Used Cars Database

Springboard DSC Capstone Project 2 by Sneha Rani

Introduction

Just because a car isn't new doesn't mean it can't be new to anyone. Buying a used car can be just an exciting as purchasing a brand new model. Moreover, buying a used car can save big bucks in several ways. I think it’s an extremely important way for a family on a budget to save money.

Unfortunately, getting a used vehicle can also be just as complicated as figuring out which new car would suit the best. But machine learning techniques have made the things simple and easy for customers to purchase pre-owned cars with the best budget.

Cars today are really good and really last a long time. Used cars are a great value. Predicting the price of used cars is both an important and interesting problem but it’s not as simple as we think. The value of used cars depends on a number of factors. For example, the most important ones are usually the age of the car, its mileage, its horsepower etc. The look and feel of the car certainly contributes a lot to the price. As we can see, price depends on a large number of factors. Unfortunately, information about all these factors are not always available and the buyer must make the decision to purchase at a certain price based on few factors only. In this work, I have considered important factors which will help buyers to take better decision.

Problem and Client

The problem is to predict the price of used cars using machine learning techniques based on the different listed features of the car and the current market price. Also, to find the top features of a car and is sold more. Our client is Ebay Kleinanzeigen.

With difficult economic conditions, it is likely that sales of second-hand imported (reconditioned) cars and used cars will increase. It will be helpful for customers to get a correct pricing based on more features. They will get more knowledge about the new better pricing model.

Approach

- I will be doing exploratory data analysis, correlation and then build different models and find which fits best in the dataset.
- I will derive the relationship between the feature score and price.
- I will build machine learning models which will predict the price of used cars based on the features.

Data Pre-processing

We are going to use the data present in the source link:-

https://www.kaggle.com/orgesleka/used-cars-database

The data was downloaded from Kaggle. This dataset has over 370000 used cars scraped with Scrapy from Ebay-Kleinanzeigen. The content of data is in German so one has to translate it first if one can't speak German. It is pretty much clear and straightforward. Each row represents a car; each column contains car attributes. The dataset consists of 20 features and consist of (Int and Object) data type which include:

Data Dictionary

Column Name	**Data Types**	**Column Description**
dateCrawled	Object	when this ad was first crawled, all field-values are taken from this date
name	Object	"name" of the car
seller	Object	private or dealer
offerType	Object	the selling type of the ca

price	Int64	the price of the ad to sell the car
abtest	Object	unknown
vehicleType	Object	type of the car. Limousine, Kleinwagen, Kombi, Bus etc
yearOfRegistration	Int64	at which year the car was first registered
gearbox	Object	manuell or automatik
powerPS	Int64	the power of the car in PS
model	Object	the model of the car
kilometer	Int64	how many kilometers the car has driven
monthOfRegistration	Int64	at which month the car was first registered
fuelType	Object	benzin, diesel, lpg etc
brand	Object	brand of the car. Mercedes, Porsche, Audi etc..
notRepairedDamage	Object	if the car has a damage which is not repaired yet. Yes or no
dateCreated	Object	the date for which the ad at ebay was created
nrOfPictures	Int64	number of pictures in the ad
postalCode	Int64	code that shows the location of the car
lastSeenOnline	Object	when the crawler saw this ad last online

Tools Used

Pandas: Loading the data, data wrangling, and manipulation

Scikit-learn: Libraries for Regressor, Model evaluation, Feature Importance

Data Visualization: MatplotLib, and Seaborn

Data Cleaning / Data Wrangling

Used car dataset was downloaded from the data source link in the .csv file format and then it was imported on the Jupyter notebook using Pandas. The dataset was not pre-cleaned. It had missing values for the columns vehicleType, gearbox, model, fuelType, notRepairedDamaged which were of the categorical type. I filled all the NaN values by introducing a new category called **'not-available'** for all these columns and checked again for any more missing values to be fixed. There were no more missing values and the dataset was then checked to determine data types, shape and definition of each column in order to filter the columns. Certain columns as the seller, offerType, abtest, monthOfRegistration, dateCreated, nrOfPictures, postalCode, lastSeenOnline, model, dateCrawled, name were found to be insignificant and contained no information. These columns were removed from the final dataset for further analysis.

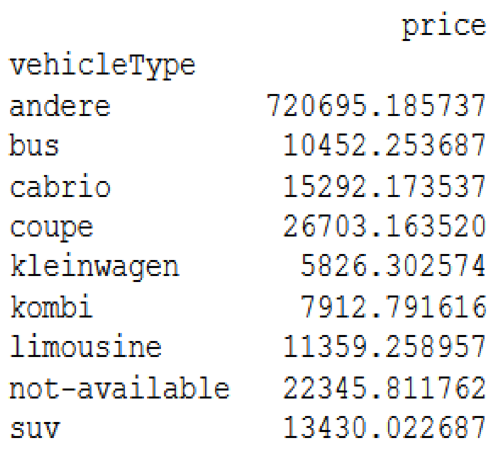
Exploratory Analysis

Certain initial findings were then performed for the used car dataset which included interesting questions as:

- What is the average price of the car based on vehicleType?
- Can we find out the min, max and average price of the car of different brands?
- How many cars are having invalid registration year?
- Which is the most popular brand among used car in the market?
- What is the average price of Volkswagen? How many cars of 'Volkswagen' brand are on sale?
- Which is the most running model in Volkswagen?
- Which vehicle type has maximum number of automatic transmission?
- What is the min, max and average horse power of different vehicle type?

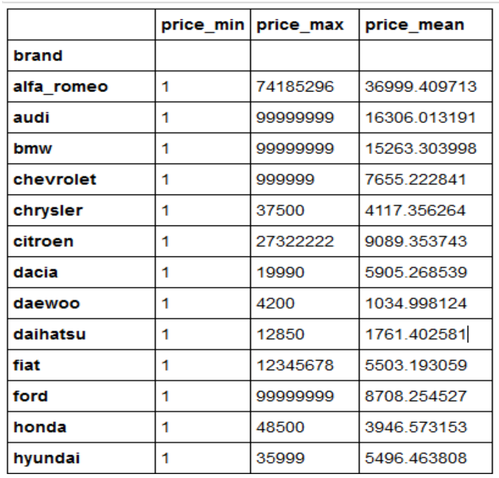
I started with the analysis on some of the fields of ‘Used Car Database’ and found that seller, offerType, monthOfRegistration, nrOfPictures, abtest, dateCreated had no distinguishing information for the model and so these columns were dropped for future analysis.

I further surveyed the average price of the car based on vehicleType. The table (Tbl. 1) showed the average price of each vehicle type among which 'andere' vehicle type having (720695.18 Euro) was a bit costlier than others (bus, cabrio, coupe, kleinwagen, kombi, limousine, suv).



```
                          price
vehicleType
andere            720695.185737
bus                10452.253687
cabrio             15292.173537
coupe              26703.163520
kleinwagen          5826.302574
kombi               7912.791616
limousine          11359.258957
not-available      22345.811762
suv                13430.022687
```

Table 1



	price_min	price_max	price_mean
brand			
alfa_romeo	1	74185296	36999.409713
audi	1	99999999	16306.013191
bmw	1	99999999	15263.303998
chevrolet	1	999999	7655.222841
chrysler	1	37500	4117.356264
citroen	1	27322222	9089.353743
dacia	1	19990	5905.268539
daewoo	1	4200	1034.998124
daihatsu	1	12850	1761.402581
fiat	1	12345678	5503.193059
ford	1	99999999	8708.254527
honda	1	48500	3946.573153
hyundai	1	35999	5496.463808

Table 2

After that, I tried finding out the min, max and average price of the car of different brands and saw from the above analysis (Tbl. 2) that car prices varied from 1 to 99999999 and had a varied average price. Prices as low as 1 and as high as 999999999 did not define well for the model. It needs to be cleaned further and filtered for better price range data.

