**KOCAELİ ÜNİVERSİTESİ**
**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**Yapay Zeka**

# Makine Öğrenmesi ile Twitlerden Cinsiyet Tespiti

**160201077 TUĞBA AYDEMİR**
**190201105 ALPEREN İLERİ**
**170201093 İLAYDA DİŞİAÇIK**
**170202127 BURAK DURSUN**
**170201135 HARUN BÜYÜKBAŞ**

## 1.Veri Okuma ve Veriyi Anlama

Twitterdan alınan verilerini latin harfle olduğunu gösterip, okuduk.

```
import pandas as pd

data=pd.read_csv('/Users/tugbaaydemir/Documents/BILGISAYAR/VI.bYAPAYZEKA/NLP/
twitter.csv',encoding="latin1")
data.head()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20050 entries, 0 to 20049
Data columns (total 26 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   _unit_id               20050 non-null  int64  
 1   _golden                20050 non-null  bool   
 2   _unit_state            20050 non-null  object 
 3   _trusted_judgments     20050 non-null  int64  
 4   _last_judgment_at      20000 non-null  object 
 5   gender                 19953 non-null  object 
 6   gender:confidence      20024 non-null  float64
 7   profile_yn             20050 non-null  object 
 8   profile_yn:confidence  20050 non-null  float64
 9   created                20050 non-null  object 
 10  description            16306 non-null  object 
 11  fav_number             20050 non-null  int64  
 12  gender_gold            50 non-null     object 
 13  link_color             20050 non-null  object 
 14  name                   20050 non-null  object 
 15  profile_yn_gold        50 non-null     object 
 16  profileimage           20050 non-null  object 
 17  retweet_count          20050 non-null  int64  
 18  sidebar_color          20050 non-null  object 
 19  text                   20050 non-null  object 
 20  tweet_coord            159 non-null    object 
 21  tweet_count            20050 non-null  int64  
 22  tweet_created          20050 non-null  object 
 23  tweet_id               20050 non-null  float64
 24  tweet_location         12566 non-null  object 
 25  user_timezone          12252 non-null  object 
dtypes: bool(1), float64(3), int64(5), object(17)
memory usage: 3.8+ MB
```

```
data.shape
(20050, 26)
```

## 2.Veri Temizleme

Amacımız atılan tweetleri analiz ederek cinsiyet tespiti yapmak olduğu için, ‘gender‘ ve ‘description‘ olan kısmı aldık diğer öznitelikleri kullanmayacağımız için sildik.

```
data=pd.concat([data.gender,data.description],axis=1)
data.head(100)
```

| | gender | description |
|---|---|---|
| 0 | male | i sing my own rhythm. |
| 1 | male | I'm the author of novels filled with family dr... |
| 2 | male | louis whining and squealing and all |
| 3 | male | Mobile guy. 49ers, Shazam, Google, Kleiner Pe... |
| 4 | female | Ricky Wilson The Best FRONTMAN/Kaiser Chiefs T... |
| ... | ... | ... |
| 95 | brand | NaN |
| 96 | male | Calm And Cool |
| 97 | female | A retro modernist suffering from unsightly vis... |
| 98 | brand | Multiple Sclerosis lives here in Brant County ... |
| 99 | male | Doctorate in Physics, 2 time Olympic swimming ... |

Veri kümemizde eksik veri  olup olmadığını arıyoruz.

```
data.isnull().sum()
```

```
gender            97
description     3744
dtype: int64
```

Alınan çıktıdan görüldüğü üzere veri kümemizde, 97 adet cinsiyet özniteliği değeri, 3744 adet tweet metni eksik olarak girilmiştir. Bunlara daha yakından bakalım.

```
data['gender'].value_counts()
```

```
female     6700
male       6194
brand      5942
unknown    1117
Name: gender, dtype: int64
```

```
data[data['gender']=='unknown'].head(10)
```

