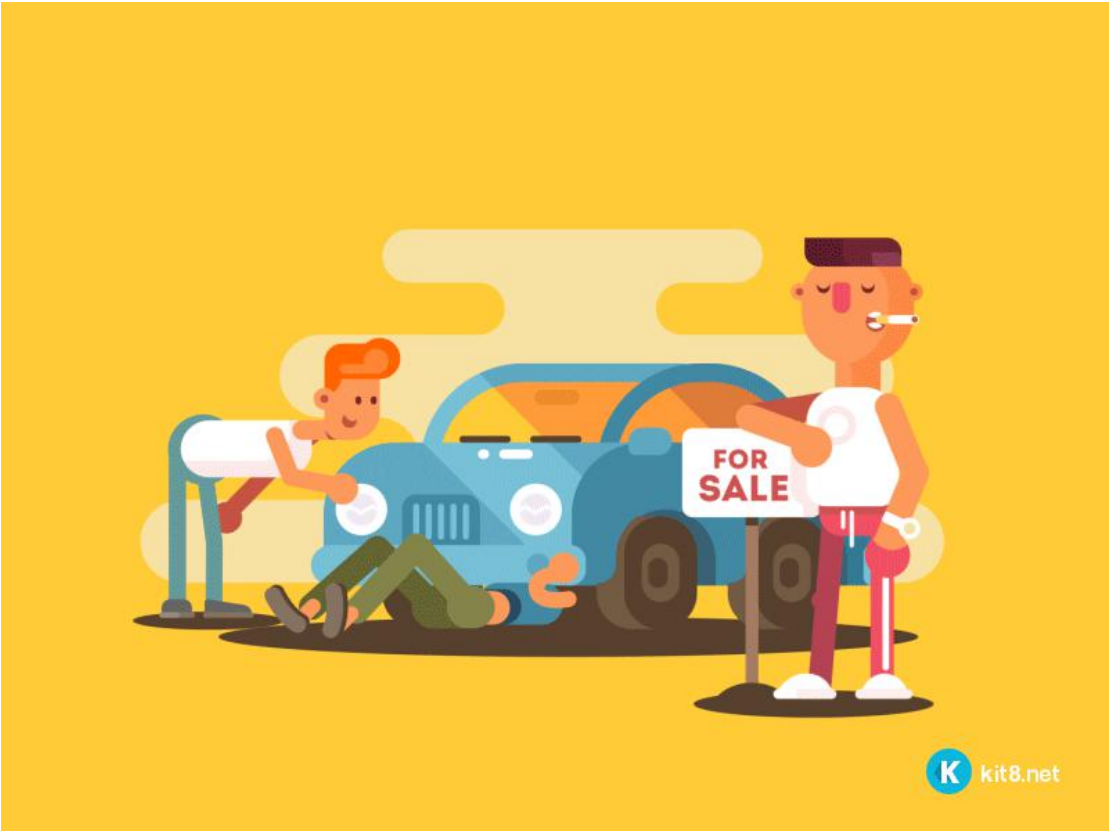
FLIP ROBO

USED CARS PRICE PREDICTION



Submitted by:

ArunPrasad.k

Table of content

Abstract – A car price prediction has been a high interest research area, as it requires noticeable effort and knowledge of the field expert. The production of cars has been steadily increasing in the past decade, with over 70 million passenger cars being produced in the year 2018. This has given rise to the used car market, which on its own has become a booming industry. The recent advent of online portals has facilitated the need for both the customer and the seller to be better informed about the trends and patterns that determine the value of a used car in the market.Considerable number of distinct attributes are examined for the reliable and accurate prediction. To build a model for predicting the price of used cars in INDIA, we applied machine Learning Algorithms such as Lasso Regression, Multiple Regression and Regression trees, we will try to develop a statistical model which will be able to predict the price of a used car, based on previous consumer data and a given set of features. We will also be comparing the prediction accuracy of these models to determine the optimal one.

Keywords – car price prediction, support vector machines, classification, machine learning

---

1. INTRODUCTION

The used car market is an ever-rising industry, which has almost doubled its market value in the last few years. The emergence of online portals such as CarDheko, OLX, Carwale, CarsTrade, and many others has facilitated the need for both the customer and the seller to be better informed about the trends and patterns that determine the value of the used car in the market. Machine Learning algorithms can be used to predict the retail value of a car, based on a certain set of features.

Different websites have different algorithms to generate the retail price of the used cars, and hence there isn't a unified algorithm for determining the price. By training statistical models for predicting the prices, one can easily get a rough estimate of the price without actually entering the details into the desired website. The main objective of this paper is to use various different prediction models to predict the retail price of a used car and compare their levels of accuracy.

2. PROBLEM STATEMENT

With the covid 19 impact in the market, we have seen lot of changes in the car market. Now some cars are in demand hence making them costly and some are not in demand hence cheaper. One of our clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models. So, they are looking for new machine learning models from new data.

