

**TC  
İZMİR BAKIRÇAY ÜNİVERSİTESİ**

**2024-2025 BAHAR DÖNEMİ**

**BİL304  
YAPAY ZEKA DERSİ  
DÖNEM SONU PROJESİ**

**TARİFBOT - CHATBOT UYGULAMASI**

**Hazırlayanlar:  
220601054 -Begüm Keleş  
210601017 -Sueda Ahsen Gökalp**

## 1. Veri Seti Tanıtımı: Yemek Tarifleri Veri Seti (yemek\_tarifleri\_501.csv)

### Veri Setinin Kaynağı ve Amacı

Bu çalışmada kullanılan “**yemek\_tarifleri\_501.csv**” adlı veri seti, Türk mutfağına ait 500’ün üzerinde yemek tarifini içeren kapsamlı bir veri kümesidir. Veri seti, bir yemek tarifi öneri sisteminin eğitilmesi amacıyla hazırlanmış olup, tariflerin içeriğinde yer alan malzemeler, kategoriler, pişirme yöntemleri ve hazırlanış adımlarını içeren nitelikler barındırmaktadır. Bu yapı sayesinde hem makine öğrenmesi algoritmalarıyla sınıflandırma yapılabilmekte, hem de içerik bazlı öneri sistemleri geliştirilebilmektedir.

### Veri Setinin İçeriği

Veri seti toplamda **501 satır (örnek)** ve birden fazla özellik (özellik sayısı: 7) içermektedir. Ana başlıklar aşağıdaki gibidir:

- **foodname:** Tarifin adı (örnek: *Elmalı Kurabiye*)
- **category:** Tarife ait üst kategori (örnek: *Tatlı, Ana Yemek, Meze, Çorba*)
- **ingredients:** Tarif için gereken temel malzemeler (örnek: *elma, tarçın, un, şeker*)
- **materials:** Daha serbest biçimde girilmiş malzeme listesi (özgün yazımla)
- **tags:** Tarifin etiketleri (örnek: *glutensiz, vegan, pratik*)
- **steps:** Tarifin hazırlanış adımları
- **prep\_time:** Hazırlık süresi bilgisi (dakika cinsinden)

Bu alanlar sayesinde hem metinsel veriye hem de kategorik veriye dayalı analizler yapılabilmektedir.

### Veri Tipleri ve Temizleme Süreci

Veri setinde yer alan sütunlar farklı veri tiplerine sahiptir:

- **foodname, category, tags, materials, ingredients, steps:** metinsel veri
- **prep\_time:** sayısal veri

İlk aşamada boş/eksik veriler (NaN) analiz edilmiş ve veri bütünlüğü açısından kritik olan **foodname, category** ve **ingredients** sütunlarının eksik olması durumunda ilgili satırlar veri setinden çıkarılmıştır. Malzeme ve adım alanlarındaki yazım hataları, büyük-küçük harf uyumsuzlukları ve noktalama işaretleri düzenlenmiştir. **ingredients** sütunu tokenize edilerek makine öğrenmesi modelleri için uygun formata getirilmiştir.

## Kategorilerin Dağılımı

Veri setinde yer alan tarifler aşağıdaki 12 kategoriye ayrılmıştır:

- Tatlı, Ana Yemek, Meze, Çorba, Kahvaltı, Salata, Yan Yemek, İçecek, Ara Sıcak, Zeytinyağlı, Hamur İş, Atıştırmalık

En fazla tarif içeren kategori "Tatlı", en az tarif içeren ise "Salata" kategorisidir. Bu dengesizlik, modelleme sürecinde **SMOTE (Synthetic Minority Oversampling Technique)** gibi yöntemlerle dengelenmiştir.

## Etiketleme ve Filtreleme İmkanları

Veri seti aynı zamanda etiketleme (tag) sistemine sahiptir. Tarifler "vegan", "glutensiz", "az kalorili", "fırınsız", "15 dakikada hazır" gibi etiketlerle işaretlenmiştir. Bu yapı sayesinde filtre tabanlı öneri sistemi geliştirilmesi mümkün kılınmıştır. Özellikle doğal dil girdilerine uygun olarak kullanıcıdan gelen istekler bu etiketlere eşlenerek çalışmaktadır.

## Örnek Bir Kayıt

Aşağıda veri setinden alınmış örnek bir kayıt yer almaktadır:

```
foodname: Elmalı Kurabiye
category: Tatlı
ingredients: elma, tarçın, toz şeker, un, tereyağı, kabartma tozu
tags: tatlı, klasik, çay saati
steps: Elmaları rendeleyin, tarçın ve şekerini karıştırarak kavurun, hamuru hazırlayıp iç harcı ekleyin, pişirin
prep_time: 45
```

## Veri Setinin Güçlü Yönleri

- Gerçek yemek tariflerinden oluşturulmuş olması sayesinde **uygulamaya dönük öneri modellerine** uygunluk
- Hem metinsel hem kategorik veri yapısı ile **çoklu analiz türlerine uygunluk**
- Etiket ve içerik yapısı sayesinde **kural tabanlı ve öğrenmeye dayalı öneri sistemlerine entegrasyon imkânı**
- Türk mutfağına özgü öğeler barındırmasıyla birlikte kültürel veri çeşitliliği

## 2. Model Eğitimi ve Yapay Zeka Tabanlı Chatbot Geliştirme Süreci

Bu bölümde, yemek tarifleri veri seti kullanılarak geliştirilen yapay zeka tabanlı chatbot sisteminin kodlama süreci adım adım açıklanmıştır. Model, hem **Naive Bayes** hem de **Yapay Sinir Ağı (MLP)** tabanlı iki farklı yaklaşımla eğitilmiştir. Aşağıda sürecin her bir aşaması detaylı olarak sunulmuştur.

