USED CAR PRICE PREDICTION

GROUP NUMBER-12

GROUP MEMBERS-

ANIRBAN KHARA 1828006

DIPANKAR JANA 1828011

ANSHUL CHAUDHARY 1828053

AVI MISHRA 1828058

ISHIKA BHAUMIK 1828070

**GUIDED BY- Mr. AJAY ANAND**

**Acknowledgement**

***We would like to express our special thanks and gratitude to our teacher Mr. AJAY ANAND, who gave us the golden opportunity to do this wonderful project of USED CAR PRICE PREDICTION.***

***Who also helped us in completing our project. We gained knowledge about many new things, we are really thankful to him.***

**Abstract**

Determining whether the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle’s price on the market. The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases.

We implement and evaluate various learning methods on a dataset consisting of the sale prices of different makes and models across cities in the United States. Our results show that Gradient boosting model with ensemble technique is performing well but is computationally heavy.Conventional linear regression also yielded satisfactory results, with the advantage of a significantly lower training time in comparison to the aforementioned methods.

Contents

**INTRODUCTION -------------------------------------------------- 5**

**TECHNOLOGIES USED ---------------------------------------- 6**

**LINK TO THE PROJECT ---------------------------------------- 7**

**ENHANCEMENTS ON THE PROJECT ---------------------- 7**

**MAJOR CORRELATIONS -------------------------------------- 8**

**PREPROCESSING OF DATA ---------------------------------10**

**OUR APPROCH TO THE PROBLEM --------------------- 12**

**CONCLUSION -----------------------------------------------------22**

**FUTURE SCOPE OF THE PROJECT ---------------------- 23**

**REFRENCES ------------------------------------------------------ 24**

**INTRODUCTION-**

Predicting the price of used cars in both an important and interesting problem.

According to data obtained from the National Transport Authority, the number of cars registered between 2003 and 2013 has witnessed a spectacular increase of 234%. From 68, 524 cars registered in 2003, this number has now reached 160, 701. With difficult economic conditions, it is likely that sales of second-hand imported (reconditioned) cars and used cars will increase. It is reported in that the sales of new cars has registered a decrease of 8% in 2013.

In many developed countries, it is common to lease a car rather than buying it

outright. A lease is a binding contract between a buyer and a seller (or a third party –usually a bank, insurance firm or other financial institutions) in which the buyer must pay fixed installment for a pre-defined number of months/years to the seller/finance. After the lease period is over, the buyer has the possibility to buy the car at its residual value, i.e. its expected resale value. Thus, it is of commercial interest to seller/finances to be able to predict the salvage value (residual value) of cars with accuracy. If the residual value is under-estimated by the seller/financier at the beginning, the installments will be higher for the clients who will certainly then opt for another seller/financier. If the residual value is over-estimated, the installments will be lower for the clients but then the seller/financier may have much difficulty at selling these high-priced used cars at this over-estimated residual value. Thus, we can see that estimating the price of used cars is of very high commercial importance as well.

Manufacturers’ from Germany made a loss of 1 billion Euros in their USA market because of miscalculating the residual value of leased cars. Most individuals in Mauritius who buy new cars are also very apprehensive about the resale value of their cars after certain number of years when they will possibly sell it in the used cars market.

**TECHNOLOGIES USED-**

**Python** - is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant white space. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

**NumPy**- it is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays

**Pandas** - it is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license

**Matplotlib- it** is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged.

**Seaborn**- it is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

**Datetime**- The [datetime](https://docs.python.org/3/library/datetime.html" \l "module-datetime" \o "datetime: Basic date and time types.) module supplies classes for manipulating dates and times.While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation.

**Tensorflow** - The core open source library to help you develop and train ML models. Get started quickly by running Colab notebooks directly in your browser.TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

# **Scikit-learn** - (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms

**Keras** - it is a deep learning API written in Python, running on top of the machine learning platform [TensorFlow](https://github.com/tensorflow/tensorflow). It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

**LINK TO THE PROJECT -**

<https://github.com/kb22/Used-Car-Price-Prediction>

The above link is the original project made by **Karan Bhanot**

It is a project that predicts the price of the car based on features like year, milage, engine, number of seats etc and uses linear regression model and random forest model for the same.

**ENHANCEMENTS ON THE PROJECT-**

In this project we are going to improve the accuracy of the model and deploy the model with the best accuracy.

The data-set of the project-

<https://www.kaggle.com/avikasliwal/used-cars-price-prediction>

**All About the data-**

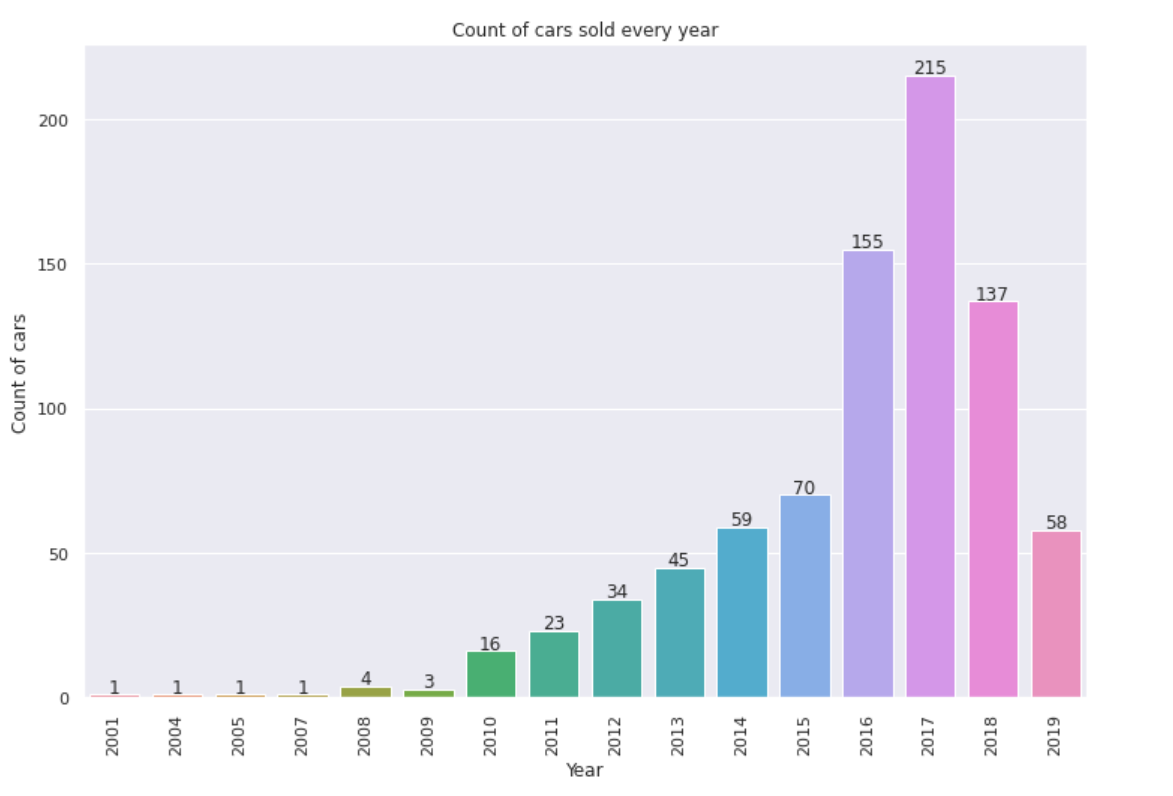
**test-data.csv** contains the test data. There are **1234** rows and **13** cols in the test data.

