NLP, Duygu Analizi Çalışmasında NLTK/Corpus/Stopwords Fonksiyonunca Karar Süreci Dışarısına Çıkarılan İngilizce "Not" Takısının, Yeniden Karar Sürecine Dahil Edilmesi Ve Etkilerinin İncelemesi

Sercan EĞİLMEZKOL
Başkent Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü
Yüksek Lisans - Öğr.No: 22210272
Ankara, TÜRKİYE

***Abstract*—In this study, the effects of removing the negative suffixes from the comments together with the Stopwords and not being included in the decision mechanism will be examined. Three solution proposals will be examined.**

Keywords—NLP, Sentiment Analysis, Stopwords, Removing Negative Suffixes

***Özet*—Bu çalışma ile olumsuzluk takılarının Stopwords'ler ile birlikte yorumlardan çıkarılarak karar mekanizmasına sokulmaması ile oluşan etkiler incelenecektir. Üç adet çözüm önerisi incelenecektir.**

Anahtar Sözcükler—NLP, Duygu Analizi, Stopwords, Olumsuzluk Takılarının Çıkarılması

I. AMAÇ

Bu çalışma ile EEM612 kodlu Örüntü Tanıma ve Makine Öğrenmesi dersi dönem ödevi sunum çalışması yapılacaktır.

II. KULLANILAN VERİ SETİ VE ÇALIŞMA DOSYALARININ PAYLAŞIMI

Aşağıdaki link içerisindeki müşteri yorumları veri seti kullanılmıştır.

- https://www.kaggle.com/datasets/vigneshwarsofficial/reviews

İndirilen veri seti dosyası (.tsv) uzantılı bir dosyadır. Windows PC'de isim değiştirme ile dosya uzantısı (.csv) olarak değiştirilmiştir.

Excel ile dosya açıldığında dosya formatı sorunludur. Farklı hücreler içerisinde verilmiş yorumlar vardır. "Liked?" (beğenme) bilgisi yorum ile bitişiktir. Ayrıca 230 nolu satırdaki 229 nolu yorum içerisine 9 adet yorum girili haldedir. Bu hataların tümü aşağıdaki dosyada düzeltilmiştir.

- https://github.com/sercanegilmezkol/EEM612_HW/blob/main/Veri_seti_belirleme/NLP/Restaurant_Reviews.csv

Çalışma dosyalarının tümünün paylaşımı için:

- https://github.com/sercanegilmezkol/EEM612_HW

III. DOĞAL DİL İŞLEME (NLP: NATURAL LANGUAGE PROCESSING)

Doğal Dil İşleme (NLP: Natural Language Processing) bir uygulama algoritmasının insan dilini konuşulduğu veya yazıldığı şekliyle anlayabilmesi yeteneğidir. Yapay Zekanın (YZ: Artificial Intelligence - AI) bir alt dalıdır.

NLP 50 yılı aşkın bir süredir varlığını sürdürmektedir ve kökleri dilbilim alanındadır. Tıbbi araştırma, arama motorları ve iş zekası gibi bir çok alanda uygulama alanına sahiptir.

NLP, insan dilinin kural tabanlı ve istatistiksel modellemesi olan hesaplamalı dilbilimi ile makine öğrenimi ve derin öğrenme modellerini birleştiren bir alandır.

A. NLP'nin Önemi ve Gerekliliği:

İşletmeler, büyük miktarlarda yapılandırılmamış, metin ağırlıklı veriler kullanır. Bu veriyi verimli bir şekilde işlemek için bir yönteme ihtiyaç vardır. Çevrimiçi olarak oluşturulan veya veritabanlarında saklanan verilerin büyük çoğunluğu NLP'dir ve yakın zamana kadar işletmeler bu verileri etkili bir şekilde analiz edemiyorlar idi. NLP'nin yararlı olduğu yer burasıdır.

Şu şekilde düşünelim: Milyonlarca kelimelik bir metin veritabanı elde etmek kolaydır. Bunları analiz etmek için programlar yazabiliriz. Ama nasıl?

- Bir metnin amacı ve içeriğini özetleyen anahtar sözcük ve tümceleri otomatik olarak nasıl çıkarabiliriz?
- NLP'nin önemli ve hassas zorlukları nelerdir?

