Deccan Education Society's

NAVINCHANDRA MEHTA INSTITUTE OF TECHNOLOGY AND DEVELOPMENT

NAAC Accredited "B++"

"**Car price prediction**"

SUBMITTED BY

C21027 Aditya Deshpande
C21002 Kunal Ambre
C21031 Tejas Gaikwad

Academic year

[2022-23]

Under the guidance of:

Associate Director Mr. Pratik Desai

Submitted to University of Mumbai
in partial fulfillment of the requirements for qualifying
MASTER OF COMPUTER APPLICATION
Examination

Deccan Education Society's

NAVINCHANDRA MEHTA INSTITUTE OF TECHNOLOGY AND DEVELOPMENT

PROJECT CERTIFICATE

This is to certify that the Project done at____by

Mr./Ms.____

(Seat No.________________) in partial fulfillment for MCA Degree Examination has been found satisfactory. This report had not been submitted for any other examination and does not form part of any other course undergone by the candidate.

Internal Guide

Director

EXAMINED BY

EXTERNAL EXAMINER

………………………………

INDEX

1. Introduction

Car price prediction is somehow interesting and popular problem. As per information that was gotten from the Agency for Statistics of BiH, 921.456 vehicles were registered in 2014 from which 84% of them are cars for personal usage [1]. This number is increased by 2.7% since 2013 and it is likely that this trend will continue, and the number of cars will DOI: 10.18421/TEM81-16
https://dx.doi.org/10.18421/TEM81-16
Corresponding author: Enis Gegic, International Burch University, Sarajevo, Bosnia and Herzegovina
Email: enis.gegic@ibu.edu.ba
Received: 29 March 2018. Accepted: 28 January 2019. Published: 27 February 2019.

The article is published with Open Access at www.temjournal.com increase in future. This adds additional significance to the problem of the car price prediction. Accurate car price prediction involves expert knowledge, because price usually depends on many distinctive features and factors. Typically, most significant ones are brand and model, age, horsepower and mileage. The fuel type used in the car as well as fuel consumption per mile highly affect price of a car due to a frequent changes in the price of a fuel. Different features like exterior color, door number, type of transmission, dimensions, safety, air condition, interior, whether it has navigation or not will also influence the car price. In this paper, we applied different methods and techniques in order to achieve higher precision of the used car price prediction. This paper is organized in the following manner: Section II contains related work in the field of price prediction of used cars. In section III, the research methodology of our study is explain. Section IV elaborates various machine learning algorithms and examine their respective performances to predict the price of the used cars. Finally, in section V, a conclusion of our work are given, together with the future works plan.

In this fast world, you don't have your own personal mode of transportation sort of an automobile, life will become even additional agitated. The public choose to obtain their automobile as a result of its convenience to commute between places, permits movement with an outsized cluster of individuals with fuel potency, and safe mode of transport. The used automobile marketplace is witnessing a boom in India, with the decision for luxurious vehicles sometimes increasing. Till a couple of years, owning a luxury automobile won't be a dream for varied shoppers, as a result of money hurdles, however, this is often bit by bit dynamic as shoppers can simply obtain used luxury vehicles. Machine Learning provides numerous ways through that it's easier to predict the worth of an automobile, by the previous information that is obtainable We've enforced the model exploitation supervised Learning techniques of Machine Learning, which is outlined by its use of labeled information sets to coach algorithms to classify data or predict outcomes accurately. As the input file is fed into the model, it adjusts its weights till the model Journal of Emerging Technologies and Innovative Research has been fitted fittingly, which happens as a part of the cross-validation method. If there is also further transparency within the marketplace and fewer intermediaries, the seller ought to get the next value for a vehicle and therefore the shopper ought to get one at a lower fee as margins get reduced on every facet.

2. Objectives :

You as a Data scientist are required to apply some data science techniques for the price of cars with the available independent variables. That should help the management to understand how exactly the prices vary with the independent variables. They can accordingly manipulate the design of the cars, the business strategy etc. to meet certain price levels.

3. TECHNOLOGY USED

Python :

What is Python? What are the benefits of using Python

Python is a high-level, interpreted, general-purpose programming language. Being a general-purpose language, it can be used to build almost any type of application with the right tools/libraries. Additionally, python supports objects, modules, threads, exception-handling, and automatic memory management which help in modelling real-world problems and building applications to solve these problems.

Benefits of using Python:

- Python is a general-purpose programming language that has a simple, easy-to-learn syntax that emphasizes readability and therefore reduces the cost of program maintenance. Moreover, the language is capable of scripting, is completely open-source, and supports third-party packages encouraging modularity and code reuse.

- Its high-level data structures, combined with dynamic typing and dynamic binding, attract a huge community of developers for Rapid Application Development and deployment.

Machine learning :

What is machine learning?

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

IBM has a rich history with machine learning. One of its own, Arthur Samuel, is credited for coining the term, “machine learning” with his research (PDF, 481 KB) (link resides outside IBM) around the game of checkers. Robert Nealey, the self-proclaimed checkers master, played the game on an IBM 7094 computer in 1962, and he lost to the computer. Compared to what can be done today, this feat seems trivial, but it’s considered a major milestone in the field of artificial intelligence.

Over the last couple of decades, the technological advances in storage and processing power have enabled some innovative products based on machine learning, such as Netflix’s recommendation engine and self-driving cars.

Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase. They will be required to help identify the most relevant business questions and the data to answer them.

Machine learning algorithms are typically created using frameworks that accelerate solution development, such as TensorFlow and PyTorch.

3 types of machine learning

Machine learning involves showing a large volume of data to a machine so that it can learn and make predictions, find patterns, or classify data. The three machine learning types are supervised, unsupervised, and reinforcement learning.

Supervised learning

Gartner, a business consulting firm, predicts that supervised learning will remain the most utilized machine learning among enterprise information technology leaders in 2022 [2]. This type of machine learning feeds historical input and output data in machine learning algorithms, with processing in between each input/output pair that allows the algorithm to shift the model to create outputs as closely aligned with the desired result as possible. Common algorithms used during supervised learning include neural networks, decision trees, linear regression, and support vector machines.

This machine learning type got its name because the machine is “supervised” while it's learning, which means that you’re feeding the algorithm information to help it learn. The outcome you provide the machine is labeled data, and the rest of the information you give is used as input features.

For example, if you were trying to learn about the relationships between loan defaults and borrower information, you might provide the machine with 500 cases of customers who defaulted on their loans and another 500 who didn't. The labeled data “supervises” the machine to figure out the information you're looking for.

Supervised learning is effective for a variety of business purposes, including sales forecasting, inventory optimization, and fraud detection. Some examples of use cases include:

- Predicting real estate prices
- Classifying whether bank transactions are fraudulent or not
- Finding disease risk factors
- Determining whether loan applicants are low-risk or high-risk
- Predicting the failure of industrial equipment's mechanical parts
-

Unsupervised learning

While supervised learning requires users to help the machine learn, unsupervised learning doesn't use the same labeled training sets and data. Instead, the machine looks for less obvious patterns in the data. This machine learning type is very helpful when you need to identify patterns and use data to make decisions. Common algorithms used in unsupervised learning include Hidden Markov models, k-means, hierarchical clustering, and Gaussian mixture models.

Using the example from supervised learning, let's say you didn't know which customers did or didn't default on loans. Instead, you'd provide the machine with borrower information and it would look for patterns between borrowers before grouping them into several clusters.

