RA1911030010030.Assignment2.MLCore.Nitish

September 13, 2021

[2]:
```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
```

[4]:
```
df = pd.read_csv("/home/waterupto/Downloads/Latest Covid-19 India Status.csv")
df.head()
```

[4]:

	State/UTs	Total Cases	Active	Discharged	Deaths	Active Ratio (%)	Discharge Ratio (%)	Death Ratio (%)
0	Andaman and Nicobar	7572	6	7437	129	0.08	98.22	1.70
1	Andhra Pradesh	2022064	14550	1993589	13925	0.72	98.59	0.69
2	Arunachal Pradesh	53408	634	52507	267	1.19	98.31	0.50
3	Assam	592616	6415	580491	5710	1.08	97.95	0.96
4	Bihar	725759	55	716048	9656	0.01	98.66	1.33

[5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 8 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   State/UTs            36 non-null     object
 1   Total Cases          36 non-null     int64
 2   Active               36 non-null     int64
 3   Discharged           36 non-null     int64
 4   Deaths               36 non-null     int64
 5   Active Ratio (%)     36 non-null     float64
 6   Discharge Ratio (%)  36 non-null     float64
 7   Death Ratio (%)      36 non-null     float64
dtypes: float64(3), int64(4), object(1)
memory usage: 2.4+ KB
```

[6]:
```
X = df[["Total Cases", "Active", "Discharged", "Deaths"]]
```

[8]:
```
from sklearn.preprocessing import MinMaxScaler
X_sca = MinMaxScaler()
X = X_sca.fit_transform(X)
```

[9]:
```
dfq = pd.DataFrame(data=X, columns=["Total Cases", "Active", "Discharged",␣
↪"Deaths"])
```

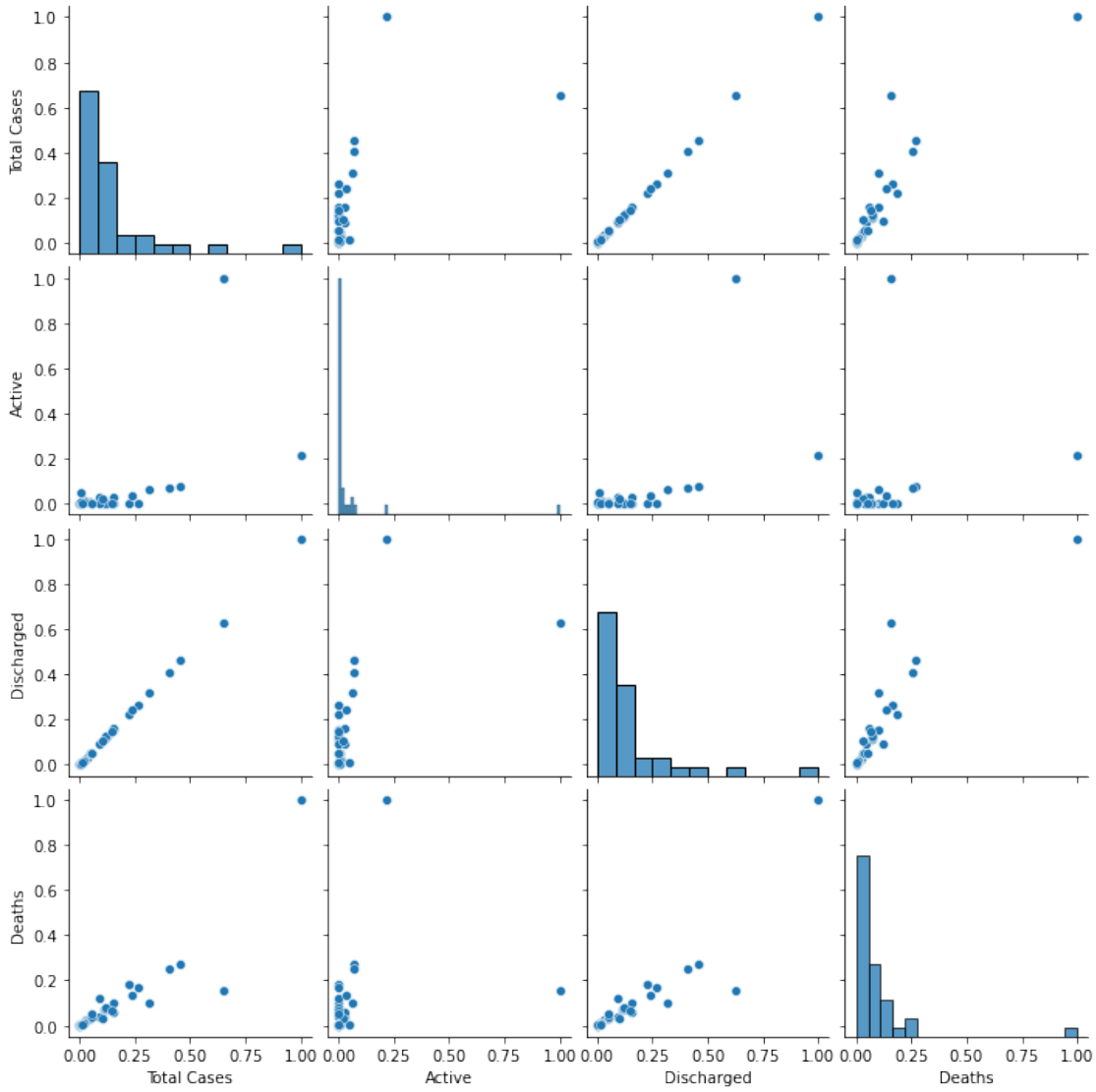
[10]:
```
dfq.describe()
```

[10]:

	Total Cases	Active	Discharged	Deaths
count	36.000000	36.000000	36.000000	36.000000
mean	0.140496	0.045588	0.141054	0.088872
std	0.208192	0.168225	0.207284	0.171517
min	0.000000	0.000000	0.000000	0.000000
25%	0.010291	0.000600	0.010156	0.005858
50%	0.071383	0.002923	0.072166	0.039312
75%	0.154109	0.023846	0.156528	0.099015
max	1.000000	1.000000	1.000000	1.000000

[12]:
```
import seaborn as sns
sns.pairplot(dfq)
```

[12]:
```
<seaborn.axisgrid.PairGrid at 0x7fece7cbce50>
```



[22]:
```
X = np.array(dfq["Active"])
y = np.array(dfq["Deaths"])
X = X.reshape(-1, 1)
y = y.reshape(-1, 1)
print(X.shape, y.shape)
```

