DESIGN, DEVELOPMENT AND IMPLEMENTATION OF DECISION SUPPORT SYSTEMS FOR PRIVATE EQUITY INVESTMENT

A dissertation submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

TECHNOLOGY AND INFORMATION MANAGEMENT

By

Paul H.F. Vroomen

June 2017

The Dissertation of Paul H.F. Vroomen is approved:

____________________________________
Subhas Desa, chair

____________________________________
Nirvika Singh

____________________________________
Patrick Mantey

Tyrus Miller
Vice Provost and Dean of Graduate Studies
Table of Contents

LIST OF FIGURES ............................................................................................................. V

LIST OF TABLES ................................................................................................................ XII

ABSTRACT .......................................................................................................................... XIV

DEDICATION ....................................................................................................................... XVI

ACKNOWLEDGEMENTS ................................................................................................. XVII

1. INTRODUCTION .............................................................................................................. 1
   1.1. BACKGROUND ........................................................................................................... 1
   1.2. RESEARCH ISSUES ............................................................................................... 3
   1.3. RESEARCH CONTRIBUTIONS ............................................................................. 6
   1.4. ORGANIZATION OF THE THESIS ...................................................................... 9

2. PROBLEM STATEMENT .................................................................................................. 10
   2.1. THESIS PROBLEM STATEMENT ...................................................................... 10
   2.2. OVERVIEW OF THE IDSS ................................................................................ 10

3. DEFINING A SUCCESSFUL PRIVATE EQUITY INVESTMENT ..................................... 15
   3.1 INTRODUCTION ....................................................................................................... 15
   3.2 ORGANIZATION OF THE CHAPTER ................................................................... 16
   3.3 PROBLEM FORMULATION .................................................................................... 17
      3.3.1 Approach .......................................................................................................... 17
   3.4 LITERATURE SURVEY ........................................................................................... 20
      3.4.1 Modern Portfolio Theory .............................................................................. 21
      3.4.2 The Theory of Efficient Markets ................................................................... 23
      3.4.3 EBO and VC Rates of Return ....................................................................... 24
      3.4.4 Angel Investment Rates of Return ................................................................ 24
      3.4.5 ECF Rates of Return ..................................................................................... 25
   3.5 THEORETICAL FRAMEWORK ............................................................................... 25
      3.5.1 Internal Rate of Return .................................................................................. 26
      3.5.2 Modern Portfolio Theory .............................................................................. 29
      3.5.3 Efficient Market Theory ................................................................................. 31
      3.5.4 The Central Limit Theorem ........................................................................... 36
   3.6 APPLICATION ............................................................................................................ 37
      3.6.1 Data .................................................................................................................. 37
         3.6.1.1 Survivorship Bias ...................................................................................... 39
         3.6.1.2 Private Equity Buy-Out .......................................................................... 41
         3.6.1.3 Venture Capital ....................................................................................... 42
         3.6.1.4 Angel Investors ....................................................................................... 43
         3.6.1.5 Addressing Limitations of Modern Portfolio Theory ................................ 45
      3.6.2 Normative Model for Private Equity Investment ............................................ 55
         3.6.2.1 ECF Systematic Risk .............................................................................. 55
         3.6.2.2 Normative model for private equity investments ...................................... 58
   3.7 RESULTS ..................................................................................................................... 62
      3.7.1 Target IRR for Individual ECF Investments ..................................................... 62
      3.7.2 The Feasible Parameter Space for a Technology Start-Up ................................ 67
         3.7.2.1 DCF Enterprise Model Assumptions ......................................................... 69
4. CLASSIFYING PROSPECTIVE INVESTMENTS .......................................................... 75
   4.1 INTRODUCTION .......................................................................................... 75
   4.2 ORGANIZATION OF THE CHAPTER ......................................................... 77
   4.3 PROBLEM FORMULATION ....................................................................... 78
   4.4 LITERATURE SURVEY .............................................................................. 79
      4.4.1 Attribute Selection .............................................................................. 79
      4.4.2 Model Selection and Validation .......................................................... 82
   4.5 THEORETICAL FRAMEWORK .................................................................. 84
      4.5.1 Attribute Selection .............................................................................. 84
      4.5.2 Model Selection .................................................................................. 86
         4.5.2.1 The 0.632+Bootstrap Algorithm ...................................................... 86
         4.5.2.2 The Paired t-test ........................................................................... 97
         4.5.2.3 The Permutation Test (Standalone t-test) .......................................... 99
   4.6 APPLICATION ............................................................................................ 100
      4.6.1 Attribute Selection .............................................................................. 100
      4.6.2 Extracting the Training Set .................................................................. 103
      4.6.3 Model Selection .................................................................................. 106
         4.6.3.1 Definition of the Loss Function ....................................................... 110
         4.6.3.2 Preliminary Model Selection Using Leave-One-Out Cross Validation ... 112
         4.6.3.3 Final Model Selection Using 0.632+Bootstrap .................................. 113
      4.6.4 Model Assessment .............................................................................. 116
   4.7 RESULTS .................................................................................................... 119
      4.7.1 Leave-one-out cross validation results .............................................. 121
      4.7.2 0.632+Bootstrap Results .................................................................... 122
         4.7.2.1 Results of e0 bootstrap: IBk (k=1) and IBk (k=2) ......................... 124
         4.7.2.2 Results of e632+Bootstrap: J48 and IBk (k=1) ............................. 127
         4.7.2.3 Results of e632+Bootstrap: SVM and IBk (k=1) ............................ 128
         4.7.2.4 Results of e632+Bootstrap: Multi-layer Perceptron and IBk (k=1)... 129
         4.7.2.5 Results of e632+Bootstrap: Naive Bayes and IBk (k=1) ............. 130
         4.7.2.6 Results of e632+Bootstrap: Logistic and IBk (k=1) ..................... 131
      4.7.3 Results of the Permutation Test on the IBk Classifier ......................... 132
   4.8 CONCLUSIONS ......................................................................................... 133

5. CONCLUSIONS AND FUTURE WORK .............................................................. 135
   5.1 IMPACT OF THE RESEARCH AND KEY CONCLUSIONS ....................... 135
   5.2 SUMMARY OF THE THESIS .................................................................... 136
   5.3 CONTRIBUTIONS OF THE RESEARCH ................................................... 136
   5.4 FUTURE WORK ......................................................................................... 138

6. APPENDICES .................................................................................................. 142
   6.1 APPENDIX 1: ATTRIBUTE SELECTION CRITERIA AND QUANTIFICATION PROCESS ........................................................................ 142
   6.2 APPENDIX 2: THE FINAL DATA SET ......................................................... 145
   6.3 APPENDIX 3: 0.632 BOOTSTRAP R CODE: .............................................. 147
   6.4 APPENDIX 4: 0.632+BOOTSTRAP R CODE: .......................................... 151
   6.5 APPENDIX 5: IBk CLASSIFIER PERMUTATION TEST R CODE: ............. 156

7. REFERENCES .................................................................................................. 159
List of Figures

1) Figure 1: A high level illustration of the IDSS, showing the two-stage evaluation process. (p.11)

2) Figure 2: A more detailed view of the architectural framework of the IDSS showing the three databases that comprise the system and their application in each stage. (p.13)

3) Figure 3: A process for determining whether an enterprise qualifies for ECF financing by identifying the feasible parameter space for revenue growth and ROIC required for the enterprise to achieve the expected IRR of an efficient ECF portfolio. (p.18)

4) Figure 4a: Efficient frontier of a portfolio of 5 blue chip stocks optimized using Monte Carlo simulation assuming returns distribution is Gaussian. The capital market line using a risk free rate of 3.5% is also shown. (From: Karandikar, 2012) (p.47)

5) Figure 4b: Efficient frontier of a portfolio of 5 blue chip stocks optimized using Monte Carlo simulation assuming returns distribution is a students-t distribution. Note that the capital market line is steeper than in Figure 4a, resulting in optimal weights being skewed toward more conservative (i.e., low risk) stocks. (From: Karandikar, 2012). (p.48)

6) Fig 5a and 5b: Cambridge Associates EBO and VC vintage year fund IRR distributions. The distributions are Gaussian-like, but with skewness and long tails. Both have finite variances. (p.49)
7) Figure 6. Raw distribution of the AIPP angel investor IRR data plotted in equal size IRR bins with width equal to 50% and with the first bin containing all investments with IRR <=0%. The best-fit parametric distribution, a Pareto distribution with tail parameter equal to 1.1, is also shown. (p.50)

8) Fig. 7. The IRR distribution for Angel investments in portfolios. The distribution becomes more Gaussian-like and has finite variance (p.53)

9) Figure 8: The normative model for the private equity market showing the 95% confidence interval for the Angel asset class, which constrains the 95% confidence interval for the ECF asset class. The graph shows, with 95% confidence, that an efficient portfolio of ECF assets should yield at least 27% IRR, with a mean of 34% IRR, for ECF systematic risk that is 10% higher than that of the Angel asset class. (p.60)

10) Figure 9: Distribution of returns for angel investment portfolios based on AIPP data. (p.63)

11) Figure 10. Bimodal distribution for an ECF portfolio that produces a portfolio IRR of 35% and in which 55% of the basis value of the investments in the portfolio are written off (loss category percentage of portfolio is 55%). This requires the remaining 45% of performing category investments to yield 58% IRR. (p.65)

12) Figure 11: Required returns multiples for the performing investments in an efficient portfolio that yields a portfolio expected IRR of 35%. (p.67)
13) Figure 12: Revenue, cumulative cash flow (excluding investment cash inflow) and cumulative investment profile for the example company. ECF investors only participate in the seed round. (p.69)

14) Figure 13: Returns to seed investors for the company with investment, revenue and cash flow profile shown in Fig. 6, and with percentage equity ownership after the seed round as the parameter. ECF investors require at least 45% equity ownership of the company after the seed round to achieve the required IRR for ECF investments. At 35% seed equity ownership, ECF investors would not achieve the required IRR at any time. At 55% seed equity ownership the IRR surpasses the portfolio required IRR over a relatively wide window. (p.72)

15) Figure 14: Returns to seed investors with 45% seed round ownership for the company with investment, baseline revenue and cash flow profile shown in Fig. 6 and with revenue growth as the parameter. If revenue grew at 125% of baseline, IRR increases by about 10% with a wide exit window. If revenue grew at 75% of baseline, the company cannot yield the required IRR for ECF investors at any time. (p.73)

16) Figure 15: The statistical learning process for selecting the optimal classifier. (p.76)

17) Figure 16: The ratio of the standard deviation of Err_{cvk} and Err_1 (solid line) as a function of true error µ for 24 sampling experiments in (Efron, 1997) shows that Err_1 consistently has lower variance than Err_{cvk}. The
dotted line is the ratio of expectations, showing that Err$_1$ has more bias than Err$_{cvk}$. (p.91)

18) Figure 17: Relative bias of Err$_{0.632+}$ (solid curve) versus the relative bias of Err$_1$ (top dotted curve) and Err$_{0.362}$ (bottom dotted curve) for the 24 synthetic and real datasets in (Efron, 1997). Relative bias is bias relative to the true error. This plot shows that Err$_{0.632+}$ has superior relative bias to both Err$_1$ and Err$_{0.362}$. (p.96)

19) Figure 18: The 20 predictor attributes of the classifier feature vector. The target attribute is the nominal IRR range for that instance, defined as Red, Yellow or Green, depending on the numerical IRR of the instance. (p.102)

20) Figure 19: Illustration of the selection funnel for extracting the 20 attributes for each instance in the dataset. (p.104)

21) Figure 20a: Error rate of synthetically generated data for two-class classification from known class prior distributions (Gaussian, equal standard deviation, Bayes error = 0.05, attribute correlation = 0) applied to a linear kernel support vector machine, by feature size and sample size. The black dotted line shows the minimum error. (From: Hua, 2005). (p.108)

22) Figure 20b: Error rate of synthetically generated data for two-class classification from known class prior distributions (Gaussian, equal standard deviation, Bayes error = 0.05, attribute correlation = 0) applied to a k-nearest neighbors algorithm with k=3, by feature size and sample size.
size. The black dotted line shows the minimum error. (From: Hua, 2005). (p.109)

23) Figure 21: The Weka Knowledgeflow diagram used for initial evaluation of the 8 classifier algorithms selected for this analysis. The diagram illustrates the loading of a .csv file (CSVLoader), class assignment in the training and test set (ClassAssigner), application of leave-one-out cross validation for testing (CrossValidationFoldMaker), the logistic, multilayer perceptron (neural net) and support vector machine classifiers and various performance evaluation and visualization tools. (p.113)

24) Figure 22: Knowledgeflow environment illustrating the modified e0 bootstrap approach to calculating Err\textsubscript{1} for Logistic Regression, Support Vector Machine and Simple Logistic classifiers. (p.115)

25) Figure 23: Knowledgeflow environment for the permutation test. (p.116)

26) Figure 24: Pseudo code for the permutation test. The R implementation is listed in Appendix 5. (p.119)

27) Figure 25: Histogram of the loss function results for 500 bootstrap iterations of the IBk (k=1) and the IBk (k=2) algorithm. The mean values are superimposed on the histograms, showing the difference in performance between the two algorithms. (p.126)

28) Figure 26: Histogram of the loss function results for 500 bootstrap iterations of the IBk (k=1) and J48 Decision Tree algorithm. The mean
values are superimposed on the histograms, showing the difference in performance between the two algorithms. (p.127)

29) Figure 27: Histogram of the loss function results for 500 bootstrap iterations of the IBk (k=1) and SMO support vector machine algorithm. The mean values are superimposed on the histograms, showing the difference in performance between the two algorithms. (p.128)

30) Figure 28: Histogram of the loss function results for 500 bootstrap iterations of the IBk (k=1) and multilayer perceptron (neural net) algorithm. The mean values are superimposed on the histograms, showing the difference in performance between the two algorithms. (p.129)

31) Figure 29: Histogram of the loss function results for 500 bootstrap iterations of the IBk (k=1) and Naïve Bayes algorithm. The mean values are superimposed on the histograms, showing the difference in performance between the two algorithms. (p.130)

32) Figure 30: Histogram of the loss function results for 500 bootstrap iterations of the IBk (k=1) and logistic regression algorithm. The mean values are superimposed on the histograms, showing the difference in performance between the two algorithms. (p.131)

33) Figure 31: The implementation process for the IDSS, showing the sections of the process completed in this research and the remaining tasks. The remaining tasks are primarily concerned with completion of the software.
independent UML definition of the IDSS, followed by a software and hardware specific implementation of the IDSS. (p.139)
List of Tables

1) Table 1: Summary data for the three private equity asset classes. Mean IRR is the arithmetic mean of the vintage year IRR and IRR standard deviation is the arithmetic mean of the vintage year standard deviations. (p.54)

2) Table 2: Private equity enterprise stages. (p.59)

3) Table 3: The standard error of the mean and confidence intervals of each of the three existing asset classes in the private equity market based on the data in section 3.6.1. (p.61)

4) Table 4: Three examples of the description, measure and quantification process for selected predictor attributes. (p.103)

5) Table 5: Summary of classifier algorithms in the preliminary set. (p.110)

6) Table 6: Confusion matrix results for the preliminary set of classifier algorithms validated using leave-one-out cross validation, ranked in order of lowest to highest loss function value. The K nearest neighbor (KNN, with k=2) algorithm has the lowest loss function of the set, followed by KNN with k=1 and the support vector machine. (p.121)

7) Table 7: Summary of the pairwise t-test of each selected classifier relative to the k-nearest neighbors algorithm with k=1. The 95% confidence interval is positive for each tested learning algorithm, showing that the k-nearest neighbors algorithm has a lower loss function value than each of
the tested classifiers with > 95% confidence. The \( k \)-nearest neighbors algorithm is therefore the optimal classifier. (p.123)

8) Table 8: The loss functions of 200 IBk classifiers each validated using 50 iterations of the 0.632+Bootstrap algorithm on randomly permuted training attributes with the target attribute held constant. Only 11 of the 200 randomized IBk classifiers (highlighted) had a lower loss function than the original training set, resulting in a p-value of 0.055. (p.132)
DESIGN, DEVELOPMENT AND IMPLEMENTATION OF DECISION SUPPORT SYSTEMS FOR PRIVATE EQUITY INVESTMENT

Paul Vroomen

Abstract

The objective of this research is to design, develop and implement an intelligent decision support system (IDSS) for making rational private equity investment decisions. (Private equity investments are capital investments in enterprises that are not traded on public equity markets; they include Equity Buy-Out, Venture Capital, and the new Equity Crowd Funding (ECF) asset classes). The design and development of the IDSS requires the integration of investment science (valuation theory, portfolio theory, efficient market theory) and information science & technology (statistical learning, software engineering).

The IDSS architecture is built around the two primary tasks in the due diligence process for evaluating private equity investments, namely, evaluating whether a prospective investment can generate returns commensurate with its risk, and then verifying that the enterprise has the appropriate attributes or success factors to achieve that rate of return. These two tasks are implemented as stages in the IDSS: the “Evaluate” stage (Stage 1), and the “Verify” stage (Stage 2). The motivating application for this IDSS is the emerging ECF asset class, which is applied in this research to demonstrate the financial engineering and statistical learning principles and processes developed.
A fundamental investment science result of this research is the derivation of the Capital Market Line (CML) for the private equity market, based on modern portfolio theory, efficient market theory, and historical financial performance metrics. The CML enables quantification of the target Internal Rate of Return (IRR) for individual assets in the market as a function of risk. Results show that the target IRR for ECF investments places significant constraints on the operating requirements of companies seeking ECF financing.

Achieving the target IRR requirement identified in Stage 1 is a necessary but not sufficient condition for making a positive investment decision. In the Verify stage statistical learning methods are applied to verify whether a prospective investment has the necessary attributes in terms of team competencies, product value proposition, and market dynamics to achieve the target IRR. This dissertation describes the processes developed and results achieved in this research for attribute selection and extraction, model selection, and model validation, which together form the core elements of Stage 2. The results of this part of the research demonstrate that the $k$-nearest-neighbors ($k$NN) algorithm yields the lowest loss function value of the set of 8 supervised, unsupervised and meta classifiers tested.
Dedication

I dedicate this dissertation to my patient, loving and supportive wife, Kathryn, who has stood by me all these years and has unhesitatingly supported me through all my endeavors, including one of my long cherished goals with this research.
Acknowledgements

Firstly, I would like to acknowledge the persistence and dedication of my advisor, Subhas Desa who showed the patience and care to coach me through this entire doctoral research.

Additionally, I would like to acknowledge the other members of my examining committee, Professor Pat Mantey and Professor Nirvikar Singh, who both provided invaluable advice during the course of this research.

I further benefited from advice and support from many members of the faculty at UCSC and also colleagues and friends in industry and at other university campuses. In particular, I want to acknowledge the following people to thank them for their help: Prof. Brent Haddad and Prof. John Musacchio of the UCSC TIM department, who provided advice and assistance in several aspects of this research; Prof. Dan Friedman of the UCSC Economics department who reviewed most of Chapter 3 and provided valuable recommendations; Prof. David Helmbold of the UCSC Computer Science department for his advice on machine learning algorithms; Prof. Athanasios Kottas of the UCSC Applied Mathematics department who reviewed and provided feedback on the statistical methods used in this research; Prof. Jianping Hua of the University of Texas at Austin who
provided valuable recommendations on model selection and validation techniques; Prof. Mohak Shah of the University of Ottawa for providing software routines for algorithms in his book “Evaluating Learning Algorithms”; Dr. Tyler Munger, of Chobanian Consulting Group LLC, who provided valuable assistance with Ruby on Rails and R coding; Ralph Eschenbach, Mark Hawkins, Dean Witter III, Sidney Faulkner, Natasha Skok, George Pavlov and Dado Banatao, who provided access to private equity databases and also provided valuable insights on venture capital and angel investment best practices.
1. Introduction

This introductory chapter provides the motivation and objectives of this research. Section 1.1 describes the background that motivated the research. Section 1.2 outlines the research issues that were addressed. Section 1.3 describes the research contributions made by this research. Section 1.4 summarizes the organization of this dissertation.

1.1. Background

The primary engine of economic prosperity in any community is the ability of that community to create and sustain innovative enterprises. Silicon Valley is a case study in how market forces can drive enterprises to build products and services that create strong and sustained worldwide demand and even entirely new industries, resulting in economic well-being for the residents of the San Francisco Bay Area that is the envy of the world. Those enterprises all were early stage start-up companies at some point in their history, where they benefited from private equity investment. Private equity investments are capital investments in enterprises that are not traded on public equity markets; they include Equity Buy-Out, Venture Capital, Angel investments and the new Equity Crowd Funding (ECF) asset classes. The well-established
and vibrant private equity investment infrastructure in Silicon Valley is therefore a key contributor to the success of the region.

One of the greatest challenges faced by private equity investors is the selection of candidate investments that are likely to succeed. The due diligence process in private equity investment is the primary vehicle for making such selection decisions. This research is focused on applying historical data together with statistical learning and data mining principles to create an on-line intelligent decision support system to automate many of the underlying common considerations of the due diligence process in private equity investment decisions.

Private equity investment decisions are complex, requiring knowledge in diverse disciplines including finance, economics, engineering, manufacturing, marketing & sales, product development, supply chain management and human behavior. In venture capital practice, extensive due diligence is performed in each of these areas prior to an investment decision, to assess whether a candidate investment can yield the required returns. Such due diligence requires expertise, experience and judgment, but has also been shown to be inconsistent, error prone and subject to human emotion (Kuncel, 2013).
Many of these due diligence processes are based on common underlying considerations that can be codified and incorporated into an intelligent decision support system (IDSS) to assess enterprise performance and quantify the probability of achieving that performance. The primary hypothesis of this research is that application of such an IDSS in private equity investment decisions results in better financial returns than decisions made without an IDSS. To verify this hypothesis requires the design, development and implementation of such an IDSS, which is the subject of this research.

1.2. Research Issues

The following research issues are addressed in this work:

1. While much empirical research has been done to identify characteristics of a start-up company that result in success, very little research has been done to apply finance theory to the private equity market to analytically identify the returns that an efficient portfolio of private equity investments should yield. This research addresses this issue by applying efficient market theory and modern portfolio theory to develop a normative model for identifying the capital market line of the private equity market. The capital market line in turn enables estimation of the
rate of return that efficient asset portfolios in the private equity market should yield, including new asset classes with no historical performance data.

2. Introduction of new asset classes in any financial market typically results in the necessity to predict the feasible region of the value creating parameters for the new asset class. For example, a new asset class introduced to the private equity market immediately raises the question of what amount of revenue growth and return on equity, the two primary value creation parameters, will result in acceptable rates of return. This research addresses the challenge of predicting the feasible region of the value parameter space for a new asset class in the existing private equity market.

3. Identifying acceptable rates of return based on the business plan of a prospective private equity investment is a necessary, but not sufficient, basis for investing. A critical issue is determining the probability that the prospective investment has the appropriate combination of characteristics in terms of team competencies, market dynamics and product value proposition to achieve that rate of return. This research addresses this issue by applying statistical learning techniques to the
prospective investment to predict the probability of that investment achieving its pro forma business plan.

4. Related to predicting whether a prospective investment has the necessary business competencies to achieve success, is the process of identifying the attributes that are primary contributors to those competencies. This research addresses the issue of attribute definition by defining a process for identifying the primary success factor attributes.

5. Besides defining attributes, a further issue is the process by which the attributes are extracted from unstructured databases, and then quantified. This research required creation of quantification algorithms for each attribute as well as an attribute extraction process to address this issue.

6. Model selection is another challenging research issue that had to be addressed in this research. Model selection is the process of selecting, evaluating, validating and performing significance tests on statistical learning algorithms.
1.3. Research Contributions

1. The primary contribution of this research is the design, development and implementation of a two-stage architectural framework for an Intelligent Decision Support System that assesses the financial return potential of a prospective enterprise and then extracts a set of parameters that is applied to statistical learning algorithms to estimate the probability of achieving the returns. Section 2.2 provides a high level overview of the framework and chapters 3 and 4 provide a detailed description of the two primary stages of the architecture.

