Veriyi Anlamak: Makine Öğrenimi Projelerinde İlk Adımlar

Veri bilimi ve makine öğrenimi projelerinde veriyi anlamak, projenin başarısında kritik bir rol oynar. Verinin kalitesi, modelin performansını doğrudan etkiler. Bu makalede, makine öğrenimi projelerinde veriyi anlama sürecinin temel adımlarını ve bu süreçte kullanılacak teknikleri ele alacağız. Örneğimizde, kalp hastalığı verilerini içeren 'heart.csv' veri setini kullanacağız.

1. Değişken Tanımlama

Veri setindeki değişkenlerin her biri tanımlanmalı ve anlaşılmalıdır. Değişkenler, bağımsız ve bağımlı değişkenler olarak sınıflandırılabilir. Örneğin, 'heart.csv' veri setinde 'age', 'sex', 'cp' gibi değişkenler bağımsız değişken, 'target' ise bağımlı değişken olarak tanımlanabilir.

2. Tek Değişkenli Analiz

Tek değişkenli analiz, her bir değişkenin dağılımını ve istatistiksel özelliklerini incelemeyi içerir. Bu adımda, ortalama, medyan, standart sapma gibi istatistiksel ölçütler hesaplanabilir ve verinin genel yapısı hakkında bilgi edinilebilir.

```
# Tek değişkenli analiz için istatistiksel ölçütlerin hesaplanması
def univariate_analysis(feature):
    print("Değişken: ", feature)
    print("Ortalama:", df[feature].mean())
    print("Medyan:", df[feature].median())
    print("Standart Sapma:", df[feature].std())
    print("Minimum Değer:", df[feature].min())
    print("Maksimum Değer:", df[feature].max())
    print("")

# Her bir sütun üzerinde tek değişkenli analiz yapma
for column in df.columns:
    univariate_analysis(column)
```

3. İki Değişkenli Analiz

İki değişkenli analiz, iki değişken arasındaki ilişkiyi incelemeyi içerir. Bu, özellikle bağımlı ve bağımsız değişkenler arasındaki ilişkileri anlamak için önemlidir. Örneğin, 'age' ve 'target' arasındaki ilişki bir dağılım grafiği ile görselleştirilebilir.

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

# 'age' ve 'target' arasındaki ilişkiyi görselleştirme
plt.figure(figsize=(8, 6))
sns.scatterplot(x='age', y='target', data=df, hue='target', palette='viridis')
plt.title('Age vs. Target')
plt.xlabel('Age')
plt.ylabel('Target')
plt.show()
```

Bu kod, yaş arttıkça kalp hastalığı riskinin nasıl değiştiğini gösterebilir.

4. Eksik Değer Düzenleme

Veri setinde eksik değerler bulunabilir. Bu eksik değerler, modelin performansını olumsuz etkileyebilir. Eksik değerleri belirlemek ve uygun yöntemlerle (örneğin, ortalama ile doldurma) düzenlemek önemlidir.

```
# Eksik değerlerin kontrolü
print("Eksik değerlerin sayısı:")
print(df.isnull().sum())

# Eksik değerleri ortalama ile doldurma
df_filled = df.fillna(df.mean())

# Eksik değerlerin kontrolü (doldurulmuş veri seti üzerinde)
print("\nEksik değerlerin sayısı (doldurulmuş veri seti):")
print(df_filled.isnull().sum())
```

5. Aykırı Veri Düzenleme

Aykırı veriler, veri setinde olağan dışı yüksek veya düşük değerlerdir ve modelin doğruluğunu bozabilir. Bu aykırı veriler belirlenmeli ve uygun şekilde düzenlenmelidir.

6. Değişken Dönüşümü

Veri setindeki bazı değişkenlerin dönüşüme ihtiyacı olabilir. Örneğin, kategorik değişkenler (cinsiyet, göğüs ağrısı türü gibi) sayısal formatlara dönüştürülebilir.

```
# One-Hot Encoding
df_encoded = pd.get_dummies(df, columns=['sex', 'chest_pain_type'])
```

7. Değişken Oluşturma

Yeni değişkenler oluşturmak, modelin daha iyi performans göstermesini sağlayabilir. Örneğin, yaş ve kolesterol seviyesinden bir risk skoru gibi yeni bir değişken oluşturulabilir.

Veriyi Görselleştirmek

Veriyi görselleştirmek, veriyi anlama sürecinin önemli bir parçasıdır. Görselleştirme teknikleri, verideki kalıpları, eğilimleri ve ilişkileri daha kolay görmemizi sağlar. Örneğin, 'heart.csv' veri setinde yaş ve hedef değişkeni arasındaki ilişkiyi daha iyi anlamak için dağılım grafiği kullanılabilir.

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

# Yaş ve hedef değişkeni (target) arasındaki ilişkiyi görselleştirme
plt.figure(figsize=(8, 6))
sns.scatterplot(x='age', y='target', data=df, hue='target', palette='viridis')
plt.title('Age vs. Target')
plt.xlabel('Age')
plt.ylabel('Target')
plt.show()
```

Veriler Arasındaki İlişkiyi İncelemek

Veriler arasındaki ilişkileri daha iyi tanımlamak için korelasyon katsayıları hesaplanabilir. Korelasyon katsayıları, iki değişken arasındaki doğrusal ilişkiyi ölçer.

```
# 'age' ve 'chol' arasındaki korelasyon katsayısı
correlation_age_chol = df['age'].corr(df['chol'])

# Korelasyon katsayısı tablosu oluşturma
correlation_table = pd.DataFrame(data={'Değişkenler': ['age', 'chol'], 'Korelasyon
print(correlation_table)
```

Bu kod, her bir çift değişken arasındaki korelasyon katsayılarını gösterir. Örneğin, 'age' ve 'chol' arasındaki korelasyon katsayısı, yaş ve kolesterol arasındaki ilişkinin gücünü gösterebilir.

Sonuç

Veriyi anlamak, makine öğrenimi projelerinin temel adımlarından biridir. Veriyi doğru şekilde analiz etmek ve görselleştirmek, modelin doğruluğunu ve genel performansını artırabilir. Bu makalede, veri analizinde kullanılan temel teknikler ve yöntemler 'heart.csv' veri seti örneğiyle ele alınmıştır. Bu adımlar, veriyi daha iyi anlamamızı ve modelin başarısını artırmamızı sağlar.