PREDICTION OF COST OVERRUNS USING ENSEMBLE METHODS

IN

DATA MINING AND TEXT MINING ALGORITHMS

By

PRATHIKSHA RAMESH

A thesis submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Civil and Environmental Engineering

written under the direction of

Dr Trefor Williams

and approved by

_____

_____

_____

New Brunswick, New Jersey

January 2014

ABSTRACT OF THE THESIS

PREDICTION OF COST OVERRUNS USING ENSEMBLE METHODS IN DATA MINING AND TEXT MINING ALGORITHMS

By **PRATHIKSHA RAMESH**

Thesis Director **PROFESSOR TREFOR WILLIAMS**

Abstract: In competitive bidding in the United States, the lowest bid is most often than not selected to perform the project. However, the lowest bidder tends to undervalue the costs in order to win the bid and as a result may incur significant cost increases during the construction life cycle due to change orders. For project owners to accurately estimate the actual project cost and to predict the bid that is close to the actual project, there is an urgent need for new decision aids to analyze the bidding patterns.

The goal of this research has been to select the predictive features in a bid package to help minimize the cost overruns with the help of open source data mining software. The features were selected based on correlation and regression analysis by studying the p-values and r-squared values. The data set was then prepared with only the features that were affecting the output, which in our case were the cost overruns. The output is divided into 4 classes depending on the percentage of overrun. The learning algorithms used for prediction were neural networks, support vector machines, decision trees along with the ensemble methods. The empirical study of the prediction models suggest

an efficiency of up to 50% in predicting whether a project will have cost overruns and what is the approximate range of percentage overrun.

Acknowledgments

I'm really grateful to my thesis advisor, Prof. Williams for his incredible direction. He has guided and helped me right from the beginning with his intellect, support and encouragement. I thank him for the all the insightful talks and for being so extremely patient while I learnt the ropes.

I want to extend my gratitude to my Master's Thesis Committee members, Prof. Jie Gong and Prof. Eric Gonzales.

I would like to thank my roommates here at Rutgers who have become my second family, for putting up with all those late nights. My friends back home in India for always encouraging me to push harder.

I cannot express how thankful I'm to my parents and loved ones, for always supporting me and believing that I am capable of doing the best.

Dedication

To my family, friends and loved ones.

Table of Contents

List of Illustrations

List of Tables

1. INTRODUCTION

Cost overruns are a very common problem in the construction industry. Many factors such as contract duration, project size, bid volume, regulation, completeness of the plans and the contractors management expertise all exert an influence on the contractors initial bid prices and affect the outcome of the completed construction project cost [1]. There are also many external sources contributing to cost overruns ranging from construction changes to differing site conditions. One of the main causes of cost overruns is the intensely competitve market in the construction industry. As required by law, government agencies have traditionally used the policy that the construction contracts are competitvely bid and must be awarded to the lowest possible bidder. Thus many bidders tend to submit bids that are lower than the actual project costs in order to outplay competitors and win the bid. As a result, competitively bid construction projects are often completed at a much greater cost than the original low bid amount. In addition, because of the lowest-bid price policy, the project owner may continually work at reducing their costs [2].

The data-mining framework proposed uses ensemble methods like boosting, bagging and stacking. The learning algorithms include neural networks, decision tree and support vector machines. The input features for the prediction models are the bidding patterns characterized by bidding ratios, proposed by [3]. Real highway bid data from the department of transportation (DOT) of California and Washington are used to test the accuracy of conventional and data mining based bid selection policies.

The main objective of this research was to experiment with the use of prediction models based on data and text mining to study and determine if they can be used as an alternative method of bid selection. The focus was on two different types models to select the optimum project bidder. The first model is a classification model using Rank 1 Bid and Engineer's Estimate for every project and predicting the output class. The second model uses only the engineer's estimate and the delay time in finishing the project to predict the cost overrun.

In this work, the development of a prediction model in RapidMiner using ensemble methods is studied. The study is divided into:

- Collecting the data sets from bid express online and cleaning the data.
- Identification of features that are statistically significant using correlation and regression analysis.
- Using ensemble methods with different classification algorithms to build prediction models.
- Testing models of trained data on testing data.

a. Literature Study

There have been several applications of data and text mining to construction management problems. Existing construction mining research has focused on methods of classifying documents and extracting information from databases. A prototype system that automatically classifies construction documents according to project

components using data mining techniques was proposed by Caldas et al. [4]. Soibelman and Kim [5] addressed the need for data mining in the construction industry, and the possibility to identify predictable patterns in construction data that were previously thought to be chaotic. In that study, a prototype knowledge discovery and data mining (KDD) system was developed to find the cause of activity delays from a U.S. Army Corps of Engineer's database called the Resident Management System. Soibelman et al. [6] have addressed the need to develop additional frameworks that allows the development of data warehouses from complex construction unstructured data and to develop data modeling techniques to analyze common construction data types. Various modeling techniques have been applied to the prediction of construction costs. They have usually focused on the use of numeric data to predict the projects outcome. Recent work has employed advanced data mining techniques to produce predictions. Son et al. [7] have developed a model using Principal Component Analysis and Support Vector Regression using 64 project definition variables to predict cost performance on building projects. Gritza and Labi [8] have applied econometric models to the analysis of highway project cost overruns. They found that for a given project type and project duration, contracts of larger size or longer duration are generally more likely to incur cost overruns. Regression analysis and neural networks have also been applied to predicting construction costs [9, 10, 11, 12, 13 and 14]. Potentially, the addition of text data to various modeling techniques can enhance the predictions made by these models by covering more of the factors that can affect construction performance than can be derived from numeric data only.

Text summaries of what is to be constructed for a particular project, and textual descriptions of project line items are available from project bidding data that is collected by state transportation agencies. Additionally, numeric data are available at the time of the bid opening including the project magnitude, and the number of bidders [15].

Several data mining frameworks including only numerical and only text based approaches have been studied to compare the results.

Seminal work like [16] used neural network classification and regression models [21] to predict the outcome. Several indicators of the nature of the submitted bids were studied

- Low bid
- Standard deviation
- Median bid
- The number of submitted bids
- Estimated project duration

The data were split in this way to maximize the cases available to train and validate the neural network while still providing sufficient cases to provide an independent test of the network's performance. This test set corresponded to the data used to test the regression models.

The training set is used to train the neural network and the validation set is used to measure model performance during training. The validation data are a proportion of the training data that are not used to build the neural network model. The error in the validation data is measured at frequent intervals during the training cycle. The optimal neural network model is the one that has the lowest validation error. Two neural network models, PNN (probabilistic neural network) and the GRNN (generalized regression neural network) are used in this study [2]. For neural network classification, the bid selection is modeled as a classification problem, which is to classify the best bid to be the lowest bid or the second-lowest bid. For neural network regression, the bid selection is modeled as a regression problem, which is to predict the optimal rank of the best bid.

Other data mining works, such as [17] use tree map visualization to find factors that contribute to cost overruns in highway projects. Tree-maps were produced in two focus areas. First, tree-maps that aid in the identification of project cost overruns will be described. Second, tree-maps that relate the concentration of cost in a limited number of line items are studied. To create the tree-maps the Tree-map software developed by the University of Maryland Human Computer Interaction Laboratory [18] has been used. However it was found that it is difficult to find a strong indicator for the potential of cost overruns on competitively bid projects. Because of the great variability in project outcomes, the tree-maps did not identify dramatic indicators of bidding trends.

In this study, like in [19] we have classified the output (cost overruns) into 4 classes depending on the percentage of overruns. This gives an idea of what range the projects come under and also helps in building a better classification model. The ensemble methods used produce better efficiency in the prediction models (75-85%) as compared to the neural network classification and regression methods. Another major advantage is the simplicity of the process and implementation. The resources and memory requirements to run the software [20] is minimal as well.