This type of machine learning is widely used to create predictive models. Common applications also include clustering, which creates a model that groups objects together based on specific properties, and association, which identifies the rules existing between the clusters. A few example use cases include:

- Creating customer groups based on purchase behavior
- Grouping inventory according to sales and/or manufacturing metrics
- Pinpointing associations in customer data (for example, customers who buy a specific style of handbag might be interested in a specific style of shoe)

Reinforcement learning

Reinforcement learning is the closest machine learning type to how humans learn. The algorithm or agent used learns by interacting with its environment and getting a positive or negative reward. Common algorithms include temporal difference, deep adversarial networks, and Q-learning.

Going back to the bank loan customer example, you might use a reinforcement learning algorithm to look at customer information. If the algorithm classifies them as high-risk and they default, the algorithm gets a positive reward. If they don't default, the algorithm gets a negative reward. In the end, both instances help the machine learn by understanding both the problem and environment better.

Gartner notes that most ML platforms don't have reinforcement learning capabilities because it requires higher computing power than most organizations have [2]. Reinforcement learning is applicable in areas capable of being fully simulated that are either stationary or have large volumes of relevant data. Because this type of machine learning requires less management than supervised learning, it’s viewed as easier to work with dealing with unlabeled data sets. Practical applications for this type of machine learning are still emerging. Some examples of uses include:

- Teaching cars to park themselves and drive autonomously
- Dynamically controlling traffic lights to reduce traffic jams
- Training robots to learn policies using raw video images as input that they can use to replicate the actions they see

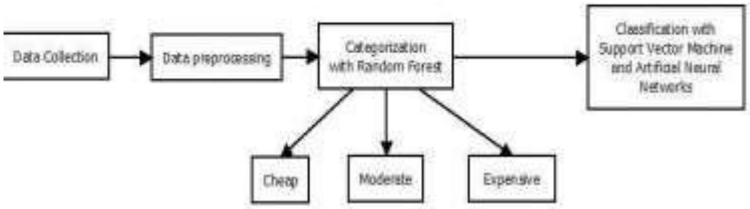
4. Related Work

Predicting price of a used cars has been studied extensively in various researches. Listian discussed, in her paper written for Master thesis [2], that regression model that was built using Support Vector Machines (SVM) can predict the price of a car that has been leased with better precision than multivariate regression or some simple multiple regression. This is on the grounds that Support Vector Machine (SVM) is better in dealing with datasets with more dimensions and

it is less prone to overfitting and underfitting. The weakness of this research is that a change of simple regression with more advanced SVM regression was not shown in basic indicators like mean, variance or standard deviation. Another approach was given by Richardson in his thesis work [3]. His theory was that car producers produce more durable cars. Richardson applied multiple regression analysis and demonstrated that hybrid cars retain their value for longer time than
traditional cars. This has roots in environmental concerns about the climate and it gives higher fuel efficiency. Wu et al. [4] conducted car price prediction study, by using neuro-fuzzy knowledge-based system. They took into consideration the following attributes: brand, year of production and type of engine. Their prediction model produced similar results as the simple regression model. Moreover, they made an expert system named ODAV (Optimal Distribution of Auction Vehicles) as there is a high demand for selling the cars at the end of the leasing year by car dealers. This system gives insights into the best prices for vehicles, as well as the location where the best price can be gained. Regression model based on k-nearest neighbor machine learning algorithm was used to predict the price of a car. This system has a tendency to be exceptionally successful since more than two million vehicles were exchanged through it [5]. Gonggie [6] proposed a model that is built using ANN (Artificial Neural Networks) for the price prediction of a used car. He considered several attributes: miles passed, estimated car life and brand. The proposed model was built so it could deal with nonlinear relations in data which was not the case with previous models that were utilizing the simple linear regression techniques. The non-linear model was able to predict prices of cars with better precision than other linear models. Furthermore, Pudaruth [7] applied various machine learning algorithms, namely: k-nearest neighbors, multiple linear regression analysis, decision trees and naïve bayes for car price prediction in Mauritius. The dataset used to create a prediction model was collected manually from local newspapers in period less than one month, as time can have a noticeable impact on price of the car. He studied the following attributes: brand, model, cubic capacity, mileage in kilometers, production year, exterior color, transmission type and price. However, the author found out that Naive Bayes and Decision Tree were unable to predict and classify numeric values. Additionally, limited number of dataset instances could not give high classification performances, i.e. accuracies less than 70%. Noor and Jan [8] build a model for car price prediction by using multiple linear regression. The dataset was created during the two-months period and included the following features: price, cubic capacity, exterior color, date when the ad was posted, number of ad views, power steering, mileage in kilometer, rims type, type of transmission, engine type, city, registered city, model, version, make and model year. After applying feature selection, the authors considered only engine type, price, model year and model as input features. With the given setup authors were able to achieve prediction accuracy of 98%. In the related work shown above, authors proposed prediction model based on the single machine learning algorithm. However, it is noticeable that single machine learning algorithm approach did not give remarkable prediction results and could be enhanced by assembling

5. Materials and Methods

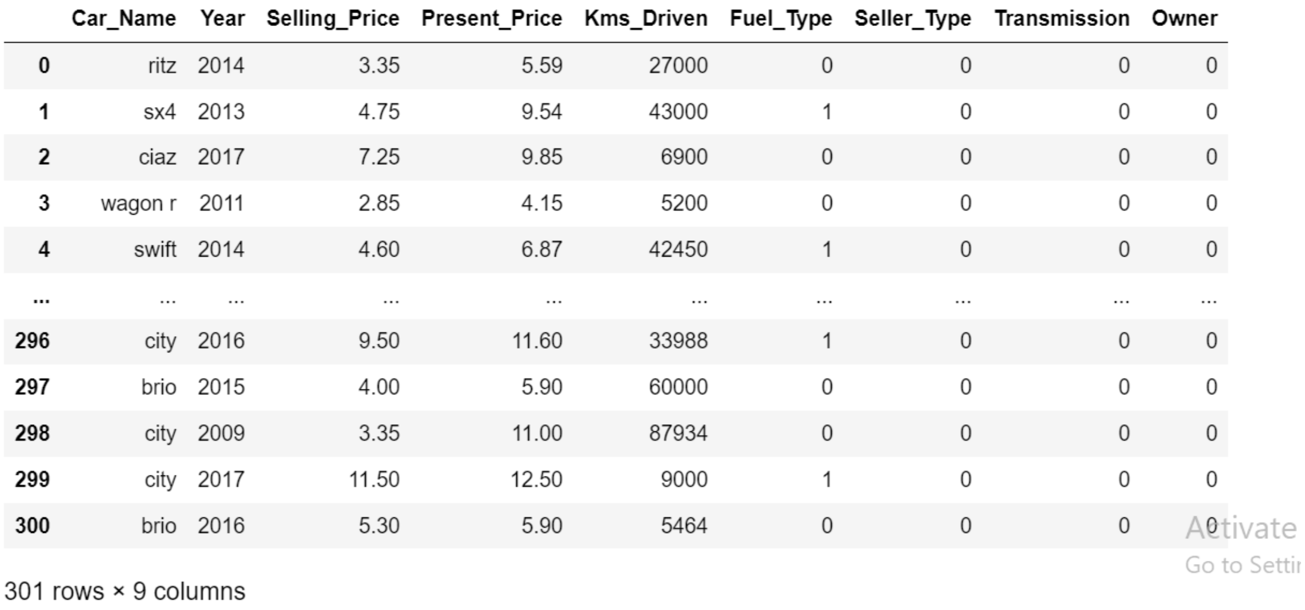
Approach for car price prediction proposed in this paper is composed of several steps, shown in Fig. 1.various machine learning methods in an ensemble


