

# Aygaz Yapay Zekaya Giriş Bootcamp Proje Sunum Dosyası

---

## Fashion MNIST ile Görüntü Sınıflandırma Projesi

**Projenin Amacı:** Bu projenin amacı, Keras kütüphanesi ile sağlanan Fashion MNIST veri setini kullanarak çeşitli makine öğrenmesi ve derin öğrenme algoritmaları ile görüntü sınıflandırma problemi üzerinde çalışmaktır. Bu sayede katılımcılar, görüntü verilerinin nasıl işlendiğini, modellerin nasıl oluşturulduğunu ve eğitildiğini, sonuçların nasıl değerlendirildiğini interaktif olarak öğreneceklerdir.

**Kullanılan Veri Seti:** Fashion MNIST veri seti, toplamda 70,000 adet 28x28 boyutunda gri tonlamalı görüntüden oluşmaktadır. Veri seti 10 farklı sınıfa ayrılmıştır ve her bir sınıfta 7,000 adet görüntü bulunmaktadır. Bu sınıflar, moda ürünlerini temsil etmektedir.

**Veri Ön İşleme:** Veri ön işleme adımları aşağıdaki gibi gerçekleştirilmiştir:

- Veri Bölme:** Veri seti, eğitim ve test verileri olarak ikiye ayrılmıştır:
  - Eğitim verisi: 60,000 görüntü
  - Test verisi: 10,000 görüntü
- Normalizasyon:** Görüntü verileri 0-255 aralığındaki piksel değerlerinden oluşmaktadır. Bu değerler 0-1 aralığına normalize edilmiştir.
- Görselleştirme:** Eğitim verisinden rastgele seçilen 10 görüntü görselleştirilmiştir.

```
python
Kodu kopyala
import tensorflow as tf
from tensorflow.keras.datasets import fashion_mnist
import matplotlib.pyplot as plt
(X_train, y_train), (X_test, y_test) = fashion_mnist.load_data()

X_train = X_train / 255.0
X_test = X_test / 255.0

plt.figure(figsize=(10, 10))
for i in range(10):
    plt.subplot(1, 10, i+1)
    plt.imshow(X_train[i], cmap='gray')
    plt.axis('off')
plt.show()
```

**Model Oluşturma ve Eğitim:** Bu projede Convolutional Neural Network (CNN) kullanılarak bir model oluşturulmuş ve eğitilmiştir. CNN, görüntü verileriyle çalışmak için oldukça etkili bir derin öğrenme algoritmasıdır.

```
python
Kodu kopyala
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.utils import to_categorical

# Etiketleri one-hot encode edin
y_train = to_categorical(y_train)
```

```
y_test = to_categorical(y_test)

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')
])

model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

history = model.fit(X_train[..., np.newaxis], y_train, epochs=10,
validation_data=(X_test[..., np.newaxis], y_test))
```

**Sonuçlar:** Eğitim ve test setleri üzerindeki model performansı aşağıdaki gibi değerlendirilmiştir:

- **Eğitim Doğruluğu:** 0.92 (örnek değer)
- **Test Doğruluğu:** 0.89 (örnek değer)

Değerlendirme metrikleri olarak doğruluk (accuracy) kullanılmıştır. Ek olarak, daha kapsamlı bir analiz için F1 skoru, kesinlik (precision) ve geri çağırma (recall) gibi metrikler de hesaplanabilir.

**Google Colab Dosyası:** Projenin kodları Google Colab üzerinde yazılmış ve .ipynb uzantılı bir dosya olarak kaydedilmiştir. Bu dosya, adım adım ilerlemeyi ve kodların nasıl çalıştığını gösteren yorum hücreleri ile zenginleştirilmiştir.

**GitHub Reposu Oluşturma:** Proje, GitHub'da merkezi bir depo oluşturularak paylaşılmıştır. Repo, projenin kod dosyalarını, veri setlerini ve projenin genel bilgilerini içeren README.md dosyasını barındırmaktadır. Teknik detaylar ve projenin nasıl gerçekleştirildiği hakkında açıklamalar .ipynb dosyasına eklenmiştir.

**Proje Sunumu:** Sunum, projenin amacı, kullanılan yöntemler, deney sonuçları ve tartışma bölümlerini içermektedir. Sunum dosyası özgün olup, projenin sonuçlarını detaylı bir şekilde ele almaktadır.