| | gender | description |
|---|---|---|
| **19** | unknown | NaN |
| **92** | unknown | Boring, boring Chelsea. |
| **115** | unknown | GB for 15 years,not usually a twitter twat, fo... |
| **116** | unknown | NaN |
| **145** | unknown | NaN |
| **164** | unknown | NaN |
| **197** | unknown | \|unknown\| |
| **201** | unknown | Current mood: Jet Black Heart.\n\nI only go in... |
| **223** | unknown | SE Asian, Goat Kid, 13 March. ¾öÔ¾÷ø¾ª«_©ÛâS... |
| **244** | unknown | NaN |

```
data[data['description'].isna()].head(10)
```

| | gender | description |
|---|---|---|
| **15** | female | NaN |
| **18** | male | NaN |
| **19** | unknown | NaN |
| **49** | brand | NaN |
| **54** | male | NaN |
| **90** | female | NaN |
| **95** | brand | NaN |
| **110** | female | NaN |
| **116** | unknown | NaN |
| **122** | male | NaN |

Yukarıda bahsedilen eksik verileri sildik. Bu nedenle, 20000 veriden yaklaşık 16000 veri düştü

```
data.dropna(axis=0 ,inplace=True)
```

Verimizin son noktada gelinen yapısı

```
data.shape
```

```
(16224, 2)
```

Veriyi analiz etmemiz için verideki kategorik değikenleri de sayısallaştırmamız gerekiyor.

```
data.gender=[1 if each=="female" else 0 for each in data.gender]
```

```
data.head(3)
```

| | gender | description |
|---|---|---|
| **0** | 0 | i sing my own rhythm. |
| **1** | 0 | I'm the author of novels filled with family dr... |
| **2** | 0 | louis whining and squealing and all |

## 3.Regular Expression

Odaklanmamız gereken veri kümemizi aldıktan sonra, bu veride makine öğrenmesi uygulayabilmek için, veri üzerinde bir takım doğal dil uygulamaları işlemleri gerçekleştirmemiz gerekiyor.

Burada bu işlemleri tüm veriye uygulamadan önce, örnek bir tweet alarak işlemlerin nasıl uygulandığını bu tweet üzerinde görerek kavramaya çalışalım.

4. tweete bu uygulama için seçelim.

```
import re
first_description=data.description[4]
first_description
```

```
'Ricky Wilson The Best FRONTMAN/Kaiser Chiefs The Best BAND Xxxx Thank yo
u Kaiser Chiefs for an incredible year of gigs and memories to cherish al
ways :) Xxxxxxx'
```

Bu aşamada regular expression kavramı ile ilgili aşağıdaki uygulamalar yardımıyla kısa bir bilgi verelim.

[^ character_group ] character_group içinde olmayan herhangi bir karakterle eşleşir. Varsayılan olarak, character_group içindeki karakterler büyük / küçük harf duyarlıdır.

Örnek: [^ae] Tugba “T”, “u”, “g”, “b”

```
re.sub("[^ae]"," ",'Tugba Aydemir')
'    a    e   '

re.sub("[^aeu]"," ",'Tugba,..Aydemir')
' u  a      e   '
```

```
re.sub("[^aeu]","?",'Tugba,..Aydemir')
'?u??a??????e???'
```

Bu örneklerde de görüldüğü üzere, re.sub komutu içinde ilk bölümde değerlendirmeye almak istediğiniz karakterleri yazıp, ikinci bölümde de bunun dışında kalan karakterleri metin içinde hangi karakterle doldurmak istediğinizi yazarak, son olarak bunu komutu işletmek istediğiniz değişkeni yazarak komutu ifade edersiniz.

Bu bilgilerden yola çıkarak örnek aldığımız 4. Tweeti istemediğimiz karakterlerden arındıralım.

```
description=re.sub("[^a-zA-Z]"," ",first_description)
```

Verideki tüm harfleri küçük harf yaptık.

```
description=description.lower()
```

```
'ricky wilson the best frontman kaiser chiefs the best band xxxx thank yo
u kaiser chiefs for an incredible year of gigs and memories to cherish al
ways    xxxxxxx'
```

## 4.Natural Language Tool Kit

Odaklanmamız gereken veri kümemizi aldıktan sonra, bu veride makine öğrenmesi uygulayabilmek için, veri üzerinde bir takım doğal dil uygulamaları işlemleri gerçekleştirmemiz gerekiyor.

