2022

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

Table of Contents

Table of Contents i
List of Figures iii
List of Tables iii
Definition of terms iv
Abstract iv
1.0.0 Introduction 1
1.1.0 Aims and Objectives 1
1.2.0 Problem statement 1
1.3.0 Scope of work 1
1.4.0 Project outline 2
2.0.0 Background study 2
3.0.0 Methodology 3
3.1.0 Flowchart representation of methodology 3
3.2.0 Data collection 3
3.3.0 Data cleaning 4
3.4.0 Data mining 6
3.5.0 Feature extraction/engineering 6
3.6.0 Performance metrics selection for regression and classification tasks 7
3.6.1 Performance metrics selection for regression task 7
3.6.2 Performance metrics selection for classification task 7
3.7.0 Training classifier and regressor models 7
3.8.0 Hyperparameter Optimisation 8
4.0.0 Results 8
4.1.0 Price prediction using regressor models 8
4.2.0 Sold prediction using classifier models 10
5.0.0 Discussion, conclusion, recommendation and further study 13
5.1.0 Business related 13
5.2.0 Technical Related 14
5.3.0 Research observations, recommendations and further study 15
6.0.0 Bibliography 17
7.0.0 Appendixes 19
7.1.0 Appendix A 19

7.2.0 Appendix B 24
7.2.1 KBest Trained Regressor models 24
7.2.2 Random Forest Features Trained Regressor models 28
7.2.3 Artificial Neural Network (ANN) using back propagation 32

List of Figures

Figure 1: Unclean torque column before cleaning. 5
Figure 2: Torque column containing unique nm values after cleaning. 5
Figure 3: Torque column containing unique rpm values after cleaning. 6
Figure 4: Picture showing correlation plot between columns with annotations. 6
Figure 5: Distribution of region performance of sold and not-sold. 14
Figure 6: XGBoost regressor price prediction for test and training data. 15
Figure 7: Linear Regressor price prediction for test and training data 16

List of Tables

Table 1: Comparison of price predicting top four feature selections. 9
Table 2: Performance of regressor models trained with top four price predicting features from KBest Selector. 9
Table 3: Performance of regressor model trained with random forest price predicting top four features. 10
Table 4: Comparison of sold predicting top five feature selections. 11
Table 5: Performance of classifier models trained with top five KBest selected sold predicting features. 11
Table 6: Confusion matrix performance report of models trained with top five KBest selected sold predicting features and 2363 test observations. 12
Table 7: Performance of classifier model trained with random forest sold predicting top five features. 12
Table 8: Confusion matrix performance report of models trained with random forest sold predicting top five features and 2363 test observations. 13

Definition of terms

- MAE - Mean Absolute Error
- MSE - Mean Squared Error
- 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
- USD - United States Dollar

Abstract

This project is about improving the used cars market through the use of machine learning techniques. The used cars market described by Akerlof as a lemons market often experience asymmetric information between dealers and consumers. This has become an advantage to some dealers with more knowledge and a disadvantage to the consumers as these dealers rack up huge profits from sale of low-quality cars because consumers and sometimes the new dealers in the market are not always aware of the factors that influence the prices of used cars. Research has shown that the mileage, reliability (cost of maintenance), extra features like higher engine power are notable factors that influence the price of used cars while biasing the mind of consumers whether or not to opt for a used car. This research is centered around finding the best machine learning features and models that best predicts the price of a used car and if it would be sold or not. The data for this project was gotten from Kaggle.com. Exploratory data analysis (EDA) packages like pandas, numpy, matplotlib and seaborn were used to analyse the available data and provide insights. Machine learning regression algorithms were used to build price prediction models and classification algorithms to build binary classification models. Finally, the two groups of models built were compared using preferred metrics so as to decide the best performing algorithm in each group. After the research, XGBoost regressor trained with KBest price predicting features proved to be the best model for price prediction with an R^2 score of 98.7% for training data and 98.0% for test data. It also had an rRMSE score of 11.1%. The linear regressor model was the least performing model with an R^2 score of 62.4% for training data and 64.0% for test data. It also had an rRMSE score of 57.2%. The best price predicting features were torque in newton meter, max power, engine capacity and year. The best model for predicting sold or not-sold binary classification task was the stacking classifier trained with random forest sold predicting features with an accuracy score of 98.9% for training data and 96.0% for the test data with only 100 mislabeled observations out of 2363 observations. The least performing model for the binary classification task was the Gaussian Naïve Bayes (GNB) trained with KBest sold predicting features with an accuracy of 63.4% for the training data and 62.6% for the test data. It had 885 mislabeled observations out of the 2363 observations.

1.0.0 Introduction

Vehicle lifetime has had significant improvement 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 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 and the market covers the sale of private and remarketed cars. A Private sale refers to sales 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 or car rental companies (Duvan & Ozturkcan, 2009). The used cars market is described as a lemons market because of the asymmetric information between dealers and consumers (Akerlof, 1978). This has been of great benefit to dealers with more knowledge as they often rack up huge profits from sale of low-quality cars because consumers and sometimes the new dealers do not have information regarding factors that influence the prices of used cars 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 consumers and new dealers by providing knowledge about market facts (Duvan & Ozturkcan, 2009). Research has shown that the mileage, reliability (cost of maintenance), extra features like higher engine power as well as 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; Phlips, 1983; Scherer, 1996). This project will also significantly help in 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.

1.1.0 Aims and Objectives

i. Finding the features that best predicts used car prices and sale probability.
ii. Building machine learning models to predict used car prices and sale probability.
iii. 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;

- **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?

- **Technical related questions:**

i. What model would best and least predict car prices and with what features?
ii. Which model best and least 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, using EDA and data mining packages like pandas, numpy, matplotlib and seaborn to analyse the available data and provide insights. Thereafter, applying feature extraction/engineering, machine learning and data processing techniques to build price prediction models using regression algorithms and classification algorithms to build binary classification

models. I will be trying out over seven different models on each machine learning task to compare the performance of the models using preferred metrics so as to decide the best and least performing model in each group. Hyperparameter optimisation will then be performed to improve and optimise model performance. Finally, results and findings will be presented and discussed.

1.4.0 Project outline

The outline for this project is as follows;

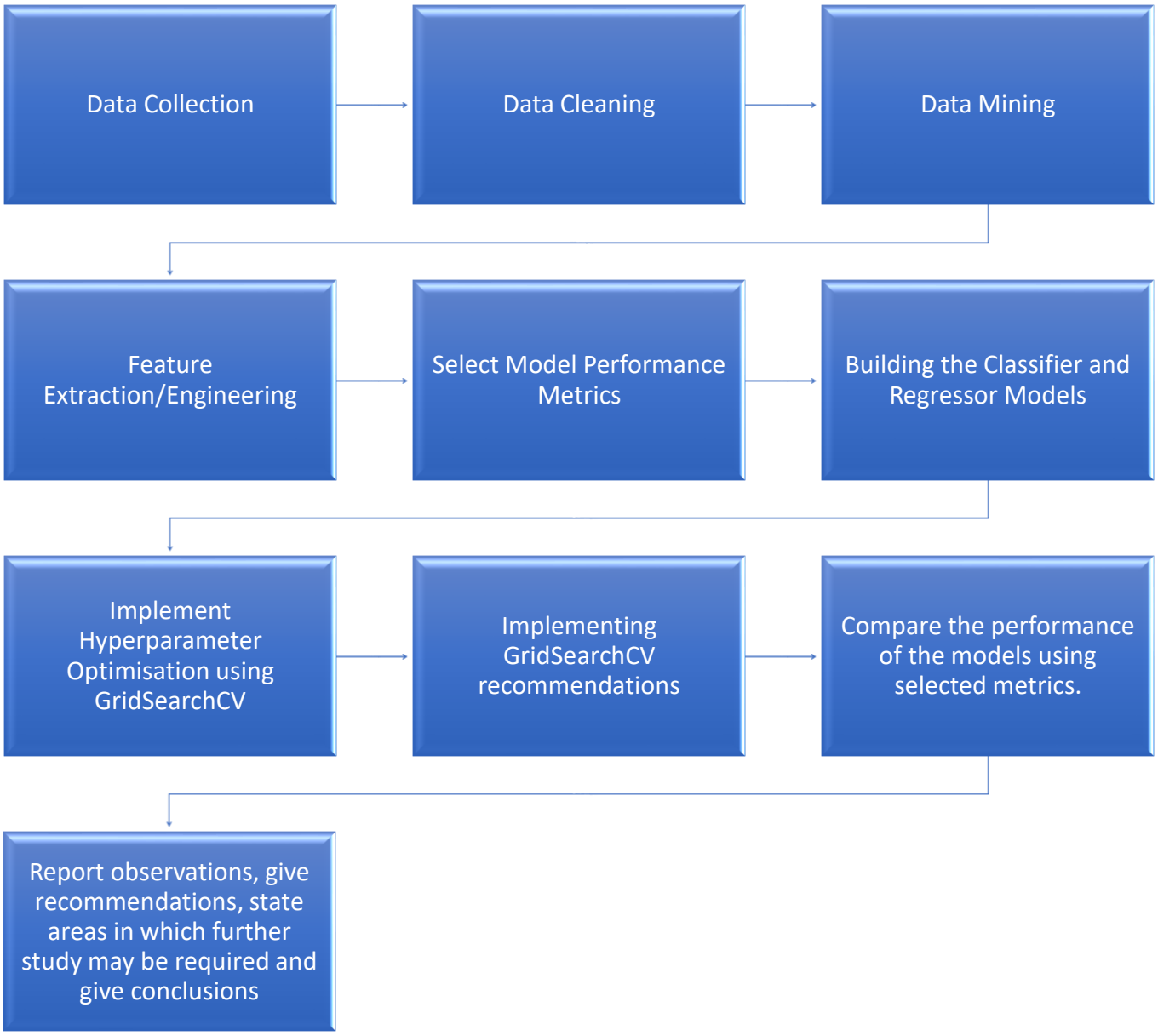
i. The project would be Introduced with an overview while stating the importance and scope.
ii. Thereafter, similar projects already completed will be highlighted stating their project aims, methodology and results.
iii. I would then discuss how I would achieve my aims and objectives.
iv. Results obtained will then be carefully outlined
v. Finally, I would discuss my findings, give recommendations and highlight areas that would require further-study.

2.0.0 Background study

In recent time a lot of research has been conducted for the prediction of used car prices using machine learning techniques and regressor models. Data for such projects have been sourced from Kaggle, company databases, web scrapping etc. Some of the regressor models built to help predict the car prices include random forest, decision tree, linear regression, gradient boosting and ensemble regressors. It is important to note that these models have been extensively used by researchers to validate the best and least price predicting model while comparing model performance using preferred metrics. They have also tried to answer the question, if prices can actually be predicted and to what degree of relative accuracy. Asghar et al. in 2021, concluded that random forest regressor was the best regressor for price prediction after getting a 95.82% training accuracy and 83.63% test accuracy. In 2017, Voß & Lessmann concluded in their research that ensemble regressors performed best with an overall best MAE of 3.97 proving earlier research by Caruana et al. in 2006 right. They also concluded that linear regression methods predict significantly less accurate than ensemble regression methods. N. Monburinon et al. in 2018, concluded that the gradient boosting performed the best in price prediction with an MSE of 0.28 followed by random forest with 0.35 while linear regressor had the least performance with an MSE of 0.55. Also, researchers have not left out model performance in classification tasks by classifier algorithms and deep learning techniques. V. Bahel et al. in 2020, using breast cancer and titanic survival datasets compared the performance of various binary classification algorithms using preferred metrics. Their correlation result showed that the breast cancer dataset had better correlation than the titanic dataset. They also concluded that Logistic Regression and AdaBoost models performed better with features with high correlation to the target while 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. Also, the decision tree and random forest performed decently with both datasets while the KNN worked well when used with fewer features but as the number of features increases the performance of the model decreases. Another research by G. Gui et al. in 2020, showed that LSTM model experienced overfitting when they attempted to use recurrent learning technique for their classification project but the random forest classifier performed the best with a test accuracy score of 90.2%.

