Group 11 – Machine Learning Project Report

By Giuliano Silva (sba20130), Yorgos Tsavidis (sba20298)
and James Smith (sba20133)

Machine Learning
Higher Diploma in Science in Data Analytics for Business
Dr Muhammad Iqbal

CCT College Dublin

10th December 2020

Table of Contents

Group 11 – Machine Learning Group Project Report

Domain Problem and Project Objectives

The number of privately registered cars has dramatically increased in the first few decades of the 21st Century (Pudaruth, 2014; Gegic et al., 2019). This has also led to an increase in the purchase of second-hand cars and a more competitive second-hand car market (Pudaruth, 2014). One consequence of this is that first-hand car buyers are very conscious about the resale value of their cars (Pudaruth, 2014), and will make a decision accordingly. Furthermore, second-hand car buyers may be exploited by car sellers who take advantage of the increase in demand for previously owned cars. Buyers are typically overwhelmed when looking to buy a used car, as there are a multitude of available cars with different ages, brands, mileages, owner history and of course, prices. It is often difficult to determine a fair price for the cars on offer as one would have to factor in multiple attributes. There is therefore a need for a fair and competitive price prediction system for vehicles based on selected features in common with other cars in the marketplace (Pal et al., 2019). However, accurate car price prediction can be difficult, often involving domain expertise, because the price depends on many factors, including brand, age, fuel type, horsepower, and mileage (Gegic et al., 2019).

Applying a supervised machine learning model to the evaluation of used car prices is a reliable solution to this domain problem. Gegic et al. (2019) even built a GUI for car buyers that allows customers to enter their details and have a car price returned based on their inputs, with a machine learning model running in the background. Gegic et al. (2019) used three different algorithms to predict cheap, moderate and expensive car prices. Utilising an Artificial Neural Network (ANN), Support Vector Machine (SVM) and a Random Forest (RF) Classification model produced predictive models with an accuracy of 86%, 90% and 89% respectively. Similarly, Pal et al. (2019) predicted the price of cars in a Kaggle dataset with 95% accuracy using an RF Regression algorithm. Puteri and Safitri (2020) built a 14 feature linear regression model that predicted car sales in the Indonesian market with an accuracy of 75%. The K-Nearest Neighbor (KNN) algorithm has also proved to be an accurate predictor of car price, with Samruddhi and Ashok Kumar (2020) proposing a KNN regression model that is 85% accurate. Certain car features also emerge from the literature as

better predictors of car price, including vehicle age. distance travelled, colour, car brand, vehicle type, transmission, and selling location (Pal et al., 2019; Puteri and Safitri, 2020).

Based on the literature, and how successful machine learning models are at predicting car prices, the objective of this study was to build a machine learning model that predicts car prices with an acceptable accuracy that customers would find useful and reliable. The researchers chose three algorithms to build and compare, based on their success in previous research: KNN Regression, Linear Regression, and Random Forest. The study also aimed to determine which of the car features included in our models had the biggest relationship with price, either positively or negatively.

Dataset Selection

Initially, several datasets were put forward and examined as candidates for this project. The researchers aimed to find a dataset where regression models could be applied. This required a continuous target variable.The proposed datasets were:

- AirBnB (Seth, 2020): A compilation of multiple datasets found on Inside Airbnb (Inside Airbnb, n.d.). It contains short term stay listings in different regions of the USA. This was a promising dataset that satisfied the requirements set for our project in terms of both entries and features. However, initial analysis and testing on this dataset produced poor results in terms of price prediction and the dataset was subsequently dropped as a candidate. One difficulty is that the prices for Airbnb lettings constantly change over time due to various factors, thus we were not able to find meaningful patterns. The dataset is also quite skewed and contains extreme outliers, making regression analysis difficult.
- Kuala Lumpur (Property Listings, 2019): A dataset of house rental listings in Kuala Lumpur, Malaysia scraped from a property listing website. It features more than 53,000 rows but only 8 columns which was insufficient based on our assessment criteria.
- World Happiness Report (World Happiness, 2019): This dataset had a limited number of both rows and columns and was deemed generally unsuitable for the needs of our project.

- Used-cars-catalog (Lepchenkov, 2019): This data was scraped from a Belarussian car ads website. It has more than 38,000 rows and 30 features in a combination of numerical and categorical data. It was decided that this would be our fallback dataset.

Ultimately, we opted for a Kaggle dataset (Birla, 2020) containing information about used cars listed on www.cardekho.com, an Indian car listing website. With thirteen columns and over eight thousand rows, the dataset was deemed suitable for exploratory data analysis and applying machine learning models. This report refers to the chosen dataset as 'cars' from hereinafter.

Data Preparation

1. Data Understanding and Exploration Data Analysis

Initially, the cars dataset had 8128 rows and 13 features. Car price was chosen as the target variable for our analysis. Table 1 below lists the column names and variable type.

Column Name/Variable	Type
name	object
year	int
selling_price	int
km_driven	int
fuel	object
seller_type	object
transmission	object
owner	object
mileage	object
engine	object
max_power	object

torque	object
seats	float

Table 1. Cars dataset. Original column names and variable types

Table 1 illustrates that the dataset contained four continuous variables and nine categorical variables. Table 2 below shows values in the first five rows of the initial cars dataset

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_power	torque	seats
0	Maruti Swift Dzire VDI	2014	450000	145500	Diesel	Individual	Manual	First Owner	23.4 kmpl	1248 CC	74 bhp	190Nm@ 2000rpm	5.0
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	Diesel	Individual	Manual	Second Owner	21.14 kmpl	1498 CC	103.52 bhp	250Nm@ 1500-2500rpm	5.0
2	Honda City 2017-2020 EXi	2006	158000	140000	Petrol	Individual	Manual	Third Owner	17.7 kmpl	1497 CC	78 bhp	12.7@ 2,700(kgm@ rpm)	5.0
3	Hyundai i20 Sportz Diesel	2010	225000	127000	Diesel	Individual	Manual	First Owner	23.0 kmpl	1396 CC	90 bhp	22.4 kgm at 1750-2750rpm	5.0
4	Maruti Swift VXI BSIII	2007	130000	120000	Petrol	Individual	Manual	First Owner	16.1 kmpl	1298 CC	88.2 bhp	11.5@ 4,500(kgm@ rpm)	5.0

Table 2. First 5 Rows of Cars dataset

The researchers also examined the central tendencies of the continuous variables. Although containing some skewness, the mean and mode of each of the variables year, selling_price, km_driven and seats were similar. However, generated histograms indicate that the data was not normally distributed, as shown in Figure 1.

	year	selling_price	km_driven	seats
count	8128.000000	8.128000e+03	8.128000e+03	7907.000000
mean	2013.804011	6.382718e+05	6.981951e+04	5.416719
std	4.044249	8.062534e+05	5.655055e+04	0.959588
min	1983.000000	2.999900e+04	1.000000e+00	2.000000
25%	2011.000000	2.549990e+05	3.500000e+04	5.000000
50%	2015.000000	4.500000e+05	6.000000e+04	5.000000
75%	2017.000000	6.750000e+05	9.800000e+04	5.000000
max	2020.000000	1.000000e+07	2.360457e+06	14.000000

Table 3. Central Tendencies of the 4 Continuous Variables in initial Cars Dataset

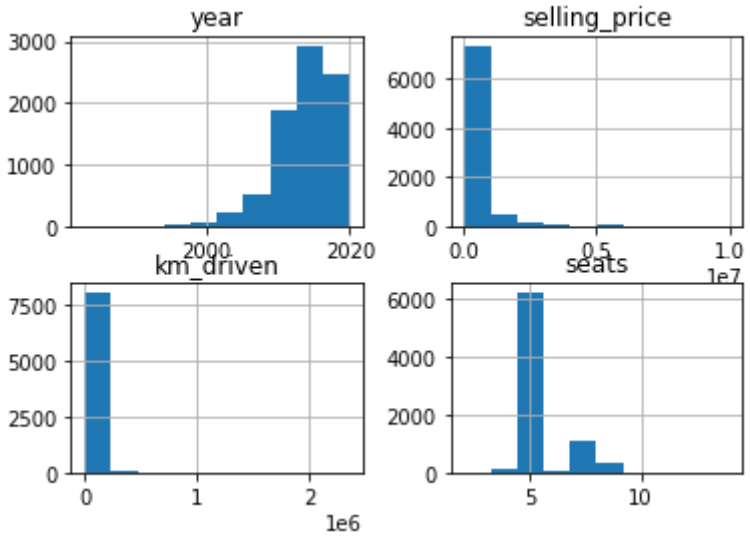


Figure 1. Histograms for year, selling_price, km_driven and seats.

The researchers also examined each of the nine categorical variables. Some variables contained a considerable number of classes.

Variable	Class Sample	No. of feature classes
name	Maruti Alto LXi	1,982
engine	1248cc	121
max_power	67 bhp	320
torque	190Nm@ 2000 rpm	441
seller_type	Dealer	3
fuel	Diesel	4
transmission	Manual	2
owner	First Owner	5
mileage	19.7 kmpl	393

Table 4. Categorical variables of initial cars dataset, with sample classes

As it was the target variable in this study, the researchers examined the distribution of price in more detail. Figure 2 below illustrates that the car price values distribution was positively

skewed, and leptokurtic in shape. The researchers noted that this would have to be addressed for a reliable linear model to be built.