```
(36, 1) (36, 1)
```

[15]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.20, random_state=0)
```

[16]:
```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
print(X_train.shape, y_train.shape)
reg.fit(X_train, y_train)
```

```
(28, 1) (28, 1)
```

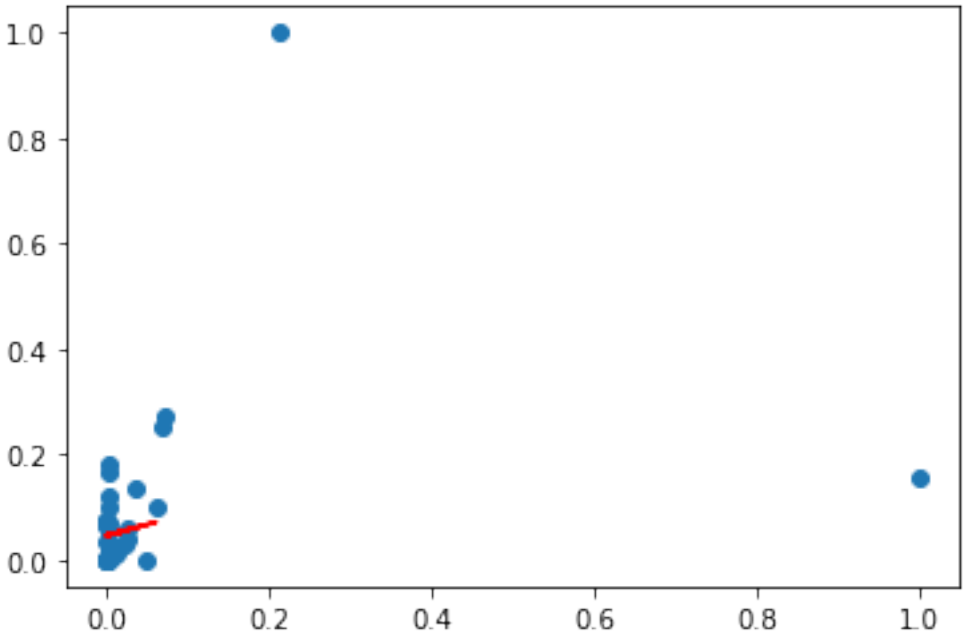
[16]:
```
LinearRegression()
```

[17]:
```
y_pred = reg.predict(X_test)
```

[18]:
```
reg.score(X_test, y_test)
```

[18]:
```
-0.19725328804505926
```

[19]:
```
plt.scatter(X, y)
plt.plot(X_train, reg.predict(X_train), color="red")
plt.show()
```



[20]:
```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
kf = KFold(n_splits=5)
model = LinearRegression()
scores = cross_val_score(model, X_train, y_train, scoring='r2', cv=kf)
print("Avg accuracy: {}".format(scores.mean()))
```

```
Avg accuracy: -0.5496354514436426
```

[ ]: