Car Price Prediction Using Machine Learning



Hello Everyone My Name is **Nivitus**. Welcome to the **Car Price Prediction** Tutorial. This is another Machine Learning Blog on **Medium** Site. I hope all of you like this blog; ok I don’t wanna waste your time. Let’s get ready to continue our Journey.

So far so good, today we are going to work on a dataset which consists information about the Car Names, Year, Selling Price, Actual Price and other aspects such as Fuel Type etc. When we work on these sorts of data, we need to see which column is important for us and which is not. Our main aim today is to make a model which can give us a good prediction on the price of the Car Price based on other variables. We are going to use Linear Regression and some other ML algorithms for this dataset and see if it gives us a good accuracy or not.

**Table of Contents:**

* Overview
* Motivation
* Understand the Problem Statement
* About the Dataset
* About the Algorithms used in
* Data Collections
* Data Preprocessing
* Exploratory Data Analysis(EDA)
* Feature Engineering
* Data Cleaning
* Feature Observation
* Feature Selection
* Model Building
* Model Performances
* Prediction and Final Score
* Project Deployement
* Output

**Overview**

In this Blog we are going to do implementing a scalable model for predicting the car price prediction using some of the regression techniques based of some of features in the dataset. In other things about we will see it in upcoming parts …

**Motivation**

The Motivation behind this blog I am always love to know the car models and its types. One day I was see an article in google it just about the top 10 power full and big budget cars in the world. At that time I got an idea about why shouldn’t I do this project for predicting car prices based on the features like fuel type and transmission type. That’s why I decided to write the blog.

**Understand the Problem Statement**

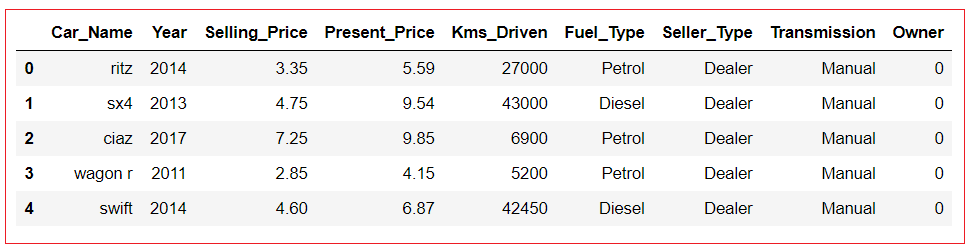


Don’t get confuse about in this project problem statement. It’s actually very simple note this we are going to predicting the selling price from the present price based on the features. It’s seems like the car was sold in a particular price from like age of cars and some other features. We’ll see about all over the things in upcoming section.

**About the Dataset**

In this Dataset I got from the kaggle. As well as here I mentioned some of the things about the dataset like features. The goal of this project is to create a regression model that is able to accurately estimate the price of the car given the features.

**Data Overview**



1. **Car\_Name** – Denotes Name of the Cars
2. **Year** – Denotes Year of Bought
3. **Selling\_Price** – Denotes Price of sold
4. **Present\_Price** – Denotes Current Price
5. **Kms\_Driven** – Counts how many number kilo meters driven in a car
6. **Fuel\_Type** – Denotes types of the fuel
7. **Seller\_Type** – Denotes Seller type
8. **Transmission** - Denotes types of the transmission in a car
9. **Owner** – Denotes how many number of the owners already kept the same car.

**About the Algorithms used in**

The major aim of in this project is to predict the car prices based on the features using some of the regression techniques and algorithms.

1. **Random Forest Regressor**

**Machine Learning Packages are used for in this Project**



**Data Collection**

I got the Dataset from [Kaggle.](https://www.kaggle.com/nehalbirla/vehicle-dataset-from-cardekho) This Dataset consist several features such as Name of the cars, Selling Price, Present price and Fuel type and so on. Let’s know about how to read the dataset into the Jupyter Notebook. You can download the dataset from [Kaggle](https://www.kaggle.com/nehalbirla/vehicle-dataset-from-cardekho)in csv file format.

