

Car Price Prediction Project

Submitted by:

ROHIT KATTEWAR

ACKNOWLEDGEMENT

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.

The dataset has 6000 different kinds used cars data. The source of this data is [www.olx.in](http://www.olx.in/).

The given dataset contains various Brands, Models, Kilometers driven, Manufacturing Year, Number of Owners, Fuel Type of the particular car, and finally the price of the car. These cars are selling in various locations in India. The given dataset includes all types of cars for example- SUV, Sedans, Coupe, etc.

INTRODUCTION

* **BUSINESS PROBLEM FRAMING**

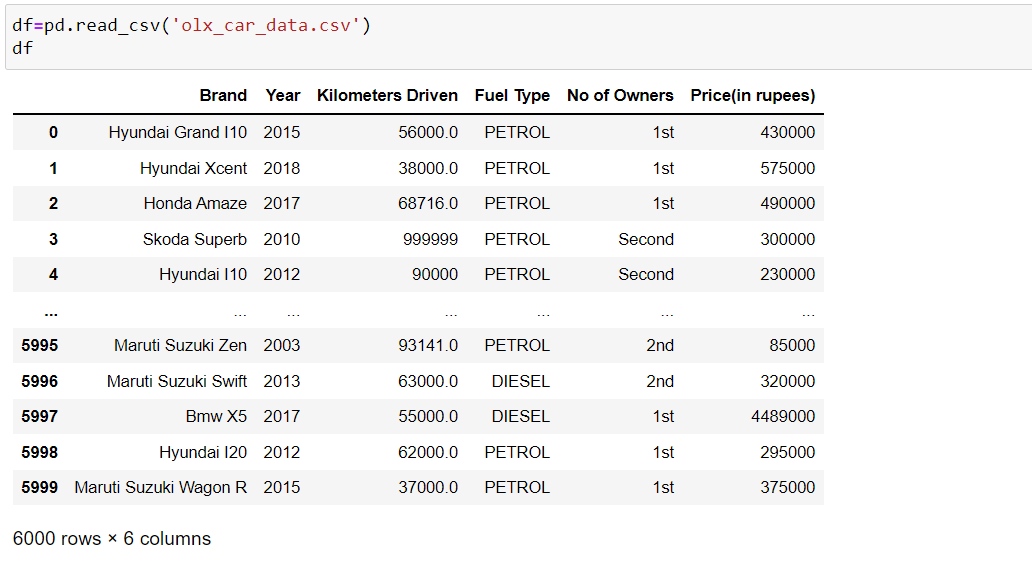
Our client works with small traders, who sell used cars. With the change in market due to Covid-19 impact, our client is facing problems with their previous car price valuation machine learning models. So, they are looking for new machine learning models from new data. We have to make car price valuation model.

ANALYTICAL PROBLEM FRAMING

* **Mathematical/Analytical Modeling of the Problem:**

We will begin with how the looks like in the Data frame, then we will be dealing with the Statistical summary of the data then we will look at the correlation between the various features with each other.

**Data frame**:

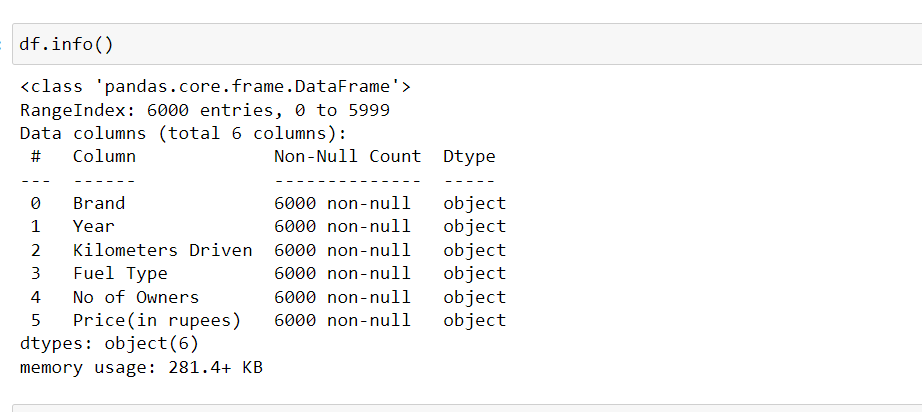


**Description of the dataset**:

**Features:**

The given dataset contains various Brands, Models, Kilometers driven, Manufacturing Year, Number of Owners, Fuel Type of the particular car, and finally the price of the car. These cars are selling in various locations in India. The given dataset includes all types of cars for example- SUV, Sedans, Coupe, etc.