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 dataset 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. The torque unclean data was a combination of two column which is torque in n/m and torque in rpm hence 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 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. Also, the selling price column was in Indian rupees and our focus area the United States hence I had covert the rupees to the dollar with the exchange rate as at 18th of July, 2022 which was at 1 rupee to 0.012 USD (Xe Currency Converter, 2022).

```
'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.  , 250.  , 140.  , 113.75,  59.  , 170.  , 160.  , 248.  ,
        78.  ,  84.  , 115.  , 200.  ,  62.  , 219.7 , 114.  ,  69.  ,
       172.5 , 114.7 ,  60.  ,  90.  , 151.  , 104.  , 320.  , 145.  ,
       146.  , 343.  , 400.  , 138.  , 360.  , 380.  , 173.  , 111.7 ,
       219.6 , 112.  , 130.  , 205.  , 280.  ,  99.04,  77.  , 110.  ,
       153.  , 113.7 , 113.  , 101.  , 290.  , 120.  ,  96.  , 135.  ,
       259.8 , 259.9 ,  91.  ,  96.1 , 109.  , 202.  , 430.  , 347.  ,
       382.  , 620.  , 500.  , 550.  , 490.  , 177.5 , 300.  , 260.  ,
       213.  , 224.  , 640.  ,  95.  ,  71.  , 117.  ,  72.  , 134.  ,
       150.  , 340.  , 240.  , 330.  , 111.8 , 135.4 , 190.25, 247.  ,
       223.  , 180.  , 195.  , 154.9 , 114.73, 108.  , 190.24, 420.  ,
       100.  ,  51.  , 132.  , 350.  , 218.  ,  85.  ,  74.5 , 180.4 ,
       230.  , 219.66, 245.  , 204.  , 125.  , 172.  , 102.  , 106.5 ,
       108.5 , 144.15,  99.  , 142.5 , 196.  , 209.  , 220.  , 171.  ,
       277.5 , 215.  , 263.7 ,  94.14, 789.  , 259.87, 436.39, 182.5 ,
        90.3 , 305.  , 540.  , 327.  , 450.  , 148.  , 116.  , 222.  ,
       135.3 ,  98.  , 402.  ,  99.07, 210.  , 197.  , 136.3 , 470.  ,
       142.  , 123.  , 510.  , 292.  , 136.  , 228.  , 149.  , 187.  ,
       225.  , 259.88, 580.  , 127.  , 132.3 , 124.6 , 219.9 , 420.7 ,
       424.  ,  99.8 , 321.  , 619.  , 560.  , 600.  , 285.  , 226.  ,
       155.  , 103.  , 175.  ,  72.9 ,  57.  , 128.  , 131.  , 185.  ,
       176.  , 121.  , 106.  , 113.8 ,  83.  , 124.5 , 171.6 ,  88.4 ,
       355.  , 119.  , 410.  , 174.  ,  99.1 , 385.  ,  53.  , 124.  ,
       159.8 , 333.  , 480.  , 250.06, 436.4 ])
```

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. I used available commands from pandas and numpy for the data mining and analysis while using matplotlib and seaborn commands for data visualisations.

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 as shown in figure 4.

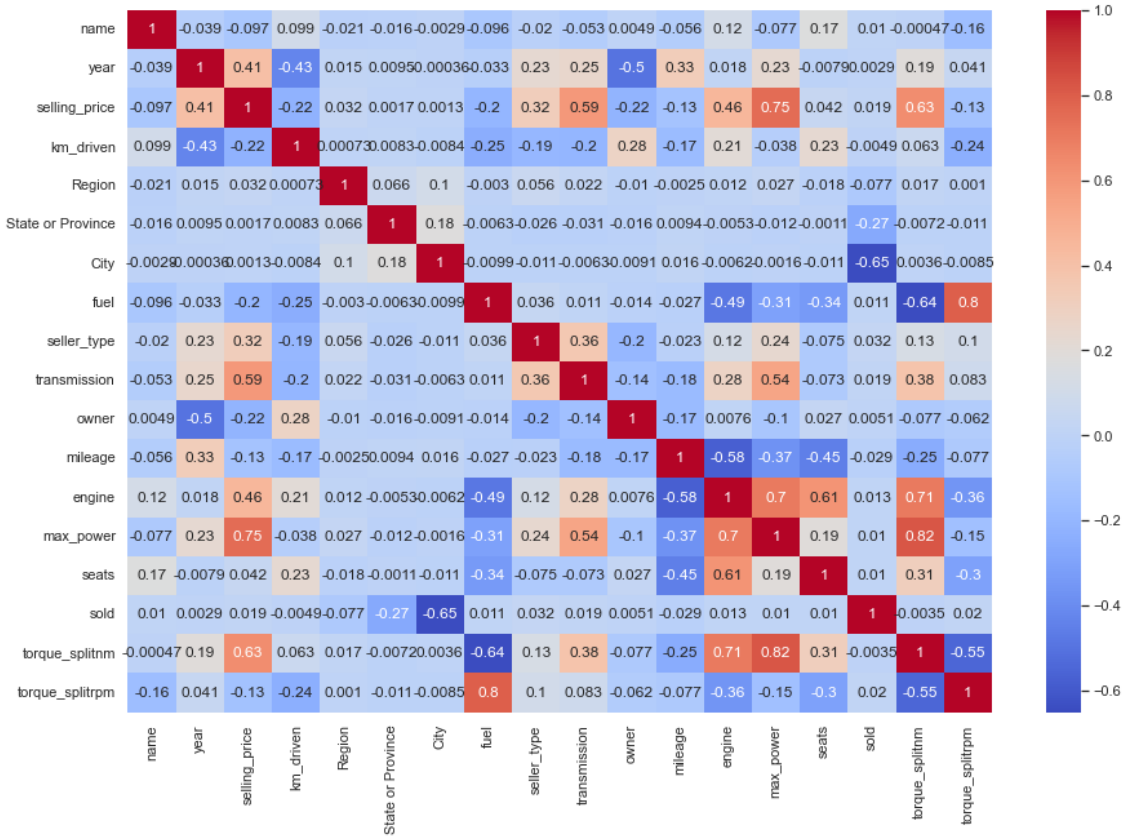


Figure 4: Picture showing correlation plot between columns with annotations.

3.6.0 Performance metrics selection for regression and classification tasks

3.6.1 Performance metrics selection for regression task

i. R^2 Score
ii. Mean Absolute error
iii. Mean Squared Error
iv. Root Mean Square Error
v. Relative Root Mean Square Error
vi. Cross-validation

(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 classification task

i. Confusion Matrix
ii. Accuracy Score
iii. F1 Score
(Raschka, 2014).

3.7.0 Training classifier and regressor models

For the regressor models, the selected features and target consisting of 7,906 observations were split into training and test data. 5,929 observations for the training data making up 80% of the entire dataset and 1,977 observations for the test data making up 20% of the entire dataset. 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 are biased by the magnitude of the training features.

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 the resample technique to up sample the available data and got 11,812 observations in total. I then split the observations into training and test data. My training data for my classification task was 9,449 observations making up 80% of the entire resampled dataset and 2,363 observations for my test data making up 20% of the entire resampled dataset. I then scaled my data using standard scaler by fitting and transforming my training data and then transforming 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 hyperparameter optimisation using GridSearchCV on Random Forest algorithm with the first 2000 observations in the dataset so as to explore the feature_importances_ attribute. This is to enable me compare listed hierarchy of random forest's feature_importances_ and best training features predicted by my KBest 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

1. **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

2. **Table 1: Comparison of price predicting top four feature selections.**

S/N	KBest Price Predict Features	Random Forest Price Predict Features
1.	Torque_splitnm	max_power
2.	max_power	Year
3.	engine	Km_driven
4.	year	torque_splitnm

3. **Model performance report for price prediction using regressors**

Table 2: Performance of regressor models trained with top four price predicting features from KBest 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	98.7	97.9	853.2	2178534.2	1476.0	11.5	98.8	97.0
2	Linear Regressor	62.4	64.0	3430.9	37644355.3	6135.5	57.2	62.9	62.8
3	Polynomial Regressor	85.3	86.2	2090.4	14484144.6	3805.8	32.3	85.7	84.0
4	Ransac Regressor	93.0	92.4	1096.8	7920226.4	2814.3	23.0	94.0	93.2
5	SVM Regressor	93.1	87.7	1387.5	12347613.2	3513.9	28.6	94.3	88.6
6	XGBoost Regressor	98.7	98.0	860.6	2049091.7	1431.5	11.1	98.8	97.1
7	Stacked Regressor	98.5	97.6	881.6	2551404.9	1597.3	12.4	98.4	96.9
8	KNN Regressor	98.1	96.9	1029.2	3281679.7	1811.5	14.1	98.1	93.1

Table 3: Performance of regressor model trained with random forest price predicting top four features.

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	98.9	97.6	848.0	2484663.6	1576.3	12.3	99.0	96.8
2	Linear Regressor	63.5	63.1	3453.7	38557269.4	6209.5	57.2	63.6	63.2
3	Polynomial Regressor	85.3	70.3	2237.1	31041586.2	5571.5	44.4	85.4	78.7
4	Ransac Regressor	94.7	94.6	982.4	5629162.1	2372.6	18.9	95.4	94.2
5	SVM Regressor	93.1	85.9	1507.4	14135702.1	3759.7	30.3	94.2	86.1
6	XGBoost Regressor	99.0	97.9	824.6	2147006.0	1465.3	11.4	99.1	97.1
7	Stacked Regressor	98.7	97.9	841.8	2211288.5	1487.0	11.5	98.8	96.9
8	KNN Optimised Regressor	99.8	85.3	1674.0	15393506.8	3923.5	31.1	99.8	84.6

4.2.0 Sold prediction using classifier models

1. 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

2. Table 4: Comparison of sold predicting top five feature selections.

S/N	KBest Sold Predict Features	Random Forest Sold Predict Features
1.	torque_splitrpm	Fuel
2.	selling_price	City
3.	engine	km_driven
4.	State or Province	Engine
5.	seats	Region

3. Model performance report for sold prediction using classifiers and artificial neural networks

Table 5: Performance of classifier models trained with top five KBest selected sold predicting features.

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	63.4	62.6	1478	885	53	69	63.4	63.3
2	Decision Tree Classifier	97.7	84.3	1991	372	83	86	97.9	81.1
3	Random Forest Classifier	97.7	86.4	2042	321	85	87	97.9	83.3
4	Logistic Regression	64.7	63.1	1492	871	56	68	63.9	63.9
5	SVM Classifier	65.1	63.1	1492	871	56	68	95.5	86.2
6	XGBoost Classifier	97.7	85.6	2023	340	84	87	97.9	83.2
7	Stacked Classifier	95.5	87.0	2056	307	87	87	93.2	85.5
8	KNN Classifier	96.9	83.9	1983	380	82	85	97.4	79.0

9	Artificial Neural Network (ANN)	60.8	60.9	-	-	-	-	-	-

Table 6: Confusion matrix performance report of models trained with top five KBest selected sold predicting features and 2363 test observations.

S/N	Model	True Positives	False Positives	False Negatives	True Negatives
1	Gaussian Naive Bayes Classifier	499	668	217	979
2	Decision Tree Classifier	884	283	89	1107
3	Random Forest Classifier	922	245	76	1120
4	Logistic Regression	554	613	258	938
5	SVM Classifier	549	618	253	943
6	XGBoost Classifier	909	258	82	1114
7	Stacked Classifier	1015	152	155	1041
8	KNN Classifier	882	285	95	1101

Table 7: Performance of classifier model trained with random forest sold predicting top five features.

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	96.0	94.2	2225	138	94	94	96.3	94.2
2.	XGBoost classifier	100	96.2	2272	91	96	96	99.9	96.0
3.	Stacked classifier	98.9	96.0	2263	100	96	96	99.1	95.5

4.	KNN optimised classifier	100	94.1	2225	138	94	94	99.9	92.4
5.	Artificial Neural Network (ANN)	93.3	92.2	-	-	-	-	-	-

Table 8: Confusion matrix performance report of models trained with random forest sold predicting top five features and 2363 test observations.

S/N	Model	True Positives	False Positives	False Negatives	True Negatives
1	Random Forest optimised classifier	1041	137	1	1184
2	XGBoost optimised classifier	1098	80	11	1174
3	Stacked optimised classifier	1118	60	40	1145
4	KNN optimised classifier	1074	104	34	1151

5.0.0 Discussion, conclusion, recommendation and further study

5.1.0 Business related

As shown from feature engineering, the top four features that influences the prices of a used car are the torque rating in newton per meter, maximum power, engine capacity and year of manufacture while the top five features that influences bias of 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).

An overview of the dataset revealed that it consists of 25.3% of sold cars and 74.7% of not-sold cars. Considering only car names with over 50 observations in the dataset with emphasis on sold to not-sold ratio, the top three names sold overall are; Nissan with 40.74%, Mercedes with 29.63% and Jaguar with 29.58%. The top two fuel types sold are petrol vehicles with 26.56% and diesel vehicles with 24.47%. The eastern region has the highest sale possibility 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 figure 5 below.