2. DATA ACQUISITION AND CLEANSING

Bid data like the number of bidders, contract id, textual information, all bid submitted and the actual completed project cost is obtained from the department of transportation website for California and Washington states.

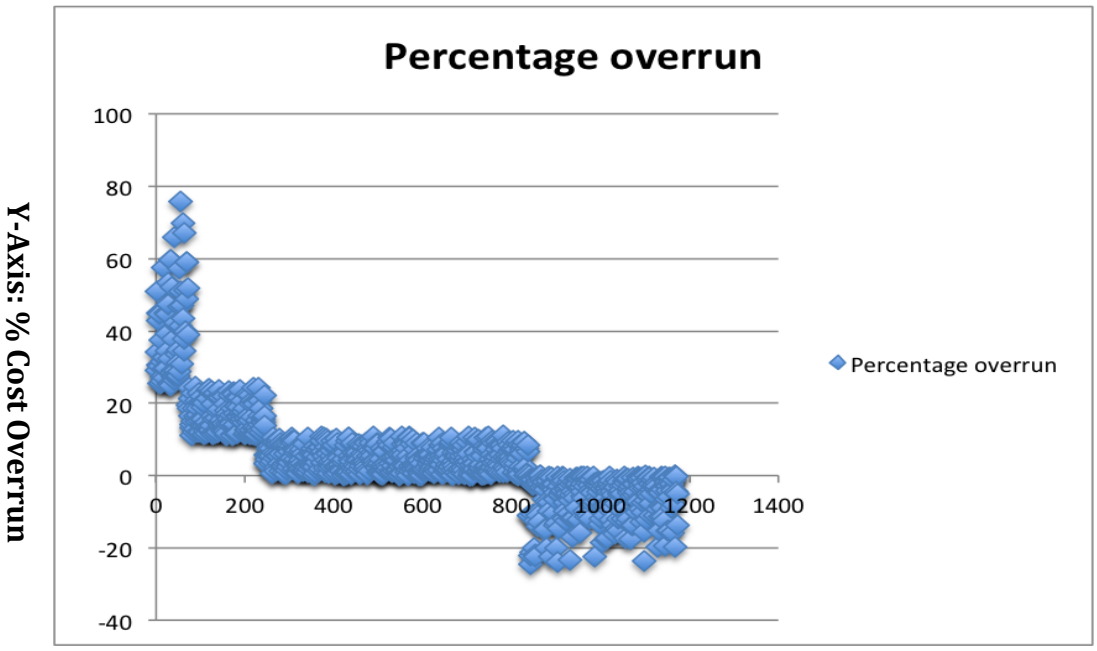
Table 2.1: Characteristics of the Datasets

Data Characteristics	California Raw Data	California Cleansed Data	Washington Raw Data	Washington Cleansed Data
Number of Projects	1201	1174	3147	2355
Number of Bids Received	4849	4608	14469	9286

The datasets contained several anomalous cases. There were several projects with extremely large overruns or under-runs. This variation was resulting in skewed graphs while plotting the data on a scatter plot. The data was cleansed to remove such projects

and provide a more uniform sampling data. The criterion to determine the percentage overrun is the ratio of the difference in cost between completed and lowest bid cost to the lowest bid.

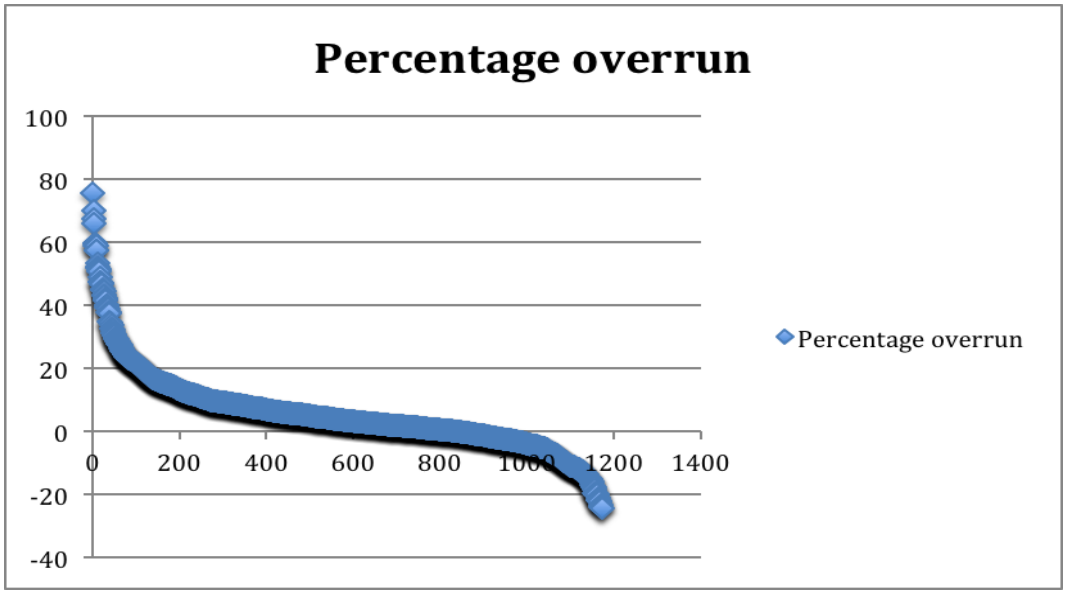
Figure 2.1: Raw Dataset



Y-Axis: % Cost Overrun

X-Axis: Project ID

Figure 2.2: Cleansed Data



Y-Axis: % Cost Overrun

X-Axis: Project ID

3. STATISTICAL ANALYSIS

There are a lot of variables included in the original bid documents acquired from the bidding express website. However to build an optimum model it is important to use only the features which are significant in predicting the output. Statistical analysis is particularly useful when there is noisy data like ours, which has quite a number of anomalous project costs.

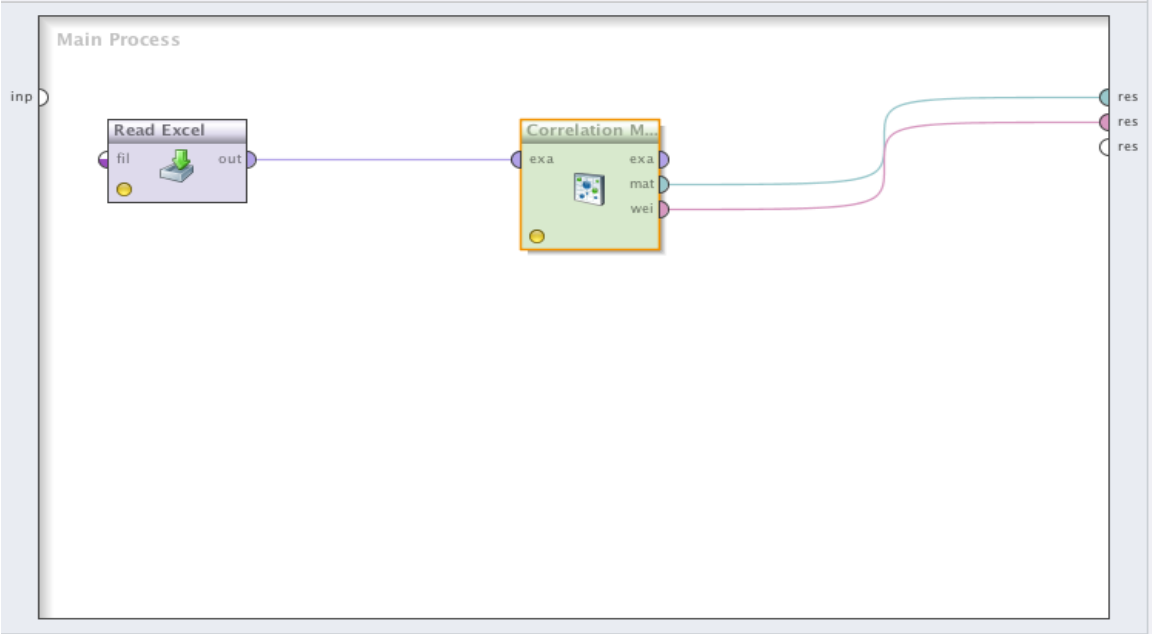
The statistical analysis methods employed in this study include correlation and regression analysis.

a. Correlation Value

Correlation refers to the statistical dependence relationship between any two sets of random variables. In this study, we are determining the Pearson correlation coefficient (p) that is sensitive only to a linear relationship between the two variables. Values of the correlation coefficient are always between -1 and +1. A correlation coefficient of +1 indicates that two variables are perfectly related in a positive linear sense, a correlation coefficient of -1 indicates that two variables are perfectly related in a negative linear sense, and a correlation coefficient of 0 indicates that there is no linear relationship between the two variables [14].

