



# **YAZILIM TASARIM TANIMLAMA DOKÜMANI (SDD)**

**Derin Öğrenme İle Twitter Platformunda**

**Yalan Haber Tespiti**

**Cansu AYTEN – 171180010**

**Gamze AKSU – 171180005**

**BM495 BİLGİSAYAR PROJESİ I**

**OCAK 2022**

# İÇİNDEKİLER

Sayfa

İÇİNDEKİLER .....	i
KISALTMALAR .....	ii
1. KAPSAM.....	1
1.1 Tanım .....	1
1.2. Sisteme Genel Bakış .....	1
1.3. Dokümana Genel Bakış .....	2
2. İLGİLİ DOKÜMANLAR.....	2
3. SİSTEM MİMARİ TASARIMI .....	2
3.1. Model Oluşturma .....	2
3.2. Seçilen Model.....	3
3.3. Alternatif Modeller .....	3
4. BİLEŞENLERİN DETAYLI ANLATIMI .....	3
4.1. Veri Seti .....	3
4.2. Veri Ön İşleme .....	4
4.3. Vektörleştirme .....	5
4.4. Tokenizasyon.....	6
4.5. Kodlama .....	6
4.6. Genel Çalıştırma (Execution) Kavramı .....	6
5. KULLANICI ARA YÜZ TASARIMI.....	6
6. GEREKSİNİMLERİN İZLENEBİLİRLİĞİ .....	7
EKLER .....	9

## KISALTMALAR

Kısaltmalar	Açıklamalar
<b>CBOW</b>	The Continuous Bag Of Words
<b>GUI</b>	Graphical User Interface
<b>HTTP</b>	Hyper-Text Transfer Protocol
<b>IDE</b>	Integrated Development Environment
<b>LSTM</b>	Long Short-Term Memory
<b>MB</b>	Megabayt
<b>NLTK</b>	Natural Language Toolkit
<b>RELU</b>	Rectified Linear Unit
<b>SDD</b>	Software Design Description
<b>SPMP</b>	Software Project Management Plan
<b>SRS</b>	Software Requirement Specifications
<b>STD</b>	Software Test Description
<b>TPU</b>	Tensor Processing Unit

## **1 KAPSAM**

### **1.1 Tanım**

Twitter’da Gerçek-Sahte Haber Tespit Projesi, sürüm 1.0 (beta)

Teknoloji günden güne gelişirken hayatımıza kattığı kolaylıklar ve tehlikeler kaçınılmazdır. Teknolojinin faydalarından yalnızca ikisi olan İnternet ve akıllı cihazlar ile sosyal medya hayatın bir parçası haline gelmiştir. İnsanlar öğrendikleri bir bilgiyi ya da yaşadıkları bir olayı gerçek zamanlı olarak sosyal medya üzerinden paylaşabilmektedirler. Bu bilgiler kimi zaman doğru ya da yanlış olup olmadığına bakılmaksızın yayılır, kimi zaman gündem değiştirme amacıyla yanlış bilgiler haber olarak insanlara sunulur, kimi zamansa öğrenilen herhangi bir haber abartılıp değiştirilerek aktarılır. Bu bilgilerin hızla yayılması bilgi kirliliği yaratabilmektedir. Oluşan bu bilgi kirliliği dünya üzerinde bazen büyük etkilere sahip olabilir, insanlar arasında kargaşa yaratabilir. Bilgi kirliliğine neden olabilen sosyal medya platformlarından biri Twitter’dır. Twitter’da herkes kendine ait bir profile sahip olmakla birlikte istediği zaman 280 karakter uzunluğunda bir metni paylaşabilmektedir. Aynı konu başlıklarından çok fazla söz edilmesi bu konu başlığının Twitter’da Gündem içerisinde yer almasını sağlar. Twitter ülkelere, dünyaya ve kullanıcıya ait farklı gündem seçenekleri de sunmaktadır. Gündemde seçilen konu başlığına ait tweetler görüntülenebilir ve böylece doğru ya da yanlış bilgiler daha hızlı yayılabilir. Bunu önlemek amacı ile sosyal medya platformlarında gerçek ya da sahte bilgileri ayırt edebilecek bir sisteme ihtiyaç duyulmaktadır. Yazılım Tasarım Tanımlama Dokümanı içerisinde bu soruna alternatif bir çözüm olarak geliştirilen derin öğrenme ile Twitter platformunda yalan haber tespiti konulu projeye ait tasarımlar belgelenecektir.

### **1.2 Sisteme Genel Bakış**

Twitter’da Gerçek-Sahte Haber Tespit Projesi içerisinde doğal dil işleme teknikleri ve derin öğrenme teknikleri birlikte kullanılarak haber içerikli bir tweetin gerçek ya da sahte olup olmadığı en doğru şekilde tahmin edilmeye çalışılacaktır. Bu projede basit ve kullanıcı dostu bir ara yüz geliştirilecek ve bu ara yüz ile kullanıcıdan alınan en fazla 280 karakter uzunluğunda olan tweetin gerçek ya da sahte olup olmadığı sonucu kullanıcıya döndürülecektir.

Twitter’da Gerçek-Sahte Haber Tespit Projesi Gazi Üniversitesi Mühendislik Fakültesi'nin Bilgisayar Mühendisliği bölümünde 4. sınıfta öğrenim gören iki kişilik bir öğrenci grubu tarafından gerçekleştirilecektir. Projenin dokümantasyon takvimi şu şekildedir:

1. Yazılım Proje Yönetim Planı (SPMP): 19 Kasım 2021
2. Yazılım Gereksinim Belirtim Dokümanı (SRS): 10 Aralık 2021
3. Yazılım Tasarım Tanımlama Dokümanı (SDD): 14 Ocak 2022
4. Yazılım Test Tanımlama Dokümanı (STD): 21 Ocak 2022
5. Proje Rapor Teslimi ve Sunumu: 28 Ocak 2022

### 1.3 Dokümana Genel Bakış

Bu Yazılım Tasarım Dokümanı (SDD -Software Design Description) tasarımının detaylı açıklanmasıdır. Burada her bir yazılım biriminin açıklaması yapılır. SRS (Software Requirement Specification) raporundaki her bir gereksinim için nasıl bir yol izlenmesi gerektiğine dair bilgiler içermektedir. Ayrıntılı bilgilerin daha açıklayıcı ve anlaşılır olması için use case, veri akış gibi bazı diyagramlardan yararlanılmıştır.

## 2. İLGİLİ DOKÜMANLAR

1. Word2vec. (2022). *Wikipedia*  
<https://en.wikipedia.org/wiki/Word2vec>
2. Keskin, M. (2018). Word2Vec, FastText, GloVe  
<https://medium.com/codable/word2vec-fasttext-glove-d4402fa8cce0>
3. Zubiaga, A. Liakata, M. Kochkina, E. (2018). PHEME dataset for Rumour Detection and Veracity Classification  
[https://figshare.com/articles/dataset/PHEME\\_dataset\\_for\\_Rumour\\_Detection\\_and\\_Veracity\\_Classification/6392078](https://figshare.com/articles/dataset/PHEME_dataset_for_Rumour_Detection_and_Veracity_Classification/6392078)

## 3. SİSTEM MİMARİ TASARIMI

### 3.1. Model Oluşturma

Model oluşturma Ek-A'da use case diyagramı ile gösterilmiştir. Model oluşturulurken Geliştirici olarak adlandırılan bir çubuk aktör vardır. Geliştirici Ham veri deposundan aldığı verileri Düzenli Veri oluşturma işleminden geçirdikten sonra Veri deposuna kaydeder. Düzenlenmiş .csv formundaki veriye büyük-küçük harflerin dönüştürülmesi, noktalama işaretlerinin kaldırılması, stopword ve http ile başlayan linklerin kaldırılması ve son olarak

kelimeleri köklerine dönüştürme işlemi gibi ön işlemlerden geçirerek veri deposuna kaydedilir. Daha sonra geliştirici verileri tokenizasyon işlemi uygular. Bu işlemden sonra veriler modele verilebilecek şekilde sayısal olarak dönüştürülmüş olur. Geliştirici Model Oluşturma Aşamasına verilerle eğitilecek olan modeli oluşturur. Geliştirici modeli eğitmek için ve modeli test etmek için veriyi ikiye böler. Eğitim verisi ile Model Eğitim işlemi gerçekleştirilir. Model Değerlendirme işleminde ise test verisi ile test edilen modeller değerlendirilip aralarından birisi seçilir.

### 3.2. Seçilen Model

Proje kapsamında 8 farklı model oluşturulmuştur. Modeller arasında 3. Model accuracy değeri (0.86) daha fazla olduğu için bu model seçilmiştir. Model içerisinde Embedding katmanı içerisinde Word2Vec kullanılmıştır. Embedding katmanı içerisinde çıkış shape'i (22,100) dür. Parametre sayısı 693300 olmuştur. Embedding katmanı sonrasında 0.5 değeri ile Dropout katmanı eklenmiştir. Daha sonra çıkış shape'i Flatten ile 2200 olarak düzeltilmiştir. Daha sonra Dense, Dropout, Dense, Dropout ve Dense olarak devam edilmiştir. Dense katmanlarında aktivasyon fonksiyonu olarak relu kullanılmıştır. Son Dense katmanında sigmoid aktivasyon fonksiyonu kullanılmıştır.

### 3.3. Alternatif Modeller

Proje kapsamında 8 farklı model oluşturulmuştur. Modeller oluşturulurken Bidirectional LSTM, LSTM, Embedding (Word2Vec, FastText), Dropout, Flatten, Dense gibi yöntemlerden yararlanılmıştır. 1. ve 2. Modelde Bidirectional LSTM kullanılmıştır. 1. Model içerisinde Embedding katmanında word2vec kullanılırken 2. Model içerisinde fasttext kullanılmıştır. 4. Model ise 3. Model ile benzer katmanlara sahiptir. Aralarındaki fark ise Embedding katmanında Fasttext kullanılmasıdır. 5. ve 6. Model içerisinde LSTM kullanılmıştır. Embedding katmanında 5. Modelde word2vec kullanılırken 6. Modelde fasttext kullanılmıştır. 7. Modelde ise CountVectorizer kullanılmıştır. Son olarak 8. Modelde ise HashingVectorizer kullanılmıştır.

## 4. BİLEŞENLERİN DETAYLI ANLATIMI

### 4.1 Veri Seti

Proje içerisinde PHEME veri seti kullanılmıştır [3]. Bu veri setinin içerisinde dokuz haber başlığına ait türü sahte, gerçek veya belirsiz olan toplamda 6425 kaynak tweet bulunmaktadır. Bu haber başlıkları; Charlie Hebdo, Ebola Essien, Ferguson, Germanwings Crash, Gurlitt,

Ottawa Shooting, Prince Toronto, Putin Missing ve Sydney Siege'dir. Bulundurdukları gerçek, sahte ve tweet sayıları aşağıdaki gibidir:

- Charlie Hebdo: 1621 gerçek, 458 sahte tweet
- Ebola Essien: 0 gerçek, 14 sahte tweet
- Ferguson: 859 gerçek, 284 sahte tweet
- Germanwings Crash: 230 gerçek, 238 sahte, 1 belirsiz tweet
- Gurlitt: 77 gerçek, 61 sahte tweet
- Ottawa Shooting: 420 gerçek, 470 sahte tweet
- Prince Toronto: 4 gerçek, 229 sahte tweet
- Putin Missing: 112 gerçek, 126 sahte tweet
- Sydney Siege: 699 gerçek, 522 sahte tweet

Her bir başlık ayrı dizinlerde tutulmaktadır. Veriler .json dosyaları olarak depolanmıştır. Her tweet için bir tweet id bulunmaktadır. Tweet'in kendisi "source-tweet" dizininde bulunur. Tepkiler yani "reactions" dizininde ise source tweet'i yanıtlayan kişilerin tweetleri bulunmaktadır. Proje içerisinde reactions dizininde bulunan tweetler kullanılmamıştır. Source tweetin sahte veya gerçek olup olmadığı bilgisi ise "annotation.json" dosyasında bulunmaktadır. Proje kapsamında bu veriler Jupyter Notebook uygulaması üzerinden Python programlama dili kullanılarak işlenmiş ve daha sonrasında işlenen bu veriler bir .csv dosyasına dönüştürülmüştür.

## 4.2 Veri Ön İşleme

Twitter'da Gerçek-Sahte Haber Tespit Projesi için kullanılan veri setindeki tweetlere bir takım ön işlemler uygulanmıştır. İlk olarak tweetler içerisindeki noktalama işaretleri kaldırılmıştır. Bunun nedeni noktalama işaretleri kelimelere dâhil değildir. Dâhil olması durumunda verilerde gürültü meydana gelmektedir. Bundan sonra tokenize işlemi gelir. Tokenize işlemi ile cümleler kelimelerine ayrılarak bir liste biçiminde tutulur. Böylelikle kelimelere ulaşım kolaylaştırılmıştır ve bu şekilde ileride kelime üzerine uygulanacak işlemlerde de kolaylık sağlanmaktadır. Daha sonrasında verideki tüm büyük harfler küçük harfe dönüştürülmüştür. Bunun nedeni ise aynı kelimenin büyük harfle başlayan ve başlamayanı birbirinden farklı değildir ve farklı kabul edilmemelidir. Bundan sonrasında ise etkisiz kelimeler (stopwords) kaldırılmıştır. Böylelikle metin içerisinde daha önemli kelimeler kalır. Daha az kelime kalmasıyla birlikte modelin eğitim süresi de kısalmıştır. Sonrasında tweetlerin içerisinde bulunan

linklerin bir kelime olarak düşünülmemesi için tweet içerisinde linklerin kaldırılması işlemi yapılmıştır. Son olarak stemming işlemi yapılmıştır. Bu işlem kelimelerin bazı kısımlarını çıkarmaktadır. Böylelikle bazı kelimeler için aynı kelimeler elde edileceğinden verideki benzersiz kelime sayısı azalmaktadır. Proje içerisinde Snowball stemmer kullanılmıştır. Bu işlemlerin yapılabilmesi için nltk ve string kütüphanelerinden yararlanılmıştır.

### 4.3 Vektörleştirme

Twitter’da Gerçek-Sahte Haber Tespit Projesi kapsamında birçok model oluşturulmuştur. Bu modellerden biri oluşturulurken Word2Vec kelime temsil yönteminden faydalanılmıştır. Bu teknik 2013 yılında Google tarafından geliştirilmiştir [1]. Word2Vec’in iki çeşit alt yöntemi bulunmaktadır. Bu yöntemler CBOW ve Skip-Gram’dır. Bu yöntemler girdi ve çıktı alışları açısından birbirlerinden ayrılmaktadır. İki yöntemde de pencere boyutu (window size) belirlenir. CBOW modelinde pencere boyutunun merkezinde bulunmayan kelimeler girdi, Skip-Gram modelinde ise merkezinde bulunan kelimeler girdi olarak alınmaktadır. Çıktı olarak ise CBOW modelinde pencere boyutunun merkezinde olanlar tahmin edilmeye çalışılırken Skip-Gram modelinde ise merkezinde olmayanlar tahmin edilmeye çalışılır. Bu işlemler bütün cümlelere uygulanmaktadır. İşlemlerin sonunda veri eğitime hazır duruma getirilmektedir. CBOW yöntemi genel olarak büyük veri kümelerinde, Skip-Gram ise daha küçük veri kümelerinde iyi çalışmaktadır. Kısacası yöntemler birbirlerinin tam tersidir. Bu yöntemlerin kullanılabilmesi için Gensim kütüphanesi proje içerisinde kullanılmıştır.

Diğer bir model oluşturulurken FastText tekniğinden yararlanılmıştır. Bu teknik 2016 yılında Facebook tarafından geliştirilmiştir. Word2Vec’in uzantısıdır. Kelime kelime işlem yapmaktansa kelimeleri birkaç harften oluşacak şekilde böler. Bölerken “n-gram” yöntemi kullanılmaktadır [2]. Bu yöntemde n harf sayısını temsil etmektedir.  $n=2$  ise bigram adı verilmekte ve örneğin ağaç kelimesi ağ, ğa, aç şeklinde parçalanmaktadır. Adından da anlaşılacağı üzere hızlı çalışmaktadır. Bu yöntemin kullanılması için de Gensim kütüphanesine ihtiyaç duyulmaktadır.

Diğer bir model oluşturulurken CountVectorizer yöntemi kullanılmıştır. Bu yöntemde metin sayısal olarak ifade edilmektedir. Bu metnin içerisinde geçen kelimelerin cümle içindeki geçme sayısı kullanılarak bir matris oluşturulur. Matriste her sütun birbirinden farklı bir kelimeyi temsil etmektedir. Projede ise matristeki her satır bir tweete karşılık gelmektedir. Yani matrisin boyutu veride bulunan benzersiz kelime sayısı ile verideki tweet sayısı kadardır. Bu da verinin



büyük olması durumunda çok fazla yer kaplayabileceği anlamına gelmektedir. Bu yöntemin kullanılabilmesi için scikit-learn kütüphanesine ihtiyaç duyulmaktadır.

Diğer bir model oluşturulurken HashingVectorizer yöntemi kullanılmıştır. Bu yöntem CountVectorizer yöntemine benzerdir. Farklarından birincisi metnin içerisinde geçen kelimelerin cümle içinde geçme sayısına bir hash fonksiyonunun uygulanmasıdır. Diğer ise benzersiz kelimeleri bellekte tutmamasıdır. Bu yöntemin de kullanılabilmesi için scikit-learn kütüphanesine ihtiyaç duyulmaktadır.

Bu işlemlerin veriye ayrı ayrı uygulanmasının ardından elde edilen matrisler oluşturulan modelleri eğitmek için kullanılmaktadır.

#### **4.4 Tokenizasyon**

Word2Vec yapısından sonra tokenize kütüphanesi ile veri içerisindeki her kelime farklı bir sayı ile temsil edilecek hale getirilir. Daha sonra her tweetin içerisindeki kelime sayısı aynı olmadığından her tweet içerisindeki kelime sayısını eşitlemek için padding işlemi yapılır. Burada tweetler içerisindeki en büyük uzunluğa sahip tweetin kelime sayısı bulunarak maksimum uzunluk olarak padding işlemine verilir ve tüm tweetlerin uzunluğu kelimelerin sonuna 0 koyularak aynı uzunluğa getirilir. Daha sonra padding yapılmış veri word2vec ve FastText modelleri ile ağırlıklı matrise dönüştürülür. Daha sonra bu ağırlıklı matris derin öğrenme modelinde Embedding katmanında kullanılmıştır.

#### **4.5. Kodlama**

Projenin kodlama işlemleri Google Colab üzerinden gerçekleştirilmiştir. Çalışma zamanı türü olarak TPU seçilmiştir. Gerekli olan pandas, numpy, matplotlib, seaborn, wordcloud, tensorflow, keras, nltk kütüphaneleri kullanılmıştır. Ara yüz gerçekleştirimi işlemleri Spyder IDE'si üzerinden yapılmıştır. Ara yüz oluşturulurken tkinter kütüphanesinden yararlanılmıştır.

#### **4.6. Genel Çalıştırma (Execution) Kavramı**

Uygulamanın genel çalışması ile ilgili veri akış diyagramları Ek-B'de gösterilmiştir

### **5. KULLANICI ARA YÜZ TASARIMI**

Ara yüzün kullanıcı dostu olması amacıyla ara yüz içerisinde çok fazla karmaşık işleme yer verilmemiştir. Kullanıcıların tweet girebileceği bir metin kutusu bulunmaktadır. Metin kutusu içeriği 280 karakterle sınırlandırılmıştır. Tahmin butonuna basıldığında bir Message Box ile

girilen tweetin sahte ya da gerçek olduğu bilgisi verilmektedir. Sayfayı Temizle butonuna tıklandığında ise metin kutusu içerisine yazılan tweet silinmektedir.



Şekil 1: GUI tasarımı

Ara yüz için giriş ve çıkışlar Ek-C'deki use case diyagramında gösterilmektedir. Diyagramda Kullanıcı ve Geliştirici olarak adlandırılan iki çubuk aktör bulunmaktadır. Kullanıcı ilk önce ara yüz içerisinde Tweet Girme işlemi yapar. Girilen tweeti Geliştirici modelin tweeti sahte ya da gerçek olarak değerlendirebilmesi için noktalama işaretlerinin kaldırılması, büyük küçük harf dönüşümü, stopword ve linklerin kaldırılması ve kelimelere köklerine dönüştürme gibi bazı ön işlemlerden geçirir. İşlenmiş veri modele tahmin etmesi için verilir. Model tweetin sahte ya da gerçek olarak tahmin eder. Bu tahmin sonucu ara yüzden kullanıcıya gösterilir.

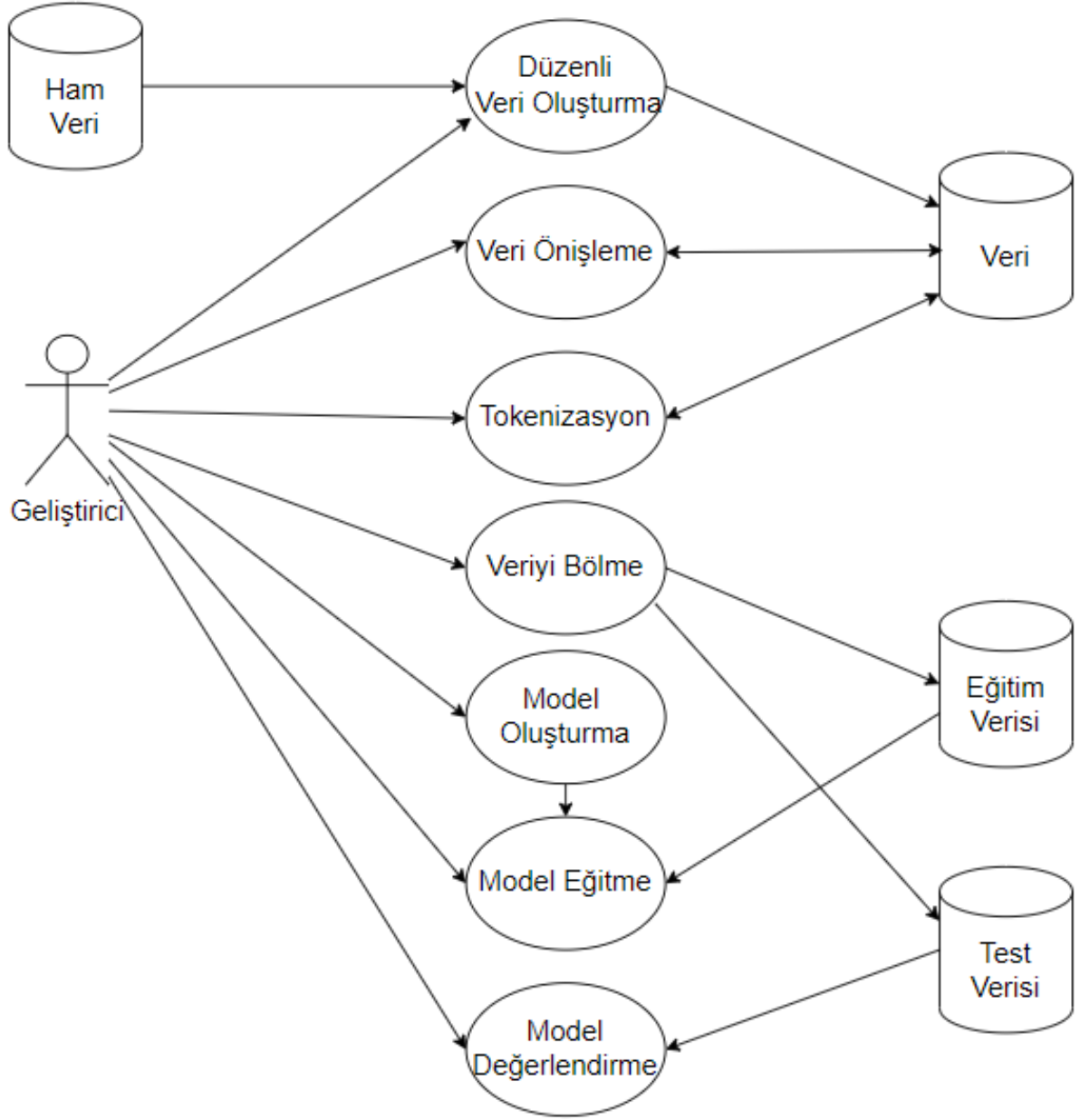
## 6. GEREKSİNİMLERİN İZLENEBİLİRLİĞİ

ID	SRS Başlık No	Gereksinim	SDD Başlık No
1	3.2.1.1	Haber içerikli tweet metin kutusu ile kullanıcıdan alınmalıdır.	5

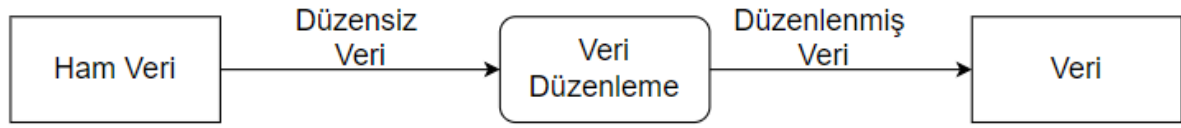
2	3.2.1.2	Alınan haber içerikli tweetin uzunluğu en fazla 280 karakter olmalıdır.	5
3	3.2.1.3	Kullanıcı Tahmin butonuna tıkladıktan sonra girilen tweetin gerçek veya sahte olduğu eğitilmiş model kullanılarak tespit edilmelidir.	5
4	3.2.1.4	Yapılan çıkarım kullanıcıya en fazla 1 saniye içerisinde gösterilmelidir.	Uygulanmadı
5	3.2.1.5	Yapılan çıkarımın doğruluğu en az %85 orana sahip olmalıdır.	3.2
6	3.2.1.6	Kullanıcı yeni bir tweet girmek istediğinde Sayfayı Temizle butonuna tıklaması sonucu sayfa ilk haline dönmelidir.	5
7	3.5	Veri seti gereksinimi	4.1
8	3.10.3.1 3.12.1	Verilere dair işlemler Google Colab üzerinden gerçekleştirilmelidir ve TPU desteği kullanılmalıdır	5
9	3.10.3.2	Ara yüz gerçekleştirimi Spyder IDE'si üzerinden yapılmalıdır.	5
10	3.10.3.3	Verilere dair işlemler yapılırken Pandas, Numpy, Matplotlib, Seaborn, Wordcloud, Tensorflow Keras, NLTK ve Re gibi kütüphaneler kullanılacaktır.	5
11	3.10.3.4	Ara yüz gerçekleştirilirken üstteki bazı kütüphanelerle birlikte Tkinter kütüphanesi kullanılmalıdır.	5
12	3.14.1 3.14.2 3.14.3 3.14.4 3.14.5	Veri önışleme işlemlerinin yapılması	4.1

## EKLER

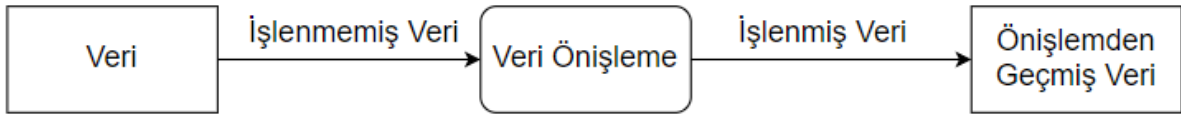
### Ek-A



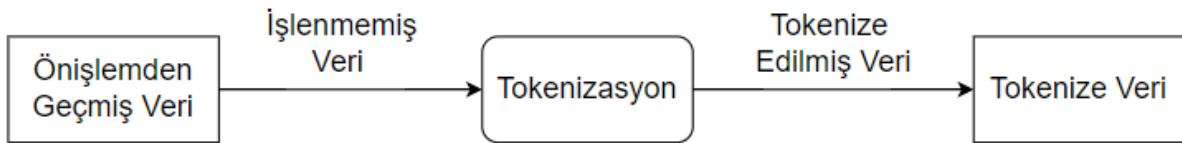
Diyagram 1: Model Oluşturma Use Case Diyagramı

**Ek-B**

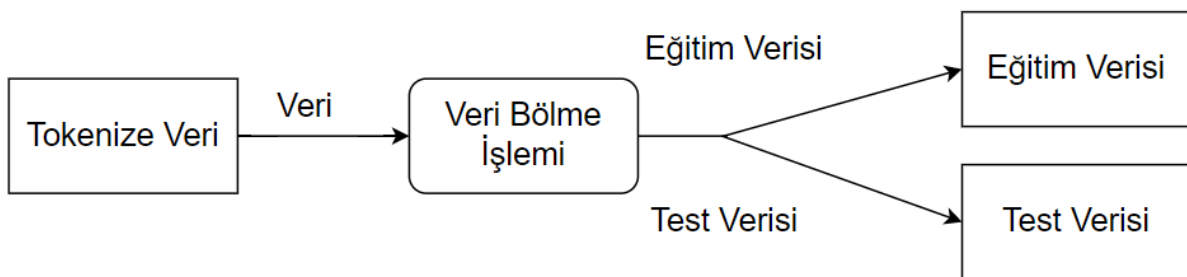
Diyagram 3: Veri Düzenleme Veri Akış Diyagramı



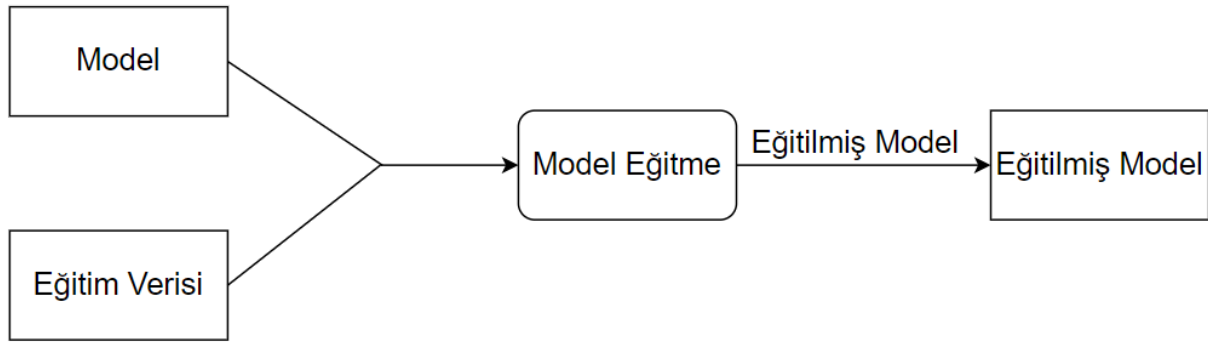
Diyagram 4: Veri Önışleme Veri Akış Diyagramı



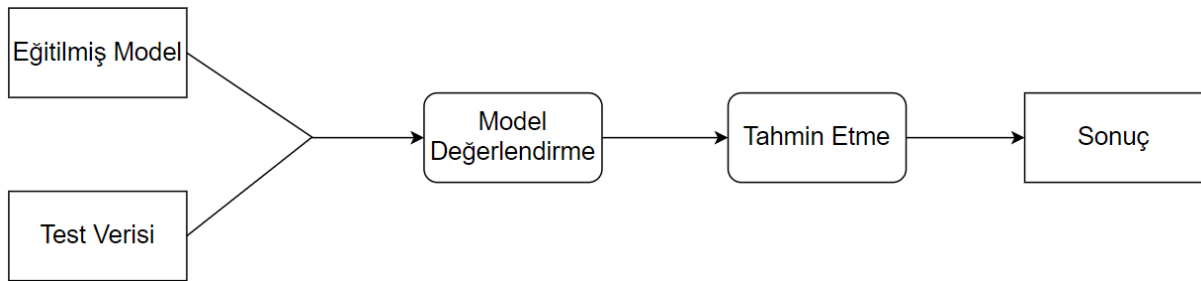
Diyagram 5: Tokenizasyon Veri Akış Diyagramı



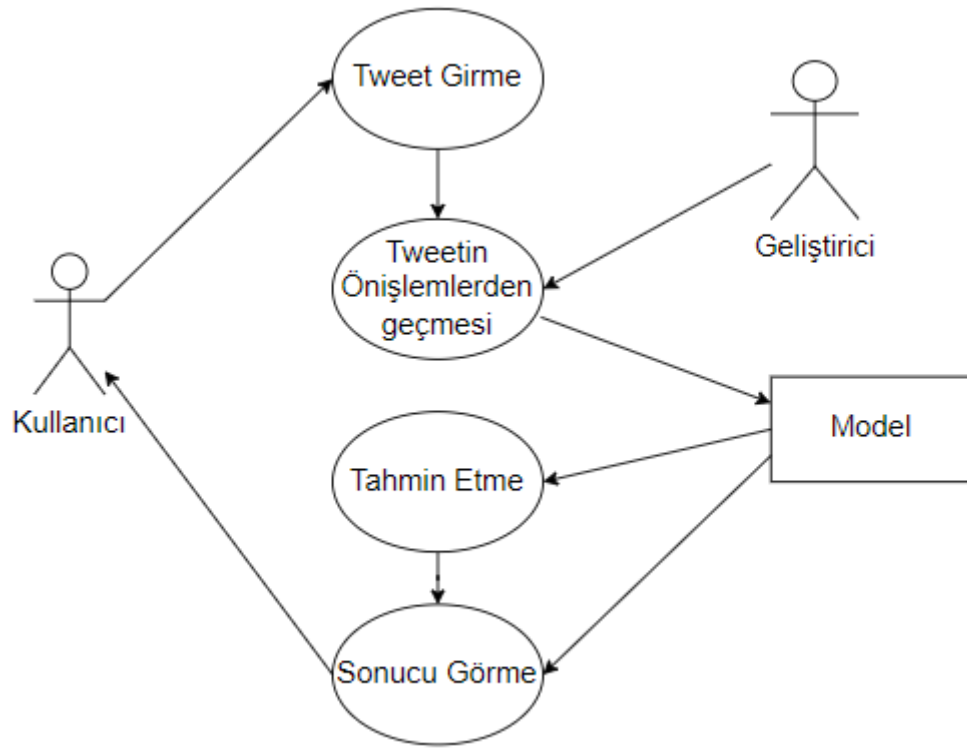
Diyagram 6: Veri Bölme İşlemi Veri Akış Diyagramı



Diyagram 7: Model Eğitim Veri Akış Diyagramı



Diyagram 8: Model Değerlendirme Veri Akış Diyagramı

**Ek-C**

Diyagram 2: Ara yüz için Use Case Diyagramı