2. The largest databases of private equity asset class returns data available were analyzed, rationalized and applied, using the principles contained in modern portfolio theory and the theory of efficient markets, to create the capital market line (CML) of the private equity market. The CML, in turn, was then applied to extract a target rate of return of a new asset class in the market, namely the equity crowd funding (ECF) asset class. Section 3.6 describes the analysis, rationalization and application of the data and synthesizes the concepts of the theoretical framework to create the CML.

3. The target rate of return for ECF assets enabled quantification of the feasible region of the value-driving parameters (revenue growth and
return on invested capital) for prospective ECF investments. The feasible region places severe constraints on the set of companies that are likely to qualify for ECF investment. Section 3.7 describes the derivation and implications of the feasible region of the value-driving parameters.

4. An attribute selection process was created to extract the primary attributes of a prospective investment that have statistically significant correlation with success. The process required creation of quantification algorithms for each attribute. The process is based on domain knowledge, practitioner interviews and a structured review of the research literature, which also required creation of a literature review process. Sections 4.5.1 and 4.6.1 describes the attribute selection process and its application.

5. The attribute selection process was applied to extract an attribute data set from the due diligence databases of a Silicon Valley venture capital firm and an association of Angel investors. These databases were unstructured and often incomplete, requiring considerable additional research to identify and quantify attributes as well as fine-tuning of the attribute quantification algorithms. Section 4.6.2 describes the data set extraction process.
6. The attribute data set was applied as the training and test set for training selected statistical learning algorithms. A process was created and implemented for selecting the optimal set of learning algorithms. The selection process is based on a literature review of applications of the learning algorithms in unrelated domains but with similar data set size and dimensionality. The selection process also ensured algorithmic diversity. Sections 4.5.2 and 4.6.3 describe the model selection process and its application.

7. The three-class target attribute applied in this research, required definition of a loss function that appropriately takes into account the cost of misclassification. A loss function was defined using domain knowledge of the cost of each false positive and false negative in the validation confusion matrix. Section 4.6.3.1 defines the loss function.

8. Given the size and dimensionality of the attribute data set, validating the accuracy of the loss function results yielded by each learning algorithm model proved challenging. A literature review, combined with consultations with data mining faculty at UCSC, the University of Texas, Austin and the University Ottawa resulted in the identification of the 0.632+Bootstrap algorithm as the optimal validation technique for the unique characteristics of the attribute data set. This algorithm was analyzed, implemented in R and
applied to validate each statistical learning algorithm. Section 4.5.2 describes the validation process as applied to model selection.

9. The model selection and validation processes defined in this research enabled the identification of the optimal model for classifying prospective ECF investments. The optimal model, extracted from the k-nearest neighbors learning algorithm, shows that a prospective investment can be classified as green or yellow with 84% accuracy and with 94.5% confidence. These results are discussed in section 4.7.

1.4. Organization of the Thesis

The thesis is organized into two primary parts, chapters 3 and 4, in each of which the literature is reviewed, the relevant theory is described, the theory is applied and results are recorded and discussed. Chapters 3 and 4 are detailed descriptions of the two stages of the IDSS. Each of chapters 3 and 4 contains two primary contributory domains as illustrated in the diagram below:
2. Problem Statement

This chapter defines the problem addressed in this dissertation and then provides a high level overview of the Intelligent Decision Support System architectural framework.

2.1. Thesis Problem Statement

The problem that this research is focused on solving is to provide a solution to the question of how to enable a large, diverse set of non-expert investors to identify investment opportunities that are likely to succeed, from a large, diverse set of privately held companies of different size, stage and quality.

2.2. Overview of the IDSS

The IDSS comprises two stages as illustrated in Figure 1.
Stage 1: Evaluate

Can a prospective enterprise yield an acceptable rate of return?

- Yes → Stage 2: Verify
- No → Reject

Stage 2: Verify

Does the prospective enterprise have the team, market and products to credibly achieve the rate of return?

- Yes → Accept
- No → Reject/Review

Figure 1: A high level illustration of the IDSS, showing the two-stage decision support process

Stage 1 examines the financial plan submitted by the issuer to determine whether the funding requirements of the company can yield an Internal Rate of Return (IRR) commensurate with the risk of the investment. If the issuer’s plan passes the Stage 1 test, then Stage 2 examines a set of 20 predictor attributes derived from responses provided by the issuer to a set of on-line questions designed to extract each of the attributes.

Attributes relating to the team, for example, quantify the leadership experience, commitment, market knowledge, technical expertise, ability to innovate and completeness of the founding team. The IDSS then applies the specific set of parameters extracted for the issuer to a trained classifier to classify the
prospective investment into one of three categories: Green (prospective investment is likely to achieve the target IRR), Yellow (data is not definitive, review manually), Red (prospective investment is unlikely to achieve the target IRR).

A more detailed view of the architectural framework is shown in figure 2. This view shows the issuer database, the industry database and the classifier database. These three databases contain the data required for the functioning of the IDSS. The issuer database contains the responses to the attribute extraction questionnaire and the pro forma financial plan of the issuer, which includes the quarterly cash flow statement and an income statement. The industry database contains industry related data, including market size and market growth for relevant industries, gross margins, net margins, price/earnings ratios and other relevant data for successful public enterprises in each industry. The industry database is primarily sourced from publicly available data on line.
Figure 2: A more detailed view of the architectural framework of the IDSS showing the three databases that comprise the system and their application in each stage.

The classifier database contains the training and test data sets that are used to identify the optimal classifier from which the classifier model is extracted. This data is sourced from existing private equity databases for investments with known IRR. Given its proprietary nature the data is anonymized. Practitioners with domain knowledge of the private equity due diligence process, who can apply judgment regarding the quantification of attributes, are required to do the extracting of attributes from the data.

The Stage 1 algorithm calculates the IRR of a candidate enterprise based on the pro forma financial plan stored in the issuer database and compares it to a target
IRR set by the investor. The investor will be assisted by the algorithm, which will advise a target IRR commensurate with the risk of the enterprise. The primary purpose of the Stage 1 algorithm is to verify that the enterprise is capable of providing an acceptable return given the size and growth of its target market, its revenue growth in that market and its cash requirements. The algorithm also compares the enterprise financial plan to typical financial ratios and metrics for successful companies in that market stored in the industry database. This ensures that the expectations represented by the plan are within typical bounds for the industry. The algorithm also provides for sensitivity analysis to enable exploration of the effect of changes to the financial plan.

If the enterprise can yield an acceptable return it passes to the stage 2 algorithm. Here the enterprise attributes stored in the issuer database are applied to a trained classifier, which classifies the enterprise into one of three categories: Red (the enterprise is unlikely to achieve the target IRR - reject), Yellow (unclear whether the enterprise can achieve the target IRR – review manually) and Green (the enterprise is likely to achieve the target IRR – accept for further due diligence). The algorithm again allows for sensitivity analysis where attributes can be adjusted to determine what changes to the enterprise attributes will result in better classification results.
3. Defining a Successful Private Equity Investment

This chapter contains a complete description of the design, implementation and application of the Evaluate stage (Stage 1) of the IDSS.

3.1 Introduction

Title III of the JOBS Act of 2012 creates a new class of private equity investment known as equity crowd funding (ECF). In this research, modern portfolio theory and the theory of efficient markets are applied to estimate the rate of return required of ECF assets. To do this, the no-leverage capital market line of the established private equity market is derived by identifying the rates of return, and variances of these rates of return, for three private equity investment classes: equity funds, venture capital and angel investments. The systematic risk of ECF assets relative to the other private equity classes is then analyzed qualitatively to enable estimation of the variance of the ECF asset class on the capital market line and thereby enabling extraction of the commensurate rate of return for individual ECF assets.

A parameterized model for a representative ECF investment case is then analyzed to investigate the implications of this rate of return. The target rate of return, derived from the capital market line, is imposed as a constraint on this
The constrained model enables identification of the feasible region for the model parameters. The two parameters that are the primary determinants of value are studied: revenue (and consequently earnings) growth and capital structure (as represented by percentage equity ownership of ECF investors). Results illustrate the trade off between capital structure and revenue growth required to achieve the target rate of return for ECF assets. Results also show that the feasible region for revenue growth and equity ownership significantly limits the set of enterprises that can qualify for equity crowd funding.

### 3.2 Organization of the Chapter

This chapter is organized into 5 primary sections:

1) Problem Formulation
2) Literature survey
3) Theoretical framework
4) Application
5) Results
3.3 Problem Formulation

As shown in Figure 1, Stage 1, the “Evaluate” stage of the IDSS, addresses the problem of how to determine the acceptable rate of return that an ECF asset should yield.

3.3.1 Approach

A two-phase process is applied in this research for this “Evaluate” Stage:

1) Development of a generic theoretical framework from which a normative model can be created for any efficient securities market; this theoretical framework is applied to determine the normative model for the private equity market; 2) application of the private equity normative model to the ECF asset class which enables determination of the feasible parameter space for a typical technology start-up enterprise whose seed round is funded by ECF. The process, highlighting these two stages, is illustrated in Fig.3.

2) This theoretical framework is applied to create the normative model of the private equity market by compiling a data set covering the widest historical period and the largest number of individual assets available in the equity buy-out (EBO), venture capital (VC) and angel investment asset classes, the three primary existing private equity asset classes. Since the three asset class data sets
differ in terms of reported data, the data is rationalized where necessary to ensure that the data reflects the same information for each asset class.

![Diagram of ECF Investment Process]

**Figure 3:** A process for determining whether an enterprise qualifies for ECF financing by identifying the feasible parameter space for revenue growth and ROIC required for the enterprise to achieve the expected IRR of an efficient ECF portfolio.

The following 3 steps comprise Phase 2:

3) In order to apply the normative model to determine the expected IRR for an efficient portfolio of private equity assets, it is necessary to identify the
systematic risk associated with the asset class under consideration. The systematic risk of ECF investments is determined from an analysis of each contributing risk factor, since no historical data is available for ECF investment performance. This is done by considering each contributing element of systematic risk for ECF and comparing each risk element to its closest proxy, angel investments. This enables estimation of a standard deviation premium for ECF investments over angel investments, which, when applied to the normative model, enables estimation of the expected IRR required of an efficient ECF portfolio.

4) Next the known IRR distribution for angel investments and new business longevity data from the US Bureau of Labor Statistics are applied to extract the IRR required of individual investments within an efficient ECF portfolio. This provides the target IRR that individual investments in an efficient ECF portfolio must yield to achieve the expected IRR of the portfolio.

6) This target IRR is the constraint that determines feasibility for any enterprise to qualify for ECF financing. To illustrate the implications of this constraint, this target IRR is applied to a typical start-up enterprise to determine the feasible parameter space for revenue growth and ROIC for that enterprise, using a discounted cash flow (DCF) enterprise model. The enterprise model is based on the enterprise's pro forma income and cash
flow forecasts and parameterizes revenue growth and ROIC while constraining the model to achieve the required IRR. This approach enables finding the thresholds of the feasible parameter space for these two parameters in which the enterprise achieves the required IRR. It also allows determination of the degree to which revenue growth and ROIC can be traded off with one another.

3.4 Literature Survey

No performance data for the ECF asset class for non-accredited investors in the US is available yet. This means that the tools of modern finance theory, such as the Capital Asset Pricing Model (CAPM) or the Generalized Black-Scholes equation, cannot be applied to determine the required price, and therefore rate of return, of an ECF asset, since no historical variance-covariance and return rate data is available for such ECF assets.

It is therefore necessary to develop a method that enables estimation of the rate of return required of a new asset class in an existing market. Ideally, a generalized method that enables estimation of any new asset class in an existing market would be most useful as it would be applicable beyond only the private equity asset class. The method developed in this research is based on modern
portfolio theory and the theory of efficient markets and uses historical data for existing asset classes in the market to determine the required returns for any efficient portfolio of that new asset class.

The method developed here is applicable to any market where a new asset class is introduced into an existing market and is therefore applicable to the private equity market, treating ECF as a new asset class in that market. This literature survey therefore includes a survey of published research in the underlying theories applied in this research, namely modern portfolio theory and the theory of efficient markets, as well as research in the returns and performance of the primary private equity asset classes, namely, equity funds (EBO), venture capital (VC) and Angel investments.

3.4.1 Modern Portfolio Theory

Extensive use was made in this research of modern portfolio theory and specifically the portfolio optimization model developed by Markowitz (Markowitz, 1952). The Markowitz model is based on a set of 7 assumptions (Sharpe, 1964, Cont, 2001) that are very unlikely to be met under real world conditions. Nevertheless, more than 60 years after its creation the Markowitz model remains the basis of much of modern finance theory including the capital asset pricing model (CAPM), development of mutual funds and index funds,
hedging strategies, value at risk analysis, funds of funds and bond portfolio optimization, among other widely applied strategies in finance practice. The Markowitz model has also proved remarkably robust to elimination of the assumptions identified by Sharpe (Fabozzi, et al. 2002).

The Markowitz model has, however, been shown to have limitations and as a consequence several modifications have been proposed, such as the two-factor model (Fama, 2004). More recent criticism of the Markowitz model has arisen in the wake of the 2008 recession. Of primary concern is the underlying assumption in the Markowitz model that the distribution of returns is of the exponential class, such as a normal distribution. This has been highlighted in the literature since soon after the original Markowitz paper was published in 1952 (Mandelbrot 1963). Mandelbrot showed that the returns distribution of public equities is not Gaussian and that a better fit is a Levy distribution, which has no standard deviation and therefore standard deviation cannot be used as a measure of risk, as in the Markowitz approach. As a result, various derivatives of the Markowitz model have been proposed, some of which resort to numerical techniques based on the median and value at risk (Karandikar 2012, Pflug 2000). These techniques adapt the Markowitz model to student’s t distributions (Hu 2010, Demarta 2004), Pareto distributions (Kluppelberg, 2008) or normal tempered and normal mixture models (Ehrler 2011, Buckley 2008), in some cases using copulas and Monte Carlo analysis.
3.4.2 The Theory of Efficient Markets

The theory of efficient markets, in conjunction with modern portfolio theory, provides the theoretical basis for the method developed in this research to predict target returns for new asset classes in an existing market. Beechey et al (Beechey 2000) provide a succinct review of the efficient market theory, its alignment with empirical data and also its shortcomings, focusing on public equities and foreign exchange markets. The theory, formulated in the early 1960s (Sharpe 1964), has been subjected to widespread verification tests and has consistently been shown to have strong correlation with empirical evidence, primarily in public equity markets (Fama 1969).

The efficient market theory has also been shown to have limitations, primarily related to the rate at which markets react to new information (Basu 1977, Fama 2004). In this research the theory of efficient markets is applied to the private equity market with a detailed analysis of the issue of efficiency in the private equity market.
3.4.3 EBO and VC Rates of Return

An extensive literature exists on methods for measuring returns in equity buy-out funds and venture capital [Lee et al., 1999; Cochrane, 2005; Salman, 1990; Manigart, 2002; Da Rin et al, 2002; Phalippou et al, 2009; Kaplan et al., 2005; MacMillan, 1985]. Of these, Da Rin, et al., provides a comprehensive survey of the issues related to valuing private equity investments, focusing primarily on venture capital investments. Most of these studies however have been conducted through empirical surveys of investors, investor groups and entrepreneurs. This research takes an analytical approach to this problem.

The seminal paper on venture capital returns (Kaplan, et al., 2005) based on 577 VC funds, finds that value-weighted net VC fund IRR is 17% with a standard deviation of 31%. The more diverse assumptions described in the literature (such as writing off all mature investments) result in the IRR ranging from as low as public equity market returns (or slightly less – (Phalippou et al, 2009) to 21% (Kaplan, et al., 2005, using the Public Market Equivalent (PME) metric).

3.4.4 Angel Investment Rates of Return

Considerable research has also been conducted on the returns performance of angel investing (Wiltbank, 2009; Wiltbank, 2005, Wiltbank et al., 2007; Mason, et
al., 2002; Van Osnabrugge, 2009). Performance results from this angel investor research yield IRR ranging from 22% for UK based angel investors to 27% for US based investors.

3.4.5 ECF Rates of Return

No research on the returns yielded by ECF investments has yet been published. This is due to the fact that very little data is available on performance of ECF investments given that ECF investment in the US was only enabled by the JOBS Act in 2012 and SEC regulations for this type of investment were finally recorded in the Federal Register in January 2016. While many patronage crowd-funding platforms already exist in the US (Best, 2013), ECF platforms are just emerging and in the US are currently primarily aimed at accredited investors. It is likely that ECF platforms targeting non-accredited investors will start proliferating now that the SEC regulations are published. It is therefore urgent that rigorous research on the returns requirements of ECF investments is conducted to ensure that future ECF investors are appropriately informed.

3.5 Theoretical Framework

The primary performance parameter of an investment applied in this research is the internal rate of return (IRR) delivered by that investment. Modern portfolio
theory and specifically the Markowitz portfolio optimization model, combined with the principles of efficient markets, are applied to analytically determine the capital market line of the private equity market. The capital market line is the normative model for the private equity market. The theoretical framework for this analysis is derived from the following three primary theoretical elements:

### 3.5.1 Internal Rate of Return

Classical portfolio optimization is based on the return that an investment can yield, with the risk of the investment represented by the standard deviation of that return. In this context, return is defined as:

\[ R_t = \frac{p_t - p_{t-1}}{p_{t-1}} \]  \[1\]

Where:

- \( R_t \) = The return for a period, \( t \)
- \( p_t \) = The price (or value) of an investment at the end of period \( t \)
- \( p_{t-1} \) = The price of an investment at the start of period \( t \)

This approach works well in public markets, such as stock exchanges, where the period, \( t \), for an investment can be arbitrarily chosen, since a portfolio can be created at any time by purchasing securities on the stock exchange. The period varies dependent on the time horizon of the investor and typically it is relatively short: one day, one month, one quarter or one year.
The option of selecting the time horizon of an investment does not exist in private equity investment due to the fact that private equity investments are illiquid. This means that the investor cannot select the period of the investment. Rather the investor has to wait until the opportunity to liquidate an investment arises, which is an event mostly outside the control of the investor.

In private equity investment the two dominant metrics for measuring the return of an investment are Internal Rate of Return (IRR) and the return multiple. Of these, the IRR measures the compound annual return that an investment yields over any period of time. It is therefore the equivalent of \( R_i \) in equation [1] where period \( t \) is one year. The return multiple is the proceeds of an investment divided by the amount initially invested. It is independent of a time period, resulting in a 10x return multiple for a 7 year time horizon being substantially less attractive than for a 3 year time horizon. It is therefore not a meaningful metric to employ in portfolio optimization.

Consequently, throughout this research, the Internal Rate of Return (IRR), denoted by \( r \), is applied as the rate of return metric, and “return” is used as synonymous with IRR. Besides the fact that IRR normalizes investments of different time periods to an annual return, it has a number of other benefits: 1) discounted future cash flow models are widely used in practice to evaluate
private equity investments (Copeland et al., 2003) and IRR directly measures the return on cash flows; 2) IRR explicitly includes the time dependency of returns; 3) IRR is an explicit measure of return on capital invested. (Koller et al., 2010, Luenberger 2014)

Let the IRR of an investment be denoted as \( r \). In order to compute \( r \) this research uses the standard technique of setting the net present value (NPV) for a series of \( n \) quarterly cash flows equal to zero and numerically computing \( r \). This technique is represented as follows:

\[
NPV_n = \sum_{i=1}^{n} \frac{c_i}{(1+r)^{i/4}} = 0
\]  

where:

\( NPV_n = \text{net present value for } n \text{ quarters} \)

\( r = \text{Internal Rate of Return (IRR) per year, in decimal form} \)

\( c_i = \text{cash flow in quarter } i \)

\( n = \text{the quarter for which } r \text{ is being calculated} \)
3.5.2 Modern Portfolio Theory

Let $R^p$ and $\text{var}(R^p)$ denote, respectively, the rate of return and the variance of the rate of return of a portfolio of investments. The Markowitz model minimizes portfolio variance, $\text{var}(R^p)$, by adjusting the weight of each security in a portfolio under required constraints, as follows (Cochrane, 2007):

Minimize with respect to $\mathbf{w}$, $\text{var}(R^p) = \frac{1}{2} \mathbf{w}' \mathbf{V} \mathbf{w}$ \hspace{1cm} [3]

subject to the constraints:

$\mathbf{w}' \mathbf{E} = \mu; \; \mathbf{w}' \mathbf{1} = 1$ \hspace{1cm} [4]

where:

$\mathbf{w} = \text{vector of } n \text{ weights for each security in the portfolio}$

$\mathbf{E} = \text{vector of } n \text{ mean returns for each security in the portfolio}$

$\mathbf{V} = \text{the } n \times n \text{ variance-covariance matrix}$

$\mathbf{1} = \text{identity vector of length } n$

$\mu = \text{expected return of portfolio}$

This is a classic non-linear optimization problem that can be solved by finding the minimum of the Lagrangian. The solution is:
\[
\text{var}(R^p)^* = \frac{C\mu^2 - 2B\mu + A}{AC-B^2}
\]

[5]

\[
\mathbf{w}^* = \mathbf{V}^{-1} \frac{E(C\mu - B) + 1(A - B\mu)}{(AC-B^2)}
\]

[6]

where:

\( \text{var}(R^p)^* \) = minimum variance of portfolio IRR for expected return of \( \mu \)

\( \mathbf{w}^* \) = optimum vector of weights for expected return of \( \mu \)

\( A \triangleq \mathbf{E}'\mathbf{V}^{-1}\mathbf{E} \);

\( B \triangleq \mathbf{E}'\mathbf{V}^{-1}\mathbf{1} \);

\( C \triangleq \mathbf{1}'\mathbf{V}^{-1}\mathbf{1} \)

Equation [5] illustrates the quadratic relationship that exists between minimum variance (portfolio risk) and the expected return of the portfolio. The upper half of this quadratic is defined as the efficient frontier for a portfolio of assets. The efficient frontier explicitly defines the relative weight that should be assigned to each asset in a portfolio to achieve a desired return with minimum risk, satisfying the non-satiation behavior of rational investors.
3.5.3 Efficient Market Theory

An efficient market, i.e., one in which all available information is fully reflected in asset prices (Fama 1969), can be represented as follows:

\[ E(P_{j,t+1} | \phi_t) = (1 + E(R_{j,t+1} | \phi_t))p_{j,t} \]  \[ \text{[7]} \]

where upper case letters are used to denote random variables, and lower case letters are used to denote actual values, \( E \) is the expected value operator, and:

\( t = \text{current time} \)

\( P_{j,t+1} = \text{price of asset } j \text{ at future time } t+1 \)

\( R_{j,t+1} = \text{return of asset } j \text{ at future time } t+1 \)

\( \phi_t = \text{the set of available information at current time } t \text{ that is fully reflected in price } p_{j,t} \)

\( p_{j,t} = \text{price of asset } j \text{ at current time } t \)

Such efficient markets also show the following characteristics:

if \( z_{j,t+1} \) is the excess market value of asset \( j \),

\[ \text{then } z_{j,t+1} = r_{j,t+1} - E(R_{j,t+1} | \phi_t) \]  \[ \text{[8]} \]
In an efficient market,

\[ E(Z_{j,t+1} | \phi_t) = 0 \]  \[9\]

If the market trades in \( n \) assets based on \( \phi_t \) and if \( w_j(\phi_t) \) is the capitalization weight of each asset, then the total excess market value at time \( t+1, v_{t+1} \), generated by this market is:

\[ v_{t+1} = \sum_{j=1}^{n} w_j(\phi_t) z_{j,t+1} \]  \[10\]

In an efficient market, the expectation of this total excess market value, \( E(V_{t+1}) \), is:

\[ E(V_{t+1}) = \sum_{j=1}^{n} w_j(\phi_t) E(Z_{j,t+1} | \phi_t) = 0 \]  \[11\]

The \( w_j(\phi_t) \) in equation [11] can be written as a function of current price as follows:

\[ w_j(\phi_t) = s_j p_{j,t} \]  \[12\]

Where:

\( s_j = \) total number of shares outstanding of security \( j \), and which can be assumed to remain unchanged from \( t \) to \( t+1 \)
Solving equation [7] for $E(R_{j,t+1} | \phi_t)$ and inserting this into equation [10] yields:

$$v_{t+1} = \sum_{j=1}^{n} w_j(\phi_t)\left[ r_{j,t+1} - \left( E(P_{j,t+1} | \phi_t) / p_{j,t} \right) - 1 \right]$$  \[13\]

Inserting equation [12] into equation [13] and recognizing that $r_{j,t+1} = \frac{p_{j,t+1}}{p_{j,t}} - 1$,

we can write equation[11] as:

$$E(V_{t+1}) = E[\sum_{j=1}^{n} s_j (p_{j,t+1} - p_{j,t}) - \sum_{j=1}^{n} s_j (E(P_{j,t+1} | \phi_t) - p_{j,t})]$$  \[14\]

Since $E(V_{t+1}) = 0$, this simplifies to:

$$\sum_{j=1}^{n} s_j (p_{j,t+1}) = \sum_{j=1}^{n} s_j (E(P_{j,t+1} | \phi_t))$$  \[15\]

Equation [15] states that the value of a market trading in $n$ securities at time $t+1$
is the sum of the expected market capitalization weight of each security in the
market at time $t+1$, given by $\sum_{j=1}^{n} s_j (E(P_{j,t+1} | \phi_t))$.

