University of Waterloo
Faculty of Engineering

ME 780: Computational Intelligence

Automobile Price Prediction Using Various Machine Learning Algorithms

Mohammed Ibrahim Shariff (21041214)
Saurav Bin Saji (21041330)

Masters in Electrical and Computer Engineering
13th December 2022

Acknowledgment

We would like to extend our thanks and appreciation to **Dr. William Melek**, for his constant encouragement, appreciation and timely advice during the implementation of this project.

His constructive skilled guidance during the tenure of the course has been of great help in the successful completion of the work.

Table of Content

List of Figures

List of tables

List of Codes

1. Executive Summary:

Source Code: https://colab.research.google.com/drive/1C5I19__hApZxqeNI-gru8KayJYilyvCO

Used vehicle price forecast is a hot issue because of the record amount of automobiles that have been bought and sold in the last five years as a result of the semiconductor scarcity and rising mortgage rates. People tend to buy used automobiles more frequently in developing nations since they are more affordable. Because used car prices fluctuate on the market, both buyers and sellers need an intelligent system that will enable them to accurately estimate the price. The collecting of the dataset, which includes all crucial information such as the car's production year, gas type, condition, mileage, horsepower, doors, the number of times it has been painted, customer reviews, the car's weight, etc., is the most challenging task facing this intelligent system.

A used car's market price is influenced by a variety of factors, making it difficult to determine whether the advertised price is accurate. The goal of this project is to build machine different learning models that can precisely forecast a used car's price based on its attributes that are highly correlated with a label (Price), and predict which model performs the most accurate price of the car so that the buyers can make an educated decisions. To accomplish this, different regression algorithms has been employed in the Machine Learning Model. Null, redundant, and missing values were removed from the dataset during pre-processing. In this supervised learning study, five regressors (Random Forest Regressor, Linear Regression, Artificial Neural Network, K nearest Neighbour Regressor and XGBoost) have been trained, tested, and compared against a benchmark dataset. Among all the experiments, the XGBoost Model had the highest score at 95%, followed by 0.01 MSE, 0.06 MAE, and 0.1 RMSE respectively. In addition to XGBoost, Random Forest algorithm performed well with an 94% score, followed by ANN Regression having an 88% mark. The researchers of this project anticipate that in the near future, the most sophisticated algorithm is used for making predictions, and then the model will be integrated into a mobile app or web page for the general public to use.

Key Words: Car Price Prediction, supervised learning, linear regression, Artificial Neural Network, K Nearest neighbors, Random Forest, XGBoost, R2 Score, MSE, MAE, RMSE

2. Approach:

The project deals with the automobiles based in America. This data was obtained using web scraping techniques on https://www.cargurus.com/. This data is for academic, research and individual experimentation only and is not intended for commercial purposes. This dataset contains 3 million new and used vehicle listings in the United States.

After data collection the dataset was pre-processed to remove samples that have missing value, and remove non-numerical part from numerical attributes, converting categorical values into numerical (if needed), fix any discrepancies in the units, as well as removing attributes that doesn't affect the price evaluations if needed to reduce the complexity of the model. Data Understanding and preparation is an essential part of building a model as it gives the insight into the data and what corrections or modifications shall be done before designing and executing the model, preliminary analysis of the data must be done to have deeper understanding into the quality of the data, in terms of outliers and the skewedness of the figures, descriptive Statistics of categorical and numerical variables was done for that to be achieved. As well as the ability to understand the main attributes that affect the results of the price. That was done through a correlation matrix for every attribute to understand the relations between the different factors.

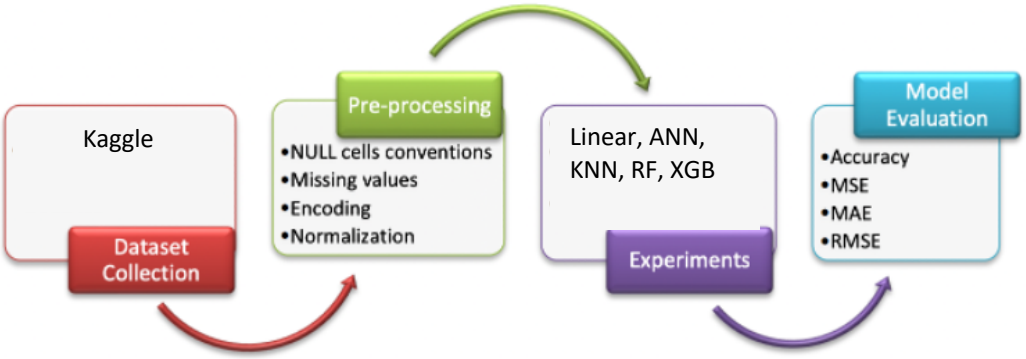

Fig 1: Methodology of the Project

Afterwards when the data is organized and transformed into a form that could be processed by the data mining technique. Different data mining models were designed to predict prices and values of used cars. In this study three models are proposed to be built using Logistic Regression model

technique, Random Forest Regressor and Bagging Regressor. Firstly, the data was portioned into section for training and the other part for testing, portioning percentage can be tested with different ratios to analyse different results. All three models were evaluated on four evaluation matrices known as model score, Mean Square Error (MSE), Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). From all, the Random Forest Regressor outperformed.

3. Literature Review:

Several studies and related works have been done previously to predict used car prices around the world using different methodologies and approaches, with varying results of accuracy from 50% to 90%. In (Pudaruth, 2014) the researcher proposed to predict used car prices in Mauritius, where he applied different machine learning techniques to achieve his results like decision tree, K-nearest neighbours, Multiple Regression and Naïve Bayes algorithms to predict the used cars prices, based on historical data gathered from the newspaper.

Achieved results ranged from accuracy of 60-70 percent, the author suggested using more sophisticated models and algorithms to make the evaluation, with the main weakness off the decision tree and naïve Bayes that it is required to discretize the price and classify it which accrue to more inaccuracies. Moreover, he suggested a larger set of data of data to train the models hence the data gathered was not sufficient.

(Monburinon, et al., 2018) Gathered data from a German e-commerce site that totalled to 304,133 rows and 11 attributes to predict the prices of used car using different techniques and measured their results using Mean Absolute Error (MEA) to compare their results. Same training dataset and testing dataset was given to each model. Highest results achieved was by using gradient boosted regression tree with a MAE of 0.28, and MEA of 0.35 and 0.55 for mean absolute error and multiple linear regression respectively. Authors suggested adjusting the parameters in future works to yield better results, as well as using one hot encoding instead of label encoding for more realistic data interpretations on categorical data.

(Gegic, Isakovic, Keco, Masetic, & Kevric, 2019) from the International Burch University in Sarajevo, used three different machine learning techniques to predict used car prices. Using data scrapped from a local Bosnian website for used cars totalled at 797 car samples after pre-processing, and proposed using these methods: Support Vector Machine, Random Forest and Artificial Neural network. Results have shown using only one machine learning algorithm achieved results less than 50%, whereas after combing the algorithms with pre calcification of prices using Random Forest, results with accuracies up to 87.38% was recorded.

(Noor & Jan, 2017) were able to achieve high level of accuracy using Multiple linear regression models to predict the price of cars collected from used cars website in Pakistan called Pak Wheels that totalled to 1699 records after pre-processing, and where able to achieve accuracy of 98%, this was done after reducing the total amount of attributes using variable selection technique to include significant attributes only and to reduce the complexity of the model.

4. Project Description:

Determine the market price of a used automobile accurately by factoring in various features that drive the current value. The purpose of this study is to determine the best model that can precisely predict the cars price and determine the challenges faced in each model and how to overcome them. Each regression model is trained using a set of data and then optimized by tuning the various hyperparameters. The various regression models used are:

1. Linear Regression
2. K-Nearest-Neighbours
3. Artificial Neural Networks
4. Random Forest
5. XGBoost

The results from these models were compared, and the best model should be predicted.

5. Motivation

Problem 1: In the past few years, the problems such as global chip shortage in the semiconductor industry and increase in mortgage rates have cropped up. To address these issues, our car price prediction model will help buyers in understanding the best value. The chip shortage has probably had the most effect on the automotive sector. The typical automobile can have more than 100 chips on board, and many vehicles need thousands of semiconductors to operate safety features, the electrical and engine systems, entertainment, connection, and more. This depends on the extent of connectivity. The cost of new autos is also rising due to chip shortages. In 2021 and 2022, the average price of a new car reached all-time highs.