NLP şu başlıklar ile ilgilenir:

- Cümle yapısını analiz etme,
- Cümlenin gramer yapısını oluşturma,
- Cümlelerin anlamını çözümleme,
- Dil verilerini yönetme.

B. Duygu Analizi (Sentiment Analysis):

Duygu analizi, bir metnin olumsuz, olumlu veya nötr gibi duygular içerip içermediğini belirlemek için kullanılır. NLP

ve makine öğrenimini (ML: Machine Learning) kullanan bir metin analitiği yöntemidir. Duygu analizi, "fikir/görüş madenciliği (opinion mining)" ve "duygusal yapay zeka (emotional AI) " olarak da bilinir.

NLP'nin duygu analizi haricinde diğer uygulama alanları şunlardır:

- Konuşma tanıma (Speech recognition)
- Konuşma etiketlemenin bir kısmı (Dilbilgisel etiketleme) (Grammatical tagging)
- Cümle içerisinde sözcüğün anlam ayrımının yapılması (Word sense disambiguation)
- Adlandırılmış varlık tanıma (Named entity recognition)
- Eş-referans çözümü (Co-reference resolution) (Örn: "O (she)" Ayşe'yi mi, çalıştığımız firmayı mı, yoksa evimizin kedisini mi ifade ediyor?)
- Doğal dil üretimi (Konuşma tanımanın tersi) (Konuşmadan metne) (Speech-to-text)

IV. İNCELEME ÇALIŞMALARI VE DEĞERLENDİRMELER

NLTK Corpus Stopwords fonksiyonu kullanılarak kullanıcı yorumları içerisindeki takılar çıkartılırken "not", "hasn't", "didn't" gibi kelimeler de karar mekanizmasından çıkartılmaktadır. Bunun sebebi duygu analizi amaçlı kullanılan kütüphanelerde bu kelimelerin bulunmasıdır. Ticari amaçlı firmalarca hazırlanan uygulamalarda da "stopwords çıkartma" uygulanan bir yöntemdir. [6]

Scikit-Learn Count Vectorizer fonksiyonu kullanılarak, yorumlar içerisinde kullanılan kelimeler, en çok kullanılan kelimeler önceliklendirilerek işaretlenmektedir (tokenize).

Olumsuzluk takıları yorumlardan çıkartıldığı için Count Vectorizer fonksiyonuna girdi olarak kelimenin kökü olumlu olarak sokulacaktır. Halbuki kullanıcı yorumu olumsuzdur. Sonuç olarak olumsuzluk takısı ve algoritma sonucu eğitim ve test veri setlerinde anlam kaymaları olacak ve başarısız bir öğrenme gerçekleşecektir.

NLTK Porter Stemmer fonksiyonu kullanılarak kelime köklerine indirgenmiş veri setimizde "Crust is not good." yorumu "crust" ve "good" kelimeleri işaretlenebilir kelimeler listesine alınmasına rağmen; beğenme bilgisi "hayır: 0" olarak eğitim veya test işlemine tabi tutulacaktır.

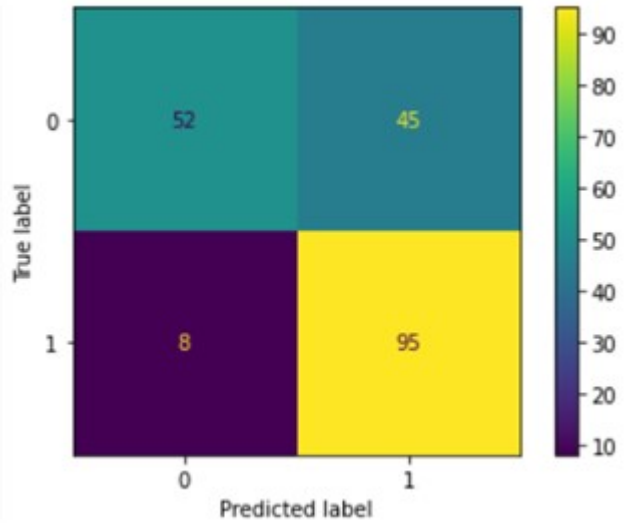
Index	Review	Liked
0	Wow... Loved this place.	1
1	Crust is not good.	0

1	str	10	crust good

Bu örnekte geçen "good" kelimesi pek çok yorumda kullanılan bir kelimedir. Bazı yorumlarda olumsuzluk ekiyle birlikte kullanılması sebebiyle bu yorumlar modelde "Good var ve Olumsuz" olarak işlenecektir. Olumlu yorumlarda ise "Good var ve Olumlu" şeklinde işlenecektir. Bu karışıklık sebepli algoritmanın doğru karar veremeyeceği ve bu karışıklığın bazı ilave algoritmalar eklenerek giderilebileceği öngörülmüştür.

