Classifier for determining good car buy

ABSTRACT

The goal of this project is to decide whether car is good buy or not. Dataset is obtained from the Kaggle - CARAVANA challenge. Dealers buy used cars from auction and some of them turn out to be costly because of transportation cost and repair work that needs to be put into it. Kicked Car is a term used to describe bad car deal. Kicked car happen because of unidentified issues in the car and fraudulent readings. A model to identify bad car at auction will save lot of money for dealers. Price of car depends on number of numerical and categorical attributes. The training data has car records classified as a good buy or a bad buy. Learning model is created using classification techniques and it is verified with test dataset. The model giving best performance is selected. This dataset was chosen since it covers different aspects of data mining like data cleaning, feature creation, feature subset selection and classification.

1. OVERVIEW

CARVANA maintains online car garage. All the details regarding car are available to view and compare [1]. Dataset provided on Kaggle is for competition for building classifier to predict if Car in auction is good buy or not [2]. Car details are described with number of attributes provided in table 1. New features will be created to describe engine capacity and engine type of car record by splitting the attributes Model and Sub-model. Numeric attributes like prices and costs are quantified for simpler analysis. Along with the provided attributes, data related to socioeconomic condition during the buy year is also important. Hence socioeconomic data can be obtained from financial site. Attributes which are highly correlated with the output class will be selected as a feature subset for creating learning model. Different classification algorithms: Random Forest, Decision tree, Naive Bayes and SVM will be considered for choosing the best model.

Table 1: Attribute and its types

Type of Attribute	Attributes
Nominal Attributes	IsBadBuy, Auction Name, Make, Model, Sub-model, Color, Transmission, Wheel-type, Nationality, Top Three American Name, PrimeUnit, IsOnlineSale, State of Purchase, ZIP of purchase state
Ratio Attributes	VehicleAge, Odometer Reading, MMRAcquisitionAuctionAveragePrice, MMRAcquisitionAuctionCleanPrice, MMRAcquisitionRetailAveragePrice, MMRAcquisitonRetailCleanPrice, MMRCurrentAuctionAveragePrice, MMRCurrentAuctionCleanPrice, MMRCurrentRetailAveragePrice, MMRCurrentRetailCleanPrice, AcquisitionCost, WarrantyCost
Interval Attributes	Vehicle Year, Purchase Date
Ordinal Attributes	Size, Wheel Type, Trim Level, Guarantee level

Section 2 describes design considerations and steps taken for preparing data. Section 3 describes implementation details and results for different data mining algorithms run on data. Section 4 compares performance of different data mining algorithms. Section 5 describes challenges faced during analysis and implementation.

2. DESIGN CONSIDERATIONS

2.1 Data Cleaning and Feature Extraction:

After visualizing data in Weka, it was found that some attributes carry redundant data. There are some attributes which are highly co-related to each other. E.g. age of car and purchase year convey the same information. Hence the year information is redundant. There is state name and ZIP code which are again highly correlated and provide overlapping information. Hence ZIP code information will be removed and only State name information will be retained with the data. Researchers at Willamette University [13] have found good results by performing principle component analysis of attributes.

There also exist composite attributes like Model and SubModel which contain descriptive information about car. SubModel attribute has 864 unique values whereas Model has 1063 unique values which makes it difficult to use in model. There are 3 types of information available in each Model and SubModel attribute. Part of Model text specifies Model name, engine type, engine power supply type and fuel capacity. E.g. EQUINOX AWD V6 3.4L Model describes EQUINOX mode, All wheel drive has V6 type engine and 3.4 liters of fuel capacity [9]. Part of SubModel text specifies the number of doors the car has, the sub model name and fuel capacity in liters. E.g. 4D SUV 4.2L SubModel describes SUV having 4 doors and 4.2 liters of fuel capacity.

Hence Model and SubModel attributes need text processing. Number of doors was easily extracted from SubModel text as it appears 1st in the SubModel description or is absent. Fuel capacity is extracted using regular expression for 'float value followed by L'. Model and SubModel name information needs de-duplication. One way of assigning definite Model or SubModel name to car is by selecting single word in description which has occurred the most over all the records. This requires splitting provided description, separating incorporated information from it and generating word count for rest of the data.

Nominal attributes described in table 2 like Auction Name, Make, Color, Transmission, Wheel-type, Nationality, Top Three American Name, State of Purchase are already categorical data and do not need any processing.

