Fake News Detection

DATA MINING PROJECT

Ramazan Emre Keskin | 030716003 | 04.01.2021

Introduction

Technology is also advancing in the developing world Social media platforms that come into our lives with this technology have also expanded their communication channels. This greatly increases the dissemination of news. The growth of this spread also causes fake news to increase and spread easily. In this study, I developed an algorithm that detects whether a news is true or false by using the news I received from social media, which I am sure is true or false.

Literature Review

We live in an age where computers can understand everything and do many things for humans. As a necessity of this, we have to explain our own language to computers. For this reason, natural language processing (NLP) applications have become widespread today. Many institutions and governments have started to use natural language processing applications in transactions such as ready bot applications, comment analysis, correspondence analysis. I also tried to use nlp in this project, albeit at an introductory level.

Structure of the Solution Proposed

I collected my data on social media, twitter. I have received 1000 news stories from the Twitter account of the "trthaber" institution that I have determined as reliable and labeled them correctly. I received 1000 pieces of news from the less reliable 'zaytung' twitter account and labeled it incorrectly. After first processing the data set I prepared, I processed the text preprocessing and then converted the news into numerical data by determining the term numbers with the countvectorization method. I taught the machine by giving the converted data as input to the logistic regression algorithm, which is a machine learning algorithm.

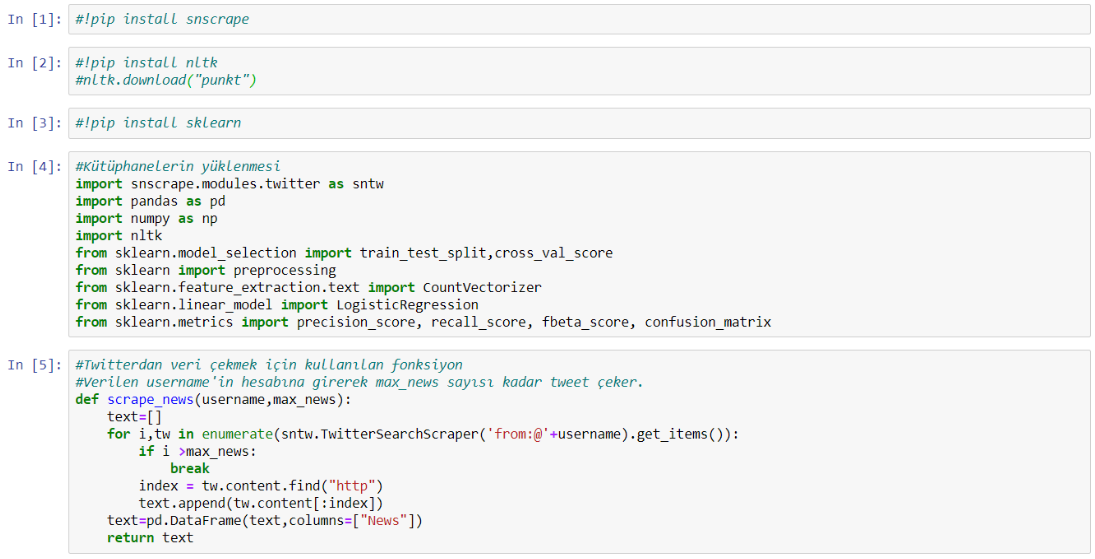
How To Use The Software ? What Are The Requrements To Run ?

While writing the code, I wrote it in functions to make it easier to use. In order 2orm y code to work, the basic python library must be installed first. In addition, libraries of functions used in functions must be loaded. I mentioned the loading of these libraries in comments at the beginning of my code and I made the code explanations in comments.

