

Name : Krish Sukhani

Batch : D

Roll No : 59

TE IT

ML Linear Regression

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

```
from google.colab import drive
drive.mount("/content/gdrive")
```

🔗 Mounted at /content/gdrive

```
austin_weather = pd.read_csv('/content/gdrive/My Drive/datasets/austin_weather.csv',encoding=
```

```
austin_weather.head()
```

	Date	TempHighF	TempAvgF	TempLowF	DewPointHighF	DewPointAvgF	DewPointLowF	Humi
0	2013-12-21	74	60	45	67	49	43	
1	2013-12-22	56	48	39	43	36	28	
2	2013-12-23	58	45	32	31	27	23	
3	2013-12-24	61	46	31	36	28	21	
4	2013-12-25	58	50	41	44	40	36	

```
austin_weather = austin_weather.drop(['Events', 'Date', 'SeaLevelPressureHighInches',
                                       'SeaLevelPressureLowInches', 'TempHighF', 'TempLowF', 'DewPointHighF', 'DewPoin
```

```
austin_weather.head()
```

	TempAvgF	DewPointAvgF	HumidityAvgPercent	SeaLevelPressureAvgInches	VisibilityAvg
0	60	49	75	29.68	
1	48	36	68	30.13	
2	45	27	52	30.49	
3	46	28	56	30.45	
4	50	40	71	30.33	

```
austin_weather = austin_weather.replace('T', 0.0)
```

```
austin_weather.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1319 entries, 0 to 1318
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   TempAvgF                             1319 non-null   int64
1   DewPointAvgF                         1319 non-null   object
2   HumidityAvgPercent                   1319 non-null   object
3   SeaLevelPressureAvgInches           1319 non-null   object
4   VisibilityAvgMiles                   1319 non-null   object
5   WindAvgMPH                          1319 non-null   object
6   PrecipitationSumInches               1319 non-null   object
dtypes: int64(1), object(6)
memory usage: 72.3+ KB
```

```
austin_weather = austin_weather.replace('-', np.NaN)
```

```
austin_weather = austin_weather.dropna()
```

```
austin_weather.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1307 entries, 0 to 1318
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   TempAvgF                             1307 non-null   int64
1   DewPointAvgF                         1307 non-null   object
2   HumidityAvgPercent                   1307 non-null   object
3   SeaLevelPressureAvgInches           1307 non-null   object
4   VisibilityAvgMiles                   1307 non-null   object
```

```

5   WindAvgMPH          1307 non-null  object
6   PrecipitationSumInches  1307 non-null  object
dtypes: int64(1), object(6)
memory usage: 81.7+ KB

```

```
austin_weather["DewPointAvgF"] = austin_weather["DewPointAvgF"].astype(float)
```

```
austin_weather["HumidityAvgPercent"] = austin_weather["HumidityAvgPercent"].astype(float)
```

```
austin_weather["SeaLevelPressureAvgInches"] = austin_weather["SeaLevelPressureAvgInches"].ast
```

```
austin_weather["VisibilityAvgMiles"] = austin_weather["VisibilityAvgMiles"].astype(float)
```

```
austin_weather["WindAvgMPH"] = austin_weather["WindAvgMPH"].astype(float)
```

```
austin_weather["PrecipitationSumInches"] = austin_weather["PrecipitationSumInches"].astype(fl
```

```
austin_weather.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1307 entries, 0 to 1318
Data columns (total 7 columns):
#   Column              Non-Null Count  Dtype
---  -
0   TempAvgF            1307 non-null   int64
1   DewPointAvgF        1307 non-null   float64
2   HumidityAvgPercent  1307 non-null   float64
3   SeaLevelPressureAvgInches  1307 non-null   float64
4   VisibilityAvgMiles  1307 non-null   float64
5   WindAvgMPH          1307 non-null   float64
6   PrecipitationSumInches  1307 non-null   float64
dtypes: float64(6), int64(1)
memory usage: 81.7 KB

```

```
Y = austin_weather['PrecipitationSumInches']
```

```
Y = Y.values.reshape(-1, 1)
```

```
X = austin_weather.drop(['PrecipitationSumInches'], axis = 1)
```

```
X.head()
```