Table 2: Nominal Attribute Values Set

Attribute	Number of values
Auction	3
Make	33
StateOfPurchase	37
Color	17
Transmission	5
Nationality	5
TopThreeAmericanName	3

Ordinal attributes will be assigned numerical values based on its order. I also learnt from a Yonsei University blog [3] and Robin High's research [4] that levels in the ordinal data do not follow the natural order. Hence it is necessary to assign numerical values to ordinal data manually so that qualitative difference between the levels of the ordinal data are taken into account [5,6,7]. This will help learning model identify the behavior of the attribute. Such ordinal numerical values were assigned to Size variable for models which cannot handle categorical values.

All ratio attributes specified in the table were normalized and quantized. Quantizing the attributes helps in creating a simpler model. As per research done about feature discretization [3] it is found that most of the learning algorithms show better results for discrete rather than continuous data. I propose to quantize continuous attributes like cost, price estimations and odometer reading to nearest multiple of 100. After quantizing, numerical attributes will be normalized. Distance based data mining algorithms give better results after normalizing the data [8,16]. E.g. in the case of my Kaggle – CARVANA dataset Vehicle Odometer and Vehicle age both are numerical attributes however Vehicle Odometer gets more weightage as range of its magnitude is higher. Hence normalization is required for the data.

2.2 Data Analysis:

After exploring data in rattle tool, it is found that some price estimations, Prime Unit and Guarantee level attributes have NA values as per percentages shown in table 3 which need to be dealt with. Information gain or Chi square test can be performed to ascertain the dependence of the target variable on unknown values of these attributes. Then data imputation can be done using kNN method for attributes in which dependency exists between unknown values and target variable as per chi square test. This data cleaning task can re-mediate the missing values. The attributes PrimeUnit and Guarantee level have 67% of NA values however [13] have found very good accuracy by creating classifier with these attributes which are highly correlated with themselves. Whereas only 0.2% of price estimations are missing in the dataset of 73k instances, records having null values for price estimations can be discarded.

Table 3: Null values in Data

Attribute	Number of Null values	Percentage of null values
PrimeUnit	48699	67%
Guarantee level	48699	67%
MMRCurrentAuctionAveragePrice	206	0.2%
MMRCurrentAuctionCleanPrice	206	0.2%
MMRCurrentRetailAveragePrice	206	0.2%
MMRCurrentRetailCleanPrice	206	0.2%

For the sake of generating correlation matrix, I let R convert some of the Categorical attributes to numeric and plotted the data. Figure 1 shows. Vehicle age is negatively correlated with vehicle cost. Vehicle odometer is positively correlated with vehicle age and warranty cost.

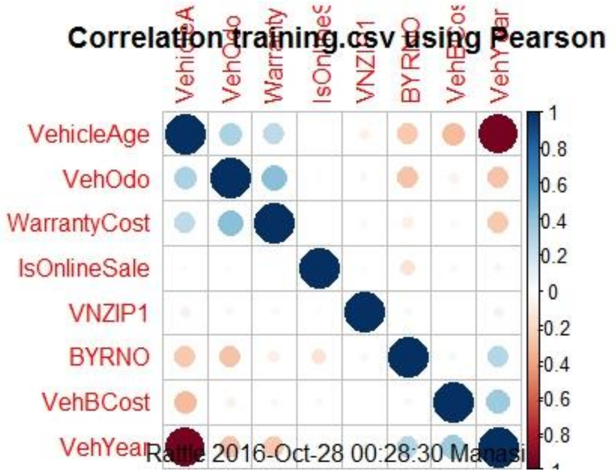


Figure 1: Correlation Plot

After plotting number of distribution of output class as seen in figure 2 it was found that only 12% of the instances classify the records as bad car, data might produce biased results. To overcome this skew, oversampling will be done to get about equal number of samples for good and bad car deals. Oversampling and Down-sampling modify the dataset in such a way that it ends up containing equal numbers of rows for all the classes of the target attribute. This will help in generating an effective and robust model.

The package which will be used to perform this task is SMOTE which is known as Synthetic Minority Oversampling Technique based on an article [15]. This works by taking the class which is the least in number or in other words the minority class and it synthetically creates samples based on nearby neighbors which are determined by attributes of the dataset. The number of the minority class members decides the number of randomly chosen nearest neighbors. For majority class, the suggested method randomly drops rows which are the nearest neighbor in feature space. Oswaldo Figueroa Domejean has done similar work and found good results [14].

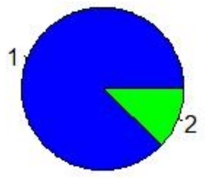


Figure 2: Output Variable Distribution

2.3 Handling large number of categories:

After cleaning the Model and SubModel attributes were derived as per table 4. Figure 3 also shows models which are seen more frequently.

