

...

mitre tool

page1

page2

ctiv 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 dataframe. 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. Return the

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 dataframe. 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. Return the python code instead of a description, don't write plt.show() at the end.

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay

df = df.head(10000)
df = df.dropna(subset=['Software Description', 'Mitigation Description'])

X = df['Software Description']
y = df['Mitigation Description']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

vectorizer = TfidfVectorizer(stop_words='english')
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)

model = LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")

cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
fig = disp.plot()
```