Bu çalışma kapsamında üç ek algoritmanın etkileri incelenecektir.

Öncelikle bu ekler yapılmadan alınan tahmin sonucunu paylaşalım:



- 200 adet test verisinin 103 adeti olumluyken bunların 95 adedi doğru tahmin edilebilmiştir. 97 olumsuz test verisinin ise 52 adedi doğru tahmin edilebilmiştir.

Algoritma içerisinde aşağıdaki satırlar bu dokümanda gösterildiği gibi olmalıdır.

```
90  for i in range(1000):

94      '''if olumsuzluk_Tespit(gecici_yrm):
95          if yorumlar.iloc[i, 1] == 0:
96              yorumlar.iloc[i, 1] = 1    '''
97          #else:
98              #yorumlar.iloc[i,1] = 0
99      #yorum = olumsuzluk_Tespit(gecici_yrm)
100     '''if yorumlar.iloc[i, 1] == 0:
101         yorum = olumsuzluk_Tespit2(gecici_yrm)    '''

111 gosterge=600
```

A. İncelenecek ilk algoritma:

Olumsuzluk tespiti sonrası eğitim ve test verilerinde "beğenme" bilgisi olumsuz olan verilerde "beğenme" bilgisinin tersini alma mantığına dayanır.

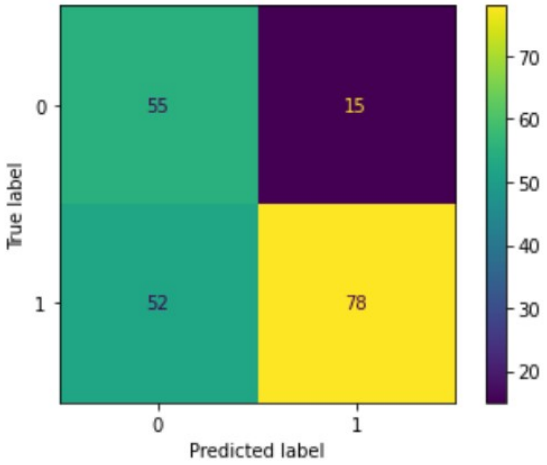
```python
def olumsuzluk_Tespit(yorum):

    _Not = 0
    ReturnSend = 0

    for i in range(len(yorum)):
        if _Not == 0 and yorum[i] == "n":
            _Not = 1
        elif _Not == 1 and (yorum[i] == "o" or yorum[i] == "'"):
            _Not = 2
        elif _Not == 2 and yorum[i] == "t":
            _Not = 3
        elif _Not == 3 and yorum[i] == " ":
            _Not = 4
            ReturnSend = 1
        else:
            _Not = 0
    if ReturnSend == 0:
        return 0
    else:
        return 1
```

```python
94      if olumsuzluk_Tespit(gecici_yrm):
95          if yorumlar.iloc[i, 1] == 0:
96              yorumlar.iloc[i, 1] = 1
```

Bu algoritmanın bir sonraki ayağı olabilecek "ilgili değişikliğin yapıldığı verilerin işaretlenmesi ve test koşturulup tahmin yapıldığında terslenen test verilerinin tahminde de terslenmesi" ilk ve temel incelememiz için gerekli olmadığından görece-zor işbu algoritma eklenmeden sadece burada verilen algoritmanın etkileri incelenecektir. Aşağıdaki sonuçlara ve algoritmanın detaylı incelemelerine bakıldığında bu algoritmanın ne kadar olumlu bir etkisinin olacağı net olarak belirlenememiştir.