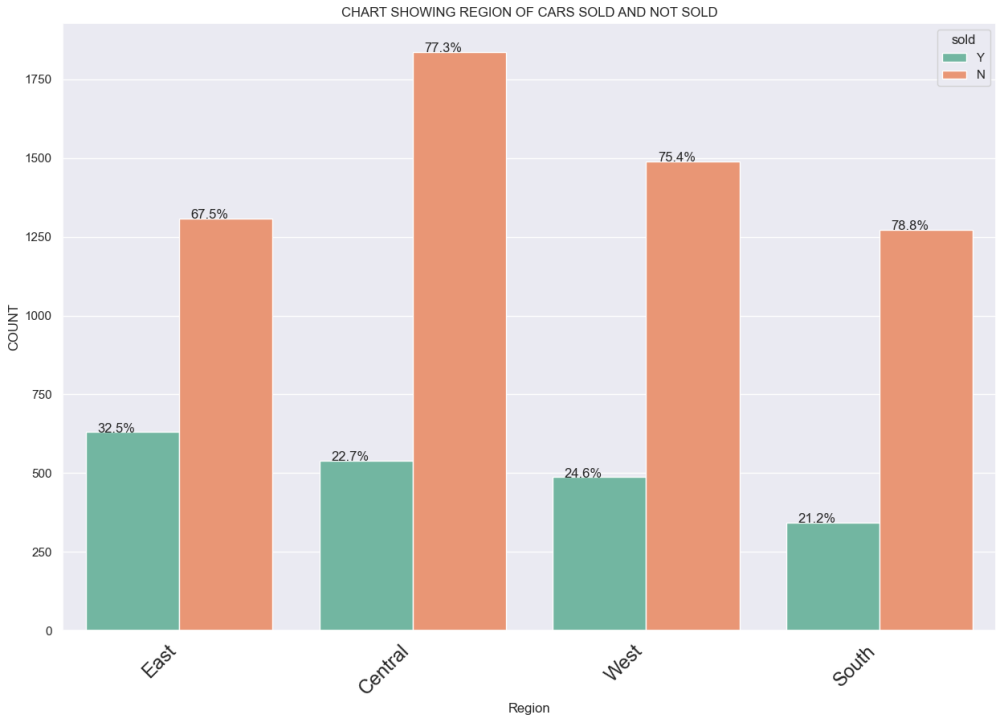


Figure 5: Distribution of region performance of sold and not-sold.

Cars from Trustmark dealer sells more than cars from other sellers with a sold percentage of 33.9% and not-sold percentage of 66.1%.

For further business-related insights, see Appendix A.

5.2.0 Technical Related

The best performing model for price prediction is the XGBoost regressor trained with KBest features. It achieved a training accuracy of 98.7%, test accuracy of 98.0%, MAE of 860.6, RMSE of 1431.5 and rRMSE of 11.1% which is a near excellent model going by the rRMSE performance score. There were other models that performed considerable well like the random forest trained by KBest features, the XGBoost and stacking regressor trained with predicted features by the random forest regressor. Though these models had better MAE than the best performing model, it had better training and test accuracy, RMSE and rRMSE than all the rest models as shown in table's 2 and 3. The model with least performance was the multi linear regressor. It performed the least in both cases, maxing a training accuracy of 63.5, test accuracy of 63.1 and an rRMSE of 57.2% at best. The KBest trained regressors had better performance when averaged across eight regressors with an rRMSE average of 23.8% than regressors trained with random forest predicted features that averaged an rRMSE of 27.1%. For the sale prediction models involving the use of binary classification, the stacking classifier trained with the random forest predicted features performed best with a training accuracy of 98.9%, and test accuracy of 96.0% while the least performing model was the gaussian naive bayes model with a training accuracy of 63.4%, test accuracy of 62.6% and having the greatest number of mislabeled points at 885 as shown in table 5 and 7. The XGBoost classifier trained with random forest selected best features also performed well with a training accuracy of 100% and a test accuracy of 96.2%. However, I preferred the stacked classifier over the XGBoost classifier because it had the best ratio balance between false positives and false negatives as shown in table 8. Based on this

outcomes, it can be said that the linear regressor struggled to predict the prices probably due to low specificity and high heterogeneity of the data which is a known problem for this model as shown in figure 7 while models using the tree based approach performed significantly well with both features as can be seen with the XGBoost regressor being the best regressor model trained as shown in figure 6 (Voß & Lessmann, 2017). Also, the naïve bayes models are known to perform better when there is no correlation between features but existence of correlation in the selected features as shown in figure 4 must have resulted in its poor performance (V. Bahel et al., 2020). In summary, the KBest selected features were the best features for the regression task while the random forest selected features were the best features for the classification task as improved accuracy and model performance can be seen when the training features were changed from the KBest to that of the random forest as shown in table's 5, 6, 7 and 8.

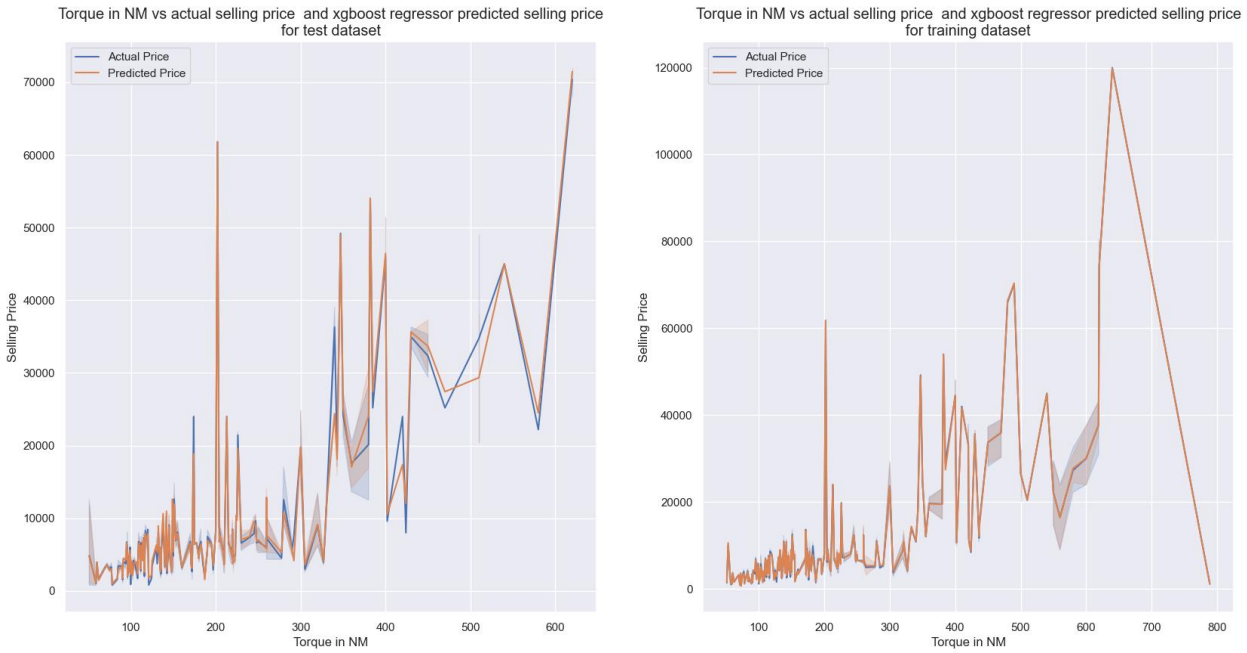


Figure 6: XGBoost regressor price prediction for test and training data.

5.3.0 Research observations, recommendations and further study

i. The stacking regressor dropped in accuracy scores and performance rating when I removed the multi linear regressor from the group due to its poor individual performance and the stacking regressor's high performance was restored when I returned it to the group.

ii. Artificial neural network trained with KBest features struggled in performance but experience significant improvement with the features predicted by random forest.

iii. Models like decision tree, random forest and XGBoost using tree-based approach performed well both for classification and regression tasks (Jiang et al., 2020).

iv. The KNN which uses an instance base learning approach did well with predicting training data but could not predict the testing data with same very high training accuracy as this is obviously due to the high heterogeneity of the data. Also, KNN models performed better with less features thus validating its limitation due to curse of dimensionality (V. Bahel et al., 2020; Protasov & Khan, 2021).

v. SVM classifier performed much better during cross-validation than during normal training and testing. This could be an area of interest for further study to find out what happened and why.

vi. I would recommend further research to explore if the level of cleanliness of a used car can influence its price.

vii. For performance visualisations for the regressor models and neural networks, see Appendix B.

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

In conclusion XGBoost regressor trained with KBest features was my best price predicting model and stacking classifier trained with random forest selected features 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.

Jiang, M., Liu, J., Zhang, L. & Liu, C. (2020) An improved stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms. *Physica A: Statistical Mechanics and its Applications,* 541 122272.

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.

Protasov, S. & Khan, A. M. (2021) Using proximity graph cut for fast and robust instance-based classification in large datasets. *Complexity,* 2021.

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.

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.*

Xe Currency Converter (2022) . Available online: https://www.xe.com/currencyconverter/convert/?Amount=1&From=USD&To=INR [Accessed Jul 18 2022].

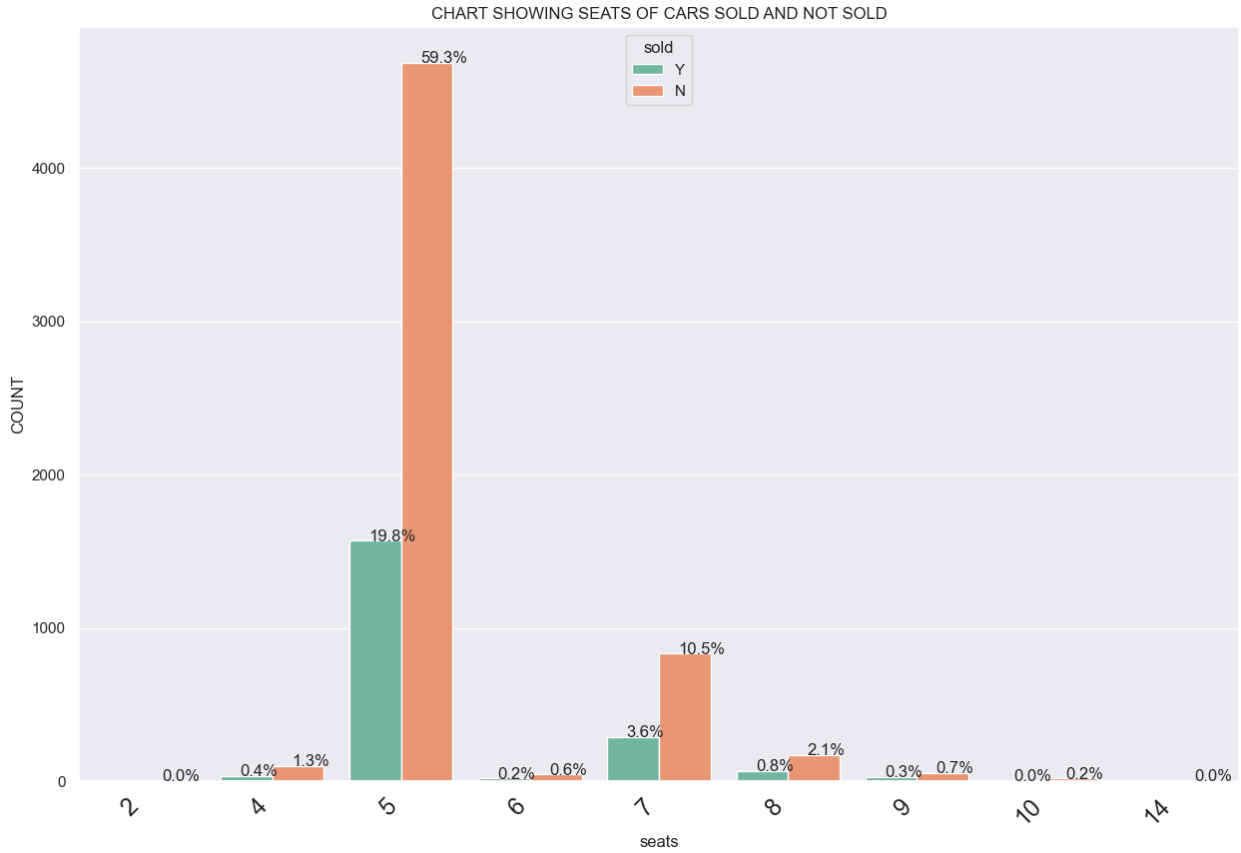
Xu, Z., Li, W., Li, Y., Shen, X. & Ruan, H. (2019) Estimation of secondary forest parameters by integrating image and point cloud-based metrics acquired from unmanned aerial vehicle. *Journal of Applied Remote Sensing,* 14 (2), 022204.