In an efficient market it can be assumed that some participants are applying the
Markowitz portfolio optimization calculation to determine the optimum
portfolio weights. This will cause the market capitalization weight, $w_j(\phi_t)$ in
equation [11], of each security traded on the market to converge to its Markowitz optimum weight through the buying and selling mechanism. Since the entire efficient market can be considered a portfolio comprised of each security in the market, the capitalization weight of each security in the market will converge to its Markowitz optimum portfolio weight (Luenberger, 2014). This means that the elements of vector \( \mathbf{w}^* \) in Equation [6] are equivalent to the \( \mathbf{w}_j(\phi_t) \) of Equation [11] in an efficient market.

Much debate exists in the literature as to whether the private equity market is efficient. The debate centers on whether the net asset value (NAV— the book value of an asset) reflects the true market value of a private equity asset (DaRin, 2011, Phallipou, 2009). Most private equity practitioners generally accept that the private equity market is efficient based on the following reasoning:

At the time of investment, \( t_i \), and of exit, \( t_e \), of a private equity security, the open buying and selling mechanism operates. Furthermore, enterprises that have received private equity financing are subject to scrutiny by their investors; the investors are subject to scrutiny by their partners and the partners are subject to scrutiny by their limited partners (in the case of venture capital and private equity buy-out firms). For example, any tendency by the general partners of a venture capital fund to under-state or over-state the NAV of investments would be questioned by the limited partners (and would likely jeopardize the general
partners’ ability to secure investment in later funds from those limited partners). This continual multi-level scrutiny tends to drive NAVs and therefore capitalization weights to their efficient market optimum during the time between $t_I$ and $t_E$ as well.

This research therefore assumes that the private equity market is efficient. Under that assumption, the mean and standard deviation of the market portfolio lies on the efficient frontier of the market. (The efficient frontier of the market is also known as the capital market line (CML)).

Furthermore, a portfolio of securities in that market will asymptotically approach the CML as the size of the portfolio approaches the market population. The Central Limit Theorem enables calculation of confidence intervals with which the mean of a sample taken from a distribution approaches the mean of the population. The CML defines the mean of the market portfolio, and therefore the population, as a function of risk.

As will be shown in section 3.6.1, the data for the existing asset classes in the private equity market that is applied to calculate the private equity CML in this research, represents a significant subset of the private equity market. Consequently, each can be regarded as a portfolio in the private equity market that is large enough to approach the CML within small and well-defined
confidence intervals. Each of the existing asset classes represents a different risk – return trade off and therefore their means and standard deviations will lie within the confidence interval at different points on the CML. These portfolios therefore provide multiple points on the CML within known confidence intervals, from which the CML can be constructed.

3.5.4 The Central Limit Theorem

The Central Limit Theorem is applied in this research to calculate confidence intervals for the Capital Market Line (CML). The Central Limit Theorem states that the distribution of the means, $\bar{x}_i$, $i=1,2,\ldots,n$, of a set of $n$ large samples taken from a population will be approximately normally distributed regardless of the distribution of the population (so the $\bar{x}_i$, are instantiations of the random variable, $\bar{X}$, which is approximately normally distributed). The theorem further requires that each variate in each sample is independent and identically distributed (iid) and each sample has both a finite mean and finite variance. The Central Limit theorem states that the standard error of the means is:

$$SE_{means} = \frac{SE_{sample}}{\sqrt{n}}$$

[16]

Where:

$n = \text{sample size}$

$$SE_{sample} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

[17]
And:

\[ x_i \] = the value of instance \( i \) of the sample set

\[ \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \]  \[18\]

3.6 Application

In this section the theory and prior research identified in the literature review is synthesized and applied to the largest known databases of historical returns performance for the existing private equity asset classes.

3.6.1 Data

The data applied in this research is sourced from Cambridge Associates, LLC (Cambridge Associates, June 30, 2014) and the Angel Investor Performance Project (AIPP), available from the Kauffman Foundation (Ewing Marion Kaufmann Foundation, 2013; Wiltbank et al, 2007).

The VC and EBO fund performance data tracked by Cambridge Associates is widely consulted by practitioners in the private equity market. The Cambridge Associates IRR by vintage year data is calculated by applying the net aggregated cash flows to limited partners (i.e., after deduction of all expenses and carry percentages) for all funds established in each vintage year. Where an individual investment in a fund portfolio has not exited, NAV reported by the fund is used...
to determine the value of the investment. The standard deviation for each vintage year is the standard deviation of the IRR for all funds established in the vintage year.

In a similar fashion to equation [2], Cambridge Associates calculates IRR for each fund, $j$, established in a vintage year, $y$, by solving for $IRR_j$ in the following equation:

$$NPV_j = \sum_{i=1}^{N} \frac{C_{ij}}{(1+IRR_j)^{i/4}} = 0$$

where:

$NPV_j =$ Net Present Value of fund $j$, which is set to 0 to calculate $IRR_j$

$j = 1, 2, 3...M; M =$ total number of funds established in vintage year $y$

$i = 1, 2, 3...N; N =$ the number of quarters that fund $j$ exists

$C_{ij} =$ the net aggregate cash flows to or from limited partners in quarter $i$ for fund $j; C_{Nj}$ is the net asset value (NAV) of the fund at the end of 2012

The IRR for vintage year $y$, $IRR_y$, is then calculated as follows:

$$IRR_y = \frac{1}{M} \sum_{j=1}^{M} IRR_j$$

Where
\[ M = \text{number of funds established during vintage year } y. \]

Similarly, the standard deviation, \( \sigma_y \), for each vintage year is calculated as follows:

\[
\sigma_y = \sqrt{\frac{1}{M-1} \sum_{j=1}^{M} (IRR_j - IRR_y)^2}
\]  

[21]

### 3.6.1.1 Survivorship Bias

Private equity funds and angel investors are not legally required to report the performance of the companies in which they have invested. Reporting is therefore voluntary, which raises the concern that failed investments are under-reported, skewing the reported performance to yield higher expected returns than actually occurred. This phenomenon is referred to as survivorship bias. Both the Cambridge Associates data and the AIPP data have been tested for survivorship bias with results showing that survivorship bias is not materially skewing the data, as described below.

Cambridge Associates requires its VC and EBO clients to report the performance of their entire fund, including non-performing investments in the fund.
Survivorship bias would then only occur if a fund ceased reporting. Cambridge Associates has found that on average in each year over the most recent six years only 0.8% of funds in their database stopped reporting, representing an average of 0.6% of the NAV of the database in each respective year. Over the same period the number of funds in the database has grown at an average of 8% per year. Additionally the small number of funds that ceased reporting has been spread over all performance quartiles and not concentrated in the lower performing quartiles as would be expected if the cessation in reporting were driven by poor performance. Survivorship bias is therefore minimal in the Cambridge Associates data.

The AIPP data was sourced from 81 angel investor groups throughout the US representing a response rate of 31% of the groups solicited. Individual response rates were 13%, with some groups showing a high response and others a low response rate. If survivorship bias was present in the data, then the low response rate groups would likely correlate to the poorer performing outcomes and the high response rate groups to better performing outcomes. The data is however uncorrelated to the response rate of the groups with seven high response rate groups (groups in which at least 2 out of 3 members respond) showing a return multiple of 2.6X, while the low response rate groups had a return multiple of 2.4X. Also the median multiple for the high response rate group was 1.2X while the low response rate group median multiple was 1.4X.
The conclusion from this discussion is that survivorship bias in the AIPP data is also negligible.

### 3.6.1.2. Private Equity Buy-Out

Private equity buy-out funds typically invest in opportunities where established enterprises are looking to spin off sections of their companies as a result of changing corporate strategy, or where an entire enterprise is seeking financing to implement a new strategy (Da Rin et al., 2011). These funds tend to be large, of the order of several billion dollars, and their focus on established enterprises with revenue history, results in their investments typically being of lower risk than the earlier stage investments made by venture capital funds, with commensurate lower returns (Cambridge Associates, June 2014). It is very unlikely that ECF investors would participate in the investments made by equity buy-out funds as the scale of these investments is such that the $1M maximum annual investment permitted by the JOBS Act for non-accredited ECF investors would not be material enough to justify raising ECF investment.

Nevertheless, equity buy-out funds constitute a key asset class of the private equity investment market and as such provide a necessary data point on the capital market line of the private equity market. Based on the Cambridge Associates data for 1,112 equity buy-out funds over the period 1986 to 2012, the
mean IRR for EBO is 14.7% and the mean standard deviation is 16.4%. In this research rounding to the nearest 0.5% is applied, yielding an IRR of 14.5% and a standard deviation of 16.5% for the EBO market.

The Morningstar financial data tracking service identified a total of 1,956 equity funds that were active in 2014. A typical equity buy-out fund has a term of 15 years. For the 15 year period ending in 2012, the Cambridge Associates data contains 904 funds. The Cambridge Associates data therefore contains about 46% of the entire EBO market.

### 3.6.1.3. Venture Capital

Venture capital funds typically invest in privately held late, early and seed stage companies that have significant growth potential (MacMillan, 1985). ECF investors will also be seeking such early stage high growth opportunities, although the scale of ECF investments vs. VC investments would differ given the restrictions on capital amounts defined by the JOBS Act.

Based on the Cambridge Associates vintage year net IRR data for 1,474 venture capital funds over the period 1981 to 2013, the mean VC IRR is 16.3% and the mean standard deviation is 25.2%. In this research I use 16.5% IRR with a standard deviation of 25% for the VC market. This is equivalent to a 3x return.
multiple over 7 years, which is consistent with return expectations for venture capital funds in practice.

The National Venture Capital Association (NVCA) reports 1,333 active venture capital funds in 2013. A typical venture capital fund term is 10 years. The Cambridge Associates data contains 556 funds for the 10 year period ending in 2012. This data therefore represents about 42% of the entire venture capital market.

3.6.1.4. Angel Investors

A closer proxy for the performance of private equity investments in early stage companies is that of Angel investors. Angel investors are typically wealthy individuals, with substantial business operating experience, who invest in early stage companies in their fields of expertise (Mason, 2002; Riding, 2000). The risk profile for Angel investors closely resembles the likely profile for ECF investors in that their transactions are conducted directly with the issuing company and usually at very early stages such as the seed round (Wiltbank, 2005).

Angel investors are also a diverse group of individuals, ranging from formally organized groups of accredited investors who share due diligence activities and pool their expertise in the decision making process, to “friends and family” of
entrepreneurs who invest small amounts (from $1,000s to $10,000s), and whose expectations on returns are not necessarily driven by IRR metrics. An assumption of this research is that ECF investors will be driven by typical investment considerations and therefore will expect a rate of return that is commensurate with the risk inherent in the asset class. Therefore the Angel investor data applied in this research is restricted to accredited Angel investors working in formal groups and investing with the expectation of acceptable rate of return.

The Angel market is informal and no formal reporting requirements exist. Various studies of this market have been performed (Wiltbank, October 2005), but the available data is much sparser and less structured than that available for the EBO and VC markets.

The Ewing Marion Kauffman Foundation has conducted the Angel Investor Performance Project (AIPP) (Ewing Marion Kauffman Foundation). The database is publicly available and contains returns data for 3,097 US angel investments with 1,137 exits, sourced from associations created by angel investors to provide better insight into the industry's performance and also to lobby on behalf of the industry. Data reported through such associations is subject to the same issues as the VC data and should therefore receive the same degree of caution. The
study nevertheless provides the most comprehensive set of data on this asset class to date.

3.6.1.5 Addressing Limitations of Modern Portfolio Theory

A fundamental assumption of modern portfolio theory is that the risk of a portfolio is measured by the covariance matrix of the IRR delivered by each investment in a portfolio, as expressed in equation [3]. Clearly, this means that the IRR distribution of the securities in a portfolio must have finite variance. This assumption is not met if the distribution of the IRR is of a form that does not have finite variance, such as a Levy distribution (which has infinite mean and variance) or a generalized Pareto distribution (which has infinite variance if its tail index lies in the interval \((0,0.5]\)).

Benoit Mandelbrot (Mandelbrot, 1963; Mandelbrot, 2004) raised this issue within a few years of Markowitz's publication of his seminal paper (Markowitz, 1952), showing that the returns in public equity and commodities markets have distributions for which a Levy distribution is the best fit. It is now widely accepted that returns distributions of public equities have kurtosis (“fat tails”) and heteroscedasticity (“skewness”) that cannot be modeled using symmetric distributions such as the normal distribution (Hu, 2010; Buckley, 2008)
Various techniques have been developed in the literature to deal with this problem, mostly focused on applying risk metrics other than variance, the most popular of which is “value at risk” (VAR) which is widely applied in financial risk evaluation and regulation (Kim, 2012). VAR is a specific quantile, such as the lowest 5% of a distribution, and is therefore an integral of the distribution, resulting in a quadratic relationship with the IRR, as is the case for variance. Researchers have applied numerical optimization techniques to non-Gaussian, fat-tailed, skew distributions to determine the efficient frontier of portfolios sampled from such distributions (Karandikar, 2012). The key conclusion from such analyses is that an assumption that returns are normally distributed, as is the case for the Markowitz model, results in a more aggressive portfolio than would be the case if numerical techniques are applied to optimize portfolios from non-Gaussian distributions. Karandikar, et al, however, show that such techniques result in only a 6% improvement in the Sharpe ratio (the parameter that is maximized) for a Monte Carlo technique that assumes a students-t distribution relative to a Gaussian distribution for a portfolio of 5 blue chip stocks (see Figure 4). This implies that use of distributions with finite variance to approximate “badly behaved” distributions that have no variance, still provides a close approximation of the optimal portfolio weights.
Figure 4a: Efficient frontier of a portfolio of 5 blue chip stocks optimized using Monte Carlo simulation assuming returns distribution is Gaussian. The capital market line using a risk free rate of 3.5% is also shown. (From: Karandikar, 2012)
Figure 4b: Efficient frontier of a portfolio of 5 blue chip stocks optimized using Monte Carlo simulation assuming returns distribution is a students-t distribution. Note that the capital market line is steeper than in Figure 4a, resulting in optimal weights being skewed toward more conservative (i.e., low risk) stocks. (From: Karandikar, 2012).

The investment options for private equity investors are: EBO funds, VC funds and individual Angel or ECF funded enterprises. Figures 5a and 5b illustrate the IRR distribution for the Cambridge Associates EBO and VC fund data. The funds are effectively portfolios whose IRR distributions have finite variance, so the problem of infinite variance does not exist for these two asset classes.
Fig 5a and 5b: Cambridge Associates EBO and VC vintage year fund IRR distributions. The distributions are Gaussian-like, but with skewness and long tails. Both have finite variances.

However, in the case of Angel investments, where investments are made in individual enterprises, the problem of infinite variance manifests itself explicitly. As shown in figure 6, the best-fit distribution of IRR to the AIPP data is a power law (Pareto) distribution. Figure 6 shows the best-fit generalized Pareto distribution (GPD) overlaid on the AIPP IRR data and validated using a bootstrap test algorithm resulting in a p-value of 0.008, or less than 1%, implying >99% confidence in the GPD fit to the AIPP IRR data. The best-fit GPD parameters are: shape = 0.3939795, scale=0.5029750 (Villasenor, 2009). The shape parameter is less than 0.5 and therefore variance is infinite, which means that the Markowitz approach of minimizing risk for a given return by minimizing variance will not provide meaningful results.
Figure 6. Raw distribution of the AIPP angel investor IRR data with the first bin containing all investments with IRR <=0% (assuming that investors are concerned only with the distribution of positive rates of return). The best-fit parametric distribution, a generalized Pareto (power law) distribution is overlaid on the histogram. The best-fit generalized Pareto distribution parameters are: shape = 0.3939795, scale=0.5029750, location = 0.

Actual variance of angel investments will of course be finite, since infinite variance, in a distribution with a finite mean, implies the possibility of infinite rates of return. An infinite rate of return is impossible since it would require infinite resources, which do not exist. Rates of return can, nevertheless, be extremely high. The AIPP data, for example, contains an investment of $10,000 that yielded $2.4M in 1 year, or
a 239X return in 1 year (an IRR of 23,900%). While extremes are possible, the maximum return that an investment could yield is always finite.

An estimate of this maximum can be determined from an analysis of the returns that the world’s fastest growing companies have yielded. The fastest growing company (as measured by company valuation) in the history of the internet is GroupOn, Inc., which was valued at $1.35B just 17 months after the company was started (Steiner, 2010). A hypothetical seed investor that purchased $1M shares in GroupOn for 45% equity ownership on the day it was founded would have had a return of ~600X in 17 months, a rate of return of ~9,000% (assuming that the investment round that set valuation at $1.35B resulted in the new investor owning only 30% of the company’s equity). A reasonable cap on the rate of return would therefore be 1,000X in 12 months, a rate of return of 100,000%, requiring that an investment of $10,000 yields $100M in 1 year (note that to sustain an IRR of 100,000% would require that in 17 months the investment needs to yield $10,000*(1,001)^1.417 or >$1.8B, more than the total valuation of the world’s fastest growing web commerce company in the same timeframe). Assuming such a large upper bound on a Pareto distribution, however, still results in finite variance. Note also that the lower bound of a private equity investment is an IRR of -100%, a total write off of the amount invested. Private equity investments are therefore bounded.

The approach taken in this research is to constrain variance further by limiting the Angel IRR data to only investors in the AIPP data that invest in portfolios of 2 or
more enterprises. These portfolio returns are then converted to vintage year returns as was done for the Cambridge Associates data. The resultant vintage year portfolio variance is then applied as a proxy for the Angel investor asset class variance. Use of portfolio variance is justified by the fact that a total of 41% of exited investments in the AIPP database were portfolio investments. Strong evidence also exists that successful Angel investors typically invest in portfolios (Shane, 2009).

To calculate vintage portfolio returns required the following process:

1) Only exited investments are extracted from the AIPP data (as non-exited investments may or may not have remaining value and NAV for non-exited investments was not recorded in the AIPP data).

2) Then the mean IRR and IRR standard deviation was calculated for individual investors that have portfolios of two or more investments. This yields $IRR_j$ for each portfolio. The AIPP data provided a total of 88 such portfolios, containing 462 exited investments.

3) Then the vintage year for each angel portfolio was extracted, using the earliest initial investment year in the portfolio, and $IRR_y$ and $\sigma_y$ for each vintage year that had 3 or more portfolios was calculated.

The resultant distribution of Angel portfolio investments is shown in Fig. 7.
Fig. 7. The IRR distribution for Angel investments in portfolios. The distribution becomes more Gaussian-like and has finite variance. 

Fig. 7 shows that the IRR distribution of portfolios of Angel investments is more Gaussian-like and has finite variance. Also, by the Central Limit Theorem, the mean IRR of these portfolios will converge to the mean IRR of the individual AIPP exited investments. The resultant mean IRR for the Angel market is 30.6% and the portfolio IRR standard deviation is 65%. The resultant data for each relevant vintage year for all three private equity markets is summarized in table 1.
Table 1: Summary data for the three private equity asset classes. Mean IRR is the arithmetic mean of the vintage year IRR and IRR standard deviation is the arithmetic mean of the vintage year standard deviations.
3.6.2 Normative Model for Private Equity Investment

The previous sections provide the necessary framework to create the normative model for the private equity market. The application of the model will be demonstrated by estimating the expected returns for the new ECF asset class. But before this can be done, the risk of the ECF asset class has to be estimated. Section 3.6.2.1 discusses the underlying risk factors affecting the ECF asset class and applies these factors to estimate the systematic risk of the ECF asset class. This is then applied to the normative model in section 3.6.2.2 to determine the consequent expected return for the ECF asset class.

3.6.2.1 ECF Systematic Risk

ECF investments will have inherently higher systematic risk than that of the other private equity classes. This increased risk is due to the following factors:

1) **Limited Liquidity:** Private equity investments are not liquid and therefore proceeds cannot be extracted whenever desired, as is the case for public equities. This is a primary contributor to the systematic risk of private equity investments (Macmillan et al., 1985; Manigart, et al., 2002). It is unclear at this time whether a secondary market will develop for ECF investments. The JOBS Act only requires that ECF investments are not transferable for a period of 1 year after investment and is silent as to whether these assets may then
be publicly sold. Nevertheless, such secondary sales will still be subject to the constraints of the JOBS Act (such as the total dollar amount that investors may invest in ECF securities per year). Also, most venture capital investment terms include buy back options for the company whereby the company has first option to purchase any shares that a venture capital investor wishes to sell, at the face value of the shares. If, as is likely, ECF investment terms include such a buy back option, then the secondary market will mostly only apply to investments that are being sold at below their face value. It can therefore be concluded that the ECF asset class will have the same liquidity risk as the other private equity classes.

2) **Reduced Control:** VC investors, and to a lesser extent, angel investors, tend to have well-structured terms and conditions to establish control over their investments, including Board seats, controlling voting rights, information rights, anti-dilution preferences and liquidation preferences. ([Van Osnabrugge, 2008; Sahlmann, 1990; Berkery, 2007, Da Rin, et al., 2011](#)). While such protective provisions can be included in the terms of ECF investments, the effect of a large group of small investors making that investment is that each individual investor will very likely own a small minority share. Voting on the ECF shares will therefore be based on the aggregate views of all investors, resulting in reduced control for individual ECF investors relative to VC and angel investors.

3) **Information Asymmetry:** Issuers of private equity securities are not required to publicly disclose material developments, as is legally mandated for public
companies (Salman, 1990; Da Rin, et al., 2011). This reduced disclosure can result in rapid changes in the net asset value of private equity investments when material information does become available to investors. Given the less active role and reduced visibility that ECF investors would have relative to VC investors and angel investors, this risk is higher in ECF investments.

4) **Legal Constraints:** The JOBS Act constrains the amount ECF issuers can raise to a maximum of $1M per year and the amount that each ECF investor can invest annually to $10,000. This limits the ability of ECF investors to participate in later stage financing rounds both for reasons of scale (the ECF investor has reached the limit of permitted investment amount and cannot participate) and timing (the ECF issuer has reached the limit of permitted ECF funding in a particular year and therefore needs to find capital from other sources than ECF) (Bradford, 2012; Best et al., 2013). The timing limit would result in ECF funded companies seeking additional funding from traditional institutional investors, diluting the ECF investors and thereby reducing their potential returns upon a liquidation event. The scale limit would result in ECF investors that participated in earlier rounds foregoing participation in a new round, resulting in uncontrollable dilution.