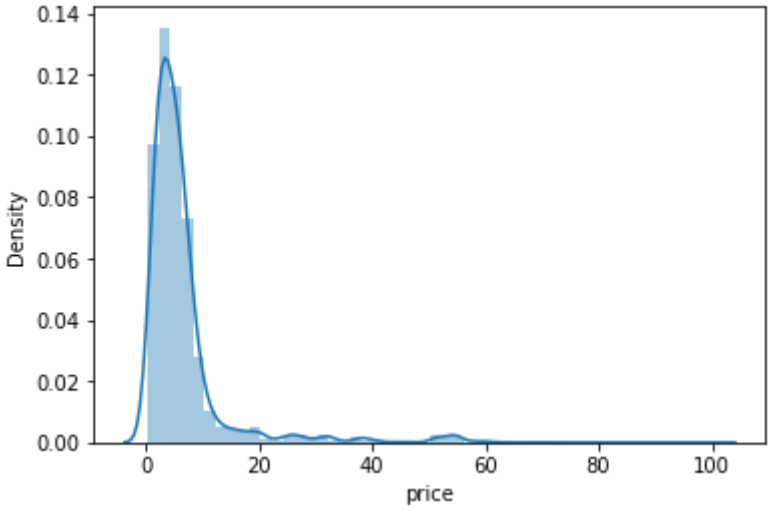


Figure 2. Original car price distribution

The researchers also examined the feature relationships with car price. Automatic cars had a higher selling price than manual cars.

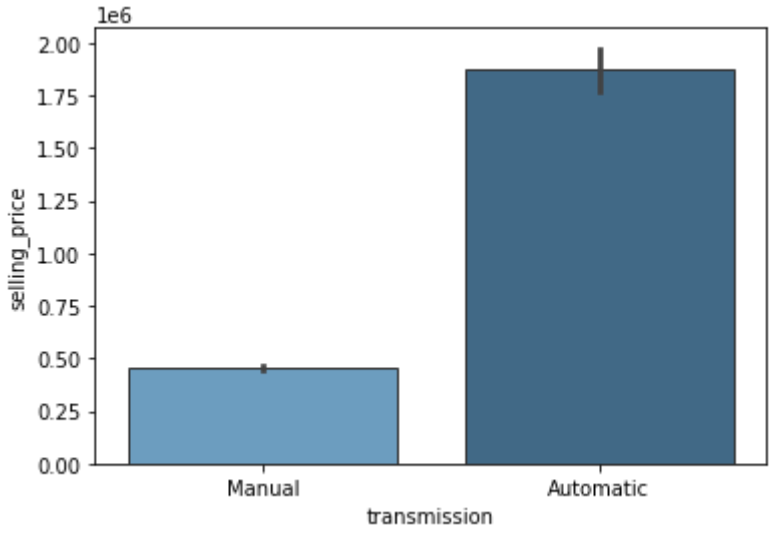


Figure 3. Car transmission and price

Max_power is positively correlated with car price, particularly in the 50 to 200 bhp range. Faster cars are more expensive as expected.

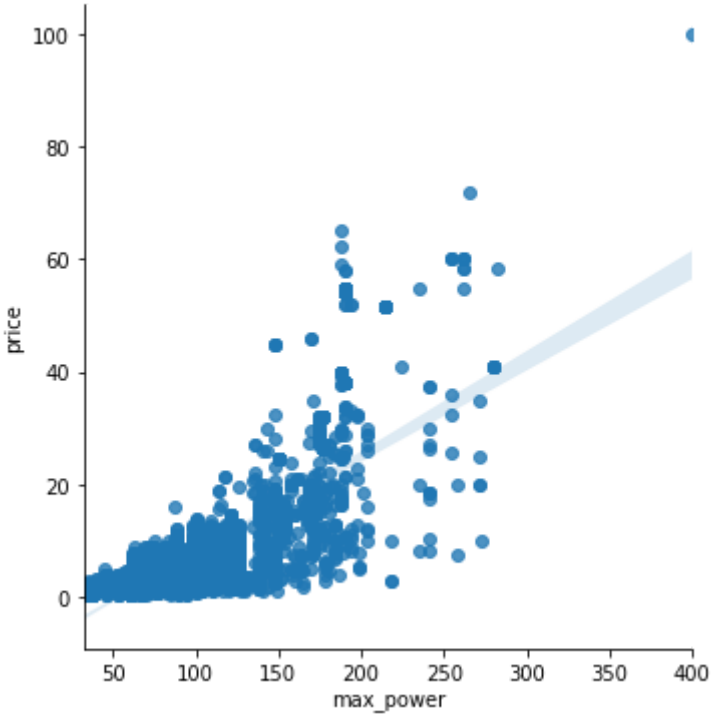


Figure 4. Max power (bhp) of cars and price

Cars with more kilometres on the clock sell for a lower price than cars with a lower mileage. After 500,000 kilometres, the price drops dramatically.

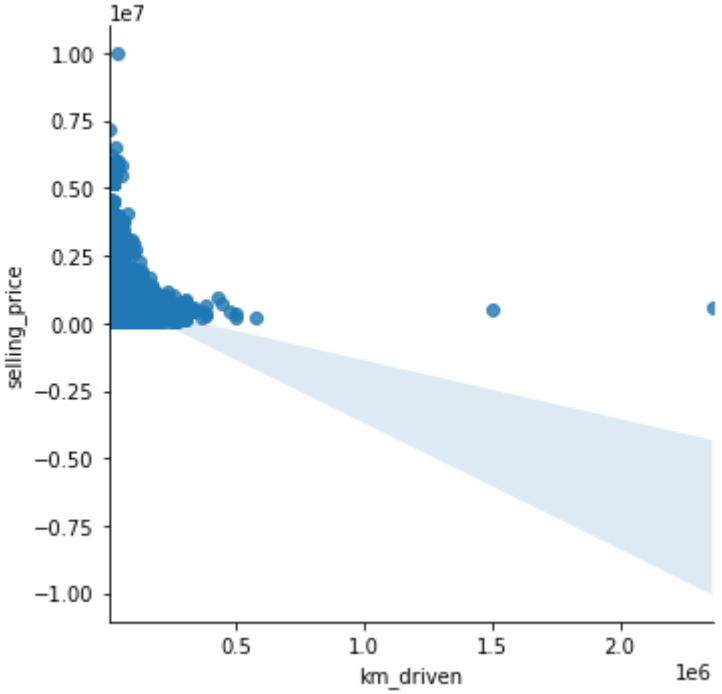


Figure 5. Mileage (Km driven) and car price

The age of the car is also correlated with price, with the price of cars dropping steeply after 5 years.

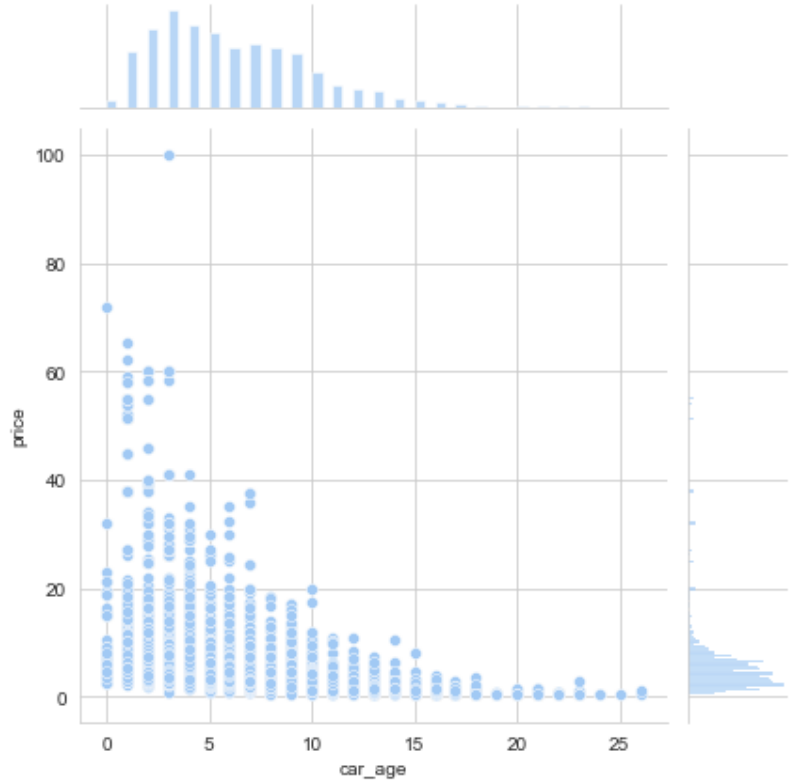


Figure 6. Age of car and car price

Finally, another insight from the data is that the more owners a car has had, the lower the resale price.

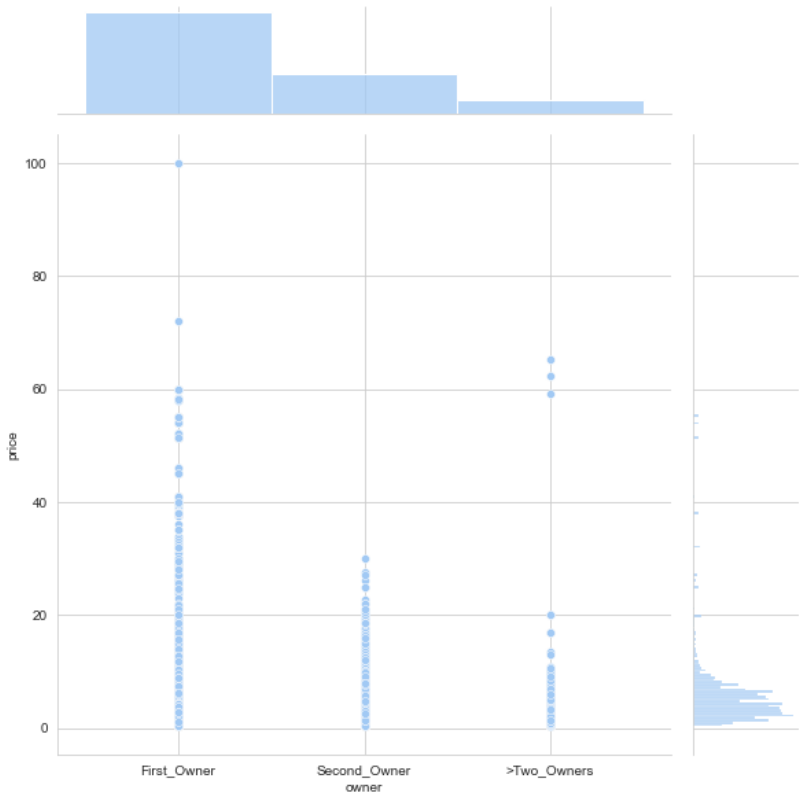


Figure 7. Number of owners and car price

Our initial exploration of the cars dataset confirmed that following data cleaning and preparation, our dataset would be suitable for inclusion in regression models. We aimed to use a linear regression model to regress price on each of the features, and compare the accuracy with additional regression models, selecting the most suitable.