Data Collection
Data preprocessing
Categorization with Random Forest
Classification with Support Vector Machine and Artificial Neural Networks
Cheap
Moderate
Expensive

Data is collected from a local web portal for selling and buying cars autopijaca.ba [9], during winter season, as time interval itself has high impact on the price of the cars in Bosnia and Herzegovina. The following attributes were captured for each car: brand, model, car condition, fuel, year of manufacturing, power in kilowatts, transmission type, millage, color, city, state, number of doors, four wheel drive (yes/no), damaged (yes/no), navigation (yes/no), leather seats (yes/no), alarm (yes/no), aluminum rims (yes/no), digital air condition (yes/no), parking sensors (yes/no), xenon lights (yes/no), remote unlock (yes/no), electric rear mirrors (yes/no), seat heat (yes/no), panorama roof (yes/no), cruise control (yes/no), abs (yes/no), esp (yes/no), asr (yes/no) and price expressed in BAM (Bosnian Mark). 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. 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 preprocessing step was applied. Many of the attributes were sparse and they do not TEM Journal. Volume 8, Issue 1, Pages 113-118, ISSN 2217-8309, DOI: 10.18421/TEM81-16, February 2019.
contain useful information for prediction. Hence, it is decided to remove them from the dataset. The attributes “state”, “city”, and “damaged” were completely removed.
The color of the cars was normalized into fixed set of 15 different colors. Continuous attributes such as “millage”, “year of manufacturing”, “power in kilowatts” and “price” are converted into categorical values using predefined cluster intervals. The millage is converted into five distinct categories, the year ofmanufacturing has been converted into seven categories and the power in kilowatts is converted into eleven categories. The price attribute has been categorized into 15 distinct categories based on price range. These categories are shown in Table 2 and similar principle was applied to other attributes. This data transformation process converted regression prediction machine learning problem into classification problem.

6. Implementation/Source code:

Dataset :



	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	0	0	0	0
1	sx4	2013	4.75	9.54	43000	1	0	0	0
2	ciaz	2017	7.25	9.85	6900	0	0	0	0
3	wagon r	2011	2.85	4.15	5200	0	0	0	0
4	swift	2014	4.60	6.87	42450	1	0	0	0
...	...	...	...	...	...	...	...	...	...
296	city	2016	9.50	11.60	33988	1	0	0	0
297	brio	2015	4.00	5.90	60000	0	0	0	0
298	city	2009	3.35	11.00	87934	0	0	0	0
299	city	2017	11.50	12.50	9000	1	0	0	0
300	brio	2016	5.30	5.90	5464	0	0	0	0

301 rows × 9 columns

The collected raw data set contains 301 samples. Since data is collected using web scraper, there are many samples that have only few attributes. In order to clean these samples, PHP scriptthat is reading scraped data from database, perform cleaning and saves the cleaned samples into CSV file. The CSV file is later used to load data into, software for building machine learning models [10]. After cleanup process, the data set has been reduced samples.

Libraries which used in module :

Scipy :

SciPy (Scientific Python) is another free and open-source Python library for data science that is extensively used for high-level computations. SciPy has around 19,000 comments on GitHub and an active community of about 600 contributors. It’s extensively used for scientific and technical computations, because it extends NumPy and provides many user-friendly and efficient routines for scientific calculations.

Numpy :

NumPy (Numerical Python) is the fundamental package for numerical computation in Python; it contains a powerful N-dimensional array object. It has around 18,000 comments on GitHub and an active community of 700 contributors. It’s a general-purpose array-processing package that provides high-performance multidimensional objects called arrays and tools for working with them. NumPy also addresses the slowness problem partly by providing these multidimensional arrays as well as providing functions and operators that operate efficiently on these arrays.

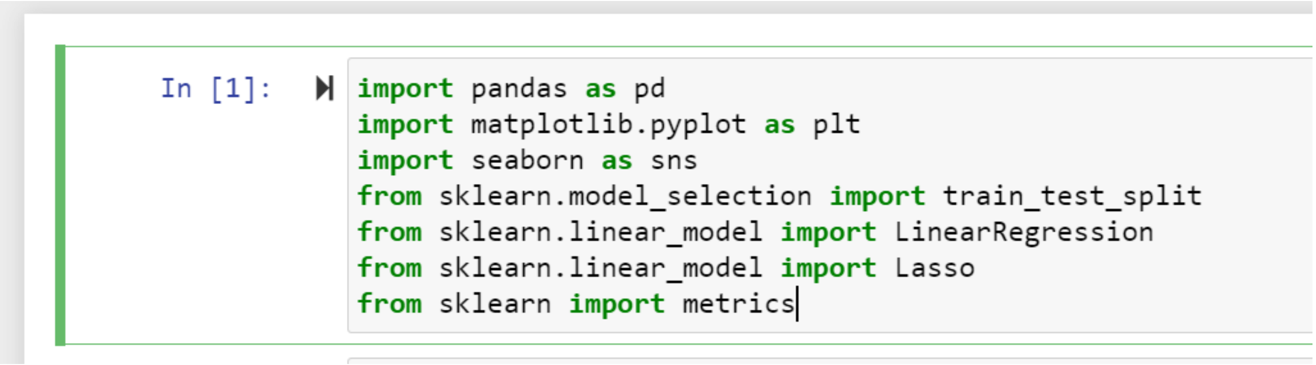
Pandas

Pandas is a must in the data science life cycle. It is the most popular and widely used Python library for data science, along with NumPy in matplotlib. With around 17,00 comments on GitHub and an active community of 1,200 contributors, it is heavily used for data analysis and cleaning. Pandas provides fast, flexible data structures, such as data frame CDs, which are designed to work with structured data very easily and intuitively.

Matplotlib

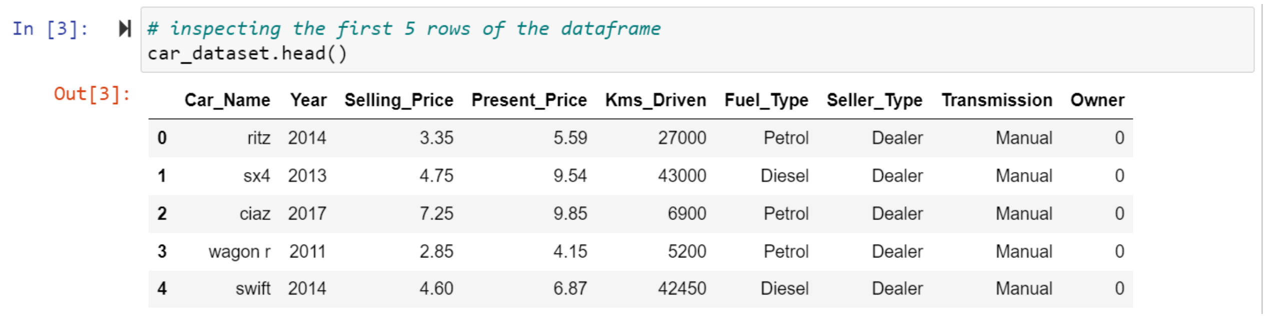
Matplotlib has powerful yet beautiful visualizations. It’s a plotting library for Python with around 26,000 comments on GitHub and a very vibrant community of about 700 contributors. Because of the graphs and plots that it produces, it’s extensively used for data visualization. It also provides an object-oriented API, which can be used to embed those plots into applications.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn import metrics
```



