Predictive Analytics using SRP-CRISP-DM framework on the outcome of the NBA Matches

Raymund Mark Herradura, Mathew Christopher

Manukau Institute of Technology, Auckland, New Zealand

Abstract- Prediction on the result of National Basketball Association (NBA) is a challenging feat to accomplish for both researchers and domain experts. In this paper we validate the problem of predicting an NBA match outcome as a classification problem and apply the principles behind SRP-CRISP-DM framework using Trees algorithms to derive a high success rate. Our results arrived at a prediction of 95.73% accuracy, outperforming any other machine learning algorithms in this experiment. Nevertheless, another approach edging the accuracy of this framework is Expert Selected approach that would be explored further in the experiments.

1. Introduction

Data Analytics is a specific branch of Data Science which over arch with the spectrum of Machine Learning. Referenced in other publications as knowledge discovery, knowledge extraction, data archaeology and data mining (Chen, M., et. al 1996).

Machine Learning is a set of computer algorithms that harness the power of self-learning thru improved experience. Machine Learning focuses on the incremental process of self-learning and data modelling to form future predictions about the future, data mining narrows in on cleaning the large datasets to glean valuable insights from the past (Theobald, O. 2017). Predictive Analytics is at the core of Machine Learning wherein it's focused in predicting the probability of future events from historical data. Predictions are built upon a model a representation or a state, process, or system that requires our comprehension (Miller, J. D., et. al 2017).

CRISP-DM or cross-industry standard processing for data mining, is a popular and proven methodology with a well-structured approach on data mining processes. The

methodology includes the description of the stages of a project and the tasks involved are explained based on the association with each other. CRISP-DM is helpful in providing an overview of the data mining life cycle which contain six stages and It is not a requirement that the stages are sequential. The approach is flexible and can be tailored to requirements and can be moving back and forth from one stage of the project to other. The importance and relevance of different stages of the project can differ with the purpose for which the methodology is used (CRISP-DM Help Overview, n.d.).

