Car Price Prediction Document

- Car Price Prediction dataset consist of 205 rows and 26 columns
- This dataset consists of lots of null values and these null values are in the form of "?"
- I have replaced these "?" with np.nan values
- Mostly these null values are in the numerical columns so I replaced these np.nan values with their respective mean for each column
- This dataset consists of large number of categorical columns
- In these categorical columns some of the columns have ordinal values like one, two and so on…
- I replaced these values with integer value like 1, 2 and change the column type to integer
- For most of the columns I used pd.get_dummies to convert it to numerical columns
- Even though pd.get_dummies increase the dimensionality of the data I don't want lose those columns, as these columns are more important in the car price prediction
- In some of the columns the value counts for some of the values are 1.
- If I replaced these values with pd.get_dummies it will lead to sparse matrix. So, I decided to remove that row
- After these pre-processing the dataset consists of 197 rows and 57 columns
- As we see that, this dataset consists of more number of columns. So, I decided to do PCA
- For that I split the data into X and Y
- Then using Sklearn libraries' StandardScaler, I standardized the data
- After that I did feature extraction using PCA method
    - Using the PCA I find eigen values
    - With that eigen values I find that using the first 48 components I can explain the 99% of variance

- I used both the data before PCA and after PCA for model building
- Using the model selection module form sklearn split the data into X_train, X_test, y_train and y_test
- Basic model that used are
    - Linear Regression
    - Decision Tree Regressor
    - Gradient Boosting Regressor
    - Random Forest Regressor
    - Support Vector Regressor

- I have trained these models to both the data before PCA and after PCA
- Surprisingly, both the data before PCA and after PCA gives the more or less same R2_score
- So, I decided to use before PCA data for further Hyper Parameter Tuning
- I have selected 3 best model based on their R2_score for Hyper Parameter Tuning, they are
    - Decision Tree Regressor
    - Gradient Boosting Regressor
    - Random Forest Regressor
- From the scores obtain Gradient Boosting Regressor having the highest score
- Final model for the Car Price Prediction is Gradient Boosting Regressor
    - R2_score of 83%
    - Mean Squared Error is 7653594.048983475
    - Mean Absolute Error is 1882.2091252991274