**train-data.csv** contains the test data. There are **6019** rows and **14** cols in the test data.

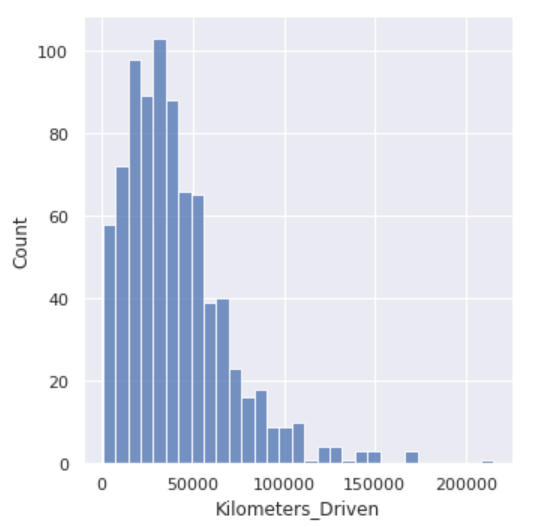
The model made by the author could achieve a maximum accuracy of 88% using random forest regressor. We aim to develop a model that has better accuracy then this.

**MAJOR CORRELATIONS-**

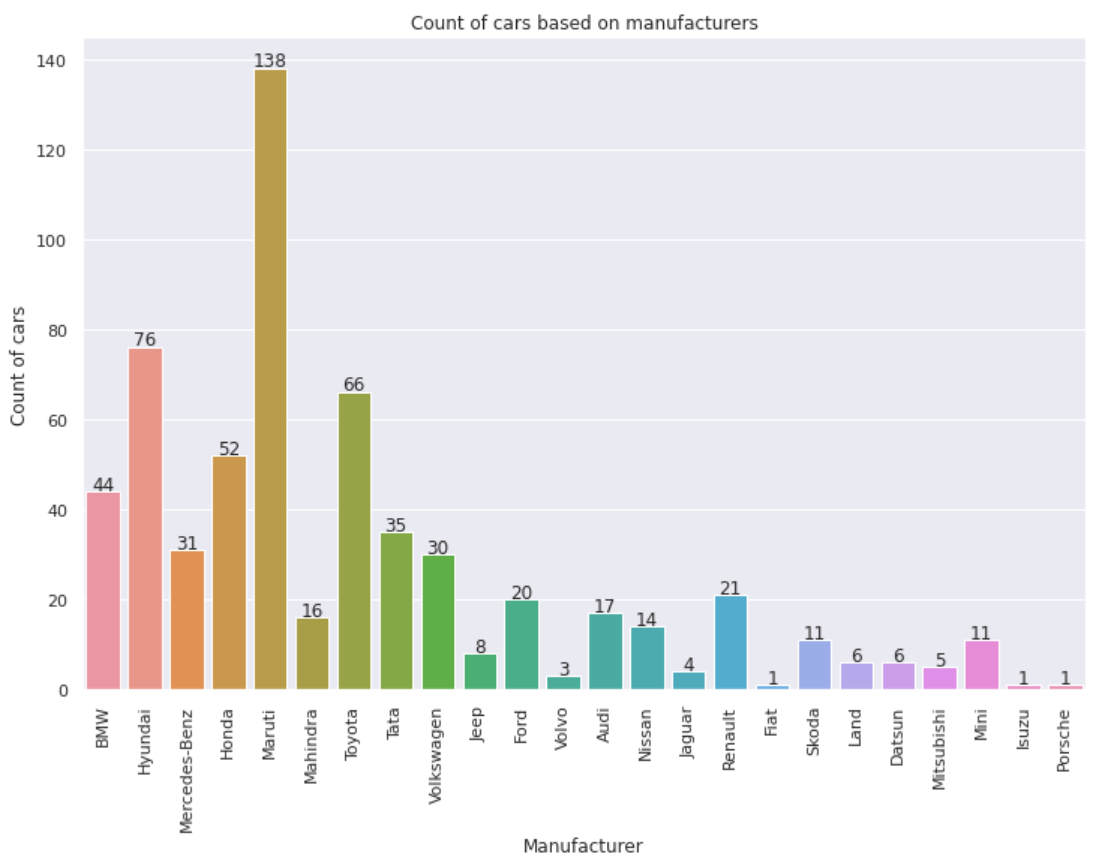
**COUNT VS YEAR-**



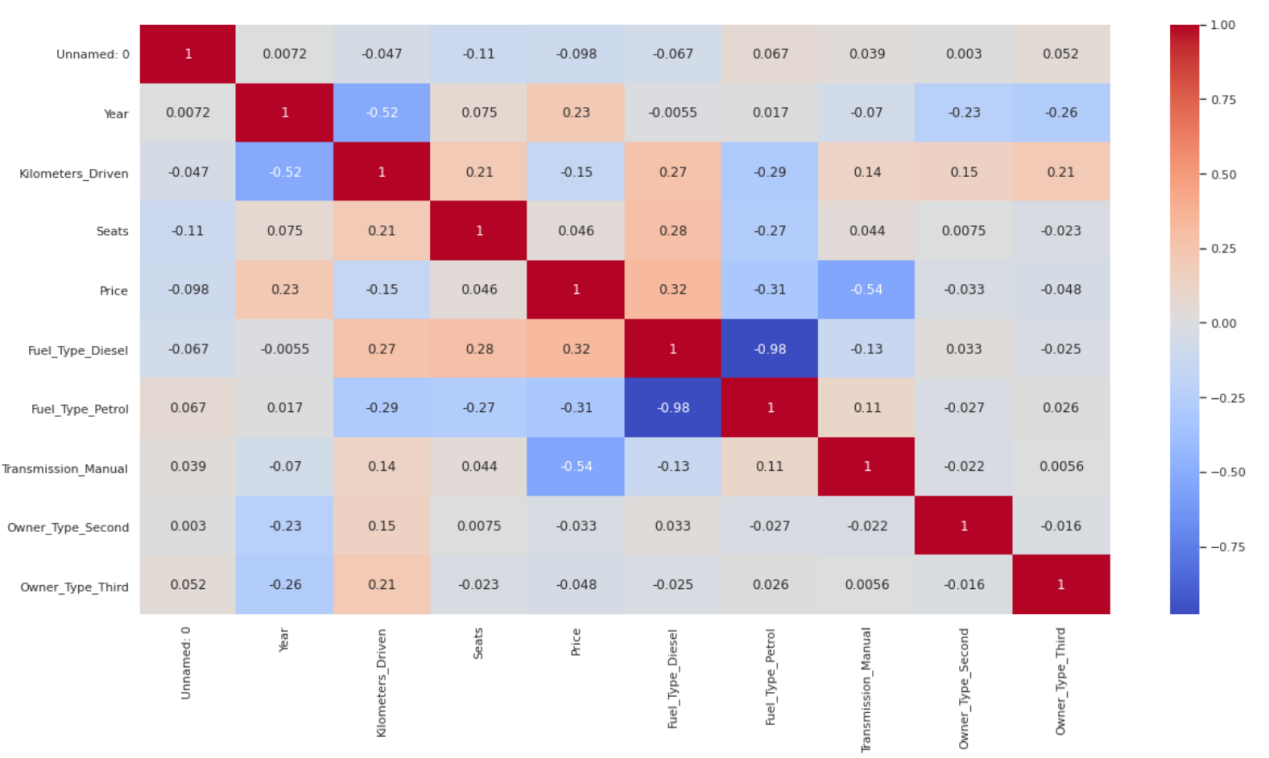
**COUNT VS KILOMETERS\_DRIVEN -**



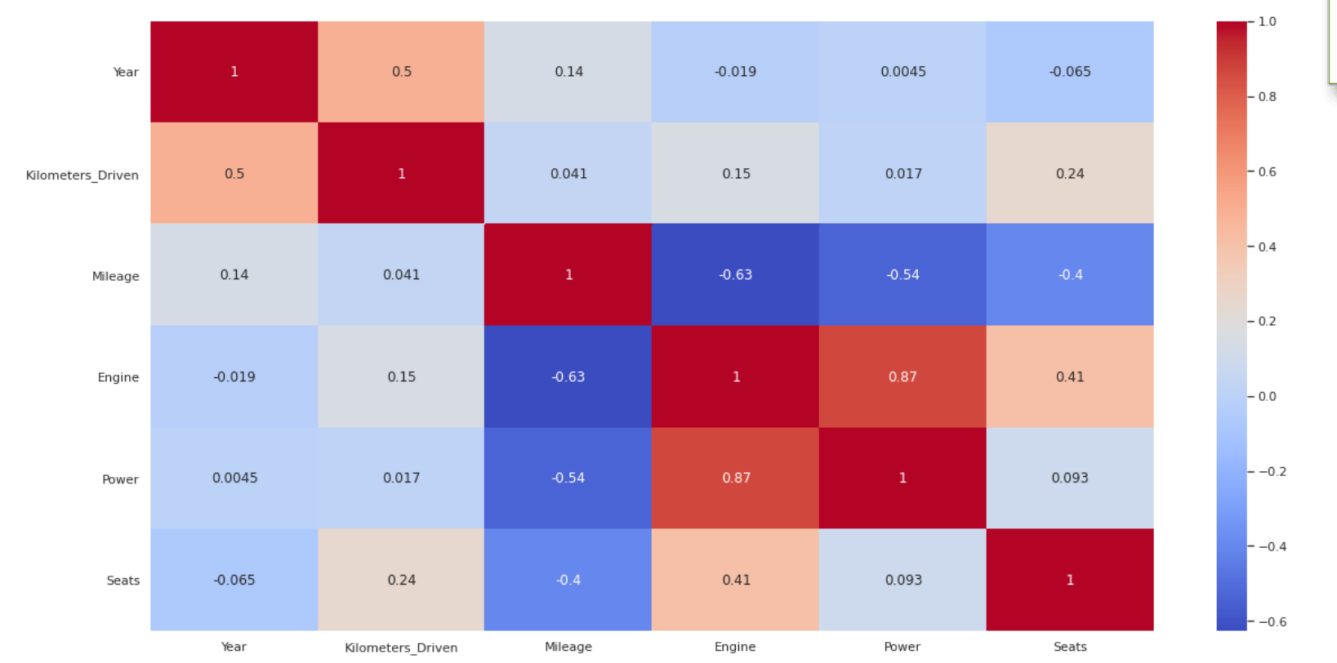
**COUNT VS MANUFACTURE -**



**CORRELATION B/W ATTRIBUTES(before preprocessing the data )**



**CORRELATION B/W ATTRIBUTES AFTER PREPROCESSING**



**PREPROCESSING OF DATA**

## So, What is Data Preprocessing & Why do we need Data Preprocessing?

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

* **Getting the data-set**
* **Importing libraries**
* **Importing datasets**
* **Finding Missing Data**
* **Encoding Categorical Data**
* **Splitting dataset into training and test set**
* **Feature scaling**
* Various libraries that are to be used were imported ex- pandas, numpy, scikit learn, Tensorflow, Matplotlib, Seaborn etc.
* The data was in .csv format so the data was read into a variable using function from pandas library to read a .csv file.
* In this data firstly the missing values of certain columns that is the Nan values were removed. After doing this the data was split into train set and test set.
* From the “name of car model” the name of manufacturer that is Hyundai or Ford etc were extracted and added to the dataset under the column name “manufacturer” and the “name of car model” column was dropped.
* Also the categorical columns "Fuel\_Type", "Transmission",

"Owner\_Type",”Manufacturer" have been changed to numerical column using get dummies function from the pandas library.

