2022

“Improving the used cars market through machine learning techniques”.

FINAL DISSERTATION PROJECT

WALLACE TUDEME - 202123922

UNIVERSITY OF HULL | Cottingham Rd, Hull HU6 7RX

Table of Contents

List of Equations

List of Figures

List of Tables

Definition of terms

- MAE - Mean Absolute Error
- MSE - Mean Squared Error
- ADS-B - Automatic Dependent Surveillance - Broadcast
- LSTM - Long Short Term Memory
- EDA - Exploratory Data Analysis
- SVM - Support Vector Machines
- RMSE - Root Mean Square Error
- rRMSE - relative Root Mean Square Error
- Cross-val - Cross-Validation
- XGBoost - Extreme Gradient Boosting

Abstract

This project is about improving the used cars market through the use of machine learning techniques and the research is centered around finding the best machine learning models that best predicts the price of a used car and if it would be sold or not. It also involves discovering the features that best influences the price of a used car and sale through the use of feature engineering techniques. After the research, XGBoost regressor proved to be the best model for price prediction with an R^2 score of 99.01% for training data and 97.93% for test data. It also had an rRMSE score of 11.51%. The linear regressor model was the worst performing model with an R^2 score of 63.46% for training data and 63.14% for test data. It also had an rRMSE score of 57.20%. The best price predicting features were max power, year, km driven and torque in newton meter. The best model for predicting sold or not-sold binary classification task was the stacking classifier with an accuracy score of 98.81% for training data and 96.36% for the test data with only 86 mislabeled observations out of 2363 observations. The features for this prediction were chosen by the Random Forest Classifier during hyper parameter optimisation. The worst model for this binary classification task was the Gaussian Naïve Bayes (GNB) with an accuracy score of 63.51% for the training data and 63.06% for the test data. It had 873 mislabeled observations out of the 2363 observations available. The features for this model training were selected using K-Best Selector.

1.0.0 Introduction

Vehicle lifetime has had significant increase over the years, rising nearly 27% from 1969 to 2014(Bento et al., 2018). The global market value of used cars was about 819.52 billion dollars in 2003 and with a compound annual growth rate of 4.2% between 1999 and 2003 (Duvan & Ozturkcan, 2009).

A used car in general term is said to be any car that has previously been registered. The used cars market covers the sale of private and remarketed second hand cars. Private sale refers to the sale when both the buyer and the seller are private individuals and remarketed sales refers to sales by companies which can be car manufacturers, car leasing and rental companies (Duvan & Ozturkcan, 2009).

The used cars market is described as a lemons market because of the asymmetric information between dealers and consumers. This has been of great benefit to dealers with more knowledge as they rack up huge profits from sale of low-quality products as there are many factors that influence the prices of used cars but this information is not available to the consumer and sometimes the new dealers in the market. But with the presence of the internet and available individual reviews a balance is gradually coming into the market. This is one area my project will help to address for the new dealers with little knowledge about market facts (Duvan & Ozturkcan, 2009).

Further research has shown that the mileage and reliability (cost of maintenance) of a used car and the future fuel cost affects the pricing of used cars while biasing the mind of consumers whether or not to opt for a used car (Sallee et al., 2016; Yerger, 1996).

(Akerlof, 1978) did research to see how depreciation affects the prices of used cars. He proposed that there is a possibility that almost new cars could exceptionally depreciate as a result of undetectable risks and such cars being termed lemon cars by Akerlof. However, such risks could be considered to be entirely hypothetical as such cars often comes with factory insurance. Though due to the asymmetrical information that exist between dealers and consumers, such cars when returned, often find their way back into the market to an unsuspecting buyer at discounted rates and having the dealer profit from the turn of events.

(Scherer, 1996) quoted an internal memorandum of a ford Galaxie four door sedan. The memo revealed that the models with extra features exceeded the base model cost by an accounting cost of 17%, that happens to be in contrast with the amount change for extra feature models which could go as high as 293% markup. Also (Phlips, 1983) concluded that cars with extra options are overpriced to extract the highest possible price from those who want extra features like higher engine power output.

1.1.0 Aims and Objectives

i. Finding the best features that best predicts the change in target data and building machine learning models to predict the price of a car and probability of sale.
ii. Comparing the performance of the machine learning models and detecting possible limitations that might result from individual models.

1.2.0 Problem statement

This project aims to answer questions from two main focus areas namely;

i. Business related questions
ii. Technical related questions

1.2.1 Business related questions

i. What are the features that influence car prices the most?
ii. What are the features that influence car sales the most?
iii. What possible facts are hidden in the data that would influence sales positively?

1.2.2 Technical related questions

i. What model would best predict car prices and with what features?
ii. Which model best predict the sale of a car and with what features?
iii. Are there any other model performance related observations?

1.3.0 Scope of work

This project will start with acquiring the required data, data cleaning and data mining. Thereafter feature extraction/engineering, machine learning data processing, training and testing of models, exploring performance metrics of the models, hyperparameter optimisation, final results and conclusion.

1.4.0 Project outline

The outline for this project is as follows;

i. Introduction
ii. Background study
iii. Methodology
iv. Result and discussion
v. Conclusion

2.0.0 Background study

In recent time a lot of research has been conducted for the prediction of used car prices.

(Asghar et al., 2021) on their project titled used cars price prediction using machine learning with optimal features, were able to establish facts using machine learning techniques on the available data to effectively predict prices using regressor models. They sourced their data from Kaggle.com and used random forest and decision tree regressors and achieved a training accuracy of 95.82% and a test accuracy of 83.63% while using 500 decision trees and 205 observations.

(Voß & Lessmann, 2017) on their project titled resale price prediction in the used cars market sourced their data from a leading German car manufacturer and were hoping to answer questions centered around the degree to which resale prices are predictable, what is the relative accuracy between price predictive models and how do they compare with each other. They concluded at the end of their findings that ensemble machine learning techniques are the best for price prediction models proving earlier research by (Caruana et al., 2006) right. They also concluded that linear regression methods predict significantly less accurate than ensemble regression methods. At the end of their project, they were able to achieve a MAE of 3.97.

(N. Monburinon et al., 2018) on their project titled prediction of prices for used car by using regression models, sourced their data by scraping it off eBay-Kleinanzeigen, a German e-commerce and consist of 371,528 observations. Their project was centered around a performance analysis of multiple linear regression, random forest regression and gradient boost regression. After building the various models,

they got MSE of 0.28 for the gradient boost, 0.35 for random forest and 0.55 for multi linear. By these results, they concluded that the gradient boost is best for regression tasks.

(G. Gui et al., 2020) on their project titled flight delay prediction based on aviation big data and machine learning, sourced their data from an automatic dependent surveillance broadcast (ADS-B) and retrieving important features such as weather condition, flight schedule and airport information. They experimented with the LSTM and random forest model and they concluded that though the LSTM is suitable to handle classification problems, it suffered from overfitting because of their limited dataset. However, the random forest proved very valuable as it obtained a high accuracy score of 90.2%.

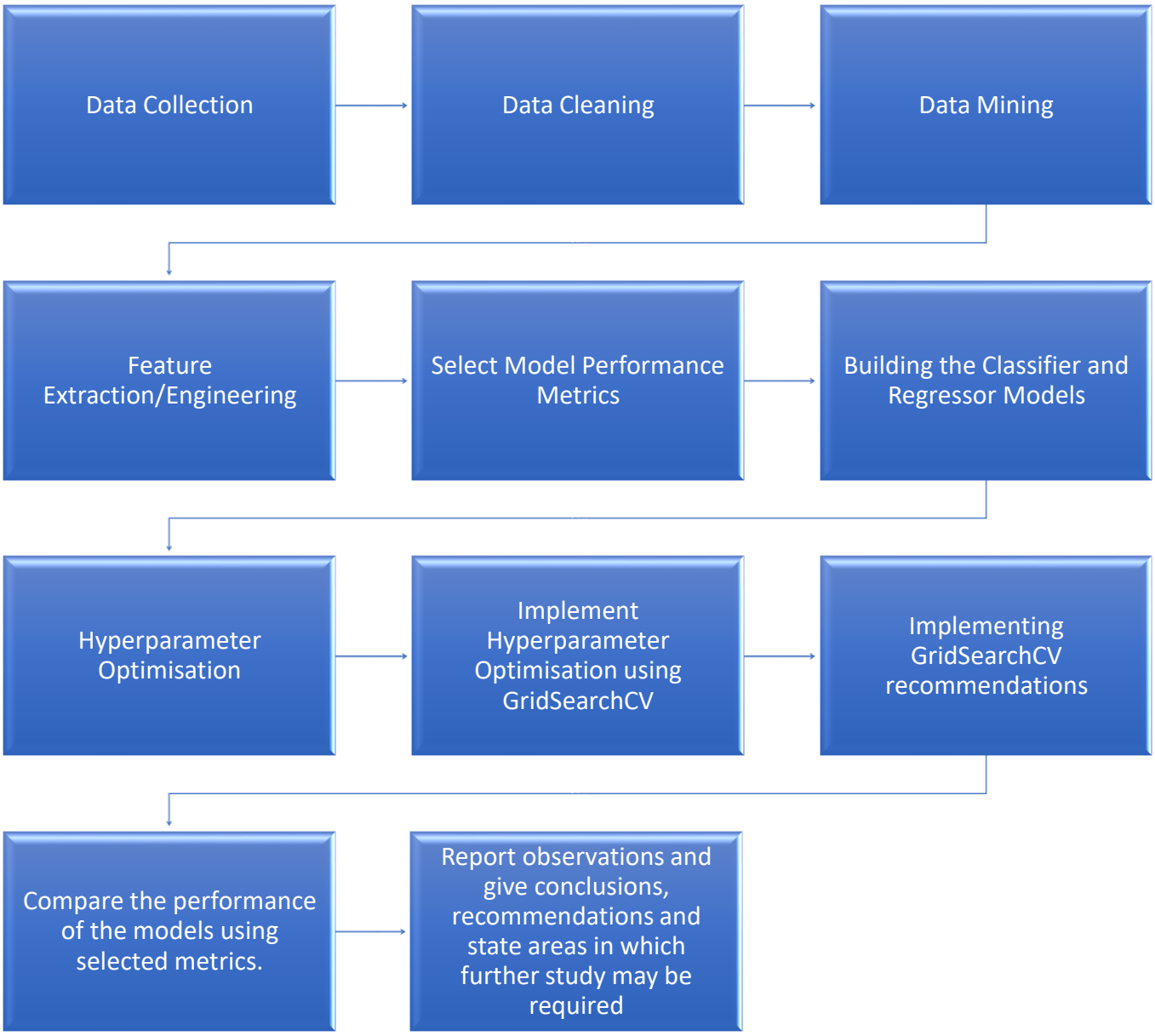
(V. Bahel et al., 2020) on their research titled a comparative study on various binary classification algorithms and their Improved variant for optimal performance, sourced their data from public domain which were breast cancer prediction and titanic survival datasets. They were able to compare the performance of various binary classification algorithms used preferred metrics. Their correlation result showed that the breast cancer dataset had better correlation than the titanic dataset while for the various models, the Logistic Regression and AdaBoost models performed better with features with high correlation to the target. They also concluded that the Naïve Bayes model worked better with the dataset with less correlation because it treats each feature as an independent class thus simplifying the model's learning procedure. The KNN worked well when used with fewer features but as the number of features increases the performance of the model while Decision Tree and Random Forest performed decently with both datasets.

For this research, I will be using exploratory data analysis (EDA) packages like pandas, numpy, matplotlib and seaborn to analyse the available data and provide insights. I will also be using machine learning regression algorithms to build price prediction models and classification algorithms to build binary classification models as I discovered that researchers are yet to explore the option of trying out up to seven different models on a machine learning task to compare the performance of the various algorithms using preferred metrics so as to decide the best performing algorithm in each group.

This project is aimed at addressing issues relating to the price disparity problem in car prices across the used cars market using the United States (US) market as a baseline.

3.0.0 Methodology

3.1.0 Flowchart representation of methodology



3.2.0 Data collection

The data for this research was retrived from Kaggle.com using the link below: (https://www.kaggle.com/datasets/shubham1kumar/usedcar-data)

The data contained 7906 observations and 18 columns as are listed below;

i. Sales_ID
ii. Name

iii. Year
iv. Selling price
v. Km driven
vi. Region
vii. State or province
viii. City
ix. Fuel
x. Seller type
xi. Transmission
xii. Owner
xiii. Mileage
xiv. Engine
xv. Max power
xvi. Torque
xvii. Seats
xviii. Sold

3.3.0 Data cleaning

Upon retrieving the data, I imported the data into the Pandas library and after initial exploration, the data had no null values but for the torque column that required cleaning which contained unwanted characters and was not in the right format as can be seen below in figure 1 I had to split the data into column into groups and then removed unwanted characters using both excel and Pandas library commands. Thereafter, I had to place the place the new cleaned data into individual columns as shown in figure 2 and 3. Finally, I converted the n/m torque column into a float datatype and the rpm torque into an int datatype. The torque unclean data was a combination of two column which is torque in n/m and torque in rpm.

```
'182.5Nm@ 1500-1800rpm' '90.3Nm@ 4200rpm' '12.5@ 2,500@kgm@ rpm@'
'215Nm@ 1750-3000rpm' '215Nm@ 1750-3000' '305Nm@ 2000rpm'
'540Nm@ 2000rpm' '327Nm@ 2600rpm' '300Nm@ 1600-3000rpm'
'620Nm@ 2000-2500rpm' '450Nm@ 1600-2400rpm' '19@ 1,800@kgm@ rpm@'
'9.2@ 4,200@kgm@ rpm@' '145@ 4,100@kgm@ rpm@' '51Nm@ 4000+/-500rpm'
'110Nm@ 3000rpm' '148Nm@ 3500rpm' '116Nm@ 4750rpm'
'48@ 3,000+/-500@NM@ rpm@' '148Nm@ 4000rpm' '222Nm@ 4300rpm'
'135.3Nm@ 5000rpm' '98Nm@ 1600-3000rpm' '170Nm@ 1400-4500rpm'
'343Nm@ 1400-2800rpm' '402Nm@ 1600-3000rpm' '113Nm@ 3300rpm'
'99.07Nm@ 4500rpm' '210nm@ 1600-2200rpm' '190 Nm @ 1750  rpm '
'32.1kgm@ 2000rpm' '224nm@ 1500-2750rpm' '400nm@ 1750-2500rpm'
'215Nm@ 1750-2500rpm' '25@ 1,800-2,800@kgm@ rpm@' '197Nm@ 1750rpm'
'136.3Nm@ 4200rpm' '470Nm@ 1750-2500rpm' '11@ 3,000@kgm@ rpm@'
'142Nm@ 4000rpm' '145Nm@ 4100rpm' '320Nm@ 1500-2800rpm'
'123Nm@ 1000-2500rpm' '218Nm@ 1400-2600rpm' '510@ 1600-2400'
'220Nm@ 1500-2750rpm' '380Nm@ 2000rpm' '104Nm@ 3100rpm' '292Nm@ 2000rpm'
'20@ 3,750@kgm@ rpm@' '46.5@ 1,400-2,800@kgm@ rpm@' '380Nm@ 2500rpm'
'15@ 3,800@kgm@ rpm@' '136Nm@ 4250rpm' '228Nm@ 4400rpm' '149Nm@ 4500rpm'
'187Nm@ 2500rpm' '146Nm@ 3400rpm' '8.6@ 3,500@kgm@ rpm@'
'219.7Nm@ 1750-2750rpm' '190Nm@ 2000-3000' '450Nm@ 2000rpm'
'300Nm@ 2000rpm' '230Nm@ 1800-2000rpm' '42@ 2,000@kgm@ rpm@'
'110Nm@ 3000-4300rpm' '110@11.2@@ 4800' '330Nm@ 1800rpm'
'225Nm@ 1500-2500rpm' '380Nm@ 1750-2750rpm' '28.3@ 1,700-2,200@kgm@ rpm@'
'259.88Nm@ 1900-2750rpm' '580Nm@ 1400-3250rpm' '400 Nm /2000 rpm'
'127Nm@ 3500rpm' '300Nm@ 1500-2500rpm' '132.3Nm@ 4000rpm'
'113nm@ 4400rpm' '151NM@ 4850rpm' '153Nm@ 3750-3800rpm'
'10.7@ 2,500@kgm@ rpm@' '124.6Nm@ 3500rpm' '78Nm@ 3500rpm'
'219.9Nm@ 1750-2750rpm' '420.7Nm@ 1800-2500rpm' '130Nm@ 3000rpm'
'424Nm@ 2000rpm' '130@ 2500@kgm@ rpm@' '99.8Nm@ 2700rpm'
'113Nm@ 4,500rpm' '11.2@ 4,400@kgm@ rpm@' '240Nm@ 1850rpm'
'16.1@ 4,200@kgm@ rpm@' '320Nm@ 1750-2700rpm' '115Nm@ 4500rpm'
```