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

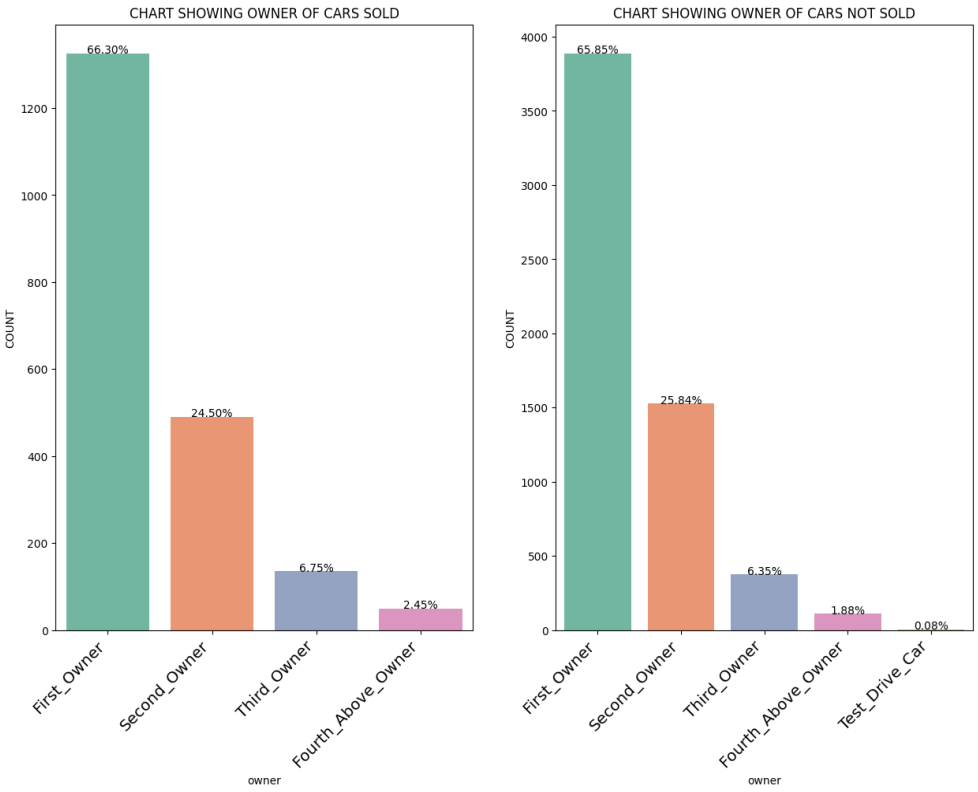
7.1.0 Appendix A

- Majority of the cars available are 5-seater cars with a sold to not-sold ratio that is dominated by not-sold and maintaining a fairly constant distribution across cars with other number of seats as shown in the chart below.

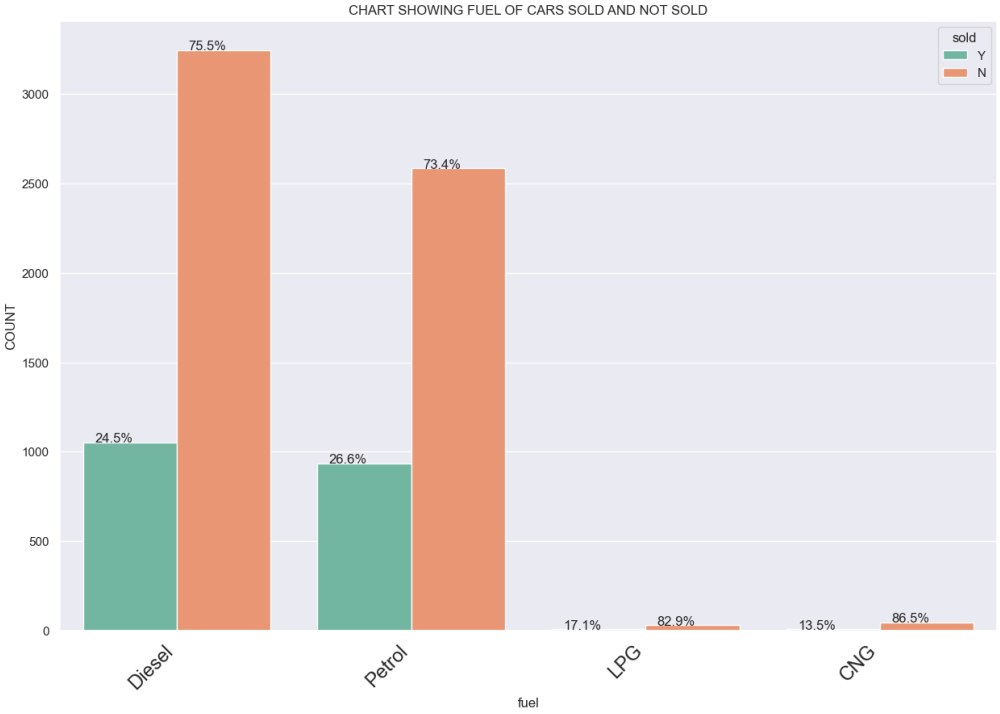


Picture showing distribution of sold and not-sold seats

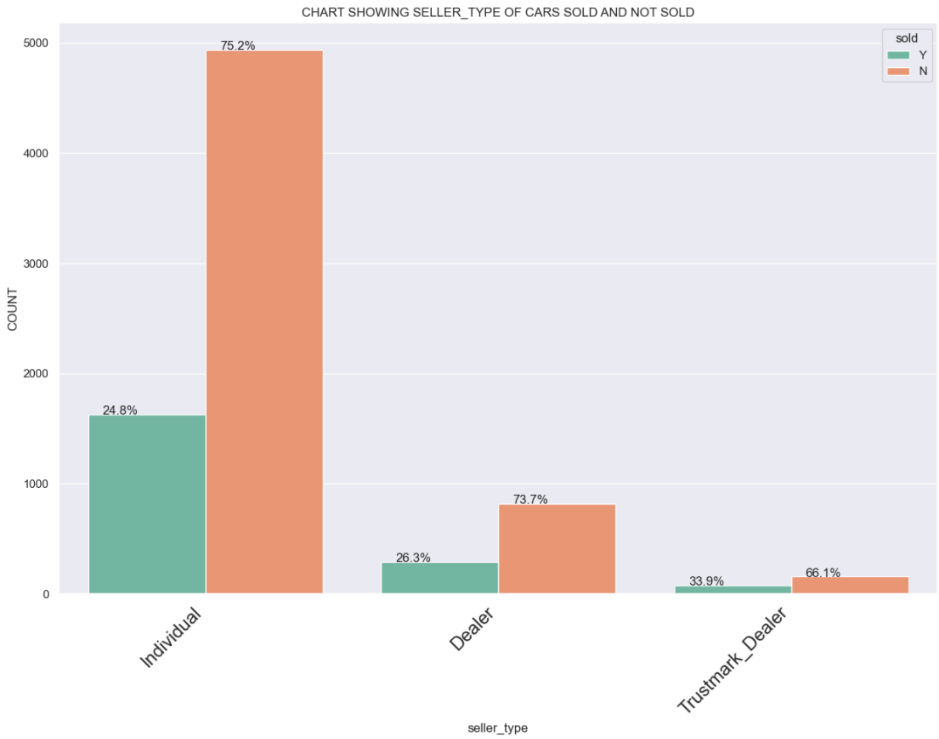
- The average selling price of an automatic car costs around 23802.2 dollars while that of manual cars costs around 5499.07 dollars. Automatic cars could cost up to four times the manual car.
- For the owner's column, the fourth and above owner column has more sold than not-sold ratio than any other owner type and as you move back from the fourth and above owner to the first owner, a decline is seen between the ratio of sold and not sold as shown below.



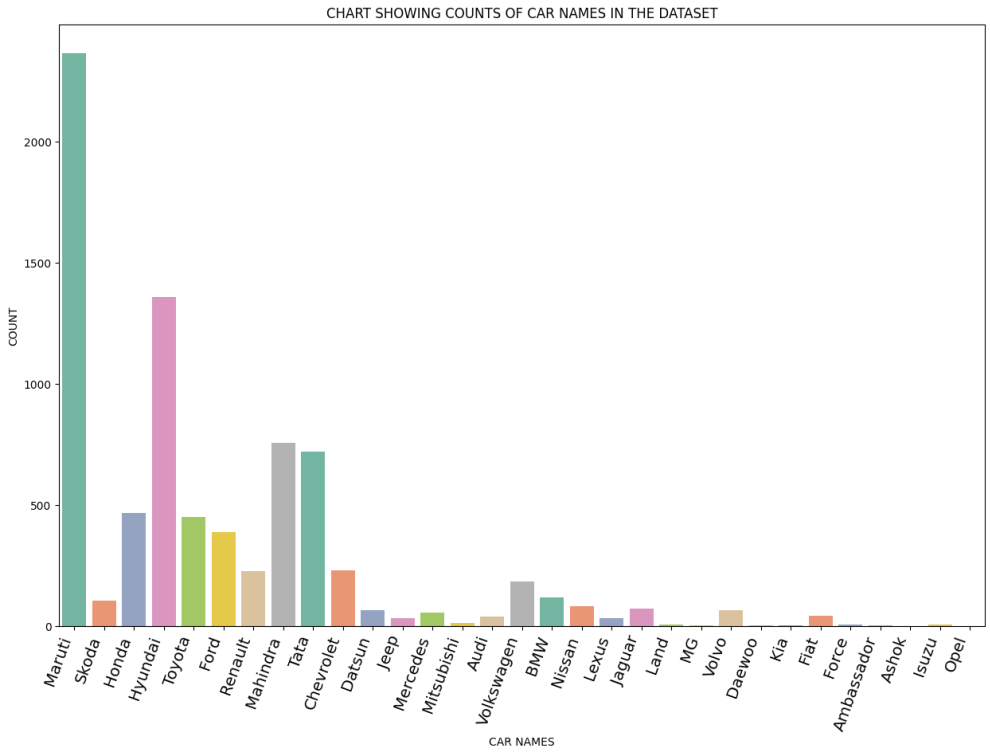
Picture showing distribution of owner column



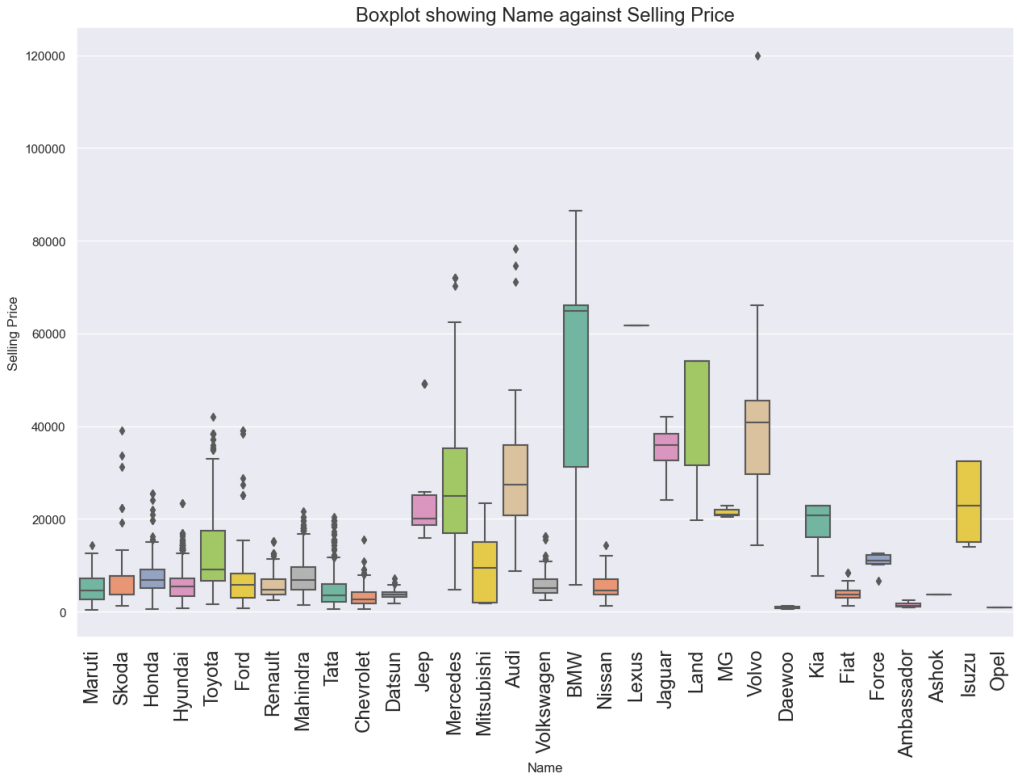
Picture showing fuel type sold to not-sold ratio for different fuel type.