Info of the dataset:

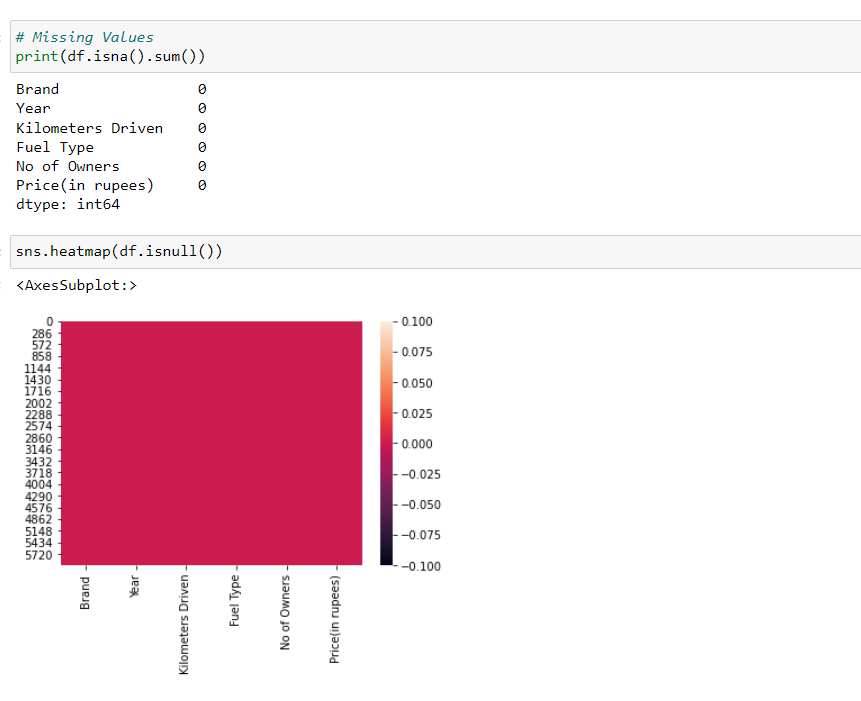


The above image gives the idea about the data types of the given data in the dataset. From above, all the data has the “Object” data type and no column in the dataset has the null values as the Non-Null Count represents the “non-null” value.

The names of the columns are “Brand”, “Year”, “Kilometers Driven”, “Fuel Type”, “No of Owners”, and “Price (in rupees)”.

**Null Values:**

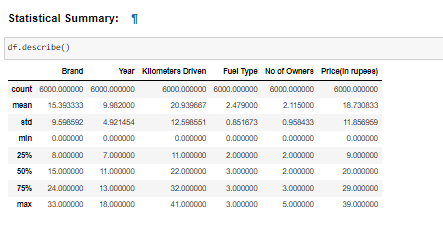
We have no null values in the dataset.



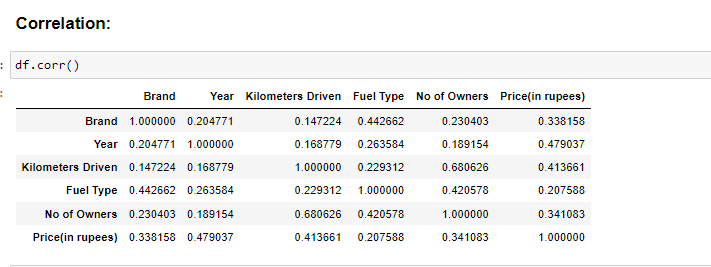
The heatmap above represents the Null values present in the dataset.

**Statistical Summary:**

It gives the basic statistics about the data like the percentile, mean, maximum, minimum etc.



**Correlation:**



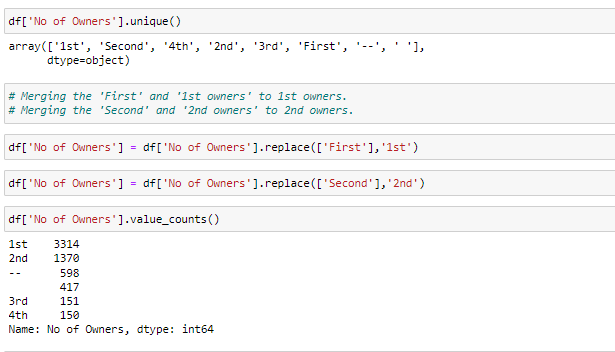
* **Data Sources and their formats**

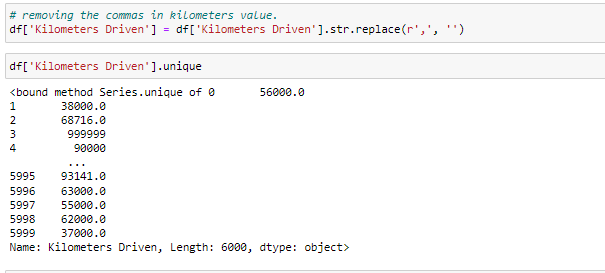
I have extracted the data from [www.olx.in](http://www.olx.in) website. This dataset includes total 6000 cars which belongs various brands, models, different manufacturing years, and selling in the different parts of India.

* **Data Pre-processing Done**

The column “No of Owners” has repeat count of owners so I have merged it into the respective category of the owner.

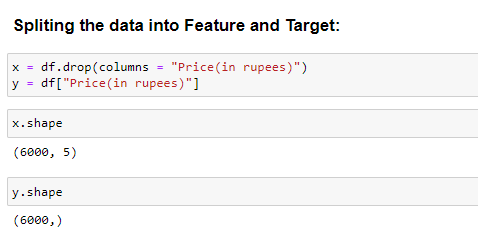
As well as the “Kilometers Driven” column has the commas in the values so I need to remove that.



****

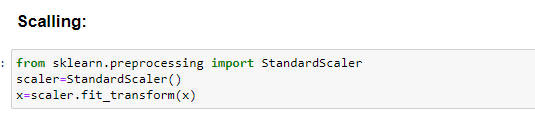
MODEL DEVELOPMENT AND EVALUATION

* **Identification of possible problem-solving approaches (methods)**

****

We will be split the data into target and feature as x and y respectively.

**Scalling and the x and y**

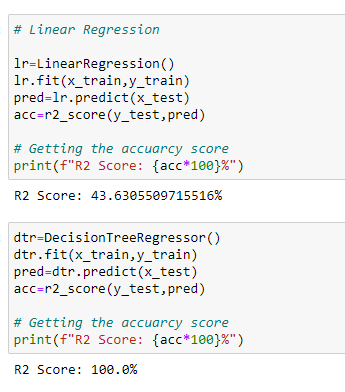
****

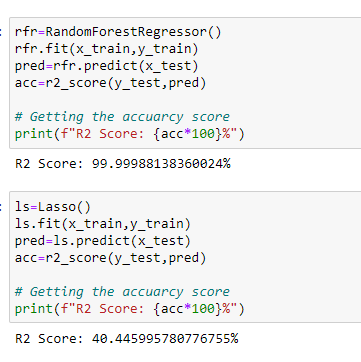
**Getting the best accuracy score and a specific random state**



