

```
1 def regressor1(x_train,x_test,y_train):
2     from sklearn.linear_model import LinearRegression
3     from sklearn.tree import DecisionTreeRegressor
4     from sklearn.ensemble import RandomForestRegressor
5     from sklearn.svm import SVR
6
7     lr = LinearRegression()
8     lr.fit(x_train,y_train)
9     y_pred = lr.predict(x_test)
10
11     dtr = DecisionTreeRegressor()
12     dtr.fit(x_train,y_train)
13     y_pred2 = dtr.predict(x_test)
14
15     rfr = RandomForestRegressor()
16     rfr.fit(x_train,y_train)
17     y_pred3 = rfr.predict(x_test)
18
19     svr = SVR()
20     svr.fit(x_train,y_train)
21     y_pred4 = svr.predict(x_test)
22
23     return y_pred,y_pred2,y_pred3,y_pred4


1 def regressor2(x_train,x_test,y_train,y_test):
2     from sklearn.metrics import mean_absolute_error
3     from sklearn.metrics import r2_score
4     from sklearn.linear_model import LinearRegression
5     from sklearn.tree import DecisionTreeRegressor
6     from sklearn.ensemble import RandomForestRegressor
7     from sklearn.svm import SVR
8
9     lr = LinearRegression()
10    lr.fit(x_train,y_train)
11    y_pred = lr.predict(x_test)
12    lre = mean_absolute_error(y_test,y_pred)
13    print("Linear Regression\nmean absolute error : ",lre,"\n")
14    print("R2_score : ",r2_score(y_test,y_pred),"\n")
15
16    dtr = DecisionTreeRegressor()
17    dtr.fit(x_train,y_train)
18    y_pred2 = dtr.predict(x_test)
19    dtre = mean_absolute_error(y_test,y_pred2)
20    print("Decision Tree Classifier\nmean absolute error : ",dtre,"\n")
21    print("R2_score : ",r2_score(y_test,y_pred2),"\n")
22
23    rfr = RandomForestRegressor()
24    rfr.fit(x_train,y_train)
25    y_pred3 = rfr.predict(x_test)
```

```

26 rfre = mean_absolute_error(y_test,y_pred3)
27 print("Random Forest Regressor\nmean absolute error : ",rfre,"\n")
28 print("R2_score : ",r2_score(y_test,y_pred3),"\n")
29 svr = SVR()
30 svr.fit(x_train,y_train)
31 y_pred4 = svr.predict(x_test)
32 svre = mean_absolute_error(y_test,y_pred4)
33 print("Suppoer Vector Regression\nmean absolute error : ",svre,"\n")
34 print("R2_score : ",r2_score(y_test,y_pred4),"\n")
35
36 return y_pred,y_pred2,y_pred3,y_pred4

```

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt

```

```
1 train = pd.read_csv("/content/train.csv")
```

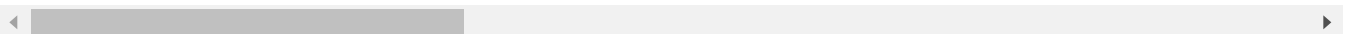
```

1 print(train.shape)
2 train.head(2)

```

```
(39499, 15)
```

	Property_ID	Property_Type	Property_Area	Number_of_Windows	Number_of_Doors	Fi
0	0x21e3	Apartment	106	NaN	1	Semi_
1	0x68d4	Apartment	733	2.0	2	U



```
1 train.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39499 entries, 0 to 39498
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Property_ID           39499 non-null  object
1   Property_Type         39499 non-null  object
2   Property_Area         39499 non-null  int64
3   Number_of_Windows     37845 non-null  float64
4   Number_of_Doors       39499 non-null  int64
5   Furnishing            38457 non-null  object
6   Frequency_of_Powercuts 38116 non-null  float64
7   Power_Backup          39499 non-null  object
8   Water_Supply          39499 non-null  object

```

```

9   Traffic_Density_Score    39499 non-null float64
10  Crime_Rate                38712 non-null object
11  Dust_and_Noise            38280 non-null object
12  Air_Quality_Index         39499 non-null float64
13  Neighborhood_Review       39499 non-null float64
14  Habitability_score        39499 non-null float64
dtypes: float64(6), int64(2), object(7)
memory usage: 4.5+ MB

```

```
1 test = pd.read_csv("/content/test.csv")
```

```
1 test.shape
```

```
(10500, 14)
```

```

1 print(train.isnull().sum())
2 print(test.isnull().sum())

```

```

Property_ID                0
Property_Type              0
Property_Area              0
Number_of_Windows         1654
Number_of_Doors            0
Furnishing                 1042
Frequency_of_Powercuts     1383
Power_Backup              0
Water_Supply              0
Traffic_Density_Score      0
Crime_Rate                 787
Dust_and_Noise            1219
Air_Quality_Index          0
Neighborhood_Review        0
Habitability_score         0
dtype: int64
Property_ID                0
Property_Type              0
Property_Area              0
Number_of_Windows         445
Number_of_Doors            0
Furnishing                 257
Frequency_of_Powercuts     366
Power_Backup              0
Water_Supply              0
Traffic_Density_Score      0
Crime_Rate                 212
Dust_and_Noise            330
Air_Quality_Index          0
Neighborhood_Review        0
dtype: int64

```

```
1 from sklearn.impute import SimpleImputer, KNNImputer
```

```
1 impute = SimpleImputer(strategy = "most_frequent")
2 knnimputer = KNNImputer(n_neighbors=15)
```

```
1 y = train.iloc[:,-1]
2 train.drop(train.columns[-1],axis=1,inplace=True)
```

```
1 train2 = train.iloc[:,[3,6]]
2 train2 = knnimputer.fit_transform(train2)
3 train2 = pd.DataFrame(train2)
```

```
1 train3 = train.iloc[:,[5,10,11]]
2 train3 = impute.fit_transform(train3)
3 train3 = pd.DataFrame(train3)
```

```
1 train.iloc[:,3] = train2.iloc[:,0]
2 train.iloc[:,6] = train2.iloc[:,1]
3 train.iloc[:,5] = train3.iloc[:,0]
4 train.iloc[:,10] = train3.iloc[:,1]
5 train.iloc[:,11] = train3.iloc[:,2]
```

```
1 test2 = test.iloc[:,[3,6]]
2 test2 = knnimputer.fit_transform(test2)
3 test2 = pd.DataFrame(test2)
```

```
1 test3 = test.iloc[:,[5,10,11]]
2 test3 = impute.fit_transform(test3)
3 test3 = pd.DataFrame(test3)
```

```
1 test.iloc[:,3] = test2.iloc[:,0]
2 test.iloc[:,6] = test2.iloc[:,1]
3 test.iloc[:,5] = test3.iloc[:,0]
4 test.iloc[:,10] = test3.iloc[:,1]
5 test.iloc[:,11] = test3.iloc[:,2]
```

```
1 from sklearn.preprocessing import OneHotEncoder
2 from sklearn.preprocessing import StandardScaler
3 ohe = OneHotEncoder()
4 sc = StandardScaler()
```

```
1 train2 = train.iloc[:,[1,5,7,8,10,11]]
2 train2 = ohe.fit_transform(train2)
3 train2 = pd.DataFrame(train2.toarray())
4
5 train3 = train.iloc[:,[2,3,4,6,9,12,13]]
6 train3 = sc.fit_transform(train3)
```

```
7 train3 = pd.DataFrame(train3)
8
9 train = pd.concat([train2,train3],axis=1)

1 test2 = test.iloc[:,[1,5,7,8,10,11]]
2 test2 = ohe.fit_transform(test2)
3 test2 = pd.DataFrame(test2.toarray())
4
5 test3 = test.iloc[:,[2,3,4,6,9,12,13]]
6 test3 = sc.fit_transform(test3)
7 test3 = pd.DataFrame(test3)
8
9 test = pd.concat([test2,test3],axis=1)

1 p1,p2,p3,p4 = regressor1(train,test,y)

1 from sklearn.metrics import mean_absolute_error
2 mean_absolute_error(p1,p2)

7.393495208333333

1 mean_absolute_error(p2,p3)

4.155949761904761

1 mean_absolute_error(p3,p4)

3.5897523347410205

1 mean_absolute_error(p4,p2)

5.796674246296543

1 p3

array([31.807 , 80.139 , 66.5328, ..., 75.0706, 79.834 , 79.2593])

1 test = pd.read_csv("/content/test.csv")
2 id = test.iloc[:,0]

1 submission = pd.DataFrame({"Property_ID":id,"Habitability_score":p3})

1 submission
```

	Property_ID	Habitability_score
0	0x6e93	31.8070
1	0x8787	80.1390
2	0x6c17	66.5328
3	0x9dbd	72.1662
4	0xbfde	77.1976
...	...	...
10495	0x423d	64.3772
10496	0x78c5	82.1809
10497	0xbf3	75.0706
10498	0x305b	79.8340
10499	0x5cff	79.2593

10500 rows x 2 columns

```
1 submission.to_csv("Submission.csv",index=False)
```

```
1 pd.read_csv("Submission.csv")
```

	Property_ID	Habitability_score
0	0x6e93	31.8070
1	0x8787	80.1390
2	0x6c17	66.5328
3	0x9dbd	72.1662
4	0xbfde	77.1976
...	...	...
10495	0x423d	64.3772
10496	0x78c5	82.1809
10497	0xbf3	75.0706
10498	0x305b	79.8340
10499	0x5cff	79.2593

10500 rows x 2 columns

---

✓

0s

completed at 8:12 AM

●

✕