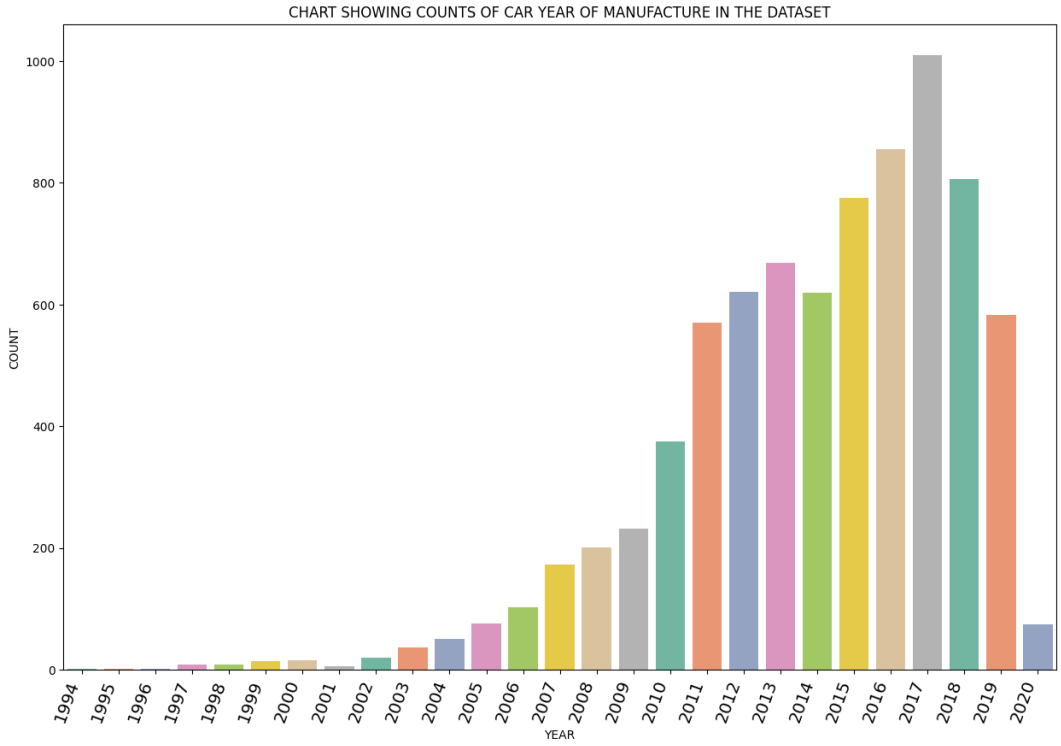
Picture showing distribution of seller type with respect to sold and not-sold.



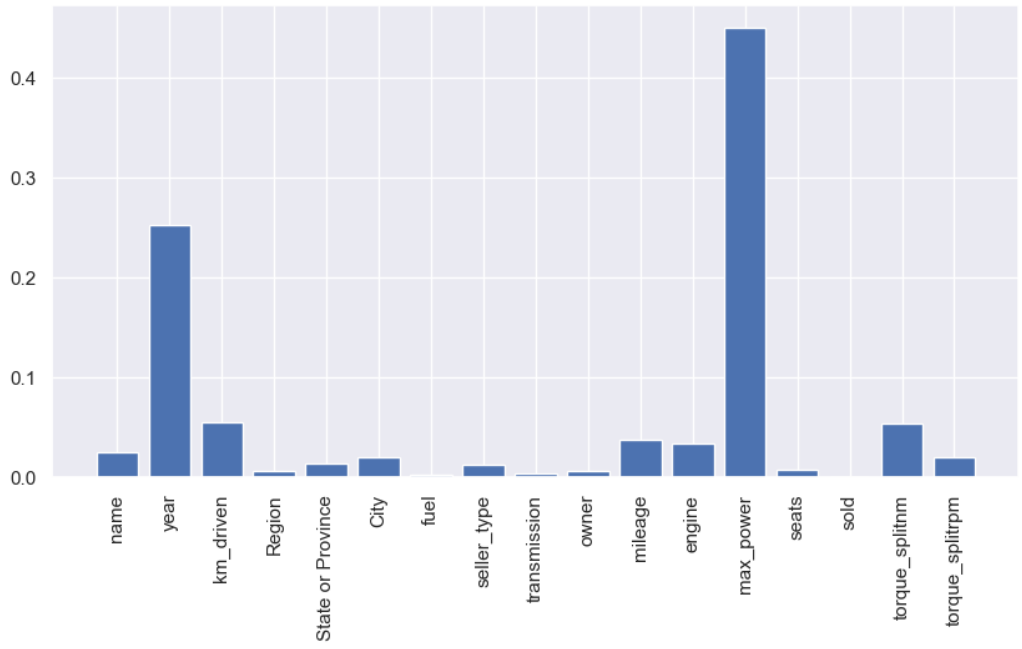
Picture showing distribution of car names vs counts



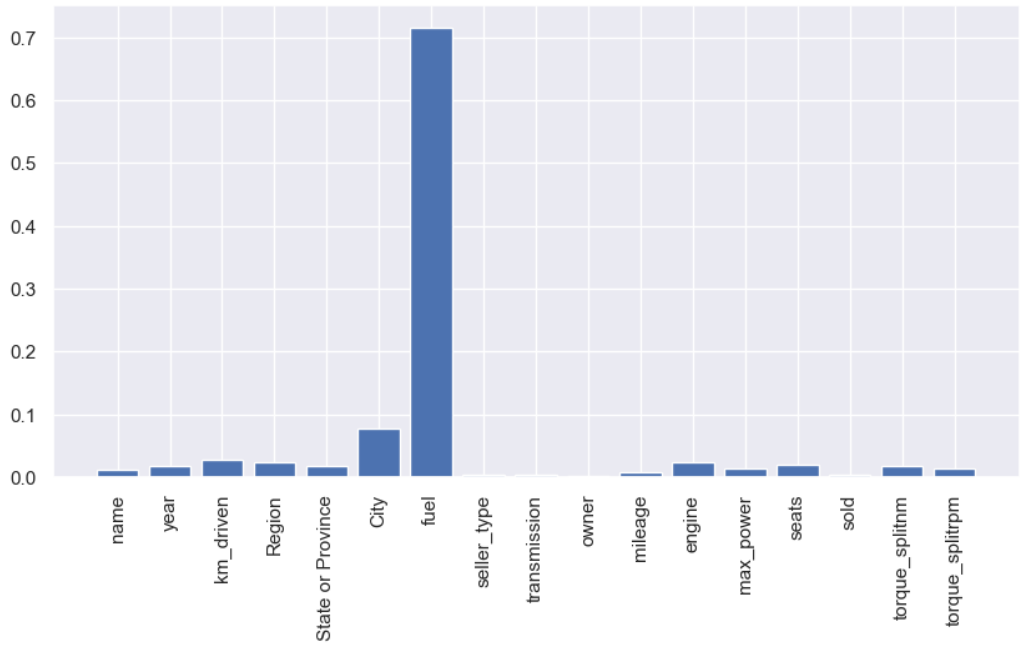
Picture showing box plot of car names vs range of selling price and average selling price



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



Plot showing distribution of random forest feature selection for price prediction during hyperparameter optimisation.

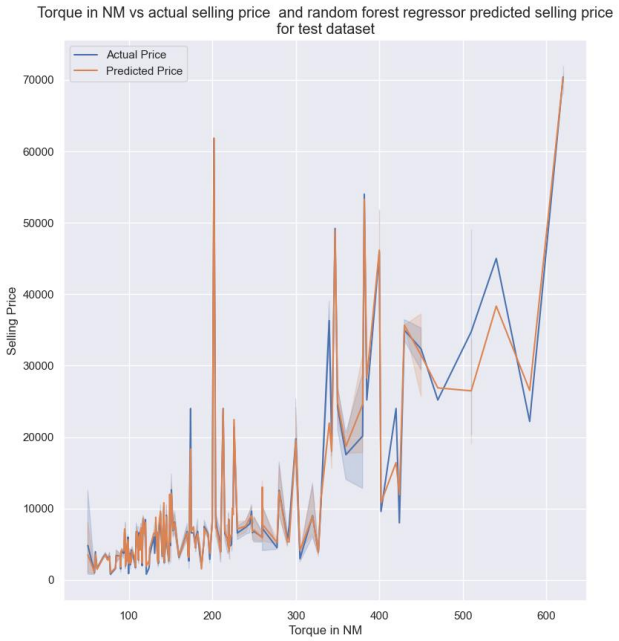


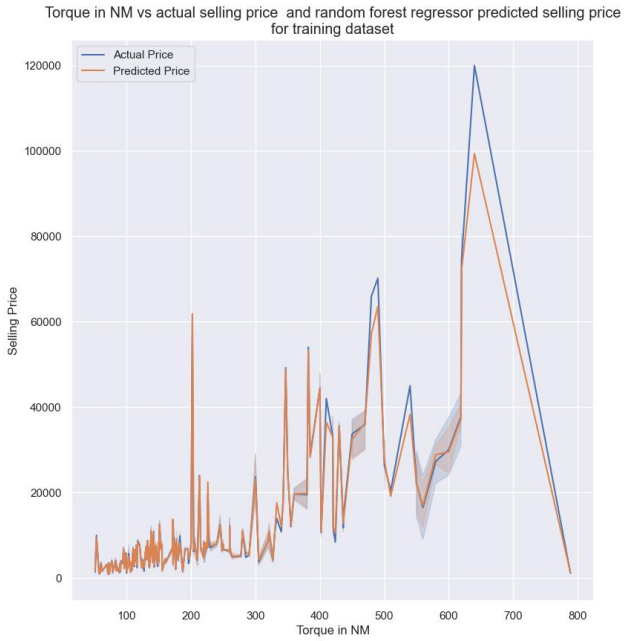
Plot showing distribution of random forest feature selection for sold prediction during hyperparameter optimisation.

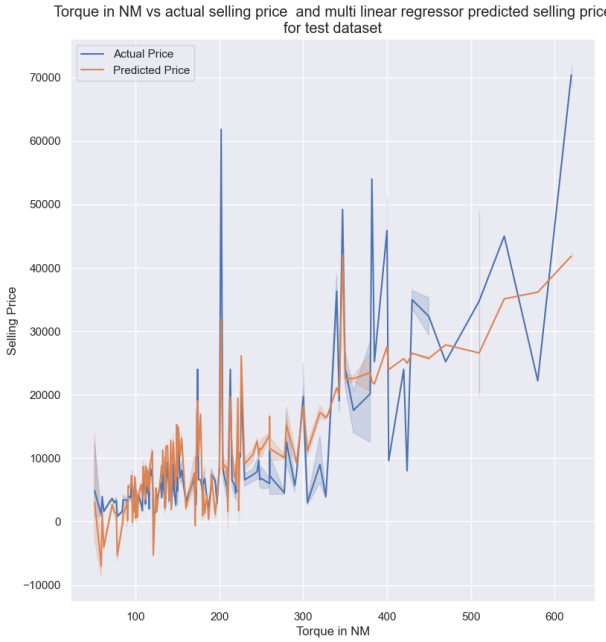
7.2.0 Appendix B

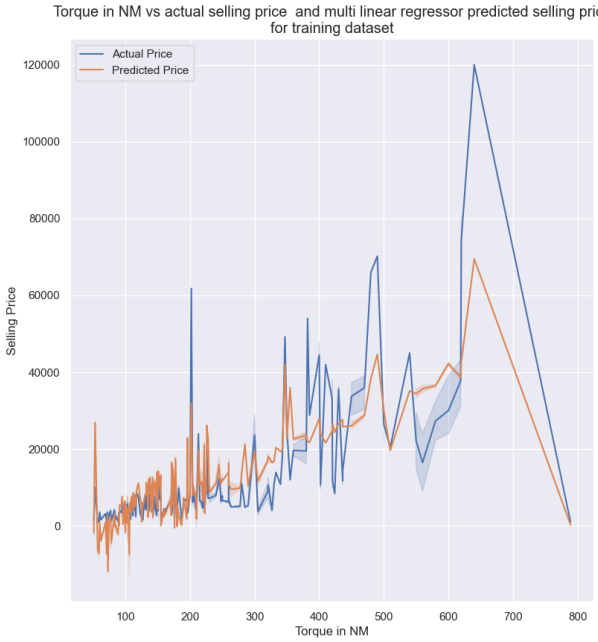
Below are the performance charts for regressor models.