	TempAvgF	DewPointAvgF	HumidityAvgPercent	SeaLevelPressureAvgInches	VisibilityAvg
0	60	49.0	75.0	29.68	
1	48	36.0	68.0	30.13	
2	45	27.0	52.0	30.49	

Y[5]

```
array([0.])
```

```
Rain = LinearRegression()
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y)
```

```
Rain.fit(X_train, Y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
Y_pred = Rain.predict(X_test)
```

```
print('Mean absolute error = ' + str(mean_absolute_error(Y_test, Y_pred)))
```

```
print('Mean squared error = ' + str(mean_squared_error(Y_test, Y_pred)))
```

```
print('R2 score = ' + str(r2_score(Y_test, Y_pred)))
```

```
Mean absolute error = 0.15867863151540623
```

```
Mean squared error = 0.07546507884560216
```

```
R2 score = 0.19619611017895366
```

```
Rain.coef_
```

```
array([[ 0.01342898, -0.01354499,  0.01168234, -0.15174905, -0.11005644,
         0.00895832]])
```

```
coeff_austin=pd.DataFrame(Rain.coef_[0],X.columns,columns=['Coefficient'])
```

```
coeff_austin
```

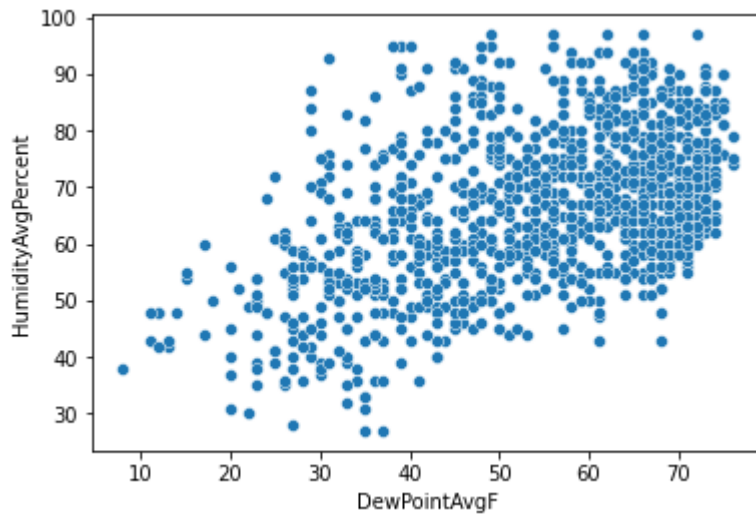
	Coefficient
TempAvgF	0.013429
DewPointAvgF	-0.013545
HumidityAvgPercent	0.011682
SeaLevelPressureAvgInches	-0.151749
VisibilityAvgMiles	-0.110056
WindAvgMPH	0.008958

```
austin_weather.corr()
```

	TempAvgF	DewPointAvgF	HumidityAvgPercent	SeaLevelPressure
TempAvgF	1.000000	0.895020	0.033040	
DewPointAvgF	0.895020	1.000000	0.453475	
HumidityAvgPercent	0.033040	0.453475	1.000000	
SeaLevelPressureAvgInches	-0.621492	-0.630993	-0.181596	
VisibilityAvgMiles	0.210172	-0.063717	-0.584849	
WindAvgMPH	0.037003	0.037517	-0.002244	
PrecipitationSumInches	-0.018420	0.127098	0.346870	

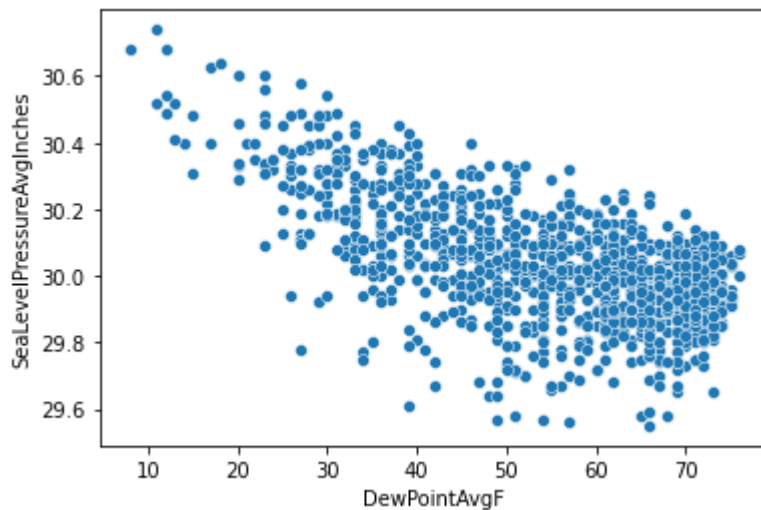
```
sns.scatterplot(x="DewPointAvgF", y="HumidityAvgPercent",data=austin_weather)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fa91d69bc50>