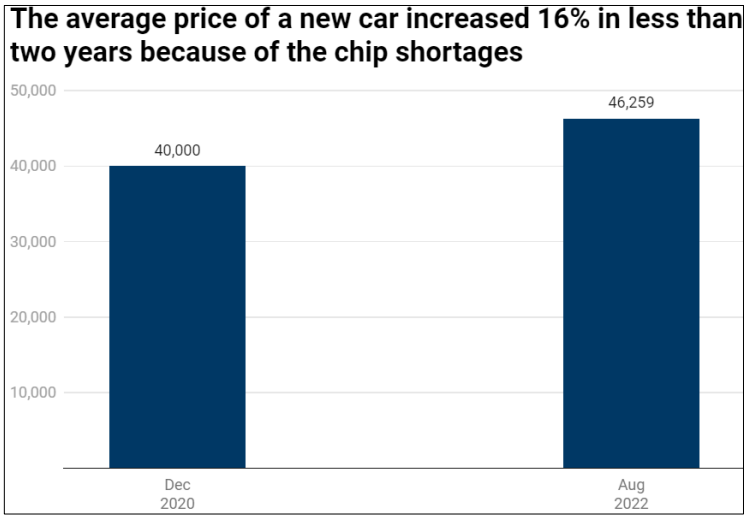

Fig 2: Global Chip shortage

Problem 2: Since the Federal Reserve raised the prime interest rate in October and November 2022, car loans have grown more expensive. In October 2022, the average interest rate on a car loan was 10.6%, about double what it was in the beginning of 2022.

However, the story is more complex than just interest rates. Due to supply chain issues that automakers are experiencing, cars are now more expensive than they were before the outbreak. In the long run, high interest rates combined with high pricing may make new cars expensive for low- and middle-income households.

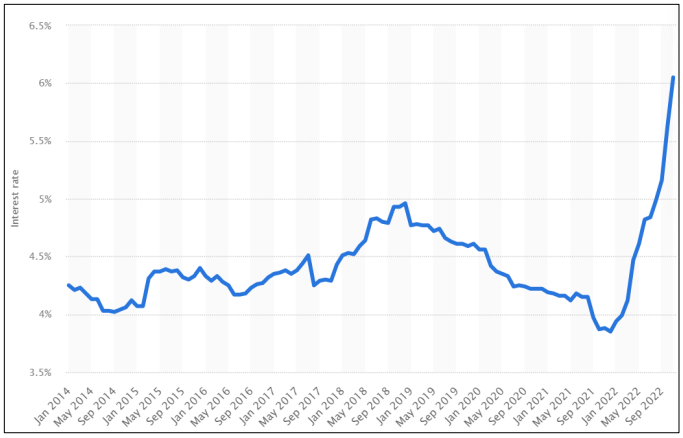

Fig 3: Year wise comparison of the interest rates

Problem 3: Environmental issues, the carbon footprint of a new car are as follows:
6 tonnes CO2e: Citroen C1, basic spec, 17 tonnes CO2e: Ford Mondeo, medium spec
35 tonnes CO2e: Land Rover Discovery, top of the range

A car's carbon footprint is incredibly intricate. The best we can do is break down the known total emissions of the world or a country into several industries and sectors using so-called input-output analysis, while also taking into consideration how each industry uses the products and services of all the others. This results in a footprint of 720 kg CO2e every £1000 spent when we split the overall emissions of the auto sector by the whole amount of money spent on new cars. Thus, it is advisable to purchase a used car or keep your current vehicle on the road rather than switching to a new model because making a new car produces the same amount of carbon pollution as driving one.

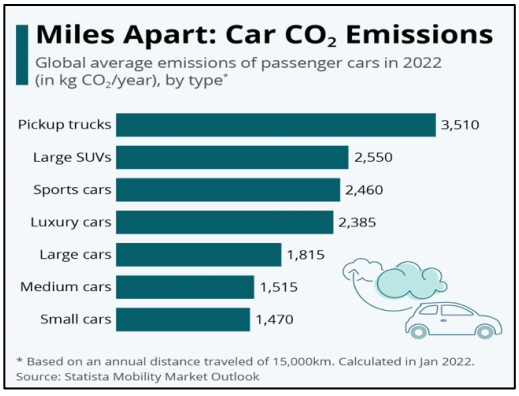

Fig 4: CO2 Emissions from new cars

6. Dataset Description

The dataset is obtained from the Kaggle website. Through multiple runs and irritations 100,000 rows of data with 26 features were collected successfully. Initially, pre-processing was used to individually fix or convert the data types of each attribute. The following table provides details and a description of the dataset:

Features	**Description**
VIN	This value is used to identify unique data points
Body Type	We categorise passenger cars based on body types (coupes, sedans, hatchbacks, etc)
Days on Market	Determines the number of days from the day the auto has been advertised
Engine Cylinders	An essential component of the engine is a cylinder. It is a chamber where fuel is used to generate power. There are 4 in a 4 cylinder, 6 in a V6, etc.
Engine Displacement	It is a measurement of the cylinder volume swept by all of the pistons in a piston engine.
Engine Type	Similar to engine cylinders
Exterior Color	The single or combination of colours on the exterior of a car
Frame Damage	Its value is True if the frame of the body is damaged
Fuel Tank Volume	The total fuel capacity of the fuel tank in litres. Its value is determined for non-electric vehicles
Fuel Type	It determines the type of fuel used by the vehicle to generate power. It takes values such as gasoline, diesel or electric
Has Accidents	It's a Boolean value which informs if the vehicle has suffered any accidents
Horsepower	The numeric value determines the power in HP
Is New	The Boolean value determines if the car is in new condition
Major Options	It is a list of unique features of the car
Make Name	The make of a car is its brand, such as Chevrolet, Land Rover, BMW, etc
Maximum Seating	The number of passengers that can be seated in the car

Milage	It determines the number of miles the auto has travelled.
Model Name	It is the model of the given automobile maker
Owner Count	The number of owners who have owned the car previously including the current owner of the car
Seller Rating	The rating given by the seller out of 5
Torque	This value gives the torque generated by the auto
Transmission	The transmission gear system present in the auto. It is either Automatic (A), or Manual (M).
Transmission Display	The display that shows the transmission mode and the gear on which the car is running
Wheel System	It depicts the wheel system in the automobile. Example Front Wheel Drive (FWD), All Wheel Drive (AWD) and Rear Wheel Drive (RWD)
Year	The Year in which the car was bought
Price	The target variable for the model which is predicted based on the above input features.

Table 1: Automobile Features and Description

7. Exploratory Data Analysis (EDA)

EDA is the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations. It consisted of the following below mentioned step in our project.

7.1 Data Cleaning and Data Wrangling

The following steps were involved in Data Cleaning Process:

- Deleting redundant rows
- Detecting multicollinearity and eliminating it.
- Replacing missing values and NaN values with useful information

- Investigating outliers and dealing with them
- Adjusting the column values and data types
- Aspect Engineering (creating new features from existing features)
- Data preparation for the model (scaling, encoding, etc.)
- Exploring the data to discover insights and respond to the relevant queries
- Eliminating useless columns from the model
- Preserving the cleansed data for the project's subsequent phases

Some of the major features of the model are as shown below:

1) Fig 4 shows the missing values detected in each feature of the dataset, they were further replaced with useful information.