Next, I counted the total number of cars having invalid registration year. About 113 cars had no proper valid registration year. I then visualized the most popular brand among used car in the market.

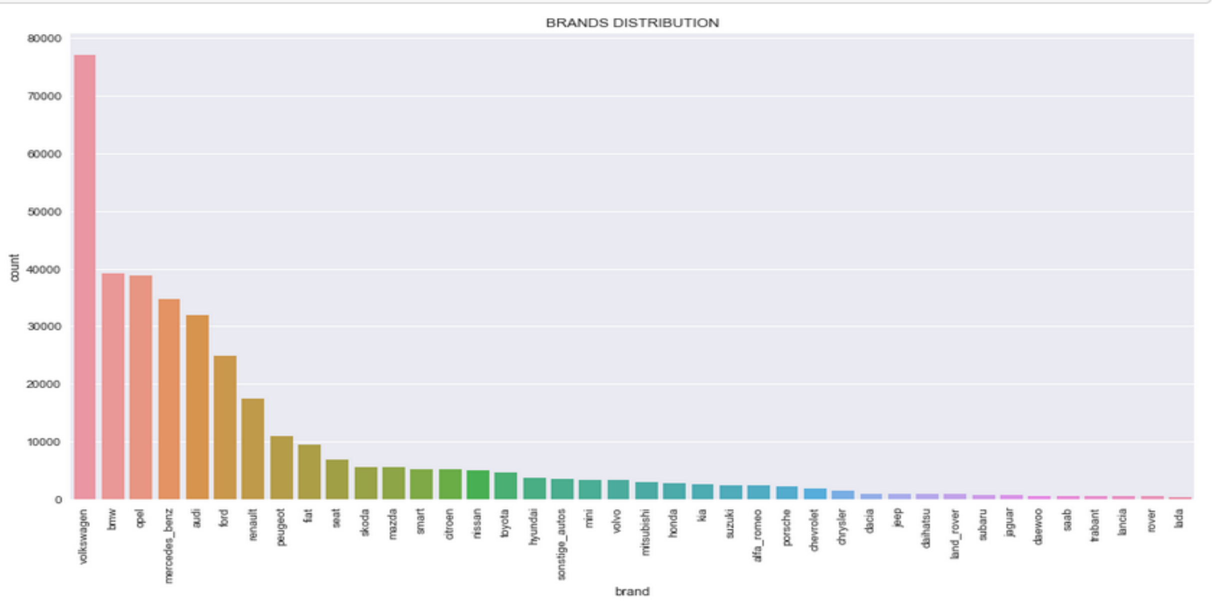


Figure 1

Based on the above bar graph (Fig. 1), this column had 40 different brands having consistent records which could be useful for the model. Volkswagen was listed as the most popular brand. BMW was the next competitor. As Volkswagen was considered to be the best brand from the above graph analysis, I also tried finding the average price of the Volkswagen and the total number of cars of 'Volkswagen' brand on sale. The result showed 77039 number of Volkswagen cars on sale having the mean price, 15024 Euros.

I deep dived more into the Volkswagen brand and also visualized the most running model on the same.

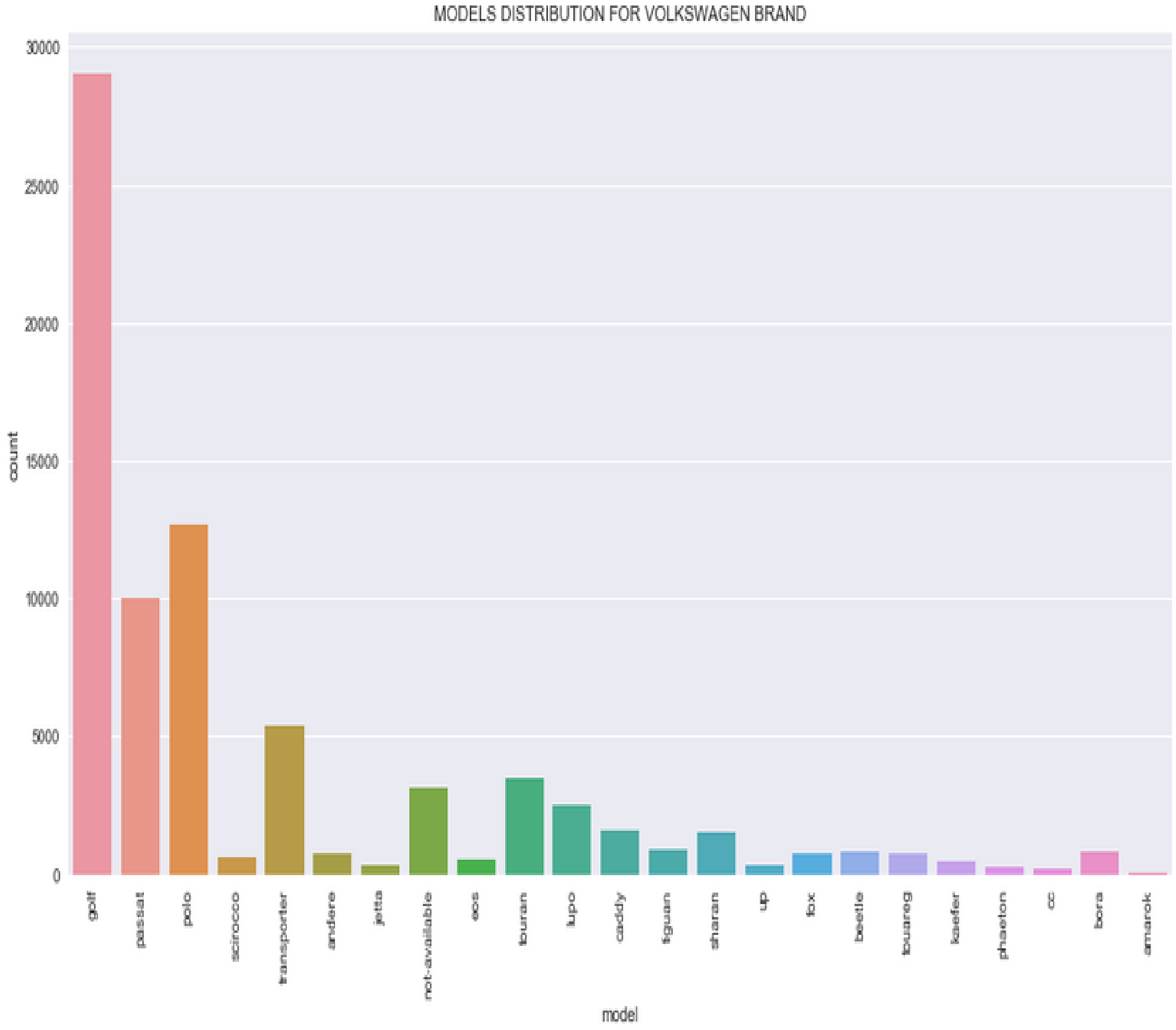


Figure 2

From the above graph (Fig. 2) of 'MODELS DISTRIBUTION FOR VOLKSWAGEN BRAND', I could see that ‘golf’ is the most popular and running model in Volkswagen. Next, I could visualize is ‘polo’ which is the second competitor model among Volkswagen brand.

My last second analysis was which vehicle type contributed the maximum number of automatic transmission.

