Doğal Dil İşleme Dersi Dönem Projesi

AD SOYAD NO: Serdar Yıldız 16011004

AD SOYAD NO: Kamil Başkut 15011091

İçindekiler

Giriş

Amaç

Günümüz dünyasında insanların büyük bir çoğunluğu internet ile etkileşimde bulunmaktadır. Bu sebeple dünya üzerinde anlık olarak milyonlarca içeriğe yorum yapılmaktadır. Bu yorumların her birinin ise olumlu, insan haklarını çiğnemeyen, güzel yorumlar olmadığı da bir gerçektir. İnternette yaşanan bu yoğun yorum trafiğinde hangi yorumun kötü hangi yorumun iyi olduğunu bir insan tarafından okuyarak sınıflandırılması ise mümkün değildir. Bu sebeple sınıflandırma işlemini İngilizce üzerinden yapacak olan bu sistemin projesi önerilmiştir.

Yöntem

Eğiticili öğrenme metotları kullanılacağı için etiketli veriler üzerinden proje ilerletilmiştir. Bu sebeple kaggle platformu üzerinden ***'Toxic Comment Classification Challenge'***[1] ve ***'Jigsaw Unintended Bias in Toxicity Classification'*** [2] veri setleri birleştirilerek projeye uygun bir veri seti elde edilmiştir. Veri setlerinden kullanılmayacak olan bazı öznitelikler çıkarılmış sadece metin ve etiketi alınarak veri sadeleştirilmiştir.

Sadeleştirilmiş veri üzerinde temizleme işlemleri yapılmış ve veri daha dengeli bir hale geritirilmiştir. Daha sonrasında bilgisayarın anlayabileceği formata dönüştürülmüştür. Elde edilen bilgisayar için anlamlı veri test ve eğitim olarak belirli bir oranda iki parçaya ayrılmıştır. Eğitim için bazı algoritmalar belirlenmiş ve bu algoritmalar test verisi kullanılarak test edilmiştir.

Tüm bu yapılanlar sonucunda elde edilen en başarılı algoritma seçilerek sistem tamamlanmıştır.

Ön İnceleme

Sistemde kullanılacak olan 2 adet veri vardır. Bunlardan biri çok büyük sayılarda veri içerirken, diğeri nispeten küçüktür. Büyük olan veri seti *Toxic Comment Classification Challenge iken daha küçük olan* Jigsaw Unintended Bias in Toxicity Classification'dır.

Verilerin Öznitelikleri

- ***Toxic Comment Classification Challenge verisinin öznitelikleri**:*

  'id', 'target', 'comment_text', 'severe_toxicity', 'obscene', 'identity_attack', 'insult', 'threat', 'asian', 'atheist', 'bisexual', 'black', 'buddhist', 'christian', 'female', 'heterosexual', 'hindu', 'homosexual_gay_or_lesbian', 'intellectual_or_learning_disability', 'jewish', 'latino', 'male', 'muslim', 'other_disability', 'other_gender', 'other_race_or_ethnicity', 'other_religion', 'other_sexual_orientation', 'physical_disability', 'psychiatric_or_mental_illness', 'transgender', 'white', 'created_date', 'publication_id', 'parent_id', 'article_id', 'rating', 'funny', 'wow', 'sad', 'likes', 'disagree', 'sexual_explicit', 'identity_annotator_count', 'toxicity_annotator_count'

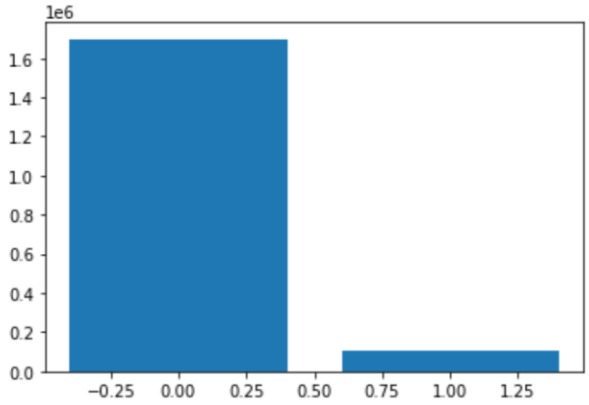
- **Jigsaw Unintended Bias in Toxicity Classification verisinin öznitelikleri:**

  'id', 'comment_text', 'toxic', 'severe_toxic', 'obscene', 'threat', 'insult', 'identity_hate'

Veri Büyüklükleri

- ***Toxic Comment Classification Challenge:***

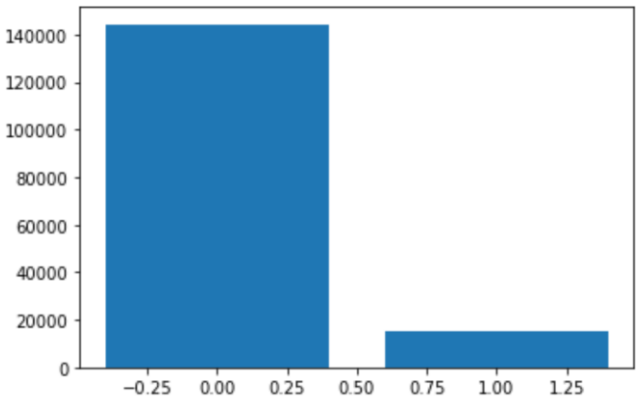
  Bu veride toplamda ***1.804.874*** adet etiketli yorum bulunmaktadır. Bunlardan 1.698.436 adet veri normal(Nontoxic) olarak, 106.438 tanesi ise toxic olarak etiketlenmiştir. Sınıf dağılımı aşağıdaki grafikte verilmiştir.