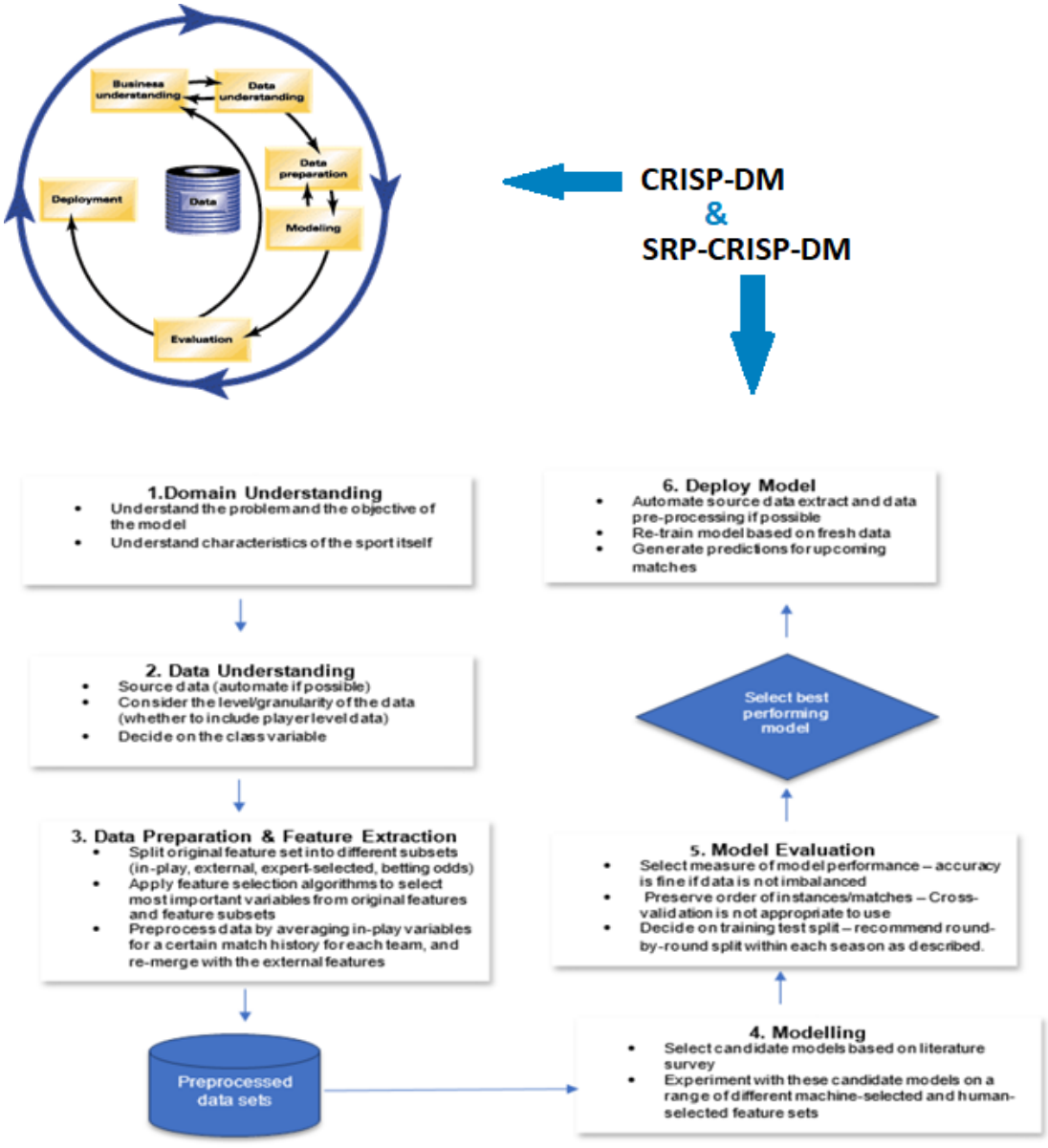

Fig. 1: SRP-CRISP- DM; Thabtah, F. (2017)

(Bunker, R. P., & Thabtah, F. 2017) put forward a new approach based on CRISP-DM specifically for sports prediction termed as SRP-CRISP-DM. The methodology was applied

in their study based on neural network. Using neural network for the purpose is a very modern approach in sport prediction compared to artificial neural networks in earlier works and it has been considered having high potential. The methodology has the six steps rooted on the steps in standard CRISP-DM and focus is fitting in the sport prediction aspect in CRISP-DM, this method is inferred to suit best for team sports. The stages of CRISP-DM and SRP-CRISP-DM lifecycle is as figure above. We did the study based on SRP-CRISP-DM methodology and incorporated all the required phases in our analysis details is as follows.

Domain Understanding in our context is understanding of basketball game and focussing on NBA season. there could be different objectives in predicting, our study focuses on the two outcomes win or lose. Data Understanding process involved checking data from different sources and to understand and come up with a dataset beneficial for the purpose. Complex datasets with player level data was not considered due to the complexity and time constrain. A publicly available dataset with team level stats was used for the analysis.

Data Preparation and Feature Extraction involved normalisation or scaling the numerical attributes which were with varying scales, input attributes causing noise was eliminated with the initial analysis of the attributes in pre-processing. Data was discretised to enable it to be used with all classification algorithms used the attributes that did not need discretisation was filtered. Feature selection applied and with different methods and classification algorithms. and the combination with the best accuracy results, another method incorporated was eliminating some attributes based on expert opinion. Modelling involved applying the pre-processed dataset with five candidate models namely Naïve Bayes, J48, Bagging (Random Forest), Random Forest and AdaBoostM1 (JRip Classifier), these models were applied with the machine selected features and the expert selected features. Model Evaluation was based on the above classification methods comparing the machine selected features and the expert selected features based on criteria like correctly number of correctly and incorrectly classified instances mean absolute error and error rate /accuracy. K-fold cross validation was used for training and testing, no held out data was used for testing. Deploy Model was not part

of the study as an automated fresh data to the model was not incorporated, but automated fresh data from seasonal games can be used to train the model further.

1.1 Project scope

Use SRP-CRISP-DM (Bunker & Thabtah, 2017) on the NBA dataset from 20014 until 2018 with 5 pre-selected teams to predict the outcome of their success/error rates in winning and losing games. The report is limited to the use of Classifiers such as Functions, Trees, Bayesian and ANN.

1.2 Goals

Use Supervised Learning Techniques such as Bayesian, Decision Trees, and Boosting to Predict the outcome of a binary class of a match either labelled win or loss.

1.3 Objectives:

Use dataset publicly available on data.world (Kelepouris, 2018) and apply SRP-CRISP-DM (Bunker & Thabtah, 2017) framework to predict the category of a two-class variable. Also, to employ Applied Machine Learning using Weka (Waikato University, n.d.) to build, train, test and evaluate the model

This report will tackle in Section I, an introduction to the framework that is utilise. Next in section II, the coverage of the report is laid-out and the what are the dataset parameters are including the goal of this research. While, In Section III a descriptive analysis of the data would be discussed, and the steps taken in transforming the data to build the models. Section IV will discuss about the results of the controlled experiments. Lastly, the report will end with the direction of future work.

2. Problem Statement

Sport Results Match Classification that uses nominal data as input features and needs Decision Trees, rules, function, and Naïve Bayes classifier to derive a high success rate

3.Data and Transformation Methods

3.1 Descriptive Analysis

The dataset was obtained from data.world (Kelepouris, 2018) it covers a 41 class variable, which are both nominal and numeric and has 9,840 instances. The historical per game statistics are from 2014 until 2018 regular season for each team. Three teams are pre-selected from this dataset playing 82 games per each regular season. The 3 pre-selected teams have a subset of data with 984 instances and 41 attributes. Now, from the subset of data each team's information will serve as the training data and will undergo data transformation and training by classification techniques to predict the accuracy of the outcome of a 2-class variable, which is either a win or loss category. The sample of the Data dictionary comprises of the following description below.

Table 1. abbreviation and attribute indices for the input variables

Index	Attribute Name	Description	Index	Attribute Name	Description
1	column_a	Null	**14**	X3PointShotsAttempted	# of 3-point shots attempted
2	Team	NBA Team name	**15**	X3PointShots.	% of 3-point shots made
3	Game	# of games played	**16**	FreeThrows	# of free-throws made
4	Date	date the game was played	**17**	FreeThrowsAttempted	total of free-throws attempted

5	Home	where the game was played	**18**	FreeThrows.	% of 3 free-throws made
6	Opponent	Name of Opponent	**19**	OffRebounds	total # of offensive rebounds
7	WINorLOSS	outcome of the match	**20**	TotalRebounds	total # of offensive + defensive rebounds
8	TeamPoints	total points scored	**21**	Assists	total # of assists made
9	OpponentPoints	total points scored by their opponent	**22**	Steals	total # of stelas committed
10	FieldGoals	# of goals made	**23**	Blocks	total # of shot blocks made
11	FieldGoalsAttempted	# of goals attempted	**24**	Turnovers	total # of turnovers committed
12	FieldGoals.	% of goals made	**25**	TotalFouls	total # of fouls committed
13	X3PointShots	# of 3-point shots made	**26**	Opp.FieldGoals	# of baskets made by an opponent

Index	Attribute Name	Description	Index	Attribute Name	Description
27	Opp.FieldGoalsAttempted	total # of baskets attempted by an opponent	**35**	Opp.OffRebounds	total # of offensive rebounds. Collected by an opponent
28	Opp.FieldGoals.	% of field goals made	**36**	Opp.TotalRebounds	total # of rebounds.collected by an opponent
29	Opp.3PointShots	total # of 3 pts. shot made by an opponent	**37**	Opp.Assists	total # of assists oponents made
30	Opp.3PointShotsAttempted	total # of 3 pts. shot attempted by an opponent	**38**	Opp.Steals	total # of steals oponents made

31	Opp.3PointShots.	% 3 pts. shot made by an opponent	**39**	Opp.Blocks	total # of blocks made by an opponent
32	Opp.FreeThrows	total # of freethrows made by an opponent	40	Opp.Turnovers	total # of turnovers oponents made
33	Opp.FreeThrowsAttempted	total # of free-throws attempted by an opponent	**41**	Opp.TotalFouls	total # of fouls oponents made
34	Opp.FreeThrows.	% of free-throws by an opponent			

Looking at fig. 1 2 characteristics can be derived from the plot of the dataset. Firstly, majority of the data types are numerical values with varying scales. Secondly, the distribution of the attributes is both normal (gaussian distribution) and not a bell curve. The attributes as described in table 1, teampoints, opponentpoints, fieldgoals, fieldgoals_2, x3pointshots_2, freethrows, assists, opp_fieldgoals, opp_fieldgoals2, opp_pointshotsattempted, opp_3pointshots_2, opp_totalrebounds are normally distributed while the rest are not on a gaussian bell curve as can be denoted from the attribute column_a which is skewed to the left and has a skewness of 2.29. With the initial analysis from the dataset it can be assumed that the input feature represented as column_a is a noisy attribute because of its skewness and can be removed from the dataset. Whilst, most of the attributes are numeric it can be rescaled so that the maximum value of each numeric attribute is 1 and the minimum value is 0. The normalise or rescaled attributes can be utilised for experimental analysis in the training dataset using a boosting technique.

figure 1: Visualisation of the all the distribution of the 41 class variables

3.2.Data Cleansing

There are no missing values identified for the subset of data under the 3 pre-selected teams of the NBA dataset. The steps involved to validate if there are missing values are as follows. Firstly, The NumericCleaner filter is selected in Weka (Waikato University, n.d.) by choosing unsupervised.attribute.NumericCleaner and then the configuration parameter is modified by specifying the appropriate indexed column for which attributes in the attributeIndex parameter. Next, set the configuration for minDefault was set to NaN, which is unknown, and this will replace the values below the default threshold of 1.8E. No missing values were identified as each attribute inspected contains a 0% missing values on their selected attribute column. Now, it is validated that this dataset does not inherit any missing values, thus this step is used to prepare for the next stage which is data pre-processing.

3.3 Data Pre-processing

The data types for the attributes of the dataset consist of 5 nominal and 36 numeric attributes. For solving the problem statement and to get a good performance out of the models the attributes needs to be discretize, so it could fit the classifiers such as trees, Bayesian, and functions algorithms. Discretization is when a real valued attribute is converted to a nominal value (Knox, 2018). The steps to discretize in Weka (Waikato University, n.d.) is for all the numeric attributes to be converted to nominal values except for the attributes of team, date, home, opponent and winorloss. The process on how to perform discretization is Firstly, navigate to the Filter pane and select unsupervised then traverse to attribute and choose discretize filter. The configuration was modified to only discretize selected numerical variables by updating the attributeIndices option to 1,8-41 only.

