HOUSING PRICE PREDICTION   Dataset

. DATASET INFORMATION
- Data contains 1460 entries each having 81 variables.

*IMPORT NECESSARY LIBRARY= PANADAS, NUMPY, MATPLOTLIB, SEABORN, SCIPY, SKLEARN, XGBOOST

DATA-PREPROCESSING

-First load the datset on jupyter notebook and print it then check the shape of the dataset
-I started looking at randomly some rows and columns of dataset and tried to understand
- Have done a lot Data-preprocessisng / EDA on this Dataset

. While checking the null elements of the dataset , it was observed that
. (PoolQC, Fence, MiscFeature, Alley) In these all Attributes more than 95 percent of elements are null value it is so many null value that it can't fill be fill if there are try to fill all the null value of these Attributes then it doesn't make any sense. therefore these Attributes will have to drop because there is no use to keeping them

. Impute the null value by mean, mode in that columns which contain minimum null value and I filled null value in some columns with groupby method
. After impute the all null value . I use describe function to check Statistical information about dataset so I found that some columns ('PoolArea','MiscVal','ScreenPorch','3SsnPorch','LowQualFinSF','BsmtFinSF2') these all columns have more than 85 or 90 percent are same value and when I plot boxplot on these columns then all values come under Outliers except similar value so i think about these columns will have to removed

. I use value_counts() function to check all categorical columns and some continuous columns that how many types of elements are in these columns ,

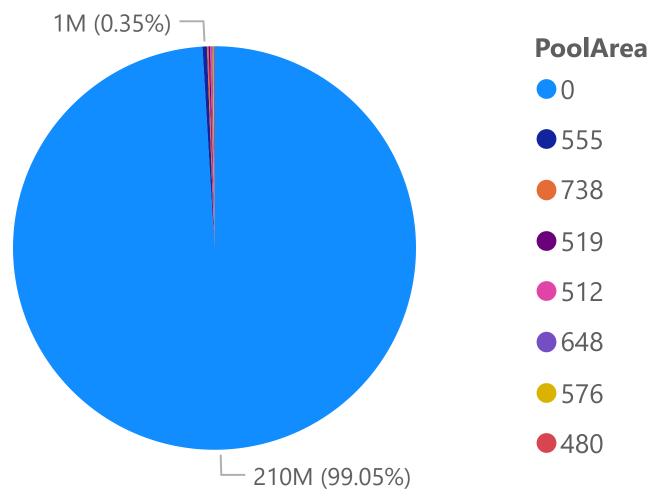
then (Street, Utilities, Condition2, Heating, RoofMatle,GarageCond) In these all Attributes have more than 1400 values are similar out of 1460 values, so we will have to be drop these all columns before trained the Model

plotting pie plot on these columns to check how many
percentage are same type of value

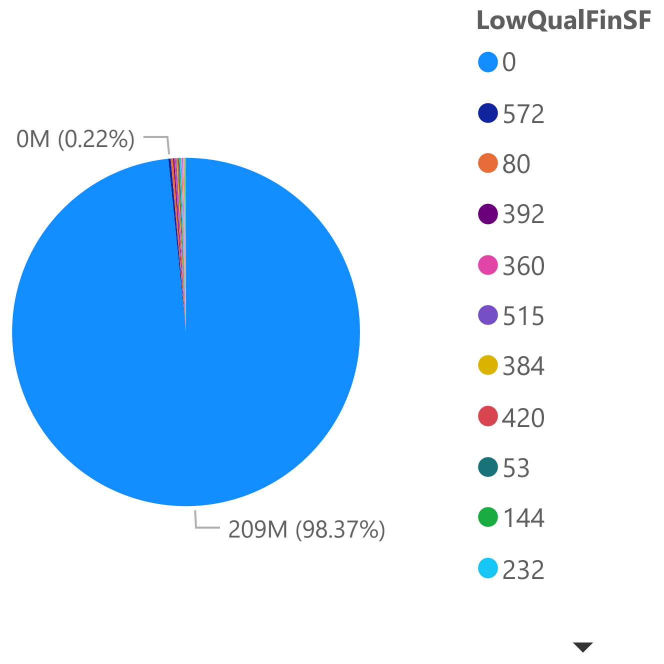
Utilities = 99 % similar value
Street = 99.6 % similar value
Condition2 = 99 % similar value
Heating = 99 % similar value
RoofMatl = 99 % similar value
GarageCond = 99 % similar value
PoolArea = 99% are similar value