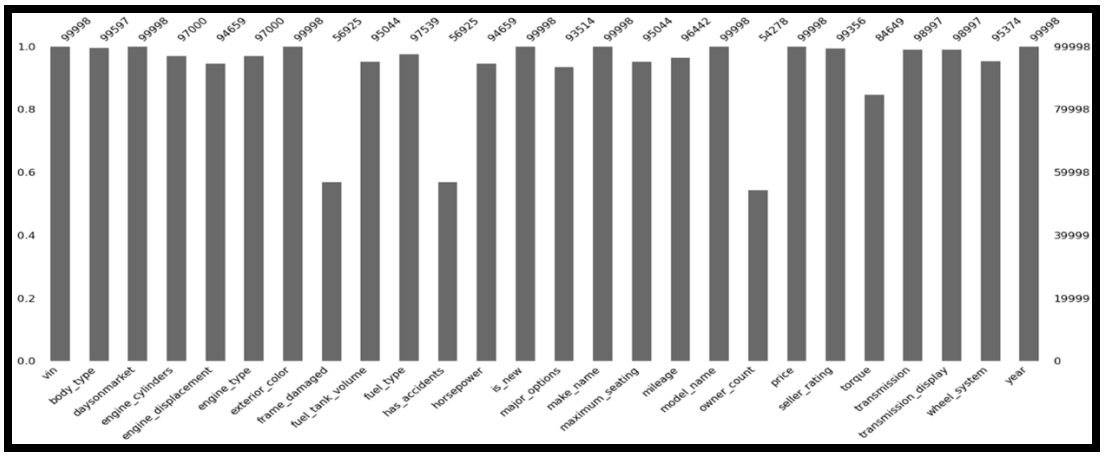

Fig 5: Visual representation of total data points in individual feature

2) Visual Representation of available body types on the Dataset. Fig-10 shows that the mode of attribute Body Type is SUV/Crossover.

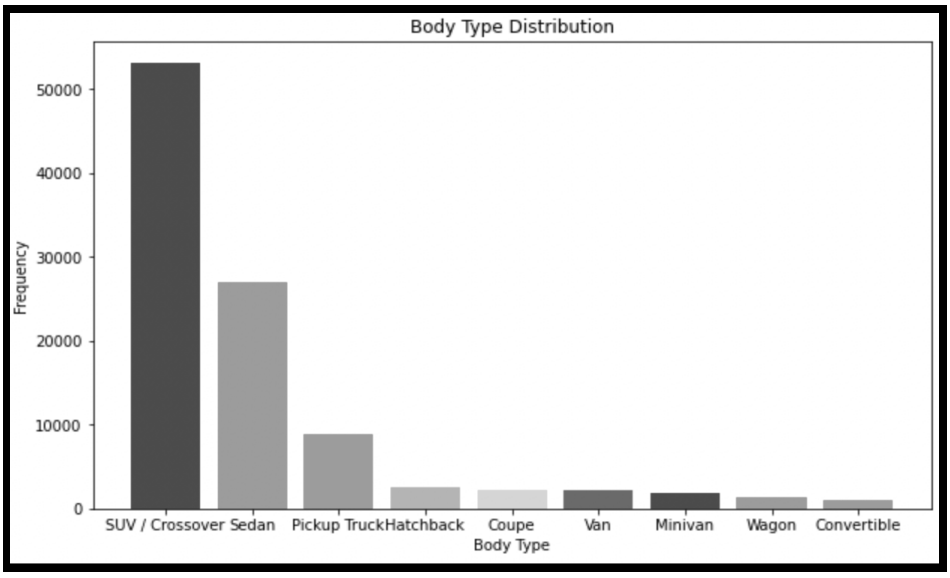

Fig 6: Body Type Distribution of Different Automobiles

3) Listing all available colors of the car in the dataset and grouping them as major ones. Fig shows that the most occuring color individually was black.

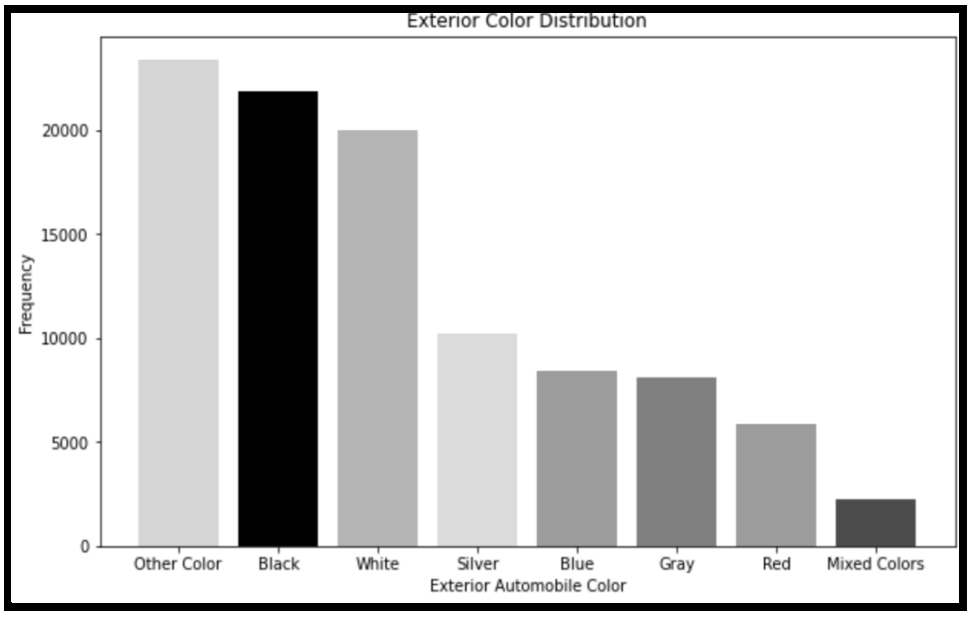

Fig 7: Exterior Color Distribution of the Automobiles

4) Figure 7 shows that the Gasoline has the highest fuel distribution amongst all the cars.

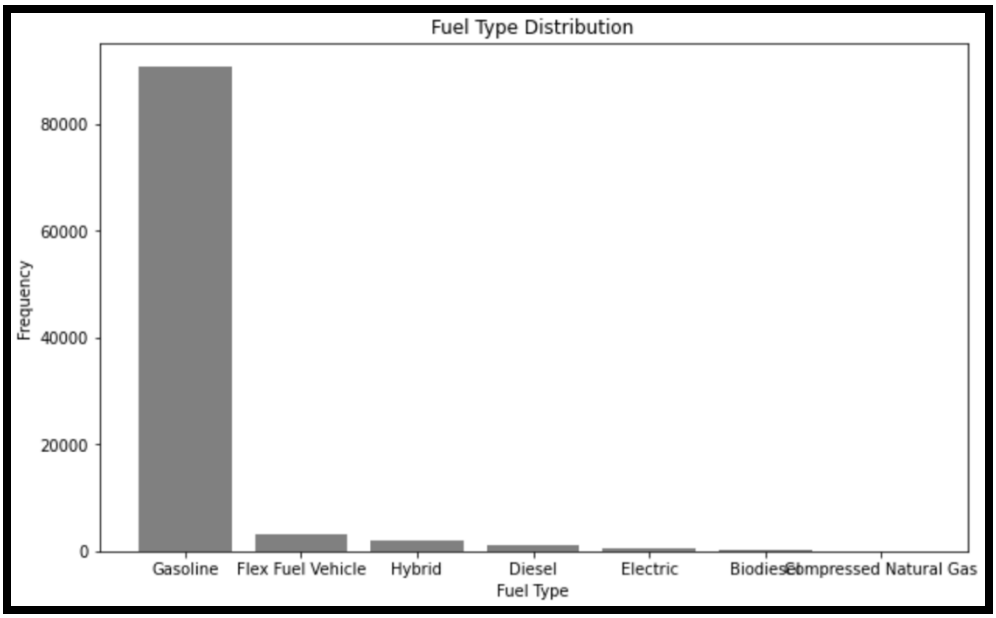

Fig 8: Fuel Type Distribution of the Automobiles

5) As shown in figure 8, most of the data lies in the range of years 2004-2021. We decided to keep the outliers to have a view of all the different classic cars.