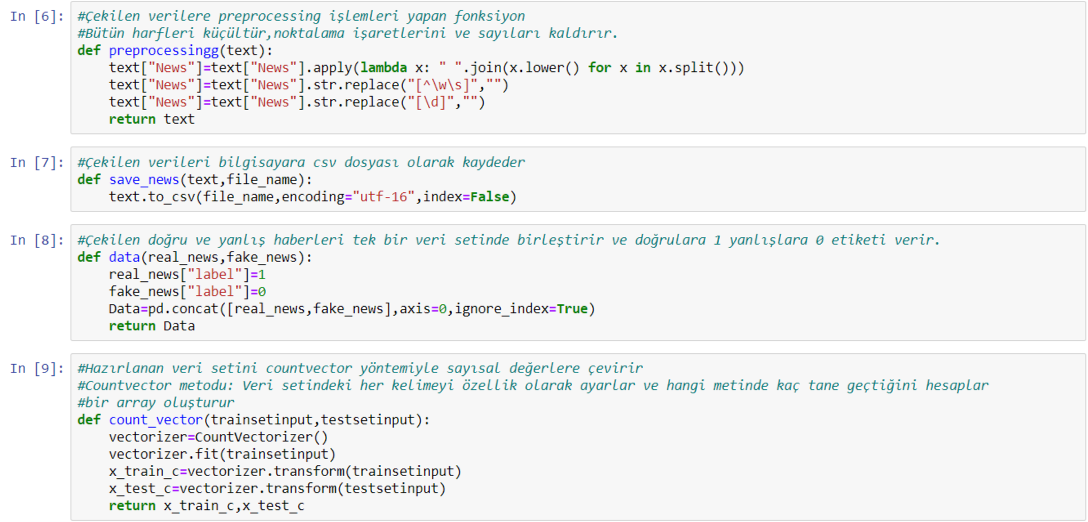
RFC for the Framework Algorithm

I used the “Countvector” method while writing the application. In the “Countvector” method, each word in the dataset is determined as a column. Each word is searched in every news and is valued as the number of occurrences in the text.