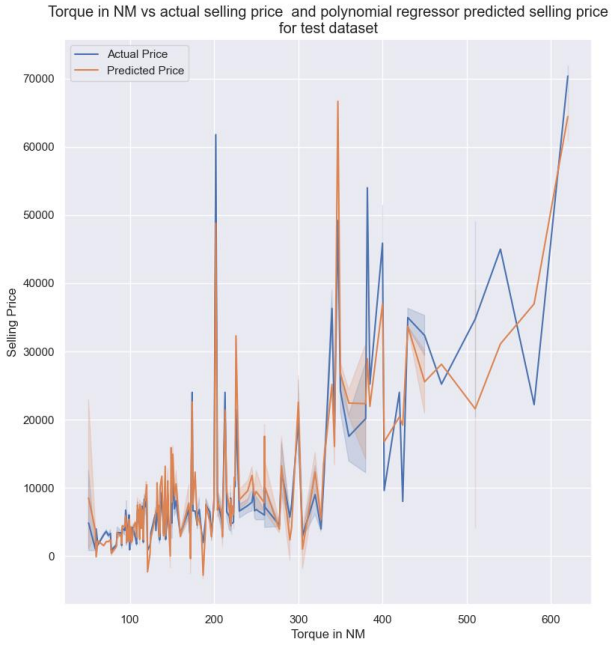
7.2.1 KBest Trained Regressor models

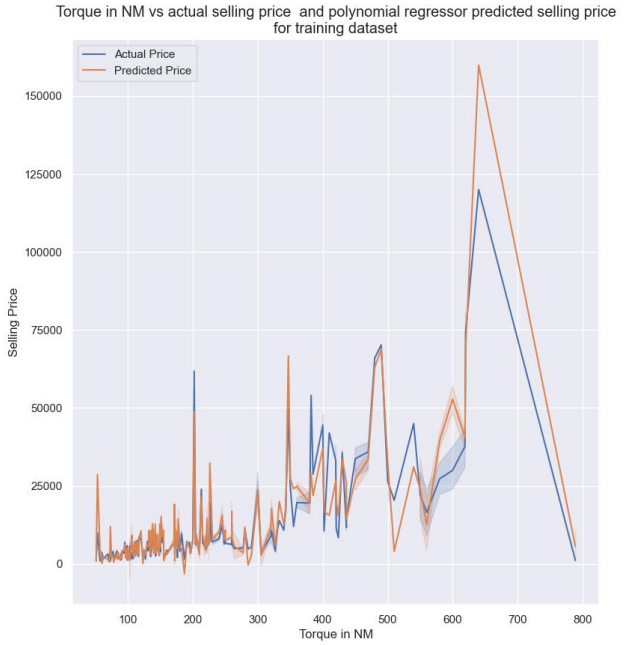


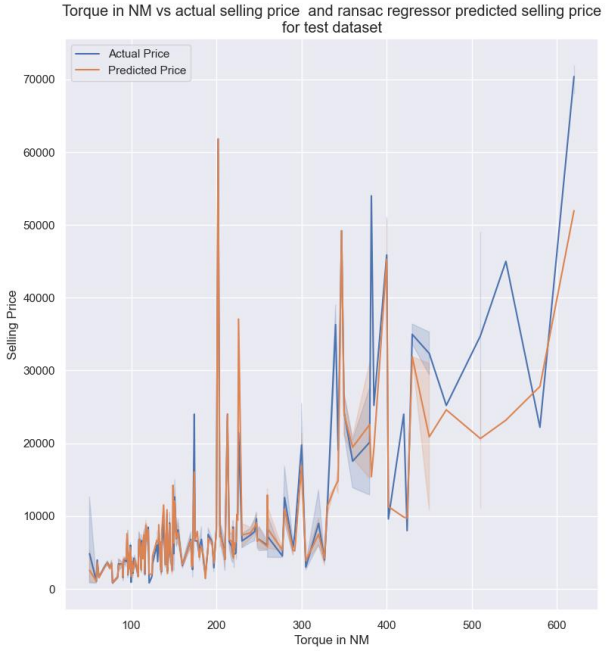


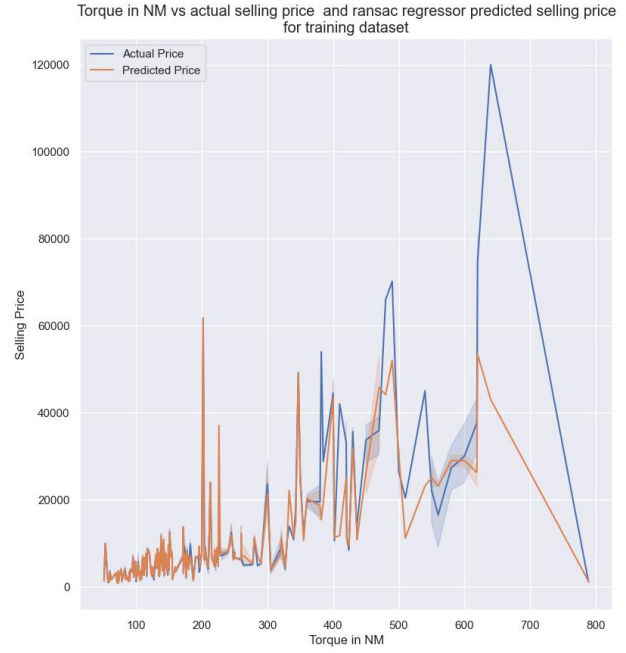


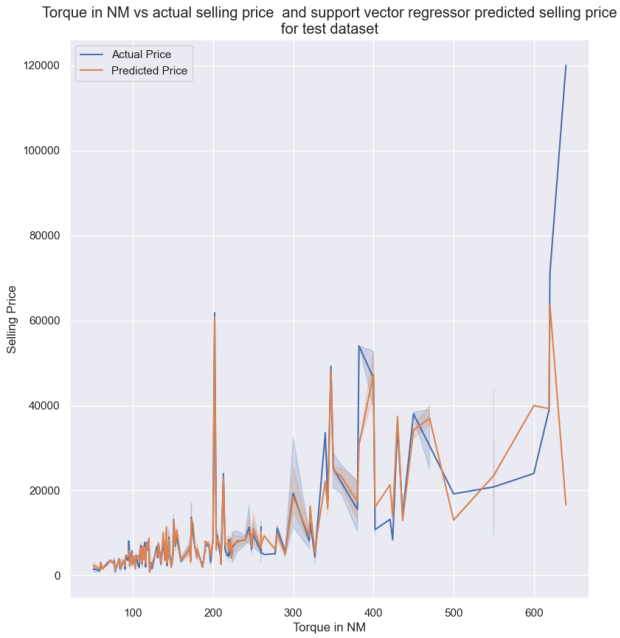


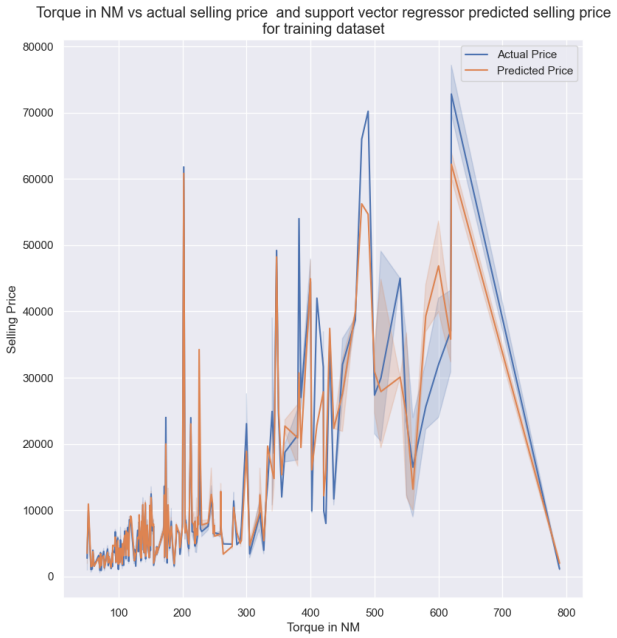

Torque in NM vs actual selling price and polynomial regressor predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Torque in NM

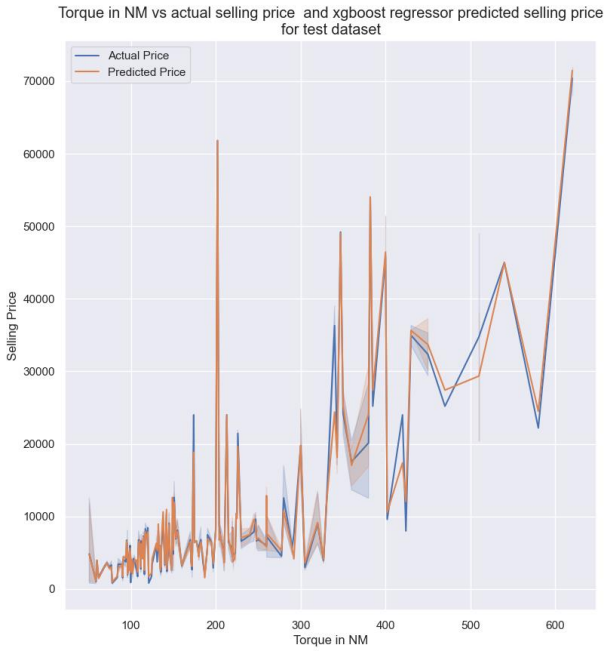

Torque in NM vs actual selling price and polynomial regressor predicted selling price for training dataset
Actual Price
Predicted Price
Selling Price
Torque in NM

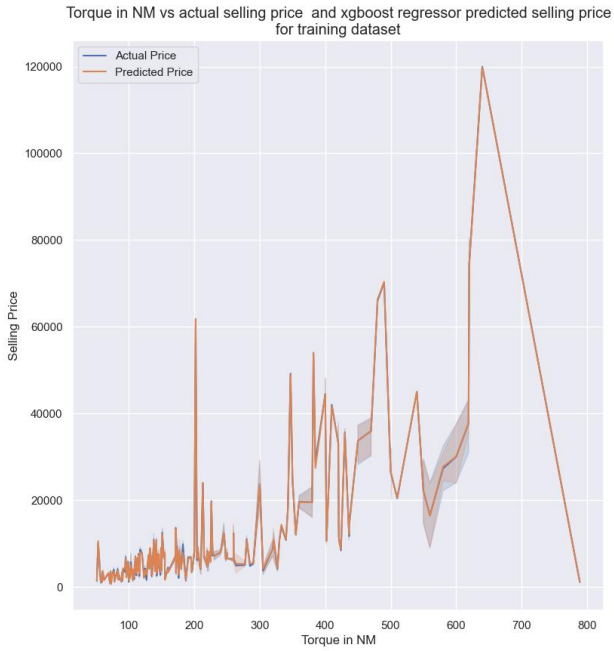

Torque in NM vs actual selling price and ransac regressor predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Torque in NM

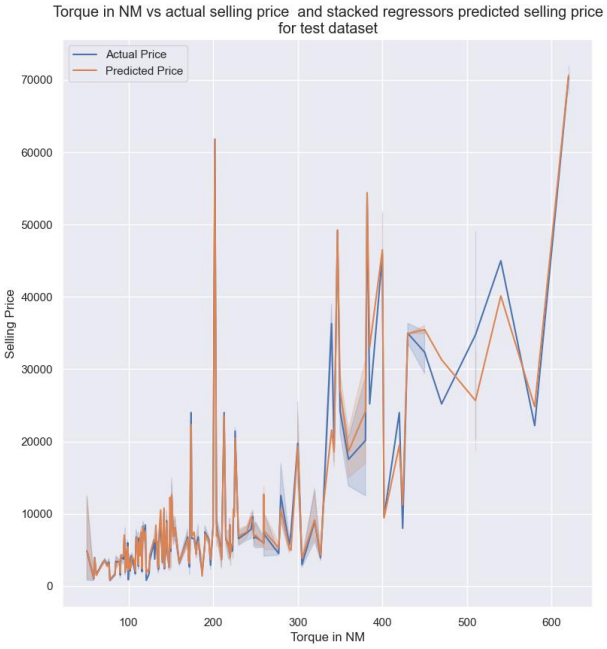

Torque in NM vs actual selling price and ransac regressor predicted selling price for training dataset
Actual Price
Predicted Price
Selling Price
Torque in NM

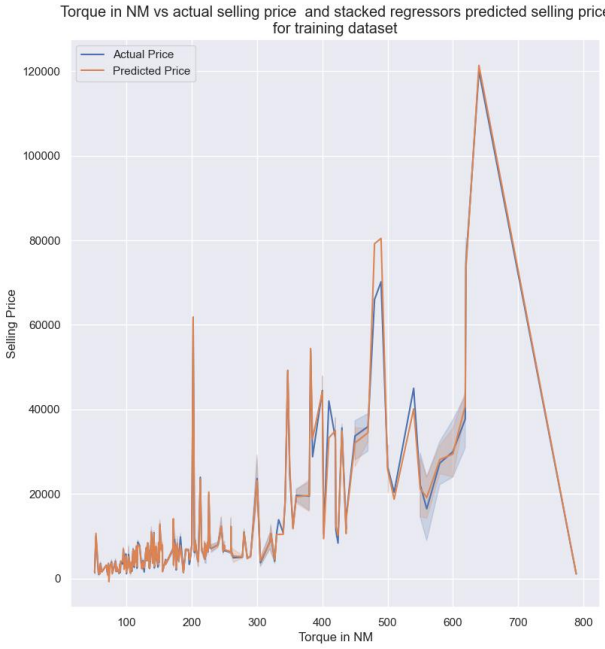

Torque in NM vs actual selling price and support vector regressor predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Torque in NM
0
20000
40000
60000
80000
100000
120000
100
200
300
400
500
600

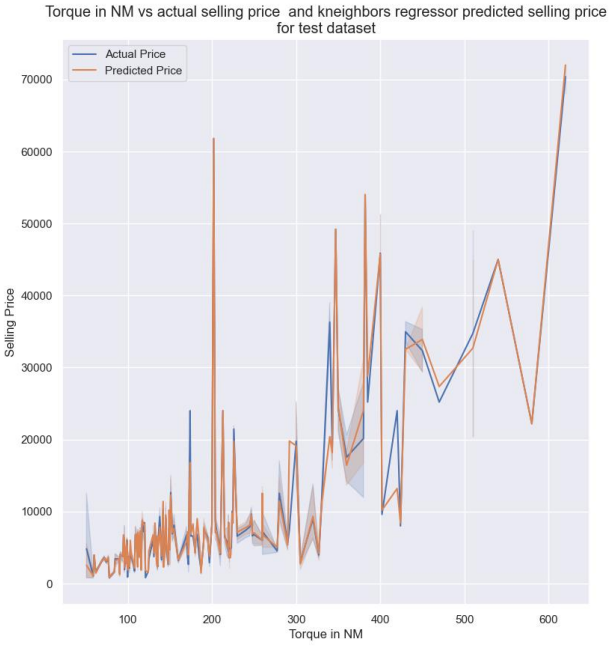

Torque in NM vs actual selling price and support vector regressor predicted selling price for training dataset
Actual Price
Predicted Price
Selling Price
Torque in NM
0
10000
20000
30000
40000
50000
60000
70000
80000
100
200
300
400
500
600
700
800

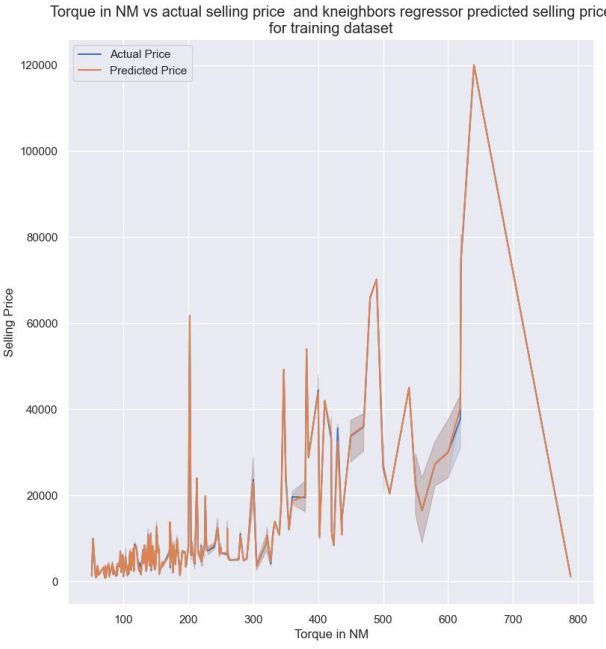

Torque in NM vs actual selling price and xgboost regressor predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Torque in NM
0
10000
20000
30000
40000
50000
60000
70000
100
200
300
400
500
600


Torque in NM vs actual selling price and xgboost regressor predicted selling price for training dataset
Actual Price
Predicted Price
Selling Price
Torque in NM
0
20000
40000
60000
80000
100000
120000
100
200
300
400
500
600
700
800

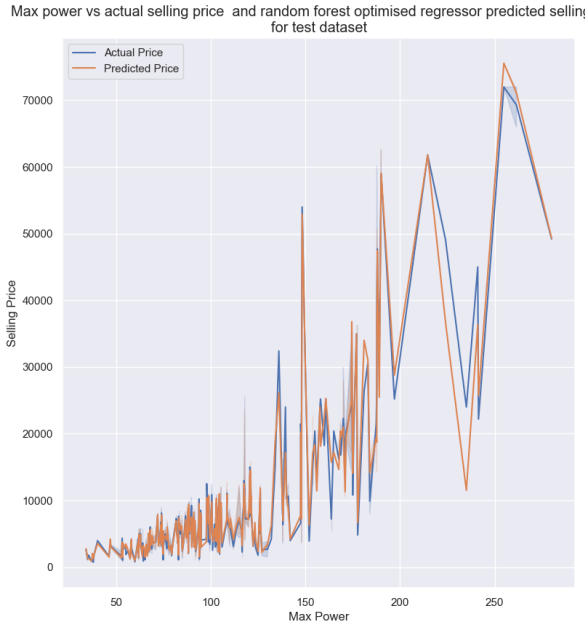

Torque in NM vs actual selling price and stacked regressors predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Torque in NM

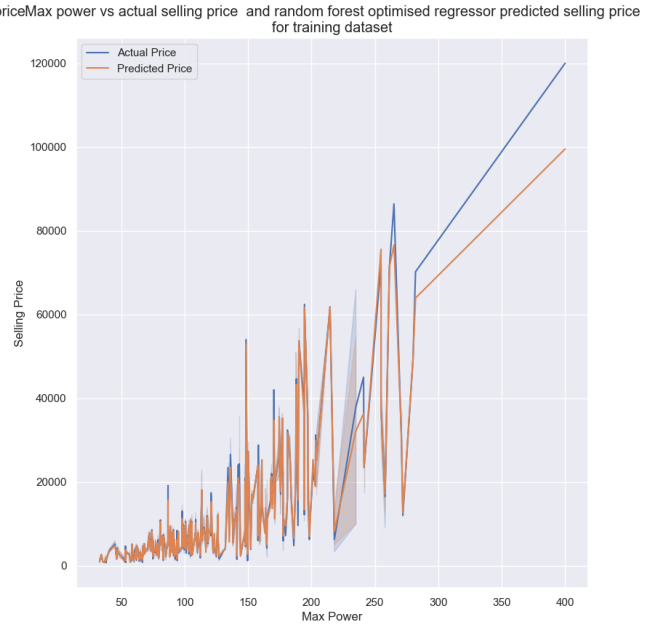

Torque in NM vs actual selling price and stacked regressors predicted selling price for training dataset
Actual Price
Predicted Price
Selling Price
Torque in NM

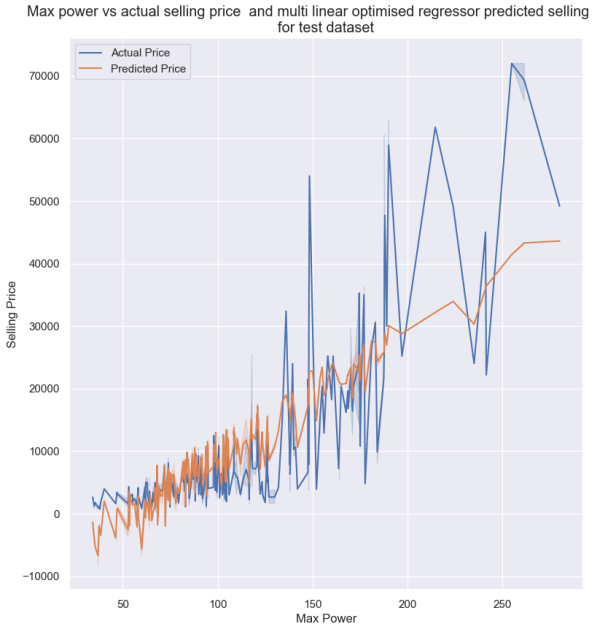

Torque in NM vs actual selling price and kneighbors regressor predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Torque in NM

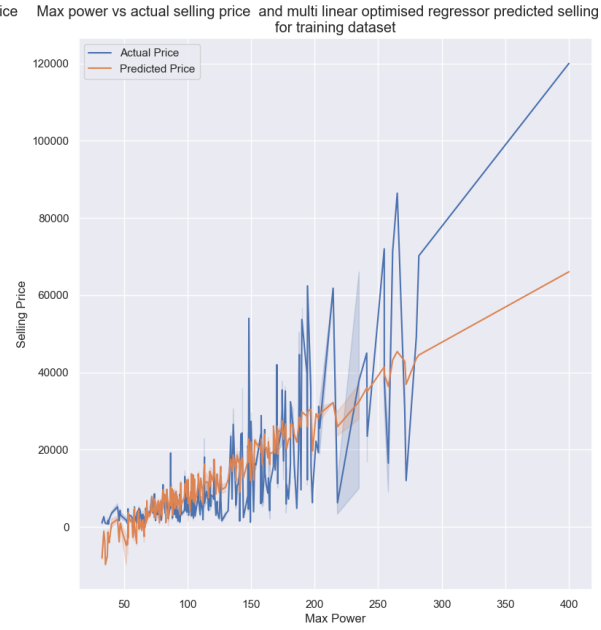

Torque in NM vs actual selling price and kneighbors regressor predicted selling price for training dataset
Actual Price
Predicted Price
Selling Price
Torque in NM

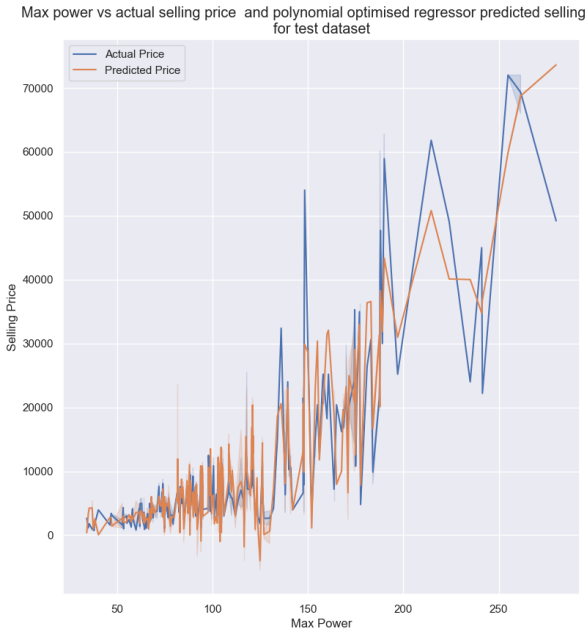
7.2.2 Random Forest Features Trained Regressor models

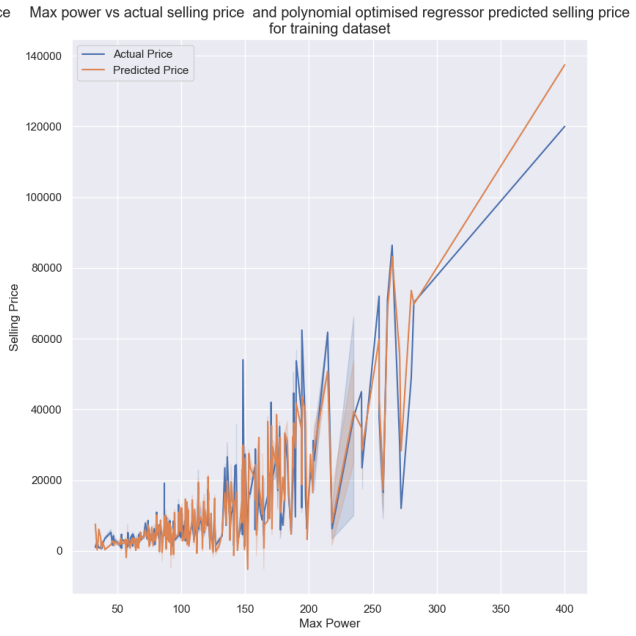


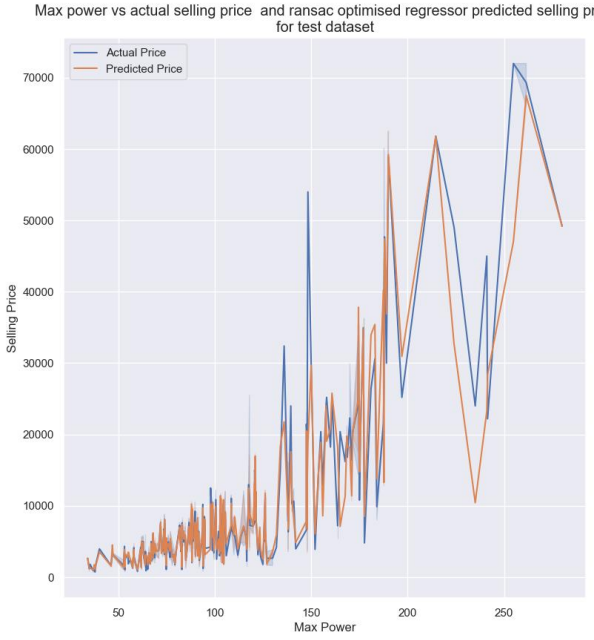


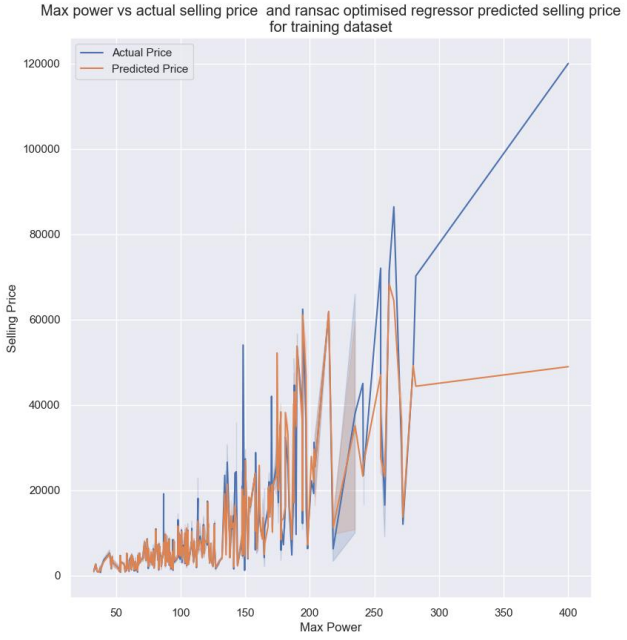




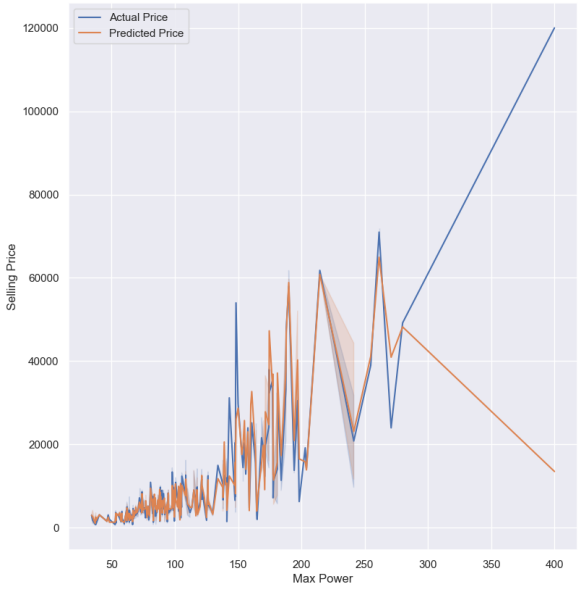

Max power vs actual selling price and polynomial optimised regressor predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Max Power


