# HOUSE PRICE PREDICTION

@KING_COUNTRY

TALAGAPU ARCHANA
RGUKT-SKLM

# KING_COUNTRY HOUSE PRICE PREDICTION

**A PROJECT REPORT**

*Submitted by*

**Talagapu.Archana&S180581**

**OF**

**Computer Science & Engineering**


**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES**

**ETCHERLA-SIRKAKULAM(DIST)**

**ANDHRA PRADESH-532410**

# TABLE OF CONTENTS

## Contents

# Abstract

House prices increase every year, so there is a need for a system to predict house prices in the future. House price prediction can help the developer determine the selling price of a house and can help the customer to arrange the right time to purchase a house. There are some factors that influence the price of a house which include number ofbedrooms,number of bathrooms,conditions,sqft_living,sqft_lot and etc. This research aims to predict house prices in KING_COUNTRY based on REGRESSION.

# Keyword

Kingcountry House Price Prediction (Kc_HPP), Pandas, NumPy, Matplotlib, Exploratory Data Analysis (EDA), Data Cleaning,Outlier detection, Dimensionality Reduction, Data Visualization,Multiple linear Regression,Polynomial Regression,Random forest etc.

# TOOLS

| | |
|---|---|
| pandas | https://pandas.pydata.org/ |
| numpy | https://numpy.org/doc/ |
| seaborn | https://seaborn.pydata.org/ |
| matplotlib | https://matplotlib.org/3.3.3/contents.html |
| sklearn | https://scikit-learn.org/stable/ |

# Objective

The objective was to forecast the price of a specific house based on market pricing while accounting for various "features".

# Applications

1.Real Estate

2.Own House Dream

# Goal

To predict the house price in KING_COUNTRY by the help of REGRESSION.The project helps to real estate business and both buyer and seller.This model is also help to buyer's who having the future goal to buy the own house.

# Description

By observing the data, we can know that the **price is dependent on various features** like bedrooms, bathrooms, sqft_living, sqft_lot, Year built etc. The price is also dependent on the location of the house where it is present. The other features like waterfront, view are less dependent on the price. Of all the records, there are **some missing values, which we have to remove so this helps us creating better model.**

# Dataset:

There are 21,613 observation and 21 variables in our dataset. Let's take a closer look at what each variable name represents:

| Variable | Description |
|---|---|
| id | Unique identifier of the house |
| date | Date of sale |
| price | Sell price |
| bedrooms | Number of bedrooms |
| bathrooms | Number of bathrooms. Noninteger values exist due to "1/2 bathrooms" and "3/4 bathrooms" |
| sqft_liv | Size of interior space in square feet |

| Variable | Description |
|---|---|
| sqft_lot | Size of land lot in square feet |
| floors | Number of floors. Noninteger values exist due to "half floor" architecture |
| waterfront | '1' if property has a waterfront, '0' if not |
| view | An index from 0 to 4 of how good the property's view is |
| condition | Condition of the house, ranked from 1 to 5, 5 being the greatest condition |
| grade | Classification by construction material and worksmanship quality. Numeric scale with higher numbers being better. For more information see the King County glossary |
| sqft_above | Square feet above ground |
| sqft_below | Square feet below ground |
| yr_built | Year built |
| yr_renov | Year renovated. '0' if never renovated |
| zipcode | 5 digit zip code |
| lat | Latitude |
| long | Longitude |
| squft_liv15 | Average size of interior space for closest 15 houses, in square feet |
| squft_lot15 | Average size of land lot for closest 15 houses, in square feet |

# Libraries

## Pandas

Pandas is mainly used for **data analysis and associated manipulation of tabular data in DataFrames**. Pandas allows importing data from various file formats such as comma-separated values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel.

## Numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

### Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython etc.

### Seaborn

Seaborn is **a Python data visualization library based on matplotlib**. It provides a high-level interface for drawing attractive and informative statistical graphics. For a brief introduction to the ideas behind the library, you can read the introductory notes or the paper.