5) **Moral Hazard:** Risk of moral hazard increases with reduced oversight and limited due diligence (Seidt, 2013). This risk is greater for ECF investments relative to venture and angel investors given the reduced visibility and control of individual ECF investors.
The conclusion from this discussion is that ECF investors will require a premium over the systematic risk faced by buy-out, venture capital and angel investor private equity asset classes.

3.6.2.2 Normative model for private equity investments

The necessary inputs are now available to apply the process outlined in section 3.3.1 to estimate the expected returns required of ECF investments. Firstly it is necessary to recognize that the EBO, VC and angel markets are portfolios comprised of four primary enterprise stages (each of which can be considered as a portfolio of individual investments of enterprises at that stage), as shown in table 2. These four enterprise stages can be regarded as portfolios that together comprise the entire private equities market. ECF investors will be making investments from this same set of enterprise stages, but with different weights for securities in each enterprise stage compared to other private equity investment classes, due to the higher risk associated with ECF investments as described in section 3.6.2.1. The EBO, VC and angel asset classes can therefore be regarded as consisting of weighted portfolios of securities each of which belong to one of these four enterprise stages.
<table>
<thead>
<tr>
<th>STAGE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seed</td>
<td>Newly formed enterprises with no revenue and no product, but a viable business idea that requires product and market validation, usually by building a prototype with which market demand can be assessed</td>
</tr>
<tr>
<td>Early</td>
<td>Enterprises that have verified their product value proposition by confirming appropriate functionality of prototypes and have validated revenue potential with confirmed customer traction, but need to convert the prototype to a commercial product</td>
</tr>
<tr>
<td>Late</td>
<td>Enterprises that are in production and shipping to customers, but not yet profitable and need working capital to grow revenue to achieve positive cash flow.</td>
</tr>
<tr>
<td>Buy-out</td>
<td>Established privately held enterprises or sub-sets of publicly held enterprises whose management is seeking acquisition due to changing business strategy.</td>
</tr>
</tbody>
</table>

Table 2: Private equity enterprise stages.

The mean IRR and IRR standard deviation data points we have identified in section 3.6.1 for the EBO, VC and Angel asset classes lie on the efficient frontier of the private equity market within well-defined confidence intervals. To calculate the exact efficient frontier of the private equity market would require the covariance matrix, $V$, and vector of means, $E$, for the four constituent enterprise stage portfolios of the private equity market to determine the constants A, B and C in equation [5]. However, since the EBO, VC and Angel data points are known to lie on the efficient frontier within known confidence intervals, a quadratic of the form of equation [5] can be fitted to these data points by adjusting the constants A, B and C such that the result is a quadratic regression of the data points. This enables explicit determination of $var(R_p)$ as a function of $\mu$ using equation [5], within the confidence intervals, without requiring $V$ or $E$.

The private equity capital market line can then be constructed by adding a risk free security to the mix of private equity constituent securities (Luenberger, 2014). The
interest rate on 10 year Treasury Bonds is used for this risk free security (at the time of writing, the rate is 2.18%). A further assumption is that borrowing is not applicable in the private equity markets (i.e. no negative weights are allowed for equation [6]) since private equity investors rarely leverage their investments. Figure 8 illustrates the resultant normative model.

**Figure 8**: The normative model for the private equity market showing the 95% confidence interval for the Angel asset class, which constrains the 95% confidence interval for the ECF asset class. The graph shows, with 95% confidence, that an efficient portfolio of ECF assets should yield at least 27% IRR, with a mean of 34% IRR, for ECF systematic risk that is 10% higher than that of the Angel asset class.

It is now possible to estimate the expected IRR required for the ECF market by applying an ECF premium over the angel investor standard deviation, as also shown...
in Fig. 8. For example, if a 10% premium for ECF investments is applied over angel investments, due to the higher systematic risk of ECF investments as described in section 3.6.2.1, then an efficient portfolio of ECF investments should yield a mean IRR of about 34%.

Applying equation [16] of the Central Limit Theorem to the data of section 3.6.1 shows that the 95% confidence interval is constrained by the +/- 5.29% of the Angel asset class – see table 3. Since we know the distribution of the AIPP data from which portfolios are sampled, random samples can be taken from that distribution to synthetically create as many angel investor portfolios as desired. However the sampled portfolios cannot exceed the confidence interval of the sample population (in this case the AIPP data) from which the samples are taken. The parameter \( n \) in equation [16] is therefore the AIPP population (i.e. \( n=462 \)).

<table>
<thead>
<tr>
<th></th>
<th>EBO</th>
<th>VC</th>
<th>Angel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Error of Mean</td>
<td>0.49</td>
<td>0.66</td>
<td>2.70</td>
</tr>
<tr>
<td>99% confidence interval</td>
<td>+/- 1.27</td>
<td>+/- 1.69</td>
<td>+/- 6.96</td>
</tr>
<tr>
<td>95% confidence interval</td>
<td>+/- 0.97</td>
<td>+/- 1.29</td>
<td>+/- 5.29</td>
</tr>
</tbody>
</table>

Table 3: The standard error of the mean and confidence intervals of each of the three existing asset classes in the private equity market based on the data in section 3.6.1, and their corresponding confidence intervals.

The 95% confidence interval for the target ECF IRR is therefore +/- 6.5%. The relatively modest 10% standard deviation premium over Angel investments is
justified by the close similarity of the systematic risk profile between ECF and Angel investments.

**3.7 Results**

In this section the implications of the target ECF IRR identified in section 3.6 are investigated. Firstly the asset class IRR is translated into a target IRR for individual investments in section 3.7.1. Then in section 3.7.2 this individual investment IRR target is applied as a constraint to identify the revenue growth and return on invested capital (ROIC) requirements for a projected typical ECF funded enterprise.

**3.7.1 Target IRR for Individual ECF Investments**

The expected IRR range identified for ECF in section 3.6.2.2 is the rate of return for an efficient portfolio comprising the entire ECF asset class. It is therefore necessary to translate this market return into a required return for individual ECF investments, since this is the constraint for the analysis of required revenue growth and ROIC for an ECF funded enterprise. Empirical data shows that VC and angel investments are distributed over a range of returns with the distribution heavily skewed to the low end of the returns spectrum (Berkery, 2007; Wiltbank, 2005; Wiltbank, 2007). The distribution for the 1,137 exited angel investments in the AIPP data is summarized in Fig. 9, illustrating that 46.7% of the exited investments were
written off (IRR = -100%) and 61.8% of angel investments had less than 0% IRR, meaning that investors did not recover their invested funds.

![Figure 9: Distribution of returns for angel investment portfolios based on AIPP data](image)

Angel investors are the closest established proxy for ECF investors given their similar investment process, limited post investment control and poor post investment visibility (although in all of these three factors angel investors are better positioned than ECF investors). The distribution for ECF investors is therefore likely to be more skewed to the low end than that of angel investors. Further, data from the Bureau of Labor Statistics (BLS) (*US Bureau of Labor Statistics, 2012, Seidt, 2013*) shows that 65% to 75% of early stage companies are shut down within 10 years of their founding. If 65% to 75% of a portfolio is written off, the portfolio expected IRR would have to be produced by the remaining 35% to 25% of the portfolio.
Given the lack of historical rate of return data to determine an actual distribution for the ECF asset class, the distribution of returns in an ECF portfolio is approximated by applying a bimodal distribution to ECF portfolios comprising two categories: enterprises that are written off and therefore yield -100% IRR (loss category), and performing enterprises that are required to achieve a sufficient average rate of return to yield the portfolio expected IRR (performing category). Based on the rate of return distribution for angel investments as well as the BLS data described above, and to compensate for the lack of ECF data, a relatively wide range for the loss category is assumed in the following analysis, from 50% to 75% of an efficient ECF portfolio, which inversely means that the portfolio expected IRR must be produced by the remaining 50% to 25% of the portfolio. The average rate of return of the performing category is defined to be the required IRR for individual ECF investments. Rational ECF investors will therefore seek out ECF investments that can credibly yield at least this required IRR to ensure that their portfolios are efficient and therefore yield the expected ECF portfolio IRR. Such a bimodal distribution is illustrated in Fig. 10.
Figure 10. Bimodal distribution for an ECF portfolio that produces a portfolio IRR of 35% and in which 55% of the basis value of the investments in the portfolio are written off (loss category percentage of portfolio is 55%). This requires the remaining 45% of performing category investments to yield 58% IRR.

The return multiple required of the performing category defined above is given by:

\[ M_i = \frac{(1+r)^i}{p} \]  \[14\]

where:

\[ M_i = \text{return multiple required in year } i \]
\[ i = \text{number of years from investment} \]
\[ r = \text{portfolio internal rate of return in decimal form} \]
\[ p = \text{the performing category percentage of the portfolio in decimal form} \]
Figure 11 illustrates the returns multiples (defined as net proceeds/capital invested) for the performing portion of the portfolio required for the overall portfolio IRR to reach a target of 35%. From this chart it can be seen, for example, that a portfolio of which 45% of investments are in the performing category (with the other 55% of investments written off, yielding -100% IRR), but which yields a portfolio IRR of 35%, requires the performing 45% of the portfolio to yield multiples of 10X the initial investment amount over a period of 5 years or 5X over a period of 3 years.

**Figure 11: Required returns multiples for the performing investments in an efficient portfolio that yields a portfolio expected IRR of 35%**
As a consequence, if an ECF investor builds an ECF portfolio by making investments with the maximum $10,000 permitted under the JOBS Act in a particular year, and $5,500 of those investments are written off because 55% of the investments fail, Fig 5 implies that the remaining $4,500 of investments are required to yield $45,000 (10X) in 5 years, or $22,500 (5X) in 3 years for the overall portfolio IRR to reach 35%. A 10X multiple in 5 years translates to an IRR from the productive portion of the portfolio of 58%, as illustrated in Fig.10. This analysis means that successful ECF investments (i.e., investments that can yield the portfolio expected IRR) must have credible potential to generate growth in enterprise value that can achieve the multiples shown in Figure 10 for a given investment horizon and performing category portfolio percentage.

3.7.2 The Feasible Parameter Space for a Technology Start-Up

It is now possible to investigate the effect on ROIC and revenue growth of the target IRR derived in section 3.7.1 for an ECF funded enterprise. Given the restricted amount that issuers can raise through ECF funding, a very likely application for ECF investment is that it provides the seed investment for a technology start-up which secures venture capital funding for its subsequent financing needs after the seed round. Seed investment, even for complex technology companies such as semiconductor and biotechnology companies, typically is below $1M and is applied primarily to create a prototype or proof of concept for the core business idea. This
investigation is based on a typical venture capital funded technology start-up where ECF investors participate in the seed round only.

The company's pro forma revenue, cash flow and equity investment profile is shown in Fig. 12 (this is an average profile for successful venture funded semiconductor companies). In this analysis it is assumed that ECF investors do not participate in the two following venture capital rounds, since the investment amounts required are well beyond the limits imposed by the JOBS Act.

Since the rate of return for the example in Fig. 12 is fixed, and a fixed seed round investment amount is assumed, the percentage of the enterprise's equity acquired in the seed investment varies the ROIC parameter (i.e., invested capital is varied by varying the percentage equity acquired for the seed amount). The ROIC parameter is therefore implemented by varying the capital structure of the enterprise (the phrase “capital structure” is used in the private equity sense, meaning the composition of equity ownership, as recorded in the capitalization table of the enterprise). The objective therefore is to determine what percentage of the company's equity ECF investors must acquire for their $750,000 seed round investment to ensure that they will achieve the ECF target returns. The effect of larger and smaller revenue growth than the baseline case shown in Fig. 12 is also examined.
Figure 12: Revenue, cumulative cash flow (excluding investment cash inflow) and cumulative investment profile for the example company. ECF investors only participate in the seed round.

3.7.2.1 DCF Enterprise Model Assumptions

The timing and size of the investments made in each round and the timing and ramp of revenue are shown in Fig. 12. The seed round investment was $0.75M. Post money valuations for the two venture capital investment rounds in this company were $15M after the A round investment of $8M, and $37.5M after the B round investment of $15M. We assume a 5-year investment horizon and that this enterprise is an investment in an ECF portfolio of which 45% is estimated to be in the performing category (defined in section 3.7.1), meaning that the enterprise target IRR is 58%. The following assumptions are also made in this example:
1) **ECF equity ownership:** The seed round investment amount is $0.75M, all of which is assume to be provided by ECF investors. The equity ownership acquired by ECF investors for this amount is varied to determine the capital structure threshold at which the company yields the ECF target return given the revenue profile in Fig. 12. This is equivalent to varying the company’s valuation to determine the capital structure that can yield the required IRR, which is common practice in private equity valuation analysis.

2) **Company Value (Price)/Earnings Ratio:** An average of public company comparable price/earnings ratios for successful semiconductor companies is applied to the example enterprise’s discounted future cash flow to calculate its value in each quarter. In this analysis a P/E ratio of 12 is used.

3) **Future Cash Flow Period:** The value of the company for each quarter is calculated by multiplying the price/earnings ratio with an annualized moving average of future forecast cash flow. In this research a twelve-month moving average of future cash flow is applied, and then company valuation is determined by multiplying this by the price/earnings ratio. No discount is therefore applied to the first 12 months, but cash flow beyond 12 months into the future is entirely discounted. This is done because longer-term revenue forecasts (more than one year) would typically be heavily discounted by a prospective acquirer given the uncertainty of forecasting long-term revenues for an early stage technology company in the formative stages of its revenue ramp.
4) **Liquidation preference:** This parameter sets the amount that is first distributed to VC investors upon a liquidation event such as acquisition or IPO. VC investors will typically hold preferred shares that have such liquidation preference as a protective provision. In this example a 1x liquidation preference is assumed, meaning that VC investors will first receive distributions equal to 1x the amount that they invested before other investors (including the seed round ECF investors) receive any distributions.

5) **Stock Option Pool:** The size of the stock option pool is the percentage of shares that the company retains for incentivizing employees in the form of stock options. In this analysis it is assumed that the stock option pool is reset to 10% of the company's post money valuation at each investment round, which is common practice in venture capital investment.

Given the data in Fig. 12 and the set of assumptions above, the distributions that the ECF investors are likely to receive can be calculated for each quarter in which company value is higher than the cumulative investment in the company (i.e., for each quarter that IRR is positive) if the company were to exit in that quarter. The percentage equity ownership received by ECF investors can then be varied to locate the boundary of the feasible region required to achieve the target IRR. Similarly, the capital structure can be held fixed while varying revenue growth to identify the feasible region boundary for revenue growth.
3.7.2.2 Results

The resultant IRR as a function of time from investment for the ECF seed investors is shown in Fig. 13 with equity ownership as a parameter. Despite dilution, the ECF seed investors can still receive returns that achieve the target IRR of 58%, provided that they acquire at least 45% of the company's equity in the seed round. Note also that this company enjoyed vigorous revenue growth averaging more than $1M per quarter in the first 3 years of revenue shipments. The effect of revenue growth on the IRR achieved by the company is illustrated in Figure 14.

Figure 13: Returns to seed investors for the company with investment, revenue and cash flow profile shown in Fig. 12, and with percentage equity ownership after the seed round as the parameter. ECF investors require at least 45% equity ownership of the company after the seed round to achieve the target IRR for ECF investments. At 35%
seed equity ownership, ECF investors would not achieve the required IRR at any time. 
At 55% seed equity ownership the IRR surpasses the target IRR over a relatively wide window.

Figure 14: Returns to seed investors with 45% seed round ownership for the company with investment, baseline revenue and cash flow profile shown in Fig. 12 and with revenue growth as the parameter. If revenue grew at 1.25X baseline, IRR increases by about 10% with a wide exit window. If revenue grew at 0.75X of baseline, the company cannot yield the target IRR for ECF investors at any time.

3.8 Conclusions

A framework has been developed based on modern portfolio theory and the theory of efficient markets to derive the capital market line for the private equity market. An
important component of deriving the capital market line was the application of large databases of historical IRR performance data for several private equity asset classes. This enabled estimation of the expected returns for an efficient portfolio of a new asset class in this market. By way of example, the private equity capital market line was applied to determine the expected IRR required of the equity crowd funding (ECF) asset class. The ECF asset class was found to have an expected IRR of greater than 27%, with 95% confidence, if the risk of the ECF asset class is assumed to be 10% higher than the Angel investment asset class. This has significant implications for enterprises seeking financing through ECF. Using an example where ECF financing provides the seed round for a technology company that undergoes two subsequent venture capital financed rounds, the revenue growth rate for the company was shown to have to exceed $1M per quarter during the first three years of revenue shipments to achieve an IRR of 58% which yields a portfolio return of 35% when 45% of the portfolio is productive. Additionally, the enterprise would have to agree to sell 45% of its equity to the seed round ECF investors. The sensitivity of the feasible parameter space for the two primary value determinants (revenue growth and ROIC) was also determined, showing that variations in revenue growth rate and variations in the equity percentage acquired in the seed round materially affect the ability of the enterprise to generate acceptable returns.
4. Classifying Prospective Investments

This chapter describes the detailed design, implementation and application of statistical learning in the Verify stage (Stage 2) of the IDSS.

4.1 Introduction

If an enterprise’s financial plan shows that it can achieve the required returns commensurate with the investment’s risk, as determined in Stage 1, that is a necessary but not sufficient condition to make a positive investment decision. If the enterprise passes Stage 1 of the IDSS, it is then necessary to determine whether the mix of competencies and assets of the enterprise can credibly achieve that financial plan. This takes place in Stage 2 of the IDSS which employs statistical learning algorithms, trained on data from existing private equity enterprises that have known outcomes, to classify a prospective enterprise’s probability of achieving its business plan.

Figure 15 illustrates the statistical learning process applied in this research to identify the optimal classifier algorithm for Stage 2 of the IDSS. The process consists of four stages each of which is described in detail in the sections that follow.
Firstly, the predictor attributes on which predictions of the target attribute will be made are defined. This includes the process of quantifying those attributes. Secondly, those attributes that are correlated with others are removed to minimize the dimensionality of the dataset while maximizing its information content. Once the attributes are defined, they are extracted from the due diligence databases of private equity investors and institutions to which access was obtained. This is the training and test set (collectively, the dataset) which will enable selection of the optimal predictor model. Finally, the dataset is applied to a set of statistical learning algorithms (classifiers) to identify the optimal predictor model.

Figure 15: The statistical learning process for selecting the optimal classifier.
The predictor model can then be applied to classify prospective investments into one of the three target attribute classes: green (prospective investment is likely to succeed), yellow (prospective investment has elements that may lead to success, but needs further scrutiny), red (prospective investment is not likely to succeed). Green investments can proceed to more detailed due diligence and the process of conformance to the requirements of the JOBS Act can be initiated (e.g., background checks on the principals and physical address verification of the enterprise). Yellow investments would require scrutiny and sensitivity analysis to determine which attributes need to be improved and to explore whether those improvements can be credibly achieved. Red investments can be rejected.

4.2 Organization of the Chapter

This chapter is organized into 5 primary sections:

1) Problem Formulation
2) Literature survey
3) Theoretical framework
4) Application
5) Results
4.3 Problem Formulation

This chapter addresses several problems that are central to the challenge of applying statistical learning to predict the likelihood that a prospective investment will succeed. This chapter addresses the following problems:

1) Identification of the enterprise attributes that are positively correlated with IRR to compile a target set of attributes that maximize information content while minimizing dimensionality. A process for determining such attributes was defined and implemented.

2) Extraction and quantification of the target attributes from existing private equity databases. A process was defined and implemented to ensure consistency of the quantification process for each instance in the dataset.

3) Selection of learning algorithms that are optimally suited to the characteristics of the final dataset. A process was defined and implemented for selecting the learning algorithms that are optimally suited to small datasets of high dimensionality, and also maximized diversity in classification technique.

4) Identification and implementation of the optimal validation techniques for the selected algorithms. The characteristics of the dataset necessitated the use of bootstrap techniques rather than the more prevalent cross validation. The optimal bootstrap technique was identified and implemented.

5) Identification and implementation of the optimal assessment technique for the selected algorithm. Both pairwise and regular t-tests were implemented to determine statistical significance of the selected model’s results.
4.4 Literature Survey

The classification process consists of 3 primary phases: attribute selection, model selection and model validation. A literature survey was therefore conducted for each of these phases.

4.4.1 Attribute Selection

An extensive body of published research exists that attempts to identify factors that contribute to success for early stage privately held start-up enterprises. The reason that such extensive research exists is that creating successful companies is the primary engine for job and wealth creation in an economy, which leads to high profile government, industry and academic focus on these success factors.

This rich body of research is a valuable source for identifying the predictor attributes for Stage 2 of the IDSS. The following process was applied to identify success factors from the literature:

a) Only papers that identified quantitative factors and showed statistically significant correlation between those factors and successful outcomes were consulted.

b) The papers were ranked by citation rate (citations per year).

c) The factors identified in the highest ranked paper were extracted, followed by the second highest, then third highest, and so on.
d) Where factors were the same or similar to factors already on the extracted list, they were not added to the list.

e) The process was stopped when no new factors were identified in five consecutive next ranked papers.

f) Factors that were identified in two or more papers were then retained and factors only mentioned in one paper were discarded.

One of the more interesting studies, and the paper with the highest citation rate identified in this research, was written by Song et al (Song, 2008) who completed a meta study of 31 papers, all of which contained studies of factors with statistically significant correlation to success, identifying 8 factors that have high success correlation across all 31 papers.

One of the earliest studies of success factors identified by practitioners in the venture capital industry, and also one of the most widely cited papers, is the survey of factors applied by the general partners of 83 venture capital firms performed by MacMillan et al (MacMillan, 1985). This paper found 10 factors, 6 related to the team, 2 each to product and market, that were identified by practitioners as primary contributors to success and became widely accepted as the primary due diligence criteria in the venture capital industry.

A second study by Macmillan et al (Macmillan, 1987), also widely cited, reviewed the criteria applied by general partners at 67 venture capital firms in the case of one
failed investment and one successful investment made by each firm, yielding an analysis of nearly 150 enterprises. Participants in the study ranked each investment on 25 screening criteria, using several metrics for outcome, and the authors then applied regression, cluster analysis, factor analysis and t-tests to identify correlations between the screening criteria and successful or unsuccessful outcomes. The paper identified three factors related to the team and two factors related to the market that were strongly correlated to success and which the venture capital industry has come to view as “table stakes” (non-negotiable requirement) for any prospective investment.

Several of the more widely cited papers contained case studies of individual early stage start-ups, mostly technology companies, in various countries including India (Kakati, 2003), Israel (Chorev, 2006), the European Union (Murray, 1996) and the US (Stuart, 1987). In each case the authors identified the factors that resulted in success using regression techniques.

A unique approach was taken by Khan (Khan 1987) in which he attempted to assess the influence of factors that are difficult to quantify, such as the influence of environment and judgment, on the success or failure of start-up ventures. He developed an environment based conjunctive model from a survey of 36 VC firms and 104 investments made by those firms. His analysis found two highly significant factors, both related to the team.
4.4.2 Model Selection and Validation

The process of model selection is primarily concerned with quantitatively assessing how accurately a model, created from a subset of the population, represents that population. The process of determining the accuracy with which a model represents the population is known as validation, and is the basis for selection of an optimal model. In most cases, the distribution of the entire population is unknown, and therefore this accuracy has to be inferred from the subset that is available. Model selection is a critical process in the field of statistical learning and consequently a large body of published research exists here too.