3.4 Feature Selection Results

To implement Feature Selection using WrapperSubsetEval, InfoGainAttributeEval, and CorrelationAttributeEval the Select Attributes tab was chosen and below this tab the Attribute Evaluator pane includes the three attribute evaluators that was picked. The corresponding configuration for WrapperSubsetEval is set to JRip as the classifier for the feature selection algorithm. RandomForest, RandomTree and J48 classifiers are interchanged accordingly to be employed for subset estimation accuracy in the controlled experiment. The reason for implementing discretization before feature selection is to utilise InfoGainAttributeEval attribute evaluator because it only accepts nominal class

attributes. Table II below explains the results of implementing the three chosen algorithms for feature selection, which produce a resulting best merit of subset that is 0.97 for RandomForest. The next types of feature selection to use are InfoGainAttributeEval and CorrelationAttributeEval paired with the ranker method. This search method will rank each attribute according to a score from 0 to 1 and for this research only the top 5 attributes obtaining scores near to 1 are selected. Also, it can be observed that the date attribute was ranked the highest with a score of 0.93,

which is not required in predicting the outcome of a match as a date variable does not account for a class to result to a win or loss.

Table II. best features selected by Wrapper

Table III. comparison of attribute ranking via search method

Feature selection	JRip (0.91)	RandomForest (0.97)	J48 (0.93)
WrapperSubsetEval	6,8,19,17	5,8,9,20	8,9

FEATURE SELECTION	**RANKED**	**ATRRIBUTE INDEX**	**ATTRIBUTE NAME**
INFOGAINATTRIBUTE	0.93992	4	date
	0.24212	8	teampoints
	0.20171	10	fieldgoals
	0.19149	12	fieldgoals_2
	0.14408	15	x3pointshots_2
CORRELATIONATTRIBUTE	0.2355	5	home
	0.1695	8	teampoints
	0.1643	10	fieldgoals
	0.1443	12	fieldgoals_2
	0.1327	36	opp_totalrebounds

Based from the results from tables II-III, the most beneficial algorithms to be employed are Wrapper using Random Forest and J48 as classifiers including BestFirst as the search methods to achieve better accuracy results in training and testing the model. Hence, features with index attributes of 5,8,9,20 (refer to Table I for list of attributes indices) are retained out of the 41 attributes. But, expert recommendation is also leveraged in the experiments for pre-selecting the unwanted attributes that would be used for training and testing the model. The attributes are hand-

picked by the expert to be excluded are with indices 1,2,3,4,5,12 (refer to Table I abbreviations).

3.4.1 Selecting the Models

In section 2, the pre-processing stage was completed to prepare the dataset for training and testing. Referring to the problem statement of this research it is already determined that Three Classification techniques would be employed, which are Trees, Bayesian, and Function algorithms will be use as the classification technique in creating the model that would be used for training and testing the data. The first classification technique, as explained by (Theobald, 2017) are Decision trees which are supervised learning techniques and mainly used for classification problems, but they can also solve regression problems. The input features used for Classification trees can vary from categorical and quantitative data that produces an output for categorical outcomes. Regression Trees also accept categorical and quantitative input features and produces a quantitative outcome. Next classification technique, which is under the radar of trees is Random Forest, the algorithm constructs multiple trees and integrate their predictions to elect an optimise track of classification and prediction (Theobald, 2017). Secondly, Naïve Bayes are simple probabilistic classifiers and is an application of Bayes Theorem that is taking the assumption that the features have strong independence (Mohammed et. al, 2017). Lastly, boosting also a variant of multiple decision trees and is a classification technique that transforms weak learners into strong learner. Boosting principle is the addition of weights to the iterations that were misclassified in previous rounds (Theobald, 2017). Every weak learner is trained by employing a simple set of training samples. As examined by (Freund & Schapire, n.d.) AdaboostM1 is employed when a weak learner is strong and capable to achieve a reasonably high-pitched accuracy. However, this method crashes if the weak learner is not capable to reach at least 50% accuracy when coursed on hard distributions.