2. Data Cleaning / Data Preparation / Feature Engineering

The first step in our data preparation was to examine the number of missing values in the dataset. We found five of the features had missing values: torque (222), seats (221), engine (221), mileage (221) and max_power (215). Since the number of missing values was not significant, we dropped these rows from the dataset.

We then dropped the feature torque as the values were a combination of strings and numbers, making this feature difficult to use for analysis. In addition, torque is calculated using horsepower, as is max_power, therefore the researchers were confident that inclusion of max_power alone would suffice for car price prediction.

There was a significant amount of feature engineering required to produce a suitable dataset for regression analysis. We removed the ‘cc’ string value from the engine feature so that we could convert the values into floats. Strings ‘kmpl’ and ‘Km/kg’ were removed from the mileage variable, and ‘Bhp’ was removed from the max_power variable, also making these features numeric. In addition, the researchers created a new continuous variable ‘car_age’ by subtracting the vehicle age from the current year, e.g., 2020 (current year) - 2015 (year) = 5 (car_age). The categorical variable year was then dropped from the dataset to avoid redundancy.

The feature ‘name’ contained a large amount of classes. To address this we used high level domain knowledge to create a new variable car_brand, a binary category with classes domestic and international. The researchers felt that this was an appropriate method of handling this data as approximately 2,367 (30%) cars belonged to the Maruti brand alone.

The owner feature originally had five classes, and the researchers decided to recode this category so that only three classes remained; first owner, second owner, and more than two owners. Exploratory data analysis had revealed that price prediction had the biggest relationship with first and second car ownership, after which the price drops considerably.

The researchers examined the target variable price. As the car prices were in Indian rupees, the values were divided by 10,000 in an attempt to standardise the means for each variable and allow for easier understanding of regression results.

To include the categorical variables in further analysis, the researchers would be required to recode the variables car_brand, engine, fuel, seller_type, transmission and owner as dummy variables based on the assumptions of the algorithm.

Following this data cleaning and feature engineering, we converted some features into new datatypes, resulting in the following dataset:

Column Name/Variable	Type
car_brand	object
car_age	int

selling_price	int
km_driven	float
fuel	object
seller_type	object
transmission	object
owner	object
mileage	float
engine	float
max_power	float
torque	object
seats	float
price	float

Table 5. Cars dataset following initial data cleaning and feature engineering

Next, the researchers standardised each of the continuous features in the dataset using the StandardScaler function, which removes the mean of each feature and scales to unit variance, resulting in a standard deviation of approximately 1 across each feature (Pedregosa et al., 2011). This standardisation resulted in the following feature central tendency values.

	km_driven	mileage	engine	max_power	seats	car_age	price
count	7.363000e+03	7.363000e+03	7.363000e+03	7.363000e+03	7.363000e+03	7.363000e+03	7.363000e+03
mean	-1.592280e-17	3.724969e-16	1.447527e-17	1.544029e-17	1.254523e-16	1.930036e-18	-1.930036e-17
std	1.000068e+00	1.000068e+00	1.000068e+00	1.000068e+00	1.000068e+00	1.000068e+00	1.000068e+00
min	-1.248278e+00	-4.944929e+00	-1.686951e+00	-2.007149e+00	-3.545125e+00	-1.621979e+00	-1.626088e+00
25%	-6.395137e-01	-6.661804e-01	-4.426558e-01	-6.546227e-01	-4.298659e-01	-8.450205e-01	-8.072794e-01
50%	-1.177012e-01	1.338547e-02	-3.319071e-01	-1.232182e-01	-4.298659e-01	-6.806171e-02	-1.373479e-01
75%	4.910800e-01	7.281881e-01	2.109789e-01	5.211531e-01	-4.298659e-01	7.088970e-01	6.070205e-01
max	3.980890e+01	3.471621e+00	4.554067e+00	7.183882e+00	8.915910e+00	5.111663e+00	3.807804e+00

Table 6: Standardised features (s ≈ 1)

Finally, by agreeing a cut-off value of 15 (based on the newly standardised values), the researchers dropped 543 rows with outlier price values, leaving 7363 rows in the cars dataset. The price variable then had a new distribution, as per Figure 8 (for comparison, see Figure 2).

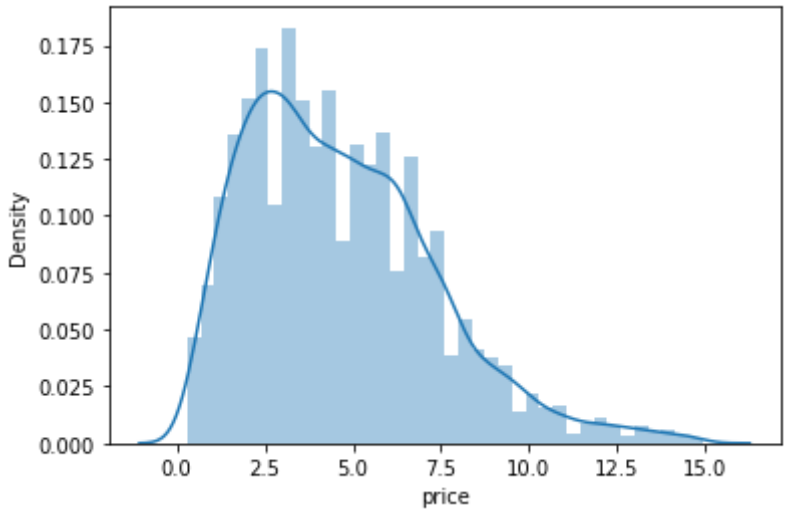


Figure 8. Price variable following removal of outliers and normalisation

Table 7 illustrates the finalised dataset prior to the regression analyses.

	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_power	seats	car_brand	car_age	price
0	1.282496	Diesel	Individual	Manual	First_Owner	0.944642	-0.331907	-0.424078	-0.429866	Domestic	-0.068062	-0.062911
1	0.838955	Diesel	Individual	Manual	Second_Owner	0.375821	0.210979	0.710199	-0.429866	International	-0.068062	-0.360658
2	1.186830	Petrol	Individual	Manual	>Two_Owners	-0.489997	0.208807	-0.270382	-0.429866	International	2.003828	-1.149689
3	0.960711	Diesel	Individual	Manual	First_Owner	0.843966	-0.010519	0.190706	-0.429866	International	0.967883	-0.900325
4	0.838955	Petrol	Individual	Manual	First_Owner	-0.892702	-0.223330	0.121543	-0.429866	Domestic	1.744842	-1.253900
...	...	...	...	...	...	...	...	...	...	...	...	...
7901	0.665017	Petrol	Individual	Manual	First_Owner	-0.288644	-0.442656	-0.084026	-0.429866	International	0.190925	-0.546750
7902	0.821561	Diesel	Individual	Manual	>Two_Owners	-0.716519	0.200121	0.959187	-0.429866	International	1.744842	-1.235291
7903	0.838955	Diesel	Individual	Manual	First_Owner	-0.087291	-0.331907	-0.427921	-0.429866	Domestic	1.226870	-0.315996
7904	-0.813451	Diesel	Individual	Manual	First_Owner	0.987430	-0.010519	-0.577775	-0.429866	Domestic	0.190925	-0.658406
7905	-0.813451	Diesel	Individual	Manual	First_Owner	0.987430	-0.010519	-0.577775	-0.429866	Domestic	0.190925	-0.658406

Table 7. Cars dataset after EDA, data cleaning and feature engineering

3. Data preparation

The standard recommendation in machine learning is to split train and test sets into 90:10 (10%) and 80:20 (20%) sets respectively (Little et al., 2017). In addition to these train-test splits, and as per the assessment criteria, the researchers also built models using a 70:30 (30%) test split for each algorithm. Where appropriate, the researchers also employed k-fold

cross-validation. Cross-validation attempts to avoid over-fitting of training models and ensure good model generalisation by dividing the dataset into train and test subsets. The purpose of cross-validation is therefore to achieve a "stable and confident estimate of model performance" (Reitermanova, 2010). K-fold cross-validation uses a combination of tests to gain a stable estimate of the model error by dividing the dataset into k-1 parts for training, and the remaining part for testing. This process is repeated for each data part to gain an estimate of model performance (Reitermanova, 2010). The model chosen by the researchers for each algorithm was the one that achieved the highest accuracy.

