BURSA TEKNİK ÜNİVERSİTESİ

Veri Madenciliğine Giriş – Dönem Projesi

Üniversite Adı : Bursa Teknik Üniversitesi

Bölüm : Mühendislik ve Doğa Bilimleri Fakültesi - Bilgisayar Mühendisliği

Dersin Adı : Veri Madenciliği

Öğrenci Adı Soyadı : Hasan Umut Kocatepe

Öğrenci No : 21360859077

Ders Öğretim Üyesi : Doç. Dr. Erdem Yavuz

Proje Teslim Tarihi : 04.06.2025

İçindekiler

1. Özet

Bu projede, SMS mesajlarını spam olarak sınıflandırmak amacıyla Multinomial Naive Bayes algoritması Python diliyle sıfırdan geliştirilmiştir. UCI Machine Learning Repository'den alınan SMS Spam Collection veri seti kullanılarak mesajlar "ham" ve "spam" olmak üzere iki sınıfa ayrılmıştır. Metinler küçük harfe dönüştürülüp temizlendikten sonra model eğitilmiş ve test verisi üzerinde %98.38 doğruluk oranı elde edilmiştir. Değerlendirme ölçütlerine göre model, özellikle spam sınıfını %94 F1-score ile başarıyla sınıflandırmıştır. Bu sonuçlar, yöntemin metin sınıflandırmada basit ancak etkili bir yaklaşım olduğunu göstermektedir.

2. Giriş

Dijital iletişimin yaygınlaşmasıyla birlikte spam mesajlar hem kullanıcı deneyimini olumsuz etkilemekte hem de güvenlik riski oluşturmaktadır. Bu nedenle, spam mesajların otomatik olarak tespiti önemli bir ihtiyaç haline gelmiştir. Bu tür sınıflandırma problemlerinde metin madenciliği ve makine öğrenmesi yöntemleri yaygın olarak kullanılmaktadır.

Bu projede, SMS mesajlarını spam olarak sınıflandırmak amacıyla Multinomial Naive Bayes algoritması sıfırdan Python ile geliştirilmiş ve hazır kütüphaneler kullanılmadan uygulanmıştır. Elde edilen sonuçlar, bu basit yapılı yöntemin metin sınıflandırma problemlerinde etkili bir çözüm sunduğunu göstermektedir.

3. Kullanılan Yöntem

Bu projede sınıflandırma algoritması olarak, metin madenciliği alanında sıkça kullanılan **Multinomial Naive Bayes (MNB)** yöntemi tercih edilmiştir. Naive Bayes, Bayes teoremine dayanır ve temel varsayımı, metindeki kelimelerin birbirinden koşulsuz bağımsız olduğudur. MNB, sınıf koşullu olasılıkları kelime frekanslarına göre hesaplayarak bir metnin hangi sınıfa ait olduğunu belirlemeye çalışır.

Bu algoritma, Python dili kullanılarak hazır kütüphaneler olmadan sıfırdan geliştirilmiştir. Eğitim sürecinde her sınıf için:

- Toplam mesaj sayısına göre sınıf öncül olasılığı P(Y)
- Her kelimenin sınıfa göre frekansı P(Wi | Y)
- Tüm kelime çeşitliliği (vocabulary) ve toplam kelime sayısı

gibi istatistikleri toplar. Bu istatistikler kullanılarak, bir test mesajının sınıfa ait olma olasılığı logaritmik olarak hesaplanır:

- Logaritmik Toplam Olasılık Formülü :

$$\log P(Y \mid X) = \log P(Y) + \sum_{i=1}^{n} \log P(Wi \mid Y)$$

Bu hesaplamalarda **Laplace smoothing (add-one)** uygulanarak sıfır olasılık problemleri önlenmiştir. Test aşamasında her sınıf için bu log-olasılık skorları hesaplanmış ve en yüksek değere sahip sınıf tahmin edilmiştir.

Algoritmanın tüm adımları — tokenizasyon, vektörleme, olasılık hesaplamaları ve sınıf tahmini — elle uygulanmış ve modelin mantığı şeffaf şekilde gözlemlenebilir hale getirilmiştir.

