BIM 407 Veri Madenciliği Proje Raporu

Ali Köse

030117090

Proje Kod Adı : MEDOYA

Proje Adı: Mesaj Doğru mu Yanlış mı

Giriş

Proje kapsamında sosyal medya platformları üzerinden veri toplanarak sosyal medya platformlarından en az biri kullanılıp veriler toplanarak proje sahibinin belirlediği bir haber konusu üzerinden yapılan haberlerin doğru veya yanlış şekilde, daha sonra yanlış haberlerin bilinçli mi yoksa bilinçsiz olarak yanlış haber yapıldığı konusunda tespit edilmesi istenmiştir. Proje konusu seçimi ve sosyal medyadan veri toplanması tamamen proje sahibine bırakılmıştır.

Proje Kapsamında belirlenecek konular şunlardır:

1. Sosyal medya platformunun belirlenmesi
2. Haber içeriğinin belirlenmesi
3. Haber konusuna göre belirlenen sosyal medya platformu üzerinden verinin çekilmesi
4. Elde edilen ham verinin alan bilgisi öğrenilmesi
5. Elde Edilen ham verinin proje kapsamına yönelik hazırlanması
6. Hazırlanan verinin çözüme yönelik modelin inşa edilmesi
7. Modelin Değerlendirilmesi

Proje kapsamında Twitter platformu üzerinden haber konusu ‘Asgari Ücret’ hakkında verilerinn elde edilmesi,elde edilen verilerin Doğal Dil işleme alanında uygulanan Kelime çantası modeline yönelik,çeşitli kütüphanelere sahip olan Python dilindeki kütüphanelerden yararlanılarak verinin incelenmesi,kullanılacak kısımların belirlenmesi, verinin hazırlanması, modelin oluşturulması ve değerlendirilmesi gerçekleştirilmiştir.

Veri Seti :

Proje kapmasında veri toplayacağım sosyal medya platformu olarak Twitter seçilmiştir. Twitter platformunun seçilmesinin başlıca sebepleri şunlardır:

- Metin konusunda herkese eşit hak -140 karakter- tanınması
- Toplumsal konularda halkın,siyasi liderlerin güncel haberleri ilk olarak bu sosyal medya platformu üzerinden görüşlerini bildirmesi
- Twitter platformu üzerinde yapılan paylaşımların #hashtag ile konu kapsamında doğal bir filtre oluşturması
- Twitter api aracılığıyla veri setini kolaylıkla oluşturma imkanı sağlayacağının düşünülmesi

Proje kapsamında seçilen haber konusu her yılın sonuna gelinmesi ile gündem olan ‘Asgari Ücret’ mesajlarıdır. Asgari Ücretin seçilme sebebi ise resmi haberin yapılma

tarihinden önce ve sonrasında üzerinde siyasi liderlerin, bakanlıkların belediyelerin ve halkın görüş bildirme seviyesi gayet yüksek olması etki etmiştir.

Ham Verinin Oluşturulması:

Ham veri, Veri madenciliği açısından kaynağından toplanıp istiflenmiş ama analize hazır hale getirilmemiş verilerdir.

Twitter platformunun geliştiricilere özel olarak Twitter Apı ile python in bir kütüphanesi olan Tweepy kullanılması düşünülüp gerekli araştırmalar ışığında karar verilmiştir.Fakat proje süresince onay alınamadığı için açık kaynak bir kütüphane olan Twint kullanılmıştır.

Veriler Jupyter notebook üzerinden python dilinde, açık kaynaklı Twint kütüphanesinin sunmuş olduğu parametreler üzerinden projeye yönelik ham veri setleri oluşturulmuştur.