### SKlearn

Scikit-learn is **a free machine learning library for Python**. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

Importing dataset:

In [37]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df= pd.read_csv(r'D:\data_science\kc_house_data.csv')
df
```

Out[37]:

| | id | date | price | bedrooms | bathrooms | sqft_living | sqft_lot | floors | waterfront | view | ... | grade | sqft_above | sqft_basement | yr_l |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 7129300520 | 20141013T000000 | 221900.0 | 3 | 1.00 | 1180 | 5650 | 1.0 | 0 | 0 | ... | 7 | 1180.0 | 0 | 1 |
| 1 | 6414100192 | 20141209T000000 | 538000.0 | 3 | 2.25 | 2570 | 7242 | 2.0 | 0 | 0 | ... | 7 | 2170.0 | 400 | 1 |
| 2 | 5631500400 | 20150225T000000 | 180000.0 | 2 | 1.00 | 770 | 10000 | 1.0 | 0 | 0 | ... | 6 | 770.0 | 0 | 1 |
| 3 | 2487200875 | 20141209T000000 | 604000.0 | 4 | 3.00 | 1960 | 5000 | 1.0 | 0 | 0 | ... | 7 | 1050.0 | 910 | 1 |
| 4 | 1954400510 | 20150218T000000 | 510000.0 | 3 | 2.00 | 1680 | 8080 | 1.0 | 0 | 0 | ... | 8 | 1680.0 | 0 | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 21608 | 263000018 | 20140521T000000 | 360000.0 | 3 | 2.50 | 1530 | 1131 | 3.0 | 0 | 0 | ... | 8 | 1530.0 | 0 | 2 |
| 21609 | 6600060120 | 20150223T000000 | 400000.0 | 4 | 2.50 | 2310 | 5813 | 2.0 | 0 | 0 | ... | 8 | 2310.0 | 0 | 2 |
| 21610 | 1523300141 | 20140623T000000 | 402101.0 | 2 | 0.75 | 1020 | 1350 | 2.0 | 0 | 0 | ... | 7 | 1020.0 | 0 | 2 |
| 21611 | 291310100 | 20150116T000000 | 400000.0 | 3 | 2.50 | 1600 | 2388 | 2.0 | 0 | 0 | ... | 8 | 1600.0 | 0 | 2 |
| 21612 | 1523300157 | 20141015T000000 | 325000.0 | 2 | 0.75 | 1020 | 1076 | 2.0 | 0 | 0 | ... | 7 | 1020.0 | 0 | 2 |

21613 rows × 21 columns

## Data Analysis

Data Analysis. Data Analysis is **the process of systematically applying statistical and/or logical techniques to describe and illustrate, condense and recap, and evaluate data.**

### .info()

8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   id              21613 non-null  int64  
 1   date            21613 non-null  object 
 2   price           21613 non-null  float64
 3   bedrooms        21613 non-null  int64  
 4   bathrooms       21613 non-null  float64
 5   sqft_living     21613 non-null  int64  
 6   sqft_lot        21613 non-null  int64  
 7   floors          21613 non-null  float64
 8   waterfront      21613 non-null  int64  
 9   view            21613 non-null  int64  
 10  condition       21613 non-null  int64  
 11  grade           21613 non-null  int64  
 12  sqft_above      21611 non-null  float64
 13  sqft_basement   21613 non-null  int64  
 14  yr_built        21613 non-null  int64  
 15  yr_renovated    21613 non-null  int64  
 16  zipcode         21613 non-null  int64  
 17  lat             21613 non-null  float64
 18  long            21613 non-null  float64
 19  sqft_living15   21613 non-null  int64  
 20  sqft_lot15      21613 non-null  int64  
dtypes: float64(6), int64(14), object(1)
memory usage: 3.5+ MB
```

Cheack duplicated values and shape of the dataset etc:

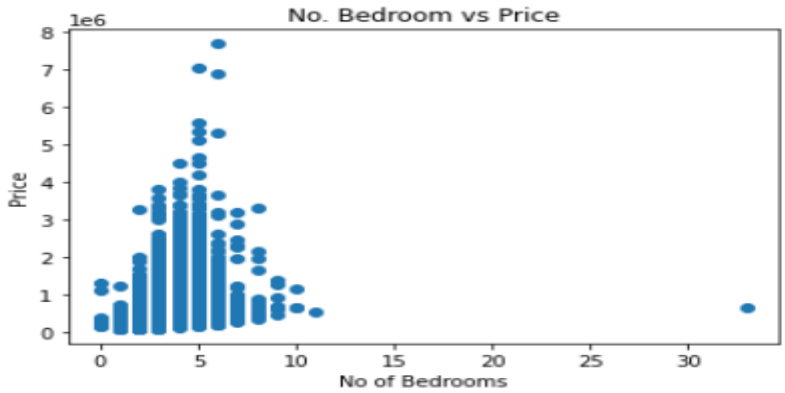
```
: sum(df.duplicated())
: 0
: df.shape
: (21613, 21)
: df.describe().transpose()
:
```

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| id | 21613.0 | 4.580302e+09 | 2.876566e+09 | 1.000102e+06 | 2.123049e+09 | 3.904930e+09 | 7.308900e+09 | 9.900000e+09 |
| price | 21613.0 | 5.400881e+05 | 3.671272e+05 | 7.500000e+04 | 3.219500e+05 | 4.500000e+05 | 6.450000e+05 | 7.700000e+06 |
| bedrooms | 21613.0 | 3.370842e+00 | 9.300618e-01 | 0.000000e+00 | 3.000000e+00 | 3.000000e+00 | 4.000000e+00 | 3.300000e+01 |
| bathrooms | 21613.0 | 2.114757e+00 | 7.701632e-01 | 0.000000e+00 | 1.750000e+00 | 2.250000e+00 | 2.500000e+00 | 8.000000e+00 |
| sqft_living | 21613.0 | 2.079900e+03 | 9.184409e+02 | 2.900000e+02 | 1.427000e+03 | 1.910000e+03 | 2.550000e+03 | 1.354000e+04 |
| sqft_lot | 21613.0 | 1.510697e+04 | 4.142051e+04 | 5.200000e+02 | 5.040000e+03 | 7.618000e+03 | 1.068800e+04 | 1.651359e+06 |
| floors | 21613.0 | 1.494309e+00 | 5.399889e-01 | 1.000000e+00 | 1.000000e+00 | 1.500000e+00 | 2.000000e+00 | 3.500000e+00 |
| waterfront | 21613.0 | 7.541757e-03 | 8.651720e-02 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 1.000000e+00 |
| view | 21613.0 | 2.343034e-01 | 7.663176e-01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 4.000000e+00 |
| condition | 21613.0 | 3.409430e+00 | 6.507430e-01 | 1.000000e+00 | 3.000000e+00 | 3.000000e+00 | 4.000000e+00 | 5.000000e+00 |
| grade | 21613.0 | 7.656873e+00 | 1.175459e+00 | 1.000000e+00 | 7.000000e+00 | 7.000000e+00 | 8.000000e+00 | 1.300000e+01 |

## Outliers

An outlier is **an observation that lies an abnormal distance from other values in a random sample from a population**.

```
2]: #Treating Outliers
    bed=df['bedrooms']
    pr=df['price']
    plt.title("No. Bedroom vs Price")
    plt.xlabel("No of Bedrooms")
    plt.ylabel("Price")
    plt.scatter(bed,pr)
     #no of bedroom=33 is an outlier which can be removed
2]: <matplotlib.collections.PathCollection at 0x1a70a32c5e0>
```



**Check Null values:** A null value in a relational database is **used when the value in a column is unknown or missing**. A null is neither an empty string (for character or datetime data types) nor a zero value (for numeric data types).

```
df.isnull().sum() #2 records in sqft_above is null
```

```
id               0
date             0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
view             0
condition        0
grade            0
sqft_above       2
sqft_basement    0
yr_built         0
yr_renovated     0
zipcode          0
lat              0
long             0
sqft_living15    0
sqft_lot15       0
dtype: int64
```

## Data Preparation

Data preparation and filtering steps can take considerable amount of processing time. Examples of data preprocessing include **cleaning, instance selection, normalization, one hot encoding, transformation, feature extraction and selection**, etc. The product of data preprocessing is the final training set.

- **Removing the Outliers**

```
#treating null values
df=df.replace(np.nan,0)
```

```
df.isnull().sum()
```

```
id               0
date             0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
view             0
condition        0
grade            0
sqft_above       0
sqft_basement    0
yr_built         0
yr_renovated     0
zipcode          0
lat              0
long             0
sqft_living15    0
sqft_lot15       0
dtype: int64
```

- Removing Outliers

```
# check for the extreme values shown in the describe.
df=df[df['bedrooms']<33]
print(df['bedrooms'])
```

```
0         3
1         3
2         2
3         4
4         3
        ..
21608     3
21609     4
21610     2
21611     3
21612     2
Name: bedrooms, Length: 21612, dtype: int64
```