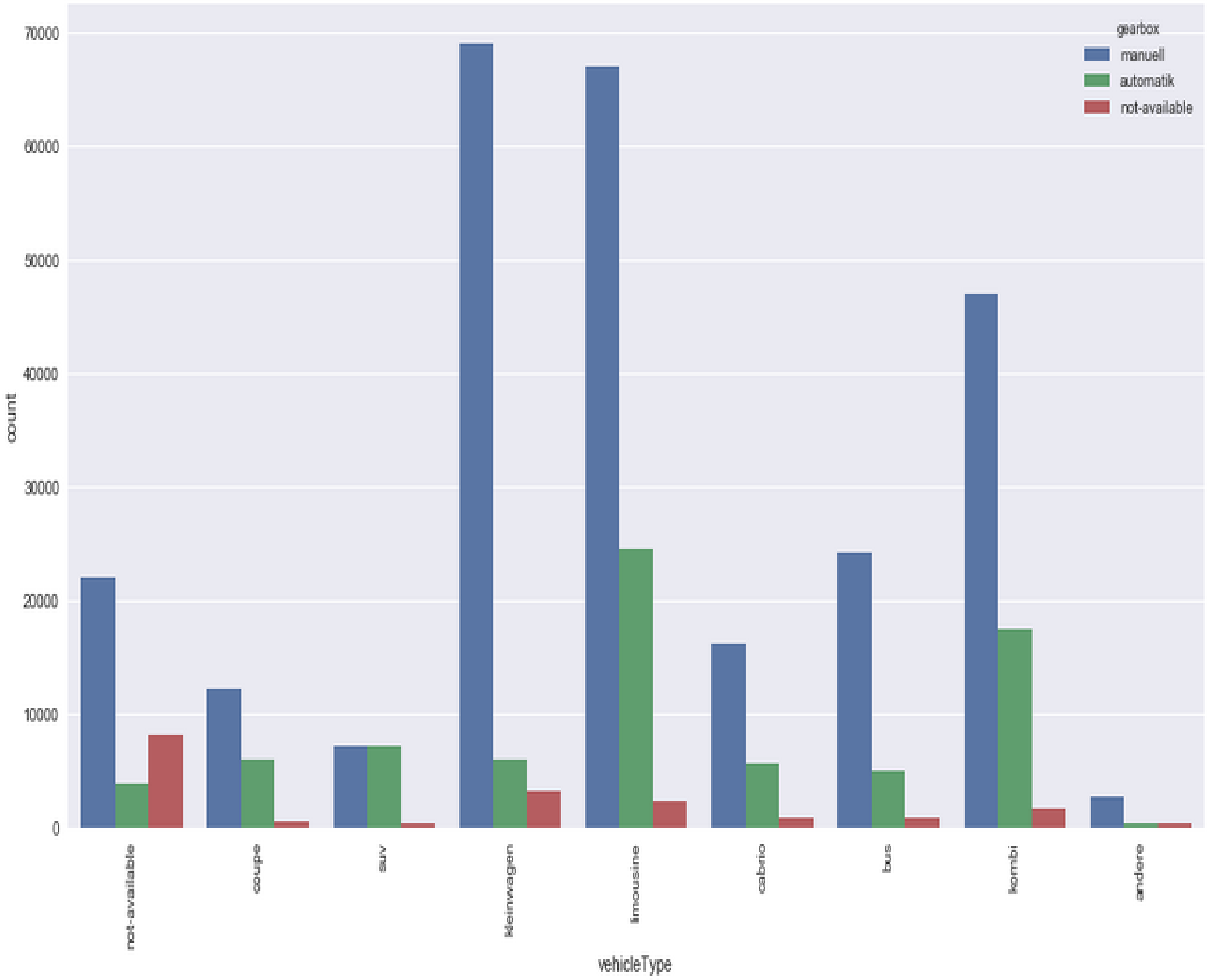
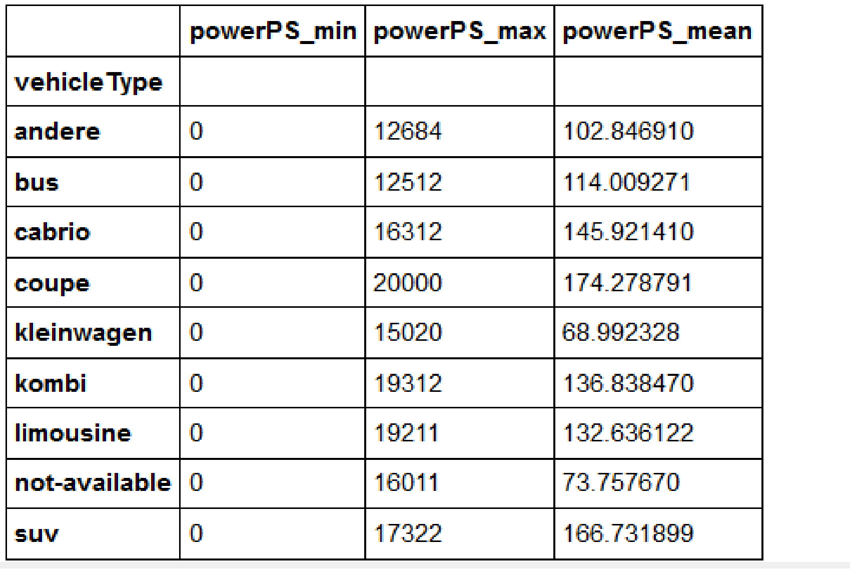


Figure 3

Based on the above bar graph (Fig. 3), I could see that ‘Limousine’ showed the maximum number of automatic transmission than ‘Kombi’. The last but not the least initial finding was the min, max and average horsepower of different vehicle type.



	powerPS_min	powerPS_max	powerPS_mean
vehicleType			
andere	0	12684	102.846910
bus	0	12512	114.009271
cabrio	0	16312	145.921410
coupe	0	20000	174.278791
kleinwagen	0	15020	68.992328
kombi	0	19312	136.838470
limousine	0	19211	132.636122
not-available	0	16011	73.757670
suv	0	17322	166.731899

Table 3

While analyzing min, max and average values of powerPS from the above table (Tbl. 3) on different vehicle type, minimum powerPS of a car is **0** and maximum is **20000** powerPS which is not a valid value. So, we can limit the powerPS values to have better data understanding in order to avoid inconsistent data.

As I was getting close to the analysis I came up with some more interesting questions as:

- Does postal code have an impact on the price? Does price vary of different region?
- Is there any variance in the price of the car based on year of registration? Does vehicle of various type impact differently on car price and year of registrations?
- How horsepower of a car impacts the price range?

Few statistics were calculated using inferential statistics for finding the relationship between the two variables using Pearson coefficient. I started with the first one to check how ‘postalCode’ had an impact on the price and whether the price of used cars really varied with the different region.