3.4.2 Model Deployment

The classifiers are now selected for the classification problem and the data has completed the preprocessing stages such as discretization and feature selection. The data can now be used to fit for the selected algorithms. The classification algorithms are implemented to execute on the NBA dataset for each pre-selected team. All the 3 teams subset of data contains 328 instances and 5 input features selected by the wrapper algorithm with J48 and RandomForest as classifiers. The steps engaged for deployment are, Firstly navigate on the classify tab in Weka (Waikato University, n.d.) then the Classifier pane is chosen and then select the choose button next, traverse to weka.classifiers.tree and elect J48 with default configuration. Secondly, direct towards the Test Options pane, the cross-validation folds parameter is modified to 5. Also, the candidate class to predict from the input features is selected. The same methods are engaged with RandomForest. Moving on, to the next classification technique is Naïve Bayes, similarly in the previous step under the Classify tab we navigate to the Classifier pane and pick choose button and traverse to weka.classifier.bayes and select NaiveBayes algorithm with the default configuration and for cross-validation is set to 5 folds. Also, under the attribute selection pane we elect the class variable to predict which is winorloss. Last classifier to choose from is Adaboost, likewise in the previous step after selecting choose under the Classifier pane we then traverse to weka.classifier.meta and elect AdaboostM1. The results of the Model deployment will be discussed further in the next section.

4. Experimental Analysis

4.1 Experimental Settings

a. Platform Used: Weka Explorer (ver. 3.9.2)

b. Hardware Specifications: Processor:

2.6 GHz Intel Core i5,

Memory: 8 GB 1600 MHz DDR3

Software Specs: macOS High Sierra 10.13.14

c. Confusion Matrix

	loss	win			
	a	b	<-	classified	as
Actual Wins	4216	234	a	=	L
Actual Loss	94	4312	b	=	W

D. Evaluation measures used: k fold cross validation

4.2 Results Discussion

4.2.1 Cleveland Predictive Model Analysis

A.) SRP-CRISP-DM Framework Application

The controlled experiment was performed for both teams Cleveland and Sacramento. Two approaches were employed to perform the experiment. Firstly, SRP-CRISP-DM framework were the principles of the methodologies are observed and incorporated to WEKA (Waikato University, n.d.) workbench. Secondly, inferring to the Expert Selection method wherein class variables are carefully selected and removed from the dataset as part of pre-processing before implementing any classification techniques.

Table IV. Summary of Success/Error rates from Classification Algorithms

CLEVELAND (CLE)	SUMMARY					
Models	**Total # of instances**	**Mean Absolute Error**	**Correctly Classified Instances**	**Success rate %**	**Incorrectly Classified Instances**	**Error rate %**
Naïve Bayes	328	0.2631	282	85.97	46	14.02
J48	328	0.1061	307	93.59	21	6.4
Bagging (Random forest)	328	0.13	311	94.81	17	5.18
RandomForest	328	0.1099	312	95.12	16	4.87
AdaBoostM1 (JRip classifier)	328	0.058	314	95.73	14	4.27

The experiment is executed after the pre-processing phase leaving the dataset discretized and applied feature selection. As a result, leaving the final dataset comprises of 328 instances and 5 attributes (home, winorloss, teampoints, opponenpoints, offrebounds). Table IV outlines the various algorithms used for the SRP-CRISP-DM framework and their corresponding results that constitutes to the overall performance of the model. AdaBoostM1with JRip as the classifier achieved the highest success rate of 96%. and Naïve Bayes attained the lowest success rate at 86%. Proportionate to Naïve Bayes attaining the lowest success rate is also scoring high for the mean absolute error rate that is the accuracy for the average error score for each prediction (Theobald, 2017).

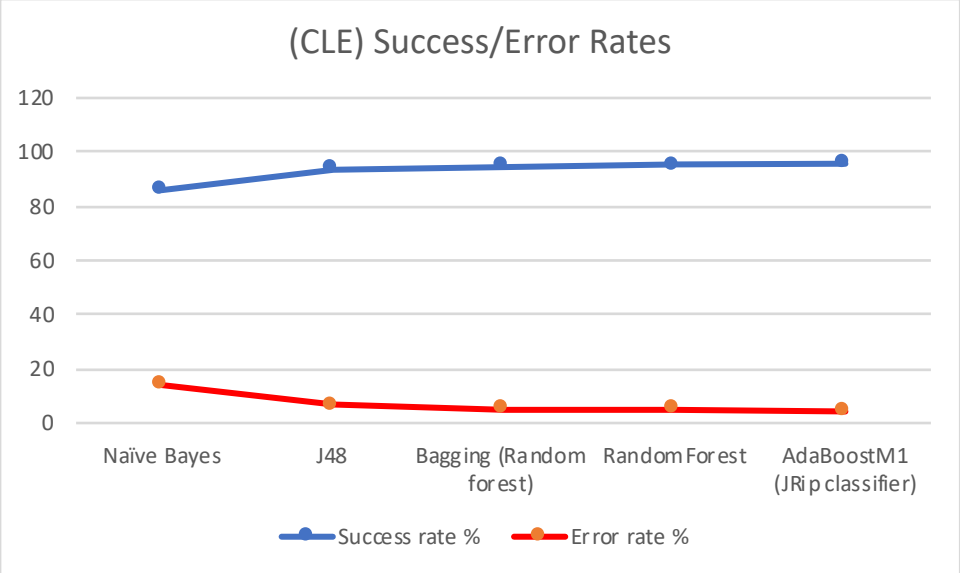

fig. 2 Line Chart on success/error rates on SRP-CRISP-DM model

The line chart above shows a holistic view of where each model performance sits based on achieving a high success rate and the figures they attained for the error rate. J48 and Bagging algorithms performed less than 1 percent against Random Forest. Their comparable success rates depict that they have a similar performance in training, testing and evaluating the model for this experiment.

B.) Expert Selected Feature Selection

This approach utilised opinions from experts to carefully select and removed noisy attributes such as 1,2,3,4,5,12 (pls. refer to table I for indices and their abbreviations). The dataset after discretization and expert selection retained a total of 325 instances with 35 class variables. Also, a percentage split of 90% training data and 10% test data was employed under this approach. AdaBoost and Random Forest achieved identical success rate of 97%. Whilst, Naïve Bayes attained the lowest at 91% and proportionately scoring relatively the highest error rate for mean absolute error.

Table V. Summary of Success/Error rates from Classification Algorithms

CLEVELAND (CLE)			SUMMARY				
Models	**% Split**	**Total # of instances**	**Mean Absolute Error**	**Correctly Classified Instances**	**Success rate %**	**Incorrectly Classified Instances**	**Error rate %**
Naïve Bayes	**90-10**	328	0.1404	30	90.91	3	9
J48	**80-20**	328	0.091	61	92.42	5	7.58
Bagging (Random forest)	**90-10**	328	0.2966	31	93.94	2	6.06
RandomForest	**90-10**	328	0.2815	32	96.97	1	3.03
AdaBoostM1 (JRip classifier)	**90-10**	328	0.0557	32	96.97	1	3.03

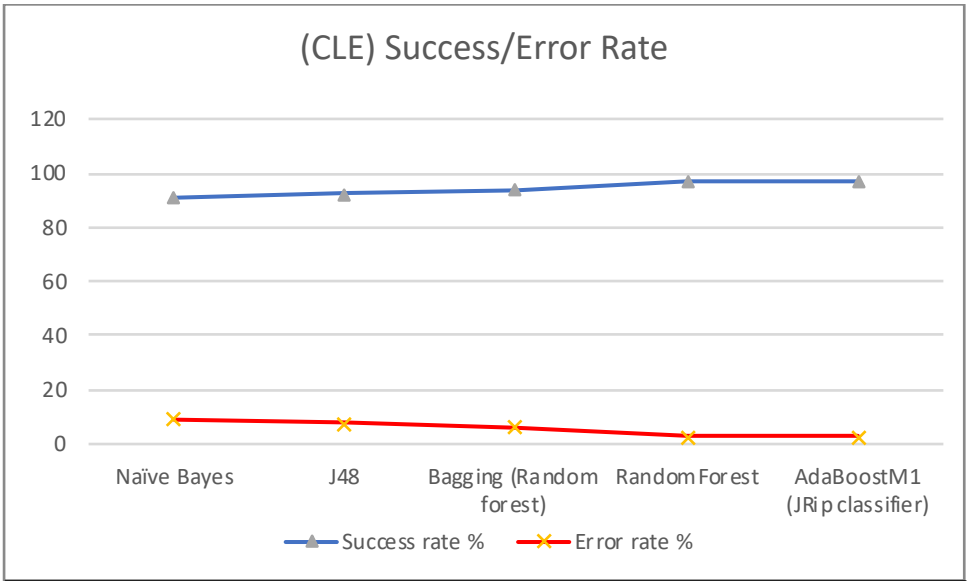

fig. 3 CLE Line Chart on success/error rates on Expert Selected approach

The line chart above shows a general view of where each individual model performance lies based on achieving a high success rate and the figures they attained for the error rate. J48 and Bagging algorithm achieved once more similar results and are less than 3 percent from the success rates of Random Forest and AdaBoost. Meaning, they have similar performance in training, testing and building the model.

`

4.2.2 Sacramento Predictive Model Analysis

A. SRP-CRISP DM Approach

The principles behind SRP-CRISP-DM was applied to this controlled experiment. A total of 5 models were built with 5 different classification techniques and implemented with the same dataset of 328 instances and 5 class variables J48 classifier achieved a fair success rate of 68% and Bagging method achieved the lowest from the group at 60%. Proportionately, Bagging classifier achieved the highest error rate from this experiment at 40%.

Table VI: SAC Summary of Accuracy/Error rates from Classification Algorithms

SACRAMENTO(SAC)	**SUMMARY**					
Models	**Total # of instances**	**Mean Absolute Error**	**Correctly Classified Instances**	**Success rate %**	**Incorrectly Classified Instances**	**Error rate %**
Bagging (Random forest classifier)	328	0.4123	197	60	131	40
RandomForest	328	0.4154	200	61	128	39
AdaBoostM1 (JRip classifier)	328	0.4216	212	65	116	35
Naïve Bayes	328	0.3724	218	66	110	34
J48	328	0.3997	224	68	104	32

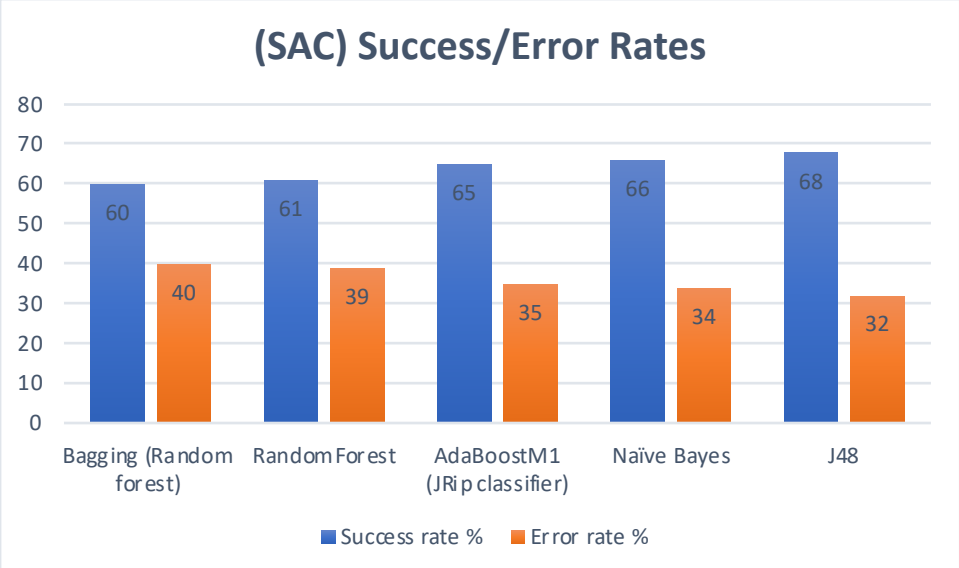

fig. 4 SAC Bar Chart on success/error rates on SRP-CRISP-DM model

The bar chart above shows an overall view of every model performance relative to their strength in percentage based on achieving a high success rate and the error rates clinched. AdaBoost and Naïve Bayes classifiers achieved almost identical results at 65% and 66% respectively and are less than plus or minus 3 percent from J48. The comparable success rates of Naïve Bayes and AdaBoost connotes that they have similar performance in training, testing and building the model.

B. Expert Selected Approach

This approach consumed the opinion from an expert to cautiously select and uninvolved noisy attributes such as 1,2,3,4,5,12 (pls. refer to table I for indices and their abbreviations). The dataset used after discretization and expert selection retained a total of 325 instances with 35 classes of variables. Also, a percentage split of 90% training data and 10% test data was employed under this approach. AdaBoost and Random Forest are the top algorithms for this set of experiment achieving 88% and 76% correspondingly. Whilst, Naïve Bayes attained the lowest at 79% and uniformly scoring relatively the highest error rate for mean absolute error at 21%.

Table VII: SAC Summary of Accuracy/Error rates from Classification Algorithms

SACRAMENTO (SAC)		SUMMARY				
Models	**% Split**	**Mean Absolute Error**	**Correctly Classified Instances**	**Success rate %**	**Incorrectly Classified Instances**	**Error rate %**
Naïve Bayes	**90-10**	0.2329	26	79	7	21
J48	**80-20**	0.1431	29	88	4	12
Bagging (Random forest)	**90-10**	0.3827	25	76	8	24
RandomForest	**90-10**	0.3781	25	76	8	24
AdaBoostM1 (JRip classifier)	**90-10**	0.1332	29	88	4	12