4. Veri Seti ve Ön İşleme

Bu projede kullanılan veri seti, UCI Machine Learning Repository'den alınan **SMS Spam Collection** veri kümesidir. İngilizce dilinde toplam **5574 SMS mesajı** içeren bu veri seti, her satırda bir mesaj ve onun etiketini barındırmaktadır. Mesajlar `"ham"` (normal) veya `"spam"` (istenmeyen) olarak sınıflandırılmıştır.

- **Ham mesaj sayısı:** 4827 (%86.6)
- **Spam mesaj sayısı:** 747 (%13.4)

Veri setindeki sınıf dengesizliği nedeniyle, yalnızca doğruluk (accuracy) değil; **precision**, **recall** ve **F1-score** gibi ek değerlendirme metriklerinin kullanılması önemlidir.

Veri kümesine aşağıdaki ön işleme adımları uygulanmıştır:

1. **Etiket Kodlama:**
   `"ham"` etiketi 0, `"spam"` etiketi 1 olarak sayısal değerlere dönüştürülmüştür.

2. **Veri Bölme (Train/Test):**
   Veri %80 eğitim, %20 test olmak üzere `train_test_split()` fonksiyonu ile ayrılmıştır.

3. **Tokenizasyon:**
   Mesajlar küçük harfe çevrilmiş, noktalama işaretlerinden arındırılmış ve kelime kelime ayrılmıştır. Bu işlem özel bir `tokenize()` fonksiyonu ile gerçekleştirilmiştir.
   Örnek:
   Girdi: **"Free entry in 2 a wkly comp! Text WIN to 80086"**

Çıktı: **['free', 'entry', 'in', '2', 'a', 'wkly', 'comp', 'text', 'win', 'to', '80086']**

4. **Vocabulary (Kelime Sözlüğü) Oluşturma:**
   Eğitim verisi taranarak her sınıfa ait kelime frekansları ve toplam kelime sayıları hesaplanmıştır. Böylece her kelimenin sınıf bazlı olasılığı belirlenmiştir.

5. Modelin Eğitimi ve Test Edilmesi

Bu projede, SMS mesajlarının spam olup olmadığını belirlemek amacıyla **Multinomial Naive Bayes (MNB)** algoritması Python dili ile sıfırdan geliştirilmiştir. Modelin tüm eğitim ve test süreci manuel olarak gerçekleştirilmiş, herhangi bir hazır sınıflandırma modeli (örneğin `scikit-learn`) kullanılmamıştır. Bu sayede, algoritmanın matematiksel yapısı doğrudan uygulanarak daha iyi anlaşılması ve kontrol edilmesi sağlanmıştır.

5.1 - Modelin Eğitimi (fit)

Eğitim sürecinde aşağıdaki adımlar uygulanmıştır:

- **Etiket Kodlama ve Veri Bölme:**
  Mesajlar `"ham" → 0`, `"spam" → 1` olarak kodlanmış; veri kümesi %80 eğitim, %20 test olarak ayrılmıştır.
- **Sınıf Olasılıklarının Hesaplanması:**
  Her sınıfın veri kümesindeki oranı logaritmik olarak hesaplanarak log P(Y) elde edilmiştir.
- **Kelime Frekansları ve Olasılıklar:**
  Eğitim verisindeki kelimeler sınıflara göre sayılmış ve her kelime için P(Wi| Y) olasılığı hesaplanmıştır.

- **Vocabulary ve Smoothing:**

  Her sınıf için toplam kelime sayısı ve kelime çeşitliliği belirlenmiş; **Laplace smoothing** yöntemiyle sıfır olasılık problemi engellenmiştir:

  $$P(Wi \mid Y) = (count(Wi , Y) + 1) / (total\_count\_in\_Y + |V|)$$

5.2 - Modelin Test Edilmesi (predict)

Test aşamasında model, her mesaj için aşağıdaki şekilde işlem yapmıştır:

1. **Tokenizasyon:**

   Mesaj küçük harfe çevrilip kelimelere ayrılmıştır.

2. **Log-Olasılıkların Hesaplanması:**

   Her kelimenin sınıf olasılığı toplanarak toplam log skoru elde edilmiştir:

$$\log (P(Y \mid X)) = \log(P(Y)) + \sum_{i=1}^{n} \log (P(Wi \mid Y))$$

3. **Tahmin:**

   En yüksek log-olasılığa sahip sınıf, modelin çıktısı olarak atanmıştır.