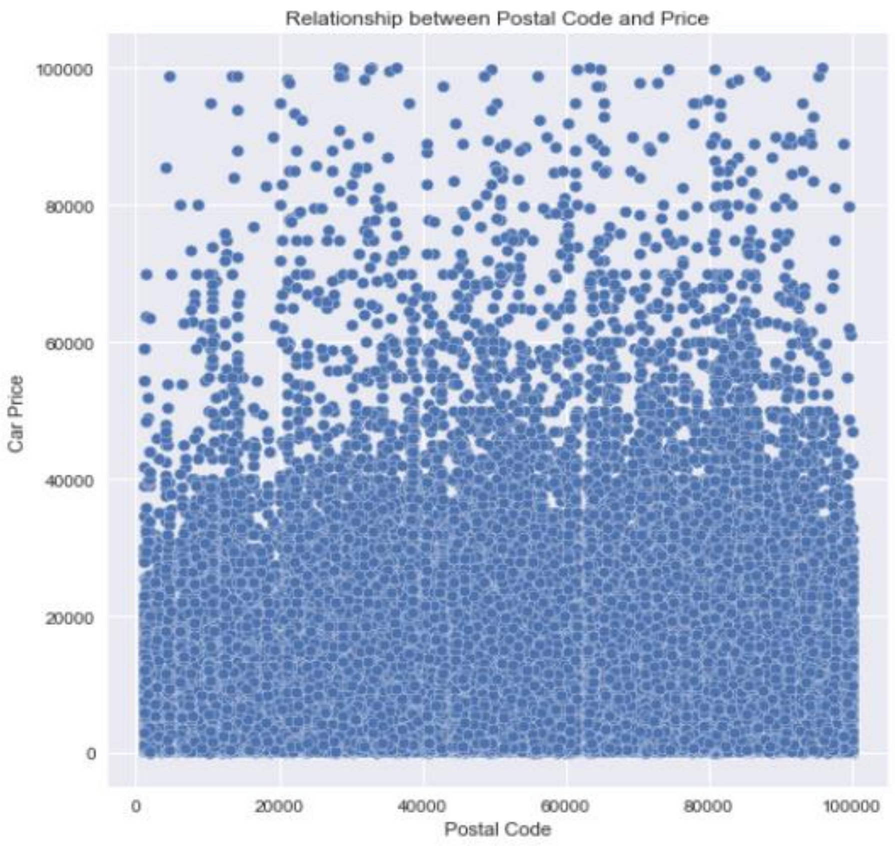
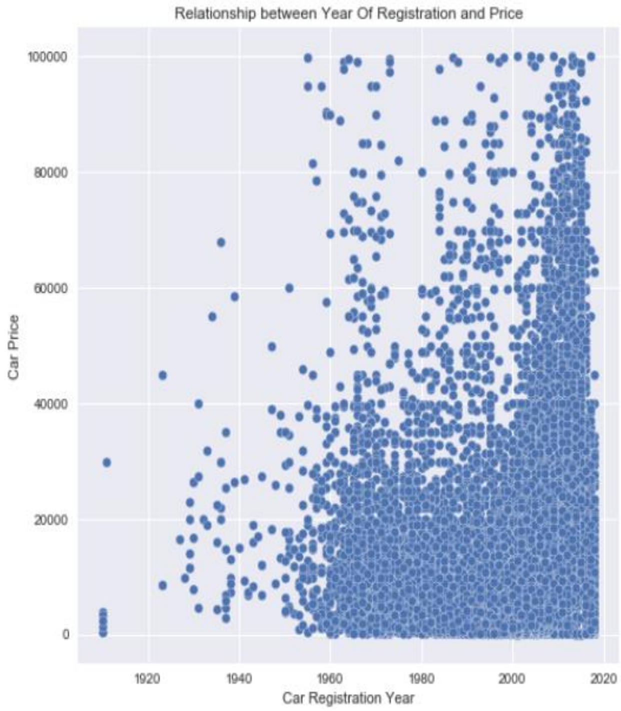


Figure 4

The postal code is one of the features which may be helpful in determining the location of the car. But, the above scatter plot (Fig. 4) didn't show any pattern between postal code and price. Prices range from the cars across various regions were distributed all along. So, I could hardly find any relation with the target variable price.

I then checked if there was any variance in the price of the car based on year of registration and if the vehicle of various type statistically impacted on car price and year of registration.

From the below scatter plot graph (Fig. 5) based on year vs. price, I found that newly registered cars seemed to be more expensive than the older ones. However, there are also some old cars which are also expensive.



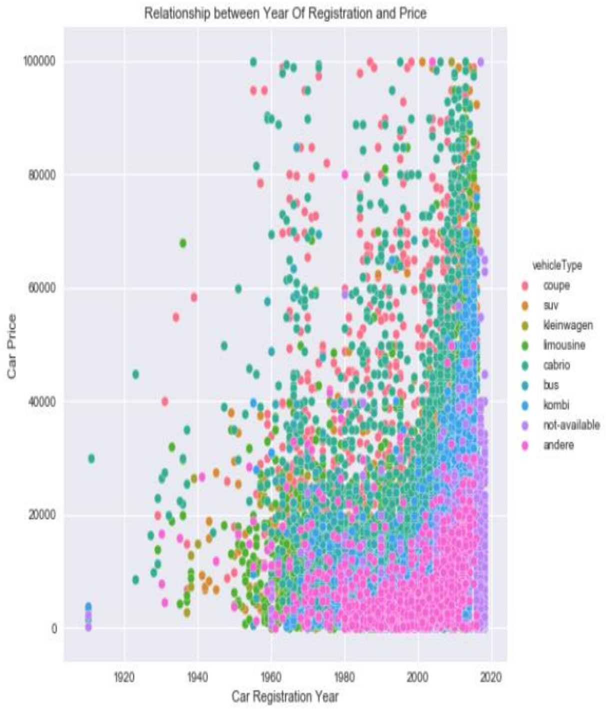
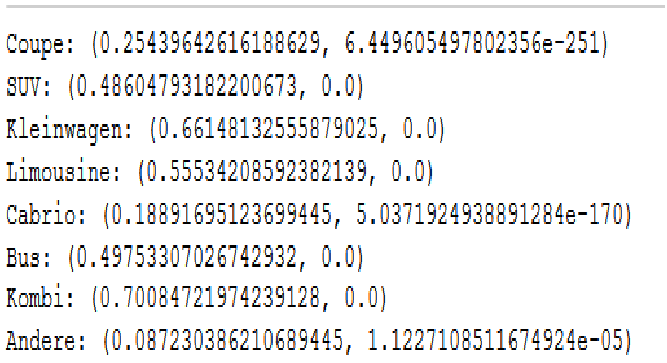


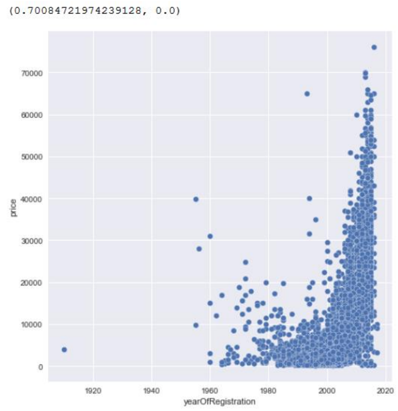
Figure 5

Figure 6

Also, next scatter plot (Fig. 6) shows that there exists a positive relationship between year of registration and price. But, Pearson correlation coefficient for the above graph is ***0.35*** which shows the strength of the relationship is weak between the year of registration and price. I further grouped the plot by vehicle type and checked Pearson correlation coefficient for each vehicle type. I specifically plotted for vehicle type 'kombi' as it showed a strong positive correlation of ***0.70*** (Tbl.4) among all the vehicle types.



```
Coupe: (0.25439642616188629, 6.449605497802356e-251)
SUV: (0.48604793182200673, 0.0)
Kleinwagen: (0.66148132555879025, 0.0)
Limousine: (0.55534208592382139, 0.0)
Cabrio: (0.18891695123699445, 5.0371924938891284e-170)
Bus: (0.49753307026742932, 0.0)
Kombi: (0.70084721974239128, 0.0)
Andere: (0.08723038621068945, 1.1227108511674924e-05)
```



```
(0.70084721974239128, 0.0)
```

Table 4

Figure 7

The above plot clearly shows that as the year of registration increases, car prices also increases which means price value of old age cars is cheaper than new ones in 'Kombi' types of vehicle.

