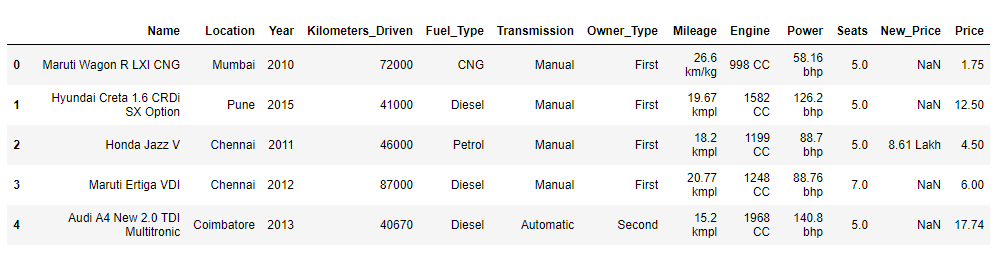
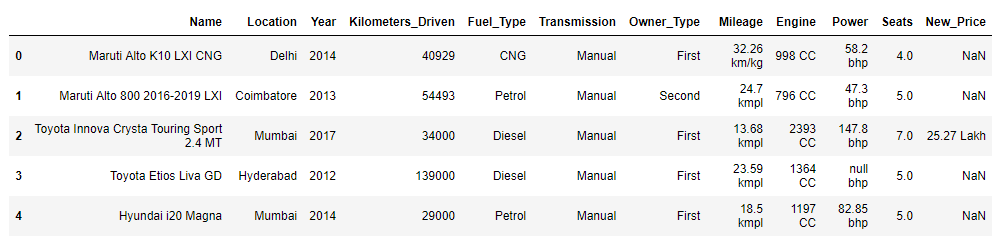
**Train Data:-**



**Test Data:-**



**Numerical values:**

* Kilometers Driven
* Mileage
* Engine
* Power
* Price

**Categorical values:**

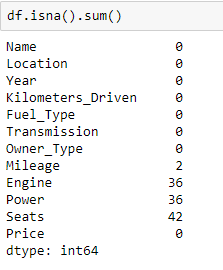
* Name
* Location
* Year
* Fuel type
* Transmission
* Owner type
* Seats

**Preprocessing :**

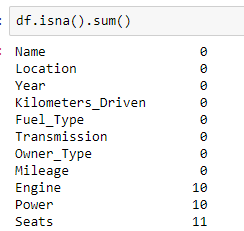
Train data preprocessing:

1.Remove **New\_price** column

2. Remove all NULL values from Train data:



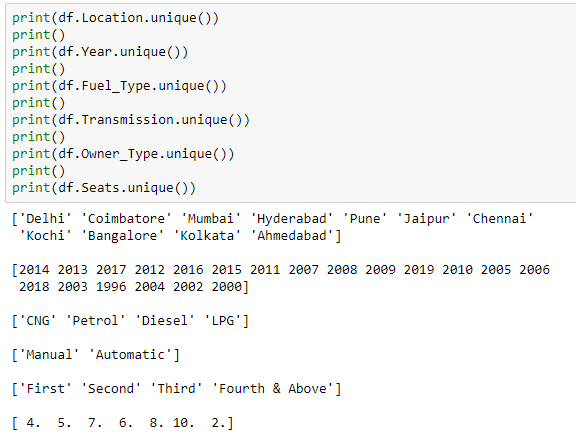
3. Handle missing values in Test data

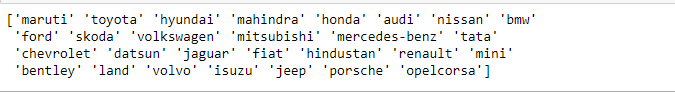


Mileage, Engine, Power is replaced with **Mean** of respected columns.

Seats are replaced with most **frequent** item.

4. Check unique variables in both test data and train data.



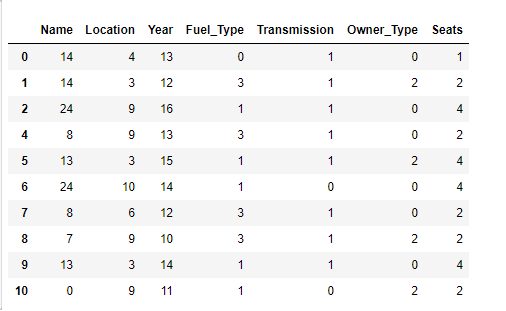


Remove Lamborghini from **Train data**. (for better decision making)

Remove 1996 row from **Test data**. (Since 1996 is not present in train data)

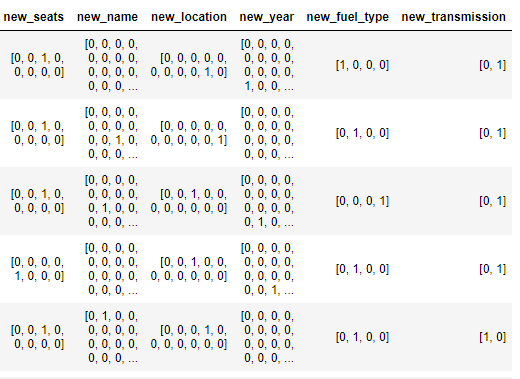
6. Combine Train data and Test data (concat)

7. Label encode Categorical variables

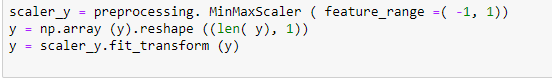


8. One hot encode the label encoded variables

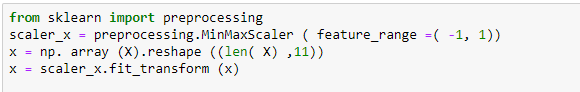
**Sample :**



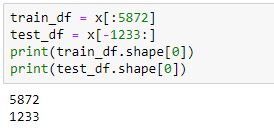
9. Extract the **Price** column into Y variable (remove from dataframe) and apply minmax transformation



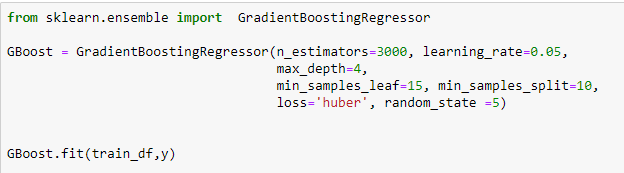
10. Apply minmax scalar over a whole dataframe (X)



11. Split the dataframe into Train and Test



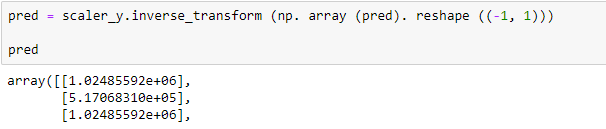
12. Train the model with 100% of **Train data**



13. Make Predictions using **Test data**



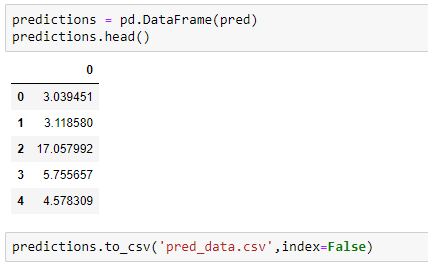
14. Inverse transform the minmax scalar applied over Y (Price from Train)



15. On the Place of year (1996 year) input the average value of predicted results.

Approx. values for 1996 is (4.0 lacks)

16. Save the results in a dataframe.



**Further optimization:**

* Handling NULL values in train data, test data with subbytes of the whole dataframe.
* Feature Importance for better correlated features.
* Feature engineering with valuable formulation with existing parameters.
* Model ensemble for better predictions.