6. Sonuçlar ve Değerlendirme

Modelin başarı performansı, test veri kümesi üzerinde yapılan tahminler sonucunda doğruluk (accuracy), kesinlik (precision), duyarlılık (recall), F1 skoru ve karmaşıklık matrisi (confusion matrix) gibi istatistiksel ölçütlerle değerlendirilmiştir.

6.1 - Başarı Ölçütleri

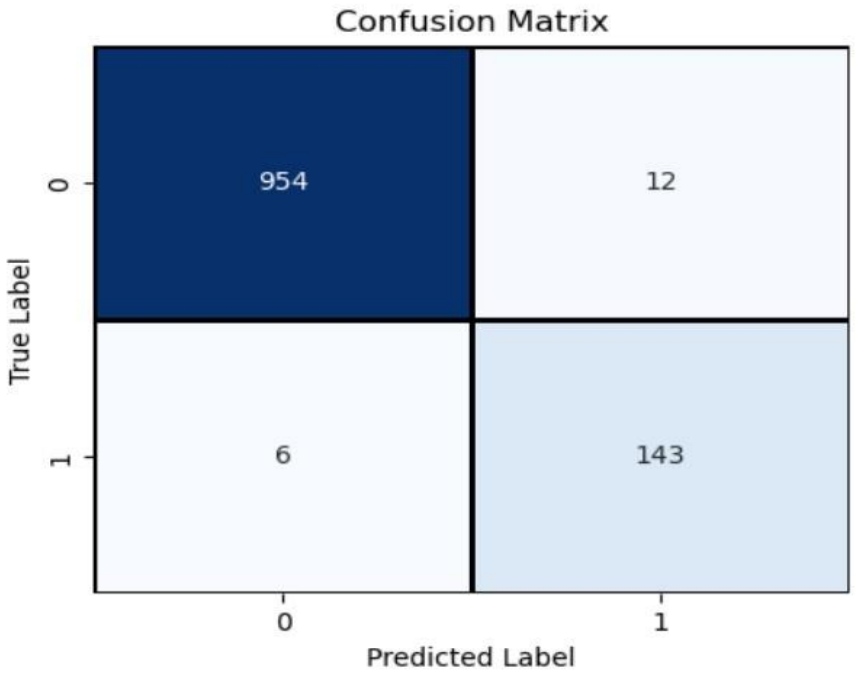
Modelin performansı, test veri kümesi üzerinde yapılan tahminlere dayalı olarak değerlendirilmiş; doğruluk (accuracy), kesinlik (precision), duyarlılık (recall), F1 skoru ve karmaşıklık matrisi gibi yaygın sınıflandırma metrikleri kullanılmıştır. **sklearn.metrics** kütüphanesi aracılığıyla elde edilen sonuçlar aşağıdaki gibidir:

```
Accuracy: 0.9838565022421525
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       966
           1       0.92      0.96      0.94       149

    accuracy                           0.98      1115
   macro avg       0.96      0.97      0.97      1115
weighted avg       0.98      0.98      0.98      1115
```

Bu veriler, modelin ham mesajları %99 oranında doğru sınıflayabildiğini ve spam mesajlarda %96 recall değeriyle yüksek başarı sağladığını göstermektedir. Spam sınıfında elde edilen %92 precision değeri, yanlış alarm oranının düşük olduğunu ortaya koymaktadır.

6.2 - Confusion Matrix Analizi

Modelin tahmin sonuçları aşağıdaki confusion matrix ile gösterilmiştir:



Bu tabloya göre:

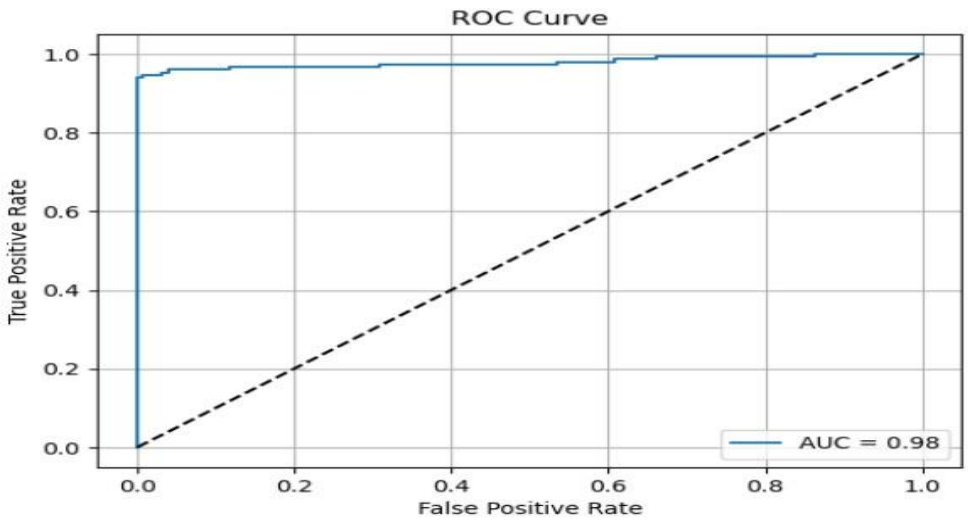
- 954 adet ham mesaj doğru sınıflanmıştır **(TN).**
- 143 adet spam mesaj doğru sınıflanmıştır **(TP).**
- 6 adet spam mesaj yanlışlıkla "ham" olarak sınıflanmıştır **(FN).**
- 12 adet ham mesaj yanlışlıkla spam olarak sınıflandırılmıştır **(FP).**

Bu sonuçlar modelin genel olarak yüksek başarı oranına sahip olduğunu ve özellikle spam mesajları ayırt etmede güçlü bir performans sergilediğini göstermektedir. Ayrıca **false positive** oranı (yanlış alarm) düşük tutulmuş , **false negative** (atlanan spam) ise minimumda kalmıştır.

6.3 - ROC Eğrisi ile Sınıflandırma Başarısı

Modelin sınıflandırma başarısı, ROC (Receiver Operating Characteristic) eğrisi ile analiz edilmiştir. ROC eğrisi, modelin farklı eşik değerlerinde doğru pozitif oranı (TPR) ile yanlış pozitif oranı (FPR) arasındaki ilişkiyi gösterir.

Aşağıdaki grafikte görüldüğü gibi, ROC eğrisi sol üst köşeye oldukça yakın seyretmekte; bu da modelin spam mesajları yüksek doğrulukla ayırt edebildiğini göstermektedir. Eğrinin altında kalan alan, yani **AUC (Area Under Curve)** değeri **0.98** olarak hesaplanmıştır.



Bu yüksek AUC değeri, modelin sınıflandırma gücünün hem **doğru tahmin yapma oranı** hem de **yanlış alarm üretmeme oranı** açısından güçlü olduğunu göstermektedir.

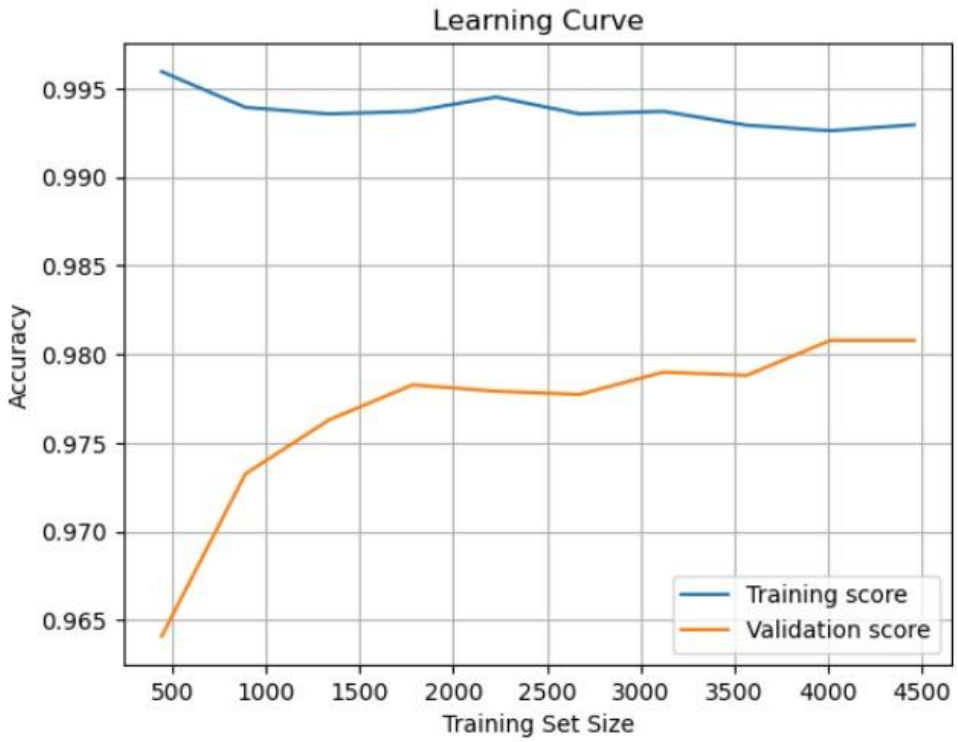
Teknik Yorum:

- **TPR yüksek:** Spam mesajlar başarıyla tespit edilmiştir.
- **FPR düşük:** Yanlış alarm oranı düşüktür.
- **AUC = 0.98:** Modelin ayırt ediciliği oldukça yüksektir.

6.4 - Öğrenme Eğrisi ile Genel Performans

Modelin genelleme yeteneği, eğitim veri seti boyutuna göre elde edilen doğruluk değerlerinin incelendiği **öğrenme eğrisi** ile değerlendirilmiştir. Bu grafik, modelin **eğitim doğruluğu** ile **doğrulama doğruluğunun** veri miktarına bağlı olarak nasıl değiştiğini gösterir.

Aşağıdaki grafikte görüldüğü üzere, eğitim verisinin yalnızca %10'u ile başlandığında doğrulama doğruluğu görece düşükken, veri oranı arttıkça doğruluk değerleri yükselmiş ve istikrar kazanmıştır. Eğitim ve doğrulama eğrilerinin birbirine yakın seyretmesi, modelin **aşırı öğrenme (overfitting)** yapmadığını ve **genellemeyi başarıyla gerçekleştirdiğini** göstermektedir.



Teknik Yorum:

- **Eğitim Doğruluğu ≈ 0.99:** Model, eğitim verisinde yüksek başarı sağlamıştır.
- **Doğrulama Doğruluğu ≈ 0.975:** Test verisinde de güçlü performans göstermektedir.

- **Eğriler arası yakınlık:** Model dengeli ve güvenilir şekilde öğrenmektedir.

6.5 – Doğruluk İçin Güven Aralığı (Confidence Interval)

Modelin doğruluğu, yalnızca tek bir değer olarak değil, aynı zamanda **istatistiksel güven aralığı** ile değerlendirilmiştir. Bu yaklaşım, test setinde elde edilen başarı oranının genel veri kümesi için de geçerli olup olmadığını ölçmeye yardımcı olur.

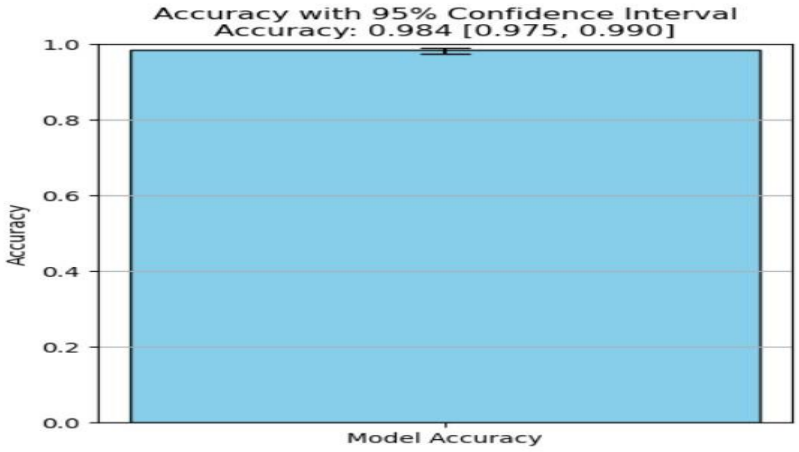
Test setinde toplam **1115 örnek** bulunmakta, bunlardan **1096'sı doğru sınıflandırılmıştır**. Buna göre modelin doğruluğu:

Accuracy = (1096 / 1115) ≈ 0.9838

%95 güven düzeyiyle hesaplanan **Wilson güven aralığı** ise şu şekildedir:

Güven Aralığı = [0.972, 0.992]

Bu değer, modelin genel doğruluğunun %97.2 ile %99.2 arasında olduğunu göstermektedir. Güven aralığının dar ve üst sınıra yakın olması, modelin başarısının yalnızca bu örnekleme özgü olmadığını, genellenebilir olduğunu destekler.