Natural language tool kit kütüphanesini import ediyoruz. Daha sonra download ile corpus diye bir klasöre indiriliyoruz. Son olarak corpus klasöründen import ederiz.

```
import nltk #natural language tool kit
nltk.download("stopwords")#corpus diye bir klasöre indiriliyor
from nltk.corpus import stopwords #sonra  corpus klasöründen import ediliyor
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/tugbaaydemir/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Dosyanın indiği yere bakalım.

```
stopwords
<WordListCorpusReader in '/Users/tugbaaydemir/nltk_data/corpora/stopwords'>
```

Tweeti, kelime-kelime ayıralım.

```
description = description.split()
```

Tweetin bu ayrılmış halindeki ilk 10 kelimeyi görelim.

```
description[:10]
```

```
['ricky',
 'wilson',
 'the',
 'best',
 'frontman',
 'kaiser',
 'chiefs',
 'the',
 'best',
 'band']
```

Bu kısımda analizimizi olumsuz yönde etkileyecek olan the, and gibi gereksiz kelimeleri atıyoruz. Burada kısımda, veri kümesi ingilizce oldugu icin stopwords.words("english") kullandık. Diğer diller icin stopwords.words("ilgili dil") kullanılabilir.

```
description=[word for word in description if not word in
set(stopwords.words("english")) ]
```

Tweetin son halindeki ilk 10 kelimeyi görelim. Aşağıdaki çıktıdan görüldüğü üzere ‘the’ kelimeleri atıldı.

```
description[:10]
```

```
['ricky',
 'wilson',
 'best',
 'frontman',
 'kaiser',
 'chiefs',
 'best',
 'band',
 'xxxx',
 'thank']
```

Daha sonra lemmatization yaparız. Lemmatization, elinizdeki metinde bulunan kelimenin köküne inmektedir. Örneğin; loved➔love gibi

```
import nltk as nlp
nltk.download('wordnet')
lemma = nlp.WordNetLemmatizer()
description=[lemma.lemmatize(word) for  word in description]
```

Son duruma bir göz atalım.

```
description[:10]
```

```
['ricky',
 'wilson',
 'best',
 'frontman',
 'kaiser',
 'chief',
 'best',
 'band',
 'xxxx',
 'thank']
```

Önişlemeden geçirdiğimiz bu metni birleştirerek, bir metin olarak son haline bakalım.

```
description=" ".join(description)
description
```

```
'ricky wilson best frontman kaiser chief best band xxxx thank kaiser chie
f incredible year gig memory cherish always xxxxxxx'
```

Örnek tweetimizin değişimini. Aşağıdaki gibi özetlersek;

**1.Durum**

'Ricky Wilson The Best FRONTMAN/Kaiser Chiefs The Best BAND Xxxx Thank you Kaiser Chiefs for an incredible year of gigs and memories to cherish always :) Xxxxxxx'

**2. Durum** (regular expressiondan gecti;noktalama isaretleri, buyuk-kucuk harften kurtuldu
ricky wilson the best frontman kaiser chiefs the best band xxxx thank you kaiser chiefs for an incredible year of gigs and memories to cherish always xxxxxxx

**3. Durum**
(once stopwords.words("english") ile and,the gibi kelimelerden kurtulduk,
sonra lemmatization kullanarak kelimelerin sadece köklerini bulduk
ornegin chiefs=> chief; memories=>memory gibi

'ricky wilson best frontman kaiser chief best band xxxx thank kaiser chief incredible year gig memory cherish always xxxxxxx

## 5. Tüm Veriye Uygulama

Bir önceki bölümde 4. Tweeti alarak, her tweet metininin işlenmesi adım adım göstermiştik. Bu bölümde bunu tüm veriye uygulayalım.