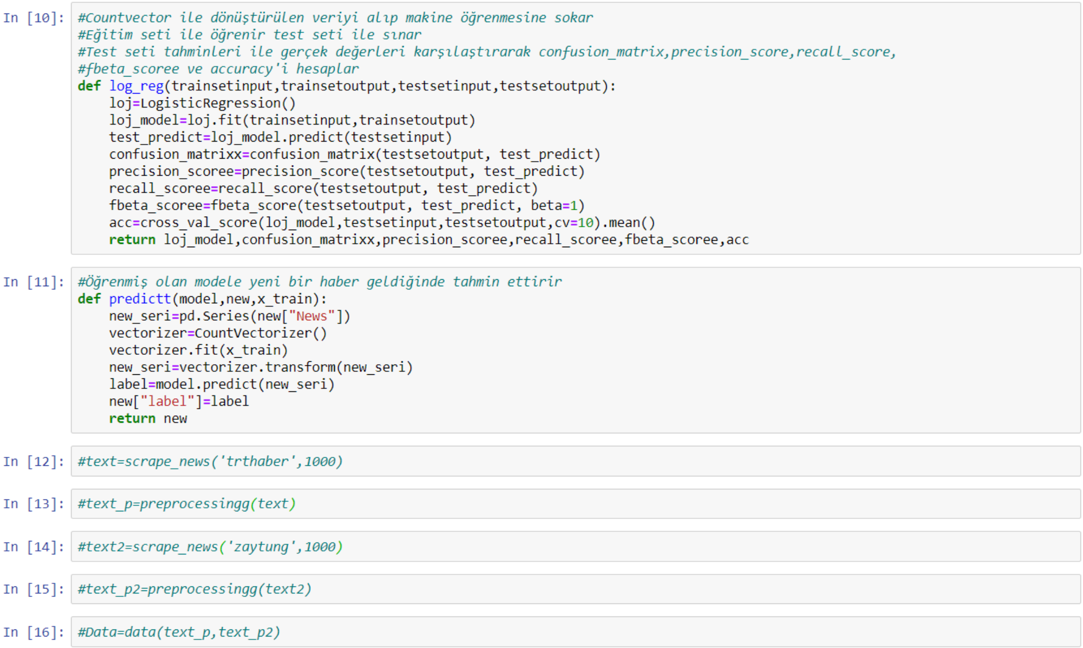
Runtime Examples



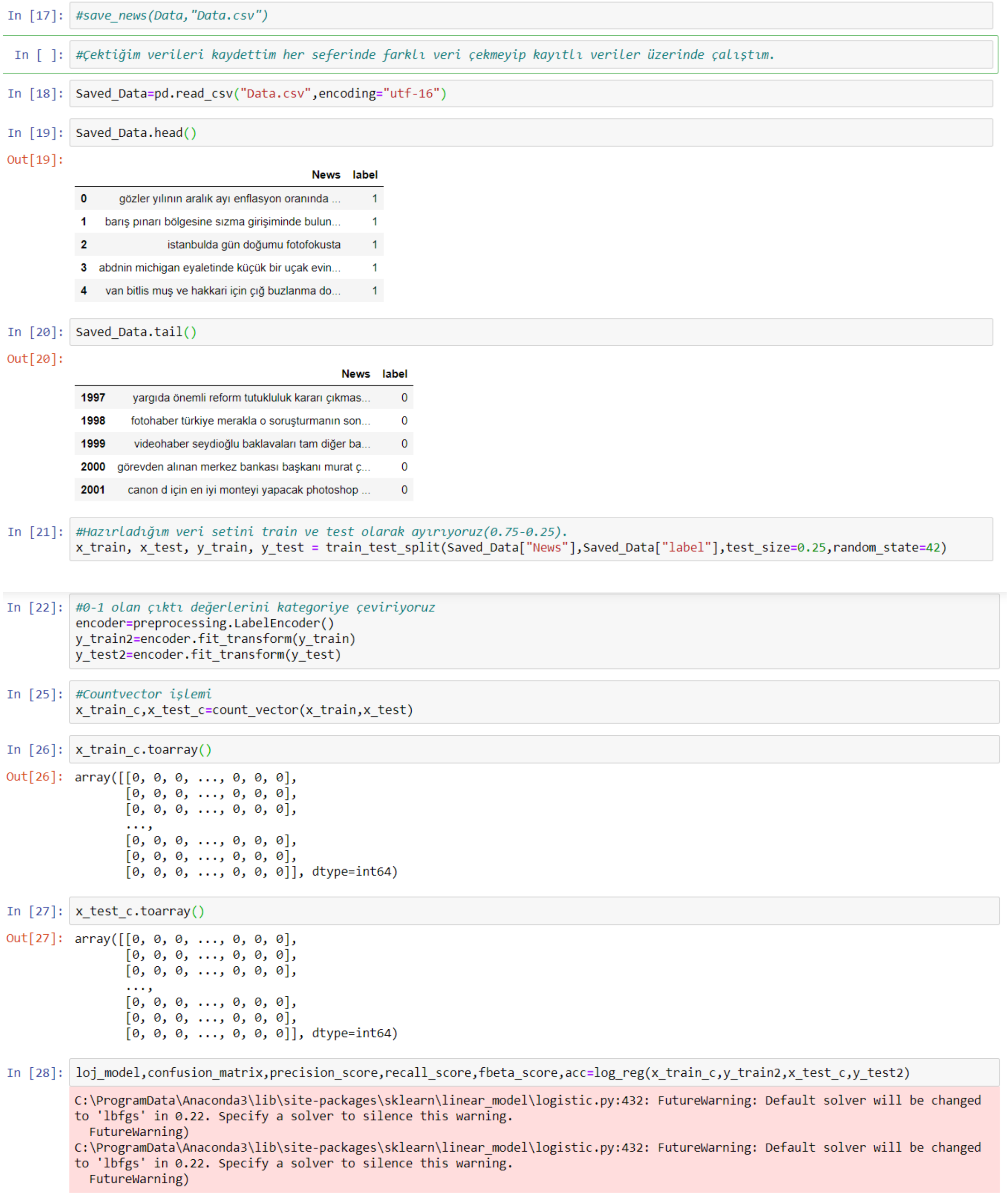
```
In [1]: #!pip install snscrape

In [2]: #!pip install nltk
        #nltk.download("punkt")

In [3]: #!pip install sklearn

In [4]: #Kütüphanelerin yüklenmesi
        import snscrape.modules.twitter as sntw
        import pandas as pd
        import numpy as np
        import nltk
        from sklearn.model_selection import train_test_split,cross_val_score
        from sklearn import preprocessing
        from sklearn.feature_extraction.text import CountVectorizer
        from sklearn.linear_model import LogisticRegression
        from sklearn.metrics import precision_score, recall_score, fbeta_score, confusion_matrix

In [5]: #Twitterdan veri çekmek için kullanılan fonksiyon
        #Verilen username'in hesabına girerek max_news sayısı kadar tweet çeker.
        def scrape_news(username,max_news):
            text=[]
            for i,tw in enumerate(sntw.TwitterSearchScraper('from:@'+username).get_items()):
                if i >max_news:
                    break
                index = tw.content.find("http")
                text.append(tw.content[:index])
            text=pd.DataFrame(text,columns=["News"])
            return text
```



```
In [6]:  #Çekilen verilere preprocessing işlemleri yapan fonksiyon
         #Bütün harfleri küçültür,noktalama işaretlerini ve sayıları kaldırır.
         def preprocessingg(text):
             text["News"]=text["News"].apply(lambda x: " ".join(x.lower() for x in x.split()))
             text["News"]=text["News"].str.replace("[^\w\s]","")
             text["News"]=text["News"].str.replace("[\d]","")
             return text

In [7]:  #Çekilen verileri bilgisayara csv dosyası olarak kaydeder
         def save_news(text,file_name):
             text.to_csv(file_name,encoding="utf-16",index=False)

In [8]:  #Çekilen doğru ve yanlış haberleri tek bir veri setinde birleştirir ve doğrulara 1 yanlışlara 0 etiketi verir.
         def data(real_news,fake_news):
             real_news["label"]=1
             fake_news["label"]=0
             Data=pd.concat([real_news,fake_news],axis=0,ignore_index=True)
             return Data

In [9]:  #Hazırlanan veri setini countvector yöntemiyle sayısal değerlere çevirir
         #Countvector metodu: Veri setindeki her kelimeyi özellik olarak ayarlar ve hangi metinde kaç tane geçtiğini hesaplar
         #bir array oluşturur
         def count_vector(trainsetinput,testsetinput):
             vectorizer=CountVectorizer()
             vectorizer.fit(trainsetinput)
             x_train_c=vectorizer.transform(trainsetinput)
             x_test_c=vectorizer.transform(testsetinput)
             return x_train_c,x_test_c
```



```
In [10]: #Countvector ile dönüştürülen veriyi alıp makine öğrenmesine sokar
         #Eğitim seti ile öğrenir test seti ile sınar
         #Test seti tahminleri ile gerçek değerleri karşılaştırarak confusion_matrix,precision_score,recall_score,
         #fbeta_scoree ve accuracy'i hesaplar
         def log_reg(trainsetinput,trainsetoutput,testsetinput,testsetoutput):
             loj=LogisticRegression()
             loj_model=loj.fit(trainsetinput,trainsetoutput)
             test_predict=loj_model.predict(testsetinput)
             confusion_matrixx=confusion_matrix(testsetoutput, test_predict)
             precision_scoree=precision_score(testsetoutput, test_predict)
             recall_scoree=recall_score(testsetoutput, test_predict)
             fbeta_scoree=fbeta_score(testsetoutput, test_predict, beta=1)
             acc=cross_val_score(loj_model,testsetinput,testsetoutput,cv=10).mean()
             return loj_model,confusion_matrixx,precision_scoree,recall_scoree,fbeta_scoree,acc

In [11]: #Öğrenmiş olan modele yeni bir haber geldiğinde tahmin ettirir
         def predictt(model,new,x_train):
             new_seri=pd.Series(new["News"])
             vectorizer=CountVectorizer()
             vectorizer.fit(x_train)
             new_seri=vectorizer.transform(new_seri)
             label=model.predict(new_seri)
             new["label"]=label
             return new

In [12]: #text=scrape_news('trthaber',1000)

In [13]: #text_p=preprocessingg(text)

In [14]: #text2=scrape_news('zaytung',1000)

In [15]: #text_p2=preprocessingg(text2)

In [16]: #Data=data(text_p,text_p2)
```