Table 4: Derived attributes

Attribute	Unique Values
Model	517
SubModel	442
Wheel Drive	5
Door Type	4
Engine Capacity	Numeric
Engine Type	4

Some in-built functions for creating learning models like random forest, decision tree cannot handle more than 50 categorical values. Hence to make data compatible and simpler, discrete probabilities were calculated for each output class w.r.t. categorical values in these attributes [21]. This probability can be called as rate of specific attribute value relating to car bad buy or Gini. These bad buy rates w.r.t categorical values of attributes were mapped in the data using category it takes and actual categorical variables were removed. Attributes like Make, State, Color, WheelType, Nationality, TopThreeAmericanNames have less than 50 unique values which can be handled by R functions. However, some functions like k-means, knn require only numerical attributes to calculate Euclidean distance in which case such attributes were converted to rate variable or eliminated [20, 22].

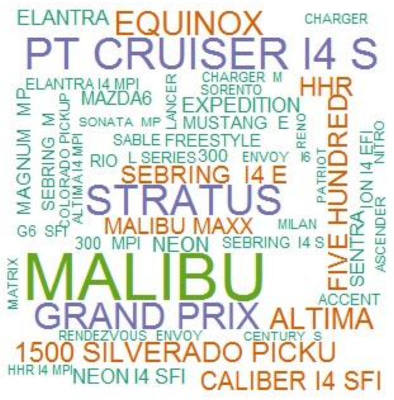


Figure 3: Word Cloud for Model attribute as per occurrence frequency

2.4 Dealing with NA values

Attributes like Prime Unit, Guarantee level had more than 60% of NA values and were eliminated. Of the derived attributes Engine Type, Engine Capacity and Wheel Drive had huge percentage of NA values hence were eliminated. Knn imputation was applied to attributes like Door Type which had less number of NA values. Knn Imputation is process of generating values for missing attributes based on k nearest neighbors. K was chosen as 5. Figure 4 shows presence of NA values in the data before processing. After processing for attributes showing good percentage of NA values, few more records were eliminated from the dataset where other attributes showed NA values.

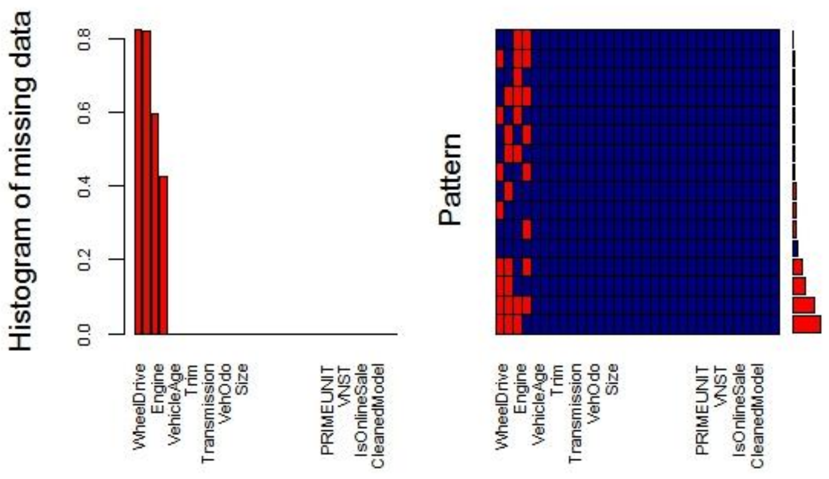
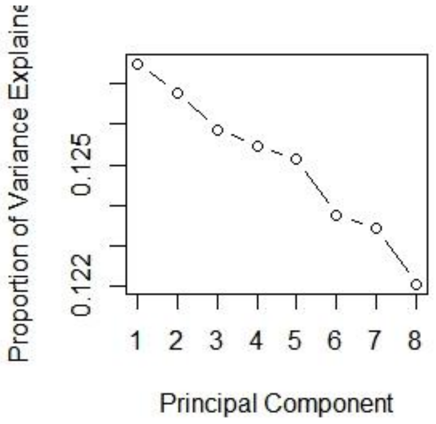


Figure 4: Information of Missing Data

2.4 Principal Component Analysis:

Initially after observing data in rattle() for numerical values, MMR price estimates showed very good cross-correlation as seen in figure 7. Hence after performing data cleaning, normalization and quantization, prcomp() function in R was applied to subset of data containing MMR price estimates for identifying principal components [24]. From figure 5 it can be seen that each principal component accounts for almost equal variance. From figure 6 which is a cumulative variance plot it was seen that in order to get about 93% of information, all 8 principal components are necessary. Hence all 8 MMR price estimate variables in the training dataset. It is also evident from the correlation plot in figure 8 obtained after normalization and quantization of these attributes where attributes showed zero cross-correlation.