The cars dataset had the following features and target variables for regression analysis:

- Features/Independent Variables - km_driven, mileage, engine, max_power, seats, car_age, seller_type, fuel, transmission, owner_type and car_brand
- Target/Dependent Variable - price

Model Building and Evaluation

1. Multiple Linear Regression

The relationship between a dependent variable and multiple independent variables can be modelled using a multiple linear regression (MLR) model, which can be written as the following equation:

$$y = b_0 + b_1x_1 + b_2x_{2...} + b_nx_n$$

where y is the dependent variable, b_0 is the intercept and each independent variable is represented in the format b_nx_n, with x_n representing an independent variable, and b_n the

regression coefficient in the n^{th} dimension for that variable (Uras et al., 2020). For our analysis, target variable price was regressed on each feature in the cars dataset.

Ruginski (2016) explains that although the MLR is a robust model, there are a number of assumptions that must be met in order to avoid biased results, including:

1. A linear relationship exists between the target variable and each feature variable
2. Homoscedasticity, i.e., there is constant variance across the range of each feature variable
3. Normality of the residuals
4. No multicollinearity (very high correlation) between the feature variables
5. No outliers

Although the researchers were only interested in the predictive accuracy of the linear regression, they felt that testing whether the MLR model met the above assumptions was warranted. Firstly, the relationship between car price and continuous features km_driven, mileage, engine, max_power, seats, and car_age, by generating scatterplots (refer to Appendix 1). The scatterplots indicated that not all features were linearly related to price. Next, the model residuals were examined to identify whether the data met non-linearity, outliers, homoscedasticity, and normal distribution assumptions. The researchers then examined the dataset residuals. Appendix 4 includes the residual plots. Actual residual values were compared to fitted values which indicated that although not plotting on a straight line, there was a positive relationship.. Comparing the fitted values against the residuals revealed that one or more of the model features had a non-linear relationship with the target variable price. A QQ plot comparing the theoretical and sample quantiles indicated that the model

residuals were not normally distributed, given the data deviation. A scale-location plot revealed that the residual variance was not equal, indicating homoscedasticity in the dataset. Next, we tested the residuals for normality by examining both skewness and kurtosis; an approach developed by D'Agostino (1971) and D'Agostino and Pearson (1973). The stats.normaltest function was used to carry out the test. The normality test indicated that the residuals were not normally distributed (t = 665.74, $p < 0.001$). The Breusch Pagan test also indicated that the residuals did not exhibit equal variance (LM = 1558.22, $p<0.001$; F = 159.10, $p<0.001$).

To test for feature multicollinearity, Pearson correlation coefficient values were calculated to examine the relationship between each feature variable (refer to Appendix 2 for correlation matrix). It was determined that several features were significantly correlated, with *r* values greater than 6.0, i.e., max_power and engine ($r = 0.64$), seats and engine ($r = 0.68$), dummy variables fuel_Diesel and fuel_Petrol (generated from fuel feature) ($r = -0.98$), and dummy variables owner_First_Owner and owner_Second_Owner (generated from owner feature) ($r = -0.81$). To address this multicollinearity issue, it was decided that the engine feature should be dropped from the MLR model. In relation to multicollinearity between the dummy variables, Allison (2012) argues that multicollinearity between dummy variables may be a necessary consequence of including a categorical variable in a MLR model, but that this will not affect other features in the model. Therefore, the dummy variables generated for features fuel and owner were included in the regression analysis. Following the removal of engine from the model, no feature correlation (excluding the dummy variables) exceeded an *r* value of 6.0. Although our dataset failed to meet the MLR assumptions, the researchers felt that given the robustness of this model, regressing price on 13 feature variables could be achieved, and a reliable price prediction accuracy achieved.

We initially regressed each feature on target variable price using simple splits of 30, 20 and 10% split. The R^2 test model accuracy for each split was 0.693, 0.687, and 0.688 respectively. Having chosen the 30% split as the most accurate model, we generated a new 13 feature MLR model with the engine feature removed to address multicollinearity. The resulting model and coefficients can be written using the following equation:

price = - 0.01 - 0.05 (km_driven) - 0.04 (mileage) + 0.40 (max_power) + 0.07 (seats) - 0.51 (car_age) - 0.13 (seller_type_Individual) - 0.29 (transmission_Manual) + 0.00 (car_brand_International) + 0.45 (fuel_Diesel) + 0.25 (fuel_LPG) + 0.08 (fuel_Petrol) + 0.16 (owner_First_Owner) + 0.01 (owner_Second_Owner)

We can determine the following from the above linear regression equation:

1. Max_power (0.40) and fuel_Diesel (0.45) and car_age(-0.51) contribute most to our regression model. As one would expect, the older the car, the lower the price.
2. Cars having only one previous owner can demand a higher resale price in c comparison to cars with multiple owners (owner_First_Owner = 0.16).
3. Dealerships are selling cars at higher prices compared to individuals (seller_type_Individual = -0.13).
4. Diesel cars are the most popular in terms of fuel type (fuel_Diesel = 0.45).
5. Cars with automatic transmissions tend to be more expensive than manual transmission cars (transmission_Manual = -0.29)
6. Fuel efficiency (mileage = -0.04) and actual car mileage (km_driven = -0.05), albeit negatively related to price, do not have a dramatic impact on car resale prices.

Figure 9 visualises actual versus predicted car prices, based on the most accurate MLR model.

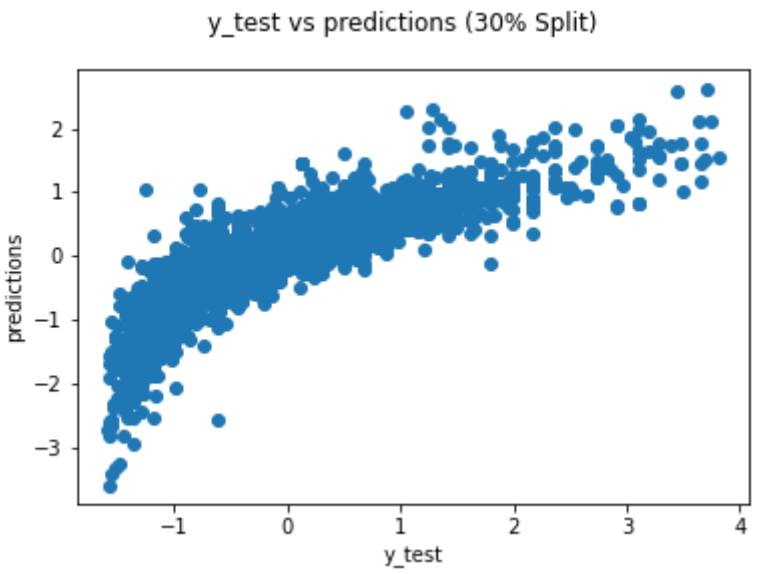


Figure 9. Predicted versus Actual price values

Finally, the researchers performed cross-validation on the multiple linear regression model using k-fold cross validation on 30, 20 and 10% splits, in an attempt to achieve a better accuracy than our 30% simple split model. The regression model results with the engine feature removed are presented in Table 8 below.

Split Type	N Features	Train Set Accuracy (R^2)	Test Set Accuracy (R^2)	Test Set Accuracy (Adj. R^2)	Test Set MAE[1]	Test Set MSE[2]	Test Set RMSE[3]
30% Simple	14[4]	0.695	0.694	0.692	0.415	0.309	0.555
20% Simple	14[4]	0.697	0.687	0.685	0.413	0.306	0.553
10% Simple	14[4]	0.696	0.688	0.682	0.402	0.278	0.523
30% Simple	13	0.693	0.691	0.689	0.418	0.311	0.558
30% K-fold	13	0.685	0.690	-	-	-	-
20% K-fold	13	0.688	0.676	-	-	-	-
10% K-fold	13	0.689	0.676	-	-	-	-

1. Mean Absolute Error. 2. Mean Squared Error. 3. Root Mean Squared Error 4. Engine feature not removed, introducing multicollinearity

Table 8. Multiple Linear Regression Model - Price Prediction Accuracy Results

Table 8 suggests from both simple splitting and k-fold cross-validation that a 30% split produces the most accurate multiple regression model and car price predictions. If larger train datasets are used, the model begins to overfit the data. Interestingly, the 14 feature model using a simple 30% split returns the highest test set accuracy of 69.4%, despite having multicollinearity issues with the inclusion of the engine feature. Despite the higher accuracy of the former, the researchers recommend using the 30% simple split regression model (69.1%) with 13 features, where the feature engine has been dropped from the dataset.

2. K-Nearest Neighbours Regression

The K- Nearest Neighbours (KNN) algorithm is a simple but effective method used to solve both regression and classification problems. It predicts the target value (regression) or class (classification) using 'feature similarity', i.e., test dataset points are assigned a value based on how closely they resemble points in the training dataset (Singh, 2018). The distance between train and test points are calculated differently depending on whether the problem is one of regression (Euclidean or Manhattan) or classification (Hamming) (Singh, 2018). The python function KNeighborsRegressor (Pedregosa et al., 2011) uses a Minkowski metric which defaults to the Euclidean distance function, give as:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

p, q = two points in Euclidean n-space

q_i, p_i = Euclidean vectors, starting from the origin of the space (initial point)

n = n-space

The KNN Regression algorithm is simple in that one only needs to define the number of nearest neighbours (n) and change the train and test data percentages (K-Nearest Neighbors, 2020). In practice, that means that a lower n value will typically achieve very high train scores and low test scores (due to overfitting) whereas a higher number of n will yield a better balance between train and test scores, generalising better.

The researchers built a KNN Regression model using the standardised cars dataset, creating one-hot encoded dummy variables for each categorical feature, meaning the model had 19 features predicting target variable price. Initially, regression models were built using a 30% split train and test dataset, and over a range of n = 1 to 30 neighbours. Figure 10 below illustrates that the model performed best between n = 1 and 5 neighbours. Accuracy (R^2) scores were generated for this range, and the results are presented in Table 9.

