YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Natural Language Processing with Disaster Tweets

PROJE RAPORU

SERCAN SÖZEN
21501071

ÖĞRETİM ÜYESİ

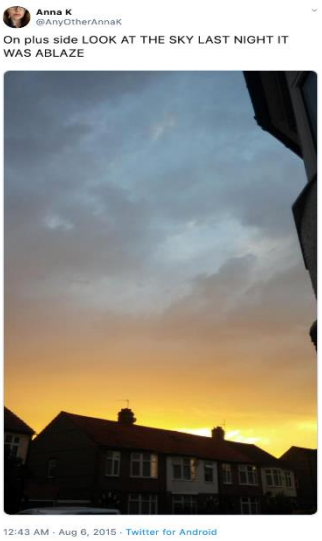
Prof.Dr. Mine Elif KARSLIGİL YAVUZ

Özet

Bu projede, hangi Tweetlerin gerçek felaketlerle ilgili olduğunu ve hangilerinin olmadığını tahmin eden bir makine öğrenimi modeli oluşturuldu. Elle sınıflandırılmış 7503 tweet'lik bir veri kümesine kullanıldı. Veri seti %70'i train, %15'i validasyon ve %15'i test olacak şekilde sınıfların oranları korunarak bölümlendi. Buna ek olarak K-Fold cross validasyon yöntemiyle K=5 ve K=10 değerleri kullanılarak çeşitli sınıflandırma modellerinin başarıları kıyaslandı. Tweet'ler farklı vektörizasyon fonksiyonlarıyla vektörlere dönüştürülüp, çeşitli sınıflandırıcıların bu vektörler üzerindeki başarı seviyeleri gözlemlendi.

1. Amaç

Twitter, acil durumlarda önemli bir iletişim kanalı haline geldi. Akıllı telefonların her yerde bulunması, insanların gözlemledikleri bir acil durumu gerçek zamanlı olarak duyurmalarını sağlıyor. Bu nedenle, daha fazla ajans (yani afet yardım kuruluşları ve haber ajansları) bilgiye hızlı erimek için Twitter'ı programlı olarak izleme yolunu tercih etmeye başladı. Ancak, bir kişinin sözlerinin gerçekten bir felaketi ilan edip etmediği her zaman net değildir. Bu örneği ele alalım:



Figür 1 Örnek bir kullanıcı tweet'i

Figür 1' deki tweet'in yazarı, alev alev yanmak anlamına gelen "ABLAZE" kelimesini kullanmakta ancak bunu mecazi anlamda ifade etmektedir. Bunu ayırt etmek, özellikle görsel yardımla, bir insan için kolaydır. Ancak bir makine için bu karmaşık bir hal alabilir.

2. Kapsam

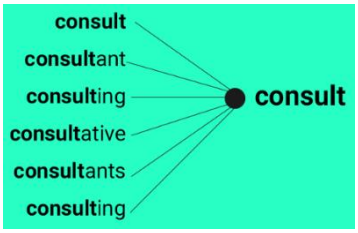
Çalışmada kullanılan veri seti 7503 adet veriden oluşmaktadır. Bu verilerin 4307 tanesi negatif sınıf, 3196 tanesi pozitif sınıf etiketine sahiptir. Aynı zamanda bu veri seti 5 adet sütundan oluşmaktadır. Bunlar sırasıyla "id, keyword, location, text, target" şeklindedir. "Keyword" sütunu ilgili tweet'te kullanılmış ve bir felaket haberi niteliği taşıyabilecek kelimeleri içerir. "Location", tweet'in yüklendiği konumu, "text" tweet'in içerdiği mesajı ve "target" ise bu tweet'in gerçek bir felaket olup olmadığını ifade eder. Eğer veri setindeki bir tweet'in "target" etiketi 1 ise bu tweet gerçek bir felaketi anlatmaktadır. Kullanılan veri seti "Figure Eight Company" şirketi tarafından sağlanmıştır. Veri setine kendi "Data for Everyone" sayfalarından erişim sağlanabilir.