Figure 1: Unclean torque column before cleaning.

```
array([190.0, 250.0, '140', 113.75, 59.0, 170.0, 160.0, 248.0, 78.0, 84.0,
       115.0, 200.0, 62.0, 219.7, 114.0, 69.0, 172.5, 114.7, 60.0, 90.0,
       151.0, 104.0, 320.0, 145.0, 146.0, 343.0, 400.0, 138.0, 360.0,
       380.0, 173.0, 111.7, 219.6, 112.0, 130.0, 205.0, 280.0, 99.04,
       77.0, 110.0, 153.0, 113.7, 113.0, 101.0, 290.0, 120.0, 96.0, 135.0,
       259.8, 259.9, 91.0, 96.1, 109.0, 202.0, 430.0, 347.0, 382.0, 620.0,
       500.0, 550.0, 490.0, 177.5, 300.0, 260.0, 213.0, 224.0, 640.0,
       95.0, 71.0, 117.0, 72.0, 140.0, 134.0, 150.0, 340.0, 240.0, 330.0,
       111.8, 135.4, 190.25, 247.0, 223.0, 180.0, 195.0, 154.9, 114.73,
       108.0, 190.24, 420.0, 100.0, 51.0, 132.0, 350.0, 218.0, 85.0, 74.5,
       180.4, 230.0, 219.66, 245.0, 204.0, 125.0, 172.0, 102.0, 106.5,
       108.5, 144.15, 99.0, 142.5, 196.0, 209.0, 220.0, 171.0, 277.5,
       215.0, 263.7, 94.14, 789.0, 259.87, 436.39, 182.5, 90.3, 305.0,
       540.0, 327.0, 450.0, 148.0, 116.0, 222.0, 135.3, 98.0, 402.0,
       99.07, 210.0, 197.0, 136.3, 470.0, 142.0, 123.0, 510.0, 292.0,
       136.0, 228.0, 149.0, 187.0, 225.0, 259.88, 580.0, 127.0, 132.3,
       124.6, 219.9, 420.7, 424.0, 99.8, 321.0, 619.0, 560.0, 600.0,
       285.0, 226.0, 155.0, 103.0, 175.0, 72.9, 57.0, 128.0, 131.0, 185.0,
       176.0, 121.0, 106.0, 113.8, 83.0, 124.5, 171.6, 88.4, 355.0, 119.0,
       410.0, 174.0, 99.1, 385.0, 53.0, 124.0, 159.8, 333.0, 480.0,
       250.06, 436.4], dtype=object)
```

Figure 2:Torque column containing unique nm values after cleaning.

```
array([ 2000,  2700,  2250,  4500,  4000,  2500,  2100,  3500,  3550,
        1750,  3000,  2125,  4850,  2200,  4600,  4800,  2400,  2625,
        4400,  2300,  2375,  2975,  3750,  3800,  4200,  4250,  2275,
        2325,  1900,  4300,  3125,  1700,  2600,  2212,  1600,  2750,
        4700,  2875,  1300,  1740,  3650,  3200,  4386,  2525,  1470,
        1800,  3275,  5000,  1950,  3600,  1820,  4388,  2150,  1650,
        4100,  4750,  2950,  3300,  3100,  3400,  3775,  1850,  2225,
        1500,  1875,  2650,  2800,  3325,  1462,  3175, 21800,  2050,
        2340,  3700])
```

Figure 3:Torque column containing unique rpm values after cleaning.

3.4.0 Data mining

After cleaning the data, data mining was done to get insights from the data including the use of Apriori to get instances with high correlation and connected events. I used available commands from pandas and numpy for the data mining and analysis and used matplotlib and seaborn commands for data visualisation.

3.5.0 Feature extraction/engineering

After data mining I carried out feature extraction to get the best features for training my models. I used the k-best algorithm to get best features to predict the selling price and sold columns. I also did a correlation plot to reveal correlation relationship between the columns with annotations.

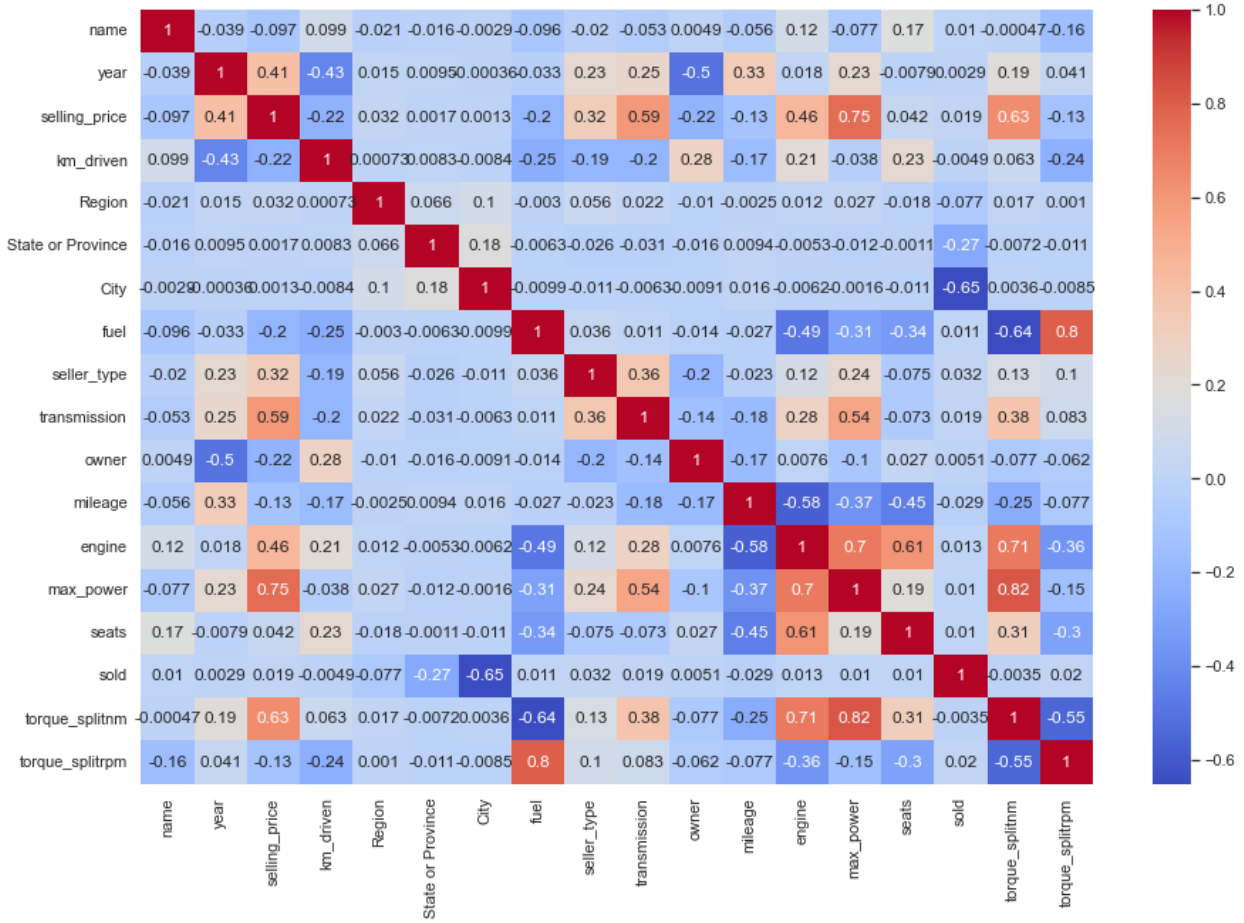


Figure 4:Picture showing correlation plot between columns with annotations.

3.6.0 Performance metrics selection for regression and classification

3.6.1 Performance metrics selection for regression task

i. ***R^2 Score:*** is also known as coefficient of determination, explains the variation expected in an output (y) which depends on input (x) while using any regression model. It is also important to note that the higher values R^2 score depicts better prediction by the model (Stojiljkovic, 2021). Mathematically, R^2 is denoted by;

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x}_i)^2} \quad \text{..... Equation i}$$

ii. ***Mean Absolute error:*** is expressed as the sum of deviation between the actual and the predicted value in ratio with number of observations as shown below;

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}_i) \quad \text{..... Equation ii}$$

The closer this value is to zero the better the model.

iii. ***Mean Squared Error:*** is expressed as the sum of the square of deviation between the actual and the predicted value in ratio with number of observations as shown below;

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}_i)^2 \quad \text{..... Equation iii}$$

The closer this value is to zero the better the model.

iv. ***Root Mean Square Error:*** is expressed as the square root of the sum of the square of deviation between the actual and the predicted value in ratio with number of observations as shown below;

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}_i)^2} \quad \text{..... Equation iv}$$

v. **Relative Root Mean Square:** is expressed as the ratio of the RMSE to the arithmetic mean of the actual values and tells to what degree the model can be relied upon.

Mathematically expressed as;

$$rRMSE = \frac{RMSE}{\bar{x}_i} \quad \text{..... Equation v}$$

It also has its grading for model performance as shown below;

Excellent	-	rRMSE < 0.10
Good	-	0.11 < rRMSE < 0.20
Fair	-	0.21 < rRMSE < 0.30

Poor - rRMSE > 0.30

vi. ***Cross-validation:*** is a statistical method used to evaluate and compare machine learning algorithms by splitting the given data into equal or almost equal parts and then using one section as training data and another section a validation data while holding out the rest splits. The procedure is being rotated throughout the splits. The type cross-validation used for this research is the k-fold cross validation with parameters n_splits=3, n_repeats=3, random_state=1.

(Xu et al., 2019; Chai & Draxler, 2014; Refaeilzadeh et al., 2009; Mehdizadeh et al., 2021; Mehdizadeh et al., 2020).

3.6.2 Performance metrics selection for binary classification task

i. ***Confusion Matrix:*** is a way to summarise the performance of an algorithm on a binary classification task in the form of true positives, false positives, true negatives and false negatives.

ii. ***Accuracy Score:*** is expressed as the ration of the sum of the correct predictions to the number of predictions.

iii. ***F1 Score:*** is expressed as the product of precision and recall.

(Raschka, 2014).

3.7.0 Training classifier and regressor models

For the regressor models, the selected features and target consisting of 7,906 observations was split into training into training and test data with 5,929 observations for the training data making up 80% of the entire data and 1,977 observations for the test data making up 20% of the entire data. The data was standardised using standard scaler by fitting and transforming the training features and target and transforming the test features and targets. This was done to improve performance of models that do not follow a rule-based approach as models that use rule-based approach are not biased by the magnitude of the training values

I trained the follow models for my price prediction task, namely;

i. Random Forest regressor
ii. Linear regressor
iii. Polynomial regressor
iv. Ransac regressor
v. SVM regressor
vi. Extreme gradient boost (XGBoost) regressor
vii. Stacking regressor
viii. KNN regressor,

While for the binary classification task, due to imbalanced nature of the target variable, I used resample technique to up sample the available data and got 11,812 observations. I then split the training and test data. My training datat for my classification task was 9,449 observations making up 80% of the entire resampled data and 2,363 observations making up 20% of the entire resampled data. I then scaled my data using standard scaler, fit and transformed my training data and transformed my test data.

I trained the following models for my binary classification task of sold or not sold.

i. Gaussian Naïve Bayes classifier
ii. Decision Tree classifier
iii. Random Forest Classifier
iv. Logistic Regression
v. SVM classifier
vi. Extreme gradient boost (XGBoost) classifier
vii. Stacking classifier
viii. Artificial Neural Network (ANN) using back propagation
ix. KNN classifier

3.8.0 Hyperparameter Optimisation

I carried out hyper parameter optimisation using GridSearchCV on Random Forest algorithm with the first 2000 observations in the dataset so as to explores the feature_importances_ attribute. This is to enable me compare listed hierarchy of feature_importances_ and best training features predicted by my k-best algorithm while also trying to improve the performance of the Random Forest classifier and regressor.

4.0.0 Results

4.1.0 Price prediction using regressor models

4.1.1 Performance of regression models trained with features from K-Best Selector

S/N	Model	R^2 train score	R^2 test score	MAE	MSE	RMSE	rRMSE	Average. cross-val train accuracy	Average cross-val test accuracy
1	Random Forest Regressor	0.9908	0.9771	855.79	2399123.35	1548.91	0.1215	0.9915	0.9680
2	Linear Regressor	0.6346	0.6314	3453.73	38557269.44	6209.45	0.5720	0.6356	0.6319
3	Polynomial Regressor	0.8527	0.7033	2237.10	31040020.36	5571.36	0.4443	0.8544	0.7872
4	Ransac Regressor	0.9468	0.9462	982.44	5629166.06	2372.59	0.1893	0.9541	0.9426
5	SVM Regressor	0.9316	0.8593	1507.43	14135702.06	3759.75	0.3035	0.9486	0.8649
6	XGBoost Regressor	0.9901	0.9793	828.20	2170510.56	1473.27	0.1151	0.9914	0.9709
7	Stacked Regressor	0.9878	0.9783	844.51	2269593.80	1506.52	0.1167	0.9878	0.9690
8	KNN Regressor	0.9976	0.8529	1674.04	15393506.76	3923.46	0.3113	0.9975	0.8456

Table 1: Performance of regression models trained with features from K-Best Selector.

4.1.2 Hyperparameter Optimisation using GridSearchCV and random forest regressor

i. **Best parameters:** {'criterion': 'absolute_error', 'max_depth': 20, 'max_leaf_nodes': 500, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

ii. **Feature importance:**

- 44.93875609849927: max_power
- 25.193036718320183: year
- 5.546869010017546: km_driven
- 5.371322594436802: torque_splitnm
- 3.7961627558994127: mileage
- 3.398323274395656: engine
- 2.5319937491949376: name
- 2.0113091921276305: torque_splitrpm
- 1.9748692211591896: City
- 1.362736515409125: State or Province
- 1.2283353074903298: seller_type
- 0.7629958730375949: seats
- 0.6585134552945917: Region
- 0.5681809550663641: owner
- 0.3998307710724765: transmission
- 0.25676450857888655: fuel
- 0.0: sold

iii. **Best Score:** 0.9344

iv. **Comparing price predicting feature selections:**

S/N	K-Best Price Predict Features	Random Forest Price Predict Features
1.	max_power	max_power
2.	year	year
3.	Km_driven	Km_driven
4.	torque_splitnm	torque_splitnm

Table 2: Comparison of price predicting feature selections.

v. **Performance of regression model trained with parameters from hyperparameter optimisation:**

S/N	Model	R^2 train score	R^2 test score	MAE	MSE	RMSE	rRMSE	Average. cross-val train accuracy	Average cross-val test accuracy
1.	Random Forest optimised regressor	0.9894	0.9762	843.55	2487678.63	1577.24	0.1236	0.9902	0.9677

Table 3: Performance of regression model trained with parameters from hyperparameter optimisation.

