# Sentiment classification of movie reviews, Air passenger satisfaction classification and Diamond price prediction

*Aloysious Brhanavan*
National College of Ireland (NCI)
*Dublin, Ireland*
*x19206984@student.ncirl.ie*

***Abstract*— This paper analysed three different datasets, Rotten Tomatoes movie reviews dataset, Airline passenger satisfaction dataset and Bluenile diamond prices dataset using supervised machine learning techniques. Three different machine learning techniques were applied and the results were evaluated to identify the best model for each dataset.**

**Linear support vector machines, multinomial Naive Bayes and Random Forest models were applied to the Rotten Tomatoes movie reviews text sentiment classification problem. Three tree-based ensemble leaning methods Random Forest, Gradient Boosting and AdaBoost were applied to both Airline passenger satisfaction dataset and Bluenile diamond prices dataset.**

**In binary text sentiment classification, Linear SVC scored higher than the other two models in all evaluation metrics considered. Linear SVC with tf-idf representation achieved a maximum accuracy of 86.0% in Rotten Tomatoes movie reviews sentiment classification. Random Forest was the best among the ensemble learning methods on both classification and regression problems. Random Forest achieved an accuracy of 96.2% on Airline passenger satisfaction classification and explained 99.2% of the variance in log diamond prices.**

***Keywords*— *Rotten Tomatoes, Movie reviews sentiment, Air Passenger Satisfaction, Diamond prices, Linear SVM, Multinomial Naive Bayes, Ensemble learning, Random Forest, Gradient Boost, AdaBoost***

## I. INTRODUCTION

In this paper, three different machine learning techniques were applied and a comparative analysis of the techniques was conducted with the ultimate goal of identifying the best in class model for each dataset. While the main objective of the study was to maximise predictive power, the report also explored ways to reduce model complexity and built parsimonious models as a supplementary analysis.

Altogether five different machine learning techniques were evaluated in this report. These are Linear SVM, multinomial Naïve Bayes, Random Forest, Gradient Boosting and AdaBoost. Both Gradient boosting and AdaBoost used decision stumps as base learners.

*Dataset 1: Rotten Tomatoes Movie Reviews (RTMR)*

The internet has revolutionized the way we evaluate products and services. Consumers are increasingly relying on feedback, impressions and reviews from other customers to make informed decisions. The power that reviews have in influencing consumer decisions and consequently the impact on sales has led to an explosion in online reviews about pretty much everything we consume.

Most reviews come in the form of unstructured text data. However manually shifting through mountains of text data to gain insights is both tedious and time consuming. Sentiment analysis, a branch of Natural Language Processing (NLP) helps to identify sentiment given a text.

The first dataset Rotten Tomatoes movie reviews consisted of 169,669 reviews. It was sourced directly from the Rotten Tomatoes website for purpose of this project using a combination of web scraping and API calls.

The machine learning problem was to classify the audience opinion as positive or negative from the text reviews. This was a challenging sentiment classification problem involving unstructured, subjective and sometimes non-coherent short text.

The main research question for this dataset was:

1. How accurately can we predict whether the sentiment of a movie review is positive or negative?

In addition, the following secondary research questions regarding implementation were also answered as part of the investigation.

2. How much can the prediction accuracy be improved by increasing the number of features?
3. Can we build a model with high accuracy and less features using feature selection?
4. Can a voting classifier that uses majority voting rule to combines all three classifiers into a single collective prediction model achieve a higher accuracy than the best machine learning-based model?

*Dataset 2: Airline Passenger Satisfaction (APS)*

Businesses solicit feedback through customer surveys in order to evaluate their service against customer expectations. This helps assess how satisfied are customers with various aspects of products and service and measure customer satisfaction. High customer satisfaction has a direct link to increased revenue via loyalty and repeat business.

The second dataset Airline Passenger Satisfaction was sourced from Kaggle. It was chosen for the project because it contained a large number of samples and consisted of a mixture of nominal, ordinal and numerical variables.

Airline Passenger satisfaction dataset involved predicting overall customer satisfaction of a particular journey using factors such as age, gender, type of travel and customer feedback regarding different aspects of the airline's services.

The research questions for this dataset were:

1. How well can we predict whether an airline passenger is satisfied or not satisfied on a particular journey using variables such as age, gender, travel type and passenger feedback on different aspects of the service?
2. What factors have the biggest impact on airline passenger satisfaction?

*Dataset 3: Bluenile Diamond Prices (BDP)*

Bluenile was found in 1999 with the aim of demystifying the experience of buying a diamond or an engagement ring. It is now the world's largest internet retailer of engagement rings and loose diamonds. The secret to Bluenile's success was its innovative busines model that was born out of the need to preserve capital after the dotcom crash. The traditional diamond retailers buy and hold diamonds in their inventory with the aim of reselling them with a markup to make a profit. Bluenile simply acts as a marketplace between buyers and sellers and charges a commission on the sale price of a diamond. As a result, Bluenile has the ability have thousands of diamonds in its virtual inventory and offer customers a wide range of selection without having to tie up a large amount of capital to hold inventory.

This dataset was sourced directly from Bluenile using its public API. It has a total of 21,245 rows and 14 columns.

The research questions for this dataset were:

1. How accurately can we predict price of a diamond based on its characteristics (such as carat, colour, clarity, cut, depth etc)?
2. What characteristics are the major determinants of price of a diamond?

## II. RELATED WORK

This section discusses journal articles and conference papers that are relevant to the research questions set out in the previous section.

*Dataset 1: RTMR*

The authors of [1] made publicly available a large movie review dataset consisting of 50,000 samples from the Internet Movie Database (IMDB). This dataset set is very similar the RTMR dataset. The researchers decided not to remove stop words in preprocessing step because some words, especially negative words could be indictive of sentiment. In addition, the authors only included highly polarized reviews and excluded neutral reviews of rating between 5 and 7 stars out of 10 from the dataset and used a balanced dataset to train the models. A fixed 5,000 most frequent term vocabulary was used for both bag-of-words and tf-idf representations. The authors demonstrated Linear SVM classifier showed superior performance compared to other methods and achieved a highest accuracy of 88.33% in classification

The authors of [2] used a novel text representation that compensated for the fuzziness and uncertainty of language found in short reviews by introducing probabilistic linguistic terms sets. The preprocessing steps consisted of spelling correction, negative words checking, Part of speech (POS) tagging and stop words removal. They compared the efficiency of lexicon-based approaches and machine learning approaches using this representation on three benchmark datasets. One of these benchmark datasets was a balanced Rotten Tomatoes movie reviews consisting of 10,662 samples. Naive Bayes, SVM and Logistic regression achieved precision of 0.75, 0.75 and 0.76 and recall of 0.78, 0.76 and 0.77 respectively on Rotten Tomatoes movie review dataset. Among the five sentiment lexicons compared, VADER achieved the best performance on movie reviews and TextBlob came second with slightly worse performance than VADER. The model achieved an accuracy of 84.22% and this was superior to other state of the art approaches for movie reviews. A summary of other state of the art approaches for movie reviews mentioned in this paper and the corresponding accuracies are summarized in Table 1.

| Author/Year | Method | Accuracy |
|---|---|---|
| Socher et al. (2011) | Semi-Supervised Recursive Auto-encoders (RAE) | 77.7% |
| Appel et al. (2016) | Semantic Rules, Fuzzy Sets | 76.0% |
| Kim et al. (2014) | CNN | 81.5% |
| Zhang et al. (2014) | CharCNN | 77.0% |
| Wang et al. (2017) | KPCNN | 83.2% |

*Table 1: State of the art approaches for movie reviews*

In [3] the authors performed a comparison of existing machine learning techniques and lexicon-based approaches for opinion mining. The researchers demonstrated machine learning methods such as SVM and Naive Bayes produce the highest accuracy for text sentiment classification and outmatched lexicon-based methods. Moreover, the paper also concluded the inclusion of bigrams results in better model accuracy.

The authors of [4] compared the performance Naïve Bayes, maximum entropy and SVM in classifying IMDB movie reviews. SVM with bag of words representation that only included unigrams achieved the highest accuracy score of 82.9%. They noted the inclusion of bigrams did not improve performance beyond that of unigrams. However, adding bigrams did not adversely affect the accuracy in a meaningful way too.

In [5] the researchers compared SVM, Naïve Bayes, Random Forest and Convolution Neural Network (CNN) for movie review sentiment classification using the IMDB 50K movie reviews dataset. The machine learning models Random Forest, SVM and Naïve Bayes achieved accuracy scores of 0.84, 0.83 and 0.78 respectively. CNN model pre-trained on word vectors and fine-tuned during training achieved the highest accuracy score of 88.9%.

The authors of [6] performed a comparative efficiency analysis of 3 machine learning algorithms on classifying Twitter sentiment. SVM achieved the highest accuracy of 82.3% followed by Naive Bayes classifier with accuracy score of 79.3%.

In [7] the authors compared lexicon-based classifiers versus supervised machine learning methods in the domain of movie reviews. The results showed lexicon-based approach was easily outclassed by machine learning based models.

*Dataset 2: APS*

The authors of [8] analyzed Airline passenger satisfaction data sourced from Skytrax air travel review portal using several standard classification algorithms provided by WEKA. The report found the factors that contributed highly to the passenger satisfaction were airport queuing time, lounge comfort, airline cabin staff quality and seat legroom.

The reference paper [9] compared the accuracy of several ensemble methods based on classification trees using 14 different data sets. It found the best overall results were obtained with Random Forest. Moreover, Random Forest proved the most robust against noise too.

The authors of [10] presented empirical evidence to support Adaboost's resistance to overfitting and discussed how this crucial property of AdaBoost is inconsistent with the statistical view point.

*Dataset 3: BDP*

In [11] the authors applied ensemble learning methods to the prediction of diamond prices using its characteristics such as carat, colour, clarity, cut etc. Random Forest, Gradient Boosting, AdaBoost and Bagging achieved R-square values of 0.978, 0.958, 0.881 and 0.965 respectively

The authors of [12] examined the predictive power of liner, Random forest, polynomial regression, Gradient descent and Neural network for predicting diamond prices and ranked Random Forest as the best estimator.

## III. CHOICE OF MACHINE LEARNING METHODS

The two crucial properties that influenced the choice of the machine learning methods employed for each dataset were predictive power and interpretability.

*Dataset 1: RMTR*

The choice of three machine learning classifiers were Linear SVM, multinomial Naïve Bayes and Random Forest.

With the exception of [5], all other reference papers in section II ranked Linear SVM as the best machine learning method for text sentiment classification and Naïve Bayes as a close second. The reference paper [5] ranked Random Forest as the best among ML methods for IMDB 50K movie reviews sentiment classification.

In addition, this study opted for Scikit learn's generalized linear classifier, SGDClassifier implementation of Linear SVM model. SGDClassifer uses stochastic Gradient Descent as solver and scales well for sparse machine learning problems encountered in text sentiment classification.

Moreover, all three models are interpretable. The coefficients of Linear SVM gives the weights assigned to each of its features. Similarly, log probabilities of features given a class in multinomial Naïve Bayes and feature importances in Random Forest can aid in understanding model the predictions.

*Dataset 2: APS and Dataset 3: BDP*

Both datase2 and dataset 3 used tree-based ensemble learning methods Random Forest, Gradient Boosting and Adaboost for classification and regression respectively. Gradient boosting and AdaBoost used decision stumps as base learners.

Both datasets had a number of nominal and ordinal variables. In dataset 2, the features such as Ease of Online booking, Gate location, Online boarding, Seat Comfort had passenger feedback scores between 0 and 5. In diamond prices dataset the attributes such as colour, clarity had a natural ordering and eight levels each.

This presented a dilemma. Transforming these ordinal variables using one hot encoding would lead to loss of useful information as independent categorical representation would ignore the natural ordering of these variables. On the other hand, representing them with discreate numerical values would be problematic for any model that belongs to Scikit learn's family of liner models and KNN.

Tree based models only use values of features to split the data thus treat both ordinal and categorical variables exactly the same.