3.REQUIREMENTS

Hardware requirements :

i. Operating system- Windows 7,8,10

ii. Processor- dual core 2.4 GHz (i5 or i7 series Intel processor or equivalent AMD)

iii. RAM-4GB

Software Requirements :

iv. Python

v. Jupyter Notebook

vi. Chrome

4.DATA SCRAPING

Data is collected from a local web portal for selling and buying cars, as time interval itself has high impact on the price of the cars in various big cities in India.

The following attributes were captured for each car: brand, model, location, fuel, year of manufacturing, price ,kilometers, number of owners. Since manual data collection is time consuming task, especially when there are numerous records to process, a "web scraper" as a part of this research is created to get this job done automatically and reduce the time for data gathering.

Web scraping is well known technique to extract information from websites and save data into local file or database. Manual data extraction is time consuming and therefore web scrapers are used to do this job in a fraction of time.

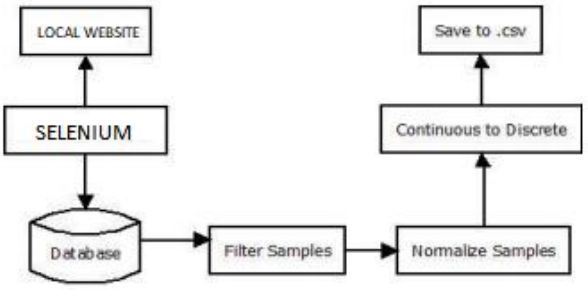


Figure 2. Data gathering and transformation workflow diagram

Web scrapers are programed for specific websites and can mimic regular users from website's point of view. After raw data has been collected and stored to local database, data pre-processing step was applied. Finally we save the Processed data set sample in CSV format.

5.DATA OVERVIEW

The data samples contain 8 fields which includes **brand, model, year, kilometers, fuel, no_of_owners, location, price.**

- ✓ **Price**: The calculated retail price of used cars. The cars which were selected for this data set were all less than a year old and were considered to be in good condition.
- ✓ **Brand**: The manufacturer of the car.
- ✓ **Model**: The specific models for each car.
- ✓ **No.of.owners**: How many owners a car has had.
- ✓ **Year:** The manufacturing year of the car Model.
- ✓ **Fuel**: Type of Fuel used in the car.
- ✓ **Kilometers**: The total number of kilometers the car has been driven.
- ✓ **Location**: Location the car is available.

6.METHODOLGY

There are two primary phases in the system:

1. **Training phase**: The system is trained by using the data in the data set and fits a model (line/curve) based on the algorithm chosen accordingly.

2. **Testing phase**: the system is provided with the inputs and is tested for its working. The accuracy is checked. And therefore, the data that is used to train the model or test it, has to be appropriate.

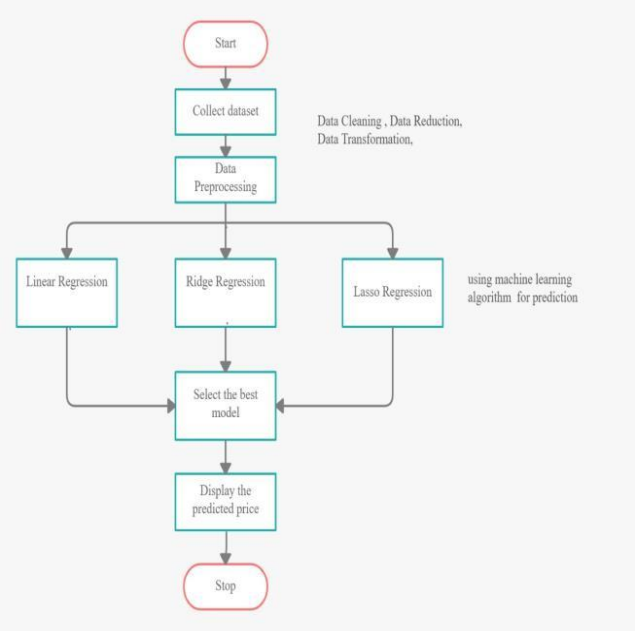


Figure 3:**Proposed System Flowchart**

The system is designed to detect and predict price of used car and hence appropriate algorithms must be used to do the two different tasks. Before the algorithms are selected for further use, different algorithms were compared for its accuracy. The well-suited one for the task was chosen.

7.DESIGN APPROACH

7.1. Linear regression:

Simple linear regression statistical method allows us to summarize and study the relationship between two continuous quantitative variables.

- ✓ One variable, denoted x, is regarded as the predictor, explanatory, or independent variable.
- ✓ The other variable, denoted y, is regarded as the response, outcome, or dependent variable

7.2. Lasso Regression:

Least Absolute Shrinkage and Selection Operator (Lasso) is an L1 norm regularised regression technique that was formulated by Robert Tibshirani in 1996 . Lasso is a powerful technique that performs regularisation and feature selection. Lasso introduces a bias term, but instead of squaring the slope like Ridge regression, the absolute value of the slope is added as a penalty term.

