Logo

Description automatically generated

**ATATÜRK ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ**

**DERİN ÖĞRENME VE EVRİŞİMLİ SİNİR AĞLARI VİZE ÖDEVİ**

**Bilgisayar Mühendisliği– Yüksek Lisans**

**21040201052**

**İsa KULAKSIZ**

**Ham - Spam Ayrımı**

Gereksinimler:

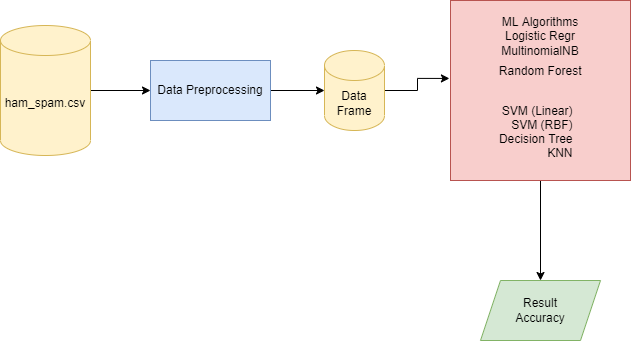
* Makine öğrenimi tekniklerinin uygulanması

1. Naive Bayes
2. Logistic Regression
3. SVM (Linear, RBF)
4. Multi-layer Perceptron Classification
5. LSTM

* K Fold Cross Validation uygulanması
* Tahmin değerlerinin hesaplanması (Accuracy, Recall, Precision vb.)
* ROC Eğrisinin gösterilmesi

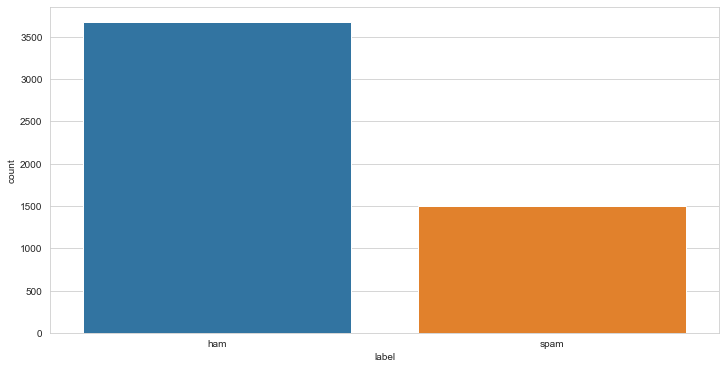
Senaryo: E-mail üzerinden gelen mesajlar üzerinde gerçek veya istenmeyen e-postaların tespit edilmesi.

Tablo 1’de kullanılacak yöntemler adım adım gösterilmiştir.

****

*Tablo 1 MLOPS*

Veri seti üzerinde “matpolotlib” kütüphanesi yardımıyla kaç adet ham veya spam olduğu Tablo 2’de gösterilmiştir.



*Tablo 2 Ham ve Spam Veriler*

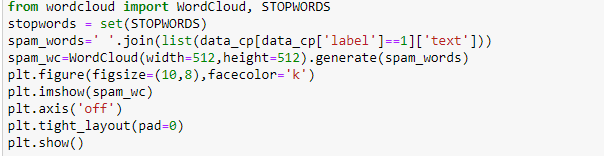
**Veri Ön İşleme (Data Preprocessing)**

Veri setinde varsayılan “label” öz niteliği “ham ve spam” olarak tanımlanmıştır. Burada “label” öz niteliğinin sayısal hale dönüştürme işlemi yapıldı.

Spam için 1 ham ise 0 olarak sayısal hale dönüştürüldü.

“WordCloud” kütüphanesini kullanarak spam maillerde en çok geçen kelimeler görselleştirildi.

Kaynak kod aşağıda yer almaktadır.