Refence papers [11] and [12] ranked Random Forest as the best machine learning model for predicting diamond prices while [9] ranked it as the best all purpose binary classifier. With enough trees in the forest, Gradient boosting can do as well as, if not better than Random Forest. AdaBoost was chosen for its resistance to overfitting [7]. Decision Tree itself was overlooked because boosting almost always leads to better predictions.

Finally, Scikit learn's implementation of Random Forest, AdaBoost and Gradient Boost gives impurity-based feature importances scores that can aid in model interpretability

## IV. METHODOLOGY

This paper followed a KDD methodology for data mining on all three datasets.

### IV.1 DATA EXPLORATION

*Dataset 1: RMTR*

This dataset had 169,669 rows and 2 columns. First column contained audience review as text and the corresponding star rating. The review text had punctuations, numbers, html tags and emoticons. The star rating was split into 10 levels ranging from half a star to 5 stars.

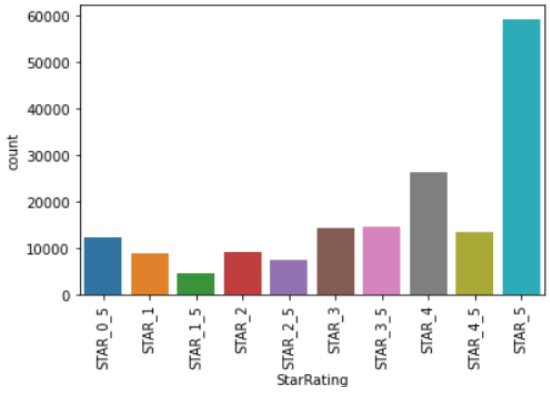



*Figure 1: Frequencies of star rating*

5-star reviews accounted for the lion's share followed by 4-star and 4.5-star. Overall, roughly 25% of the reviews were below 3 stars and 75% were 3 stars or above, Figure 1.

The shortest review was only 10 characters long whereas the longest review was 55,269 characters long. The average length of a review was 218 characters and the majority of the reviews were less than 600 characters long.

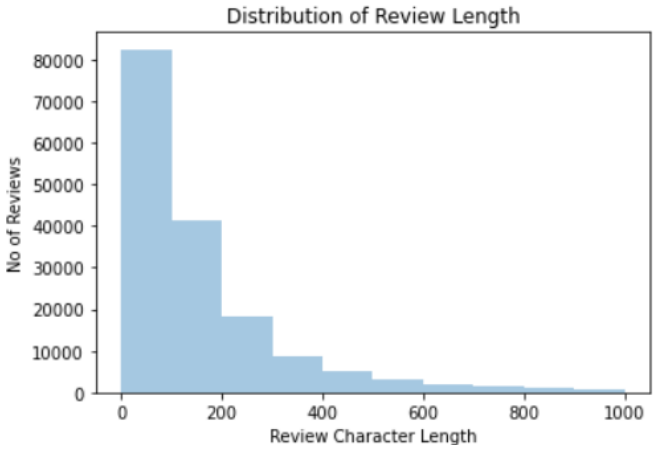



*Figure 2: Distribution of review length*

*Dataset 2: APS*

APS dataset consisted on training and test datasets consisting of 103,904 and 25,976 rows respectively. Arrival delay column had some missing values. It had 4 nominal variables, 4 continuous numerical variables and 12 ordinal variables.

The target variable Satisfaction had 2 levels, 43.4% satisfied and 56.6% neutral or dissatisfied

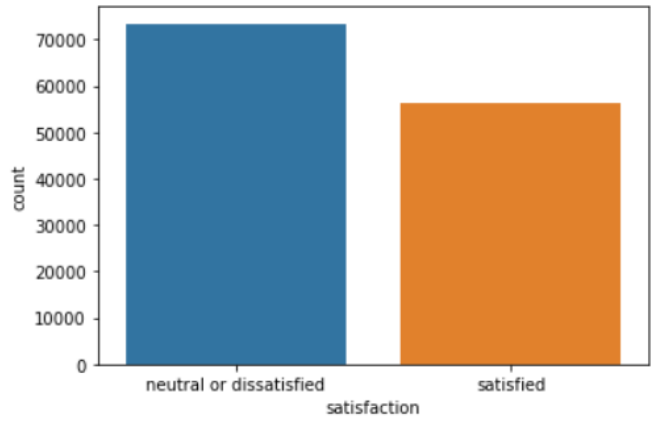



*Figure 3: Passenger satisfaction*

Bar charts of categorical variables grouped by the target variable showed customer satisfaction was lower among disloyal customers, personal travel and economy class passengers, Figure 4.

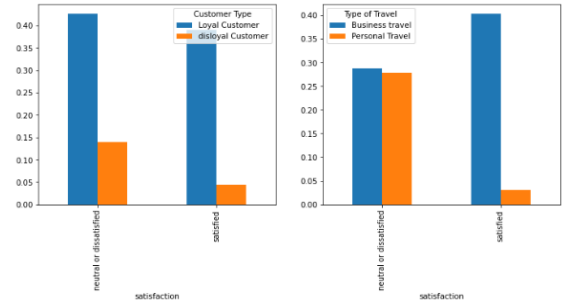



*Figure 4: Customer type, travel type vs target variable*

The continuous variable Age was nearly normally distributed while the distribution of flight distance, departure and arrival delay were right skewed.

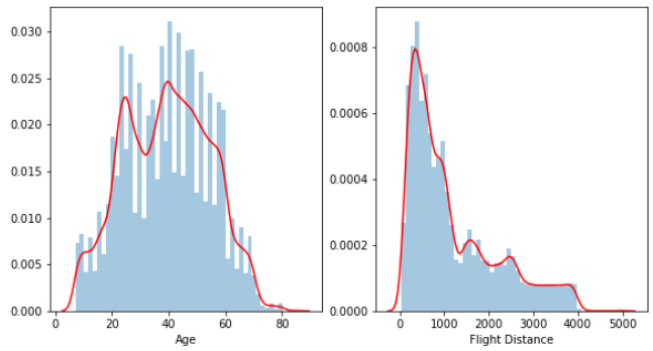



*Figure 5: Distribution of Age and Flight distance*

*Dataset 3: BDP*

BDP dataset had 21,245 rows and 14 columns. This dataset had four numerical variables and 8 ordinal variables. The numerical variables were carat, depth, lxwRatio and table.

The bulk of the diamonds were between 1 and 2 carats. The depth of diamonds was nearly normally distributed.

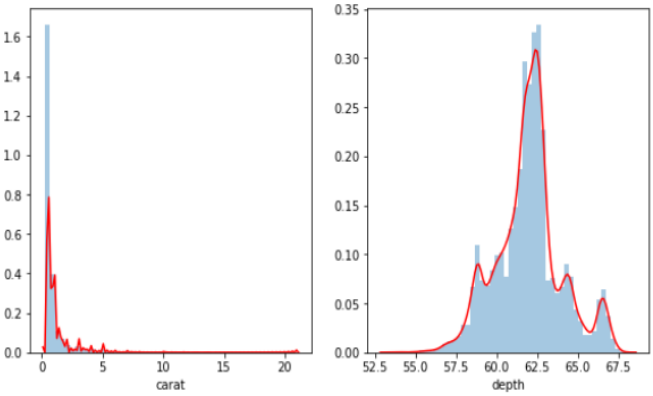



*Figure 6: Distribution of Age and Flight distance*

The ordinal variables colour and clarity were fairly evenly distributed across all different levels. However, some ordinal variables such as fluorescence had a few levels with a very low frequency, Figure 8.

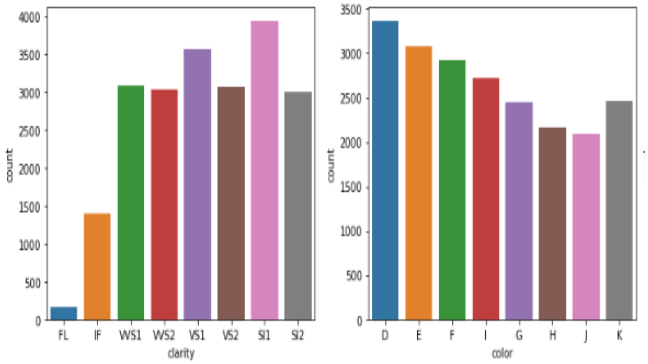



*Figure 7: Frequencies of colour and clarity*

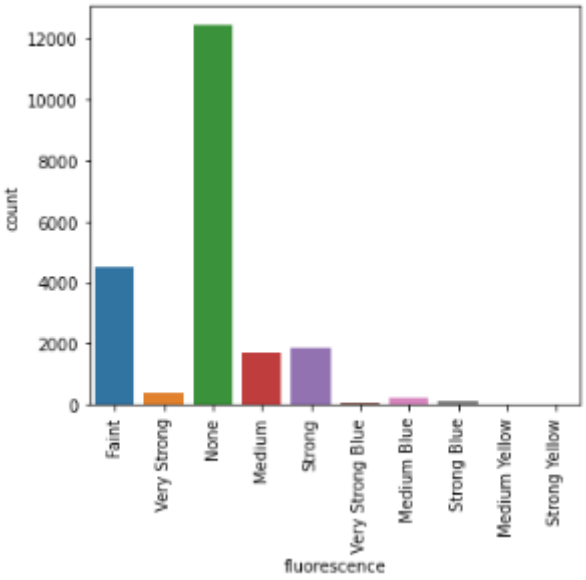



*Figure 8: Fluorescence value counts*

The distribution of the target variable price was heavily skewed to the right. It had a mean of 805 dollars and upper and lower bounds of 250 dollars and 2.56 million dollars respectively. Moreover 99% of the diamond prices were under 131,575 dollars. Figure 9 shows the distributions of log diamond prices.

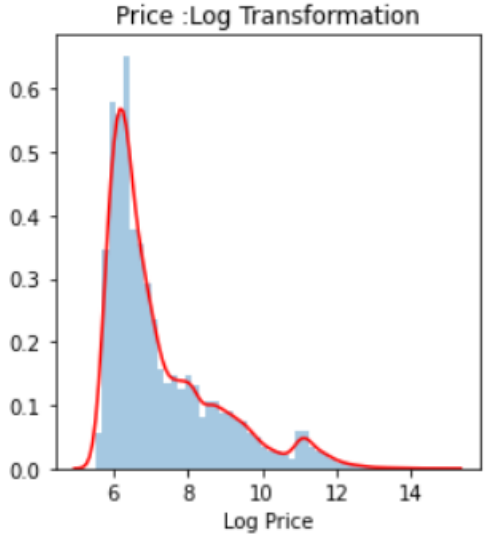



*Figure 9: Distribution of Log diamond prices*

## IV.2 DATA PREPROCESSING & TRANSFORMATION

*Dataset 1: RMTR*

The target variable sentiment was derived from the star rating. Sentiment of reviews with star rating less than or equal to 2.5 were defined as negative and star rating of 3 and above as positive.

Preprocessing steps involved case normalization, removal of numbers, punctuations, html tags, underscore and other special characters using regular expressions. The cleaned reviews corpus had a vocabulary of 69,563 words.

Stemming was applied to covert words into their root form. This reduced the number of unique words by over 20,000 to 48,678.

The word cloud of positive reviews contained words associated with positive sentiment such as great, good, best, fun etc.


*Figure 10: Word cloud of positive reviews*

The word cloud of negative reviews was a little puzzling at first. While it contained words such as bad, bored and worst, it also featured words such as good, great and funny. One explanation is negative reviews often contains negotiation (for example not good, not great, not funny) and looking at term frequency alone has taken these words out of context. Therefore, it was decided both unigrams and bigrams would be included in the word representation.


*Figure 11: Word cloud of negative reviews*

Another interesting observation was both positive and negative word clouds featured some domain specific words such as movie, film, act, watch and scene. As a result, a custom stopwords list was created by removing words that convey negation (such as not and never) from the standard English stopwords and including the domain specific most frequent words found in the word clouds.

Random down-sampling technique was used to create a balanced dataset consisting of 42,138 positive and negative reviews each.

Finally, a fixed 1,000 most frequent term vocabulary was used to create both bag-of-words and tf-idf word representations. Figure 12 shows the top 50 words featured in the bag-of-words representation.