Ardından modelleme, görselleştirme, metin ön işleme ve derin öğrenme süreçleri için gerekli tüm kütüphaneler içe aktarılmıştır:

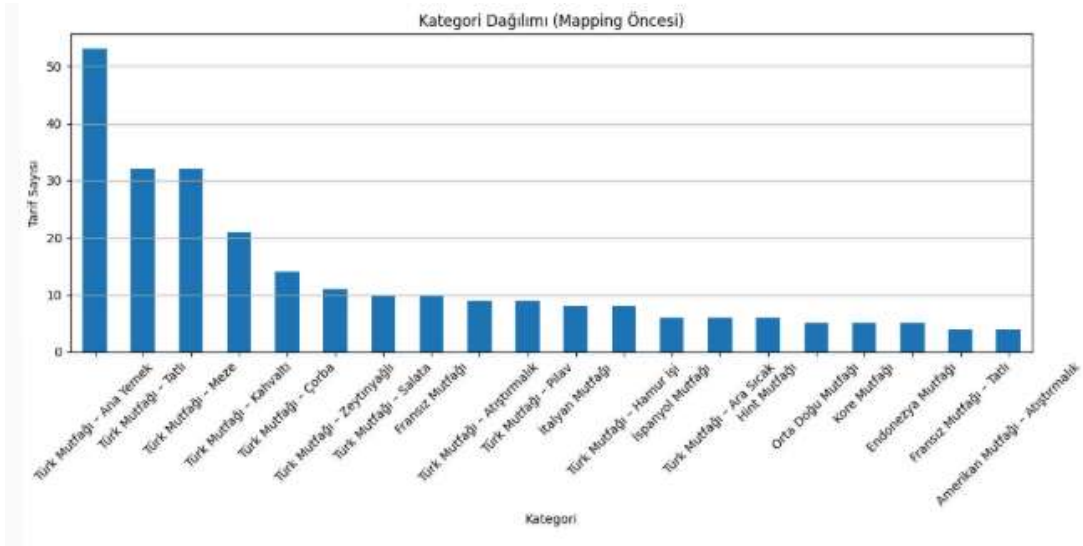
```
[2]: import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.multiclass import OneVsRestClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.regularizers import l2
from imblearn.over_sampling import RandomOverSampler
from collections import Counter
```

## Veri Setinin Yüklenmesi

```
print(df.columns.tolist())
```

```
['foodname', 'foodcategory', 'city', 'materials', 'description', 'hazirlama_suresi_dk', 'pisirme_suresi_dk', 'toplam_sure_dk', 'porsiyon_sayisi', 'zorluk', 'vejetaryen', 'vegan', 'etiketler', 'gluten', 'pisirme_yontemi']
```

Modelden önce veri kümesindeki **kategori dengesizliği** analiz edilmiştir. Özellikle **foodcategory** sütunundaki değerlerin dağılımı aşağıdaki gibi görselleştirilmiştir:



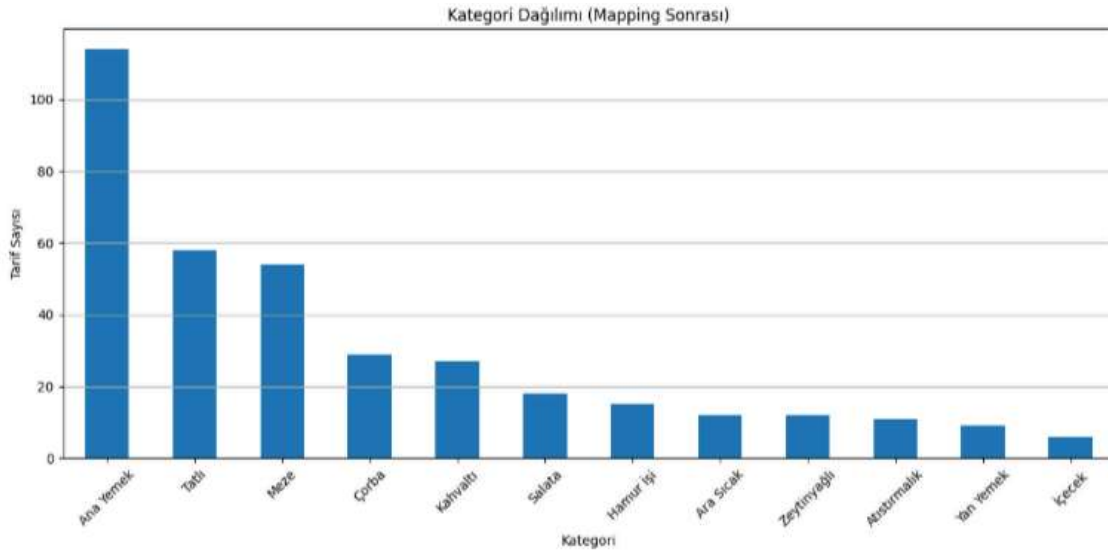
Bu grafik, bazı kategorilerin veri setinde çok fazla yer kapladığını, bazı kategorilerin ise oldukça az temsil edildiğini ortaya koymuştur. Bu durum, model eğitimi sırasında sınıf

dengesizliği sorununa yol açabileceğinden sonraki adımlarda **oversampling** ile çözülmüştür.