**Model Building:**









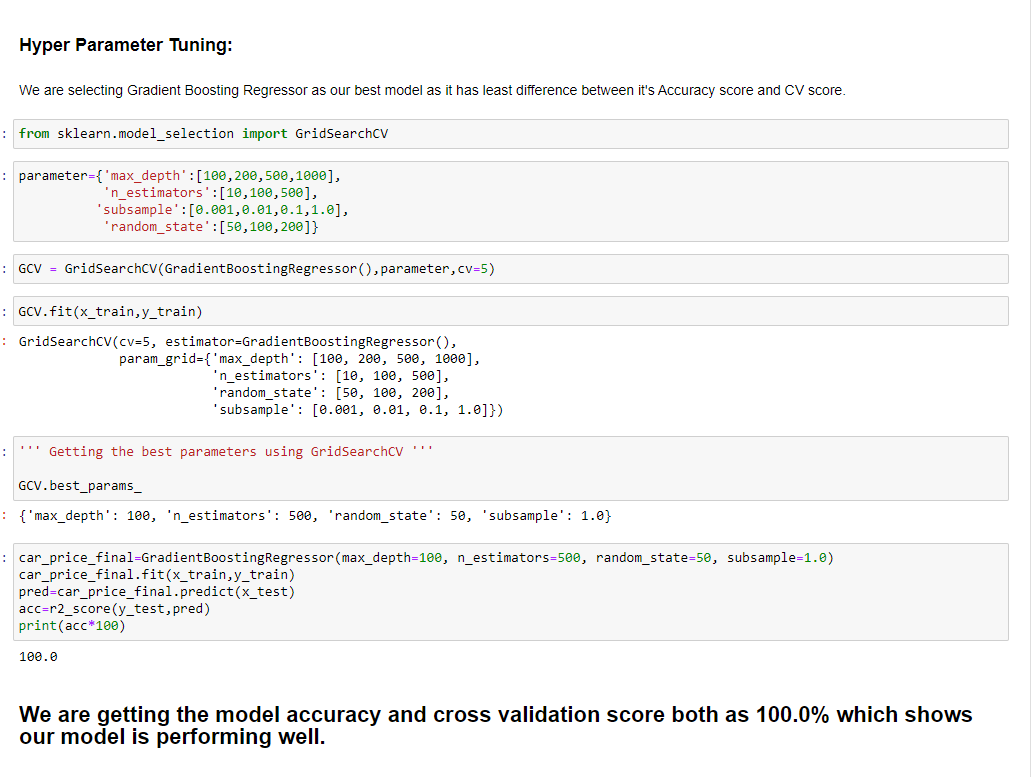
**Cross Validation Score**:



**Gradient Boosting Regressor**:

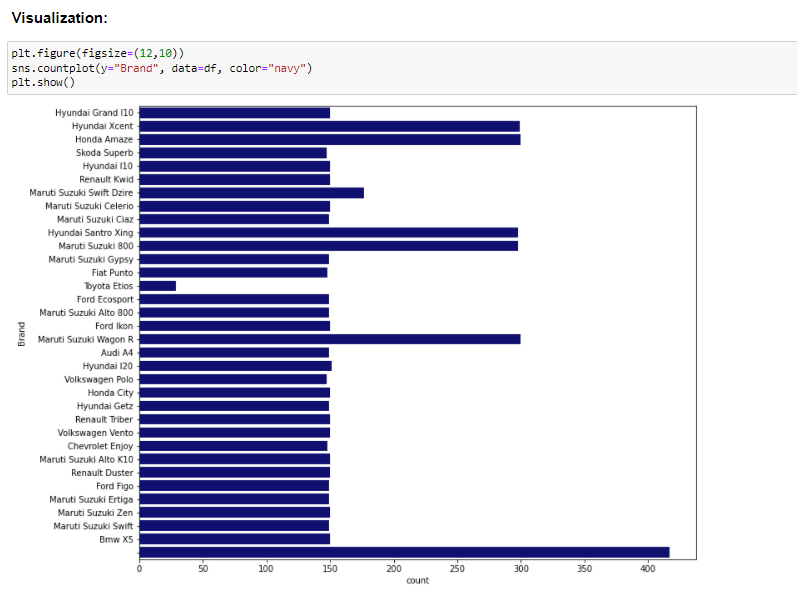
Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest. A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

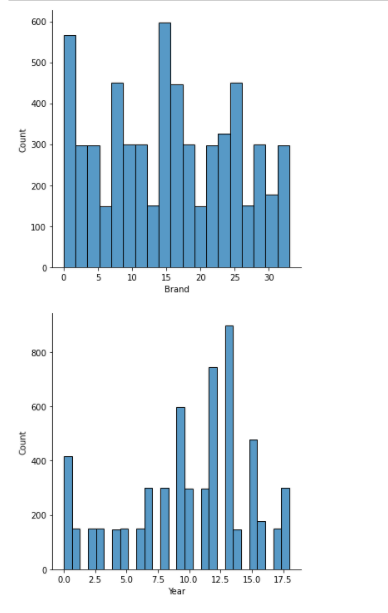
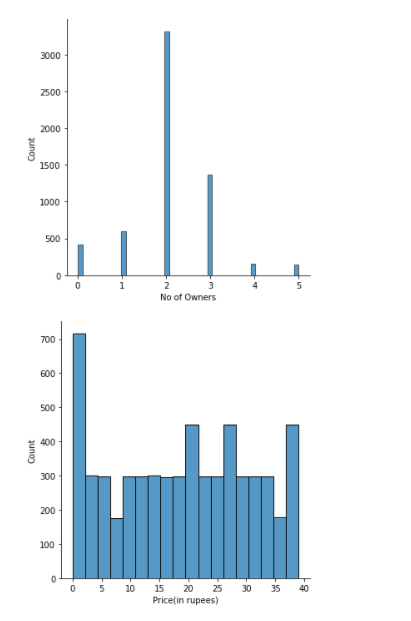
**Hyper Parameter Tuning of the model:**