Lasso is defined as:

$$L = \text{Min}(\text{sum of squared residuals} + \alpha * [\text{slope}])$$

Where Min(sum of squared residuals) is the Least Squared Error, and + α ∗ [slope]) is the penalty term. However, alpha α is the tuning parameter which controls the strength of the penalty term. In other words, the tuning parameter is the value of shrinkage.

7.3. Ridge Regression:

The Ridge Regression is an L2 norm regularised regression technique that was introduced by Hoerl in 1962. It is an estimation procedure to manage col linearity without removing variables from the regression model.

In multiple linear regression, the multicollinearity is a common problem that leads least square estimation to be unbiased, and its variances are far from the correct value.

Therefore, by adding a degree of bias to the regression model, Ridge Regression reduces the standard errors, and it shrinks the least square coefficients towards the origin of the parameter space .

Ridge formula is

$$L = \text{Min}(\text{sum of squared residuals} + \alpha * [\text{Slope}]_2$$

Where Min(sum of squared residuals) is the Least Squared Error, and $[\text{Slope}]_2$ is the penalty term that Ridge adds to the Least Squared Error.

7.4. Random Forest Regression:

A Random Forest is an ensemble technique qualified for performing classification and regression tasks with the help of multiple decision trees and a method called Bootap Aggregation known as Bagging .Decision Trees are used in

classification and regression tasks, where the model (tree) isformed of nodes and branches. The tree starts with a root node, while the internal nodes correspond to an input attribute. The nodes that do not have children are called leaves, where each leaf performs the prediction of the output variable.

A Decision Tree can be defined as
$\phi = x \rightarrow y$

7.5. Gradient Boosting algorithm:.

Gradient boosting is a machine Taking in strategy to relapse Also arrangement problems, that produces a prediction model in the structure of an group from claiming powerless prediction models.

The exactness of a predictive model might be helped to two ways. Possibly by grasping characteristic building alternately. Toward applying boosting calculations straight far. There are a significant number boosting calculations in

- ✓ Gradient Boosting
- ✓ XGBoost
- ✓ AdaBoost
- ✓ Gentle Boost etc.

Each boosting algorithm need its own underlying math. Also, a slight variety may be watched same time applying them. Boosting calculation will be a standout among those The greater part capable Taking in thoughts acquainted in the final one twenty A long time. It might have been intended to order problems, yet all the it can be developed should relapse too.

The inspiration to gradient boosting might have been An technique. That combines those outputs about large portions "weak" classifiers to process An capable "committee. " a powerless classifier will be person whose slip rate is main superior to irregular guessing.

8. IMPLEMENTATION

8.1. Importing Dataset:

Reading the data to plot the graphs:

```
import io
df = pd.read_excel(io.BytesIO(uploaded['usedcars_webscraping.xlsx']))
df
```

	Unnamed: 0	Brand	Model	Year	Kilometers	Fuel	No.of.owners	Location	Price
0	0	Nissan	Sunny	2017.0	86000.0	Diesel	1st	KOCHI	399999.0
1	1	Maruti Suzuki	Baleno	2018.0	51300.0	CNG & Hybrids	1st	RAJKOT	651000.0
2	2	NaN	NaN	NaN	NaN	NaN	NaN	DELHI	618000.0
3	3	Mahindra	Bolero Power Plus	2019.0	28400.0	Diesel	1st	BAREILLY	665000.0
4	4	Honda	Brio	2011.0	48000.0	Petrol	1st	DELHI	235000.0
...	...	...	...	...	...	...	...	...	...
11417	9601	Audi	Q3	2017.0	44600.0	Petrol	1st	Gurgaon	2400000.0
11418	9602	Audi	A3	2017.0	34000.0	Diesel	1st	Gurgaon	2200000.0
11419	9603	Mercedes-Benz	E-Class Cabriolet	2016.0	33100.0	Petrol	2nd	Delhi	4475000.0
11420	9604	Audi	A4	2011.0	70000.0	Petrol	2nd	Delhi	900000.0

8.2 Data Preprocessing

Before applying any type of data analytics on he data set, the data should be first cleaned.

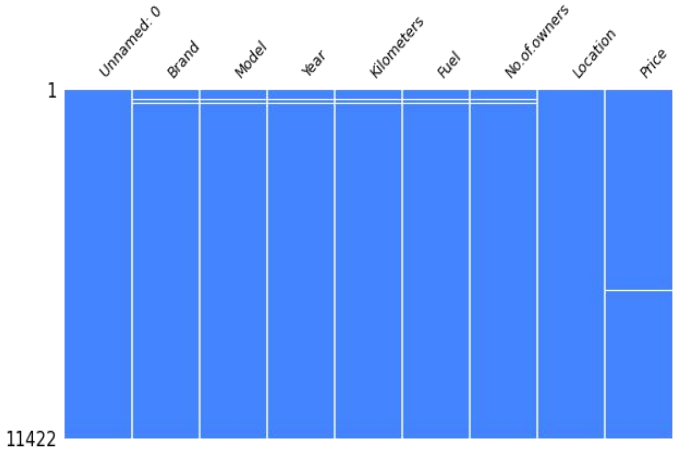


Figure 3 Heatmap on missing values

We find many attributes with missing values.The following is the list of Columns with missing values.