Veri Setinin incelenmesi:

Veri setinin incelenip, elde edilen veri üzerinden kullanılacak niteliklerin belirlenmesi ve diğer niteliklerle nasıl kullanılacağı veri madenciliği projesi için çok büyük önem arz eder. Ders kapsamı boyunca bu hususta çokça bahsedilmiştir. Elde edilen veri setinde nitelikler ve örnek içerikleri şu şekildedir:

1. İd : 1343597180934909953
2. conversation_id: 1343597180934909953
3. created_at: 2020-12-28 19:38:25 Türkiye Standart Saati
4. date: 2020-12-28
5. time: 19:38:25
6. timezone : 300
7. user_id: 244276679
8. username: mehmetkesimoglu
9. name : Mehmet Siyam Kesimoğlu
10. place : nan
11. tweet : Kırklareli Belediyesi çalışanlarının aldığı en düşük ücret 3.520 TL’dir. 😊 #asgariucret https://t.co/wIWoyYZltt
12. language: tr
13. mentions : []
14. urls: []
15. photos : ['https://pbs.twimg.com/media/EqVqgyMXIAAY__c.jpg']
16. replies_count : 938
17. retweets_count : 3910

18. likes_count : 61135
19. hashtags : ['asgariucret']
20. cashtags: []
21. link: https://twitter.com/mehmetkesimoglu/status/1343597180934909953
22. retweet: False
23. quote_url : nan
24. video: 1
25. thumbnail : https://pbs.twimg.com/media/EqVqgyMXIAAY__c.jpg
26. near: nan
27. geo: nan
28. source: nan
29. user_rt_id: nan
30. user_rt : nan
31. retweet_id: nan
32. reply_to : []
33. retweet_date : nan
34. translate: nan
35. trans_src : nan
36. trans_dest : nan

Elde edilen veri seti üzerinde tarih, dil, retweet, replies ve like sınırlandırması yapılmış kaliteli veri elde edileceği düşünülmüştür. Burada bahsedilen kaliteli veri, haber niteliği taşıyıp taşımadığıdır.

Bir yazının haber niteliği taşıması, sosyal medya nazarında diğer kullanıcılarla etkileşime geçmesi, belirli bir kesimce kabul görmüş olması baz alınmıştır.

Veri seti üzerinde yapılan alan incelemesi sonucunda elde edilen bazı niteliklerin tüm veri seti için boş değer ürettiği; bazı sütunların ise benzer bilgileri içerdiği görülmüştür.

Bu durum araştırılmış ve Twitter ' ın geliştiriciler için sağladığı api harici yöntemlerle veri elde edilmesinin önüne geçmeye çalışmaktadır. Bu sebepten dolayı veri setinde bazı niteliklere artık erişilememektedir. Sonuç olarak kullanılacak sütunlar şu şekilde belirlenmiştir:

1. date: 2020-12-28 ,

7 username: mehmetkesimoglu

11. tweet : Kırklareli Belediyesi çalışanlarının aldığı en düşük ücret 3.520 TL'dir. 😊 #asgariucret https://t.co/wIWoyYZltt

16. replies_count : 938

17. retweets_count : 3910

18. likes_count : 61135

Veri Önişleme:

Veri seti içinde belirlenen nitelikler ile modele yönelik bir hazırlık yapılması şarttır. Belirlenen nitelikler Twint kütüphanesinin veri setinde belirli filtreleme işlemlerine yarayan: Filter_retweets, Min_likes, Min_retweets, Min_replies, Search, Since ve Until metotları kullanılarak gerçekleştirilmiştir. Bu metotların kullanımları çeşitli şekillerde deneme-yanılma yöntemi ile ihtiyaç olduğu görülüp kullanılmıştır.

Asgari Ücret her sene yaşadığımız şartlar doğrultusunda değişikliğe uğramaktadır. Bu sebeple zaman sınırı konulmadığında en son atılan tweetten başlayarak geriye doğru bir veri seti oluşturulmuştur. İlk veri setinde 52.000 civarında tweet bulunmaktadır. Daha sonra bu seneki asgari ücretin T.C. Aile, Çalışma ve Sosyal Hizmetler Bakanı Zehra Zümrüt Selçuk'un açıkladığı tarih 28/12/2020 ve bir hafta öncesi baz alınmıştır.

Bir hafta öncesinin baz alınmasının sebebi resmi açıklama yapılmadan önce yapılan yorumların doğrudan yanlış olarak sınıflandırılıp kurulan modelde kullanılması planlanmıştır.

