SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

2023-2024

VERİ BİLİMİ ÖDEVİ

SAKARYA
ÜNİVERSİTESİ

Furkan Tataroğlu

G201210089

Giriş

Bu proje, Pokémon verisi üzerinde Exploratory Data Analysis (EDA) ve makine öğrenimi modeli oluşturma sürecini kapsamaktadır. Projede, verinin yüklenmesi, temel analizler, eksik değerlerin doldurulması, özellik mühendisliği ve tahmin modeli oluşturma adımları yer almaktadır.

Veri Setinin Yüklenmesi

Veri seti Kaggle'dan indirilmiş ve pandas kütüphanesi kullanılarak yüklenmiştir:

```
# İndirilen zip dosyasını açmak
with zipfile.ZipFile("pokemon-dataset.zip", 'r') as z:
    z.extractall("pokemon_dataset")

# Açılan CSV dosyasını okumak
pokemon_df= pd.read_csv("pokemon_dataset/Pokemon.csv")

# DataFrame'in ilk birkaç satırını göstermek
print(pokemon_df.head())
```

Verinin ilk birkaç satırı ve temel istatistikleri incelenmiştir:

```
# Temel İstatistikler ve Eksik Değerler
print(pokemon_df.describe())
print(pokemon_df.isnull().sum())
```

type2 sütununda eksik değerler bulunmuştur ve bu eksik değerler 'None' ile doldurulmuştur:

```
# 'type2' eksik değerlerini 'None' ile doldur
pokemon_df['type2'].fillna('None', inplace=True)
```

Kategorik Değişkenlerin Analizi

Kategorik değişkenlerin dağılımı bar grafikleri ile incelenmiştir:

```
# Kategorik Değişkenlerin Grafiksel Analizi
def bar_plot(variable):
    var = pokemon_df[variable]
    varValue = var.value_counts()
    plt.figure(figsize=(16,8))
    plt.bar(varValue.index, varValue, width=0.8)
    plt.xticks(varValue.index, varValue.index.values, rotation=90)
    plt.ylabel("Frequency")
    plt.title(variable)
    plt.tight_layout()
    plt.show()
    print("{}: \n {}".format(variable, varValue))

category1 = ["type1", "type2", "generation", "legendary"]
for c in category1:
    bar_plot(c)
```

Sayısal Değişkenlerin Analizi

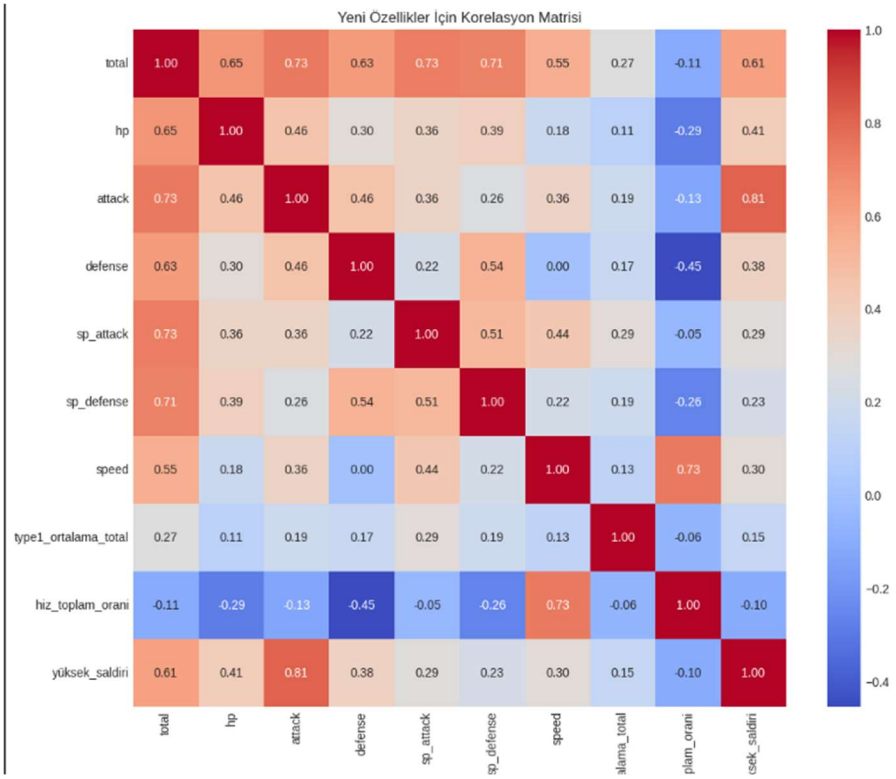
Sayısal değişkenlerin dağılımları histogramlar ile incelenmiştir:

```python
# Sayısal Değişkenlerin Grafiksel Analizi
def plot_hist(variable):
    plt.figure(figsize=(9,3))
    plt.hist(pokemon_df[variable], bins=50)
    plt.xlabel(variable)
    plt.ylabel("Frequency")
    plt.title("{} distribution with hist".format(variable))
    plt.show()

numericVar = ['total', 'hp', 'attack', 'defense', 'sp_attack', 'sp_defense', 'speed']
for n in numericVar:
    plot_hist(n)
```

Korelasyon Analizi ve Isı Haritası

Özellikler arasındaki korelasyonlar ve ısı haritası:



Doğrusal Regresyon modeli eğitilmiş ve performansı değerlendirilmiştir:

Tahmin Modeli Hazırlığı

```
# Kategorik sütunları dummy değişkenlerine dönüştür
pokemon_df = pd.get_dummies(pokemon_df, columns=['type1', 'type2'])

# Özellikleri ve hedef değişkeni tanımla
features = pokemon_df.drop(columns=['name', 'total', 'number', 'legendary'])
target = pokemon_df['total']

# Veriyi eğitim ve test setlerine ayır
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.3, random_state=42)
```

```
# Model Eğitimi: Doğrusal Regresyon
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
# Modeli başlat ve eğit
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
# Tahminler yap
y_pred_train = lr_model.predict(X_train)
y_pred_test = lr_model.predict(X_test)
# Modeli değerlendirme
train_mse = mean_squared_error(y_train, y_pred_train)
test_mse = mean_squared_error(y_test, y_pred_test)
train_r2 = r2_score(y_train, y_pred_train)
test_r2 = r2_score(y_test, y_pred_test)
# HATA ORANLARİ
print(f'Eğitim MSE: {train_mse:.2f}, R2: {train_r2:.2f}')
print(f'Test MSE: {test_mse:.2f}, R2: {test_r2:.2f}')
```

```
Eğitim MSE: 1.96, R2: 1.00
Test MSE: 0.06, R2: 1.00
```

Sonuçlar

- **Eğitim MSE (Mean Squared Error)**: 1.96
- **Eğitim R2 (R-squared)**: 1.00
- **Test MSE**: 0.06
- **Test R2**: 1.00

Bu sonuçlar, modelin eğitim ve test verilerinde neredeyse mükemmel performans gösterdiğini, yani çok düşük hata oranına sahip olduğunu ve veriyi çok iyi açıkladığını göstermektedir.