97 olumsuz test verisinden 27 adedi algoritmaca olumluya çevrilerek eğitim ve tahmin yapılmıştır. Sonuç:

- 200 adet test verisinin 130 adeti olumlu olarak işleme sokulup bunların 78 adedi doğru tahmin edilebilmiştir. 70 adedi ise olumsuz olarak işleme sokulup 55 adedi doğru tahmin edilebilmiştir.
- Olumsuz olarak işleme (tahmin algoritmasına) sokulan verilerde oransal olarak % 53,6'lık başarı oranı %78,6'ya; sayısal olarak da 52 başarılı tahminden 55 tahmine yükselmiştir.



- Bunun yanında olumlu olarak tahmin algoritmasına sokulan verilerde oransal olarak % 92,2'lik başarı oranı %60,0'a; sayısal olarak da 95 başarılı tahminden 78 tahmine düşmüştür.
- Tahmin algoritmasının genel başarısı ise 200 veride 147 (% 73,5) doğru tahminden 133 (%66,5) tahmine düşmüştür.

Aşağıdaki değerlendirmelerin yapılabileceği düşünülmektedir:

- Aslında amacımız olumsuz yorumlardaki başarı oranını yükseltmek idi. Bu konuda %53,6'dan %78,6'ya gibi ciddi bir iyileşme gerçekleşmiştir.
- Olumsuzluk tespit edilen tüm olumsuz işaretli verilerin olumlu olarak işleme sokulması ise algoritmanın genel başarısında büyük bir olumsuz etki yaratmıştır.

Bu algoritmanın detaylı incelenmesi:

- Test verisinden 30 adedi bu kapsamdadır. "Liked" bilgisini terslemek yerine "2" olarak işaretlersek veri seti aşağıdaki şekilde olacaktır. "0" ve "1"ler orjinal verilerdir. "2" olarak değiştirmeyi, kontrol amaçlı olarak, biz yaptırtıyoruz.

Review	Liked
Wow... Loved this place.	1
Crust is not good.	2
not tasty and the texture was just nasty.	2
Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.	1
The selection on the menu was great and so were the prices.	1
Now I am getting angry and I want my damn pho.	0
Honeslty it didn't taste THAT fresh.	2
The potatoes were like rubber and you could tell they had been made up ahead of time being kept under a warmer.	0

- Tüm seri seti içerisinde 1000 adet veriden 156 adedi bu şekilde “2” olarak işaretlenmektedir. Yani eğitim ve test veri setleri içerisinde bu kategoride olan toplam veri sayısı 156’dır.

- Bu 156 verinin ilk 29’u aşağıdaki şekilde incelenmiştir:

- Buna göre,

- Şekilde sadece sarı ile gösterilenler, olumlu bir kelimenin olumsuzluk takısı sebebiyle hatalı olarak karar mekanizmasına sokulacağının düşünüldüğü veri setlerini göstermek için kullanılmıştır.

- Şekilde sadece turuncu ile gösterilenler, olumsuzluk takısı olmasına rağmen; ilave olarak olumsuz bir kelime de içerdiği için karar mekanizmasına hatasız (doğru) sokulacak veri setlerini göstermek için kullanılmıştır.

NOT

Yukarıdaki iki madde için karar mekanizması, “not” gibi olumsuzluğun “stopwords” ile atıldığı algoritmadır. Ekstra bir önlem yoktur.

- Şekilde sadece mavi ile gösterilenler, yorum yapmanın zor olduğu veri setleridir.

- Şekilde hem sarı, hem de turuncu gösterimler mavi olarak da düşünülebilir. Bu örnekler için sarı olumsuzluk eki ve olumlu kelimeyi (veya kelimeleri), turuncu olumsuz kelimeyi (veya kelimeleri) gösterir.

Sadece Sarı	Sadece Turuncu	Sadece Mavi	Sarı ve Turuncu İçeren
14 Adet	3 Adet	8 Adet	4 Adet
		12 det	

- (8 + 4)’ten 12 veri yorum yapmanın zor olacağı veridir.