3. Sistem Tasarımı

Bu bölümde sistem tasarımından bahsedilecek. Verinin önişlenmesi, kullanılacak sınıflandırıcılar, ve parametre ayarlamalarına değinilecek.

3.1 Verinin Önişlenmesi

Her doğal dil işleme projesinde olduğu gibi, bu projede de öncelikle yapılması gereken elimizdeki verinin makine öğrenmesi modellerine girdi olarak verilebilecek şekilde temizlenmesi gerekmektedir. Tweet'leri cümle içinde kullanılan ve bağlamı etkilemeyecek olan "stopwords" kelimelerinden, HTML etiketlerinden, emojilerden arındırmak gerekir. Ayrıca bunlara ek olarak kelimeler NLTK kütüphanesinin PorterStemmer metodu yardımı ile eklerinden ayrılarak farklı ek almış aynı köke sahip kelimeler Figür 2'deki gibi gruplanabilir.



Figür 2 Aynı köke sahip kelimelerin gruplanması

Bu projede tweet'lerdeki HTML etiketlerive etiketlenen kullanıcı adları, Regular Expression kütüphanesi kullanılarak boş bir stringle değiştirildi. Tweet içerisinde bulunan emojiler ise Figür 3' teki emoji sözlüğü kullanılarak benzer şekilde değiştirildi.

```
EMOJIS = {':)': 'smile', ':-)': 'smile', ';d': 'wink', ':-E': 'vampire', ':(': 'sad',
          ':-(': 'sad', ':-<': 'sad', ':P': 'raspberry', ':O': 'surprised',
          ':-@': 'shocked', ':@': 'shocked',':-$': 'confused', ':\\': 'annoyed',
          ':#': 'mute', ':X': 'mute', ':^)': 'smile', ':-&': 'confused', '$_$': 'greedy',
          '@@': 'eyeroll', ':-!': 'confused', ':-D': 'smile', ':-0': 'yell', 'O.o': 'confused',
          '<(-_-)>': 'robot', 'd[-_-]b': 'dj', ":'-)": 'sadsmile', ';)': 'wink',
          ';-)': 'wink', 'O:-)': 'angel','O*-)': 'angel','(:-D': 'gossip', '=^.^=': 'cat'}
```

Figür 3 Emojilerin değimi için kullanılan sözlük

3.2 Eğitim, Validasyon ve Test Bölümlemesi

Projede kullanılan veri seti Sklearn kütüphanesinin train_test_split metodunun 2 kez kullanımıyla %70'lik kısım eğitim, %15'lik kısım validasyon ve %15'lik kısım test için olacak şekilde ayırıldı.

```
X_train, x_temp, y_train, y_temp = train_test_split(X, y, stratify=y,
                                                    train_size=0.7)

X_val, x_test, y_val, y_test = train_test_split(x_temp, y_temp, stratify=y_temp,
                                                test_size=0.5)
```

Figür 4 Eğitim, validasyon ve test bölümlemesi

Buna ek olarak aynı veri seti K-Fold Cross Validation yöntemiyle, katlama sayısı 5 ve 10 olacak şekilde bölünerek kullanılacak sınıfılandırıcılar üzerindeki etkisi gözlemlendi.

3.3 Kelimelerin Vektörizasyonu

Kelimelerin makine öğrenmesi modellerine girdi olarak verilebilmesi için vektörize edilerek sayılarla temsil edilmesi gerekmektedir. Bu projede vektörizasyon fonksiyonları olarak Count Vectorizor (Bag of Words) ve TF-IDF vectorizor kullanıldı. Özellik sayısı parametresi olarak (bu projede her bir özellik bir kelimedir) her iki fonksiyonun parametresi de maksimum 4000 olacak şekilde belirlendi.

3.4 Kullanılan Sınıflandırıcılar ve Parametreler

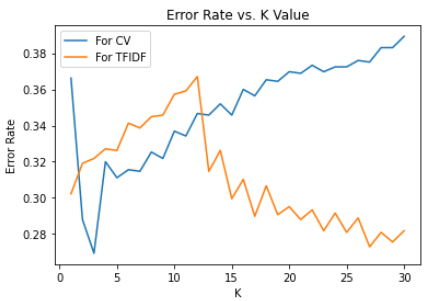
Projede farklı sınıflandırıcılar kullanılarak (Lojistik regresyon, karar ağacı, naive bayes, perceptron ve k – en yakın komşuluk) başarıları kıyaslandırıldı.