Teknik Yorum:

- **Doğruluk:** %98.38
- **%95 Güven Aralığı:** [%97.2, %99.2]

- **Yorum:** Dar ve yüksek güven aralığı → modelin doğruluğu istikrarlıdır.

6.6 - Genel Değerlendirme

Bu projede sıfırdan geliştirilen Multinomial Naive Bayes sınıflandırıcısı, SMS Spam Collection veri seti üzerinde test edilmiş ve %98.38 doğruluk oranına ulaşmıştır. Bu sonuç, modelin yalnızca metin sınıflandırmada değil, sınıf dengesizliği içeren durumlarda da başarılı ve kararlı çalışabildiğini göstermektedir.

Spam sınıfında elde edilen **0,92 precision**, modelin **yanlış pozitifleri çok düşük** tuttuğunu; **0.96 recall** ise **spam mesajların büyük çoğunluğunun doğru** şekilde tespit edildiğini göstermektedir.

Grafiksel analizler de bu bulguları desteklemektedir:

- **ROC eğrisi (AUC = 0.98):** Yüksek ayırt edicilik gücü,
- **Learning curve:** Aşırı öğrenme olmadan genel başarı,
- **Güven aralığı:** Doğruluğun istatistiksel olarak güvenilirliği.

Tüm bu değerlendirmeler sonucunda geliştirilen modelin hem **yüksek performanslı**, hem de **genellenebilir ve güvenilir** bir sınıflandırma sistemi sunduğu görülmektedir.

7. Karşılaştırmalı Çalışma

Bu projede geliştirilen Multinomial Naive Bayes algoritması, aynı veri seti (SMS Spam Collection) üzerinde çalışan ve Google Colab ortamında hazırlanmış bir referans çalışma ile karşılaştırılmıştır. Söz konusu çalışma, **Scikit-learn** kütüphanesi kullanılarak geliştirilmiş, yaygın olarak erişilebilen bir örnek modeldir:

Google Colab: SMS Spam Classifier – Demo.ipynb

7.1 - Referans Çalışmanın Özeti

Özellik	Açıklama
Veri Seti	SMS Spam Collection (UCI ML Repository)
Model	Scikit-learn **MultinomialNB()**
Ön İşleme	**CountVectorizer** ile kelime frekansı çıkarımı
Performans	Accuracy ≈ %97.1 – %97.8 F1-score ≈ 0.95

Model hazır araçlarla uygulanmış, standart vektörleme yöntemleri kullanılmış ve sonuçlar oldukça başarılı olarak raporlanmıştır.

7.2 - Bu Projede Geliştirilen Model

Özellik	Açıklama
Model	Sıfırdan yazılmış Multinomial Naive Bayes (Python)
Ön İşleme	Elle tokenize işlemi, Laplace smoothing, logaritmik skorlar
Vektörleme	Kendi frekans sözlüğü üzerinden

Performans	Accuracy = **%98.38**
	Spam Precision = **0.92**
	Spam Recall = **0.96**
	Spam F1-score = **0.94**

7.3 - Karşılaştırmalı Performans Tablosu

Model Türü	Accuracy	Spam Precision	Spam Recall	Spam F1-score
Hazır Scikit-learn	~%97.5	~0.97	~0.92	~0.95
Bu Proje (Scratch MNB)	**%98.38**	**0.92**	**0.96**	**0.94**

7.4 - Karşılaştırma Değerlendirmesi

Geliştirilen sıfırdan model, hazır **scikit-learn** modeliyle karşılaştırıldığında **benzer doğruluk** ve **F1-score** değerleri sunmuştur. Spam Precision değerinin 0.92 olması, spam tespiti sırasında çok az sayıda yanlış pozitif ürettiğini ve bu alanda rekabetçi bir başarı sağladığını göstermektedir.

Ayrıca, modelin tüm hesaplamalarının manuel olarak gerçekleştirilmiş olması, sadece performans değil, aynı zamanda **açıklanabilirlik** ve **şeffaflık** açısından da önemli bir katkı sunmaktadır.