. Used pieplot to ensure this , I have plotted it on the bottom page

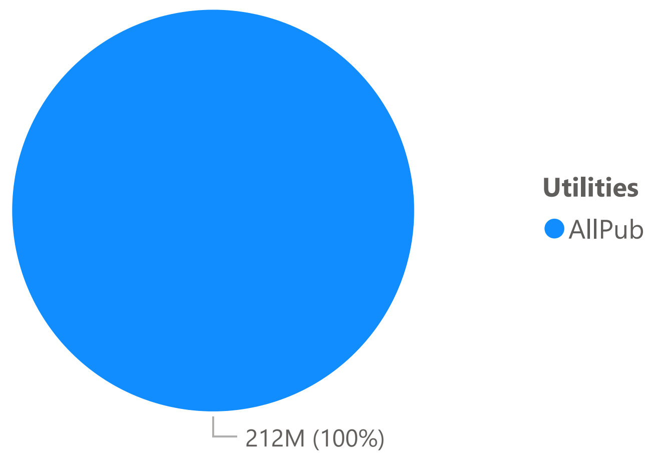
SalePrice by PoolArea



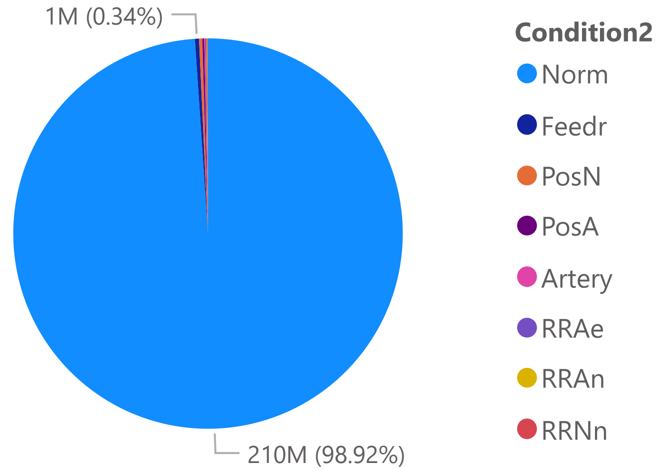
SalePrice by LowQualFinSF



SalePrice by Utilities



SalePrice by Condition2



. (Street, Utilities, Condition2, Heating, RoofMatl, GarageCond , PoolArea, LowQualFinSF,.) these all Attributes have more than 1430 values are similar out of 1460 values, so we will have to be drop these all columns before trained the Model

. The columns which are showing in this pieplot whose 98 percent value is same

. so I decide to dropped these columns because these columns giving no any information or which 1 to 2 percent value of these columns removed by Outliers method when using Zscore

visualize the data using Matplotlib & seaborn

-Now I started the visualization plot like Countplot , scatterplot, Lineplot, barplot

- plotting countplot on Categorical Variables, plotting scatterplot on Countinuous columns, in Barplot we taking both Categorical & continuous columns

. plotting heatmap on dataset_correlation

- When I have done all the visualization part

.I found that some Attributes are a little important or no any relation with 'SalePrice'