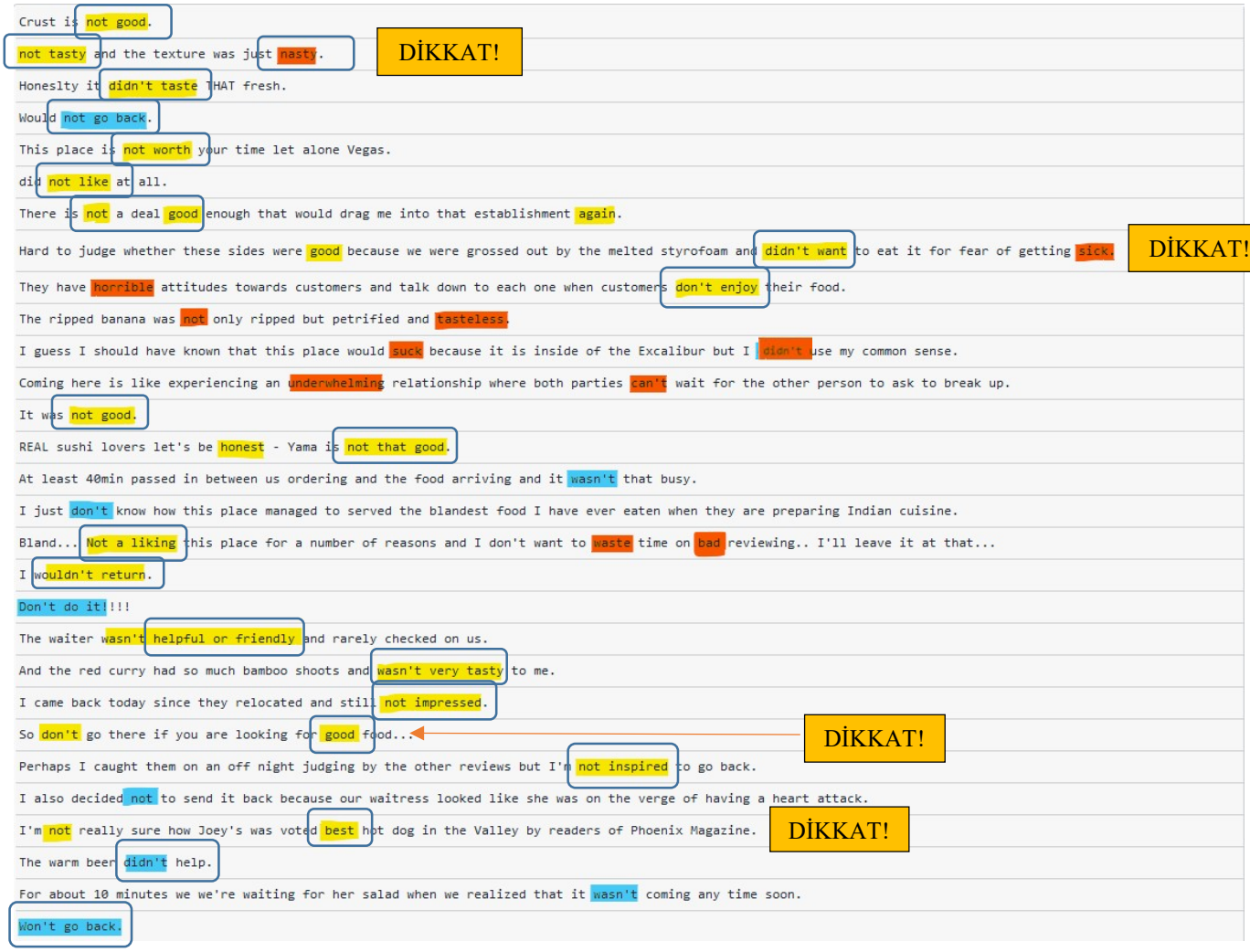
- 3 veri için: Beğenme bilgisini olumlu olarak işlem yapmamız karar mekanizmasına zarar verici niteliktedir.

- 29 veri, verilerin toplanma sırasına göre baştan seçilen ilk 29 veridir. Bunun ötesinde herhangi bir rastgelelilik yoktur. Tüm veri seti için, 156 veride bu dağılım farklı olabilir ve “Sadece Sarı”ların oranı 14/29 yani %50’ler mertebesinde olmayabilir.

- “Sadece Sarı” ve “Sadece Mavi” ((14+8)/29)’dan yaklaşık %75’lik veriyi oluşturur. “Sadece Mavi”lerin muhtemelen karar mekanizmasına olumsuz bir etkisi olmayacaktır. Ayrıca tüm veri

setinde, yani 156 adet veride de %75'lik bu oran üç aşağı, beş yukarı benzer bir ağırlıkta olacaktır çok muhtemelen.

