Used Cars Price Prediction

Team Members:

1. ASHISH M K – 200911236
2. M MARUTHI SANJEEV – 200911138
3. RISHI LODHIYA – 200911130

Introduction:

Cars which were previously owned by one or more retail owners can be called Used Cars. The technique of estimating a used car's value and potential performance based on details such its make, model, year of production, mileage, condition, and market demand is known as used car prediction.

Acquiring used cars can present a challenge unlike buying new ones from dealerships. To begin with, used cars can be purchased from different sources, including auctions, private sellers, and dealerships, which can result in varying prices. Secondly, the manufacturer's suggested retail price (MSRP) is not applicable to used cars as the sellers themselves determine the selling price. Sellers typically make known the condition of the used vehicle and set a corresponding price tag, and buyers can either purchase it at the set price or negotiate before buying. Finally, the condition of used cars can be intricate and require careful consideration before purchase.

For both buyers and sellers of used automobiles, this kind of prediction is essential since it enables them to make well-informed choices regarding pricing, buying, and selling. Used car prediction models can precisely predict a vehicle's depreciation rate, potential maintenance expenses, and resale value by using historical data and machine learning algorithms. Accurate prediction models are becoming more and more crucial for both consumers and dealers as the used automobile market continues to expand and change.

Our project aims to develop such a model which can predict used cars price with good accuracy in order to assist in making decisions regarding purchasing of used cars and which helps the person/organization to keep a healthy financial plan which involves less risk and less burden. The suggested model also has the ability to demonstrate the rate at which used cars lose their value over time. This feature can assist potential buyers in making an informed decision about which model to purchase, as it takes into consideration the car's potential resale value.

For this project, we have used 2 datasets in order to predict the price. The first dataset was obtained from CarDekho.com and the second dataset was obtained from Quikr.com. CarDekho is a famous and well-reputed platform for buying new or used cars in India based on user preferences. Quikr is a platform used for buying and selling any kind of product where users can advertise their product free-of-cost and buyers can buy products based on their preferences. Both the datasets were uploaded on Kaggle, which is where we retrieved them from.

Literature Review:

Chuyang Jin[1] proposed an article “Price Prediction of Used Vehicles Using Machine Learning” in which his objective was to develop a model that would forecast the costs of used cars taking a number of factors into account. He used various Regression algorithms and compared the results between them. He concluded by saying that Random forest regression has the greatest R-square of all five regressions, coming in at 0.90416. Hence, the employed vehicle prediction model was built using random forest regression.

Nitis Monburinon and his fellow mates[2] differentiated the execution of regression using supervised machine learning models in their work, "Prediction of pricing for secondhand cars by utilising regression models." The models were trained using information about the second-hand cars market gathered from a German E-commerce website. Gradient Boosting, RF Regression, and Decision Tree Regression were among the regression methods they employed. With only an MAE of 0.28, they came to the conclusion that Regression trees with gradient boost outperformed all other methods. The error for regression using a random forest model was 0.35, but the error for regression using a multiple linear model was 0.55. In light of this, they came to the conclusion that gradient boosted regression trees are recommended for the creation of the price evaluation model.

In the following case study, "Used Car Price Prediction using Machine Learning: A Case Study," Mustapha Hankar[3] and his colleagues used a number of regression models to predict the resale cost by taking multiple characteristics into consideration like mileage, year of production and fuel it runs on. KNN, Regression Models, and Artificial Neural Network were all implemented. The results showed that gradient boosting regressor outperformed them all, with the best R^2 rating and the lowest root mean squared error of any evaluated model.

In the paper “Research on used car price prediction based on random forest and LightGBM” which were published by Yashi Li[4] and his teammates, the price of used cars is predicted using the random forest and LightGBM algorithms, and the prediction results are compared and examined. Research demonstrates that LightGBM's prediction effect is superior to random forest's. This article and the test only choose a portion of the features, and the consideration may not be thorough enough, therefore the findings can only be used as a reference because the market price of used automobiles is affected by a variety of circumstances.

In a study titled "Review on the Pre-owned Vehicle Pricing Determination Using Machine Learning Methods," Punitha and Angelin[5] focus on a number of machine learning strategies that have been proposed by researchers for calculating the cost of used cars as well as their limitations. It also emphasises a hybrid machine learning strategy that may be altered to produce accurate forecasts regarding the price of

used autos. It also serves as an example of how the suggested approach for forecasting the cost of used cars performed. They employed Support Vector Machine, Simple Linear Regression, Random Forests, K-Nearest Neighbors, and Artificial Neural Networks. All of the above strategies has a particular advantage over a specific dataset.

A study titled "Machine Learning Methods To Forecast The Price Of Used Cars: Predictive Analytics in Retail Industry" was written by Chejarla Venkat Narayana[7] and colleagues. This study helps determine the value of a pre-owned car depending on its characteristics and decreases business risk for both the seller and the buyer. The recommended model combines machine learning algorithms with statistical regression techniques including random forest, linear, and decision tree regressions to achieve this purpose. The random forest model matches their data the best and has a respectable accuracy of 85%. They have considered that model in order to estimate the cost of a secondhand car.

Feng Wang[6] and his colleagues used a dataset that was preprocessed using the Pycaret module of Python in their article, "Prediction of Used Vehicle Price Based on Supervised Learning Algorithm," and they compared the output of all the algorithms they used. In this investigation, both the Additional Trees Regressor and the Random Forest Regressor perform admirably. Following that, the algorithm was optimised using the hyperparameter function. They concluded by showing that ET was the best model to predict the price of used cars which had a very high R^2 value.

An article titled "Prediction Of Used Vehicle Prices Using Artificial Neural Networks And Machine Learning" was written by Janke Varshitha[8] and her mates. This research provides a practical model with low error for calculating used automobile pricing. Several distinctive factors are taken into account for reliable and accurate projections. The observed results are in line with theoretical hypotheses and show improvement over models that rely on straightforward linear models. They employed Linear Regressions, Lasso, Ridge, Random Forest, Artificial Neural Networks (ANN), and Keras Regressor. These methods are tested using the car dataset. The Random Forest model has, among all the other algorithms, produced the least amount of inaccuracy, according to experimental results.

Han Zhang[9] published a paper **“**Prediction of Used Car Price Based on LightGBM”. Using actual transaction records from a used automobile trading website, this study suggests a more advanced machine learning technique based on LightGBM to estimate used car values authoritatively and creatively. In addition to LightGBM, he also used Linear Regression, SVM, Random Forest, GBDT and XGBoost. The prediction accuracy of LightGBM is higher than that of other techniques. We pioneer the application of LightGBM in the prediction of used automobile transaction prices.

Abdulla AlShared[12] published a paper **"**Used Cars Price Prediction and Valuation using Data Mining Techniques". This project's main goal is to calculate used car costs using attributes that have a strong correlation with the price attribute. Data mining methods were used to achieve this. During pre-processing, missing, redundant, and null values were eliminated from the dataset. Random Forest, Linear Regression and Bagging Regression were used. After pre-processing and transformation, the Random Forest Regressor, which had an accuracy of 95%, triumphed, followed by the Bagging Regressor, which had an accuracy of 88%.

A work titled "Used Vehicle Price Prediction using K-Nearest Neighbor Based Model" was proposed by K. Samruddhi and Dr. R.Ashok Kumar[10]. This article suggested an ML model using the KNN (K Nearest Neighbor) regression technique to analyse the price of used cars. The accuracy gained in this case using the K closest neighbour strategy was 85% as opposed to 71% for linear regression. The proposed model is additionally verified with 5 and 10 folds using the K Fold Method.

A paper titled "Price Prediction and Classification of Used-Vehicles Using Supervised Machine Learning" was proposed by Lucija Bukvic[11] and others. This study gives an overview of data-driven algorithms for predicting the cost of pre-owned cars on the Croatian market using related features in terms of manufacturing year and kilometres driven. Data mining from the online retailer "Njukalo" was used to achieve this. Several regression models were employed. In terms of prediction accuracy, the random forest classifier outperforms all other models.

Chadraprakash Trivendra[13] and his co-workers published a paper "The Price Prediction for used Cars using Multiple Linear Regression Model". The application of a controlled AI algorithm to forecast the cost of traded-in autos or vehicles has been investigated in this study. The expectations are predicated on accurate data received from a reliable source. A unique methodology and several direct relapse tests were used to generate the prediction. The cost of trade-in vehicles has been estimated in this paper using a unique AI approach. The results show a small improvement in the cost estimation of the vehicle. The value credit must now be divided into classes that include both specialized and non-specialized vehicle traits. The small number of records that were used in this analysis is its main limitation, which results in a good accuracy.

Anu Yadav, Ela Kumar, Piyush Kumar Yadav[14] proposed a paper **"**Object detection and used car price predicting analysis system (UCPAS) using machine learning technique". The notion of object detection, such as the identification of an automobile, has been understood in this work in order to investigate the cost of a used car utilizing automatic machine learning techniques. The algorithms used were Linear Regression and Random Forest. The outcomes of this experiment show that the best accuracy outcomes are achieved by clustering in conjunction with the

Random Forest model and linear regression. The machine learning model provides a reasonable result quickly when compared to the previously discussed self.

The paper “Fair Price Prediction System for Used Cars in Sri Lanka Using Machine Learning and Robotic Process Automation” which was published by T P Jayadeera and D J Jayamanne[15], focuses on developing a supervised learning-based used car pricing forecast system for Sri Lanka. To find the most appropriate regression models for the study, they analyzed with various relevant ML models, utilizing a data set which was available online. Multiple Linear Regression, Random Forest, Decision Tree, K-Nearest Neighbors and Gradient Boosting. According to the analysis's findings, Random Forest Regression and Multiple Linear Regression algorithms perform the best in predicting prices, whereas Decision Tree Regression and Gradient Boosting Regression algorithms perform just fair to poorly. Yet, the Sri Lankan and internet data sets provide unexpected results for the K-Nearest Neighbors algorithm.

In[16], the study evaluates the effectiveness of support vector regression, random forests, decision trees, and linear regression among other machine learning algorithms for estimating the price of secondhand cars. According to the study, support vector regression and random forest outperformed the other algorithms in terms of accuracy with an R-squared value of 0.95 and 0.94, respectively, the random forest and support vector regression techniques were found to have the highest accuracy.

The paper[17] uses a Random Forest Regression algorithm for estimating the Used Car Price. The study found that the algorithm was very good at predicting used car prices and had an accuracy of more than 90%.

The paper[18] compares the results of predicting used car prices using machine learning techniques like random forest, gradient boosting, and neural networks. Random forest and neural networks came in second and third, respectively, in terms of accuracy, according to the study. The gradient boosting algorithm was found to have the highest accuracy in the study, with a mean absolute error of 1601.46 and a root mean squared error of 2568.86, respectively.

The effectiveness of linear and non-linear regression models for used car price prediction is compared in a study[19]. The findings demonstrated that non-linear regression models performed better than linear regression models in terms of accuracy, including neural networks and support vector regression. The study discovered that, with R-squared values of 0.92 and 0.88, respectively, non-linear regression models such as neural networks and support vector regression were more accurate than linear regression models.

An article[20] compares the performance of different machine learning methods for predicting used car prices, including linear regression, decision tree, random forest, and neural networks. Research has shown that random forests and neural networks

outperform other methods in terms of accuracy. Research has shown that random forests and neural networks have the highest accuracy, with an R-squared value of 0.91 and 0.90, respectively.

The study[21] used machine learning algorithms such as decision trees,random forests,and neural networks to predict the prices of used cars in the Indian automarket. The results show that the random forest algorithm is the most accurate. The study found that random forest algorithms have the highest accuracy and the R square value is 0.94.

The objective of the research[22] is to evaluate the effectiveness of various machine learning techniques, such as linear regression, decision tree, random forest, gradient boosting, and neural networks, in forecasting the prices of pre-owned vehicles. After careful examination, it was concluded that the random forest algorithm was the most successful in terms of precision, followed by gradient boosting and neural networks. The results revealed that the random forest algorithm achieved the greatest precision, with a mean absolute error of 1701.66 and a root mean squared error of 2887.26. Despite having slightly lower precision, gradient boosting and neural networks were still capable of accurately predicting the prices of used cars.

In the manuscript[23], the proposal is to employ machine learning methods to anticipate the prices of pre-owned vehicles. The authors have utilized several regression algorithms, including support vector regression, decision tree regression, and random forest regression, to train models on a dataset of used car listings. They have also analyzed the influence of distinct feature sets, such as vehicle age, mileage, and make and model, on the models' performance. The outcomes reveal that the random forest regression model surpasses the other models, attaining an R-squared value of 0.95.

The article[24] employs machine learning methods to forecast the prices of used cars, with a specific focus on deep learning models. The writers utilize a convolutional neural network (CNN) to extract features from the images of the used cars, along with a recurrent neural network (RNN) to model the temporal data of the listings. They integrate these two models to predict the prices of the vehicles. The findings of their experiment show that their model surpasses traditional regression models, obtaining a mean absolute error of 9.95% on a Korean used car listings dataset.

In the paper[25], it proposes a feature engineering-based approach for used car price prediction. The authors use various techniques, such as one-hot encoding, feature scaling, and polynomial feature generation, to transform the raw data into more meaningful features. They then use gradient boosting regression, a popular ensemble learning method, to train a model on the transformed data. The experimental results show that their approach achieves an R-squared value of 0.91 on a dataset of US used car listings, outperforming other regression models that do not incorporate feature engineering.

Methodology for Dataset 1:

We obtained our first data set, Dataset 1, from Cardekho.com which was uploaded in Kaggle[26]. Our data set contained the following attributes:

```
In [5]: df.head()
```

Out[5]:

	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

Figure 1 – Dataset 1 Attributes

Data Cleaning: To begin with our methodology, we found out that our data set had quite a large number of null values in 5 attributes which would hinder our further proceedings.

```
In [7]: df.isnull().sum()

Out[7]: name             0
        year             0
        selling_price    0
        km_driven        0
        fuel             0
        seller_type      0
        transmission     0
        owner            0
        mileage        221
        engine         221
        max_power      215
        torque         222
        seats          221
        dtype: int64
```

Figure 2 – Null Values in Dataset 1

Therefore we removed all the null values by using the dropna() function in Python.

The attribute “owners” needed to be changed to a numerical format as it has a huge impact while selling used cars and needs to be fitted wherever possible. So, the following changes were done:

- Wherever there was “First Owner”, we replaced it with the number “1”.
- Wherever there was “Second Owner”, we replaced it with the number “2”.

- Wherever there was “Third Owner”, we replaced it with the number “3”.
- Wherever there was “Fourth Owner & Above”, we replaced it with the number “4”.
- Wherever there was “Test Drive Cars”, we replaced it with the number “0”.

Furthermore, other attributes like mileage, engine, max_power (Figure 1) where also converted into a numerical format. String components like “kmpl” for the mileage attribute, “CC” for the engine attribute as well as “rpm” for the max_power attribute were removed.

A much more refined data set (Figure 3) has been achieved by following the previous steps.

	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	1	23.40	1248	74	190Nm@ 2000rpm	5.0
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	Diesel	Individual	Manual	2	21.14	1498	103.52	250Nm@ 1500-2500rpm	5.0
2	Honda City 2017-2020 EXi	2006	158000	140000	Petrol	Individual	Manual	3	17.70	1497	78	12.7@ 2,700(kgm@ rpm)	5.0
3	Hyundai i20 Sportz Diesel	2010	225000	127000	Diesel	Individual	Manual	1	23.00	1396	90	22.4 kgm at 1750-2750rpm	5.0
4	Maruti Swift VXI BSIII	2007	130000	120000	Petrol	Individual	Manual	1	16.10	1298	88.2	11.5@ 4,500(kgm@ rpm)	5.0

Figure 3 – Refined Version of Dataset 1

Distribution of price based on fuel:

Figure 4 – Distribution of price based on fuel

We may infer that diesel fuel cars are typically more expensive than petrol fuel cars by plotting the price versus fuel type. The plot shown above backs up our suspicions. Cars that run on petrol are generally less expensive. Because of their slightly better fuel usage and storage than LPG vehicles, CNG vehicles cost more than LPG vehicles on average.

Pair Plot:

We can view single variable distributions as well as connections between two variables using a pair plot. Pair plots are a fantastic tool for spotting tendencies for further investigation.Here we have used pair plot using selling_price, km_driven, engine, mileage, owner and hue value as “owner”

```
In [42]: sns.pairplot(df[["selling_price","km_driven","engine","mileage","owner"]],hue="owner")
```

Figure 5 – Attributes used for Pair Plot in Dataset 1

Nearly all of the characteristics are falling as ownership rises (Figure 6). For instance, compared to succeeding owners, the initial owner's car often has a larger mileage. Same rules apply to engine and km_driven. This also makes a lot of sense.

The visualization of outliers is also aided by scatter plots in pair plots. We can safely overlook a few small outliers in our data that are there.

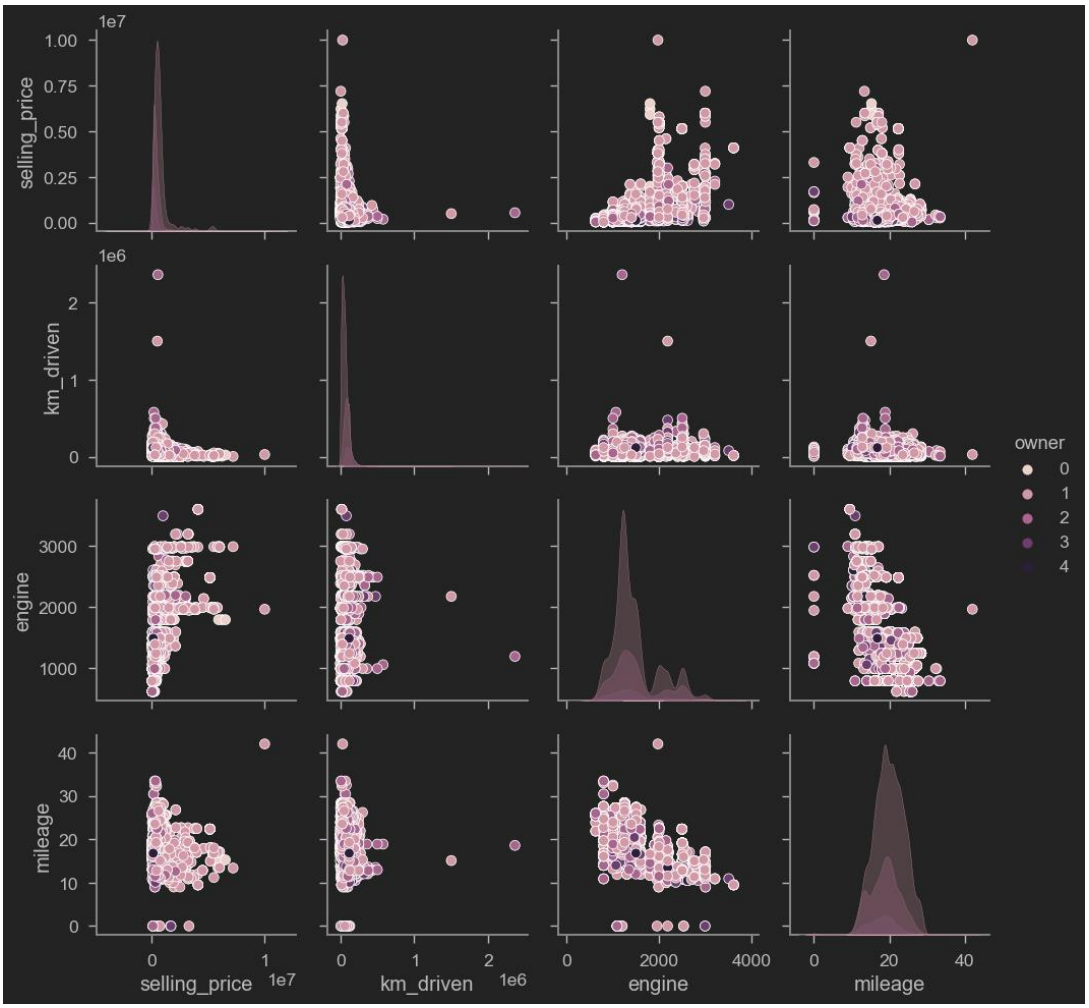

Figure 6 – Pair Plot of Dataset 1

Correlation Matrix:

	year	selling_price	km_driven	owner	mileage	engine	seats
year	1.000000	0.414092	-0.418006	-0.513541	0.329145	0.018848	-0.009144
selling_price	0.414092	1.000000	-0.225534	-0.243316	-0.126054	0.455734	0.041358
km_driven	-0.418006	-0.225534	1.000000	0.288681	-0.173073	0.205914	0.227336
owner	-0.513541	-0.243316	0.288681	1.000000	-0.173399	0.005067	0.029770
mileage	0.329145	-0.126054	-0.173073	-0.173399	1.000000	-0.575831	-0.452085
engine	0.018848	0.455734	0.205914	0.005067	-0.575831	1.000000	0.610309
seats	-0.009144	0.041358	0.227336	0.029770	-0.452085	0.610309	1.000000

Figure 7 – Correlation Matrix of Dataset 1

It is efficient to identify the dependencies by visually representing the correlations. The manufacture year and engine have a strong relationship with the selling price (0.41 and 0.46) in the specified plot (Figure 8).

The following is the heatmap of the correlation matrix:

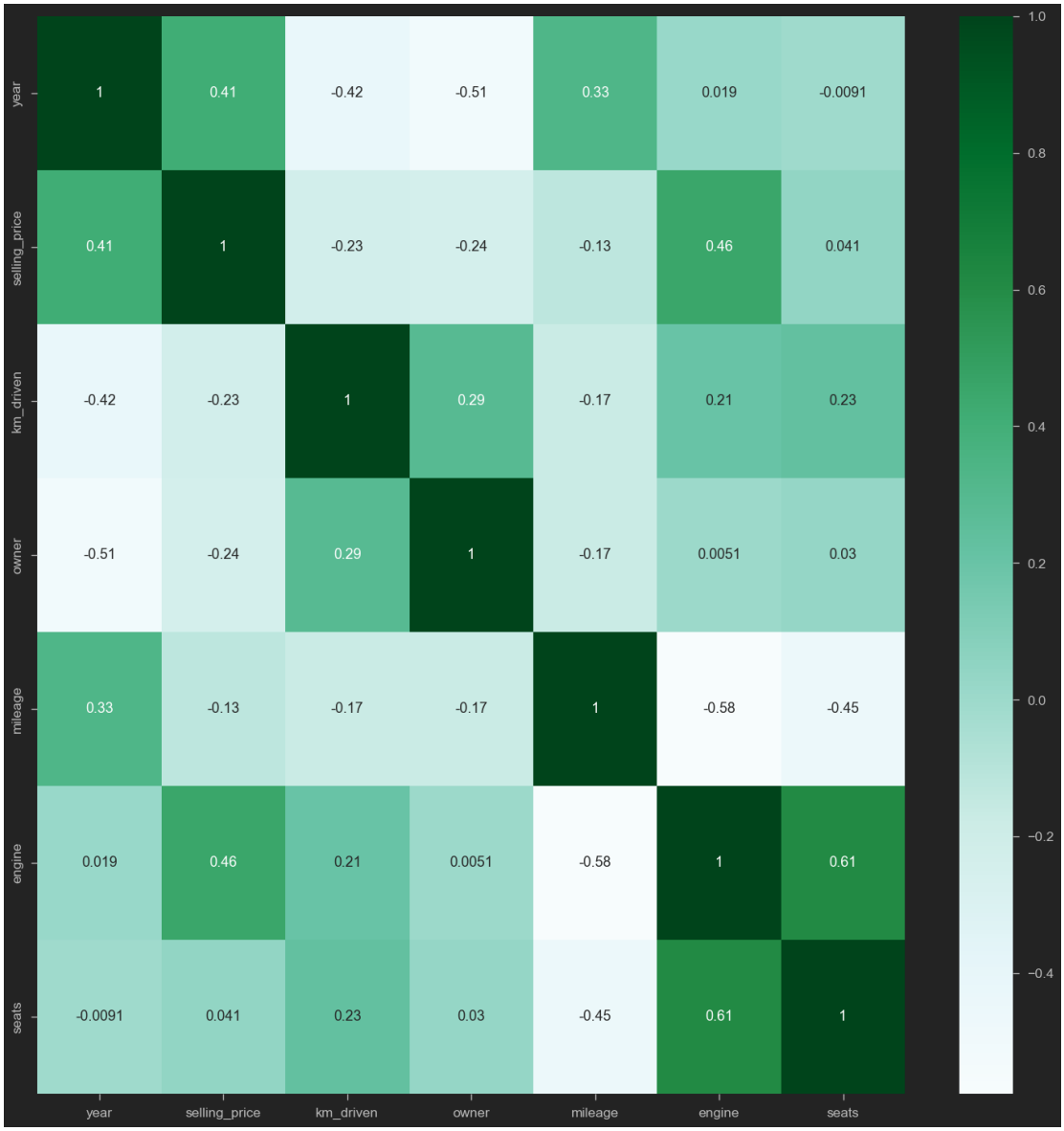

Figure 8 – Correlation Heatmap of Dataset 1

Algorithm Implementation:

Before implementing the algorithms, we need to transform the attributes whose data type is "object" to a numerical data type in order to fit it in any model. The following figure (Figure 9) describes how it is done. The operation resulted in total of 2837 columns.

```
df_new = pd.get_dummies(df, columns= ['name', 'fuel', 'seller_type', 'transmission', 'max_power','torque'])
```

```
df_new.head()
```

	year	selling_price	km_driven	owner	mileage	engine	seats	name_Ambassador CLASSIC 1500 DSL AC	name_Ambassador Classic 2000 DSZ AC PS	name_Ambassador Grand 1500 DSZ BSIII	...	torque_96.1Nm@ 3000rpm	torque_96Nn 2500r
0	2014	450000	145500	1	23.40	1248.0	5.0	0	0	0	...	0	
1	2014	370000	120000	2	21.14	1498.0	5.0	0	0	0	...	0	
2	2006	158000	140000	3	17.70	1497.0	5.0	0	0	0	...	0	
3	2010	225000	127000	1	23.00	1396.0	5.0	0	0	0	...	0	
4	2007	130000	120000	1	16.10	1298.0	5.0	0	0	0	...	0	

5 rows × 2837 columns

Figure 9 – Converting "Object" data type to Numerical format in Dataset 1

We then take a variable X and load it with the independent variables whom we want to compare our target variable to. Our target variable here is selling_price.

In the variable y we load it with our target variable (selling_price), so that we can compare it with other independent variables in the data set.

```
X = df_new.drop("selling_price", axis=1)
y = df_new["selling_price"]
```

Figure 10 – Determining Dependent and Independent Variables for Dataset 1

Now in order to use these in our algorithms, we need to split them into test and train datasets.

```
X = np.array(X)
y= np.array(y)
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
                                                    random_state=2)
```

Figure 11 – Splitting into Train and Test datasets for Dataset 1

For this project, we will use the following algorithms:

1. Multiple Linear Regression
2. Random Forest
3. XGBoost

We are using Multiple Linear Regression as there are several independent variables whom we want to compare our target variable to.

Random Forest is used as it gives high accuracy without any extra adjustments and also it handles large datasets efficiently.

XGBoost has multiple regularization parameters which improves the model performance and it also performs well even if there are missing values and has the ability to handle large datasets.

1. **Multiple Linear Regression:**

Multiple Linear Regression is a technique where we can predict a dependent variable based on other independent variables. In our case, we want to predict the "Price" attribute (dependent variable) based on all other independent variables.

The following is the mathematical imputation:

$$y = \beta_0 + \beta_1 x_1 + .\beta_2 x_2 + \beta_3 x_3 \ldots$$

Equation 1 – Relationship between dependent and independent variables

Here, y is the dependent variable.

x1, x2,x3,… are independent variables.

b0 = intercept of the line.

b1, b2, … are coefficients.

For this algorithm, we will use the previously built datasets (test and train) to get the result.

```
In [33]: from sklearn.linear_model import LinearRegression

In [34]: linear_model = LinearRegression()
         linear_model.fit(X_train, y_train)
         linear_model.score(X_test, y_test)

Out[34]: 0.9250624302486585
```

Figure 12 – Multiple Linear Regression on Dataset 1

Results obtained using Multiple Linear Regression:

We have obtained an accuracy of 92% (0.9250) which is a good accuracy for a Linear Regression Model. We can expect better results compared to this using other enhanced algorithms which will be demonstrated further.

Residuals Plot: The discrepancy between the values of the fitted response and the observed response is displayed on a residual plot. The null residual plot, which represents the ideal residual plot, exhibits a random distribution of points that form a band around the identity line that is roughly constant in width.

Residuals plot for Multiple Linear Regression:

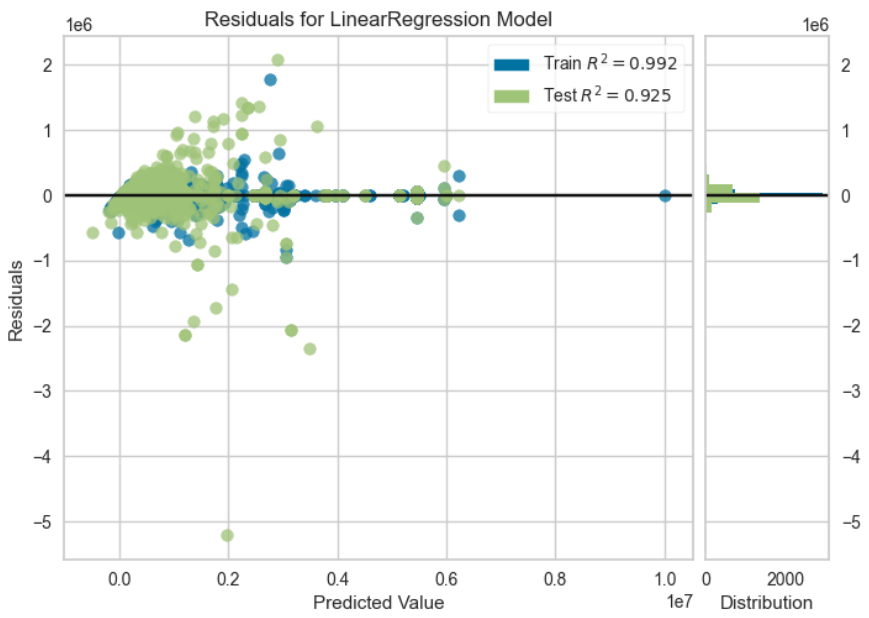

Figure 13 - Residuals plot for Multiple Linear Regression (Dataset 1)

We can observe that there is slight deviation in the residuals plot (Figure 13) for Multiple Linear Regression (92.5% correct).

Random Forest:

Random Forest uses an ensemble method learning technique for classification and regression. The basic tenet of ensemble method is that weak learners can combine with strong learners. At training time, Random Forest builds a number of decision trees. For bootstrapped datasets, these decision trees are individually trained. By averaging the predictions made by each individual tree, the final prediction value is determined.

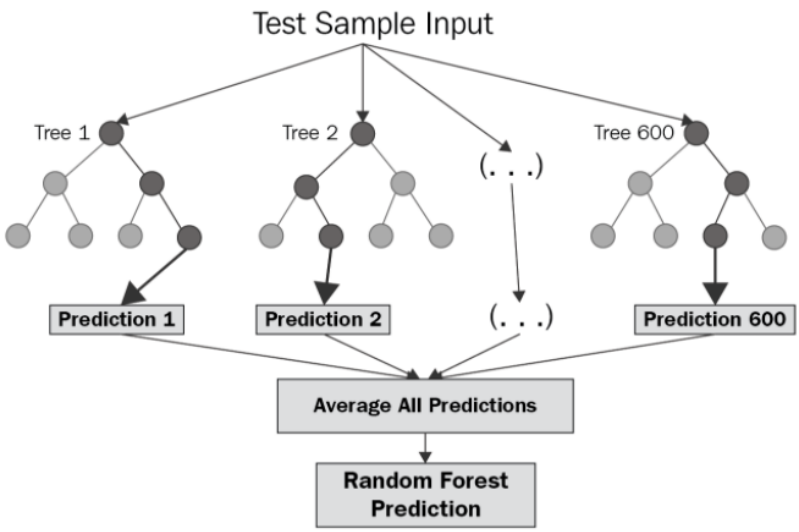

Figure 14 – Random Forest

For this algorithm, we will use the previously built datasets (test and train) to get the result.

```
In [61]: random_model = RandomForestRegressor()
         random_model.fit(X_train, y_train)
         random_model.score(X_test, y_test)

Out[61]: 0.9584908806653372
```

Figure 15 – Random Forest Regression on Dataset 1

Results obtained using Random Forest:
0.9584 (95% accurate)
The percentage of the dependent variable's variation that can be predicted from the independent variable is the definition of the coefficient of determination, also known as "R squared," in statistics (s). In our situation, R squared is nearer 1, demonstrating the model's dependability in forecasting the selling price. Without hyper-parameter adjustment, Random Forest is renowned for its great accuracy.

Residuals plot for Random Forest:

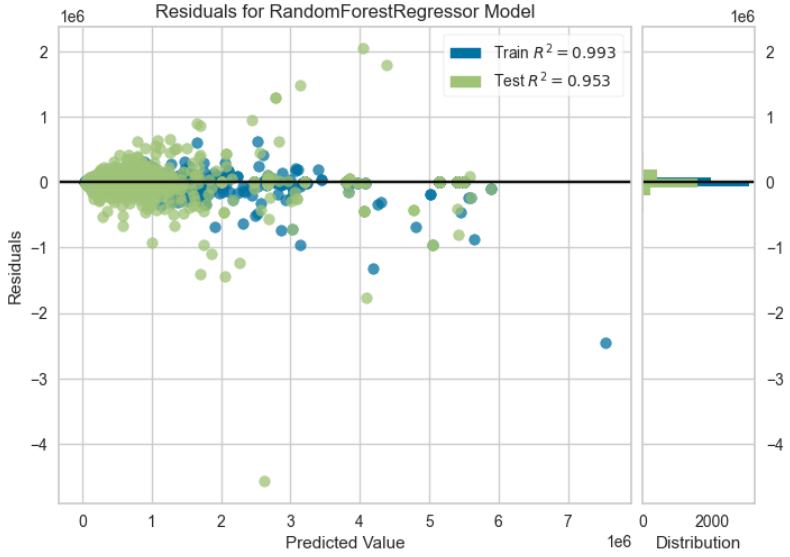

Figure 16 - Residuals plot for Random Forest (Dataset 1)

We can observe that there is an even slighter deviation (Figure 16) when compared to Multiple Linear Regression in the residuals (95.3% correct).

2. **XGBoost:**

XGBoost is a very efficient way of forming a supervised regression models. If we understand this statement's objective function and base learners, we can deduce its accuracy. Objective

function has two parts, i.e Regularisation term and Loss function. It gives insights on the inconsistency between the actual and predicted values or how far the models predictions differ from the actual results. The two most used loss functions in XGBoost for binary and regression classification are Reg:linear and Reg:logistics. It is one of the ensemble learning techniques, which demands training and integrating multiple independent models to give a single prediction.

We will use the same data we used for the Random Forest algorithm.

```
In [65]: from xgboost import XGBRegressor

In [66]: xgb_model = XGBRegressor()
         xgb_model.fit(X_train, y_train)
         xgb_model.score(X_test, y_test)

Out[66]: 0.9342274902724732
```

Figure 17 – XGBoost on Dataset 1

Results obtained using XGBoost:
0.9342 (93% accuracy)
We can observe that Random Forest Regressor performs a little bit better than XGBoost Regressor when we compared both of them.

Residuals plot for XGBoost:

Figure 18 - Residuals plot for XGBoost Dataset 1

We are getting a similar observation (Figure 18) in the case of XGBoost as well (94.1% correct).

Comparison of all the results obtained on Dataset 1:

Table 1:

Algorithms	Score
Multiple Linear Regression	0.9250
Random Forest	0.9584
XGBoost	0.9342

Here we can see that Random Forests performs slightly better than the other two algorithms by having an accuracy of 95% while XGBoost lags a bit lower with an accuracy of 93% and last the Multiple Linear Regression, with an accuracy of 92%.

Now, let's move on to our second dataset, Dataset 2. This dataset was obtained from Quikr which was uploaded in Kaggle[27]. Our 2nd Dataset contained the following attributes (Figure 19):

Out[3]:

	Name	Label	Location	Price	Kms_driven	Fuel_type	Owner	Year	Company
0	Ford Figo Duratec Petrol EXI 1.2 - 2015	PLATINUM	Bangalore	3,80,000	35,056 kms	Petrol	NaN	2015	Ford
1	Maruti Suzuki Wagon R VXI BS IV - 2016	PLATINUM	Bangalore	4,65,000	44,000 kms	Petrol	NaN	2016	Maruti
2	Hyundai Creta 1.6 SX PLUS AUTO PETROL - 2018	PLATINUM	Bangalore	13,50,000	42,917 kms	Petrol	NaN	2018	Hyundai
3	Hyundai Venue - 2019	PLATINUM	Chennai	10,19,699	16,112 kms	Petrol	2nd Owner	2019	Hyundai
4	Honda Jazz - 2017	PLATINUM	Pune	7,13,499	30,988 kms	Petrol	2nd Owner	2017	Honda

Figure 19 – Attributes of Dataset 2

Methodology for Dataset 2:

Data Cleaning: We have to again clean our dataset in order to proceed further without any problems in the future. We again drop all the null values and remove string components from numerical attributes.

Here we again replaced "1st Owner" with 1, "2nd Owner" with 2, and "3rd Owner" with 3 due to the same reason mentioned earlier for Dataset 1.

We did the same for other attributes like Price and Kms_driven by removing the commas and "kms" and converted them into numerical format.

```
In [7]: df["Owner"] = df["Owner"].str.replace("1st Owner", "1")
        df["Owner"] = df["Owner"].str.replace("2nd Owner", "2")
        df["Owner"] = df["Owner"].str.replace("3rd Owner", "3")

In [8]: df["Price"] = df["Price"].str.replace(",", "")
        df["Kms_driven"] = df["Kms_driven"].str.replace(",", "")

In [9]: df["Kms_driven"] = df["Kms_driven"].str.replace("kms", "")

In [10]: df["Owner"] = df["Owner"].astype("int")
         df["Price"] = df["Price"].astype("int")
         df["Kms_driven"] = df["Kms_driven"].astype("int")
```

Figure 20 - Converting Datatypes

We have obtained a refined dataset (Figure 21):

	Name	Label	Location	Price	Kms_driven	Fuel_type	Owner	Year	Company
3	Hyundai Venue - 2019	PLATINUM	Chennai	1019699	16112	Petrol	2	2019	Hyundai
4	Honda Jazz - 2017	PLATINUM	Pune	713499	30988	Petrol	2	2017	Honda
5	Hyundai i20 - 2013	PLATINUM	Pune	391099	69163	Diesel	2	2013	Hyundai
6	Maruti Suzuki Swift Dzire VXi 1.2 BS IV - 2015	PLATINUM	Pune	474299	42859	Petrol	2	2015	Maruti
7	Toyota Corolla Altis VL AT - 2017	PLATINUM	Pune	1252999	34919	Petrol	1	2017	Toyota

Figure 21 - Refined Version of Dataset 2

Pair Plot:

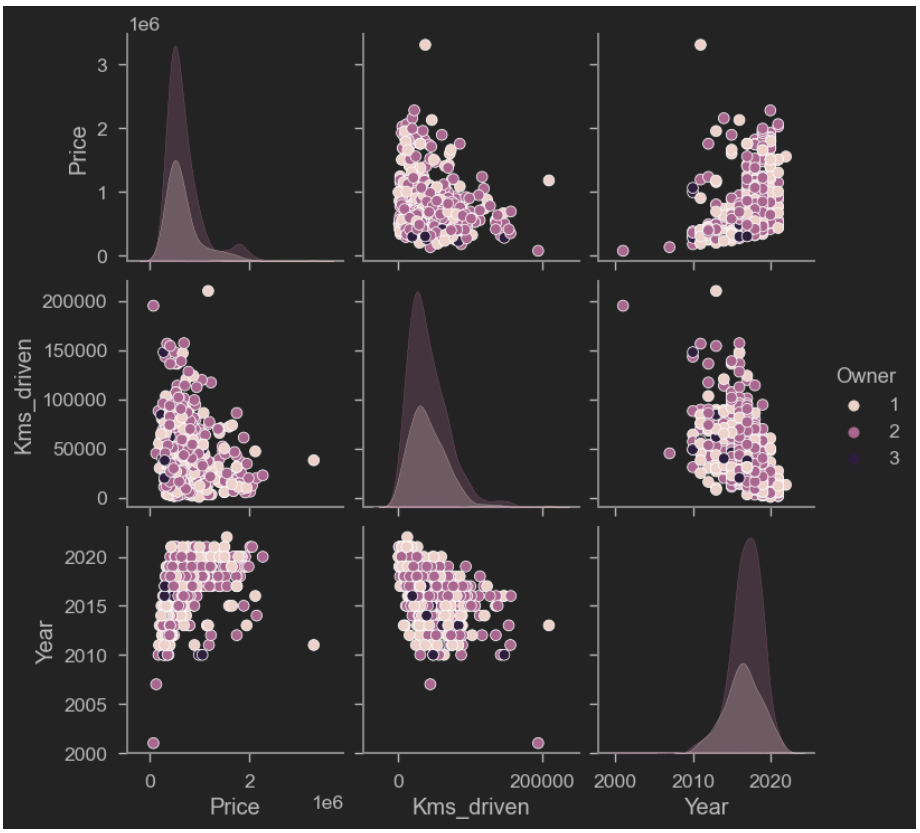

Figure 22 - Pair Plot for Dataset 2

Here again, nearly all of the characteristics are falling as ownership rises.

Correlation Matrix:

	Price	Kms_driven	Owner	Year
Price	1.000000	-0.160056	-0.027680	0.372076
Kms_driven	-0.160056	1.000000	0.019785	-0.525997
Owner	-0.027680	0.019785	1.000000	0.013313
Year	0.372076	-0.525997	0.013313	1.000000

Figure 23 - Correlation Matrix for Dataset 2

We can see that Price has a kind of strong correlation with the attribute Year.

The following is the heatmap:

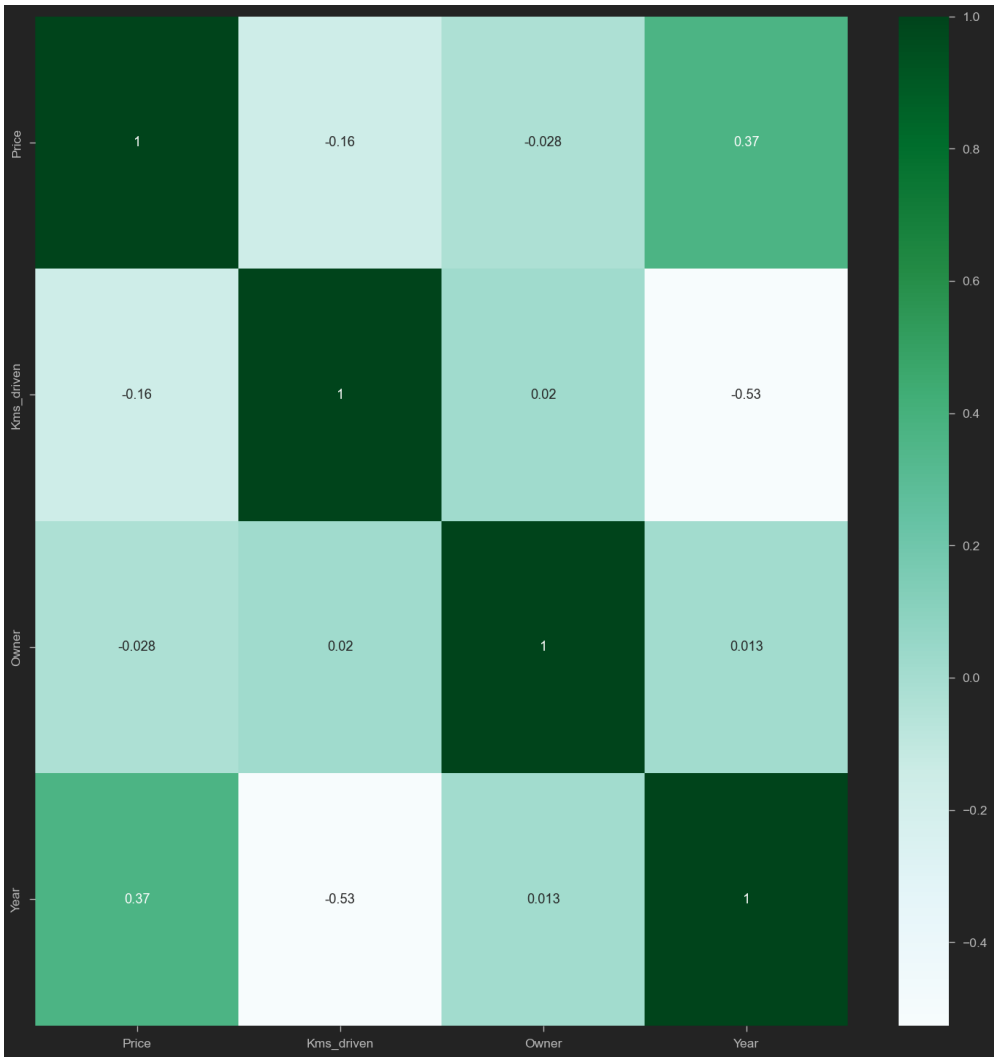

Figure 24 - Heatmap for Dataset 2

Algorithm Implementation:

Again, we will transform all our non-numerical attributes to numerical format in order to fit it in our models.

```
In [21]: df_new = pd.get_dummies(df, columns= ['Name', 'Label', 'Location', 'Fuel_type', 'Company'])

In [22]: df_new.head()
```

Out[22]:

	Price	Kms_driven	Owner	Year	Name_Audi Q3 - 2015	Name_BMW 3 Series 320d - 2013	Name_BMW 3 Series 320d Luxury Line - 2015	Name_BMW 5 Series 523i Sedan - 2010	Name_Chevrolet Sail 1.2 LS - 2013	Name_Datsun Redi GO - 2017	...	Company_Maruti	Company_Merc
3	1019699	16112	2	2019	0	0	0	0	0	0	...	0	
4	713499	30988	2	2017	0	0	0	0	0	0	...	0	
5	391099	69163	2	2013	0	0	0	0	0	0	...	0	
6	474299	42859	2	2015	0	0	0	0	0	0	...	1	
7	1252999	34919	1	2017	0	0	0	0	0	0	...	0	

5 rows × 476 columns

Figure 25 – Converting "Object" data type to Numerical format in Dataset 2

The above operation resulted in a total of 476 new columns.

Next, we will load a variable X with all our independent variables and a variable y with our dependent variable (Price).

```
X = df_new.drop("Price", axis=1)
y = df_new["Price"]
```

Figure 10 - Loading X and y

Then, we split our dataset into train and test

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state=2)
```

Figure 26 - Splitting into train and test for Dataset 2

1. **Multiple Linear Regression:**

```
In [33]: from sklearn.linear_model import LinearRegression

In [34]: linear_model = LinearRegression()
         linear_model.fit(X_train, y_train)
         linear_model.score(X_test, y_test)

Out[34]: 0.8056085663543814
```

Figure 27 - Multiple Linear Regression on Dataset 2

Results obtained using Multiple Linear Regression:

For this dataset, we obtained an accuracy of 80% (0.8056) using this algorithm.

Residuals Plot for Multiple Linear Regression:

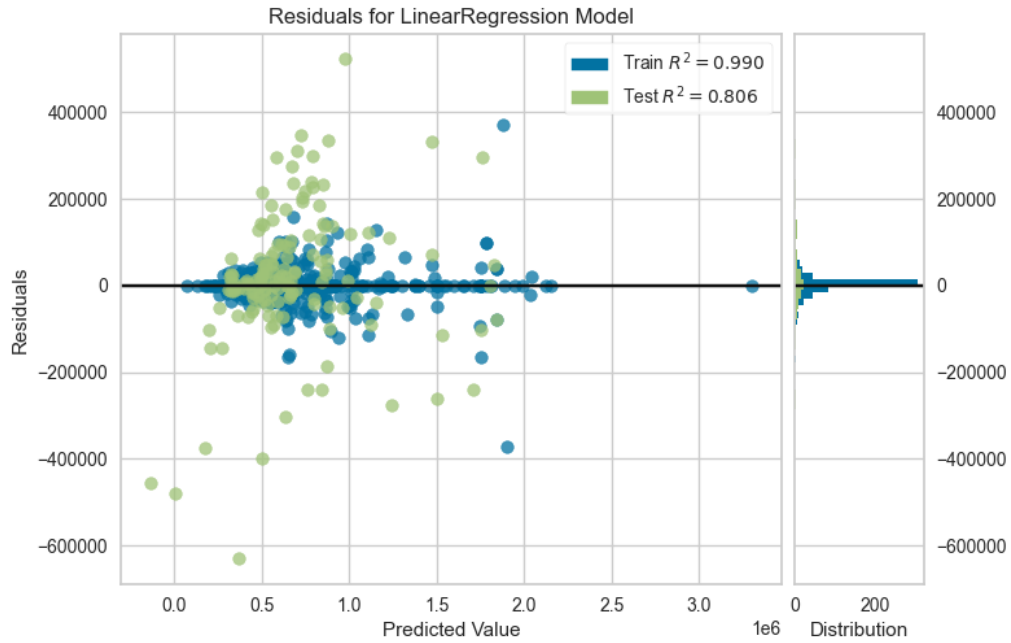

Figure 28 - Residuals Plot for Multiple Linear Regression on Dataset 2

In the above figure (Figure 28), we can see more deviation from the ideal result when compared to the first dataset.

2. Random Forest:

```
In [29]: from sklearn.ensemble import RandomForestRegressor

In [30]: random_model = RandomForestRegressor()
         random_model.fit(X_train, y_train)
         random_model.score(X_test, y_test)

Out[30]: 0.8103951223415314
```

Figure 29 - Random Forest on Dataset 2

Results obtained using Random Forest:

We obtained an accuracy of 81% (0.8103) which is slightly better than that of Multiple Linear Regression.

Residuals Plot for Random Forest:

Figure 30 - Residuals Plot for Random Forest on Dataset 2

The above figure (Figure 30) shows a slightly better deviation (1% better than Multiple Linear Regression).

3. XGBoost:

```
In [31]: from xgboost import XGBRegressor

In [32]: xgb_model = XGBRegressor()
         xgb_model.fit(X_train, y_train)
         xgb_model.score(X_test, y_test)