```
In [1]:  import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns
         from sklearn.model_selection import train_test_split
         from sklearn.linear_model import LinearRegression
         from sklearn.linear_model import Lasso
         from sklearn import metrics
```

Data set head (Top 5 data) :



```
In [3]:   # inspecting the first 5 rows of the dataframe
          car_dataset.head()
```

Out[3]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

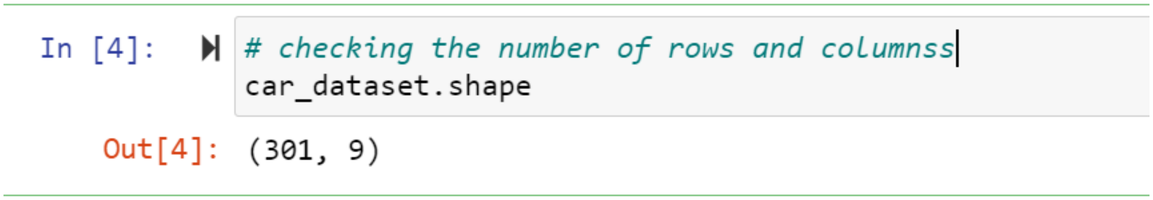
Command :

car_dataset.head()

these command use for print the top 5 rows of the dataset.

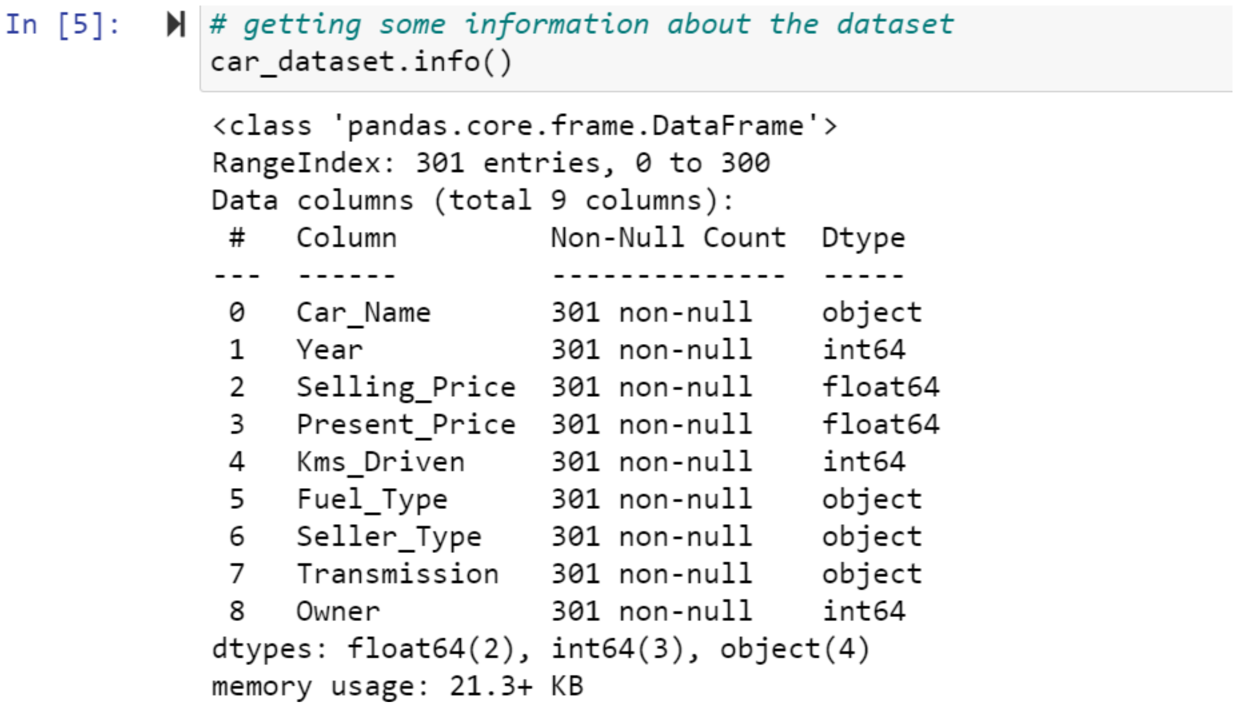
# data shape

car_dataset.shape ()

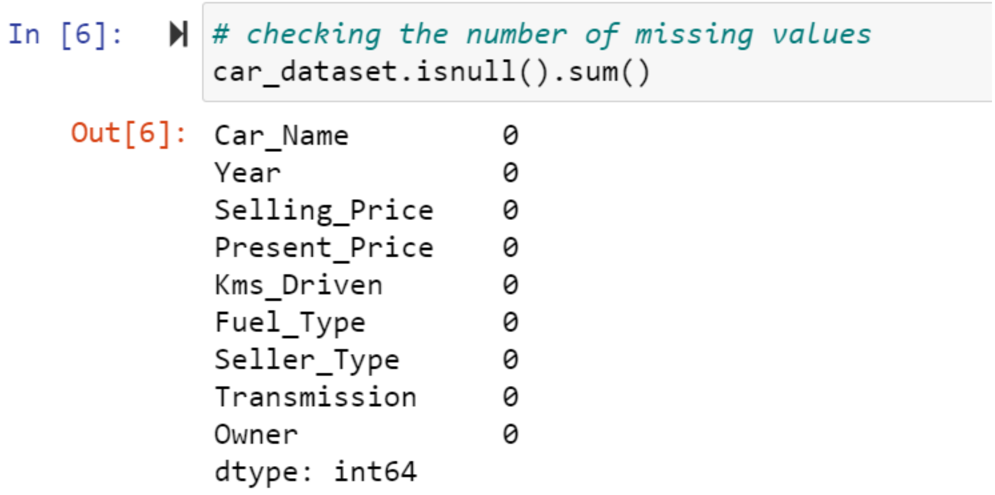


```
In [4]:   # checking the number of rows and columnss
          car_dataset.shape

Out[4]:   (301, 9)
```

# In the data set is their any null value is available or not (Preprocessing)



```
In [5]:  # getting some information about the dataset
         car_dataset.info()