- Yukarıdaki tüm verilere bakıldığında sadece son maddede ciddi anlamda etkili bir veri vardır (%75). Bu bilgiler ışığında bu algoritma için çok da net bir fikre varılamamaktadır.
- Bazı ek incelemeler yapılırsa aşağıdaki bazı detayların öne çıktığı düşünülmektedir. Bunlar:



- Şekilde sarı ile gösterilen olumsuzluk takısı ile anlam olarak olumlu / yapıcı kelimeler genel olarak birbirine konum olarak çok yakındır. "Help", "go back" gibi kelimelerin olduğu mavi kategorisinde incelenen bazı yorumlar da bu kapsamdadır.
- Bazı noktalarda dikkatli olunarak bir algoritma geliştirilebileceği değerlendirilmektedir. Örneğimiz için bu noktalar, aşağıdaki resimde "DİKKAT!" ifadesi ile netleştirilmiştir.

  Örneğin, en basitinden: Bir olumlu ifadeler kütüphanesi ("good", "tasty" ve "enjoy" gibi) oluşturup olumsuzluk eki ile aralarında örneğin 4 kelimelik bir mesafe varsa aktif olacak şekilde "yorumu olumluya çekme" eklentisi düşünülebilir. (Not: Bu mesafenin 4 gibi bir sayıyla sınırlandırılmasının olumsuz etkileri olacağı kesindir. Bkz. resimde "DİKKAT!" işaretli üçüncü ve dördüncü örnekler. Herhangi bir mesafe sınırı koymadan böyle bir algoritma eklersek yanlış karar vermemizi sağlayacak örnekler de elbette olacaktır. 29 adetlik bu veri setinde böyle bir örneğe denk gelmedik. Daha detaylı bir inceleme yapılabilir.)

  Ya da örneğin: İlk yorumdaki gibi hem olumlu hem olumsuz kelimelerin "not/n't" olumsuzluk ifadesiyle birlikte aynı yorum içinde kullanılması durumunun olumsuz etkilerini ortadan kaldırmak için Olumsuz_Durumda_Havuzdan_Çıkartılabilecek_Olumsuz_Kelimeler (ODHÇOK) gibi bir kütüphane oluşturup; "yorumu olumluya çekme" eklentisini uygularken, ayrıca "ODHÇOK kütüphanesinde olan kelimeleri yorumdan silme" eklentisi de yapılabilir. (Bkz. resimde "DİKKAT!" işaretli ilk iki örnek.)

- Sonuç olarak: Algoritmanın ne kadar olumlu bir etkisinin olacağı net olarak belirlenememiştir. Daha detaylı inceleme çalışmaları gereklidir. Bu doküman kapsamında yapılmayacaktır.

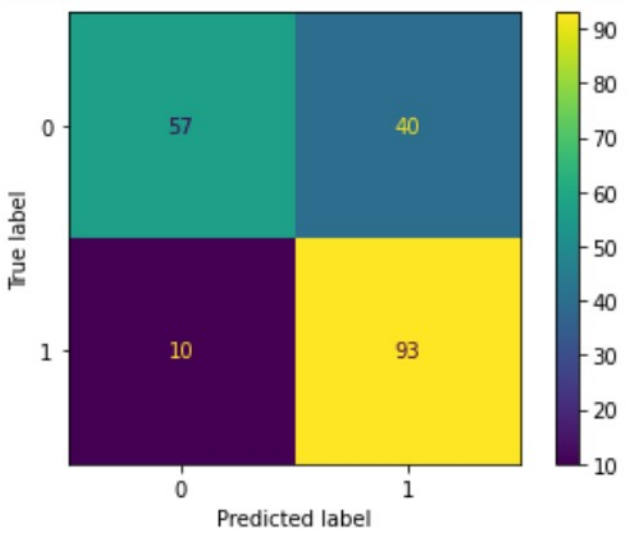
B. ***İncelenecek ikinci algoritma:***

Olumsuzluk ekini sonrasındaki kelime ile alt tire: "_" karakteri gibi bir karakterle birleştirme ve "not_good" gibi ikinci bir işaretleme kelimesi türetme temellidir.

```
def olumsuzluk_Tespit(yorum):

    _Not = 0
    ReturnSend = 0
    yrm = ""

    for i in range(len(yorum)):
        if _Not == 0 and yorum[i] == "n":
            _Not = 1
            yrm += yorum[i]
        elif _Not == 1 and (yorum[i] == "o" or yorum[i] == "'"):
            _Not = 2
            yrm += yorum[i]
        elif _Not == 2 and yorum[i] == "t":
            _Not = 3
            yrm += yorum[i]
        elif _Not == 3 and yorum[i] == " ":
            _Not = 4
            ReturnSend = 1
            yrm += "_"
        else:
            _Not = 0
            yrm += yorum[i]

    return yrm
```

```
99      yorum = olumsuzluk_Tespit(gecici_yrm)

111     gosterge=800
```

