

Kötü amaçlı URL'leri Algılayan Sistem Tasarımı

Hazırlayanlar

17110131006-Mehmet OKYAY

18110131505-Abdourazak ALİ EGUEH

Projenin Amacı

- Projemiz bilgisayar ve ağ güvenliğinde herhangi bir internet sitesinin güvenilir olup olmadığını tahmin eden bir makine öğrenmesi uygulamasıdır. Bu uygulama ile veri setinde bulunan güvenilir veya güvenilmeyen siteler makine öğrenmesi yolu ile eğitilip yeni listede olmayan yeni bir domain adresinin güvenilirlik güvenilemeyeceği konusunda fikir sahibi olmamızı sağlamaktadır.

Projenin Hazırlanma Aşamaları

- Veri Toplama
- Veri ön işleme
- Verinin eğitilmesi
- Verinin test Edilmesi
- Eğitimin Değerlendirilmesi ve Geliştirilmesi

Veriyi Toplama

- Projemizde kullanılan veri seti Kaggle üzerinden indirilip kullanılmıştır. Veri setimiz **Unnamed**, **url**, **label**, **result** olmak üzere 4 niteliğe sahip ve 450176 adet kayıttan oluşmaktadır.

```
In [74]: import pandas as pd
import numpy as np
import random
import re

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

```
In [72]: dataset = pd.read_csv("urldata.csv")
```

```
In [75]: dataset
```

```
Out[75]:
```

	Unnamed: 0	url	label	result
0	0	https://www.google.com	benign	0
1	1	https://www.youtube.com	benign	0
2	2	https://www.facebook.com	benign	0
3	3	https://www.baidu.com	benign	0
4	4	https://www.wikipedia.org	benign	0
...
450171	450171	http://ecct-it.com/docmmnn/aptgd/index.php	malicious	1
450172	450172	http://faboleena.com/ajs/infortis/jquery/plugin...	malicious	1
450173	450173	http://faboleena.com/ajs/infortis/jquery/plugin...	malicious	1
450174	450174	http://atualizapj.com/	malicious	1
450175	450175	http://writeassociate.com/test/Portal/inicio/l...	malicious	1

Veri Önışleme

- Veri setindeki label alanındaki değeri result değerlerine göre yeniden etiketlendi.
- Veri setinde null değere sahip bir kayıtlın olup olmadığına bakılarak eğitim modelinde hata oranının artmasına neden olabilecek değerler tespit edildi.

```
In [66]: dataset.replace({'label':{'malicious':'kötü_amaçlı', 'benign':'iyi_amaçlı'}},inplace=True)
```

```
dataset.isnull()
```

Unnamed: 0	url	label	result
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
...
450171	False	False	False
450172	False	False	False
450173	False	False	False
450174	False	False	False
450175	False	False	False

```
print (dataset.isnull().sum())
```

```
Unnamed: 0    0  
url           0  
label         0  
result        0  
dtype: int64
```


Veri Önışleme

- Web sitelerinin domain adresleri ile güvenilir veya güvenilirmez site ilişkisini kurabilmek için url adreslerindeki geçen kelimeleri gruplarını ayırmak gerekir. Yani url de geçen kelime ile label arasında bir bağlantı kurulacaktır. Bunun için Python'da bir fonksiyon tanımladık.
- Ayrıca bu bağlantıyı kurmak için re(regular expression) kütüphanesi tanımlandı.

```
def makeTokens(f):  
    tkns_BySlash = str(f.encode('utf-8')).split('/')  
    total_Tokens = []  
    for i in tkns_BySlash:  
        tokens = str(i).split('-')  
        tkns_ByDot = []  
        for j in range(0, len(tokens)):  
            temp_Tokens = str(tokens[j]).split('.')  
            tkns_ByDot = tkns_ByDot + temp_Tokens  
            total_Tokens = total_Tokens + tokens + tkns_ByDot  
    total_Tokens = list(set(total_Tokens))  
    if 'com' in total_Tokens:  
        total_Tokens.remove('com')  
    return total_Tokens
```

Eğitim Modeli Oluşturma

- Bağımlı(y) ve bağımsız değişkenler(X) veri setinden ilgili sütunları seçilerek alındı ve daha sonra metni ayıklama fonksiyonu TF x IDF Skorlama Modeline ile kelimelerin sıklık ve baskınlık oranları ile bağımlı değişken arasındaki ilişki kurmak amacıyla modelin eğitimi lojistik regresyon algoritması ile gerçekleştirildi.

```
y = dataset["label"]

url_list = dataset["url"]

vectorizer = TfidfVectorizer(tokenizer=makeTokens)

X = vectorizer.fit_transform(url_list)

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

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

LogisticRegression()
```


Projenin Test Edilmesi

- Veri setinin eğitilmesinden sonra tahmin işlemleri için random olarak belirlenen web sitelerinin test edilmesi aşamasıdır.

```
X_predict = ["https://bm.ksu.edu.tr/", "https://www.kervaniho.meb.k12.tr/",  
"https://www.w3schools.com/python/numpy/default.asp", "https://www.udemy.com/"]
```

```
X_predict
```

```
['https://bm.ksu.edu.tr/',  
 'https://www.kervaniho.meb.k12.tr/',  
 'https://www.w3schools.com/python/numpy/default.asp',  
 'https://www.udemy.com/']
```

```
X_predict = vectorizer.transform(X_predict)  
New_predict = logit.predict(X_predict)
```

```
print(New_predict)
```

```
['iyi_amaçli' 'iyi_amaçli' 'iyi_amaçli' 'iyi_amaçli']
```

```
X_predict1 = ["www.buyfakebillsonlinee.blogspot.com",  
"www.unitedairlineslogistics.com",  
"www.stonehousedelivery.com",  
"www.silkroadmeds-onlinepharmacy.com" ]
```

```
X_predict1 = vectorizer.transform(X_predict1)  
New_predict1 = logit.predict(X_predict1)  
print(New_predict1)
```

```
['kötü_amaçli' 'kötü_amaçli' 'kötü_amaçli' 'kötü_amaçli']
```


Projenin Değerlendirilmesi

- Yaptığımız projenin %99'un üzerinde bir başarı ile web sitelerinin domain adreslerine bakarak bu sitenin güvenli veya güvenli olmayan şeklinde tahminde bulunması bilgisayar ve ağ güvenliğinde bilgi güvenliği açısından yorumlandığında gizlilik ilkesine uygudur

```
print("Başarı: ",logit.score(X_test, y_test))
```

```
Başarı: 0.9948020791683326
```