1. brand,
2. model,
3. year,
4. kilometers,
5. fuel,
6. no.of.owners,
7. Price.

It is very important to not loss too much data.But still filling the NaN values with mean or median or mode may result in high bias if we train ML algorithms especially when the column has high percentage of missing values.Using mean value for replacing missing values may not create a great model and hence gets ruled out. Since the missing values are very low we prefer to drop all the missing values.

```
#Removing NaN values
df.dropna(axis="rows",inplace=True)
df
```

Rechecking for the Missing value.Here we find there are no missing values in the data set which has been handled in this case**.**

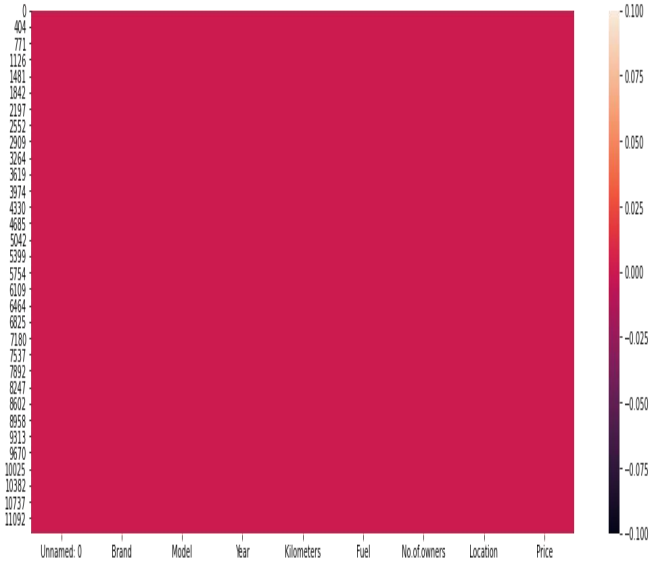


Figure 4 Heatmap on missing values

9. EXPLORATORY DATA ANALYSIS

We are going to perform exploratory data analysis for our problem in the first stage. In exploratory data analysis data set is explored to figure out the features which would influence the used car price. The data is deeply analyzed by finding a relationship between each attribute and our target label.

9.1 Uni-variate Analysis

i) Target variable “price”

First we will see on the Basic Descriptions on the target variable “price”.

```
#description about column SalesPrice
df["Price"].describe()
```

```
count    1.133500e+04
mean     3.497560e+06
std      1.359257e+08
min      1.500000e+04
25%      4.349990e+05
50%      7.200000e+05
75%      1.525000e+06
max      9.000000e+09
Name: Price, dtype: float64
```

```
#Let's visualize the distribution of sale price
plt.figure(figsize=(18, 8))
sns.distplot(df['Price'])
plt.show()
#skewness and kurtosis
print("Skewness: %f" % df['Price'].skew())
print("Kurtosis: %f" % df['Price'].kurt())
```