Grafik 1

- **Jigsaw Unintended Bias in Toxicity Classification:**

Bu veride toplamda ***159.571*** adet etiketli yorum bulunmaktadır. Bunlardan 144.277 adet veri normal(Nontoxic) olarak, 15.294 tanesi ise toxic olarak etiketlenmiştir. Sınıf dağılımı aşağıdaki grafikte verilmiştir.



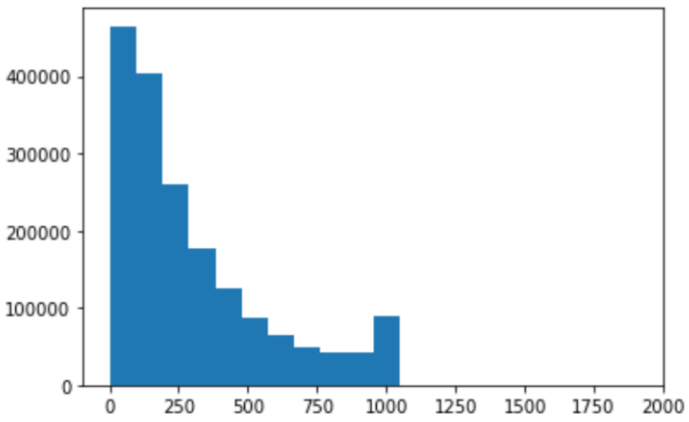
Grafik 2

Verilerdeki Metinlerin Karakter Boyutları

Verinin her birindeki metnin karakter uzunlukları birbirinden doğal olarak farklıdır. Kimi metin çok uzun iken kimi metin çok daha kısadır.

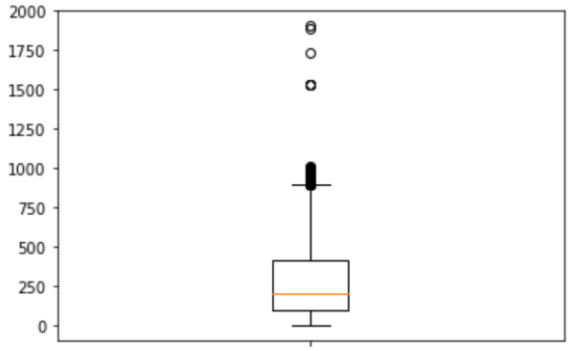
- ***Toxic Comment Classification Challenge:***

Aşağıdaki dağılım grafiğinden de görüleceği üzere yığılma kısa metinler üzerindedir. Yaklaşık 500.000 metin karakter sayısı 0-100 arasındadır. Yaklaşık 400.000 adet de 100-200 karakter sayısı arasında vardır. Bu da verinin neredeyse yarısının metin uzunluğunun 200'ü geçmediğini göstermektedir. Grafikte x ekseni metin uzunluğunu gösterirken y ekseni kaç adet bulunduğunu gösterir.



Grafik 3

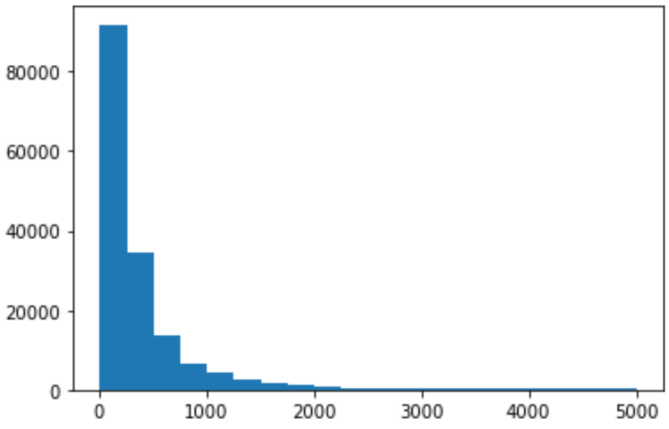
Dağılımın kutu grafiği aşağıdaki gibidir.



Grafik 4

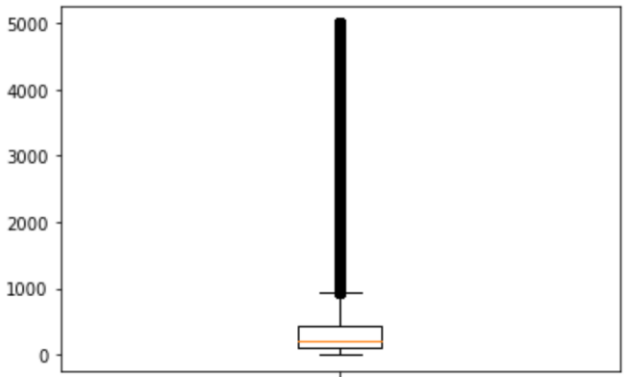
- **Jigsaw Unintended Bias in Toxicity Classification**

Aşağıdaki dağılım grafiğinden de görülebileceği üzere bir önce bahsedilen verideki dağılıma benzerdir. Bu durum iki verinin birleştirildiğinde sorun olmayacağı anlamına da gelmektedir. Grafikte x ekseni metin uzunluğunu gösterirken y ekseni kaç adet bulunduğunu gösterir.