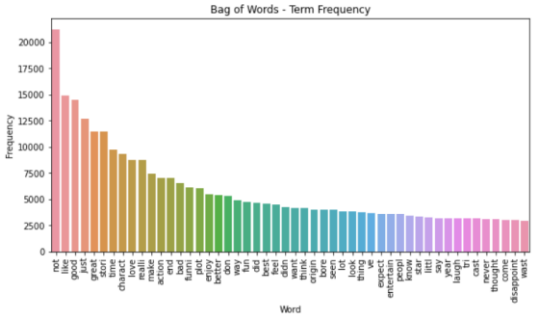



*Figure 12: Top 50 most frequent words in Bag-of-words*

*Dataset 2: APS*

The following steps were taken in data cleaning and transformation:

i. Missing values were filled with zero: Arrival delay had missing values for flights that that arrived on time. Rows with missing values were filled with zero.

ii. Binarized target variable to 0 and 1, 1 meaning satisfied

iii. Dummy encoding of categorical variables.

iv. Dropped encoded and irrelevant columns from the dataset.

The ordinal variables had numerical values between 0 and 5, therefore no transformation was applied to them.

*Dataset 3: BDP*

Preprocessing and transformation steps involved:

i. Removal of comma, space and dollar sign from price column and converting it a float

ii. Log transformation of the target variable price

iii. Combing low frequency levels of categorical variables into one group

iv. Mapping string labels of ordinal variables into discrete integer values

v. Dropping the *Id* column that contained a unique diamond identifier and other encoded categorical variables from the dataset

### IV.3 DATA MINING

*Dataset 1: RMTR*

The dataset was split into 80% training and 20% test in a stratified fashion. The machine learning models Linear SVM, multinomial Naive Bayes and Random Forest were trained with bag-of-words representation. 5-fold cross validation accuracy, test set accuracy, precision, recall, F1 score and Cohen's Kappa were calculated for each model.

Then the whole process was repeated again with tf-idf representation. A hyper parameter optimization of best model was carried out to see if the performance could be improved.

The impact having more features on accuracy was studied by increasing the most frequent term vocabulary from 100 to 10,000

A model with approximately 200 features was built to see if feature selection can help reduce model complexity without seriously affecting the accuracy.

Finally, the all three machine learning models with tf-idf representation was combined into one voting classifier to evaluate the performance of a collective prediction approach. The rationale behind the voting classifier was simple, if each model is making mistake on different instances then combining all 3 would lead to better predicative power.

*Dataset 2: APS*

Random Forest, Gradient Boost and AdaBoost were fitted to the training data. 5-fold cross validation accuracy, test set accuracy, precision, recall, F1 score and Cohen's Kappa were calculated for each model.

*Dataset 3: BDP*

The dataset was split into 80% training and 20% test sets. Random Forest regressor, Gradient Boosting regressor and AdaBoost regressor were trained using the training set. 5-fold cross validation r-square, test set r-square, MSE, RMSE and MAE were calculated for each model.

### IV.4 RESULTS & EVALUATION

*Dataset 1: RMTR*

Results summary of all 6 models is shown in Table 2

| Classifier | Avg CV Score | Accuracy | Precision | Recall | F1 Score | Kappa |
|---|---|---|---|---|---|---|
| Random Forest BOW | 0.821 | 0.817 | 0.821 | 0.812 | 0.816 | 0.635 |
| Naive Bayes BOW | 0.830 | 0.828 | 0.821 | 0.838 | 0.830 | 0.656 |
| Linear SVC BOW | 0.837 | 0.840 | 0.855 | 0.819 | 0.837 | 0.681 |
| Random Forest tf-idf | 0.830 | 0.831 | 0.841 | 0.815 | 0.828 | 0.661 |
| Naive Bayes tf-idf | 0.828 | 0.829 | 0.844 | 0.808 | 0.826 | 0.659 |
| Linear SVC tf-idf | 0.840 | 0.847 | 0.850 | 0.842 | 0.846 | 0.694 |

*Table 2: RTMR results summary*

Bag-of-words: Linear SVC outperformed Naïve Bayes and Random Forest with highest scores in cross validation accuracy, test set accuracy, precision, F1 score and Kappa. In fact, the only evaluation metric in which it lost to Naïve Bayes was Recall.

Tf-idf: Linear SVC beat other 2 ML methods in every single evaluation metric.

Among the 6 models, Linear SVC with tf-idf representation scored the highest in all metrics but precision. Linear SVC with BOW had the highest precision of 0.855 compared to 0.850 for Linear SVC with tf-idf.

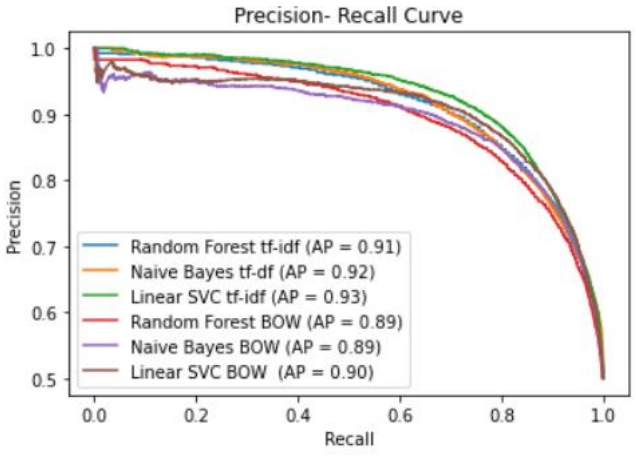



*Figure 13: Precision-recall curves RMTR*

However, Linear SVC with tf-idf had a higher area under the precision-recall curve than the other 5 models including Linear SVC with BOW, Figure 13. Similarly, Linear SVC with tf-idf representation also had highest area under ROC curve. Linear SVC with tf-idf was the overall best model on this dataset.

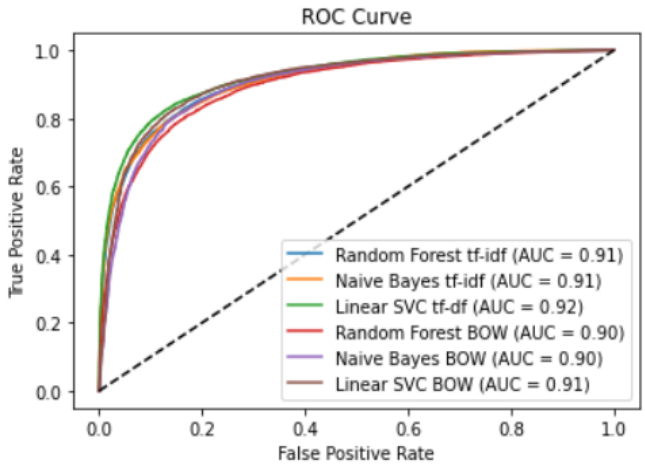



*Figure 14: ROC AUC curve RMTR*

Linear SVC with tf-idf correctly classified 84% of the positive reviews and 85% of the negative reviews. The model achieved the exact same accuracy score on positive reviews that were excluded from both the training and test data in random down-sampling. This provided further evidence that the model generalized well.

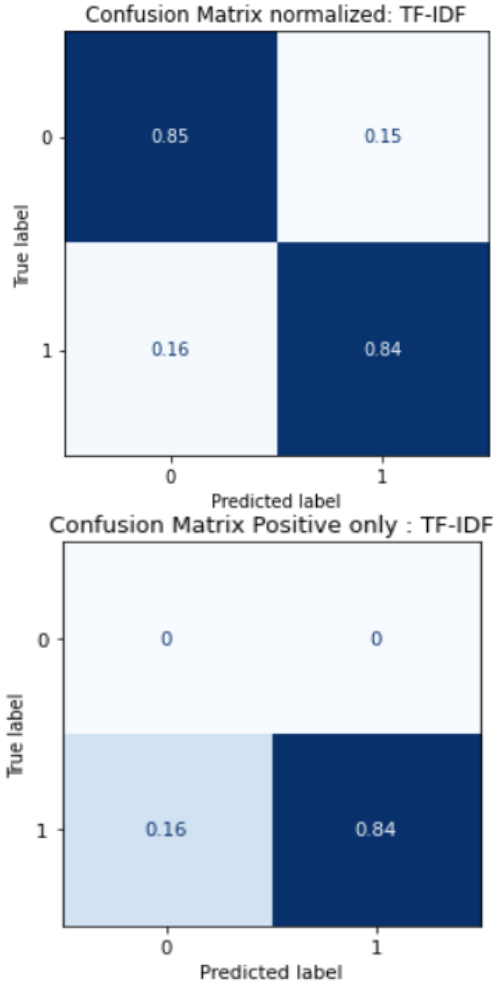



*Figure 15: Confusion Matrix: Linear SVC tf-idf*

Figure 16 shows the top 20 features with highest positive and negative coefficients in BOW representation. One interesting observation is, even though bigrams accounted for less than 10% of the original 1000 most frequent terms, 7 bigrams made it to the top 40 (17.5%). Of these 7 bigrams, 4 contained the word 'not' (not bad, not recommend, not funny and not worth). Moreover, the bigram 'not worth' had the highest negative coefficient. In contrast, only 2 bigrams appeared among the top 20 most positive and negative tf-idf features.

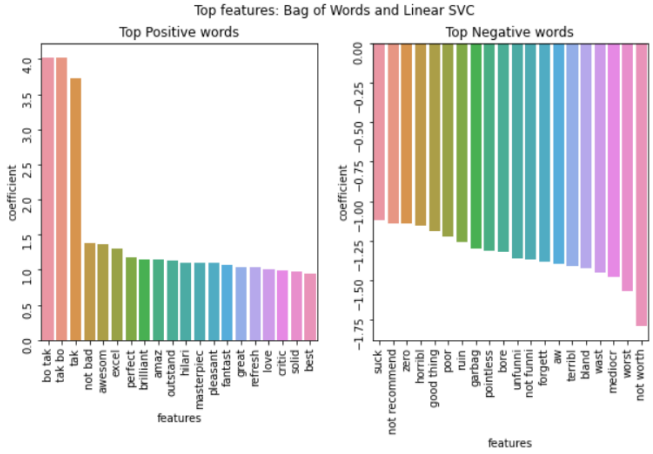



*Figure 16: Top features Linear SVC with BOW*

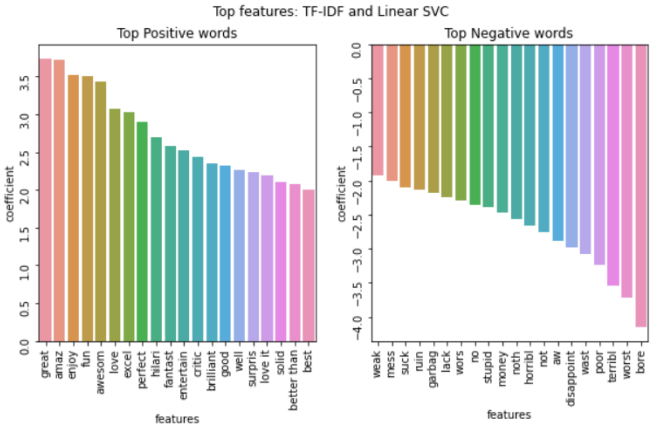



*Figure 17: Top features Linear SVC with tf-idf*

Hyperparameter optimization of the regulation parameter alpha of Linear SVC with tf-idf did not result in a better model. The default value of 0.0001 was in fact the most optimal.

Varying the number of features from 100 to 10,000 showed accuracy of Linear SVC with tf-idf model could be improved from 84.5% to 86.0% with 5000 features. However, including more than 5000 features did not result in any gain in accuracy.

| No of features | Accuracy |
|---|---|
| 100 | 0.726 |
| 200 | 0.781 |
| 500 | 0.831 |
| 1000 | 0.845 |
| 2000 | 0.853 |
| 5000 | 0.860 |
| 10000 | 0.860 |

*Table 3: Linear SVC with tf-idf accuracy vs No of features*

The model with 10,000 features had only 215 features with coefficients higher than 1 or less than -1. Linear SVC trained with a custom vocabulary consisting of these 215 features achieved an accuracy of 83.4%. For comparison, limiting the

features to only 200 would have resulted in an accuracy of only 78.1%, Table 3.