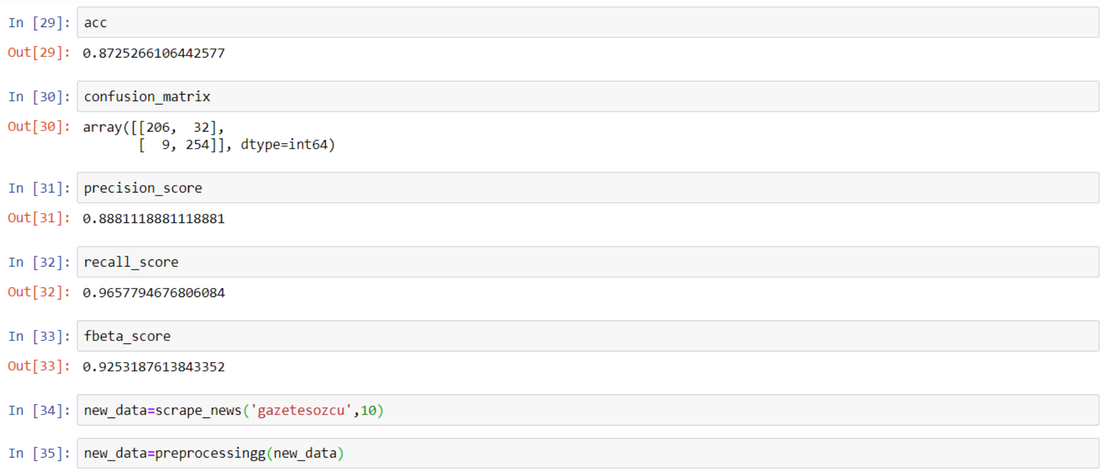
```
In [17]: #save_news(Data,"Data.csv")

In [ ]: #Çektiğim verileri kaydettim her seferinde farklı veri çekmeyip kayıtlı veriler üzerinde çalıştım.

In [18]: Saved_Data=pd.read_csv("Data.csv",encoding="utf-16")

In [19]: Saved_Data.head()
Out[19]:
```

