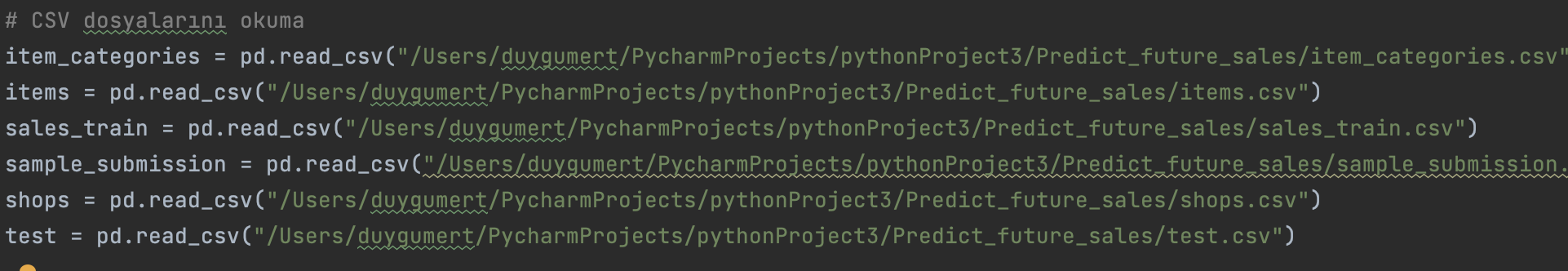
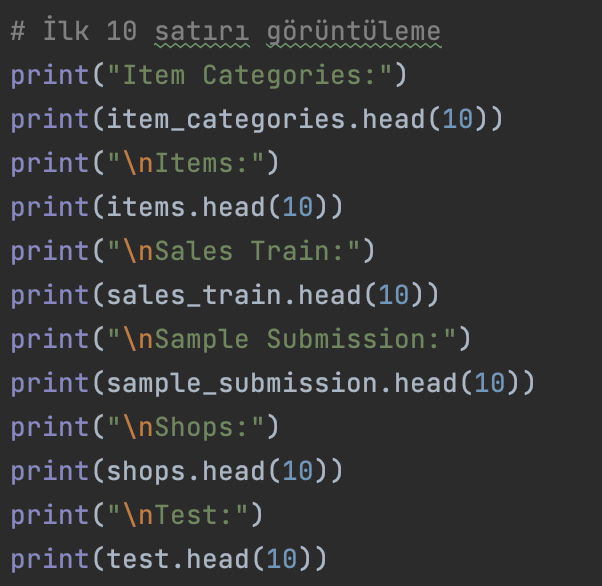
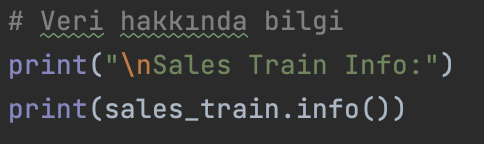
 YÜKSEK DÜZEY PROGRAMLAMA ÖDEVİ DUYGU MERT 20201317207

import matplotlib.pyplot as plt  
import pandas as pd

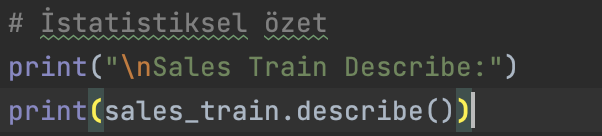
**Matplotlib**: Python'da veri görselleştirmesi için kullanılan en yaygın kütüphanedir. Kullanıcıların, verilerden grafikler, çizgi grafikleri, histogramlar, çubuk grafikler ve daha fazlasını oluşturmasına olanak tanır. Özellikle bilimsel verileri görselleştirmek için sıklıkla tercih edilir. pyplot modülü, bu grafiklerin hızlı ve kolay bir şekilde çizilmesini sağlayan fonksiyonlar sunar.

**Pandas**: Veri manipülasyonu ve analizinde en çok kullanılan kütüphanedir. Verileri okuma, yazma, temizleme, dönüştürme ve filtreleme gibi işlemleri kolayca yapmayı sağlar. DataFrame adı verilen veri yapısı, tablolarla çalışmayı çok daha verimli hale getirir. Pandas, özellikle büyük veri setlerinde hızlı analizler yapabilmek için tasarlanmıştır ve veri bilimi, finansal analiz ve zaman serisi analizi gibi birçok alanda yaygın olarak kullanılır.