         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 301 entries, 0 to 300
         Data columns (total 9 columns):
          #   Column         Non-Null Count  Dtype  
         ---  ------         --------------  -----  
          0   Car_Name       301 non-null    object 
          1   Year           301 non-null    int64  
          2   Selling_Price  301 non-null    float64
          3   Present_Price  301 non-null    float64
          4   Kms_Driven     301 non-null    int64  
          5   Fuel_Type      301 non-null    object 
          6   Seller_Type    301 non-null    object 
          7   Transmission   301 non-null    object 
          8   Owner          301 non-null    int64  
         dtypes: float64(2), int64(3), object(4)
         memory usage: 21.3+ KB
```

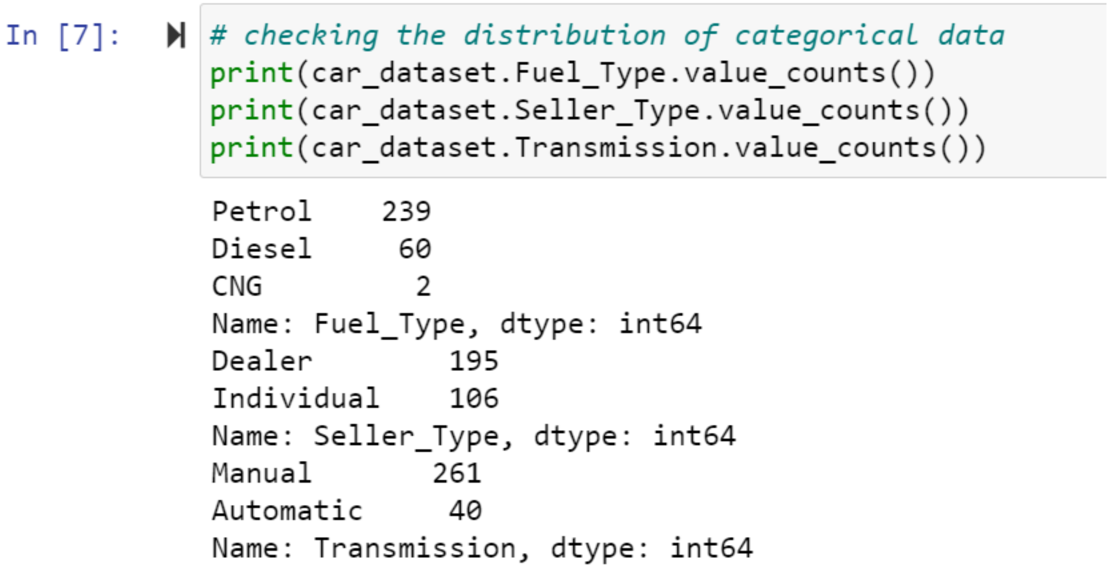


```
In [6]:  # checking the number of missing values
         car_dataset.isnull().sum()

Out[6]:  Car_Name         0
         Year             0
         Selling_Price    0
         Present_Price    0
         Kms_Driven       0
         Fuel_Type        0
         Seller_Type      0
         Transmission     0
         Owner            0
         dtype: int64
```

# checking the distribution of categorical data

Categorical data is a type of data that is used to group information with similar characteristics, while numerical data is a type of data that expresses information in the form of numbers.

Example of categorical data: **gender**



```
In [7]:  # checking the distribution of categorical data
         print(car_dataset.Fuel_Type.value_counts())
         print(car_dataset.Seller_Type.value_counts())
         print(car_dataset.Transmission.value_counts())
```

```
Petrol    239
Diesel     60
CNG         2
Name: Fuel_Type, dtype: int64
Dealer        195
Individual    106
Name: Seller_Type, dtype: int64
Manual       261
Automatic     40
Name: Transmission, dtype: int64
```

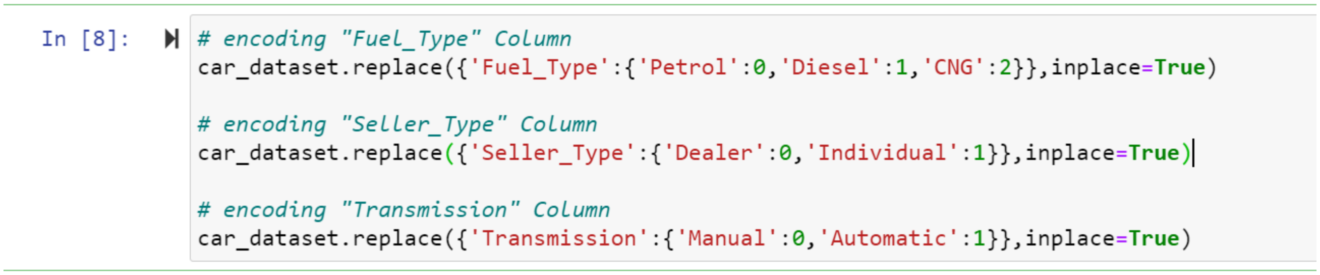
# encoding "Fuel_Type" Column :

Label Encoding: Label encoding algorithm is quite simple and it considers an order for encoding, Hence can be used for encoding ordinal data.

One-Hot Encoding: To overcome the Disadvantage of Label Encoding as it considers some hierarchy in the columns which can be misleading to nominal features present in the data. we can use the One-Hot Encoding strategy.
One-hot encoding is processed in 2 steps:

1. Splitting of categories into different columns.
2. Put ‘0 for others and ‘1’ as an indicator for the appropriate column.

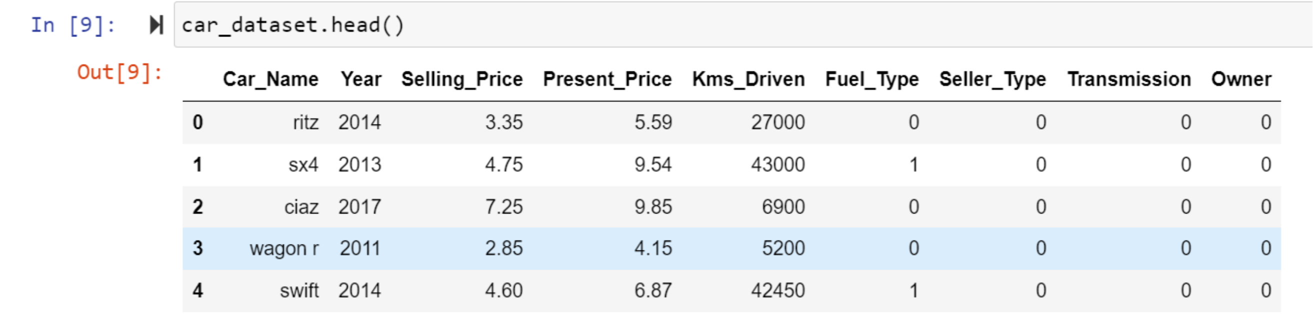
Frequency Encoding: We can also encode considering the frequency distribution. This method can be effective at times for nominal features.



In [8]:

```python
# encoding "Fuel_Type" Column
car_dataset.replace({'Fuel_Type':{'Petrol':0,'Diesel':1,'CNG':2}},inplace=True)

# encoding "Seller_Type" Column
car_dataset.replace({'Seller_Type':{'Dealer':0,'Individual':1}},inplace=True)

# encoding "Transmission" Column
car_dataset.replace({'Transmission':{'Manual':0,'Automatic':1}},inplace=True)
```

After Encoding :



In [9]:

```python
car_dataset.head()
```

Out[9]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	0	0	0	0
1	sx4	2013	4.75	9.54	43000	1	0	0	0
2	ciaz	2017	7.25	9.85	6900	0	0	0	0
3	wagon r	2011	2.85	4.15	5200	0	0	0	0
4	swift	2014	4.60	6.87	42450	1	0	0	0