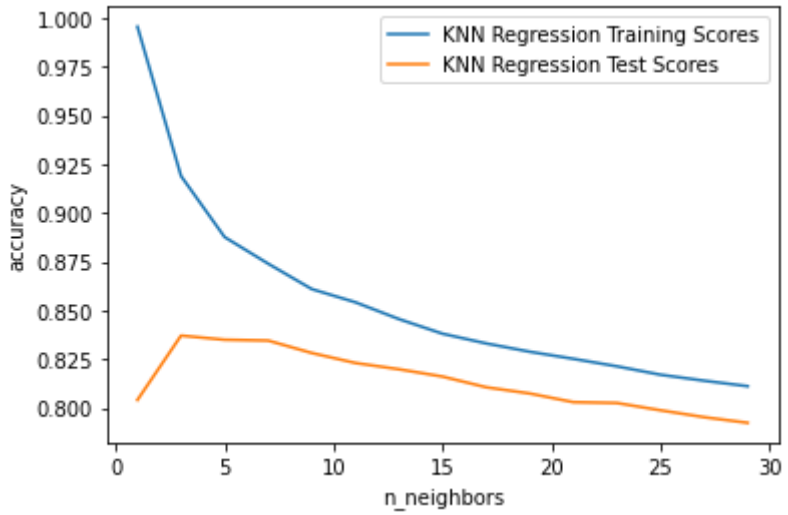


Figure 10. KNN Regression Accuracy - 1 to 30 neighbours

N neighbours	Training Accuracy	Test Accuracy
1	0.996	0.804
2	0.945	0.832
3	0.919	0.8370
4	0.901	0.8372
5	0.888	0.835

Table 9. KNN Regression Train and Test Accuracy Scores (30% Split)

As per the above results, the regression model performed best at n = 4 neighbours, with an accuracy of 84%. Model overfitting resulted in lower accuracy scores from n = 1 to 3, and when n was increased from 4 to 5, the model began to underfit the training data and produce poorer test scores.

The researchers then compared the above KNN regression model (30% split), where n = 4, with splits of 20% and 10%. The resulting test accuracies for the 20% and 10% splits were 0.842 and 0.843 respectively; both outperforming the 30% split, with the 10% split suggesting better generalisation from the training set to the test set in comparison to the other models.

The researchers then examined the 10% split KNN Regression model in more detail, using the test features to predict price values. Figure 10 outlines visually how well the test model predicts car prices (84%), with Figure 11 examining the distribution of the model residuals.

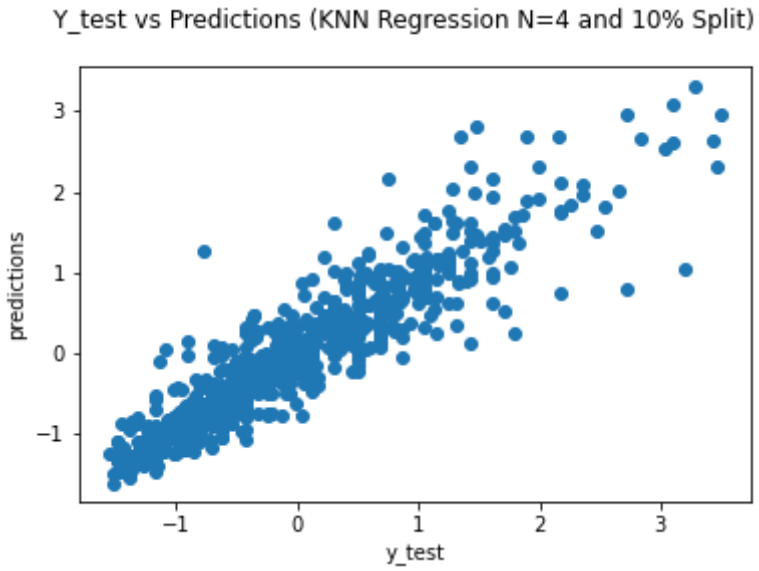


Figure 10. KNN Regression Model (10% Split) - Test Dataset Price Predictions

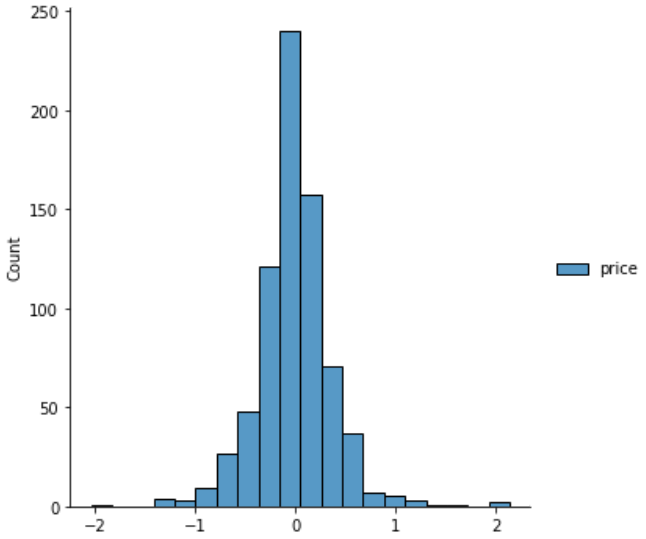


Figure 11. KNN Regression Residuals (10% Split)

An overview of the most accurate KNN Regression model is outlined in Table 10 below.

Test Split	N neighbours	Training Accuracy	Test Accuracy	MSE[1]	RMSE[2]

10%	4	0.909	0.843	0.139	0.374

1. Mean Squared Error. 2. Root Mean Squared Error

Table 10. KNN Regression Results for 10% Split with n=4 Neighbours

3. Random Forest Regression

The researchers determined from linear regression analysis that the cars dataset is not normally distributed. To overcome this issue, the researchers also applied a Random Forest (RF) regression to the dataset. RF belongs to the category of ensemble methods, and can be used for classification and regression problems. The algorithm was developed to reduce overfitting seen in other decision tree algorithms (Gegic et al., 2019). RandomForestRegressor specifically is a powerful algorithm that divides the dataset into a subset of samples and generates multiple decision trees based on the mean prediction (Pedregosa et al., 2011). The algorithm can be tuned in terms of number of estimators (decision trees) and maximum features in a single decision tree (Pal et al., 2019). The researchers did not amend the default number of decision trees (100) during analysis or any hyperparameters as the model performed well in terms of both train and test datasets, indicating good generalisation and avoiding model overfitting. The researchers chose RF over boosting algorithms such as XGBoost and Gradient Boosting because RF decreases the variance in the model prediction values by generating multi-sets of the training set, potentially overcoming the issues met when applying the MLR model (Garg, 2018).

RF models were built using a 30, 20, and 10% split, producing models with test set accuracies of 0.898, 0.895 and 0.895 respectively, indicating that a 30% split had less train set overfitting and produced the most accurate model. The researchers also examined the

importance of each feature in the model. In order of significance, car_age (0.41), max_power (0.31), engine (0.10), mileage (0.5) and km_driven (0.04) were the most important price prediction features in the RF model.

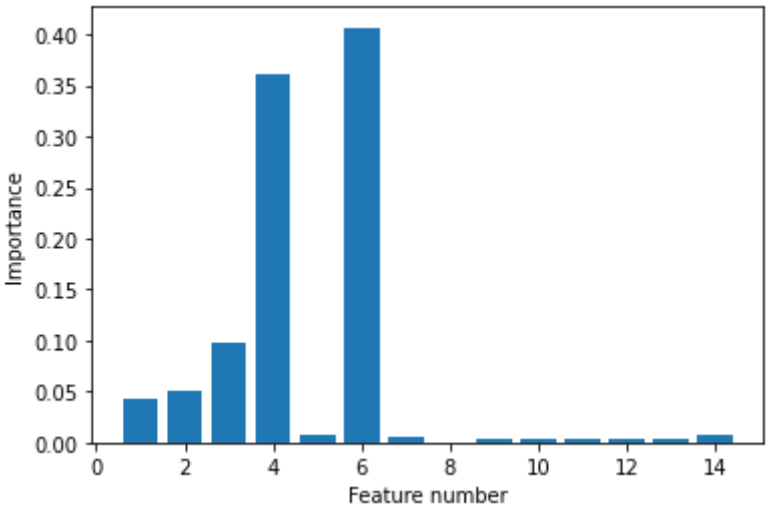


Figure 12: Random Forest Regression - Feature Importance

The test set was then used to predict price values, as per Figure 13 below.

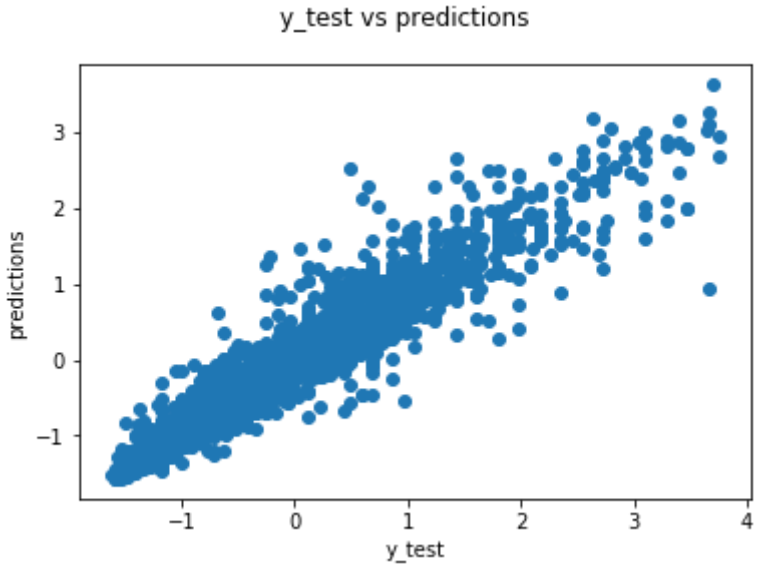


Figure 13: Random Forest Regression - Price Prediction using Test Dataset

The residual analysis of the dataset also produced good results in terms of normal distribution and homoscedasticity, as seen in Figures 14 and 15.