Bu durumda "good" kelimesi kümesi; "good" ve "not_good" olarak iki kelime kümesine bölünerek işlemlere tabi olacaktır. Scikit-Learn Count Vectorizer fonksiyonu n adet kelimenin işaretlemesi mantığına göre çalıştığı için artan küme sayısını ve örnek sıklığını telafi etmek için işaretlenecek kelime sayısı (gösterge değişkeni) arttırılmıştır.



200 adet test verisinden:

- **2 adet** olumlu test verisi **hatalı olarak olumsuz** tahmin edilmişken,
- **5 adet** olumsuz test verisi **daha başarılı olarak olumsuz** tahmin edilmiştir.

Başka bir ifadeyle,

- Olumsuz yorumlara ait 97 veri için başarı oranı (52 adet için) %53,6'dan (57 adet için) %58,8'e çıkmıştır.
- Olumlu yorumlara ait 103 veri için başarı oranı (95 adet için) %92,2'den (93 adet için) %90,3'e gerilemiştir.
- Tahmin algoritmasının genel başarısı ise 200 veride 147 (% 73,5) doğru tahminden 150 (%75,0) tahmine çıkmıştır.

Aşağıdaki değerlendirmeler yapılabilir:

- Asıl amacımız olan olumsuz yorumlardaki başarı oranını yükseltme gayesine %5'lik katkı sağlayan bir çalışma olmuştur. Modele girecek kelime ve örnek sayısını arttırır isek: (1) Yapılan bu eklentinin başarısının biraz daha hissedilir seviyelere çıkacağı düşünülmektedir. (2) Ayrıca olumlu test sonuçları için de iyileşme olacağı öngörülmektedir.
- Algoritma 8 GB RAM'li bir Intel i5 işlemcili PC'de koşturulmuştur. Biraz daha güçlü bir işlemci ve elbette RAM kapasitesi ile etiketleme kümesi daha büyük verilere çıkartılabilir.

Not: Yukarıdaki amaçlarla 65 Mbyte boyutunda 50K'lık bir veri seti (ilk 1K olumlu ve ilk 1K olumsuz yorum alınarak) 2K'lık bir veri setine dönüştürülmüştür.

https://github.com/sercanegilmezkol/EEM612_HW/blob/main/Veri_seti_belirleme/NLP/Restaurant_Reviews2.csv

Asıl kaynak:

https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

Bu veri setinde izlenilen filmlerin konusu (özeti) verilmektedir. Duygu analizinde kullanmak için kullanışlı bir veri seti değildir. Buna karşın 2K'lık ve çok daha fazla kelime kullanılarak verilmiş yorumlar için; yani daha fazla kelime ve veri girişi için işlem gücü ihtiyacını belirlemede veri elde edebileceğimiz bir çalışma olmuştur. Asıl veri setimizde aynı algoritma 2-3 sn.de koşturulabiliyorken, bu veri seti için 3-4 dk. bekleme sürelerine çıkılmıştır.