4.2.0 Sold prediction using classifier models

4.2.1 Performance of classifier models trained with features from K-Best Selector and 2363 test observations

S/N	Model	Training accuracy	Test accuracy	No. of accurately predicted test observations	No. of mislabeled test observations	F1_0 score	F1_1 score	Average. cross-val train accuracy	Average cross-val test accuracy
1	Gaussian Naive Bayes Classifier	0.6351	0.6306	1490	873	0.54	0.69	0.6343	0.6335
2	Decision Tree Classifier	0.9836	0.8388	1982	381	0.82	0.85	0.9852	0.8165
3	Random Forest Classifier	0.9836	0.8629	2039	324	0.85	0.87	0.9852	0.8407
4	Logistic Regression	0.6457	0.6306	1490	873	0.56	0.68	0.6423	0.6420
5	SVM Classifier	0.6528	0.6327	1495	868	0.56	0.68	0.9624	0.8697
6	XGBoost Classifier	0.9836	0.8684	2052	311	0.86	0.88	0.9851	0.8396
7	Stacked Classifier	0.9661	0.8904	2104	259	0.89	0.89	0.8979	0.8459
8	Artificial Neural Network (ANN)	0.6091	0.6162	-	-	-	-	-	-

Table 4: Performance of classifier models trained with features from K-Best Selector and 2363 test observations.

4.2.2 Confusion matrix performance report of models trained with K-Best features and 2363 test observations

S/N	Model	True Positives	False Positives	False Negatives	True Negatives
1	Gaussian Naive Bayes Classifier	507	660	213	983
2	Decision Tree Classifier	867	300	81	1115
3	Random Forest Classifier	912	255	69	1127
4	Logistic Regression	559	608	265	931

5	SVM Classifier	553	614	254	942
6	XGBoost Classifier	932	235	76	1120
7	Stacked Classifier	1039	128	131	1065

Table 5: Confusion matrix performance report of models trained with K-Best features and 2363 test observations.

4.2.3 Hyperparameter Optimisation using GridSearchCV and random forest classifier

i. **Best parameters:** {'criterion': 'entropy', 'max_depth': 20, 'max_leaf_nodes': 200, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 600}.

ii. **Feature importance:**

- 71.4700363072066: fuel
- 7.797646514938992: City
- 2.6723011979711213: km_driven
- 2.441696952459786: engine
- 2.3685928992635703: Region
- 1.9856988886852056: seats
- 1.8366477519907733: year
- 1.7439231157725756: torque_splitnm
- 1.6921312647828788: State or Province
- 1.449048863618103: max_power
- 1.3948884430467985: torque_splitrpm
- 1.2318923352832665: name
- 0.7333329922747106: mileage
- 0.3856831506623154: transmission
- 0.3500769846740044: sold
- 0.31186314572658724: seller_type
- 0.1345391916427289: owner

iii. **Best Score:** 0.9389

iv. **Comparing sold predicting feature selections:**

S/N	K-Best Sold Predict Features	Random Forest Sold Predict Features
1.	torque_splitrpm	fuel
2.	State or Province	City
3.	engine	km_driven
4.	mileage	engine
5.	selling_price	Region

Table 6: Comparison of sold predicting feature selections.

v. **Performance of classifier model trained with features from hyperparameter optimisation and 2363 test observations:**

S/N	Model	Training accuracy	Test accuracy	No. of accurately predicted points	No. of mislabeled points	F1_0 score	F1_1 score	Average. cross-val train accuracy	Average cross-val test accuracy
1.	Random Forest optimised classifier	0.9594	0.9412	2224	139	0.94	0.94	0.9633	0.9417
2.	XGBoost optimised classifier	0.9995	0.9615	2272	91	0.96	0.96	0.9995	0.9576
3.	Stacked optimised classifier	0.9881	0.9636	2277	86	0.96	0.96	0.9916	0.9538
4.	KNN optimised classifier	0.9995	0.9416	2225	138	0.94	0.94	0.9990	0.9237
5.	Optimised Artificial Neural Network (ANN)	0.9297	0.9289	-	-	-	-	-	-

Table 7: Performance of classifier model trained with features from hyperparameter optimisation.

vi. **Confusion matrix performance report of models trained with hyperparameter features and 2363 test observations:**

S/N	Model	True Positives	False Positives	False Negatives	True Negatives
1	Random Forest optimised classifier	1041	137	2	1183
2	XGBoost optimised classifier	1098	80	11	1174
3	Stacked optimised classifier	1123	55	31	1154
4	KNN optimised classifier	1074	104	34	1151

Table 8: Confusion matrix performance report of models trained with hyperparameter features and 2363 test observations.

5.0.0 Discussion, conclusion, recommendation and further study

5.1.0 Discussion and conclusion

5.1.1 Business related

As shown from the data set, the top four features that influences the prices of a used car are the maximum power, year of manufacture, total distance already covered by the car (odometer meter) and torque rating in newton per meter (Phlips, 1983; Scherer, 1996).

Also, from the data set top five features that influences bias if a used car would be sold or not sold are fuel type, City of purchase, total distance already travelled (odometer reading), engine capacity and region of sale (Sallee et al., 2016; Yerger, 1996).

For further business-related insights on the dataset see Appendix A.

5.1.2 Technical Related

The best model for price prediction is the Xgboost regressor. It had a near excellent rating of 11% relative root mean square error. It also had the best accuracy for training and test datasets at 99.01% and 97.93% respectively. Finally, it had the best MAE and RMSE of 828.20 and 1473.27 respectively. Predictions using the XGBoost model can be seen in figure 5.

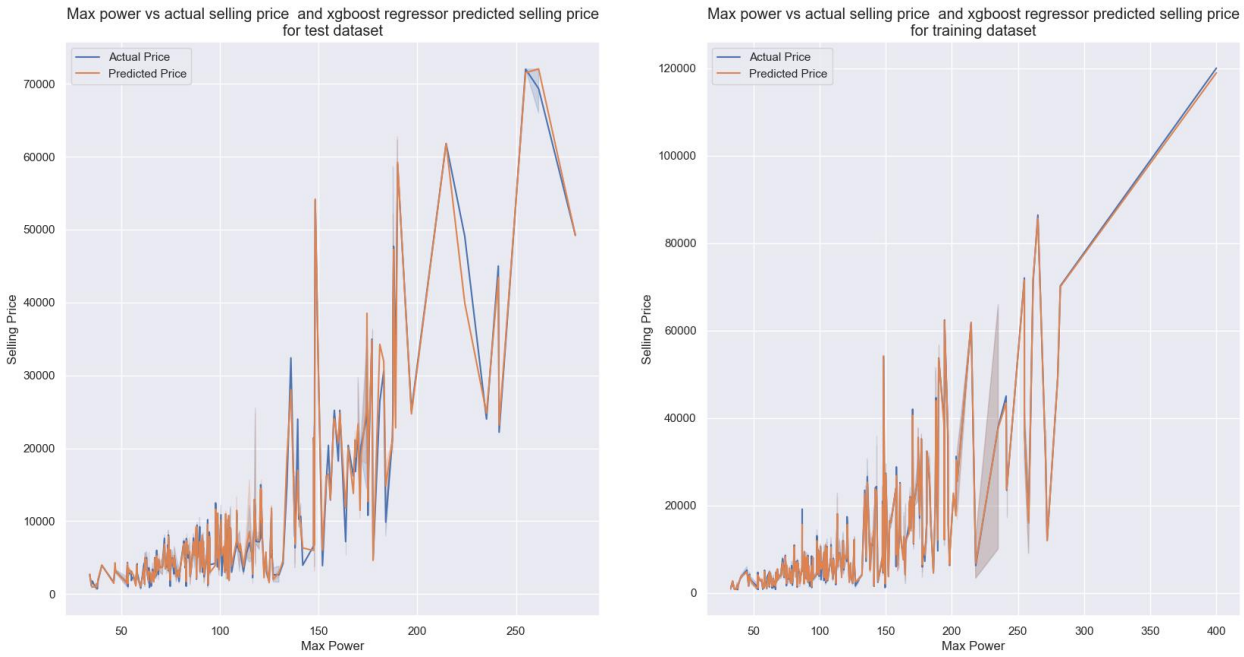


Figure 5:XGBoost regressor price prediction for test and train data

The best model for predicting sold and not sold is the stacking classifier using the best predicting features as suggested by the random forest classifier during hyperparameter optimisation. It had the least number of mislabeled points of 86 out of 2363 observations. It also had a training and test accuracy of 98.81% and 96.36% respectively thereby having the best generalising accuracy. Finally, it had a very good balance

between mislabeled points of zeros and ones in 55 and 31 respectively being the best recorded amongst all models trained and validated.

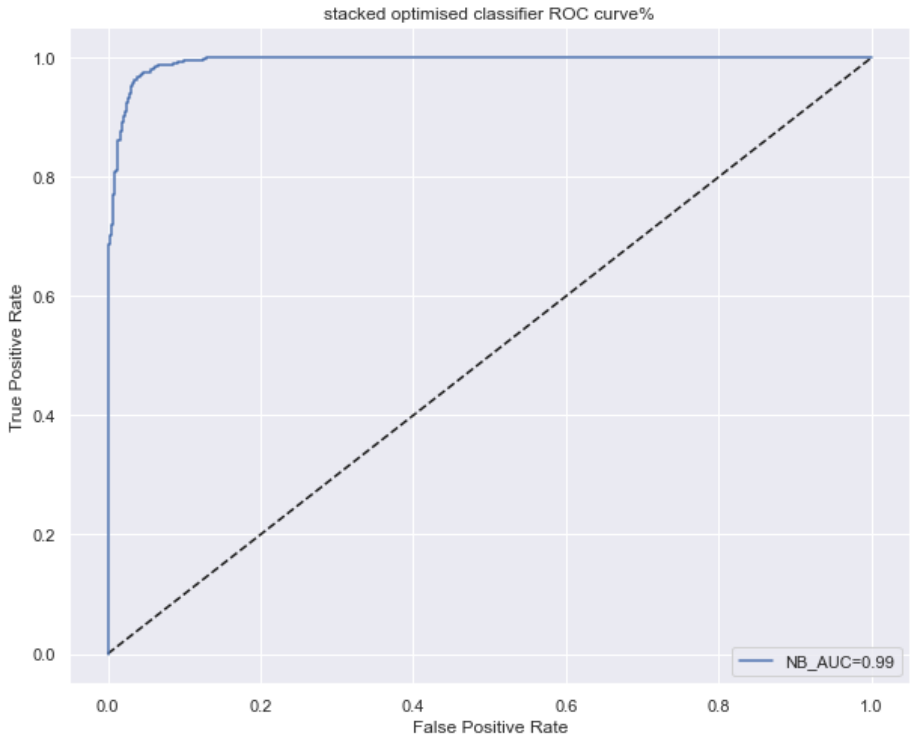


Figure 6: ROC curve performance of the stacking classifier trained with random forest best predicting features.

5.3.0 Research observations, recommendations and further study

i. Linear regressors struggled with predicting the prices probably due to low specificity and high heterogeneity of the data which is a known problem for this type of model as shown in figure 7 (Voß & Lessmann, 2017).
ii. Random Forest feature selection outperformed K-Best feature selection in classification task.
iii. SVM classifier performed much better during cross-validation that during normal training and validation. This could be an area of interest for further study to find out what happened and why.
iv. I would recommend further research to explore if the level of cleanliness of a used car can influence its price.
v. Performance charts for the regressor and classifier models can be found in Appendix B.

Figure 7: Linear Regressor price prediction for test and train data

In conclusion, XGBoost regressor was my best price predicting model and stacking classifier was my best performing model for the binary classification task.

6.0.0 Bibliography

Akerlof, G. A. (1978) The market for “lemons”: Quality uncertainty and the market mechanism. In Anonymous *Uncertainty in economics.* Elsevier, 235-251.

Asghar, M., Mehmood, K., Yasin, S. & Khan, Z. M. (2021) Used cars price prediction using machine learning with optimal features. *Pakistan Journal of Engineering and Technology,* 4 (2), 113-119.

Bento, A., Roth, K. & Zuo, Y. (2018) Vehicle lifetime trends and scrappage behavior in the U.S. used car market. *The Energy Journal,* 39 (1), .

Caruana, R., Munson, A. & Niculescu-Mizil, A. (2006) Getting the most out of ensemble selection. *Sixth International Conference on Data Mining (ICDM'06).* IEEE.

Chai, T. & Draxler, R. R. (2014) Root mean square error (RMSE) or mean absolute error (MAE). *Geoscientific Model Development Discussions,* 7 (1), 1525-1534.

Duvan, B. S. & Ozturkcan, S. (2009) Used car remarketing. *International Conference on Social Sciences (ICSS).*

G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou & D. Zhao. (2020) Flight delay prediction based on aviation big data and machine learning.

Mehdizadeh, S., Fathian, F., Safari, M. J. S. & Khosravi, A. (2020) Developing novel hybrid models for estimation of daily soil temperature at various depths. *Soil and Tillage Research,* 197 104513.

Mehdizadeh, S., Mohammadi, B., Pham, Q. B. & Duan, Z. (2021) Development of boosted machine learning models for estimating daily reference evapotranspiration and comparison with empirical approaches. *Water,* 13 (24), 3489.

N. Monburinon, P. Chertchom, T. Kaewkiriya, S. Rungpheung, S. Buya & P. Boonpou. (2018) Prediction of prices for used car by using regression models. - *2018 5th International Conference on Business and Industrial Research (ICBIR).*

Phlips, L. (1983) *The economics of price discrimination* Cambridge University Press.

Raschka, S. (2014) An overview of general performance metrics of binary classifier systems. *arXiv Preprint arXiv:1410.5330,* .

Refaeilzadeh, P., Tang, L. & Liu, H. (2009) Cross-validation. *Encyclopedia of Database Systems,* 5 532-538.

Sallee, J. M., West, S. E. & Fan, W. (2016) Do consumers recognize the value of fuel economy? evidence from used car prices and gasoline price fluctuations. *Journal of Public Economics,* 135 61-73.

Scherer, F. M. (1996) *Industry structure, strategy, and public policy* Prentice Hall.

Stojiljkovic, M. (2021) Linear regression in python. *Real Python.Https://Realpython.Com/Linear-Regression-in-Python/.Accessed,* 8 .

V. Bahel, S. Pillai & M. Malhotra. (2020) A comparative study on various binary classification algorithms and their improved variant for optimal performance. - *2020 IEEE Region 10 Symposium (TENSYMP).*

Voß, S. & Lessmann, S. (2017) Resale price prediction in the used car market. *International Journal of Forecasting,* .

Xu, Z., Li, W., Li, Y., Shen, X. & Ruan, H. (2019) Estimation of secondary forest parameters by integrating image and point cloud-based metrics acquired from unmanned aerial vehicle. *Journal of Applied Remote Sensing,* 14 (2), 022204.

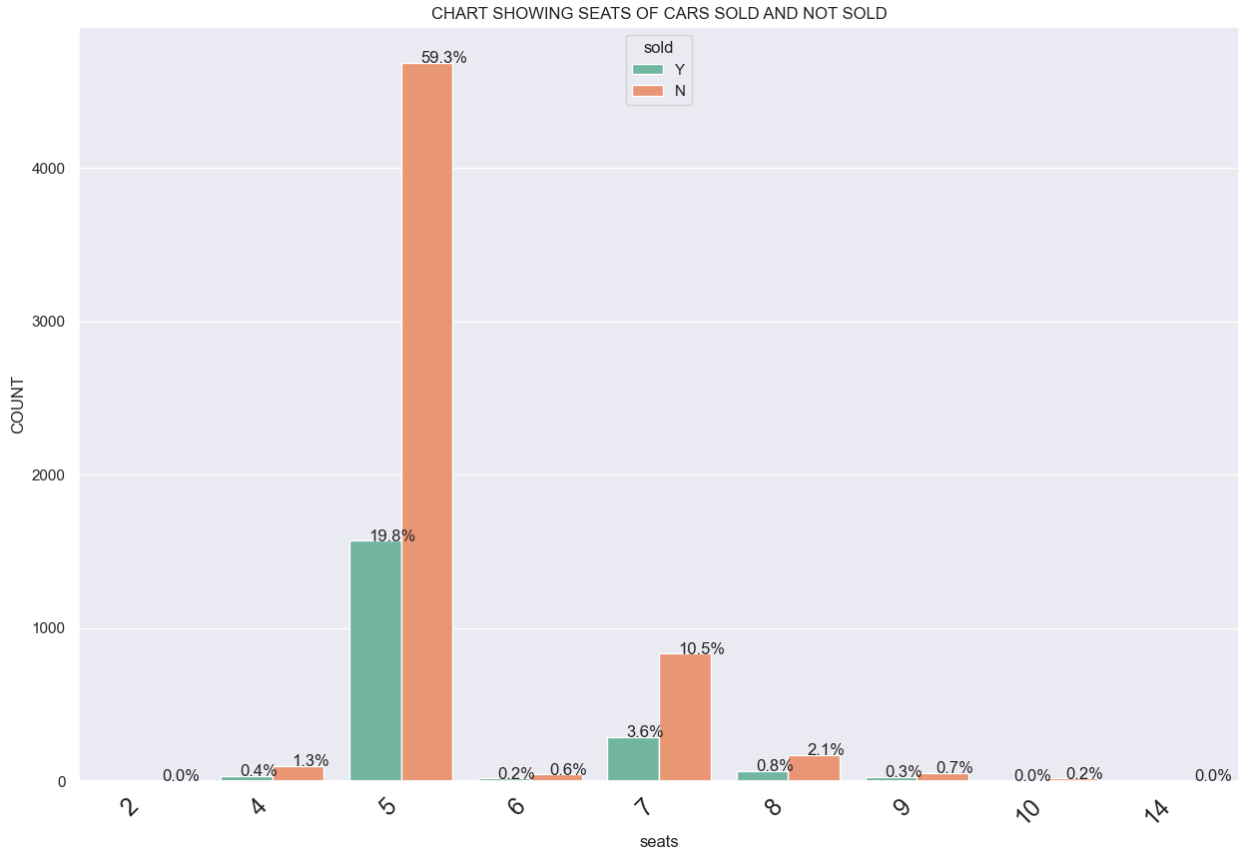
Yerger, D. B. (1996) Used car markets: Reliability does matter, but do consumer reports? *Applied Economics Letters,* 3 (2), 67-70.

7.0.0 Appendixes

7.1.0 Appendix A

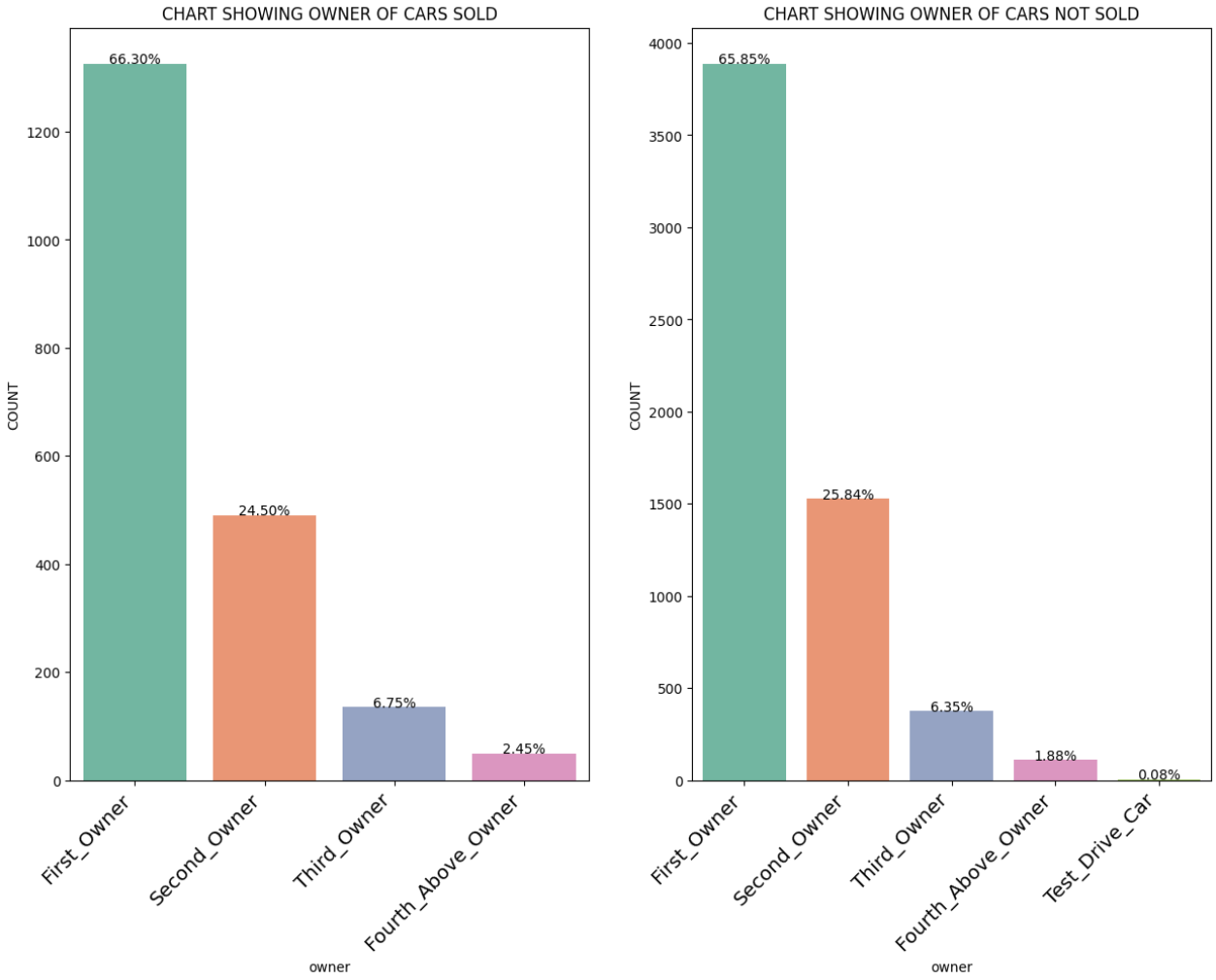
The following analysis was done considering only car models that have over 50 observations in the dataset.

- Majority of the cars available are 5-seater cars sold to non-sold ratio is fairly balanced as shown in the chart below.



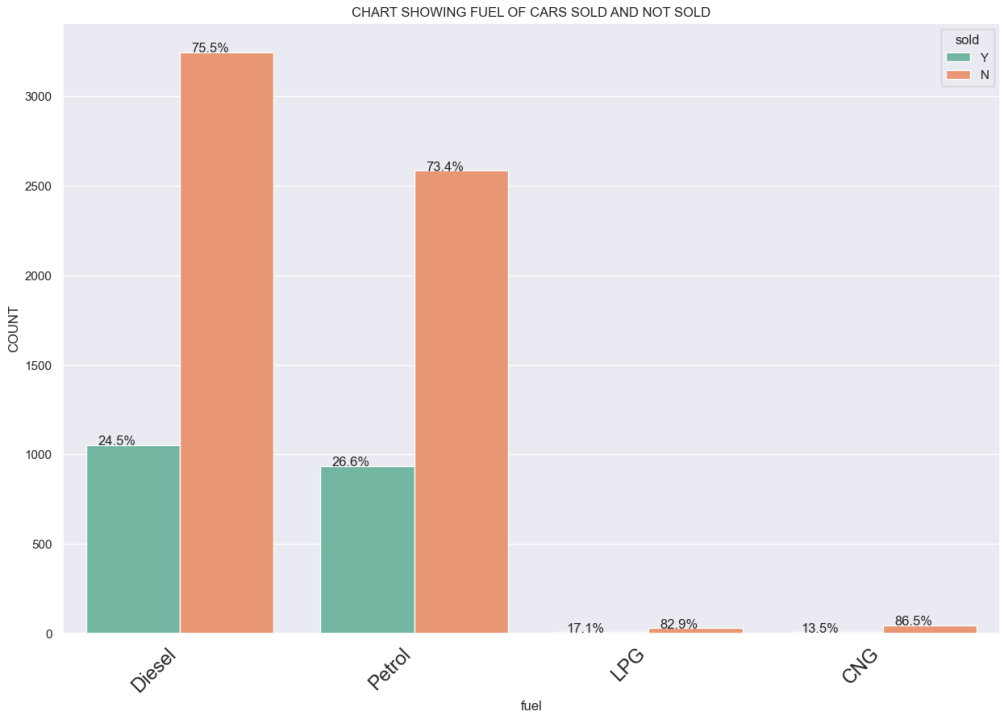
Picture showing distribution of sold and not-sold seats

- For the owner's column, the fourth and above owner column has more sold than not-sold ratio than any other owner type and as you move back from the fourth seller to the first



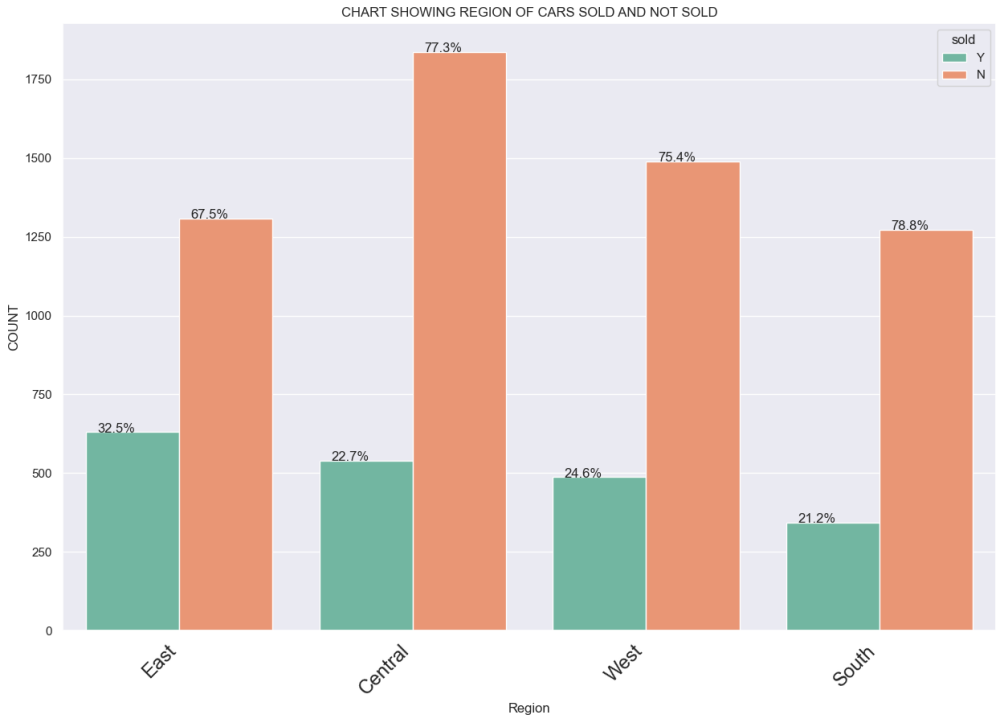
Picture showing distribution of owner column

- The top three cars with sale probability includes Volkswagen which has the highest sale probability with respect to the fourth owner and above with a sold percentage of 7.69% and then Chevrolet with 7.14% and finally Hyundai with 4.22%
- The top three sold cars overall are; Nissan with 40.74%, Mercedes with 29.63% and Jaguar with 29.58% of sold cars in ratio with not-sold cars.
- The top two fuel types sold are petrol vehicles with 26.56% and diesel vehicles with 24.47%.



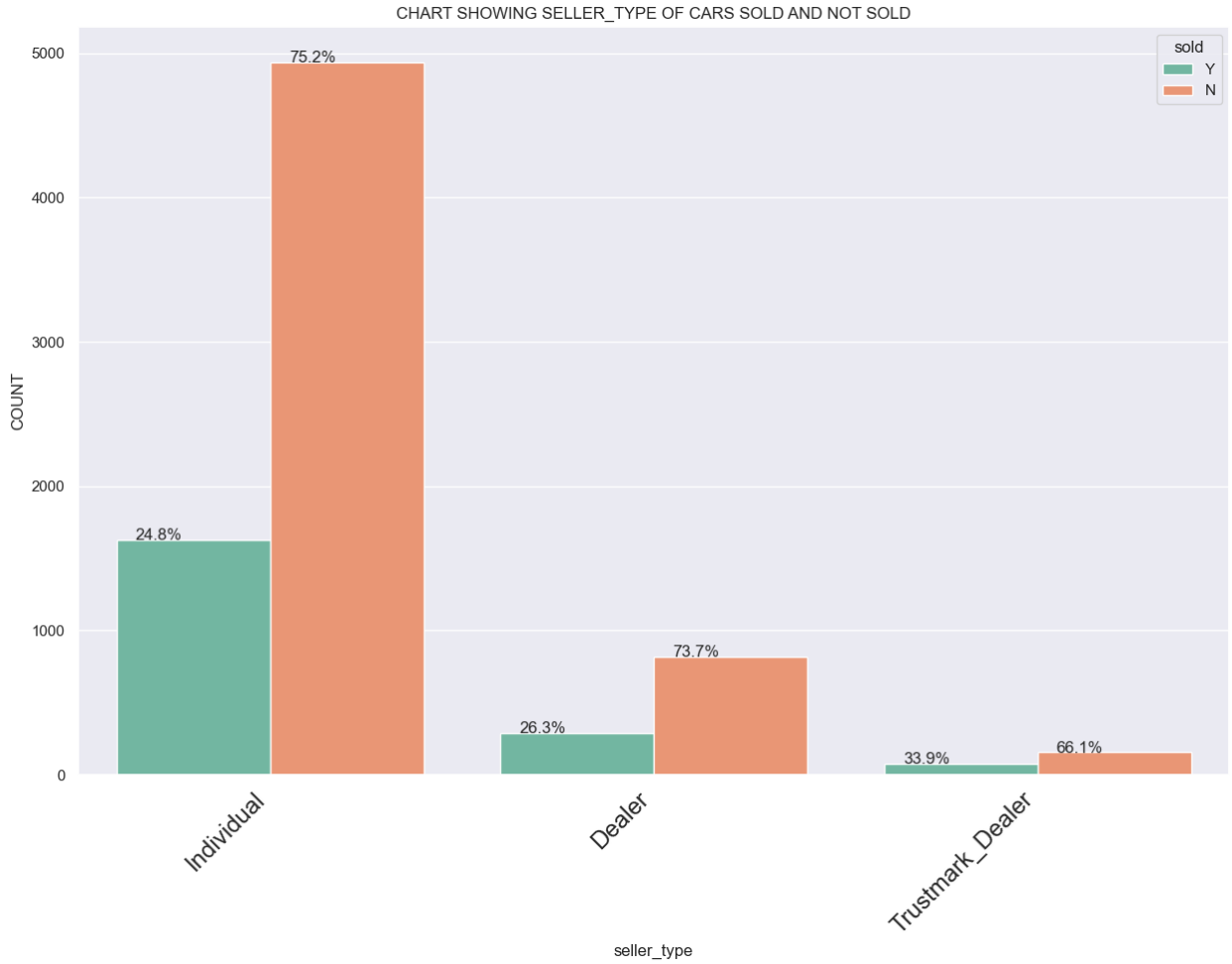
Picture showing fuel type sold to not-sold ratio for different fuel type.

Eastern region has the highest sale probability with statistics showing 32.5% sold and 67.5% not-sold. The stats of the next region closest to the stats of eastern region has approximately about 10% difference. See picture below.

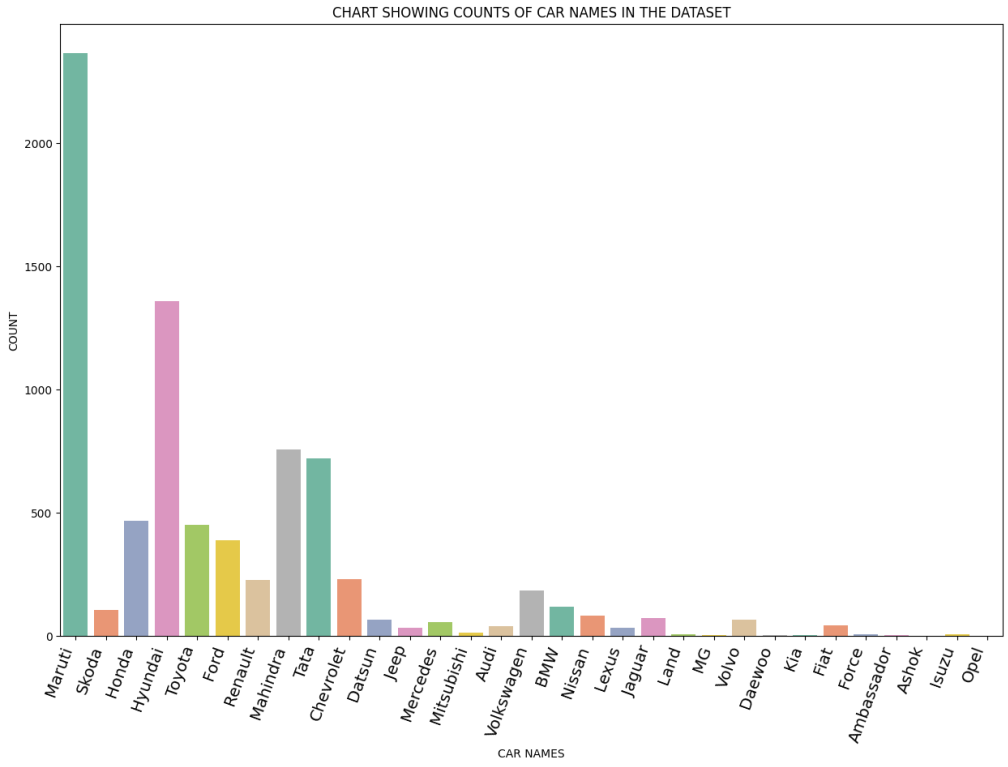


Picture showing the distribution of region performance to sold and not-sold.

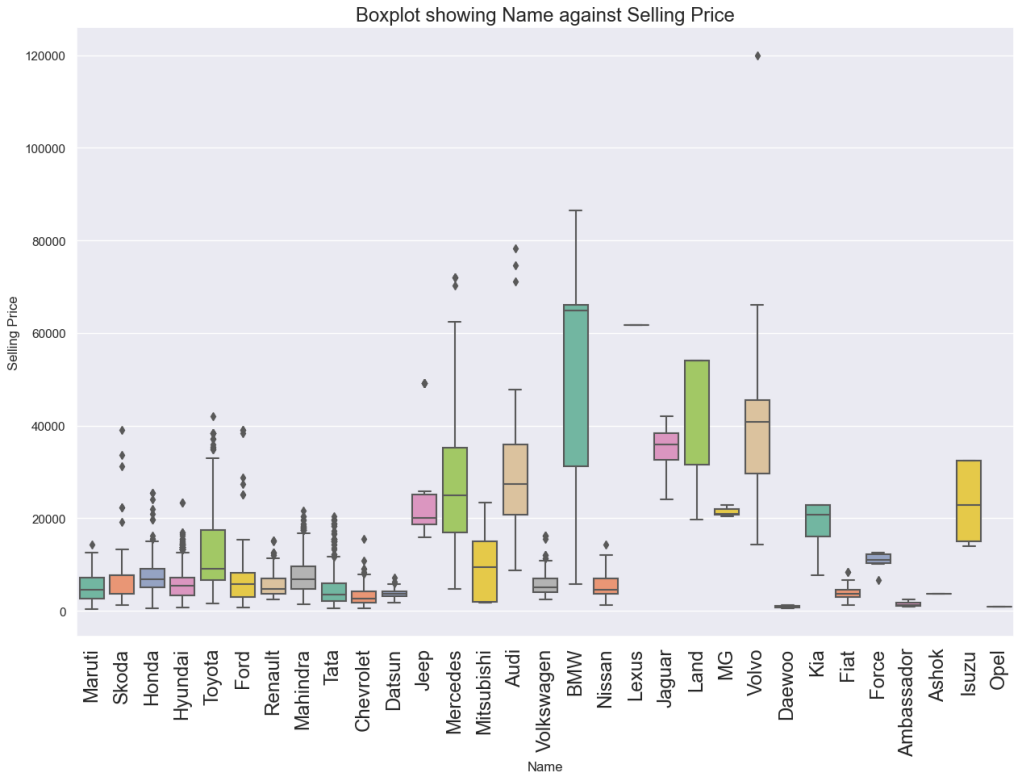
- Average selling price of an automatic car costs around 23802.2 dollars while that of manual car costs around 5499.07 dollars. Automatic cars seem to cost up to four times the manual cost on average.
- Car from Trustmark dealer sells more than cars from other sellers with a sold percentage of 33.19% and not-sold percentage of 66.1%



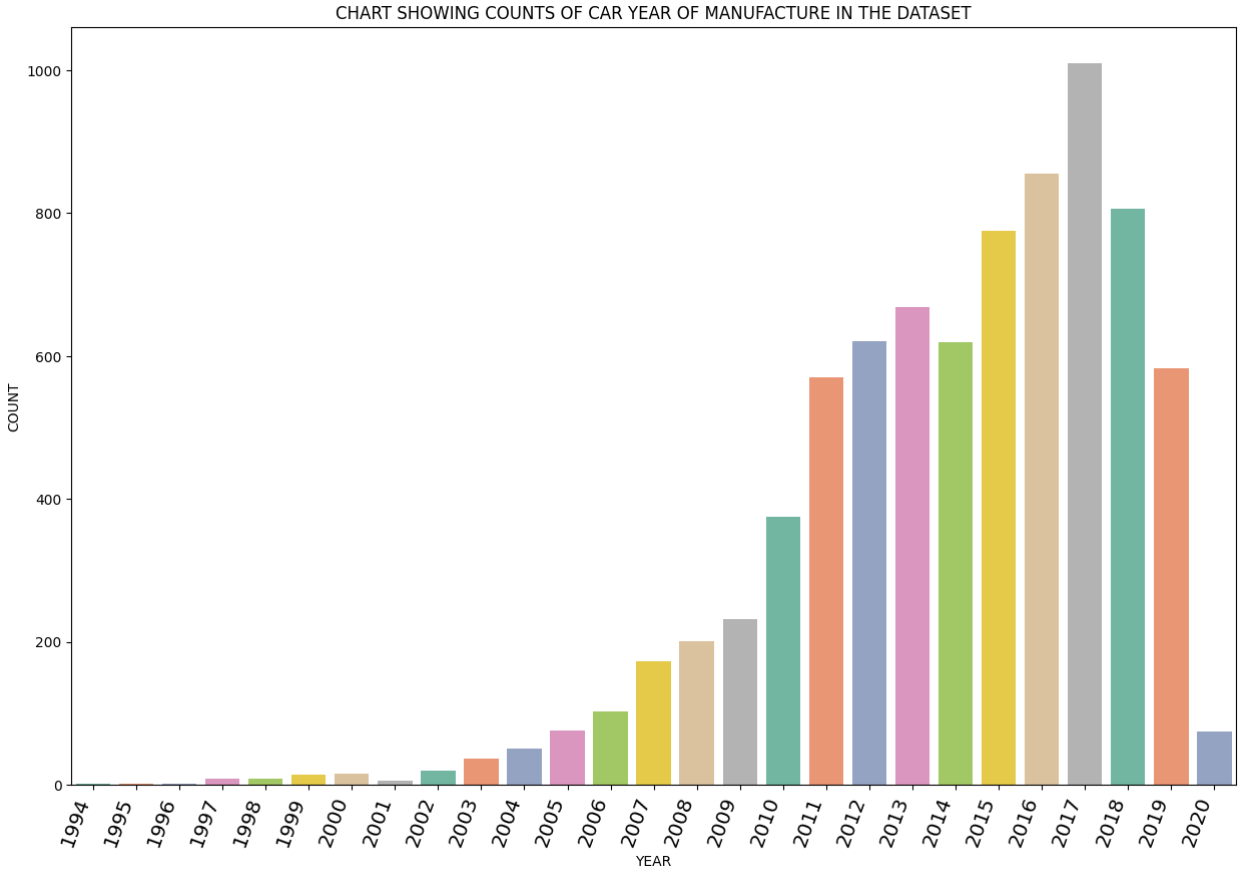
Picture showing distribution of seller type with respect to sold and not-sold.



Picture showing distribution of car names vs counts



Picture showing box plot of car names vs range of selling price and average selling price

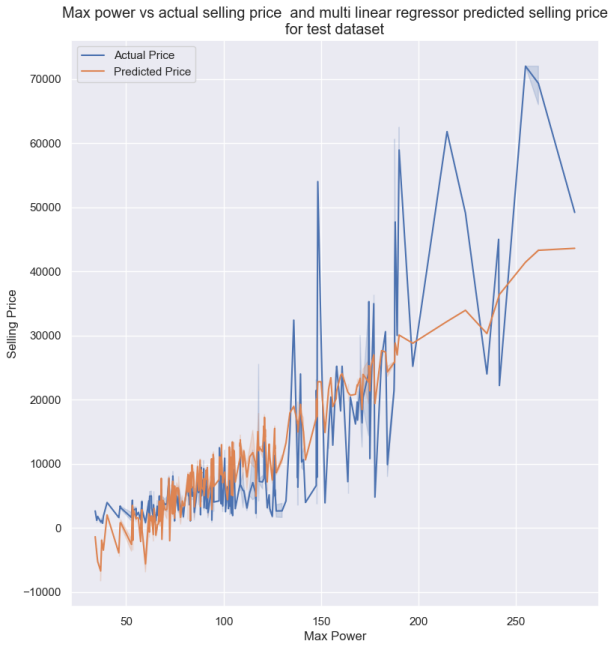


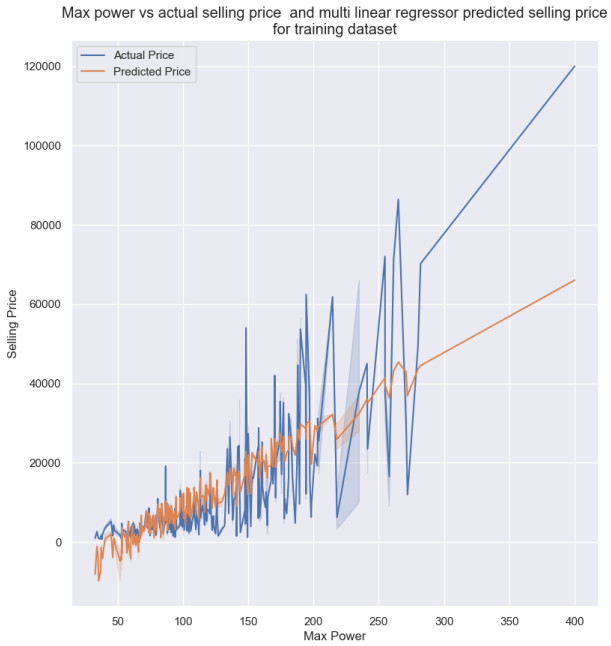
Plot showing distribution of car year of manufacture in the dataset.

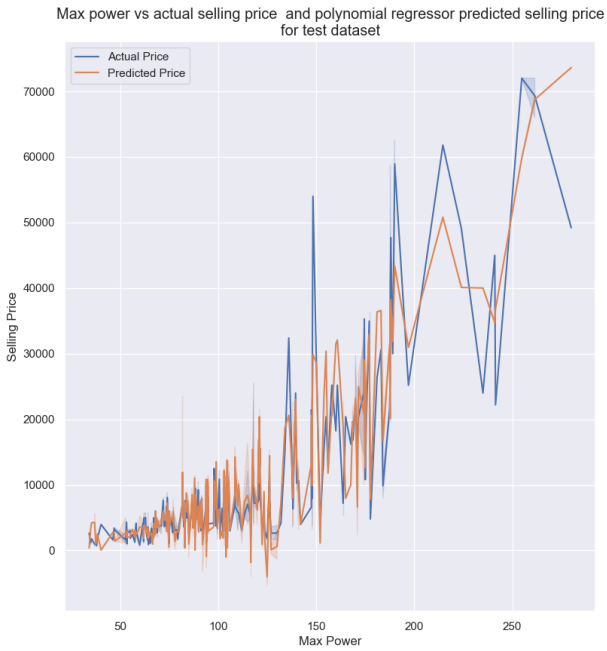
7.2.0 Appendix B

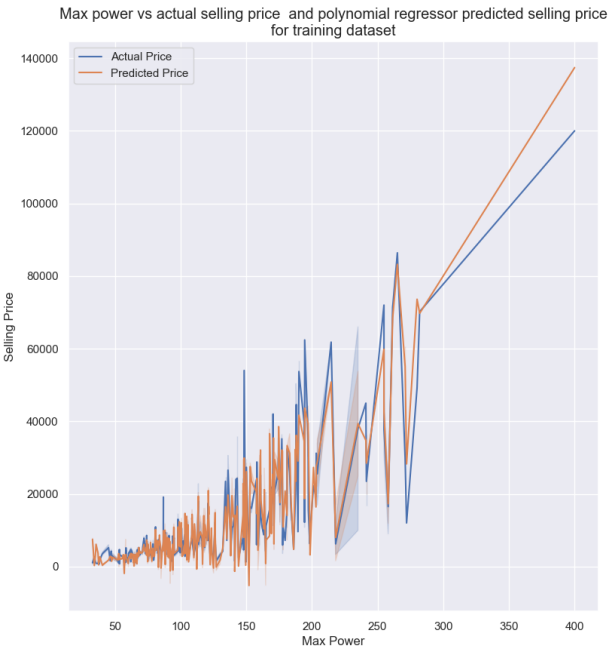
Below are the performance charts for regressor and classifier models.

