

DEP TASK 4

The screenshot shows a Google Colab notebook titled 'Untitled8.ipynb'. The code cell contains the following Python code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import LabelEncoder
```

The next cell shows the result of loading a CSV file:

```
data = pd.read_csv('/content/CloudWatch_Traffic_Web_Attack.csv')
data.head()
```

The output is a preview of the CSV data:

creation_time	end_time	src_ip	src_ip_country_code	protocol	response_code	dst_port	dst_ip	rule_names	observation_name	source.meta	source.name	time	detection_types
2024-04-25T23:00:00Z	2024-04-25T23:10:00Z	147.161.161.82	AE	HTTPS	200	443	10.138.69.97	Suspicious Web Traffic	Adversary Infrastructure Interaction	AWS_VPC_Flow	prod_webserver	2024-04-25T23:00:00Z	waf_rule
2024-04-25T23:00:00Z	2024-04-25T23:10:00Z	165.225.33.6	US	HTTPS	200	443	10.138.69.97	Suspicious Web Traffic	Adversary Infrastructure Interaction	AWS_VPC_Flow	prod_webserver	2024-04-25T23:00:00Z	waf_rule
2024-04-25T23:00:00Z	2024-04-25T23:10:00Z	165.225.212.255	CA	HTTPS	200	443	10.138.69.97	Suspicious Web Traffic	Adversary Infrastructure Interaction	AWS_VPC_Flow	prod_webserver	2024-04-25T23:00:00Z	waf_rule
2024-04-25T23:00:00Z	2024-04-25T23:10:00Z	136.226.64.114	US	HTTPS	200	443	10.138.69.97	Suspicious Web Traffic	Adversary Infrastructure Interaction	AWS_VPC_Flow	prod_webserver	2024-04-25T23:00:00Z	waf_rule
2024-04-25T23:00:00Z	2024-04-25T23:10:00Z	165.225.240.79	NL	HTTPS	200	443	10.138.69.97	Suspicious Web Traffic	Adversary Infrastructure Interaction	AWS_VPC_Flow	prod_webserver	2024-04-25T23:00:00Z	waf_rule

Next steps: [Generate code with data](#) [View recommended plots](#) [New interactive sheet](#)

The screenshot shows the same Google Colab notebook with the following code cell:

```
data.info
```

The output shows the DataFrame information:

```
pandas.core.frame.DataFrame.info
def info(verbose: bool | None=None, buf: WriteBuffer[str] | None=None, max_cols: int | None=None,
memory_usage: bool | str | None=None, show_counts: bool | None=None) -> None
/usr/local/lib/python3.10/dist-packages/pandas/core/frame.py
Print a concise summary of a DataFrame.
This method prints information about a DataFrame including
the index dtype and columns, non-null values and memory usage.
```

The next cell shows the result of the `data.info()` command:

```
[25] data.shape
(282, 16)
```

The next cell shows the result of the `missing = data.isnull().sum()` command:

```
[26] missing = data.isnull().sum()
```

The next cell shows the result of the `data['time'] = pd.to_datetime(data['time'])` command:

```
[27] data['time'] = pd.to_datetime(data['time'])
data['time_diff'] = data['time'].diff().dt.total_seconds().fillna(0)
features = ['src_ip', 'time_diff']
ds = data[features]
ds.fillna(0, inplace=True)
print("missing values after preprocessing", missing)
```

The output shows the missing values after preprocessing:

```
missing values after preprocessing bytes_in      0
bytes_out      0
creation_time  0
end_time      0
src_ip        0
```

SYED FARIS HUSSAIN NAQVI

```
colab.research.google.com/drive/1148BxUrCrAKB2AjjK-bUDx_ADsQ3Z7authuser=0#scrollTo=O1mdpC-KaXm6

Course learning

Untitled8.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
RAM Disk
+ Gemini
[27] missing values after preprocessing bytes_in 0
bytes_out 0
creation_time 0
end_time 0
src_ip 0
src_ip_country_code 0
protocol 0
response.code 0
dst_port 0
dst_ip 0
rule_names 0
observation_name 0
source.meta 0
source.name 0
detection_types 0
dtype: int64
<ipython-input-27-2b60669005f9>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
ds.fillna(0, inplace=True)

[30] cat_col = ['src_ip', 'src_ip_country_code', 'protocol', 'dst_ip', 'rule_names', 'observation_name', 'source.meta', 'source.name', 'detection_types']
for col in cat_col:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])

[33] ds = data.drop(columns=['time', 'creation_time', 'end_time'])

[34] ds

bytes_in bytes_out src_ip src_ip_country_code protocol response.code dst_port dst_ip rule_names observation_name source.meta source.name detection_types time_diff
0 5602 12990 5 0 0 200 443 0 0 0 0 0 0 0 0.0
1 30912 18186 12 6 0 200 443 0 0 0 0 0 0 0 0.0
2 28506 13468 8 2 0 200 443 0 0 0 0 0 0 0 0.0
✓ 1s completed at 10:03 PM
```

```
colab.research.google.com/drive/1148BxUrCrAKB2AjjK-bUDx_ADsQ3Z7authuser=0#scrollTo=O1mdpC-KaXm6

Course learning

Untitled8.ipynb
File Edit View Insert Runtime Tools Help All changes saved
RAM Disk
+ Gemini
[36] from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
ds = scaler.fit_transform(ds)

