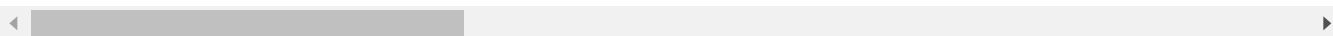


```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
```

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

```
1 df.head(2)
```

	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 df.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 df.dropna(inplace=True)
```

```
1 df.head(2)
```

	Property_ID	Property_Type	Property_Area	Number_of_Windows	Number_of_Doors	Furni
1	0x68d4	Apartment	733	2.0	2	Unfur
2	0x7d81	Apartment	737	4.0	2	Fur



```
1 df1 = df.iloc[:,[1,5,7,8,10,11]]
2 df1
```

	Property_Type	Furnishing	Power_Backup	Water_Supply	Crime_Rate	Dust_and_I
1	Apartment	Unfurnished	No	Once in a day - Evening	Well below average	Mε
2	Apartment	Fully Furnished	No	Once in a day - Morning	Slightly below average	Mε
3	Apartment	Unfurnished	Yes	Once in a day - Morning	Well above average	Mε
4	Bungalow	Fully Furnished	No	All time	Well below average	Mε
5	Single-family home	Unfurnished	No	Once in a day - Morning	Well below average	Mε
...
39491	Bungalow	Semi_Furnished	Yes	All time	Well below average	Mε
39492	Bungalow	Fully Furnished	No	All time	Slightly above average	Mε
39495	Apartment	Fully Furnished	No	All time	Slightly above average	Mε
39496	Bungalow	Unfurnished	Yes	Once in two days	Well below average	Mε
39498	Single-family home	Fully Furnished	No	All time	Well below average	Mε

23772 rows x 6 columns

```
1 for a in df1.columns:
```

```
2 print(a,len(df1[a].unique()))
```

```
Property_Type 6
Furnishing 3
Power_Backup 3
Water_Supply 5
Crime_Rate 4
Dust_and_Noise 3
```

```
1 from sklearn.preprocessing import OneHotEncoder
```

```
1 ohe = OneHotEncoder()
2 df1 = ohe.fit_transform(df1)
```

```
1 df1=pd.DataFrame(df1.toarray())
```

```
1 df2 = df.iloc[:,[2,3,4,6,9,12,13]]
2 df2.head()
```

	Property_Area	Number_of_Windows	Number_of_Doors	Frequency_of_Powercuts	Traffic_I
1	733	2.0	2	1.0	
2	737	4.0	2	0.0	
3	900	3.0	2	2.0	
4	2238	14.0	6	0.0	
5	1185	3.0	3	0.0	



```
1 from sklearn.preprocessing import StandardScaler
```

```
1 sc = StandardScaler()
```

```
1 df2 = sc.fit_transform(df2)
```

```
1 df2 = pd.DataFrame(df2)
```

```
1 final = pd.concat([df2,df1],axis=1)
```

```
1 final
```

	0	1	2	3	4	5	6	0	1
0	-0.337143	-0.715193	-0.284456	0.685475	-1.627038	-0.407715	-0.231744	0.0	1.0
1	-0.335111	0.027420	-0.284456	-0.643602	0.896389	-0.007047	0.094363	0.0	1.0
2	-0.252293	-0.343886	-0.284456	2.014552	-0.160501	-0.343608	-3.003652	0.0	1.0
3	0.427527	3.740487	3.099358	-0.643602	-0.734007	-0.087181	1.298450	0.0	0.0
4	-0.107488	-0.343886	0.561498	-0.643602	-0.545569	-0.487849	0.947258	0.0	0.0
...
33767	1.329890	2.997874	3.099358	-0.643602	0.937354	-0.664143	1.348620	0.0	0.0
33768	0.975753	2.255260	1.407451	2.014552	-1.889212	-0.327582	-1.812108	0.0	0.0
33769	-0.483472	-1.086500	0.561498	0.685475	-0.537376	-0.567983	-1.009383	0.0	1.0
33770	1.210997	0.770034	3.099358	-0.643602	0.396620	0.249380	0.081820	0.0	0.0
33771	-0.085132	-0.715193	1.407451	-0.643602	1.584597	0.169247	1.173024	0.0	0.0

33772 rows × 31 columns



