EDHEC Business School

MSc. In Data Analytics and Artificial Intelligence

Master Thesis:

"Prediction of Price of Used Cars in the USA using Regression Techniques"

By

Charles Martin Raj PIDAKALA

Student ID: 79738

May 30, 2022

EDHEC Business School does not express approval or disapproval concerning the opinions given in this paper which are the sole responsibility of the author.

I certify that: (a) the master project being submitted for examination is my research, the data and results presented are genuine and obtained by me during the conduct of the research and that I have properly acknowledged using parenthetical indications and a full bibliography all areas where I have drawn on the work, ideas and results of others and that the master project has not been presented to any other examination committee before and has not been published before.

ACKNOWLEDGEMENTS

I sincerely thank Professor BENITEZ José for his guidance and help in accomplishing this research work. I would also like to thank all the MSc in Data Analytics and Artificial Intelligence Programme Directors and Professors for guiding us through all the subjects and our projects during our course work.

Finally, I thank the EDHEC Business school for the infrastructure and resources provided for all of us students to seamlessly work on projects.

ABSTRACT

Prediction of used cars is a topic that’s been in high demand because of the unprecedented number of cars being purchased not just from one owner to another but from companies and businesses that emerged especially to handle the sales and purchases of used cars. The primary objective of this research is to develop a Machine Learning model that would estimate and predict the prices of the used cars by using the features that are highly affecting our dependent variable 'price'. Data mining and pre-processing, and statistical analysis on this data were done with the help of the Data Science and Machine Learning libraries of Python. In this learning study, multiple regression algorithms were used from the Python Machine Learning libraries such as Sklearn, etc., to come up with a model that can be trained, tested, and compared to each other. Among all the models developed and experimented with, XGB Regressor had the highest score of 75% followed by MAE of $ 4083 and 93 % of MAPE. We expect to improve these scores if we consider even more relevant features such as description, model, mileage, and ML models prepared for company-specific cars. This study could be further developed into an application for mobile for the general public or for a firm that sells and buys used cars.

TABLE OF CONTENTS:

LIST OF FIGURES:

Appendix

LIST OF TABLES

Chapter 1: INTRODUCTION:

1.1 Background:

The automobile industry is one of the major industries and transport has become next to the necessity. The automobile evolution, say a different type of vehicle, their production process, and their availability to the consumers has been so remarkable so far and it's not going to take a halt any time soon. Cars have been the most used automobiles all around the world by almost all sections of people. There are innumerable car manufacturers all around the world and still, new companies are emerging day by day. It is not an exaggeration to say that "no invention affected the humans every life in the 20th Century more than the automobile." One major shareholder of car sales all around the world after China is the United States of America. The USA stands 2nd position by the statistics of 2021 with a total number of 14,947,180 cars sold. The percentage of increase in sales from 2020 to 2021 is 3% (*Car Sales by Country | Global Car Sales Data | 1. China 2. the US*, 2021).

Even though automobiles had been there since the 19th century, the accessibility of cars to the general public started after the advent of Ford. The growth of the Automobile industry led to an economic evolution across the US (ushistory.org, n.d.). "Automakers and their suppliers are America's largest industries in the manufacturing sector, contributing 3% to the country's GDP" (*US Economic Contributions*, 2020). Not just the purchase of new cars, but the purchase of old and used cars are also quite fluid in the US. As per the News article and a report by (Technavio, 2022) the used car market size in the US is expected to increase by 3.91 million units before 2025 with an accelerated momentum at a CAGR of 1.98% during this forecast period.

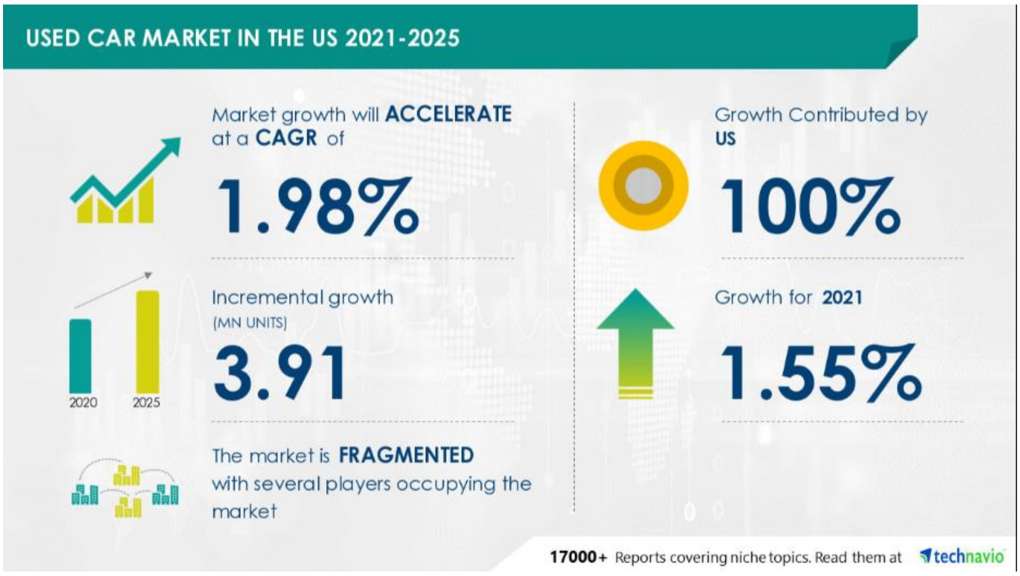


Figure 1: Used Car Market in US 2021-25 (Technavio, 2022)

From the above statistics, we can say that people in the US are inclined toward buying used cars. Because of the scope of getting a business from introducing a vendor to a buyer, the consultancy has emerged who mediates the transaction sale and purchase of the vehicle and thereby generates revenue. This would reduce the hassle for both buyer and vendor to search for each other in the market. In recent years, many businesses are emerging who buy the used car from one user and then repair them and sell it again to users who are looking to buy a used car. These vendors might take advantage of the general public by buying the car for a very low price and then selling it for a high price. Since selling and buying a used car is not easy as it is to buy a new car from a manufacturer, users might lose a lot of money if they are ignorant of the estimated price of a used car.

1.2 Research Vision and mission:

This research study mainly focuses on this issue of the used car market to provide a solution to know their vehicle's price. This study answers the question of **"How to estimate the price of used cars?"**. We try to take the help of features such as Manufacturer, year of manufacture, distance travelled, condition etc and understand the relation of these features and estimate a price for the sale/purchase of the car with the application of statistical data analysis and Machine learning algorithms.

1.3 Project Goals:

This study is sought after because of the high demand for used cars in the US and because of the major contribution to generating revenue in the automobile market. The aim of conducting this study is to help the general public and used car vendors determine the price of a used car based on different features. Towards this end goal, as a research project, we have some subsequent goals to reach. Some such goals are as follows:

1. Research and find relevant data sources which would provide us with the data required to conduct the study
2. If we did not find the data on the web, scrape the data from Craigslist.
3. Finding out other requirements to achieve the end goal. Requirements such as Tools and Technologies applicable, the infrastructure needed.
4. Application of the Data cleaning, pre-processing, EDA and Modelling techniques in Python using data science libraries.
5. Come up with a sound model that can be trained with the data available and tested with good metrics.

1.4 High-level Project Design and flow:

The high-level design of the project is as follows. The entire project is broadly classified into four steps:

i. *Data Collection*: Used car sales and purchase data need to be collected from a relevant source. The data collected should have features that are required to decide a price for a car such as a manufacturer, make, etc.,

ii. *Data Analysis*: The data analysis part has two subparts

a. *Data Exploration*: If the data set is huge take a sample of data and find out the statistical analysis of these features with the dependent variable 'Price'. Visualise the data to find out the correlation and trend of these features with price. For the continuous variables adjust the scale by standardising them and finding the distributions.

b. *Data Cleaning*: Finding the percentage of null values in the features and handling the null values by removing the column if it does not affect the price variable or Interpolate values with the help of the other features. Find out the outliers and remove them for better model training.

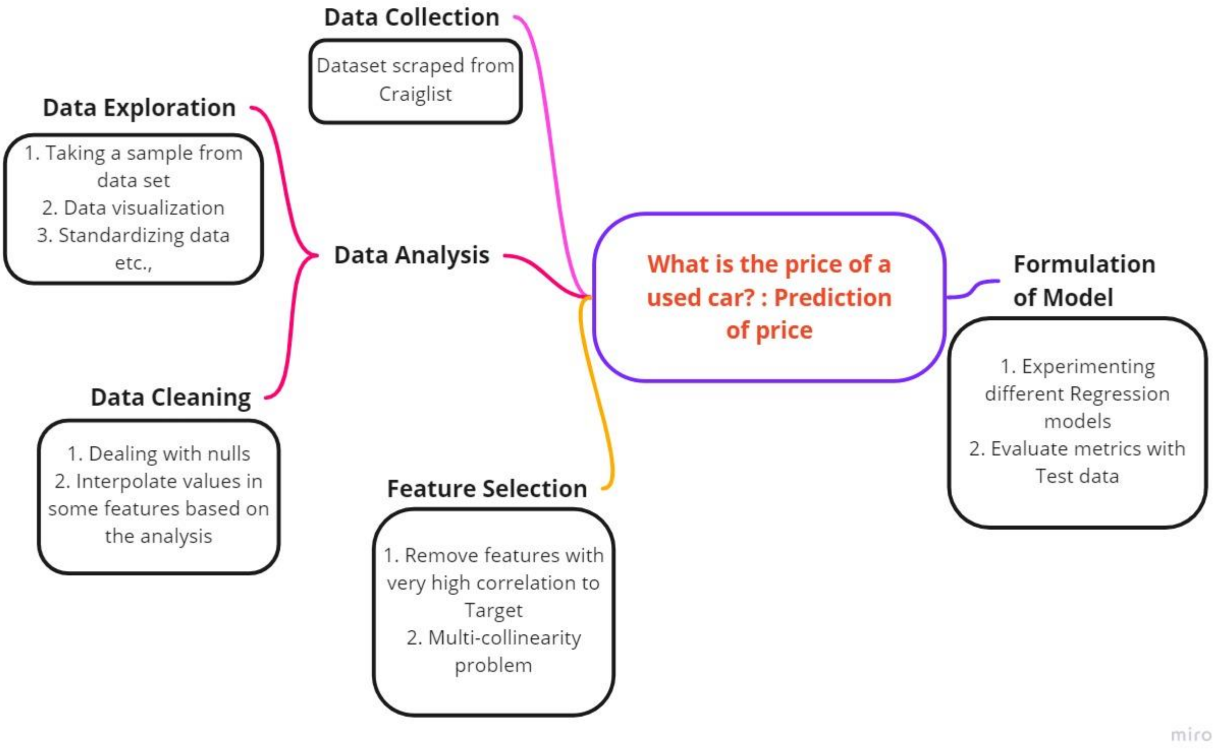

Data Collection
Dataset scraped from Craiglist
Data Exploration
1. Taking a sample from data set
2. Data visualization
3. Standardizing data etc.,
Data Analysis
Data Cleaning
1. Dealing with nulls
2. Interpolate values in some features based on the analysis
What is the price of a used car? : Prediction of price
Formulation of Model
1. Experimenting different Regression models
2. Evaluate metrics with Test data
Feature Selection
1. Remove features with very high correlation to Target
2. Multi-collinearity problem
miro

Figure 2: High-level project design and flow

iii. *Feature Selection*: Finding the feature correlation with the target and removing those columns which are highly correlated with the price variable. Similarly, check for multi-collinearity problems and drop the columns. Vectorize the qualitative variable to make use of them in our model.

iv. *Formulation of Model*: Splitting the data into train set and test set, and experimenting with the data with different regression models. Evaluating all these experimented models and finding the best model that predicts the prices well.

1.5 Resources Required:

We require the following tools, Technologies and libraries

- Python and Data Science libraries of Python such as Pandas, NumPy, SK-learn etc.,
- Tableau, Matplotlib, and Seaborn for Visual data analytics
- Python IDE such as Jupyter Notebook, Spyder or Google Colab
- Data mining, analysing, visualising and ML modelling skills are required.

Chapter 2: LITERATURE REVIEW

From the statistics mentioned in the background, the demand for cars has increased. There is a surge in the prices of cars all around the world. (*U.S.*, 2022) From the surveys and the statistics from Statista, it is observed that the average selling price of Light motor vehicles (LMV) is increased by at least 1.4k – 6.7K from 2018 – to 2021 in the USA. Now the average price of an LMV is standing at $ 42,380 in 2021. This increase is because of the result of the hike in the prices of semiconductors which were in shortage during the COVID-19 pandemic.

