RealState_Prediction

September 7, 2023

To explain various steps in ML project. We will use “Real estate price prediction” data set . The Kaggle link for the data set

https://www.kaggle.com/datasets/quantbruce/real-estate-price-prediction

This dataset has 414 sample data . We will create a ML model to predict the price per unit area of a house based on -

1. Age of the house
2. Distance from nearest MRT satiation
3. Number of convenience stores

STEP 1 - Data Collection

We will used python panda library to load data (collected from Kaggle)

[3]:
```
import pandas as pd
df=pd.read_csv("/content/drive/MyDrive/Python/AI/datasets/Real estate.csv")
df.head()
```

[3]:

	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores
0	1	2012.917	32.0	84.87882	10
1	2	2012.917	19.5	306.59470	9
2	3	2013.583	13.3	561.98450	5
3	4	2013.500	13.3	561.98450	5
4	5	2012.833	5.0	390.56840	5

```
   X5 latitude  X6 longitude  Y house price of unit area
0     24.98298     121.54024                        37.9
1     24.98034     121.53951                        42.2
2     24.98746     121.54391                        47.3
3     24.98746     121.54391                        54.8
4     24.97937     121.54245                        43.1
```

STEP 2 - Data Cleaning

Logically Colums like - 1. transaction date 2. latitude 3. longitude

Should not have any influence on the outcome (price per unit). Hence, we should remove them

[4]:
```
df=df.drop(['No','X1 transaction date', 'X5 latitude','X6 longitude'], axis=1)
```

STEP 3 - Data Preparation

In this step we will select input and result data and create train and test data sample.

In current example we will consider

input x -> age of the house, distance from nearest MRT satiation, number of convenience stores

ouput y ->house price per unit area

[5]:
```
x=df.iloc[:,0:3]
y=df.iloc[:,3:4]
```

Using Python library for creating train and test split .2 = 80-20% split

[6]:
```
from sklearn import model_selection
X_train,X_test,y_train,y_test=model_selection.train_test_split(x,y,test_size=0.
↪2,random_state=2)
```

STEP 4 - Model Selection

Process of fitting the dataset into a standard ML model algorithm . Our current dataset is best fit for Linear Regression algorithm. We will use standard Python sklearn "LinearRegression" ML model algorithm

[7]:
```
from sklearn import linear_model
model=linear_model.LinearRegression()
```

STEP 5 - Model Training

We will used standard libraray method to train our model with train data set created in step-3

[ ]:
```
model.fit(X_train,y_train)
```

STEP 6 - Model Evaluation

We will used sample test data created in step-3 to evaluate the model performance. Model's standard methods will be used.

[9]:
```
y_predict=model.predict(X_test)
```

Depending on the use-case and type of the model selection various performance matrices will be collected to evaluate the accuracy of the model. (standard libraried will be used)

[10]:
```
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
MAE = mean_absolute_error(y_test,y_predict)
MSE = mean_squared_error(y_test,y_predict)
R2_score = r2_score(y_test,y_predict)
print('MAE -> ',MAE)
print('MSE -> ',MSE)
print('R2 score -> ',R2_score)
```

```
MAE ->  6.362370630628733
MSE ->  119.36009326395077
R2 score ->  0.4615959935866515
```

[ ]:
```
model.predict([[16,100,7]])
```

STEP 7 - Model deployment

Trained model will be exported. This will be used in web framework to host the model as API

[12]:
```
import pickle
with open('/content/drive/MyDrive/Python/AI/models/model.pkl', 'wb') as file:
    pickle.dump(model, file)
```