## Spliting Dataset into Training and Testing

Here we will discuss how to split a dataset into Train and Test sets in python.The train-test split is used to estimate the performance of machine learning alogrithms that are applicable for prediction-based alogorithms/Applicatins.This method is a fast and easy procedure to perform such that we can compare our own achine learning model results to machine results.By default,the test set is split into 20% of actual data and the traning set is split into 80% of the actual data.

```
#!pip install scikit-learn
from sklearn.model_selection import train_test_split
x_train, x_test , y_train , y_test = train_test_split(x,y,test_size=0.20,random_state=101)
```

# Model Training

Here I used 3 types of regression.Those are

- Multiple
- Polynomial
- Random Forest

# Multiple Linear Regression

In the previous topic, we have learned about Simple Linear Regression, where a single Independent/Predictor(X) variable is used to model the response variable (Y). But there may be various cases in which the response variable is affected by more than one predictor variable; for such cases, the Multiple Linear Regression algorithm is used.

Moreover, Multiple Linear Regression is an extension of Simple Linear regression as it takes more than one predictor variable to predict the response variable. We can define it as:

*Multiple Linear Regression is one of the important regression algorithms which models the linear relationship between a single dependent continuous variable and more than one independent variable.*

```
9]: #!pip install scikit-learn
    from sklearn.model_selection import train_test_split
    x_train, x_test , y_train , y_test = train_test_split(x,y,test_size=0.20,random_state=101)

0]: from sklearn.linear_model import LinearRegression
    from sklearn.metrics import r2_score
    lr=LinearRegression()

1]: lr.fit(x_train,y_train)

1]: LinearRegression()

2]: y_pred=lr.predict(x_test)
    r2_score(y_test,y_pred)

2]: 0.5746585054942905
```

# Polynomial Regression

- Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial. The Polynomial Regression equation is given below:

$$y= b_0+b_1x_1+ b_2x_1^2+ b_2x_1^3+..... b_nx_1^n$$

- It is also called the special case of Multiple Linear Regression in ML. Because we add some polynomial terms to the Multiple Linear regression equation to convert it into Polynomial Regression.
- It is a linear model with some modification in order to increase the accuracy.
- The dataset used in Polynomial regression for training is of non-linear nature.
- It makes use of a linear regression model to fit the complicated and non-linear functions and datasets.

## POLYNOMIAL

```
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree =2)
x_poly = poly_reg.fit_transform(x)
lin_reg2 = LinearRegression()
lin_reg2.fit(x_poly,y)


print(f'R² score: {r2_score(y, lin_reg2.predict(poly_reg.fit_transform(x)))*100}')
```

```
R² score: 60.22231360110801
```

# Random Forest Regression

Random Forest Regression is **a supervised learning algorithm that uses ensemble learning method for regression**. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single modelThe greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

## RANDOMFOREST

```
from sklearn.ensemble import RandomForestRegressor
rf_reg=RandomForestRegressor(n_estimators = 10,random_state=0)
rf_reg.fit(x,y)

print(f'R² score: {r2_score(y, rf_reg.predict(x))*100}')
```

```
<ipython-input-54-6ef2ce8ae430>:3: DataConversionWarning: A colu
nge the shape of y to (n_samples,), for example using ravel().
  rf_reg.fit(x,y)
```

```
R² score: 92.24570347793016
```

# Model Prediction

Model evaluation is **the process of using different evaluation metrics to understand a machine learning model's performance, as well as its strengths and weaknesses**. Model evaluation is important to assess the efficacy of a model during initial research phases, and it also plays a role in model monitoring.

```
new_house=[[3,
3,1870
,2120,
1970]]
print('predicted value is:',int(rf_reg.predict(new_house)))
```

```
predicted value is: 669175
```

The actual house rate is 6,99,000 but the model predicts the rate of house is 6,69,175 .

# Resources&References

- ✓ Kaggle website for collecting the dataset.
- ✓ Any standard textbook or some blogs for getting a concrete idea on regressions.
- ✓ Python documentations on visualization tools and libraries for working with datasets.

# Gratitude

It feels so fruitful at the semester for learning a subjects at its high peak possible.Moreever,having a project done on the basis of all the theoretical and practical knowledge that we gained throughout the semester from you is a thing that we should be greatful for.Hereby,submitting the project I,from the bottom of my heart,want to convey myimmense gratitude for guiding us all through Data Science and even other aspects of life.

Thank you,sir.

**Subject Faculty**:Asst.Prof.Ch.Satish Kumar,Department of Computer Science and Engineering,RGUKT IIIT Srikakulam,AP.

## Links:

https://github.com/ArchanaTalagapu/Dsp-project-.git

Done By,

T.Archana_S180581_Cse-2A