PREDICTING HOUSE AND PRICES USING ML

PHASE 3 : DEVELOPMENT PART

PROJECT : LOADING ANDPRE-PROCESSING DATASETUSING ML

» **Data cleaning** can be applied to filling in missing values, remove noise, resolving inconsistencies, identifying and removing outliers in the data.

» **Data integration** merges data from multiple sources into a coherent data store, such as a data warehouse.

» **Data transformations**, such as normalization, may be applied. For example, normalization may improve the accuracy and efficiency of mining algorithms involving distance measurements.

» **Data reduction** can reduce the data size by eliminating redundant features, or clustering, for instance

```
import warnings

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from operator import itemgetter

from sklearn.experimental  import
enable_Iterative_imputer

from sklearn.impute import IterativelImputer

from sklearn.preprocessing import OrdinalEncoder

from category_encoders.target_encoder

import TargetEncoder

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import (Gradient
```

~~~
BoostingRegressor, GradientBoostingClassifier)

import xgboost

miss_df = find_missing_percent(train)

```Displays columns with missing value

s```

Display(miss_df[miss_df[‘PercentMissing’]>0.0)

print("\n")

print (f "Number of columns with missing

values:{str(miss_df[miss_df['PercentMissing']>0.0].s
hape[6])}")
~~~

	ColumnName	TotalMissingVals	PercentMissing
3	LotFrontage	259.0	
6	Alley	1 369.0	93.77
25	MasVnrType	8.0	0.55
26	MasVnrArea	8.0	0.55
30	BsmtQual	37.0	2.53
31	BsmtCond	37.0	2.53
32	BsmtExposure	38.0	2.60
33	BsmtFinType1	37.0	2.53
35	BsmtFinType2	38.0	2.60
42	Electrical	1 .0	O. 07
57	FireplaceQu	690.0	47.26
58	GarageType	81.0	5.55
59	GarageYrBlt	81.0	5.55
60	GarageFinish	81.0	5.55
63	GarageQual	81.0	5.55
64	GarageCond	81.0	5.55

72	PoolQC	1453.0	99.52
73	Fence	1 179.0	80.75
74	MiscFeature	1406.0	96.30

Drop the columns which have more than 70% of missing values

In [5]:

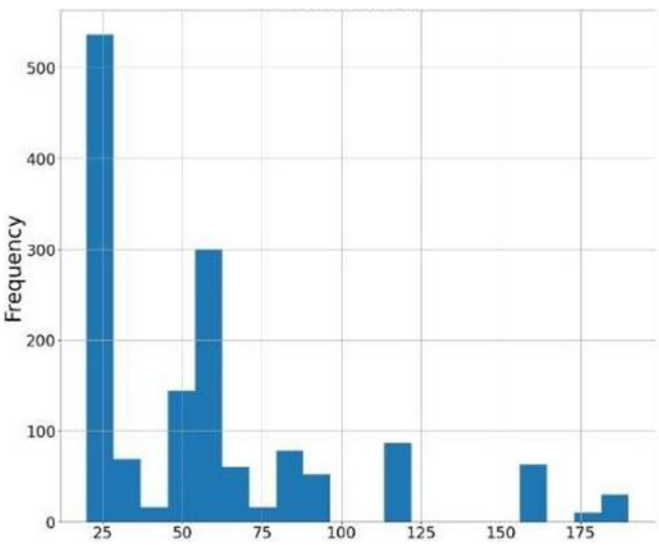
```
drop_cols = miss_df[miss_df['PercentMissing']
>70.0].ColumnName.tolist()
print (f"Number of columns with more than 70%:
{len(drop_cols)}")
train = train.drop(drop_cols,axis=1)
test = test.drop(drop_cols, axis =1)
```