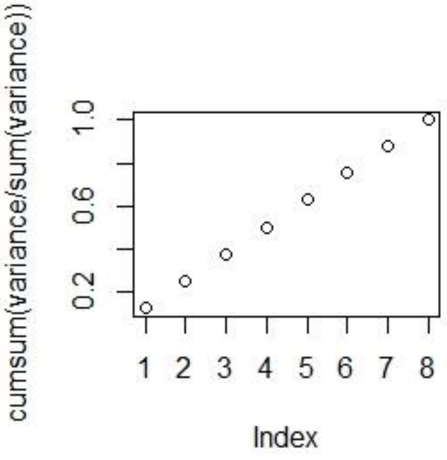


Figure 5: Proportion of Variance of Principal Components

Figure 6: Cumulative Variance of Principal Components

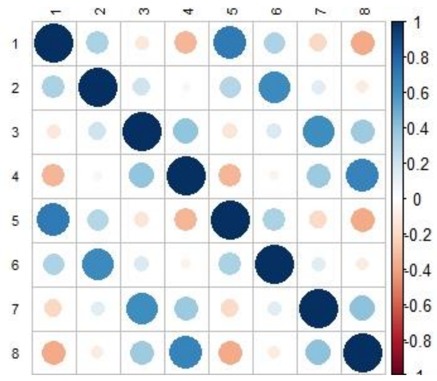


Figure 7: Before Normalization

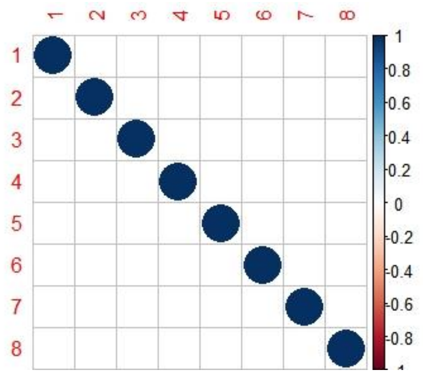


Figure 8: After Normalization

3. IMPLEMENTATION DETAILS

After cleaning data and extracting the features, dataset is used for following purposes.

3.1 Trends in the dataset:

By converting data to numerical, k-means clustering was performed on the dataset. kmeans() function in R was used for this purpose. However, since dataset consists of combination of numerical and categorical values, single distance measure could not be applied to all attributes. Converting categorical values to numeric does not add much value to information. Hence k-means clustering was used only for numerical values and extracting pattern out of those clusters was not useful.

3.2 Classification model for predicting car deal:

3.2.1 Random Forest

Random Forest method was applied to cleaned data. The classifier gives very good accuracy of 90%. However, it is failure since it cannot identify single record which relates to bad buy. Model chose IsBadBuy value of 0 to be positive class value. Car dealer would want to eliminate as much cars as possible which might be bad buys even if some good buys are missed. Hence it is important for model to have good negative prediction value.

Model gave low negative prediction value because dataset is skewed. Data contains only 12 % records that classify car as bad buy. Hence it is necessary to balance the dataset before creating model. Such imbalanced dataset can be balanced by synthetically creating records for class having lesser number of values. SMOTE package in R was used to create balanced dataset. Smote under-samples the class records which are higher in percentage and oversamples class records which are lower in percentage. After balancing data random forest model was fixed as seen in table 5.

Random forest model creates ensemble of decision trees and combines multiple weak classifiers to create one strong classifier. It uses multiple random subsets to create ensemble of decision trees [19]. Random subsets are created by selecting random predictor variables also called as bagging. Figure 9 is visualization of confusion matrix and ROC obtained after applying RandomForest model again to the balanced dataset [17].