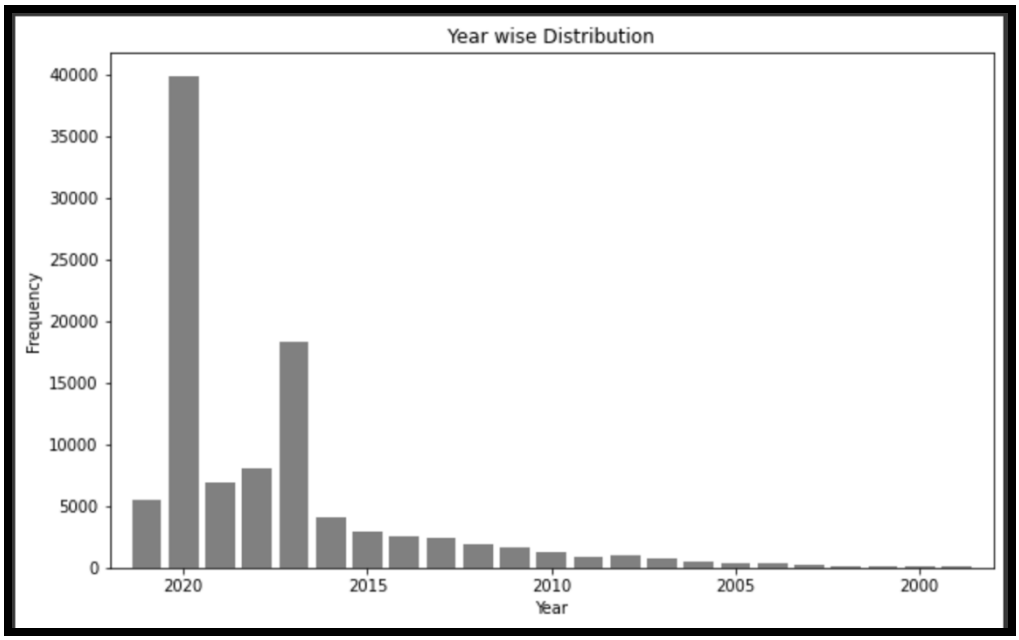

Fig 9: Year wise distribution of the Automobiles

6) The graph demonstrates that the column contains a significant number of outliers. Therefore, before replacing the null values with the mean or median value, outliers have been removed. To eliminate the outliers, interquartile range has been used.

```
# using the interquartile range to remove outliers
q1 = cars_df2['mileage'].quantile(0.25)
q3 = cars_df2['mileage'].quantile(0.75)
# calculating the interquartile range
iqr = q3 - q1
# removing outliers
cars_df2 = cars_df2[(cars_df2['mileage'] >= q1 - 1.5*iqr) & (cars_df2['mileage'] <= q3 + 1.5*iqr)]
```

Code 1: Depicts the process of removing outliers using interquantile

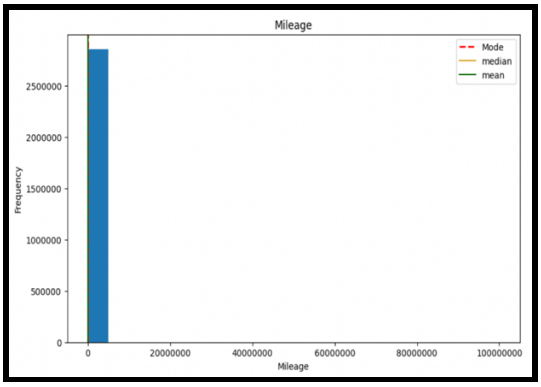

Fig 10a: Mileage of the car before removing outliers

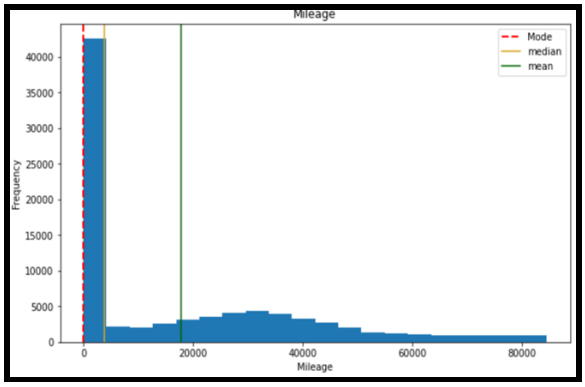

Fig 10b: Mileage of the car after removing outliers

It appears that most cars have a mileage of zero miles. This is because the vehicles are new. In this instance, the mean value of the column can be used to fill in the null values.

7) From fig, it can be visualized that Ford is the most occurring Brand from the dataset.

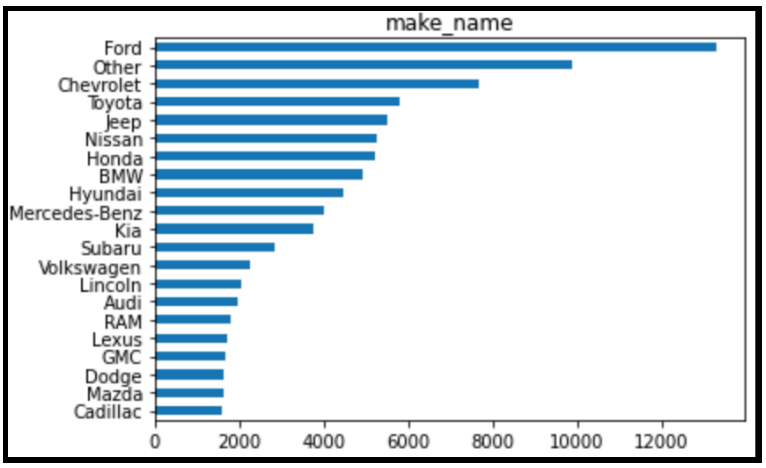

Fig 11: Make Name distribution of the Automobiles

8) The seller rating is the rating of the dealership that is selling the vehicle. This is a rating from 1 to 5 stars. We can see that the seller_rating column is heavily skewed to the top. This means that there are a large amount of 4 star ratings which tells us that most sellers have a high rating.

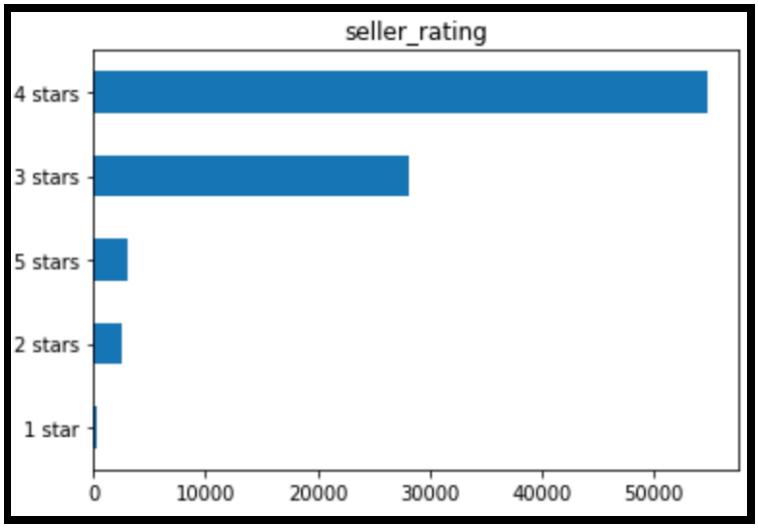

Fig 12: Seller Rating Distribution given in the dataset

9) Visualization of Transmission type is as shown, most automobiles are Automatic, followed by CVT and Manual.

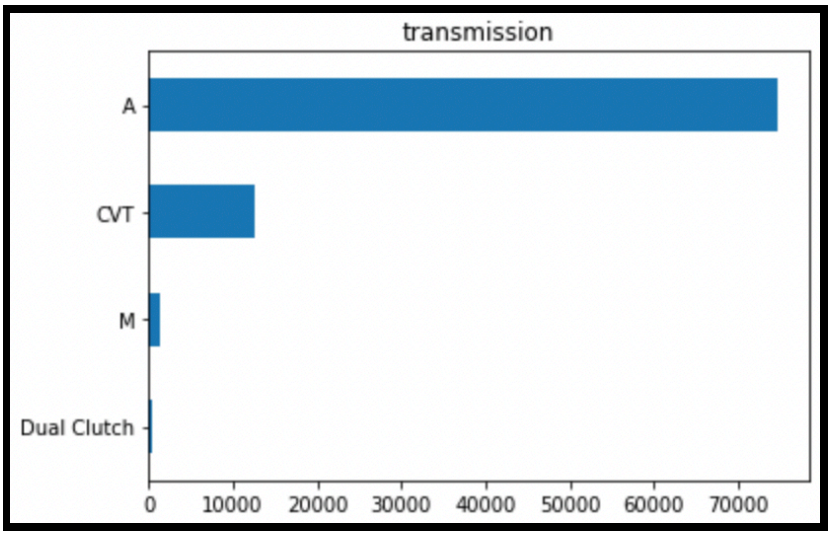

Fig 13: Transmission Distribution of the Cars in the Dataset

10) Visualization of Torque is as shown, most vehicles have Torque in the range 200 – 400 (N-m)