Daha sonra bu filtreleme işlemi sonrası 42.285 tweet elde edilmiştir. Fakat elde edilen tweetlerin haber niteliği taşıması hususunda belirlenen ölçütlere göre tekrar filtrelendiğinde ise 1.419 tweet elde etmiş olundu.

Son olarak kelime çantası modeline uygun elde edilen tweetler işlenerek haberin doğru ve yanlışlığı belirlenmiştir.

b.Literatür Özeti:

Proje kapsamında yapılan araştırmalar sonucu birçok farklı şekilde yanlış haber tespiti yapılmıştır. Bunlardan bazıları :

- Eugenio Tacchini 1 , Gabriele Ballarin 2 , Marco L. Della Vedova 3 , Stefano Moret 4 , and Luca de Alfaro5. (25 Apr 2017) Some Like it Hoax : Automated Fake News Detection in Social Networks.

- Abdullah Hamid1*, Nasrullah Sheikh 2*, Naina Said1*,Kashif Ahmad3*, Asma Gul4, Laiq Hasan 1, Ala Al-Fuqaha Fake News Detection in Social Media using Graph NeuralNetworks and NLP Techniques: A COVID-19 Use-case

- Fake News Detection on Social Media : A Data Mining Perspective
  Kai Shu 1 ,Amy Sliva 2 ,Suhang Wang 3, Jiliang Tang 4 , and Huan Liu5

Probleme yönelik başlıca incelediğim kaynaklar bu şekildedir. Elde edilen bilgiler doğrultusunda Doğal Dil işleme kullanılması gerektiği anlaşılmıştır. Doğal Dil İşleme temel olarak 2 yaklaşım bulunmaktadır : Linguistik ve analitik yaklaşımlar.

Linguistik, dili bir sistem olarak gören ve niteliğini, yapısını, birimlerini ve dönüşümlerini inceleyen bir dalıdır. Linguistik yaklaşım ise dil bilimi ışığında dil bilgisi kuralları baz alınarak yapılan dil işleme yaklaşımıdır. Analitik yaklaşıma göre daha iyi sonuç verir fakat çalışma süresi ve geliştirilme aşaması çok daha derin bir yaklaşımdır.

Analitik yaklaşım ise dil bilgisi, kelime önceliği, cümle yüklem ilişkisi gözetmeksizin analitik olarak dilin işlenmesidir.Dil bilimi yaklaşımına göre daha kötü sonuç verir. Fakat algoritma geliştirmesi daha basit ve daha hızlı sonuç vermektedir.

Proje kapsamında gerçekleştirilen literatür taraması sonucunda Doğal Dil İşleme yaklaşımı olarak Analitik yaklaşım üzerinden Kelime Çantalama yöntemi; sınıflandırma yöntemi olarak ise Kural tabanlı sınıflandırma yöntemi kullanılmasına karar verilmiştir.

c.Çerçevenin Yapısı,Bileşenler,Mimari Tasarım:

Proje kapsamında elde edilen veri seti Python kütüphaneleri kullanılarak modele uygun hale getirilmiştir.Tweetler tekrardan bir filtreleme yapılıp sadece haberin yayınlandığı gün ve sonraki günü içeren tweetler üzerinden yapılmıştır.

Tweetler içeresinde geçen noktalama işaretleri, büyük-küçük harflerin yazımı,veri setinde geçen kelimelerin belirlenmesi,veri setinde geçen sayıların belirlenmesi vb. işlemler ile kurulacak olan modele hazır hale getirilmiştir.

Kullanılacak model için kural tabanlı bir sınıflandırma modeli oluşturulmuştur. Basitçe kullanılan kural şu şekildedir:

-Eğer model için hazırlanan kelime çantasındaki kelime ve sayılarla 8 den fazla bir eşleşme var ise bu haber doğrudur, 8 den az sayıda kelime ve sayı eşleştirmesi var ise haber yanlıştır. Eşik değer olarak belirlenen 8 ise kelime çantasında bulunan değer sayısı baz alınarak %40 eşleşmeden fazla olan tweetleri doğru kabul etmiştir.