```
sns.scatterplot(x="DewPointAvgF", y="SeaLevelPressureAvgInches",data=austin_weather)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fa91d64a358>



```
y_test = list()
for i in Y_test:
    y_test.append(i[0])
```

y_test

```
[0.0,
 0.0,
 0.0,
 0.01,
 0.0,
 0.0,
 0.0,
 0.03,
 0.0,
 0.0,
 0.0,
 0.1,
 0.0,
 0.0,
 0.0,
 0.01,
 0.3,
 0.0,
 0.07,
 0.0,
 0.0,
 0.0,
 0.0,
 0.0,
 0.0,
 0.0,
 0.0,
 0.73,
 0.0,
 0.1,
 0.0,
 0.07,
 0.0,
 0.0,
 0.0,
 0.11,
 0.11,
 0.58,
 0.0,
 0.0,
 0.0,
 0.98,
 0.0,
 0.0,
 1.46,
 0.0,
 0.0,
 0.0,
 0.0,
 0.0,
 0.0,
 0.0]
```

```
0.0,  
0.21,  
0.0,  
0.0,  
0.0,  
0.12,  
0.0,  
0.11,  
0.0,
```

```
y_pred = list()  
for i in Y_pred:  
    y_pred.append(i[0])
```

y_pred

```
0.07081029039403349,  
0.02351540438472899,  
0.012253157849771767,  
0.15379850642268256,  
0.4934325474095038,  
0.035549429041270564,  
-0.04889430505622716,  
-0.02083050060815993,  
0.15663302077650965,  
-0.07304020453675353,  
0.0016510492904888707,  
-0.08039827223500229,  
0.02477341675618039,  
0.052744801286849,  
0.412414095637307,  
0.05905736540300577,  
0.0566508005553672,  
-0.006299282592729938,  
0.025385302626603767,  
0.0004394995419572112,  
0.11005267390537199,  
-0.07562266374943061,  
0.20188338298239028,  
0.2615308197847641,  
0.04153232564131315,  
0.034315619910507245,  
-0.07207861862288034,  
0.09578885293619699,  
0.011141084182460403,  
0.01593228875856223,  
0.01636836896452465,  
-0.185944784817754,  
0.1687286902861116,  
-0.07456921375122594,  
0.0657265968269245,  
0.10634289376300554,  
0.14043859763076405,  
0.37991478520975885,  
0.05282979274401001,  
0.1823784780602571,  
0.3462582527592195,
```

```
-0.013767854496583354,
0.021015518014458223,
0.21812037755664182,
0.15586495567857028,
0.7782315679329788,
0.6050198082092191,
0.6959927671350394,
0.018975655139789716,
-0.03497899485789979,
-0.11213405349933048,
0.16856246392212348,
0.10646679513522184,
-0.003878575473538426,
0.05490389747658764,
-0.018539687368512858,
0.04057788502049586,

0.42620004757795993,
-0.13746375533822874]
```

```
austin_weather2 = pd.DataFrame(data = (zip(y_test,y_pred)),columns=['Actual','Predicted'])
```

```
austin_weather2
```

	Actual	Predicted
0	0.00	0.144245
1	0.00	0.012934
2	0.00	0.040951
3	0.01	0.477221
4	0.00	0.074004
...
322	0.00	0.054904
323	0.00	-0.018540
324	0.00	0.040578
325	0.00	0.426200
326	0.00	-0.137464

```
327 rows × 2 columns
```

```
sns.lmplot(x='Actual',y='Predicted',data=austin_weather2)
```


<seaborn.axisgrid.FacetGrid at 0x7fa91e714390>

