**REAL ESTATE PRICE PREDICTION**

1. **DATA CLEANING**

PROCESS

1. Import the necessary libraries (numpy.pandas,matplotlib)
2. Import/read csv file
3. Group it using one parameter
4. Drop unnecessary columns
5. Check total null values
6. Check types of data in size column by using unique()
7. Right a lambda function to retrieve only the preferred data from size and save as bhk
8. Check unique values in bhk column
9. Check anomaly or outlier using any simple check function
10. Normalize the values in total\_sqft column by using is\_float function and convert\_sqft\_to\_num function

FUNCTIONS AND KEYWORDS

* Pd.read\_csv() – read csv file
* Head() – shows 5 rows on top
* Shape() – shows dimensions (rows,cols)
* Groupby() – sort and arrange
* Agg() – aggregate
* Drop() – drops columns
* Isnull() – check for null values
* Sum() – sums up
* Dropna() – drops rows with null values
* Lambda function – to create simple and small functions without name
* Apply – apply a function
* Copy() – copy a df to other df
* Loc – display the specified rows

1. **FEATURE ENGINEERING**

PROCESS

1. Create a new column called price per sqft from using total sqft and price columns
2. Check number of unique locations. If it is more then
3. Use lambda function to strip empty rows, then
4. Use groupby and sort the locations which number of locations in descending order
5. Check the number or locations less that 10 frequency and then store it in a variable
6. Create lambda function to group those locations to ‘others’

FUNCTIONS AND KEYWORDS

1. Sort\_values() – sort function
2. Strip() – to remove extra whitespaces
3. **OUTLIER REMOVAL**

PROCESS

1. Remove price per sqft outliers by using a function which removes values which are less than the mean-std and more than mean+std
2. Plot in scatterplot and check for bhk anomaly by using each individual locations. Then find cost of bhk3 less than bhk2 and remove the outliers
3. It uses a function which which assigns mean,std and count for each bhk and compares
4. Remove no of bathrooms outlier by using logic to remove values if bathroom no is greater than bathrooms+2.

FUNCTIONS

* New\_df = pd.DataFrame() ------- to initialize dataframe
* Concat()---------- join dfs
* matplotlib.rcParams['figure.figsize'] = (15,10) ---------------assiging size
* plt.scatter()---------------scatterplot
* xlabel and ylabel ---------label text
* legend()----- ref box
* title ---heading
* get() --- fetch funct
* plt.hist() ----- histogram

1. **MODEL TRAINING**

PROCESS

1. To represent the locations data use Hot Encoding technique create dummies
2. Concat d10 and dummies after dropping ‘other’ from dummies
3. Drop location column and create d12
4. Create variable x with independent variable columns by dropping price column
5. Create var y with price values from d12
6. Import train\_test\_split from sklearn.model\_selection to split training and test data
7. Create the linear regression model and use fit and score functions
8. Import shufflesplit and cross val score from sklearn.mode\_selection and use then to test the models score using 5 splits
9. Import GridSearchCV and train three models linear regression , lasso and decision tree regressor
10. Create a dictionary with for all the three models
11. Create scores array and use gridsearchcv to get the model with best scores and display the scores and models and parameters in a table
12. Write a function to predict the price of house using linear regression with inputs location,sqft,bath,bhk
13. Create location index variable and inputarray
14. Call the predict function and check output
15. Export the model by converting it into pickle file
16. And export the column names in json file.

FUNCTIONS AND KEYWORDS

* ONE HOT ENCODING – it converts categorical data and represents each as binary vectors
* Get\_dummies() – to convert the categorial column into dummies
* Train\_test\_split – to split the dataset into training and testing
* LinearRegression model --- ml algorithm
* Fit() – to train the model
* Score() – evaluate the score by testing
* ShuffleSplit() – shuffles the dataset and trains
* Cross\_val\_score – score calculator
* GridSearchCV – this is used to train and test multiple models with same data and select the best
* Lasso and DecisionTreeRegressor ---ml algorithms
* Np.where() ------- conditional indexing
* Zeros() – create arrays of zeros using methods
* Pickle file – exporting of python files
* Json – exporting column names in this case
* Pickle.dump(‘filename’,’wb’’) --- where dump is to store the file and wb is write binary
* With open --- used to check is the file is safely closed by opening it
* .write(json,dump()) -------- write and store the json file