Gamze Balaban

Süreç

- ML Pipeline
- Paper Okumak
- EDA
- Makin Öğrenme Modelleri
- Transformer
- Bert
- Sentiment Analysis
- Değerlendirme Metrikleri
- API Kullanımı
- Clustering

ML Pipeline

1- Veri Toplama
2- Veri Temizleme ve Ön İşleme
3- EDA(Keşifsel veri analizi)
4- Feature Engineering (embedding ,encoding)
5- Veri Bölme
6- Model seçimi ve Eğitimi
7- Model Değerlendirme
8- Model Tuning
9- Model Deploy(API)

Paper Okumak

1- Abstract : Makalenin amacı ve temel bilgiler bu kısımda yer alır. İlk okunması gereken kısım burasıdır.

2- Introduction : Burada problem tanımı , makalenin katkısı ve mevcut yapılardan nasıl bir fark ortaya koyduğu belirtilir.

3- Background : Önceki yöntemlerle karşılaştırma kısmıdır. Buraya ilk aşama için kısaca göz atılır.

4- Model Architecture : Kullanılan algoritma, model, mimarı detaylı bir şekilde anlatılır. Bu kısım da sayısal verilerde boğulmadan ana mantığı kavramaya odaklanılmalıdır.

5- Experiments : Kullanılan veri seti,ayarlar ve parametreler burada yer alır. Yine sayısal kısımlara fazla odaklanmadan karşılaştırmalı tablolara bakılmalıdır.

6- Results : Performans metrikleri yorumlanır ve diğer yöntemlerle kıyaslama yapılır. Bu kısım bize makaledeki yöntemin eski yöntemlerden ne kadar iyi ya da kötğ olduğunu gösterir.

7- Conclusion : Son kısıma da yine kısaca göz gezdirilir ve gelecek için nasıl kullanılmasının planlanıldığı öğrenilir.

EDA

Veriyi anlamak için çok önemli bir adımdır.
Veri Tipi Analizi
Eksik Değer Analizi
Dağılım Analizi
Korelasyon Matrisi
Aykırı Değer Tespiti (Boxplot, Z-score,IQR)

Makine Öğrenmesi Modelleri Değerlendirme

Veri Ön İşleme adımlarını gerçekleştirdim. Burada bir temizleme fonksiyonu yazdım ve metinlerimi en yalın haline dönüştürdüm.
SentimentIntensityAnalyzer kullanılarak her yorumun duygu durumunu (Positive, Negative, Notr) belirledim.
compound skoru ile metnin etiketi belirlendi.
TfidfVectorizer kullanarak metin verisi sayısal verilere dönüştürdüm.
Farklı makine öğrenmesi modelleriyle eğitim yaptım.
Model eğitildikten sonra doğruluk skoru ve sınıflandırma raporu yazdırdım.

Lojistik Regresyon (LR) – En iyi performans: Doğrusal ilişkileri öğrenmede etkilidir ve metin verisi gibi yüksek boyutlu ve seyrek veri setlerinde iyi çalışır.

Naive Bayes – İyi performans: Hızlıdır ve genellikle yeterince iyi sonuçlar verir, ancak bağımsızlık varsayımı(her özellik birbirinden bağımsız) sınırlayıcı olabilir.

Random Forest – İyi performans: Karmaşık veri setlerinde iyi çalışır, ancak hesaplama maliyeti yüksektir ve overfitting riski vardır.