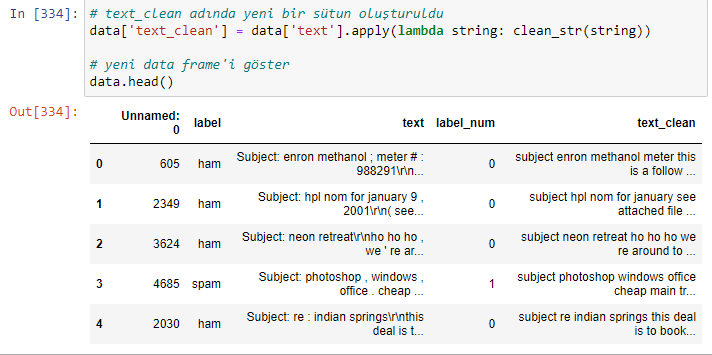
İlgili görsel Tablo 3’te yer almaktadır.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Tablo 3 Spam maillerde en çok tekarlanan kelimeler*

“Text” öz niteliğinde kelimelerin yanında sorun olabilecek pek çok karakterlere sahiptir. Bunun için noktalama işaretlerinin önemsiz kabul edildiği bir normalizasyon yapıldı.

Elde edilen yeni bulgular mevcut data frame üzerinde yeni bir öz nitelik oluşturularak Tablo 4’te gösterildi.

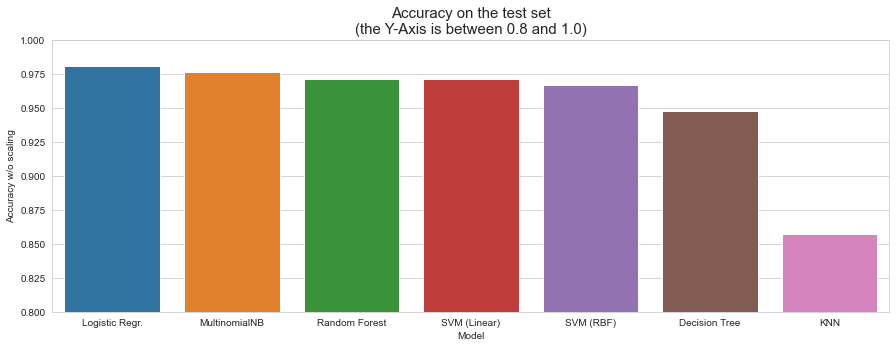
*Tablo 4 data frame’in ilk 5 elemanı*

Veri setinin %30’u test %70’i ise eğitim seti için ayrıldı.

Kullanılan Makine Öğrenimi algoritması ve tahmin skorları Tablo 5’te gösterilmiştir.

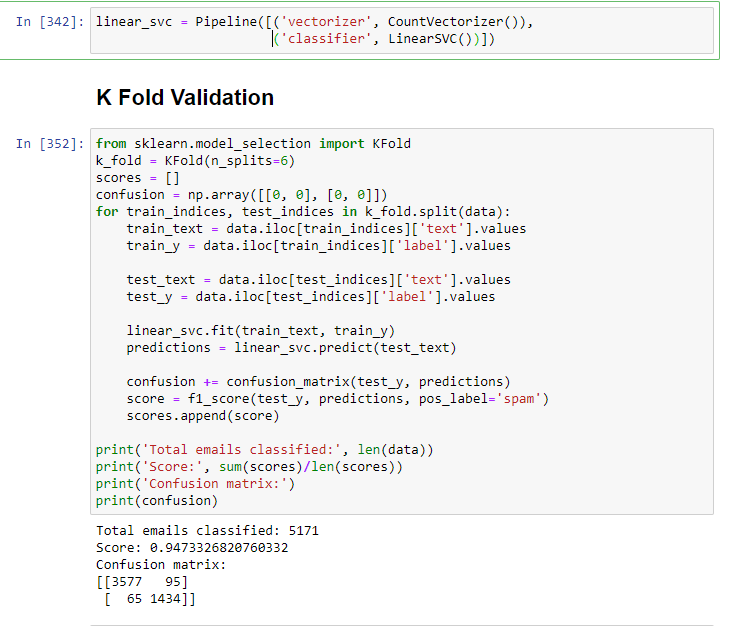
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MLOPS | Accuracy | Precision | Recall | F-measure |
| Logistic Reg | %98,07 | %98,06 | %98,06 | %98,06 |
| MultiNominal Naive Bayes | %97,61 |  |  |  |
| Random Forest | %97,10 |  |  |  |
| SVM (Linear) | %97,10 |  |  | %94,73 |
| SVM (RBF) | %96,71 |  |  |  |
| Decision Tree | %94,78 | %94,90 | %94,90 | %94,90 |
| KNN | %85,76 | %85,76 | %85,06 | %85,06 |

*Tablo 5 MLOPS sonuçları*



**K Fold Cross Validation**

Tanım: Elimizde çok az veri varsa test için ayırmanız halinde çok küçük bir veri seti ile karşı karşıya kalabiliriz. Bunun sonucu olarak da; test verileri hangi verinin test verisi hangi verinin eğitim olduğuna bağlı olarak değişecektir. Bu durumu önlemek içim eldeki veriyi K parçaya genelde (K=5 veya K=10) ayrılır. K tane aynı model oluşturulup her biri K-1 parça veriyle eğitilip kalanı da değerlendirmeye tabi tutulur.