```
miss_df=miss_df[miss_df['ColumnName’].isin(train.
colums)]
"""Columns to Impute
impute_cols =
miss_df[miss_df['TotalMissingVals’]>0.0].ColumnNa
me.tolist()
miss_df[miss_df[ 'TotalMissingVals']>0.
0]
```

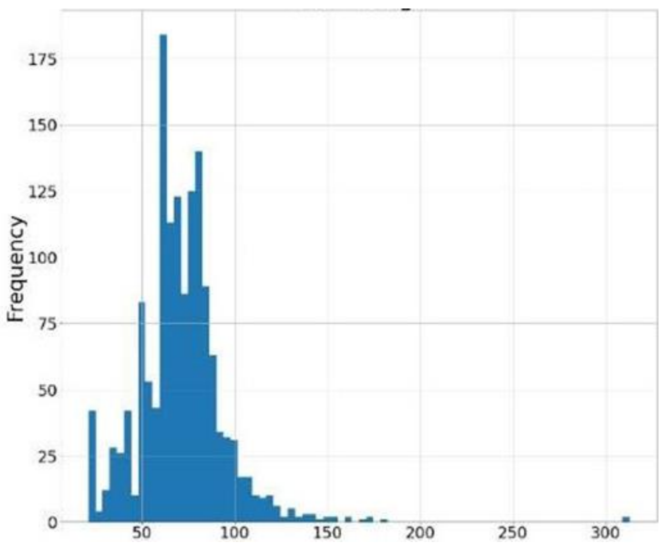
```
Number of columns with more than 70%:
4
```

	ColumnName	TotalMissingVals	PercentMissing
3	LotFrontage	259.0	
25	MasVnrType	8.0	0.55
26	MasVnrArea	8.0	0.55
30	BsmtQual	37.0	2.53
31	BsmtCond	37.0	2.53
32	BsmtExposure	38.0	2.60
33	BsmtFinType1	37.0	2.53
35	BsmtFinType2	38.0	2.60
42	Electrical	1.0	
57	FireplaceQu	690.0	47.26
58	GarageType	81.0	5.55
59	GarageYrBlt	81.0	5.55
60	GarageFinish	81.0	5.55
63	GarageQual	81.0	5.55
64	GarageCond	81.0	5.55