* The “location column” and “new price” Column was dropped as it was not putting major effect on the target variable.
* The “year of manufacture” column was converted to number of years from the current year that is if the manufacturing year is 2015 then it will be converted to 6 years that is 2021-2015.
* From the “mileage” column only the first numerical part was extracted that is if the string is 26.6 km/kg then only 26.6 that is the numeric part was extracted as ML models cannot take string inputs. Similar technique was applied to columns “engine” and “power”.
* The Nan values in the “mileage”, “engine”, “power” and “seats” column were replaced by the mean of the values in the respective columns.
* After performing all these the train data and test data was standardized using the standardscaler function from sci-kit learn library.

**OUR APPROCH TO THE PROBLEM-**

Since it the data was Continuous**in nature so we had to plan our model in the same way. We have tried applying regressor models on the data-set.**

****APPROCH 1: LINEAR REGRESSION-****

Linear regression is a ****linear model****, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

When there is a single input variable (x), the method is referred to as ****simple linear regression****. When there are ****multiple input variables****, literature from statistics often refers to the method as multiple linear regression.

Different techniques can be used to prepare or train the linear regression equation from data, the most common of which is called ****Ordinary Least Squares****. It is common to therefore refer to a model prepared this way as Ordinary Least Squares Linear Regression or just Least Squares Regression.

The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric.