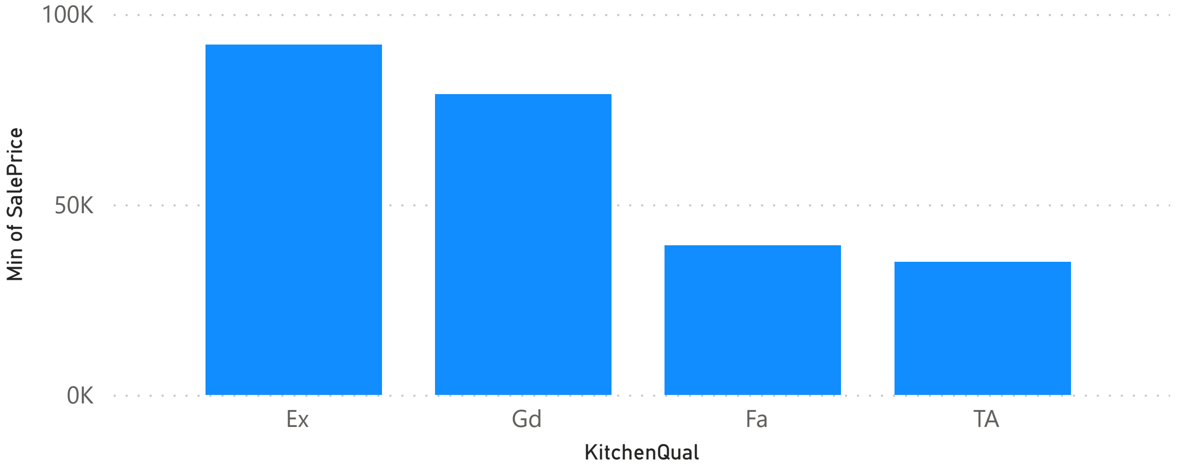
. some Attributes are such types that only one value contain in whole datapoints so no any use point to keep them because those Attributes give zero percent information to predict the SalePrice , then I dropped it

According to EDA / Data_Preprocessing and in this process.

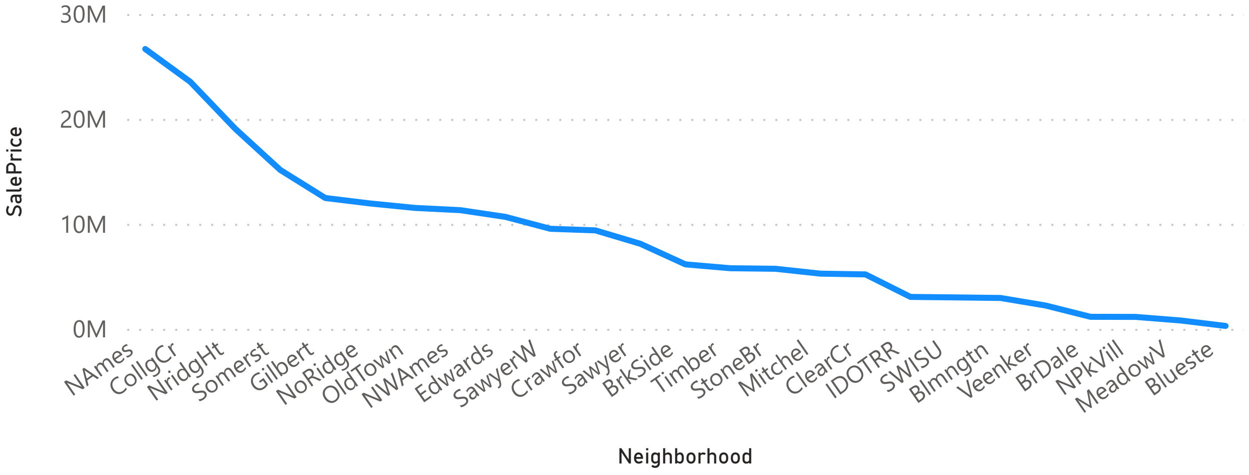
1. I used many methods like first Data-Cleaning followed by Data-Visualization(lineplot, countplot, barplot), checking dataset Correlation,
   Then I found that there are some such Attributes which are more important for predicting the Target-Attribute('SalePrice') like

(lotArea,OverallQual,OverallCond,YearBuilt,RoofStyle,Exterior1st,ExterQual,BsmtQual,BsmtCond,BsmtFinType1,BsmtFinType2,
HeatingQC,GrLivArea,FullBath,KitchenQual,GarageArea,GarageCars,GarageFinish,GarageQual,OpenPorchSF,.)

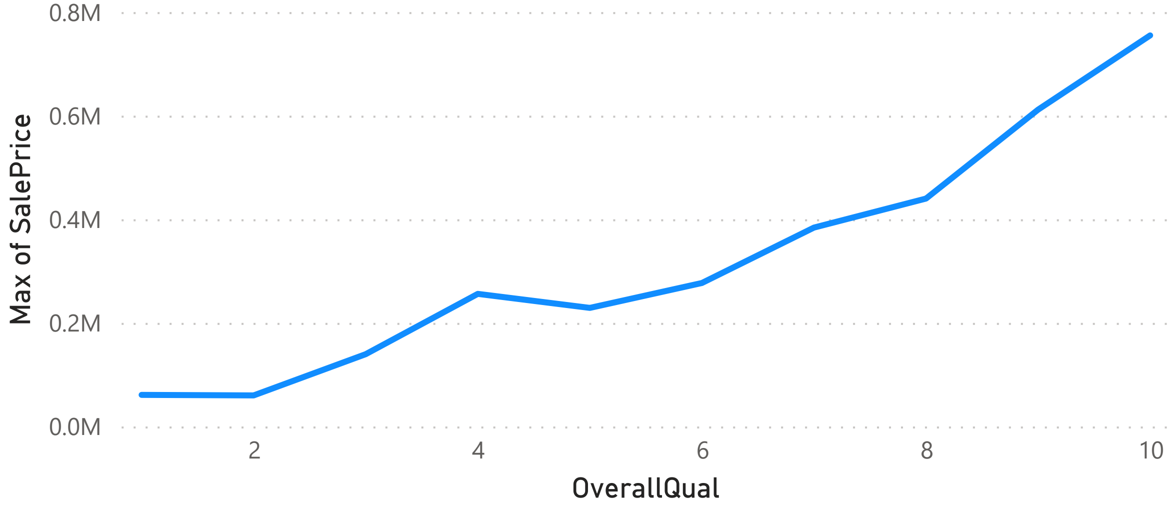
Min of SalePrice by KitchenQual



SalePrice by Neighborhood



Max of SalePrice by OverallQual



. These all columns of this dataset are example of positive correlated with Target

. These columns are important to predict the Target

we have written all about these columns on above page had to show some example only through visualization

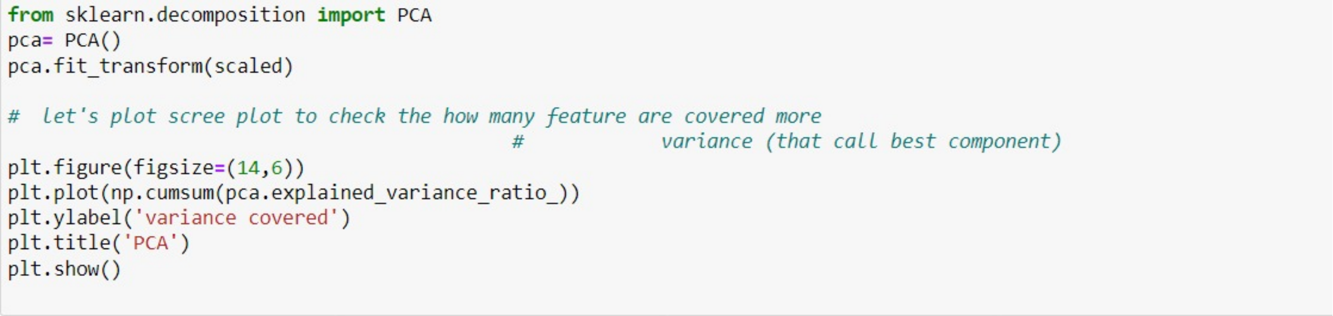
. If such Attributes are more in the REGRESSION-PROBLEM , then the accuracy of prediction value is good

2. we use Label_Encoding on ordinal type of Categorical Attributes & Onehot-Encoding on Nominal type columns,
   But the model was made by Label-Encoding the nominal Attributes there was no difference in accuracy

3. In this dataset some columns are very huge skewed then we use np.log1p & np.sqrt to remove skewness
4. After removing skewness we use Boxplot for checking Outlier, boxplot show OUtliers in most of columns then we use
   Zscore method to confirm its
5. use VIF to solve Multicollinearity_Problem then we found that some columns are multicollinearity so we removed its

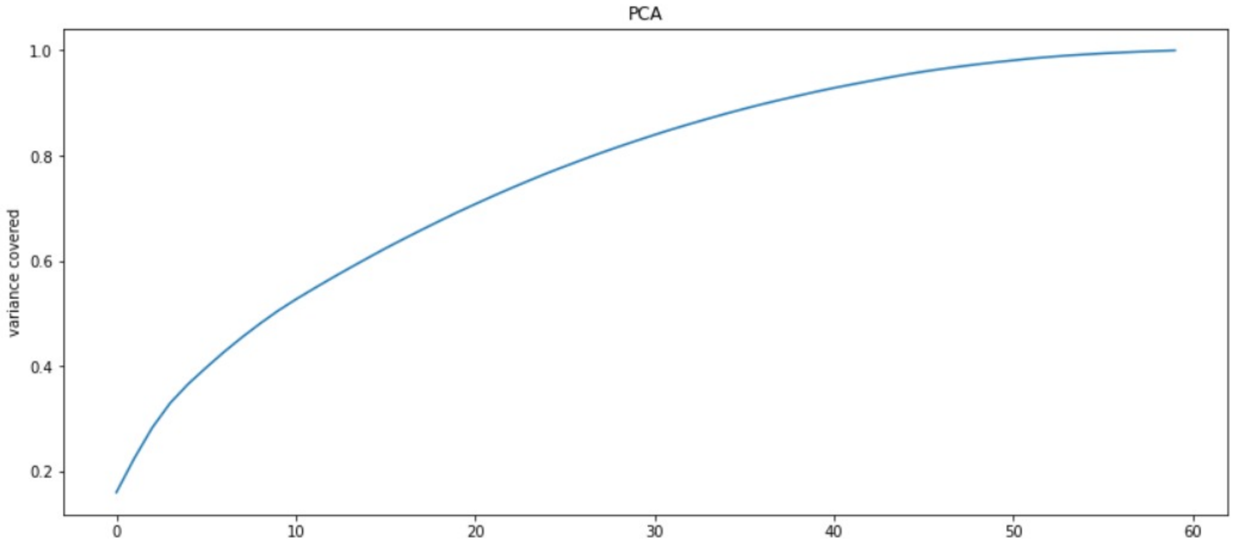
6. Seperate the dataset into two part Input-data, Target-Attribute
7. Scaling the Input-data with StandardScaler

# Finally after Data-Preprocessing , 60 Attributes were left on which the model was to be fitted,
# After this , PCA was used on the scaled data, then 55 PCA_componenets covered the 90 percent of variance of data
# But we don't use PCA



```
from sklearn.decomposition import PCA
pca= PCA()
pca.fit_transform(scaled)

#  let's plot scree plot to check the how many feature are covered more
                                                               #         variance (that call best component)
plt.figure(figsize=(14,6))
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.ylabel('variance covered')
plt.title('PCA')
plt.show()
```



MACHINE LEARNING MODEL

1. First we do Train_test_split on input-data & target-attribute then data split into train & test data , fit the model on train data then test the model accuracy on test data

2. I used LinearRegression, RandomForest-Regression, AdaBoost-Regressor & XGBRegressor to fit the models and after train the models checking r2_score so the highest r2_score = 90 percent , obtained from RandomForest-Regressor & XGB-Regressor . In machine learning in the REGRESSION-problem, the RandomForest r2_score on test data is more compare than another algorithms

3. applying cross_validation with Kfold on all four model then we get maximum cross_validation_score from RandomForest-Regressor = 89 percent

4. We done Hyperparameter tunning over RandomForest_Model because 'r2_score' of RandomForest was highest compare to all model & 'root_mean_squared_error' was less compare to another model
. we get highest Cross_validation_score from RandomForest_Model
HyperTunned_Model r2_score = 91%

5. . we make HyperParameter tunning on Randomforest-Model and last predict the test data with this Model

6. we used all metrics like(MSE, r2_score, AbsoluteMSE, RMSE) in built models for checking that which models performed more compare to another on this dataset

7. Import pickle module save the hypertunned model