```
description_list = []
for description in data.description:
    description = re.sub("[^a-zA-Z]"," ",description)
    description = description.lower()   # buyuk harftan kucuk harfe cevirme
    description = description.split()
    #description  =  [  word  for  word  in  description  if  not  word  in
set(stopwords.words("english"))]
    lemma = nlp.WordNetLemmatizer()
    description = [ lemma.lemmatize(word) for word in description]
    description = " ".join(description)
    description_list.append(description)
```

Verimize bulunan metinlerin herbirini uygun olmayan kelimelerden arındırarak temizledik. Şimdi bu metinleri kelime sıklığına göre sayısalllaştıralım. Sayısallaştırma aşamasında kullanacaağımız Bag of Words yöntemini önce aşağıdaki cümlelerle ifade etmeye çalışalım.

(1) John likes to watch movies. Mary likes movies too.
(2) Mary also likes to watch football games.

"John","likes","to","watch","movies","Mary","likes","movies","too"
"Mary","also","likes","to","watch","football","games"

BoW1 = {"John":1,"likes":2,"to":1,"watch":1,"movies":2,"Mary":1,"too":1};
BoW2 = {"Mary":1,"also":1,"likes":1,"to":1,"watch":1,"football":1,"games":1};

BoW3=BoW1 U BoW2
BoW3 = {"John":1,"likes":3,"to":2,"watch":2,"movies":2,"Mary":2,"too":1,"also":1,"football":1,"games":1};

(1) [1, 2, 1, 1, 2, 1, 1, 0, 0, 0]
(2) [0, 1, 1, 1, 0, 1, 0, 1, 1, 1]

```
from sklearn.feature_extraction.text import CountVectorizer # bag of words yaratmak icin
max_features = 5000
count_vectorizer = CountVectorizer(max_features=max_features,stop_words = "english")
sparce_matrix = count_vectorizer.fit_transform(description_list).toarray()
```

En sık kullanılan 5000 kelime aşağıdaki komut ile yazdırılmıştır.

```
print("EN SIK KULLANILAN {} KELIME GRUBU:\n{}".format(max_features,count_vectorizer.get_feature_names()))
```

```
EN SIK KULLANILAN 5000 KELIME GRUBU:
 ['aa', 'aaron', 'abc', 'ability', 'able', 'absolute', 'absolutely', 'abu
se', 'ac', 'academia', 'academic', 'academy', 'acc', 'accept', 'accepted
', 'access', 'accessory', 'accident', 'account', 'accountant', 'accountin
g', 'ace', 'achieve', 'act', 'acting', 'action', 'active', 'activist', 'a
ctivity', 'actor', 'actress', 'actual', 'actually', 'ad', 'adalah', 'adam
', 'add', 'added', 'addict', 'addicted', 'addiction', 'addition', 'addres
s', 'admin', 'administrator', 'admirer', 'adopted', 'adoption', 'adorable
', 'adore', 'adult', 'advance', 'advanced', 'advancing', 'advantage', 'ad
venture', 'adventurer', 'adventurous', 'advertising', 'advice', 'advise',
'adviser', 'advisor', 'advisory', 'advocacy', 'advocate', 'advocating', '
aerial', 'aerospace', 'aesthetic', 'af', 'afc', 'affair', 'affiliate', 'a
ffiliated', 'affiliation', 'affordable', 'aficionado', 'afraid', 'africa
', 'african', 'afrikaner', 'afro', 'afternoon', 'ag', 'age', 'agency', 'a
gender', 'agent', 'agile', 'ago', 'agree', 'agreement', 'agriculture', 'a
h', 'ahead', 'ahs', 'ai', 'aim', 'aime', 'aiming', 'ain', 'aint', 'air',
'airplane', 'aj', 'aka', 'akps', 'akun', 'al', 'alabama', 'alberta', 'alb
um', 'alcohol', 'ale', 'alert', 'alex', 'alexis', 'alfie', 'algorithm', '
ali', 'alice', 'alien', 'alive', 'allah', 'allergic', 'alliance', 'alot',
```

Verimizi artık etiket verisini ayırarak makine öğrenmesi algoritmalarını kullanır hale getirelim

```
y = data.iloc[:,0].values   # male or female classes
x = sparce_matrix

x.shape
(16224, 5000)
```

Son olar veriyi test-train olarak ayıralım.

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.1,
random_state = 42) from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb_model=nb.fit(x_train,y_train)
```