The linear equation assigns one scale factor to each input value or column, called a coefficient and represented by the capital Greek letter Beta (B). One additional coefficient is also added, giving the line an additional degree of freedom (e.g. moving up and down on a two-dimensional plot) and is often called the intercept or the bias coefficient.

For example, in a simple regression problem (a single x and a single y), the form of the model would be:

y = B0 + B1\*x

In higher dimensions when we have more than one input (x), the line is called a plane or a hyper-plane. The representation therefore is the form of the equation and the specific values used for the coefficients (e.g. B0 and B1 in the above example).

## **Making Predictions with Linear Regression**

Given the representation is a linear equation, making predictions is as simple as solving the equation for a specific set of inputs.

Let’s make this concrete with an example. Imagine we are predicting weight (y) from height (x). Our linear regression model representation for this problem would be:

y = B0 + B1 \* x1

or

weight =B0 +B1 \* height

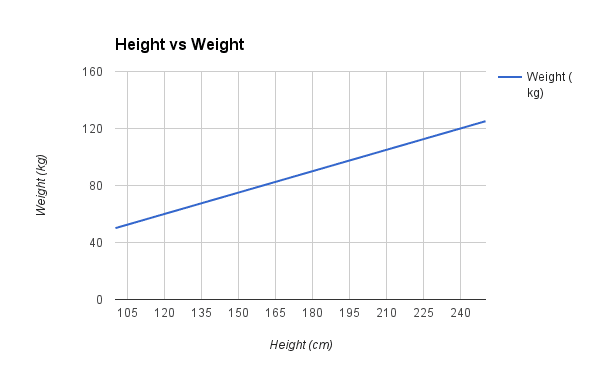
Where B0 is the bias coefficient and B1 is the coefficient for the height column. We use a learning technique to find a good set of coefficient values. Once found, we can plug in different height values to predict the weight.

For example, lets use B0 = 0.1 and B1 = 0.5. Let’s plug them in and calculate the weight (in kilograms) for a person with the height of 182 centimeters.

weight = 0.1 + 0.5 \* 182

weight = 91.1

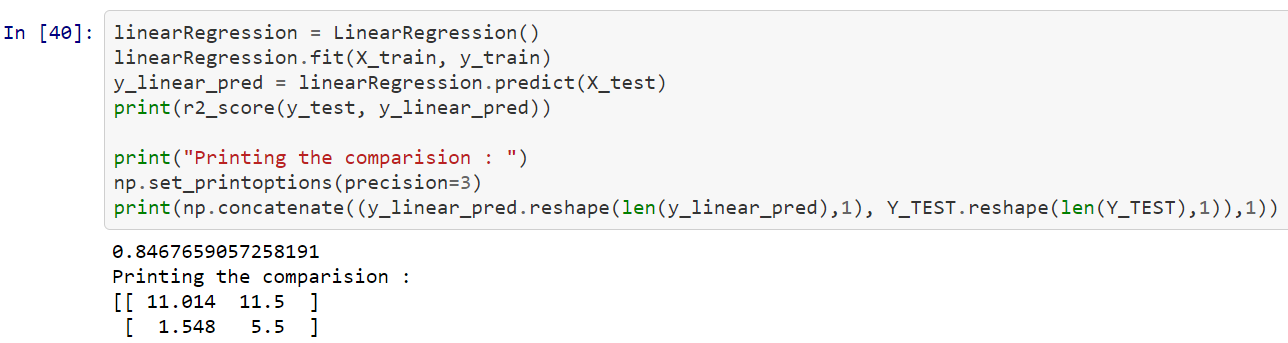
You can see that the above equation could be plotted as a line in two-dimensions. The B0 is our starting point regardless of what height we have. We can run through a bunch of heights from 100 to 250 centimeters and plug them to the equation and get weight values, creating our line.