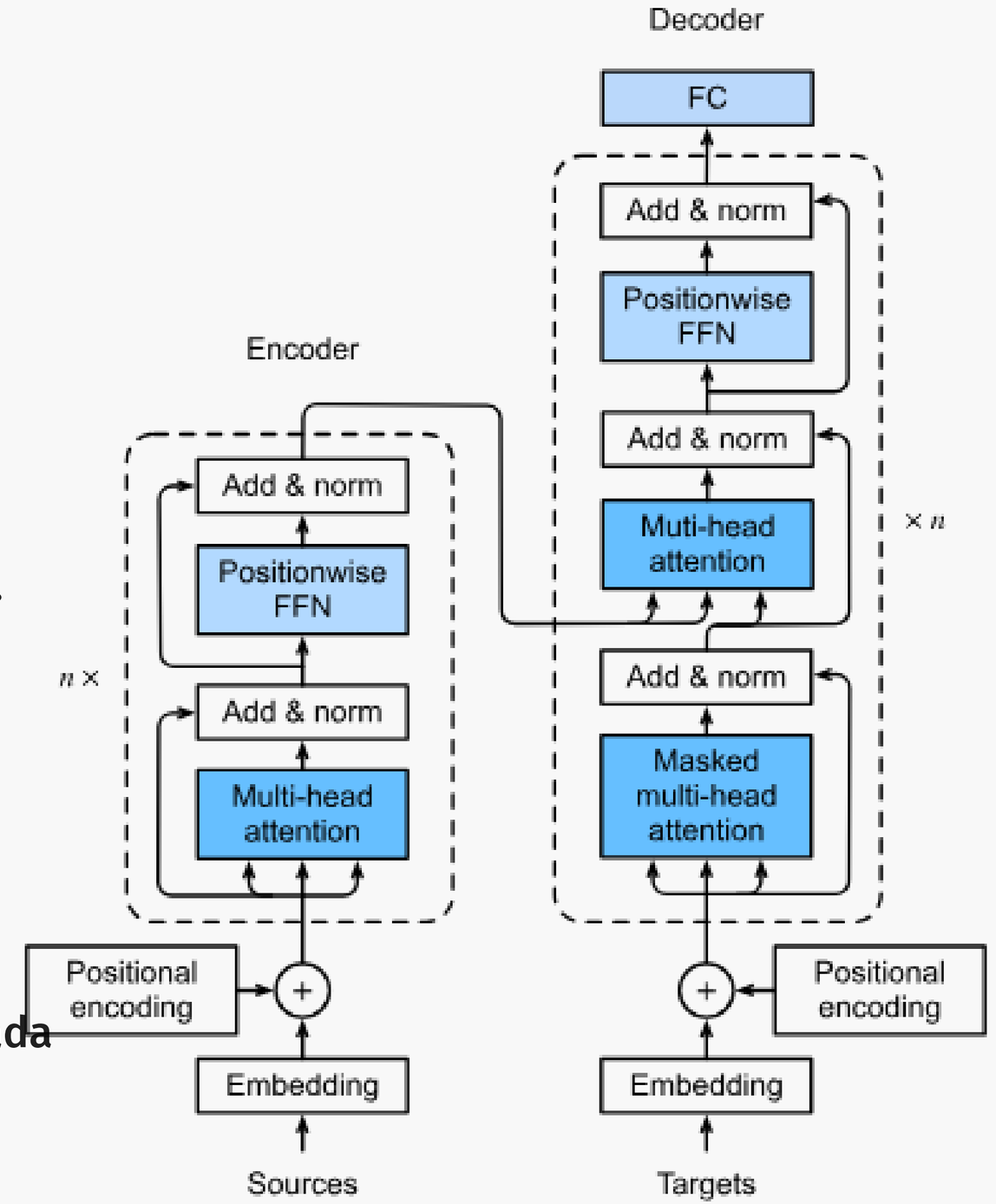
K-Nearest Neighbors (KNN) – En kötü performans: Veri boyutunun artmasıyla zayıflar.
Uzaklık ölçümü ve komşu sayısı gibi parametreler başarıyı doğrudan etkiler.
Metinlerde kelimeler arasındaki benzerlikler lineer değil, çok boyutlu ve karmaşık olabilir. Bu sebeple KNN zayıf kalır.
Fazla özelliğe sahip verilerde de her bir komşunun önemini belirlemek zorlaşır.

Transformer Modeli

Attention is All You Need
RNN, LSTM gibi sıralı (sequential) modellere kıyasla, veriyi paralel olarak işler.