3.4.1 Naive Bayes

Naive bayes sınıflandırıcısı olarak Sklearn Naive Bayes kütüphanesindeki Gaussain Naive Bayes ve Multinomial Naive Bayes sınıflandırıcıları kullanıldı.

3.4.2 K – en yakın komşuluk

K en yakın komşu sınıflandırıcı için 1-31 arasında komşu sayısı değerleri denendi. Bu aralıktaki komşuluk değerleri için Figür 5' te de görülebileceği üzere CV vektörizasyonu için en az hata 3 komşulukta, TF-IDF vektörizasyonu için ise en az hata 27 komşulukta belirlendi.



Figür 5 Her iki vektörizasyon fonksiyonu için farklı komşuluk değerlerindeki hata oranı

3.4.3 Karar Ağacı

Karar ağacı da benzer şekilde hem CV, hem de TF-IDF için denendi. En optimal parametreler olarak maksimum ağaç derinliği 70, maksimum yaprak düğüm sayısı ise 100 olarak seçildi.

3.4.4 Perceptron

Çok katmanlı perceptron için aktivasyon fonksiyonu relu, çözücü olarak sigmoid fonksiyonu, başlangıç learning rate olarak 0.1 değeri belirlendi. Ayrıca learning rate adaptive olarak ayarlandı. Böylece iterasyonlardaki hata oranı aşağı yönde ilerlediği taktirde learning rate parametresi olarak başlangıçta verilen sabit değerin kullanılması sağlandı. Kullanılan 2 adet gizli katmandaki nöron sayıları da sırasıyla 5 ve 10 olarak belirlendi.

3.4.5 Lojistik Regresyon

Lojistik regresyon da diğer sınıflandırıcılara benzer şekilde her iki vektörizasyon fonksiyonunun çıktısıyla denendi ve başarıları gözlemlendi. Parametre olarak sadece sınıfların ağırlığı "balanced" olarak belirlendi.

4. Deneyler

Sınıflandırma modellerinin değerlendirilmesi için F1 – measure kullanıldı. F1 aşağıdaki gibi hesaplanır:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Denklem 1 F1-Measure hesaplanması

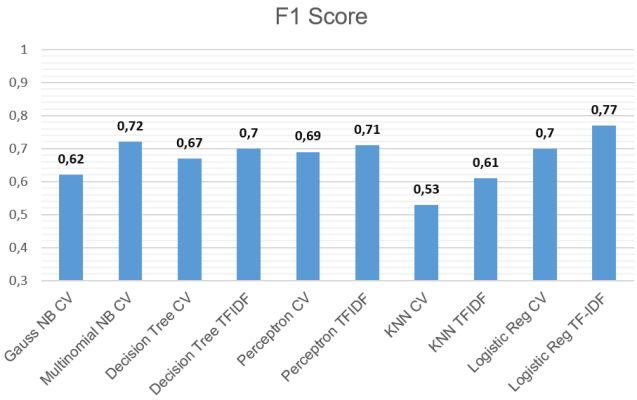
Denklem 1 F1-Measure hesaplanmasıDenklem 1' deki precision ve recall değerleri karışıklık matrisindeki doğru tahmin yapılan pozitif örnek sayısı (TP), yanlış tahmin yapılan pozitif örnek sayısı (FP), doğru tahmin yapılan negatif örnek sayısı (TN) ve yanlış tahmin yapılan negatif örnek sayısı (FN) kullanılarak aşağıdaki gibi hesaplanırlar:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