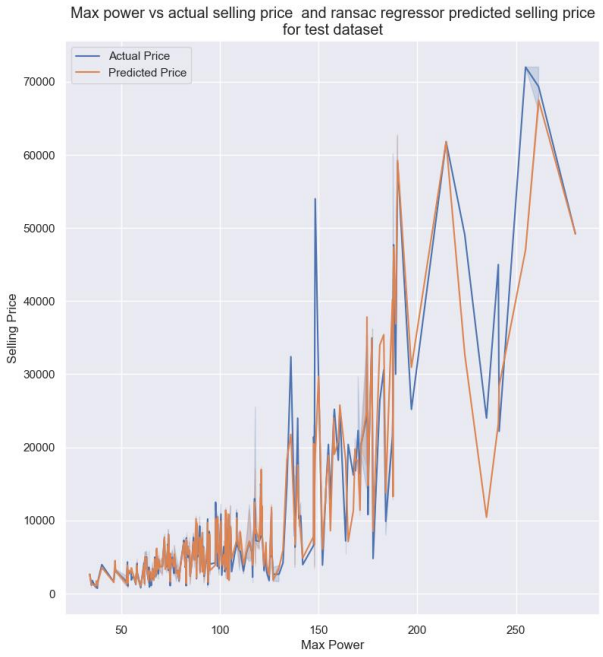
7.2.1 Regressor models

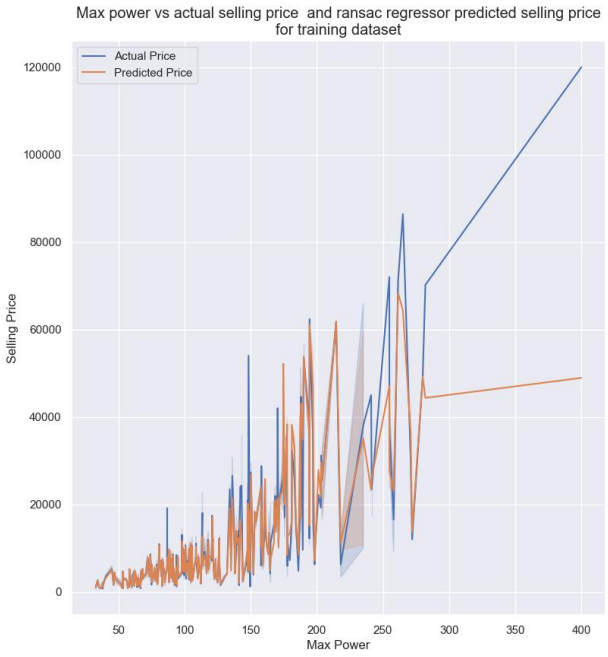


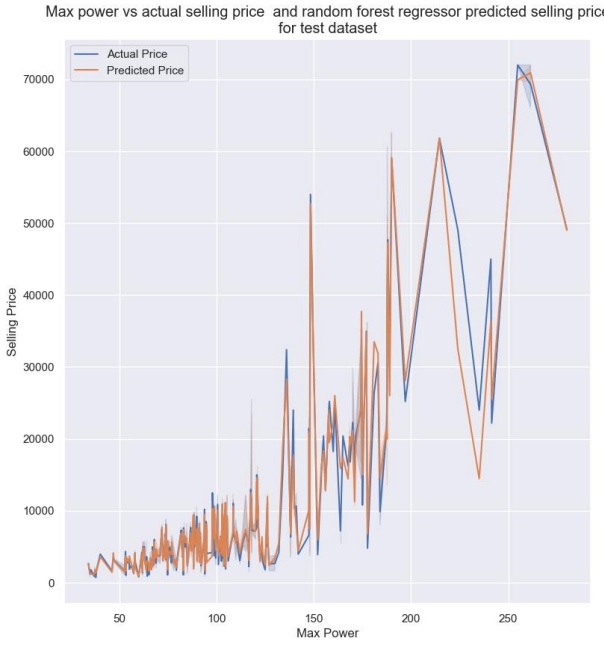


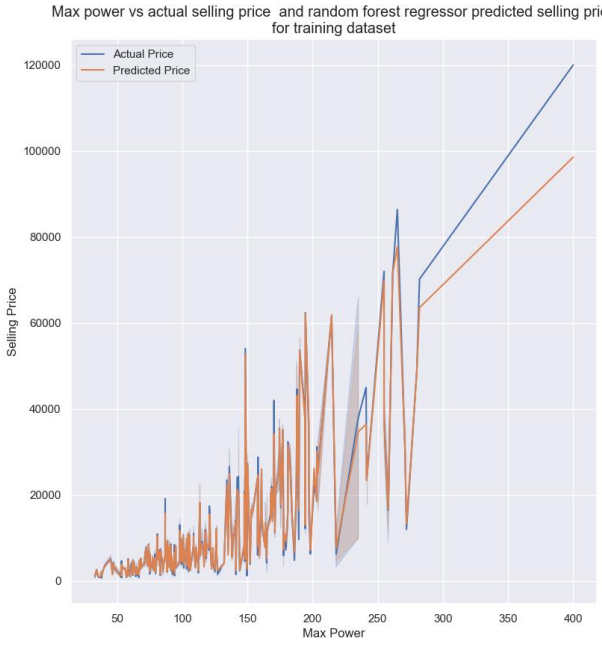


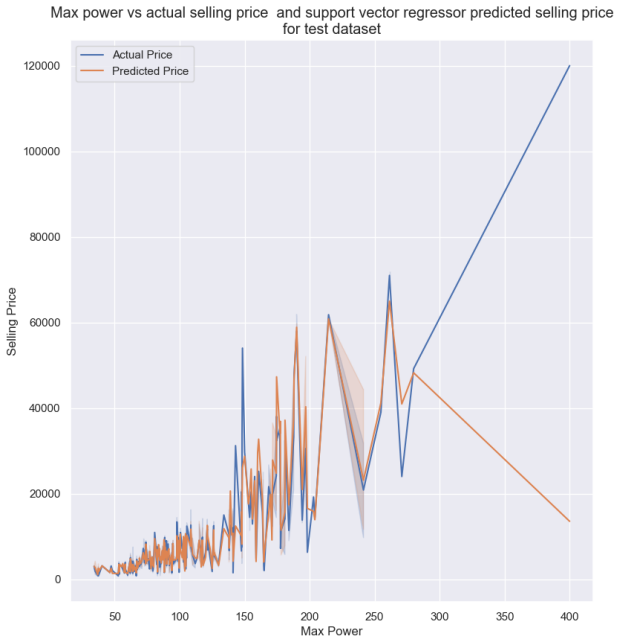


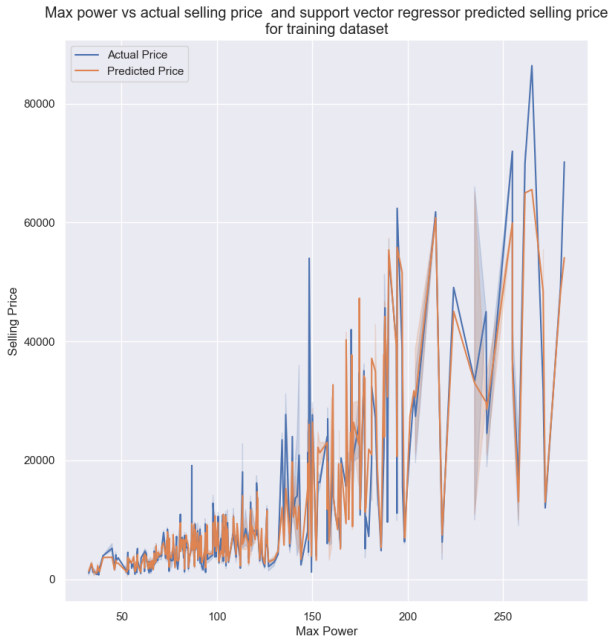

Max power vs actual selling price and ransac regressor predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Max Power

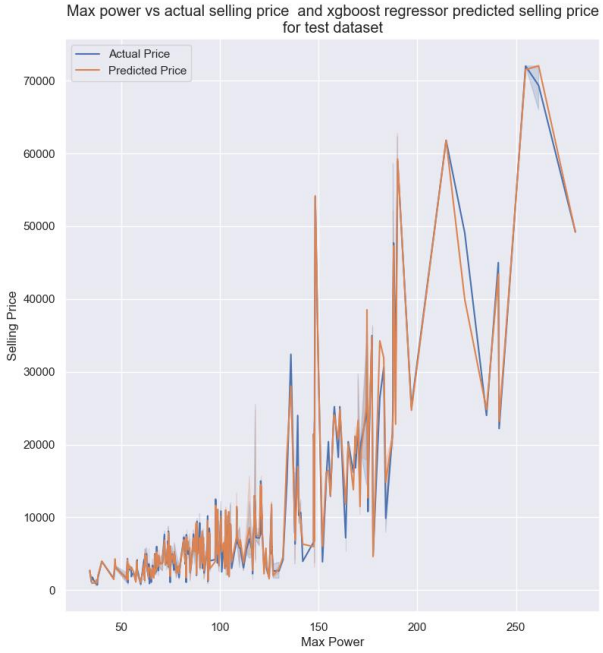

Max power vs actual selling price and ransac regressor predicted selling price for training dataset
Actual Price
Predicted Price
Selling Price
Max Power

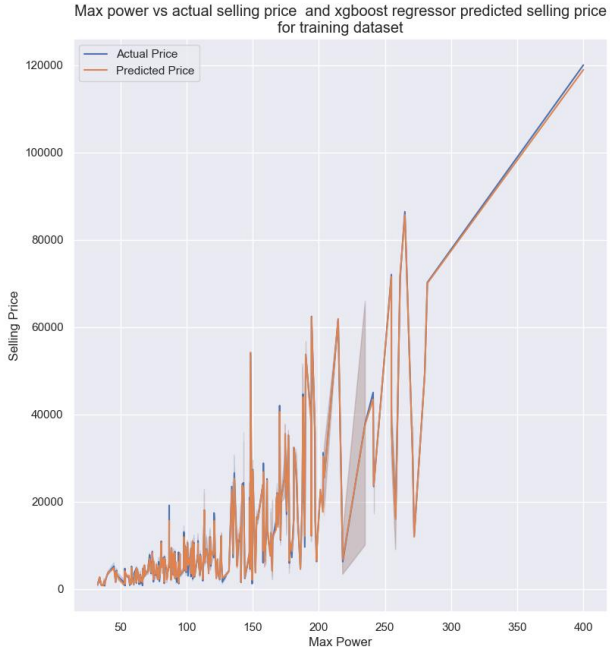

Max power vs actual selling price and random forest regressor predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Max Power

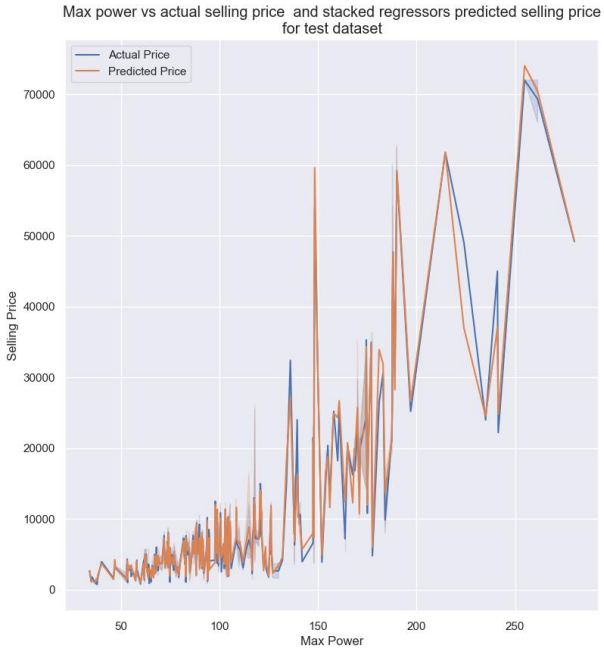

Max power vs actual selling price and random forest regressor predicted selling price for training dataset
Actual Price
Predicted Price
Selling Price
Max Power

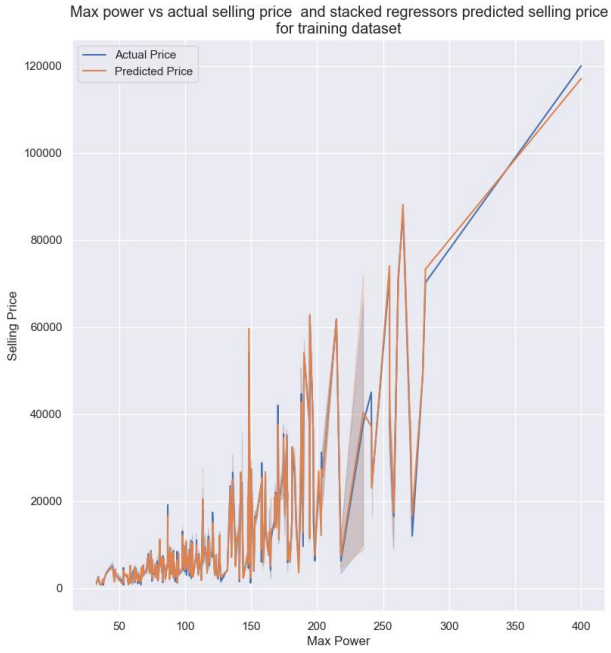

Max power vs actual selling price and support vector regressor predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Max Power

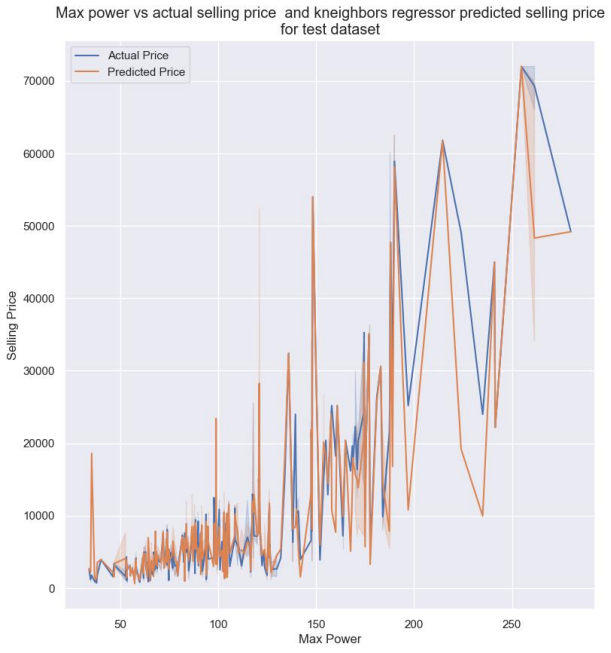

Max power vs actual selling price and support vector regressor predicted selling price for training dataset
Actual Price
Predicted Price
Selling Price
Max Power

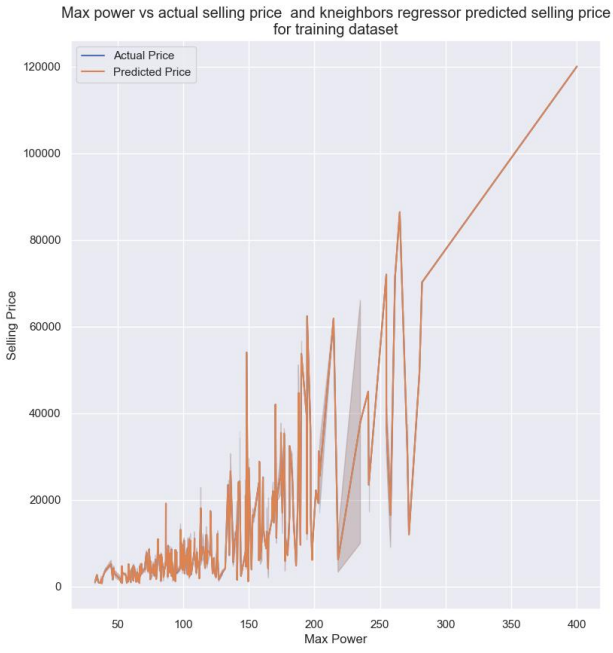

Max power vs actual selling price and xgboost regressor predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Max Power

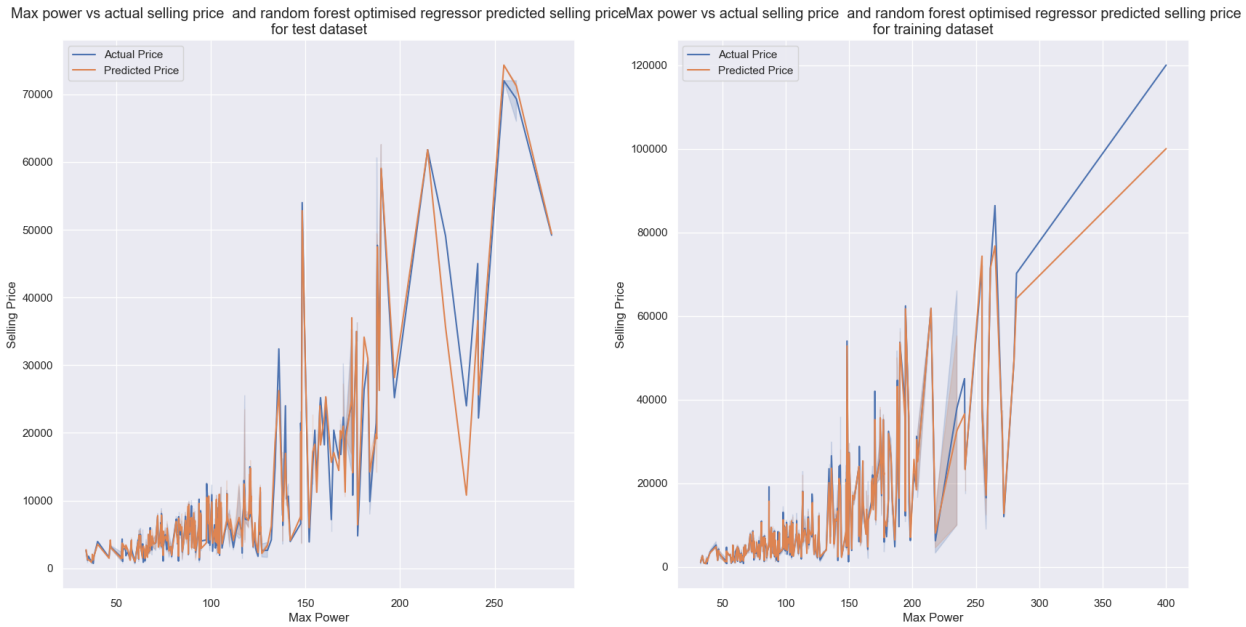

Max power vs actual selling price and xgboost regressor predicted selling price for training dataset
Actual Price
Predicted Price
Selling Price
Max Power


Max power vs actual selling price and stacked regressors predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Max Power

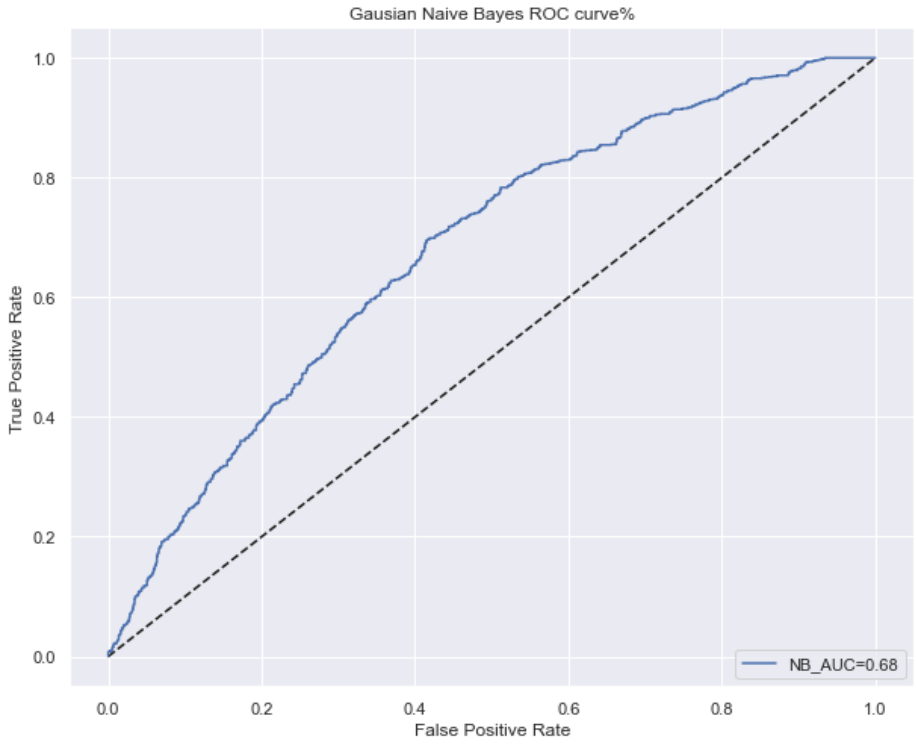

Max power vs actual selling price and stacked regressors predicted selling price for training dataset
Actual Price
Predicted Price
Selling Price
Max Power

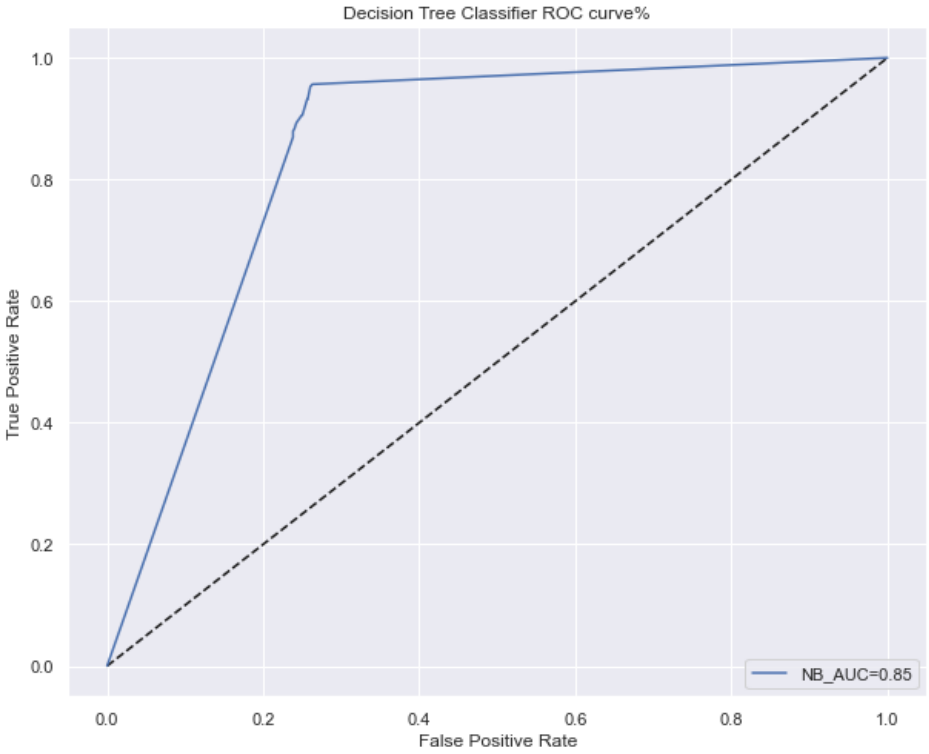

Max power vs actual selling price and kneighbors regressor predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Max Power

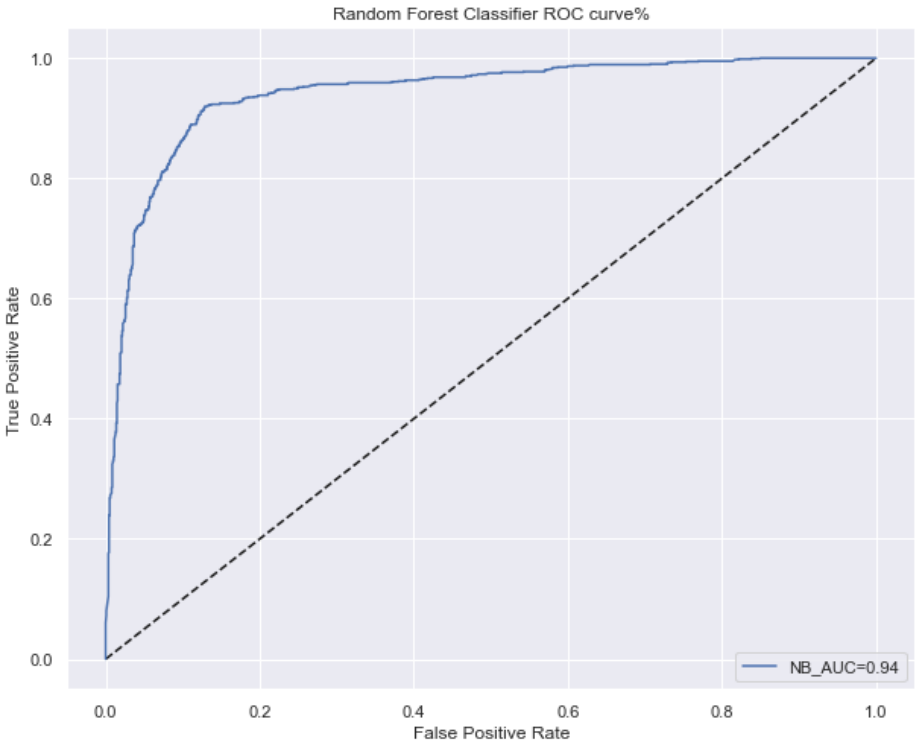

Max power vs actual selling price and kneighbors regressor predicted selling price for training dataset
Actual Price
Predicted Price
Selling Price
Max Power

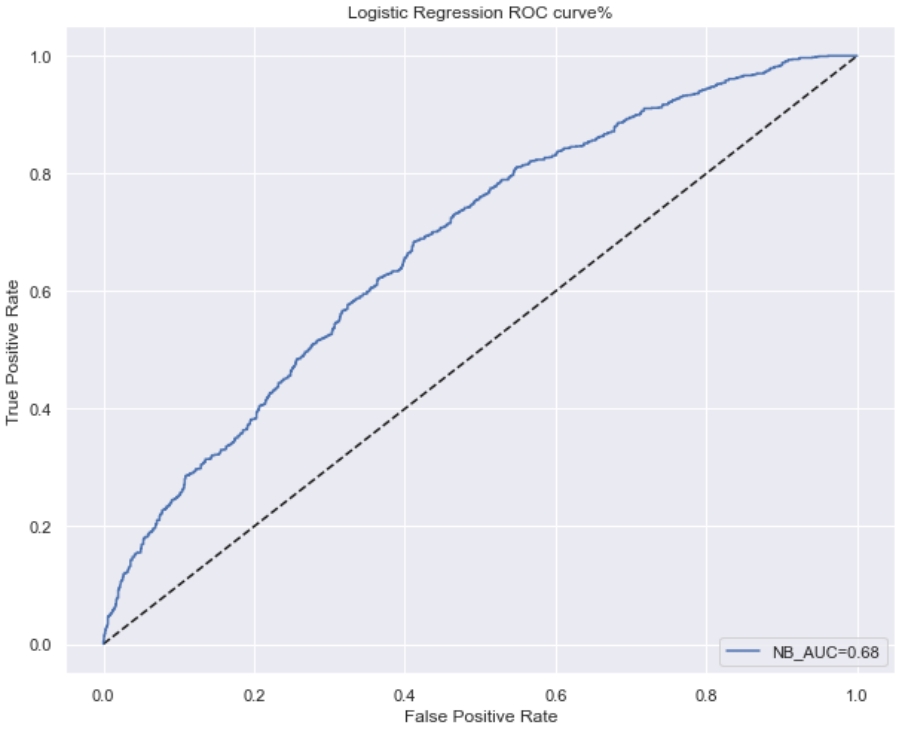


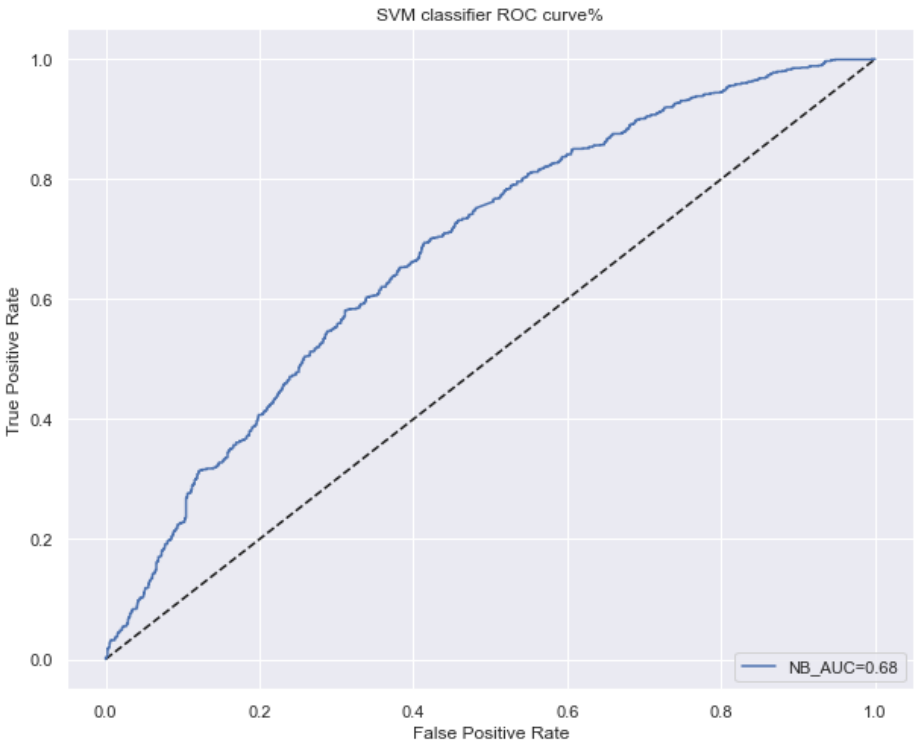
7.2.2 Classifier model

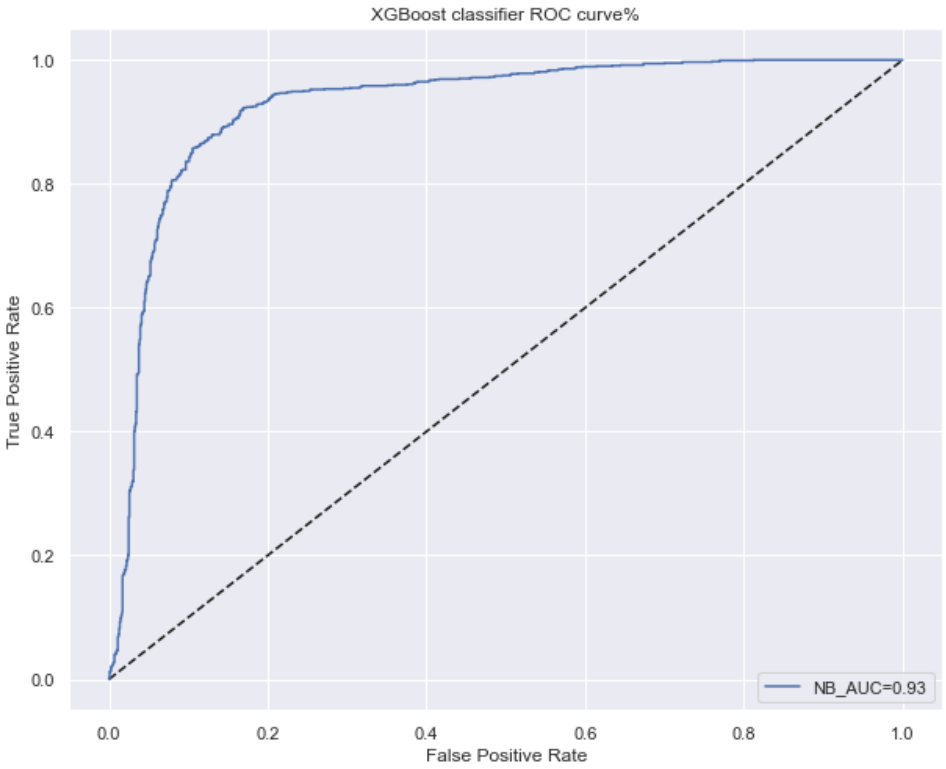


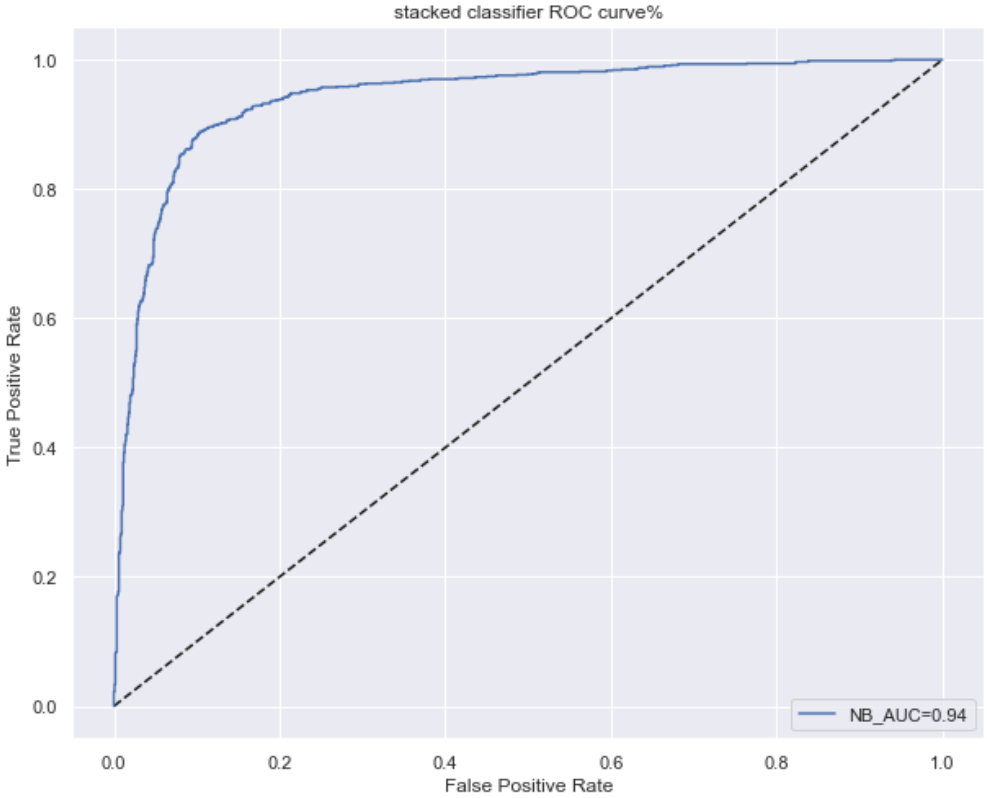

Decision Tree Classifier ROC curve%
True Positive Rate
False Positive Rate
NB_AUC=0.85
0.0
0.2
0.4
0.6
0.8
1.0

