**Konu:**

Sınıflandırma ve bir tane dil modeli kullanılmıştır. Çalışma mantıkları sınıflandırma modelleri kullanıcının yazdığı metinden yola çıkarak konuya uygun film bulmaya çalışır. Dil modeli yine aynı şekilde kullanıcıdan alınan metinden yola çıkarak ve veri setinin içindeki konu kısmını kullanarak belirttiğimiz karakter sayısında bize bir cümle oluşturur.

**Kullanılan kütüphaneler:**

Verdiğiniz kod örnekleri aşağıdaki kütüphaneleri ve modülleri kullanmaktadır:

1.pandas

2.TfidfVectorizer

3.linear\_kernel

4.spacy

5.text\_lemmatizer (simplemma)

6.numpy

7.ast

8.re

9.stopwords (nltk)

10.word\_tokenize (nltk)

11.train\_test\_split (sklearn)

12.MultinomialNB (sklearn)

13.svm (sklearn)

14.classification\_report (sklearn)

15.accuracy\_score (sklearn)

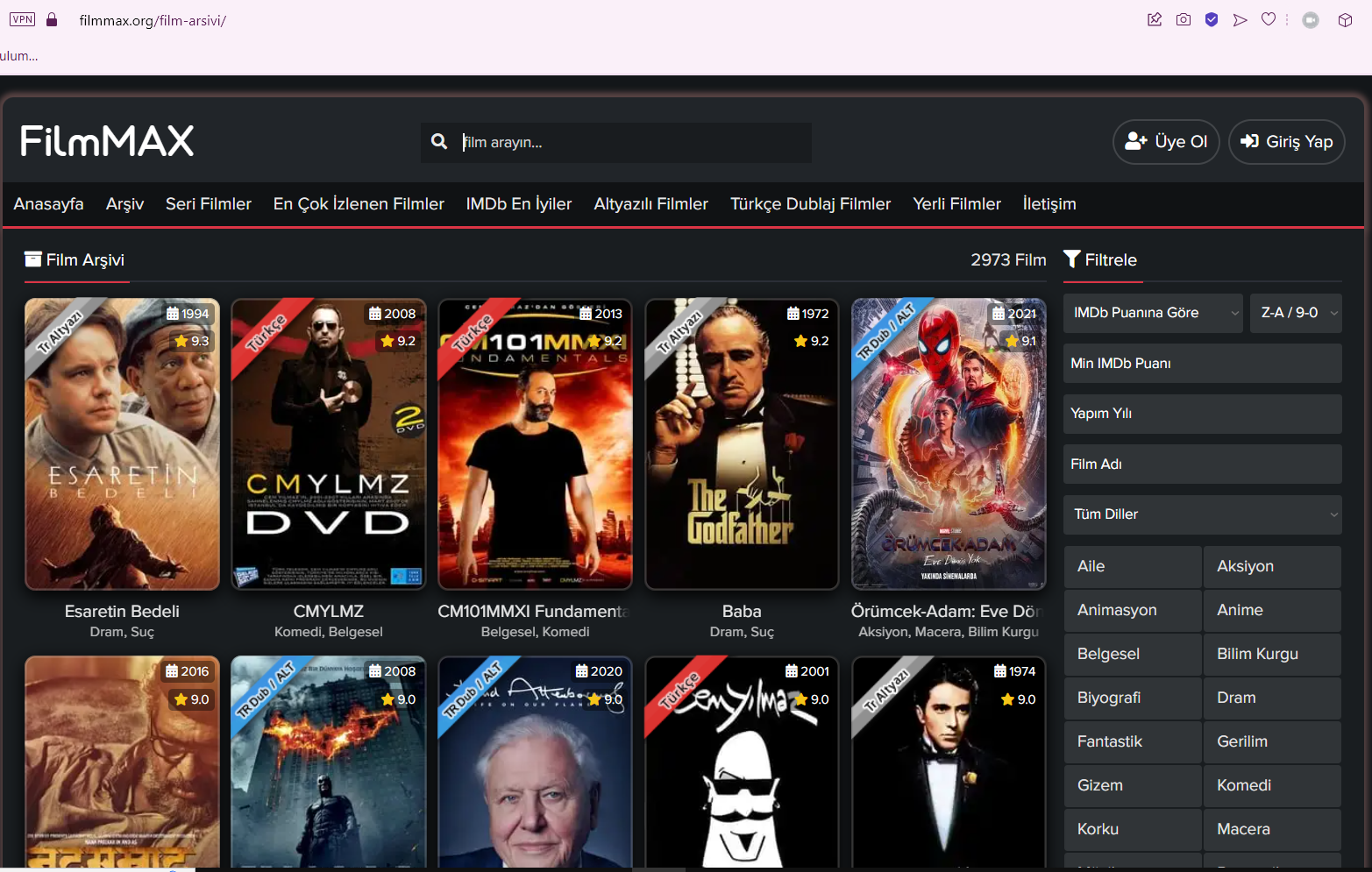
16.SimpleImputer (sklearn)

**VERİ SETİ'NİN OLUŞTURULMASI**

Veri seti film sitesinden çekilen film adı, türü ve konusundan oluşmaktadır. Veri setindeki verileri elde edebilmek için öncelikle siteden filmlerin linklerini çektik ve daha sonra o linklere girerek filmlerin adını, türünü ve konusunu çekerek veri setimizi oluşturduk. Sitenin film arşivinde buluna bütün filmlerin verilerini çektik çünkü ve veri setimizin boyutu arttıkça modellerimizin daha başarılı olacağını varsaydık. Veri setimizde 2913 filmin verisi bulunuyor ve bir film birden fazla türe sahip olabiliyor.

Veri setindeki sütunlar aşağıdaki gibidir:

|  |  |  |
| --- | --- | --- |
| **Film Adı** | **Türler** | **Konu** |



# Lemmatization İşlemi

Lemmatizasyon, kelimenin kökünü bulma işlemidir. Örneğin, "koştu" kelimesini "koş" olarak ele almak gibi. Bu, modelin daha iyi öğrenmesine yardımcı olabilir.Simplemma kütüphanesi Python fonksiyonunda bir metin dizgisini temizleme ve lemmatizasyon işlemlerini uygulama amacıyla tasarlanmıştır. Türkçe lemmatizasyon işlemi uygulanır. Bu işlem, kelimeleri temel veya kök formlarına indirgeme sürecidir.

from simplemma import text\_lemmatizer

kodu ile konu isim türler içerisinde emojiler, noktalama işaretleri, stopwordsler, linkler gibi istenmeyen ve modelin başarısını düşürecek veriler tweetlerin içerisinden temizleniyor.

def clean\_text(text)

def clean\_row(row):

simplemma ile temizlenen konu isim ve türler sorgu için hazır hale getirilmiştir.



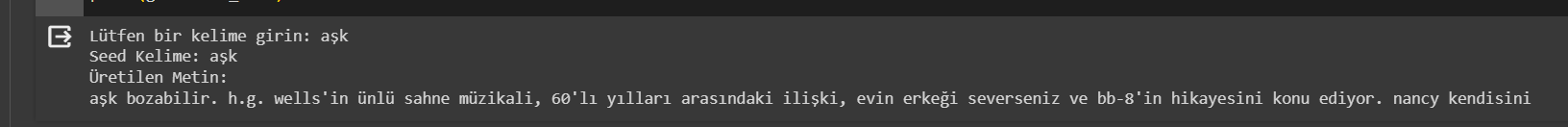
# Modelin Oluşturulması ve Filmlerin Kategorilendirilmesi

# Model Seçimi ve Model Oluşturma:

Projemizde 3 farklı tipte model kullandık. Bunlar bir metin girdiğimizde bize metne göre veri setinden istenilen karakter de cümle yazan Markov modeli ,

Cosine Similarity temelli bir öneri sistemi, benzer filmleri sıralayan Rondom Forest ve Support Machine modeli. Örnek gösterimler aşağıdaki gibidir:

Markov Modeli



### Cosine Similarity

metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

### Random Forest ve SVM Modelİ

metin, ekran görüntüsü, menü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, siyah, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, menü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

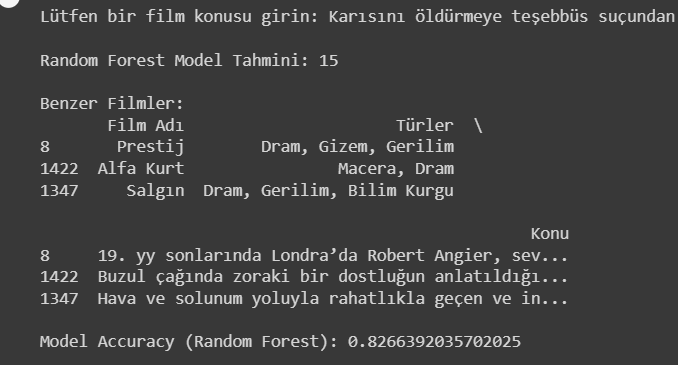
### **Filmlerin Vektörel Matrisinin Çıkarılması**

Filmler metinden oluştuğu için bunun bilgisayar ortamında işlenmesi mümkün değildir bu yüzden veriler sayısal değerlere dönüştürülmelidir. Bir sözlük oluşturularak dökümandaki her kelime için bir indexleme yapılır. Daha sonra hangi index numarasına sahip kelimenin hangi filmi kaç kere geçtiği hesaplanarak sayma matrisi oluşturulur.

**TfidfVectorizer** kullanarak metin verilerini vektörlere dönüştürme işlemini gerçekleştirir. Önce, **X\_train** ve **X\_test** dizileri 2D dizilere dönüştürülür ve ardından **SimpleImputer** kullanılarak eksik değerler temizlenir. **TfidfVectorizer** ise metin verilerini sayısal vektörlere dönüştürmek için kullanılır **TfidfVectorizer** bir kelimenin döküman içindeki önemi istatistiksel olarak hesaplamıştır. Bu vektörler, her bir kelimenin ve kelime kombinasyonunun belirli bir ağırlıklandırma ile ifade edildiği sayısal temsillerdir. Bu sayısal temsiller, ardından bir sınıflandırma modeline veya başka bir makine öğrenimi modeline giriş olarak kullanılabilir.

### **Modellerin Eğitilmesi ve Başarısının Hesaplanması**

Modeller birbirlerinde farklı şekilde eğitilmiştir.Cosine Similarity bir öneri algoritmasıdır bu yüzden accuracy yani başarı hesaplamak pek mümkün değildir çünkü Cosine Similarity, unsupervised bir öğrenme yöntemidir ve genellikle kullanıcının ilgisini çekebilecek önerileri sağlamak amacıyla kullanılır.Markov modeli kullanıcıdan aldığı kelimeden başlayarak veri setinden rastgele gelicek şekilde belirtilen sayıda karakterle cümle oluşturur başarı hesabı yapmak mümkün değildir.Random modelinin eğitimi şu şekildedir. Modelin tanımlandığı kısımda, **RandomForestClassifier** sınıfı kullanılarak bir Random Forest modeli belirlenir. Bu modelin tekrarlanabilirliğini sağlamak için **random\_state** parametresi kullanılmıştır.Verinin eğitim ve test setine ayrıldığı bölümde, **train\_test\_split** fonksiyonu kullanılarak **tfidf\_matrix** özellik matrisi ve **df['labels']** etiketleri, veriyi %80 eğitim ve %20 test olacak şekilde bölmek üzere ayrılır. Bu işlem, modelin eğitildikten sonra performansının değerlendirileceği bağımsız bir test setinin oluşturulmasını sağlar.Modelin eğitildiği kısımda, **fit** metodunu kullanarak model, eğitim seti üzerindeki özellikler ve etiketler arasındaki ilişkiyi öğrenir. Modelin eğitilmesi, bu ilişkiyi modelin içindeki karar ağaçlarına öğretir.   
Destek Vektör Makinesi (SVM), sınıflandırma ve regresyon problemlerini çözmek için kullanılan bir makine öğrenimi algoritmasıdır. Temelde, iki sınıf arasında bir hiper düzlem oluşturarak veri noktalarını en iyi şekilde ayırmaya odaklanır. SVM'nin önemli bir özelliği, sınıflar arasındaki marjı maksimize etmeye çalışmasıdır. Bu marj, sınıflar arasındaki en yakın destek vektörlerin uzaklığına dayanır. SVM, lineer olarak ayrılamayan veri setlerini işleyebilme yeteneği ile kernel trick gibi teknikleri kullanarak geniş bir uygulama yelpazesi sunar. Parametreler arasında C ve gamma'nın bulunması, algoritmanın esnekliğini belirler. SVM'nin başarısı, sınıflar arasında net bir ayrım yapma kabiliyeti ve düşük boyutlu veri setlerinde etkili performansıyla öne çıkar.



Modellerin Hataları

Random Forest ve Support Vector Machine modellerinin hataları farklılık göstermektedir. Random Forest veri setinden tam bir konuyu girdi olarak verdiğimizde filmi çoğunlukla bulabilmektedir ancak SVM için aynısı söyleyemeyeceğim.Aşağıda örnekler mevcuttur.

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Konuya ait film Siyah Telefondu

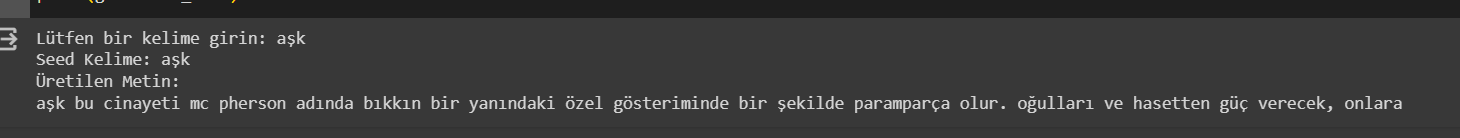
metin, ekran görüntüsü içeren bir resim

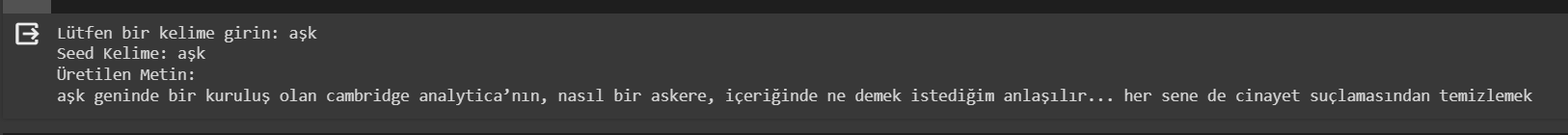
Açıklama otomatik olarak oluşturuldu

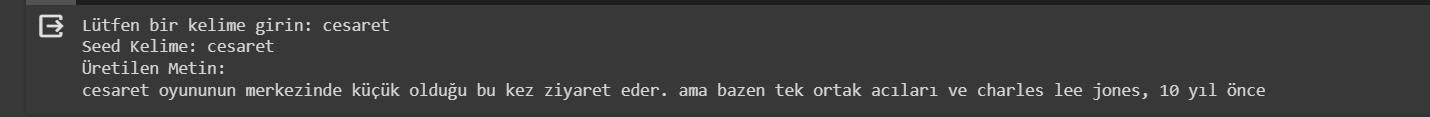
Konuya ait film Korku Seansı

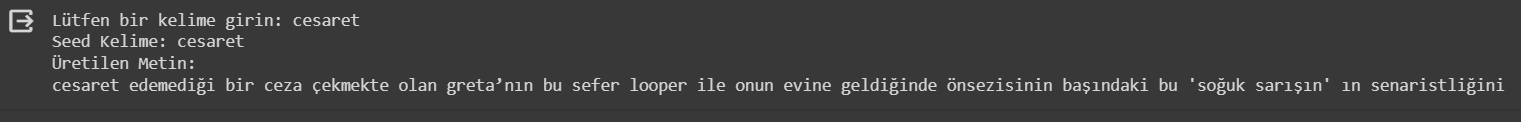
### **Modellerin Konulara Göre Yaptığı Tahminler**

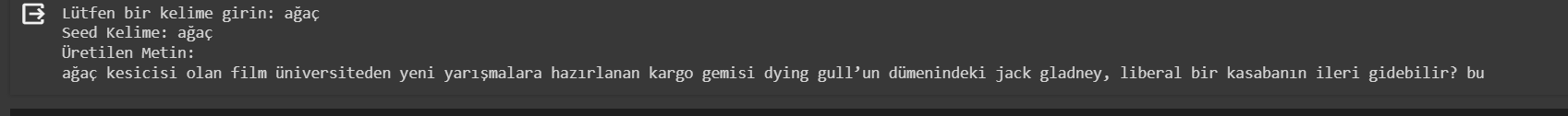
Markov Model

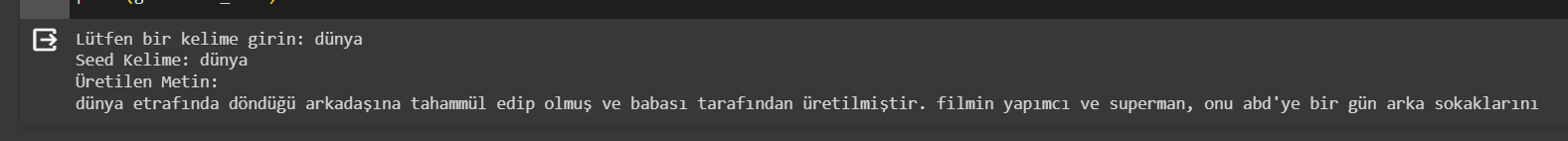












Cosine Similarity

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, multimedya yazılımı, yazılım, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Random Forest ve SVM Modeli

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, menü, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, menü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, menü içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

# Sonuç

Sonuca bakıldığında yapılan projede seçilen modelin ne kadar önemli olduğu görülmektedir. Farklı projelerde farklı modellerin daha etkili olabileceği görülebilir.Bizim modellerimiz içerisinde benzer film bulurken Random Forest Modeli , SVM ye göre daha başarılıydı. Ancak, model seçimi kadar, veri setinin dikkatlice hazırlanması da önemlidir. Noktalama işaretlerinden arındırılmış ve olumsuz etkileri minimize edilmiş stopwordslerden temizlenmiş bir veri seti, modelin başarı oranını belirlemede kritik bir faktördür. Veri setindeki özelliklerin niceliği, eğitim-test veri setinin uygun bir şekilde bölünme oranı gibi faktörler de modelin performansını önemli ölçüde etkileyebilir.