Grafik 5

Dağılımın kutu grafiği aşağıdaki gibidir.

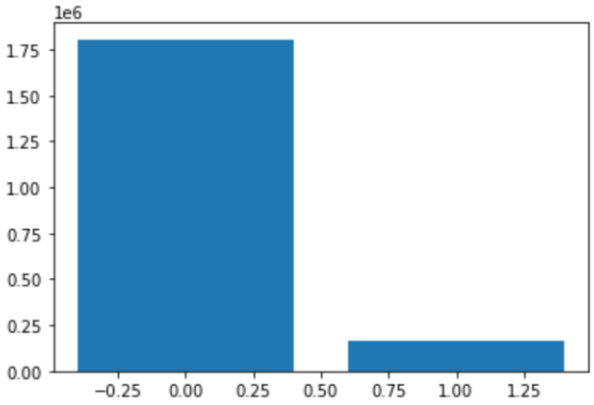


Grafik 6

Sistem Analizi

Verilerin Birleştirilmesi

Sistemi eğitmek için bir eğitim verisine ihtiyaç vardır. Bu eğitim verisini elde etmek için iki veri seti birleştirilmiştir. Birleştirilen iki veri seti sonucunda elde edilen eğitim verisinin dağılımı aşağıdaki grafikte görüldüğü gibidir.



Grafik 7

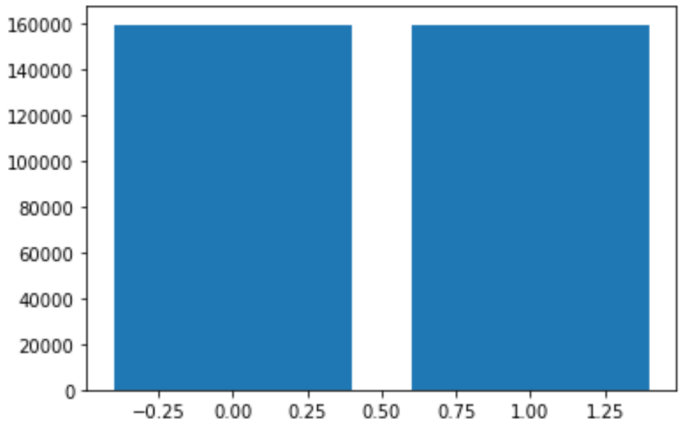
'Toxic Comment Classification Challenge' veri setinde etiketler 0 ve 1 arasında değerler alabilmektedir. Eğitimin regresyon değil sınıflandırma üzerinden yapılacağı için 0.5 eşik değeri belirlenip bu değerden eşit ve yüksel olanlar 1 küçük olanlar 0 olarak etiketlenmiştir.

Sınıfların Dengelenmesi

Grafik-1 ve grafik-2'den de görüleceği üzere sınıflar arasında bir dengesizlik bulunmaktadır. 1'e 10'luk bir oranda sistemin tamamına 'nontoxic' olarak etiketlemesi durumunda dahi sistem başarısı %90'ın üzerindedir. Aynı zamanda sınıf dağılımlarının bir tarafta yoğunlaşması eğitimi de olumsuz etkilemetedir. Bu sebeple sınıfların dağılımlarının eşitlenmesi gerekmektedir.

Sınıf dağılımlarının eşit hale getirilmesi amacı ile iki veri seti için de tüm 'toxic' etiketli veriler alınmıştır. Bu şekilde hedeflenen sınıfın tamamı eğitime dahil edilmiştir. Daha sonrasında 'toxic' etiketli veriler kadar 'nontoxic' etiketli verilerden rastgele bir seçim yapılmıştır. Bu şekilde 'toxic' ve 'nontoxic' olarak etiketli iki sınıftan eşit sayıda alınarak dengeli bir eğitim verisi oluşturulmuştur.

Elde edilen sonucu gösteren sınıf dağılım grafiği aşağıdaki grafikte görüldüğü gibidir.



Grafik 8

Veriyi dengeleme işleminden sonra her bir sınıftan 159.628 adet metin, toplamda 319.256 adet metin verisi ile eğitim verisi oluşturulmuştur.

Ön işleme

Metin içerisinde bazı istenmeyen karakterler vardır. Bunlara örnek vermek gerekirse noktalama işaretleri, tırnak sonrasında gelen kısaltılmış kelimeler('I'm' metnindeki 'm' gibi), başlangıçta ve sonda oluşan boşluklar gibi. Bunların ortadan kaldırılması için Regex komutları yazılmıştır. Yazılan regex komutları aşağıda listelendiği gibidir.

- '\W'

  Harf ve rakam dışındaki karakterleri atmak için kullanılır. Bu karakterler yerine boşluk yazılır. Bu şekilde noktalama işaretleri çıkarılmış olur.

- '\s+[a-zA-Z]\s+'

  Noktalama işaretlerinin çıkarılması ile tek harften oluşan öbekler kalmış olabilir. Bunları da silmek için yukarıda yazılmış olan regex komutu kullanılır. 'I'm' örneğinde olduğu gibi tırnaktan sonra gelen m harfinin yerine de boşluk karakteri getirilir.