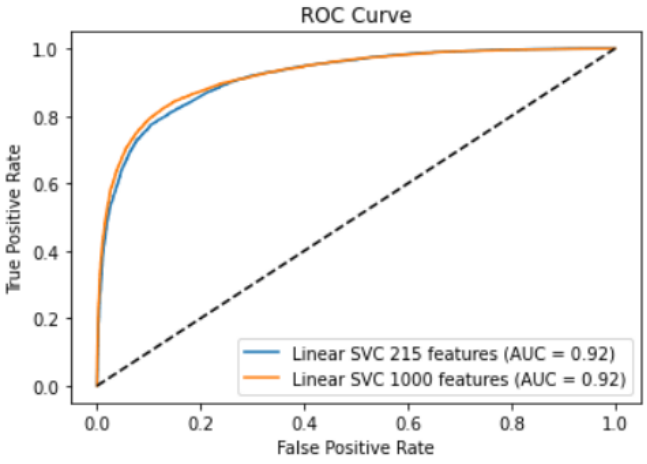



*Figure 18: ROC AUC curve 215 features vs 1000 features*

The voting classifier combined the predictions of Linear SVC, multinomial Naive Bayes and Random Forest classifier using a majority voting rule. Compared to the best model, the voting classifier was slightly better at predicting negative reviews and slightly worse at correctly classifying positive class. Overall, the voting classifier produced an accuracy score of 84.4% compared to 84.6% for the best model.

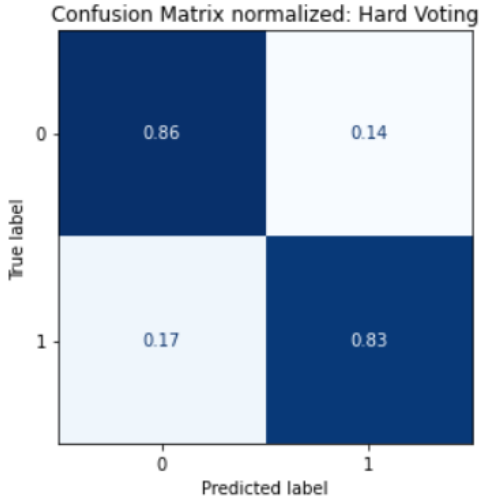



*Figure 19: Confusion Matrix: Voting Classifier with tf-idf*

For completeness, the accuracy scores of the two lexicon-based approaches that performed the best in the domain of movie reviews sentiment classification in reference paper [2] were also calculated. VADER and TextBlob correctly classified 73.4% and 73.2% of the test data respectively.

*Dataset 2: APS*

Summary of data mining is shown in Table 4.

| Classifier | Avg CV Score | Accuracy | Precision | Recall | F1 Score | Kappa |
|---|---|---|---|---|---|---|
| Random Forest | 0.962 | 0.963 | 0.972 | 0.942 | 0.957 | 0.924 |
| Ada Boost | 0.928 | 0.926 | 0.922 | 0.909 | 0.915 | 0.850 |
| Gradient Boost | 0.942 | 0.941 | 0.945 | 0.920 | 0.932 | 0.881 |

*Table 4: APS results summary*

Random Forest beat the other two ensemble learning methods on every single evaluation metric and achieved a cross validation and test accuracy of 0.962 and 0.963 respectively.

Random Forest also had the highest area under the curve in both Precision-Recall and ROC plots.

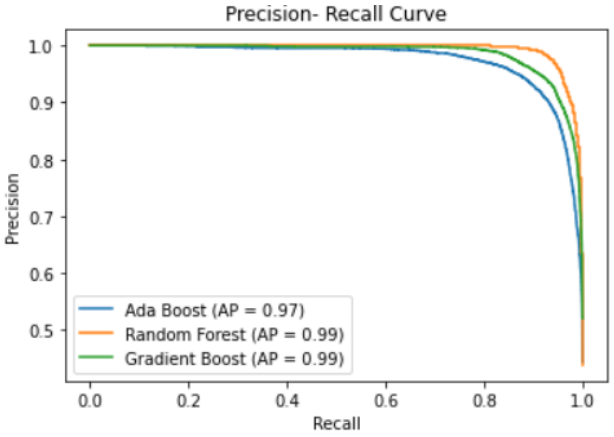



*Figure 20: Precision-Recall curves APS*

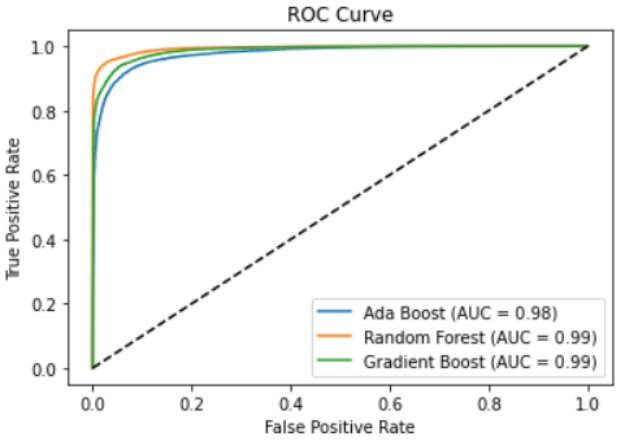



*Figure 21: ROC curves APS*

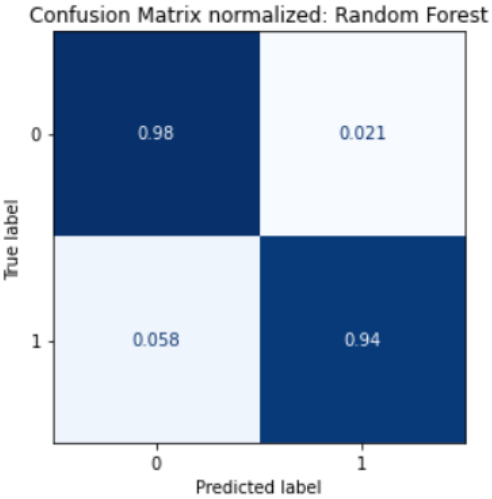



*Figure 22: Confusion Matrix Random Forest*

Random Forest correctly classified 98% of the negative class and 94% of the positive class, Figure 22.

Increasing the number of estimators from the default setting of 100 made little difference to the accuracies of Random Forest and AdaBoost. However, accuracy of Gradient Boost increased with number of estimators and reached an accuracy

score that was similar to Random Forest with 1000 estimators and flatlined after that, Figure 23.

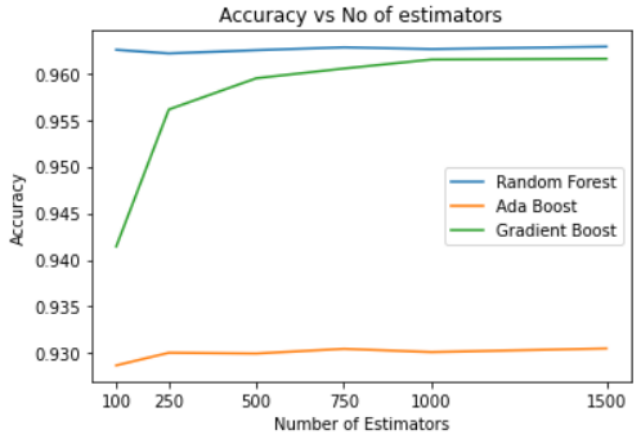



*Figure 23: Accuracy vs No of estimators*

Both Random Forest and Gradient Boosting identified similar features as the top contributors to Air passenger customer satisfaction. These were Online Boarding, Inflight Wifi, whether the travel was business related, whether the passenger flew business or economy, Inflight entertainment and Leg room.

The main difference was, almost all factors made some contribution to the prediction of Random Forest, whereas in Gradient Boosting only the top 7 factors had the biggest weights. Therefore, the Gradient boosting classifier was chosen for feature selection.

A parsimonious model with only 8 features was built using automatic recursive feature elimination (RFE). This parsimonious model scored an accuracy of 94.4%.

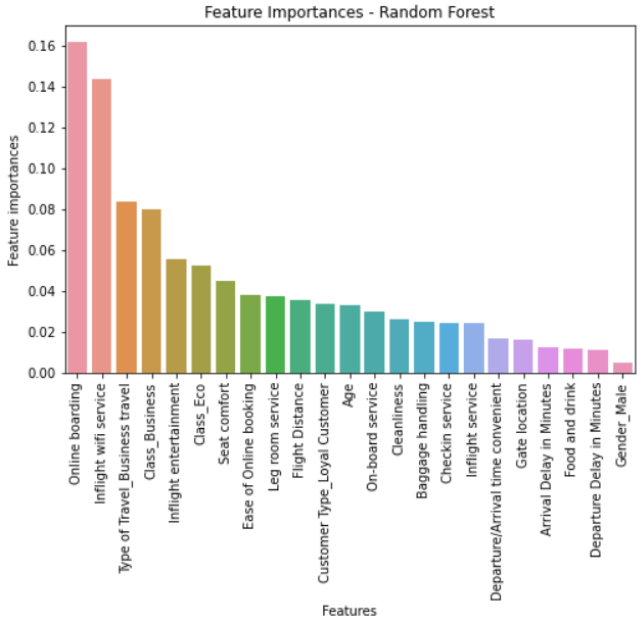



*Figure 24: Feature Importances Random Forest*

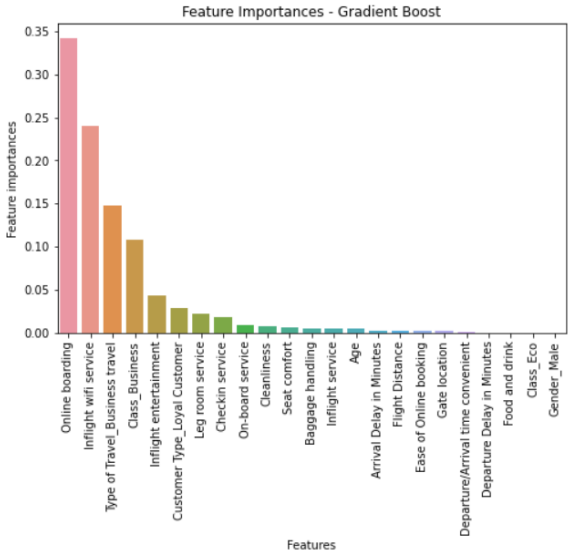



*Figure 25: Feature Importances Gradient Boosting*

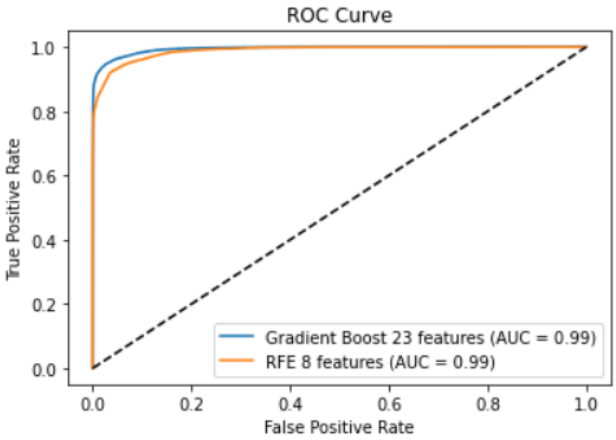



*Figure 26: ROC Curves Parsimonious model and GB*

AdaBoost identified a completely different set of features as most important for Airline passenger satisfaction. These factors were flight distance, departure delay, arrival delay and age of the passenger. Most importantly, there was zero overlap between the factors identified by AdaBoost and the other 2 models.

