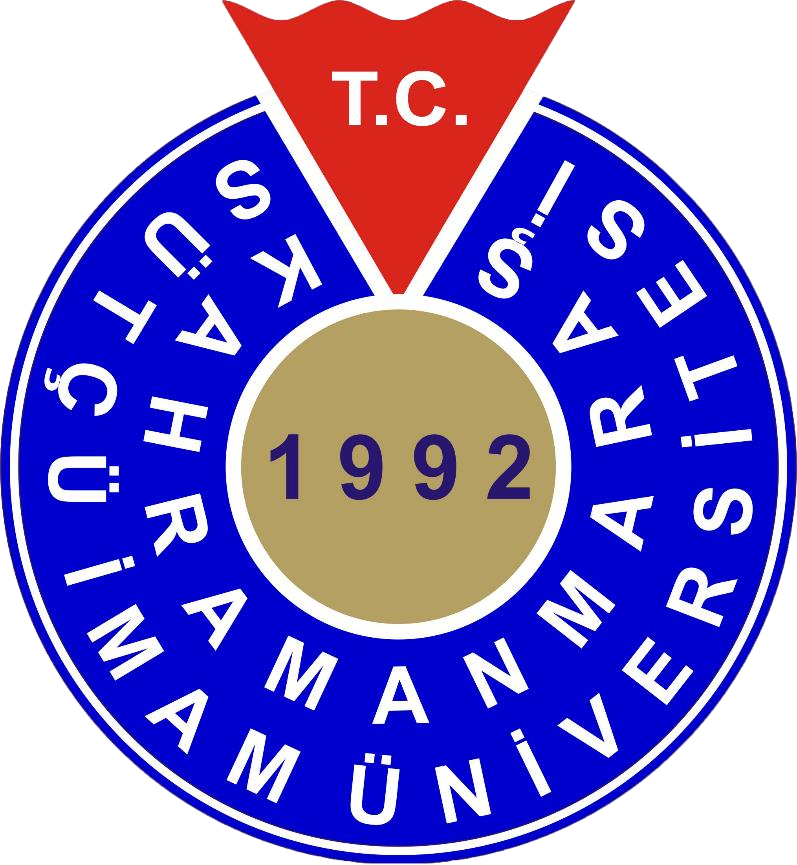
T.C. KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ

MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİL412 DOĞAL DİL İŞLEME DERSİ

BAHAR DÖNEMİ FİNAL RAPORU



**19\*\*\*\*\*\*\*\* – Merve Nilgün Yılmaz**

**BÖLÜM 1 : Problemin Belirlenmesi , Verilerin Üretilmesi**

# **Problemin Tanımı**

**E-Commerce Clothing Reviews=** E-Commerce Clothing Reviews.xlsx, Bir e-ticaret sitesinden alışveriş yapan müşterilerin aldıkları ürüne veya hizmete dair yorumlarını ve alınan ürün bilgilerini içermektedir.

**Amaç:** Kullanıcının yaptığı yorumun hangi departmana yönlendirilmesi gerektiğini belirlemek.

# **Kullanılacak Verilerin Üretilmesi**

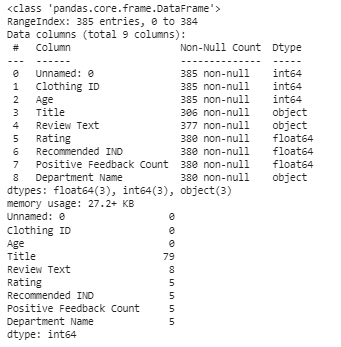
**.Veri Seti**

**İçerik:** Bu, müşteriler tarafından yazılan incelemeler etrafında dönen bir Kadın Giyim E-Ticaret veri kümesidir. Dokuz destekleyici özelliği, metni çoklu boyutları aracılığıyla ayrıştırmak için harika bir ortam sunar. Bu gerçek bir ticari veri olduğu için anonim hale getirilmiş, inceleme metninde ve metninde şirkete yapılan atıflar “perakendeci” ibaresi ile değiştirilmiştir. Bu veri kümesi 467 satır ve 8 özellik değişkeni içerir. Her satır bir müşteri incelemesine karşılık gelir ve aşağıdaki değişkenleri içerir.

* **Clothing ID** Tamsayı İncelenmekte olan belirli bir parçayı ifade eden Kategorik değişken..
* **Age:** İnceleyenlerin yaşının Pozitif Tamsayı değişkeni.
* **Title:** İnceleme başlığı için dize değişkeni..
* **Review Text:** İnceleme gövdesi için dize değişkeni.
* **Rating:** Müşteri tarafından verilen ürün puanı için 1 En Kötü'den 5 En İyi'ye kadar Pozitif Sıralı Tamsayı değişkeni
* **Recommended IND:** Müşterinin ürünü nerede tavsiye ettiğini, 1'in tavsiye edildiğini, 0'ın tavsiye edilmediğini belirten ikili değişken..
* **Positive Feedback Count:** Bu incelemeyi olumlu bulan diğer müşterilerin sayısını belgeleyen Olumlu Tamsayı.
* **Department Name:** Ürün departman adının kategorik adı.

Şekil 1: *Kullanılan Veri Setinden Bir Kesit*

Şekil 1 ‘ de veri setimizden bir kesit verilmiştir. Veri setimiz hakkında bilgi almak ve veri setimizi biraz inceleyelim.



Şekil 2: Veri seti hakkında sütun bilgileri

Şekil 2 ‘ de data.info() ile veri setimiz hakkında kısa bilgiler edindik .

**Metaryal ve Metot**

Veriler %80 eğitim seti, %20 test seti olacak şekilde rastgele bölünerek 4 farklı makine öğrenmesi modeli ile sınıflanarak test edilmiştir. Lojistik Regresyon (LR) ,Destek Vektör Makinesi (SVM), Naive Bayes (NB), Rastgele Orman (RF sınıflayıcılarının test başarıları karşılaştırılmıştır.

### Text Preprocessing( Metin Ön işleme )

Metin, mevcut tüm verilerin en yapılandırılmamış şeklidir, bu nedenle içinde çeşitli gürültü türleri mevcuttur. Bu, verilerin herhangi bir ön işleme olmaksızın kolayca analiz edilemeyeceği anlamına gelir. Metnin temizlenmesi ve standartlaştırılması, gürültüsüz ve analize hazır hale getirilmesi sürecinin tamamı, metin ön işleme olarak bilinir.

Metin ön işleme adımına geldiğimizde önce Veri setimizde bulunan metinleri büyük küçük harf duyarlılığı kazandırdık daha sonra gereksiz kullanılan noktalama işaretlerini kaldırdık.

Metin ön işlemenin üç temel adımı:

Tokenization:

Metin madenciliği üzerinde çalışmak söz konusu olduğunda bu adım en önemli önceliklerden biridir. Belirteçleştirme, temel olarak bir cümleyi, cümleyi, paragrafı veya tüm bir metin belgesini tek tek kelimeler veya terimler gibi daha küçük birimlere bölmektir. Bu küçük birimlerin her birine jeton denir.

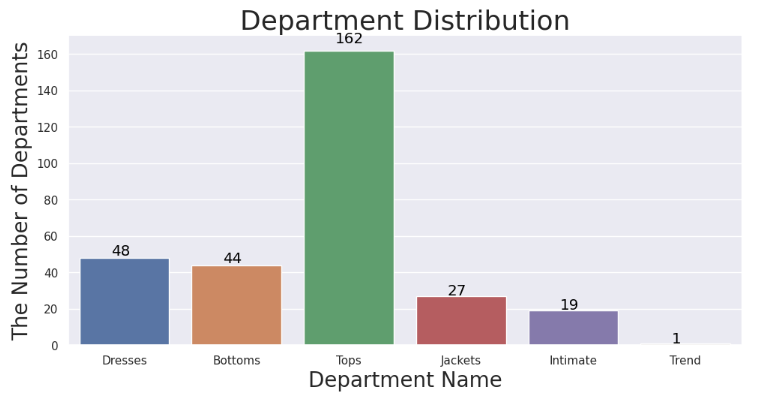
Noise Removal:

Verilerin bağlamı ve son çıktı ile ilgili olmayan herhangi bir metin parçası gürültü olarak belirtilebilir. Örneğin – dil engel sözcükleri (bir dilin yaygın olarak kullanılan sözcükleri – is, am, the, of, in vb.), URL'ler veya bağlantılar, büyük ve küçük harf ayrımı, noktalama işaretleri ve sektöre özgü sözcükler. Bu adım, metinde bulunan her türlü gürültülü varlığın kaldırılmasıyla ilgilidir.

Lexicon Normalization:

Diğer bir metinsel gürültü türü, tek bir kelimenin sergilediği çoklu temsillerle ilgilidir. Örneğin - "play", "player", "played", "plays" ve "playing" - "play" kelimesinin farklı varyasyonlarıdır. Farklı anlamlara gelseler de, bağlamsal olarak hepsi benzerdir. Bu adım, bir kelimenin tüm eşitsizliklerini normalleştirilmiş biçimlerine (lemma olarak da bilinir) dönüştürür. Sözlüğü normalleştirmenin iki yöntemi vardır; Stemming veya Lemmatizasyon. Lemmatizasyon bu durum için tavsiye edilir, çünkü Lemmatizasyon bu şekilde her kelimenin kök biçimini döndürür (yalnızca köklenme olan sonekleri çıkarmak yerine). İlk adım olarak, metni belirteçlere çevirin ve tüm kelimeleri küçük harfe çevirin. Ardından noktalama işaretlerini, kötü karakterleri, sayıları ve durma sözcüklerini kaldırın. İkinci adım ise Lemmatizasyon yöntemi ile bunların normalleştirilmesi amaçlanmaktadır.

**1.3.Veri Setine Genel Bakış**

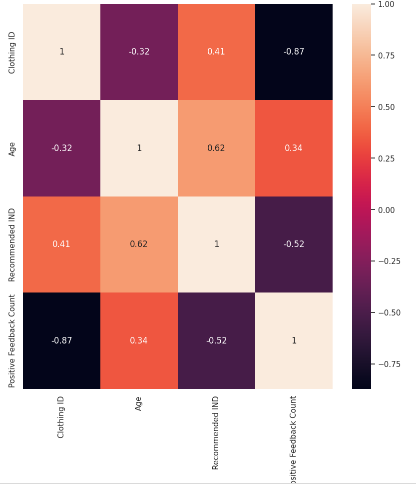


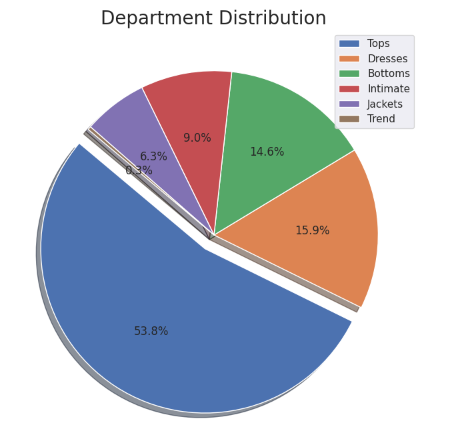
Yukarıdaki bar grafiğinde Top için yaklaşık 162 adet giriş var hemen ardında ise dresses için yaklaşık 48 giriş olduğu gözlemlenmektedir. Bunu takip eden 44 giriş ile bottoms olduğu görülmektedir. Böylece görüldüğü gibi sıralanmıştır.

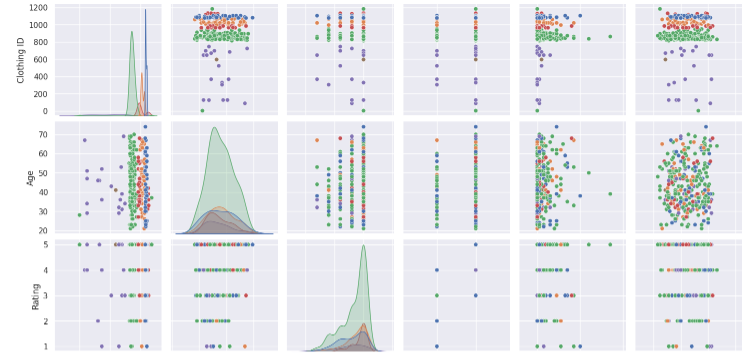
Burada ise department distribution nun teker yüzdelik oranını gözlemledik. Top için yaklaşık 162 için giriş var demiştik buda veri setinin genelinde %53 olarak gözlemlenmiştir.

**Correlation Matris**

Elimizdeki veriler nümerik değerlerden oluştuğu için correlation matrix’ine bakılması gerekmektedir. Bu matix ile elimizdeki 2 özelliğin(feature) arasındaki ilişkiye bakılır. Eğer bu ilişki 1 ise bu iki özellik %100 doğru orantılıdır denir. Eğer bu ilişki -1 ise iki özellik %100 ters orantılıdır anlamına gelmektedir. Eğer 0 ise herhangi bir ilişki yoktur denir.



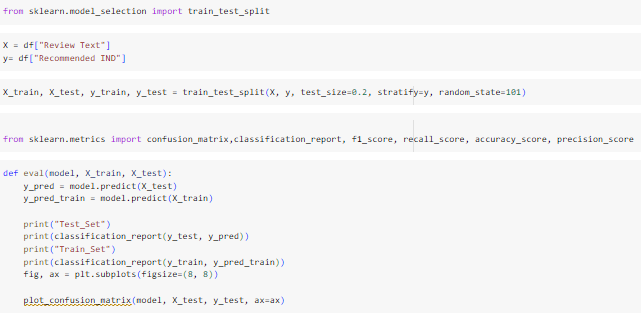


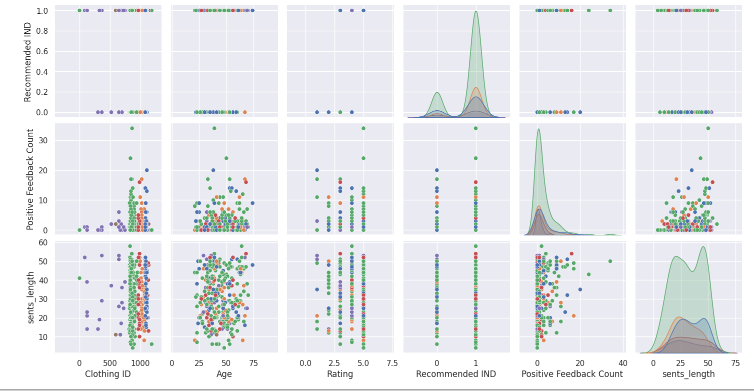
 **PairPlot Gösterimi**

Kullanılan veri setimiz üzerinde PairPlot gösterimine bakılarak dağılımlar incelendi.

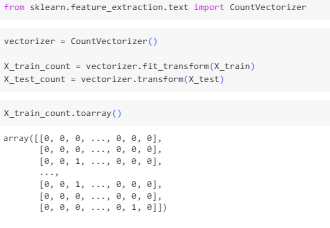
**Train-Test Split**

Oluşturduğumuz modeli eğitmek için train -test işlemleri yapacağız burda 80% eğitim 20% test olmak üzere veri setimizi parçaladık.





**Count Vectorizer**

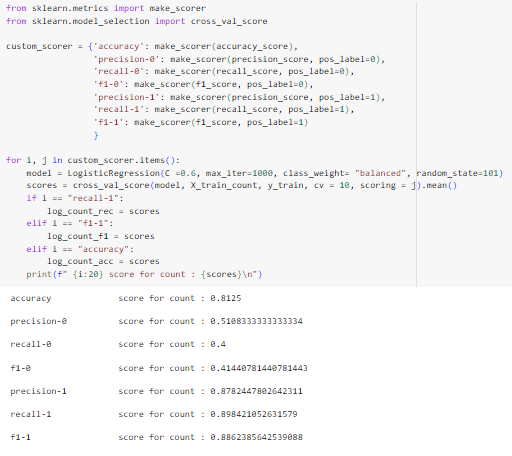
Count Vectorizer, belirli bir diziyi bir frekans temsiline dönüştürmenin bir yoludur. Herhangi bir NLP Veri Bilimcisinin karşılaştığı en büyük zorluklardan biri, Makine Öğrenimi modellerini çalıştırmak için metin dizilerinin mümkün olan en iyi sayısal/vektörel temsilini seçmektir. Genel bir senaryoda, etiketlenmiş veya etiketlenmemiş bir grup metin bilgisi üzerinde çalışıyoruz ve ardından yeni bir metin verisini tahmin etmek için dizelerde bulunan sözcüklere dayalı modeli anlayabilen bir makine öğrenimi modeli oluşturmak istiyoruz. Örneğin, "Dışarıda çok yağmur yağıyor" ile "Dışarıda sağanak yağıyor" aynı anlama geldiğini biliyoruz. Ama bilgisayarın bunu anlamasını nasıl sağlayacağız. Bildiğimiz kadarıyla bilgisayarlar sadece sayısal verileri anlayabilirken, bilgisayarlar için doğal dil verileri herhangi bir sayısal veya istatistiksel bilgi içermeyen metin dizileridir.

Veri setimiz üzerinde Count Vectorizetion uyguladık ve belirli bir diziyi frekans temsiline dönüştürmüş olduk.

**Modellerin Oluşturulması**

Gerekli Ön işleme adımlarımızı bitirdikten sonra modelimizi oluşturuyoruz.

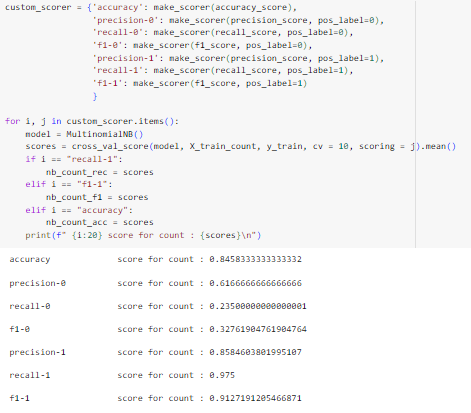
**1-Logistic Regression**

Lojistik regresyon, iki veri faktörü arasındaki ilişkileri bulmak için matematikten yararlanan bir veri analizi tekniğidir. Lojistik regresyon, daha sonra diğerine dayalı bu faktörlerden birinin değerini tahmin etmek için bu ilişkiyi kullanır. Tahminin genellikle evet ya da hayır gibi sınırlı sayıda sonucu vardır.

Yukarıda görüldüğü gibi Logistic Regression modelimizi eğittiğimiz zaman %81.25 başarı oranını yakalamış olduk.

**2-Naive Bayes**

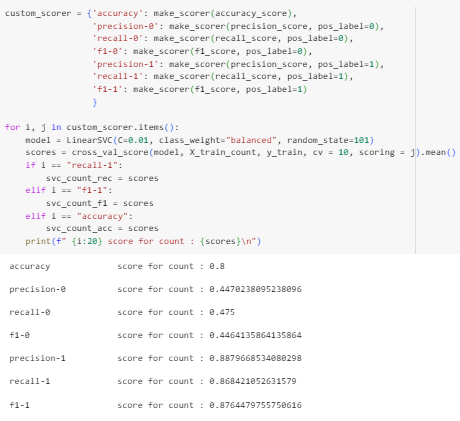
**Bayes teoremi**, olasılık kuramı içinde incelenen önemli bir konudur. Bu **teorem** bir rassal değişken için olasılık dağılımı içinde koşullu olasılıklar ile marjinal olasılıklar arasındaki ilişkiyi gösterir.



Naive bayes sınıflandırıcımızda eğitilen model üstünde yakaladığımız başarıya baktığımız zaman %84.58 başarı oranı ile en yüksek başarı oranı veren algoritmamızdır. Bir önceki algoritmamız olan Logistic Regression ile kıyasladığımızda daha yüksek bir başarı yakaladığımızı görmüş olduk.

**3-Destek Vektör Makinesi(SVM)**

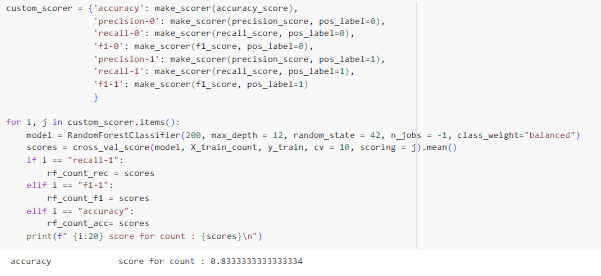
Destek Vektör Makineleri (Support Vector Machine) genellikle sınıflandırma problemlerinde kullanılan gözetimli öğrenme yöntemlerinden biridir. Bir düzlem üzerine yerleştirilmiş noktaları ayırmak için bir doğru çizer. Bu doğrunun, iki sınıfının noktaları için de maksimum uzaklıkta olmasını amaçlar. Karmaşık ama küçük ve orta ölçekteki veri setleri için uygundur.

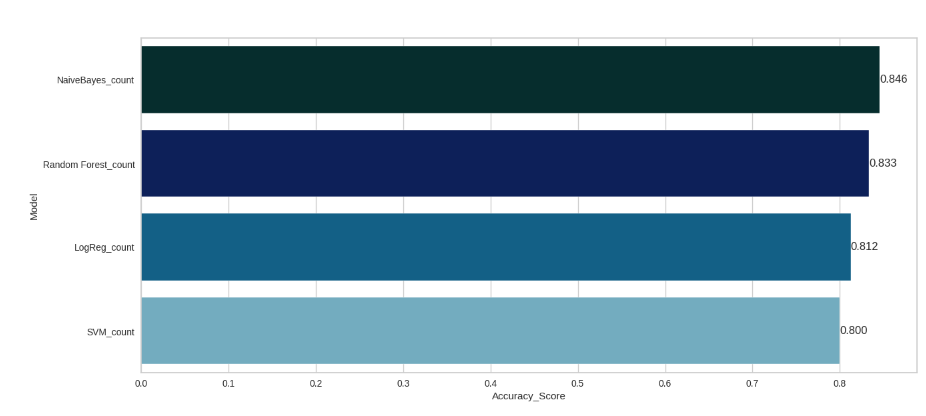


Bu modelimizde ise diğerlerine göre daha düşük bir başarı oranı elde ettik buda %80 ‘lik bir başarı oranıdır.

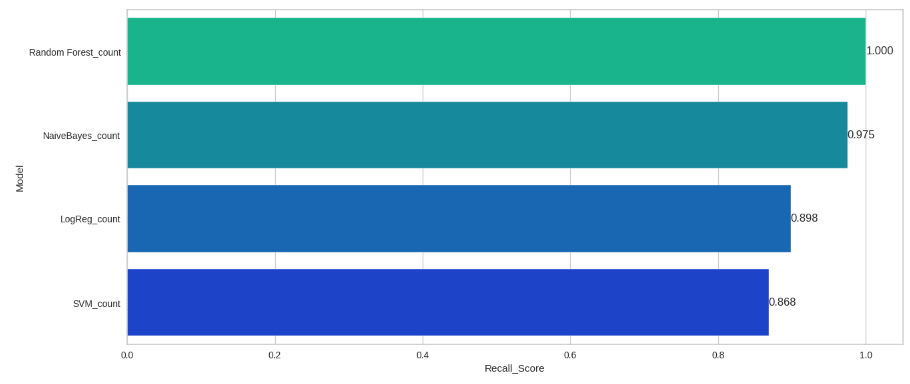
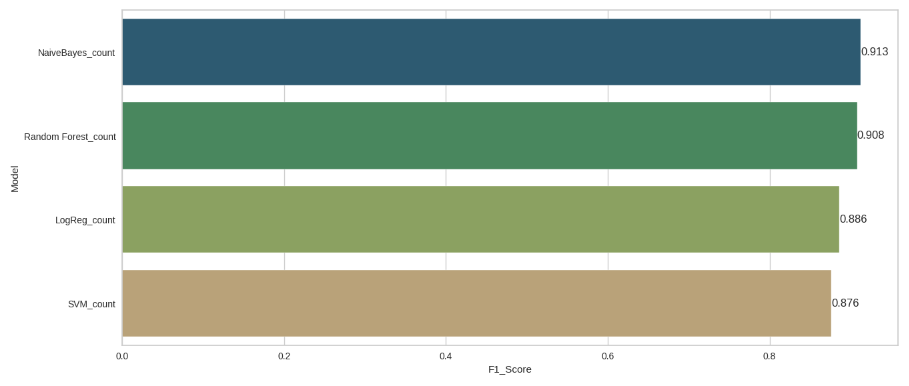
**4-Rastgele Orman(Random Forest)**

Random forest algoritması ,denetimli sınıflandırma algoritmalarından biridir. Hem regresyon hem de sınıflandırma problemlerinde kullanılmaktadır. Algoritma ,birden fazla karar ağacı üreterek sınıflandırma işlemi esnasında sınıflandırma değerini yükseltmeyi hedefler. Random forest algoritması birbirinden bağımsız olarak çalışan birçok karar ağacının bir araya gelerek aralarından en yüksek puan alan değerin seçilmesi işlemidir.

Bu algoritmada ise %83.30 bir başarı oranı yakalanmıştır. En yüksek Naive Bayes algoritmasından sonra en yüksek başarı oranı bu algoritma ile yakalanmıştır.

**Projede Kullanılan Modellerin Karşılaştırılması**

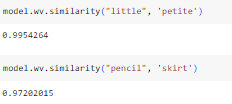
Görüldüğü gibi en yüksek başarı oranını Naive Bayes, sonraki en yüksek başarı oranı random forest daha sonrasında onu takip eden logistic regression ve en düşük başarı oranına sahip ise destek vektör makinesi sınıflandırıcımız olmuştur. Böylece 4 modeli uygulayıp başarı oranlarını kıyaslamış olduk.



Yukarıdaki 2 sonuç tablosunda ise F1\_Score ve Recall\_Score için başarı oranları gözlemlenmiştir.

**Word2Vec Modeli**

Word2Vec modelinde her kelime temel olarak tek bir sayı yerine 32 veya daha fazla boyutlu bir vektör olarak temsil edilir. Burada semantik bilgi ve farklı kelimeler arasındaki ilişki de korunur. Word2vec, kelime yerleştirmeleri üretmek için en yaygın kullanılan modellerden biridir. Bu modeller, kelimelerin dilsel bağlamlarını yeniden oluşturmak için eğitilmiş sığ, iki katmanlı sinir ağlarıdır. Word2Vec iki şekilde uygulanabilir, biri Gram Atla, diğeri ise Sürekli Sözcük Torbasıdır.



İki kelime ne kadar benzerlik gösteriyor yan tarafta word2vec modelimizi kullanarak bunun sonuç başarı oranı gözlemlenmiştir.



Ele alınan iki kelimeden diğerleri arasında hangi kelime uyuşmuyor buna göz atıldı.

**KAYNAKÇA**

1. <https://towardsdatascience.com/clothes-reviews-analysis-with-nlp-part-1-d81bdfa14d97>
2. <https://towardsdatascience.com/clothes-reviews-analysis-with-nlp-part-1-bfb8a3a2c4bd>
3. <https://medium.com/@ANDREY_FATEEV/womens-e-commerce-review-dataset-data-analysis-fab4bccf99b4>
4. <https://learn.hackwagon.com/showcase/6ARd2TMtWgGLgcTgx>
5. <https://datauab.github.io/sentiment_predictions/>
6. <https://www.kaggle.com/code/saiprasadpadala/starter-women-s-e-commerce-clothing-72789f05-0>
7. <https://www.kaggle.com/code/gourabroy553/women-s-e-commerce-clothing-reviews>