The correlation is tested between the different attributes (feature variables) and the output (cost overruns). The dataset (excel sheet) is imported into the RapidMiner process window and the correlation matrix function is connected to the dataset.

Figure 3.1: RapidMiner Process Window



i. California Dataset Correlation

The different variables provided in the original bid document are

- Project ID
- Number of items in the bid
- Number of Bids
- Rank 1 Bid
- Top 5 Bid Average
- Bid Rand 1 vs. Average Bid Percentage Difference
- Engineer's Estimate
- Amount Over
- Percentage Overrun

The excel data sheet is imported as a “Read Excel” operator in RapidMiner. The “Correlation Matrix” is then connected and the process is executed. The result is a correlation confusion matrix, which is symmetric about its diagonals. The correlation of a variable with itself is always 1.

ii. Washington Dataset Correlation

The items in the original bid package are

- Contract Status
- Contract Number
- Federal aid number
- State Route
- Contract Title
- Contract Name
- Engineer’s Estimate
- Bid Amount
- Amount Paid

The correlation matrices for the two datasets are shown below.

Figure 3.2: Correlation Matrix for California Dataset

Attributes	Number of...	Number of...	Log(Low Bid)	Rank 1 Bid	Top 5 Bid ...	Bid rank 1...	Engineer's ...	Amount Over	Ratio of Lo...	Bid Rank 1...	Reciprocal...	Ratio Over...	% Overrun
Number of Bids	1	-0.053	-0.083	-0.114	-0.113	0.471	-0.111	0.038	-0.401	0.393	0.214	-0.012	0.030
Number of Number of	-0.053	1	0.667	0.284	0.287	-0.089	0.269	0.140	0.128	-0.118	-0.380	0.462	-0.031
Log(Low Bid)	-0.083	0.667	1	0.511	0.513	-0.235	0.503	-0.220	0.166	-0.173	-0.604	0.148	0.036
Rank 1 Bid	-0.114	0.284	0.511	1	1.000	-0.047	1.000	-0.801	-0.033	0.008	-0.126	-0.033	0.151
Top 5 Bid Avg	-0.113	0.287	0.513	1.000	1	-0.046	1.000	-0.798	-0.033	0.008	-0.126	-0.032	0.151
Bid rank 1 vs Avg Bid % Diff	0.471	-0.089	-0.235	-0.047	-0.046	1	-0.046	-0.005	-0.377	0.481	0.587	-0.040	0.127
Engineer's Estimate	-0.111	0.269	0.503	1.000	1.000	-0.046	1	-0.815	-0.040	0.016	-0.122	-0.038	0.151
Amount Over	0.038	0.140	-0.220	-0.801	-0.798	-0.005	-0.815	1	0.193	-0.202	0.020	0.148	-0.111
Ratio of Low Bid to EE	-0.401	0.128	0.166	-0.033	-0.033	-0.377	-0.040	0.193	1	-0.890	-0.233	-0.008	0.397
Bid Rank 1 vs EE % Diff	0.393	-0.118	-0.173	0.008	0.008	0.481	0.016	-0.202	-0.890	1	0.273	-0.023	-0.224
Reciprocal Values	0.214	-0.380	-0.604	-0.126	-0.126	0.587	-0.122	0.020	-0.233	0.273	1	-0.167	0.118
Ratio Overrun	-0.012	0.462	0.148	-0.033	-0.032	-0.040	-0.038	0.148	-0.008	-0.023	-0.167	1	-0.307
% Overrun	0.030	-0.031	0.036	0.151	0.151	0.127	0.151	-0.111	0.397	-0.224	0.118	-0.307	1

Figure 3.3: Correlation Matrix for Washington Dataset

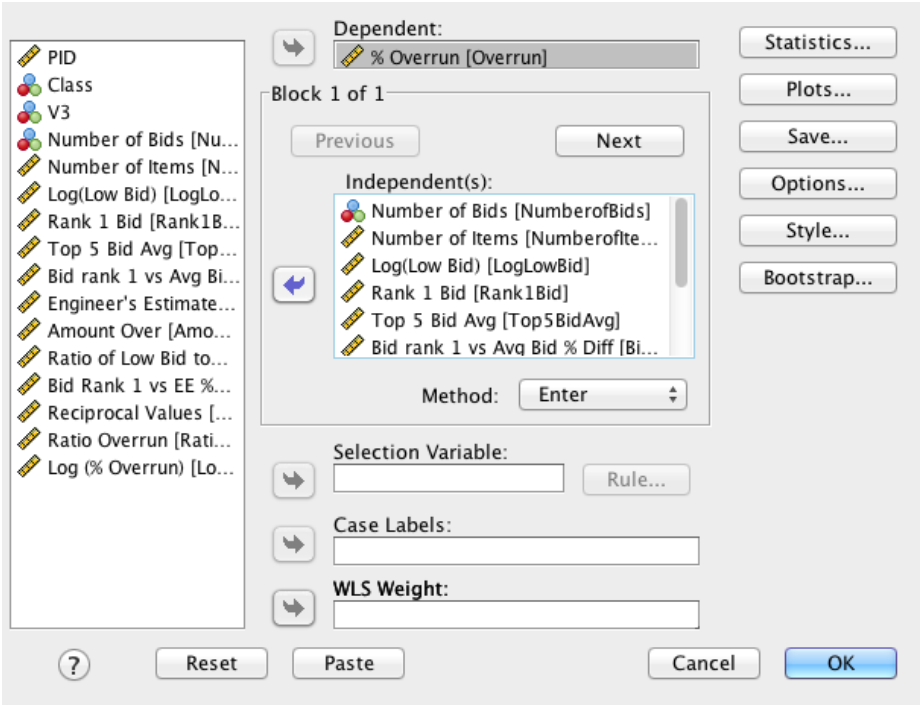
Attributes	PID	Engineer's Estimate	Bid Amount	Amount Paid	Diff b/w bid and actual	Original Working Day...	Authorized Days Co...	Remaining Working Days Count
PID	1	0.034	0.032	0.029	-0.015	-0.108	-0.121	0.136
Engineer's Estimate	0.034	1	0.998	0.995	0.479	0.634	0.617	0.461
Bid Amount	0.032	0.998	1	0.997	0.476	0.624	0.607	0.467
Amount Paid	0.029	0.995	0.997	1	0.549	0.642	0.628	0.452
Diff b/w bid and actual	-0.015	0.479	0.476	0.549	1	0.503	0.525	0.081
Original Working Days Count	-0.108	0.634	0.624	0.642	0.503	1	0.984	0.146
Authorized Days Count	-0.121	0.617	0.607	0.628	0.525	0.984	1	0.120
Remaining Working Days Count	0.136	0.461	0.467	0.452	0.081	0.146	0.120	1

b. Regression Value

Regression analysis is a statistical process for estimating the relationships among variables. It helps to understand how the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held fixed [14].