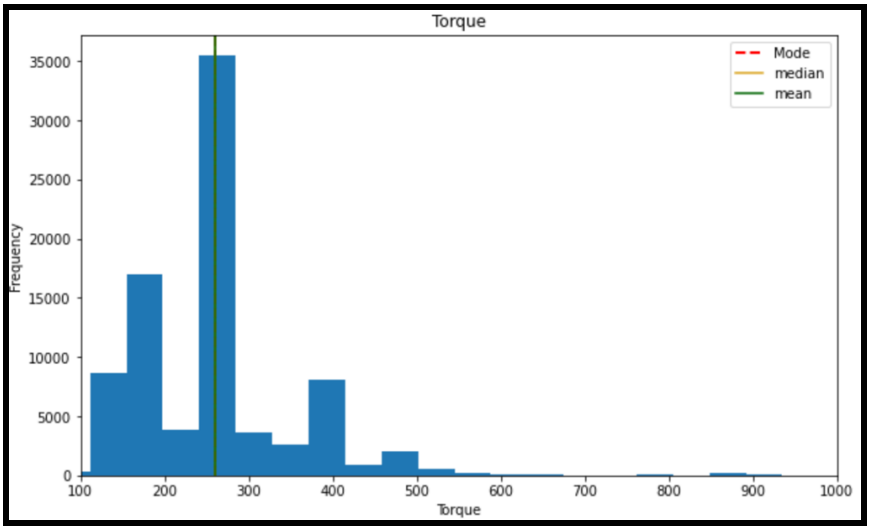

Fig 14: Torque Distribution of the Cars in the Dataset

11) The graph shows that our target variable price is right skewed. This means that the majority of the values are clustered on the left side of the graph. This is not ideal for our model because it will be biased towards the lower values. Thus, the target variable has been transformed to make it more normal.

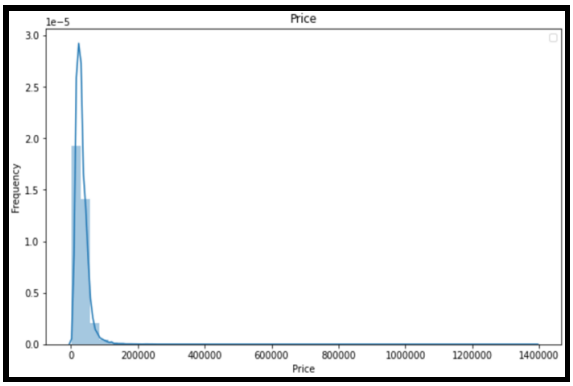

Fig 15a: Price of the car before Normalization

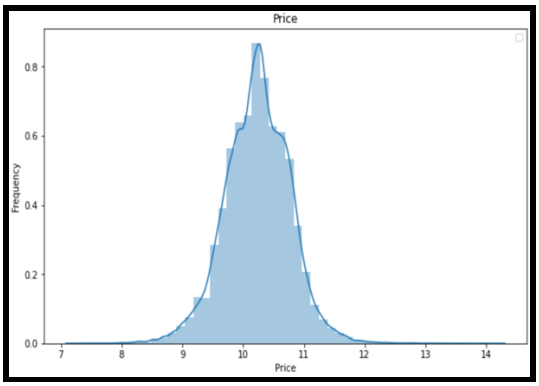

Fig 15b: Price of the car after Normalization

7.2 Dataset pre-processing

Pre-processing is a Data Mining approach that entails putting unstructured data into a format that can be understood. Real-world data frequently lacks particular information on activity or trends and also contains erroneous information. As a result, this can lead to bad data gathering, which could then lead to poor models that are built using the data. The data can be pre-processed to address such issues. Input modification or encoding is the process of pre-processing in machine learning, which makes data easier for the computer to parse. The algorithm can now correctly comprehend the data as a result.

The following methods can be used to detect multicollinearity.

1. **Pairwise Correlations:** Examining the pairwise correlations between various pairs of independent variables might provide useful information in diagnosing multicollinearity.

2. **Variance Inflation Factor (VIF):** Pairwise correlations are not always beneficial because one variable may not be able to entirely explain another variable, but other variables combined may be able to do so. Thus, VIF can be used to test various types of variable-variable relationships. VIF illustrates the relationship between one independent variable and all other independent variables. VIF is provided by,

$$\text{VIF} = \frac{1}{1-R^2}$$

where i refers to the i^{th} variable which is being represented as a linear combination of the rest other independent variables. The common heuristic followed for the VIF values is if VIF > 10 then the value is definitely high and it should be dropped. And if the VIF=5 then it may be valid but should be inspected first. If VIF < 5, then it is considered a good vif value.

In this model, there is no multicollinearity between the numerical columns. This means that all of the numerical columns in the model can be used. The encoding Technique is used to turn categorical data into numerical data since machine learning algorithms cannot process categorical data. One hot encoding is used in this project.

Now that we have explored the data, we can begin to encode the categorical columns. We will use One Hot Encoding to encode the categorical columns. This will allow us to use the categorical columns in our model.

```
# looping through the categorical columns and using get_dummies to create dummy variables
for col in cat_col:
    dummies = pd.get_dummies(cars_df2[col], prefix=col, drop_first=True)
    cars_df2 = pd.concat([cars_df2, dummies], axis=1)
    cars_df2.drop(col, axis=1, inplace=True)

# checking the result
cars_df2.head()
```

Code 2: To convert the categorical data to numerical Values

vin	Unnamed: 0	daysonmarket	fuel_tank_volume	horsepower	maximum_seating	mileage	price	torque	year	body_type_Coupe
ZACNJABB5KPJ92081	0	522	12.700	177.000	5	7.000	10.049	200.000	2019	0
SALCJ2FX1LH858117	1	207	17.700	246.000	7	8.000	10.747	269.000	2020	0
SALRR2RV0L2433391	3	196	23.500	340.000	7	11.000	11.119	332.000	2020	0
SALCJ2FXXLH862327	4	137	17.700	246.000	7	7.000	10.797	269.000	2020	0
SALYK2EX1LA261711	5	242	16.600	247.000	5	12.000	11.111	269.000	2020	0

Table 2: Categorical Data converted to Numerical values

Table 2 validates that all categorical features have been converted into numerical features.

Checking the correlation between the numerical columns:

```
# instantiating the correlation matrix
corr_df = cars_df2.corr()

# Create a mask to only show the lower triangle of the correlation matrix
mask = np.triu(corr_df)

# Set up the matplotlib figure
plt.figure(figsize=(15, 10))
sns.heatmap(corr_df.round(2), annot=True, vmax=1, vmin=-1,
            center=0, cmap="coolwarm", mask=mask, linewidths=1)
plt.show()
```

Code 3: To display the correlation matrix

After data transformation, it is time to look for attribute associations. To determine whether or not relationships exists, a correlation matrix was used.

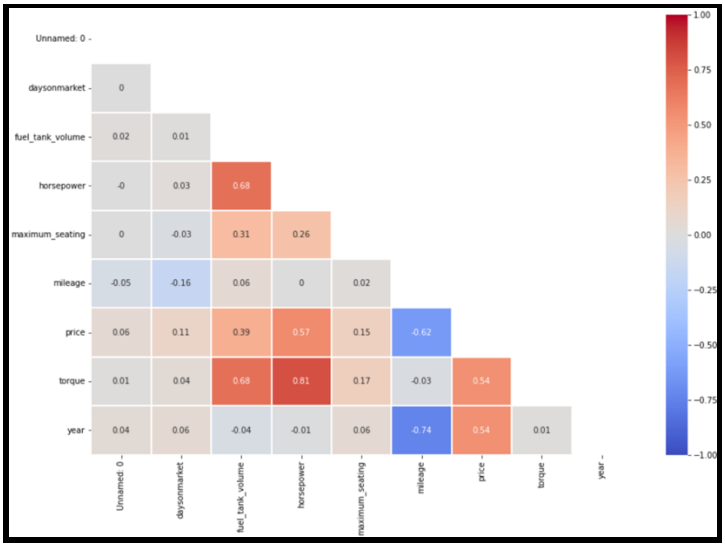

Fig 16: Correlation Matrix of the Car

Observations from the Correlation Matrix:

- The torque of the vehicle has the highest positive correlation with the target variable. This means that the faster the vehicle, the more expensive it is.
- The horsepower of the vehicle has a high positive correlation with the target variable. This means that the higher the horsepower of the car, the more expensive it is.
- We can see the lowest negative correlation with mileage. This means that the more miles the vehicle has, the lower the price of the vehicle.
- What year the vehicle was made has a high positive correlation with the target variable. This means that the newer the vehicle, the more expensive it is.
- We can also see that the engine type has an effect on the price of the vehicle.
- If the vehicle has an incident, it will lower the price of the vehicle.
- Brand names have an effect on the price of the vehicle. The more popular the brand name, the more expensive the vehicle.