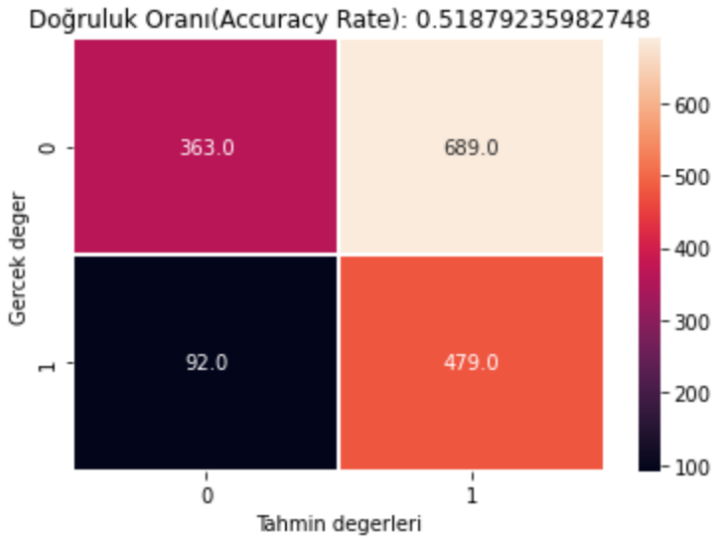
## 6. Makine Öğrenmesi Algoritmalarını Uygulama

### 6.1. Naïve Bayes

```
from sklearn.metrics import confusion_matrix, accuracy_score,
classification_report
import seaborn as sns
import matplotlib.pyplot as plt
```

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb_model=nb.fit(x_train,y_train)
y_pred = nb_model.predict(x_test)

sns.heatmap(cm,annot=True,fmt=".1f", linewidths=.3)
plt.ylabel('Gercek deger')
plt.xlabel('Tahmin degerleri')
plt.title('Doğruluk Oranı(Accuracy Rate): {0}'.format(nb.score(x_test,
y_test)), size = 12)
plt.show()
```




```
print(classification_report(y_test, y_pred))
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.80 | 0.35 | 0.48 | 1052 |
| 1 | 0.41 | 0.84 | 0.55 | 571 |
| accuracy | | | 0.52 | 1623 |
| macro avg | 0.60 | 0.59 | 0.52 | 1623 |
| weighted avg | 0.66 | 0.52 | 0.51 | 1623 |

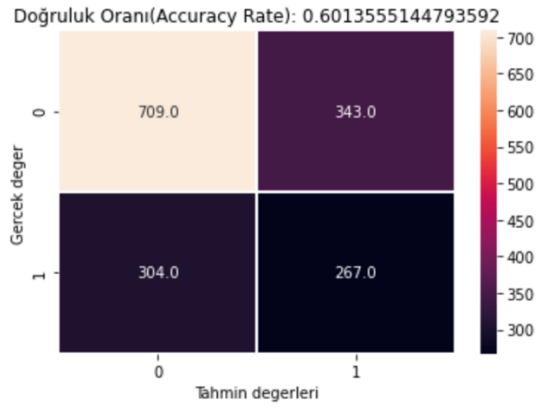
Max_features değerini ne kadar arttırıp deneyerek başarı oranını arttırabiliriz.değeri arttırmak başarı oranını düşürebilir de. Naive bayes modeli bu uygulama için çok da uygun değildir.

## 6.2. KNN

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(3)
knn_model = knn.fit(x_train, y_train)
y_pred = nb_model.predict(x_test)

accuracy_score(y_test, y_pred)

sns.heatmap(cm,annot=True,fmt=".1f", linewidths=.3)
plt.ylabel('Gercek deger')
plt.xlabel('Tahmin degerleri')
plt.title('Doğruluk Oranı(Accuracy Rate): {0}'.format(nb.score(x_test,
y_test)), size = 12)
plt.show()
```