**Code for collecting data from CSV file into Jupyter Notebook!**

*# Import libraries*

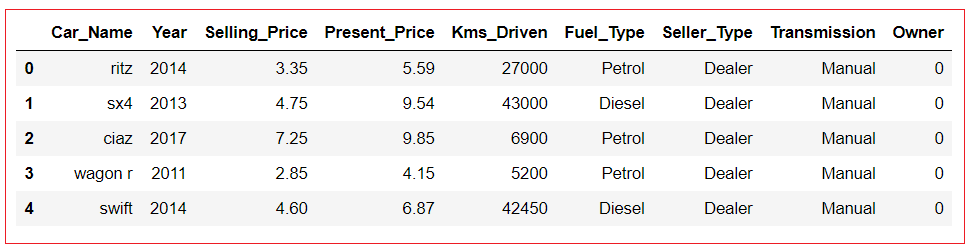
**import** **numpy** **as** **np**

**import** **pandas** **as** **pd**

*# Import the dataset*

df = pd.read\_csv("train.csv")

df.head()



**Data Preprocessing**

In this Car Price Dataset we need not to clean the data. The dataset already cleaned when we download from the Kaggle. For your satisfaction I will show to number of null or missing values in the dataset. As well as we need to understand shape of the dataset.

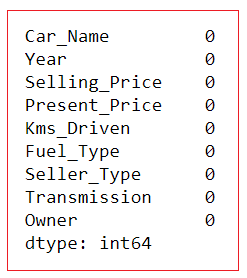
# Shape of the Dataset

print("Shape of the Dataset",df.shape)

Shape of the Dataset (301, 9)

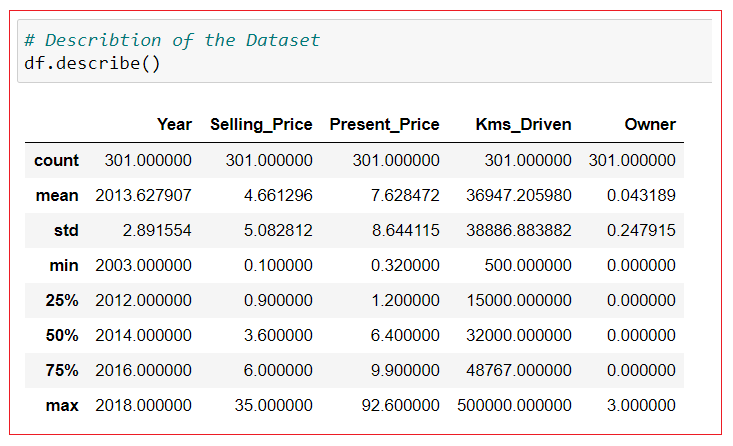
# Checking the Null or Missing Values

df.isnull().sum()

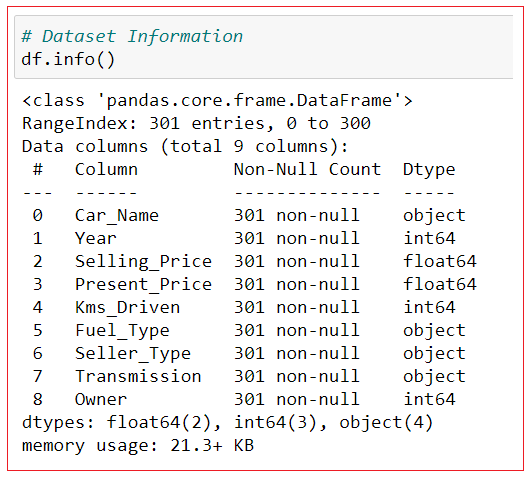


**Exploratory Data Analysis**

In statistics, exploratory data analysis (**EDA**) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily **EDA** is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.



Note: Here the Features are only Numerical values left are Categorical values. We’ll see about it in the upcoming section.



**Feature Engineering**

Feature engineering is the process of using domain knowledge to extract features from raw data via data mining techniques. These features can be used to improve the performance of machine learning algorithms. Feature engineering can be considered as applied machine learning itself.

Note: Here we are going to extracting the new feature from the

Existing features for predicting the output. That’s the technique will prove your domain knowledge. Here as well as we are going to do handing the categorical features in the car price dataset.

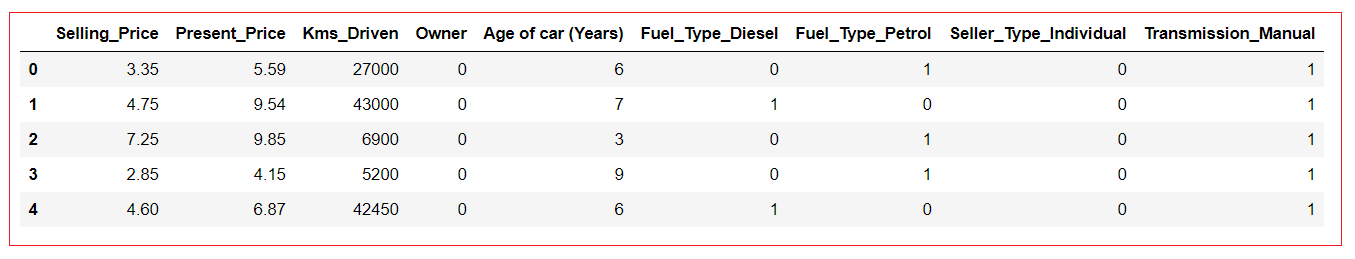
**Data Cleaning**

# Handing the Categorical Values

final\_df = pd.get\_dummies(final\_df,drop\_first=True)