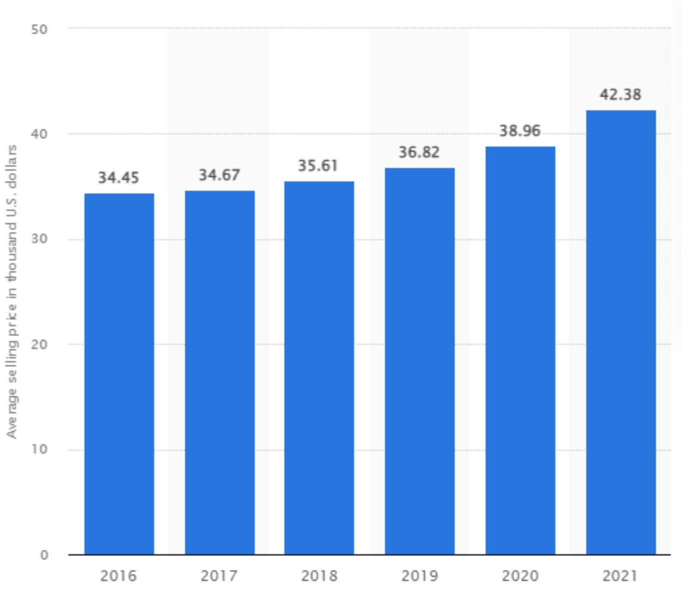


*Figure 3: Average Selling price of a new car by year(*U.S.*, 2022)*

Similarly, the average selling price for used cars amounted to $ 26700 in 2021. This price gap between the new cars and used cars is attracting the households to choose a used vehicle over a new vehicle. As discussed previously, estimation of price for a used car is highly in demand and hence there emerged many consultants and businesses that handle the sale/purchase of the used car in the market. Making these estimates readily available for the general public who does not have the expertise in the sale/purchase of cars domain knowledge has a huge scope for research and development.

Several studies and researches had been made previously to estimate the car prices using different methodologies, approaches and using different features for study. (Pudaruth, 2014) in his research, he proposed some ML models to predict the used car prices in Mauritius. The data was collected from the newspapers for over a month and features like CC, year, mileage, make and the price was collected. In his study, he experimented with models such as Decision Tree, KNN, Multiple Regression and Naïve Bayes algorithms and obtained accuracies of 60-70%. The author stated his limitations in his research were due to insufficient data for the models Decision tree and Naïve Bayes.

(Noor & Jan, 2017) Two other researchers, Noor and Jan proposed a multivariate regression model to predict the price of the used cars. They collected used car data (around 2000 records) from "PakWheels" which contains variables such as price, engine capacity, mileage, colour, and transmission, etc., They ran the Linear regression model in Minitab with all the features and found some of the features as insignificant. The features they used for their final model are year, model, engine type and price and obtained an R2-score of 98% and a standard deviation of the residuals of Rs.92622.

(Monburinon et al., 2018) Researchers conducted a comparative study to predict the prices of the used cars in Germany using the data scraped from the German e-commerce site (about 304133 records). They took some categorical and numerical features such as fuel type, model, seller, price, and year etc., They trained ML models such as Linear regression, Random Forest, and Gradient Boosting using some of the features mentioned and used MAE for evaluating their models.

(Wu et al., 2009) proposed a new price forecasting technique for used cars using BP networks (Back Propagation Neural network), and the ANFIS model (Adaptive Neuro-fuzzy Inference system). Both these Artificial Neural networks are quite widely used ANNs and using these, the researchers were successful in forecasting the price and Absolute Percentage error (APE) was used for evaluating the prediction efficiency. Though this research was done back in 2009, the application of ANN on large data sets to predict the price is much more efficient and reliable

(AlShared, 2021) A student of RIT Dubai, for his MSc thesis, proposed ML models to predict the price of used cars in the United Arab Emirates (UAE). Data was scraped from buyanycar.com and it’s cleaned for the usage of the models. Features such as brand, model, year, colour, engine capacity, etc., were used as independent variable inputs to train the ML regression models such as Random Forest regressor, Bagging regressor and linear regressor. The models were evaluated and the R2-score of 85-88% was obtained with minimal MAEs.

From all the above research, it is quite evident that this research on the prediction of price is very sought after and it has a lot of scope for business in the used car markets. So far, all these researches were done using a ton of different features and many different models. This study especially focuses on the regression and ensemble models from the Machine Learning libraries of python and both categorical features and numerical features would be used for training these models.

Chapter 3: PROJECT DESCRIPTION

3.1 Data Source and Data Collection

As this study only talks about the used car market in the USA, we need the data from all the states in the USA. The data set is available on Kaggle for download at “Used Cars Dataset”. The data set is scraped by (Austin Reese) from Craiglist: the world's largest collection of used vehicles for sale. This specific data is scraped for all the states in the USA.

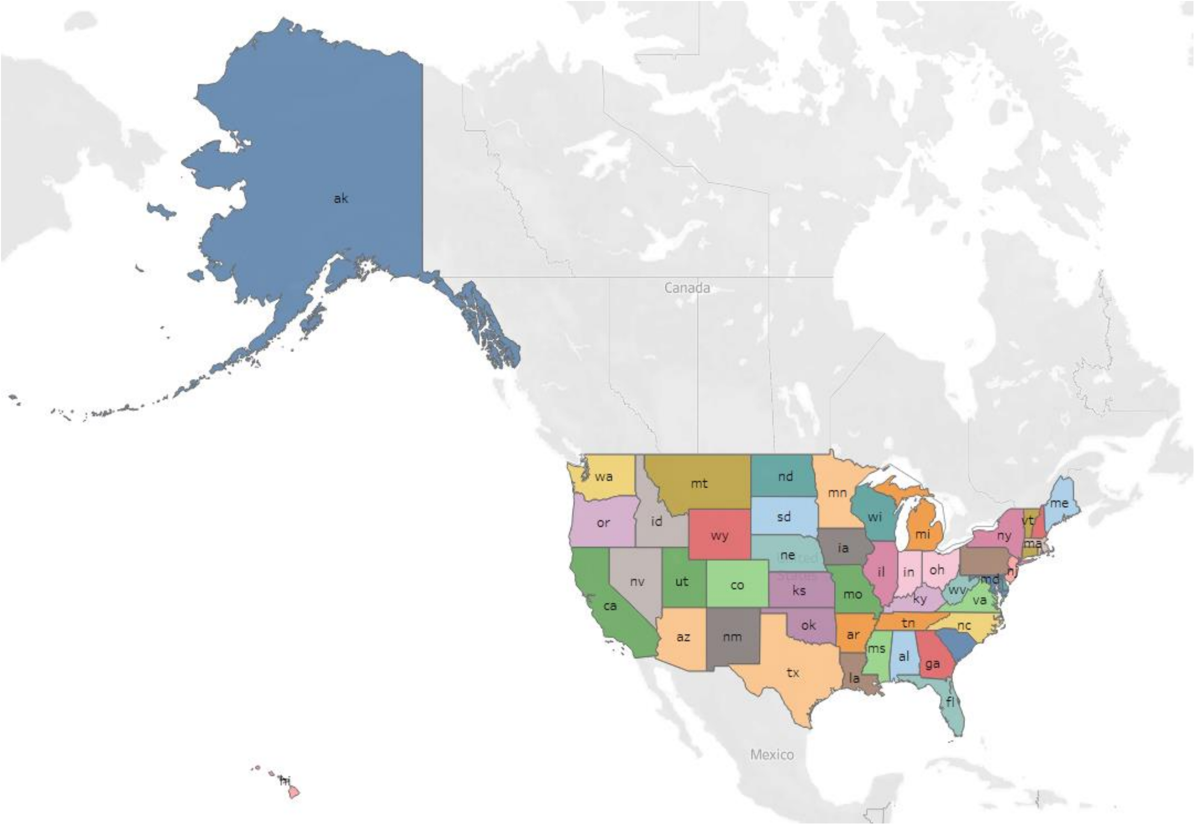


Figure 4: Tableau map of Different states data available in the data set.

The data set is named "vehicles.csv" and it contains more than 460000 records with 26 columns. To optimise the performance and take some of the burdens on the processor, only a sample of 50000 rows of data is taken randomly (fig.25 Appendix). The sampled dataset is now “vehicle_sample.csv” and this is read into python as vehicles_df data frame for analysis and modelling and this data frame is duplicated as df1. All the operations are made on this df1 data frame.

3.2 Data Set Description

There are completely unusable columns among the 26 columns such as: [Unnamed: 0, id, URL, region_url, VIN, posting_date, image_url, lat, long]. These features might be useful for a user to manually look but they do not help train our model. So, all these columns are dropped before further analysis. Additionally, two columns: “image_available”, “desc_available” of Boolean type and one numerical column “Age” are created. All the features that may be needed for the model are as follows:

Features	***Description***
region	The region where the car is available for sale
year	purchase year. (Values are in years)
manufacturer	Car manufacturers such as Ford, and Chevrolet.
model	Model of the car.
condition	The current condition of the car.
cylinders	The number of cylinders in the car (talks about the engine cc).
fuel	fuel type gas, diesel or petrol.
odometer	The number of miles the car was driven.
title_status	The current state of the car (clean or not)
transmission	Transmission type of the car (automatic, manual or hybrid)
drive	Drive type whether it's RWD,4wd, or fwd.
size	Size of the car.
type	Car family type (hatchback, sedan, or some other category)
paint_color	Colour of the car
description	a short description of the car written by the seller
desc_available	Has true or false values based on the availability of description
image_available	Has true or false values based on the availability of image

Age	calculated by subtracting the year from the current year

Table 1: Feature descriptions

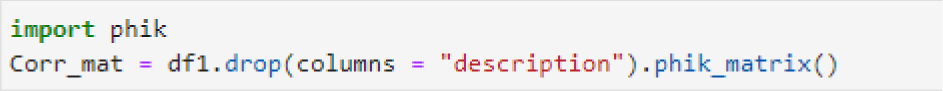
3.3 Tools and Technologies: Libraries & Technologies used

Tools	Jupyter-Lab version 3.2.8, Spyder, Tableau and Microsoft VS Code
Technologies	Python 3
Libraries	Pandas, NumPy, Scikit-Learn (sklearn), Seaborn, Matplotlib, Datetime, Phi-K, SciPy

Table 2: Tools and Technologies used

***New Libraries used*:**

Phi-K: Phi-k is a new correlation coefficient which refines the traditional Pearson's correlation. The three main advantages of Phi-k over other correlation methods: “First, it works consistently between categorical, ordinal and interval variables. Second, it captures non-linear dependency. Third, it reverts to the Pearson correlation coefficient in case of the bivariate normal input distribution.” This is especially used to study variables of mixed data types (*Phi_K Correlation Analyzer Library*, 2018). Fig. 5 shows the library and syntax for Phi-k.



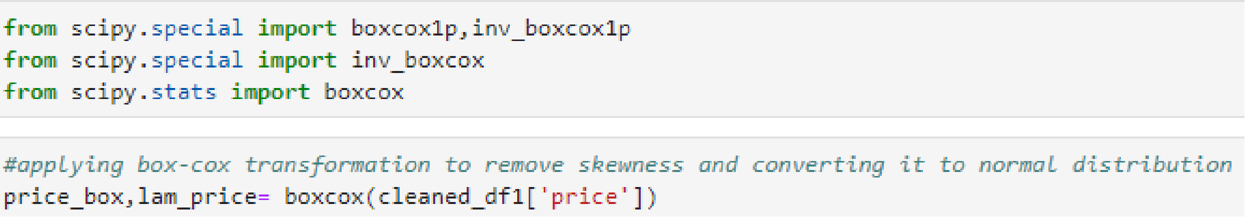
```
import phik
Corr_mat = df1.drop(columns = "description").phik_matrix()
```

Figure 5: Phi-k library and syntax

Box-Cox transformation: This is a transformation technique that is used to transform a non-normal feature into a normal distribution. This transformation technique is proposed and developed by statisticians George Box and Sir David Roxbee Cox. At the core of this technique, is an exponent value lambda (λ), which varies from -5 to 5. All values for λ are considered optimal values for the data selected (Stephanie, 2021). The optimal value gives the best approximation for a normal distribution curve. This transformation follows the following formula:

$$\mathrm{y}(\lambda) = \begin{cases} \frac{y^{\lambda}-1}{\lambda}, & if \ \lambda \neq 0; \\ \log y, & if \ \lambda = 0. \end{cases}$$

This is available under the SciPy library and can be used as shown in fig. The inverse transformation has to be used on the target variable while calculating performance metric scores using inv_boxcox.