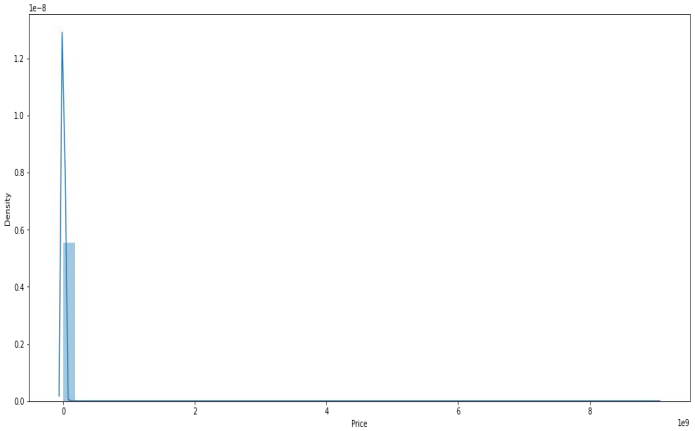


Figure 5 Displacement graph on Target

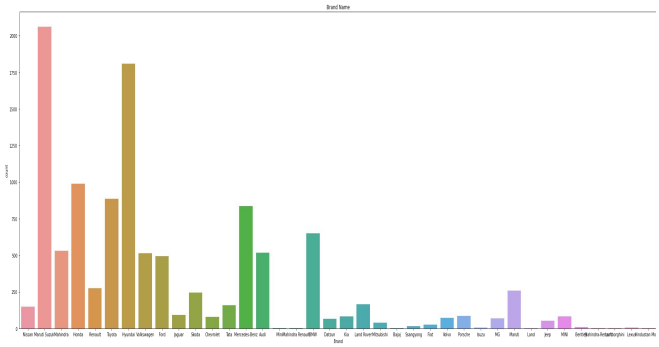
ii) Categorical Features

The following are some of the feature with categorical features

- ✓ Brand
- ✓ Model
- ✓ Fuel
- ✓ No.of.owners
- ✓ Location

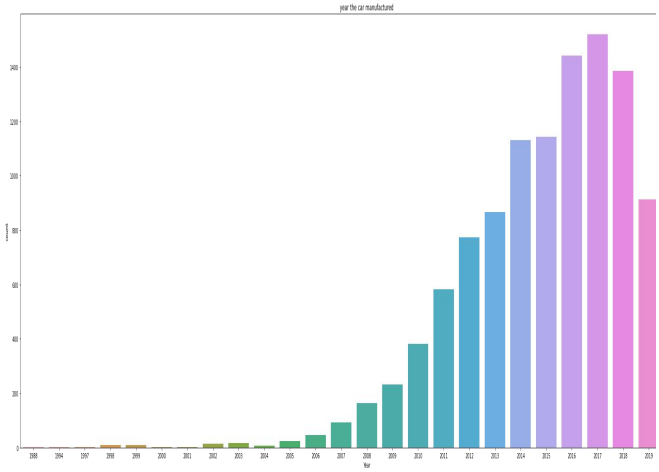
Brand:

Vehicle or Car branding is a term used to describe anything on your vehicle that promotes your business from a sticker on the side to an entire vehicle wrap.



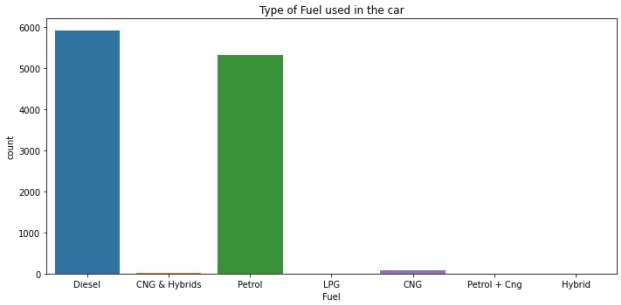
Model:

The model refers to the name of a car product and sometimes a range of products.



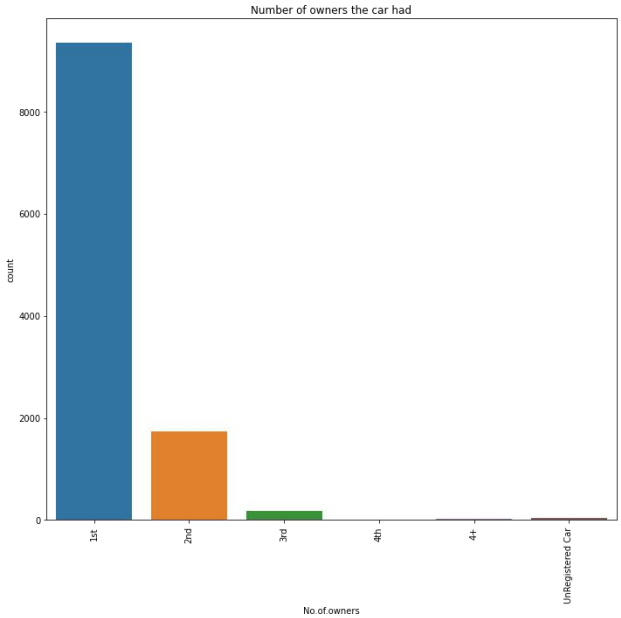
Fuel:

Currently, the majority of motor vehicles worldwide are powered by **gasoline or diesel**. Other energy sources include ethanol, biodiesel, propane, compressed natural gas (CNG), electric batteries, and hydrogen (either using fuel cells or combustion). There are also cars that use a hybrid of different power sources.



No.of.owners:

The number of owners and its influence on the value of a **car completely depends on the body type and its age**. For example, it's considered acceptable if an older convertible car has had over five owners because it has most likely been bought for summer enjoyment and then moved on come winter.



Plot Insight:

- **Brand** - Brand name Maruthi suzuki has been seen more for the sale.
- **Model**- model name city of honda brand is high for sale with respect to model value count.
- **year**- car with manufacturing year 2017 was found to be more in the dataset respect to model year count.
- **Fuel**- Diesel and petrol cars is found in almost equal numbers.
- **No.of.owners**- 1st owners cars have seen high in dataset.

10. Feature Engineering

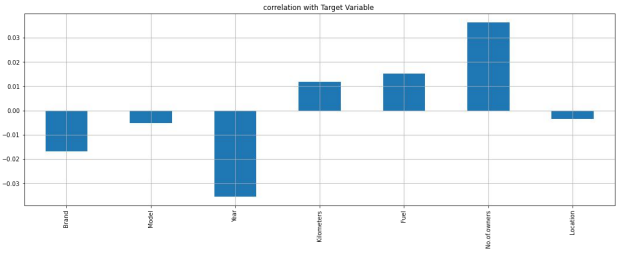
Feature engineering is the most important part of data analytic process. It deals with, selecting the features that are used in training and making predictions. In feature engineering the domain knowledge is used to find features in the dataset which are helpful in building machine learning model. It helps in understanding the dataset in terms of modeling. A bad feature selection may lead to less accurate or poor predictive model. The accuracy and the predictive power depend on the choice of correct features. It filters out all the unused or redundant features.