- '\^[a-zA-Z]\s+'

  Başlangıçta bulunan tek karakterin yerine boşluk karakteri basar.

- '\s+'

  Birden fazla boşluk karakteri için tek boşluk karakteri basar.

Yukarıdaki regex komutlarının çalışması ile metin sadeleşmiş ve makine için anlamlandırma işlemine alınması için daha uygun hale getirilmiştir.

Regex komutları sonucunda elde edilen yeni metin karakterlerinin hepsi küçük harf yapılarak devam edilir. Bu işlemden sonra son adım olarak lemmatization yapılır ve önişleme süreci tamamlanmış olur.

"The best course's Natural Language Processs that I have ever studied!! " örnek metni yukarıdaki yukarıda anlatıldığı işlenirse elde edilen sonuç metni ***"the best course natural language process that have ever studied'"*** şeklindedir.

Metinleri Algoritmalar İçin Uygun Hale Getirme

Metin halinde bunları makine öğrenmesi modelleri veya yapay sinir ağı modellerine veremeyeceğimiz için bunları anlamlı sayılara dönüştürmek zorundayız. Bu amaç doğrultusunda kullanılan 2 ana yöntem vardır. Bunlar Bag of Words modeli[3] ve Word Embeding modelidir[4].

Elimizdeki verinin yeteri kadar kapsayıcı olduğunu düşüncesi ve aynı zamanda daha hızlı olması sebebiyle Bag of Words modeli tercih edilmiştir.

Bu amaç doğrultusunda Bag of Words modelini oluşturmak için sklearn[5] kütüphanesi kullanılmıştır. Bu kütüphane kullanılarak elde edilen deneyimler göz önünde bulundurularak "*max_features=512, min_df=5, max_df=0.95, stop_words=stopwords.words('english'*)" olarak sabitlenmiştir. Stopword kavramı kullanılarak anlamlandırmada işe yaramaz kelimeler çıkarılmıştır.

Sonuç olarak metin girdisinden 512 adet yeni öznitelik elde edilmiştir. Öznitelik sayısının artması sistem çalışma süresinin kabul edilemez sürelere bulaşması sebebiyle 512 değerinin optimum değer olduğu sonucuna varılmıştır.

Makine Öğrenmesi Modelleri

Sınıflandırma problemi için çok uzun zamandır kullanılan çok meşhur algoritmalar bulunmaktadır. Bu projede karşılaştırılabilmesi için öncelikle denen algoritmaların bu algoritmalar olması kararlaştırılmıştır. Bahsi geçen algoritmalar

- ***Decision Tree algoritması***
- ***Random Forest algoritması***
- ***AdaBoost algoritması***
- ***SVM algoritması***
- ***K-nn algoritması***
- ***Gaussian Naive Bayes algoritması***

Yukarıda bahsi geçen algoritmalardan 2 tanesi için sonuç alınamamıştır. Eğitim süreleri 6 saatin üzerinde olduğu kesindir. Bu algoritmalar SVM ve K-nn algoritmalarıdır.

Bu algoritmalara ek olarak daha önceden yapılan çalışmalarda başarısını ispatlamış ***XGBoost***[6] algoritması kullanılmıştır. Bu algoritma birçok algoritmanın başarılı taraflarını alarak yepyeni bir model ortaya koymuştur. Veri ile random forest algoritmasında olduğu gibi birçok karar ağacı oluşturmaktadır. Bu karar ağaçları sonucunda elde edilen sınıf çıktılarını ağırlıklandırmaktadır. Bu ağrılıkları oylayarak en doğru kararı vermektedir.

Eldeki verinin %20'si test olarak ayrılarak algoritmaların başarıları ölçülmüştür. Bu algoritmalar sonucunda elde edilen başarılar ve ortalama çalışma süreleri aşağıdaki tabloda verilmiştir.

Tablo 1

MODEL	Doğruluk (%)	Run Time
Decision Tree	69.17	3min 58s
AdaBoost	74.66	3min 7s
Random Forest	75.71	11min 42s
SVM	?	+6H
KNN	?	+6H
Naive Bayes	69.06	2.03s
XGBoost	76.17	1min 10s

XGboost algoritması ile elde edilen başarı en yüksek başarıdır. Bu sebeple xgboost algoritmasının parametrelere denemeleri yaparak başarı arttırılmaya çalışılmıştır. Xgboost algoritmasında öznitelik seçimini kendi yapabildiği için çıkarılan öznitelik sayısı 512 den 1500'e çıkarılmıştır. Yapılacak olan denemelere bu şekilde devam edilmiştir.

Yapılan denemeler aşağıdaki tabloda görüldüğü gibidir.