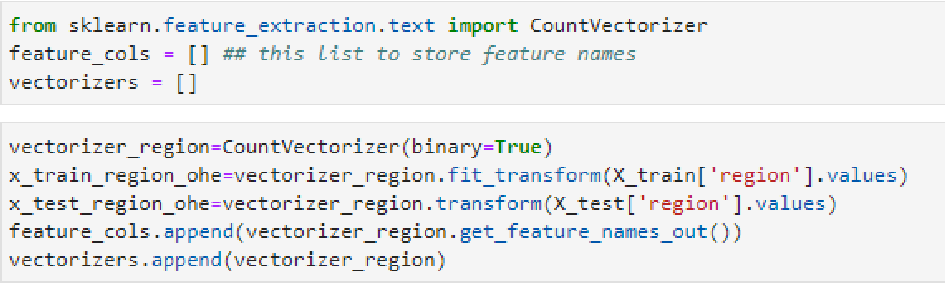


```
from scipy.special import boxcox1p,inv_boxcox1p
from scipy.special import inv_boxcox
from scipy.stats import boxcox
```

```
#applying box-cox transformation to remove skewness and converting it to normal distribution
price_box,lam_price= boxcox(cleaned_df1['price'])
```

Figure 6: Box-cox Transformation

Count-Vectorizer: This function is used to convert textual data into a vector of terms or tokens. And these words are then encoded as integers or floats using a one-hot encoding technique, to be used in ML algorithms. This process is called feature extraction or Vectorisation (*CountVectorizer in Python*, 2015). Since we have many categorical features, these features are vectorised using a count-vectorizer before using them for training the models.



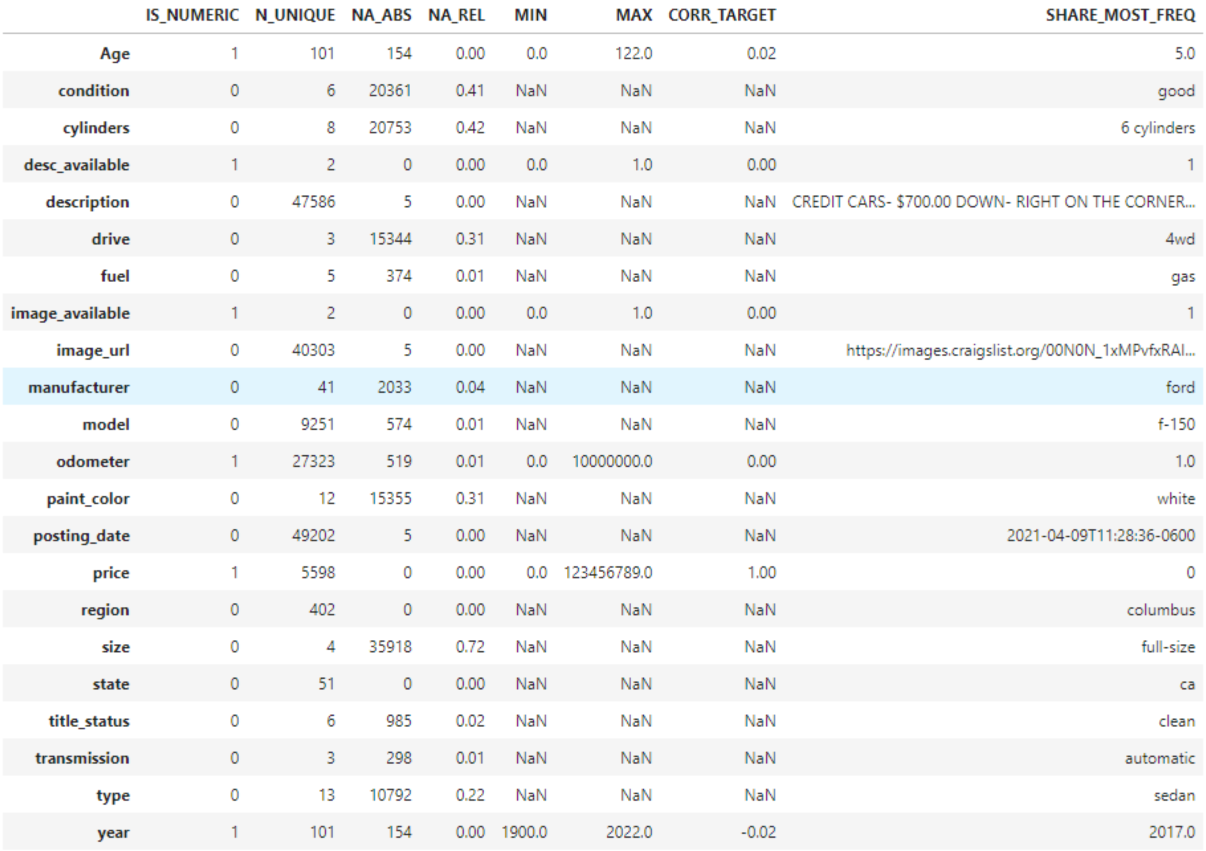
```
from sklearn.feature_extraction.text import CountVectorizer
feature_cols = [] ## this list to store feature names
vectorizers = []
```

```
vectorizer_region=CountVectorizer(binary=True)
x_train_region_ohe=vectorizer_region.fit_transform(X_train['region'].values)
x_test_region_ohe=vectorizer_region.transform(X_test['region'].values)
feature_cols.append(vectorizer_region.get_feature_names_out())
vectorizers.append(vectorizer_region)
```

Figure 7: Count-Vectorizer

3.4 Exploratory Data Analysis (EDA) and Data Cleaning:

The EDA will give in-depth exposure to the different features one by one. Before jumping into understanding each feature. The function “get_summary_table()” would give some insights into the data we have. Refer to fig 2 in the appendix for the code inside the function.



	IS_NUMERIC	N_UNIQUE	NA_ABS	NA_REL	MIN	MAX	CORR_TARGET	SHARE_MOST_FREQ
Age	1	101	154	0.00	0.0	122.0	0.02	5.0
condition	0	6	20361	0.41	NaN	NaN	NaN	good
cylinders	0	8	20753	0.42	NaN	NaN	NaN	6 cylinders
desc_available	1	2	0	0.00	0.0	1.0	0.00	1
description	0	47586	5	0.00	NaN	NaN	NaN	CREDIT CARS- $700.00 DOWN- RIGHT ON THE CORNER...
drive	0	3	15344	0.31	NaN	NaN	NaN	4wd
fuel	0	5	374	0.01	NaN	NaN	NaN	gas
image_available	1	2	0	0.00	0.0	1.0	0.00	1
image_url	0	40303	5	0.00	NaN	NaN	NaN	https://images.craigslist.org/00N0N_1xMPvfxRAl...
manufacturer	0	41	2033	0.04	NaN	NaN	NaN	ford
model	0	9251	574	0.01	NaN	NaN	NaN	f-150
odometer	1	27323	519	0.01	0.0	10000000.0	0.00	1.0
paint_color	0	12	15355	0.31	NaN	NaN	NaN	white
posting_date	0	49202	5	0.00	NaN	NaN	NaN	2021-04-09T11:28:36-0600
price	1	5598	0	0.00	0.0	123456789.0	1.00	0
region	0	402	0	0.00	NaN	NaN	NaN	columbus
size	0	4	35918	0.72	NaN	NaN	NaN	full-size
state	0	51	0	0.00	NaN	NaN	NaN	ca
title_status	0	6	985	0.02	NaN	NaN	NaN	clean
transmission	0	3	298	0.01	NaN	NaN	NaN	automatic
type	0	13	10792	0.22	NaN	NaN	NaN	sedan
year	1	101	154	0.00	1900.0	2022.0	-0.02	2017.0

Table 3: get_summary_table summary

From the table.3 summary, we can see the insights of each feature such as is_numeric or not, No. of unique values, number of nulls and percentage of nulls, min, max, mode and correlation with price. The columns 'posting_date', and 'image_url' are removed for further analysis as they don't contribute to our model. Similarly, 'size' is removed as it has more than 72% nulls. Since most of the above variables are qualitative, the Phi_k correlation analyzer was used to find the correlation between other features.

Using the phik_matrix, a Correlation matrix is generated and a heat map is plotted (fig.26 Appendix) to find the correlation between different variables. From the heatmap in fig 5 in the appendix, it is observed that:

1. "model" is highly correlated with manufacturer, price and many others (> 90%).
2. 2. "state" is highly correlated with "region" (= 1).

This poses a multicollinearity problem. So, the Model and state are removed from the data frame df1. And it is observed that there are 1323 duplicate records in the data and these duplicates are removed as well.

Analysis of each variable

Dependent variable – "price": There are many outliers in the price distribution on both sides with max going more than $ 100K and less than $200. These records with these anomalies and records with prices less than $ 1000 and miles less than 60,000 and age greater than 10 are dropped as they don't make sense and contribute to our model (fig.27 Appendix). After the cleaning of the data, the price distribution is as below which is still right-skewed. Price is normalised using the Box-cox transformation to obtain price as a normal distribution.

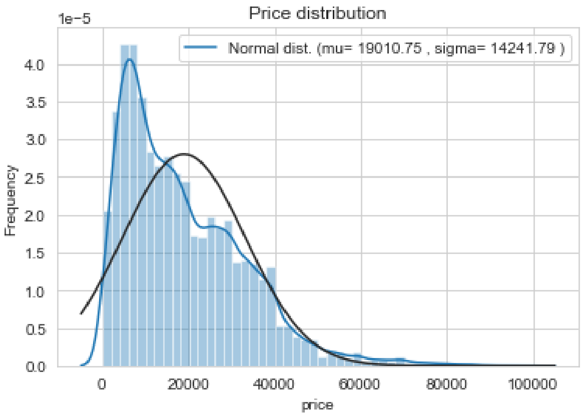


Figure 8: Price Distribution

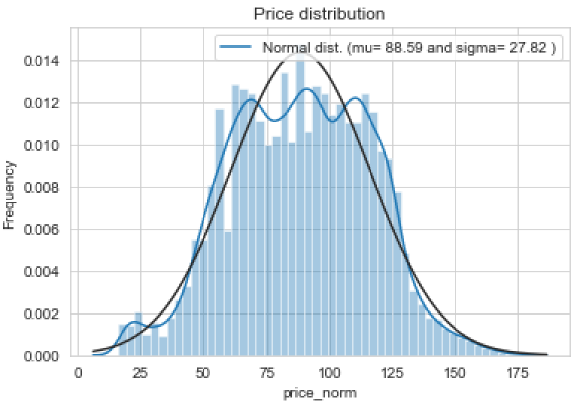


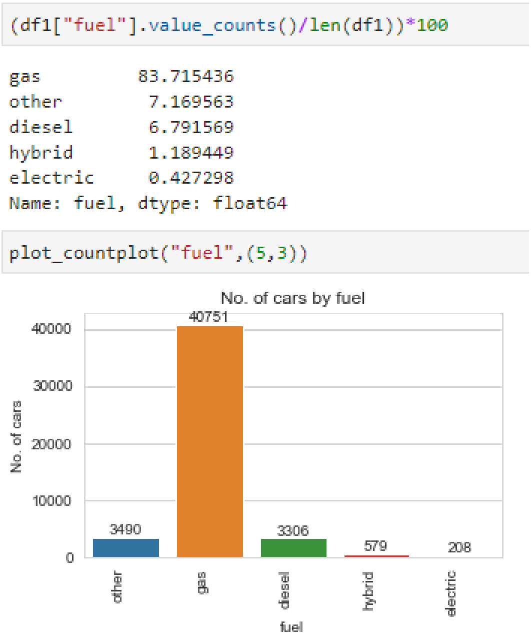
Figure 9: Normalised Price Distribution

Control variables:

i. *"title_status":* 95% of the cars are clean and the distribution can be observed from the fig.28 in appendix

ii. *"condition":* 29% of the cars are in good condition and 23% are in excellent condition.

iii. *"fuel":* 83% of the cars are gas-driven cars showing people in the US would prefer to use gas-driven cars. From the following figures 6 & 7: it is observed that 75% of the gas fuel type cars are under $ 23K while 75% of the diesel fuel type cars are under 45K USD. On average Diesel cars are costlier in the market whereas hybrid and gas cars have an average low price.