```
# Splitting the data and Target

X = car_dataset.drop(['Car_Name','Selling_Price'],axis=1)
Y = car_dataset['Selling_Price']
```

Train test split is a model validation procedure that allows you to simulate how a model would perform on new/unseen data. Here is how the procedure works:

1. ARRANGE THE DATA

Make sure your data is arranged into a format acceptable for train test split. In scikit-learn, this consists of separating your full data set into “Features” and “Target.”

2. SPLIT THE DATA

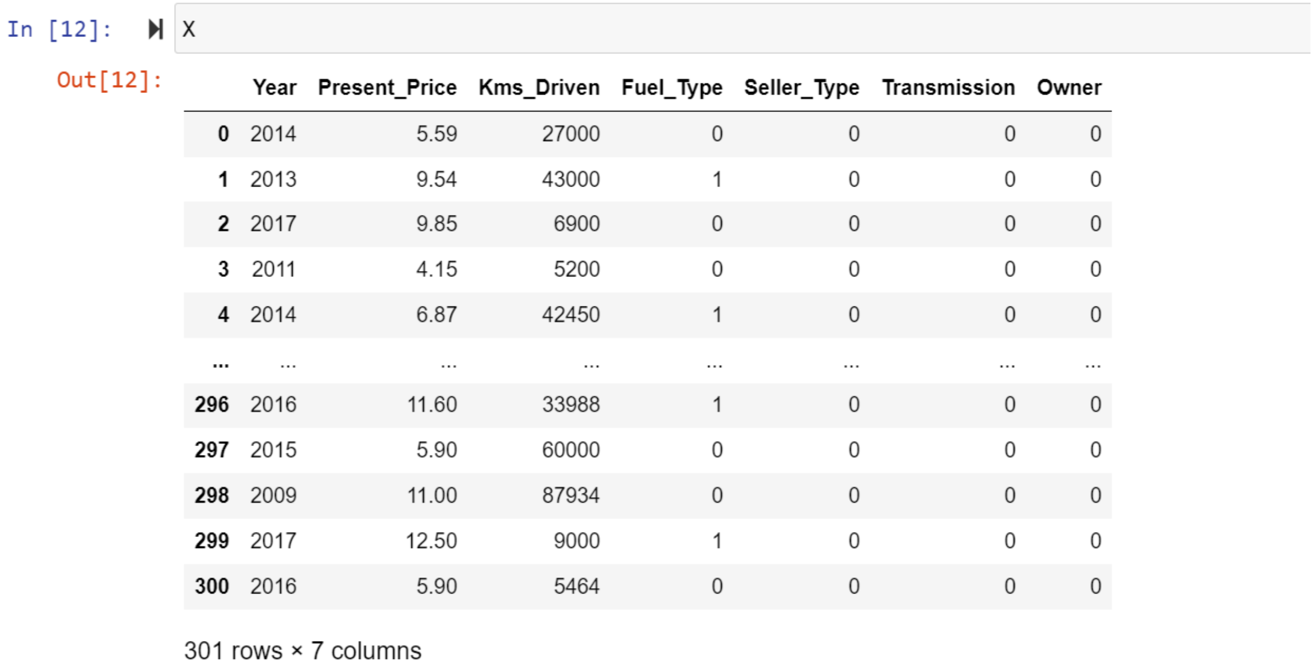
Split the data set into two pieces — a training set and a testing set. This consists of random sampling without replacement about 75 percent of the rows (you can vary this) and putting them into your training set. The remaining 25 percent is put into your test set. Note that the colors in “Features” and “Target” indicate where their data will go (“X_train,” “X_test,” “y_train,” “y_test”) for a particular train test split.

3. TRAIN THE MODEL

Train the model on the training set. This is “X_train” and “y_train” in the image.

4. TEST THE MODEL

Test the model on the testing set (“X_test” and “y_test” in the image) and evaluate the performance.



```
In [12]:    X
```

Out[12]:

	Year	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	2014	5.59	27000	0	0	0	0
1	2013	9.54	43000	1	0	0	0
2	2017	9.85	6900	0	0	0	0
3	2011	4.15	5200	0	0	0	0
4	2014	6.87	42450	1	0	0	0
...	...	...	...	...	...	...	...
296	2016	11.60	33988	1	0	0	0
297	2015	5.90	60000	0	0	0	0
298	2009	11.00	87934	0	0	0	0
299	2017	12.50	9000	1	0	0	0
300	2016	5.90	5464	0	0	0	0

301 rows × 7 columns

After encoding values become 0 & 1 & 2.

# Train , Test , Split

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.1, random_state=2)
```

# Linear Regression

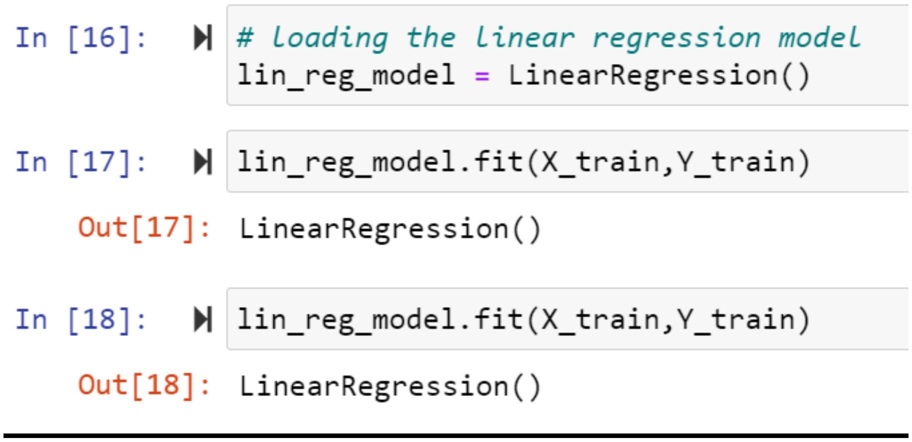
Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

Types of Linear Regression

Linear regression can be further divided into two types of the algorithm:

- **SimpleL inear Regression:**
  If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.
- **Multiple Linear regression:**
  If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.



```
In [16]:  # Loading the linear regression model
           lin_reg_model = LinearRegression()

In [17]:  lin_reg_model.fit(X_train,Y_train)

Out[17]:  LinearRegression()

In [18]:  lin_reg_model.fit(X_train,Y_train)

Out[18]:  LinearRegression()
```

Model Evaluation

The train/test/validation split

The most important thing you can do to properly evaluate your model is to not train the model on the entire dataset. I repeat: do not train the model on the entire dataset. I talked about this in my post on preparing data for a machine learning model and I'll mention it again now because it's that important. A typical train/test split would be to use 70% of the data for training and 30% of the data for testing.

As I discussed previously, it's important to use new data when evaluating our model to prevent the likelihood of overfitting to the training set. However, sometimes it's useful to evaluate our model as we're building it to find that best parameters of a model - but we can't use the test set for this evaluation or else we'll end up selecting the parameters that perform best on the test data but maybe not the parameters that generalize best. To evaluate the model while still building and tuning the model, we create a third subset of the data known as the validation set. A typical train/test/validation split would be to use 60% of the data for training, 20% of the data for validation, and 20% of the data for testing.

I'll also note that it's very important to shuffle the data before making these splits so that each split has an accurate representation of the dataset.

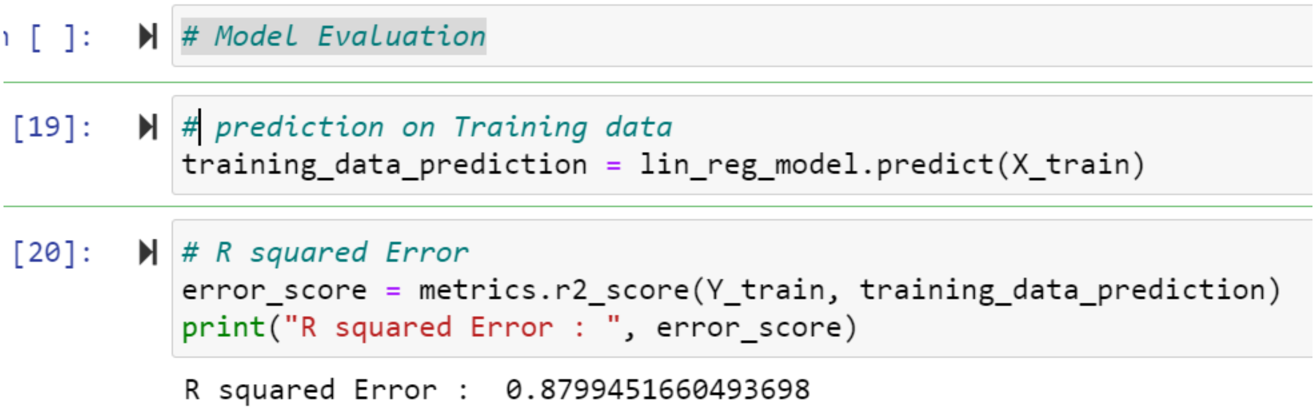
Metrics

In this session, I'll discuss common metrics used to evaluate models.