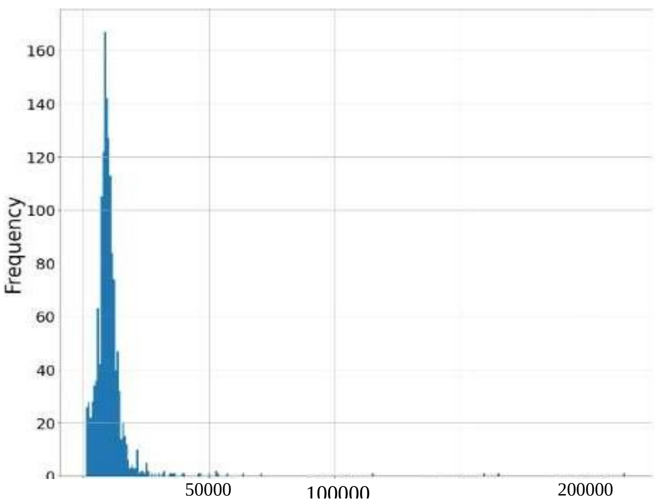
MSSubClass



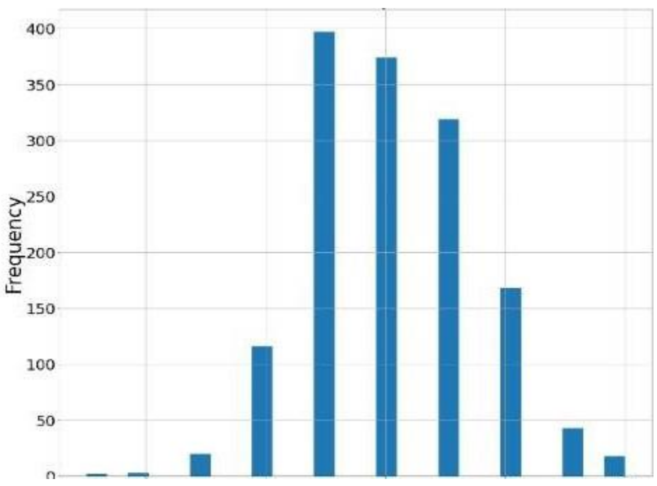
LotFrontage



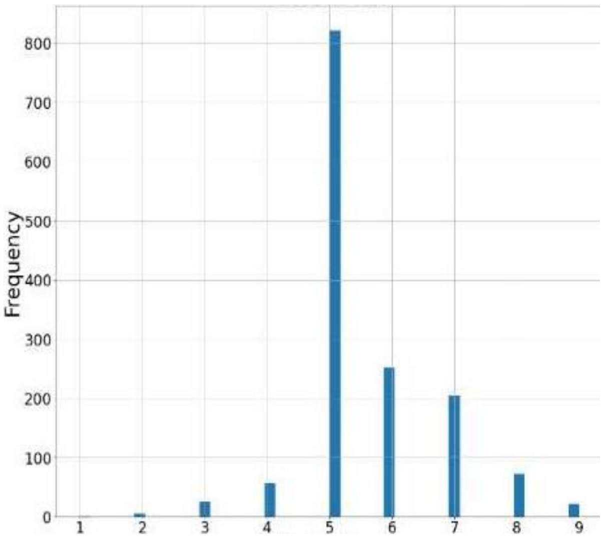
LotArea



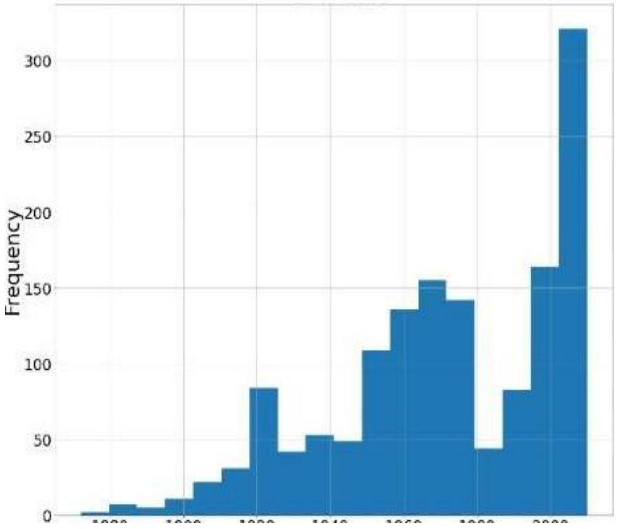
OverallQual



OverallCond



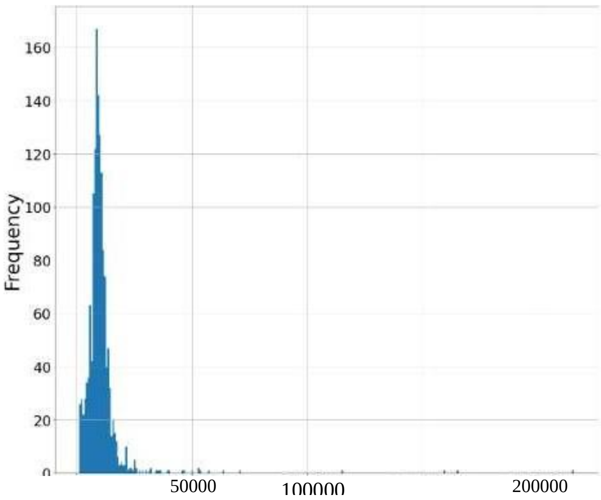
YearBuilt



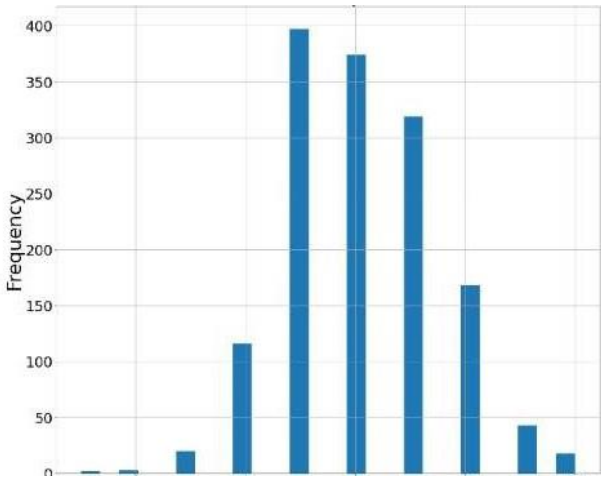
~~~
def fit_model(x_train,y_train, model):
“””
Fits x_train to y_train for the giv
en
        model.
        “””
        Model.fit(x_train,y_train)
        return model

```Xtreme Gradient Boosting Regressor```
Model=xgboost.XGBRegressor(objective="reg:squarederror", random_state=42)
model = fit_model(x_train,y_train, model)
'''Predict the outcomes'''
predictions = model.predict(test)
~~~

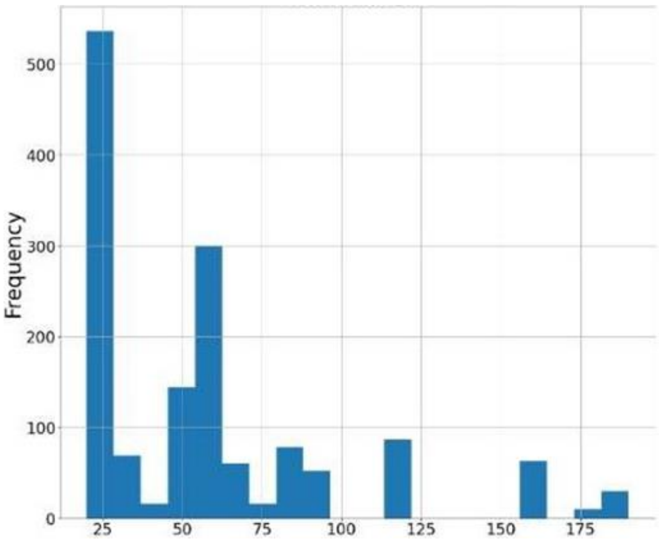
MSSubClass



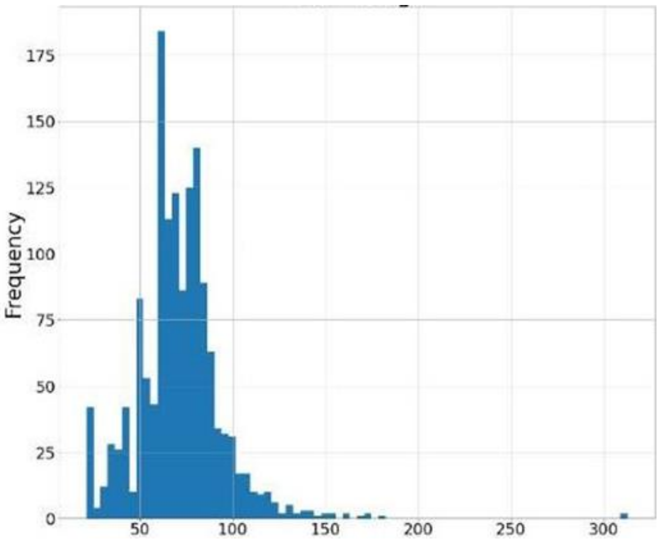
LotFrontage



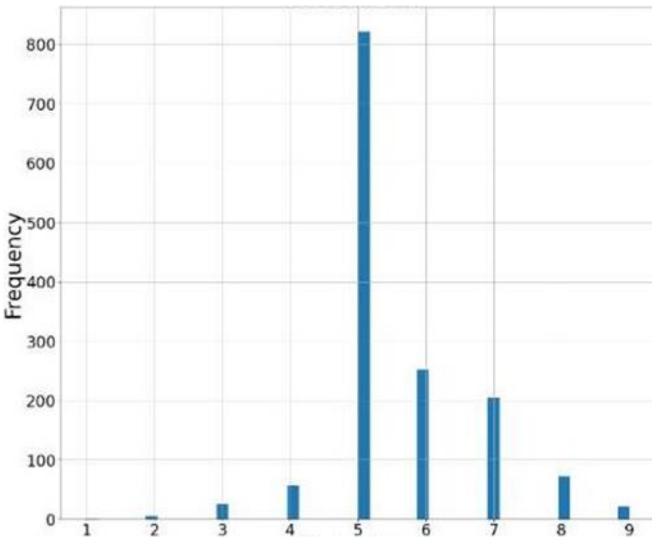
LotArea



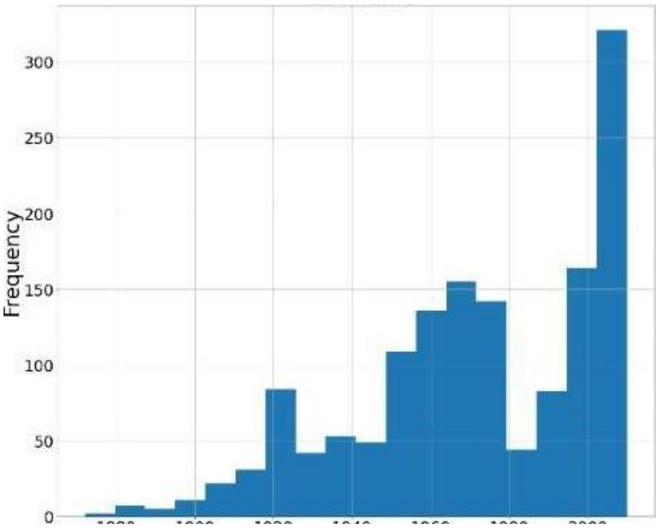
OverallQual



OverallCond



YearBuilt

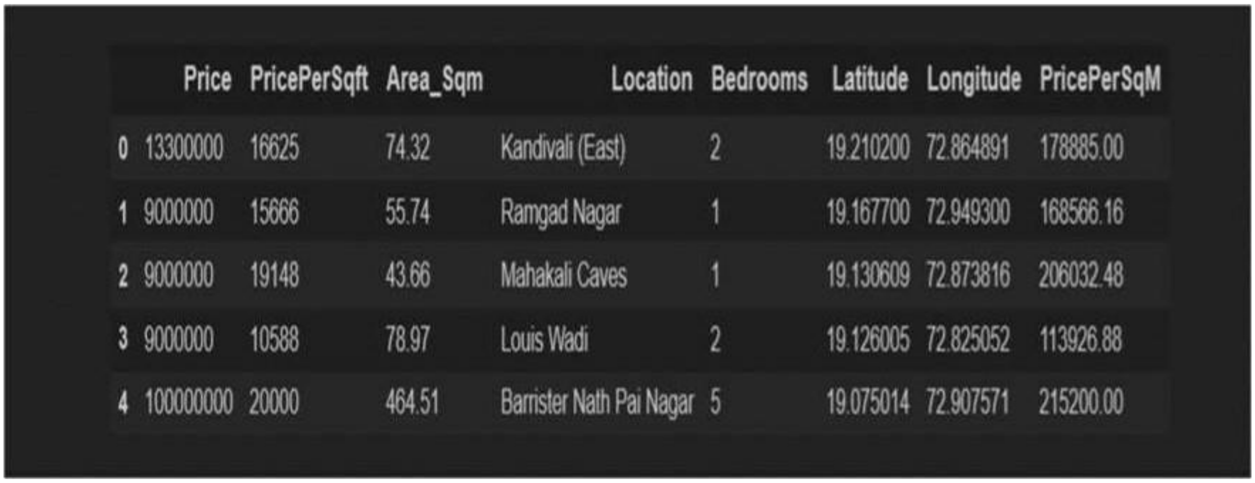


DATASET

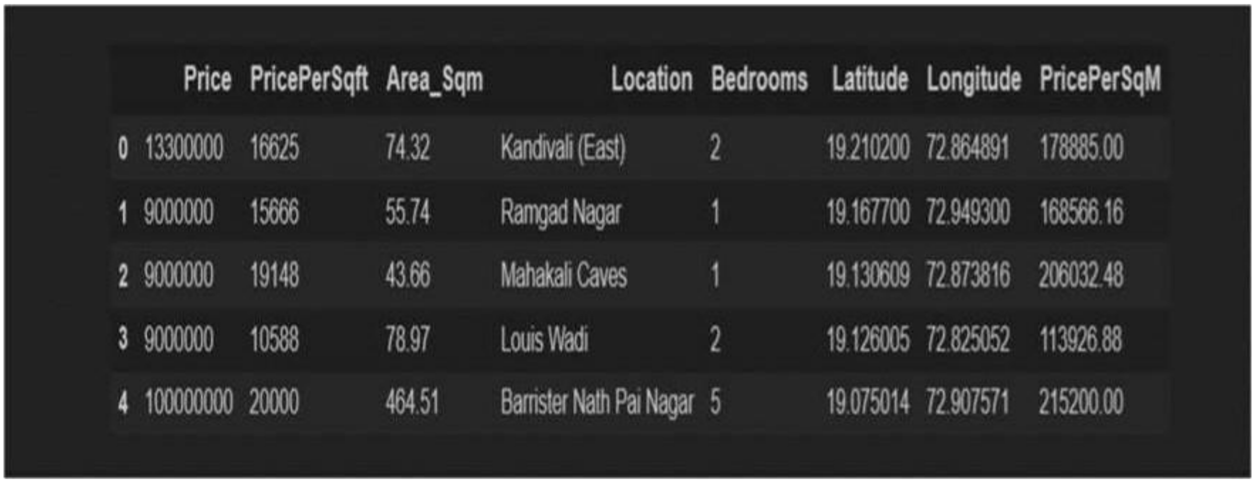
Here we have web scrapped the Data from 99acres.com website which is one of the leading real estate websites operating in INDIA.

Our Data contains Bombay Houses only.

Dataset looks as follows-



	Price	PricePerSqft	Area_Sqm	Location	Bedrooms	Latitude	Longitude	PricePerSqM
0	13300000	16625	74.32	Kandivali (East)	2	19.210200	72.864891	178885.00
1	9000000	15666	55.74	Ramgad Nagar	1	19.167700	72.949300	168566.16
2	9000000	19148	43.66	Mahakali Caves	1	19.130609	72.873816	206032.48
3	9000000	10588	78.97	Louis Wadi	2	19.126005	72.825052	113926.88
4	100000000	20000	464.51	Barrister Nath Pai Nagar	5	19.075014	72.907571	215200.00



	Price	PricePerSqft	Area_Sqm	Location	Bedrooms	Latitude	Longitude	PricePerSqM
0	13300000	16625	74.32	Kandivali (East)	2	19.210200	72.864891	178885.00
1	9000000	15666	55.74	Ramgad Nagar	1	19.167700	72.949300	168566.16
2	9000000	19148	43.66	Mahakali Caves	1	19.130609	72.873816	206032.48
3	9000000	10588	78.97	Louis Wadi	2	19.126005	72.825052	113926.88
4	100000000	20000	464.51	Barrister Nath Pai Nagar	5	19.075014	72.907571	215200.00

Thank you