🎯 Transformer Ne Yapar?
Metinlerdeki kelimeler arasındaki ilişkiyi anlamaya çalışır. (attention mekanizması)

Encoder ve Decoder Bloklarından oluşur.
Embedding-> kelimeler sayısal vektöre dönüştürülür.
Positinal Encoding-> her vektöre konum bilgisi eklenir.
Mutli-head attention-> Her kelimenin diğer kelimelerle ilişkisine bakar.(self attention)
Feed Forward Neural Network-> Her kelimenin bilgisi burada da işlenir.

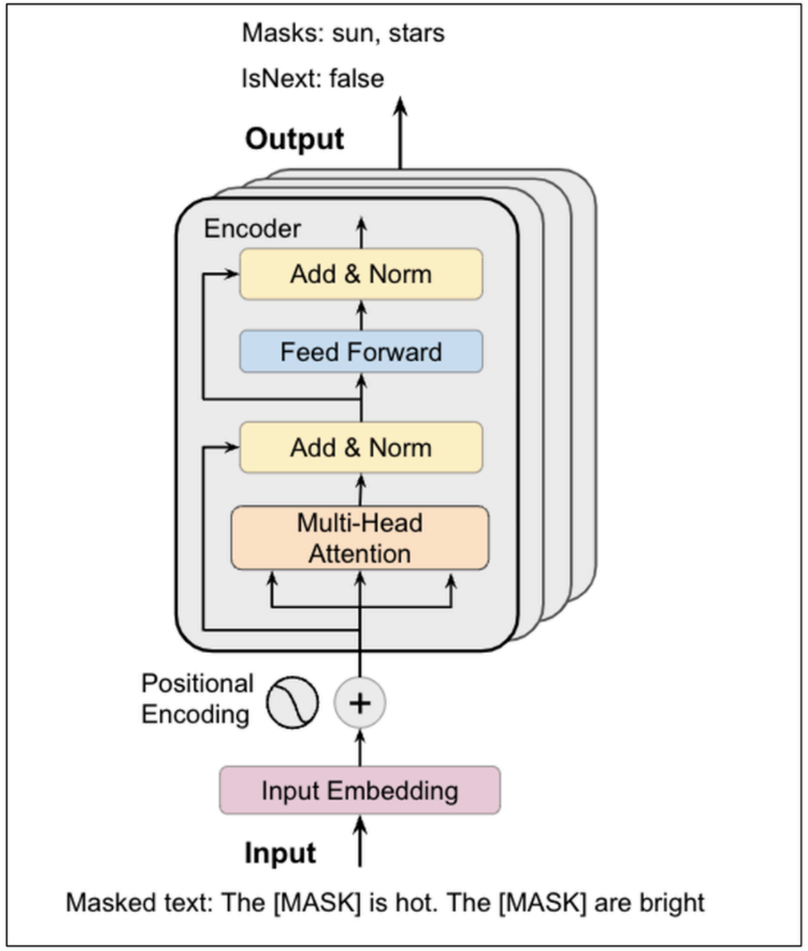


🎯 BERT Modeli

İki yönlüdür.
Pretrained + Fine-tuned : Önceden dev veriyle eğitilir (Wikipedia + BooksCorpus), ardından görevine özel olarak yeniden eğitilir (fine-tuning).
Masked Language Model (MLM)
NSP (Next Sentence Prediction)

[CLS] Cümle 1 [SEP] Cümle 2 [SEP]

- [CLS]: anlamsal özet (sınıflandırmada kullanılır).
- [SEP]: Cümle sınırı



Sentiment Analysis

1.Veri Temizleme:
rating sütunu, sayılarla sınırlı hale getirilir (örneğin, '3 star' ifadesinden sadece '3' sayısı çıkarılır).
review sütununda boş olan ya da sadece boşluk içeren yorumlar silinir.