A critical factor in the research described in this dissertation is that the dataset used for model selection is small, defined as \( n \leq 50 \), where \( n \) is the number of instances in the dataset. The reason for this is that the dataset has to be extracted from institutional due diligence files that vary substantially in their information content and structure, often requiring research into other sources of data, such as LinkedIn or other websites to determine the employment history of members of a founding team, for example. This need to maximize validation accuracy for small datasets required a survey of research literature that specifically deals with model selection for small datasets. Fortunately, validation of small datasets has been widely studied, as this is also relevant to model selection in the biological sciences where, in proteomic profiling studies, for example, datasets are also small and often have high dimensionality, in some cases containing more than 15,000 attributes in datasets as small as \( n \leq 100 \) instances (Molinaro, 2005).
The implications, conclusions and recommendations of several key papers were directly applied in this research. Efron and Tibshirani have done valuable and extensive research into validating small datasets, and their seminal paper on the 0.632+Bootstrap validation technique (Efron, 1997) provided the basis for the validation process used in this research. Hua et al (Hua 2005) performed an extensive set of simulations to identify the optimal combination of attribute dimensionality and number of instances for a range of statistical learning algorithms, both with and without correlation among predictor attributes. They found, for example, that accuracy increases with predictor attribute dimensionality for a support vector machine with datasets as small as \(n=10\), provided the attributes are uncorrelated. In the research described in this dissertation, therefore, extensive effort was made to remove any correlation between the predictor attributes. The primary author, Jianping Hua, also kindly provided online access to the entire results database of 560 3-dimensional graphical images illustrating the result of each simulation.

Valuable insights were also gained from the text “Evaluating Learning Algorithms” (Japkowicz, 2011), which contained extensive R code examples of various validation techniques, including stratified k-fold cross validation, various bootstrapping algorithms, including 0.632 Bootstrap, permutation tests and paired t-tests to determine statistical significance of performance differences between various
learning algorithms. The co-author, Mohak Shah, kindly provided access to the entire code set in .docx form.

The learning algorithms applied in this research were all sourced from Weka (Eibe, 2016), accessed through R by invoking the RWeka library. The original papers in which each of the algorithms used in this research were published are summarized in table 5.

4.5 Theoretical Framework

The Stage 2 theoretical framework has two primary domains: 1) attribute selection, and 2) model selection, validation and assessment. Section 4.5.1 addresses attribute selection and section 4.5.2 deals with model selection, validation and assessment.

4.5.1 Attribute Selection

The dataset available for this research is limited in terms of dataset size, \( n \) (\( n=17 \) in the final dataset), but each instance in the dataset has high dimensionality, \( d \) (\( d=21 \) in the final dataset, 20 predictor attributes and a target attribute). It is therefore important in the process of attribute selection to understand the optimal characteristics of the attributes that yield best performance in terms of minimizing the error rate produced by a classifier trained on the dataset. Two theoretical considerations apply to identify these characteristics. The first is the identification of
the effect of covariance among the attributes and the second is the identification of
the minimum error rate of classifiers as a function of both dataset size and
dimensionality to identify the classifiers that are likely to be optimal for the final
dataset used this research.

Intuitively one would expect that error rate should decline as the number of
attributes increases. This is indeed the case when the joint attribute-class
distributions are known. But for sample data, which is the typical application domain
of statistical learning algorithms, a peaking effect has been demonstrated (Hughes,
1968) where the error rate declines as the number of attributes increases up to a
point, after which the error rate begins to increase. This peaking effect is more
critical for small dataset sizes, as is the case in this research. Knowing which
classifiers, and under which attribute characteristics this peaking effect occurs is
therefore a key consideration for both attribute selection and model selection in this
research.

Hua et al. (Hua, 2005) performed a thorough simulation study to identify the optimal
combination of attribute dimensionality and dataset size for various classifiers. Their
analysis was based on simulations using synthetic data, with known class conditional
distributions, applied to several classifiers with feature sizes varying from 2 to 30
and dataset sizes varying from 10 to 200. Knowledge of the prior distribution
enabled extraction of the true error rate for each classifier and each combination of
feature size and dataset size. The computational complexity of this exercise required the use of contemporary massively parallel computing techniques.

4.5.2 Model Selection

A primary challenge in model selection is validation of the model being evaluated. The literature review and recommendations from experts in the field all pointed to the 0.632+Bootstrap algorithm as the optimum validation technique for small, high dimensionality data sets, as is the case in this research. This model selection section therefore starts with a detailed review of the 0.632+Bootstrap validation algorithm in section 4.5.2.1. This is followed with a review of the paired t-test for comparative performance evaluation between two algorithms in section 4.5.2.2. The section concludes in sub-section 4.5.2.3 with a review of the standalone t-test, which is the primary method applied in this research for model assessment.

4.5.2.1 The 0.632+Bootstrap Algorithm

The original derivation of the 0.632+Bootstrap algorithm was presented by Efron and Tibshirani (Efron 1997). The algorithm yields the lowest combination of bias and variance among the primary validation methods (cross validation, bootstrap, 0.632 bootstrap), and is especially suited to small data sets. This is because bootstrapping has the effect of multiplying the size, \( n \), of a dataset by roughly 1.6, due to the lower variance resulting from the bootstrap process compared to cross-validation (the
most commonly used validation method). The rest of this section describes the
theory of the 0.632+Bootstrap algorithm by developing the theory in steps starting
from the resubstitution error and building from there to leave-one out cross-
validation, leave-one-out bootstrapping, 0.632 bootstrap and ultimately the
0.632+Bootstrap algorithm.

Let $x = (x_1, x_2, x_3, ... x_n)$ be a dataset consisting of $n$ instances $x_i = (t_i, y_i)$ where $t_i$ is
the predictor vector containing $k$ predictor attributes and $y_i$ is the target attribute.
Let the distribution of the population be $F$. Then we define a model as a function of $x$
that applies a prediction rule, $r$, using only the predictor vectors, $t$, to predict the
target attribute of a new instance whose target attribute is unknown. This model can
be written: $r_x(t)$. The true error rate of $r_x(t)$ is defined as:

$$Err_t = E_{0F}(Q(y_0, r_x(t_0)))$$  \[14\]

Where:

$E_{0F}$ is the Expectation operator indicating that the test sample, $x_0$, is an instantiation
of the random variable, $X_0$, with distribution, $F$

$x_0 = (t_0, y_0)$ is a test instance, randomly sampled from $F$, which is not in the dataset
$x$

$$Q(y, r) = \begin{cases} 
0 & \text{if } y_0 = r_x(t_0) \\
1 & \text{if } y_0 \neq r_x(t_0)
\end{cases}$$  \[15\]

Since $X_0$ is a random variable, $Q(y_0, r_x(t_0))$ is also a random variable
The true error rate is, strictly, the conditional error rate, conditional on the training set, $x$. The distribution $F$ is unknown in the vast majority of practical applications, which are typically constrained to a sample from $F$. Validation methods for predictor rules therefore use error rate estimators based on such a sample. Consequently, identifying the optimal model is the process of identifying the model whose validation result most closely approximates $Err_t$.

While $F$ is unknown, the empirical distribution, $\hat{F}$, of a dataset, $x$, assumes a probability of $\frac{1}{n}$ for each instance in a dataset. $Err_t$ can then be estimated by substituting $\hat{F}$ for $F$. Such an estimate is called the resubstitution error rate:

$$Err_{resub} = E_{0\hat{F}}(Q(y_0, r_x(t_0))) = \frac{1}{n} \sum_{i=1}^{n} (Q(y_i, r_x(t_i)))$$

[16]

The random samples, $X_0$, now are taken from the distribution $\hat{F}$. The resubstitution error rate tends to be biased downward (i.e., results in a smaller error rate than the true error rate) because the same instances are used for the test set and the training set. This bias can be corrected by applying cross validation. K-fold cross validation divides the dataset into $n/k$ unique sets and removes each successive set of $k$ instances, which become the test set, with the remaining instances comprising the training set. The error rate for each $n/k$ training and test set is then averaged. The k-fold cross validation error rate is:
\[ Err_{cvk} = E_{0f}(Q(y_0, r_{x(k)}(t)) = \frac{1}{n} \sum_{i}^{n/k} Q(y_i, r_{x(k)}(t)) \]  \[ \quad \text{[17]} \]

Where:

\( r_{x(k)}(t) \) is trained on a training set that excludes the \( k \) instances now used in the test set. When \( k=1 \), this is known as leave-one-out cross validation. \( Err_{cvk} \) is a discontinuous function. When \( n \) is small, this discontinuity in \( Err_{cvk} \) can be especially pronounced, resulting in high variance. Bootstrap smoothing is a way of reducing this variance by averaging.

Bootstrap smoothing is the process of resampling the dataset such that the attributes of the resampled instances have the same marginal distribution as the original dataset. Let \( x^* \) be an independent, identically distributed (iid) random sample of \( m \) samples from \( \hat{F} \), each sample of size \( n \), such that \( x_1^*, x_2^*, x_3^*, ..., x_m^* \sim \hat{F} \). Then the \( x_i^* \) are bootstrapped samples from the dataset \( x \) with distribution \( \hat{F} \). Bootstrapping results in smoothing of the variance of \( Err_{cvk} \) because \( m \) can be much larger than \( n \), resulting in less discontinuity in the distribution of the variance.

To compare \( Err_{cv1} \), the leave-one-out cross validation error, to the error of bootstrapping, it is necessary to define the error of the leave-one-out bootstrap. The error rate of leave-one-out bootstrap is defined as follows:

\[ Err_1 = \frac{1}{n} \sum_{i}^{n} E_{F(i)}(Q(y_i, r_{x(i)}(t^*_i))) \]  \[ \quad \text{[18]} \]
Where:

\( x_{(i)} \) is the dataset with instance \( i \) removed

\( r_{x_{(i)}}(t_{(i)}^\ast) \) is the result of the prediction rule on a bootstrapped sample of size \( n \) taken from \( x_{(i)} \)

\( F_{(i)} \) is the empirical distribution \( F \), with instance \( i \) removed and which therefore has a population of \( n-1 \) instances.

And

\[
E_{F_{(i)}} \left( Q \left(y_i, r_{x_{(i)}}(t_{(i)}^\ast) \right) \right) = \frac{1}{m} \sum_{j=1}^{m} Q \left(y_i, r_{x_{(i)}}(t_{j(i)}^\ast) \right)
\]  \hspace{1cm} [19]

Where:

\( m \) is the number of bootstrap samples

\( t_{j(i)}^\ast \) is the predictor vector for bootstrap sample \( j \) from \( x_{(i)} \)

Simulations on 24 synthetic and real datasets (synthetic datasets have known population distributions, \( F \)) in (Efron, 1997) enable comparison of \( Err_{cv1} \) with \( Err_1 \). Their results show that the variance in \( Err_1 \) is consistently lower than that of \( Err_{cv1} \), but the bias of \( Err_1 \) is consistently higher than that of \( Err_{cv1} \). This is illustrated in Figure 16, taken from (Efron, 1997), showing that the median ratio of the variances of \( Err_1 \) and \( Err_{cv1} \) is 0.79 while the median ratio of expectations (which is equivalent to bias) is 1.07.
Figure 16: The ratio of the standard deviation of $\text{Err}_{cv}$ and $\text{Err}_1$ (solid line) as a function of true error $\mu$ for 24 sampling experiments in (Efron, 1997) shows that $\text{Err}_1$ consistently has lower variance than $\text{Err}_{cv}$. The dotted line is the ratio of expectations, showing that $\text{Err}_1$ has more bias than $\text{Err}_{cv}$.

The 0.632 Bootstrap algorithm is an attempt to optimize the balance between variance and bias. The algorithm recognizes that, since the $m$ bootstrap samples are taken from the data set $x_{(i)}$ with replacement and are distributed as $\mathcal{F}_{(i)}$, which is a uniform distribution, then the probability of a particular instance not being chosen is $1 - \frac{1}{m}$. This means that the probability of this particular instance not being chosen after $m$ samples is $(1 - \frac{1}{m})^m$. Hence the resultant number of distinct examples in a sample of $m$ instances is $\left(1 - \left(1 - \frac{1}{m}\right)^m\right) \approx 0.632m$. This means that a classifier
trained on the $m$ bootstrap samples is only trained on 63.2% of the data in the original dataset, resulting in downward bias (i.e., accuracy appears to be better than it actually is).

The 0.632 Bootstrap algorithm attempts to offset this downward bias by balancing it with the upward bias of the resubstitution error, $Err_{resub}$, as follows:

$$Err_{0.362} = 0.632 Err_1 + 0.368 Err_{resub}$$  \[20\]

Efron et al \cite{Efron, 1983} show that this algorithm is more accurate than all other validation methods. But the algorithm was not tested on classifiers that tend to highly overfit to the training data, such as the k nearest neighbors (KNN) algorithm, which fits perfectly to the training data, resulting in $Err_{resub} = 0$. Under these conditions the 0.632 Bootstrap algorithm is itself biased downward. This can be easily seen in the case of a distribution in which the target attribute, $y$, is independent of the predictor attributes $t_i$. In this case, the error rate is known as the no-information error rate. For example, in a dichotomous application (where the target attribute can take on only one of two classes, usually 1 and 0) with target attribute independent of the predictor attributes and where 50% of the instances have a target attribute of class 0 and the other 50% have a target attribute of class 1, the error rate is 0.5. Both $Err_1$ and $Err_{cvk}$ yield errors of 0.5 for such a dataset. But in the case of the KNN algorithm, $Err_{resub} = 0$ and the 0.632 Bootstrap algorithm yields
an error of $0.632*0.5=0.316$, which is clearly biased downward. Addressing this issue was the motivation for the development of the 0.632+Bootstrap algorithm.

The 0.632+Bootstrap algorithm places more weight on $Err_1$ in situations where the amount of overfitting is high. Overfitting is measured by $Err_1 - Err_{resub}$ and normalized using the no-information error rate. The no-information error rate can be estimated using Monte Carlo techniques by permuting the predictor attributes while holding the target attribute constant. (This is similar to the t-test simulation of the null hypothesis described in section 4.5.2.2). An estimate of the no-information error rate, $\hat{\rho}$, can be obtained as follows:

$$\hat{\rho} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} Q(y_i, r_x(t_j))$$  \[21\]

Note that $\hat{\rho}$ is independent of $r_x(t_j)$ (i.e., as an example, $\hat{\rho}$ has the same value when $r$ is a support vector machine as when $r$ is a k-nearest neighbor rule). It is dependent only on the distribution of the classes in the dataset, $x$. This is expected since, if the predictor vector is independent of the target attribute (which is the definition of “no-information”), then no information exists for a rule to outperform any other rule.

The relative overfitting rate is defined as:

$$\hat{R} = \frac{Err_1 - Err_{resub}}{\hat{\rho} - Err_{resub}}$$  \[22\]
\( \hat{R} \) varies from 0, when no overfitting occurs \((Err_1 = Err_{resub})\) to 1 when overfitting equals the no information error rate \((Err_1 = \hat{\gamma})\). Using an argument based on the Euclidean distance of test points to the training points, Efron et al \((Efron, 1997)\) show that a less biased version of \(Err_{0.632} \) can then be written as follows:

\[
Err_{0.632+} = (1 - \hat{\omega})Err_{resub} + \hat{\omega}Err_1
\]  \[23\]

Where:

\[
\hat{\omega} = \frac{0.632}{1 - 0.368\hat{R}}
\]  \[24\]

\( \hat{\omega} \) varies from 1, when \( \hat{R} = 1 \), to 0.632, when \( \hat{R} = 0 \), so that \(Err_{0.632+}\) varies from \(Err_{0.362}\) to \(Err_1\) depending on the relative overfitting rate, \( \hat{R} \). However, it can happen that \(Err_1 \geq \hat{\gamma} > Err_{resub}\) or \( \hat{\gamma} \leq Err_{resub}\) in which case \( \hat{R} \) can fall outside \([0,1]\). To correct for this, \( \hat{R} \) and \(Err_1\) are modified as follows:

\[
Err_1' = \min (Err_1, \hat{\gamma})
\]  \[25\]

\[
\hat{R}' = \begin{cases} \frac{Err_1 - Err_{resub}}{\hat{\gamma} - Err_{resub}} & \text{if } Err_1, \hat{\gamma} > Err_{resub} \\ 0 & \text{otherwise} \end{cases}
\]  \[26\]

This now finally allows definition of the 0.632+Bootstrap algorithm. Note that equation \[23\] can be rewritten as follows:
\[ \text{Err}_{0.632^+} = \text{Err}_{0.632} + (\text{Err}_1 - \text{Err}_{\text{resub}}) \cdot \frac{0.368 \cdot 0.632 \cdot \hat{R}}{1 - 0.368 \cdot \hat{R}} \]  \hspace{1cm} [27]

Substituting \( \text{Err}_1 \) and \( \hat{R}' \) in equation [27] yields the final form of the 0.632+Bootstrap algorithm:

\[ \text{Err}_{0.632^+} = \text{Err}_{0.632} + (\text{Err}_1' - \text{Err}_{\text{resub}}) \cdot \frac{0.368 \cdot 0.632 \cdot \hat{R}'}{1 - 0.368 \cdot \hat{R}'} \]  \hspace{1cm} [28]

Equation [28] is the form of the 0.632+Bootstrap algorithm that was used for validation, pairwise comparison and t-test assessment in the model selection process for this research.

Figure 17, taken from (Efron, 1997) compares the relative bias of \( \text{Err}_{0.632^+} \) to \( \text{Err}_1 \) and \( \text{Err}_{0.632} \), illustrating the superior bias performance of \( \text{Err}_{0.632^+} \).
Figure 17: Relative bias of $Err_{0.632^+}$ (solid curve) versus the relative bias of $Err_1$ (top dotted curve) and $Err_{0.362}$ (bottom dotted curve) for the 24 synthetic and real datasets in (Efron, 1997). Relative bias is bias relative to the true error. This plot shows that $Err_{0.632^+}$ has superior relative bias to both $Err_1$ and $Err_{0.362}$.

Calculating $Err_{0.632^+}$ is computationally intensive, since it relies on a bootstrapped sample of size $m$ evaluated $n$ times to calculate $Err_1$ and $Err_{0.362}$. However, Efron et al also show that $m>50$ results in less than 10% improvement in the standard deviation of $Err_1$ calculated from the bootstrap. This implies that $m$ can be reasonably limited to 50.
Stage 2 of the IDSS uses a three-class dataset (target attributes red, yellow and green) with 20 uncorrelated predictor attributes per instance. The 0.632+Bootstrap method applied in this research therefore uses a modified form of bootstrap sampling to calculate $\text{Err}_1$, as described in section 4.6. This research also uses a loss function, which is different to the accuracy metric $Q(y, r)$, (equation [15]). The loss function is described in detail in section 4.6.3.1. The loss function is preferred to the accuracy metric because it takes into account the cost of specific errors in the three-class confusion matrix, rather than relying only the accuracy metric $Q(y, r)$.

4.5.2.2 The Paired t-test

Observing differences in performance of two classifiers on a dataset is necessary but not sufficient to conclude that a particular classifier has better performance than another. To confirm superior performance the difference in performance needs to be shown to be statistically significant and not merely due to chance. Null-hypothesis statistical testing is therefore widely applied in the machine-learning community to obtain more precise assessments of significance of performance results.

To validate that a particular classifier has superior performance that is statistically significant in comparison to another classifier, a paired t-test is applied during the model selection process. The null hypothesis of the paired t-test for two classifiers can be stated as follows: “The probability that the mean loss function difference of the two classifiers is zero, is larger than a statistically significant threshold”. The significance threshold is set in advance and is usually either 5% or 1%. So, for the 1%
threshold, the null hypothesis can be stated as: “The probability that the loss function values of the two classifiers are not different, is larger than 1%”. If this is not the case, i.e., the probability is less than 1%, then the null hypothesis can be discarded and the loss function difference is statistically significant.

The t-test is based on the following statistic (DeGroot, 2002):

\[ U = \frac{\sqrt{n}(\bar{X}_n - \mu_0)}{\sigma_t} \]  

[29]

Where:

\[ \bar{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i \]  

[30]

\[ \sigma_t = \sqrt{\frac{\sum_{i=1}^{n} (X_i - \bar{X}_n)^2}{n-1}} \]  

[31]

\( X_i \) is a random sample from a normal distribution with mean = \( \mu_0 \) and variance = \( \sigma^2 \), and \( i = 1,2,3,...,n \)

Under these conditions, the statistic \( U \) has the students-t distribution with \( n-1 \) degrees of freedom, regardless of the value of \( \sigma^2 \) (provided \( \sigma \) is finite). Furthermore, if we have a sufficiently large sample size (typically \( >200 \) – see (Japkowicz, 2011)) the difference in the means of two distributions will asymptotically approach a normal distribution irrespective of the population distribution, due to the Central Limit Theorem. Therefore, assuming \( n>200 \), we can set:

\[ \bar{X}_n = \bar{L}(f_1) \cdot \bar{L}(f_2) \]  

[32]
Where

\[ L(f) = \frac{1}{n} \sum_{i=1}^{n} L(f)_i \]  \[33\]

\( L(f)_i \) = the \( i \)th iteration of the loss function value for classifier \( f \)

We can therefore calculate the \( U \) statistic defined in equation [29] for the difference in means of the loss function values of two classifiers, and then use the students-t distribution to determine the probability of the differences in means calculated in equation [32]. If that probability is below a pre-set threshold, typically 5% or 1%, then the null hypothesis can be rejected with 95% or 99% confidence.

### 4.5.2.3 The Permutation Test (Standalone t-test)

Once an optimal classifier is identified using the 0.632+Bootstrap validation algorithm, thereby ensuring that the optimal trade off of bias and variance has been achieved, then it remains necessary to verify that the selected classifier is performing better than a classifier trained on predictors that contain no information. This can be achieved using the permutation test. A permutation test randomly varies the predictor attributes in training set instances while holding the target attribute constant. This is equivalent to a t-test where the null hypothesis is: “the probability that the mean loss function difference between a classifier trained on the base training set and the same classifier trained on randomly generated noise (with the same marginal distributions as the training set) is larger than a pre-specified threshold”. Again, this threshold is typically set at 1% or 5%. For the 1% case, the
null hypothesis would read: “the probability that the loss function difference between a classifier trained on the base training set and the same classifier trained on random noise is greater than 1%”. If this is not the case, i.e., this probability is less than 1%, then the null hypothesis can be rejected.

4.6 Application

In this section the theoretical framework of section 4.5 and the prior research reviewed in section 4.4 are synthesized and applied to a dataset that is manually extracted from industry sources. In section 4.6.1 that attribute selection process is defined and applied. In section 4.6.2 the process of extraction of attributes from industry databases is described and applied. These two sections yield the final dataset applied in this research. Section 4.6.3 then defines and applies the model selection and validation processes, followed in section 4.6.4 by model assessment.

4.6.1 Attribute Selection

The attribute selection process applied in this research was customized for the proposed application. The first step in the process was to define the optimal training set attributes for each vector in the training set. Three sources were applied to do this, which resulted in a set of 20 uncorrelated predictor attributes.

1) An initial set was defined by applying domain knowledge acquired from several years in venture capital practice and from more than 10 years of
raising venture capital funding for semiconductor technology start-up companies. This resulted in a set of 9 attributes.

2) This initial set was adjusted by interviewing practicing general partners at Silicon Valley venture capital firms. This resulted in the addition of two attributes. Most VC practitioners used smaller sets of attributes than the 9 defined in step 1 and most of the attributes that they used were already included in that set of 9 attributes.

3) Finally, the list was extended by extracting attributes that were identified in the literature to have statistically significant correlation with successful outcomes. To do this, an extensive literature search was conducted to identify papers that met this criterion. These papers were then ranked by citation rate (number of citations per year from date of publication), and, starting at the highest citation rate, success factors (attributes) were extracted that were identified in at least two separate publications. The 10 most cited papers yielded a set of 33 attributes (including the 11 attributes from steps 1 and 2), after which no new success factors were identified.