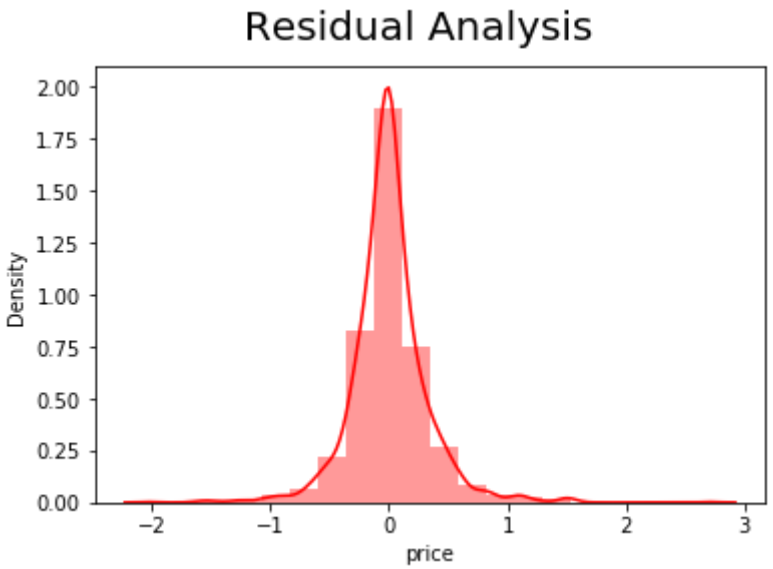


Figure 14. Random Forest Regression - Residual Distribution Curve

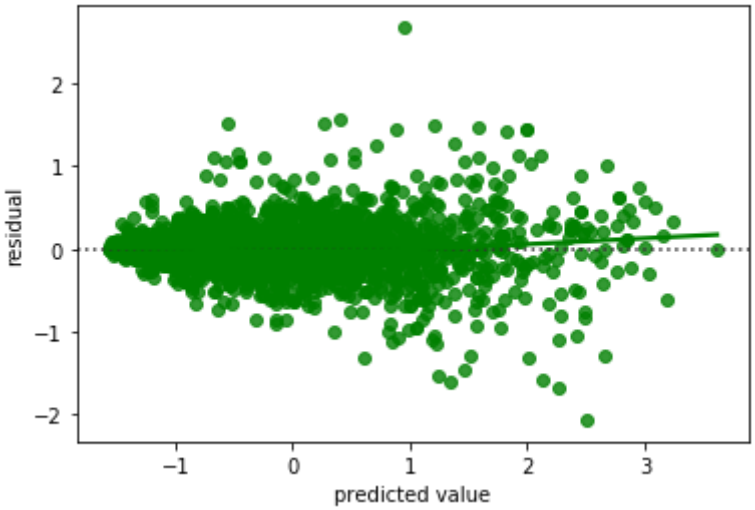


Figure 15. Random Forest Regression - Scatterplot of Price Prediction and Residuals

The results of the RF model with a 10% split are outlined in the table below.

Test Split	N Estimators	Train Set Accuracy (R^2)	Test Set Accuracy (R^2)	Test Set Accuracy (Adj. R^2)	Test Set MAE[1]	Test Set MSE[2]	Test Set RMSE[3]
30%	100	0.983	0.898	0.987	0.216	0.103	0.321

1. Mean Absolute Error 2. Mean Squared Error. 3. Root Mean Squared Error

Table 11. Random Forest Regression Results

4. Model Evaluation

The results of the 3 models built as part of this research are shown below. The results indicate that the RF regression model is the best predictor of car price in the dataset, based on the R^2 values of the test datasets.

Model	N Features	Test Split	Train Set Accuracy (R^2)	Test Set Accuracy (R^2)	Test Set Accuracy (Adj. R^2)	Test Set MAE[1]	Test Set MSE[2]	Test Set RMSE[3]
MLR	13	30%	0.693	0.691	0.689	0.418	0.311	0.558
KNN	14	10%	0.909	0.843	-	-	0.139	0.374
RF	14	30%	0.983	0.898	0.987	0.216	0.103	0.321

1. Mean Absolute Error 2. Mean Squared Error. 3. Root Mean Squared Error

Table 11. Comparison of Regression Model Accuracy - Price Prediction

Conclusion

The aim of this study was to predict the price of cars in a dataset using three different regression models: Multiple Linear Regression, K-Nearest Neighbours, and Random Forest (RF). Multiple train-test splits and cross-validation techniques were employed to determine the most efficient model for each algorithm. Results indicated that RF was the best predictive model for car price following its regression on fourteen car features in the dataset. The researchers achieved a respectable predictive accuracy of 89.8% using RF. By applying our RF model, customers can reliably determine the predicted selling price of a used car, and make an informed decision. The researchers applied only three machine learning models to the dataset, and would recommend that additional algorithms be explored in future research, e.g., Artificial Neural Networks, XGBoost or Gradient Boosting.

References

Allison, P. (2012). *When Can You Safely Ignore Multicollinearity?*. Available at: https://statisticalhorizons.com/multicollinearity (Accessed 30 November 2020).

Birla, N. (2020). *Vehicle dataset from cardekho*. Available at: https://www.kaggle.com/nehalbirla/vehicle-dataset-from-cardekho (Accessed 3 December 2020).

D'Agostino, R. B. (1971). An omnibus test of normality for moderate and large sample size. *Biometrika, 58*, 341-348

D'Agostino, R. and Pearson, E. S. (1973). Tests for departure from normality. *Biometrika, 60,* 613-622

Garg, R. (2018). A Primer to Ensemble Learning - Bagging and Boosting. Available at: https://analyticsindiamag.com/primer-ensemble-learning-bagging-boosting/#:~:text=Bagging%20is%20a%20way%20to,based%20on%20the%20last%20classification (Accessed 5 December 2020).

Gegic, E. et al. (2019). Car price prediction using machine learning techniques. *TEM Journal, 8*(1), p.113.

Inside Airbnb (n.d.). Available at: http://insideairbnb.com/index.html (Accessed 1 December 2020)

K-Nearest Neighbors (2020). Available at: https://datasciencewithsan.com/knn/ (Accessed 4 December 2020)

Lepchenkov, L. (2019). *Used-cars-catalog*. Available at: https://www.kaggle.com/lepchenkov/usedcarscatalog (Accessed 3 December 2020).

Little, M.A., et al. (2017). Using and understanding cross-validation strategies. Perspectives on Saeb et al. *GigaScience, 6*(5), p.gix020.

Pal et al. (2019). How Much Is My Car Worth? A Methodology for Predicting Used Cars' Prices Using Random Forest. In *Future of Information and Communication Conference* (pp. 413-422). Springer, Cham. https://doi.org/10.1007/978-3-030-03402-3_28

Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. *JMLR, 12*, pp. 2825-2830.

Property Listings (2019). Available at https://www.kaggle.com/dragonduck/property-listings-in-kuala-lumpur/notebooks (Accessed 3 December 2020).

Pudaruth, S., (2014). Predicting the price of used cars using machine learning techniques. *Int. J. Inf. Comput. Technol, 4*(7), pp.753-764.

Reitermanova, Z. (2010). Data splitting. In *WDS*, *10*, pp. 31-36.

Ruginski, I. (2016). Checking the assumptions of linear regression. *Accessed, 11*, p. 2018.

Seth, K. (2020). *U.S. Airbnb Open Data*. Available at: https://www.kaggle.com/kritikseth/us-airbnb-open-data (Accessed 1 December 2020).

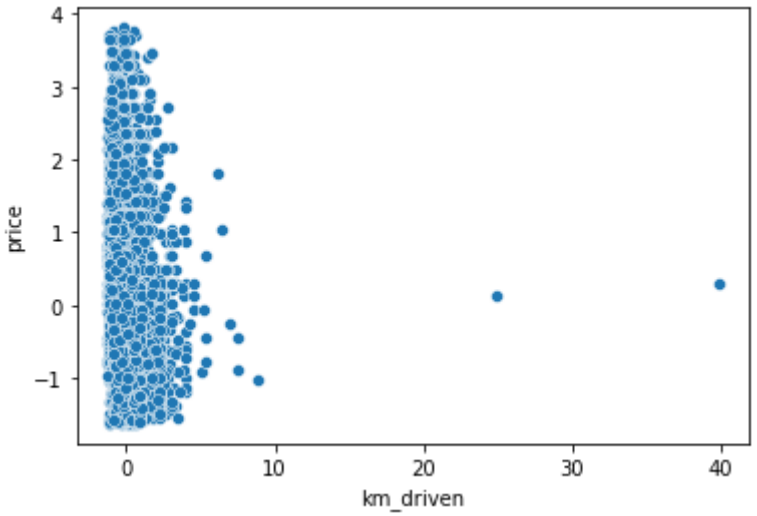
Singh, A. (2018). *A Practical Introduction to K-Nearest Neighbors Algorithm for Regression (with Python code)*. Available at: https://www.analyticsvidhya.com/blog/2018/08/k-nearest-neighbor-introduction-regression-python/ (Accessed 4 December 2020).