Kelime çantası modeli, doğal dil işlemede kullanılan basitleştirici bir temsildir. Bu modelde bir metin, kelimelerinin çantası halinde temsil edilir. Çoksallık tutulurken gramer ve kelime sırası göz ardı edilir.

Belirlenen bu basit kuralın uygulanabilmesi için ortak bir kelime çantası çıkarılması gerekmektedir.Bunun için yine Python kütüphanlerinden yararlanılarak kelimeler köklerine ayrıştırılmış, daha sonra ise tüm veri seti üzerinde bu kökler sayılarak en çok kullanılan 10 kelime ve yine aynı şekilde en çok kullanılan 10 sayı ile kelime çantası oluşturulmuştur.

Modelin değerlendirilmesi ise veri seti içerisinde bulunan bütün tweetler manuel olarak okunarak seçilen haber konusu ‘Asgari Ücret’ hakkında bakanlığın ve resmi gazetede yayınlanan bilgiler doğrultusunda etiketlenmiştir.

Sonuç olarak model tarafından yapılan etiketler ile manuel olarak yapılan etiketler karşılaştırılmıştır.Proje kapsamında yapılan işlemler sınıflandırma yapmak için kullanıldığından değerlendirme ölçütleri de sınıflandırma başarım ölçütleridir.Bu ölçütler karmaşıklık matrisi, anma , kesinlik,doğruluk ve f-ölçütü değerleri ile hesaplanır.

d. Yazılımın Kullanılması:

Yazılım, Python dilinde geliştirilmiş, Python kodu derlenebilen herhangi bir entegre geliştirme ortamında kullanılabilir. Herhangi bir arayüze sahip değildir. Proje Anaconda paket yöneticisi altında gelen Jupyter Notebook ve Spyder programlarında geliştirilmiştir. Python 3.x üzeri kullanılması gerekmektedir Projenin çalıştırılabilmesi için gerekli Python kütüphaneleri şunlardır:

- Twint
- Nest_asyncio
- Pandas
- String
- SnowballStemmer
- Collection
- Sklearn
- Matplotlib

e. Çerçevenin algoritmasının açıklanması:

Proje kapsamında verilen probleme yönelik çerçeve şu şekildedir:

1. Veri setinin oluşturulması
2. Verilerin probleme yönelik niteliklerin seçilmesi.
3. Nitelikler kullanılarak veri üzerinde filtreleme yapılması.
4. Filtrelenen veri üzerinde temizleme işlemleri yapılması.
5. Temizlenen veri setinde bulunan kelimelerin ve sayıların elde edilmesi.
6. Elde edilen kelimelerin köklerine ayrılması ve en çok geçen ilk 10 kelimenin seçilmesi.
7. Elde edilen sayılar üzerinden en çok geçen ilk 10 kelimenin seçilmesi .
8. Seçilen kelime ve sayıların birleştirilip kelime çantası oluşturulması.
9. Oluşturulan kelime çantası kural üzerinde veri setine uygulanarak sınıflandırılması.
10. Veri seti üzerinde manuel olarak sınıflandırma sonucu ile model sonucunun sınıflandırma başarım değerleri ile karşılaştırılması.
11. Sınıflandırma sonucunun görselleştirilmesi.

f. Çalışma Örnekleri:

Verinin Çekilmesi:

In [1]:
```python
import twint
import nest_asyncio
nest_asyncio.apply()
c = twint.Config()
```

In [2]:
```python

c.Lang = "tr"
c.Store_csv = True
c.Output = "./Deneme.csv"
x = 1
c.Filter_retweets = True
c.Min_likes = 10 * x
c.Min_retweets = 3 * x
c.Min_replies = x
c.Search = "#asgariucret"
c.Since ='2020-12-22 00:00:00'
c.Until = '2020-12-30 00:00:00'
```

In [3]:
```python
twint.run.Search(c)
```

```
1344016889584955393 2020-12-29 23:26:11 +0300 <workingnamer> Adaleti Twitter'dan arayan bir millet olduk malesef #SağlıkcıYok
Sayılamaz  #SağlıkcıyaAsgaridenAzMaas #SağlıkcıyaSeyyanenZam #SağlıkcılarTakipte #SağlıkcınınMaasıYatmadı #asgariucret  #ayli
nsozer  #seldatas #vesiledonmez #KadınKatliamıVar  #TekAlımda60Bin . . . . . .
1343999550487408649 2020-12-29 22:17:17 +0300 <NaciyeKaplan13> #asgariucret 👇👇  https://t.co/IVi494YYae
1343962582214959105 2020-12-29 19:50:23 +0300 <SahinalpBulent> Son 1 yıl içerisinde, temel gıda ürünleri olan; -yumurta %76,
-domates %68,  -kuru soğan %38  -mercimek %67,  -ayçiçek yağı % 43  -et % 37,  -un ve ekmek ort. %25 zamlanırken, #asgariucre
t %21,5 artış ile 2.825,90 TL oldu.
1343928518200205314 2020-12-29 17:35:02 +0300 <Rehber_Gazetesi> 30.12.2020 Manşetimiz... #AylinSözer #asgariucret #VETO #çina
şısı #IrkçıSedaAkgül #2020yeBirSözBırak @btmnbld @batmanvaliligi @ZehraZumrutS @suleymansoylu @hcemal571 @haksiad72  https://
t.co/0PBGPKDgRl
1343910287066468353 2020-12-29 16:22:35 +0300 <Skara_5734> Almanya da en düşük ücret yaklaşık 1500 €, saat ücreti 9,35€. Yani
bir Alman vatandaşı asgari ücretle 1500 şişe litrelik yağ alabilirken, Türkiye'de asgari ücret 2825 lira oldu. Biz burda o pa
rayla ancak 190 şişe litrelik yağ alabiliyoruz,   #asgariucret #yağ
1343903949993213952 2020-12-29 15:57:24 +0300 <MedyasMehmet>  https://t.co/VDiyO0nAZx #2021asgariücret #asgariucret #asgariuc
ret2021 #AsgariUEcret2825TL #btp #hüseyinbaşhaberleri #btphaberleri #mem #milliekonomimodeli #haydarbaş #ekonomimasalıbitti
https://t.co/cPe4oJYatz
1343896443267010560 2020-12-29 15:27:35 +0300 <turkiyediyince> Kaşıkla verdik, vidanjörle geri alıyoruz dönemi başlamıştır. İ
```

In [4]:
```python
import pandas as pd
```

In [5]:
```python
data = pd.read_csv("./Deneme.csv")
```

In [ ]:

In [ ]:

In [8]:
```python
pd.set_option("display.max_rows", 2800, "display.max_columns", 100)
```

In [9]:
```python
data.head()
```

Out[9]:

	id	conversation_id	created_at	date	time	timezone	user_id	username	name	place	
0	1344016889584955393	1344016889584955393	2020-12-29 23:26:11 Türkiye Standart Saati	2020-12-29	23:26:11	300	1330469795066679297	workingnamer	Leylim 🦋	NaN	Adaleti Twi arayan bir mill
1	1343999550487408649	1343999550487408649	2020-12-29 22:17:17 Türkiye Standart Saati	2020-12-29	22:17:17	300	1116376072864641026	naciyekaplan13	Naciye Kaplan	NaN	#asgariuc https://t.co/IVi4
2	1343962582214959105	1343962582214959105	2020-12-29 19:50:23 Türkiye Standart Saati	2020-12-29	19:50:23	300	1249691032893227008	sahinalpbulent	Bülent Şahinalp	NaN	Son 1 yıl içe temel gıda

```python
# -*- coding: utf-8 -*-
"""
Created on Wed Dec 30 15:38:07 2020

@author: kosea

"""

"""Kullanılan kütüphanelerin import edilmesi"""
import pandas as pd
from string import punctuation
from snowballstemmer import stemmer
from collections import Counter

"""Veri Setinin import edilmesi"""
#  Kaydedilen verinin okunması ve kullanılacak niteliklerin seçilmesi

data_filtreli = pd.read_csv("veri_tarih_begeni.csv")
pre_data = data_filtreli[["date", "username", "tweet", "replies_count", "retweets_count", "likes_count"]]

# seçilen özniteliklerin yedeğinin alınması

pre_data.to_csv("pre_data.csv")
```

```python
"""Verinin Filtrelenmesi"""
# Veri üzerinde tekrardan tarihe göre filtrelenmesi

mask = pre_data['date'] == "2020-12-28"
mask_2 = pre_data['date'] == "2020-12-29"
data_28tarihli = pre_data[mask]
data_29tarihli = pre_data[mask_2]
data_digertarihli = pre_data[~mask]

# Filtre uygulanan verilerin düzenlenmesi
data_digertarihli = data_digertarihli[~mask_2]
data_2829_tarihli = pd.concat([data_28tarihli, data_29tarihli])



# Metin olarak işlem yapılacak niteliğin veri seti üzerinden seçilmesi
tweetler_2829 = data_2829_tarihli[["tweet"]]
tweetler2829_degerler = tweetler_2829.iloc[:, 0].values

"""Verinin Temizlenmesi"""

# Veri içerisinde geçen noktalama işaretlerinin kaldırılması

cevirici = str.maketrans('', '', punctuation)

# Veri içerisinde geçen türkçe karakterlerin latin karakterlere çevrilip bir standarta oturtulması
Tr2Eng = str.maketrans("çğıöşü", "cgiosu")

kelime_listesi = []
sayi_listesi = []
tweet_listesi = []
```

```python
for i in range(1336):
    yorum = tweetler2829_degerler[i]
    yorum = yorum.lower() # cümledeki her karakterin küçük hale çevrilmesi
    yorum = yorum.translate(cevirici) # noktalama işaretlerinin kaldırılması
    yorum = yorum.translate(Tr2Eng)  # latin alfabesine çevrilmesi
    tweet_listesi.append(yorum) # tweetlerin şimdiki hali ile cümle olarak listelenmesi
    yorum = yorum.split() # tüm cümlelerdeki kelimelerin ayrılması
    yorum_length = len(yorum)
    """Kelime çantası için gerekli verinin hazırlanması"""
    for j in range(yorum_length):
        kelime_listesi.append(yorum[j]) #Tüm Kelimelerin listelenmesi

    yorum_rakam = [int(s) for s in yorum if s.isdigit()] # cümlelerde geçen rakamların tespiti
    yorum_rakam_length = len(yorum_rakam)
    for k in range(yorum_rakam_length):
        sayi_listesi.append(yorum_rakam[k]) # Tüm sayıların listelenmesi

# Snowballstemmer kütüphanesi ile kelimelerin köklerine ayrılması
kokbul = stemmer('turkish')

yorumlist_kok = kokbul.stemWords(kelime_listesi) # köklerine ayrılan kelime listesi