Sample Height vs Weight Linear Regression

Now that we know how to make predictions given a learned linear regression model, let’s look at some rules of thumb for preparing our data to make the most of this type of model.

**OUR CODE AND OUTPUT-**



As it is clearly seen that the r2 score is only **84.67%** we had to think of something else.

The already made project had  **70%** as the r2score hence our linear regression model is out performing the authors model

**APPROCH 2 DESCISION TREE -**

Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

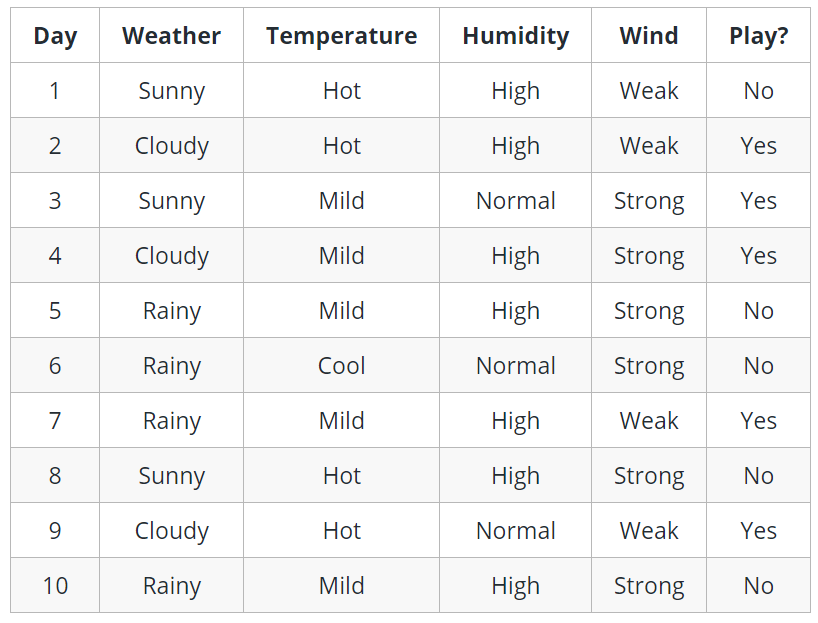
The decision rules are generally in form of if-then-else statements. The deeper the tree, the more complex the rules and fitter the model.

A decision tree is a tree-like graph with nodes representing the place where we pick an attribute and ask a question; edges represent the answers the to the question; and the leaves represent the actual output or class label. They are used in non-linear decision making with simple linear decision surface.

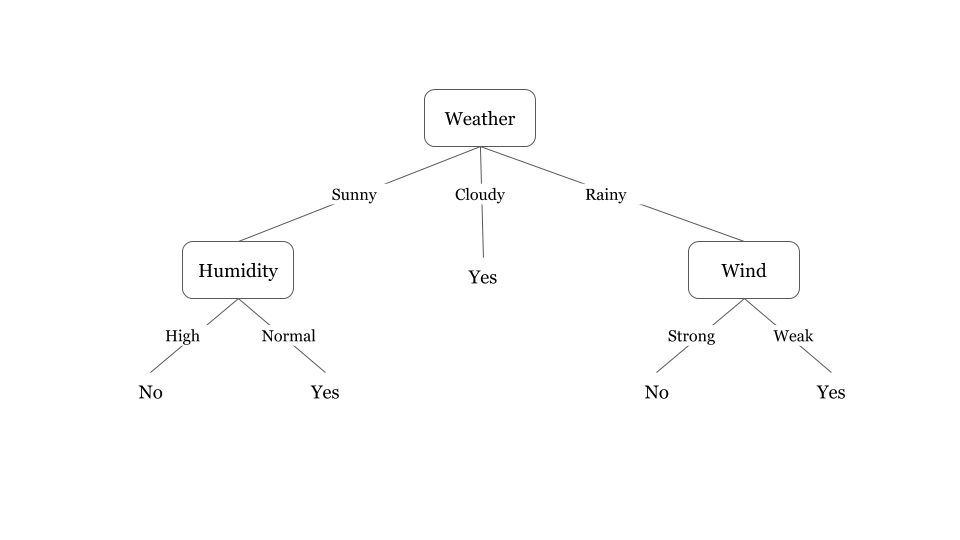
Decision trees classify the examples by sorting them down the tree from the root to some leaf node, with the leaf node providing the classification to the example. Each node in the tree acts as a test case for some attribute, and each edge descending from that node corresponds to one of the possible answers to the test case. This process is recursive in nature and is repeated for every subtree rooted at the new nodes.

et's illustrate this with help of an example. Let's assume we want to play badminton on a particular day — say Saturday — how will you decide whether to play or not. Let's say you go out and check if it's hot or cold, check the speed of the wind and humidity, how the weather is, i.e. is it sunny, cloudy, or rainy. You take all these factors into account to decide if you want to play or not.

So, you calculate all these factors for the last ten days and form a lookup table like the one below.



Now, you may use this table to decide whether to play or not. But, what if the weather pattern on Saturday does not match with any of rows in the table? This may be a problem. A decision tree would be a great way to represent data like this because it takes into account all the possible paths that can lead to the final decision by following a tree-like structure.



### Decision tree boundary

Decision trees divide the feature space into axis-parallel rectangles or hyperplanes. Let’s demonstrate this with help of an example. Let’s consider a simple AND operation on two variables. Assume X and Y to be the coordinates on the x and y axes, respectively, and plot the possible values of X and Y represents the formation of the decision boundary as each decision is taken. We can see that as each decision is made, the feature space gets divided into smaller rectangles and more data points get correctly classified.