The statistical analysis software used is the IBM SPSS Statistic package. Regression method used in this study is the linear regression. Where the dependent variable is the output (cost overrun) and the independent variable are the different attributes or features present in the bid documents.

Figure 3.4: SPSS Process Window



i. California Dataset

For the regression, the dataset is first imported into SPSS in the form of an excel sheet. The data is then analyzed for linear regression.

- Independent Variable : Percentage Overrun
- Dependent Variables: Number of Bids, Number of Items, Rank 1 Bid, Top 5 Bid Avg, Bid Rank1 vs. Avg Bid Percentage Difference, Engineer's Estimate, Amount Over.

Once the dataset is imported and the analysis is run. It gives the following results.

Figure 3.5: Model Summary for California Dataset

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.598[a]	.358	.341	12.37231

a. Predictors: (Constant), Bid Rank 1 vs EE % Diff, Rank 1 Bid, Number of Items, Number of Bids, Bid rank 1 vs Avg Bid % Diff, Log(Low Bid), Ratio of Low Bid to EE, Amount Over

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	25622.454	8	3202.807	20.923	.000[b]
	Residual	45922.237	300	153.074		
	Total	71544.691	308			

a. Dependent Variable: % Overrun

b. Predictors: (Constant), Bid Rank 1 vs EE % Diff, Rank 1 Bid, Number of Items, Number of Bids, Bid rank 1 vs Avg Bid % Diff, Log(Low Bid), Ratio of Low Bid to EE, Amount Over

The ANOVA table reports a significant f statistic indicating that using the model is better than guessing the mean. A good R square value indicates that there is a good chance of predicting the dependent variable given the independent variables.

Even though the model fit looks positive, the first section of the coefficients table shows that there are far too many predictors for the model. There are several non-significant coefficients, indicating that these variables do not contribute much to the model.

Table 3.1: Coefficients Table for California Dataset

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-28.903	10.905		-2.651	.008
	Number of Bids	1.680	.523	.171	3.215	.001
	Number of Items	.028	.025	.096	1.126	.261
	Top 5 Bid Avg	1.298E-8	.000	.151	1.347	.179
	Log(Low Bid)	-.645	.718	-.075	-.899	.369
	Bid rank 1 vs Avg Bid % Diff	.134	.063	.141	2.119	.035
	Amount Over	-1.823E-7	.000	-.079	-.750	.454
	Ratio of Low Bid to EE	38.258	4.178	.930	9.158	.000
	Bid Rank 1 vs EE % Diff	.237	.060	.425	3.929	.000
	Reciprocal Values	491616.608	555114.896	.063	.886	.377
	Ratio Overrun	-5.509	1.016	-.288	-5.423	.000

a. Dependent Variable: % Overrun

Table 3.2: Excluded Coefficients for California Dataset

Excluded Variables[a]

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics
						Tolerance
1	Rank 1 Bid	7.535[b]	1.061	.290	.061	3.866E-5
	Engineer's Estimate	7.784[b]	1.061	.290	.061	3.623E-5

a. Dependent Variable: % Overrun

b. Predictors in the Model: (Constant), Ratio Overrun, Ratio of Low Bid to EE, Top 5 Bid Avg, Reciprocal Values, Number of Bids, Number of Items, Bid rank 1 vs Avg Bid % Diff, Log(Low Bid), Amount Over, Bid Rank 1 vs EE % Diff

ii. Washington Dataset

- Dependent Variable: Difference between Actual Cost and Bid Cost.
- Independent Variables: Engineer's Estimate, Bid Amount, Original Working Days Count, Authorized Days Count, Remaining Working Days Count.

The results are as follows.

The R square value of 0.069 indicates that there could be more variables that are acting on the outcome. There are additional features that need to be considered. Since there are no excluded coefficients we will consider all the variables present for the model attributes

Figure 3.6: Model Summary for Washington Dataset

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.262[a]	.069	.067	11.57685304

a. Predictors: (Constant), Remaining Working Days Count, Engineer's Estimate, Authorized Days Count, Original Working Days Count, Bid Amount

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	22039.515	5	4407.903	32.889	.000[b]
	Residual	298068.323	2224	134.024		
	Total	320107.838	2229			

a. Dependent Variable: Percentage Overrun

b. Predictors: (Constant), Remaining Working Days Count, Engineer's Estimate, Authorized Days Count, Original Working Days Count, Bid Amount

Table 3.3: Coefficients Table for Washington Dataset

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	6.325	.363		17.428	.000
	Engineer's Estimate	-4.290E-7	.000	-.068	-.656	.512
	Bid Amount	-5.251E-7	.000	-.078	-.758	.448
	Original Working Days Count	-.102	.013	-.614	-7.775	.000
	Authorized Days Count	.127	.012	.840	10.755	.000
	Remaining Working Days Count	.005	.006	.016	.786	.432

a. Dependent Variable: Percentage Overrun

After studying the results of both statistical analyses, the most statistically significant features were chosen to build the prediction model.

4. MODEL ATTRIBUTES

After the data cleansing and statistical analysis, the dataset is put into the form of an excel sheet to prepare for the modeling.

The final attributes for the California Dataset are

- Number of Bidders
- Rank 1 Bid
- Ratio of Rank 1 Bid to Engineer's Estimate

- Engineer's Estimate
- Text Information from Bid Documents

The percentage of overrun is classified into 4 classes for the modeling.

Table 4.1: Cost Overruns Classes

Class	Percentage Overrun
1	< 0%
2	0-11%
3	11-25%
4	>25%

The final attributes for the Washington Dataset are

- Engineer's Estimate
- Original Working Days Count
- Authorized Days Count
- Remaining Working Days Count
- Text Information from Bid Documents

The percentage overruns are classified the same as the California Dataset.

5. MODELING PROCESS

a. Computer Software Employed

RapidMiner is an open-source environment for machine learning, data mining, text mining, predictive analysis and business analytics [20]. RapidMiner provides a GUI to design an analytical pipeline (the "operator tree"). The GUI generates an XML (extensible markup language) file that defines the analytical processes the user wishes to apply to the data. Alternatively, the engine can be called from other programs or used as an API. Individual functions can be called from the command line.