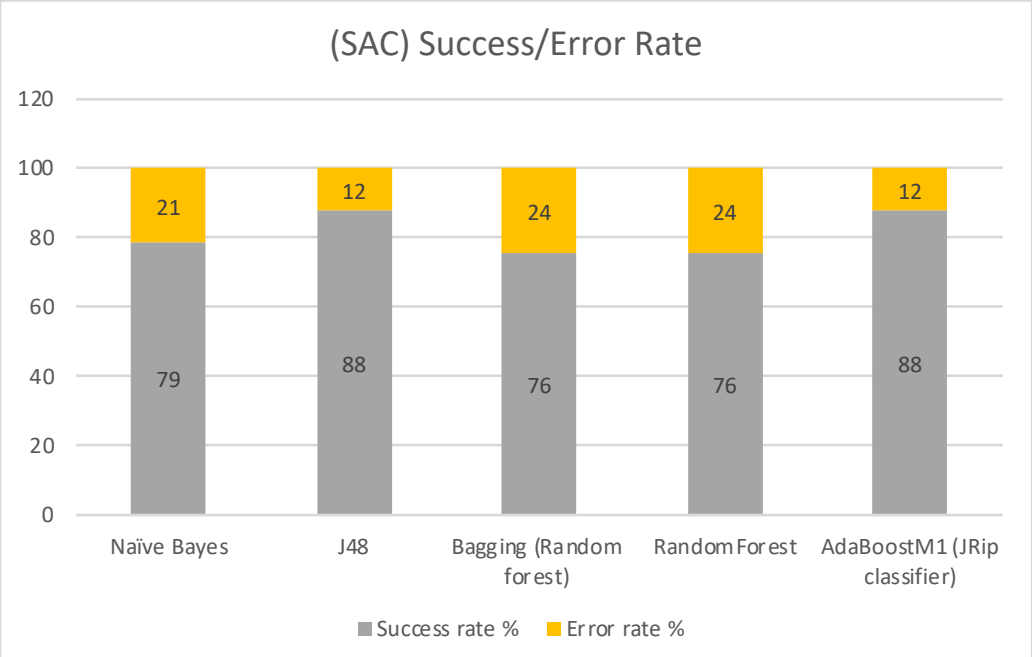

fig. 5 SAC Line Chart on success/error rates based on expert selection

The bar chart above depicts an overall view of every model performance in percentage, which is based in accomplishing a high success rate and the error rates clinched. AdaBoost and J48 classifiers both achieved the top success rate at 88%. While, similar results were achieved by Bagging and Random Forest both at 76% respectively and have a 12 percent difference from

AdaBoost. The similar success rates of Bagging and Random Forest connotes that they have similar performance in training, testing and building the model.

C. Link to the issues and reality

Machine Learning (ML) methods can be successfully infused to a broad spectrum of problems involving speech recognition, cybersecurity, natural language processing (NLP), bioinformatics and financial market analysis (Stamp, 2017). Cleveland ranks 4th overall in the Eastern Conference at 61% winning percentage, while Sacramento ranks 12th overall in the Western conference at 33% winning percentage. To conclude this experiment, the results for expert selection approach yields the best outcome by achieving high success rates and low precision scores on error for both teams. Therefore, Expert Selection using Boosting and Random Forest algorithms for predicting the outcome of a class variable will achieve a high accuracy in classifying whether an NBA match will win or lose.

4. Recommendations / Conclusions and Future Work

Even when expert selected features yield the highest success rate in categorising the outcome of the match for a two-class variable, to improve the machine learning model performance more a sample of 5 more teams from different ranking and divisions can be sampled. Repeated training and testing as discussed by (Witten, n.d.) , wherein evaluation is implemented by the classifier by training-test split with training it on the previous and testing on the later. To make different split the random number generator seed. Include player-level data to determine how each player statistics affect the match outcomes.

REFERENCES

1. CRISP-DM Help Overview. (n.d.). Retrieved from IBM Knowledge Center: https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.crispdm.help/crisp_overview.htm
2. Chen, M.-S., Jiawei, H., & Yu, P. (1996). Data Mining: An Overview from Database Perspective, *8*(6). https://doi.org/10.1109/69.553155
3. Freund, Y., & Schapire, R. E. (1999). A Short Introduction to Boosting.
4. FutureLearn. (n.d.). Repeated training and testing - Data Mining with Weka - The University of Waikato. Retrieved 21 June 2018, from https://www.futurelearn.com/courses/data-mining-with-weka/0/steps/25382
5. Hiremath, M. (2016). An Introduction to Machine Learning. *Open Source for You*, *4*(4), 87.
6. Kelepouris, I. (2018). NBA Teams Stats for every game 2014 - 2018 period. Retrieved 22 June 2018, from https://data.world/ionaskel/nba-teams-stats-for-every-game-2014-2018-period/workspace/query?queryid=d4cd54bc-2a5a-456d-bb38-d7901a52c37e
7. Knox, S. W. (2018). *Machine Learning: Topics and Techniques*. Newark, UNITED STATES: John Wiley & Sons, Incorporated. Retrieved from http://ebookcentral.proquest.com/lib/manukau/detail.action?docID=5323676
8. Mohammed, M., Khan, M. B., & Eihab Bashier Mohammed, B. (2017). *Machine Learning Algorithms and Applications*. CRC press.
9. Stamp, M. (2017). *Introduction to Machine Learning with Applications in Information Security*. London, UNITED KINGDOM: CRC Press. Retrieved from http://ebookcentral.proquest.com/lib/manukau/detail.action?docID=5056387
10. Theobald, O. (2017). *Machine Learning For Absolute Beginners* (2nd ed.).
11. Waikato University. (n.d.). Machine Learning Project at the University of Waikato in New Zealand. Retrieved 24 June 2018, from https://www.cs.waikato.ac.nz/ml/index.html
12. Bunker, R. P., & Thabtah, F. (2017, 09). A machine learning framework for sport result prediction. *Applied Computing and Informatics*. doi:10.1016/j.aci.2017.09.005
13. Witten, I. (n.d.). Repeated training and testing - Data Mining with Weka - The University of Waikato. Retrieved 21 June 2018, from https://www.futurelearn.com/courses/data-mining-with-weka/0/steps/25382