Lemmatizasyon ve Stopwords Temizleme:
WordNetLemmatizer ile kelimeler lemmatize edilir .
İngilizce stopwords (yaygın kelimeler) stopwords listesinden çıkarılır.
contractions.fix() fonksiyonu ile kısaltmalar açılır (örneğin: "don't" → "do not").
Yalnızca anlamlı kelimeler (isalpha) ve stopwords içermeyen kelimeler bırakılır.

2. Etiketleme (Labeling)
Yorumlardaki rating değerlerine göre, 1-2 → Negatif (0), 3 → Nötr (1), 4-5 → Pozitif (2) etiketleri atanır.

3. Tokenizasyon
BERT Tokenizer: BertTokenizer kullanarak her bir yorumu BERT modelinin anlayacağı şekilde encode eder
(yani her bir kelimeyi sayısal verilere dönüştürür).

4. Veri Setinin Hazırlanması
TensorDataset ile veriler input_ids, attention_mask ve etiketler (labels) şeklinde birleştirilir.
Eğitim ve doğrulama verisi %80-%20 oranında ayrılır.

Sentiment Analysis

5. Model Tanımlaması
SentimentModel sınıfı, BERT'in önceden eğitilmiş modelini alır ve num_labels=3 (3 sınıf: Negatif, Nötr, Pozitif) ile yapılandırır.

6. Eğitim Ayarları
Optimizer ve Kayıp Fonksiyonu:
AdamW optimizasyon algoritması ve CrossEntropyLoss kayıp fonksiyonu yapılandırıldı.
Eğitim Fonksiyonu:
Model her epoch'da eğitim verisi ile güncellenir.
Eğitim sırasında, her batch için loss ve doğruluk hesaplanır.
Eğitimdeki doğruluk ve F1 skoru takip edilir.
Değerlendirme Fonksiyonu:
Doğrulama verisi üzerinde modelin doğruluğu ve F1 skoru hesaplanır.

