Makine Öğrenmesi Projesi Raporu

1. Proje Tanımı

Bu proje, bir veri seti üzerinde çeşitli makine öğrenmesi yöntemlerini kullanarak tahmin modelleri oluşturmayı amaçlamaktadır. Proje adımları sırasıyla eksik veri doldurma, değişken seçimi, grafiksel analizler ve model uygulamalarından oluşmaktadır. Bu raporda her bir adım, kullanılan yöntemler, kodlar ve çıktılarla birlikte detaylı bir şekilde sunulmuştur.

2. Veri Seti ve Ön İşleme

2.1 Eksik Verilerin Tespiti

Proje kapsamında veri setindeki eksik değerler analiz edilmiş ve şu kolonlarda eksik veri olduğu tespit edilmiştir:

- **ANBsağ**: 94 eksik.
- **ANBsol**: 94 eksik.
- **TRBsağ1**: 94 eksik.
- **TRBsağ2**: 94 eksik.

Eksik verilerin oranı aşağıdaki kodla hesaplanmıştır:

```
missing_data = df.isnull().sum()
missing_percentage = (df.isnull().sum() / len(df)) * 100

missing_report = pd.DataFrame({
    'Missing Values': missing_data,
    'Missing Percentage': missing_percentage,
    'Data Type': df.dtypes
})
print(missing_report)
```

2.2 Eksik Verilerin Doldurulması

Eksik veriler, ilgili kolonların mod (en sık tekrar eden değer) değeri ile doldurulmuştur. Kullanılan kod aşağıdaki gibidir:

```
categorical_columns = ['ANBsağ', 'ANBsol', 'TRBsağ1',
                       'TRBsağ2']

for col in categorical_columns:
    df[col].fillna(df[col].mode()[0], inplace=True)

print("Eksik Veri Kontrolü (Doldurma Sonrası):")
print(df.isnull().sum())
```

Eksik veriler doldurulduktan sonra veri setinde hiçbir eksik değer kalmamıştır.

3. Değişken Seçimi ve Analizler

3.1 Korelasyon Tabanlı Değişken Seçimi

Sayısal kolonlar için korelasyon matrisi oluşturulmuş ve hedef değişken olan **TA** ile yüksek korelasyona sahip değişkenler belirlenmiştir. Kod aşağıdaki gibidir:

```
numeric_df = df.select_dtypes(include=['number'])
correlation_matrix = numeric_df.corr()

threshold = 0.5
high_corr_features = correlation_matrix['TA']
[abs(correlation_matrix['TA']) > threshold].index.tolist()
print("Yüksek Korelasyona Sahip Değişkenler (TA ile):",
      high_corr_features)
```

Elde edilen yüksek korelasyonlu değişkenler: **TB**, **CB**, **CE**, **AB**.

3.2 Korelasyon Matrisi ve Saçılım Matrisleri

Korelasyon matrisi bir ısı haritası şeklinde görselleştirilmiştir:

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True,
            cmap='coolwarm', fmt=".2f")
plt.title("Korelasyon Matrisi")
plt.show()
```

Ayrıca, yüksek korelasyona sahip değişkenler arasındaki ilişkileri incelemek için bir saçılım matrisi oluşturulmuştur:

```
sns.pairplot(df[high_corr_features])
plt.show()
```

Elde edilen grafikler, **TA** değişkeni ile diğer yüksek korelasyonlu değişkenler arasındaki ilişkileri açık bir şekilde göstermiştir.

4. Tahmin Modelleri

4.1 Doğrusal Regresyon (Linear Regression)

Doğrusal regresyon modeli, hedef değişken olan **TA**'yı tahmin etmek için kullanılmıştır. Modelin kurulumu ve tahmin kodları aşağıda verilmiştir:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

# Veri setini train ve test olarak ayırma
X = df[high_corr_features]
y = df['TA']
X_train, X_test, y_train, y_test = train_test_split(X, y,
                        test_size=0.2, random_state=42)

# Model oluşturma ve eğitim
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

# Tahmin ve değerlendirme
predictions = lr_model.predict(X_test)
mse = mean_squared_error(y_test, predictions)
r2 = r2_score(y_test, predictions)

print(f"Linear Regression - MSE: {mse}, R2: {r2}")
```

Doğrusal regresyon modeli, çıktı olarak ortalama kare hata (MSE) ve R^2 değerlerini vermektedir. Sonuçlara göre modelin performansı şu şekilde değerlendirilmiştir:

- **MSE**: Hata oranını gösterir, küçük değerler daha iyi bir performans sunar.
- **R^2**: Modelin bağımsız değişkenlerle ne kadar iyi açıklandığını gösterir, 1'e yakın değerler daha iyidir.