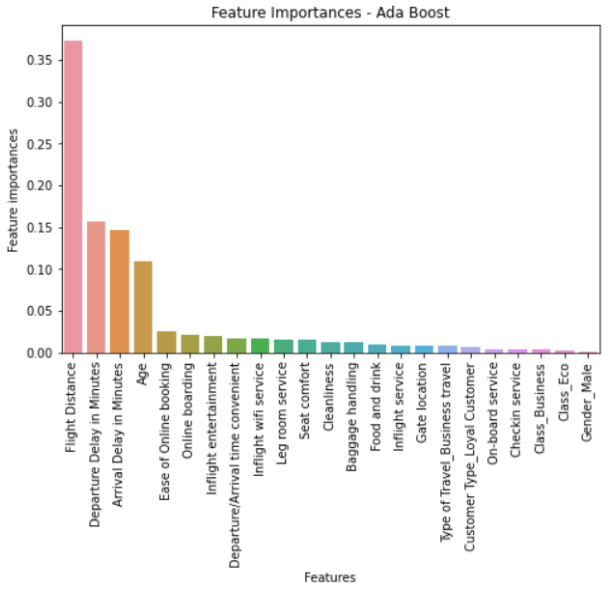



*Figure 27: Feature Importances AdaBoost*

*Dataset 3: BDP*

| Regressor | Crossval R-sq | R-square | MSE | RMSE | MAE |
|---|---|---|---|---|---|
| Random Forest | 0.992 | 0.993 | 0.017 | 0.130 | 0.095 |
| Gradient Boost | 0.991 | 0.991 | 0.020 | 0.142 | 0.110 |
| Ada Boost | 0.959 | 0.961 | 0.089 | 0.298 | 0.248 |

*Table 5: BDP results summary*

Strictly speaking the winner was Random Forest with the highest scores in cross validation and test set r-square and lowest scores in MSE, RMSE and MAE. However, the difference between Random Forest and Gradient Boost in all evaluation metrics was extremely small. Therefore it was a tie for the first place.

Scatterplot of log price against model prediction showed the datapoints mostly fell along the 45-degree diagonal line. When plotted against log prices, the residual errors were randomly distributed around the horizontal axis, Figure 29.

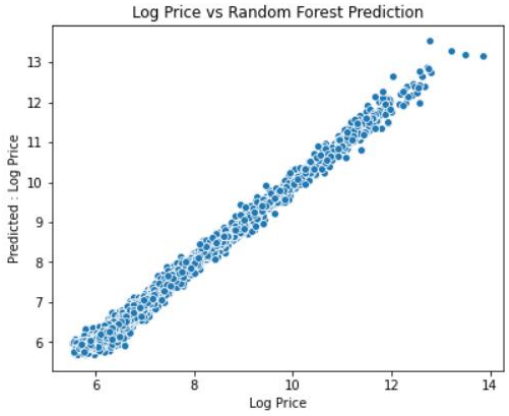



*Figure 28: Log price vs Random Forest Prediction*

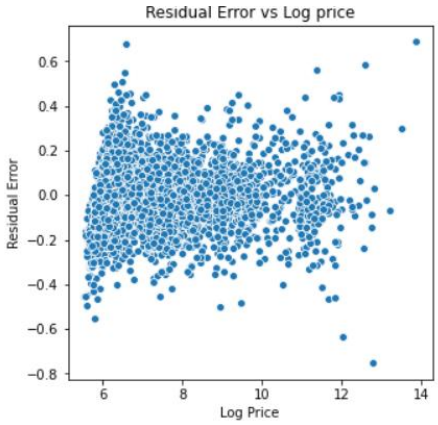



*Figure 29: Residual erros vs Target Variable*

All three ensemble learning methods ranked Carat, Color and Clarity as the top 3 determinants of price a diamond. However, Random Forest ranked the attribute depth in number 4 position, where as Gradient Boost and AdaBoost ranked Cut as the fourth most important feature. Domain knowledge suggests the four most important characteristics of a diamond are Carat, Color, Clarity and Cut, collectively known as the *4Cs of a diamond*.

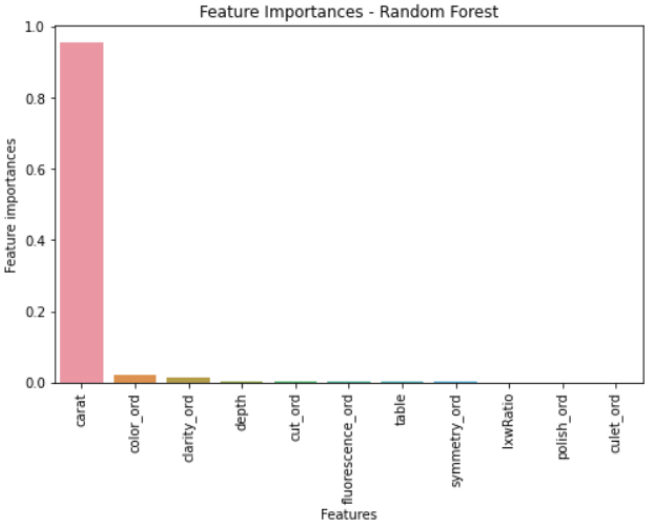



*Figure 30: Diamond price feature importances Random Forest*

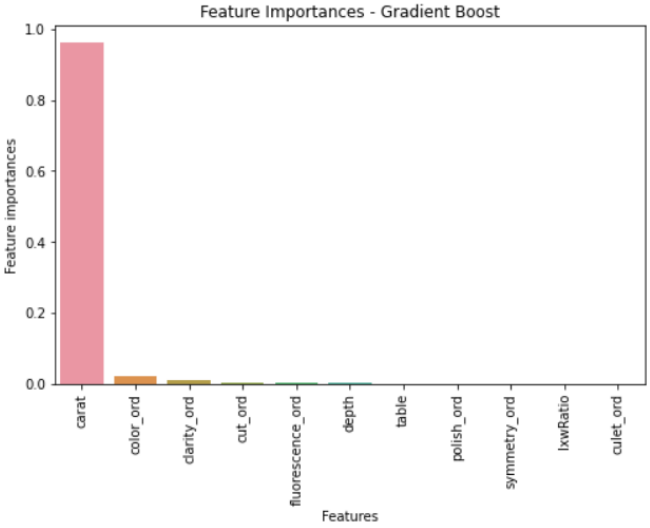



*Figure 31: Diamond price feature importances Gradient Boost*

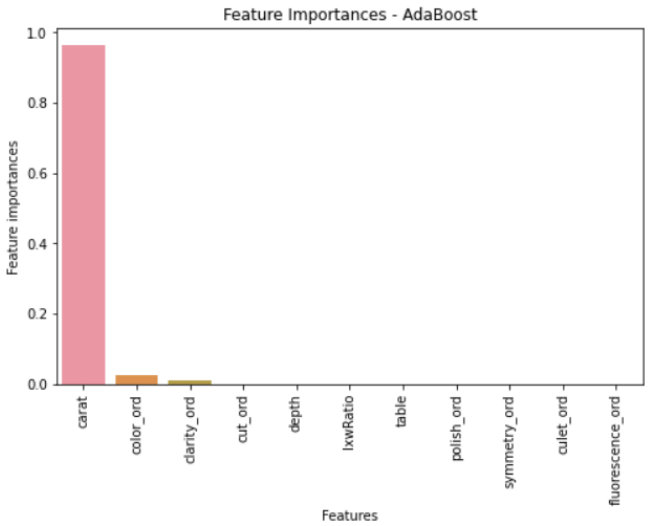



*Figure 32: Diamond price feature importances AdaBoost*

Because the features importances of Gradient boost was in line with the industry knowledge and it was equally as good as Random Forest, Gradient Boost was chosen for feature selection. A parsimonious model with only 4 features was built using automatic recursive feature elimination (RFE).

Unsurprisingly, the four features selected by RFE were in fact the 4Cs of a diamond. The r-square, MSE, RMSE and MAE values for this parsimonious model were 0.989,0.023,0.151 and 0.116 respectively, almost as good as the full model.

## V. CONCLUSIONS

*Dataset 1: RMTR*

Linear SVC produced the highest accuracy scores in both Bag-of-words and tf-idf representations. This was broadly consistent with the academic literature. ([1], [3], [4])

With highest scores in accuracy, recall, F1 score, Kappa, area under precision-recall and ROC curve, Linear SVC with tf-idf was the best classifier for Rotten tomatoes movie reviews sentiment classification. It achieved a highest accuracy of 86.0% with 5000 most frequent term vocabulary, which was higher than all the state-of-the art methodologies for movie reviews ([2]) that did not use word vectors. ([1]) or deep learning ([5]).

The academic consensus was mixed ([3], [4]) in the benefits of including bigrams in word representation. This report found some evidence to suggest inclusion of bigrams made a positive contribution to model accuracy when bag-of-words representation was used.

Naive Bayes, Random Forest and Linear SVC achieved accuracy scores of 82.9%, 83.1% and 84.7% with a fixed 1000 most common term frequency and tf-idf representation. Therefore, the differences in accuracies were small. However, a voting classifier with majority voting rule failed to outperform the best model. This implied there were a large number of instances in which 2 or all 3 models were getting the classification wrong.

Finally, the all 3 machine learning models easily outclassed lexicon-based approaches in movie review sentiment classification. This was consistent with the literature review. ([2], [3], [7])

*Dataset 2: APS*

With highest score in every single evaluation metric, Random Forest was the clear winner on this data set.

Increasing the number of estimators had a positive effect on the predictive power of the Gradient Boost model but made little difference to Random Forest and AdaBoost.

Random Forest and Gradient Boost identified similar factors as most important to Air passenger customer satisfaction namely Online Boarding, Inflight Wifi, whether the travel was business related, whether the passenger flew business or economy, Inflight entertainment and Leg room.

This report found feature importances scores from Gradient Boost classifier to be more useful in feature selection.

*Dataset 3: BDP*

Random Forest and Gradient Boost achieved impressive R-square values of 99.3% and 99.1% in predicting log diamond prices. With similar low scores on MSE, RMSE, MAE it was a tie for the first place.

The four most important determinants of a diamond were Carat, Colour, Clarity and Cut. Interestingly, domain knowledge also identifies these four as the most important characteristics of a diamond.

## VI. FUTURE WORK

*Dataset 1: RMTR*

There are a few ways in which text preprocessing could be improved:

i. Typos and spelling mistakes are often found in online reviews. Therefore, adding Spelling correction to preprocessing steps would be helpful.

I. Emoticons such as smiley face are used for expressing sentiment in reviews. Rather than removing all special characters, incorporating commonly used emoticons in vocabulary could lead to better predictions.

Word vectors or word embeddings is a multi-dimensional vector representation of a word. This representation can be used find relationship between words based on their relative proximity (semantic similarity). to each other. Using word vector representation could be very effective in sentiment classification.

*Dataset 2: APS*

This paper chose to represent the ordinal variables as discrete numerical values. This effectively restricted the choice of classifiers. Explore other machine learning methods by using one hot encoding may give interesting results.

The dataset only had numerical values for passenger feedback. By collecting textual passenger feedback, NLP techniques could be used to develop further insights into factors that influence Air passenger satisfaction.

*Dataset 3: BDP*

The restrictions imposed by Bluenile severely limited the amount of data that could be downloaded using its public API. The dataset used in this report contained less than 10% of what was available on Bluenile website. It would be possible to extract the entire list of diamond prices on Bluenile website by using Selenium.

## REFERENCES


[1] Maas, Andrew & Daly, Raymond & Pham, Peter & Huang, Dan & Ng, Andrew & Potts, Christopher. (2011). Learning Word Vectors for Sentiment Analysis. 142-150.

[2] Song, C., Wang, X., Cheng, P., Wang, J., & Li, L. (2020). SACPC: A framework based on probabilistic linguistic terms for short text sentiment analysis. Knowledge-Based Systems, 194, 105572. doi:10.1016/j.knosys.2020.105572

[3] Kharde, Vishal & Sonawane, Sheetal. (2016). Sentiment Analysis of Twitter Data: A Survey of Techniques. International Journal of Computer Applications. 139. 5-15. 10.5120/ijca2016908625

[4] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP 02, 2002

[5] Kaur, Jaspinder & Dara, Rozita & Matsakis, Pascal. (2018). Sentiment Classification of Short Texts. 10.1007/978-3-319-92058-0_73.