Classification metrics

When performing classification predictions, there's four types of outcomes that could occur.

- **True positives** are when you predict an observation belongs to a class and it actually does belong to that class.
- **True negatives** are when you predict an observation does not belong to a class and it actually does not belong to that class.
- **False positives** occur when you predict an observation belongs to a class when in reality it does not.
- **False negatives** occur when you predict an observation does not belong to a class when in fact it does.



```
# Model Evaluation
```

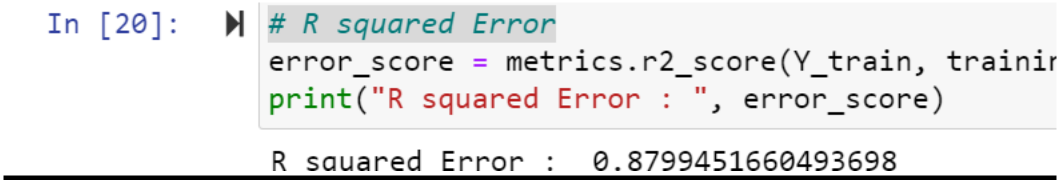
```
# prediction on Training data
training_data_prediction = lin_reg_model.predict(X_train)
```

```
# R squared Error
error_score = metrics.r2_score(Y_train, training_data_prediction)
print("R squared Error : ", error_score)
```

```
R squared Error :  0.8799451660493698
```

R squared Error

R-Squared (R² or the coefficient of determination) is **a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable**. In other words, r-squared shows how well the data fit the regression model (the goodness of fit).



In [20]:

```
# R squared Error
error_score = metrics.r2_score(Y_train, trainir
print("R squared Error : ", error_score)
```

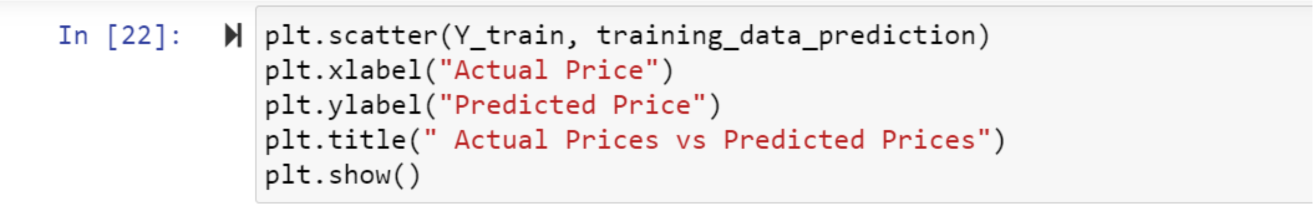
```
R squared Error :  0.8799451660493698
```

# Visualize the actual prices and Predicted prices :

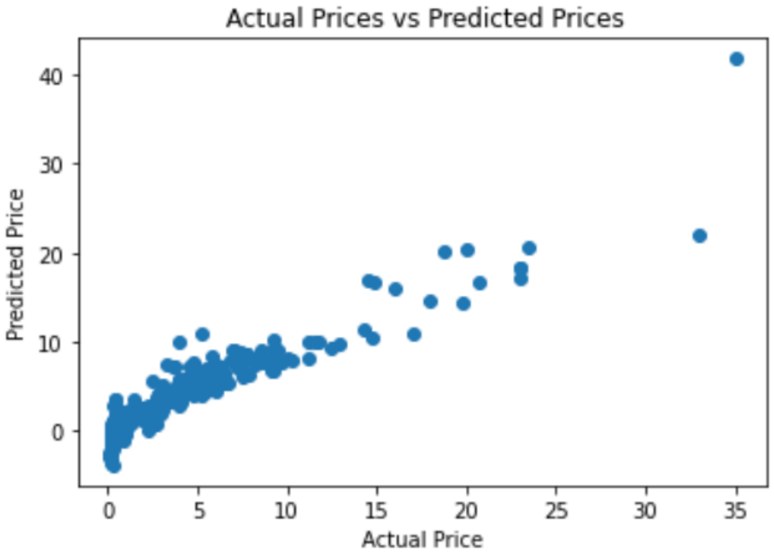
In today's world, a lot of data is being generated on a daily basis. And sometimes to analyze this data for certain trends, patterns may become difficult if the data is in its raw format. To overcome this data visualization comes into play. Data visualization provides a good, organized pictorial representation of the data which makes it easier to understand, observe, analyze.

Python provides various libraries that come with different features for visualizing data. All these libraries come with different features and can support various types of graphs. In this tutorial, we will be discussing four such libraries.

- Matplotlib
- Seaborn
- Bokeh
- Plot



```
In [22]:  plt.scatter(Y_train, training_data_prediction)
          plt.xlabel("Actual Price")
          plt.ylabel("Predicted Price")
          plt.title(" Actual Prices vs Predicted Prices")
          plt.show()
```