b. The Model

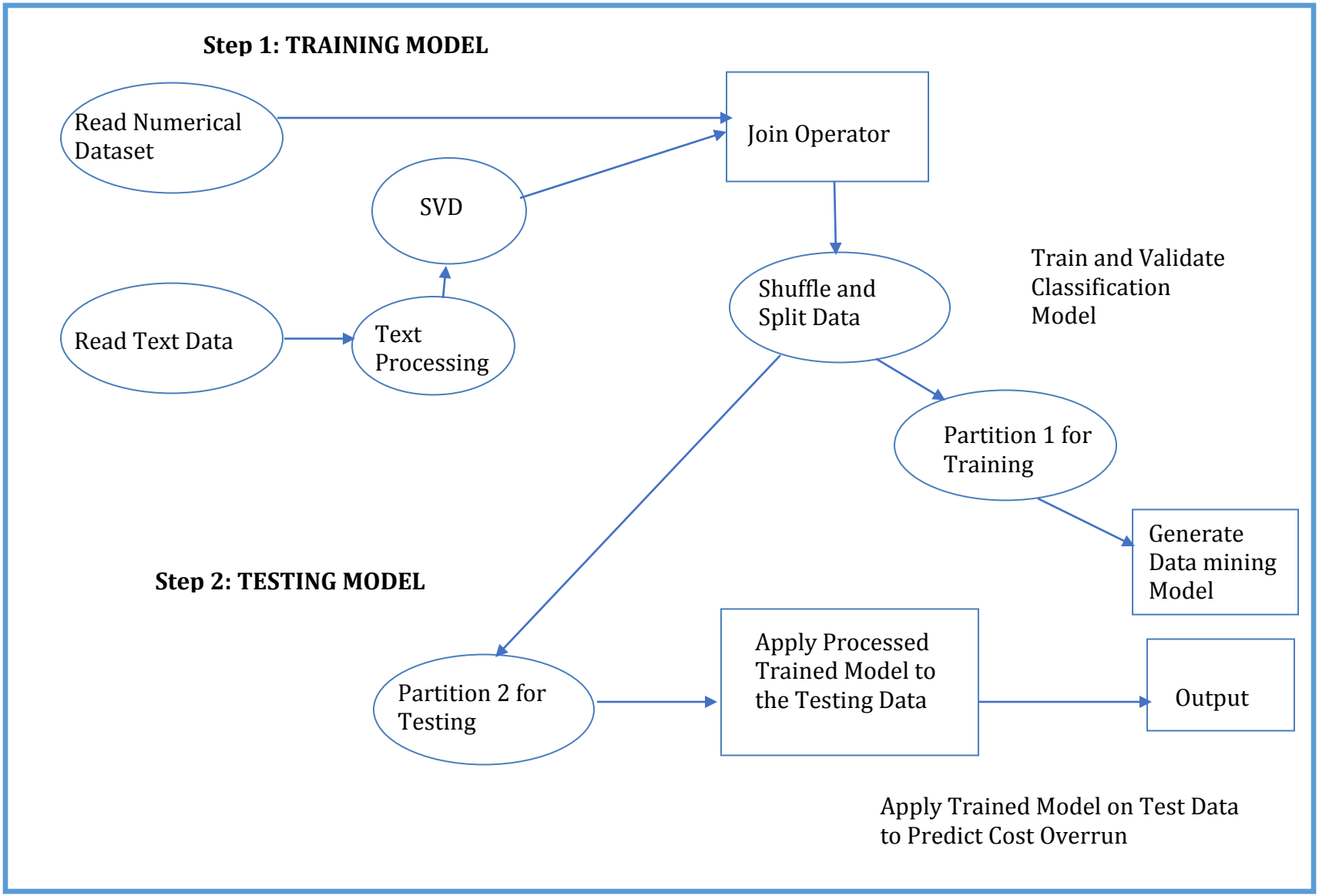
Various models were constructed that combined the text and numerical data to predict the level of cost overrun (or under run). Several different data mining algorithms were employed in the models with varying levels of success.

As is seen from the flowchart, the text data and numerical data were input into the model separately. The text data is then submitted to various text-mining algorithms to transform the text into a useable format and to provide data about the words and word pairs that are indicative of certain levels of cost overrun.

The numerical data is already in the structured form for the software to read and does not need any processing. However the textual data has to be processed for the algorithms to recognize the data. The purpose of text mining is to transform text into

numeric attributes that can then be used in data mining algorithms. Then Singular Value Decomposition (SVD) was used to reduce the text matrix to a single column of numerical data for each project. Singular value decomposition provides a convenient way for breaking the large matrix of projects and words output from the text processing models, into simpler, meaningful pieces. This was done to reduce the size of the problem so that it could be run on a workstation. It was found through experimentation that significant amounts of memory are required to run this model, which requires both text processing and analysis of numeric data.

Figure 5.1: Model Process Flow



The numerical and text data are shuffled and split in the required ratios to form the training data (partition 1) and the test data (partition 2). The first partitions from both the datasets are then combined using the “join” operator to form one unified dataset. This data forms the training model, which is validated and used to build a classification model using the ensemble methods.

The second partitions from the split dataset form the testing data. The trained model is applied on the test data and the output generated is the classification model to predict cost overruns.

c. Ensemble Methods

An ensemble of classifiers is a set of classes whose individual decisions are combined in some way (typically by weighted or un-weighted voting) to classify new examples. They use multiple models to obtain a better performance than could be obtained from any of the constituent models.

Ensemble methods are distinguished into 2 categories:

- Averaging Methods: In this method the driving principle is to build several models independently and then to average their predictions. On average, the combined model is usually better than any of the single model because its variance is reduced.
- Boosting Methods: in boosting methods, models are built sequentially and one tries to reduce the bias of the combined model. The motivation is to combine several weak models to produce a powerful ensemble.

In this study we've used the following ensemble methods

1. AdaBoost: The core principle of AdaBoost is to fit a sequence of weak learners (i.e. models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction.
2. Bagging Methods: It trains multiple models on different sample subsets and averages their predictions. It then predicts on the test data by averaging the result of the multiple models trained. It improves the accuracy of one model by using it multiple times.
3. Stacking: it involves training a learning algorithm to combine the predictions of several other learning algorithms. First, all of the other algorithms are trained using the available data, then a combiner algorithm is trained to make a final prediction using all the predictions of the other algorithms as additional inputs. Stacking typically yields performance better than any single one of the trained models.

The learning algorithms used are a combination of Decision Trees,

Neural Network and Support Vector Machines.

6. IMPLEMENTATION

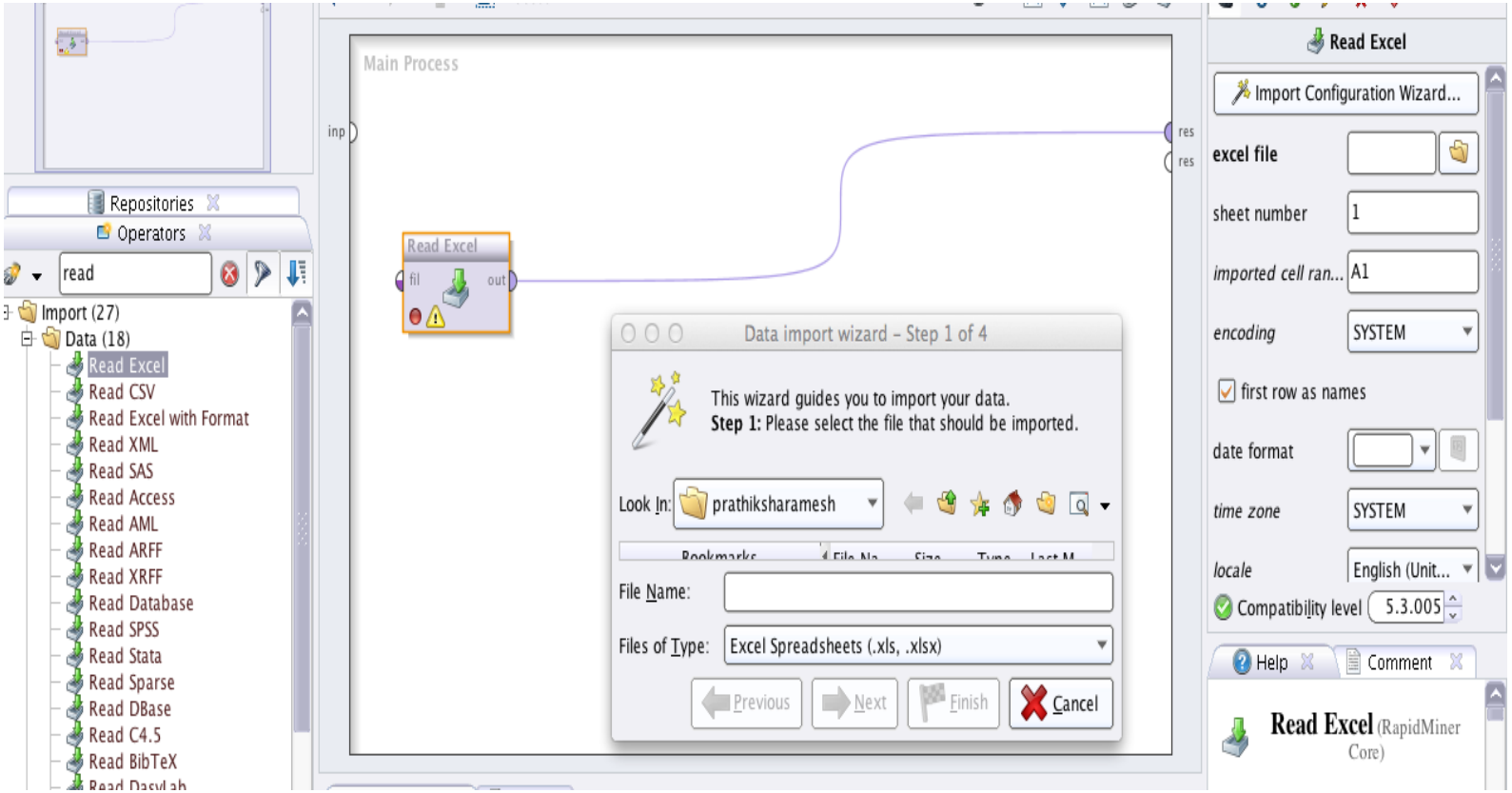
After all the data preparation, the learning algorithms are now implemented on the dataset to train classification models and then test them.