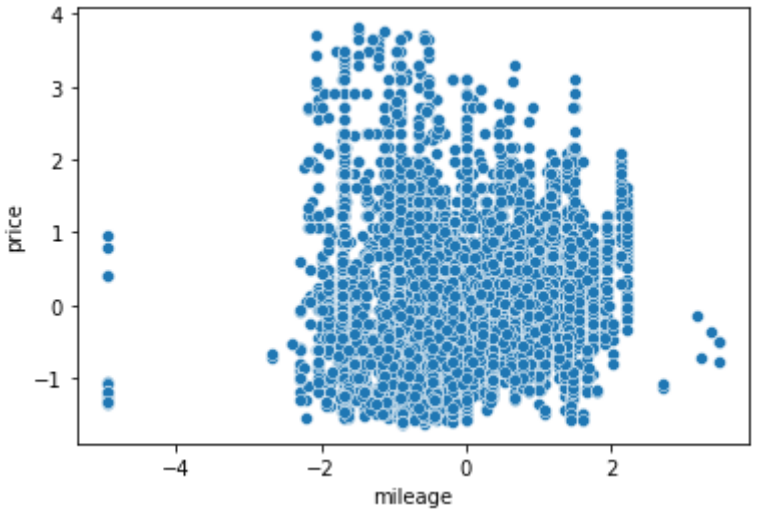
Puteri, C.K. and Safitri, L.N. (2020). Analysis of linear regression on used car sales in Indonesia. *JPhCS, 1469*(1), pp. 1-9. doi:10.1088/1742-6596/1469/1/012143

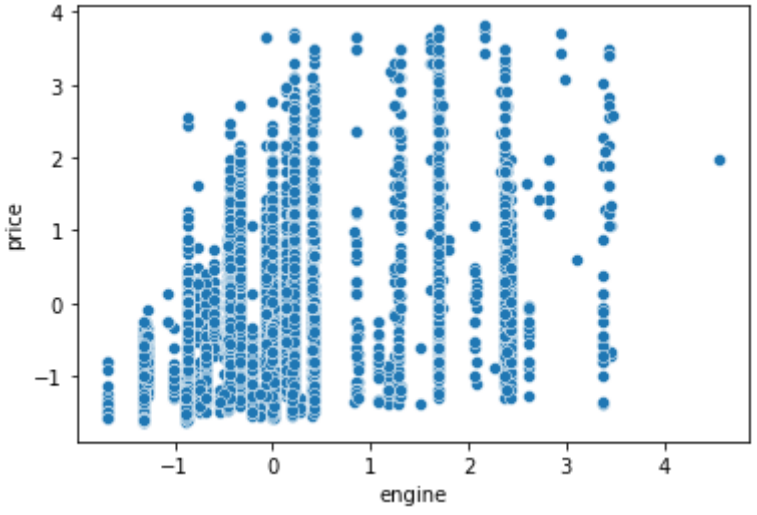
Uras, N., Marchesi, L., Marchesi, M. and Tonelli, R. (2020). Forecasting Bitcoin closing price series using linear regression and neural networks models. *arXiv preprint arXiv:2001.01127*.

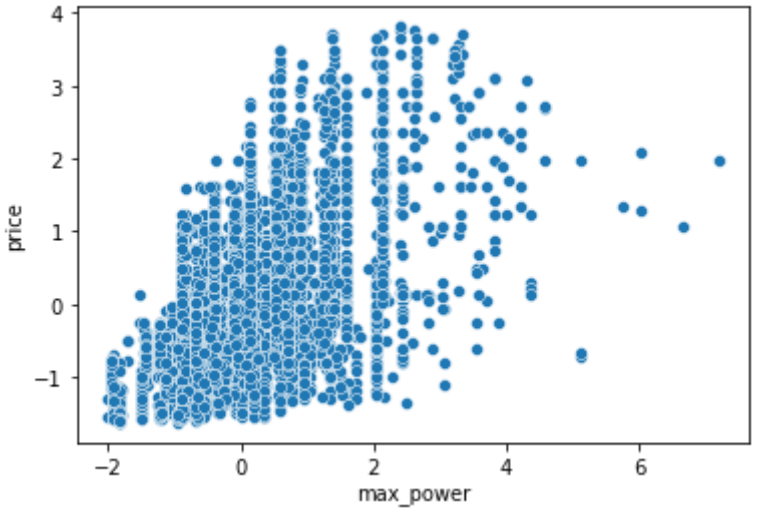
World Happiness (2019). Available at https://www.kaggle.com/unsdsn/world-happiness (Accessed 3 December 2020).

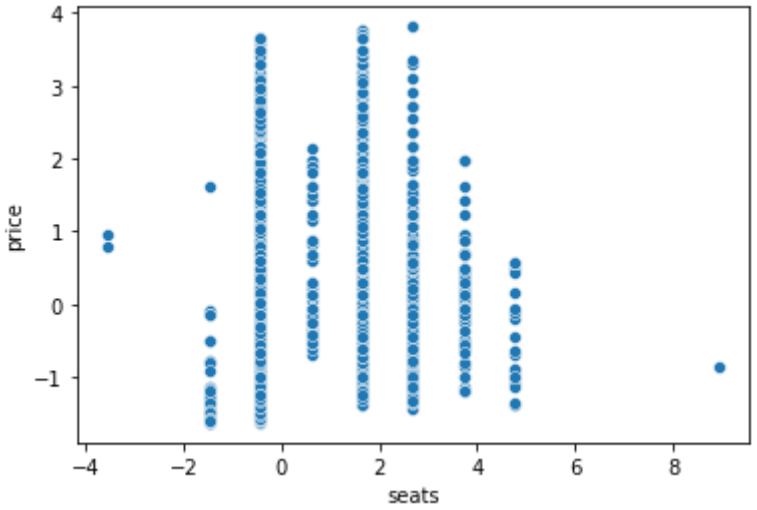
Appendix 1: Scatterplots for MLR analysis