10.1 Categorical encoding using Label-Encoder

Label Encoding refers to converting the **labels** into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those **labels** must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

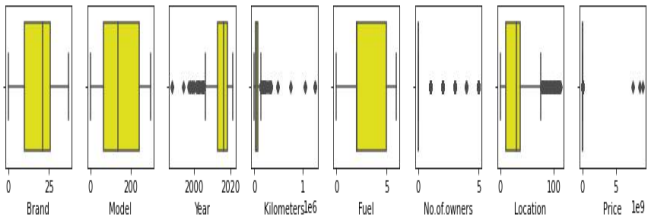
```
#From the given data-set we can infer that some of the data types are categorical
#we convert them to integer type by using the label encoder method
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df['Brand'] = le.fit_transform(df['Brand'])
df['Model'] = le.fit_transform(df['Model'])
df['Location'] = le.fit_transform(df['Location'])
df['No.of.owners'] = le.fit_transform(df['No.of.owners'])
df['Fuel'] = le.fit_transform(df['Fuel'])
```

10.2 Correlation



10.3. Looking for Outliers

The difference between a good and an average machine learning model is often its ability to clean data. One of the biggest challenges in data cleaning is the identification and treatment of outliers. In simple terms, outliers are observations that are significantly different from other data points. Even the best machine learning algorithms will underperform if outliers are not cleaned from the data because outliers can adversely affect the training process of a machine learning algorithm, resulting in a loss of accuracy.



10.4.Treating Outliers:Z score Technique

In this procedure we calculate the z-score for each observation (fix this). Any z-score greater than 3 or less than -3 is considered to be an outlier. This rule of thumb

is based on the empirical rule. From this rule we see that almost all of the data (99.7%) should be within three standard deviations from the mean. By calculating the z-score we are standardizing the observation, meaning the standard deviation is now 1. Thus from the empirical rule we expect 99.7% of the z-scores to be within -3 and 3.

```
#Removing Outliers
#Z-score Techinique
from scipy.stats import zscore
z=np.abs(zscore(df))
z
```

Percentage Loss

```
loss_of_data=(11335-10955)/11335*100
```

loss_of_data=3.3524

The loss of data is only 4% which can be acceptable.so proceeding with outlier removal data.

```
threshold=3
print(np.where(z>3))
df_new=df[(z<3).all(axis=1)]
```

11.Splitting data to fit any Machine learning model

After we have performed data cleaning, data visualizations, and learned details about our data it is time to fit the first machine learning model into it.

Separating features from the target variable:

We should start with separating features for our model from the target variable. Notice that in our case all columns except 'Price' are features that we want to use for the model. Our target variable is 'Price'. We can use the following code to do target separation.

```
#importing Libraries
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.20,random_state=42)
```

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
((8764, 7), (2191, 7), (8764,), (2191,))
```

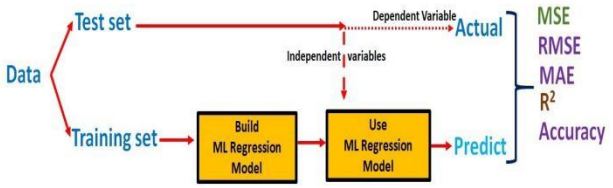
The data that you have prepared is now ready to be fed to the machine learning model.

Scaling Input

Many machine learning algorithms perform better when numerical input variables are scaled to a standard range. This includes algorithms that use a weighted sum of the input, like logical regression, and algorithms that use distance measures, like k-nearest neighbors.

12.Machine Learning Models

Various machine learning models are implemented to validate and predict the used cars price.



First we will import all the necessary Libraries of our Machine Learning Model.

```
#importing our model libraries
from sklearn.linear_model import LinearRegression,Lasso,Ridge,ElasticNet
lr=LinearRegression()
ls=Lasso()
rd=Ridge()
en=ElasticNet()
from sklearn.svm import SVR
svr=SVR()
from sklearn.neighbors import KNeighborsRegressor
knn=KNeighborsRegressor()
from sklearn.tree import DecisionTreeRegressor
dt=DecisionTreeRegressor()
#importing error Metrics
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score

```

Once the libraries are imported,we proceed to predict the required output with the help of different Machine Learning algorithm. The Different Machine learning algorithm includes

1. Linear Regression
2. Lasso
3. Ridge
4. Elastic Net
5. SVR
6. K Neighbors Regressor
7. Decision Tree Regressor

```
#scoring the model
model=[lr,svr,knn,ls,rd,en]
for m in model:
    m.fit(x_train,y_train)
    print("Training score of ",m,"is",m.score(x_train,y_train))
    predm=m.predict(x_test)
    print("r2 score is :",r2_score(y_test,predm))
    print("Error:")
    print("mean_absolute_error is :",mean_absolute_error(y_test,predm))
    print("mean_squared_error is :",mean_squared_error(y_test,predm))
    print("root mean_absolute_error is :",np.sqrt(mean_squared_error(y_test,predm)))

    print("**************************************************")
    print("\n\n")

```