a. Data Importing

RapidMiner is a java program, which is accessed by running the rapidminer.jar file. After a new process is selected, the data is imported by adding the “Read Excel” operator into the process window. Once the operator is in the process window, the “import configuration wizard” opens a data import wizard from which we can choose the excel sheet to be imported. After the data is imported there are a series of steps, which help in better preparing the data.

The features should be chosen as “Real and Attributes”. Only the Class Overrun is chosen as “Nominal and Label” the label is nothing but the variable which is the output which is to be predicted. Numerical and Text Data are imported in separate “Read Excel” Operators.

Figure 6.1: Data Importing Window in Rapidminer



b. Numerical and Text Data

The numerical data, which is already in the form of numbers does not need any further structuring. However for the text data several processing operators need to be added. In the Text Dataset the text feature is chosen as “Text and Attribute”

The first processing operator is the “Process Documents from Data” operator is added. This operator contains a nested window, which means it requires a sub process/processes. Once the nested window is opened the following text structuring operators are added.

- Transform Cases: It keeps all the words in uniform case either upper/lower.
- Filter Stopwords: Choose the English filter stop words. This operator removes stop words like a, an, is, am, the etc.
- Tokenize: This splits the document into a sequence of words. In this model the tokens were equivalent to single words.
- Stemming. In this data transformation related word tokens are normalized into a single form. For example “walking” would be transformed to “walk” (Miner et al. 2012). The Porter method of stemming was used. For the collected data stemming has been found to increase the accuracy of the classification algorithm.
- Filter Tokens: This filters tokens based on the minimum and maximum characters chosen. We chose to filter all token less than 2 characters long and more than 25 characters long.

- Generate n-grams: This identifies 2 or 3 word pairs and combines them to form additional tokens.

Singular Value Decomposition: It is a data reduction operator that converts the output matrix from the text-processing data into a simpler matrix with lesser number of variables. The number of dimensions in the SVD operator is the number of clusters or the number of variables in the reduced data matrix. Increasing or decreasing the number of dimensions improves the matrix by adding or removing terms from it.

SVD dimensions: 2

c. Join Operator

After all the text processing is complete, the output from the "SVD" and the output from the numerical data are combined using the "Join" operator. The join operator works on the principle of joining the two datasets using a common column or ID present. Check the "use id as attribute key" option and connect the numerical and text data on the left and right inputs.

d. Shuffle and Split

To prepare the training model, the unified dataset is first shuffled using the "Shuffle" operator. The shuffle eliminates any chance of over fitting the model by not placing all classes of overruns together and randomly shuffling the dataset using a local random seed. The shuffled data is then input into the "Split Data" operator.

The split data operator splits the data into 2 parts depending on the ratios for linear sampling.

- Split ratio of 65% for the training model and 35% for the testing model.

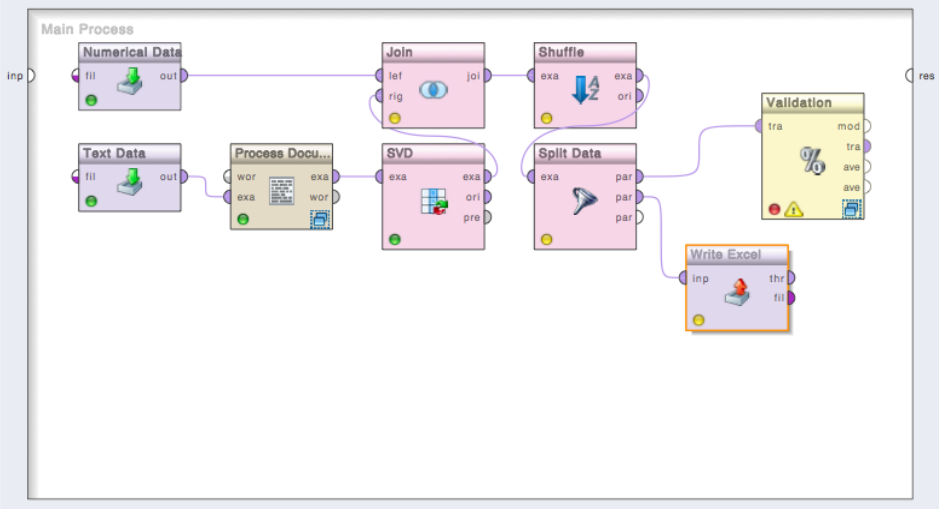
The second partition is written into an excel file using “Write Excel” operator, to be used later for the testing model.

e. Validation

This operator performs validation in order to estimate the statistical performance of a learning operator. The first partition, which is the training model, is connected to the validation operator. The validation operator we’ve chose for this study is the “Bootstrapping Validation” The validation operator is nothing but the main process for which the sub process is the ensemble method. The bootstrapping validation has a number of validations, which are the number of times that the model will be trained. Connect both the model and performance vector output from the Validation to the results.

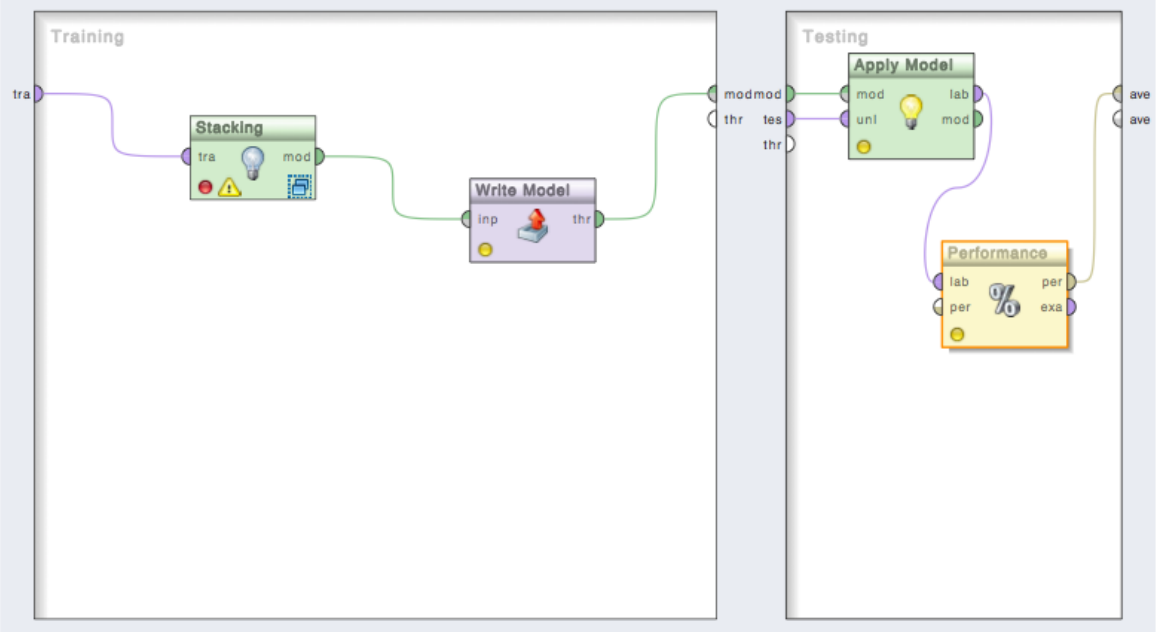
- Validation number: 7
- Validation ratio: 0.7

Figure 6.2: Training Process Window in Rapidminer



The sub process of the “Bootstrapping Validation” contains training and testing process windows. The ensemble method (Bagging, AdaBoost and Stacking) is put in the training window. The model process is written onto a model file using “Write Model” operator. This model file will then be used for the testing model. In the testing process window we use “Apply Model” and “Performance” operators. We use the Classification Performance. The performance operator is used for statistical performance evaluation of classification tasks. This operator delivers a list of performance criteria values of the classification task. Check the “accuracy” and “classification error” for the performance operator.

Figure 6.3: Validation Sub-Process in Rapidminer



The actual learning algorithms (Decision Trees, Neural Networks and Support Vector Machines) are applied in the sub process of the ensemble methods.

i. Classification Algorithms

The ensemble methods used are

- Stacking:
- AdaBoost: With iterations taken as 10.
- Bagging: Sample ratio 0.75 and iterations 10.

In the sub process of the ensemble method, the classification algorithms are applied. These are

- Decision Tree:

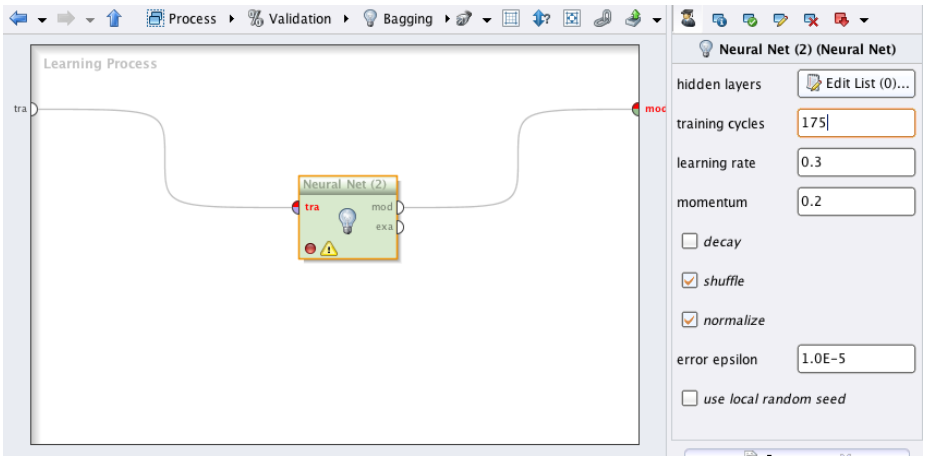
  Criterion- gain ratio; minimum size for splitting- 4, minimum leaf size- 2; minimal gain- 0.3, maximal depth- 20; confidence- 0.25

- Neural Network:

  Training cycles- 175; learning rate- 0.2; momentum- 0.2

- Support Vector Machine: We use the SVM (lib) operator which is a multiclass classifier as opposed to a normal SVM which is a binomial classifier. Cache size- 50; c- 0.5. Keep the cache size low to optimize memory

Figure 6.4: Sample Bagging Sub-Process



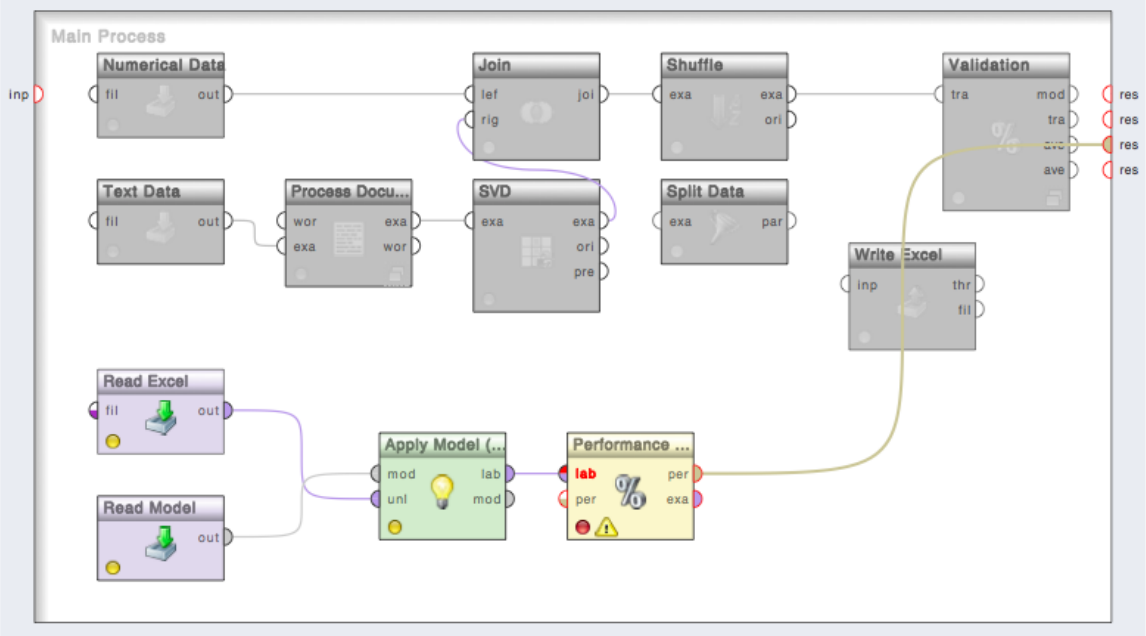
After all the necessary operators are applied, the training model is run. The result is a class precision table showing the accuracy of the trained model.

f. Testing Model

The second partition which is written in an excel sheet and the model written from the ensemble method are opened using “Read Excel” and “Read Model” operators respectively. The trained model is used as the example model to apply on the test data and predict the output.

The output from the data and model is connected to “Apply Model” which in turn is connected to the “Performance (classification) Operator”. The resulting output is the classification model, which predicts the overrun class. The result is a class precision table.

Figure 6.5: Testing Model Window



7. RESULTS

a. CALIFORNIA DATASET

The results comparison tables for the different models are as follows.

Table 7.1: Result Comparison of California Dataset

Model	Accuracy	Class 1 Precision	Class 2 Precision	Class 3 Precision	Class 4 Precision
AdaBoost	44%	36.98%	51.83%	10%	10.83%
Bagging	47%	35.71%	52.36%	9.58%	0.00%
Stacking	38%	27.91%	50%	14.33%	6%

b. WASHINGTON DATASET

Table 7.2: Result Comparison of Washington Dataset

Model	Accuracy	Class 1 Precision	Class 2 Precision	Class 3 Precision	Class 4 Precision
AdaBoost	32%	17.34%	46.27%	12%	16.93%
Bagging	46%	36.99%	51.44%	8.47%	0.00%
Stacking	41%	22.43%	50%	19.86%	11.46%

c. Data Mining Output

Table 7.1 and 7.2 show a summary of the predictions produced by the models. Each of the models was run 3 times. Each model run used a different mixture of training and testing cases by varying the local random seed for shuffling. This insured that the training and testing sets used in each run was unique. The precision of the prediction represents the percentage of time a prediction made by the model is correct. The prediction recall represents the percentage a project's actual level of cost overrun is correctly predicted. The average accuracy of the models ranged from 34% to 47%. However, overall prediction accuracy was reduced by the poor performance in predicting projects with significant under runs.