Veri setindeki kategoriler bazı durumlarda alt gruplar ya da farklı adlandırmalar içerdiğinden, bu isimler birleştirilerek 12 ana kategoriye dönüştürülmüştür. Bu işlem bir **kategori\_map** sözlüğü kullanılarak gerçekleştirilmiştir:

```
[4]:  
kategori_map = {  
    "Ana Yemek": "Ana Yemek", "Kebap": "Ana Yemek", "Mantı": "Ana Yemek", "Pide": "Ana Yemek",  
    "Ara Sıcak": "Ara Sıcak", "Döner": "Ana Yemek", "Lahmacun": "Ana Yemek", "Deniz Ürünü": "Ana Yemek",  
    "Tatlı": "Tatlı", "Meze": "Meze", "Sos": "Meze", "Turşu": "Meze", "Çorba": "Çorba",  
    "Kahvaltı": "Kahvaltı", "Gözleme": "Kahvaltı", "Ekmek": "Kahvaltı",  
    "Atıstırmalık": "Atıstırmalık", "Sokak Lezzeti": "Atıstırmalık", "Hamur İşi": "Hamur İşi",  
    "Börek": "Hamur İşi", "Salata": "Salata", "Zeytinyağlı": "Zeytinyağlı", "Makarna": "Ana Yemek",  
    "Pilav": "Yan Yemek", "Garnitür": "Meze", "İçecek": "İçecek",  
    "Vejetaryen": "Ana Yemek", "Başlangıç": "Ara Sıcak", "Fermente": "Meze",  
    "Çiğ Köfte": "Meze", "Dolma": "Ara Sıcak", "Sebze Yemeği": "Ana Yemek"  
}  
  
def extract_category(raw_cat):  
    if pd.isna(raw_cat): return []  
    parts = re.split(r"-|-", raw_cat)  
    return [parts[-1].strip()] if len(parts) >= 2 else [raw_cat.strip()]  
  
def extract_and_map_category(raw_cat):  
    extracted = extract_category(raw_cat)  
    kategoriler = [kategori_map.get(k.strip(), None) for k in extracted]  
    result = []  
    for k in kategoriler:  
        if isinstance(k, list): result.extend(k)  
        elif k: result.append(k)  
    return list(set(result))  
  
df["etiketler"] = df["foodcategory"].apply(extract_and_map_category)
```

Kategorilerin standart hale getirilmesi, sınıflandırma problemlerinde modelin daha doğru öğrenmesini sağlamaktadır.



## Çoklu Etiketleme Yapısının Kurulması ve Metin Verisinin Sayısal Temsili: TF-IDF

Her tarif birden fazla kategoriye ait olabileceğinden, etiketler çoklu olacak şekilde `MultiLabelBinarizer` sınıfı kullanılarak ikili formata dönüştürülmüştür ve Tariflerin `materials` sütununda yer alan malzemeler, metinsel ifade oldukları için `TfidfVectorizer` ile sayısal vektörlere dönüştürülmüştür:

```
mlb = MultiLabelBinarizer()
etiket_encoded = mlb.fit_transform(df["etiketler"])
df = pd.concat([df, pd.DataFrame(etiket_encoded, columns=mlb.classes_), axis=1]

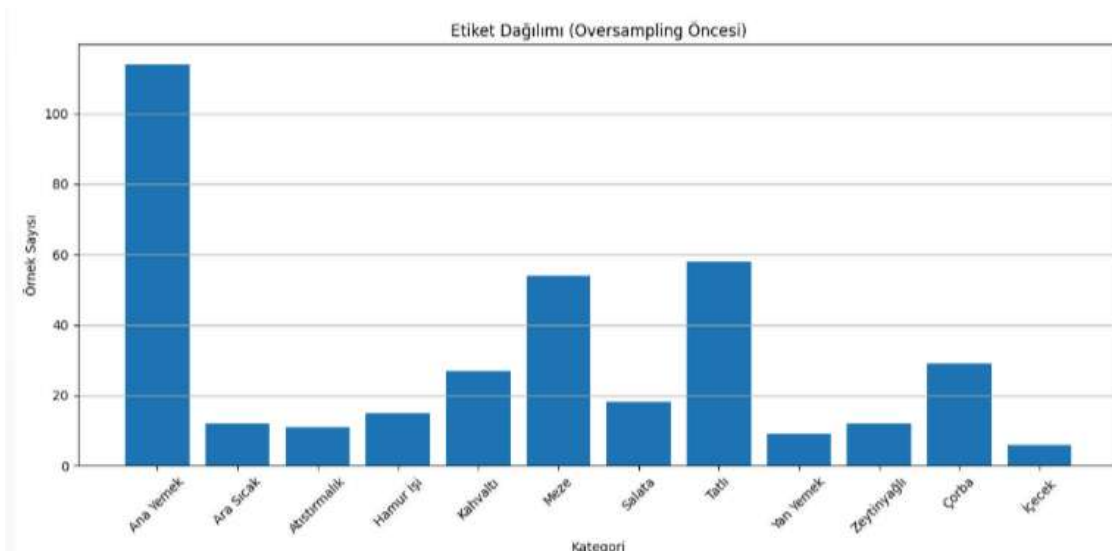
X_raw = df["materials"].fillna("")
y = df[mlb.classes_].values
tfidf = TfidfVectorizer()
X = tfidf.fit_transform(X_raw)
```