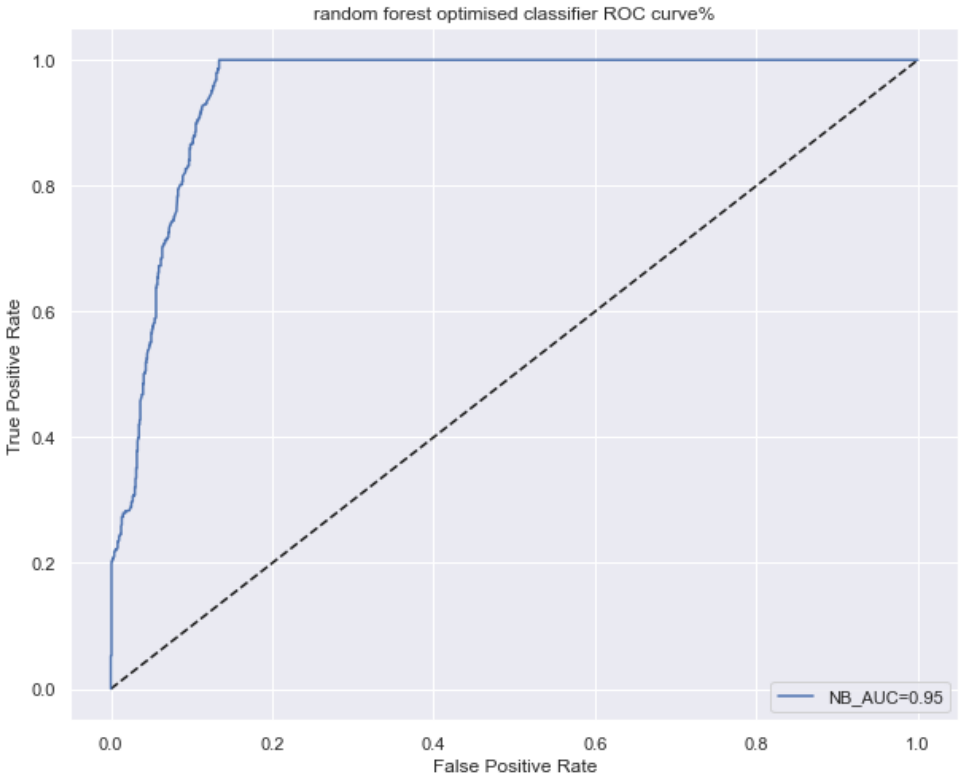

Random Forest Classifier ROC curve%
True Positive Rate
False Positive Rate
NB_AUC=0.94
0.0
0.2
0.4
0.6
0.8
1.0

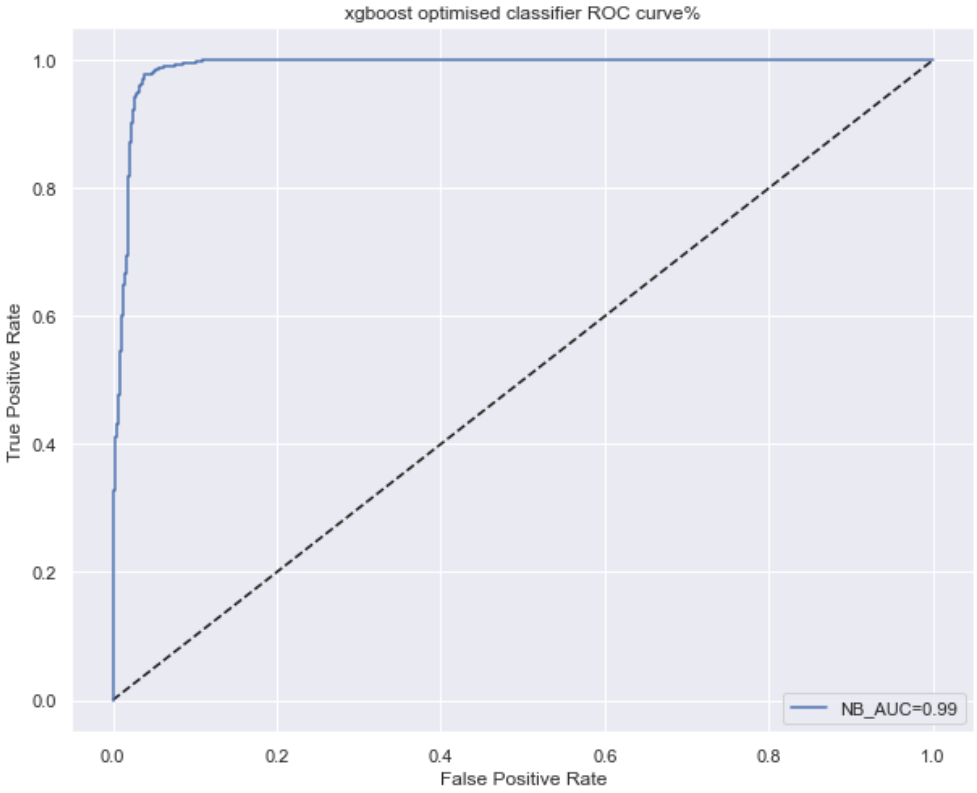

Logistic Regression ROC curve%
True Positive Rate
False Positive Rate
NB_AUC=0.68
0.0
0.2
0.4
0.6
0.8
1.0

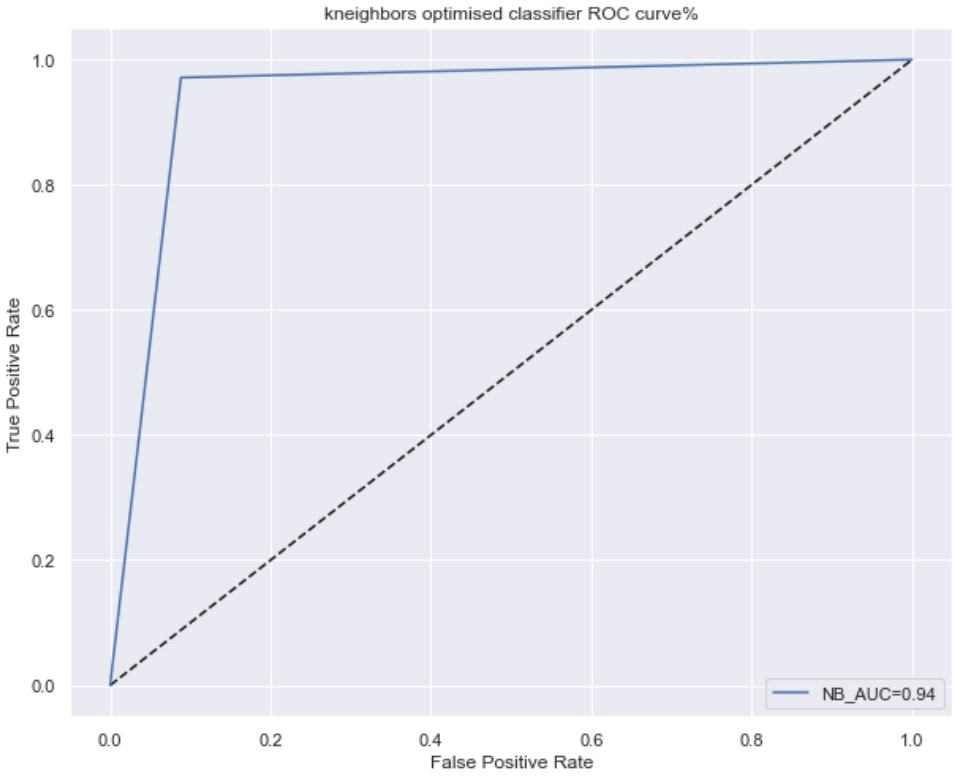

SVM classifier ROC curve%
True Positive Rate
False Positive Rate
NB_AUC=0.68
0.0
0.2
0.4
0.6
0.8
1.0

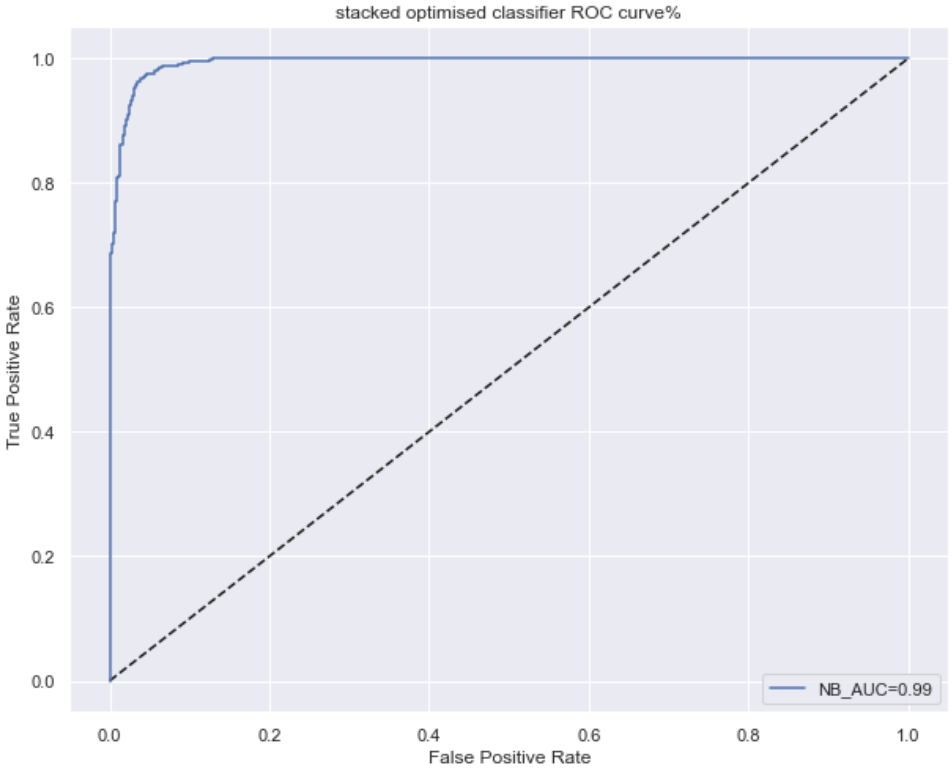

XGBoost classifier ROC curve%
True Positive Rate
False Positive Rate
NB_AUC=0.93
0.0
0.2
0.4
0.6
0.8
1.0


stacked classifier ROC curve%
True Positive Rate
False Positive Rate
NB_AUC=0.94
0.0
0.2
0.4
0.6
0.8
1.0


random forest optimised classifier ROC curve%
True Positive Rate
False Positive Rate
NB_AUC=0.95
0.0
0.2
0.4
0.6
0.8
1.0

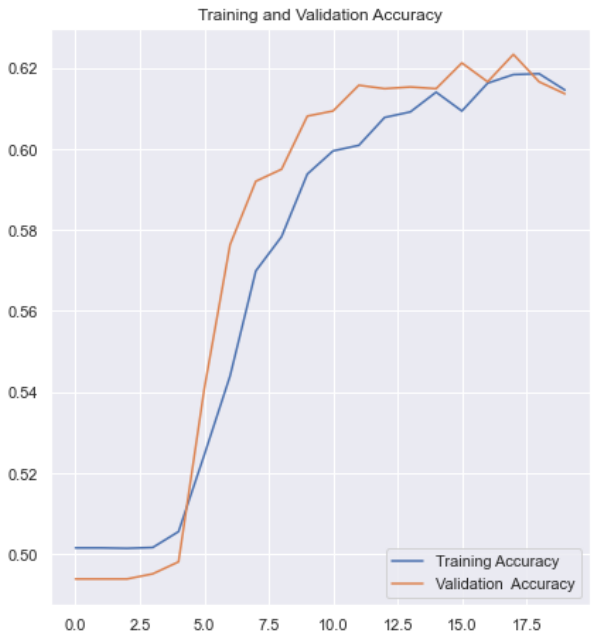

xgboost optimised classifier ROC curve%
True Positive Rate
False Positive Rate
NB_AUC=0.99
0.0
0.2
0.4
0.6
0.8
1.0

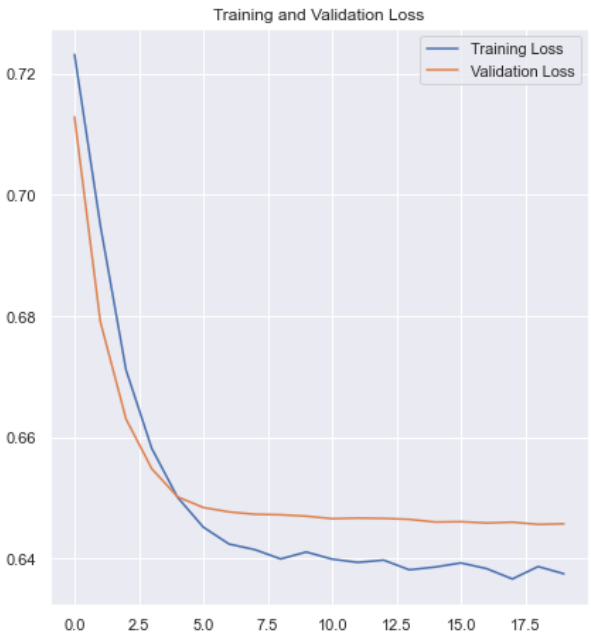

kneighbors optimised classifier ROC curve%
True Positive Rate
False Positive Rate
NB_AUC=0.94
0.0
0.2
0.4
0.6
0.8
1.0


stacked optimised classifier ROC curve%
True Positive Rate
False Positive Rate
NB_AUC=0.99
0.0
0.2
0.4
0.6
0.8
1.0

7.2.3 Artificial Neural Network (ANN) using back propagation

- Using K-Best Features





- Using random forest best predicting features for sold or not-sold