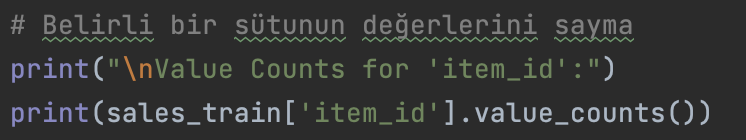
 Farklı veri kümelerini yükleyerek bir veri analizi ve modelleme sürecine başlamak için **pandas** kütüphanesini kullanır. Yüklenen dosyalar şunlardır: item\_categories.csv, ürünlerin kategorileri hakkında bilgi; items.csv, her bir ürünün detaylarını içerir; sales\_train.csv, geçmiş satış verilerini barındırır; sample\_submission.csv, tahmin formatı için örnek bir dosyadır; shops.csv, mağaza bilgilerini içerir; ve test.csv, gelecekteki satış tahminlerinin yapılacağı test verileridir. Bu dosyalar, verileri analiz etmek ve gelecekteki satışları tahmin etmek için gerekli veriyi sağlar. Yüklenen bu veri setleri, genellikle bir makine öğrenimi modelinin eğitilmesi ve değerlendirilmesi amacıyla kullanılır. Bu kısım, çeşitli veri setlerinin ilk 10 satırını görüntüleyerek her dosyanın yapısı hakkında hızlı bir inceleme yapmanızı sağlar. İlk olarak, **Item Categories** veri seti, ürün kategorilerinin kimlik numaralarını ve isimlerini içerir. Bu, hangi ürünlerin hangi kategoriye ait olduğunu anlamanızı sağlar. **Items** veri seti, her bir ürünün ID'si, adı ve kategorisi gibi bilgileri içerir, böylece her ürünün özelliklerini inceleyebilirsiniz. **Sales Train** veri seti, geçmiş satışlarla ilgili detayları sunar; tarih, mağaza kimliği, ürün kimliği ve satış miktarı gibi bilgiler içerir. **Sample Submission** dosyası, tahmin sonuçlarının formatını gösteren örnek bir şablondur ve model çıktılarını bu formatta kaydetmek için kullanılır. **Shops** veri seti, mağazaların kimlik numaraları ve adları hakkında bilgi sunar, bu da mağaza bazında analiz yapmanızı sağlar. Son olarak, **Test** veri seti, gelecekteki satış tahminleri yapılacak ürün-mağaza kombinasyonlarını içerir. Bu inceleme, veri setlerinin yapısını hızlıca anlamanıza ve modelleme sürecinde nasıl kullanılacaklarını belirlemenize yardımcı olur.

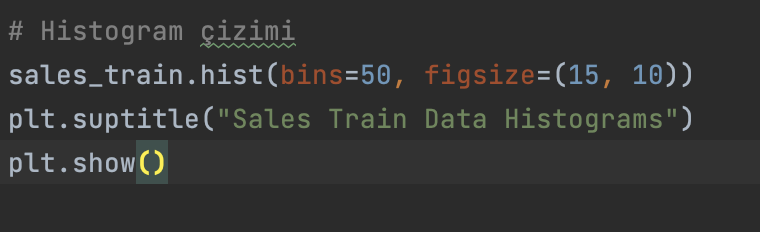


print(sales\_train.info()) komutu, sales\_train veri setinin genel bilgilerini görüntüler. Bu fonksiyon, veri setindeki sütun sayısı, her sütunun veri türü, eksik değerlerin olup olmadığı, bellek kullanımı gibi bilgileri gösterir. Bu şekilde, veri seti hakkında temel bir özet alabilirsiniz. Örneğin, veri türlerinin doğru olup olmadığını kontrol edebilir, eksik verileri analiz edebilir ve veri setinin büyüklüğü hakkında bilgi sahibi olabilirsiniz. info() fonksiyonu, genellikle veri temizliği ve analiz sürecine başlamadan önce ilk adım olarak kullanılır.



print(sales\_train.describe()) komutu, sales\_train veri setindeki sayısal sütunların istatistiksel özetini sağlar. Bu özet, her bir sayısal sütunun temel özelliklerini içerir. Veri setindeki her bir sütun için toplam geçerli değer sayısı (count), ortalama (mean), standart sapma (std), minimum ve maksimum değerler (min, max), ayrıca yüzde 25, 50 (medyan) ve 75'lik dilimler gibi çeyrek değerler yer alır. Bu istatistikler, verinin dağılımı hakkında genel bir bakış açısı sunar. Örneğin, veri setindeki anormal uç değerleri, genel eğilimleri ve verinin nasıl dağıldığını incelemenizi sağlar. Bu tür bir analiz, veri temizleme ve modelleme sürecinde önemli bir ilk adımdır, çünkü verinin yapısını anlamak ve gerektiğinde ön işleme yapmak için temel bilgiler sunar.

 print(sales\_train['item\_id'].value\_counts()) komutu, sales\_train veri setindeki item\_id sütunundaki her bir benzersiz ürün kimliğinin kaç kez tekrar ettiğini sayar. Yani, her ürünün kaç kez satıldığını veya veri setinde kaç kez göründüğünü belirler. Bu işlem, hangi ürünlerin daha sık satıldığını ya da veri setinde hangi ürünlerin yoğun olduğunu görmek için kullanılır. Bu tür bir analiz, veri setindeki dağılımı anlamanızı sağlar ve ürünlerin satış performansına dair bazı çıkarımlar yapmanıza yardımcı olabilir. Örneğin, bazı ürünlerin çok sayıda kez satıldığını, bazılarının ise nadiren görüldüğünü fark edebilirsiniz.

 print(sales\_train.hist(bins=50, figsize=(15, 10))) komutu, sales\_train veri setindeki sayısal sütunlar için histogramlar çizer. Histogram, verilerin dağılımını görsel olarak gösteren bir araçtır ve her sütunun değerlerinin frekansını görselleştirir. Burada, bins=50 parametresi, histogramın 50 farklı aralığa (bin) bölünmesini sağlar, yani veri aralığındaki farklı değerler 50 eşit bölüme ayrılır. figsize=(15, 10) parametresi ise grafiğin boyutlarını belirler, burada genişlik 15 inç, yükseklik ise 10 inç olarak ayarlanmıştır. plt.suptitle("Sales Train Data Histograms") komutu, histogramlar için başlık ekler. Son olarak, plt.show() komutu, çizilen histogramları ekranda gösterir.

Bu görselleştirme, verinin dağılımını anlamanıza yardımcı olur. Örneğin, bazı sütunların değerlerinin yoğunlaşıp yoğunlaşmadığını veya uç değerlere sahip olup olmadığını görmek için kullanılır. Histogramlar, veri setindeki belirgin eğilimleri ve dağılımı hızlı bir şekilde görselleştirir.

PROGRAM ÇIKTILARI:



