NAME OF THE PROJECT

HOUSING PROJECT

SUBMITTED BY

SONAL MISAL

INTRODUCTION –

House is a basic living thins for human being. House is a very important and needy thing for every human being in the world. In this housing project we have a dataset which consist of different features. In this project we have to predict the price for a house.

On the basis of given features we have to predict the sale price of house. Price of the house is one of the most important thing for commom man.

ANALYTICAL PROBLEM FRAMING –

To predict the sale price for a house we have to follow some steps to find the best model for price prediction. First, we have to import some libraries and then read the data. After reading the data we got dataset of housing project with the number of rows and columns. This dataset consists of 1168 rows and 81 columns.

First, we have to find is there any null values in dataset and handling those null values. If objective column has a null values, then it fills with mode() and if numerical column has null values, then it fills with mean of that column. After filling all the null values in dataset next step is visualization.

Visualization is plot between feature columns and the target column. Description is the important process in machine learning program. Description helps to cleaning the data. Following plot shows the description for the dataset.

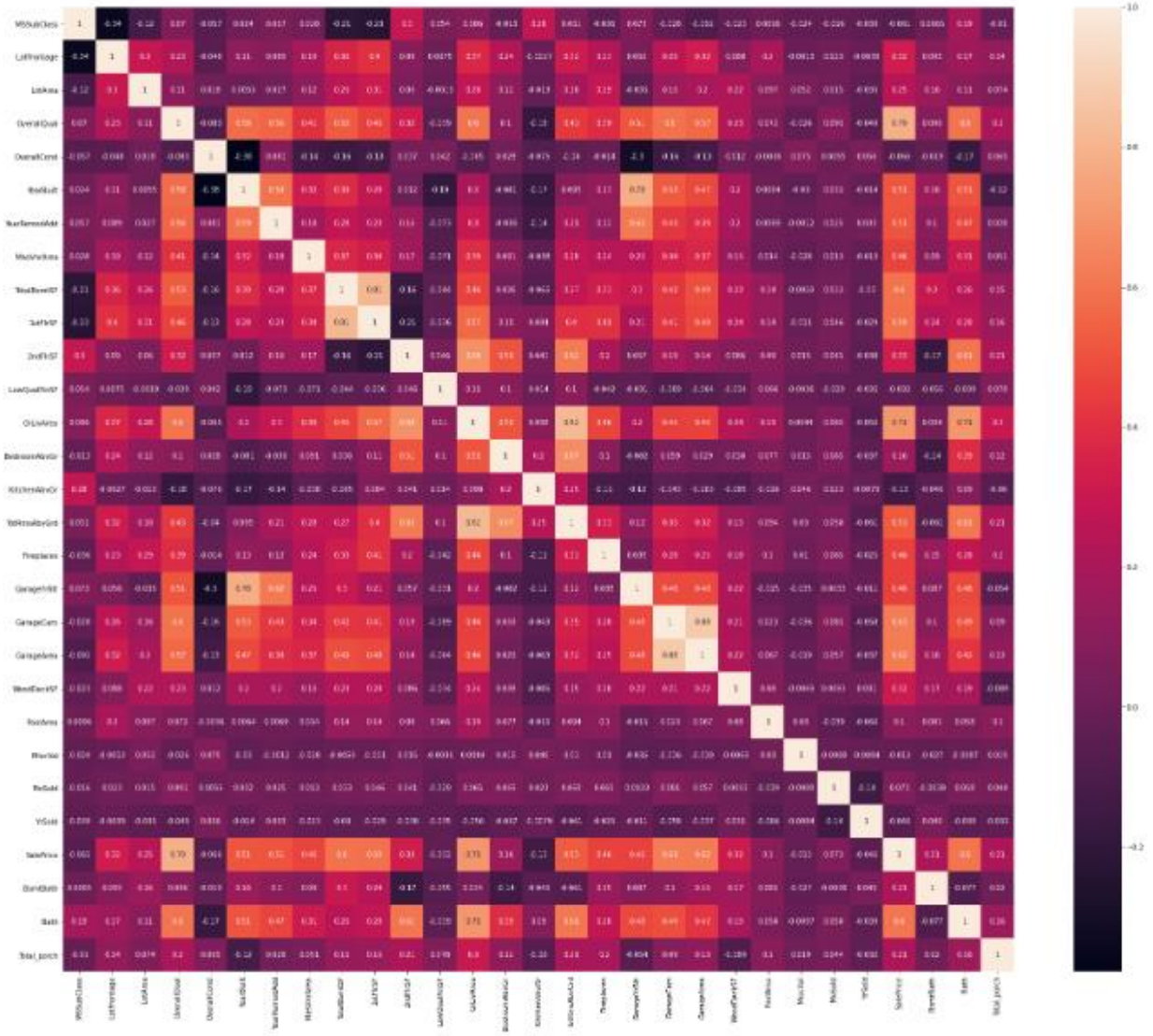
```
df.describe()
```

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	TotalBsmtSF	1stFlrSF	2ndFl
count	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000
mean	56.767979	70.988470	10484.749144	6.104452	5.595890	1970.930651	1984.758562	102.310078	1061.095034	1169.860445	348.826
std	41.940650	22.437056	8957.442311	1.390153	1.124343	30.145255	20.785185	182.047152	442.272249	391.161983	439.696
min	20.000000	21.000000	1300.000000	1.000000	1.000000	1875.000000	1950.000000	0.000000	0.000000	334.000000	0.000
25%	20.000000	60.000000	7621.500000	5.000000	5.000000	1954.000000	1966.000000	0.000000	799.000000	892.000000	0.000
50%	50.000000	70.988470	9522.500000	6.000000	5.000000	1972.000000	1993.000000	0.000000	1005.500000	1096.500000	0.000
75%	70.000000	79.250000	11515.500000	7.000000	6.000000	2000.000000	2004.000000	160.000000	1291.500000	1392.000000	729.000
max	190.000000	313.000000	164660.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	6110.000000	4692.000000	2065.000

Description shows the count, mean, standard deviation, minimum value, maximum value etc. of each column.

After finding the description of dataset we have to find the correlation of dataset. Correlation is the important process in machine learning program. It shows that collinearity between the feature columns and target column.

Following plot shows the collinearity between the columns.



We can see the collinearity of columns with each other and target column. Some columns are positively correlated whereas some are negatively correlated.

After finding correlation preprocessing process is performed. Preprocessing is the very important process in program. Finding skewness, Outliers, VIF and Data conversion are the process included in preprocessing.

MODEL DEVELOPMENT AND EVALUATION

Training Process

```
from sklearn.linear_model import LinearRegression,Ridge,Lasso
lr = LinearRegression()

from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.neighbors import KNeighborsRegressor

from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_absolute_error,mean_squared_error
```

Random State

```
for i in range(0,100):
    x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=.30,random_state=i)
    lr.fit(x_train,y_train)
    pred_tr = lr.predict(x_train)
    pred_ts = lr.predict(x_test)
    print(f"At random state {i},the training accuracy is : {r2_score(y_train,pred_tr)}")
    print(f"At random state {i},the testing accuracy is : {r2_score(y_test,pred_ts)}")
    print("\n")
```

r2 score is 85.30.

Cross Validation Score

```
tr_accu = r2_score(y_train,pred_tr)
ts_accu = r2_score(y_test,pred_ts)

from sklearn.model_selection import cross_val_score
for j in range(2,20):
    cv_score = cross_val_score(lr,x,y,cv=j)
    cv_mean = cv_score.mean()
    print(f"At cross fold{j} the cv score {cv_mean} and accuracy score for training is {tr_accu} and accuracy for testing is
    print("\n")
```

At cross fold 8 we got our best cv score : 81.31.

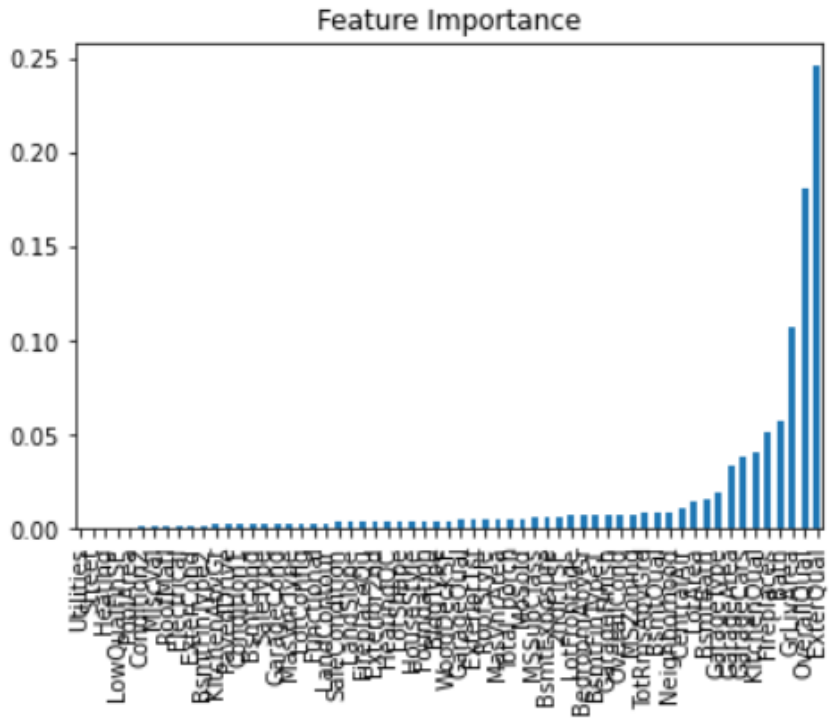
Model Testing

```
def train(model,x,y):
    model.fit(x,y)
    pred = model.predict(x)
    cv_score = cross_val_score(model,x,y,cv=8)
    cv_score = np.abs(np.mean(cv_score))
    print('Model Report')
    print('MSE',mean_squared_error(y,pred))
    print('CV',cv_score)
```

```
model = ExtraTreesRegressor()
train(model,x,y)
coef = pd.Series(model.feature_importances_,x.columns).sort_values()
coef.plot(kind = 'bar',title = 'Feature Importance')
```

```
Model Report
MSE 0.0
CV 0.8583491262158044
```

```
]: <AxesSubplot:title={'center':'Feature Importance'}>
```



Extra Tree Regressor has the highest CV score 85.83 than other models so we choose Extra tree regressor as our final model.