**ROC Eğrisi ve Altında Kalan Alan**

Tanım: True Positive Rate (TPR) / False Positive Rate (FPR) esas olarak. Sınıflandırma problemlerinin iyi çalışıp çalışmadığını kontrol eden bir metriktir.

ROC Curve temelde TPR’in 1’e yakınsaması durumunun yanında FPR’nin alacağı değerleri gösterir.

**Chart, scatter chart

Description automatically generated**

**ANN (Yapay Sinir Ağları)**

**Diagram

Description automatically generated**

**Jupyter Notebook üzerinde yapay sinir ağları çalışırken “Kernel Died” problemi ile karşılaştım. Bu problem çoğu zaman kullanılan makinenin işlemci kapasitesinin yetersizliği ile alakalı bir problem olduğu için Google Colab IDE üzerinde çalışmalarımı yaptım.**

**Veri setinin üzerinde bölme işlemi yapılarak %30 test için geri kalan %70 ise eğitim seti için ayrıldı.**

**Text

Description automatically generated**

**Oluşturacağımız model için Hidden Layer eklendi. Burada verinin aşırı öğrenmesinden kaçınmak için “Dropout” kullanıldı. Model Özeti Tablo 6’da gösterilmiştir.**

**Overfitting (Aşırı Öğrenme) durumunu önlemenin diğer önemli bir yöntemi K Fold Cross Validation’dır.**

**Text

Description automatically generated**

***Tablo 6 Model Özeti***

Yapay sinir ağını eğitmek için kullandığımız batch\_size değerini 100 ve epochs değerini ise 30 olarak tanımlandı.

**Text

Description automatically generated**

Eğitilen modelin test veri üzerinde tahminlerinin gösterilmesi

Confusion Matrix:

[[1023, 78],[303, 148]]

**LSTM (Long Short Term Memory)**

**Graphical user interface, application

Description automatically generated**

1997 yılında Hochreiter Schmidhuber tarafından geliştirilmiştir. Çalışma SimpleRNN’in bir eksiği üzerinde durmuştur. Teorik olarak t anında önceki tüm adımlardaki girdilerle ilgili bilgiyi tutabilir ancak uygulamada bu kadar uzun dönemli gereksinimilerin öğrenilmesi imkânsız hale gelebilir.

SimpleRNN benzer bir versiyonudur. Birçok zaman diliminde bilgi taşınmasını sağlayan bir yol ekler. İşlemeye çalışılan bir taşıyıcı kayışı düşünelim. Dizideki bilgi herhangi bir noktadan taşıyıcı kayışa geçebilir böylece bir sonraki zaman adımına ulaştırılır ve tekrar ihtiyaç olması durumunda eski yerine geçebilir.

**Chart, line chart

Description automatically generated**

**Accuracy (Doğruluk): %94,94**

**Multi-Layer Perceptron Classifier**

**Temelde XOR probleminin Single Perceptron Model tarafından çözülememesinden ortaya çıkmıştır. XOR fonksiyonunda iki boyutlu düzlem üzerinde çıktıları tek bir çizgiyle iki ayrı parçaya bölünememektedir. Bunun için en az iki doğru gerekmektedir.**

**Graphical user interface

Description automatically generated**

**Veri setinde uygulandıktan sonra elde edilen Accuracy (Doğruluk ) %70,95 olduğu kanıtlandı.**

**Text

Description automatically generated**

**Kaynak kodlar versiyon kontrol sistemi (git) kullanılarak GitHub hesabım’da yer almaktadır.**

**Aşağıdaki link üzerinden erişilebilir.**

[**https://github.com/isakulaksiz/SpamEmail**](https://github.com/isakulaksiz/SpamEmail)

**İsa Kulaksız**