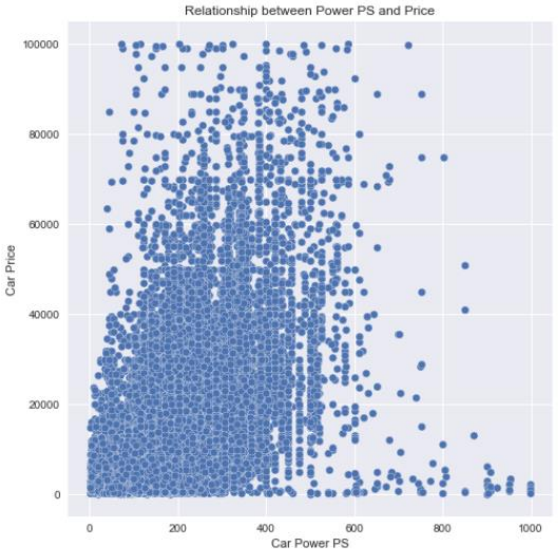
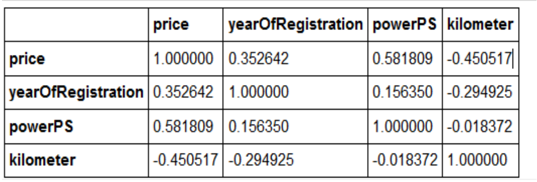


Figure 8

Last analysis was how horse power of a car had an impact on the price. I analyzed from the above graph (Fig.8) that based on powerPS vs. price graph, low powerPS cars are cheaper than others. Most of the data points lie between 0-500 ps power car which does indicate the most consuming powerPS of used cars which are on sale. Similarly, power cars do have varied price range which does indicate that there are other factors contributing to that power variation also.

At last, I tried to visualize how each variable are correlated to one another and more importantly seeing its influence on the prices of the used cars by plotting heatmap.



	price	yearOfRegistration	powerPS	kilometer
price	1.000000	0.352642	0.581809	-0.450517
yearOfRegistration	0.352642	1.000000	0.156350	-0.294925
powerPS	0.581809	0.156350	1.000000	-0.018372
kilometer	-0.450517	-0.294925	-0.018372	1.000000

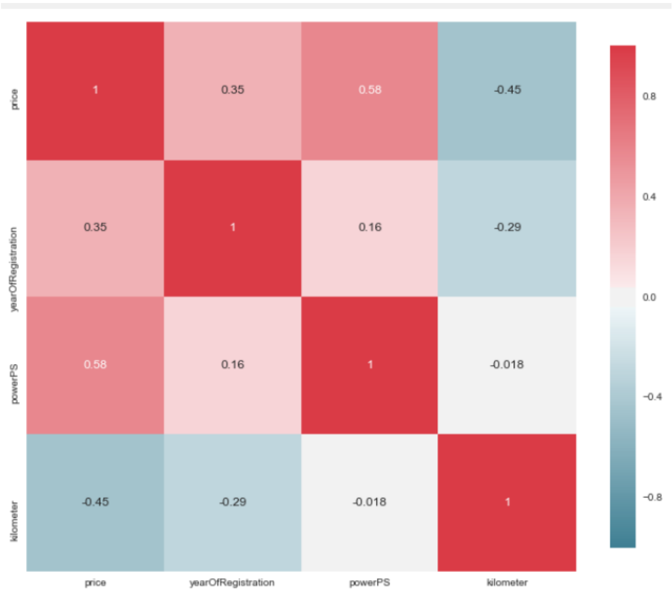


Table 5 **Figure 9**

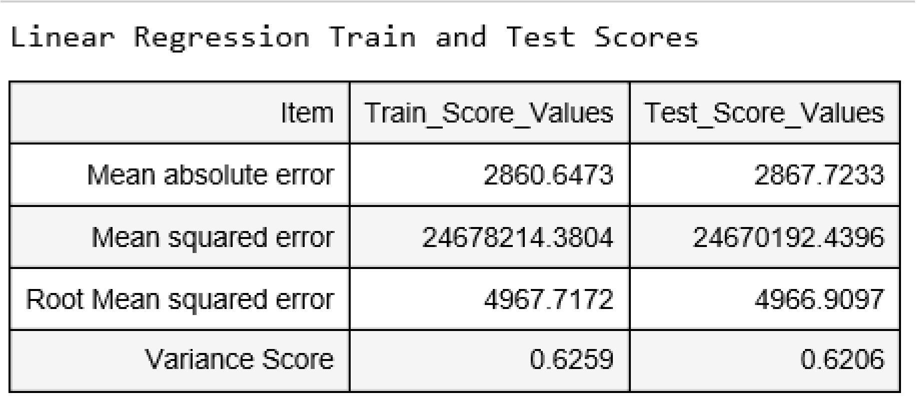
From the above heatmap, powerPS showed one of the most influencing features for the price.

Modelling the Data

Now knowing the fact that how each variable behaved on price, it was time to prepare my first baseline model using linear regression. Before preparing the model, I removed some of the insignificant columns from the analysis done earlier and few extreme outliers on the columns price, powerPS, yearOfRegistration by providing certain range of values. Next, I had created dummy variables for categorical type for easy interpretation. I created dummy variables on vehicleType, gearbox, fuelType, brand, notRepairedDamage. After creating dummy variables, final dataframe was created and it included 67 columns.

Now, the final data was ready to be modeled with the first column as ‘Price’ in order to define the parameter with ‘X’ (features) and ‘y’ (target). It is denoted by Matrix ‘X’ and vector ‘y’.

When we talk about fitting the model we would like to ensure two things - we have found the best model (in terms of model parameters) and next to the model is highly likely to generalize i.e. perform well on unseen data. First, I tried building linear regression baseline model. The data was then split into a training and test (hold-out) set. Data was then trained on the training set and tested for accuracy on the testing set.



Linear Regression Train and Test Scores

Item	Train_Score_Values	Test_Score_Values
Mean absolute error	2860.6473	2867.7233
Mean squared error	24678214.3804	24670192.4396
Root Mean squared error	4967.7172	4966.9097
Variance Score	0.6259	0.6206

Table 6

From the above table (Tbl. 6) variance score of 0.6259 and low MSE (24670192.4396) quantifies the quality of the model trained on the training set and tested on the test set. After that, I checked the magnitude of the coefficients.

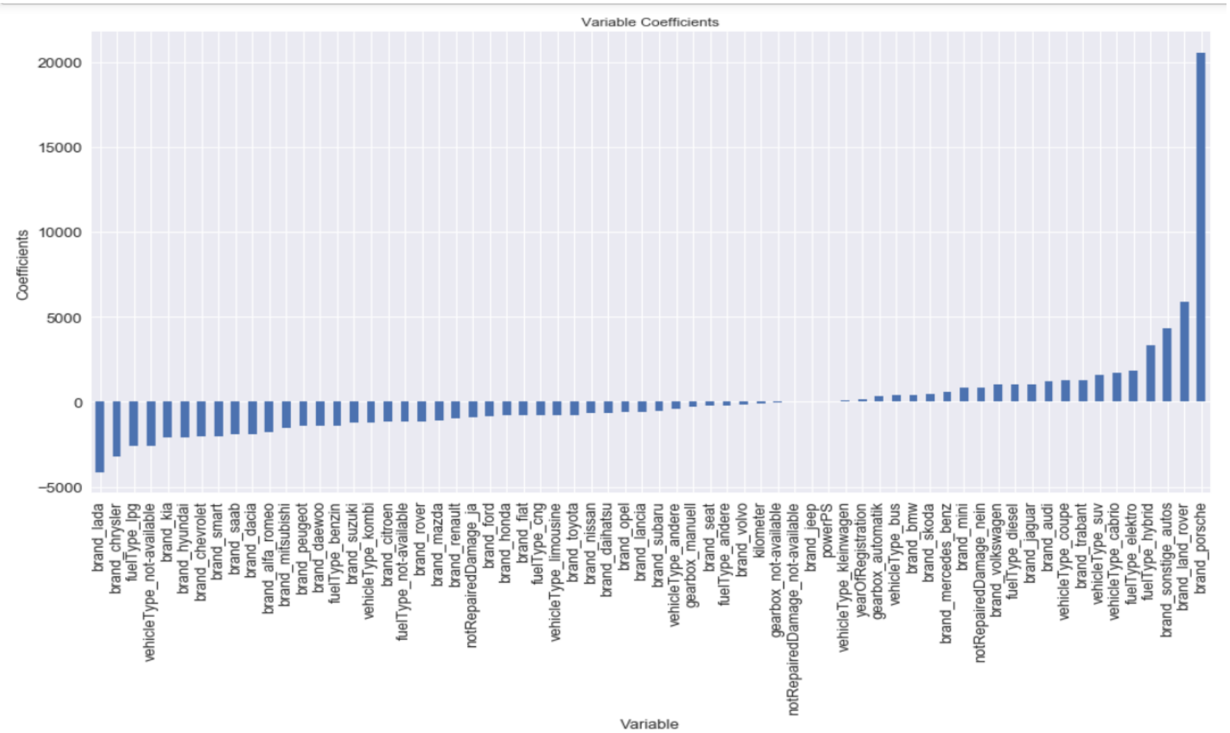
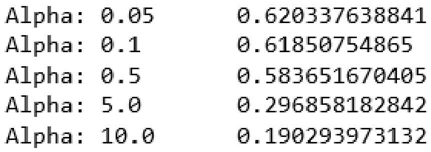