Tablo 2

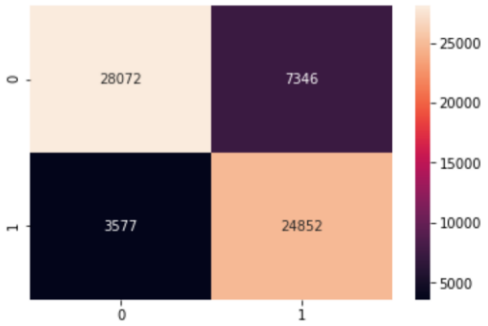
Oluşturulacak Ağaç Sayısı	***Maksimum Ağaç Derinliği***	***Test Başarısı (%)***
100	25	82,65
200	25	82,84
200	20	82,97
250	20	82,90
180	20	82,86
180	16	82,73
160	20	82,84
160	16	82,73

Tablo 2'den de görüleceği üzer en yüksek başarı ağaç sayısı 200, maksimum ağaç derinliği 20 olarak ayarlandığında elde edilmiştir ve bu modelin başarısı %82,97'dir. Ancak diğer elde edilen başarılara bakıldığında %1'den fazla bir değişim görülmemektedir. Fakat test boyutunun 63,847 olması sebebi ile %1'lik bir hata 600'den fazla metnin yanlış sınıflandırılması anlamına gelir. Bu sebeple en başarılı olan parametreler karmaşıklığı ve eğitim süresi uzun olsa da seçilmiştir.

Başarı değerleri aşağıdaki tabloda görüldüğü gibidir.

Tablo 3

Başarı Ölçütleri	**%**
Accuracy	82,97
Precision	77,18
Recall	87,41
F1	81,98



Grafik 9 – Karmaşıklık Matrisi

Yapay Sinir Ağları

Günümüzde yapay sinir ağları sınıflandırma problemlerini çözmek için yaygın olarak kullanılmaktadır. Bu projede de amaçlanan binary sınıflandırma olduğu için yapay sinir ağı modellerine çok uygun bir veri elimizde bulunmaktadır.

Elimizdeki metin verilerini Bag of Word yöntemini kullanarak 512 adet öznitelik çıkarılmıştı. Keras kütüphanesini kullanarak yapılması gereken matris işlemlerini GPU üzerinde yaparak işlemler kat kat hızlandırılabileceğinden 512 öznitelik 1500 değerine çıkarılmıştır. Bu şekilde anlamsız olabilecek öznitelikler de çıkarılmış olsa da bunları seçme işlemi yapay sinir ağlarına bırakılmıştır.

En uygun modelinin bulunması için bir formül olmadığı için tasarlanan yapay sinir ağı modellerine deneysel yaklaşılmıştır. Bu sebeple birden fazla model tasarlanıp, eğitilip, test edilmiştir.

Yapılan denemelerde optimizasyon algoritması olarak ‘adam’ algoritması kullanılmıştır. Çıkış katmanının neuron sayısı 1’e, aktivasyon fonksiyonu sigmoid olarak ve kayıp(loss) fonksiyonu ‘binary cross entropy’ olarak sabitlenmiştir.

Eğitim verisi modele verilmeden önce karıştırılarak algoritmanın aldığı her bir mini bach’in verinin tamamını temsil etmesi sağlanmıştır. Bu şekilde eğitim süreci hızlandırılmıştır.

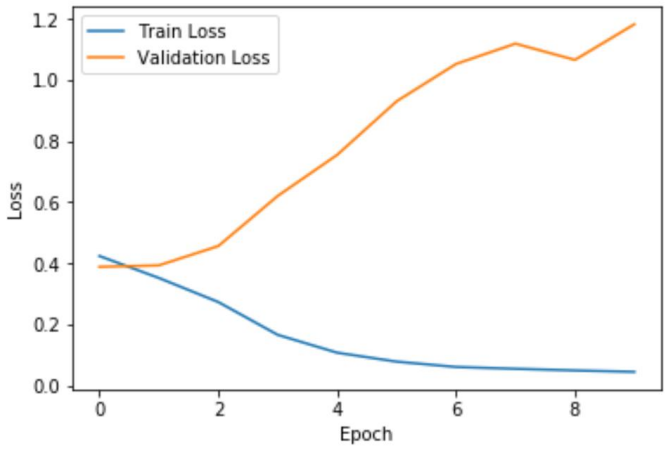
Eğitim verisinin %10’u validation veri seti olarak ayarlanarak modelin eğitim sırasındaki başarısı ölçülmüştür.

Yapay Sinir Ağı Modeli -1-

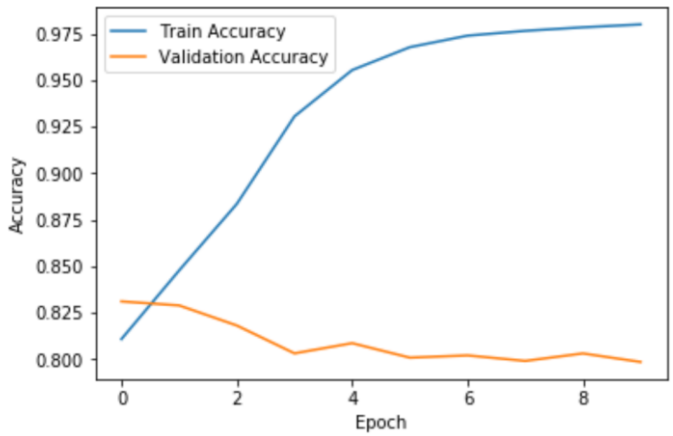
Layer (type)	Output Shape	Param #
dense_7 (Dense)	(None, 1024)	1537024
dense_8 (Dense)	(None, 512)	524800
dense_9 (Dense)	(None, 512)	262656
dense_10 (Dense)	(None, 128)	65664
dense_11 (Dense)	(None, 64)	8256
dense_12 (Dense)	(None, 1)	65