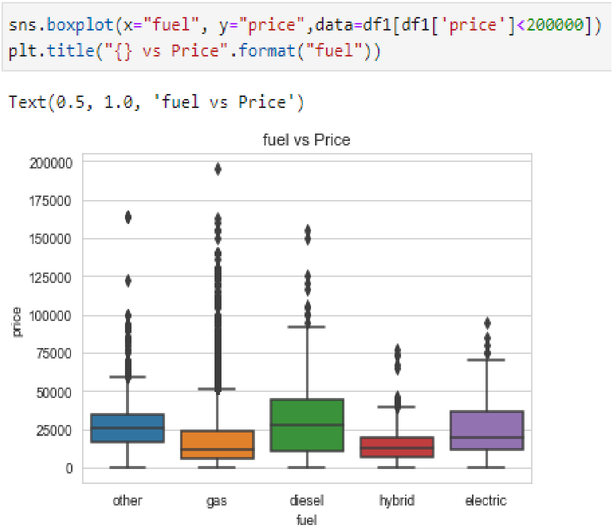


```
(df1["fuel"].value_counts()/len(df1))*100
```

```
gas          83.715436
other         7.169563
diesel        6.791569
hybrid        1.189449
electric      0.427298
Name: fuel, dtype: float64
```

```
plot_countplot("fuel",(5,3))
```

Figure 5: fuel distribution

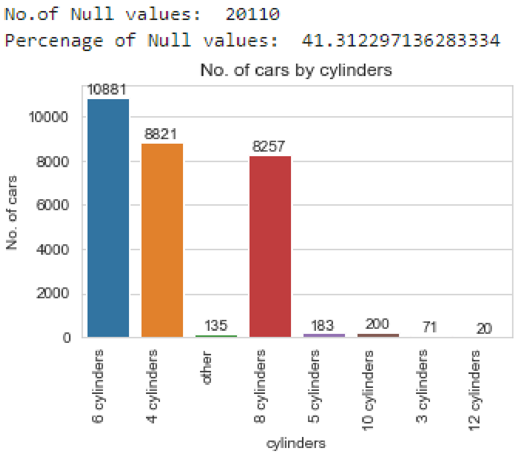


```
sns.boxplot(x="fuel", y="price",data=df1[df1['price']<200000])
plt.title("{} vs Price".format("fuel"))
```

```
Text(0.5, 1.0, 'fuel vs Price')
```

Figure 6: fuel vs price distribution

iv. *"cylinders":* 41% of the data in the cylinders feature are null values. Out of the remaining records, there are at least 8 types of cylinders. 38% of the cars have 6 cylinder engines. 4- and 8-cylinder engine cars are 30% and 28% of the existing values. 12-cylinder cars are expensive when compared to the other ones on an average. it is as expected because, with the increase of the number of cylinders, the power of the engine increases so does the cost. From these observations, we can infer that majority of the US people use cars with 4 and 6 cylinders. (fig. 7, 8, 29 - Appendix)



```
No.of Null values:  20110
Percenage of Null values:  41.312297136283334
```

Figure 7: cylinder distribution

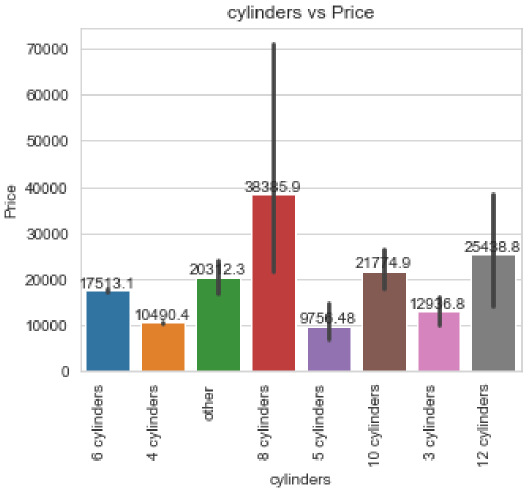


Figure 8: cylinder vs price distribution

v. *“year”:* the year column has values from 1900 but for this study, records between 1990 and 2021 are taken into consideration. Since the price of a car is largely affected by age normally, we only consider cars of age less than 32 or 30. From fig.12 an increasing trend is observed from 1991 to 2018 and it’s been decreasing from 2019 to 2021. This could be because of the effect of covid 19.

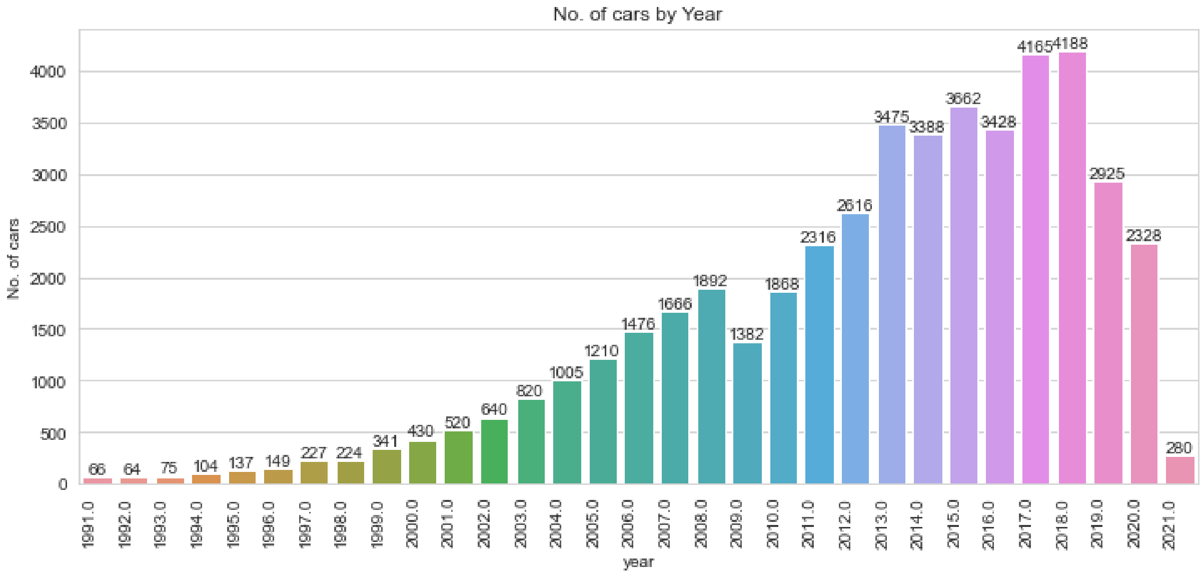


Figure 9: year by year distribution

vi. *“manufacturer”:* It is observed that Ford and Chevrolet (big American brands) are the most sold cars for seconds. Then we have (Japanese brands) Toyota, Honda, Nissan etc. Refer the fig.30, and 31 from the appendix for the distribution graph. Ferrari and Aston-martin are expensive cars. While Chevrolet is like the standard price range as we can observe high prices as well as sales.

vii. “type”: Major types of cars that are sold in this market are sedans with 20% records and then SUVs with 18% (fig.32 Appendix). The average price for pick-up trucks is much higher than other types. Refer fig. for the distribution graph.

viii. “transmission”: Almost 80% of the cars are automatic transmissions. The average price for other transmission types is higher than automatic and manual. It says that people of USA prefer automatic cars

ix. "drive": 30% of the cars are 4-wheel drive cars and 25% are front-wheel drive. The average price for 4WD cars is much higher than for FWD and RWD. We can infer that people from the US prefer 4- wheel drive and front-wheel drive cars more.

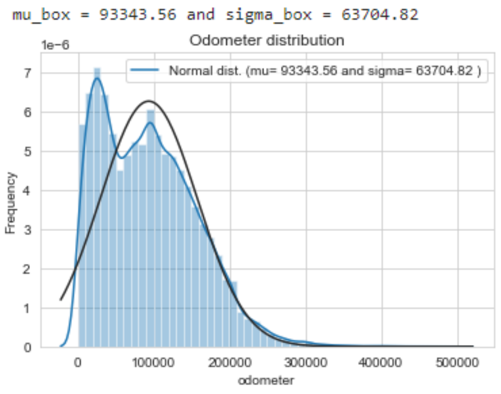
Based on all the observations, a function called clean_data() (fig.33 Appendix) was created to remove all the outliers. This function limits the data from the year 1990 to 2021, the minimum price considered is $ 200 and the maximum is $ 100K and with an odometer more than 500K miles.

3.5 Data Pre-processing and Feature Selection:

Data Pre-processing: Handling Missing values

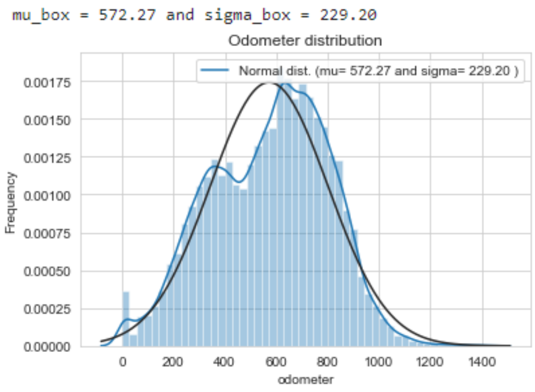
Numerical variables:

i. “odometer”: There are 251 missing values under the feature odometer. The function imputation_numeric (fig.34 Appendix) takes the input of the numeric feature of the data frame and a regressor. This function is using IterativeImputer to impute the missing values by finding the best fit values with the help of a regressor. Then the odometer values are standardised using Box-cox transformation.



```
mu_box = 93343.56 and sigma_box = 63704.82
```

Figure 10:odometer distplot



```
mu_box = 572.27 and sigma_box = 229.20
```

Figure 11:standardised odometer distplot

ii. “Age”: Similarly, age is standardised using box-cox transformation to bring all numerical features into the same scale. But transformation using box-cox also did not transform into a normal distribution.

Categorical Variables:

There are many null values under the categorical vale. Among them, condition and cylinders have 37% and 40% of null values.

i. “condition”: Usually the condition of a car is given by how old a car is, and how far was it driven. These values can be obtained from the odometer and age features. All cars with the year 2020 are filled as 'new' and for the year 2018-2019 as like new and years less than 2018 values are filled concerning odometer values. The lowest median value is for “like_new” and the highest is for “fair”. The odometer median was calculated for different conditions and If odometer values are less than the median of "like_new”, it is filled as “like_new” and greater than the "fair" median, it is filled as "fair". "good" can be filled where the odometer is greater than “like_new” and less than “excellent”, “excellent” can be filled where the odometer is greater than “good” and less than “salvage”. “salvage” can be filled where it is greater than “excellent” and less than “fair”.

ii. “manufacturer”: Only 3% of the values are missing under manufacturer. These missing values are filled with the mode (= "ford") of the manufacturer.

iii. “cylinders”: Some information regarding the cylinders is found in some of the descriptions. Records that have the number of cylinders under the description are filled using those values. From the correlation matrix, it is observed that the cylinders-column is more correlated with the manufacturer. cars with fuel type as “electric” have 0 cylinders

iv. "drive": It is highly correlated with the manufacturer. So, the records were grouped by manufacturer and missing values are filled with respective mode values of each manufacturer or "4wd"

v. "type": similar to the drive, records were grouped by the drive and missing values are filled with respective mode values of each drive type or “sedan”

vi. “transmission”: If the fuel type is electric, missing values under transmission are filled “automatic”. Some of the records have the transmission information in the descriptions. These are filled with their respective values when there is a missing value. All others are filled with automatic as we observed that most of the cars are automatic.

vii. “fuel”, “title_status” and “paint_color”: These are filled with their respective mode values. Gas for fuel, clean for title_status and white for paint_color.

Feature Selection:

A new feature “avg_odo_year” is created from the odometer and Age. It is calculated to know the average distance travelled per year. The features like “description” and “year” are not used in our further process. The target variable and all the features of the entire data set are divided into the train and test split data with the help of SK-learn library “train_test_split” with a test size of 0.2. The numerical features are scaled using Standard Scalar for further processing. All the categorical variables are converted into usable features by a One hot encoding method: Count-Vectorizer, from the SK-learn library. After all this pre-processing and feature selection, there are 613 features that go into the model with 33873 records in train data and 8469 records in test data. All these are features are sparse into hstack to save memory and improve performance while modelling.

3.6 Model Evaluation Metrics

For evaluating the performance of the models, there are so many scoring metrics from the SK-learn library. The metrics that are used in this project are as follows:

1. *Mean Absolute Error (MAE):* MAE is the average absolute value of residual at each data. It is given by the following formula:

$$\text{MAE} = \frac{1}{n} \sum |y - \hat{y}\,|$$

where n = number of data points, y – actual value and $\hat{y}$ – predicted value

A small MAE means, the model is good at prediction while a large error may indicate the model is not performing well for some data points.