```
#using Random Forest Regressor
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor()
rf.fit(x_train,y_train)
rf_pred=rf.predict(x_test)
print("score is",rf.score(x_train,y_train))
print("r2 score is",r2_score(y_test,rf_pred))
print("mean absolute error is : ",mean_absolute_error(y_test,rf_pred))
print("mean squared error is : ",mean_squared_error(y_test,rf_pred))
print("root mean absolute error is : ",np.sqrt(mean_absolute_error(y_test,rf_pred)))

```

```
#using AdaBoostRegressor
from sklearn.ensemble import AdaBoostRegressor
rf=RandomForestRegressor()
ada=AdaBoostRegressor(base_estimator=rf,n_estimators=20,learning_rate=0.1,random_state=1)
ada.fit(x_train,y_train)
ada_pred=ada.predict(x_test)
ada_score=ada.score(x_train,y_train)
print("score is",ada_score)
print("r2 score is",r2_score(y_test,ada_pred))
print("mean absolute error is : ",mean_absolute_error(y_test,ada_pred))
print("mean squared error is : ",mean_squared_error(y_test,ada_pred))
print("root mean absolute error is : ",np.sqrt(mean_absolute_error(y_test,ada_pred)))
```

Model	**Training accuracy**	**Testing accuracy**
LinearRegression	19%	19%
SVR	15%	-6%
KNeighborsRegressor	85%	74%
Lasso	19%	19%
Ridge	19%	19%
ElasticNet	17%	17%
RandomForestRegressor	99%	97%
AdaBoostRegressor	99%	98%

13.MODEL EVALUATION

The model developed in this research will be tested using several methods such as Mean Absolute Percentage Error(MAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). MAPE is calculated by making an average percentage of the absolute error of each predicted result. Thus, MAPE can indicate how much prediction error.

R-squared (R^2), Mean Squared Error (MSE), Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are the most commonly used metrics to measure accuracy

for continuous variables. In this post, we will observe the coefficient of variables (CoV) effect on the MAE, MSE, R^2, and accuracy. We will apply the same linear regression to 4 different data which has variables with different coefficients to explain how and why the MSE, MAE, R^2, and Accuracy are changing. First, while we keep the MSE and MAE fixed, we will observe the R^2 and accuracy with the change of coefficient of variables. Secondly, while we keep the R^2, and accuracy set constant, we willobserve the MSE and MAE with the change of coefficient of variables.

I) Mean Absolute Error (MAE):

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

If the absolute value is not taken (the signs of the errors are not removed), the average error becomes the Mean Bias Error (MBE) and is usually intended to measure average model bias. MBE can convey useful information, but should be interpreted cautiously because positive and negative errors will cancel out.

II) Mean absolute percentage error (MAPE):

The mean absolute percentage error (MAPE) is a measure of how accurate a forecast system is. It measures this accuracy as a percentage, and can be calculated as the average absolute percent error for each time period minus actual values divided by actual values.

$$\mathrm{M} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

Where:
n is the number of fitted points,
At is the actual value,
Ft is the forecast value.
Σ is summation notation (the absolute value is summed for every forecasted point in time). The mean absolute percentage error (MAPE) is the most common measure used to forecast error, and works best if there are no extremes to the data (and no zeros).

III) Root Mean Square Error (RMSE):

RMSE is a quadratic scoring rule that also measures the average magnitude of the error. It's the square root of the average of squared differences between prediction and actual observation.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Model	MAE	MSE	RMAE
LinearRegression	864633	1661930	12891
SVR	852664	2387939	65452
KNeighborsRegressor	864633	1661930	12891
Lasso	864633	1661930	12891
Ridge	864633	1661930	12891
ElasticNet	864633	1661930	12891
RandomForest Regressor	74379	4527054	272.72
AdaBoostRegressor	64796	3253376	254.55

14.PREDICTION

Since we have evaluated all models by using **MAE,RMSE,MAPE** we will predict by using model which has highest accuracy.Here we can choose Adaboost Regressor models to predict the Car Price .

14.1 Hyper Tuning using GridSearchCV

Hyperparameters are crucial as they control the overall behavior of a machine learning model. The ultimate goal is to find an optimal combination of hyperparameters that minimizes a predefined loss function to give better results.