Total params: 2,398,465
Trainable params: 2,398,465
Non-trainable params: 0

Model 1



Grafik 10



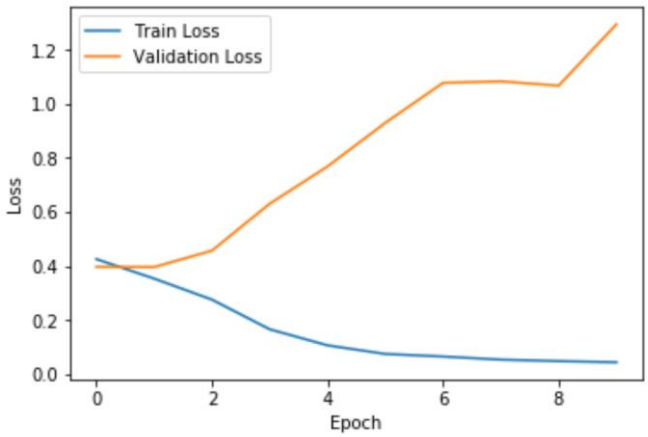
Grafik 11

Yapay Sinir Ağı Modeli -2-

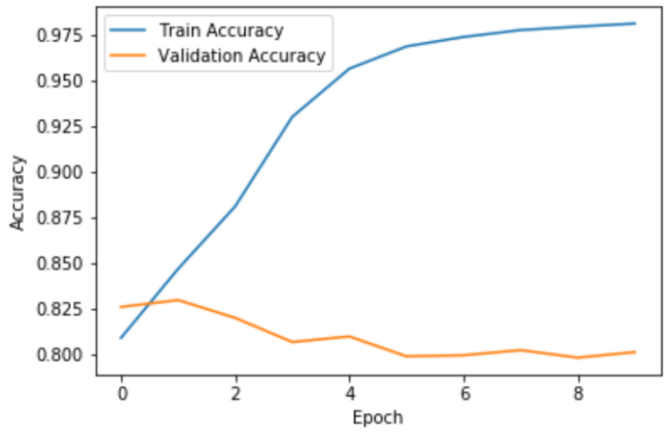
Layer (type)	Output Shape	Param #
dense_13 (Dense)	(None, 1024)	1537024
dense_14 (Dense)	(None, 1024)	1049600
dense_15 (Dense)	(None, 512)	524800
dense_16 (Dense)	(None, 512)	262656
dense_17 (Dense)	(None, 128)	65664
dense_18 (Dense)	(None, 64)	8256
dense_19 (Dense)	(None, 1)	65

Total params: 3,448,065
Trainable params: 3,448,065
Non-trainable params: 0

Model 2



Grafik 12



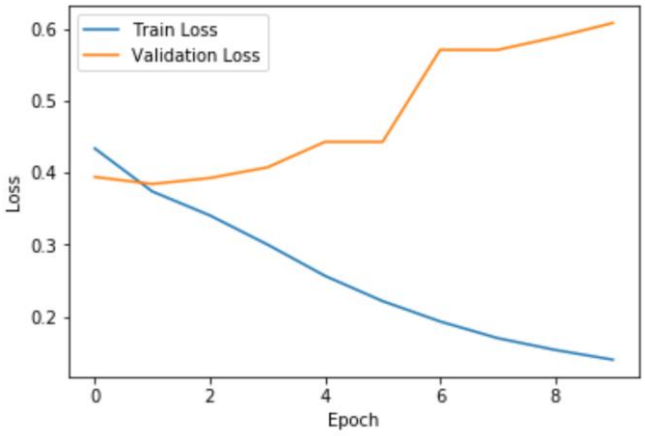
Grafik 13

Yapay Sinir Ağı Modeli -3-

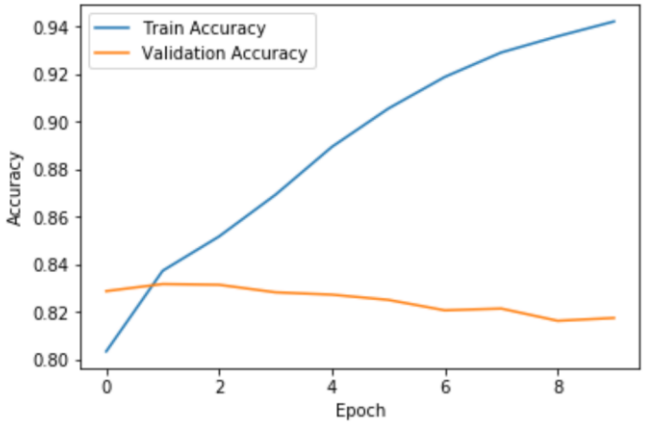
Layer (type)	Output Shape	Param #
dense_27 (Dense)	(None, 1024)	1537024
dropout_1 (Dropout)	(None, 1024)	0
dense_28 (Dense)	(None, 1024)	1049600
dense_29 (Dense)	(None, 512)	524800
dense_30 (Dense)	(None, 512)	262656
dense_31 (Dense)	(None, 128)	65664
dense_32 (Dense)	(None, 64)	8256
dense_33 (Dense)	(None, 1)	65