[6] Sajib, Mahamudul & Shargo, Shoeib & Hossain, Md. (2019). Comparison of the efficiency of Machine Learning algorithms on Twitter Sentiment Analysis of Pathao. 1-6. 10.1109/ICCIT48885.2019.9038208.

[7] L. Augustyniak, T. Kajdanowicz, P. Kazienko, M. Kulisiewicz, and W. Tuliglowicz, "An Approach to Sentiment Analysis of Movie Reviews: Lexicon Based vs. Classification," Lecture Notes in Computer Science Hybrid Artificial Intelligence Systems, pp. 168–178, 2014.

[8] Lacic, Emanuel & Kowald, Dominik & Lex, Elisabeth. (2016). High Enough? Explaining and Predicting Traveler Satisfaction Using Airline Review.

[9] Hamza, Mounir & Larocque, Denis. (2005). An empirical comparison of ensemble methods based on classification trees. Journal of Statistical Computation and Simulation - J STAT COMPUT SIM. 75. 629-643. 10.1080/00949650410001729472

[10] Mease, David & Wyner, Abraham. (2008). Evidence Contrary to the Statistical View of Boosting. Journal of Machine Learning Research. 9. 131-156. 10.1145/1390681.1390687.

[11] Pandey, Avinash & Misra, Shubhangi & Saxena, Mridul. (2019). Gold and Diamond Price Prediction Using Enhanced Ensemble Learning. 1-4. 10.1109/IC3.2019.8844910

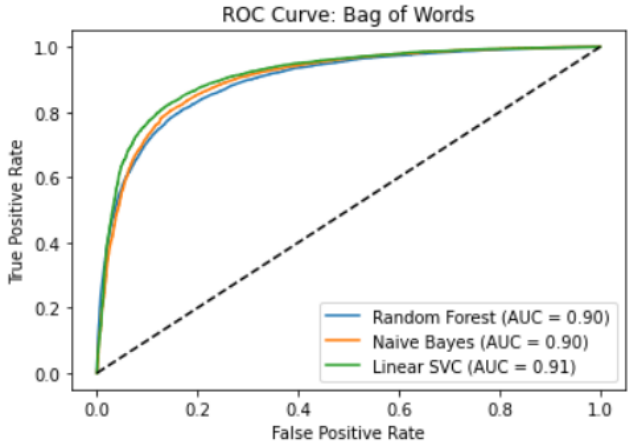
[12] Alsuraihi, Waad & Al-hazmi, Ekram & Bawazeer, Kholoud & Alghamdi, Hanan. (2020). Machine Learning Algorithms for Diamond Price Prediction. 150-154. 10.1145/3388818.3393715.

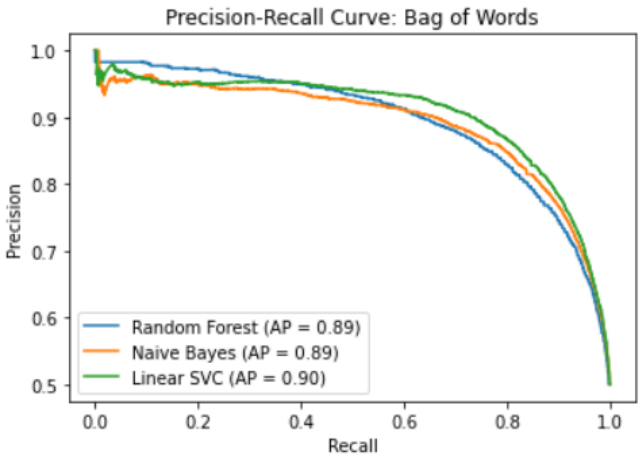
## APPENDIX

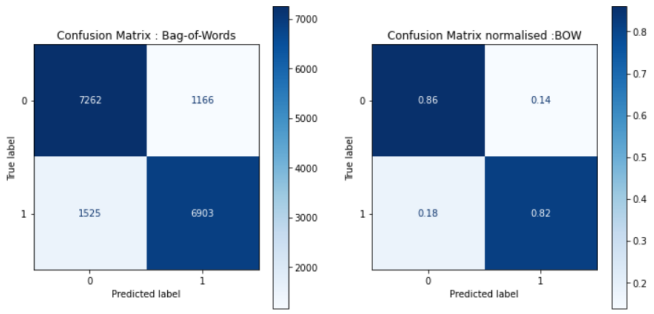
This section shows charts and Tables that were not included in the report write up.

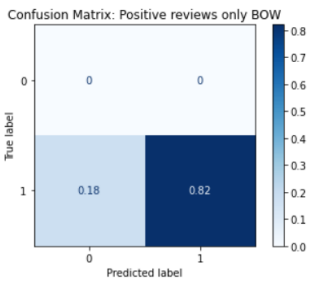
### *Dataset 1: RMTR*

| Classifier | Avg CV Score | Accuracy | Precision | Recall | F1 Score | Kappa |
|---|---|---|---|---|---|---|
| Random Forest BOW | 0.821 | 0.817 | 0.821 | 0.812 | 0.816 | 0.635 |
| Naive Bayes BOW | 0.830 | 0.828 | 0.821 | 0.838 | 0.830 | 0.656 |
| Linear SVC BOW | 0.837 | 0.840 | 0.855 | 0.819 | 0.837 | 0.681 |



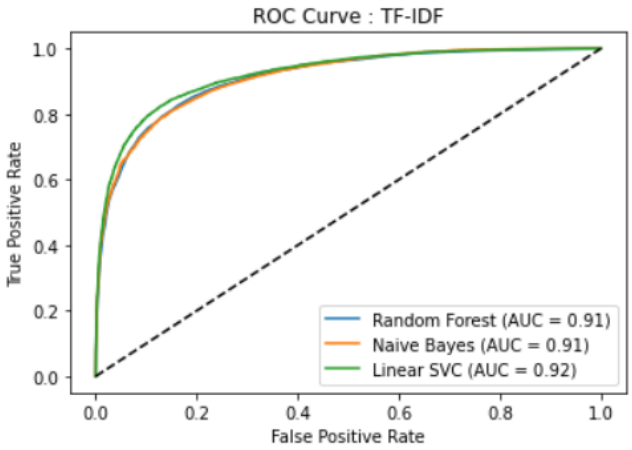



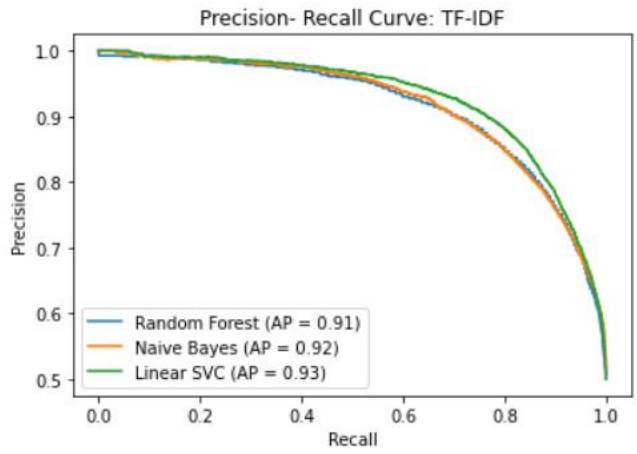



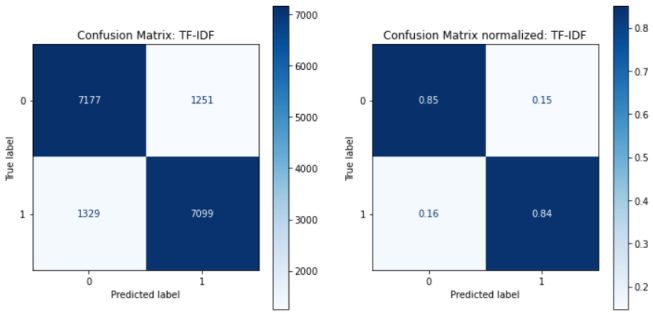



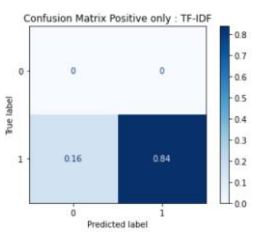

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.83 | 0.86 | 0.84 | 8428 |
| 1 | 0.86 | 0.82 | 0.84 | 8428 |
| accuracy | | | 0.84 | 16856 |
| macro avg | 0.84 | 0.84 | 0.84 | 16856 |
| weighted avg | 0.84 | 0.84 | 0.84 | 16856 |

| | Classifier | Avg CV Score | Accuracy | Precision | Recall | F1 Score | Kappa |
|---|---|---|---|---|---|---|---|
| 0 | Random Forest tf-idf | 0.830 | 0.831 | 0.841 | 0.815 | 0.828 | 0.661 |
| 1 | Naive Bayes tf-idf | 0.828 | 0.829 | 0.844 | 0.808 | 0.826 | 0.659 |
| 2 | Linear SVC tf-idf | 0.840 | 0.847 | 0.850 | 0.842 | 0.846 | 0.694 |

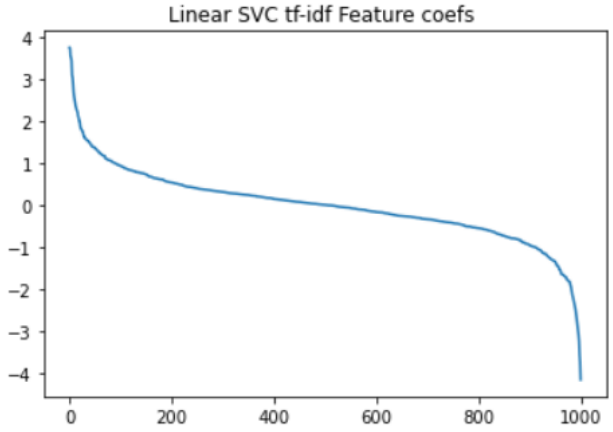


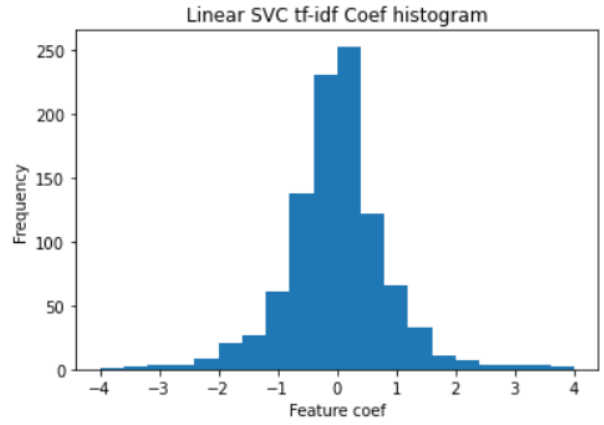



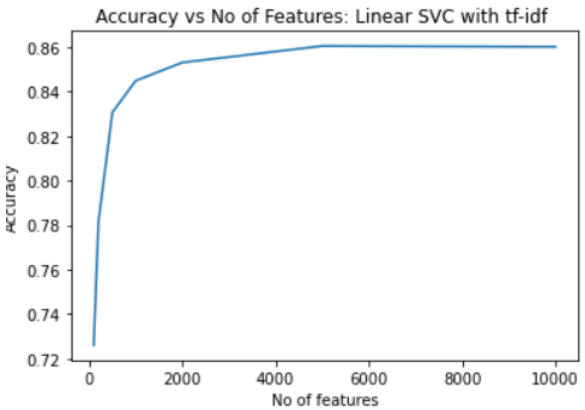



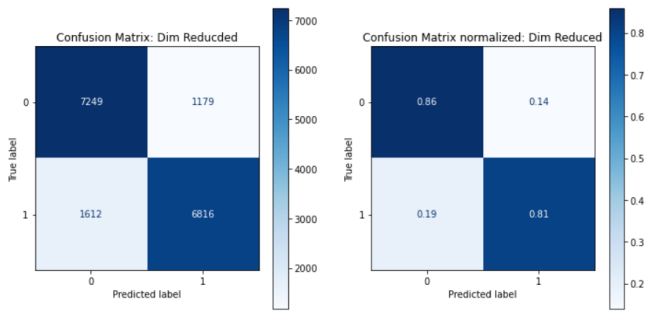



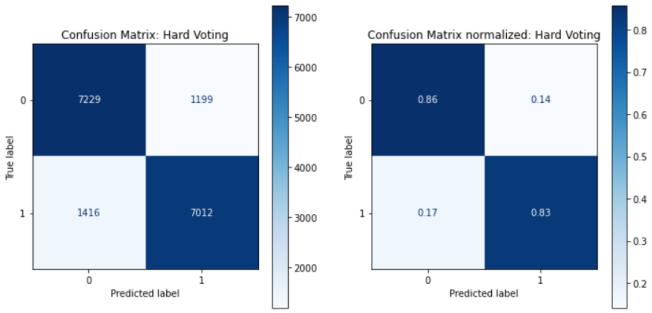
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.84 | 0.85 | 0.85 | 8428 |
| 1 | 0.85 | 0.84 | 0.85 | 8428 |
| accuracy | | | 0.85 | 16856 |
| macro avg | 0.85 | 0.85 | 0.85 | 16856 |
| weighted avg | 0.85 | 0.85 | 0.85 | 16856 |

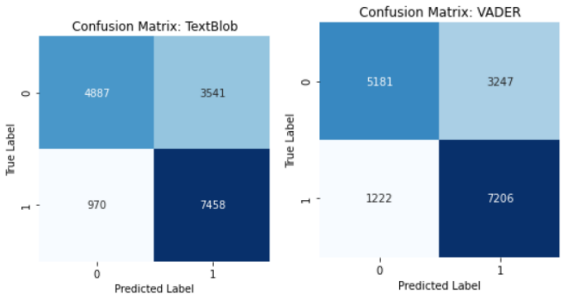


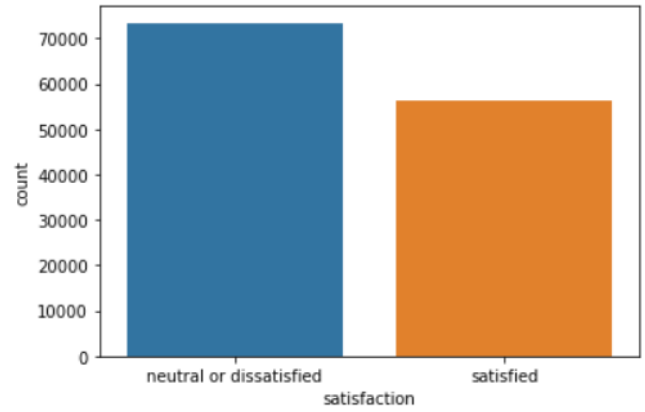



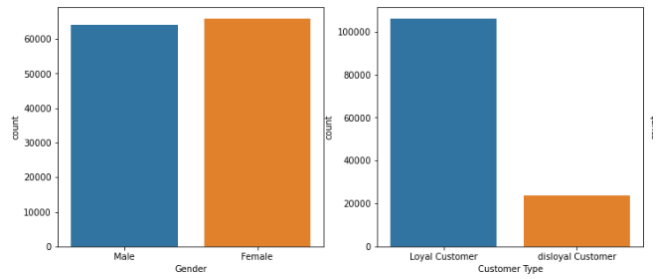



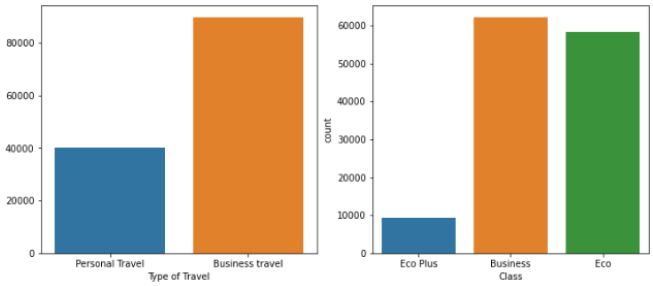



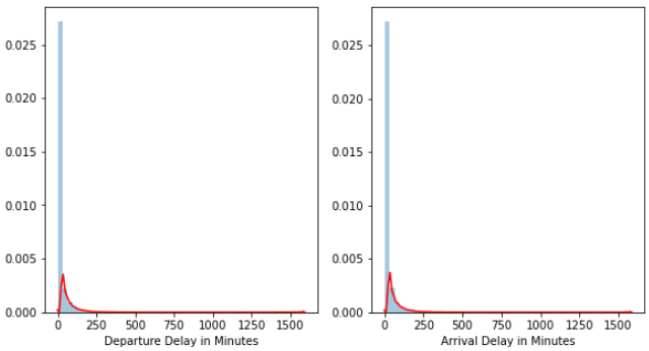



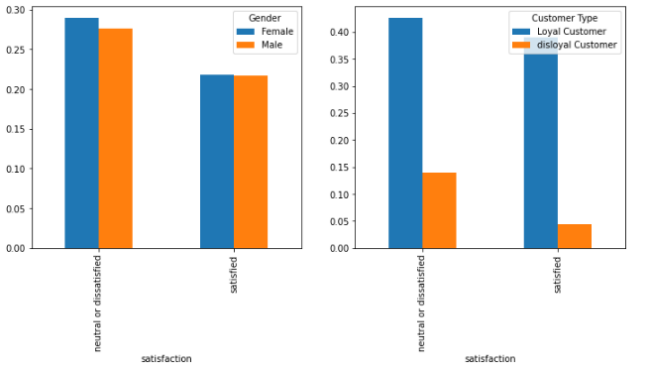

***Dataset 2: APS***

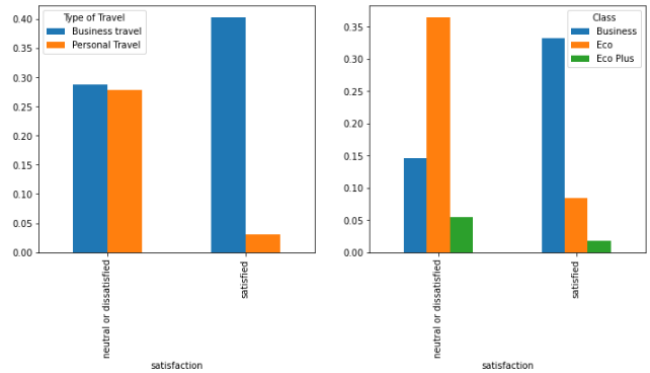


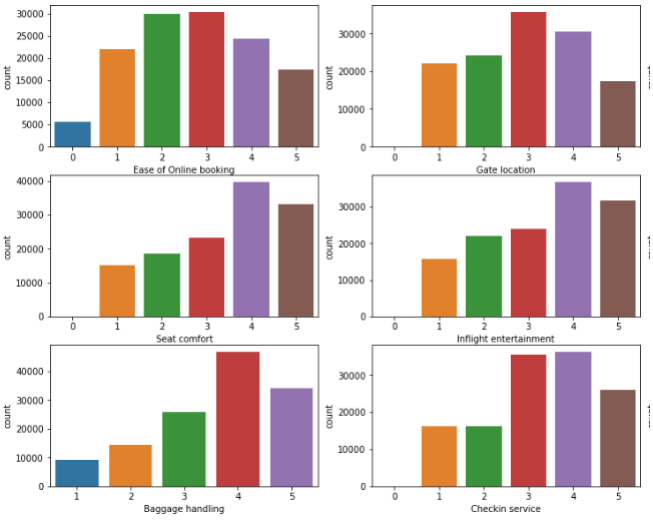



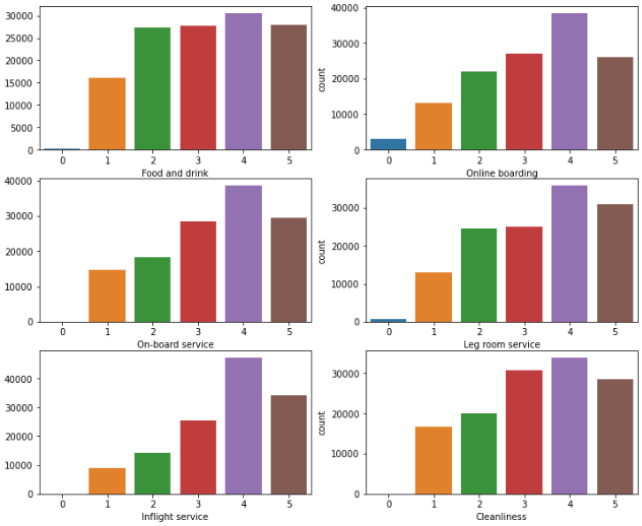



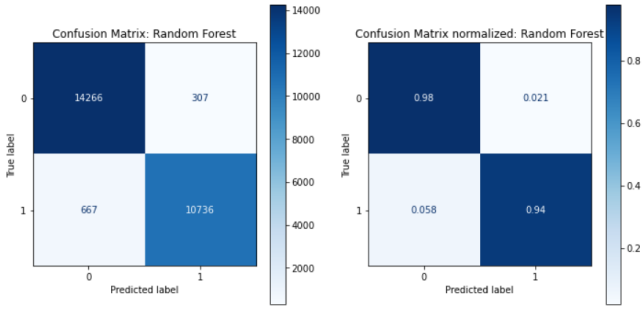


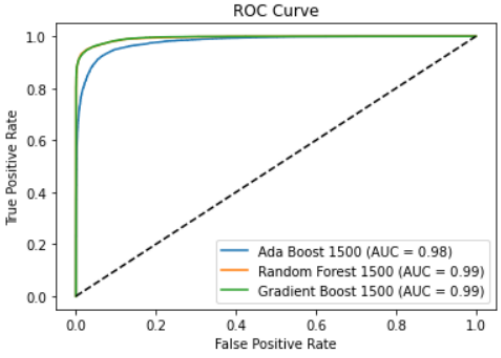



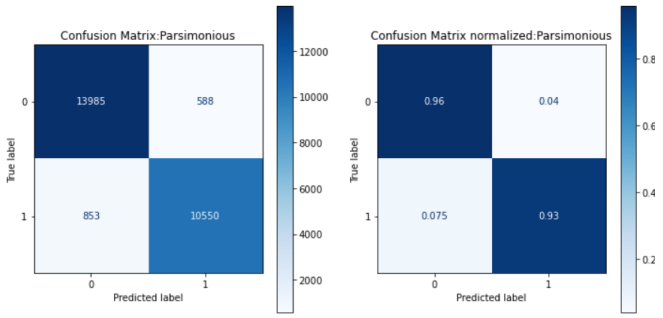



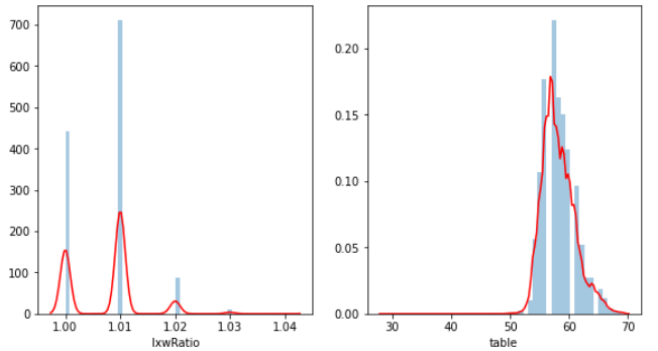



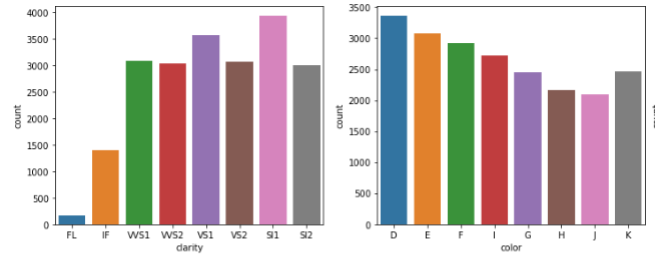

| | Classifier | Accuracy | Precision | Recall | F1 Score | Kappa | No of Trees |
|---|---|---|---|---|---|---|---|
| 0 | Random Forest 100 | 0.963 | 0.972 | 0.942 | 0.957 | 0.924 | 100 |
| 1 | Ada Boost 100 | 0.929 | 0.925 | 0.911 | 0.918 | 0.855 | 100 |
| 2 | Gradient Boost 100 | 0.941 | 0.945 | 0.920 | 0.932 | 0.881 | 100 |
| 3 | Random Forest 250 | 0.962 | 0.972 | 0.941 | 0.956 | 0.923 | 250 |
| 4 | Ada Boost 250 | 0.930 | 0.926 | 0.913 | 0.920 | 0.858 | 250 |
| 5 | Gradient Boost 250 | 0.956 | 0.963 | 0.936 | 0.949 | 0.911 | 250 |
| 6 | Random Forest 500 | 0.962 | 0.972 | 0.941 | 0.957 | 0.924 | 500 |
| 7 | Ada Boost 500 | 0.930 | 0.926 | 0.913 | 0.920 | 0.858 | 500 |
| 8 | Gradient Boost 500 | 0.959 | 0.968 | 0.939 | 0.953 | 0.917 | 500 |
| 9 | Random Forest 750 | 0.963 | 0.973 | 0.942 | 0.957 | 0.924 | 750 |
| 10 | Ada Boost 750 | 0.930 | 0.927 | 0.914 | 0.920 | 0.859 | 750 |
| 11 | Gradient Boost 750 | 0.961 | 0.968 | 0.941 | 0.954 | 0.920 | 750 |
| 12 | Random Forest 1000 | 0.963 | 0.972 | 0.942 | 0.957 | 0.924 | 1000 |
| 13 | Ada Boost 1000 | 0.930 | 0.926 | 0.914 | 0.920 | 0.858 | 1000 |
| 14 | Gradient Boost 1000 | 0.961 | 0.969 | 0.943 | 0.956 | 0.922 | 1000 |
| 15 | Random Forest 1500 | 0.963 | 0.972 | 0.942 | 0.957 | 0.924 | 1500 |
| 16 | Ada Boost 1500 | 0.931 | 0.927 | 0.914 | 0.920 | 0.859 | 1500 |
| 17 | Gradient Boost 1500 | 0.962 | 0.969 | 0.942 | 0.956 | 0.922 | 1500 |

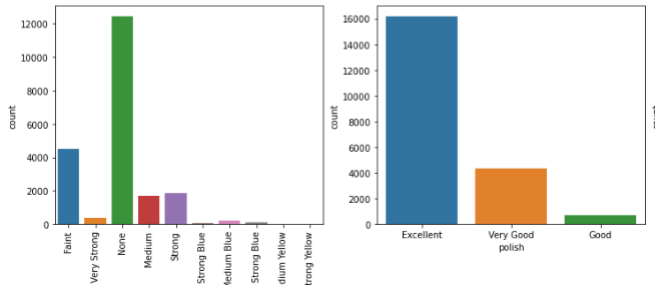


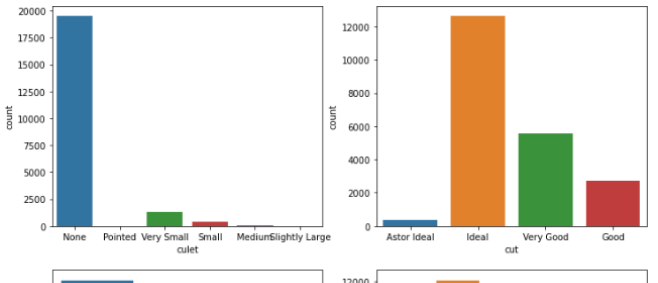


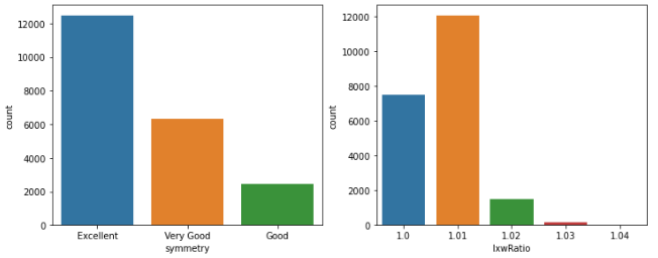

***Dataset 2: BDP***

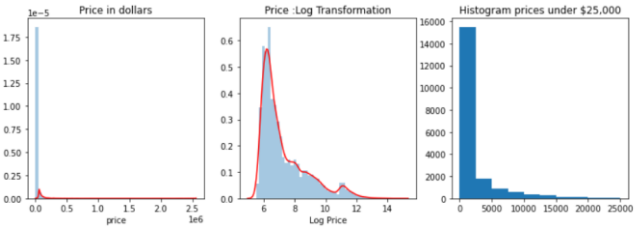


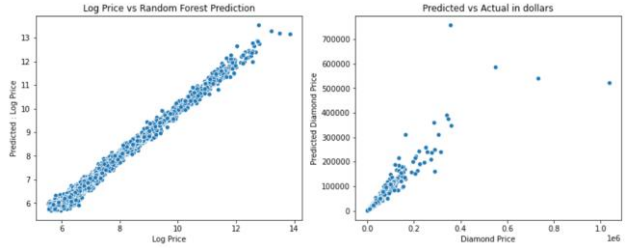



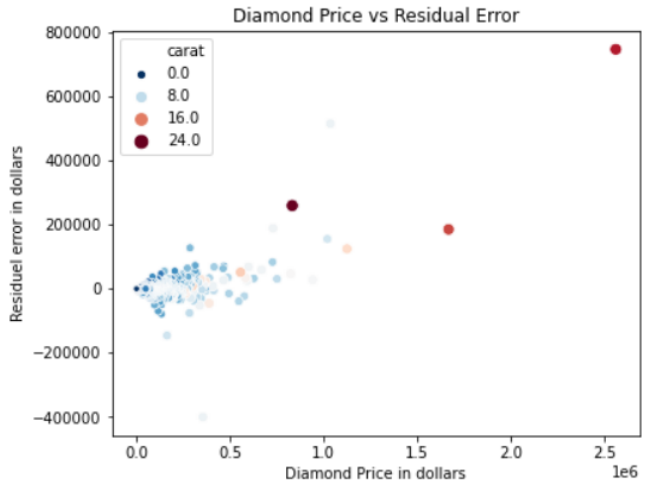



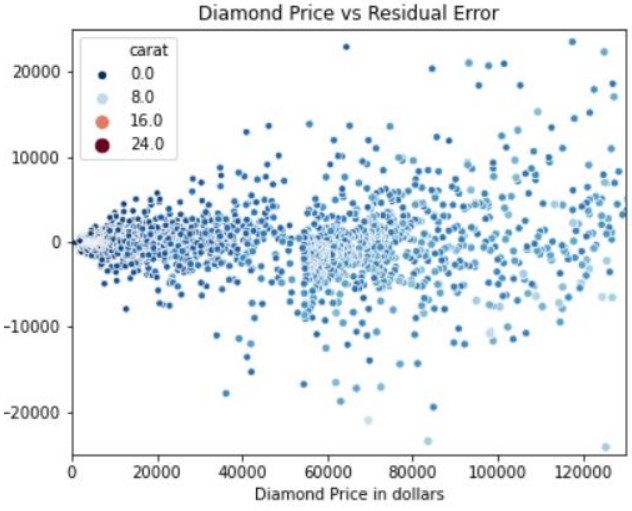



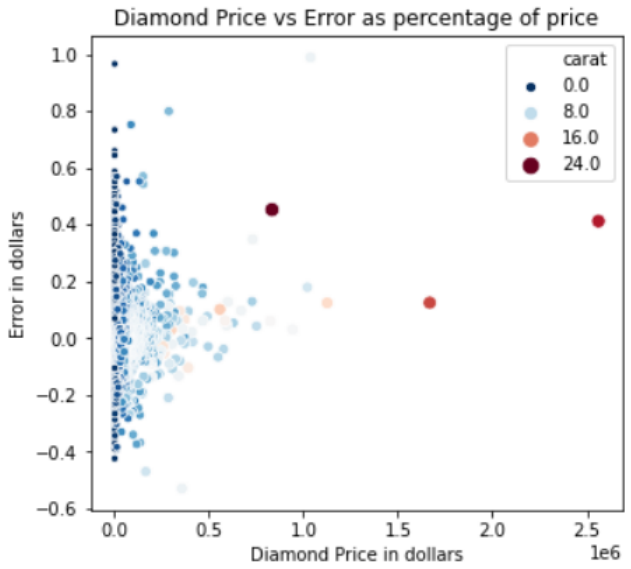