Figure 10

From the above bar graph, I visualized that coefficient of brand Porsche was the one having a higher coefficient as compared to the rest. This means that the price of the used car may be more driven by this feature.

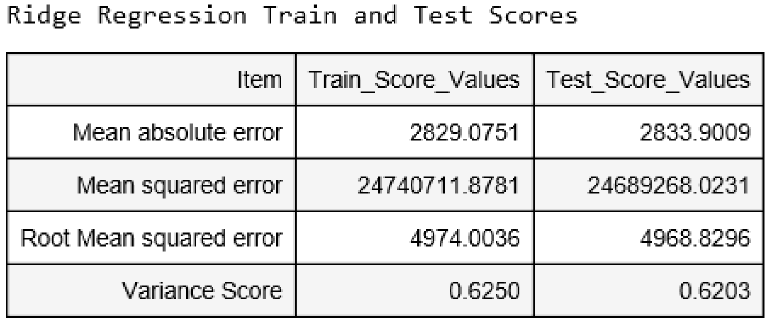
As we know that large coefficients can lead to overfitting, we need to penalize the large coefficients. For this, I have used regression techniques (Ridge and Lasso) which uses regularization to overcome this problem. I started off with Ridge Regression technique for computing R-squared scores over a certain range of alphas as shown below: (Tbl.7)



```
Alpha: 0.05      0.620337638841
Alpha: 0.1       0.61850754865
Alpha: 0.5       0.583651670405
Alpha: 5.0       0.296858182842
Alpha: 10.0      0.190293973132
```

Table 7

From the above findings, we could see that as the value of alpha increased, the magnitude of the coefficients decreased. I calculated R-square for each alpha and saw that the value of R-square (0.62) is maximum at alpha=0.05 which gives the lowest error. Keeping alpha value = 0.05, I then computed Ridge Regression train and test scores based on Mean absolute error, Mean squared error, Root Mean squared error, Variance Score which is shown below: (Tbl.8)



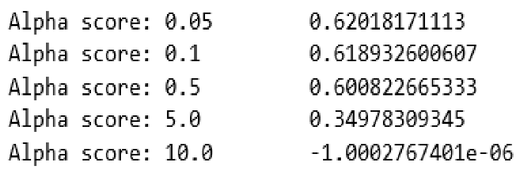
Ridge Regression Train and Test Scores

Item	Train_Score_Values	Test_Score_Values
Mean absolute error	2829.0751	2833.9009
Mean squared error	24740711.8781	24689268.0231
Root Mean squared error	4974.0036	4968.8296
Variance Score	0.6250	0.6203

Table 8

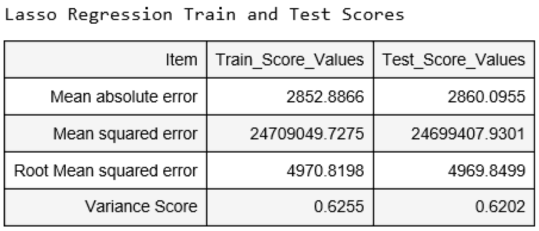
From the above table, I could see that variance score (0.6203) of test dataset is almost same as compared to the linear model, only mean squared error has slightly increased which intends not to be a good model.

I applied the same with Lasso Regression and computed the train and test scores based on Mean absolute error, Mean squared error, Root mean squared error, Variance score by selecting alpha as 0.05 which shows high R-square value among all.



```
Alpha score: 0.05        0.62018171113
Alpha score: 0.1         0.618932600607
Alpha score: 0.5         0.600822665333
Alpha score: 5.0         0.34978309345
Alpha score: 10.0        -1.0002767401e-06
```

Table 9



Lasso Regression Train and Test Scores

Item	Train_Score_Values	Test_Score_Values
Mean absolute error	2852.8866	2860.0955
Mean squared error	24709049.7275	24699407.9301
Root Mean squared error	4970.8198	4969.8499
Variance Score	0.6255	0.6202

Table 10

From the above table (Tbl. 10), I could see that variance score (0.6203) of test dataset is almost same as compared to the linear and ridge model, only mean squared error has increased which intends not to be a better fit for the model.

From all the analysis done till now, Linear Regression (lr1) model is predicting better as it has a good R-squared score with minimum MSE (*(24670192.4396)* both Lasso and Ridge. Considering Linear Regression as a better model, I created a scatterplot between the predicted prices and the original prices and tried to evaluate this plot to see how well my regression model predicts the price of given the data.

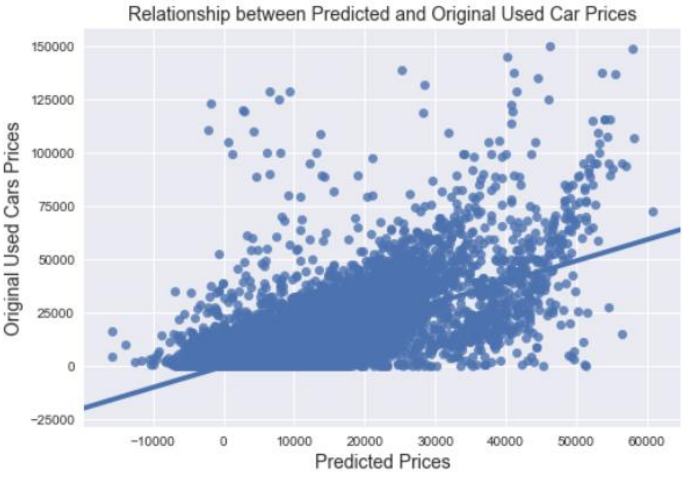


Figure 11

From the above graph, the predictions are very widely distributed along the predicted pricing axis which seems to be underpredicting the price, as the data falls below the 45-degree line. After that, I ended up by plotting residual scatter and histogram graph on test data.