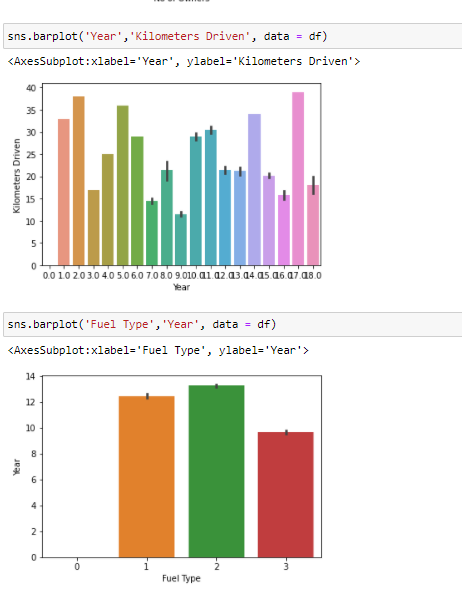


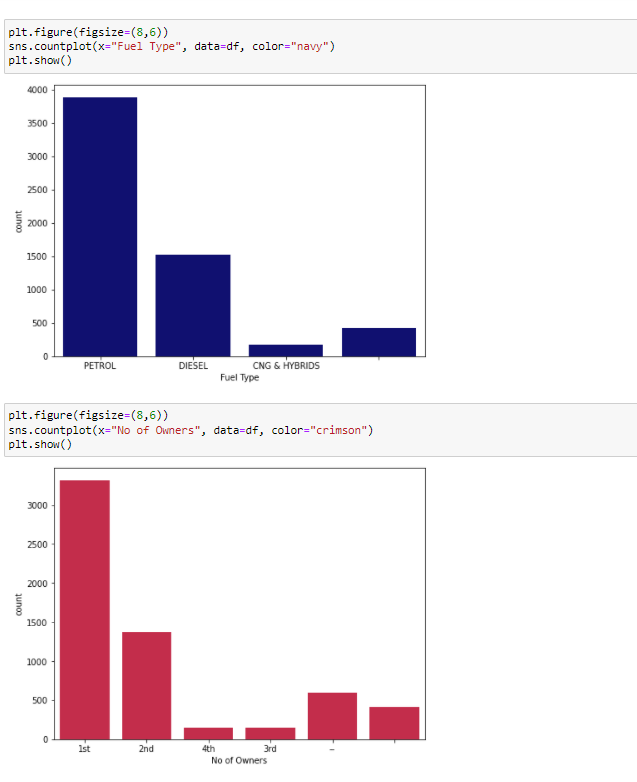
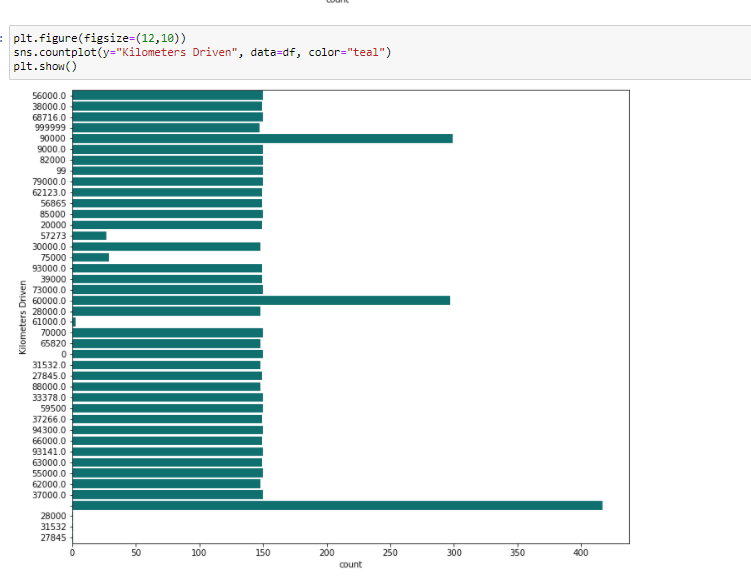
**Here we are getting our Model Accuracy Score and Cross Validation Score both as 100.0%.**