2. *Mean Absolute Percentage Error (MAPE):* This metric is useful in evaluating the prediction accuracy. It is a measure of the size of the error terms in terms of percentage. This formula is given by

$$\text{MAPE} = \frac{100}{n} \sum \left| \frac{y - \hat{y}}{y} \right|$$

where n = number of data points, y – actual value and $\hat{y}$ – predicted value

The lesser the percentage the better the prediction. Both MAE and MAPE are good measures for regression models and they are robust to outliers

3. *Root Mean Squared Error (RMSE):* It is the square root of the Mean Squared Errors (MSE). MSE is a risk function that helps to determine the average squared difference between the predicted and the actual value of a feature. It helps plot the differences between the estimated and the actual values of the model. The lower the RMSE the better the accuracy. It is given by the following formula.

$$\text{RMSE} = \sqrt{\frac{\sum (y - \hat{y})^2}{n}}$$

where n = number of data points, y – actual value and $\hat{y}$ – predicted value

4. *R2- Score:* It is a statistical measure that represents the proportion variance for a dependent variable that's explained by independent variables in regression models. The formula is given by

$$R^2 = 1 - \frac{Unexplained\ Variation}{Total\ Variation} = 1 - \frac{RSS}{TSS}$$

where R = coefficient of determination, RSS = Sum of squares of residuals; TSS = Total Sum of squares

Chapter 4: Machine Learning Modelling

The pre-processed data now have divided into train set and test set with 613 features in total. Before experimenting with different models, to evaluate the models by observing different metrics, three functions are developed:

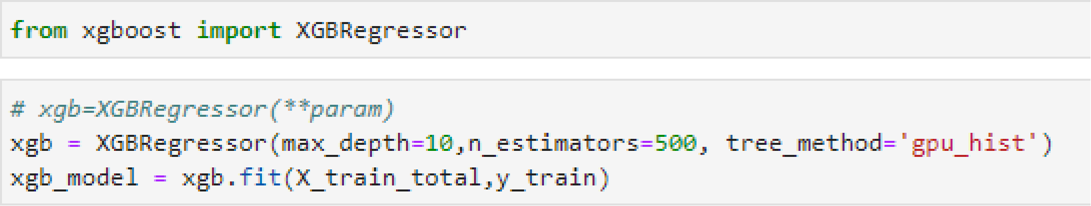
"performance_metric()" – this function takes X, y and model as its input and gives out the metrics such as MAE, MAPE, RMSE, R2 square, etc.

"plot_impfeatures()" – this function takes the model and the features list as its input and then shows the importance that features that are contributing more to the model.

"compare_plot()" – this function takes X, y and model as its input and gives out a graph which shows the actual and the prediction of y values.

4.1 XGB Regressor:

Extreme Gradient Boosting or XG Boosting is one of the ensemble ML algorithms. This can be used for both classification and regression predictive modelling problems. XG Boost is an efficient open-source implementation of the gradient boosting algorithm and hence it is available as its library. As with any other ensemble model, XG boost is also constructed from multiple decision trees. It is both computationally efficient and highly effective when it comes to predicting physical and monetary measures (Brownlee, 2021). The syntax of usage can be observed in fig. After hyper tuning the parameters of the XG boost using optuna and grid search cv, the best estimation is obtained with max_depth=10, no. of estimators = 500 and the tree method as 'gpu_hist'.



```
from xgboost import XGBRegressor
```

```
# xgb=XGBRegressor(**param)
xgb = XGBRegressor(max_depth=10,n_estimators=500, tree_method='gpu_hist')
xgb_model = xgb.fit(X_train_total,y_train)
```

Figure 12: XG Boost Regressor

After training the lasso model with our train data set, it is predicted for the test data and metrics are evaluated. This model has an R2-score of 75, MAPE of 93% and an MAE of 4453, which means it predicts in the error range of ± $2226. Fig. 16 is the plot of y actual and the predicted value of y for some of the data point samples. The prediction is very close to the actual value of the price. Fig. 17 shows the feature importance of the features in the contribution of the model. "Age", "fwd", “4-cylinders” and “diesel” are the top 4 contributing features.

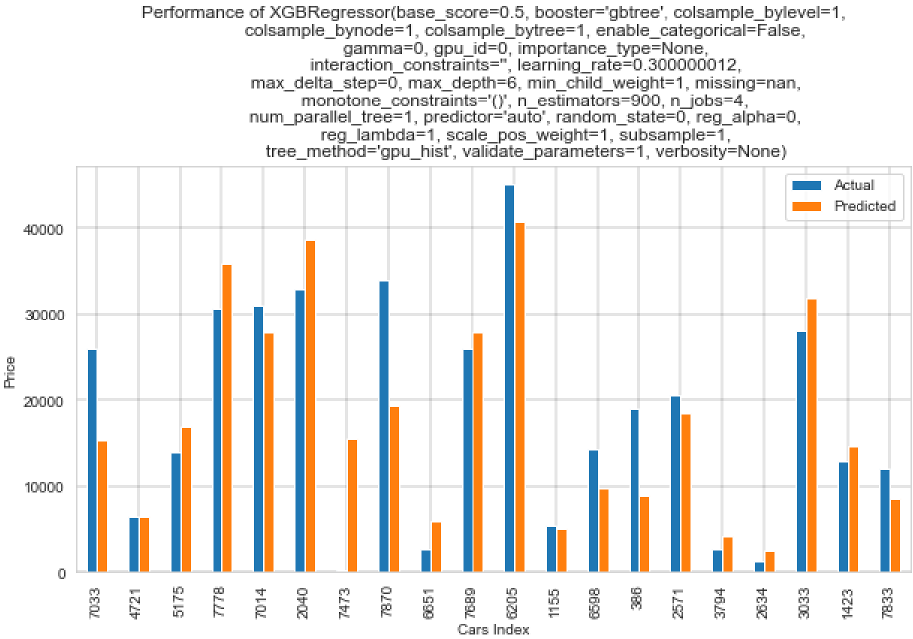


Figure 13:XGBRegressor: y-actual vs y-predicted

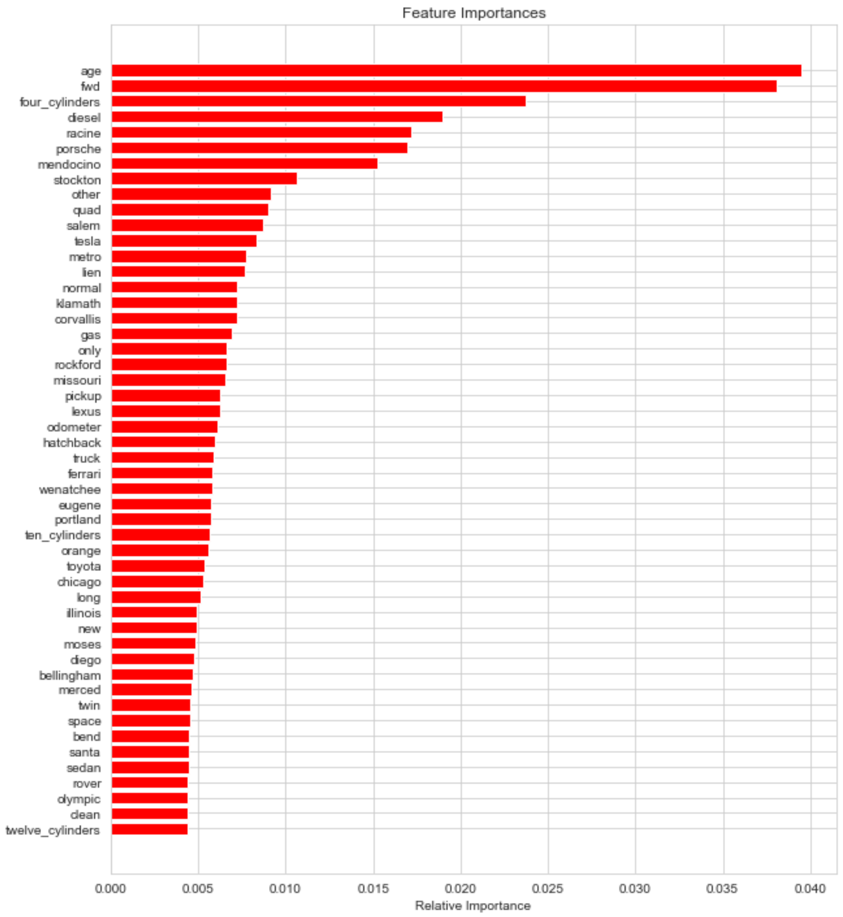


Figure 14:XGBRegressor-Feature importance

4.2 Random Forest Regressor

Random forest is a supervised learning algorithm that uses an ensemble learning method for classification and regression. Random forest is a bagging technique. It also operates from a collection of decision trees during its training and gives the mean prediction of all the individual trees as its output. The trees in random forests run in parallel, meaning is no interaction between these trees while building the trees. Simply put, it is a meta-estimator that aggregates many decision trees with some helpful modifications (Chakure, 2020).

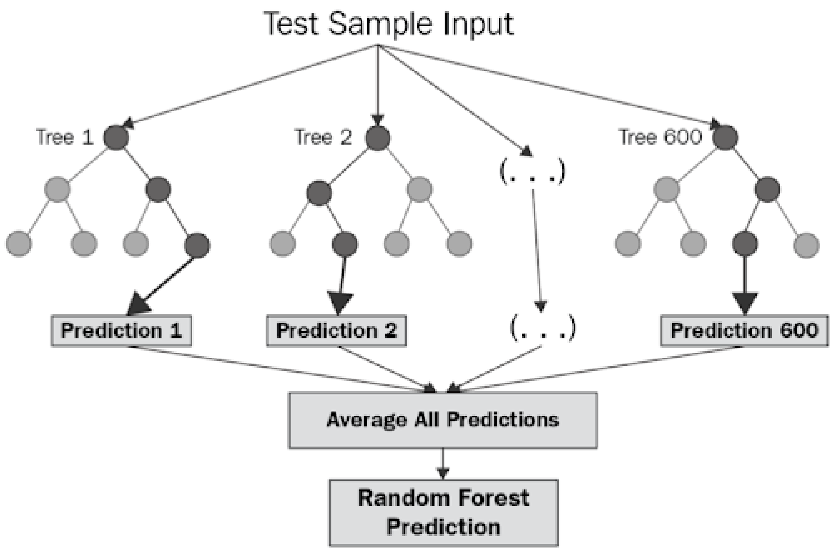
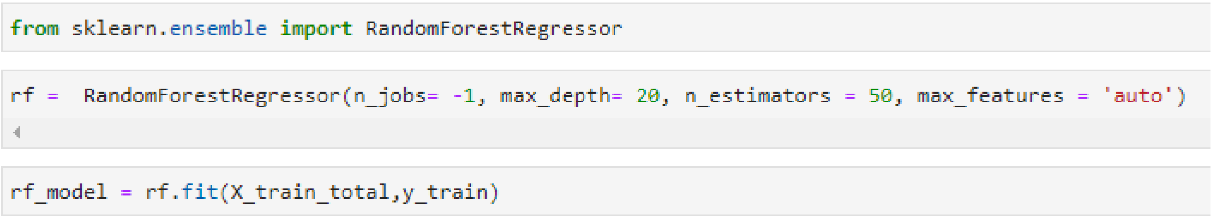


Figure 15: Random Forest diagram(Chakure, 2020)

Random forest regressor is part of the ensemble library and the following parameter were used for the experimentation.



```
from sklearn.ensemble import RandomForestRegressor
```

```
rf =  RandomForestRegressor(n_jobs= -1, max_depth= 20, n_estimators = 50, max_features = 'auto')
```

```
rf_model = rf.fit(X_train_total,y_train)
```

Figure 19: Random Forest Regressor Python code

After training the lasso model with our train data set, it is predicted for the test data and metrics are evaluated. This model has an R2-score of 74.5, MAPE of 101 and an MAE of 4252, which means it predicts in the error range of ± $2126. Fig. 19 is the plot of y actual and the predicted value of y for some of the data point samples. The prediction is very close to the actual value of the price. Fig. 20 shows the feature importance of the features in the contribution of the model. "Age", "odometer", "4-cylinders" and "fwd" are the top 4 contributing features.

