**200401020 Özgür Karli**

**210401091 Melih Durmaz**

**210401012 Muhammed Yusuf Köse**

**Proje Raporu: Futbolcu Verilerini Makine Öğrenmesi ile Sınıflandırma**

**1. Giriş**

Bu proje, futbolcu istatistiklerini içeren bir CSV dosyasından verileri alarak bir veritabanına kaydetmeyi, bu verileri çekip makine öğrenmesi modeline hazırlamayı ve sınıflandırma yapmak için bir Decision Tree modelini eğitmeyi amaçlamaktadır. Projenin sonunda yeni bir futbolcu verisi oluşturarak modelle tahmin yapmayı ve sonuçları değerlendirmeyi hedefliyoruz.

**2. Adımlar**

**2.1. ORM Yapılandırması ve Veritabanı Hazırlığı**

**Entity Class ve DbContext Tanımlama**

İlk olarak, Player sınıfı ve AppDbContext sınıfını tanımladık. Bu sınıflar Entity Framework Core kullanarak veritabanı işlemlerini gerçekleştireceğimiz yapı taşlarını oluşturur.

public class Player

{

public int PlayerID { get; set; }

public required string PlayerName { get; set; }

public int PlayerAge { get; set; }

public string PlayerPosition { get; set; }

public string PlayerSquad { get; set; }

public int PlayerMinutesPlayed { get; set; }

public float PlayerTouches { get; set; }

public float PlayerTackles { get; set; }

}

**AppDbContext.cs**

public DbSet<Player> Players { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer(@"Server=(localdb)\MSSQLLocalDB;Database=İleriVeriTabani;Trusted\_Connection=True;");

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

base.OnModelCreating(modelBuilder);

}

**Migration ve Veritabanı Güncelleme**

Terminal komutları ile migration oluşturduk ve veritabanını güncelledik:

**2.2. CSV Dosyasını Okuma ve Veritabanına Yazma**

Python kullanarak CSV dosyasını okuduk ve veritabanına yazdık:

SERVER = r'(localdb)\MSSQLLocalDB'

DATABASE = r'İleriVeriTabani'

USERNAME = r''

PASSWORD = r''

connection\_string = f'DRIVER={{ODBC Driver 17 for SQL Server}};SERVER={SERVER};DATABASE={DATABASE};UID={USERNAME};PWD={PASSWORD}'

conn = pyodbc.connect(connection\_string)

cursor = conn.cursor()

file\_path = r'C:\Users\ozzgur\Desktop\2022-2023 Football Player Stats.csv'

df\_csv = pd.read\_csv(file\_path, encoding='ISO-8859-1', delimiter=';')

**2.3. Veritabanından Veri Çekme ve Pandas DataFrame Oluşturma**

Veritabanından verileri çekip bir pandas DataFrame'e yükledik:

for index, row in df\_csv.iterrows():

    row = row.where(pd.notnull(row), None)

    query = """

        INSERT INTO Players(PlayerName, PlayerAge, PlayerPosition, PlayerSquad, PlayerMinutesPlayed, PlayerTouches, PlayerTackles)

        VALUES (?, ?, ?, ?, ?, ?, ?)

    """

    data = [

        row['Player'], row['Age'], row['Pos'], row['Squad'], row['Min'], row['Touches'], row['TklW']

    ]

**2.4. Decision Tree Modeli Oluşturma ve Değerlendirme**

Verileri eğitim ve test setlerine ayırarak, modelimizi oluşturup eğittik ve sonuçları değerlendirdik:

**2.5. Yeni Bir Vaka Yaratma ve Tahmin Etme**

Yeni bir futbolcu vakası yaratarak modelle tahmin yaptık:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Veriyi ölçeklendirme

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Decision Tree model oluşturma ve eğitme

dtc = DecisionTreeClassifier(random\_state=42)

dtc.fit(X\_train, y\_train)

cursor.execute("SELECT TOP 1 \* FROM Players ORDER BY PlayerID DESC")

new\_case = cursor.fetchone()

new\_case\_data = [new\_case.PlayerAge, new\_case.PlayerMinutesPlayed, new\_case.PlayerTouches, new\_case.PlayerTackles]

prediction = dtc.predict([new\_case\_data])

print(f"Yeni vakanın tahmini sonucu: {prediction[0]}")

**Sonuç**

Bu projede, futbolcu istatistiklerini içeren bir CSV dosyasından verileri okuyarak SQL veritabanına kaydettik, verileri çekip temizledik ve bir Decision Tree modelini kullanarak sınıflandırma yaptık. Yeni bir vaka oluşturarak modelin tahmin yeteneklerini test ettik. Modelin performansı ve tahmin sonuçları olumlu çıktı, ancak daha fazla veri ve özelliklerle modelin daha da geliştirilebileceğini gözlemledik.