Out[32]: 0.8203022394714998
```

Figure 31 - XGBoost on Dataset 2

Results obtained using XGBoost:

An accuracy of 82% (0.8203) was achieved which is better than the previous algorithms we used.

Residuals Plot for XGBoost:

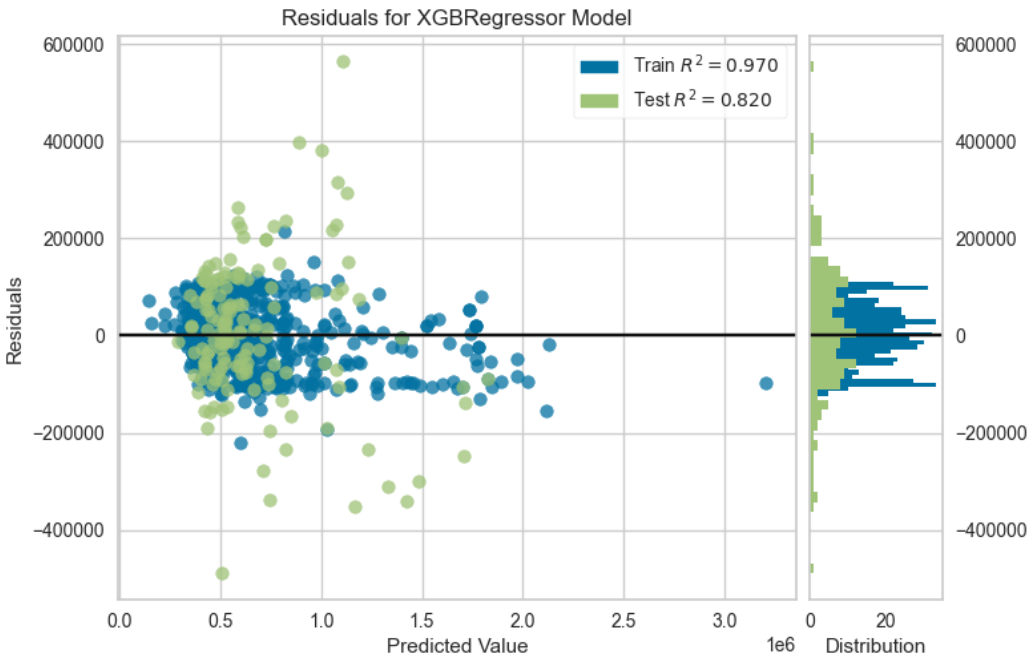

Figure 32 - Residuals Plot for XGBoost for Dataset 2

The above figure (Figure 32) shows a slightly better deviation (2% better than Multiple Linear Regression and 1% better than Random Forest).

Comparison of all the results obtained on Dataset 2:

Table 2

Algorithms	Score
Multiple Linear Regression	0.8056
Random Forest	0.8103
XGBoost	0.8203

We can see that XGBoost performs slightly better than the other two algorithms by having an accuracy of 82% (0.8203) while the other two algorithms gave an accuracy of 80% (0.8056 – Multiple Linear Regression) and 81% (0.8103 – Random Forest).

Final comparison of results obtained on both datasets:

Table 3

Algorithms	Score of Dataset 1	Score of Dataset 2
Multiple Linear Regression	0.9250	0.8056
Random Forest	0.9584	0.8103
XGBoost	0.9342	0.8203

We can see that results obtained on Dataset 1 gives more accuracy than that obtained on Dataset 2. This is because Dataset 1 had few more important independent attributes like mileage, engine and max_power which helps in the prediction of the selling price with better accuracy. Dataset 2 lacked the above independent attributes and hence had a little bit lower accuracy compared to Dataset 1. Although we say that important attributes were missing, we still managed to get an above average accuracy in Dataset 2 by performing appropriate data pre-processing techniques.

Conclusion:

In this project, we have predicted the price of used cars using authentic second-hand car datasets by efficient data mining and machine learning techniques. In order to predict the price, we used Multiple Linear Regression, Random Forest and XGBoost as Regression models are very efficiently used in prediction. We came into a conclusion that on average, Random Forest performs the best for predicting the price of used cars given certain independent attributes. It gave an accuracy of 95% for Dataset 1 which performed better than both Multiple Linear Regression and XGBoost. For Dataset 2, it gave an accuracy of 81% coming second to XGBoost which gave an accuracy of 82%.

References:

1. C. Jin, "Price Prediction of Used Cars Using Machine Learning," 2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT), Chongqing, China, 2021, pp. 223-230, doi: 10.1109/ICESIT53460.2021.9696839.
2. N. Monburinon, P. Chertchom, T. Kaewkiriya, S. Rungpheung, S. Buya and P. Boonpou, "Prediction of prices for used car by using regression models," 2018 5th International Conference on Business and Industrial Research (ICBIR), Bangkok, Thailand, 2018, pp. 115-119, doi: 10.1109/ICBIR.2018.8391177.
3. M. Hankar, M. Birjali and A. Beni-Hssane, "Used Car Price Prediction using Machine Learning: A Case Study," 2022 11th International Symposium on Signal, Image, Video and Communications (ISIVC), El Jadida, Morocco, 2022, pp. 1-4, doi: 10.1109/ISIVC54825.2022.9800719.
4. Y. Li, Y. Li and Y. Liu, "Research on used car price prediction based on random forest and LightGBM," 2022 IEEE 2nd International Conference on Data Science and Computer Application (ICDSCA), Dalian, China, 2022, pp. 539-543, doi: 10.1109/ICDSCA56264.2022.9988116.
5. P. Ponmalar P and A. Christinal C, "Review on the Pre-owned Car Price Determination using Machine Learning Approaches," 2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 2022, pp. 274-278, doi: 10.1109/ICAISS55157.2022.10010958.
6. F. Wang, X. Zhang and Q. Wang, "Prediction of Used Car Price Based on Supervised Learning Algorithm," 2021 International Conference on Networking, Communications and Information Technology (NetCIT), Manchester, United Kingdom, 2021, pp. 143-147, doi: 10.1109/NetCIT54147.2021.00036.

7. C. V. Narayana, C. L. Likhitha, S. Bademiya and K. Kusumanjali, "Machine Learning Techniques To Predict The Price Of Used Cars: Predictive Analytics in Retail Business," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2021, pp. 1680-1687, doi: 10.1109/ICESC51422.2021.9532845.
8. J. Varshitha, K. Jahnavi and C. Lakshmi, "Prediction Of Used Car Prices Using Artificial Neural Networks And Machine Learning," 2022 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2022, pp. 1-4, doi: 10.1109/ICCCI54379.2022.9740817.
9. H. Zhang, "Prediction of Used Car Price Based on LightGBM," 2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Wuhan, China, 2022, pp. 327-332, doi: 10.1109/AEMCSE55572.2022.00073.
10. K.Samruddhi¸ Dr. R.Ashok Kumar, "Used Car Price Prediction using K-Nearest Neighbor Based Model", International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE), Volume 4, Issue 3, DOI: 10.29027/IJIRASE.v4.i3.2020.686-689, September 2020
11. Bukvić, L.; Pašagić Škrinjar, J.; Fratrović, T.; Abramović, B. Price Prediction and Classification of Used-Vehicles Using Supervised Machine Learning. Sustainability 2022, 14, 17034.
12. AlShared, Abdulla, "Used Cars Price Prediction and Valuation using Data Mining Techniques" (2021). Thesis. Rochester Institute of Technology.
13. Trivendra, Chandraprakash. "The Price Prediction for Used Cars Using Multiple Linear Regression Model." International Journal for Research in Applied Science and Engineering Technology 8.5 (2020): 1801–1804.
14. Yadav, A., Kumar, E., & Yadav, P. K. (2021). Object detection and used car price predicting analysis system (UCPAS) using machine learning technique. Linguistics and Culture Review, 5(S2), 1131-1147. https://doi.org/10.21744/lingcure.v5nS2.1660
15. T P Jayadeera, D J Jayamanne, "Fair Price Prediction System for Used Cars in Sri Lanka Using Machine Learning and Robotic Process Automation", 2019 International Conference On Business Innovation (ICOBI), 22 November, Colombo, Sri Lanka.
16. Kumar, G. K., & Murthy, G. R. K. (2018). Predictive Models for Used Car Price Estimation: A Comparative Study. International Journal of Advanced Computer Science and Applications, 9(9), 491-497
17. Kumar, R., & Tiwari, S. (2020). Used Car Price Prediction using Random Forest Regression. 2020 6th International Conference on Computing, Communication and Security (ICCCS), 1-5.
18. Suresh, S., & Suresh Kumar, S. (2020). Predicting Used Car Prices Using Machine Learning Techniques. 2020 International Conference on Smart Electronics and Communication (ICOSEC), 1-5.
19. Islam, M. R., Hasan, M. M., Hasan, M. K., Ahmed, A., & Mahmud, T. (2019). A Comparative Study of Linear and Non-linear Regression Models for Used Car Price Prediction. 2019 22nd International Conference on Computer and Information Technology (ICCIT), 1-6.
20. Barani, G., Kumar, G. R. S., & Prasad, R. (2018). Comparison of Machine Learning Techniques for Predicting Used Car Prices. 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 1710-1714.

21. Roy, S., Chakraborty, S., & Mukhopadhyay, S. (2018). Prediction of Used Car Price using Machine Learning: A Case Study on the Indian Automobile Market. 2018 8th International Conference on Cloud Computing, Data Science & Engineering - Confluence, 555-558.
22. Kumari, S., & Suresh Kumar, S. (2021). A Comparative Study of Machine Learning Models for Used Car Price Prediction. International Journal of Machine Learning and Computing, 11(1), 23-28.
23. Vu, H.T., Nguyen, D.T., Pham, Q.T., & Nguyen, T.H. (2019). Prediction of Used Car Prices Using Machine Learning Techniques. Proceedings of the 2019 IEEE-RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future, pp. 1-6. doi: 10.1109/RIVF.2019.8935846.
24. Lee, S., & Cho, S. (2020). Predicting Used Car Prices with Machine Learning Techniques. Journal of Advanced Research in Dynamical and Control Systems, 12(7), 523-530.
25. Kim, J., Park, J., & Lee, K. (2021). Used Car Price Prediction with Feature Engineering and Gradient Boosting Regression. Symmetry, 13(5), 789. doi: 10.3390/sym13050789.
26. Neha B, Nishant V, Nikhil K (2023, January). Vehicle dataset Retrieved from https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho?select=Car+details+v3.csv
27. Riyuzaki U (2022, May). Car_Price_Prediction Retrieved from https://www.kaggle.com/datasets/raihansoniwala/quikr-cars