**OUR CODE AND OUTPUT-**



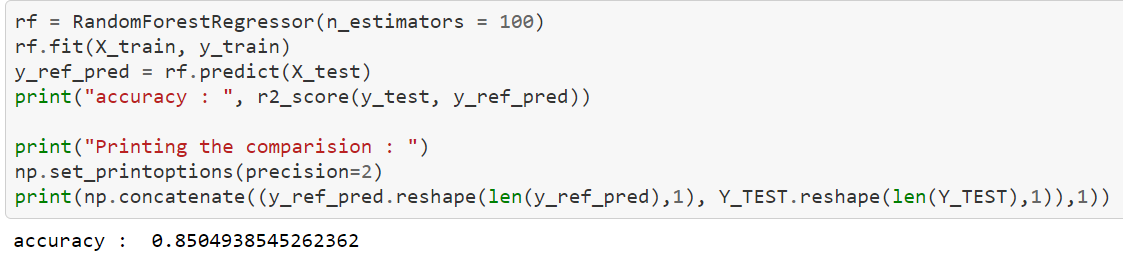
As we can see that the accuracy is **81.3%** we tried ensemble methods for significant improvements.

**APPROACH 3 RANDOM FOREST**

**Random forests** or **random decision forests** are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees.

Ensembles are a divide-and-conquer approach used to improve performance. The main principle behind ensemble methods is that a group of “weak learners” can come together to form a “strong learner”.Each classifier, individually, is a “weak learner,” while all the classifiers taken together are a “strong learner”.

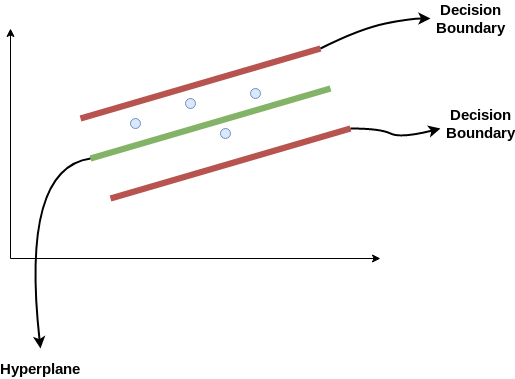
Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction. Sklearn provides a great tool for this that measures a feature's importance by looking at how much the tree nodes that use that feature reduce impurity across all trees in the forest. It computes this score automatically for each feature after training and scales the results so the sum of all importance is equal to one.



The accuracy is calculated to be **85%** which is good but not as expected so we applied SVR for further improving the accuracy.

**APPROACH 4 SUPPORT VECTOR REGRESSOR-**

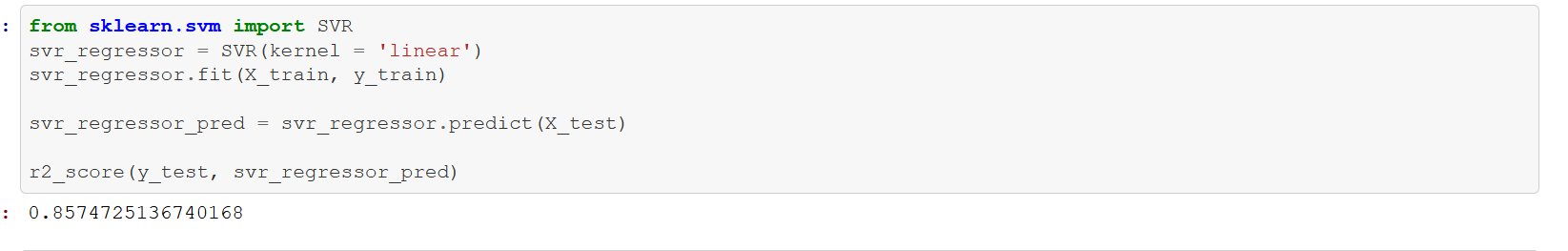
The problem of regression is to find a function that approximates mapping from an input domain to real numbers on the basis of a training sample. So let’s now dive deep and understand how SVR works actually.



Consider these two red lines as the decision boundary and the green line as the hyperplane. **Our objective, when we are moving on with SVR, is to basically consider the points that are within the decision boundary line.** Our best fit line is the hyperplane that has a maximum number of points.

The first thing that we’ll understand is what is the decision boundary (the danger red line above!). Consider these lines as being at any distance, say ‘a’, from the hyperplane. So, these are the lines that we draw at distance ‘+a’ and ‘-a’ from the hyperplane.

Hence, we are going to take only those points that are within the decision boundary and have the least error rate, or are within the Margin of Tolerance. This gives us a better fitting model.

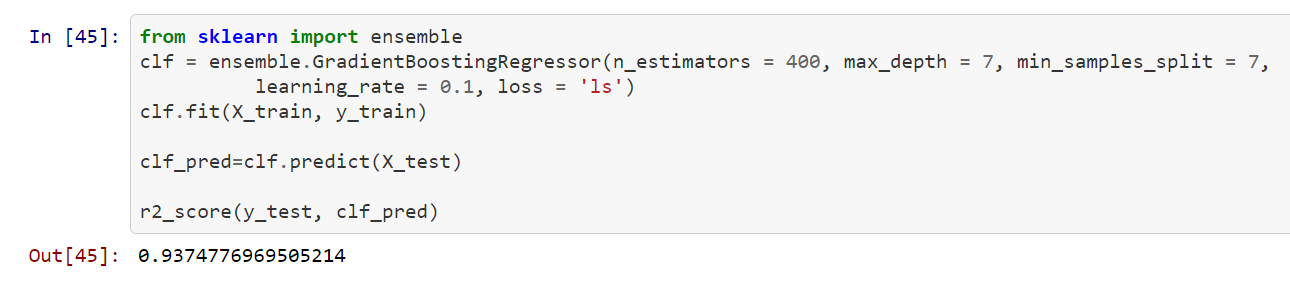


The r2 score is **85.7%** which did not improve much from random forest so we decided to try gradient boosting.