The list was then reviewed to remove attributes that were duplicates or were highly correlated with other attributes. For example, the success factor “Has a track record relevant to the venture; Successful and fast rising career in previous organization” and “Demonstrated leadership ability in the past; At least 2 years of experience in same position as in new venture” are effectively duplicates and therefore highly correlated. Only the latter was kept. This final step resulted in a set of 20 attributes
in four categories, as follows: founding team: 8; product: 6; market: 5; other: 1.

Figure 18 summarizes each of the 20 predictor attributes by category. The full set of attributes, together with details of the journal from which they were extracted and the process for quantifying each attribute, is summarized in Appendix 1. Table 4 illustrates quantification for three of the attributes.

![Diagram of Predictor Attributes]

**Predictor Attributes**

**Founding Team**
- EFF - Capable of intense sustained effort
- GRO - Able to evaluate and react to risk
- MKT - Market Experience
- EXP - Leadership Experience
- INO - Able to Innovate
- CPL - Team Completeness
- TCH - Quality of Technical Team
- JNT - Joint Experience of Team

**Product Value Proposition**
- PTN - Intellectual Property
- MAC - Market Acceptance
- VPR - Value Proposition
- PPL - Product Execution Plan
- SCH - Supply Chain Integration
- PRT - Functioning Prototype

**Market Dynamics**
- MGR - Market Growth Rate
- SOM - Target Market Share
- MXS - Targets Existing Market
- CMP - Competitive Landscape
- MDV - Market Diversity

**Other**
- AGE - Company Age

**Figure 18:** The 20 predictor attributes of the classifier feature vector. The target attribute is the nominal IRR range for that instance, defined as Red, Yellow or Green, depending on the numerical IRR of the instance.
Table 4: Three examples of the description, measure and quantification process for selected predictor attributes.

### 4.6.2 Extracting the Training Set

The next step in the statistical learning process illustrated in figure 15 is to extract the 21 attributes (now including the target attribute) for each company in the training set. Access was provided to the due diligence databases of a Silicon Valley venture capital fund and a Silicon Valley association of angel investors, together containing 170 investments. Of these, 14 Venture Capital investments and 32 Angel investments have had exits, so that the actual IRR of each of these 46 investments is known. A portion of these 46 investments did not contain enough information in the
due diligence files to extract at least 16 of the attributes and therefore were unusable. This extraction process is illustrated in Fig. 18.

![Diagram](image)

**Figure 19:** Illustration of the selection funnel for extracting the 20 attributes for each instance in the dataset.

Extracting the 21 attributes for each company in these due diligence databases was an extremely time consuming process. The desired 21 attributes had to be extracted from unstructured files of varying depth and completeness. In many cases information had to be sought from sources other than the due diligence files, such as
the LinkedIn website which contains resumes or biographies of many of the founders of the companies included in the training set.

In several cases judgment had to be exercised to extract the attributes. For example, judgment had to be exercised whether certain levels of working experience qualified as being equivalent to that occupied at the new enterprise. In some instances, the granularity of quantification had to be increased to better differentiate among enterprises. For example, the attribute SCH, which relates to the supply chain contracts that the enterprise has signed, was initially a binary attribute (the enterprise either had contracts with suppliers or did not). Analysis of the databases however showed that some enterprises had several such contracts for various aspects of their business, while others had only one or none. Clearly, for the same industry, the enterprise with several supply chain contracts is in better position than one which does not have such arrangements, so a range was built into this attribute with higher scores for the enterprises that have multiple supply chain contracts. This necessitated that the dataset had to be revised to ensure that all instances correctly reflected this change to the quantification of the SCH attribute.

Each feature vector took an average of about 30 hours to extract, if sufficient information was available in the due diligence files. In several cases the files did not contain sufficient information to extract all 20 attributes, a condition that could also take a considerable amount of time to discover.
If four attributes or less were missing for an enterprise, that enterprise was still included in the dataset, but if more than 4 attributes were missing, the enterprise was excluded from the dataset. It was necessary, therefore, to ensure that the classifier algorithms selected in step 3 of the statistical learning process all were capable of dealing with missing attributes in the training set.

To eliminate the effect of different absolute values in some of the attributes, each attribute was normalized to a value between 1 and 0. For example, the number of years of marketing experience was expressed as a fraction of the total number of years of working experience, effectively eliminating any effect that differences in the total work experience of the founding team at each company may have on the MKT parameter (the MKT parameter measures only the marketing experience of each founding team).

Appendix 2 contains the final data set with notes on any assumptions or judgments that had to be made during the extraction process.

### 4.6.3 Model Selection

A process for model selection also had to be developed during this research to maximize the probability that the population of classifiers tested yielded the optimal classifier for this application, given the size and dimensionality of the dataset. The process applied was as follows:
1) Review the literature to identify candidate learning algorithms that have performed well in similar conditions.

2) Complement the initial set from step 1 with classifiers that use fundamentally different classification methods, to maximize the diversity of classifiers applied.

3) Add meta-classifiers (ensemble techniques that take a vote among several classification results).

The literature survey revealed that three statistical learning algorithms appear to be well suited to small datasets of high dimensionality, namely, \( k \)-nearest neighbors, support vector machines and logistic regression. Both the \( k \)-nearest neighbors algorithm and support vector machines are known to have low error rates for small data sets with high dimensionality, provided that the attributes are uncorrelated, as illustrated in Figures 20a and 20b (from Hua, 2005). Note in Fig. 20a that the error rate for an SVM rapidly declines to a steady state well below 5% at \( n \) (sample size) \( \approx 20 \) for \( d \) (feature size) > 20, and also that error rate declines monotonically with \( d \). The same holds for the \( k \)-nearest neighbor algorithm, but with a steeper decline in error rate as sample size increases to \( \sim 30 \). This means that for both a support vector machine and the \( k \)-nearest neighbors algorithm, the larger the number of (uncorrelated) attributes the lower the error rate, starting with sample sizes as small as \( n = 10 \).

Also, logistic regression has been widely applied in the biological and medical sciences to datasets of high dimensionality and small size, with various techniques developed to identify sample size (Hsieh, 1998).
Figure 20a: Error rate of synthetically generated data for two-class classification from known class prior distributions (Gaussian, equal standard deviation, Bayes error = 0.05, attribute correlation = 0) applied to a linear kernel support vector machine, by feature size and sample size. The black dotted line shows the minimum error. (From: Hua, 2005).
Figure 20b: Error rate of synthetically generated data for two-class classification from known class prior distributions (Gaussian, equal standard deviation, Bayes error = 0.05, attribute correlation = 0) applied to a k-nearest neighbors algorithm with k=3, by feature size and sample size. The black dotted line shows the minimum error. (From: Hua, 2005).

This process resulted in a set of 6 supervised and unsupervised classifier algorithms and 2 meta classifier algorithms (see summary in table 5).
Table 5: Summary of classifier algorithms in the preliminary set

<table>
<thead>
<tr>
<th>WEKA CLASSIFIER</th>
<th>DESCRIPTION</th>
<th>CHARACTERISTICS</th>
<th>SOURCE</th>
</tr>
</thead>
</table>

4.6.3.1. Definition of the Loss Function

The measure of performance, $L(f)$ of a statistical learning algorithm can be any function that monotonically orders performance of classifiers. This means that for a set of $n$ classifiers $f_1, f_2, \ldots, f_n$, ranking them by their performance measure, such that, say, $L(f_2) < L(f_5) < L(f_1)$ ... implies that $f_2$ performed better than $f_5$, which in turn performed better than $f_1$. In this research the performance measure is called a Loss Function because it measures the loss of performance of each classifier. A custom loss function is applied in this research, as opposed to the usual measures of performance such as accuracy or root mean square error. This is because the private equity application of the classifiers results in certain misclassification errors being considerably more costly than others.
Since the task of the classifier is to classify prospective new issuers into the three target categories, misclassification of Red issuers as Green and Green issuers as Red should be most penalized, because that will cause potentially attractive investments to be classified as bad, and vice versa. Misclassifying Red or Green issuers as Yellow or Yellow issuers as Red or Green has less consequence, as in both cases classification as Yellow will likely result in further scrutiny of the company and an appropriate decision can then be taken, and misclassifying a Yellow company does not represent a significant loss. Therefore a lower penalty should be applied. Correct classification should be appropriately awarded. The loss function applied to the classifier results was therefore defined as follows:

\[
L_c(x) = \frac{2n_t - (2(T_g + T_y + T_r) - 4(F_{rg} + F_{gr}) - (F_{ry} + F_{yr} + F_{gy} + F_{yg}))}{n_t}
\]  

[29]

Where:

\[x = \text{the } n \times m \text{ training set matrix, } m = \text{number of attributes (incl. target), } n = \text{number of instances,}\]

\[n_t = T_g + T_y + T_r + F_{rg} + F_{gr} + F_{ry} + F_{yr} + F_{gy} + F_{yg} = \text{the number of test instances}\]

\[L_c(x) = \text{the loss function for the model generated by classifier C on training set } x\]

and whose resultant confusion matrix is defined as:

<table>
<thead>
<tr>
<th>Classified as</th>
<th>Red</th>
<th>Yellow</th>
<th>Green</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(T_r)</td>
<td>(F_{ry})</td>
<td>(F_{rg})</td>
</tr>
<tr>
<td>(F_{yr})</td>
<td>(T_y)</td>
<td>(F_{yg})</td>
<td>Yel</td>
</tr>
<tr>
<td>(F_{gr})</td>
<td>(F_{gy})</td>
<td>(T_g)</td>
<td>Grn</td>
</tr>
</tbody>
</table>
$L_c(\mathbf{x})$ varies from 0 for the best case scenario when the classifier result perfectly matches the training data, to 5.25 for the worst case scenario when the classifier classifies all Reds as Green, all greens as Red and all Yellows as either Red or Green.

The loss function penalizes serious misclassification by a factor of 4, less serious misclassification by a factor of 1, and rewards correct classification by a factor of 2. Division by $n_c$ of the result is required to normalize the data, thereby allowing comparison to results where differing numbers of test instances are used (as is the case for each run of the 0.362+bootstrap validation technique, for example). In applying 0.632+Bootstrap validation in this research $L_c(\mathbf{x})$ replaces the accuracy metric $Q(y, x)$ (equation [15]) described in section 4.5.1.2. $Q(y, x)$ (equation [15]) is, however, still required to calculate $\tilde{R}'$ (equation[26]).

4.6.3.2 Preliminary Model Selection Using Leave-One-Out Cross Validation

A preliminary assessment of each classifier was completed by applying the training set to each classifier, using the Weka Knowledgeflow and Experimenter environments, to identify the classifier that yields the lowest loss function value based on leave-one-out cross validation.

An instantiation of the Weka Knowledgeflow environment used to test the preliminary set of classifiers is shown in Fig. 21, illustrating how a test result was captured to a .csv file for further analysis using CSVSaver, as well as providing the
option to display results as text files using TextViewer. The confusion matrices of each classifier in the preliminary set are illustrated in Table 6.

Figure 21: The Weka Knowledgeflow diagram used for initial evaluation of the 8 classifier algorithms selected for this analysis. The diagram illustrates the loading of a .csv file (CSVLoader), class assignment in the training and test set (ClassAssigner), application of leave-one-out cross validation for testing (CrossValidationFoldMaker), the logistic, multilayer perceptron (neural net) and support vector machine classifiers and various performance evaluation and visualization tools.

4.6.3.3 Final Model Selection Using 0.632+Bootstrap

Leave-one-out cross validation, while minimizing bias by using each instance in the data set as a test set during each fold, still suffers from considerable variance, as
discussed in section 4.4.1. In small datasets, as is the case here, removing even one of the instances in a dataset set for the test set can materially affect the distribution of the remaining instances, especially in multi-class applications where the classes are not equally distributed, as is the case here. This will cause high variance in the classifier models for each fold.

Several techniques have been developed by researchers to reduce the effect of variance resulting from small data sets and $k$-fold or leave-one-out cross validation (Japkowicz, 2011). Of these the 0.632+Bootstrap algorithm provides the optimum balance for minimizing bias and variance effects in small datasets, as described in section 4.5.1.1.

A Knowledgeflow environment illustrating the bootstrap process to calculate $Err_1$, applied in equations 20, 23 and 27, is shown in Figure 22. The resample filter ("Resample" icon in Fig. 22) has options for randomly resampling with or without replacement, using a random number generator (RNG) seed, and produces the selected instances or their complement (i.e., the instances that were not selected) when sampling without replacement. The first two resample filters are used to randomly extract a test set and its complement, by sampling without replacement. The test set size and its complement are parameters that can be set by the user by using two Resample filters with the same RNG seed, as shown in Fig. 21.
Figure 22: Knowledgeflow environment illustrating the modified e0 bootstrap approach to calculating $\text{Err}_1$ for Logistic Regression, Support Vector Machine and Simple Logistic classifiers.

The complement set is then again resampled with replacement to create a training set that contains the same number of instances as the original data set. Changing the RNG seed then results in a different random dataset for each iteration. The loss functions for a large number of iterations are then averaged to determine the performance of each of the classifiers.

The 0.632+ algorithm cannot be implemented directly in the Weka Knowledgeflow environment due to the limitations of the Knowledgeflow tool. The Knowledgeflow
environment illustrated in Fig. 22 is useful for testing small numbers of iterations under different conditions, but is not suited for the large numbers of iterations required for effective bootstrapping, since Knowledgflow requires manual changes to the random number generator (RNG) seed for each iteration. Consequently the 0.632+Bootstrap algorithm was implemented in R using the RWeka library to access the Weka classifiers. The R code for all functions to implement the 0.632+Bootstrap algorithm is listed in Appendix 3. The code was applied to calculate $Err_{0.632+}$ using equation [28] for each of the selected classifiers, enabling comparisons of their loss function values under conditions of optimal bias – variance trade off.

**4.6.4 Model Assessment**

Given the small dataset, a key concern is to assess the quality of the resultant model generated by the selected optimum classifier, the $k$-nearest neighbors algorithm ($k$NN). Various techniques have been developed to assess the strength or weakness of a learning algorithm (Japkowicz, 2011), the primary one of which is the t-test described in section 4.5.2.2 and the related permutation test described in section 4.5.2.3. The permutation test calculates a probability that a random classifier could achieve the same or better results than the base classifier. A random classifier is defined as a classifier trained on random data with the same distribution of attributes and classes as the base training set.
Figure 23 illustrates the Knowledgeflow environment for the permutation test. The key element of this environment is the PartitionedMultiFilter. This is a filter that applies filters on subsets of attributes and assembles the output into a new dataset. Attributes that are not covered by any of the ranges can be either retained or removed from the output. In Fig. 23 this filter is applied to partition the attributes in the base data set into the two sets, the class attributes and the predictor attributes. A Randomizer filter is then applied to the partitioned attributes such that the predictor attributes of each instance is randomly varied while the target attribute is held constant as per the original distribution.

![Knowledgeflow environment for the permutation test](image)

**Figure 23: Knowledgeflow environment for the permutation test**

This results in a new dataset of the same size as the base set but with the predictor attributes of each instance randomly assigned to the class attributes, effectively “permuting” the instances. This process is equivalent to holding instances constant while randomly assigning class attributes to those instances with the same distribution as the base data set.
This randomization process is repeated for N runs, with N>200, creating a new random classifier with each iteration. On each iteration, the loss function for each random classifier is then computed from the confusion matrix and the value of the loss function compared to the loss function value achieved by the base classifier under 0.632+Bootstrap validation. For each iteration, if the random classifier loss function equals or is lower than that of the base classifier, that iteration is scored as a “1”, and if the random classifier loss function is higher than that of the base classifier, that run is scored as a “0”. Once N iterations are completed, the iteration scores are summed and the result divided by N, to yield the probability that a random kNN classifier can achieve equal or better performance as the base classifier.

Figure 24 lists the pseudo-code for this permutation test. The R code is listed in Appendix 5.
Pseudo-code for permutation test of a single classifier:

- Initialize (set size & dimension, t-test iterations (>=200), bootstrap iterations (>=50), data set, classifier)
- Calculate mean of “bootstrap iterations” 0.632+Bootstrap loss functions for “classifier” #"actual loss” of “classifier”
- for (i in 1:"t-test iterations")
  {
    - Randomly shuffle predictor attributes of instances to create “no info set”
    - Calculate mean of “bootstrap iterations” 0.632+Bootstrap “shuffled loss functions” for “classifier” using “no info set”
      - if mean of “shuffled loss functions” of “classifier”<=”actual loss”, increment “counter”
  }
- Calculate p-value = “counter”/”t-test iterations”
- Print("The probability of a “classifier” trained on randomized data achieving the same results as the “classifier” trained on the dataset is: “p-value”")

Figure 24: Pseudo code for the permutation test. The R implementation is listed in Appendix 5.

4.7 Results

The results obtained from applying the theory and processes outlined in the previous sections are presented in this section. The results are organized into three sub-sections:
1) Section 4.7.1 provides the results of applying leave-one-out (LOO) cross validation on each of the 8 selected learning algorithms. LOO cross validation is defined by equation [17] with $k=1$ and provides the lowest bias for the cross validation technique on small datasets such as the final dataset. This provides a preliminary indication of performance for each of the selected algorithms.

2) Section 4.7.2 provides both a summary (Table 7) and the detailed results of the 0.632+Bootstrap validation technique on 6 of the 8 selected algorithms. The meta classifiers were eliminated after LOO cross validation as both performed poorly relative to the base $k$NN and SVM classifiers, illustrating that bagging and boosting added no improvement in the case of small datasets. The LOO cross validation results also identified the $k$NN algorithm as the likely best performing algorithm, and so it was selected as the primary comparative algorithm in the pairwise t-test comparisons for the remaining 5 classifiers.

3) Section 4.7.3 provides the assessment results for the best performing algorithm, namely $k$NN with $k=1$. A t-test was applied to determine the statistical significance for rejecting the null hypothesis, which in this case is: “The probability of a “no information” classifier achieving the same results as the base classifier is $>5\%$”. The “no information” classifier results were obtained by randomly scrambling the predictor attributes while maintaining the target attributes in the final dataset. This ensured that the distribution of target classes remained the same as in the final dataset.
4.7.1 Leave-one-out cross validation results

Preliminary model selection was done using leave-one-out cross validation. This identified the classifiers that were likely to perform best under bootstrapping also, since the 0.632+Bootstrap includes a bootstrapped leave-one-out component. Results for leave-one out cross validation are shown in Table 6.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Confusion Matrix</th>
<th>Loss Function</th>
<th>Classifier</th>
<th>Confusion Matrix</th>
<th>Loss Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>k Nearest Neighbor</td>
<td>a b c &lt;-- classified as 6 0 0</td>
<td>a = red</td>
<td>1.3</td>
<td>a b c &lt;-- classified as 5 0 1</td>
<td>a = red</td>
</tr>
<tr>
<td>(k=2)</td>
<td>1 2 0</td>
<td>b = green</td>
<td></td>
<td>1 0 2</td>
<td>b = green</td>
</tr>
<tr>
<td></td>
<td>2 1 0</td>
<td>c = yellow</td>
<td></td>
<td>1 2 0</td>
<td>c = yellow</td>
</tr>
<tr>
<td>k Nearest Neighbor</td>
<td>a b c &lt;-- classified as 5 0 1</td>
<td>a = red</td>
<td>1.8</td>
<td>a b c &lt;-- classified as 4 1 1</td>
<td>a = red</td>
</tr>
<tr>
<td>(k=1)</td>
<td>0 0 3</td>
<td>b = green</td>
<td></td>
<td>0 0 3</td>
<td>b = green</td>
</tr>
<tr>
<td></td>
<td>1 2 0</td>
<td>c = yellow</td>
<td></td>
<td>1 2 0</td>
<td>c = yellow</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>a b c &lt;-- classified as 5 0 1</td>
<td>a = red</td>
<td>1.8</td>
<td>a b c &lt;-- classified as 3 2 0</td>
<td>a = red</td>
</tr>
<tr>
<td></td>
<td>0 0 3</td>
<td>b = green</td>
<td></td>
<td>1 0 2</td>
<td>b = green</td>
</tr>
<tr>
<td></td>
<td>1 2 0</td>
<td>c = yellow</td>
<td></td>
<td>2 1 0</td>
<td>c = yellow</td>
</tr>
<tr>
<td>Bagging</td>
<td>a b c &lt;-- classified as 4 0 2</td>
<td>a = red</td>
<td>2.0</td>
<td>a b c &lt;-- classified as 0 4 2</td>
<td>a = red</td>
</tr>
<tr>
<td>(KNN, k=1, 100 iterations)</td>
<td>0 0 3</td>
<td>b = green</td>
<td></td>
<td>1 0 2</td>
<td>b = green</td>
</tr>
<tr>
<td></td>
<td>1 2 0</td>
<td>c = yellow</td>
<td></td>
<td>3 0 0</td>
<td>c = yellow</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>a b c &lt;-- classified as 5 0 1</td>
<td>a = red</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 0 2</td>
<td>b = green</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 2 0</td>
<td>c = yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Confusion matrix results for the preliminary set of classifier algorithms validated using leave-one-out cross validation, ranked in order of lowest to highest loss function value. The k nearest neighbors (kNN, with k=2) algorithm has the lowest loss function of the set, followed by kNN with k=1 and the support vector machine.
These results show that $L_{K(2)NN}(\mathbf{x})$ has the lowest loss function value of 1.3, followed by $L_{K(1)NN}(\mathbf{x}) = L_{SVM}(\mathbf{x}) = 1.8$, and the loss function values of all other classifiers are higher.

4.7.2 0.632+Bootstrap Results

The 0.632+Bootstrap validation algorithm was applied to compare the performance of classifier pairs. Given the results of the leave-one-out cross validated preliminary analysis, the IBk ($kNN$) classifier was selected as the base classifier against which the other classifiers in table 5 would be compared, using a pairwise t-test. The results for each of the 5 pairwise classifier t-test comparisons are summarized in Table 7, followed by the detailed results for each pairwise test and the distribution histogram for the 500 bootstrap iterations applied in the 0.632+Bootstrap algorithm for each algorithm pair.