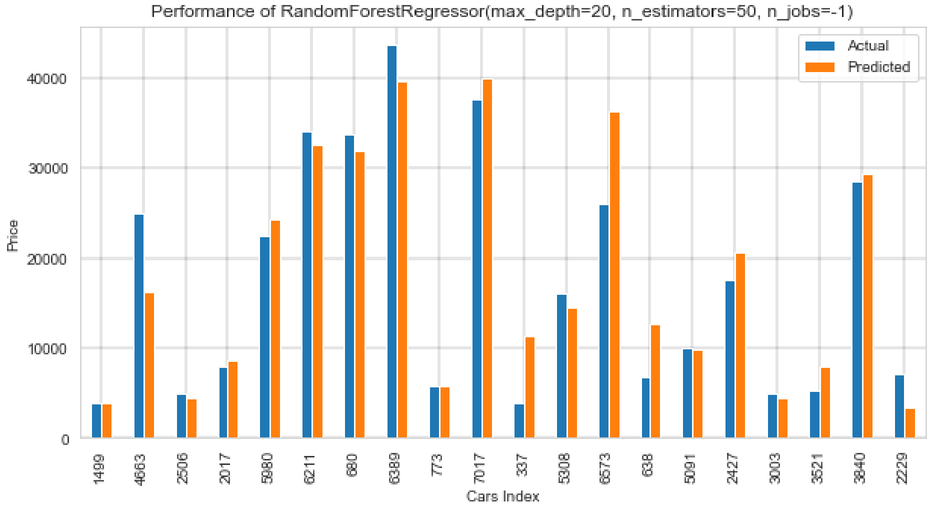


Figure 20: RF comparison plot

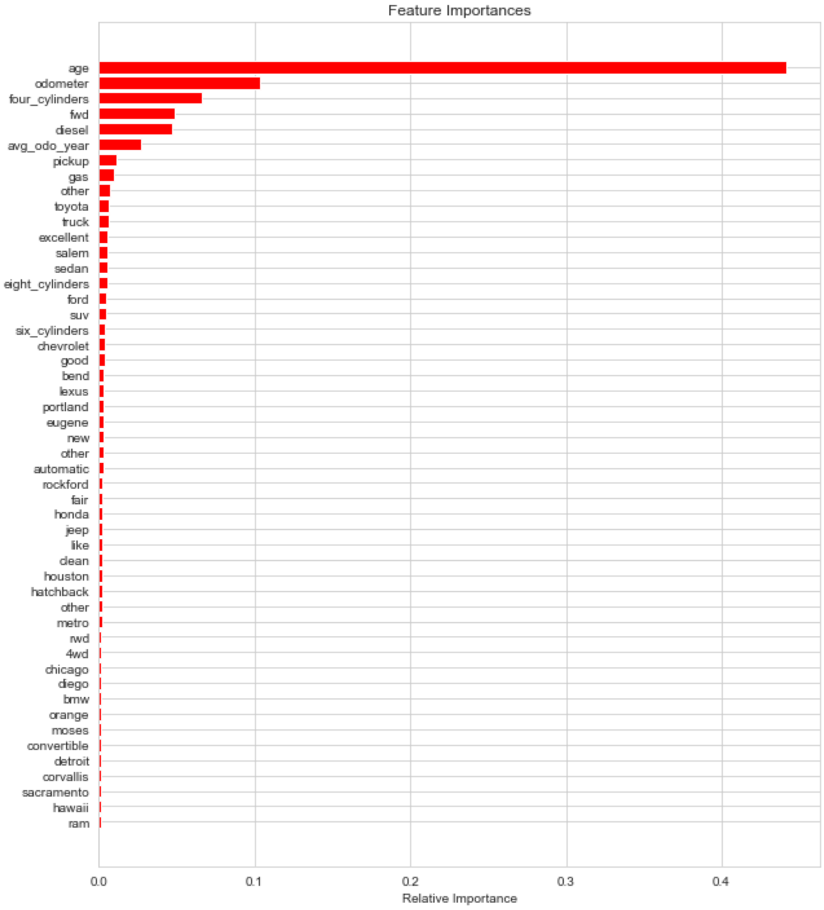


Figure 21: RF Feature Importance plot

4.2 Other Regression Models experimented

Linear Regression: LR is the most common regression algorithm in the machine learning algorithm. It assumes a linear relationship between the different features (X) and the target variable (y). y can be estimated using the linear combination of the input variables (X)

(Brownlee, 2016). For this study, three types of regression models were experimented and their insights are given below.

i. *Simple linear regression model from sklearn.linear*: This model has an R2_score of 67.7, MAPE of 109% and MAE of $ 5180. This means that this model predicts the error range of ±$2590. The prediction vs actual graph can be observed in the below figure.

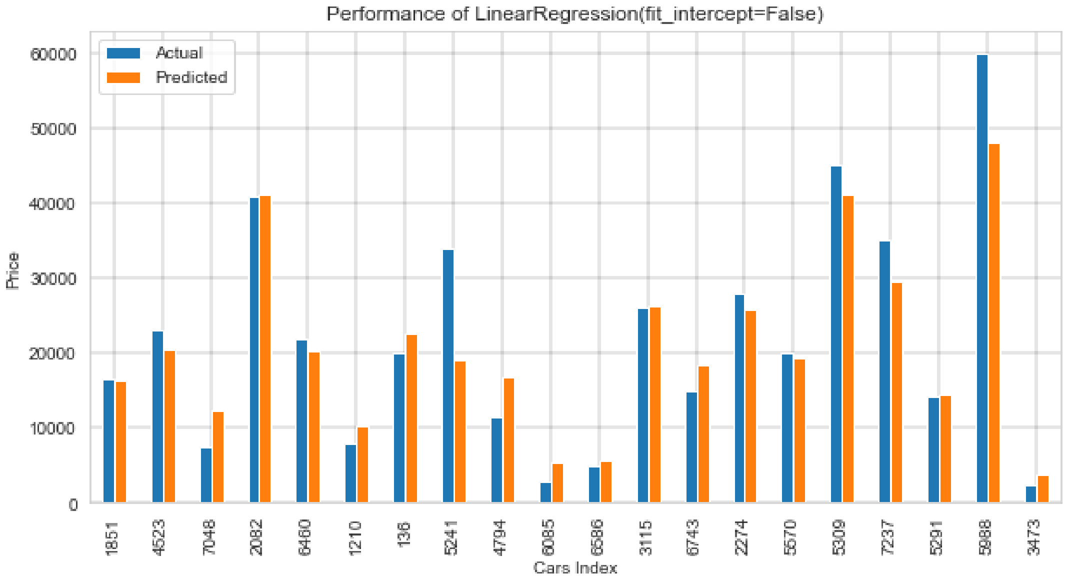


Figure 16: y actual vs y pred linear regression

ii. *Lasso regression:* Lasso regression is a type of linear regression that uses shrinkage. It performs L1 regularisation, which adds a penalty equal to the absolute value of the coefficients(Stephanie, 2015). Its formula is given by

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Figure 23: Lasso Regression equation(Stephanie, 2015)

Where λ = tuning parameter, it controls the strength of the L1 penalty i.e., the amount of shrinkage

After training the lasso model with our train data set, it is predicted for the test data and metrics are evaluated. This model has an R2-score of 68, MAPE of 109 and an MAE of 5159, which means it predicts in the error range of ± $2579

iii. *Elastic Net regression:* Elastic net is a combination of Ridge regression and lasso regression. It mainly focuses on the minimisation of the loss function (*Regularization Tutorial*, 2022).

$$L_{enet}(\hat{\beta}) = \frac{\sum_{i=1}^{n}(y_i - x_i'\hat{\beta})^2}{2n} + \lambda(\frac{1-\alpha}{2}\sum_{j=1}^{m}\hat{\beta}_j^2 + \alpha\sum_{j=1}^{m}|\hat{\beta}_j|),$$

*Figure 17: Elastic net regression expression(*Regularization Tutorial*, 2022)*

Where α is the mixing parameter between ridge (α = 0) and lasso (α = 1) and λ = tuning parameter
After training the lasso model with our train data set, it is predicted for the test data and metrics are evaluated. This model has an R2-score of 68, MAPE of 111 and an MAE of 5150, which means it predicts in the error range of ± $2575

Support Vector Regressor (SVR): SVM is a supervised machine learning algorithm that can be used for both classification and regression. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well. ("SVM | Support Vector Machine Algorithm in Machine Learning," 2017). SVR works on the same principle as SVM. It tries to find the function that approximates mapping from an input domain to real numbers based on a training sample. After training the lasso model with our train data set, it is predicted for the test data and metrics are evaluated. This model has an R2-score of 67.6, MAPE of 125, and an MAE of 5056, which means it predicts in the error range of ± $2528

4.3 Result Comparison

From Table 1, it is observed that XGB Regressor and Random Forest model are performing comparatively better among all the other models with a MAPE and R2-score of 93% & 75 and 101% & 74.5 respectively. This means both the model can explain 75% of data approximately but XGB Regressor performs slightly better than Random Forest with less MAPE.

Model	MAE ($)	MAPE (%)	RMSE	R2_score
XGBRegressor	4453	93	7094	75
Random Forest	4257	101	7162	74.5
Linear Regression	5180	109	8056	67.7
Lasso	5159	109	8039	68
Elastic Net	5150	111	8036	68
SVM	5056	125	8078	67.6

Table 4: ML model Result Comparison

4.4 Limitations and Ethical Considerations

Predicting the value of the used car is not a trivial task as it depends on a variety of factors, not just the factors, we used for our model but many more such as model and make, mileage, maintenance, spare availability in the market, history of breakdowns or engine troubles, and the UI and infotainment features of the car. Even the description and image of the car while selling gives a lot of information as people would like to know the user review and how the car looks. The major limitation of our model not having enough features that affect the price of a car. The other limitation of this particular study is the lack of a powerful machine or CPU to process all the data we have. Since the performance of the CPU is largely affected by the huge input of data, only 50000 records were taken into consideration at random even though we have around 460000 records. The performance of the model would significantly improve with more data.

Ethical Considerations:

As this data is primarily scraped from craigslist, we need to check if there are any copyright issues and if yes, we need to buy this data from craigslist before the official development of the model for the business application.

Chapter 5: CONCLUSION

5.1 Conclusion

Using the dataset obtained from Kaggle (*Used Cars Dataset*, n.d.) which is scraped from Craigslist by Austin Reese, this research project studied different features that affect the price of used cars in the USA. After careful cleaning and processing of data, several different regression ML algorithms from Python Data science libraries were used to come up with different models and compared them with each other. With the observed results, the best model studied in this research is the XGB Regressor with an R2-score of 75% and MAPE of 93%. Though many papers were published, there were not so many companies making use of these researches and converting them into a real-time solution. (*Used Car Price Trends - CarGurus*, n.d.) is one such rare website that gives real-time statistical trends and forecasting of prices for used cars. There is large scope for mobile applications or Software services that can use this research and can be used in real-time forecasting of cars during a Sale or Purchase.

5.2 Outlook and Recommendations

These models discussed in this study, even though they can explain 75% of the variance of the data (R2-score), there is still a large scope for improvement especially in minimizing the MAE and MAPE. As mentioned in the limitations, these models' performance could improve significantly with more input data for training. Similarly, if other important features like Mileage, number of previous owners, previous owner's information, infotainment features available, engine breakdown history could also significantly improve. Further Natural Language Processing and Computer Vision could also be used in reading the description and image features to improve the prediction power of the models. Artificial Neural Networks such as multi-layer perceptron and fuzzy logic techniques could also be used and can be compared with the models observed in this study.

BIBLIOGRAPHY:

AlShared, A. (2021). Used Cars Price Prediction and Valuation using Data Mining Techniques. *Theses*. https://scholarworks.rit.edu/theses/11086

Brownlee, J. (2016, March 24). Linear Regression for Machine Learning. *Machine Learning Mastery*. https://machinelearningmastery.com/linear-regression-for-machine-learning/

Brownlee, J. (2021, March 11). XGBoost for Regression. *Machine Learning Mastery*. https://machinelearningmastery.com/xgboost-for-regression/

Car Sales by Country | Global Car Sales Data | 1. China 2. The US. (2021). F&I Tools. https://www.factorywarrantylist.com/car-sales-by-country.html

Chakure, A. (2020, November 6). Random Forest and Its Implementation. *The Startup*. https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f

CountVectorizer in Python. (2015). Educative: Interactive Courses for Software Developers. https://www.educative.io/edpresso/countvectorizer-in-python

Monburinon, N., Chertchom, P., Kaewkiriya, T., Rungpheung, S., Buya, S., & Boonpou, P. (2018). Prediction of prices for a used car by using regression models. *2018 5th International Conference on Business and Industrial Research (ICBIR)*, 115–119. https://doi.org/10.1109/ICBIR.2018.8391177

Noor, K., & Jan, S. (2017). Vehicle Price Prediction System using Machine Learning Techniques. *International Journal of Computer Applications*, *167*(9), 27–31. https://doi.org/10.5120/ijca2017914373

Phi_K Correlation Analyzer Library. (2018). https://phik.readthedocs.io/en/latest/

Pudaruth, S. (2014). Predicting the Price of Used Cars using Machine Learning Techniques. *International Journal of Information & Computation Technology*, *4*, 753–764.