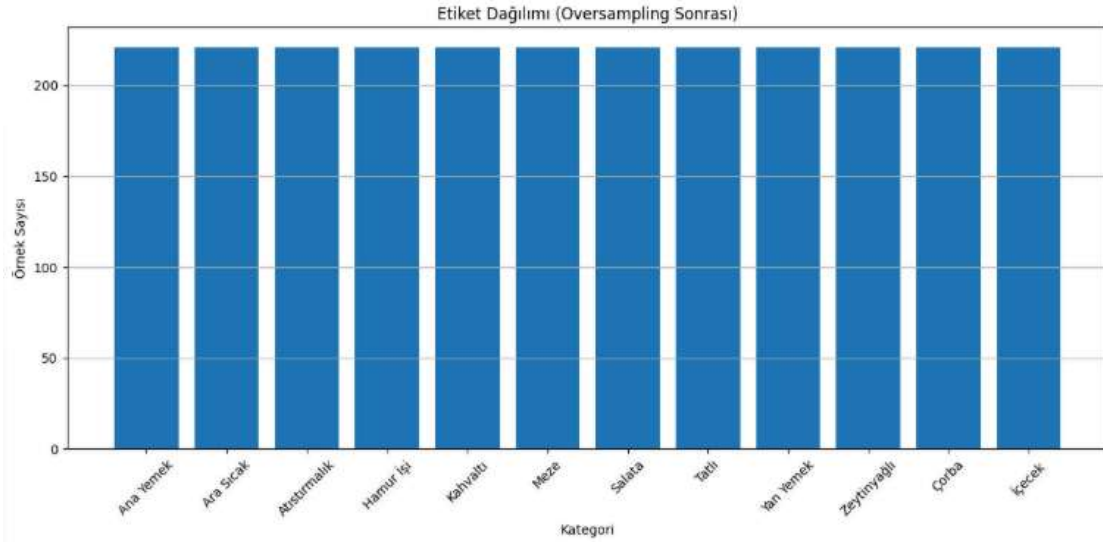
## Veri Dengesizliğinin Giderilmesi: Oversampling

Veri setinde bazı kategoriler çok az sayıda örnek içerdiğinden dolayı, eğitim verisi `RandomOverSampler` ile dengelenmiştir:

```
[9]: label_counts_before = np.sum(y, axis=0)
etiket_isimleri = mlb.classes_
```

Bu teknik, nadir görülen etiketleri çoğaltarak modelin bu sınıfları öğrenmesini kolaylaştırır.





## Eğitim ve Test Verilerinin Ayrılması

Modelin performansını değerlendirebilmek için veriler %80 eğitim, %20 test olacak şekilde ayrılmıştır:

```
[10]:  
ros = RandomOverSampler(random_state=42)  
X_ros, y_ros = ros.fit_resample(X, y)  
X_train_ros, X_test_ros, y_train_ros, y_test_ros = train_test_split(X_ros, y_ros, test_size=0.2, random_state=42)
```

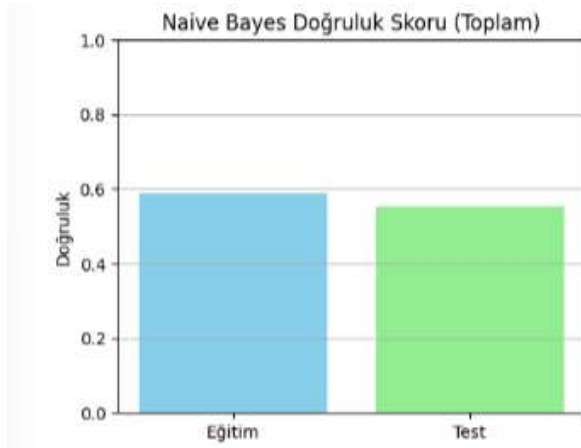
## 3. Model Eğitimi (Makine Öğrenmesi ve Derin Öğrenme)

### Naive Bayes Modeli

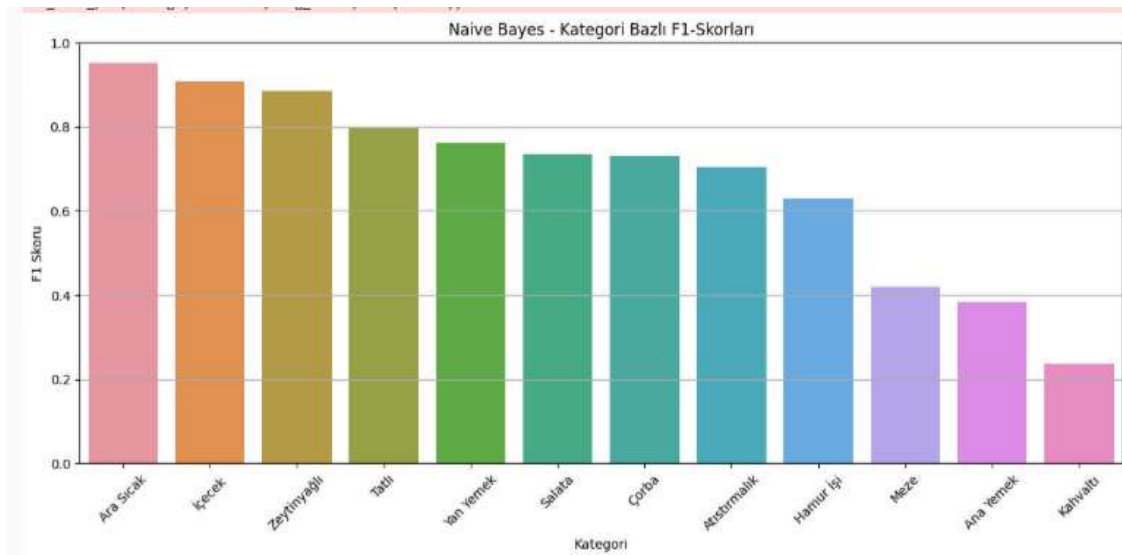
Metin verisiyle çalışırken sıkça kullanılan Multinomial Naive Bayes algoritması, çoklu etiketli sınıflandırma problemine uygun hale getirilmiştir. Bu işlem, `OneVsRestClassifier` kullanılarak her bir kategori için bağımsız sınıflandırıcılar oluşturularak gerçekleştirilmiştir:

Naive Bayes Sonuçları:				
	precision	recall	f1-score	support
Ana Yemek	0.91	0.24	0.38	41
Ara Sıcak	0.98	0.93	0.95	43
Atıstırmalık	0.92	0.57	0.71	42
Hamur İşi	0.88	0.49	0.63	47
Kahvaltı	0.64	0.15	0.24	48
Meze	0.86	0.28	0.42	43
Salata	0.96	0.60	0.74	42
Tatlı	0.88	0.73	0.80	49
Yan Yemek	0.91	0.66	0.76	44
Zeytinyağlı	0.90	0.88	0.89	40
Çorba	0.92	0.61	0.73	56
İçecek	1.00	0.83	0.91	36
micro avg	0.91	0.57	0.71	531
macro avg	0.90	0.58	0.68	531
weighted avg	0.89	0.57	0.67	531
samples avg	0.56	0.57	0.57	531

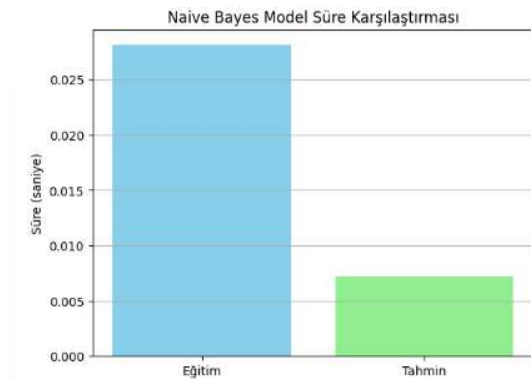
Eğitim ve test doğruluğu görselleştirilmiştir:



Her bir kategori için ayrı ayrı **F1-score** değerleri hesaplanarak modelin hangi kategoride daha iyi çalıştığı analiz edilmiştir:



Modelin eğitimi ve test sırasında geçen süre aşağıdaki şekilde ölçülmüş ve karşılaştırmalı olarak görselleştirilmiştir:





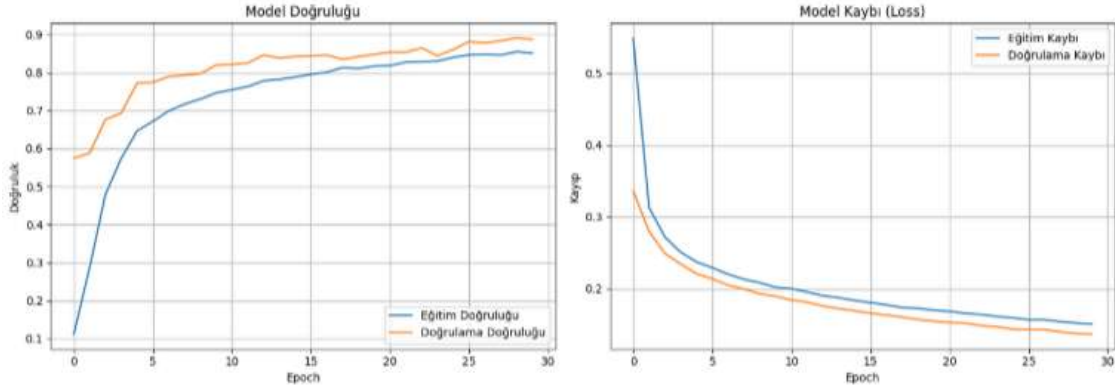
## MLP (Çok Katmanlı Perceptron) Modeli

Karmaşık ilişkileri öğrenebilmek için yapay sinir ağı (MLP - Multi-Layer Perceptron) modeli oluşturulmuştur. Modelin katman yapısı aşağıdaki gibidir:

```
model = Sequential([
    Dense(256, activation='relu', input_shape=(X.shape[1],), kernel_regularizer=l2(0.001)),
    Dropout(0.5),
    Dense(128, activation='relu', kernel_regularizer=l2(0.001)),
    Dropout(0.5),
    Dense(y.shape[1], activation='sigmoid')
])
```

**L2 regularizasyonu** ve **Dropout** katmanları ile overfitting azaltılmıştır. Çıkış katmanında **sigmoid** aktivasyonu çoklu etiket tahmini için uygundur. Model **Adam** optimizasyonu ve **binary\_crossentropy** kayıp fonksiyonu ile derlenmiştir.

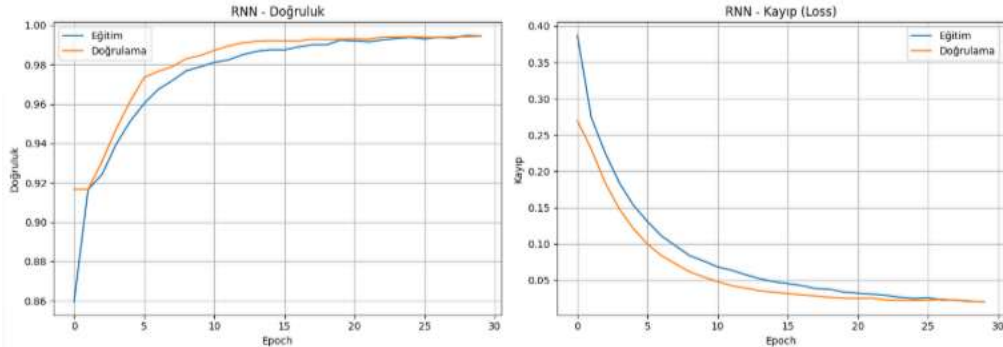
Eğitim doğruluğu ve kaybı ile doğrulama doğruluğu ve kaybı aşağıdaki gibi grafiklerle görselleştirilmiştir:



## RNN Modeli

Embedding + SimpleRNN + Dropout + Dense yapısı

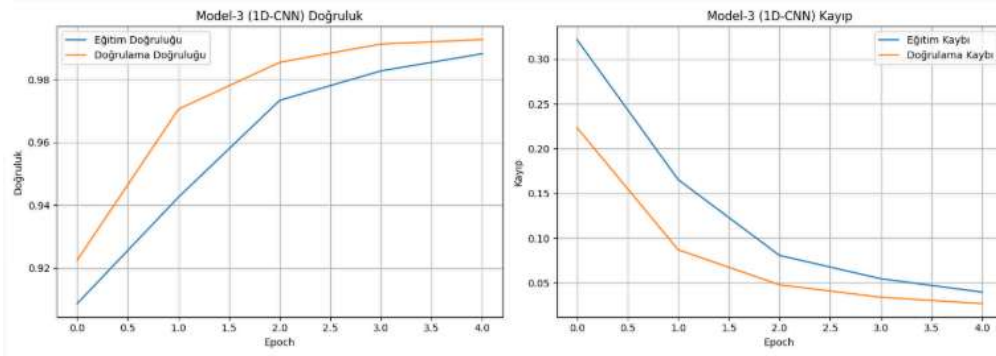
Daha çok sıralı verilerde tercih edilen bu model, yemek tariflerindeki malzeme sıralamalarını dikkate alarak öğrenir.



## 1D-CNN (Bir Boyutlu Evrişimsel Sinir Ağı) Modeli

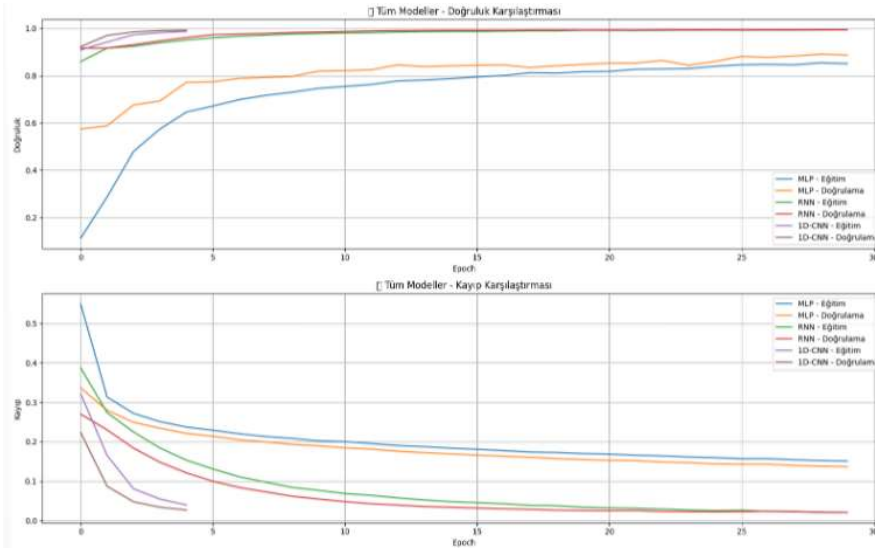
Embedding + Conv1D + GlobalMaxPooling1D + Dense

Bu model metin sıralamalarında lokal örüntüleri yakalamada başarılıdır.

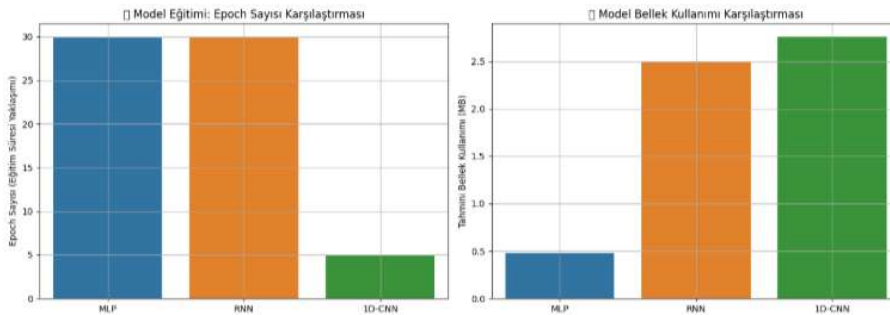


## Modellerin Karşılaştırması

Sonuç olarak, tüm modellerin doğruluk skorları aşağıda özetlenmiştir:



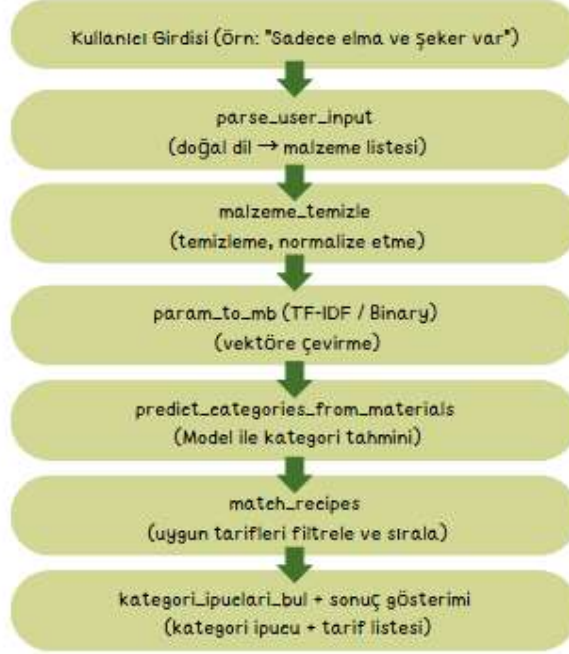
1D-CNN modeli en yüksek doğruluk oranını vermiştir; ancak MLP modeli eğitimi ve yorumlanabilirliği açısından daha basittir ve bu nedenle chatbot uygulamasında tercih edilmiştir.