```
1 y = df.iloc[:,14]
```

```
1 x = final
2 x
```

	0	1	2	3	4	5	6	0	1
0	-0.337143	-0.715193	-0.284456	0.685475	-1.627038	-0.407715	-0.231744	0.0	1.0
1	-0.335111	0.027420	-0.284456	-0.643602	0.896389	-0.007047	0.094363	0.0	1.0
2	-0.252293	-0.343886	-0.284456	2.014552	-0.160501	-0.343608	-3.003652	0.0	1.0

```

1 from sklearn.model_selection import train_test_split

1 x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)
...
1 from sklearn.linear_model import LinearRegression

1 lr = LinearRegression()

1 lr.fit(x_train,y_train)

LinearRegression()
...

1 y_pred = lr.predict(x_test)

1 from sklearn.metrics import mean_absolute_error

1 mean_absolute_error(y_test,y_pred)

7.230068836508659

1 from sklearn.tree import DecisionTreeRegressor

1 dtr = DecisionTreeRegressor()

1 dtr.fit(x_train,y_train)

DecisionTreeRegressor()

1 y_pred2 = dtr.predict(x_test)

1 mean_absolute_error(y_test,y_pred2)

5.762447076239822

1 from sklearn.ensemble import RandomForestRegressor

```

```
1 rfr = RandomForestRegressor()
```

```
1 rfr.fit(x_train,y_train)
```

```
RandomForestRegressor()
```

```
1 y_pred3 = rfr.predict(x_test)
```

```
1 mean_absolute_error(y_test,y_pred3)
```

```
4.423675692079941
```

```
1 from sklearn.svm import SVR
```

```
1 svr = SVR()
```

```
1 svr.fit(x_train,y_train)
```

```
SVR()
```

```
1 y_pred4 = svr.predict(x_test)
```

```
1 mean_absolute_error(y_test,y_pred4)
```

```
5.7100159987768935
```

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

```
1 df_test.fillna(method="ffill",axis=0,inplace=True)
```

```
1 df.head(2)
```

	Property_ID	Property_Type	Property_Area	Number_of_Windows	Number_of_Doors	Furni
1	0x68d4	Apartment	733	2.0	2	Unfur
2	0x7d81	Apartment	737	4.0	2	Fur



```
1 df1.head(2)
```

	0	1	2	3	4	5	6	7	8	9	...	14	15	16	17	18	19	20	21
0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
1	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0

2 rows × 24 columns



```
1 df2.head(2)
```

	0	1	2	3	4	5	6
0	-0.337143	-0.715193	-0.284456	0.685475	-1.627038	-0.407715	-0.231744
1	-0.335111	0.027420	-0.284456	-0.643602	0.896389	-0.007047	0.094363



```
1 dft1 = df_test.iloc[:,[1,5,7,8,10,11]]
2 dft1
```

	Property_Type	Furnishing	Power_Backup	Water_Supply	Crime_Rate	Dust_and_I
0	Apartment	Unfurnished	No	Once in a day - Morning	Well above average	Me
1	Apartment	Semi_Furnished	No	Once in a day - Evening	Well below average	Me

```
1 for a in dft1.columns:
2     print(a,len(dft1[a].unique()))
```

```
Property_Type 6
Furnishing 3
Power_Backup 3
Water_Supply 5
Crime_Rate 4
Dust_and_Noise 3
```

```
1 from sklearn.preprocessing import OneHotEncoder
```

```
1 ohe = OneHotEncoder()
2 dft1 = ohe.fit_transform(dft1)
```

```
10496      Constanca  Semi_Furnished      Yes      All time      Well below      Me
```