Total params: 3,448,065
Trainable params: 3,448,065
Non-trainable params: 0

Model 3



Grafik 14



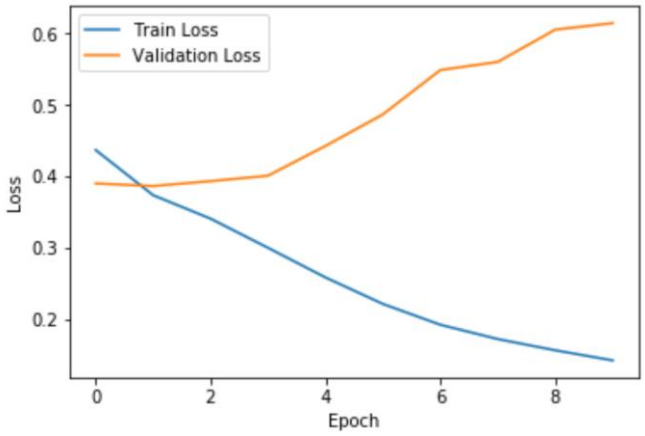
Grafik 15

Yapay Sinir Ağı Modeli -4-

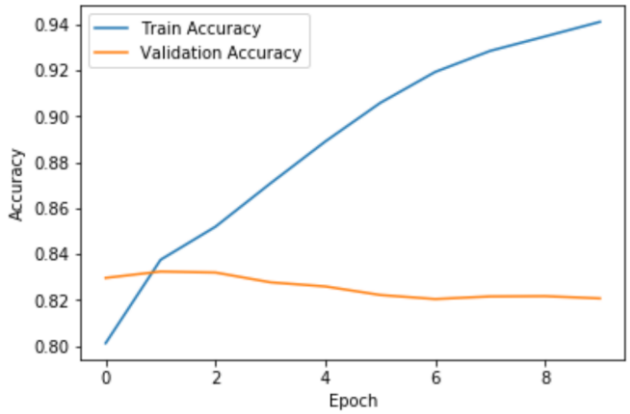
Layer (type)	Output Shape	Param #
dense_34 (Dense)	(None, 1024)	1537024
dropout_2 (Dropout)	(None, 1024)	0
dense_35 (Dense)	(None, 1024)	1049600
dense_36 (Dense)	(None, 512)	524800
dense_37 (Dense)	(None, 512)	262656
dropout_3 (Dropout)	(None, 512)	0
dense_38 (Dense)	(None, 128)	65664
dense_39 (Dense)	(None, 64)	8256
dense_40 (Dense)	(None, 1)	65

Total params: 3,448,065
Trainable params: 3,448,065
Non-trainable params: 0

Model 4



Grafik 16



Grafik 17

Yapay Sinir Ağı Modellerinin Başarı Değerleri Karşılaştırması

Tablo 4

Modeller	***Train Accuracy***	***Validation Accuracy***	***Test Accuracy***
Model 1	98.01	79.84	80.13
Model 2	98.08	80.12	80.00
Model 3	94.21	81.76	81.82
Model 4	94.11	82.07	82.03

Tablo-3'ten de görüleceği gibi train accuracy her zaman en yüksek noktada ve %100'e çok yakındır. Bunun sebebi sistemin aşırı öğrenmeye gitmesi, modelin veriyi ezberlemesidir. En yüksek elde edilen accuracy değeri 98.08 olsa da bu train accuracy'dir ve sistemin başarılı çalıştığını söylemek için doğru parametre bu değildir.

Dikkat edilmesi gereken parametre test accuracy değeridir. Bu değer hiçbir zaman eğitim verisine katılmadığı için sistemin nasıl sonuç verebileceğini en doğru söyleyen bu veri seti ile yapılmış testtir. Test Accuracy değerlerine bakıldığında %80-82 arasında kaldığı görülmektedir. XGBoost algoritması ile elde edilen başarıyı geçememiştir.