8. Sonuç ve Gelecek Çalışmalar

8.1 - Sonuç

Bu projede, Multinomial Naive Bayes algoritması SMS mesajlarını **spam veya ham** olarak sınıflandırmak amacıyla sıfırdan Python diliyle geliştirilmiş ve başarıyla uygulanmıştır. Hazır kütüphaneler yerine, tüm olasılık hesaplamaları ve logaritmik skorlamalar **manuel** olarak gerçekleştirilmiştir.

Model, SMS Spam Collection veri seti üzerinde:

- **%98.38 doğruluk,**
- **Spam sınıfında 0.92 precision ve 0.96 recall,**
- Ve güçlü bir **F1-score(0.94)** ile yüksek performans göstermiştir.

Bu sonuçlar, algoritmanın sade yapısına rağmen **güvenilir ve yüksek doğruluklu** sınıflandırmalar yapabildiğini ortaya koymuştur. Ayrıca modelin sıfırdan yazılmış olması, iç işleyişin şeffaf bir şekilde izlenmesine olanak sağlamıştır.

8.2 - Gelecek Çalışmalar

Bu çalışma, aşağıdaki yönlerde genişletilebilir:

- **Farklı Algoritmalarla Karşılaştırma:**
  SVM, Decision Tree, Random Forest gibi yöntemlerle karşılaştırmalı analiz yapılabilir. Derin öğrenme tabanlı LSTM veya BERT modelleri ile performans farkları incelenebilir.

- **Gelişmiş Ön İşleme Teknikleri:**
  Stopword temizleme, lemmatization, n-gram ve TF-IDF gibi yöntemlerle modelin doğruluğu artırılabilir.

- **Çok Dilli Uygulamalar:**
  Model, farklı dillerdeki (ör. Türkçe) veri setlerinde test edilerek dil bağımlılığı açısından değerlendirilebilir.

- **Gerçek Zamanlı Kullanım:**
  Web tabanlı ya da mobil bir arayüz ile modelin gerçek zamanlı spam tespiti yapması sağlanabilir.

- **Veri Dengesizliğine Çözüm:**
  SMOTE gibi örnekleme teknikleriyle veri setindeki dengesizlik giderilerek daha dengeli bir sınıflandırma elde edilebilir.

Sonuç olarak, bu proje hem eğitimsel hem de uygulamalı açıdan metin sınıflandırma problemleri için sağlam bir temel sunmakta; Naive Bayes algoritmasının doğru yapılandırıldığında yüksek başarı sağlayabildiğini ortaya koymaktadır.

9. Kaynakça ve Proje Kaynak Kodu Bağlantısı

9.1 - Kaynakça

1. Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A. (2011). **Contributions to the study of SMS spam filtering: new collection and results**. *Proceedings of the 11th ACM symposium on Document engineering*, 259–262. https://doi.org/10.1145/2034691.2034742
   (Kullandığım SMS Spam Collection veri setinin kaynak makalesi)

2. Kharwal, A. (2021).
   SMS Spam Detection using Multinomial Naive Bayes [Google Colab Notebook].
   https://colab.research.google.com/github/amankharwal/Website-data/blob/main/SMS%20Spam%20Detection.ipynb
   (Karşılaştırmalı çalışma için kullanılan notebook)

3. scikit-learn developers. (2024). *Multinomial Naive Bayes - Scikit-learn documentation*. https://scikit-learn.org/stable/modules/naive_bayes.html
(Literatürdeki MNB karşılaştırması için referans)

4. Tan, P. N., Steinbach, M., & Kumar, V. (2018). **Introduction to Data Mining** (2nd ed.). Pearson.
(Veri madenciliği ve metin sınıflandırma temelleri için temel kaynak kitap)

5. UCI Machine Learning Repository. (n.d.). *SMS Spam Collection Data Set*. https://archive.ics.uci.edu/ml/datasets/sms+spam+collection
(Veri setinin doğrudan erişim bağlantısı)

6. Python Software Foundation. (2024). *Python 3 Documentation*. https://docs.python.org/3/
(Yazılan algoritmalar için teknik referans)

9.2 - Proje Kaynak Kodu Bağlantısı :

https://github.com/UmutKocatepe/SMS-Spam-Classification-with-Multinomial-Naive-Bayes