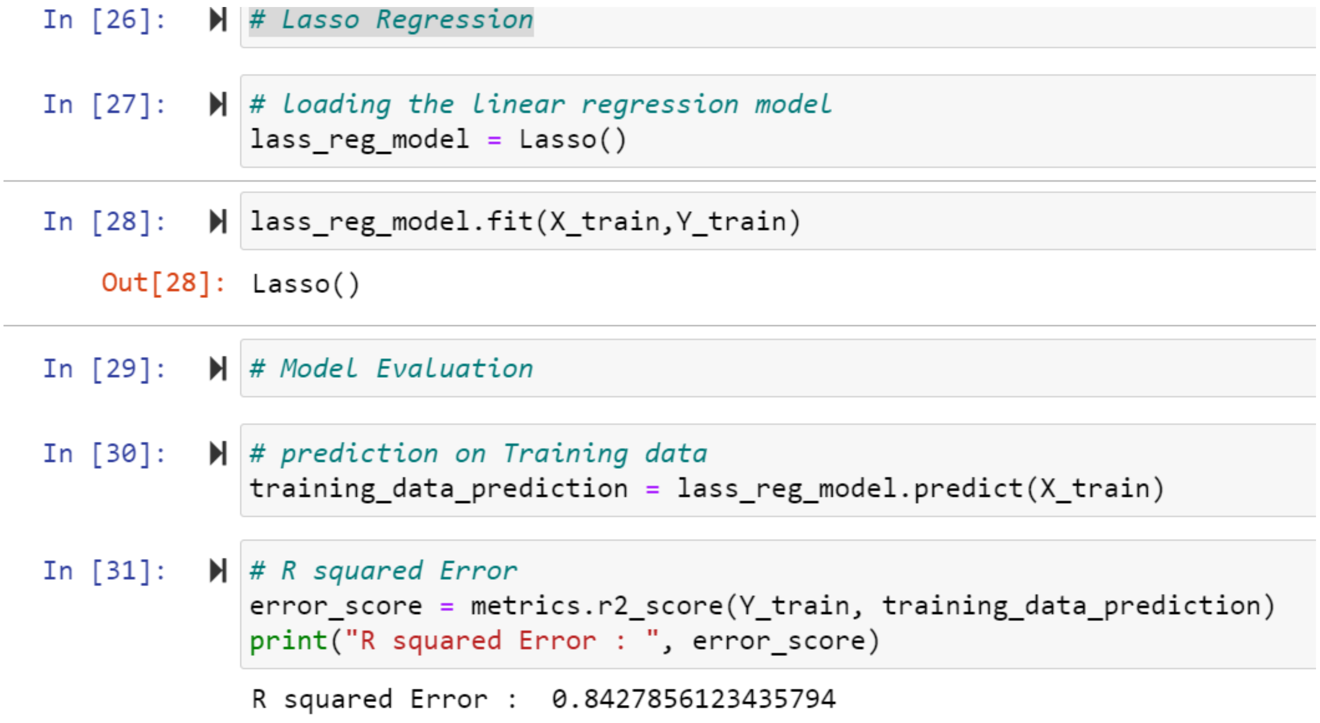
Lasso Regression

Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

Lasso Regression uses L1 regularization technique (will be discussed later in this article). It is used when we have more features because it automatically performs feature selection.

Lasso Meaning

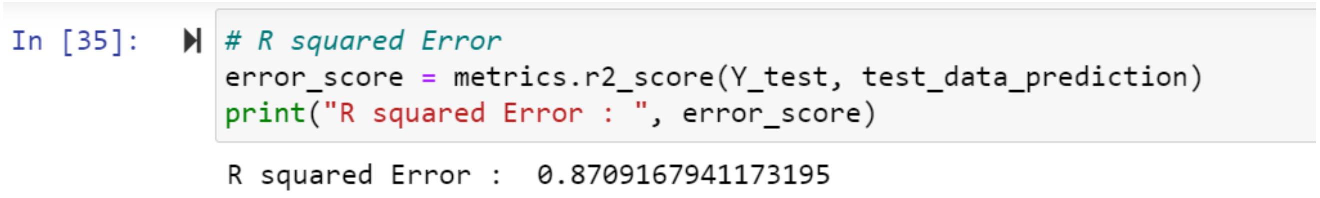
The word “LASSO” stands for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator. It is a statistical formula for the regularisation of data models and feature selection.



```
In [26]:  # Lasso Regression

In [27]:  # loading the linear regression model
          lass_reg_model = Lasso()

In [28]:  lass_reg_model.fit(X_train,Y_train)

Out[28]:  Lasso()

In [29]:  # Model Evaluation

In [30]:  # prediction on Training data
          training_data_prediction = lass_reg_model.predict(X_train)

In [31]:  # R squared Error
          error_score = metrics.r2_score(Y_train, training_data_prediction)
          print("R squared Error : ", error_score)

          R squared Error :  0.8427856123435794
```

R – Squared error in lasso regression

Lasso regression, or the Least Absolute Shrinkage and Selection Operator, is also a modification of linear regression. In lasso, the loss function is modified to minimize the complexity of the model by limiting the sum of the absolute values of the model coefficients (also called the `l1-norm`).
The loss function for lasso regression can be expressed as below:
`Loss function = OLS + alpha * summation (absolute values of the magnitude of the coefficients)`
In the above function, `alpha` is the penalty parameter we need to select. Using an `l1-norm` constraint forces some weight values to zero to allow other coefficients to take non-zero values.



```
In [35]:  # R squared Error
          error_score = metrics.r2_score(Y_test, test_data_prediction)
          print("R squared Error : ", error_score)

          R squared Error :  0.8709167941173195
```

7. 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.

- In the Proposed solution the Web Application " TypoGrapher web application" will use knowledge representation and semantic web technology in the form of the hand written Ontology APIs to produce the customized fonts for the users.

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 singlemachine 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 92.38%. 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.

8 .Referance :

[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.

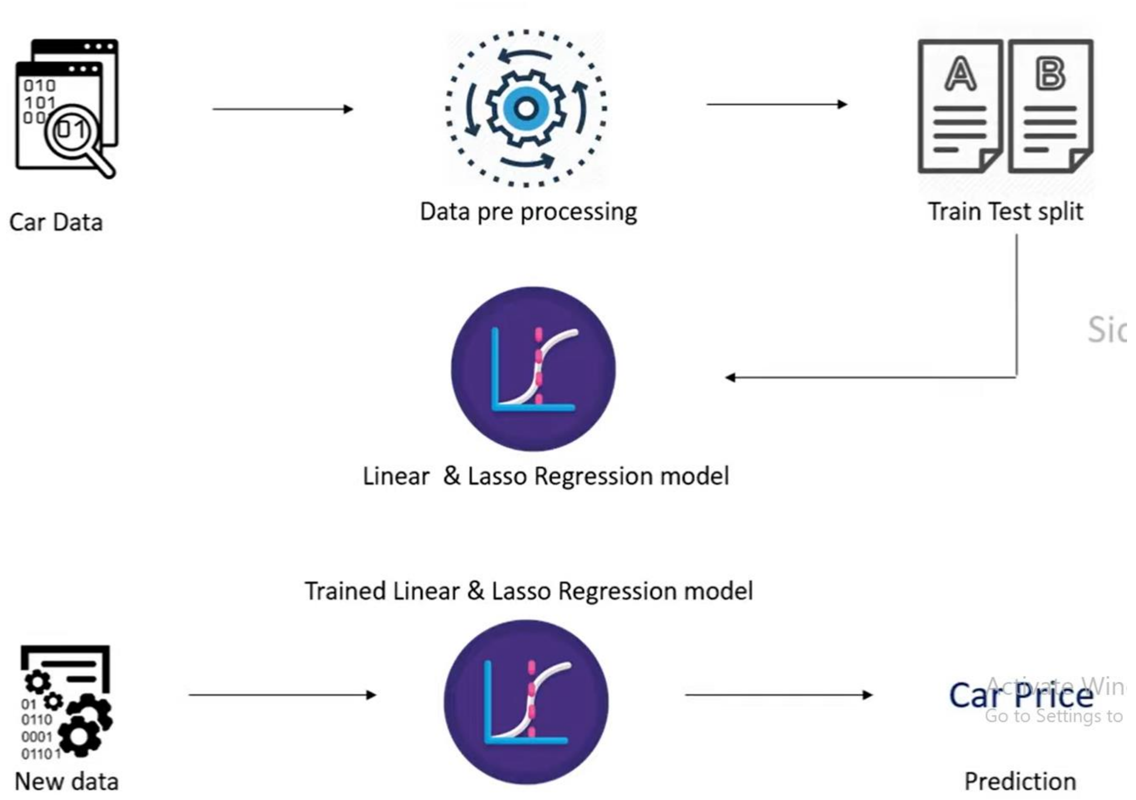
[13] Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. (2001). Support vector clustering. Journal of machine learning research, 2(Dec), 125-137
.
[14] Aizerman, M. A. (1964). Theoretical foundations of the potential function method in pattern recognition learning. Automation and remote control, 25, 821- 837
[15] 3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — 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].

[15] 3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — scikit-learn 0.19.2 documentation. (n.d.). Retrieved from:

9 . Flow diagram :


Car Data
Data pre processing
Train Test split
Linear & Lasso Regression model
Trained Linear & Lasso Regression model
New data
Car Price
Prediction