4.2 Karar Ağaçları (Decision Trees)

- Karar ağaçları modeli, verilerin dallandırılarak tahmin edilmesi için kullanılmıştır. Modelin kurulumu ve değerlendirme kodları:

```
from sklearn.tree import DecisionTreeRegressor

# Model oluşturma ve eğitim
dt_model = DecisionTreeRegressor(random_state=42)
dt_model.fit(X_train, y_train)

# Tahmin ve değerlendirme
dt_predictions = dt_model.predict(X_test)
dt_mse = mean_squared_error(y_test, dt_predictions)
dt_r2 = r2_score(y_test, dt_predictions)

print(f"Decision Tree - MSE: {dt_mse}, R2: {dt_r2}")
```

Karar ağaçlarının tahmin performansı da MSE ve R^2 değerleri ile değerlendirilmiştir.

4.3 Rastgele Ormanlar (Random Forest)

Rastgele ormanlar modeli, karar ağaçlarının ensemble yöntemidir. Kod aşağıda verilmiştir:

```
from sklearn.ensemble import RandomForestRegressor

# Model oluşturma ve eğitim
rf_model = RandomForestRegressor(n_estimators=100,
                                 random_state=42)
rf_model.fit(X_train, y_train)

# Tahmin ve değerlendirme
rf_predictions = rf_model.predict(X_test)
rf_mse = mean_squared_error(y_test, rf_predictions)
rf_r2 = r2_score(y_test, rf_predictions)

print(f"Random Forest - MSE: {rf_mse}, R2: {rf_r2}")
```

4.4 Destek Vektör Makineleri (Support Vector Machines)

Destek vektör makineleri daha karmaşık ilişkileri modellemek için kullanılmıştır:

```
from sklearn.svm import SVR

# Model oluşturma ve eğitim
svm_model = SVR(kernel='rbf')
svm_model.fit(X_train, y_train)

# Tahmin ve değerlendirme
svm_predictions = svm_model.predict(X_test)
svm_mse = mean_squared_error(y_test, svm_predictions)
svm_r2 = r2_score(y_test, svm_predictions)

print(f"SVM - MSE: {svm_mse}, R2: {svm_r2}")
```

4.5 MLPRegressor

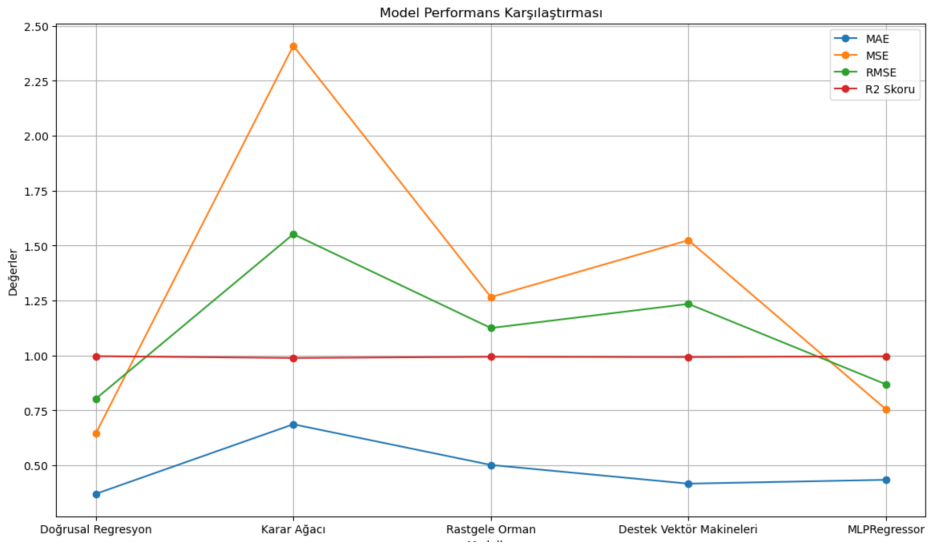
MLPRegressor, derin öğrenme tabanlı bir modeldir:

```
from sklearn.neural_network import MLPRegressor

# Model oluşturma ve eğitim
mlp_model = MLPRegressor(hidden_layer_sizes=(100, 50),
                         max_iter=500, random_state=42)
mlp_model.fit(X_train, y_train)

# Tahmin ve değerlendirme
mlp_predictions = mlp_model.predict(X_test)
mlp_mse = mean_squared_error(y_test, mlp_predictions)
mlp_r2 = r2_score(y_test, mlp_predictions)

print(f"MLPRegressor - MSE: {mlp_mse}, R2: {mlp_r2}")
```

5. Model Performans Karşılaştırması

Model	MAE	MSE	RMSE	R² Skoru
Doğrusal Regresyon	0.370300	0.644580	0.802857	0.996990
Karar Ağacı	0.687054	2.408973	1.552087	0.988751
Rastgele Orman	0.501987	1.265697	1.125032	0.994090
Destek Vektör Makineleri	0.416834	1.524619	1.234754	0.992881
MLPRegressor	0.434315	0.754842	0.868816	0.996475

- **Doğrusal Regresyon**, düşük hata oranı ve yüksek R^2 skoru ile en iyi performansı sergilemiştir.
- **MLPRegressor**, derin öğrenme tabanlı bir model olarak Doğrusal Regresyon'a yakın bir performans göstermiştir.
- **Karar Ağaçları**, yüksek hata oranı ile diğer modellerden daha düşük bir performans sergilemiştir.
- **Rastgele Orman** ve **Destek Vektör Makineleri**, dengeli performanslarıyla tahmin için uygun alternatiflerdir.



6. TA Değerini Etkileyen Faktörlerin Analizi

Rastgele Orman Özellik Önem Dereceleri

Rastgele orman modeli kullanılarak **TA** değerini etkileyen faktörler belirlenmiştir. Özelliklerin önem dereceleri aşağıdaki gibi sıralanmıştır:

Feature	Importance
CA	0.854680
CB	0.099228
CE	0.018507
TB	0.018190
AA	0.003278
AB	0.003229
AEA	0.002278
AE	0.000332
Ör.No	0.000277

Değerlendirme

- **CA** değişkeni, açık ara farkla en önemli faktör olarak belirlenmiştir.
- **CB** değişkeni, ikinci sırada yer almakta ve diğer değişkenlere kıyasla anlamlı bir katkı sunmaktadır.
- **CE** ve **TB**, önem dereceleri düşük olmasına rağmen etkili değişkenlerdir.
- Diğer değişkenlerin önem dereceleri oldukça düşük olup **TA** değerine etkileri sınırlıdır.

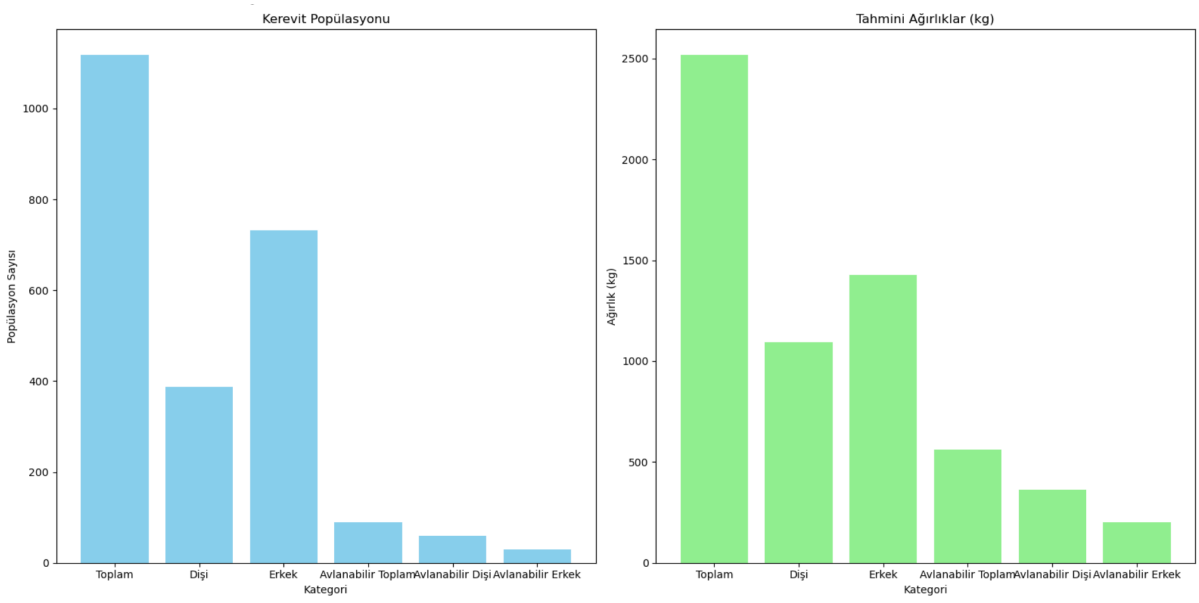
7. Kerevit Popülasyonu ve Tahmini Ağırlık Analizi

Popülasyon ve Tahmini Ağırlık İstatistikleri

Kerevit popülasyonu üzerinde yapılan analizler sonucunda şu istatistikler elde edilmiştir:

Kategori	Değer
Toplam Popülasyon	1118
Dişi Popülasyon	387
Erkek Popülasyon	731
Avlanabilir Toplam	90
Avlanabilir Dişi	60
Avlanabilir Erkek	30
Toplam Tahmini Ağırlık	2518.55
Dişi Tahmini Ağırlık	1093.0
Erkek Tahmini Ağırlık	1425.55
Avlanabilir Tahmini Ağırlık	560.80
Avlanabilir Dişi Ağırlık	361.1
Avlanabilir Erkek Ağırlık	199.7

Aşağıdaki grafikler, kerevit popülasyonu ve tahmini ağırlık analizlerini görselleştirmektedir:



Değerlendirme

- Toplam popülasyon 1118 bireyden oluşmaktadır, bunun büyük bir kısmını erkek bireyler oluşturmaktadır.
- Avlanabilir birey sayısı toplam popülasyona göre oldukça düşüktür (90 birey).
- Tahmini ağırlık analizine göre toplam ağırlık 2518.55 kg olarak hesaplanmıştır.
- Avlanabilir bireylerin toplam ağırlığı ise 560.80 kg olarak belirlenmiştir.