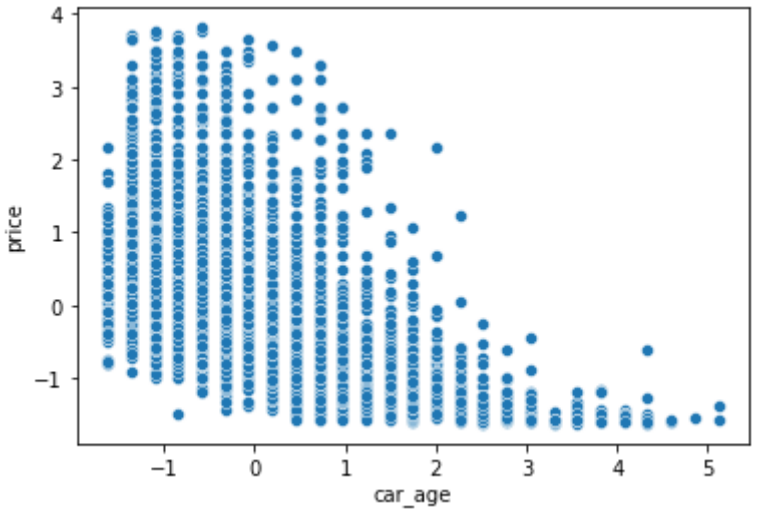




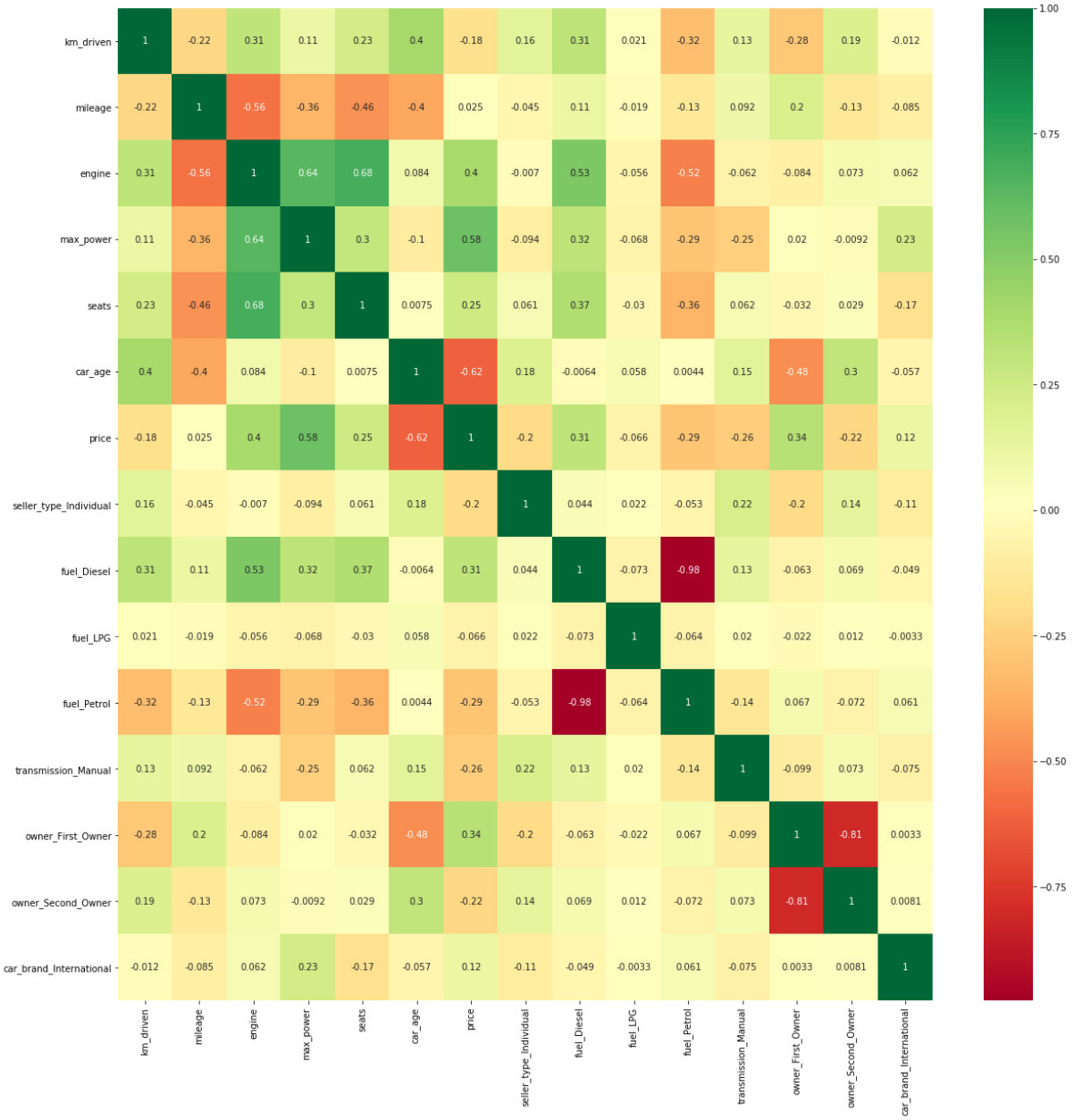



price
max_power
4
3
2
1
0
−1
−2
0
2
4
6


price
seats
4
3
2
1
0
−1
−4
−2
0
2
4
6
8


price
car_age
4
3
2
1
0
−1
−1
0
1
2
3
4
5

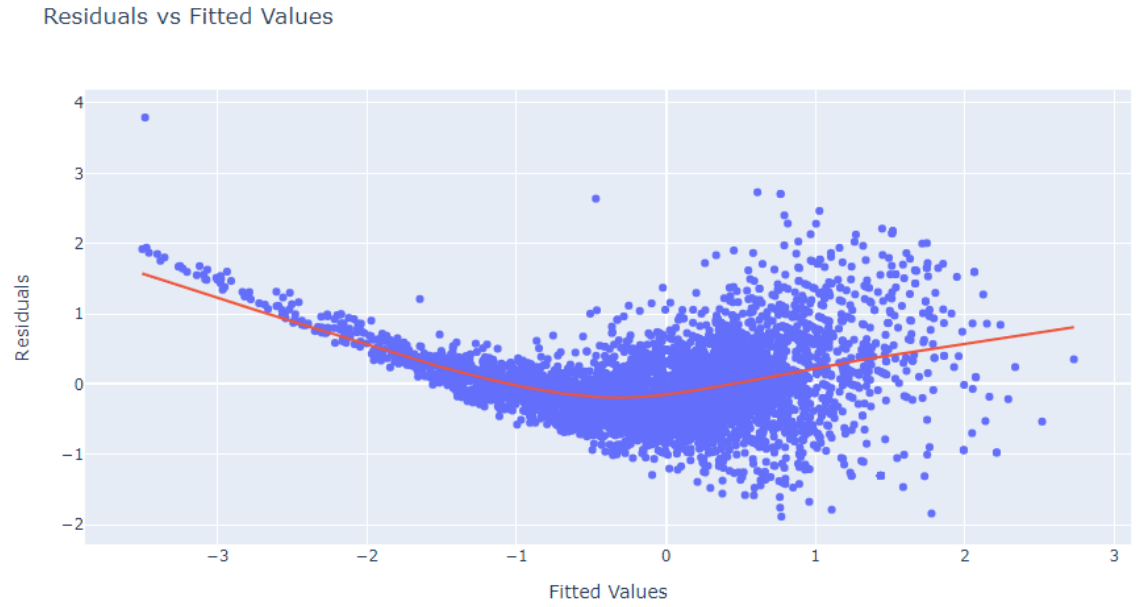
Appendix 2: Pearson Correlation Coefficient Heatmap

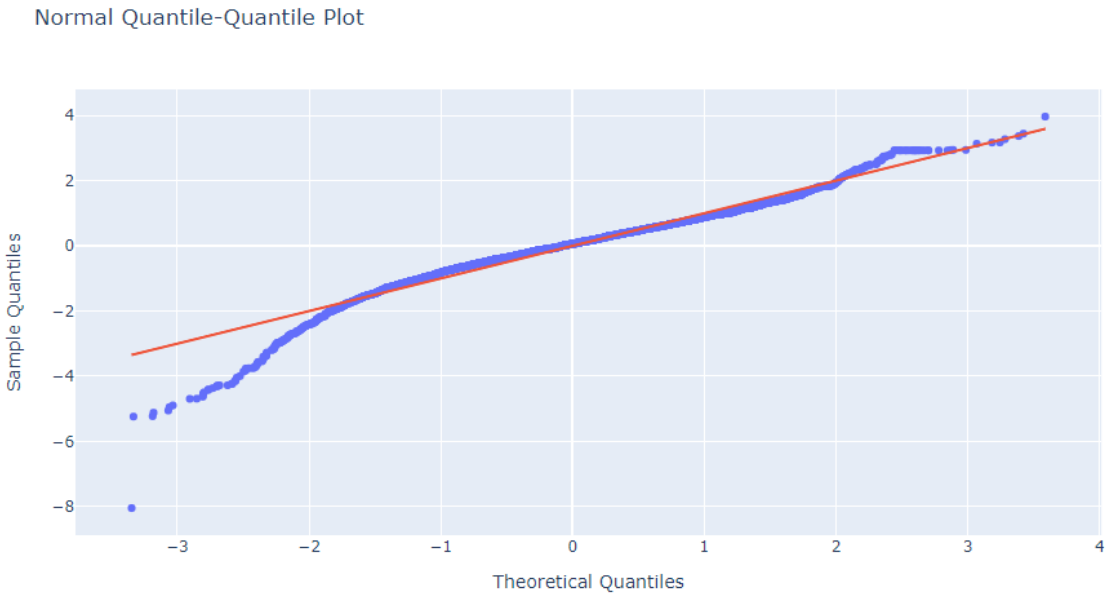


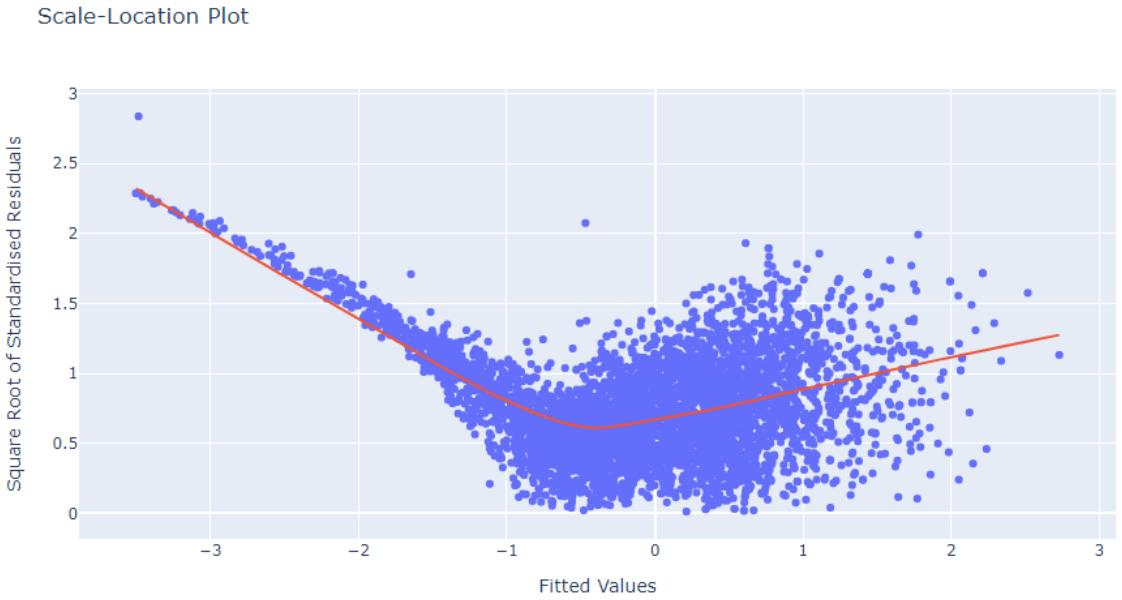
Appendix 3: Group 11 Roles and Responsibilities

Task	Task Owner
Find and choose suitable dataset	Giuliano, James, Yorgos
EDA, Data Cleaning and Data Preparation	James, Giuliano
Introduction and Research Objectives	James, Giuliano, Yorgos
Build KNN Regression Model	James, Yorgos
Build Linear Regression Model	James, Giuliano
Build Random Forest Model	Giuliano, James
ML Model Evaluation	Yorgos, Giuliano, James
Conclusion	Giuliano, James
Lead Python Programmer/Model Builder	Giuliano
ML Model Quality Analysis	Yorgos, James
HTML and IPYNB File Cleanup	James, Giuliano, Yorgos
Document Write-up Coordinator, Referencing	James

Appendix 4: Testing Linear Regression Assumptions with Residuals




Normal Quantile-Quantile Plot
Sample Quantiles
Theoretical Quantiles
4
2
0
−2
−4
−6
−8
−3
−2
−1
0
1
2
3
4


Scale-Location Plot
Square Root of Standardised Residuals
Fitted Values
3
2.5
2
1.5
1
0.5
0
−3
−2
−1
0
1
2
3