kelime_sayi = Counter(yorumlist_kok) # kök kelimelerin aynı olanların sayılması
# köklere ayırınca 9433 den 6722 ye kadar kelime sayısı indi.

rakam_sayi = Counter(sayi_listesi) # aynı sayıların kaç tane olduğunun sayılması
```

```python
"""Kelime Çantasının Hazırlanması"""

list_of_words_and_counter = []
list_of_numbers_and_counter = []


counter_kelime = 0

 # en çok geçen 20 kelimenin ilk 10 tanesinin alınması
for kelime in kelime_sayi.most_common(20):
    if counter_kelime < 10:
        list_of_words_and_counter.append(kelime[0])
    else:
        break

    counter_kelime +=1


counter_kelime = 0

#  en çok geçen 20 sayının ilk 10 tanesinin alınması
counter_rakam = 0

for rakam in rakam_sayi.most_common(20):

    if counter_rakam < 10:
        list_of_numbers_and_counter.append(rakam[0])
    counter_rakam += 1

"""Kelime Çantasının Oluşturulması"""

words_and_number_list = []
words_and_number_list.extend(list_of_numbers_and_counter)
words_and_number_list.extend(list_of_words_and_counter)
```

```
"""Kurala göre Kelime Çantası Kullanılarak Tweetlerin Sınıflandırılması"""

match_points_list = []

tweet_etiketi = []

counter_tweet = 0

match_counter = 0

for i in tweet_listesi:

    for j in words_and_number_list:

        if str(j) + ' ' in i:
            match_counter +=1

    if match_counter < 8:
        tweet_etiketi.append(0) # true list
    else:
        tweet_etiketi.append(1)
    match_points_list.append(match_counter)
    match_counter = 0
print()
```

```
# -*- coding: utf-8 -*-
"""
Created on Sun Jan  3 01:36:20 2021

@author: kosea
"""
"""VERİLERİN DEGERLENDİRİLMESİ"""
import pandas as pd

data = pd.read_csv('Son_veri_ve_Sonuc.csv')

y_test = data[['dogruluk']]
y_predict = data[['tahmin']]
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_predict,y_test)
print(cm)
from sklearn.metrics import f1_score
f_score = f1_score(y_predict,y_test)

print('f olcutu: ',f_score)

from sklearn.metrics import classification_report

class_report = classification_report(y_predict,y_test)

print(class_report)

import sklearn.metrics as mtc

fpr,tpr,threshold = mtc.roc_curve(y_predict, y_test)
roc_auc = mtc.auc(fpr, tpr)

import matplotlib.pyplot as plt
plt.title('ROC Egrisi')
plt.plot(fpr, tpr,'b',label = 'AUC = %0.2f' % roc_auc)
plt.plot([0,1],[0,1], 'r--')
plt.xlim([0,1])
plt.ylim([0,1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

g. Sonuçlar ve Yorumlanması:

Karmaşıklık matrisi :

Kullanılan model sonucunda elde edilen karmaşıklık matrisi bu şekildedir.

TP = 907 TN = 215 FP = 186 FN =28

TP : Yanlış olarak sınıflandırılan ve yanlış olan tweet miktarı

TN: Doğru olarak sınıflandırılan ve doğru olan tweet miktarı

FP : Yanlış olarak sınıflandırılan ama doğru olan tweet miktarı

FN : Doğru olarak sınıflandırılan ama yanlış olan tweet miktarı

Doğruluk = TP + TN / (TP+TN+FN+FP) = 1122 /1336 = 0.8398

Hata Oranı = FN + FP / (TP+TN+FN+FP) = 214 / 1336 = 0.1602

Kesinlik = TP / (TP + FP) = 907/935 = 0.970

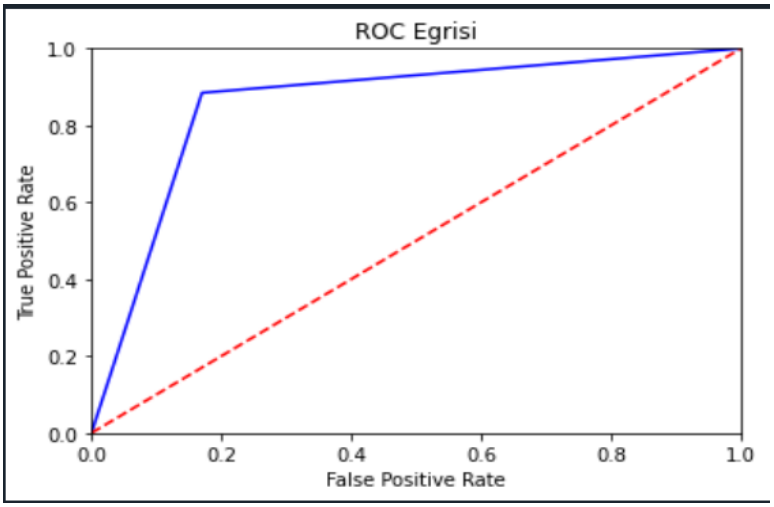
Anma = TP / (TP+FN) = 0.8298

F ölçütü = 2 * kesinlik * anma / (kesinlik +anma) = 2*0.970*0.8298 /1.7998 = 0.8944

F skoruna bakacak olursak % 89 doğru sınıflandırma yapan bir model elde edilmiştir.

