

MAKİNE ÖĞRENMESİ İLE METİN SINIFLANDIRMA

Hazırlayan: Elif Eroğlu

Teslim Tarihi: 24 Mayıs 2025

Görev: Görev 2- Makine Öğrenmesi İle Metin Sınıflandırma

Firma: DFA Teknoloji

1. Giriş

Günümüzde dijital ortamlarda üretilen metin miktarı hızla artmaktadır. Sosyal medya paylaşımları, haber içerikleri, müşteri yorumları gibi kısa metinler, kullanıcı tercihlerini anlamak, içerikleri düzenlemek ve otomatik analiz yapmak açısından büyük önem taşımaktadır. Bu nedenle metin sınıflandırma, doğal dil işleme (NLP) alanında sıkça karşılaşılan ve uygulamalarda kullanılan bir problemidir.

Metin sınıflandırma, bir metnin içerdiği bilgiye göre önceden tanımlanmış kategorilerden birine atanması işlemidir. Örneğin, haber metinlerinin “ekonomi”, “spor”, “magazin” gibi konulara otomatik olarak ayrılması, kullanıcıya uygun içerik önerileri sunmak ya da haber sitelerini içerik yönetimini kolaylaştırmak gibi faydalar sağlar.

Bu çalışmanın amacı; ekonomi, spor, magazin ve gündem gibi kategorilere basit ama etkili modellerle çalışarak doğal dil işleme (NLP) yetkinliğini göstermek, iki farklı makine öğrenmesi modeliyle eğitim-tes süreçleri yürütülecek ve sonuçlarla karşılaştırılacaktır.

2. Veri Seti

Bu çalışmada kullanılan veri seti, Kaggle platformunda Rishabh Misra tarafından yayımlanan “**News Category Dataset**” isimli açık veri setidir.

Kaynak bağlantısı: <https://www.kaggle.com/datasets/rmisra/news-category-dataset>

2.1 Veri Seti Seçim Gerekçesi

Veri seti seçilirken aşağıdaki kriterler göz önünde bulundurulmuştur.

- Etiketlerin açık ve belirgin olması:** “sport”, “business”, “tech” gibi doğrudan sınıflandırılabilir başlıklar barındırması, modelin öğrenmesini kolaylaştırmaktadır.
- Çok kategorili yapı:** Sadece ikili değil, çoklu sınıflandırma problemi sunması, modelin genel becerisini test etmek açısından daha değerlidir.
- Yaygın kullanımı ve referans verilebilirliği:** Kaggle üzerinde sıkça kullanılan ve alıntılanan bir veri seti olması, çalışmanın güvenilirliğini artırmaktadır.
- İngilizce içerik tercihi:** Türkçe yerine İngilizce metinlerden oluşan bir veri seti tercih edilmiştir. Çünkü Türkçe için güçlü tokenizer, stopword listesi vs.

kütüphaneler sınırlı. İngilizce için ise doğal dil işleme (NLP) alanında çok daha gelişmiş ve hazır ön işleme araçları bulunmaktadır. Bu sayede daha iyi bir analiz ve sınıflandırma süreci mümkün olur.

Bu nedenlerden dolayı veri kaynağı olarak bu veri seti tercih edilmiştir.

3. Ön İşleme Aşaması

Metin verileri doğrudan makine öğrenmesi modellerine verilmeden önce çeşitli ön işleme adımlarından geçirilmelidir. Bu süreç, modelin daha doğru öğrenmesini sağlar. Uygulanan temel ön işleme adımları şu şekildedir:

3.1.Küçük Harfe çevirme

Tüm metinler, büyük-küçük harf duyarlılığını ortadan kaldırmak için amacıyla küçük harfe çevrilmiştir.

3.2.Noktalama İşaretlerinin Ve Özel Karakterlerin Temizlenmesi

Noktalama işaretleri, sayılar ve metin analizi için anlam ifade etmeyen özel karakterler metinlerden çıkarılmıştır.

3.3.Stopword (Anlamsız kelime) Temizliği

Sık geçen ancak sınıflandırma için anlamlı bilgi taşımayan kelimeler (stopwords) temizlenmiştir. Bu işlem için NLTK (Natural Language Toolkit) kütüphanesi kullanılmıştır.

3.4. Tokenizasyon

Metinler, kelime bazlı parçalara ayrılmıştır. Bu işlem, her bir metni kelime sıklığı temelli bir temsil yöntemine dönüştürmek için uygulanmıştır. Python ortamında bu adım NLTK ve Scikit-learn kütüphanelerinde uygun fonksiyonlar yardımıyla yapılmıştır.

3.5.Lemmatizasyon

Kelimelerin kök hallerine indirgenmesi işlemidir (örneğin: “running”=”run”). Bu adım, veri setinin yapısına göre tercih edilmiştir fakat bazı sınıflandırma modellerinde fark yaratmayabilir.

4.Modelleme Ve Eğitim Süreci

Ön işleminden geçen metinler, makine öğrenmesi modelleri ile sınıflandırılmaya hazır hale getirilmiştir. Bu aşamada amaç her metni doğru kategoriye atayabilecek bir sistem yapmaktır.

Bu çalışmada iki farklı model tercih edilmiştir:

Multinomial Naive Bayes ve **logistic Regression**. İki modelin seçilme nedeni, metin verileriyle yaygın ve etkili şekilde çalışmalarınıdır.

4.1. Özellik Dönüştürme: TF-IDF Vektörleştirme

Metinlerin doğrudan algoritmalara verilemeyeceği için, kelimeler sayısal verilere dönüştürülmelidir. Bu amaçla **TF_IDF (Term Frequency- Inverse Document Frequency)** yöntemi kullanılmıştır.

Bu teknik sayesinde, metinlerdeki kelimelerin önemi belirlenir ve modelin gerçekten anlamlı kelimeler üzerinde odaklanması sağlanır.

4.2.Eğitim Ve Test Ayrımı

Veri seti, modele öğretmek ve başarıyı test etmek için ikiye bölünmüştür:

1. **%80 eğitim verisi:** Modelin örüntüleri öğrenmesi için kullanılır.
2. **%20 test verisi:** Modelin hiç görmediği verilerle başarısı ölçülür.

Bu oran dengesizliğe neden olmadan modelin genelleme yeteneğini ölçmek için idealdir.

4.3. Kullanılan Modeller

4.3.1 Multinomial Naive Bayes

Naive Bayes modeli, kelimelerin birbirinden bağımsız olduğu varsayımıyla çalışır. Her kelimenin ait olduğu sınıfla olan ilişkisine bakarak tahmin yapar.

Avantajı: Hızlıdır, düşük kaynak kullanır ve metin sınıflandırma gibi problemler için uygundur

4.3.2. Logistic Regression

Her sınıf için ayrı bir olasılık hesaplayarak en yüksek ihtimalli sınıfı seçer.

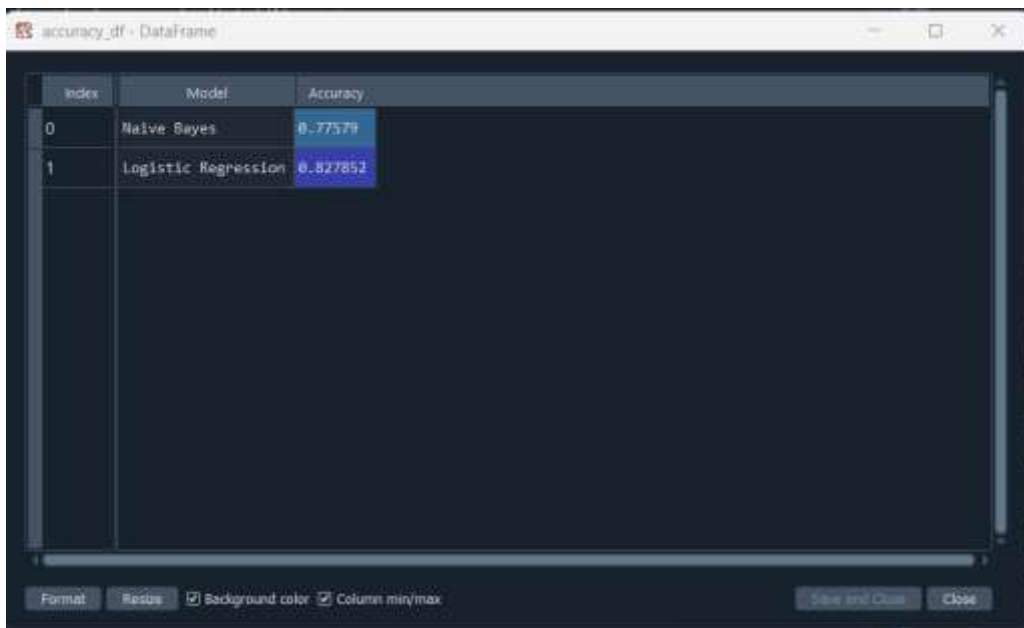
Avantajı: Özellikle çok sınıflı problemler için başarılı sonuçlar verir. Daha esnek bir yapıya sahiptir ve kelimeler arası ilişkilere duyarlıdır.