Regularization Tutorial: Ridge, Lasso & Elastic Net Regression. (2022). https://www.datacamp.com/tutorial/tutorial-ridge-lasso-elastic-net

Stephanie. (2015, September 24). *Lasso Regression: Simple Definition*. Statistics How To. https://www.statisticshowto.com/lasso-regression/

Stephanie. (2021, August 20). *Box-Cox Transformation: Definition, Examples*. Statistics How To. https://www.statisticshowto.com/box-cox-transformation/

SVM | Support Vector Machine Algorithm in Machine Learning. (2017, September 12). *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

Technavio. (2022, March 25). *Used Car Market size in the US to increase by 3.91 million units | High growth expected in mid-size segment | Technavio*. https://www.prnewswire.com/news-releases/used-car-market-size-in-the-us-to-increase-by-3-91-million-units--high-growth-expected-in-mid-size-segment--technavio-301510065.html

U.S.: Average selling price of new vehicles 2021. (2022). Statista. https://www.statista.com/statistics/274927/new-vehicle-average-selling-price-in-the-united-states/

US Economic Contributions. (2020). US Economic Contributions | American Automotive Policy Council. https://www.americanautomakers.org/us-economic-contributions

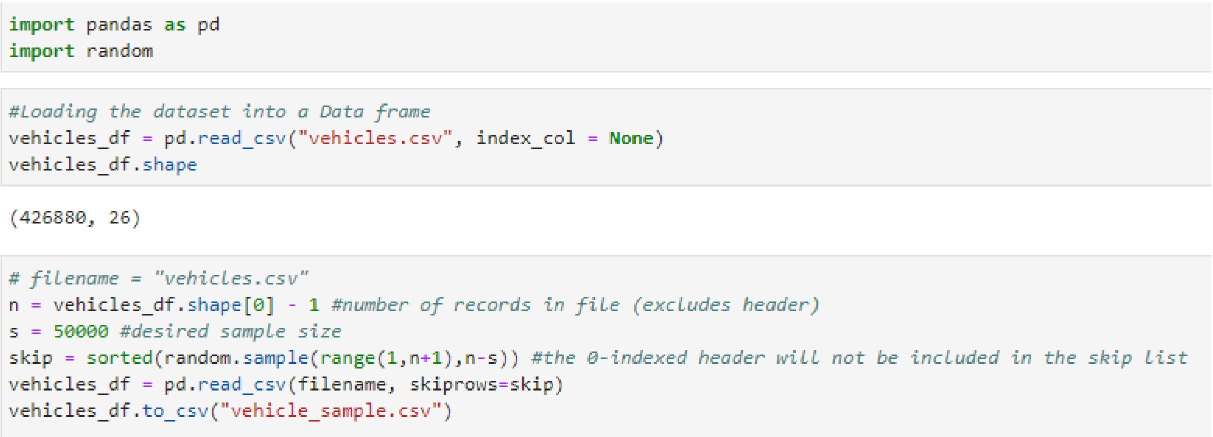
Used Car Price Trends—CarGurus. (n.d.). Retrieved May 29, 2022, from https://www.cargurus.com/Cars/price-trends/

Used Cars Dataset. (n.d.). Retrieved May 30, 2022, from https://www.kaggle.com/austinreese/craigslist-carstrucks-data

ushistory.org. (n.d.). *The Age of the Automobile [ushistory.org]*. Retrieved May 30, 2022, from https://www.ushistory.org/us/46a.asp

Wu, J.-D., Hsu, C.-C., & Chen, H.-C. (2009). An expert system of price forecasting for used cars using adaptive neuro-fuzzy inference. *Expert Systems with Applications*, *36*(4), 7809–7817. https://doi.org/10.1016/j.eswa.2008.11.019

Appendix:



```
import pandas as pd
import random
```

```
#Loading the dataset into a Data frame
vehicles_df = pd.read_csv("vehicles.csv", index_col = None)
vehicles_df.shape
```

```
(426880, 26)
```

```
# filename = "vehicles.csv"
n = vehicles_df.shape[0] - 1 #number of records in file (excludes header)
s = 50000 #desired sample size
skip = sorted(random.sample(range(1,n+1),n-s)) #the 0-indexed header will not be included in the skip list
vehicles_df = pd.read_csv(filename, skiprows=skip)
vehicles_df.to_csv("vehicle_sample.csv")
```

Figure 25: sampling the data set

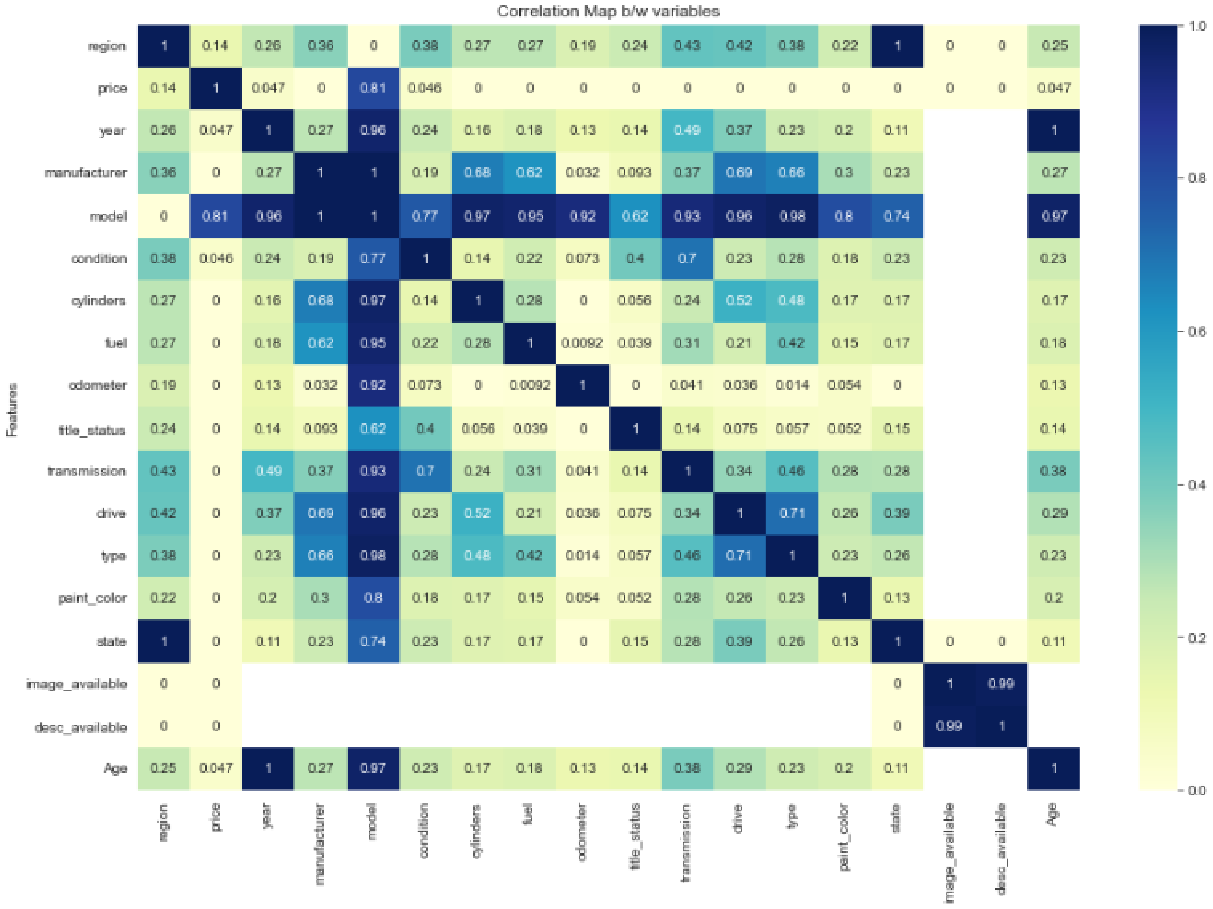
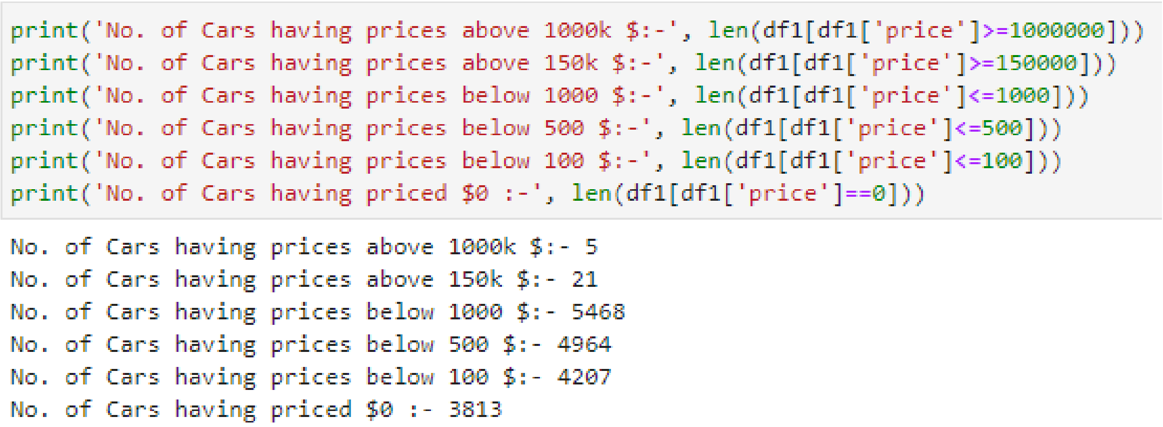


Figure 26: Phi_k correlation matrix heatmap



```
print('No. of Cars having prices above 1000k $:-', len(df1[df1['price']>=1000000]))
print('No. of Cars having prices above 150k $:-', len(df1[df1['price']>=150000]))
print('No. of Cars having prices below 1000 $:-', len(df1[df1['price']<=1000]))
print('No. of Cars having prices below 500 $:-', len(df1[df1['price']<=500]))
print('No. of Cars having prices below 100 $:-', len(df1[df1['price']<=100]))
print('No. of Cars having priced $0 :-', len(df1[df1['price']==0]))
```

```
No. of Cars having prices above 1000k $:- 5
No. of Cars having prices above 150k $:- 21
No. of Cars having prices below 1000 $:- 5468
No. of Cars having prices below 500 $:- 4964
No. of Cars having prices below 100 $:- 4207
No. of Cars having priced $0 :- 3813
```

Figure 27: price distribution values

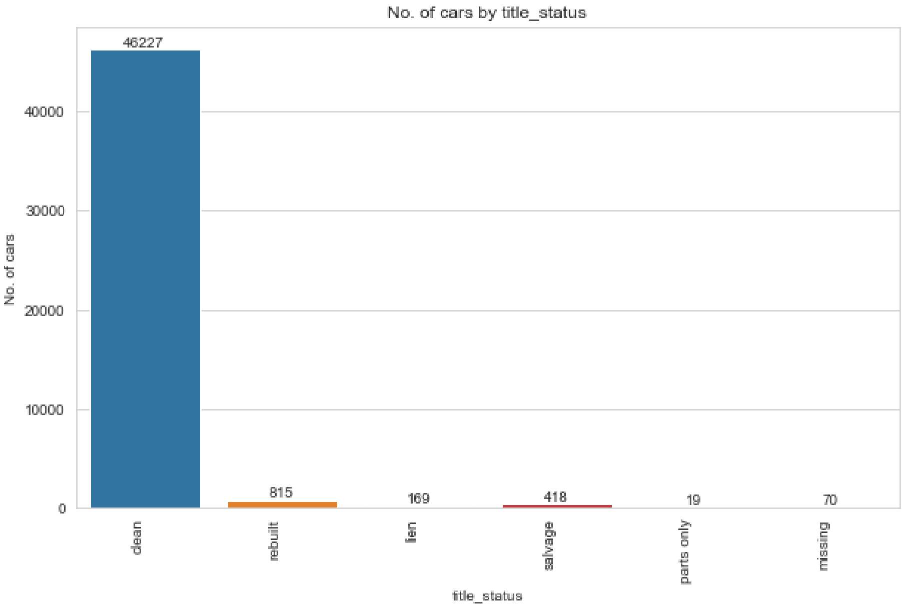
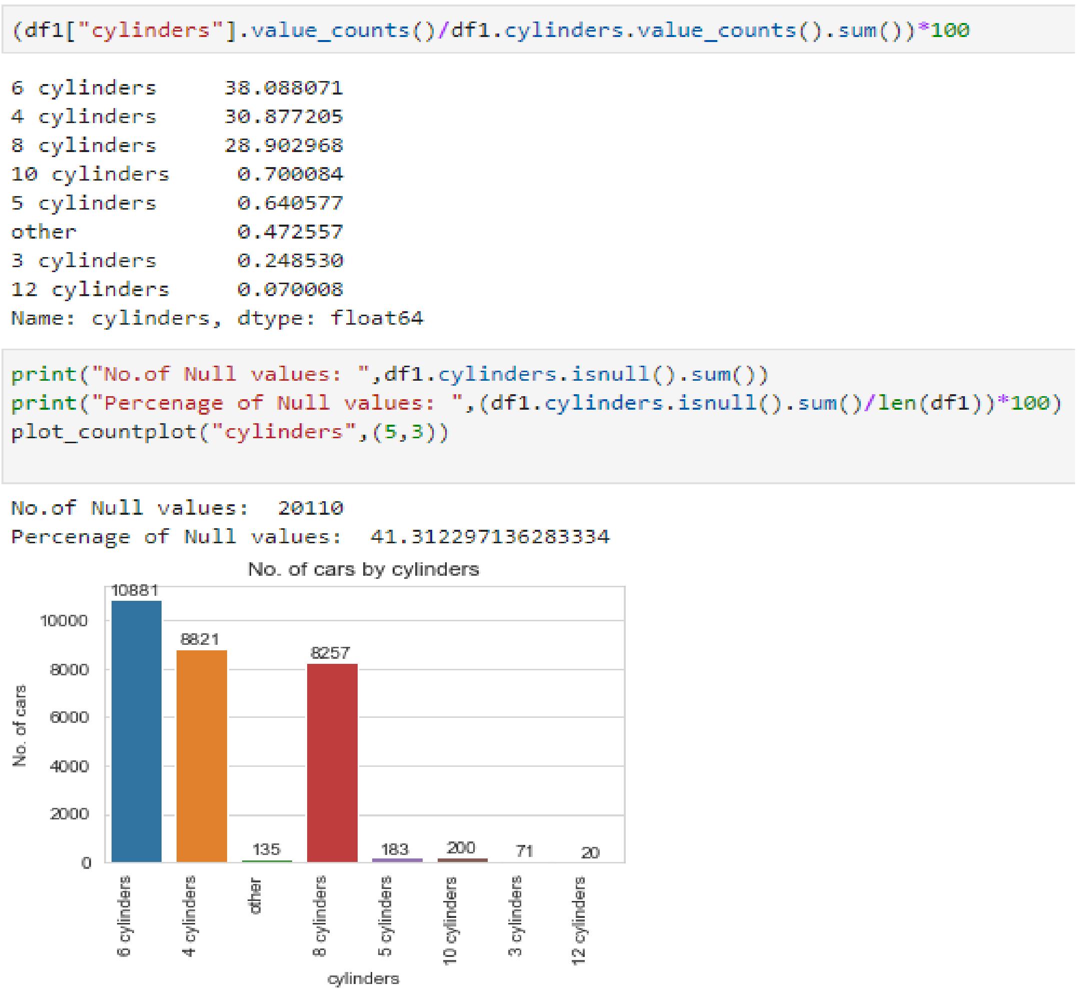


Figure 28: title_status distribution



```
(df1["cylinders"].value_counts()/df1.cylinders.value_counts().sum())*100
```

```
6 cylinders     38.088071
4 cylinders     30.877205
8 cylinders     28.902968
10 cylinders     0.700084
5 cylinders      0.640577
other            0.472557
3 cylinders      0.248530
12 cylinders     0.070008
Name: cylinders, dtype: float64
```

```
print("No.of Null values: ",df1.cylinders.isnull().sum())
print("Percenage of Null values: ",(df1.cylinders.isnull().sum()/len(df1))*100)
plot_countplot("cylinders",(5,3))
```

```
No.of Null values:  20110
Percenage of Null values:  41.312297136283334
```

Figure 29: Cylinder distribution and Null values

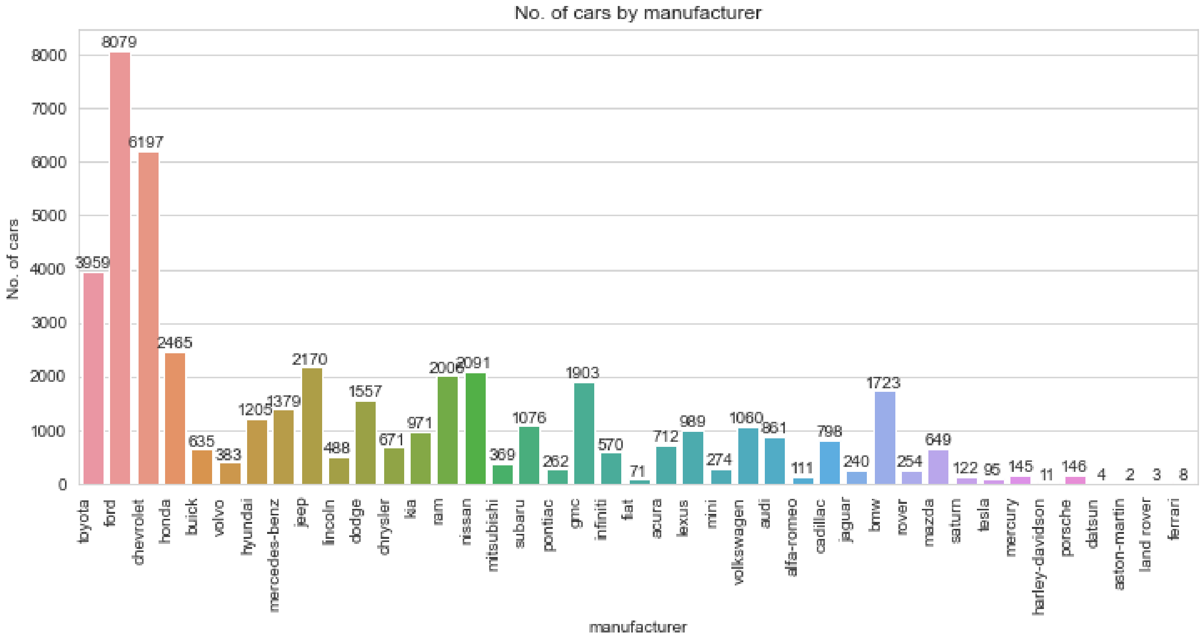


Figure30: Manufacturer distribution

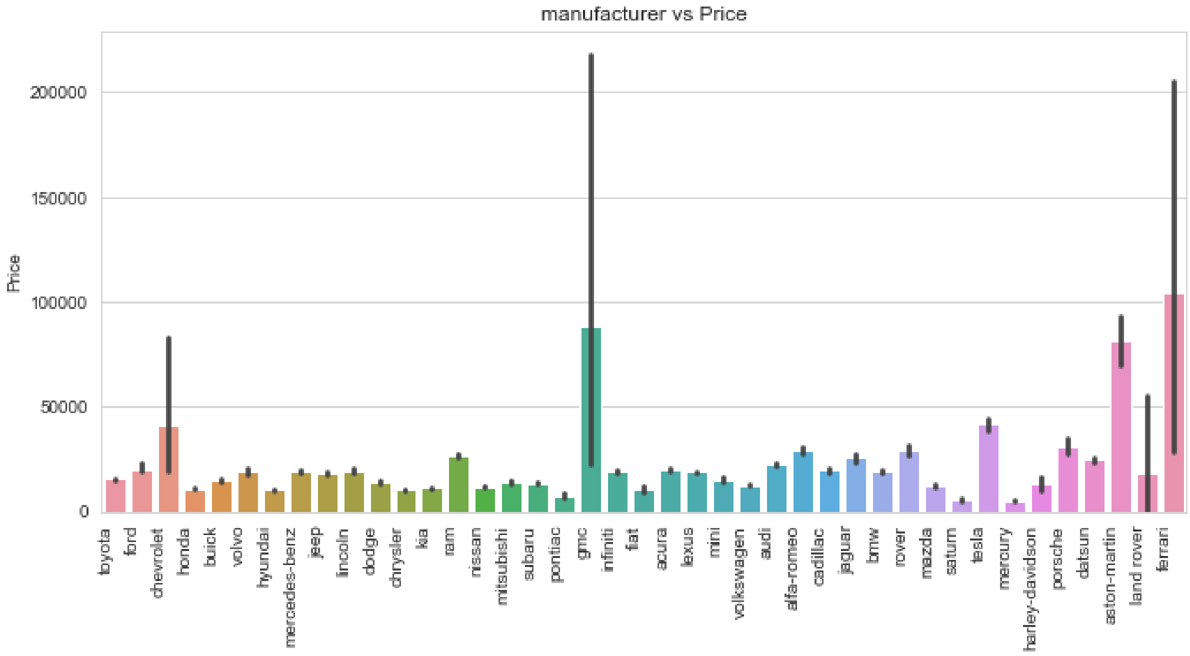


Figure 31: manufacturer vs price distribution

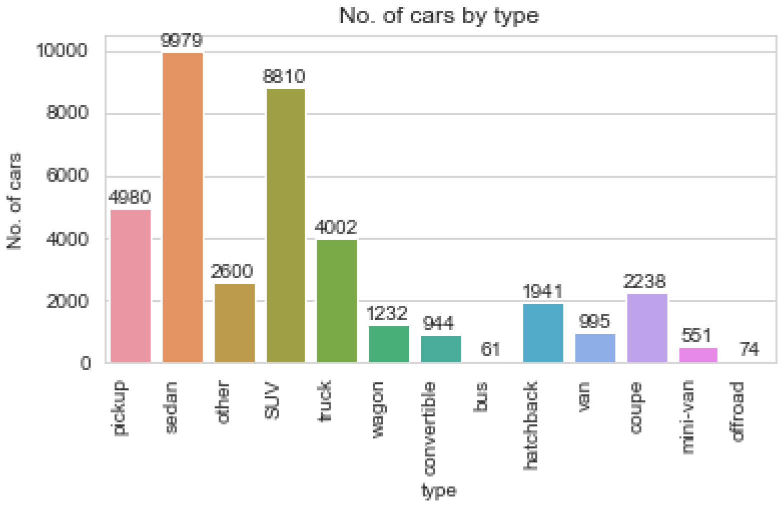
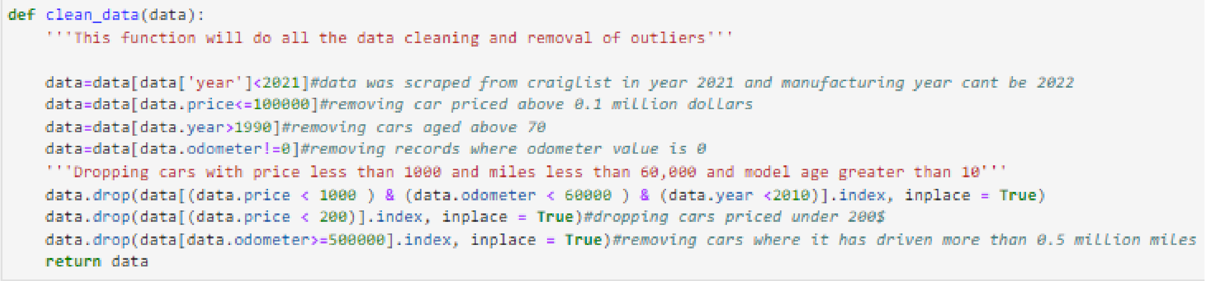


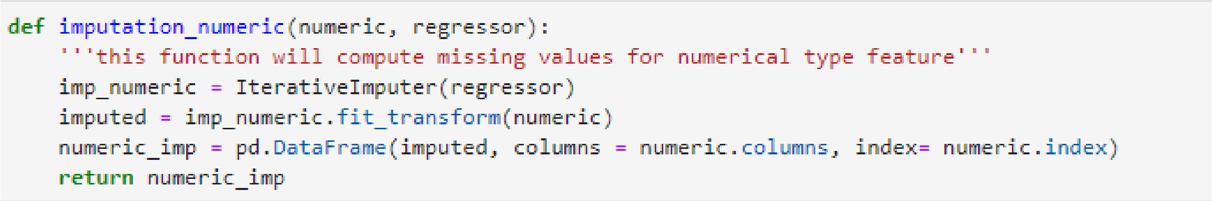
Figure 18: type Distribution



```
def clean_data(data):
    '''This function will do all the data cleaning and removal of outliers'''

    data=data[data['year']<2021]#data was scraped from craiglist in year 2021 and manufacturing year cant be 2022
    data=data[data.price<=100000]#removing car priced above 0.1 million dollars
    data=data[data.year>1990]#removing cars aged above 70
    data=data[data.odometer!=0]#removing records where odometer value is 0
    '''Dropping cars with price less than 1000 and miles less than 60,000 and model age greater than 10'''
    data.drop(data[(data.price < 1000 ) & (data.odometer < 60000 ) & (data.year <2010)].index, inplace = True)
    data.drop(data[(data.price < 200)].index, inplace = True)#dropping cars priced under 200$
    data.drop(data[data.odometer>=500000].index, inplace = True)#removing cars where it has driven more than 0.5 million miles
    return data
```

Figure 33: Clean_data() – function



```python
def imputation_numeric(numeric, regressor):
    '''this function will compute missing values for numerical type feature'''
    imp_numeric = IterativeImputer(regressor)
    imputed = imp_numeric.fit_transform(numeric)
    numeric_imp = pd.DataFrame(imputed, columns = numeric.columns, index= numeric.index)
    return numeric_imp
```

Figure 34: imputation_numeric() - function