```
#AdaBoostRegressor
#using GridSearchCV
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import AdaBoostRegressor
parameters={"n_estimators" :[1,10,100],
            "learning_rate" : [0.15,0.1,0.05,0.01],
            "loss" : ['linear', 'square', 'exponential']}
ada=AdaBoostRegressor()
clf=GridSearchCV(ada,parameters)
clf.fit(x_train,y_train)
print(clf.best_params_)
```

```
{'learning_rate': 0.15, 'loss': 'square', 'n_estimators': 10}
```

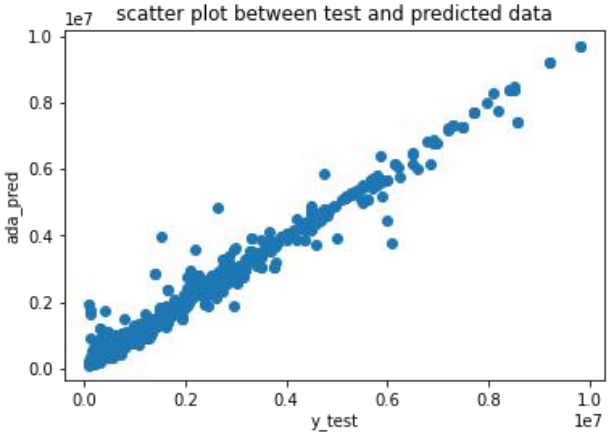
Here we first find the best parameters for the Adaboost regressor Model and indulge it to the model to improve our predicting accuracy. There are several ways for finding the parameters, here we use the most powerful and most commonly used method named GridSearchCV.

14.2 Further Evaluation

14.2.1 Cross-validation

The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias and to give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem).

14.2.2 Scatter Plot b/w Test-data and Predicted-data



15. CONCLUSION

Car price prediction can be a challenging task due to the high number of attributes that should be considered for the accurate prediction. The major step in the prediction process is collection and preprocessing of the data. In this research, PHP scripts were built to normalize, standardize and clean data to avoid unnecessary noise for machine learning algorithms. Data cleaning is one of the processes that increases prediction performance, yet insufficient for the cases of complex data sets as the one in this research. Applying single machine algorithm on the data set accuracy was less than 50%. Therefore, the ensemble of multiple machine learning algorithms has been proposed and this combination of ML methods gains accuracy of 99%. This is significant improvement compared to single machine learning method approach. However, the drawback of the proposed system is that it consumes much more computational resources than single machine learning algorithm.

Although, this system has achieved astonishing performance in car price prediction problem our aim for the future research is to test this system to work successfully with various data sets. We will extend our test data

with eBay and OLX used cars data sets and validate the proposed approach.

Future scope: In future this machine learning model may bind with various website which can provide real time data for price prediction. Also we may add large historical data of car price which can help to improve accuracy of the machine learning model. We can build an android app as user interface for interacting with user. For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data rather than the whole dataset.

References

[1] Agencija za statistiku BiH. (n.d.), retrieved from: http://www.bhas.ba . [accessed July 18, 2018.]

[2] Listiani, M. (2009). Support vector regression analysis for price prediction in a car leasing application (Doctoral dissertation, Master thesis, TU Hamburg-Harburg).

[3] Richardson, M. S. (2009). Determinants of used car resale value. Retrieved from: https://digitalcc.coloradocollege.edu/islandora /object /coccc%3A1346 [accessed: August 1, 2018.]

[4] 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.

[5] Du, J., Xie, L., & Schroeder, S. (2009). Practice Prize Paper—PIN Optimal Distribution of Auction Vehicles System: Applying Price Forecasting, Elasticity Estimation, and Genetic Algorithms to Used-Vehicle Distribution. Marketing Science, 28(4), 637-644.

[6] Gongqi, S., Yansong, W., & Qiang, Z. (2011, January). New Model for Residual Value Prediction of the Used Car Based on BP Neural Network and Nonlinear Curve Fit. In Measuring Technology and Mechatronics Automation (ICMTMA), 2011 Third International Conference on (Vol. 2, pp. 682-685). IEEE.

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

[8] Noor, K., & Jan, S. (2017). Vehicle Price Prediction System using Machine Learning Techniques. International Journal of Computer Applications, 167(9), 27-31.

[9] Auto pijaca BiH. (n.d.), Retrieved from: https://www.autopijaca.ba. [accessed August 10, 2018].

[10] Weka 3 - Data Mining with Open Source Machine Learning Software in Java. (n.d.), Retrieved from: https://www.cs.waikato.ac.nz/ml/weka/. [August 04, 2018].

[11] Ho, T. K. (1995, August). Random decision forests. In Document analysis and recognition, 1995., proceedings of the third international conference on (Vol. 1, pp. 278-282). IEEE.

[12] Russell, S. (2015). Artificial Intelligence: A Modern Approach (3rd edition). PE.

[13] Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. (2001). Support vector clustering. Journal of machine learning research, 2(Dec), 125-137.

[14] Aizerman, M. A. (1964). Theoretical foundations of the potential function method in pattern recognition learning. Automation and remote control, 25, 821- 837.

[15]3.2.4.3.1.sklearn.ensemble.RandomForest Classifier — scikit-learn 0.19.2 documentation. (n.d.). Retrieved from:http://scikit learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html accessed: August 30, 2018].

[16] Used cars database. (n.d.) Retrieved from: https://www.kaggle.com/orgesleka/used-cars database. [accessed: June 04, 2018].

[17] OLX. (n.d.), Retrieved from: https://olx.ba. [accessed August 05,2018]