**APPROACH 5 GRADIENT BOOSTING-**

Let’s start by understanding Boosting! Boosting is a method of converting weak learners into strong learners. In boosting, each new tree is a fit on a modified version of the original data set. The gradient boosting algorithm (gbm) can be most easily explained by first introducing the AdaBoost Algorithm.The AdaBoost Algorithm begins by training a decision tree in which each observation is assigned an equal weight. After evaluating the first tree, we increase the weights of those observations that are difficult to classify and lower the weights for those that are easy to classify. The second tree is therefore grown on this weighted data. Here, the idea is to improve upon the predictions of the first tree. Our new model is therefore *Tree 1 + Tree 2*. We then compute the classification error from this new 2-tree ensemble model and grow a third tree to predict the revised residuals. We repeat this process for a specified number of iterations. Subsequent trees help us to classify observations that are not well classified by the previous trees. Predictions of the final ensemble model is therefore the weighted sum of the predictions made by the previous tree models.

**OUR CODE AND r2\_score**

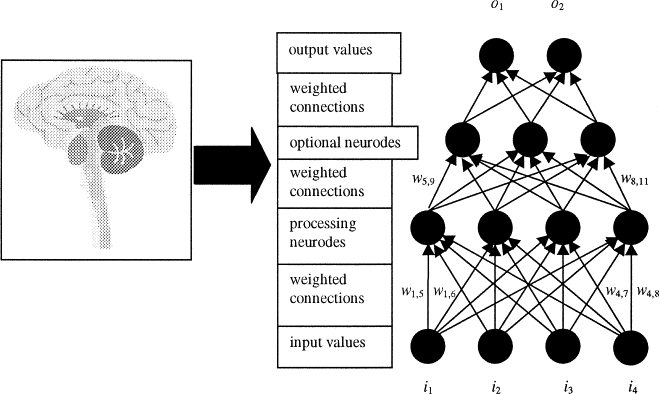


As we can see above that the r2\_score is **93.74%** so it shows significant increase in the accuracy.

This could be achieved by doing hyper parameter optimization so the best results could be yielded with these particular set of hyper parameters that is n\_estimators as 400,max\_depth = 7, min\_samples\_split = 7, learning\_rate = 0.1, loss = 'ls'.

**APPROCH 6 ANN -**

ANN is a technology based on studies of the brain and nervous system as depicted in. These networks emulate a biological neural network but they use a reduced set of concepts from biological neural systems. Specifically, ANN models simulate the electrical activity of the brain and nervous system. Processing elements (also known as either a neurode or perceptron) are connected to other processing elements. Typically the neurodes are arranged in a layer or vector, with the output of one layer serving as the input to the next layer and possibly other layers. A neurode may be connected to all or a subset of the neurodes in the subsequent layer, with these connections simulating the [synaptic connections](https://www.sciencedirect.com/topics/mathematics/synaptic-connection" \o "Learn more about Synaptic Connection from ScienceDirect's AI-generated Topic Pages) of the brain. Weighted data signals entering a neurode simulate the electrical excitation of a nerve cell and consequently the transference of information within the network or brain. The input values to a processing element, in, are multiplied by a connection weight, wn,m, that simulates the strengthening of neural pathways in the brain. It is through the adjustment of the connection strengths or weights that learning is emulated in ANNs.



## Understanding an Artificial Neural Network (ANN)

Artificial neural networks are built like the human brain, with neuron nodes interconnected like a web. The human brain has hundreds of billions of cells called neurons. Each neuron is made up of a cell body that is responsible for processing information by carrying information towards (inputs) and away (outputs) from the brain.

An ANN has hundreds or thousands of artificial neurons called processing units, which are interconnected by nodes. These processing units are made up of input and output units. The input units receive various forms and structures of information based on an internal weighting system, and the neural network attempts to learn about the information presented to produce one output report. Just like humans need rules and guidelines to come up with a result or output, ANNs also use a set of learning rules called backpropagation, an abbreviation for backward propagation of error, to perfect their output results.

An ANN initially goes through a training phase where it learns to recognize patterns in data, whether visually, aurally, or textually. During this supervised phase, the network compares its actual output produced with what it was meant to produce—the desired output. The difference between both outcomes is adjusted using backpropagation. This means that the network works backward, going from the output unit to the input units to adjust the weight of its connections between the units until the difference between the actual and desired outcome produces the lowest possible error.

RELU -

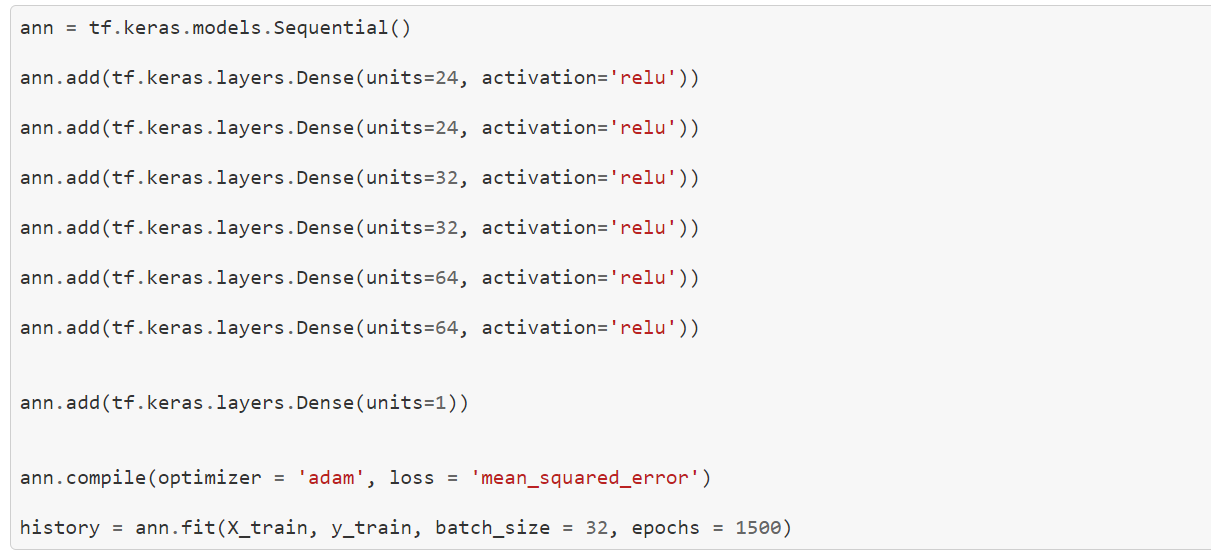
In a neural network, the activation function is responsible for transforming the summed weighted input from the node into the activation of the node or output for that input.