Each of the following pairwise classifier results contains:

1) The mean loss function values for each of the two classifiers;

2) The results of the paired t-test;

3) The distribution histogram of loss function values for each of the two classifiers, with the mean superimposed on the histogram for comparison.
### 0.632+Bootstrap Results (vs. $k$-Nearest Neighbor, $k=1$)
(500 iterations)

<table>
<thead>
<tr>
<th></th>
<th>Mean of Differences in Loss Function</th>
<th>95% Confidence Interval</th>
<th>Paired t-test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>p-value</td>
</tr>
<tr>
<td>J48</td>
<td>0.0859</td>
<td>0.0191</td>
<td>0.0118</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.5282</td>
</tr>
<tr>
<td>SVM</td>
<td>0.0896</td>
<td>0.0236</td>
<td>0.0079</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.6667</td>
</tr>
<tr>
<td>Neural Net</td>
<td>0.1674</td>
<td>0.0979</td>
<td>2.87E-06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.7341</td>
</tr>
<tr>
<td>IBk(k=2)</td>
<td>0.2129</td>
<td>0.2835</td>
<td>6.01E-09</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.9196</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>0.2339</td>
<td>0.1678</td>
<td>1.11E-11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6.9558</td>
</tr>
<tr>
<td>Logistic</td>
<td>0.3418</td>
<td>0.2713</td>
<td>&lt; 2.2E-16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9.5251</td>
</tr>
</tbody>
</table>

Table 7: Summary of the pairwise t-test of each selected classifier relative to the $k$-nearest neighbors algorithm with $k=1$. The 95% confidence interval is positive for each tested learning algorithm, showing that the $k$-nearest neighbors algorithm has a lower loss function value than each of the tested classifiers with > 95% confidence. The $k$-nearest neighbors algorithm is therefore the optimal classifier.
4.7.2.1 Results of e0Bootstrap: IBk(k=1) and IBk(k=2)

Two instantiations of the same learning algorithm, but with different parameters, result in the same resubstitution error, \( E_r \), in equation [14]. Therefore to compare performance in such a case the difference in 0.632+Bootstrap results and modified e0Bootstrap yield the same results. Since the e0Bootstrap is computationally simpler, the following results are for a pairwise t-test where the e0Bootstrap was applied as the validation method. This required some simplification of the pairwise t-test code. The simplified code is listed below.

dataSet=CombData  # this is the 17 instance final dataset
setSize=17
dimension=21
iter=500
classifier1e0Boot<-numeric(iter)
classifier2e0Boot<-numeric(iter)
for(i in 1:iter){
    Subsamp<-sample(setSize, setSize, replace=TRUE)
    Basesamp<-1:setSize
    oneTrain<-dataSet[Subsamp, 1:dimension]
    oneTest<-dataSet[setdiff(Basesamp, Subsamp), 1:dimension]
    classifier1model<-IBk(IRR_Nominal~., data=oneTrain)
    classifier2model<-IBk(IRR_Nominal~.,
data=oneTrain, control=Weka_control(K=2))
    classifier1eval<-evaluate_Weka_classifier(classifier1model, newdata=oneTest)
    classifier1acc<-lossFunct(classifier1eval)
    classifier2eval<-evaluate_Weka_classifier(classifier2model, newdata=oneTest)
    classifier2acc<-lossFunct(classifier2eval)
    classifier1e0Boot[i]=classifier1acc
    classifier2e0Boot[i]=classifier2acc
}
mean(classifier1e0Boot)
mean(classifier2e0Boot)

par(mfrow=c(2,1))
hist(classifier1e0Boot, breaks=c(0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0, 3.25, 3.5, 3.75, 4.0, 4.25, 4.5, 4.75, 5), density = 20, col="red", main="Histogram of e632 for IBk(k=1)", xlab= "Loss Function", xlim=c(0,5.5), ylim=c(0,100), labels=TRUE)
lines(c(mean(classifier1e0Boot), mean(classifier1e0Boot)), c(0,150), lwd=3)

hist(classifier2e0Boot, breaks=c(0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0, 3.25, 3.5, 3.75, 4.0, 4.25, 4.5, 4.75, 5), density = 20, col="blue", main="Histogram of e632 for..."
The means of the loss function results for each algorithm, the results of the pairwise t-test and the histogram of loss function results for the 500 iterations follow:

**Mean of Loss Functions:**

```r
> e632+J48
[1] 1.478081
> e632+IBk
[1] 1.39217
```

**Paired t-test**

data: classifier1e0Boot and classifier2e0Boot
t = -5.9196, df = 499, p-value = 6.008e-09
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.2834990 -0.1422058
sample estimates:
mean of the differences
-0.2128524
Figure 25: Histogram of the loss function results for 500 bootstrap iterations of the IBk ($k=1$) and the IBk ($k=2$) algorithm. The mean values are superimposed on the histograms, showing the difference in performance between the two algorithms.

Sections 4.7.2.2 through 4.7.2.6 each show the outputs for 500 iterations of the 0.632+Bootstrap pairwise t-tests, using the same layout, from the remaining 5 learning algorithms tested. Each algorithm is compared to the kNN ($k=1$) algorithm.
4.7.2.2 Results of e632+Bootstrap: J48 and IBk (k=1)

Mean of Loss Functions:
> e632+J48
[1] 1.478081
> e632+IBk
[1] 1.392176

Paired t-test:
data: e632+Bootstraps[1,] and e632+Bootstraps[2,]
t = 2.5282, df = 499, p-value = 0.01177
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.01914579 0.15266441
sample estimates:
mean of the differences
0.0859051

Figure 26: Histogram of the loss function results for 500 bootstrap iterations of the IBk (k=1) and the J48 Decision Tree algorithm. The mean values are superimposed on the histograms, showing the difference in performance between the two algorithms.
4.7.2.3 Results of e632+Bootstrap: SVM and IBk (k=1)

Mean of Loss Functions:
> e632IBk
[1] 1.541084
> e632SVM
[1] 1.630635

Paired t-test:
data: e632+Bootstraps[1, ] and e632+Bootstraps[2, ]
t = 2.6667, df = 499, p-value = 0.007909
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
0.02357276 0.15552812
sample estimates:
mean of the differences
0.08955044

Figure 27: Histogram of the loss function results for 500 bootstrap iterations of the IBk (k=1) and the SMO support vector machine algorithm. The mean values are superimposed on the histograms, showing the difference in performance between the two algorithms.
4.7.2.4 Results of e632+Bootstrap: Multilayer Perceptron and IBk (k=1)

Mean of Loss Functions:
> e632+MP
[1] 1.719997
> e632+IBk
[1] 1.552578

Paired t-test
data: e632+Bootstraps[1, ] and e632+Bootstraps[2, ]
t = 4.7341, df = 499, p-value = 2.87e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
0.09793787 0.23689990
sample estimates:
mean of the differences
0.1674189

Figure 28: Histogram of the loss function results for 500 bootstrap iterations of the IBk (k=1) and the multilayer perceptron (neural net) algorithm. The mean values are superimposed on the histograms, showing the difference in performance between the two algorithms.
4.7.2.5 Results of e632+Bootstrap: Naïve Bayes and IBk (k=1)

Mean of Loss Functions:
> e632+NB
[1] 1.632355
> e632+IBk
[1] 1.398482

Paired t-test
data: e632+Bootstraps[1, ] and e632+Bootstraps[2, ]
t = 6.9558, df = 499, p-value = 1.105e-11
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
0.1678137 0.2999324
sample estimates:
mean of the differences
0.2338731

Figure 29: Histogram of the loss function results for 500 bootstrap iterations of the IBk (k=1) and the Naïve Bayes algorithm. The mean values are superimposed on the histograms, showing the difference in performance between the two algorithms.
4.7.2.6 Results of e632+Bootstrap: Logistic and IBk (k=1)

Mean of Loss Functions:
> e632+Logistic
[1] 1.508825
> e632+IBk
[1] 1.380553

Paired t-test
data: e632+Bootstraps[1,] and e632+Bootstraps[2,]
t = 9.5251, df = 499, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
0.2712714 0.4122625
sample estimates:
mean of the differences
0.3417669

Figure 30: Histogram of the loss function results for 500 bootstrap iterations of the IBk (k=1) and the logistic regression algorithm. The mean values are superimposed on the histograms, showing the difference in performance between the two algorithms.
4.7.3 Results of the Permutation Test on the IBk Classifier

To verify that the performance of the IBk classifier trained on the final dataset is better than the performance of the IBk classifier trained on random noise, a permutation test was performed. The R code for the permutation test is listed in appendix 5. Results of the permutation test are shown in table 8.

Table 8: The loss functions of 200 IBk classifiers each validated using 50 iterations of the 0.632+Bootstrap algorithm on randomly permuted training attributes with the target attribute held constant. Only 11 of the 200 randomized IBk classifiers (highlighted) had a lower loss function than the original training set, resulting in a p-value of 0.055. Loss function of the original data set validated using 50 iterations of the 0.632+Bootstrap algorithm: 1.301017

Table shows that the probability of an IBk classifier trained on randomly permuted training attributes achieving the same results as the IBk classifier trained on the original dataset is: 0.055. The null hypothesis in this case is that no difference exists between the IBk classifier trained on the original dataset and an IBk classifier trained...
on randomized data with the same marginal distributions for the training and target attributes. This result is equivalent to a t-test with a p-value of 0.055, implying 94.5% confidence for rejecting the null hypothesis.

4.8 Conclusions

The results show that the IBk ($k$-nearest neighbors, $k=1$) classifier has the lowest loss function of the classifiers tested, with >95% confidence. This result implies that the underlying structure of the relationship between the target attributes and the predictor attributes is complex and the boundaries between each class are not well defined by discriminatory hyperplanes, as for logistic regression. Logistic regression specifically depends on a logit function hyperplane boundary between two classes. The poor results for logistic regression imply that a single hyperplane boundary probably does not exist in the final dataset. The normalized information gain splitting criterion of the decision tree algorithm (J48) works well in such complex boundary datasets. The J48 algorithm does not attempt to define a boundary between classes, but rather splits only on the information gain content of the attributes, hence its relatively good performance. The support vector machine should work well with complex datasets such as the final dataset, but is somewhat dependent on kernel function selection. Experimentation with different SVM kernel functions is included in the future work planned for this research. The multilayer perceptron is ideal for very large datasets where sufficient data is available for accurate back-propagation calculations. The MP therefore was not expected to

133
perform well on the final dataset, however it did perform better than expected, implying that the high dimensionality and orthogonality of the attributes of the final dataset contributed to its performance. Confirming this is another avenue for future research.

The $k$NN algorithm was therefore selected as the optimal classifier for this application due to the fact that it outperformed all other algorithms with $>98\%$ confidence. Note that the fact that all attributes are normalized ensures that the Euclidian distance calculation of the $k$NN algorithm is not dominated by large absolute values of any one attribute. Note also that the $k$NN algorithm is unsupervised and therefore requires no training.
5. Conclusions and Future Work

In this section the primary conclusions and the impact of the research is reviewed in sections 5.1. and 5.2. The contributions of the research are listed in section 5.3 showing how this research has addressed the research issues identified in chapter 1. The research has resulted in several new avenues for future work, which are discussed in section 5.4.

5.1 Impact of the Research and Key Conclusions

Prior to this research no framework and computational tools existed to perform due diligence on private equity investments. A framework and tools have now been developed and the results show that this framework can provide better than 80% accuracy in predicting whether a prospective investment is in the Green or Yellow IRR category.

Key conclusions from this work are:

1. The proposed IDSS architecture can classify prospective ECF investments into 3 IRR categories with the probability of an enterprise being categorized correctly and incorrectly but as “Yellow” = 0.8435

2. The IDSS classifies prospective investments into the three categories with a mean loss function value of 1.3 versus a “no information” mean loss function
value of 1.9, yielding a p-value of 0.055 implying that the null hypothesis can be rejected with >94.5% confidence

3. In Stage 1 of the IDSS this research showed that only a small subset of high growth enterprises qualifies for ECF investment, since at 10% higher risk, the expected IRR target for an efficient portfolio of ECF assets is >= 28%, with 95% confidence

4. In Stage 2 of the IDSS this research showed that the k-Nearest Neighbor (kNN) algorithm is the optimal classifier with >98% confidence based on paired t-tests

5.2 Summary of the Thesis

This thesis describes a set of methodologies and processes and their implementation and performance, that together enable building of an interactive web based Intelligent Decision Support System to partially automate the due diligence process in private equity investment.

5.3 Contributions of the Research

The contributions made in this research are as follows:

1) A software framework was defined for building an interactive web based IDSS, comprising two stages as defined in this research, for use by non-experts to enable on-line due diligence of new ECF issuers.
2) The largest database of rate of return data for the three primary asset classes in the private equity industry has been applied to the optimization process developed by Markowitz to determine the capital market line for the private equity industry.

3) The systematic risk of ECF investments has been qualitatively analyzed allowing for a rational estimation of the risk and rate of return required for an optimal portfolio of ECF investments, using the capital market line developed for the private equity industry.

4) The process applied to identify this ECF portfolio risk has been generalized to create a normative model for identifying the target returns that any optimal asset portfolio in a new asset class should generate.

5) A methodology was developed to determine the target return that an individual ECF asset should return, based on the optimal portfolio return identified from the normative model.

6) These principles were encoded into a software tool that enables identification of the rate of return of an asset based on user supplied data provided via an interactive website. This formed Stage 1 of the Intelligent Decision Support System proposed in this research.

7) A statistical learning process was developed for attribute selection, model selection and model validation to identify the optimal statistical learning algorithm for the dataset size and dimensionality available for this research.
8) A process was developed for optimal attribute selection using domain knowledge, industry interviews and research literature to define a set of 20 predictor attributes.

9) A dataset was manually extracted from the due diligence databases of a Silicon Valley venture capital firm and a Silicon Valley association of angel investors, with each instance containing 16 – 20 of the attributes identified in the attribute definition process. Each instance had known rates of return as the target attribute.

10) R code was written and verified for 0.632+Bootstrap validation and t-test verification to rank a set of 8 supervised, unsupervised and meta classifiers using the extracted dataset. The optimal classifier is the core intellectual property for stage 2 of the IDSS.

5.4 Future Work

Several future tasks are currently in process from this research:

1. Build the web based interactive IDSS platform:

This research has provided the architectural framework, identified the primary functional elements and defined an implementation process that enable building of the IDSS. The remaining tasks to implement the IDSS include:

• Completion of the s/w and h/w independent UML specification;
• Selection of a commercial grade software environment and database (Ruby on Rails, Python, MongoDB, SQL);

• Selection of a commercial grade, scalable, secure hardware platform (AWS, Azure);

• S/w & h/w specific implementation of the UML architecture to create a scalable, secure, commercial grade IDSS.

The implementation process for developing a commercial grade IDSS is shown in Figure 24 below.

![Diagram](image)

**Figure 31:** The implementation process for the IDSS, showing the sections of the process completed in this research and the remaining tasks. The remaining tasks are primarily concerned with completion of the software independent UML definition of the IDSS, followed by a software and hardware specific implementation of the IDSS.

2. Expand the classifier database:
The classifier database applied in this research is relatively small (17 instances). While the permutation test showed a statistically significant difference between the $k$-nearest neighbors algorithm trained on the source data vs. one trained on random data, the permutation test itself is based on data sampled from the small dataset, raising some of the bias concerns discussed in section 4.5.2.2. A larger dataset would reduce this concern. Also, a larger dataset would enable analysis of market segments, such as web-based enterprise software, mobile phone applications and SaaS companies, rather than a generic set of enterprises from widely differing market segments, as is the case for the final dataset applied in this research. Therefore, identifying and analyzing due diligence databases of more Angel/VC investors to further grow the classifier training & test set is planned.

3. Investigate performance of more classifiers:

While a diverse set of statistical learning algorithms have been applied in this research using a structured model selection process, it is possible that algorithms that have not been applied in this research could yield better results. Therefore, applying additional classifiers to the classifier database is also planned. The following have been identified as potentially interesting candidates:

- Ensemble classifiers beyond those applied in this research
- Parametric variations of the classifiers applied in this research

4. How to validate the IDSS in an actual application environment?
Validating the IDSS in real world applications remains a significant challenge. Given the complexity and time required to reach an investment decision on a prospective investment, as well as the proprietary nature of the data used in the classifier database, it is not feasible to test the IDSS performance by asking likely users to apply it to the data in the classifier database. Such a test would enable direct comparison between the IDSS performance versus that of various classes of users. Nevertheless, various ideas are being considered to emulate how likely users would perform relative to the IDSS.

5. Investigate generalization of the IDSS to other investment markets beyond private equity:

The principles and processes identified in this research are applicable to markets other than private equity. Widening the applicability of the IDSS to more markets would amplify the potential market for a commercial implementation of the IDSS and therefore is included in the on-going work.
6. Appendices

6.1 Appendix 1: Attribute Selection Criteria and Quantification Process

<table>
<thead>
<tr>
<th>Attribute:</th>
<th>Founders/Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>Capable of sustained</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>intense effort (team</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>commitment)</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>Able to evaluate and</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>react to risk well</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>Thoroughly familiar</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>with market targeted</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>by venture; Marketing</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>experience/capability;</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>experience of the</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>venture’s management</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>team in marketing</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>Demonstrated leadership</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>ability in the past;</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>At least 2 years of</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>experience in same</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>position as in new</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>venture; Managerial</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>Capability</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>Completeness of</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>management team (5%</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>key positions filled);</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>size of founding team</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>Creativity - ability</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>to innovate</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>Capability of the</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>technical team (tech</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>nical expertise,</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>expertise in product</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>development)</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>Degree of prior joint</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>experience of team (yrs</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>team members worked</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>together as % of total</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>possible time worked</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>together)</td>
<td>Founders/Management</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric:</th>
<th>Founders/Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of years each</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>founding team member</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>has been in executive</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>management or was a</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>founding team member</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>at successful, high</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>growth start-ups;</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>express as a fraction</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>of total work experience</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>of founding team</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>No. of years each</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>founding team member</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>worked at an</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>organization &gt;500</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>employees that had &gt;25%</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>revenue CAGR; express</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>as a fraction of total</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>work experience of</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>founding team</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>No. of years each</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>founding team member</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>has been in marketing</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>management position or</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>had direct responsibility for marketing function; express as a fraction of total work experience of founding team in years.</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>No. of years each</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>founding team member</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>has been in same position as in new venture; express as a fraction of total work experience of founding team in years.</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>Percentage of key positions filled in founding team; express as a fraction of total number of key positions</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>Number of successful (Annual revenue &gt;$2M) new products for which each founding team member had direct responsibility (i.e., secured internal or external funding to develop or led the technical development); =&gt;5=1; 4=0.8; 3=0.6; 2=0.4; 1=0.2; 0=0</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>Number of years each</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>founding team member</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>has worked with another founding team member; express as a fraction of total work experience of team</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>No. of years each</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>founding team member</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>has been in executive management or was a founding</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>team member at successful, high growth start-ups; express as a fraction of total work experience of founding team</td>
<td>Founders/Management</td>
</tr>
<tr>
<td>Attribute:</td>
<td>PTN</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Description:</td>
<td>The product is proprietary or can otherwise be protected; Patent protection</td>
</tr>
</tbody>
</table>

<p>| Metric: | Number of patents owned by the company; score by: &gt;40 = 1, else no. patents/40; issued patents count as a patent, provisional and applications pending count as half patent | Does the company have committed customer demand (PO, JDA, Partnership Agreement, NRE Agreement) for the product; 5 or more signed commitments = 1, otherwise no. of commitments/5 | Express as ((1 + \text{decimal cost reduction}) \times (1 + \text{decimal performance improvement}) = t); if (t &gt; 1.5) score = 1; if (t &gt; 1.4) score = 0.8 if (t &gt; 1.3) score = 0.6; if (t &gt; 1.2) score = 0.4; if (t &gt; 1.1) score = 0.2; else score = 0 | Product Development Plan scoring: At least weekly granularity = 1; each task staffed = 1; critical path identified = 1; dependencies identified = 1; key milestones identified = 1; Total score = sum of the above/5 | Does the company have committed supplier support (supply agreements, JDA, volume purchase agreements); Binary (yes; no) | Does the company have a functioning prototype of its first product? Binary (yes; no) |</p>
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Metric</th>
<th>Strategy</th>
<th>Other</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>The target market enjoys a significant growth rate</td>
<td>CAGR of SAM; if CAGR&gt;25% score =1; if CAGR&gt;20% score =0.8; if CAGR&gt;15% score =0.6; if CAGR&gt;10% score =0.4; if CAGR&gt;5% score =0.2; else score =0</td>
<td>Target SoM 3 years after product launch; if SoM&gt;25% score =1; if SoM&gt;20% score =0.8; if SoM&gt;15% score =0.6; if SoM&gt;10% score =0.4; if SoM&gt;5% score =0.2; else score =0</td>
<td>Number of direct competitors with more than 25% market share (=large competitor) in first two years after initial product release. Zero competitors =1; 1 competitor SoM=10% = 0.8; 1 or 2 comp: SoM &lt;=20% = 0.6; SoM &lt;=40% =0.4; 1 or 2 large competitors: &gt;50% SoM = 0.2; &gt;60% SoM = 0.1; &gt;75% SoM=0.</td>
<td>Number of distinct markets to be targeted in the three years from initial product release; 1 market=0; 2 markets=0.33 3 markets=0.66 more than 3 markets = 1</td>
</tr>
<tr>
<td>Metric:</td>
<td></td>
<td></td>
<td>Business plan contains accurate and detailed Industry, Market, Customer, Competition and Product Positioning analysis; score each category = 1, total is sum of categories divided by 5</td>
<td>Number of years company has existed since its registration; &lt;1 yr=0; 1yr=0.2; 2yr=0.4; 3 yr=0.6; 4yr=0.8; &gt;5yr=1</td>
<td>Discount rate that results in NPV=0 for all cash flows from and to investors</td>
</tr>
</tbody>
</table>
6.2 Appendix 2: The Final Data Set

The final data set used for training and testing the learning algorithms applied in this research is shown in this appendix 6.2. The company names have been anonymized, as the data is proprietary to the angel investor group and venture capital fund that provided access to the due diligence files of these companies. The process of defining, quantifying and extracting the attributes is described in chapter 4 and appendix 6 of this dissertation. This dataset is the result of the funnel process illustrated in figure 19.
<table>
<thead>
<tr>
<th>Company</th>
<th>EFF</th>
<th>GGD</th>
<th>MKT</th>
<th>EXP</th>
<th>OPL</th>
<th>INV</th>
<th>TCH</th>
<th>JKY</th>
<th>PTH</th>
<th>MAC</th>
<th>VPR</th>
<th>PPL</th>
<th>SCH</th>
<th>PRT</th>
<th>MGR</th>
<th>SOM</th>
<th>MIS</th>
<th>CMP</th>
<th>MOV</th>
<th>AGE</th>
<th>BMI Nominal</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>0.31</td>
<td>0.12</td>
<td>0.14</td>
<td>0.84</td>
<td>0.90</td>
<td>0.34</td>
<td>0.40</td>
<td>1.00</td>
<td>1.00</td>
<td>0.91</td>
<td>1.80</td>
<td>0.00</td>
<td>0.40</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.96</td>
<td>1.00</td>
<td>red</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>0.00</td>
<td>0.13</td>
<td>0.00</td>
<td>0.50</td>
<td>0.50</td>
<td>0.11</td>
<td>0.60</td>
<td>0.00</td>
<td>0.40</td>
<td>0.52</td>
<td>0.00</td>
<td>1.00</td>
<td>0.40</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.40</td>
<td>red</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OC</td>
<td>0.16</td>
<td>0.11</td>
<td>0.15</td>
<td>0.88</td>
<td>0.52</td>
<td>0.14</td>
<td>0.40</td>
<td>1.00</td>
<td>0.20</td>
<td>0.80</td>
<td>0.00</td>
<td>0.50</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.20</td>
<td>0.00</td>
<td>1.00</td>
<td>red</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CA</td>
<td>0.35</td>
<td>0.42</td>
<td>0.17</td>
<td>0.75</td>
<td>0.48</td>
<td>0.35</td>
<td>0.35</td>
<td>0.00</td>
<td>0.60</td>
<td>0.40</td>
<td>0.00</td>
<td>0.30</td>
<td>0.20</td>
<td>0.20</td>
<td>1.00</td>
<td>0.40</td>
<td>0.00</td>
<td>0.60</td>
<td>yellow</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WS</td>
<td>0.15</td>
<td>0.00</td>
<td>0.20</td>
<td>0.40</td>
<td>0.00</td>
<td>0.00</td>
<td>0.40</td>
<td>1.00</td>
<td>0.40</td>
<td>0.00</td>
<td>0.00</td>
<td>0.20</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>red</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ES</td>
<td>0.35</td>
<td>0.00</td>
<td>0.94</td>
<td>0.67</td>
<td>0.20</td>
<td>0.20</td>
<td>0.40</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.20</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>red</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Me</td>
<td>0.03</td>
<td>0.34</td>
<td>0.07</td>
<td>0.38</td>
<td>0.00</td>
<td>0.00</td>
<td>0.19</td>
<td>1.00</td>
<td>1.00</td>
<td>0.20</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.20</td>
<td>0.40</td>
<td>0.60</td>
<td>green</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI</td>
<td>0.00</td>
<td>0.21</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.45</td>
<td>green</td>
<td></td>
</tr>
<tr>
<td>Gl</td>
<td>0.05</td>
<td>0.40</td>
<td>0.29</td>
<td>0.16</td>
<td>0.50</td>
<td>0.60</td>
<td>0.19</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.20</td>
<td>0.40</td>
<td>1.00</td>
<td>1.00</td>
<td>0.80</td>
<td>0.00</td>
<td>1.00</td>
<td>0.30</td>
<td>0.60</td>
<td>0.60</td>
<td>green</td>
</tr>
<tr>
<td>Ae</td>
<td>0.22</td>
<td>0.21</td>
<td>0.35</td>
<td>0.08</td>
<td>0.33</td>
<td>0.20</td>
<td>0.12</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.40</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>yellow</td>
</tr>
<tr>
<td>Wa</td>
<td>0.17</td>
<td>0.09</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.20</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.40</td>
<td>yellow</td>
</tr>
<tr>
<td>Po</td>
<td>0.29</td>
<td>0.11</td>
<td>0.28</td>
<td>0.33</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.20</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.40</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>red</td>
</tr>
<tr>
<td>Nut1</td>
<td>0.13</td>
<td>0.26</td>
<td>0.13</td>
<td>0.33</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.10</td>
<td>0.30</td>
<td>0.10</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.20</td>
<td>green</td>
<td></td>
</tr>
<tr>
<td>Pys2</td>
<td>0.05</td>
<td>0.42</td>
<td>0.31</td>
<td>0.67</td>
<td>0.20</td>
<td>0.45</td>
<td>0.38</td>
<td>0.00</td>
<td>1.00</td>
<td>0.30</td>
<td>0.20</td>
<td>1.00</td>
<td>1.00</td>
<td>0.20</td>
<td>0.00</td>
<td>1.00</td>
<td>0.20</td>
<td>0.30</td>
<td>0.30</td>
<td>0.66</td>
<td>green</td>
</tr>
<tr>
<td>GFF</td>
<td>0.03</td>
<td>0.83</td>
<td>0.18</td>
<td>0.34</td>
<td>0.83</td>
<td>0.40</td>
<td>0.49</td>
<td>0.13</td>
<td>0.20</td>
<td>0.10</td>
<td>1.00</td>
<td>0.20</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.10</td>
<td>1.00</td>
<td>0.30</td>
<td>0.20</td>
<td>red</td>
</tr>
<tr>
<td>Via</td>
<td>0.45</td>
<td>0.32</td>
<td>0.07</td>
<td>0.38</td>
<td>0.33</td>
<td>0.40</td>
<td>0.20</td>
<td>0.05</td>
<td>0.20</td>
<td>0.00</td>
<td>0.40</td>
<td>0.60</td>
<td>0.00</td>
<td>0.00</td>
<td>0.40</td>
<td>0.00</td>
<td>1.00</td>
<td>0.10</td>
<td>0.30</td>
<td>0.40</td>
<td>red</td>
</tr>
<tr>
<td>AM</td>
<td>0.27</td>
<td>0.05</td>
<td>0.36</td>
<td>0.40</td>
<td>0.57</td>
<td>0.20</td>
<td>0.17</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.80</td>
<td>0.20</td>
<td>1.00</td>
<td>1.00</td>
<td>0.40</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.20</td>
<td>red</td>
</tr>
</tbody>
</table>
6.3 Appendix 3: 0.632 Bootstrap R Code:

The 0.632 Bootstrap algorithm was originally proposed in a paper by Efron (Efron, 1983). The algorithm is aimed at optimally balancing variance and bias in the validation of classifiers. The paper shows that 10-fold cross validation (CV), the most widely used validation technique in machine learning, effectively minimizes bias but can have high variance, especially when applied to small datasets. The following R code provides the functions and invocation code to perform 0.632 Bootstrap validation on two classifiers to compare their loss function values. The code also performs a paired t-test to determine statistical significance of the difference between the two classifiers.

