Hack-o-mania Project

Nitin Kumar
Team Alpha

Introduction

In the project, I have tried to adress the cyber threat of phishing attacks and frauds which are delivered via emails.
Harmful viruses and malaware enter into our computer through mails if we unknowingly or naively download attachments and /or click on links given in the mail without suspecting and so become prey of the phishing attack.

Project working idea

The project is a machine learning model that works on the dataset taken from two different websites and used it to train a Bernoulli Naive Bayes Model for the classification with high accuracy.

phishing_model.ipynb

E: > Hack-o-mania-advitiya- > phishing_model.ipynb > so now we will be start working on my project for the hackathon

+ Code + Markdown | Run All | Clear All Outputs | Outline ··· Select Kernel

[28] Python

```python
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(text_vectorized,df['Type'],test_size=0.20)
```

[29] Python

+ Code + Markdown

```python
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

[30] Python

```
(125, 7370)
(32, 7370)
(125,)
(32,)
```

```python
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB,BernoulliNB
# svm = SVC()
# svm= MultinomialNB()
svm=BernoulliNB()
svm.fit(X_train,y_train)
```

[34] Python

```
BernoulliNB
BernoulliNB()
```

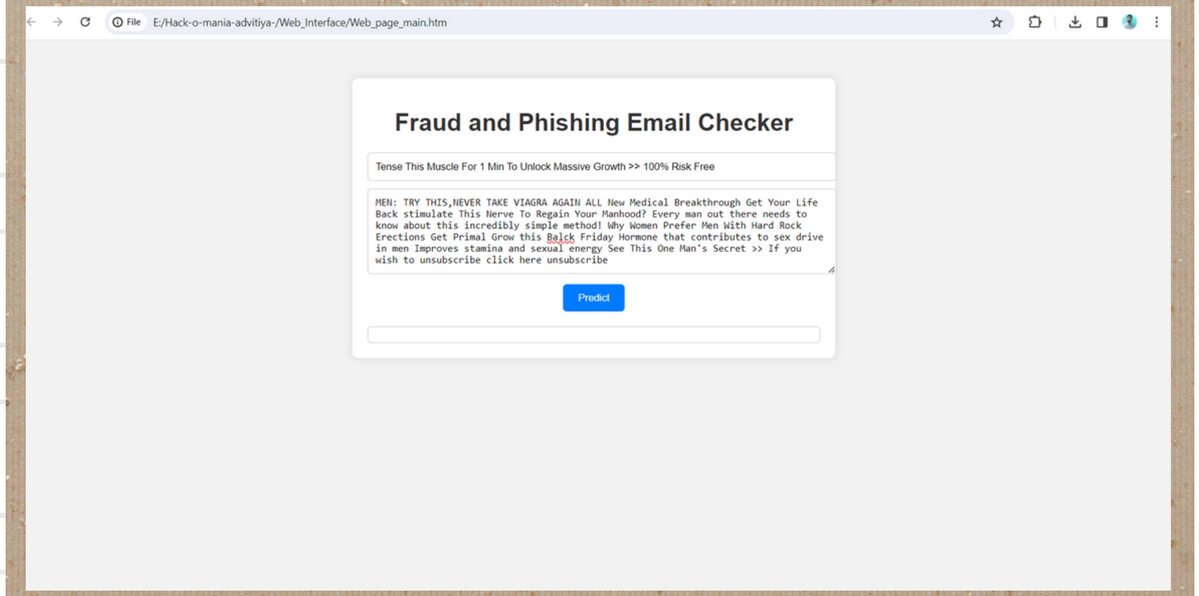
Process

- First step was to clean the data. For which I used the sklearn.preprocessing library.
- Then reading of the text and converting it into tokens and then finally a vector which represents all the important words of the email.
- Then the next step, is the one of the most crucial steps, i.e. selecting the model based on the type of data and the final findings.
- Then I trained my model over the dataset and then used the joblib library to serialize it as a fixed modal which is compatible with both app and web applications.
- Then i developed a basic web interface for the ML model which receives subject and body of an email and then gives the verdict about its potentail of being a fishing emaiil.

Web Interface

Front-end

The front-end designing of the page is done by using HTMl code and basic CSS code. The main page consists of two boxes to input the subject and text of the mail and the predict button and the output log .



Back-end

This part consists of the javascript and python based files to link the ML model to the predict button and its input and output features.

98.8%

```
accuracy= 100*accuracy_score(y_test,y_pred,normalize= True)
print(accuracy)
print(classification_report(y_test,y_pred))
```

```
98.85268471776044
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      1052
           1       1.00      0.98      0.99      1127

    accuracy                           0.99      2179
   macro avg       0.99      0.99      0.99      2179
weighted avg       0.99      0.99      0.99      2179
```

By applying different models and calculating their accuracy score, I finally used the Bernoulli Naive Bayes model from the sklearn library of python and achieved an impressive score of 98%.

Thank you!