Table 5: Random forest fixed model details

Accuracy	0.7741
Sensitivity	0.7557
Specificity	0.8130
Pos Pred Value (Car is good buy)	0.8953
Neg Pred Value (Car is bad buy)	**0.6113**

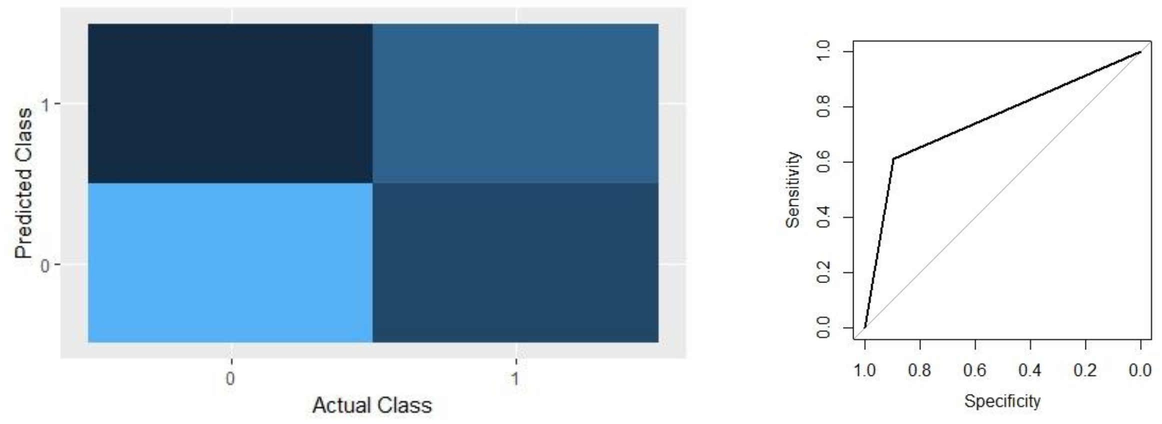


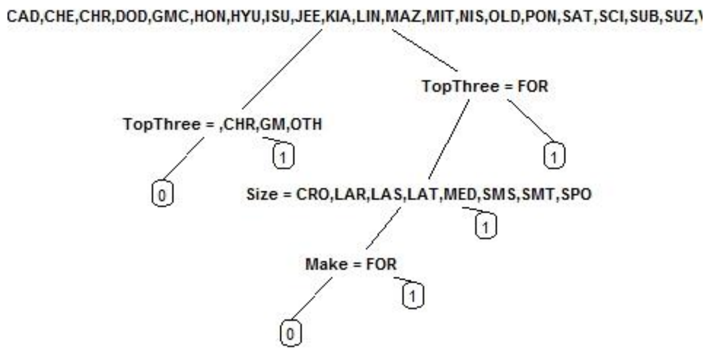
Figure 9: Confusion Matrix and ROC for Random Forest Prediction

3.2.2 Decision Trees

Decision tree model is created by choosing decision nodes recursively based on Information gain of the different attributes. Entropy of the records is used as stopping criteria. Categorical attributes will be split based on categories whereas numerical attributes will be split on best threshold given by information gain. Decision tree was created using rpart() method in R. Figure 10 shows decision tree created with rpart(). Decision tree model chose Make as its 1st node, Top Three American names as 2nd node, Size as 3rd node and Make as 4th node. Figure 11 is confusion matrix and ROC for Decision tree model [23].

Table 6: Decision Tree Model statistics

Accuracy	0.7442
Sensitivity	0.7103
Specificity	0.8483
Pos Pred Value (Car is good buy)	0.9350
Neg Pred Value (Car is bad buy)	**0.4878**



CAD,CHE,CHR,DOD,GMC,HON,HYU,ISU,JEE,KIA,LIN,MAZ,MIT,NIS,OLD,PON,SAT,SCI,SUB,SUZ,
TopThree = FOR
TopThree = ,CHR,GM,OTH
1
1
0
Size = CRO,LAR,LAS,LAT,MED,SMS,SMT,SPO
1
Make = FOR
1
0

Figure 10: Decision Tree Model

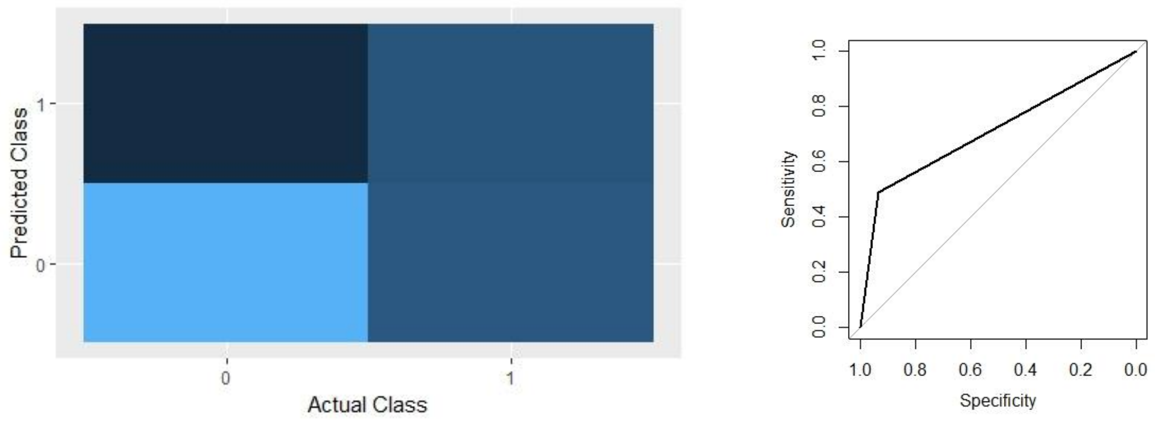


Figure 11: Confusion Matrix and ROC for Decision Tree Prediction

3.2.3 Naïve Bayes

Naïve Bayes classification model is based on Bayes Theorem and takes probabilistic approach towards the data. Attribute showing lowest error rate is chosen for classification model by Bayes rule [10]. Naïve Bayes is mostly used in multi class problems and since there are number of multi- class categorical attributes in the data this classifier should be tried out. Naïve Bayes classifier was created using the naiveBayes() function in R. Figure 12 shows confusion matrix and ROC for Naïve Bayes classifier.

Table 7: Naive Bayes Model Statistics

Accuracy	0.7184
Sensitivity	0.7561
Specificity	0.6686
Pos Pred Value (Car is good buy)	0.7511
Neg Pred Value (Car is bad buy)	**0.6745**

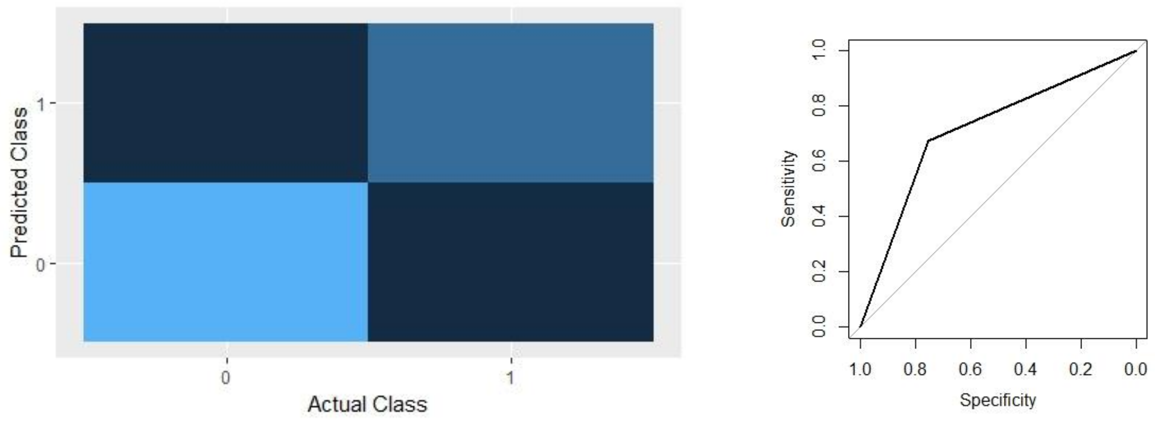


Figure 12: Confusion Matrix and ROC for Naive Bayes Classifier

3.2.4 SVM

SVM is considered best classifier for two-class problem. Hence SVM classifier can be created for classifying if the car deal is good or bad. Support Vector Machine algorithm finds optimal separating hyperplane by learning the training data [11]. Researchers at Stanford have also found good results with SVM classifier [12]. There is available package e1071:svm() for Support Vector Machine prediction. Figure 13 shows confusion matrix and ROC for SVM classifier.

Table 8: SVM Model Statistics

Accuracy	0.7325
Sensitivity	0.7378
Specificity	0.7229
Pos Pred Value (Car is good buy)	0.8273
Neg Pred Value (Car is bad buy)	**0.6051**