Note: After cleaning the dataset our final dataset look like

final\_df.head()



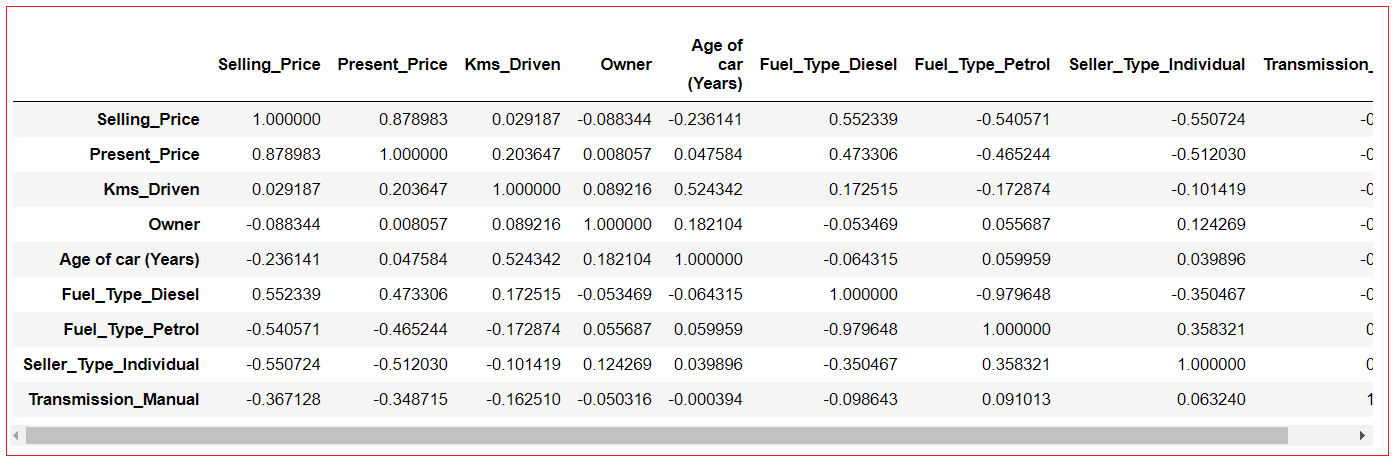
Note: Here the one of the feature is **Age of Cars** its driving from given below method.

Age of Car = Current Year – Bought Year

For Example,

6 = 2020 – 2014

Correlation of each Features

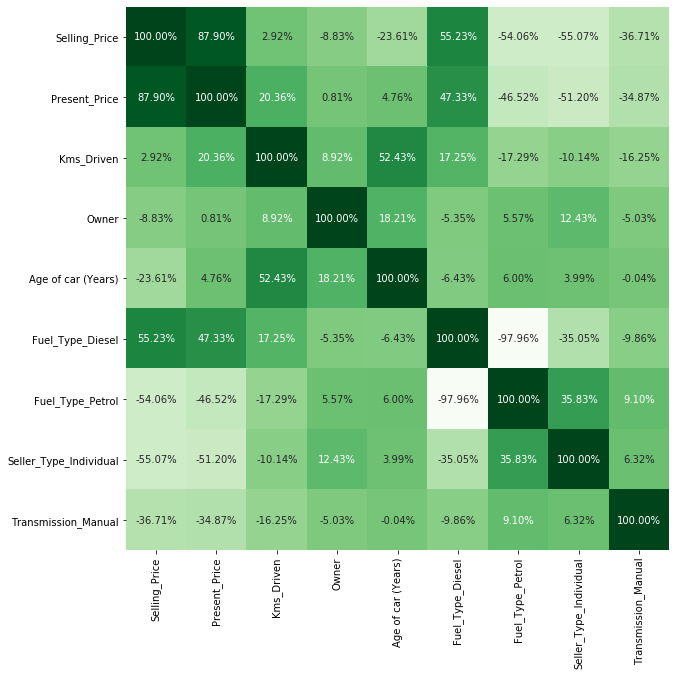


**Feature Observation**

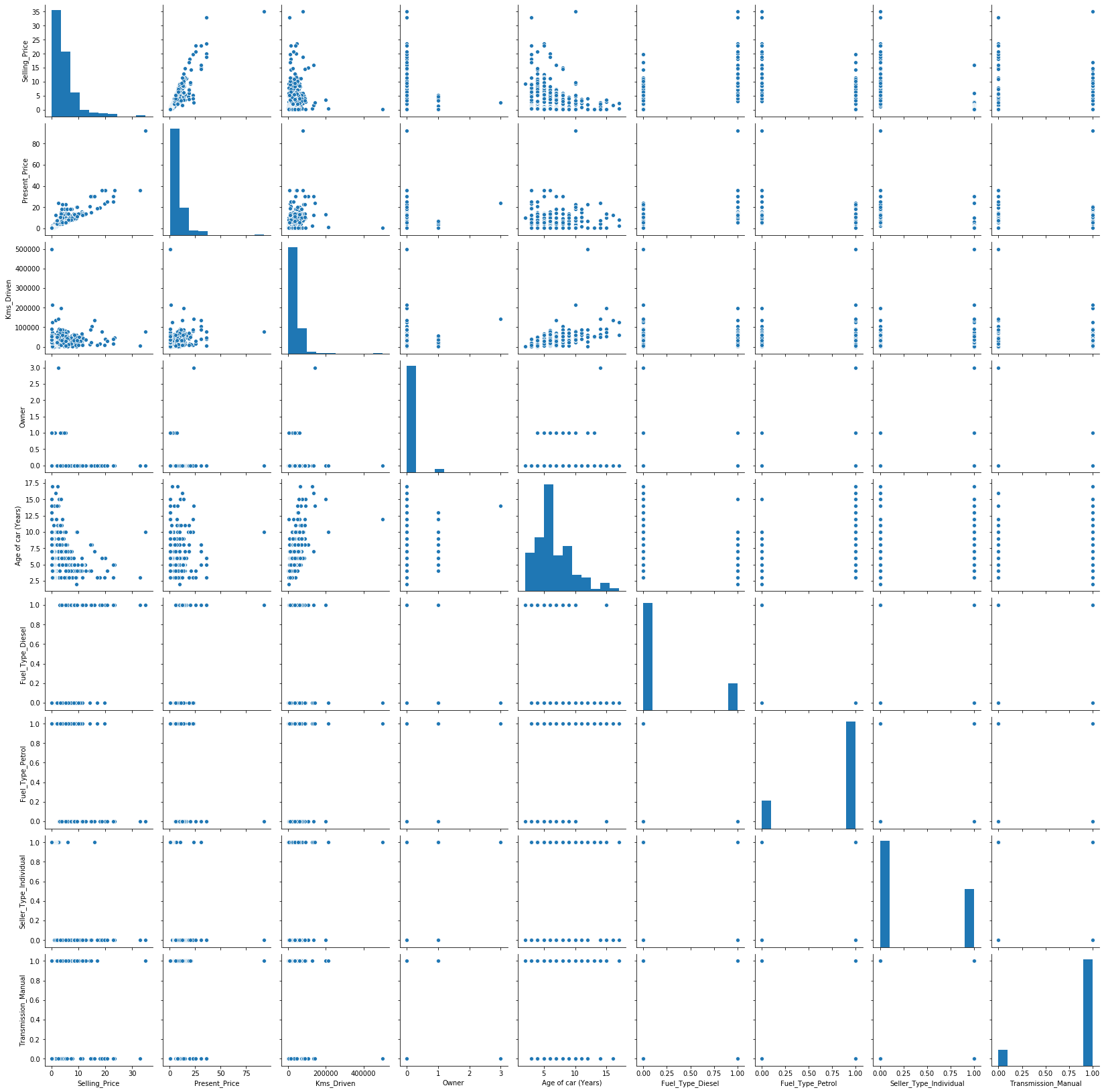
# Plotting the heatmap of correlation between features

plt.figure(figsize=(10,10))

sns.heatmap(final\_df.corr(), cbar=False, square= True, fmt='.2%', annot=True, cmap='Greens')



sns.pairplot(final\_df)



import seaborn as sns

#get correlations of each features in dataset

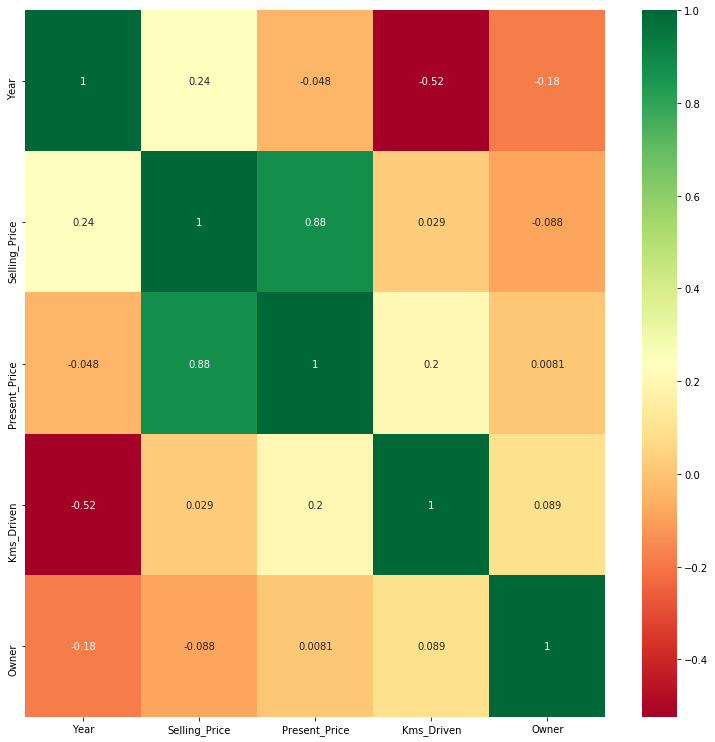
corrmat = df.corr()

top\_corr\_features = corrmat.index

plt.figure(figsize=(13,13))

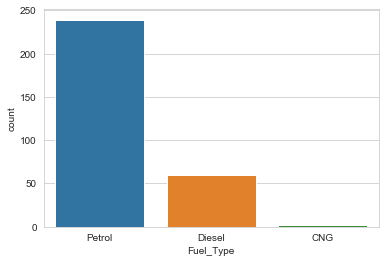
#plot heat map

g=sns.heatmap(df[top\_corr\_features].corr(),annot=True,cmap="RdYlGn")



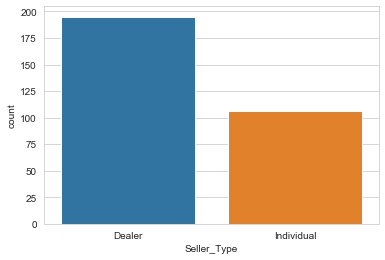
**sns.set\_style('whitegrid')**

**sns.countplot(x='Fuel\_Type',data=df)**

****

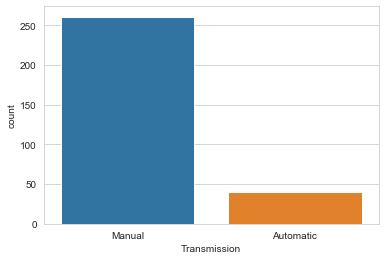
**sns.set\_style('whitegrid')**

**sns.countplot(x='Seller\_Type',data=df)**

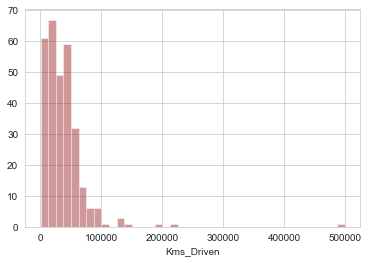
****

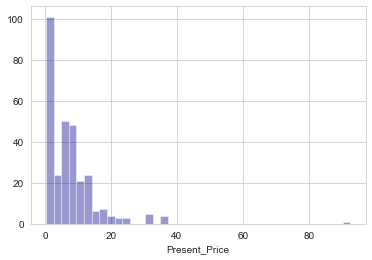
**sns.set\_style('whitegrid')**

**sns.countplot(x='Transmission',data=df)**

****

**sns.distplot(df['Kms\_Driven'].dropna(),kde=False,color='darkred',bins=40)**

**** **sns.distplot(df['Present\_Price'].dropna(),kde=False,color='darkblue',bins=40)**

****

**Feature Selection**

**Feature Selection** is the process where you automatically or manually **select** those **features** which contribute most to your prediction **variable** or output in which you are interested in. Having irrelevant **features** in your data can decrease the accuracy of the models and make your model learn based on irrelevant **features**.

*# Lets try to understand which are important feature for this dataset*

**from** **sklearn.feature\_selection** **import** SelectKBest

**from** **sklearn.feature\_selection** **import** chi2

X = final\_df.iloc[:,1:]

y = final\_df.iloc[:,0]

from sklearn.ensemble import ExtraTreesRegressor

model = ExtraTreesRegressor()

model.fit(X,y)

print(model.feature\_importances\_)

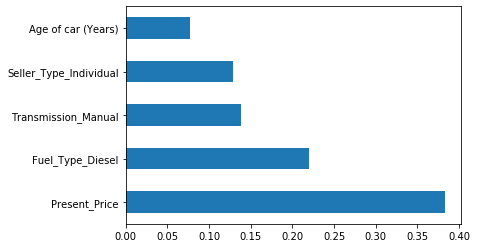
[0.38268003 0.04197867 0.00119488 0.07688097 0.22028522 0.011101890.12807213 0.13780622]

# Important features for Car Price Prediction Dataset

feat\_importances = pd.Series(model.feature\_importances\_, index=X.columns)

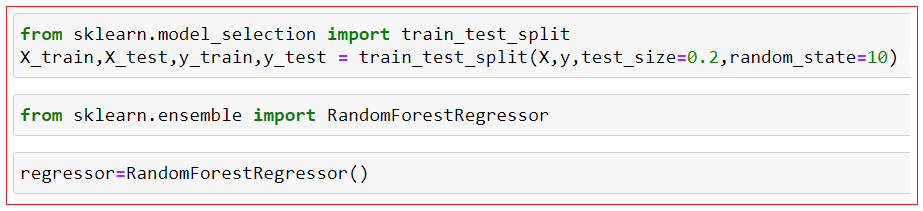
feat\_importances.nlargest(5).plot(kind='barh')

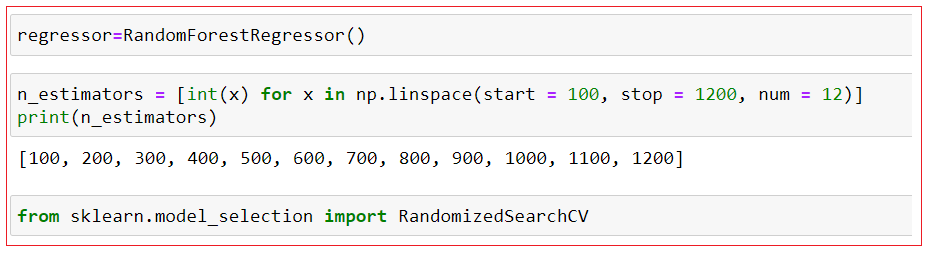
plt.show()



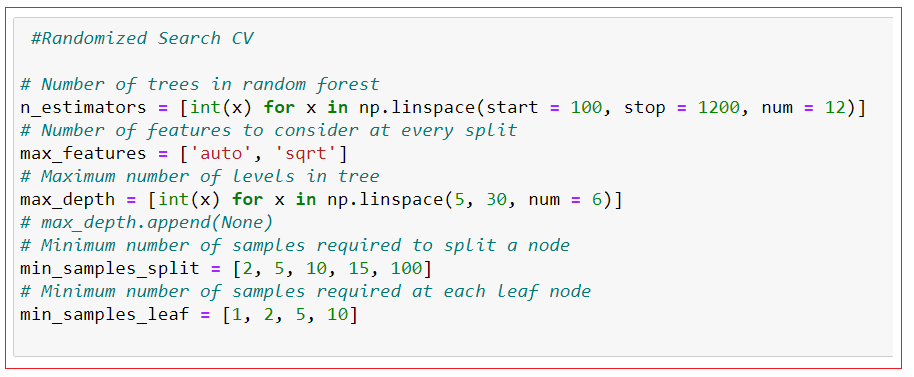
**Model Fitting**

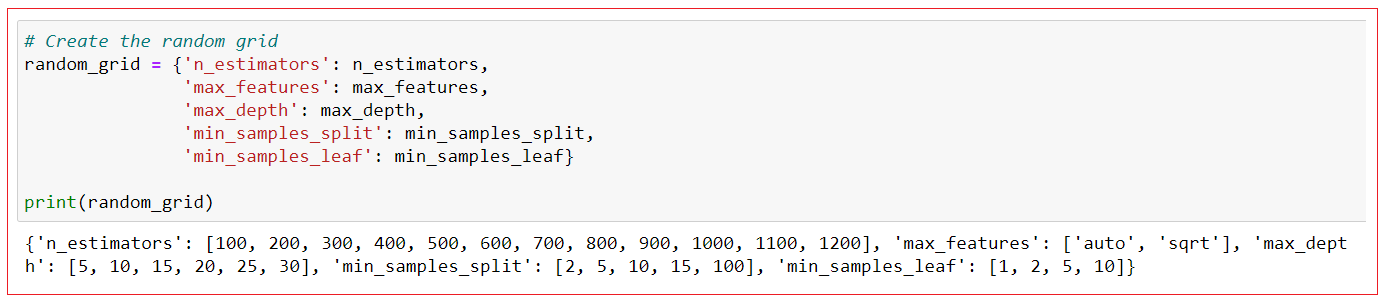
**Random Forest Regressor**

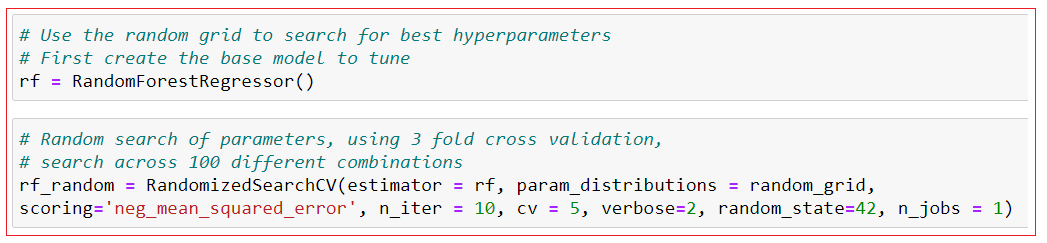


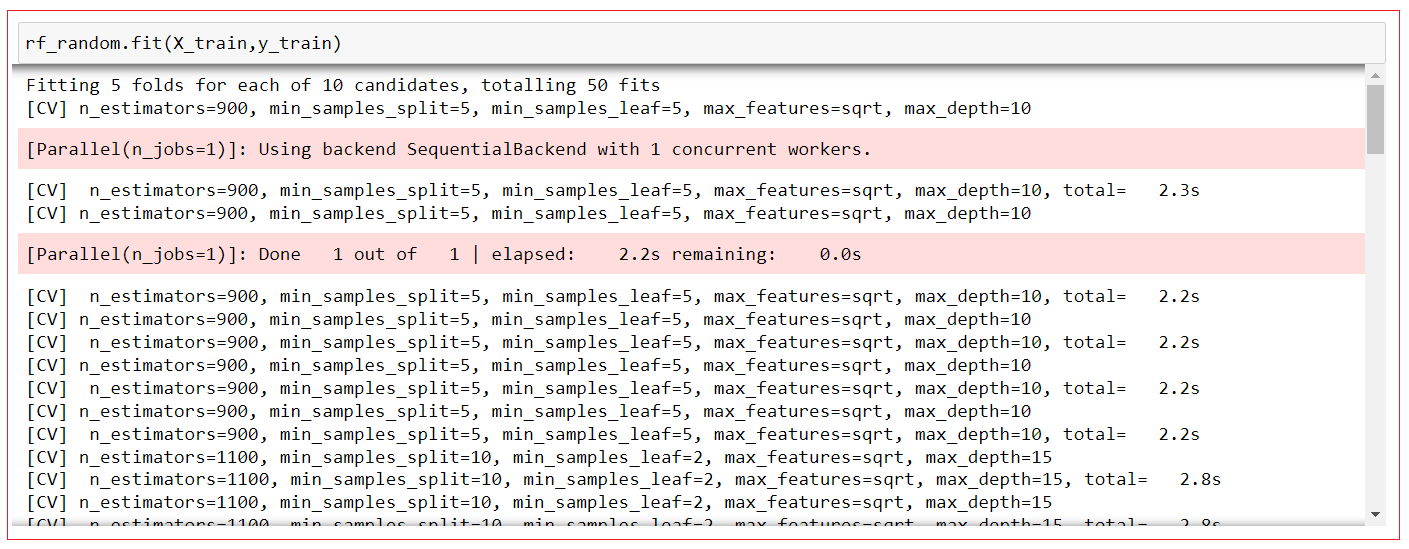


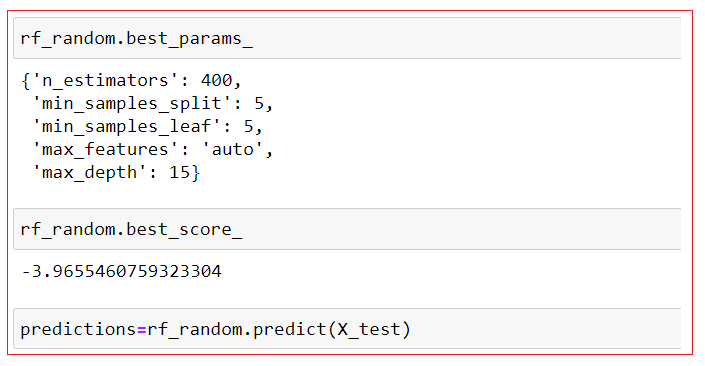
Note: Here we have use gridsearch cv for better prediction.

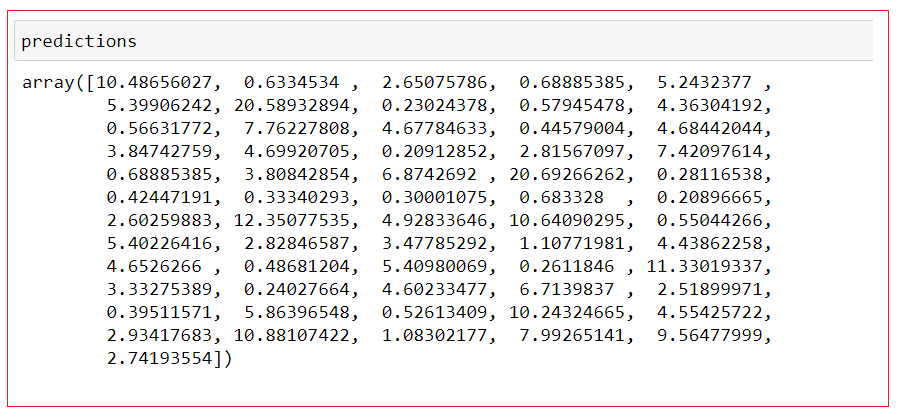






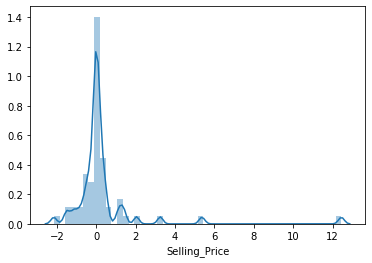




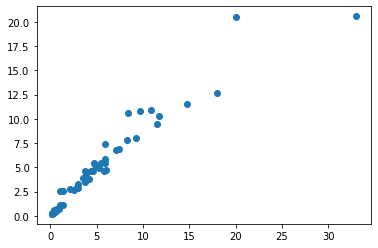


**Model Performance**

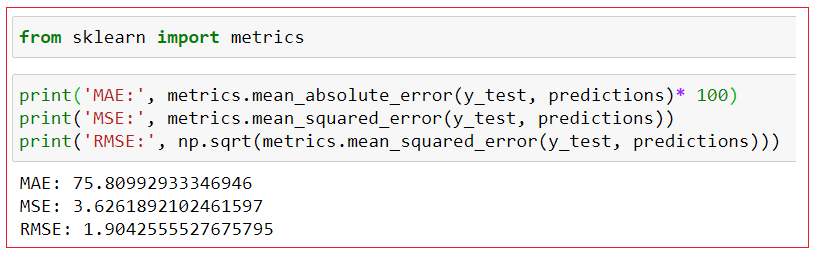
sns.distplot(y\_test-predictions)

****

plt.scatter(y\_test,predictions)



**Prediction and Final Score:**



**Project Deployment**

In this project I already deployed in one the cloud Platform which is Heroku. Here I’ll give my [project demo](https://carspred.herokuapp.com/) you can check it out. If you don’t know about Heroku platform just [click here.](https://en.wikipedia.org/wiki/Heroku)

**Output & Conclusion**

****

From the Exploratory Data Analysis, we could generate insight from the data. How each of the features relates to the target. Also, it can be seen from the evaluation of three models that Random Forest Regressor performed well.

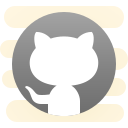
I Hope all of You Like this blog. If you wanna say more about in this blog just contact me.

**Name: Nivitus**

**Mobile Number: 9994268967**

**Email: nivitusfdo007@gmail.com**

You can Ping me on these

[](https://www.linkedin.com/in/nivitus-fernandez-999a6815b/) [](https://github.com/Nivitus) [](https://www.instagram.com/nivitus_fernandez/) [](https://www.facebook.com/profile.php?id=100004639637591)