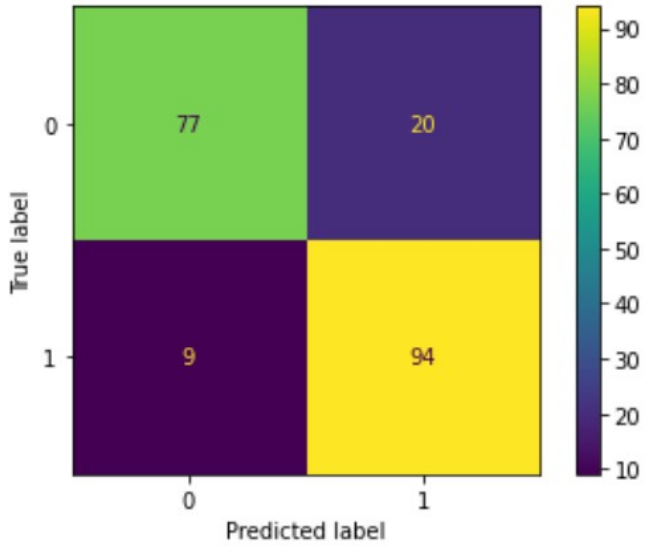
C. *İncelenecek üçüncü algoritma:*

Olumsuz yorumlarda olumsuzluk eki varsa yorum içerisine "EEM612_SE" gibi bir yapay kelime yorum içerisine katılacaktır. Olumsuzluk ekinin yorum içerisinden çıkartılması (stopwords) algoritmasına dokunulmayacaktır.

```
63  def olumsuzluk_Tespit2(yorum):
64      # Olumsuzluk_Dataset=["not","hasn't","didn't", ...]
65      _Not = 0
66      yrm = ""
67
68      for i in range(len(yorum)):
69          if _Not == 0 and yorum[i] == "n":
70              _Not = 1
71              yrm += yorum[i]
72          elif _Not == 1 and (yorum[i] == "o" or yorum[i] == "'"):
73              _Not = 2
74              yrm += yorum[i]
75          elif _Not == 2 and yorum[i] == "t":
76              _Not = 3
77              yrm += yorum[i]
78          elif _Not == 3 and yorum[i] == " ":
79              _Not = 4
80              yrm += " EEM612_SE "
81          else:
82              _Not = 0
83              yrm += yorum[i]
84
85      return yrm
```

```
100     if yorumlar.iloc[i, 1] == 0:
101         yorum = olumsuzluk_Tespit2(gecici_yrm)
```

```
111 gosterge=600
```



200 adet test verisinden:

- **1 adet** olumlu test verisi **hatalı olarak olumsuz** tahmin edilmişken,
- **(77-52)'den 25 adet** olumsuz test verisi **daha başarılı olarak olumsuz** tahmin edilmiştir.

Başka bir ifadeyle,

- Olumsuz yorumlara ait 97 veri için başarı oranı (52 adet için) %53,6'dan (77 adet için) %79,4'e çıkmıştır.
- Olumlu yorumlara ait 103 veri için başarı oranı (95 adet için) %92,2'den (94 adet için) %91,3'e gerilemiştir.
- Tahmin algoritmasının genel başarısı ise 200 veride 147 (% 73,5) doğru tahminden 171 (%85,5) tahmine çıkmıştır.

Aşağıdaki değerlendirmeler yapılabilir:

- Asıl amacımız olan olumsuz yorumlardaki başarı oranını yükseltme gayesine %25,8'lik katkı sağlayan bir çalışma olmuştur. (79,4 – 53,6 = 25,8)
- Genel algoritma başarısını da önemli ölçüde yükseltmiştir. (85,5 – 73,5 = 12)'den %12'lik katkı sağlamıştır.

V. SONUÇLAR VE İLERİ ÇALIŞMALAR

Yapılan ilk iki iyileştirme çalışmaları genel olarak büyük bir iyileşme sağlamamıştır. Daha çok örnek ve daha fazla kelime içeren veri setleri üzerinde inceleme çalışması yapılamamıştır.

Son iyileştirme çalışması ise kayda değer ve önemli bir iyileştirme ile sonuçlanmıştır.

Bu çalışmanın devamı olarak:

- Olumlu ifadeler kütüphanesi ("good", "tasty" ve "enjoy" gibi) oluşturup olumsuzluk eki geçen olumsuz işaretli yorumlar için "yorumu olumluya çekme" eklentisi düşünülebilir.
- Daha güçlü sistemler kullanılarak daha çok örnek ve daha fazla kelime içeren veri setleri verilerek ikinci iyileştirme algoritması gibi ilave bir algoritmanın modele eklenmesinin etkileri incelenebilir.

REFERANSLAR

[1] https://www.nltk.org/book/
[2] https://www.ibm.com/cloud/learn/natural-language-processing
[3] https://www.nltk.org/search.html?q=stopwords
[4] https://www.nltk.org/_modules/nltk/stem/porter.html
[5] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
[6] https://getthematic.com/sentiment-analysis/