## 4.Chatbot Sisteminin Tasarımı ve Çalışma Mantığı

Amaç, kullanıcının serbest metinle yazdığı tarif isteklerini doğal dil işleme ve makine öğrenmesi ile analiz ederek uygun bir tarif önermektir.

### Akış Şeması



- Chatbot ilk olarak kullanıcının yazdığı metinden malzeme, kategori ipuçları ve malzeme kısıtlılığı gibi bilgileri çıkarmaktadır: Kullanıcının cümlesinde geçen bilinen malzemeleri tanır. "sadece", "tek", "başka yok" gibi ifadelerle malzeme azlığına dair sinyalleri tespit eder. "kurabiye", "çorba", "salata" gibi kelimeleri analiz ederek kategori tahmini yapar.

```
malzemeler, kategori_ipuclari, malzeme_kisitli = parse_user_input(user_input)
```

- Çıkarılan malzeme listesi kullanılarak daha önce eğitilmiş olan **MLP modeli** ile ilgili kategoriler tahmin edilir: Bu tahminler, etiket skorları ile birlikte değerlendirilir. En az bir etiketin  $\geq 0.1$  puan alması durumunda öneri süreci devam eder.

```
tahminler = predict_categories_from_materials(malzemeler)
```

- Hem MLP'den gelen tahminler hem de kullanıcı cümlesinden gelen kategori ipuçları birlikte kullanılarak, veri setinde filtreleme yapılır:

```
yanit = öner_tarif(tahminler, kategori_ipuclari, malzemeler, malzeme_kisitli)
```

Filtreleme kuralları:

- Kullanıcının yazdığı malzemelerin çoğunu içeren tarifler önceliklidir.
- Eğer 3 veya daha fazla eksik malzeme varsa öneri yapılmaz.
- 1-2 eksik varsa uyarı verilir.
- Hiç eksik yoksa doğrudan öneri sunulur.
- Eğer kullanıcı doğrudan bir yemek adını yazdıysa, sistem bu adı veri setinde arar ve eşleşme bulursa öneriyi doğrudan sunar:

```
yanit_ismi = yemek_adi_ara(user_input)
```

## KULLANIM ÖRNEKLERİ:

Kullanıcı yalnızca bir malzeme (örneğin "elma") yazdığında, sistem bu malzemeyi içeren tarifleri tarar ve doğrudan adı ya da içeriği eşleşen en uygun tarifi önerir:

```
👤 TarifBot'a hoş geldin! Ne yapmak istediğini yaz (örnek: elma tarçınlı tatlı):  
👤 Sen: elma  
  
📌 Tarif Adına Göre Eşleşme Bulundu!  
📌 Tarif: Elmalı Kereviz Salatası  
📌 Malzemeler: kereviz, elma, ceviz  
🕒 Süre: 16 dk | 🍽️ Porsiyon: 2.0  
📌 Açıklama:  
Kereviz ve elma rendelenir, yoğurtla karıştırılır. Ceviz serpilerek servis edilir.
```

Kullanıcı birden fazla kelimeyle (örneğin "bana elmalı tatlı tarif") sorgu yaptığında, sistem hem malzemeleri hem de kategori ipuçlarını analiz ederek en uygun eşleşmeleri bulur ve eksik malzeme uyarısıyla birlikte öneri sunar:

```
👤 TarifBot'a hoş geldin! Ne yapmak istediğini yaz (örnek: elma tarçınlı tatlı):  
👤 Sen: bana elmalı tatlı tarifi  
1/1 ████████████████████ 0s 23ms/step  
⚠️ MLP kararsız kaldı, tahmin yapılmayacak.  
  
🔍 Elde edilen malzemeler: ['elma', 'tatlı']  
📌 Tahmin edilen kategoriler: []  
📌 Cümleden anlaşılan kategori ipuçları: ['Tatlı']  
  
⚠️ Bu tarif için eksik malzeme olabilir!  
Yine de deneyebilirsiniz:  
  
📌 Tarif: Tarte Tatin  
📌 Malzemeler: elma, şeker, tereyağı, hamur  
🕒 Süre: 43 dk | 🍽️ Porsiyon: 4.0  
📌 Açıklama:  
Karamelize elmalar fırın kabına yerleştirilir, üzerine hamur serilip pişirilir ve ters çevrilerek servis edilir.
```

Kullanıcının belirttiği birden fazla malzeme ("elmalı tarçınlı tarif") doğrultusunda, sistem bu malzemeleri içeren ve tahmin edilen kategoriyle uyumlu en uygun tarifi başarıyla önerir:

Sen: elmalı tarçınlı tarif  
1/1 0s 21ms/step  
MLP Tahmin Skorları:  
Tatlı: 0.47200000286102295  
Meze: 0.04600000008940697  
Ana Yemek: 0.03700000047683716  
Çorba: 0.032999999821186066  
Kahvaltı: 0.017000000923871994  
İçecek: 0.014000000432133675  
Zeytinyağlı: 0.009999999776482582  
Yan Yemek: 0.00800000037997961  
Ara Sıcak: 0.0010000000474974513  
Hamur İşi: 0.0010000000474974513  
Salata: 0.0010000000474974513  
Atıstırmalık: 0.0  
Elde edilen malzemeler: ['elma', 'tarçın']  
Tahmin edilen kategoriler: ['Tatlı']  
Cümleden anlaşılan kategori ipuçları: []  
Verdiğin malzemeleri içeren bir tarif buldum! İşte önerim:  
Tarif: Apple Pie  
Malzemeler: elma, tarçın, şeker, hamur, tereyağı  
Süre: 60 dk | Porsiyon: 4.0  
Açıklama:  
Elmalı iç harç hamura sarılır ve fırında pişirilerek servis edilir.

