



Veri Madenciliği Müşteri Davranışlarının Analizi İle Perakende Sektöründe Bir İnceleme: Ödev Raporu

Proje Özeti:

Bu proje, perakende sektöründeki müşteri davranışlarını anlamak ve analiz etmek amacıyla veri madenciliği tekniklerini kullanmayı hedeflemektedir. Müşteri davranışlarının derinlemesine analizi, işletmelerin stratejik kararlar almasını sağlayarak müşteri memnuniyetini artırabilir ve satışları optimize edebilir.

Proje Kodu Ve Açıklamaları:

Kullanılan Kütüphaneler:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
```

Pandas (import pandas as pd): Veri analizi ve işlemleri için kullanılır. Veri setini yüklemek, düzenlemek ve işlemek için gereklidir.

sklearn (from sklearn...): Scikit-learn, makine öğrenimi modelleri ve metrikleri içeren bir kütüphanedir.

- train_test_split: Veriyi eğitim ve test setlerine bölmek için kullanılır.
- DecisionTreeClassifier: Karar ağacı sınıflandırıcısını oluşturmak için kullanılır.
- metrics: Model performansını ölçmek için metrikler içerir.

Veri Yükleme:

```
# Veri setini yükleme
data = pd.read_csv('musteri_veri_seti.csv')
```

pd.read_csv('dosya_adi.csv'): Pandas ile CSV formatındaki veri setlerini okumak için kullanılır. Bu projede, 'musteri_veri_seti.csv' adlı dosyadan veri seti yüklenir.

Veri Hazırlama:

```
# Bağımsız değişkenleri seçme
X = data[['Yaş', 'ÜrünMiktarı', 'ToplamTutar']]

# Hedef değişkeni seçme
y = data['ÜrünKategori']
```

Bağımsız değişkenler (Yaş, Ürün Miktarı, Toplam Tutar) ve hedef değişken (Ürün Kategorisi) belirlendi.

X: Bağımsız değişkenler olarak belirlenen 'Yaş', 'ÜrünMiktarı' ve 'ToplamTutar' sütunlarını içeren veri seti.

y: Hedef değişken olarak belirlenen 'ÜrünKategori' sütunu.

Veriyi Eğitim ve Test Setlerine Ayırma:

```
# Eğitim ve test verilerini ayırma
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

train_test_split: Veriyi eğitim ve test setlerine bölmek için kullanılır.

test_size=0.2: Verinin %20'sini test seti olarak ayarlar. Veri seti %80 eğitim ve %20 test olarak ayrıldı.

random_state=42: Veriyi rastgele bölerken kullanılan bir tohum değeridir, tekrarlanabilirliği sağlar.

Karar Ağacı Modelinin Oluşturulması ve Eğitimi:

```
# Karar ağacı modelini oluşturma
model = DecisionTreeClassifier()

# Modeli eğitme
model.fit(X_train, y_train)
```

Bu kısımda, DecisionTreeClassifier kullanılarak bir karar ağacı modeli oluşturulur ve fit fonksiyonu ile bu model eğitim verileriyle (X_train ve y_train) eğitilir.

Tahmin Yapılması ve Modelin Performansının Değerlendirilmesi:

```
# Test verileri ile tahmin yapma
y_pred = model.predict(X_test)

# Modelin performansını değerlendirme
accuracy = metrics.accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```

Eğitilmiş model, test verileri üzerinde predict fonksiyonu ile tahminler yapar. Daha sonra bu tahminler gerçek değerlerle (y_test) karşılaştırılarak modelin doğruluk oranı (accuracy) hesaplanır ve ekrana yazdırılır.

Tahmin Sonuçlarının Analizi:

```
# Tahmin edilen sınıfları içeren bir tablo oluşturma
prediction_table = pd.DataFrame({'Gerçek Ürün': y_test, 'Tahmin Edilen Ürün': y_pred})

# Tabloyu ekrana yazdırma
print("\nTahmin Tablosu:")
print(prediction_table)
```

Burada, gerçek ve tahmin edilen ürün kategorilerini içeren bir tablo (prediction_table) oluşturulur. Bu tablo, her bir test verisi için gerçek ve model tarafından yapılan tahminleri içerir.

En Sık Tercih Edilen Ürünün Belirlenmesi:

```
# En çok tercih edilen ürünü belirleme
en_cok_tercih_edilen_urun = prediction_table['Tahmin Edilen Ürün'].mode().iloc[0]
print(f"\nEn Çok Tercih Edilen Ürün: {en_cok_tercih_edilen_urun}")
```

Bu adımda, tahmin tablosu üzerinde en sık tercih edilen ürün kategorisi belirlenir.

Alışveriş Analizinin Yapılması:

```
# Alışveriş analizini belirleme
alisveris_analizi = data.groupby('ÜrünKategori').agg({'ToplamTutar': 'sum', 'ÜrünMiktarı': 'sum'})
print("\nAlışveriş Analizi:")
print(alisveris_analizi)
```

Bu kısımda, veri seti üzerinde ürün kategorilerine göre yapılan toplam tutar ve ürün miktarı analizi yapılır.

Bir Sonraki Ay İçin Ürün Tahmini:

```
# Bir sonraki ay için ürün tahminini yapma
# Gelecekteki verilere dayanmıyor, sadece modelin eğitildiği verilere göre tahmin
tahmin_edilen_urun = model.predict([[30, 2, 1500]]) # Örnek olarak yaş=30, ürün
print(f"\nBir Sonraki Ay İçin Tahmin Edilen Ürün: {tahmin_edilen_urun[0]}")
```

Eğitilmiş model kullanılarak, örnek bir giriş değeri üzerinde bir sonraki ay için ürün kategorisi tahmini yapılır. Bu kod, 'Yaş=30', 'ÜrünMiktarı=2', 'ToplamTutar=1500' değerlerine sahip bir girişin model tarafından işlenerek bir sonraki aydaki müşteri alışveriş kategorisinin tahmin edilmesini sağlar. Ancak bu tahmin, gerçek gelecekteki verilere dayanmaz. Sadece modelin eğitildiği veri seti üzerinden yapılan bir öngörüdür.

Sonuç:

```
Accuracy: 0.9375

Tahmin Tablosu:
Gerçek Ürün Tahmin Edilen Ürün
30 Elektronik Elektronik
0 Elektronik Elektronik
22 Giyim Giyim
31 Giyim Giyim
18 Elektronik Elektronik
28 Giyim Gıda
10 Gıda Gıda
70 Giyim Giyim
4 Elektronik Elektronik
12 Giyim Giyim
49 Gıda Gıda
33 Gıda Gıda
67 Elektronik Elektronik
35 Elektronik Elektronik
68 Elektronik Elektronik
45 Elektronik Elektronik

En Çok Tercih Edilen Ürün: Elektronik

Alışveriş Analizi:
ToplamTutar ÜrünMiktarı
ÜrünKategori
Elektronik 34000.0 55
Giyim 13850.0 64
Gıda 6360.0 89

Bir Sonraki Ay İçin Tahmin Edilen Ürün: Elektronik
```

Kodun çıktısı yandaki gibidir.

Bu adımlar, müşteri davranışlarını analiz etmek için kullanılmış ve makine öğrenimi algoritması olan karar ağacı sınıflandırıcısını kullanarak bu analizi gerçekleştirmiştir. Bu proje, perakende sektöründeki işletmelerin müşteri davranışlarını anlamasına ve stratejik kararlar almasına yardımcı olabilir. Veri madenciliği, müşteri tercihlerini anlamak ve gelecekteki eğilimleri öngörmek için güçlü bir araçtır.

Çalışma Ekibi:

Ahmet GÜR - 02200201053

Dilara KARATAŞ - 02200201044

Burak KUTLUK - 02200201040

Ayşe DEVEDEN - 02200201081

Ahmet TAKCI - 02200201055