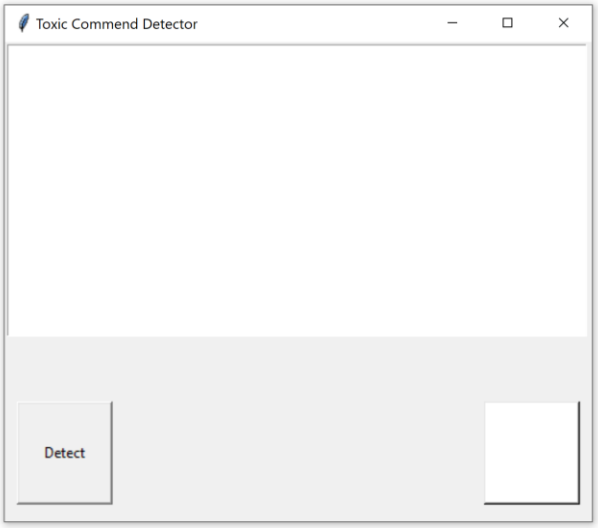
Kullanıcı Arayüzü

Kullanıcı arayüzü 3 bölümden oluşmaktadır. Bunlar,

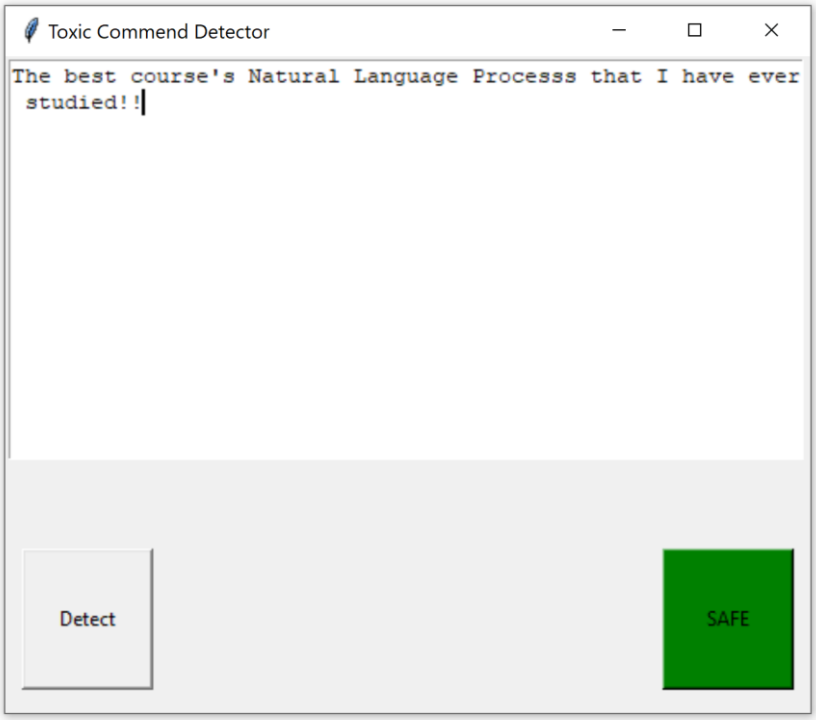
1. Kullanıcı metin girişi
2. ‘Detect’ butonu
3. Sonuç göstergesi

Kullanıcı metni girdikten sonra detect tuşuna basması ile işlem başlar. Veri temizlenir, sayısal özniteliklere çevrilir ve makine öğrenmesi modeline giriş olarak verilir. Modelin çıktısı 1 veya 0 olarak etiketlidir. 1 çıktısı ‘toxic’ 0 çıktısı ‘nontoxic’ anlamına gelmektedir. Sonuç göstergesi bu çıktıya göre rengini kırmızı yaparak ‘toxic’ veya yeşil yaparak ‘safe’ yazar.

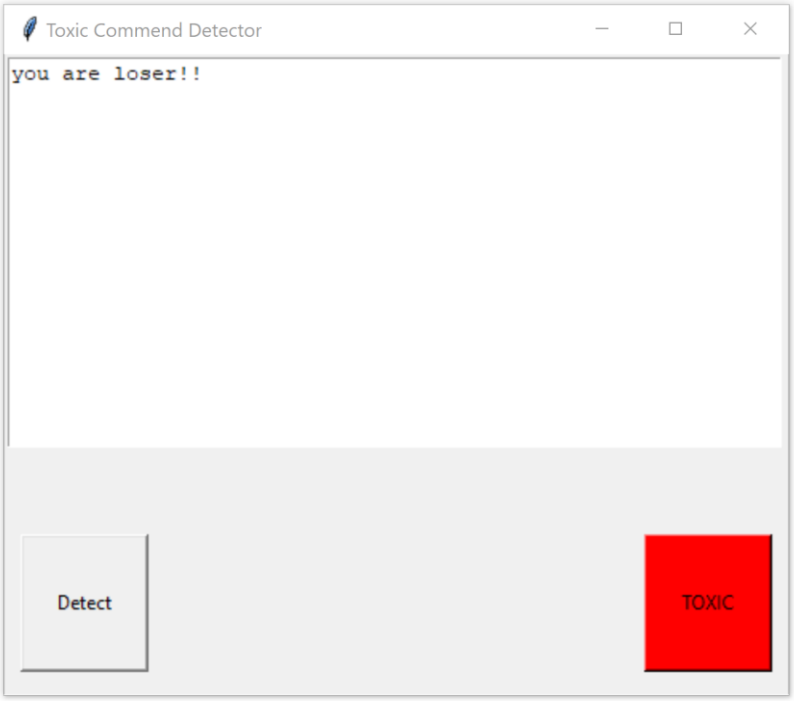
- Arayüz başlangıç ekranı:



- Nontoxic bir veri girişi yapılıp detect duşuna basıldığında:



- Nontoxic bir veri girişi yapılıp detect duşuna basıldığında:



Sonuç

Yapılan tüm denemeler sonucunda en uygun algoritmanın XGBoost algoritması olduğu görülmüştür ve bu algoritma üzerinden ilerlenmiştir.

Sklean kütüphanesinden metinleri algoritmalar için anlamlı özniteliklere dönüştüren 'CountVectorizer' modeli ve veriden anlam çıkarmak için kullanılan XGBoost makine öğrenmesi modelini kullanarak sonuçlandırılmıştır. Bu iki model de eldeki veri ile eğitilerek/hesaplanarak modellerin parametreleri kaydedilmiştir. Kaydedilen modeller kullanılarak da kullanıcıdan alınan metinler sınıflandırılmıştır. Gerçekleştirilen uygulamada metin işleme sırası şu şekildedir:

Tüm yapılan çalışmalar sonucunda eğitim verisine hiç eklenmemiş bir veri seti ile elde edilen sınıflandırma başarısı ***%82.97***'dir.

Kaynakça

[1] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview

[2] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/overview

[3] https://en.wikipedia.org/wiki/Bag-of-words_model

[4] https://en.wikipedia.org/wiki/Word_embedding

[5] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[6] https://xgboost.readthedocs.io/en/latest/