	News	label
0	gözler yılının aralık ayı enflasyon oranında ...	1
1	barış pınarı bölgesine sızma girişiminde bulun...	1
2	istanbulda gün doğumu fotofokusta	1
3	abdnin michigan eyaletinde küçük bir uçak evin...	1
4	van bitlis muş ve hakkari için çığ buzlanma do...	1

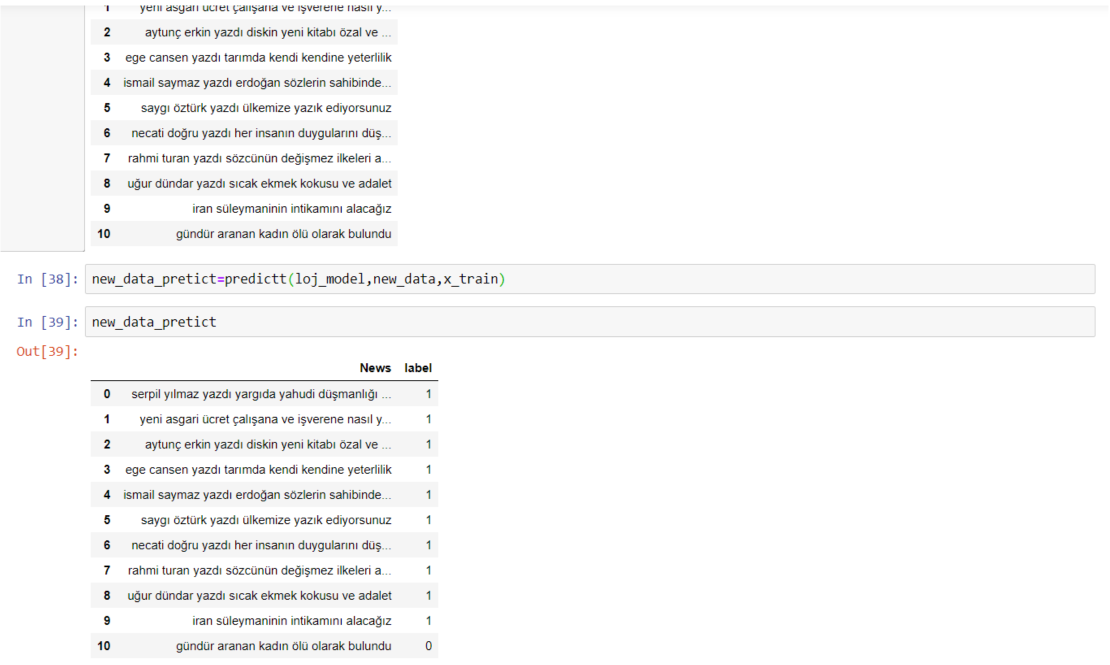
```
In [20]: Saved_Data.tail()
Out[20]:
```