7. Modelin Eğitimi
train() fonksiyonu ile model 3 epoch boyunca eğitilir.

8. Modelin Kaydedilmesi
Eğitilen modelin ağırlıkları Google Drive'a kaydedilir (torch.save()).

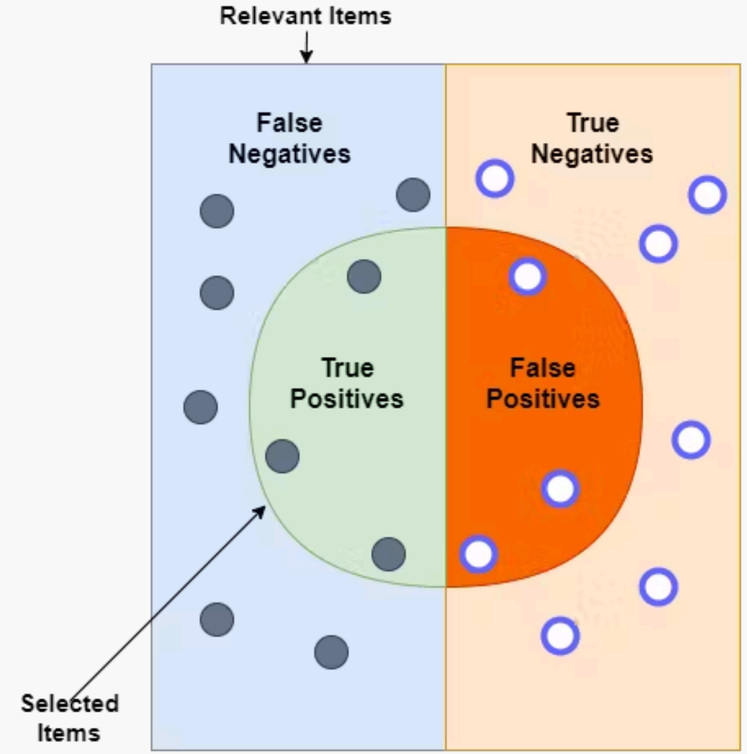
Değerlendirme Metrikleri

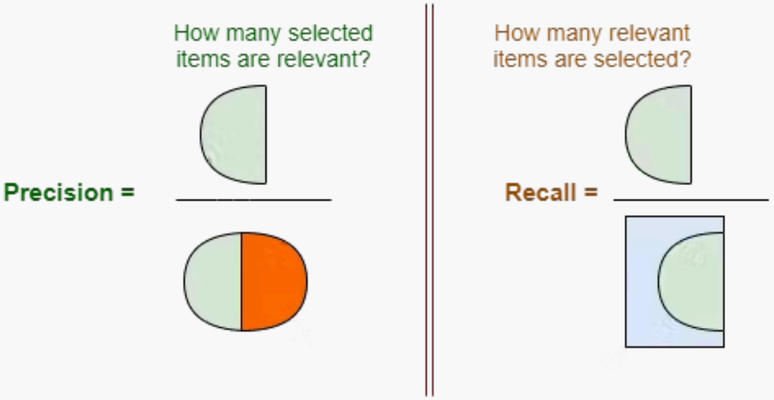
📌 **1. Accuracy (Doğruluk Oranı):**
Sınıflar dengeli ise kullanılır.

📌 **2. Precision (Kesinlik / Pozitif Tahmin Doğruluğu):**
"Pozitif" dediğim örneklerin kaçı gerçekten pozitif?

📌 **3. Recall (Duyarlılık / Hassasiyet):**
Gerçek pozitiflerin ne kadarını bulabildim?
*kanser tespiti

📌 **4. F1-Score (Harmonik Ortalama):**
Hem precision hem recall yüksekse F1 yüksek olur.

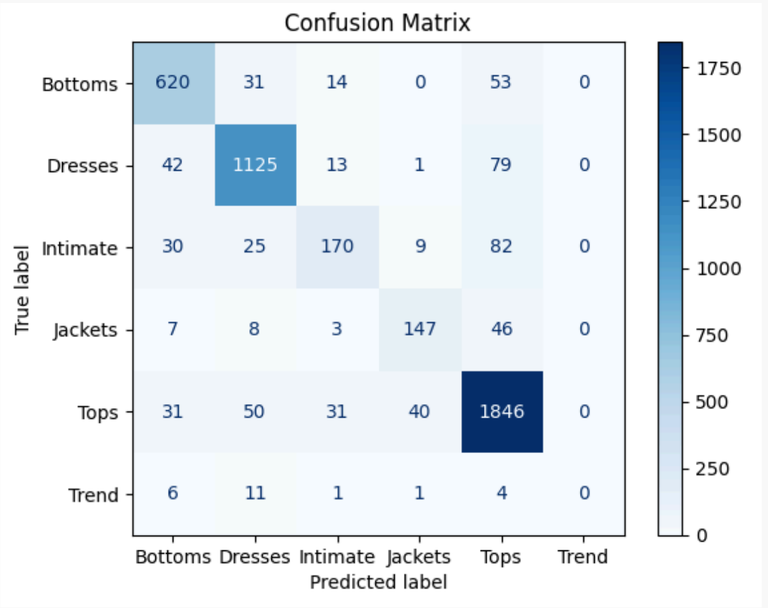




Değerlendirme Metrikleri

📌 5. Confusion Matrix (Karışıklık Matrisi)

Modelin her sınıfı nasıl tahmin ettiğini tablo şeklinde gösterir.

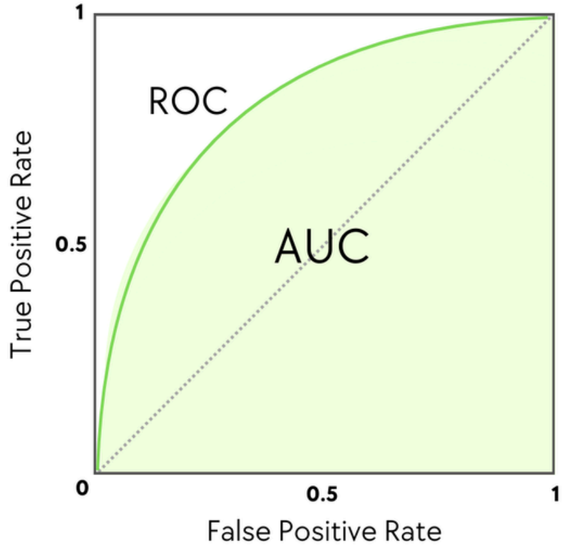


📌 6. ROC-AUC

Altındaki alan → 1'e ne kadar yakınsa model o kadar iyi.