The following conclusions were drawn from our Explanatory Data Analysis (EDA):

- The most common vehicles that are listed are SUV / Crossover, sedans, and pickup trucks.
- The most common vehicle make is Ford, Chevrolet, and Toyota.
- The Most common type of engine is an I4 engines, followed by V6's and V8's.
- The most common exterior colors are Black, White and Silver.
- At least 11% of vehicles have been in an accident, damaged, salvaged, or had been stolen.
- The average fuel tank size is 26 gal. Not including Electric vehicles and hybrids.
- Gas power vehicles represent 86% of the data.
- The average horsepower of a vehicle falls between 180 – 200 horsepower
- The average torque speed is 264 lb-ft.
- The average vehicle has 4-6 major options installed.
- The most common seating capacity is 5 seats, followed by 7 seats, and 6 seats.
- Majority of the vehicles on this list is new. Showing zero mileage.
- The most common transmission is an Automatic transmission, followed by a Manual transmission.

- The top wheel systems in order are: FWD, AWD, 4WD, RWD, and 4X2.
- The average age of a vehicle is 2 years old.
- The most popular brand names are Ford, Chevrolet, and Toyota.

8. Model Building

8.1 Linear Regression (Base Model):

Linear regression is a base regression learning algorithm used to compare the performance with respect to other models. The model is trained on the linear combination of features of the input vectors. To calculate best-fit line linear regression uses a slope-intercept form which is given below,

$$Y_i = \beta_0 + \beta_1 Xi$$

Where, Y_i = Dependent variable, β_0 = constant/Intercept, β_1 = Slope/Intercept, X_i = Independent variable.

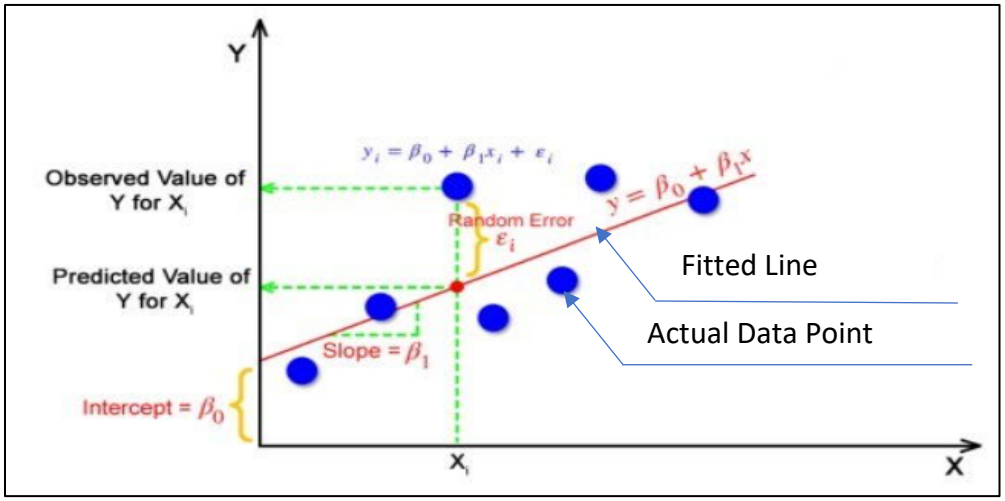

Fig 17: Graph showing actual and predicted points using Linear Regression

There are many factors to consider in Linear Regression such as the following:

Overfitting: This is the phenomena where the model consists of too many variables, it may become overly complicated and the model ends up memorizing all the values from the training dataset. It results in low test accuracy and high training accuracy.

Multicollinearity: It is the situation where multiple independent variables may have some inter dependencies. Multicollinearity makes it difficult to determine which variable is truly influencing the prediction of the response variable, which makes it easy to draw the wrong conclusions about a variable's effects on the target variable. The presence of multicollinearity in the model must be properly identified and dealt with even though it has no impact on the accuracy of the predictions. This is because when one of these correlated variables is arbitrarily removed from the model, the coefficient values can swing wildly and even change signs. Since we don't have high multicollinearity in our data we need not worry about this factor.

Overfitting and Underfitting in Linear Regression

One of the common flaws of regression models is overfitting and underfitting.

- Bias is a way to gauge how accurate a model is likely to be with respect to upcoming, unforeseen data. If there are sufficient training data available, complex models can make precise predictions. While too naive models are very likely to underperform in terms of predictions. Bias is just inaccuracies in the training data.
- In general, linear algorithms are less flexible but have a high bias that makes them quick to learn and easy to grasp. lower prediction performance on difficult issues that don't produce the desired results
- For our car price prediction model, low bias and low variance should be achieved for the best performance
- The optimum parameter combination is generated using scikit's GridSearchCV, learn's which iterates through all the different input parameters and does so based on a score criteria. Below is the set of best parameters obtained from GridSearchCV for Linear regression

```
{'copy_X': True, 'fit_intercept': True, 'n_jobs': -1, 'normalize': True}
```

Code 4: Hyperparameters of Linear Regression

The following are the metrics obtained after tuning the hyperparameters:

```
Train score: 0.8306195520377068
Test score: 0.8275485041804996
MSE: 0.049142338364299314
RMSE: 0.22168071265741482
MAE: 0.1578162674626966
```

8.2 K-Nearest-Neighbours (KNN) Regression:

KNN is non-parametric model while compared to Linear regression which is parametric model. In KNN, we use K nearest neighbours to approximate the price of the car. KNN is computationally heavy and usually requires time to compute as it has to keep real time track of all the training data and find the neighbouring nodes. If we train the model over the entire dataset, it takes about 20 minutes to build the model. Hence, we train the model using only 10% of the training data. It takes a few seconds to train the model and following are the metrics obtained

```
Train score: 0.9121576742992203
Test score: 0.8523592569125434
MSE: 0.04270028716075994
RMSE: 0.206640478030709
MAE: 0.13321558141742787
```

The model will have better results when evaluated over the entire dataset than on the sampled one but, this comes at the cost of computation time. The performance of this model can be improved by tuning the hyperparameter. Since KNN uses the K-nearest neighbours to predict the price, it is very important to select the optimal value of K (The number of neighbours)

```python
# list of K values to try
neighbors_settings = range(1, 11)

# looping through the K values on scaled data
for n_neighbors in neighbors_settings:
    # building the model
    knn = KNeighborsRegressor(n_neighbors=n_neighbors, n_jobs=-1)
    knn.fit(X_train_scaled, y_train)

    # recording the training set accuracy
    training_accuracy.append(knn.score(X_train_scaled, y_train))

    # recording the generalization accuracy
    validation_accuracy.append(knn.score(X_val_scaled, y_val))
```

Code 5: Tuning the K value in KNN model

On plotting the training accuracy for different values of K, the validation accuracy is noted and plotted.