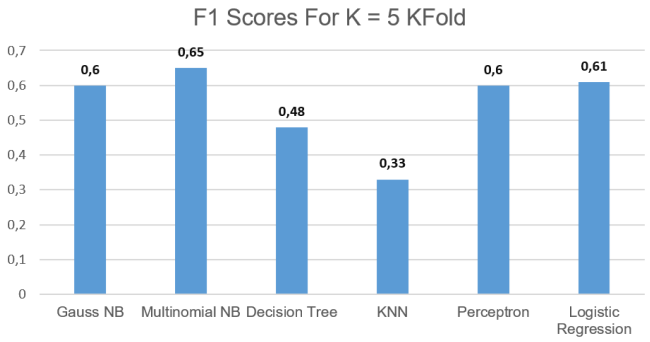
Denklem 2 Recall ve Precision değerlerinin hesaplanması

Her sınıflandırıcının ayrı ayrı F1 değerleri hesaplanarak karşılaştırıldı. Figür 6 'da gözlemlenebileceği gibi veriler train-validation-test şeklinde bölümlendiğinde en yüksek başarıyı sağlayan sınıflandırıcı 0.77'lik F1 skoru ile TF-IDF vektörizasyonuyla sağlanan veriye uygulanan lojistik regresyon oldu.

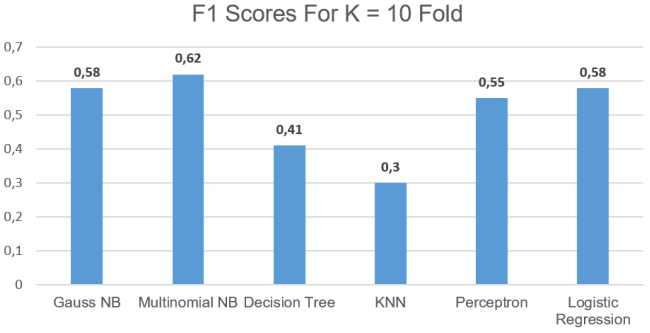


Figür 6 Farklı sınıflandırıcıların farklı vektörizasyonla sağlanan matris üzerindeki sınıflandırma başarıları

Daha sonrasında aynı veri seti K-Fold cross validation yöntemiyle önce 5 sonrasında 10 katlamayla bölünerek aynı sınıflandırıcıların başarıları Figür 7 ve gözlemlendi.



Figür 7 Farklı sınıflandırıcıların K-fold (K=5) validasyonu üzerindeki sınıflandırma başarıları



Figür 8 Farklı sınıflandırıcıların K-fold (K=10) validasyonu üzerindeki sınıflandırma başarıları

5. Çıktılar ve Yapılacaklar

Yapılan deneylerden sağlanan en yüksek başarı 0.77'lik F1 değeri ile lojistik regresyon ile sağlandı. K-Fold cross validasyonda genel olarak başarının düştüğü gözlemlendi. Özellikle K-en yakın komşuluk sınıflandırıcısında komşuluk parametresi artırıldığında başarının yüksek oranda düştüğü gözlemlendi. Bundan yola çıkarak sınıflandırılan veri kümesindeki örneklerden farklı sınıfta olanlar birbirlerine yakın konumda olduğunda komşuluk sınıflandırılmasının başarısız olduğu yorumu yapılabilir.

Projenin devamı niteliğinde bir çalışma olarak 2019'de Google'ın önerdiği "semi-supervised" bir derin öğrenme modeli olan BERT (Bidirectional Encoder Representations from Transformers) (Devlin, Chang, Lee, & Toutanova, 2019) kullanırak aynı veri seti üzerinde bir sınıflandırma gerçekleştirilebilir. Bu proje bir "Kaggle Competition" projesi olduğundan, Kaggle'daki skor tablolarını incelediğimde buradaki BERT ile yapılan çalışmaların 0.84 F1 skoru başarısına ulaştıklarını gözlemledim.

Bunun da ilerisinde bir çalışma olarak yine Google'ın 2020 Mart ayında tanıttığı ELECTRA (Clark, Luong, Le, & Manning, 2020) modeli yine aynı veri seti üzerinde denenip başarı ölçümlenebilir. ELECTRA' nın GLUE (general language understanding) skoru 85 iken BERT' in GLUE skoru 80 olarak ölçümlenmiştir. Dolayısıyla ELECTRA kullanıldığında daha yüksek bir başarı öngörülebilir.

Referanslar

Clark, K., Luong, M.-T., Le, Q., & Manning, C. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *ICLR*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.