[42] Suggested code may be subject to a license
ds = pd.DataFrame(ds)

ds

0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 -0.288219 -0.281223 -0.452528 -1.829139 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.103522
1 -0.282108 -0.260604 0.636851 1.055207 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.103522
2 -0.282689 -0.279344 0.014348 -0.867690 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.103522
3 -0.282197 -0.276161 -1.230656 1.055207 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.103522
4 -0.287996 -0.277678 0.325600 0.574483 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.103522
...
277 -0.279592 -0.280476 -0.919405 -0.867690 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.103522
278 -0.280693 -0.319733 0.481225 -0.386966 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.103522
279 5.796403 5.802781 -0.296903 1.055207 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.103522
280 -0.286187 -0.284655 -0.141277 -0.867690 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.103522
281 -0.287391 -0.309233 -0.608154 -1.348415 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.103522
282 rows x 14 columns

Next steps: Generate code with ds View recommended plots New interactive sheet

[46] X_train, X_test = train_test_split(ds, test_size=0.2, random_state=42)
model = RandomForestClassifier(X_train, y_train)
✓ 1s completed at 10:03 PM
```

SYED FARIS HUSSAIN NAQVI

```
Untitled8.ipynb - Colab
Task 4
How to create a github repository
colab.research.google.com/drive/1148BxUrCrAKB2Ajk-bUDx_ADsQ3Z7authuser=0fscrollTo=O1mdpC-KaXm6
Course learning

Untitled8.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share 8

Files
-
sample_data
CloudWatch_Traffic_W...

[46] X_train, X_test = train_test_split(ds, test_size=0.2, random_state=42)

print(f"Training set shape: {X_train.shape}")
print(f"Testing set shape: {X_test.shape}")

Training set shape: (225, 15)
Testing set shape: (57, 15)

[61] X_train.columns = X_train.columns.astype(str)
X_test.columns = X_test.columns.astype(str)

[58] X_train.columns = range(X_train.shape[1])
X_test.columns = range(X_test.shape[1])

Suggested code may be subject to a license [sun-1508/Test-spam-detection]
from sklearn.ensemble import IsolationForest

iso_forest = IsolationForest(contamination=0.02, random_state=42)
iso_forest.fit(X_train)

X_train['anomaly'] = iso_forest.predict(X_train)
X_test['anomaly'] = iso_forest.predict(X_test)

X_train['anomaly'] = X_train['anomaly'].map([1: 0, -1: 1])
X_test['anomaly'] = X_test['anomaly'].map([1: 0, -1: 1])

print("Training set anomalies:")
print(X_train[X_train['anomaly'] == 1])

print("Test set anomalies:")
print(X_test[X_test['anomaly'] == 1])

1s completed at 10:03 PM
```

```
Untitled8.ipynb - Colab
Task 4
How to create a github repository
colab.research.google.com/drive/1148BxUrCrAKB2Ajk-bUDx_ADsQ3Z7authuser=0fscrollTo=O1mdpC-KaXm6
Course learning

Untitled8.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share 8

Files
-
sample_data
CloudWatch_Traffic_W...

[62] Training set anomalies:
      0      1      2      3      4      5      6      7      8      9  \
232  4.101545  4.157770 -0.296903  1.055207  0.0  0.0  0.0  0.0  0.0  0.0
229  4.804988  4.117358 -0.296903  1.055207  0.0  0.0  0.0  0.0  0.0  0.0
245 -0.288900 -0.314751 -0.452523 -1.029139  0.0  0.0  0.0  0.0  0.0  0.0
267  5.794326  5.788548 -0.296903  1.055207  0.0  0.0  0.0  0.0  0.0  0.0
257  5.583737  5.676386 -0.296903  1.055207  0.0  0.0  0.0  0.0  0.0  0.0

      10      11      12      13      14      15 anomaly
232  0.0  0.0  0.0 -0.552648  0  1  1
229  0.0  0.0  0.0  0.345604  0  0  1
245  0.0  0.0  0.0  0.794730  0  0  1
267  0.0  0.0  0.0 -0.103522  0  1  1
257  0.0  0.0  0.0 -0.103522  1  1  1
Test set anomalies:
      0      1      2      3      4      5      6      7      8      9  \
279  5.796483  5.802781 -0.296903  1.055207  0.0  0.0  0.0  0.0  0.0  0.0
165 -0.288901 -0.178887 -1.075030  1.055207  0.0  0.0  0.0  0.0  0.0  0.0

      10      11      12      13      14      15 anomaly
279  0.0  0.0  0.0 -0.103522  1  1  1
165  0.0  0.0  0.0  16.514141  1  0  1

[70] Suggested code may be subject to a license [Albanobbb/eq | AeshanaShahindia/crypto-currency-price-prediction]
print("Test set anomalies:")
print(X_test[X_test['anomaly'] == 1])

import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
column_name = 'time_difference'

plt.figure(figsize=(12, 6))
plt.plot(X_test.index, X_test['3'], label='Time Difference')
plt.scatter(X_test[X_test['anomaly'] == 1].index, X_test[X_test['anomaly'] == 1]['3'], color='r', label='Anomaly')
plt.legend()
plt.show()

Test set anomalies:
1s completed at 10:03 PM
```

SYED FARIS HUSSAIN NAQVI