```
print(classification_report(y_test, y_pred))
```

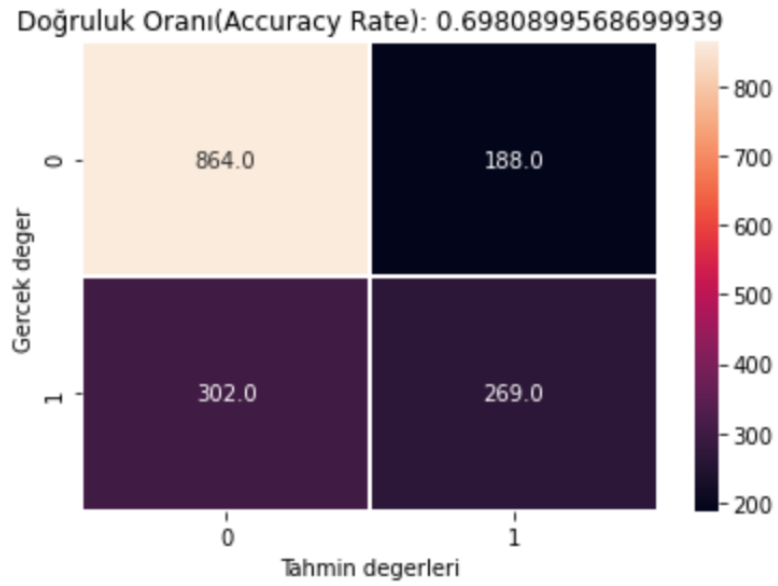
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.70 | 0.67 | 0.69 | 1052 |
| 1 | 0.44 | 0.47 | 0.45 | 571 |
| accuracy | | | 0.60 | 1623 |
| macro avg | 0.57 | 0.57 | 0.57 | 1623 |
| weighted avg | 0.61 | 0.60 | 0.60 | 1623 |

## 6.3. Lojistik Regresyon

```
from sklearn.linear_model import LogisticRegression
```

```
lr = LogisticRegression(solver = "liblinear")
lr_model=lr.fit(x_train, y_train)y_pred = nb_model.predict(x_test)
y_pred = nb_model.predict(x_test)
```

```
sns.heatmap(cm,annot=True,fmt=".1f", linewidths=.3)
plt.ylabel('Gercek deger')
plt.xlabel('Tahmin degerleri')
plt.title('Doğruluk Oranı(Accuracy Rate): {0}'.format(nb.score(x_test,
y_test)), size = 12)
plt.show()
```




```
print(classification_report(y_test, y_pred))
```

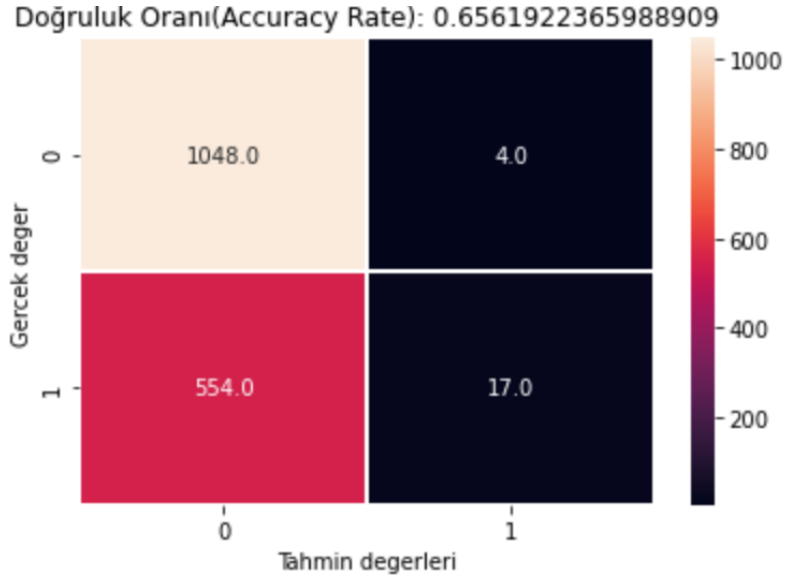
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.74 | 0.82 | 0.78 | 1052 |
| 1 | 0.59 | 0.47 | 0.52 | 571 |
| accuracy | | | 0.70 | 1623 |
| macro avg | 0.66 | 0.65 | 0.65 | 1623 |
| weighted avg | 0.69 | 0.70 | 0.69 | 1623 |

## 6.4. Karar Ağaçları