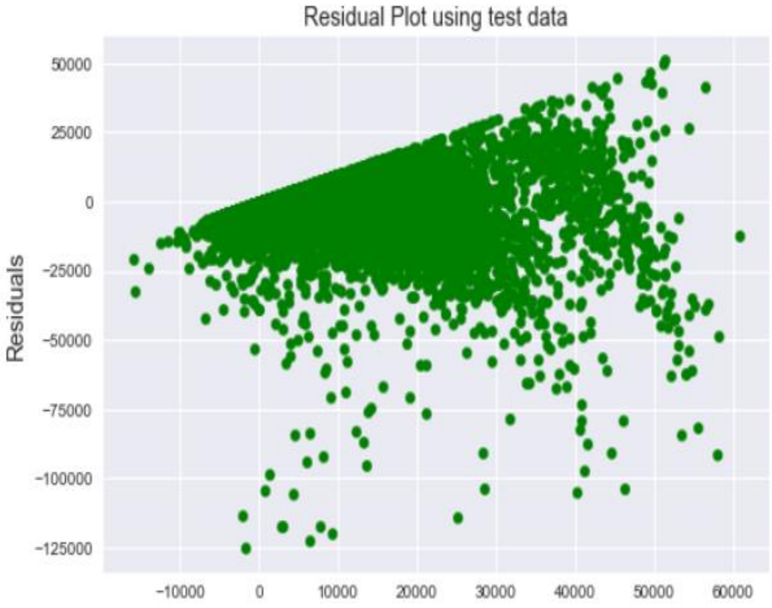


Figure 11.a

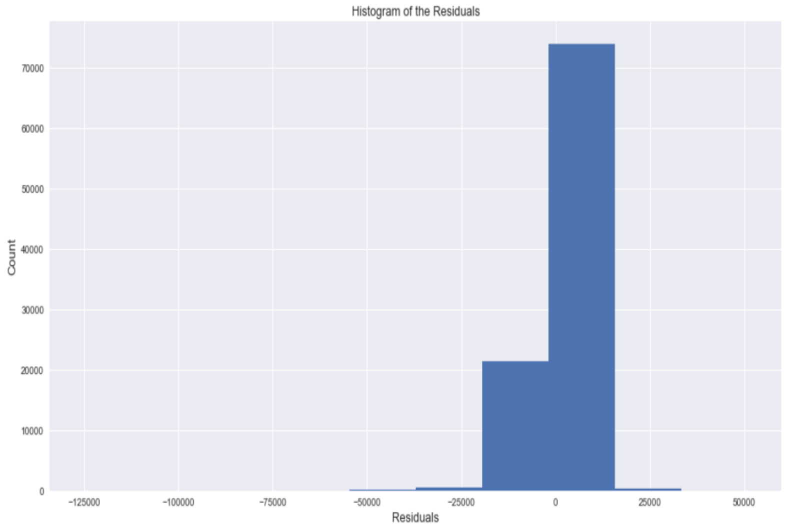
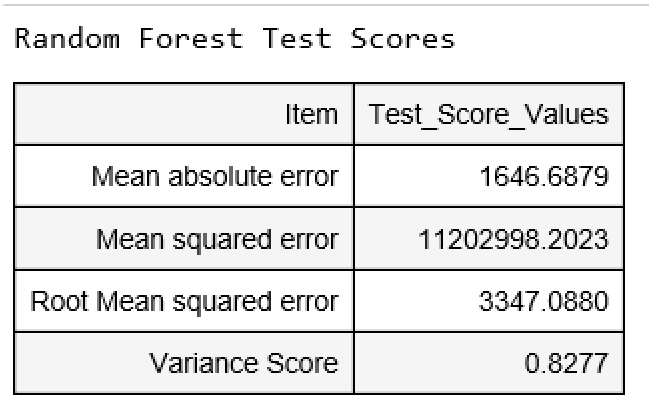


Figure 11.b

As shown in above plot (Fig. 11.a), it can be seen that the variance of error terms (residuals) was not constant. It was spreading out as the predicted value increased thereby unevenly scaling down the model’s performance. The plot showed the non-linearity in data which was not captured by the model.

Tuning the Models

In order to get a better model, I applied hyper-parameter tuning on Random Forest algorithm in order to make sure that our model is general and it works beyond our data sets. I used Grid Search CV for tuning the parameters of my model. In order to select set of hyperparameters, a good place to start with was the documentation on the random forest in Scikit-Learn among which n_estimators and max_features were the most important ones. Since I was running out of time, I tried choosing a specific range of values and see what works by adjusting a set of hyperparameters. I then computed Random Forest train and test scores based on Mean absolute error, Mean squared error, Root Mean squared error, Variance Score which is shown below: (Tbl. 11)



Random Forest Test Scores

Item	Test_Score_Values
Mean absolute error	1646.6879
Mean squared error	11202998.2023
Root Mean squared error	3347.0880
Variance Score	0.8277

Table 11

From the above table, we can see that we have increased the R-squared value 0.62 to 0.83 with minimum MSE (11202998.2023) score by removing some of the outliers initially and tuning the model by selecting the best set of hyperparameters which clearly shows the best fit from all the above models.

Considering Random Forest Regressor as the best model after tuning, I created a scatterplot between the predicted prices and the original prices and tried to evaluate this plot to see how well the model predicts the price of given the data.

From the below graph, we see very strong correlation showing the upward trend. Also, the data points were not too dispersed and lied on the 45-degree line which tends to be a perfect model. I then plotted residual plot for this model in order to assess whether the observed error is consistent with random error.

