mitre tool

page1

page2

stix tool

Mitre CSV

CyberGPT

About

CyberGPT is a smart AI assistant for cyber security analysts.

Your documents

Select your PDFs here

Drag and drop files here

Limit 200MB per file

Browse files

Submit

New Chat

Made with ❤️ by GeekWeek Team 5.2

Mitre CSV

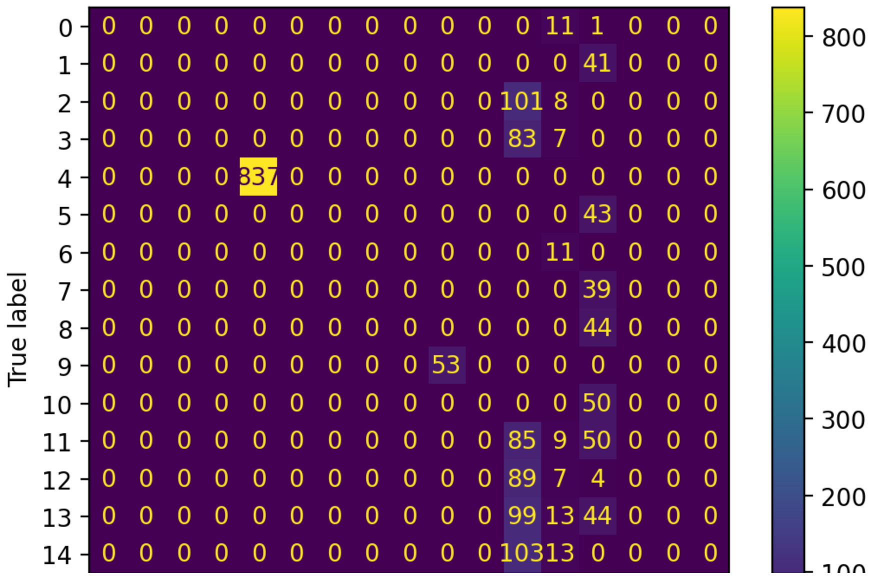
This page loads information from MITRE ATT&CK for the LLM to query.

Use python to build a machine learning model to predict the Mitigation column using text from the Description fields, using only the first 10000 rows of the data frame. Then evaluate the model and output the accuracy results. Also plot a Confusion Matrix display using an sklearn ConfusionMatrixDisplay, with the generated matplotlib figure assigned to variable fig. Include the import statements in your python code. Don't set the display_labels parameter.

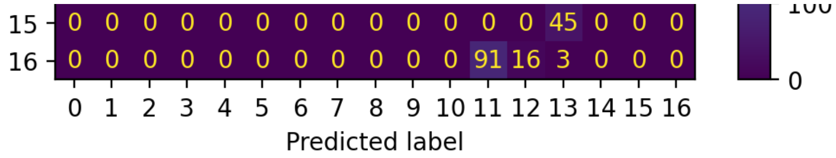
Conversation

🙂 Use python to build a machine learning model to predict the Mitigation column using text from the Description fields, using only the first 10000 rows of the data frame. Then evaluate the model and output the accuracy results. Also plot a Confusion Matrix display using an sklearn ConfusionMatrixDisplay, with the generated matplotlib figure assigned to variable fig. Include the import statements in your python code. Don't set the display_labels parameter.

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, ConfusionMatrixD

# Preprocess text data
df = df[['Technique Name', 'Mitigation Name']]
df = df.dropna()
df = df[:10000]
df['Technique Name'] = df['Technique Name'].str.lower()

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['Technique Name'], df['

# Vectorize text data
vectorizer = TfidfVectorizer(stop_words='english')
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)

# Build machine learning model
model = LogisticRegression()
model.fit(X_train, y_train)

# Evaluate model
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)

# Plot confusion matrix
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
fig = disp.plot()
```



0	0	0	0	0	0	0	0	0	0	0	0	0	11	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	101	8	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	83	7	0	0	0	0
4	0	0	0	0	837	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	43	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	44	0	0	0
9	0	0	0	0	0	0	0	0	0	53	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	85	9	50	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	89	7	4	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	99	13	44	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	103	13	0	0	0	0



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
15	0	0	0	0	0	0	0	0	0	0	0	0	0	45	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	91	16	3	0	0	0

Download

Made with Streamlit