```
1 dft1=pd.DataFrame(dft1.toarray())
```

```
10497      Duplex      Fully_Furnished      No      Once in a day      Well below
```

```
1 dft2 = df_test.iloc[:,[2,3,4,6,9,12,13]]
2 dft2.head()
```

	Property_Area	Number_of_Windows	Number_of_Doors	Frequency_of_Powercuts	Traffic_I
0	293	3.0	1		0.0
1	586	4.0	1		0.0
2	305	1.0	2		1.0
3	258	2.0	1		1.0
4	3031	12.0	4		0.0



```
1 from sklearn.preprocessing import StandardScaler
```

```
1 sc = StandardScaler()
```

```
1 dft2 = sc.fit_transform(dft2)
```



```
1 dft2 = pd.DataFrame(dft2)

1 final2 = pd.concat([dft2,dft1],axis=1)

1 dft1.head(2)
```

	0	1	2	3	4	5	6	7	8	9	...	14	15	16	17	18	19	20	21
0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0
1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0

2 rows × 24 columns



```
1 dft2.head(2)
```

	0	1	2	3	4	5	6	
0	-0.590717	-0.333468	-1.128739	-0.647132	0.755214	0.565622	-1.530498	
1	-0.429161	0.038983	-1.128739	-0.647132	1.043000	-0.512868	0.536849	

```
1 final2
```

	0	1	2	3	4	5	6	0	1
0	-0.590717	-0.333468	-1.128739	-0.647132	0.755214	0.565622	-1.530498	0.0	1.0

```

1 p1 = lr.predict(final2)
2 p2 = dtr.predict(final2)
3 p3 = rfr.predict(final2)
4 p4 = svr.predict(final2)

```

	0	1	2	3	4	5	6	0	1
1	0.019075	0.019505	1.122860	0.647132	1.006907	1.176766	1.270282	0.0	0.0

```

1 mean_absolute_error(p1,p2)

7.261287617935883

1 mean_absolute_error(p2,p3)

4.054170942857143
-----
1 mean_absolute_error(p3,p4)

3.782548119975854
🚀
1 mean_absolute_error(p4,p2)

5.833542737963133

1 p3

array([30.9022, 81.0418, 65.2626, ..., 75.4154, 79.661 , 79.8095])

1 df_test.head()

```

	Property_ID	Property_Type	Property_Area	Number_of_Windows	Number_of_Doors	Fi
0	0x6e93	Apartment	293	3.0	1	U

```
1 id = df_test.iloc[:,0]
```

1	0x0707	Apartment	300	4.0	1	Semi_
---	--------	-----------	-----	-----	---	-------

```
1 pd.read_csv("/content/sample_submission.csv")
```

	Property_ID	Habitability_score
0	0x21e3	71.98
1	0x68d4	71.20
2	0x7d81	71.39
3	0x7a57	31.46
4	0x9409	93.70

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

```
1 submission
```

	Property_ID	Habitability_score
0	0x6e93	30.9022
1	0x8787	81.0418
2	0x6c17	65.2626
3	0x9dbd	71.2135
4	0xbfde	74.3830
...
10495	0x423d	60.9617
10496	0x78c5	83.0341
10497	0xbf3	75.4154
10498	0x305b	79.6610
10499	0x5cff	79.8095

10500 rows × 2 columns

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

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

```
1 pd.read_csv('submission.csv'),
```

	Property_ID	Habitability_score
0	0x6e93	30.9022
1	0x8787	81.0418
2	0x6c17	65.2626
3	0x9dbd	71.2135
4	0xbfde	74.3830
...
10495	0x423d	60.9617
10496	0x78c5	83.0341
10497	0xbf3	75.4154
10498	0x305b	79.6610
10499	0x5cff	79.8095

10500 rows × 2 columns

1