The AdaBoost ensemble model had an average accuracy of 39% from both datasets. The model performed best in predicting cost overruns for Class 2 projects that had cost overruns between 1-11%. Prediction accuracy was low for projects with large overrun (>25%).

The Bagging ensemble model had the highest average accuracy of 46.5% from both datasets. The model performed best in predicting cost overruns for Class 2 projects with cost overruns between 1-11%.

The Stacked model had an average prediction accuracy of 38%. However the Stacked model is unable to predict large cost over runs. This model gave highly accurate prediction for projects completed near the low bid amount.

d. Word Lists

After running our text processing we generated a word frequency table. The operator "Word to Data" is connected to the output of the text processing. This result is written into an excel sheet. This table gives the words that have the highest frequency in each document (each project ID) as well the total occurrences. It also shows the number of times the word is present for each class of percentage overrun.

Table 7.3: California High Frequency Word List

Word	Total	Count in Documents	In Class (1)	In Class (2)	In Class (3)	In Class (4)
concrete	1677	842	457	863	266	91
asphalt	1110	601	364	542	158	46
traffic	750	601	253	364	93	40
type	702	572	218	353	103	28
system	672	572	205	344	85	38
control	611	554	198	292	81	40
pavement	349	274	112	170	55	12
mobility	333	332	76	180	55	22
bridge	305	164	38	156	73	38
sign	297	194	84	176	24	13
roadway	276	252	57	153	49	17
structure	244	186	33	148	46	17

Table 7.4: Washington High Frequency Word List

Word	Total	Count in Documents	In Class (1)	In Class (2)	In Class (3)	In Class (4)
bridge	378	359	91	198	61	28
pave	255	236	59	123	53	20
creek	195	189	50	85	43	17
river	186	179	40	113	20	13
road	162	154	39	80	31	12
repair	127	125	31	66	18	12
vicinity	96	96	27	47	16	6
safety	90	87	28	39	16	7
signal	82	80	18	38	19	7
improve	80	80	21	38	14	7
ramp	76	72	16	42	10	8
slope	73	69	20	32	14	7

8. CONCLUSION

On the basis of two real construction data sets the prediction models have performed considerably well. In both the data sets the class precision for Class 2 is always the highest. Class 2 is for projects with overruns between the percentages of 1-11%. This is because majority of the projects fall under this category. The ensemble models

developed are best able to predict cost overruns that fall near the low bid. The second on the list is Class 1, this is the project under runs. Therefore the classification models can be used for a great extent to predict cost overruns close to the bid amount as well as cost overruns.

Classes 3 & 4 have the least precision. This is because the ensemble models have a very low class recall (ability to actually predict the exact percentage of cost overrun) and also a small pool of projects fall under this class. Unless there are gross deficiencies in managing a project there are very few instances where a project has had overruns of more than 25%.

Since all data is available during the time of bidding. Prediction models can be used as an alternative method of bid selection.

9. FUTURE WORK

This study was conducted by using a fraction of the projects that have been completed. The best extension of this work would be collect more data so that there is a large pool of data to train and test models. Also one of reasons for a varied result in this study has been because we incorporated small and big projects alike. It would make more sense to build separate models based on the scale of the projects.

There has been considerable development in the field of text mining. The future work should definitely be focusing on gaining more textual data pertaining to the projects,

such as change orders. A lot of information is in the form of text that would be critical for the owner when it comes choosing a bid.

Although a basic statistical analysis was carried out to point out features that were statistically significant, it would be interesting to see more variables collected from the bidding documents so that there is an extensive list of attributes to choose.

Another extension of this study would be to include the contracting companies and the engineers along with the bid data, so that it provides information to the prospective owners regarding which company has incurred maximum overruns in the past.

10. REFERENCES

[1] Wilmot, C., and Cheng, G. (2003). “Estimating future highway construction costs.” J. Constr. Eng. Manage., 129(3), 272-279.

[2] Chaovalitwongse, W. Art., Wang, Wanbin., Williams, T.P., and Chaovalitwongse, Paveena. (2012). “Data mining framework to optimize the bid selection policy for competitively bid highway construction projects.” J. Constr. Eng. Manage. 138(2), 277-286.

[3] Williams, T.P. (2005). “Bidding ratios to predict highway project costs.” Eng. Constr. Archit. Manage., 12(1), 38-51.

[4] Caldas, C.H. & Soibelman, L., 2003, “Automating hierarchical document classification for construction management information systems.” Automat. Construct. 12(4), 395-406.

[5] Soibelman, L. & Kim, H., Data preparation process for construction knowledge generation through knowledge discovery in databases, J. Comput. Civil Eng. 2002, 16(1), 39-48.

[6] Soibelman L, Wu J, Caldas C, Brilakis I, Lin K-Y. Management and analysis of unstructured construction data types. Adv. Eng. Inform. 2008, 22(1), 15-27.

[7] Zhang, J. and El-Gohary, N. M., Information Transformation and Automated Reasoning for Automated Compliance Checking in Construction. Proceedings of the ASCE International Workshop on Computing in Civil Engineering, Los Angles, CA, 2013, 701-708.

[8] Son, H., Kim, C. & Kim, C., Hybrid principal component analysis and support vector machine model for predicting the cost performance of commercial building projects using pre-project planning variables, Automat. Construct. 2012, 27, 60-6.

[9] Gkritska, C. & Labi, S.S., Estimating cost discrepancies in highway contracts: Multistep econometric approach, J Constr Eng. 2008, 134(12), 953-62.

[10] Trost, S.M. & Oberlender, G.D., Predicting accuracy of early cost estimates using factor analysis and multivariate regression, J Constr Eng. 2003, 129(2), 198-204.

[11] Nassar, K.M., Nassar, W.M. & Hegab, M.Y., Evaluating cost overruns of asphalt paving project using statistical process control methods, J Constr Eng. 2005 131(11), 1173-8.

[12] Petroutsatou, K., Georgopoulos, E., Lambropoulos, S. & Pantouvakis, J.P., Early Cost Estimating of Road Tunnel Construction Using Neural Networks, J Constr Eng. 2011,138(6), 679-87.

[13] Williams, T.P., Bidding ratios to predict highway project costs, Eng., Const. Archit. Manag. 2005, 12(1), 38-51.

[14] Wilmot, C.G. & Cheng, G., Estimating future highway construction costs, J Constr Eng. 2003,129(3), 272-279.

[15] Williams, T.P. (2013). "Construction Automation." Journal Paper.

[16] Williams, T.P. (2002). "Predicting completed project cost using bidding data." Constr. Manage. Economics. 20, 225-235.

[17] Williams, T.P, Traina, B., and Whitehouse, L. (2011). "Using the tree map data visualization technique to study the likelihood of cost overruns from bidding data." Int. Constr. Specialty. Conf.

[18] http://www.cs.umd.edu/hcil/treemap/.

[19] Delen, Dursun., and Sharda, Ramesh. (2006). "Predicting box-office success of motion pictures with neural networks." Expert Systems with Applications 30(2), 243-254.

[20] RapidMiner software.

[21] Williams, T.P. (2003). "Predicting final cost for competitively bid projects using regression models." Int. J. Project. Manage. 21, 593-599.

[22] Oza, Nikung, J. "Ensemble data mining methods." NASA Ames Research Center.

[23] Witten, Ian, H. (1999). "Text Mining." Computer Science, University of Waikato, Hamilton, New Zealand.

[24] Williams, T.P., and Gong, Jie. (2013). "Predicting Construction Cost Overruns Using Text Mining, Numerical data and Ensemble Classifiers". Journal Paper.

[25] http://cs229.stanford.edu/notes/cs229-notes3.pdf "Support Vector Machines".

[26] http://abyss.uoregon.edu/~js/glossary/correlation.html “Correlation and regression analysis”