	News	label
1997	yargıda önemli reform tutukluluk kararı çıkmas...	0
1998	fotohaber türkiye merakla o soruşturmanın son...	0
1999	videohaber seydioğlu baklavaları tam diğer ba...	0
2000	görevden alınan merkez bankası başkanı murat ç...	0
2001	canon d için en iyi monteyi yapacak photoshop ...	0

```
In [21]: #Hazırladığım veri setini train ve test olarak ayırıyoruz(0.75-0.25).
         x_train, x_test, y_train, y_test = train_test_split(Saved_Data["News"],Saved_Data["label"],test_size=0.25,random_state=42)

In [22]: #0-1 olan çıktı değerlerini kategoriye çeviriyoruz
         encoder=preprocessing.LabelEncoder()
         y_train2=encoder.fit_transform(y_train)
         y_test2=encoder.fit_transform(y_test)

In [25]: #Countvector işlemi
         x_train_c,x_test_c=count_vector(x_train,x_test)

In [26]: x_train_c.toarray()

Out[26]: array([[0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                ...,
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0]], dtype=int64)

In [27]: x_test_c.toarray()

Out[27]: array([[0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                ...,
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0]], dtype=int64)

In [28]: loj_model,confusion_matrix,precision_score,recall_score,fbeta_score,acc=log_reg(x_train_c,y_train2,x_test_c,y_test2)

         C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: Default solver will be changed
         to 'lbfgs' in 0.22. Specify a solver to silence this warning.
           FutureWarning)
         C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: Default solver will be changed
         to 'lbfgs' in 0.22. Specify a solver to silence this warning.
           FutureWarning)
```



```
In [29]: acc
Out[29]: 0.8725266106442577

In [30]: confusion_matrix
Out[30]: array([[206,  32],
               [  9, 254]], dtype=int64)

In [31]: precision_score
Out[31]: 0.8881118881118881

In [32]: recall_score
Out[32]: 0.9657794676806084

In [33]: fbeta_score
Out[33]: 0.9253187613843352

In [34]: new_data=scrape_news('gazetesozcu',10)

In [35]: new_data=preprocessingg(new_data)
```



1	yeni asgari ücret çalışana ve işverene nasıl y...
2	aytunç erkin yazdı diskin yeni kitabı özal ve ...
3	ege cansen yazdı tarımda kendi kendine yeterlilik
4	ismail saymaz yazdı erdoğan sözlerin sahibinde...
5	saygı öztürk yazdı ülkemize yazık ediyorsunuz
6	necati doğru yazdı her insanın duygularını düş...
7	rahmi turan yazdı sözcünün değişmez ilkeleri a...
8	uğur dündar yazdı sıcak ekmek kokusu ve adalet
9	iran süleymaninin intikamını alacağız
10	gündür aranan kadın ölü olarak bulundu

```
In [38]: new_data_pretict=predictt(loj_model,new_data,x_train)

In [39]: new_data_pretict
Out[39]:
```

	News	label
0	serpil yılmaz yazdı yargıda yahudi düşmanlığı ...	1
1	yeni asgari ücret çalışana ve işverene nasıl y...	1
2	aytunç erkin yazdı diskin yeni kitabı özal ve ...	1
3	ege cansen yazdı tarımda kendi kendine yeterlilik	1
4	ismail saymaz yazdı erdoğan sözlerin sahibinde...	1
5	saygı öztürk yazdı ülkemize yazık ediyorsunuz	1
6	necati doğru yazdı her insanın duygularını düş...	1
7	rahmi turan yazdı sözcünün değişmez ilkeleri a...	1
8	uğur dündar yazdı sıcak ekmek kokusu ve adalet	1
9	iran süleymaninin intikamını alacağız	1
10	gündür aranan kadın ölü olarak bulundu	0

Result And Interpretation

The model taught with the dataset we have predicted new news. Although the Accuracy value is 0.87, we may not be able to achieve this success in new news due to our limited data set.

Future Work

The data set can be increased to improve this study. Stopwords in English (meaningless and highly repetitive words) can be edited for Turkish and applied to the data set. Different machine learning algorithms can be tried. However, as the data set will increase, there will be applications that require high processing. It is necessary to use computers with high hardware.

References:

https://towardsdatascience.com/14-popular-evaluation-metrics-in-machine-learning-33d9826434e4

https://www.akademikkaynak.com/literatur-taramasi-nedir-ve-nasil-yapilir.html

https://www.youtube.com/watch?v=Fp4AnPVDRMk

https://www.udemy.com/course/python-egitimi/