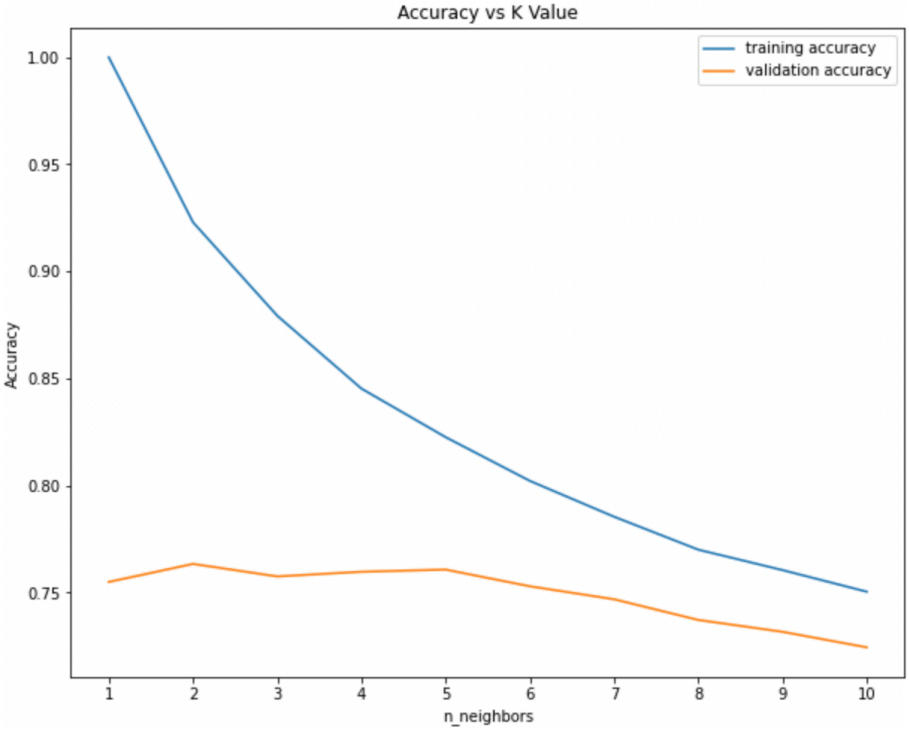

Fig 18: Accuracy VS K value for KNN

```
# the optimal value for K
optimal_k = neighbors_settings[validation_accuracy.index(max(validation_accuracy))]
print('The optimal value for K is: ', optimal_k)
The optimal value for K is:  2
```

Code 6: Determining the optimal K value in the KNN model

Building the model for optimal value of K, we get the following results:

```
Train score: 1.0
Test score: 0.9103153458739524
MSE: 0.025556830331935107
RMSE: 0.15986503786611728
MAE: 0.09857698962757659
```

We can observe that KNN performs better than Linear Regression model.

8.3 Random Forest

Random Forest is a bagging model. By averaging the results of various models, this model is utilized to lower its variance. It accomplishes this by building several decision trees and averaging their forecasts. We don't have to scale the data as this model is interdependent by the range of data.

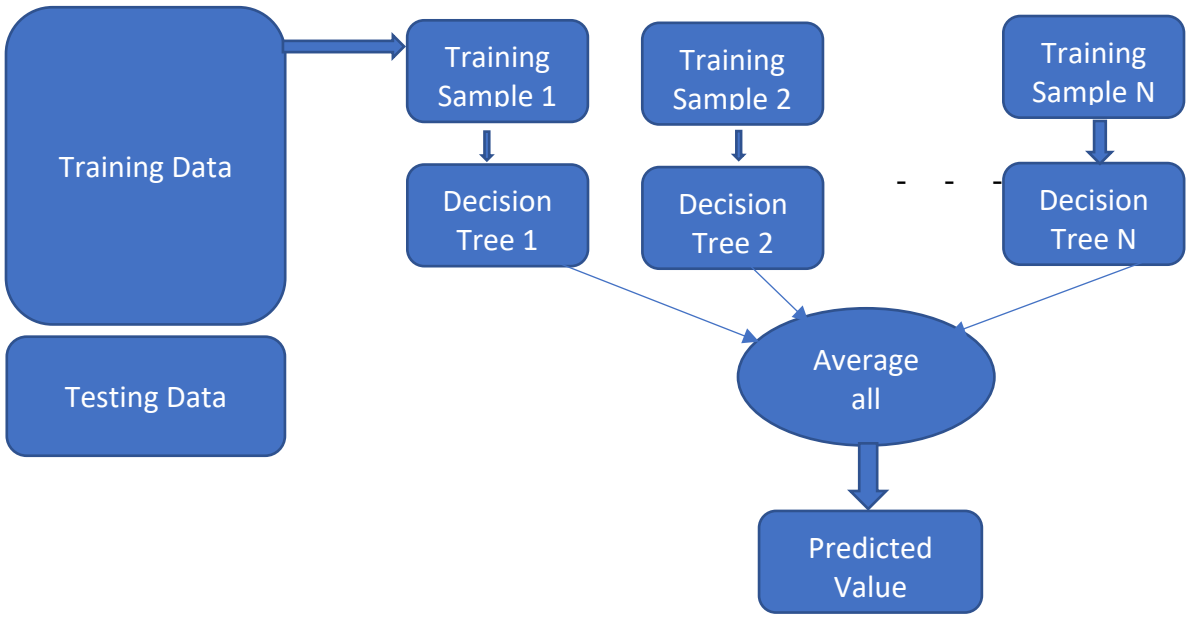

Fig 19: Random Forest Model

It does so by implementing two concepts:

Ensemble Learning: Random Forest used the concept of ensemble learning where in we use multiple trained models over the same data, averaging the output of each model to find a more accurate predictive system.

Bootstrapping: is the process of randomly sampling subsets of a dataset over a given number of iterations and a given number of variables. These results are then averaged together to obtain a more powerful result. Bootstrapping is an example of an applied ensemble model.

Some are the important parameters of Random Forest:

n_estimators: The number of decision trees that will be running in the model.

max_depth: The maximum possible depth of each tree

Bootstrap: The default value for this is True, which means the model follows bootstrapping principles

max_samples: This parameter assumes bootstrapping is set to True, if not, this parameter doesn't apply. In the case of True, this value sets the largest size of each sample for each tree.

The results obtained by Random Forest are very promising after tuning the hyperparameters, we perform a grid search to optimize the model.

```
{'max_depth': 30, 'n_estimators': 300, 'n_jobs': -1}

Train score: 0.992948279300798
Test score: 0.9486029307513758
MSE: 0.014646275788717182
RMSE: 0.12102179881623468
MAE: 0.07360446928410826
```

Code 7: Scores after Hypertuning the parameters for Random Forest Model

The problem of overfitting is overcome by combining many Decision Trees which will eventually give us low bias and low variance. But due to large number of trees this algorithm will take long time to train and it is ineffective for real time predictions.

8.4 Artificial Neural Network (ANN)

To avoid overfitting of the model we implement to following strategies:

1. Simplifying the model: We use minimal number of hidden layers to simplify the model. To decrease the complexity, we can simply remove layers or reduce the number of neurons to make the network smaller.

2. Early Stopping: This method update the model so as to make it better fit the training data with each iteration. Up to a point, this improves the model's performance on data on the test set. Past that point however, improving the model's fit to the training data leads to increased generalization error.

3. Regularization is a technique to reduce the complexity of the model. It does so by adding a penalty term to the loss function. The most common techniques are known as L1 and L2 regularization: The L1 penalty aims to minimize the absolute value of the weights. This is mathematically shown in the below formula. The L2 penalty aims to minimize the squared magnitude of the weights. This is mathematically shown in the below formula.

4. It randomly drops neurons from the neural network during training in each iteration. When we drop different sets of neurons, it's equivalent to training different neural networks. The different networks will overfit in different ways, so the net effect of dropout will be to reduce overfitting

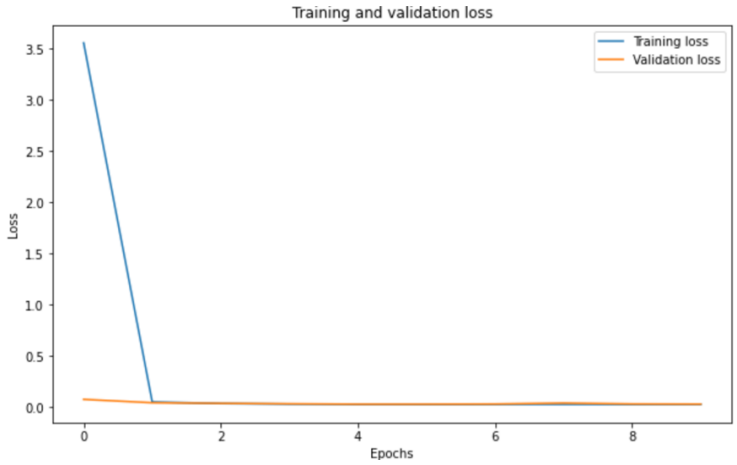

Fig 20: Epochs VS Loss for ANN

Train Loss: 0.0288
Test Loss: 0.0282
R2 Score: 0.8878
MSE: 0.02821876749443196
RMSE: 0.1679844263449203

Code 8: Scores for ANN Model after tuning the hyperparameters

8.5 XGBoost

XGBoost or Extreme Gradient Boosting algorithm is developed to for speed and performance. It uses tree-based model to predict the price of the automobile. This model is similar to LightGBM or Light gradient boosting machine which can handle large scale data. The following are the advantages of using XGBoost

- ***Regularization*** - In tree-based methods regularization is usually understood as defining a minimum gain so which another split happens. Minimum loss reduction required to make a further partition on a leaf node of the tree. The larger gamma is, the more conservative the algorithm will be. This minimum gain can usually be set for anything between (0,∞). This helps in abstaining the model from overfitting
- ***Parallel Processing*** - XGBoost implements parallel processing and is blazingly faster as compared to GBM. For our dataset, the processing time is about 20 seconds.
- ***Handling Missing Values*** - It has an in-built routine to handle missing values.
- ***Built-in Cross-Validation*** - allows user to run a cross-validation at each iteration of the boosting process.