99% of the diamonds in the dataset are less than 132K

```
# price quantitles
diamonds['price'].quantile([0, 0.05, 0.25, 0.5, 0.75, 0.95,0.99,1])
```

```
0.00        250.0
0.05        342.0
0.25        492.0
0.50        805.0
0.75       2899.0
0.95      43795.0
0.99     131575.2
1.00    2561004.0
Name: price, dtype: float64
```







Top cheap diamonds

| | id | carat | price | Predicted | Error |
|---|---|---|---|---|---|
| 1033 | LD13151377 | 10.03 | 356542.0 | 757202.0 | -400660.0 |
| 1139 | LD11860282 | 8.43 | 164992.0 | 311070.0 | -146078.0 |
| 1199 | LD12468309 | 5.02 | 135673.0 | 214705.0 | -79032.0 |
| 1055 | LD13941280 | 8.01 | 284880.0 | 360491.0 | -75611.0 |
| 1277 | LD13702073 | 5.00 | 117285.0 | 187116.0 | -69831.0 |
| 1037 | LD13126106 | 10.08 | 338916.0 | 390762.0 | -51846.0 |
| 1369 | LD13130587 | 5.35 | 98477.0 | 149124.0 | -50647.0 |
| 1202 | LD10702992 | 5.01 | 134584.0 | 184430.0 | -49846.0 |
| 1028 | LD13705309 | 11.56 | 389526.0 | 434819.0 | -45293.0 |
| 1229 | LD13697954 | 5.01 | 126463.0 | 167731.0 | -41268.0 |

Top cheap diamonds % discount

| | id | carat | price | Predicted | Error_pct |
|---|---|---|---|---|---|
| 1033 | LD13151377 | 10.03 | 356542.0 | 757202.0 | -0.53 |
| 1139 | LD11860282 | 8.43 | 164992.0 | 311070.0 | -0.47 |
| 6865 | LD13884849 | 0.35 | 323.0 | 560.0 | -0.42 |
| 6541 | LD13885379 | 0.31 | 264.0 | 432.0 | -0.39 |
| 5437 | LD13870396 | 1.50 | 7654.0 | 12586.0 | -0.39 |
| 13391 | LD12463682 | 1.51 | 12754.0 | 20666.0 | -0.38 |
| 5131 | LD13585693 | 0.71 | 1697.0 | 2678.0 | -0.37 |
| 7260 | LD09851468 | 0.32 | 350.0 | 557.0 | -0.37 |
| 1277 | LD13702073 | 5.00 | 117285.0 | 187116.0 | -0.37 |
| 1199 | LD12468309 | 5.02 | 135673.0 | 214705.0 | -0.37 |

Find a cheap diamond in your budget (Example 5000-6000)

| | id | carat | price | Predicted | Error |
|---|---|---|---|---|---|
| 2421 | LD13974854 | 1.00 | 5414.0 | 7140.0 | -1726.0 |
| 21090 | LD13343734 | 1.29 | 5607.0 | 6656.0 | -1049.0 |
| 13709 | LD12734797 | 1.32 | 5651.0 | 6577.0 | -926.0 |
| 21086 | LD10063099 | 1.00 | 5452.0 | 6376.0 | -924.0 |
| 18247 | LD13962247 | 1.13 | 5213.0 | 6087.0 | -874.0 |
| 5875 | LD13798829 | 1.00 | 5387.0 | 6238.0 | -851.0 |
| 2691 | LD13974282 | 1.11 | 5446.0 | 6291.0 | -845.0 |
| 15272 | LD13515590 | 0.90 | 5246.0 | 6082.0 | -836.0 |
| 5177 | LD13803916 | 1.00 | 5713.0 | 6539.0 | -826.0 |
| 2004 | LD13976248 | 1.00 | 5217.0 | 6011.0 | -794.0 |