```
from sklearn.tree import DecisionTreeClassifier
```

```
DTree=DecisionTreeClassifier(max_depth=3,random_state=42)
DTree.fit(x_train, y_train)
y_pred = DTree.predict(x_test)

sns.heatmap(cm,annot=True,fmt=".1f", linewidths=.3)
plt.ylabel('Gercek deger')
plt.xlabel('Tahmin degerleri')
plt.title('Doğruluk Oranı(Accuracy Rate): {0}'.format(nb.score(x_test,
y_test)), size = 12)
plt.show()
```




```
print(classification_report(y_test, y_pred))
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.65 | 1.00 | 0.79 | 1052 |
| 1 | 0.81 | 0.03 | 0.06 | 571 |
| accuracy | | | 0.66 | 1623 |
| macro avg | 0.73 | 0.51 | 0.42 | 1623 |
| weighted avg | 0.71 | 0.66 | 0.53 | 1623 |

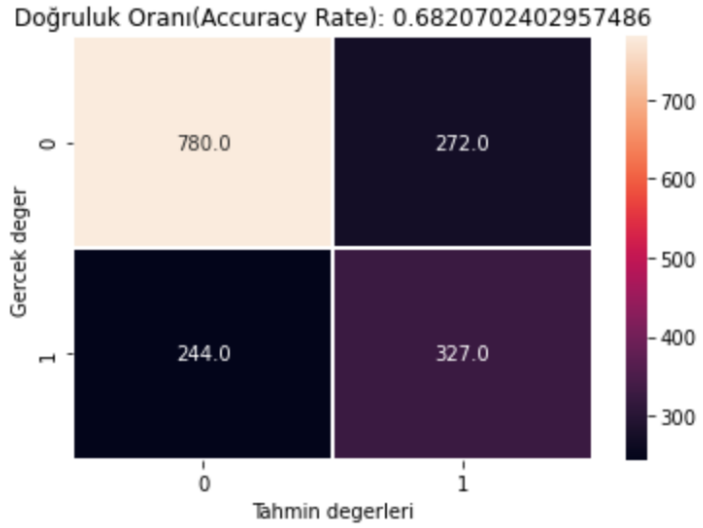
## 6.5. Random Forest

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier()
rf_model=rf.fit(x_train, y_train)
```

```
y_pred = rf_model.predict(x_test)

sns.heatmap(cm,annot=True,fmt=".1f", linewidths=.3)
plt.ylabel('Gercek deger')
plt.xlabel('Tahmin degerleri')
plt.title('Doğruluk Oranı(Accuracy Rate): {0}'.format(nb.score(x_test,
y_test)), size = 12)
plt.show()
```




```
print(classification_report(y_test, y_pred))
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.76 | 0.74 | 0.75 | 1052 |
| 1 | 0.55 | 0.57 | 0.56 | 571 |
| accuracy | | | 0.68 | 1623 |
| macro avg | 0.65 | 0.66 | 0.66 | 1623 |
| weighted avg | 0.69 | 0.68 | 0.68 | 1623 |

## 7. Tahmin

Etiket verisi olarak ayırdığımız cinsiyet verisini; erkek 0, kadın 1 kodu ile temsil edilecek şekilde sayısallaştırmıştık.

Şimdi yaptığımız farklı makine öğrenmelerinin, verdiğimiz tweet ya da tweet grubu ile ilgili değerlendirmelerini inceleyelim.

```
metin=pd.Series(['aa just for you deliver to your inbox','hi'])


nb_model.predict(sparce_matrix[-len(metin):])
array([0, 1])

knn_model.predict(sparce_matrix[-len(metin):])
array([0, 1])

rf_model.predict(sparce_matrix[-len(metin):])
array([0, 1])

lr_model.predict(sparce_matrix[-len(metin):])
array([0, 0])

DTree.predict(sparce_matrix[-len(metin):])
array([0, 0])
```

Görüldüğü üzere ilk tweet kullanılan tüm makine öğrenmeleri tarafından erkek olarak sınıflandırılırken, ikinci tweet de farklılıklar oluşmaktadır.

[1]https://www.kaggle.com/crowdflower/twitter-user-gender-classification/activity