5.Değerlendirme ve Karşılaştırma

Model başarısı sadece doğruluk oranıyla değil, aynı zamanda detaylı analizlerle değerlendirmek gerekir.

5.1. Doğruluk (Accuracy)

Modelin doğru yaptığı tahminlerin, toplam tahmin sayısına oranıdır. Her iki model için doğruluk hesaplanmış ve aşağıdaki tabloda karşılaştırılmıştır.

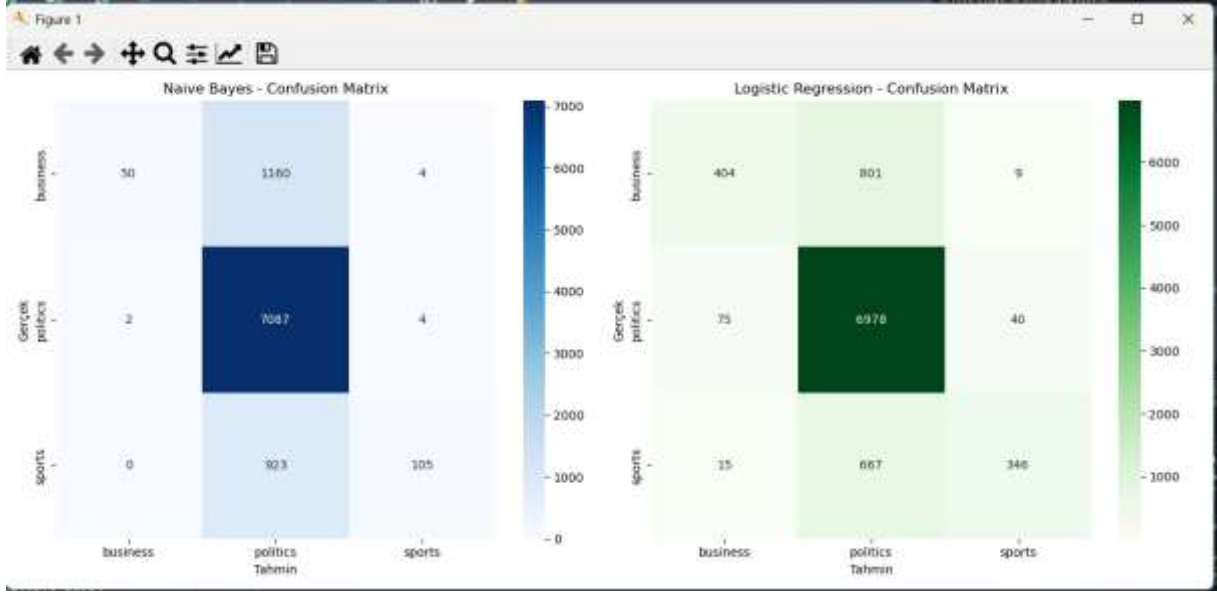


The screenshot shows a Jupyter Notebook window titled 'accuracy_df - DataFrame'. It displays a DataFrame with two rows of data. The first row is for 'Naive Bayes' with an accuracy of 0.77579. The second row is for 'Logistic Regression' with an accuracy of 0.827852. The DataFrame is shown in a table view with columns 'index', 'Model', and 'Accuracy'.

index	Model	Accuracy
0	Naive Bayes	0.77579
1	Logistic Regression	0.827852

5.2. Karmaşıklık Matrisi (Confusion Matrix)

Her sınıf için kaç adet doğru ve yanlış tahmin yapıldığını gösterir. Örneğin, “sport” kategorisindeki 200 örnekten 170’i doğru, 30’u yanlış tahmin edildiyse bu tablo bunu gösterir.



Bu matris, özellikle modelin hangi sınıfları karıştırdığını görmek açısından önemlidir.

5.3. Sonuçların Karşılaştırması (Örnek Tablo)

MODEL	DOĞRULUK ORANI	GÖZLEM
NAİVE BAYES	%77.5	Bu model metin sınıflandırma görevinde makul bir başarı sağladığını ancak bazı sınıflarda hata yapabileceğini göster
LOGİSTİK REGRESİYON	%82.7	Bu model daha yüksek bir doğruluk oranı elde ederek, sınıflandırma görevinde daha güçlü bir performans sergilemiştir.

Elde edilen doğruluk oranlarına göre Lojistik Regresyon modeli (%82.7) Naive Bayes modeline (%77.5) kıyasla daha başarılı sonuçlar vermiştir. Karmaşıklık matrisleri incelendiğinde, her iki modelin de bazı sınıflarda karışıklık yaşadığı, ancak genel olarak sınıfları ayırt etmede başarılı olduğu görülmektedir.