The results of XGBoost are precise and accurate with the following values

```
Train score: 0.9582885797343887
Test score: 0.9416416883705738
MSE: 0.01662997402738789
RMSE: 0.12895725659065443
MAE: 0.09012279120003651
```

We further improved the accuracy by tuning the hyperparameters

```
# instantiating the model
xgb = XGBRegressor(
    n_jobs=-1,
    learning_rate=0.2,
    max_depth=8,
    min_child_weight=3,
    n_estimators=1000,
    subsample=0.7,
    colsample_bytree=0.9
    )
```

```
Train score: 0.9965873765361224
Test score: 0.9563237422413876
MSE: 0.012446128269635917
RMSE: 0.11156221703442397
MAE: 0.07154460240804766
```

Code 9: Setting the Hyperparameters for XGBoost Model and determining the score

We received the Best Scores for the XGBoost Model. The Actual Price vs Predicted Price curve is as shown:

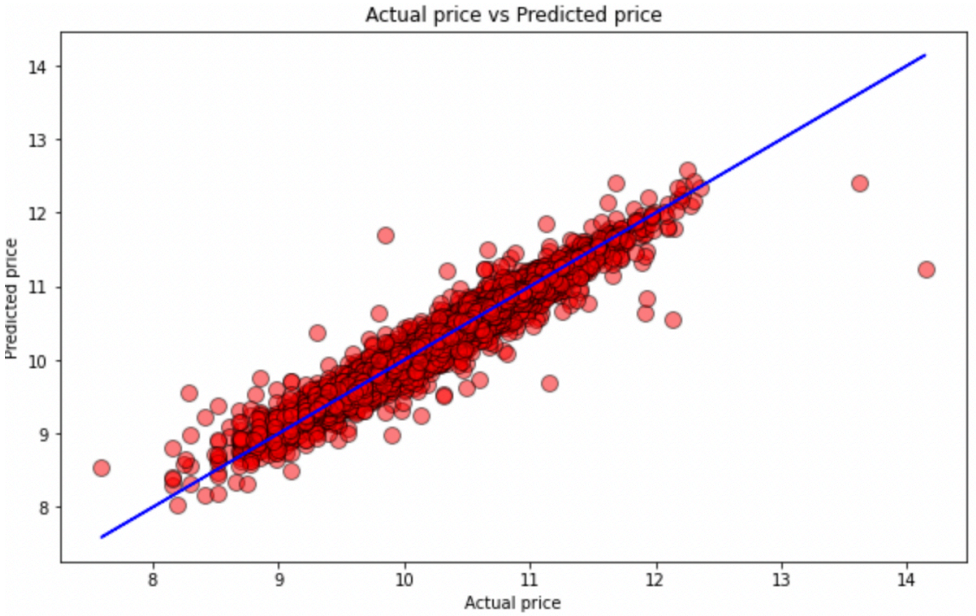

Fig 21: Actual Price vs Predicted Price

9. Results

We have obtained the results from six different machine learning models. The following are the outcomes:

Model	R2 Score	MSE	RMSE	MAE
Linear Regression	0.81	0.04	0.21	0.15
KNN	0.77	0.05	0.244	0.15
Random Forest	0.94	0.01	0.12	0.07
XGBoost	0.95	0.01	0.1	0.06
ANN	0.88	0.02	0.16	0.09

Table 3: Results

- While Linear Regression is simple, it is still a good model to consider with an R2 score of 0.81. The model can be trained easily.
- Conversely the KNN model is trained with only 10% of data as it is computationally expensive and hence the model doesn't get trained enough. Instead, if we use the entire dataset to train the KNN model, we get a very good R2 score of 0.91 which comes at the cost of heavy computation, KNN is not suitable for real time computation.
- The Random Forest algorithm predicts the price with higher precision, but this again comes at a high cost of computation.
- While ANN gives a decent R2 score of 0.88, it is a good model which requires many hyperparameter tuning
- The XGBoost algorithm has dominated the machine learning for structured or tabular data. This model gives an R2 score of 0.95. The model is built easily without much computation.

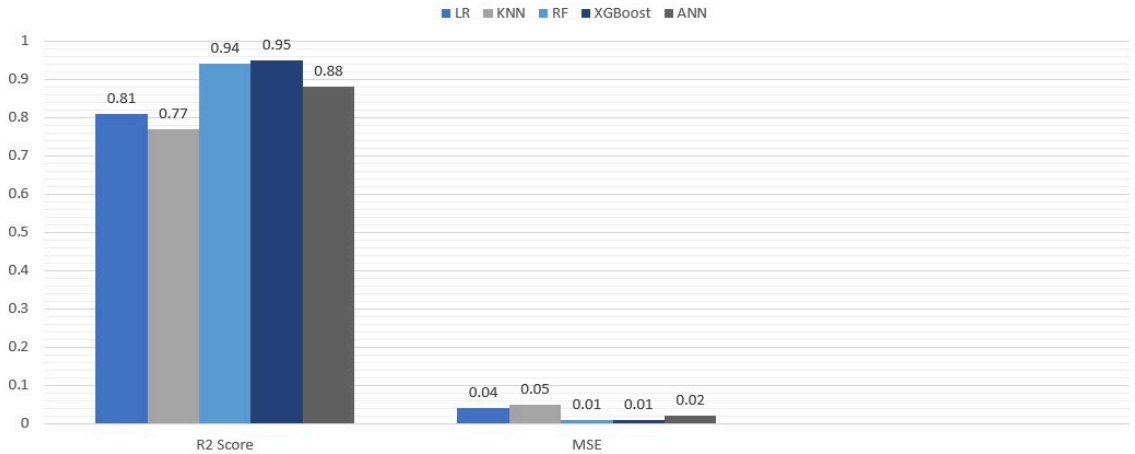

Fig 22: Conclusion- comparison between different models

10.References:

- Kuiper, S. (2008). Introduction to Multiple Regression: How Much Is Your Car Worth? Journal of Statistics Education. doi:10.1080/10691898.2008.11889579

- Listiani, M. (2009). Support Vector Regression Analysis for Price Prediction in a Car Leasing Application. Master Thesis. Hamburg: Hamburg Univesity of Technology .

- Matthew Botvinick, S. R.-N. (May 2019). Reinforcement Learning, Fast and Slow. Trends in cognitive sciences, 23(5), 408-422.

- Monburinon, N., Chertchom, P., Kaewkiriya, T., Rungpheung, S., Buya, S., & Boonpou, P. (2018). Prediction of Prices for Used Car by Using Regression Models. 5th International Conference on Business and Industrial Research (ICBIR), (pp. 115-119). Bangkok.

- Nabarun Pal, P. A. (2018). How much is my car worth? A methodology for predicting used cars prices using Random Forest. Future of Information and Communications Conference (FICC) 2018 , 1-6.

- Noor, K., & Jan, S. (2017). Vehicle Price Prediction System using Machine Learning Techniques. International Journal of Computer Applications, 27-31.

- Pudaruth, S. (2014). Predicting the Price of Used Cars using Machine Learning. International Journal of Information & Computation Technology, 754-764.

- Rizvi, R. (2019, April). Car Production is on the Rise in Dubai. Retrieved September 10, 2019, from https://propakistani.pk/2019/04/08/car-production-is-on-the-rise-in-dubai/