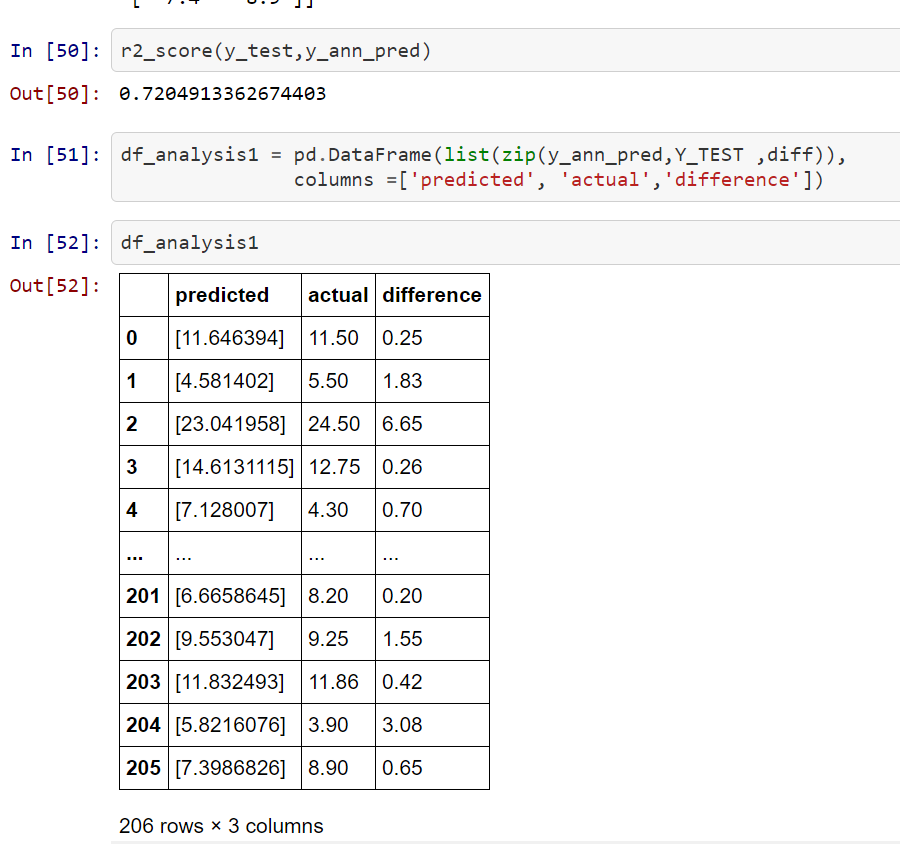
The ****rectified linear activation function**** or ****ReLU**** for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

ADAM - The ****Adam optimization algorithm**** is an extension to stochastic gradient descent that has recently seen broader adoption for deep learning applications in computer vision and natural language processing.

MEAN SQ. ERROR - [Mean Square Error](https://medium.freecodecamp.org/machine-learning-mean-squared-error-regression-line-c7dde9a26b93) is the most commonly used regression loss function. MSE is the sum of squared distances between our target variable and predicted values.

**OUR CODE -**





As it is clear from the figure that the r2 score is **72.04%** which is lower than expected.

Graph for training loss vs epochs.



It can be observed that the loss has decreased considerably over number epochs.

**CONCLUSION-**

**So the best performing model among st all regressor models used on this data-set was gradient boosting regressor with an r2score of 93.7%.**

**The values of hyper parameters used where n\_estimators as 400,max\_depth = 7, min\_samples\_split = 7, learning\_rate = 0.1, loss = 'ls'.**

**FUTURE SCOPE OF THE PROJECT-**

Predicting the resale value of a car is not a simple task. It is trite knowledge that the value of used cars depends on a number of factors. The most important ones are usually the age of the car, its make (and model), the origin of the car (the original country of the manufacturer), its mileage (the number of kilometers it has run) and its horsepower. Due to rising fuel prices, fuel economy is also of prime importance. Unfortunately, in practice, most people do not know exactly how much fuel their car consumes for each km driven. Other factors such as the type of fuel it uses, the interior style, the braking system, acceleration, the volume of its cylinders (measured in cc), safety index, its size, number of doors, paint colour, weight of the car, consumer reviews, prestigious awards won by the car manufacturer, its physical state, whether it is a sports car, whether it has cruise control, whether it is automatic or manual transmission, whether it belonged to an individual or a company and other options such as air conditioner, sound system, power steering, cosmic wheels, GPS navigator all may influence the price as well. Some special factors which buyers attach importance in Mauritius is the local of previous owners, whether the car had been involved in serious accidents and whether it is a lady-driven car. The look and feel of the car certainly contributes a lot to the price. As we can see, the price depends on a large number of factors. Unfortunately, information about all these factors are not always available and the buyer must make the decision to purchase at a certain price based on few factors only. In this work, we have considered only a small subset of the factors mentioned above.

**REFRENCES-**

datetime : <https://docs.python.org/3/library/datetime.html>

numpy : [https://numpy.org/doc/stable/user/quickstart.html](https://numpy.org/doc/stable/user/quickstart.html/)

pandas : <https://pandas.pydata.org/>

matplotlib : <https://matplotlib.org/>

seaborn : <https://seaborn.pydata.org/>

sklearn : <https://scikit-learn.org/stable/>

tensorflow : <https://www.tensorflow.org/>

keras : <https://keras.io/>

Geeksforgeeks : <https://www.geeksforgeeks.org/>

Kaggle : <https://www.kaggle.com/>

Machinelearningmastery : <https://machinelearningmastery.com/>

Towardsdatascience : <https://towardsdatascience.com/>