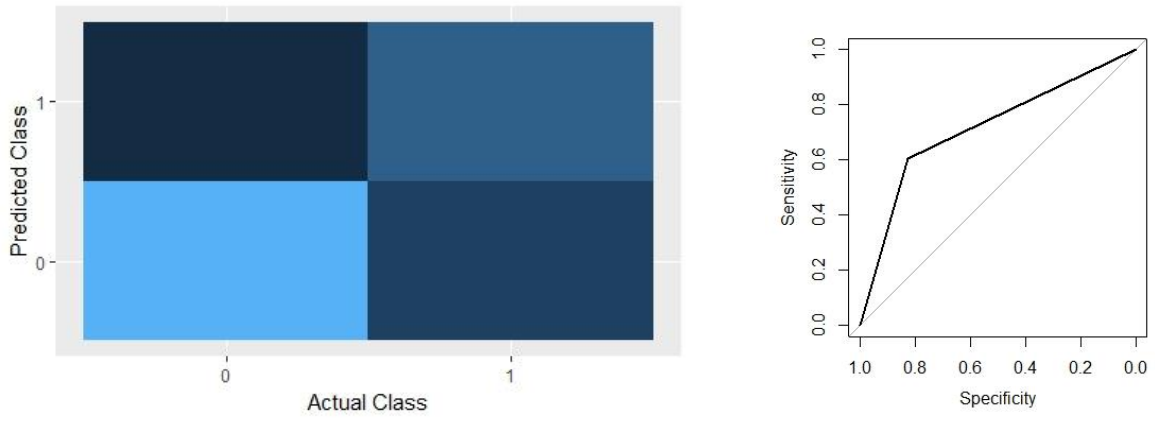


Figure 13: Confusion Matrix and ROC for SVM Classifier

3.2.5 knn with N-fold cross validation

Since Negative Prediction Value is measure of performance for classifier, N fold cross validation was done. Value of N is taken as 10. Training dataset was divided into 10 non-overlapping sets. N fold cross validation was done by running kNN for 1 to 10 values of k. The highest value for Negative Prediction was found for k=2 which is 0.43. Hence kNN model showed the least performance as compared to other models. Which again shows that some of the categorical variables which were removed for dataset to be compatible with knn() method in R, are good predictor variables [20].

3.2.6 Logistic Regression

Logistic regression is process of creating linear classifier. Logistic regression can be applied to categorical target variable. Logistic regression model can be implemented using gradient descent [18]. Model can predict the probability of each output class based on independent variables and output class having higher probability is selected. Logistic regression and SVM model have shown very similar statistics as seen from table 8 and 9.

Table 9: Logistic Regression Model Statistics

Accuracy	0.7296
Sensitivity	0.7360
Specificity	0.7182
Pos Pred Value (Car is good buy)	0.8238
Neg Pred Value (Car is bad buy)	**0.6031**

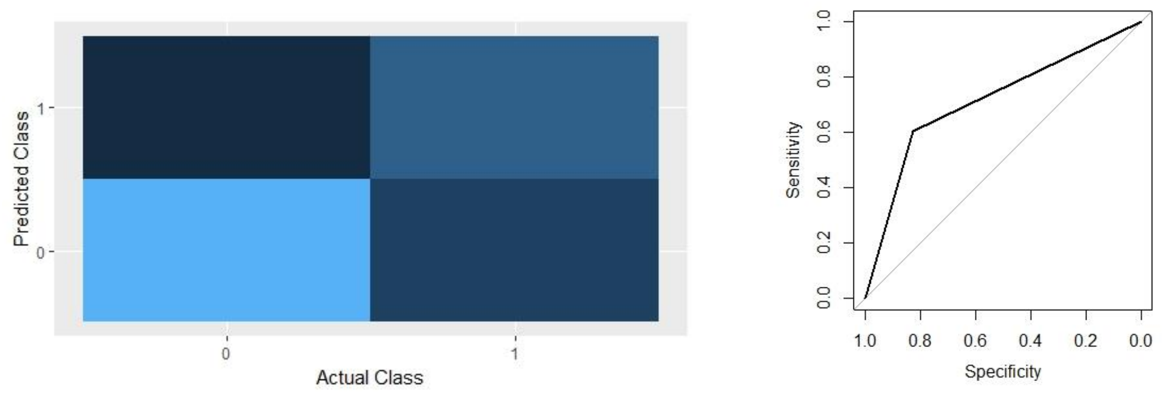


Figure 14: Confusion matrix and ROC for Logistic regression

4. PERFORMANCE MEASURE

Since the goal of project is a classification task, performance of model can be calculated by creating confusion matrix and drawing ROC curve for test data. ROC curves are not prone to error in case of skewed data as well hence it proves to be a good measure for Kaggle – CARVANA dataset. However, car dealer would rely more on the model that identifies bad cars readily at expense of declaring some good cars as bad cars since there are always plenty of cars to choose from when car resale comes in picture. Hence while comparing performance of model, negative prediction value is considered as measure. As seen from confusion matrix and classifier statistics, Random forest model gives best accuracy however its negative prediction value is less. Decision tree gives the best positive prediction value and specificity however its Negative Prediction value is the worst. Similar to Random Forest, SVM and logistic regression give good accuracy however its negative prediction value is less. It's Naïve Bayes classifier that give the best Negative Prediction Value and hence recommended. From ROC curve it is observed that Random forest has maximum area under the curve however when classifier giving the best class separation is considered, Naïve Bayes model is again comparable.