The algorithm is based on a resampling process whereby the original dataset is randomly resampled with replacement to generate a new dataset of the same size as the original dataset but, as a result of resampling with replacement, with some instances omitted and some instances appearing more than once. The instances that are omitted become the test set. For each resampled dataset a new classifier is trained and its loss function is calculated using the test set. This process is repeated for a user-defined number of iterations (usually 50 or more). The loss function values from all iterations are then averaged and then added to the resubstitution loss (the loss function of the classifier using the original dataset as both the training set and the test set, also known as the training error) of the classifier as follows:
\[ L_c(x) = \frac{1}{n} \sum^n_l (0.368 \times L^{resub}_c(x) + 0.632 \times L^{boot}_c(x_i)) \]

Where:

\( L_c(x) \) = Value of the loss function of classifier \( c \) on dataset \( x \)

\( L^{resub}_c(x) \) = Value of the resubstitution loss function for classifier \( c \) on dataset \( x \)

\( L^{boot}_c(x_i) \) = Value of the loss function for classifier \( c \) on the \( i \)th bootstrapped dataset, \( x_i \)

\( n \) = number of bootstrap iterations

The code for invocation of the 0.632 Bootstrap algorithm prints the loss function values for each of the two classifiers and plots a histogram of the loss function distribution of each of the classifiers. The code also provides a t-value and p-value for the paired t-test using the loss function distributions of each classifier. The p-value of the paired t-test provides explicit measurement of the statistical significance of the difference in loss function distributions.

**R Code**

```r
# Function to calculate loss function from the confusion matrix generated by classifierEval

lossFunct=function(classifierEval) {
    Tr<as.numeric(substr(classifierEval$string,621,622))
    Fry<as.numeric(substr(classifierEval$string,623,624))
    Frg<as.numeric(substr(classifierEval$string,619,620))
    Fyr<as.numeric(substr(classifierEval$string,637,638))
    Ty<as.numeric(substr(classifierEval$string,639,640))
    Fyg<as.numeric(substr(classifierEval$string,635,636))
    Fgr<as.numeric(substr(classifierEval$string,602,603))
    ...}
```
Fgy<-as.numeric(substr(classifierEval$string,604,605))
Tg<-as.numeric(substr(classifierEval$string,600,601))
NumTest=Tr+Ty+Tg+Frg+Fgr+Fry+Fyr+Fgy+Fyg
L= (2*NumTest-(2*(Tr+Ty+Tg)-4*(Frg+Fgr)-(Fry+Fyr+Fgy+Fyg)))/NumTest
}

# e0Boot
e0Boot = function(iter, dataSet, setSize, dimension, classifier1, classifier2){
classifier1e0Boot<-numeric(iter)
classifier2e0Boot<-numeric(iter)
for(i in 1:iter){
  Subsamp<-sample(setSize, setSize, replace=TRUE)
  Basesamp<-1:setSize
  oneTrain<-dataSet[Subsamp, 1:dimension]
  oneTest<-dataSet[setdiff(Basesamp, Subsamp), 1:dimension]
  classifier1model<-classifier1(IRR_Nominal~., data=oneTrain)
  classifier2model<-classifier2(IRR_Nominal~., data=oneTrain)
  classifier1eval<-evaluate_Weka_classifier(classifier1model, newdata=oneTest)
  classifier1acc<-lossFunct(classifier1eval)
  classifier2eval<-evaluate_Weka_classifier(classifier2model, newdata=oneTest)
  classifier2acc<-lossFunct(classifier2eval)
  classifier1e0Boot[i]=classifier1acc
  classifier2e0Boot[i]=classifier2acc
}
return(rbind(classifier1e0Boot, classifier2e0Boot))
}

# e632 Bootstrap
e632Boot=function(iter, dataSet, setSize, dimension, classifier1, classifier2){
classifier1appModel<-classifier1(IRR_Nominal~., data=dataSet)
classifier1appEvaluation<-evaluate_Weka_classifier(classifier1appModel)
classifier1appAccuracy<-lossFunct(classifier1appEvaluation)
classifier1FirstTerm=.368*classifier1appAccuracy
classifier2appModel<-classifier2(IRR_Nominal~., data=dataSet)
classifier2appEvaluation<-evaluate_Weka_classifier(classifier2appModel)
classifier2appAccuracy<-lossFunct(classifier2appEvaluation)
classifier2FirstTerm=.368*classifier2appAccuracy
e0Terms=e0Boot(iter, dataSet, setSize, dimension, classifier1, classifier2)
classifier1e632Boot<-classifier1FirstTerm+.632*e0Terms[1,]
classifier2e632Boot<-classifier2FirstTerm+.632*e0Terms[2,]
return(rbind(classifier1e632Boot, classifier2e632Boot))
}

# Invocation of .632 Bootstrap for the Combined Data Set for SVM and Logistic classifiers
library(RWeka)
Logistic<-make_Weka_classifier("weka/classifiers/functions/Logistic")
SVM<-make_Weka_classifier("weka/classifiers/functions/SMO")
CombData<-read.csv("Combined_Data_Set.csv")
setSize=12
dimension=20
iterations=500
e632Bootstraps<-e632Boot(iterations, CombData, setSize, dimension, SVM, Logistic)
e632SVM<-mean(e632Bootstraps[,1])
e632Logistic<-mean(e632Bootstraps[,2])
e632SVM
e632Logistic
par(mfrow=c(2,1))
hist(e632Bootstraps[,1], breaks=c(0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0, 3.25, 3.5, 3.75,4.0), density = 20, col="red", main="Histogram of e632 for SVM", xlab= "Loss Function", xlim=c(0,3.5), ylim=c(0,150), labels=TRUE)
lines(c(mean(e632Bootstraps[,1]),mean(e632Bootstraps[,1])), c(0,150), lwd=3)
hist(e632Bootstraps[,2], breaks=c(0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0, 3.25, 3.5, 3.75,4.0), density = 20, col="blue", main="Histogram of e632 for Logistic", xlab="Loss Function", xlim=c(0,3.5), ylim=c(0,150), labels=TRUE)
lines(c(mean(e632Bootstraps[,2]),mean(e632Bootstraps[,2])), c(0,150), lwd=3)
t.test(e632Bootstraps[,1], e632Bootstraps[,2], paired=TRUE)
6.4 Appendix 4: 0.632+Bootstrap R Code:

The 0.632+Bootstrap algorithm was originally proposed by Efron and Tibshirani (Efron, 1997). It was developed to remove the inherent bias in the 0.632 Bootstrap algorithm that arises in the case of highly overfitted classifiers, such as the $k$ nearest neighbors algorithm (which will fit perfectly to the training data). Such algorithms have resubstitution error of zero. This means that even for a training set containing no information and with target attributes split evenly at 50% each (for a dichotomous target attribute), the original 0.632 Bootstrap will then yield an error of $0.632 \times 0.5$, which is clearly downward biased. The 0.632+Bootstrap algorithm overcomes this inherent bias for overfitted classifiers.

The code for calculating the 0.632+Bootstrap is shown below. This algorithm requires the calculation of both $\varphi'$ and $\widehat{R}'$ as described in section 4.1.1.1. These further require the use of $Q(y,r)$ as the loss function rather than $L_{c}(x)$. The necessary functions to calculate these parameters are included in the following R code.

```r
#Function to determine the resubstitution error of a classifier for a given dataset using Q(yi,rx(t)) as the error function
eResubw=function(dataSet, classifier){
    classifier1appModel<classifier(Irr_Nominal~., data=dataSet)
    classifier1appEvaluation<evaluate_Weka_classifier(classifier1appModel)
    classifier1appAccuracy<lossFunctw(classifier1appEvaluation)
    errResub<classifier1appAccuracy
    return(errResub)
}

#Function to calculate accuracy, Q, from the confusion matrix generated by classifierEval for the
lossFunctw=function(classifierEval) {
  Tr<-.numeric(substr(classifierEval$string,621,622))
  Fry<-.numeric(substr(classifierEval$string,623,624))
  Frg<-.numeric(substr(classifierEval$string,619,620))
  Fyr<-.numeric(substr(classifierEval$string,637,638))
  Ty<-.numeric(substr(classifierEval$string,639,640))
  Fyg<-.numeric(substr(classifierEval$string,635,636))
  Fgr<-.numeric(substr(classifierEval$string,602,603))
  Fgy<-.numeric(substr(classifierEval$string,604,605))
  Tg<-.numeric(substr(classifierEval$string,600,601))
  NumTest=Tr+Ty+Tg+Frg+Fgr+Fry+Fyr+Fgy+Fyg
  Q=1-(Tr+Tg+Ty)/NumTest
  return(Q)
}

#e0Boot for a single classifier for calculating rHatDash in the 0.632+Bootstrap algorithm, using Q(yi,rx(t))as the error function
e0BootSinglew = function(iter, dataSet, setSize, dimension, classifier){
  classifier1e0Boot<-numeric(iter)
  for(i in 1:iter){
    Subsamp<-sample(setSize, setSize, replace=TRUE)
    Basesamp<-1:setSize
    oneTrain<-dataSet[Subsamp, 1:dimension]
    oneTest<-dataSet[setdiff(Basesamp, Subsamp), 1:dimension]
    classifier1model<-classifier(IRR_Nominal~., data=oneTrain)
    classifier1eval<-evaluate_Weka_classifier(classifier1model, newdata=oneTest)
    classifier1acc<-lossFunctw(classifier1eval)
    classifier1e0Boot[i]=classifier1acc
  }
  return(classifier1e0Boot)
}

#Function to calculate no-information error rate, gammaHat, for a given classifier, as defined in [Efron, 1997]
gammaHat<-.function(dataSet, setSize, dimension, classifier){
  appModel<-classifier(IRR_Nominal~., data=dataSet)
  Result<-numeric(setSize^2)
  k=1
  Q=0
  for(i in 1:setSize){
    for(j in 1:setSize){
      t<-.dataSet[i,]
      tj<-.dataSet[j,dimension] #remove the target attribute
      IRR_Nominal<-.dataSet[i,dimension]
      #add the ith target attribute to the jth predictor attributes
      Result[k]<-1-Q
      k=k+1
    }
  }
  return(Result/2)
}
permData<-cbind(tj, IRR_Nominal)
appEvaluation<-evaluate_Weka_classifier(appModel,
newdata=permData)
appAccuracy<-modlossFunct(appEvaluation)
Result[k]<-appAccuracy
k=k+1
if(appAccuracy==0) Q<Q+1
}
gammaHat<-Q/setSize^2
return(gammaHat)
}

#Modified loss function to find the proportion of correctly classified instances (used to calculate Q in gammaHat)

modlossFunct=function(classifierEval) {
  Tr<-as.numeric(substr(classifierEval$string,621,622))
  Fry<-as.numeric(substr(classifierEval$string,623,624))
  Frg<-as.numeric(substr(classifierEval$string,619,620))
  Fyr<-as.numeric(substr(classifierEval$string,637,638))
  Ty<-as.numeric(substr(classifierEval$string,639,640))
  Fyg<-as.numeric(substr(classifierEval$string,635,636))
  Fgr<-as.numeric(substr(classifierEval$string,602,603))
  Fgy<-as.numeric(substr(classifierEval$string,604,605))
  Tg<-as.numeric(substr(classifierEval$string,600,601))
  T=Tr+Ty+Tg
  return(T)
}

#Function to calculate rHatDash for 0.632+Bootstrap as defined in [Efron, 1997]

rHatDash<-function(iter, dataSet, setSize, dimension, classifier){
  gHat<-gammaHat(dataSet, setSize, dimension, classifier)
  errBoot<-mean(e0BootSingle(iter, dataSet, setSize, dimension, classifier))
  errResub<-eResubW(dataSet, classifier)
  errBootDash<-min(errBoot,gHat)
  if (gHat>errResub) if(errBoot>errResub) rHatDash<-(errBootDash-errResub)/(gHat-
  errResub) else rHatDash=0
  return(rHatDash)
}

#e632 Bootstrap for a single classifier using the loss function

e632BootSingle=function(iter, dataSet, setSize, dimension, classifier){
  classifier1appModel<-classifier(IRR_Nominal~., data=dataSet)
  classifier1appEvaluation<-evaluate_Weka_classifier(classifier1appModel)
  classifier1appAccuracy<-modlossFunct(classifier1appEvaluation)
  classifier1FirstTerm=.368*classifier1appAccuracy
  e0Terms=e0BootSingle(iter, dataSet, setSize, dimension, classifier)
}
classifier1e632Boot<e1classifier1FirstTerm+.632*e0Terms
return(classifier1e632Boot)
}

#e0Boot for a single classifier

e0BootSingle = function(iter, dataSet, setSize, dimension, classifier){
    classifier1e0Boot<e0numeric(iter)
    for(i in 1:iter){
        Subsamp<e0sample(setSize, setSize, replace=TRUE)
        Basesamp<e01:setSize
        oneTrain<e0dataSet[Subsamp, 1:dimension]
        oneTest<e0dataSet[setdiff(Basesamp, Subsamp), 1:dimension]
        classifier1model<e0classifier(IRR_Nominal~., data=oneTrain)
        classifier1eval<e0evaluate_Weka_classifier(classifier1model, newdata=oneTest)
        classifier1acc<e0lossFunct(classifier1eval)
        classifier1e0Boot[i]=classifier1acc
    }
    return(classifier1e0Boot)
}

#e632plus Bootstrap function for a single classifier using the loss function

e632BootPlus<e0function(iter, dataSet, setSize, dimension, classifier){
    #Calculate rHatDash and gammaHat
    rHDash<e0rHatDash(iter, dataset, setSize, dimension, classifier)
    gHat<e0gammaHat(dataSet, setSize, dimension, classifier)

    #Calculate terms of the e632plus formula
    appModel<e0classifier(IRR_Nominal~., data=dataSet)
    appEvaluation<e0evaluate_Weka_classifier(appModel)
    appAccuracy<e0lossFunct(appEvaluation)
    eBar=appAccuracy
    eBootTerms<e0e0BootSingle(iter, dataSet, setSize, dimension, classifier)
    eBoot<e0mean(eBootTerms)
    eBootDash<e0min(eBoot,gHat)
    e632Terms<e0e632BootSingle(iter, dataSet, setSize, dimension, classifier)
    e632<e0mean(e632Terms)

    #Calculate 0.632+Bootstrap loss function
    e632BootPlus<e0e632+(eBootDash-eBar)*0.368*0.632*rHDash/(1-0.368*rHDash)
    return(e632BootPlus)
}

#Invocation of 0.632+Bootstrap for IBk classifier
#Functions required: lossFunct, lossFunctw; modlossFunct, gammaHat, rHatDash, e0BootSingle, e632BootSingle, e0BootSinglew, eResub, e632BootPlus

#Initialize
library(RWeka)
IBk<-make_Weka_classifier("weka/classifiers/lazy/IBk")
classifier<-IBk
dataset<-read.csv("Combined_Data_Set.csv") #Assumes that .csv file is in home directory
setSize=12
dimension=20
iter=50

#Calculate loss function using 0.632+Bootstrap
err632BootPlus<-e632BootPlus(iter, dataset, setSize, dimension, classifier)
print(err632BootPlus)
6.5 Appendix 5: IBk Classifier Permutation Test R Code:

The following R code contains the necessary functions and invocation code to perform a t-test on the IBk classifier selected for stage 2 of the IDSS. The t-test retains the marginal distributions of predictor and target attributes by randomly permuting only the predictor attributes of instances but keeping the target attributes the same as the original dataset’s target attributes. In this way the predictor attributes and the target attributes are effectively independent, simulating the null hypothesis that the IBk classifier trained on the original dataset does not have a statistically significant lower loss function than an IBk classifier trained on random data with the same marginal distributions as the original dataset.

The t-test code performs 200 iterations (as recommended in Japkowicz, 2011), in each of which an IBk classifier trained on permuted instances is validated using a 50-iteration (as recommended in Efron, 1997) 0.632+Bootstrap validation algorithm. The resultant loss function of each of these classifiers is compared to the loss function of an IBk classifier trained on the original dataset. If the loss function of a classifier trained on the permuted dataset is lower than that of the classifier trained on the original dataset, a counter is incremented. At completion of the 200 iterations, the counter value is divided by the number of iterations to yield the probability of the loss function of a classifier trained on permuted data being lower than that of a classifier trained on the original dataset (the p-value). If this probability is lower
than 1%, representing strong statistical significance, then the null hypothesis can be safely discarded.

**R Code:**

```r
# e0Boot for a single classifier

e0BootSingle = function(iter, dataSet, setSize, dimension, classifier){
  classifier1e0Boot<-numeric(iter)
  for(i in 1:iter){
    Subsamp<-sample(setSize, setSize, replace=TRUE)
    Basesamp<-1:setSize
    oneTrain<-dataSet[Subsamp, 1:dimension]
    oneTest<-dataSet[setdiff(Basesamp, Subsamp), 1:dimension]
    classifier1model<-classifier(IRR_Nominal~., data=oneTrain)
    classifier1eval<-evaluate_Weka_classifier(classifier1model, newdata=oneTest)
    classifier1acc<-lossFunct(classifier1eval)
    classifier1e0Boot[i]=classifier1acc
  }
  return(classifier1e0Boot)
}

# e632 Bootstrap for a single classifier

#function to calculate loss function from the confusion matrix generated by classifierEval

#This function extracts the result data by counting characters in the results file; it may need to be adjusted if the results file contains additional characters than used to build this function

lossFunct=function(classifierEval) {
  Tr<-as.numeric(substr(classifierEval$string,621,622))
  Fry<-as.numeric(substr(classifierEval$string,623,624))
  Frg<-as.numeric(substr(classifierEval$string,619,620))
  Fyr<-as.numeric(substr(classifierEval$string,637,638))
  Ty<-as.numeric(substr(classifierEval$string,639,640))
  Fyg<-as.numeric(substr(classifierEval$string,635,636))
  Fgr<-as.numeric(substr(classifierEval$string,602,603))
  Fgy<-as.numeric(substr(classifierEval$string,604,605))
  Tg<-as.numeric(substr(classifierEval$string,600,601))
  return(classifier1e0Boot)
}
```

157
NumTest=Tr+Ty+Tg+Frg+Fgr+Fry+Fyr+Fgy+Fyg
L= (2*NumTest-(2*(Tr+Ty+Tg)-4*(Frg+Fgr)-(Fry+Fyr+Fgy+Fyg)))/NumTest

# Function to shuffle the first dimension-1 attributes and keep the target; assumes target is the last attribute in the dataset
shuffle<-function(dimension, setSize, dataSet){
  Subsamp<-sample(setSize, setSize, replace=FALSE)
  newData<-dataSet[Subsamp, 1:dimension-1]
  IRR_Nominal<-dataSet[,dimension]
  shufData<-cbind(newData, IRR_Nominal)
  return(shufData)
}

# Invocation of a t-test on a single IBk classifier using .632 Bootstrap validation
# Functions required: lossFunct; e0BootSingle; e632BootSingle; shuffle

# Initialize
library(RWeka)
IBk<-make_Weka_classifier("weka/classifiers/lazy/IBk")
# Assumes that .csv file is in home directory
CombData<-read.csv("Combined_Data_Set.csv")

setSize=12
dimension=20
iterationsTtest=200
iterationsBoot=50  # From Efron, Tibshirani, “Improvements on Cross Validation: the 0.632+Bootstrap algorithm”
count=0
Avg=numeric(iterationsTtest)

# Calculate the true IBk loss function
e632IBkRes<- e632BootSingle(iterationsBoot, CombData, setSize, dimension, IBk)
e632IBk<-mean(e632IBkRes)

# Calculate iterationsTtest loss functions using e632Bootstrap validation for randomly permuted data and count the number of times the result is better than the true IBk loss function
for(j in 1:iterationsTtest){
  ShufData<-shuffle(dimension, setSize, CombData)
e632Bootstraps<-e632BootSingle(iterationsBoot, ShufData, setSize, dimension, IBk)
e632IBkShuf<-mean(e632Bootstraps)
  Avg[j]<-e632IBkShuf
  if(e632IBkShuf<e632IBk) count=count+1
}
p=count/iterationsTtest
print("The probability of an IBk classifier trained on randomized data achieving the same results as the IBk classifier trained on the dataset is: ")
print(p)
7. References


72. Steiner, C., “Meet the Fastest Growing Company Ever”, Forbes, December 8, 2010


78. Robert E. Wiltbank “Siding with the Angels – Business Angel Investing: promising outcomes and effective strategies” NESTA Research Report, April 2009