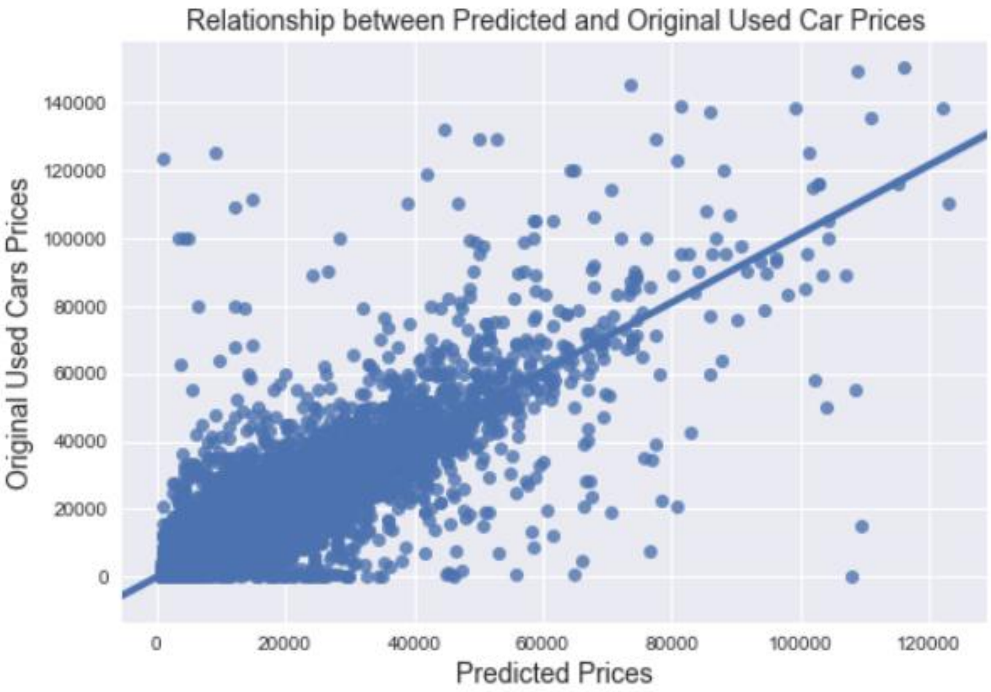


Figure 12

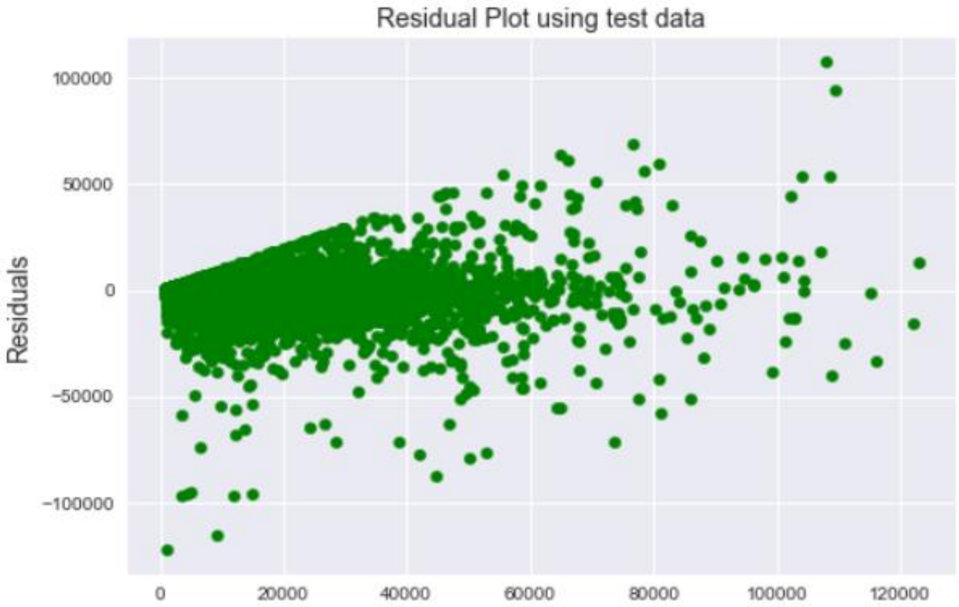


Figure 13.a

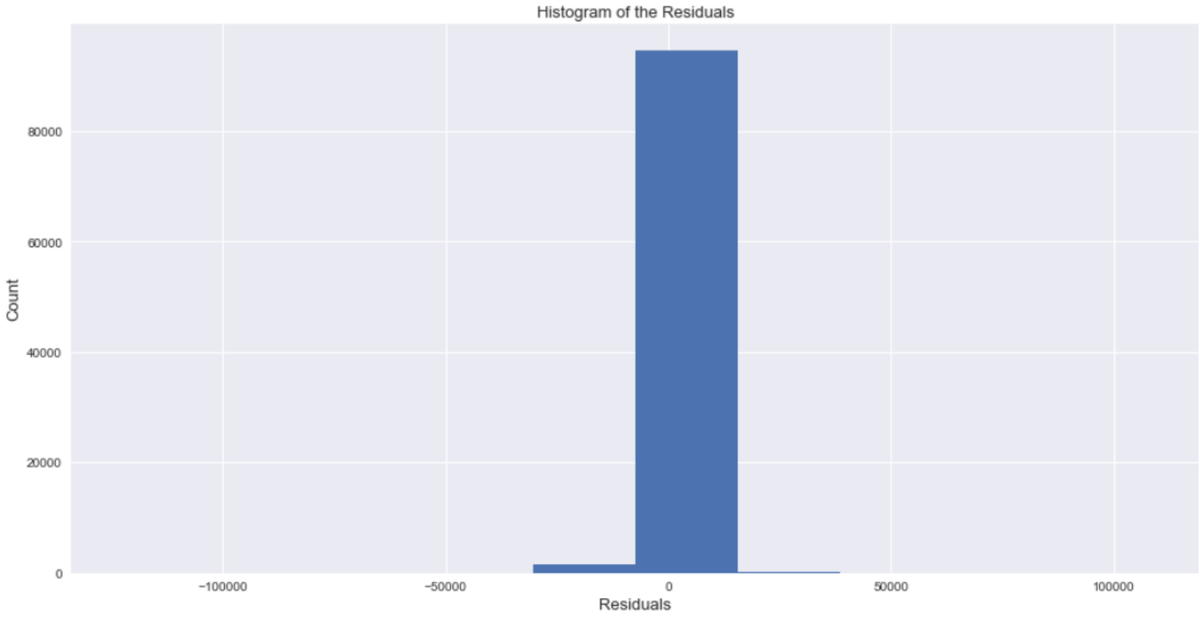
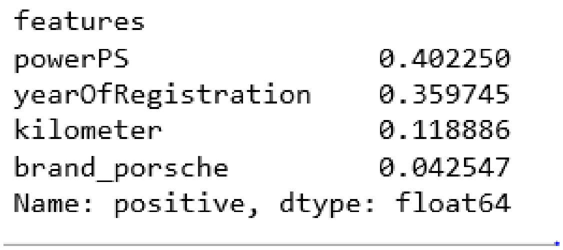


Figure 13.b

As shown in above plot (Fig. 13.a), it can be seen that the variance of error terms(residuals) is moderately constant. It is not spreading out anymore as the predicted value increases. As one said, non-constant variance arises in presence of outliers or extreme leverage values. These values get less weight, thereby increasing the model's performance. The plot shows better linearity in data than Linear Regression Model.

Feature Importance

I tried computing the relative importance of each attribute by finding the highest positive coefficient value as this could be further used to inform a feature selection process.



```
features
powerPS                0.402250
yearOfRegistration     0.359745
kilometer              0.118886
brand_porsche          0.042547
Name: positive, dtype: float64
```

Table 12

From the above feature importance analysis report (Tbl. 12), we could see that features as powerPS, yearOfRegistration, kilometer, brand_porsche with the high positive coefficients (towards price) makes some sense and are somewhat correlated to the things which influence customers to buy a used car with better information. In short, we can say that model has good understandability.

Conclusions

I was able to develop Random Forest Regression Model with a good R-squared value of 0.83 and minimum MSE (11202998.2023) which will help the client to use machine learning model to predict the price of used cars now on specific features which can help customers to take better decision for buying used cars. After doing exploratory data analysis I could suggest different useful findings on different features which helps to understand used car prices.

Recommendations for the Client

Above findings can help the client in different aspects and can become instrumental in providing better pricing of the used cars to the customer. Exploratory Data Analysis in giving some interesting information from used car data that can help to have a better understanding of the market e.g. Min, Max and Average price of the car of different brands, Most popular brands, Most sold car etc. This will be really beneficial to the buyer and seller. Feature Importance is another useful finding which can help in knowing the top features contributing most to the pricing of the car.

Model created will be useful to predict the price of the new data of used car. Since this model is based on multiple features, it can help in finding a better price which will be useful to both buyer and seller and since this is in the interest of the consumer it may drive more purchase on the site.

The client should implement this model and based on new data, the model will learn and do better prediction over a period of time.

Further Research

1. Since I was running out of time, I tried choosing a specific range of values for tuning the model using Random Regressor which is not a good approach to start with. I would rather try to use two things for tuning the model for my future work.

   a. Random Search CV which will able to narrow the range of values for each hyperparameter by randomly finding the best parameters. I will try using a wide range of values and see what works by adjusting a set of hyperparameters.
   b. I will then apply Grid Search CV around the above-obtained parameters and will find the best parameter. I will compare the model performance using Random Search CV parameters and Grid Search CV parameters.

2. I will try to use advanced techniques as **Gradient Boosting Regression** for my car dataset which is a really brilliant tool to enable better predictions.

Resources Used

- http://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html
- http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html
- https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74
- http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.GridSearchCV.html#sklearn.grid_search.GridSearchCV
- http://scikit-learn.org/stable/auto_examples/ensemble/plot_ensemble_oob.html#sphx-glr-auto-examples-ensemble-plot-ensemble-oob-py
- http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
- https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf
- http://localhost:8888/notebooks/Mini_Project4_Linear_Regression/linear_regression/Mini_Project_Linear_Regression.ipynb