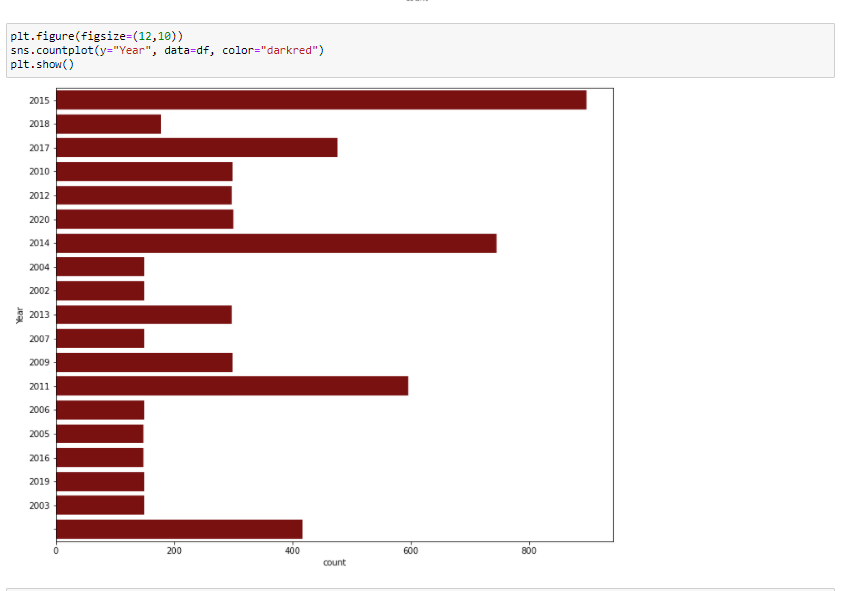
* **Visualization**

****

****

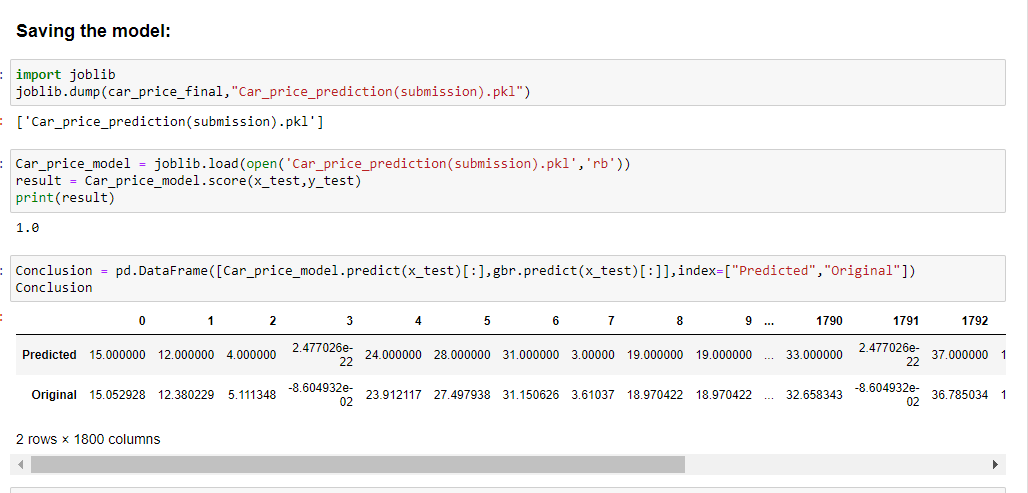
****

****

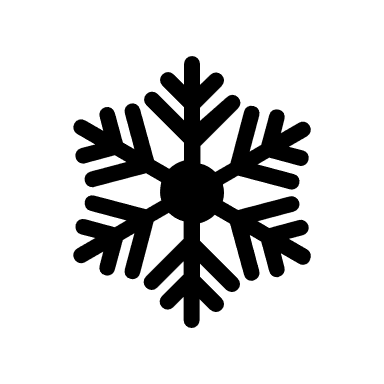
****

CONCLUSION

* **Conclusion:**



Our model is showing the best accuracy and cv score as 100.0%. Hence, we can conclude that our model is performing best.

----- --:-- ----- ----- --:-- -----  ----- --:-- ---------- --:-- -----