Kullanıcı yalnızca tek bir malzeme ("sadece elma") belirttiğinde, sistem bunu algılar, kullanıcıyı sınırlı tarif seçeneği konusunda uyarır ve yine de malzemeyi tam olarak içeren uygun bir tarif önerir:

Sen: sadece elma  
1/1 0s 23ms/step  
MLP Tahmin Skorları:  
Tatlı: 0.4350000023841858  
Meze: 0.09200000017881393  
Ana Yemek: 0.045000000178813934  
Çorba: 0.029999999329447746  
Kahvaltı: 0.02500000037252903  
İçecek: 0.008999999612569809  
Ara Sıcak: 0.006000000052154064  
Zeytinyağlı: 0.006000000052154064  
Hamur İşi: 0.0010000000474974513  
Salata: 0.0010000000474974513  
Yan Yemek: 0.0010000000474974513  
Atıstırmalık: 0.0  
Elde edilen malzemeler: ['elma']  
Tahmin edilen kategoriler: ['Tatlı']  
Cümleden anlaşılan kategori ipuçları: []  
Uyarı: Elinizde sadece 1 malzeme var. Tarifler sınırlı olabilir.  
Verdiğin malzemeleri içeren bir tarif buldum! İşte önerim:  
Tarif: Elma Dilim  
Malzemeler: elma  
Süre: 59 dk | Porsiyon: 4.0  
Açıklama:  
Elma dilimlenerek taze meyve olarak ya da hafif tatlı niyetine servis edilir.

Kullanıcı yalnızca tek bir malzeme belirttiğinde (örneğin “sadece domates”), sistem sınırlı tarif seçeneği nedeniyle uyarı verir ve yeterli eşleşme bulunamadığı durumda uygun bir tarif öneremez:

👤 Sen: sadece domates

1/1  0s 22ms/step

📊 MLP Tahmin Skorları:

Salata: 0.19300000369548798  
Çorba: 0.1850000023841858  
Ana Yemek: 0.10199999809265137  
Meze: 0.028999999165534973  
İçecek: 0.01600000075995922  
Atıstırmalık: 0.00800000037997961  
Kahvaltı: 0.00800000037997961  
Yan Yemek: 0.004999999888241291  
Zeytinyağlı: 0.004999999888241291  
Ara Sıcak: 0.0  
Hamur İşi: 0.0  
Tatlı: 0.0

🔍 Elde edilen malzemeler: ['domates']  
📁 Tahmin edilen kategoriler: ['Ana Yemek', 'Salata', 'Çorba']  
🗨 Cümleden anlaşılan kategori ipuçları: []  
⚠ Uyarı: Elinizde sadece 1 malzeme var. Tarifler sınırlı olabilir.  
❌ Uygun tarif bulunamadı.  
🔍 Malzeme: ['domates']  
📁 Kategori: ['Çorba', 'Salata', 'Ana Yemek']

Kullanıcı doğrudan bir yemek adı yazdığında (“fella”), sistem tarif adıyla eşleşme yaparak en ilgili tarifi detaylarıyla birlikte doğrudan önerir:

👤 Sen: fella

📌 Tarif Adına Göre Eşleşme Bulundu!

📌 Tarif: Fella Köftesi

📌 Malzemeler: ince bulgur, ırmik, salça, un, sarımsak, maydanoz

🕒 Süre: 50 dk | 🍽 Porsiyon: 4.0

📌 Açıklama:

Bulgur, ırmik, salça ve un yoğrularak küçük köfteler yapılır. Haşlanıp sarımsaklı domates sosuyla karıştırılır. Üz erine maydanoz serpilerek servis edilir.



Kullanıcının malzeme odaklı isteği ("mercimekli yemek tarifi") doğrultusunda, sistem hem malzemeyi hem de uygun kategorileri değerlendirerek en ilgili tarifi önerir:

Sen: mercimekli yemek tarifi  
1/1 0s 22ms/step

MLP Tahmin Skorları:

Meze: 0.2540000081062317  
Çorba: 0.2529999911785126  
Ana Yemek: 0.12999999523162842  
Kahvaltı: 0.02500000037252903  
Tatlı: 0.013000000268220901  
Salata: 0.008999999612569809  
Ara Sıcak: 0.007000000216066837  
Atıştırma: 0.004999999888241291  
İçecek: 0.004999999888241291  
Yan Yemek: 0.001000000474974513  
Zeytinyağlı: 0.001000000474974513  
Hamur İş: 0.0

Elde edilen malzemeler: ['mercimek']  
Tahmin edilen kategoriler: ['Ana Yemek', 'Meze', 'Çorba']  
Cümleden anlaşılan kategori ipuçları: []

Verdiğin malzemeleri içeren bir tarif buldum! İşte önerim:

Tarif: Koshari  
Malzemeler: makarna, pirinç, mercimek, domates sosu, soğan  
Süre: 58 dk | Porsiyon: 6.0  
Açıklama:  
Malzemeler ayrı ayrı haşlanır. Üzerine domates sosu ve kızarmış soğan eklenerek servis edilir.

Proje kaggle linki: <https://www.kaggle.com/code/begumkeles/chatbot>