5. CHALLENGES

Since there are 73,014 instances of data, it is possible to handle it in R. Dealing with large NA values and text processing could have been faster with Spark and MLlib. The major challenge was finding the best features out of available features. Different combinations of independent variables were tried for RandomForest algorithm and one giving best performance was chosen for all the models. Chosen attribute set excluded Engine type and State variable from it.

6. REFERENCES

[1] Carvana, 09/25/2016, https://www.carvana.com/how-it-works.
[2] Kaggle competition, 09/25/2016, https://www.kaggle.com/c/DontGetKicked.
[3] Yonsei University, Andrew Teoh, 10/20/2016, https://sites.google.com/site/multimediasecuritylab/feature-discretization
[4] Models for Ordinal Response Data, Robin High, 10/26/2016, http://support.sas.com/resources/papers/proceedings13/445-2013.pdf
[5] What to do with Ordinal Data, Dr Nic, 10/26/2016, https://learnandteachstatistics.wordpress.com/2013/07/08/ordinal/
[6] Basic Data Types, Michael Castello, 10/26/2016, https://infoactive.co/data-design/ch01.html
[7] Learning to Classify Ordinal Data: The Data Replication Method, Jaime S. Cardoso, Joaquim F. Pinto da Costa, 10/26/2016, http://www.jmlr.org/papers/volume8/cardoso07a/cardoso07a.pdf
[8] K Nearest Neighbors – Classification, Saed Sayed, 10/27/2016, http://www.saedsayad.com/k_nearest_neighbors.htm
[9] Car types, 10/27/2016, https://www.quora.com/How-do-you-know-if-a-car-is-4WD-2WD-FWD-RWD-or-AWD
[10] Naive-Bayes Classification Algorithm, Dr. Ing. Cristian Mihaescu, 10/27/2016, http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Lab4-NaiveBayes.pdf

[11] SVM Tutorial, Alexandre Kowalczyk, 10/28/2016, http://www.svm-tutorial.com/2014/11/svm-understanding-math-part-1/
[12] Don’t Get Kicked - Machine Learning Predictions for Car Buying, Albert Ho, Robert Romano, Xin Alice Wu, 10/28/2016, http://cs229.stanford.edu/proj2012/HoRomanoWu-KickedCarPrediction.pdf
[13] Don’t be Kicked, 10/28/2016, https://gsm672.wikispaces.com/Final+Report+-+Don%27t+Get+Kicked!
[14] Data Science with Kaggle´s Competition "Don´t Get kicked!", Oswaldo Figueroa Domejean, 10/27/2016, https://www.researchgate.net/publication/262523736_Data_Science_with_Kaggles_Competition_Dont_Get_kicked
[15] Jason Brownlee. 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset, 10/24/2016, http://machinelearningmastery.com/tactics-to-combat-imbalanced/classes-in-your-machine-learning-dataset/
[16] The Use of Normalization, 10/26/2016, http://aimotion.blogspot.com/2009/09/data-mining-in-practice.html
[17] CRAN-R Project, Xavier Robin, 11/26/2016, https://cran.r-project.org/web/packages/pROC/pROC.pdf
[18] Kan Deng’s Corner, Kan Deng, Dr, 11/21/2016, https://www.cs.cmu.edu/~kdeng/thesis/logistic.pdf
[19] Blog & Press, Dan Benyamin, 11/27/2016, https://citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics/
[20] CRAN-R Project, Brian Ripley, 11/27/2016, https://cran.r-project.org/web/packages/class/class.pdf
[21] R Documentation, 11/26/2016, https://stat.ethz.ch/R-manual/R-devel/library/stats/html/ftable.html
[22] R Documentation, 11/25/2016, https://stat.ethz.ch/R-manual/R-devel/library/stats/html/kmeans.html
[23] CRAN-R Project, Stephen Milborrow, 11/26/2016, https://cran.r-project.org/web/packages/rpart.plot/rpart.plot.pdf
[24] Analytics Vidhya, Analytics Vidhya Content Team, 11/26/2016, https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/