# Assignment Report and Approach

**Assignment:** Prediction of house prices based on the dataset provided using ml algorithms.

**Description:** Based on the different features which were provided in the datasets such as, house age, latitude & longitude for precise location of the house, house size, no of stores the model has to predict the house prices based on the input provided.

**Colab link:**
https://colab.research.google.com/drive/1zoTADvwB1chQPWoo7E8SaVfAfCUn-GiO?usp=sharing

**Approach:**

The task which we needed to accomplish was prediction of house prices which was a **continuous variable** so we are going with the **Supervised Regression Algorithms**.

The code consists of mainly 4 blocks:

>> **Importing libraries and datasets**

All the required libraries and the datasets were imported in this section.
Libraries which were essential such as pandas, numpy, matplotlib and seaborn were imported.

>> **Observations**

Initial observations such as the shape of the dataset, no of features , looking for presence of any null values and data type of the features were carried out here.

- Data consists of 414 rows and 9 columns including target columns.
- No null values were found in the data
- All features were either integer type or float data type, so no need for data type conversion required.

>> **EDA**

Over here using the correlation properties between the features , I have tried to drop some features so the model will not suffer and predict the house prices correctly. But unfortunately whenever I have dropped a feature, the accuracy of the model has reduced and no improvements were found. Which were clearly observable from the correlation plot graph made with the sns library.

```
correlation = data.corr()
correlation['House price of unit area'].sort_values(ascending=False)
```

```
House price of unit area                  1.000000
Number of convenience stores              0.571005
latitude                                  0.546307
longitude                                 0.523287
Transaction date                          0.087529
Number of bedrooms                        0.050265
House size (sqft)                         0.046489
House Age                                -0.210567
Distance from nearest Metro station (km) -0.673613
Name: House price of unit area, dtype: float64
```

**>>MODELS**

So I have tried three regression models: Linear Regression , Random Forest Regressor and finally a neural network for predicting house prices.

Before moving further when we look into the values in the datasets , every feature has its own scale of representation. Like Area in sq.feet , no of stores in discrete values, metro station distance in km and so on. **Features scaling** was done where every value of the values were divided by the highest value of the particular feature so the range of the values will be in between 0-1.This helps in decrease in **mathematical computation** in the models.

```
for i in final_data.columns:
  final_data[i]=final_data[i].div(max(final_data[i]))
```

```
final_data.head(5)
```

| | Transaction date | House Age | Distance from nearest Metro station (km) | Number of convenience stores | latitude | longitude | Number of bedrooms | House size (sqft) | House price of unit area |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.999669 | 0.730594 | 0.013082 | 1.0 | 0.998736 | 0.999786 | 0.333333 | 0.383333 | 0.322553 |
| 1 | 0.999669 | 0.445205 | 0.047256 | 0.9 | 0.998631 | 0.999780 | 0.666667 | 0.826667 | 0.359149 |
| 2 | 1.000000 | 0.303653 | 0.086619 | 0.5 | 0.998915 | 0.999816 | 1.000000 | 0.706667 | 0.402553 |
| 3 | 0.999959 | 0.303653 | 0.086619 | 0.5 | 0.998915 | 0.999816 | 0.666667 | 0.583333 | 0.466383 |
| 4 | 0.999628 | 0.114155 | 0.060198 | 0.5 | 0.998592 | 0.999804 | 0.333333 | 0.327333 | 0.366809 |

**Every model and the required libraries were mentioned in the sections allocated respectively.**
At last I used a library called PYCARET, and the dataset was tested against all the regression models.

**The results of the models were:**

| MODEL | ACCURACY OBTAINED |
|---|---|
| Linear Regression | 60.09% |
| Random Forest Regressor | 67.96% |
| Neural Network (50 epoch) | 63.27% |