📌 7. Log Loss (Logaritmik Kayıp)

Olasılık tahmininde kullanılır. Olasılık tahminlerinin doğruluğunu değerlendirir.



API Kullanımı

FASTAPI AVANTAJLARI

Starlette & Uvicorn sayesinde asenkron destekli, çok hızlıdır.

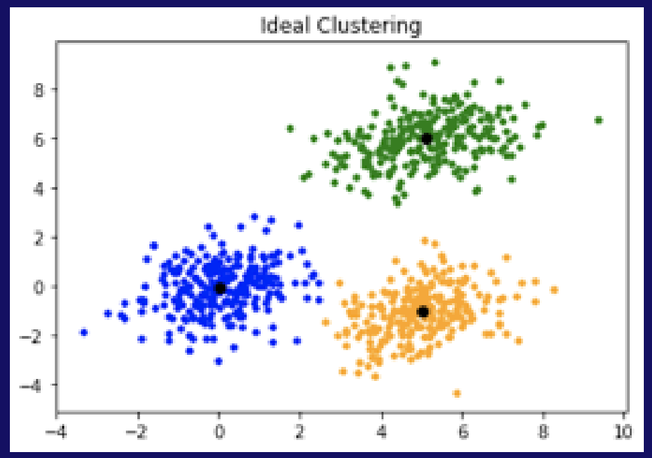
Otomatik Swagger Desteği -> API dökümantasyonu otomatik olarak oluşur.

Kolay Tip Tanımlama-> pydantic ile JSON girişleri kolayca tanımlanır

Makine öğrenmesi modelleriyle çok kolay entegre olur (sklearn, keras, transformers, torch)

Gerçek zamanlı servisler için uygundur

Clustering



Verilerdeki doğal olarak oluşan gruplamayı keşfeder.
Etiketsiz verilerde kullanılır.

Kmeans algoritmasının çalışma prensibi ;
Öncelikle her kümenin merkez noktasını veya ortalamasını temsil etmek üzere K adet nesne rastgele seçilir.
Kalan diğer nesneler, kümelerin ortalama değerlerine olan uzaklıkları dikkate alınarak en benzer oldukları kümelere dahil edilir.
Daha sonra, her bir kümenin ortalama değeri hesaplanarak yeni küme merkezleri belirlenir ve tekrar nesnelerin merkeze uzaklıkları incelenir. Herhangi bir değişim olmayıncaya kadar algoritma tekrarlamaya devam eder.

DBSCAN: benzer yoğunlukta kümeler içeren veriler içn iyidir.
DBSCAN algoritması, KMeans algoritmasının aksine, yoğunluk olarak kümeleri değerlendirmektedir.

Clustering Proje

Veri Ön işleme
SBert ile embedding oluşturma:
Her temizlenmiş yorum için SBert modeli kullanılarak embedding(sayısal vektör) çıkarılır.

Kümeleme(Clustering):
KMeans algoritması ile yorumlar kümelere ayrılır ve her kümeye bir etiket atanır.

KeyBERT ile Anahtar Kelime Çıkarımı:
Her küme için en anlamlı 3 anahtar kelime çıkartılır.

Otamatik Kategori Etiketleme:
Küme temsilcisi yorumlardan elde edilen embeddingler, önceden var olan kategorilerle karşılaştırılır ve her kümeye uygun kategori etiketlemesi yapılır.

Manual Etiketleme:
Wordcloud çıktıları ve gerçek etiketlere bakarak kümelere tahmini olarak manuel kategoriler eklenir.