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

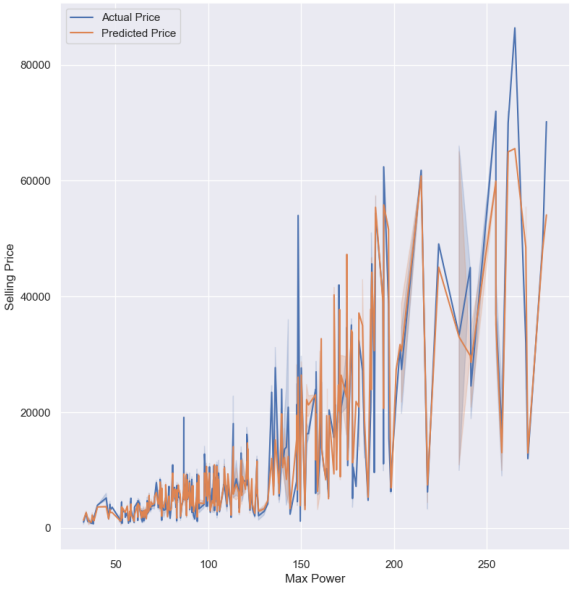

Max power vs actual selling price and ransac optimised regressor predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Max Power


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

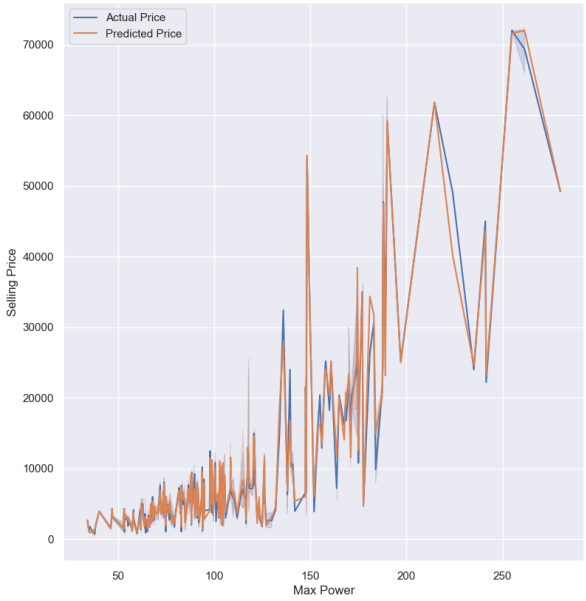
Max power vs actual selling price and support vector optimised regressor predicted selling price for test dataset



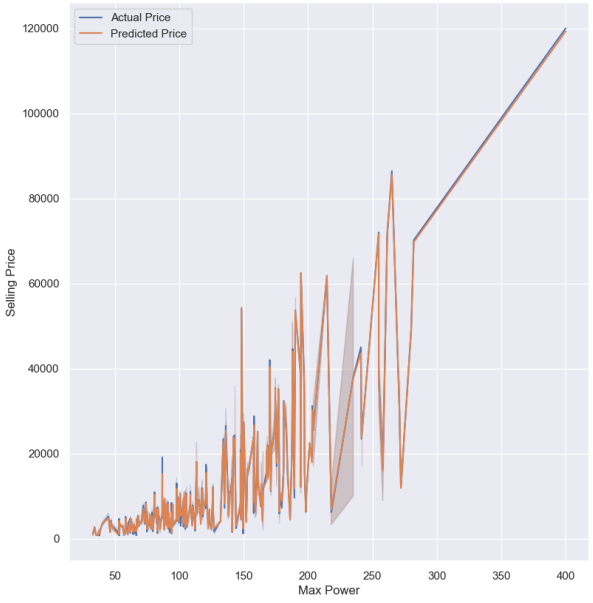
Max power vs actual selling price and support vector optimised regressor predicted selling price for training dataset

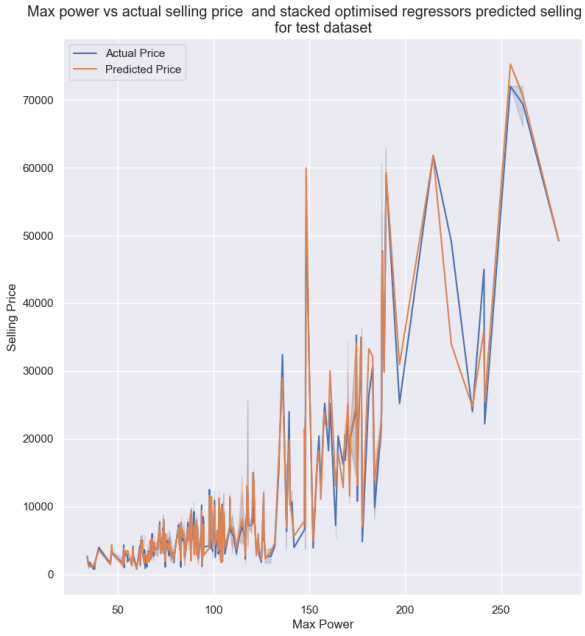


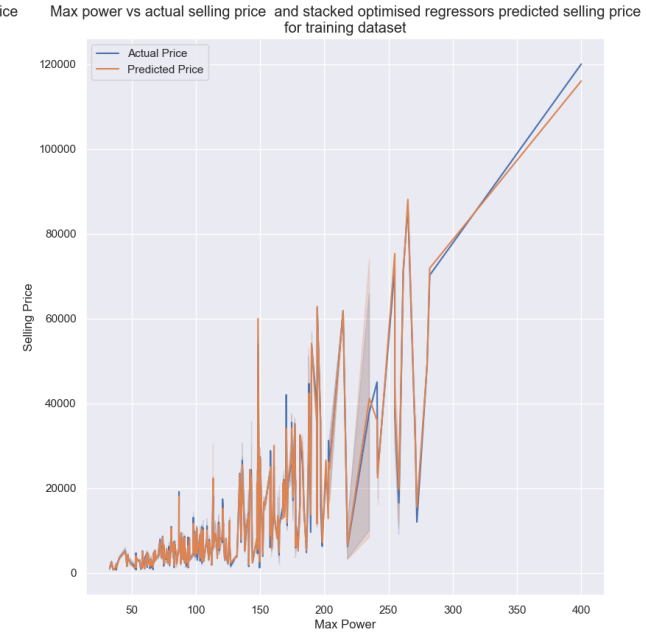
Max power vs actual selling price and xgboost optimised regressor predicted selling price for test dataset

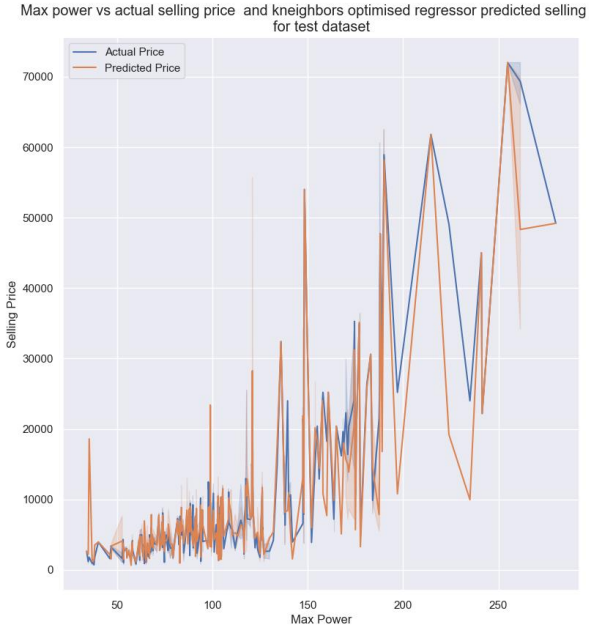


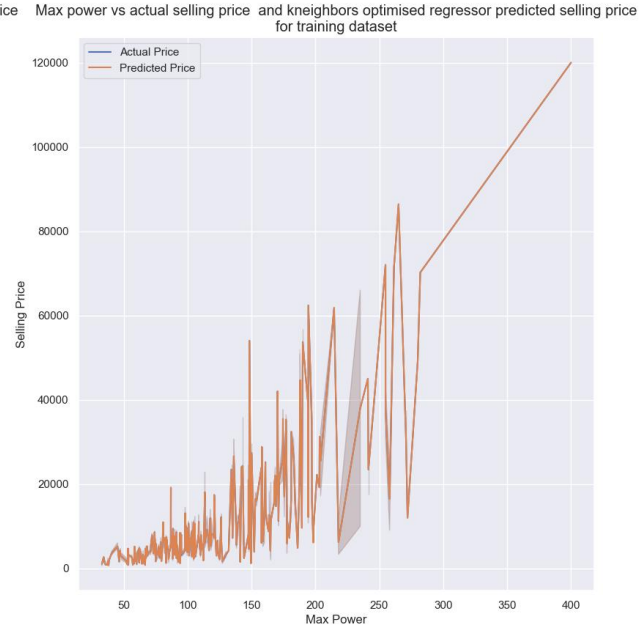
Max power vs actual selling price and xgboost optimised regressor predicted selling price for training dataset




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

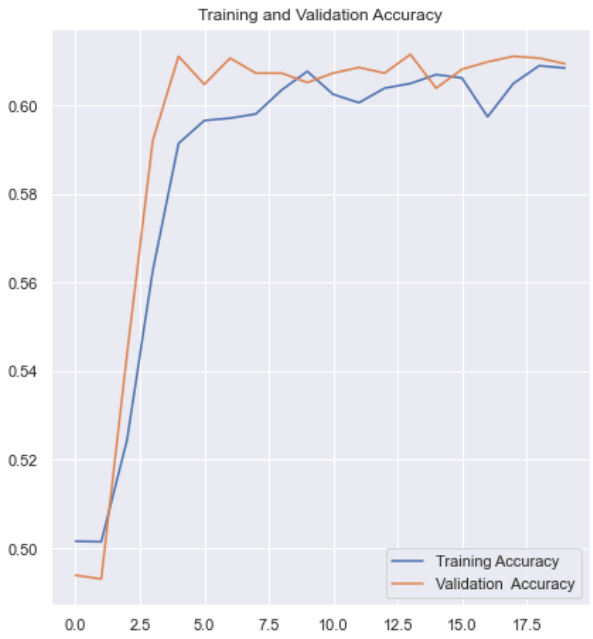

Max power vs actual selling price and stacked optimised regressors predicted selling price for training dataset
Actual Price
Predicted Price
Selling Price
Max Power

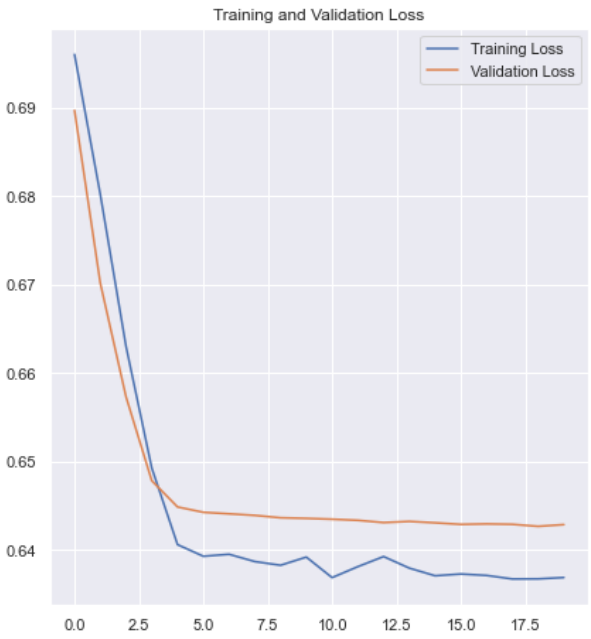

Max power vs actual selling price and kneighbors optimised regressor predicted selling price for test dataset
Actual Price
Predicted Price
Selling Price
Max Power


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

7.2.3 Artificial Neural Network (ANN) using back propagation

- Using K-Best best predicting features for sold or not-sold.





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