```
accuracy:   83.982

In [10]: runcell(0, 'C:/Users/kosea/.spyder-py3/Veri Proje/Veri_kontrol.py')
[[907 186]
 [ 28 215]]
              precision    recall  f1-score   support

           0       0.97      0.83      0.89      1093
           1       0.54      0.88      0.67       243

    accuracy                           0.84      1336
   macro avg       0.75      0.86      0.78      1336
weighted avg       0.89      0.84      0.85      1336

accuracy:   83.982
```

Elde edilen sonuçlar doğrultusunda ROC eğrisine bakacak olursak çizilen grafiğin altta kalan alanı ne kadar büyük ise o kadar başarılı bir sınıflandırma yapıldığı söylenebilmektedir.



h. Gelecekte yapılabilecekler:

Gelecekte yapılabilecek geliştirmeler şu şekilde olabilir:

-Kurulan model üzerinden sonuçların iyileştirilmesi için kelime çantası oluşturulmadan önce etkisiz kelimeler çıkartılarak oluşturulması modelin daha iyi sonuç vermesine yardımcı olabilir.

-Verilerin filtrelenmesi aşamasında retweetler temizlenmiş olmasına rağmen retweet yapılmamış fakat farklı hesaptan aynı şekilde yazılmış, haber içeriği taşımayan tweetler veri setine dahil edilmezse daha iyi bir sonuç çıkabilir.

- Veri seti farklı modeller kullanılarak başarı değerlendirmeleri sonucu farklı yaklaşımlarla hibrit şekilde kullanılabilir.

-Proje kapsamında kullanılan kelime çantası yöntemi Doğal Dil İşleme alanında kullanılan analitik yaklaşımlardan bir tanesidir. Kullanılan bu analitik yaklaşım linguistik yaklaşımlarla birleştirilerek hibrit bir şekilde kullanılması modelin daha iyi bir sonuç elde edilmesini sağlayabilir.

- Türkçe, dil bilgisi olarak çok zengin bir dildir. Çeşitli kullanım şekilleri ve karmaşık cümle çeşitlerinin daha iyi bir şekilde anlaşılması için Türkçe kelimelerinin köklerini bulmada başarılı sonuç verecek bir kütüphane geliştirilip projeye eklenmesi ve açık kaynak olarak sunulması dilimiz üzerinde yapılacak projelerde çok daha iyi çalışmaların çıkmasına yardımcı olacaktır.

i. Kullanılan Kaynaklar:

- https://github.com/twintproject/twint
- https://medium.com/towards-artificial-intelligence/how-to-scrape-tweets-without-twitters-api-using-twint-797b196b951
- https://www.udemy.com/course/veri-bilimi-icin-python/
- http://www.veridefteri.com/2017/11/20/turkce-metin-islemede-ilk-adimlar/
- https://pandas.pydata.org/pandas-docs/version/0.15/tutorials.html
- https://python-istihza.yazbel.com/listeler_ve_demetler.html
- https://www.bilkav.com/makine-ogrenmesi-egitimi/
- https://tr.wikipedia.org/wiki/Kelime_%C3%A7antas%C4%B1_modeli#:~:text=Kelime%20%C3%A7antas%C4%B1%20modeli%20do%C4%9Fal%20dil,hatta%20kelime%20s%C4%B1ras%C4%B1%20g%C3%B6zard%C4%B1%20edilir.
- http://ftp-kampus.izu.edu.tr//ALLFILES/bim407verimadenciligich4farklisiniflandirmayontemleri_7ff0f9ebfb6593817158787788743 20b_.pdf slayt 58-70
- Muhammed Çağrı Aksu1* , Ersin Karaman2 (14 Ekim 2020) FastText ve Kelime Çantası Kelime Temsil Yöntemlerinin Turistik Mekanlar İçin Yapılan Türkçe İncelemeler Kullanılarak Karşılaştırılması. Avrupa Bilim ve Teknoloji Dergisi, (20), 311-320 https://dergipark.org.tr/en/download/article-file/1225461
- https://stackoverflow.com/questions/33355678/python-list-object-attribute-append-is-read-only
- https://emrahmete.wordpress.com/2020/06/09/dogal-dil-isleme-problemlerinin-cozum-yaklasimlari-ve-gelisimi/