**Final Year B. Tech. (CSE) – I: 2022-23**

**5CS462: PE5 - Data Mining Lab**

**Assignment No. 9**

**Mini Project: Data analytics form amazon reviews**

**PRN: 2019BTECS00077 Date:20 Nov 2022**

**Full name: Biradar Avinash Vishnu**

**Title : Data analytics of amazon reviews**

**Code :**

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import matplotlib.pyplot as plt

import seaborn as sns

import streamlit as st

sns.set\_style('white')

# Split

from sklearn.model\_selection import train\_test\_split

from surprise import Reader, Dataset, SVD

from surprise.model\_selection import cross\_validate

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

    for filename in filenames:

        print(os.path.join(dirname, filename))

df = pd.read\_csv('Electronics\_data.csv')

st.title("Data Analytics for Amazon Reviews")

st.write("")

st.subheader("Dataset:")

st.write(pd.DataFrame(df.head(20)))

st.write("Total Reviews:",df.shape[0])

st.write("Total Columns:",df.shape[1])

st.write("Total number of ratings :",df.rating.nunique())

st.write("Total number of users   :", df.userId.nunique())

st.write("Total number of products  :", df.productId.nunique())

st.subheader("Ratings summary ")

df.describe()['rating']

# Average rating of products

st.subheader("Average rating of products")

ratings = pd.DataFrame(df.groupby('productId')['rating'].mean())

ratings['ratings\_count'] = pd.DataFrame(df.groupby('productId')['rating'].count())

ratings['ratings\_average'] = pd.DataFrame(df.groupby('productId')['rating'].mean())

st.write(ratings.head(10))

#histogram

ratings['rating'].hist(bins=70)

st.text(" ")

data=ratings['rating'].to\_list()

plt.rcParams['figure.figsize'] = [10, 4]

st.write("Histogram")

fig, ax = plt.subplots()

plt.locator\_params(nbins = 10)

plt.xlabel('rating')

plt.ylabel("rating\_count")

ax.hist(data)

st.pyplot(fig)

st.write(" ")

st.write(" ")

st.subheader("Most Popular products by using weighted rating method")

st.subheader("WR = (v ÷ (v+m)) × R + (m ÷ (v+m))× C")

st.text("\n\nR is the average rating for the item.\nv is the number of votes for the item.\nm is the minimum votes required to be listed in the popular items(defined by > percentile 80 of total votes).\nC is the average rating across the whole dataset.")

vote\_counts = ratings[ratings['ratings\_count'].notnull()]['ratings\_count'].astype('int')

vote\_averages = ratings[ratings['ratings\_average'].notnull()]['ratings\_average'].astype('int')

C = vote\_averages.mean()

st.write("Average rating of product across the whole dataset is i.e C = ",C)

m = vote\_counts.quantile(0.95)

st.write("Minimum votes required to be listed in the chart is i.e m = ",m)

qualified = ratings[(ratings['ratings\_count'] >= m) & (ratings['ratings\_count'].notnull()) & (ratings['ratings\_average'].notnull())][['ratings\_count', 'ratings\_average']]

qualified['ratings\_count'] = qualified['ratings\_count'].astype('int')

qualified['ratings\_average'] = qualified['ratings\_average'].astype('int')

st.write(qualified.sort\_values(by='ratings\_count', ascending=False))

st.write("No of items qualified",qualified.shape)

def weighted\_rating(x):

    v = x['ratings\_count']

    R = x['ratings\_average']

    return (v/(v+m) \* R) + (m/(m+v) \* C)

qualified['wr'] = qualified.apply(weighted\_rating, axis=1)

qualified = qualified.sort\_values('wr', ascending=False).head(20)

st.subheader("Top 10 Popular Products")

st.write(qualified.head(10))

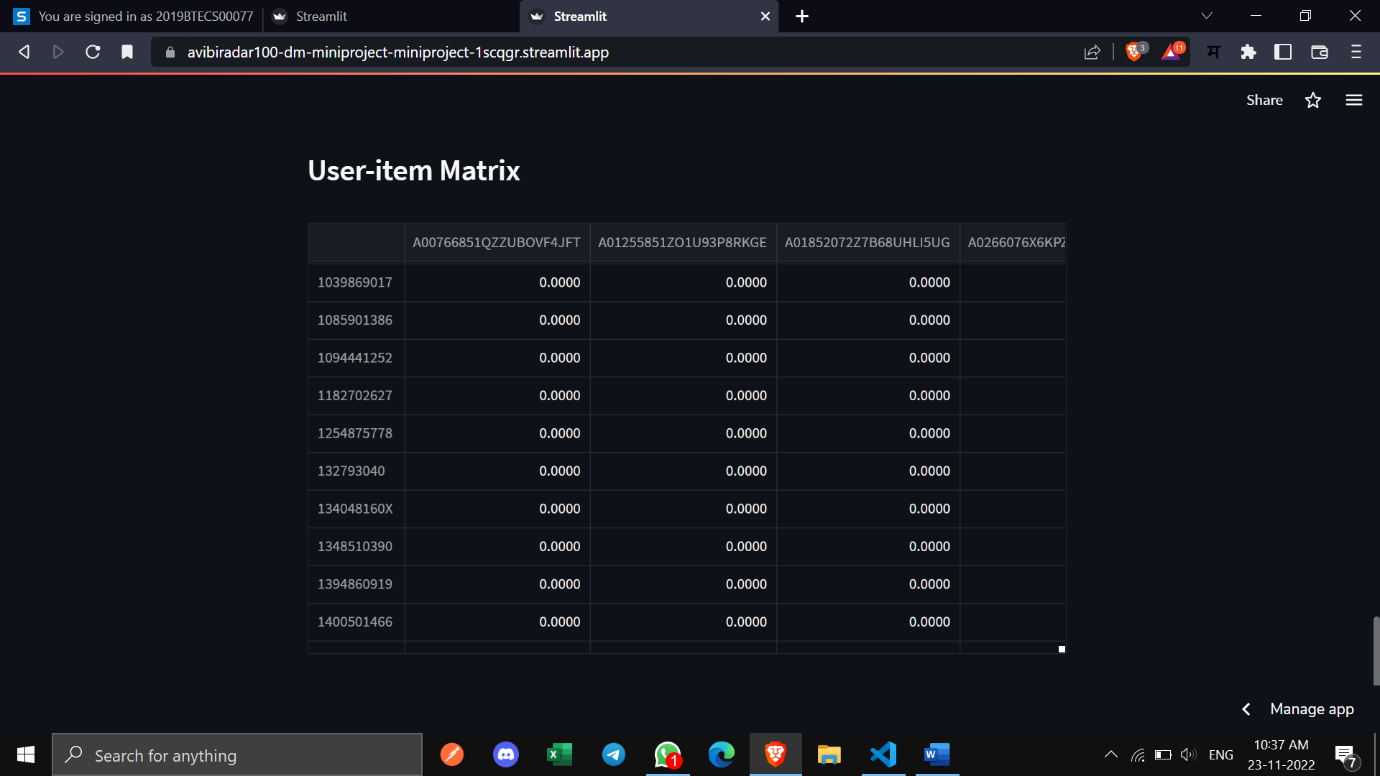
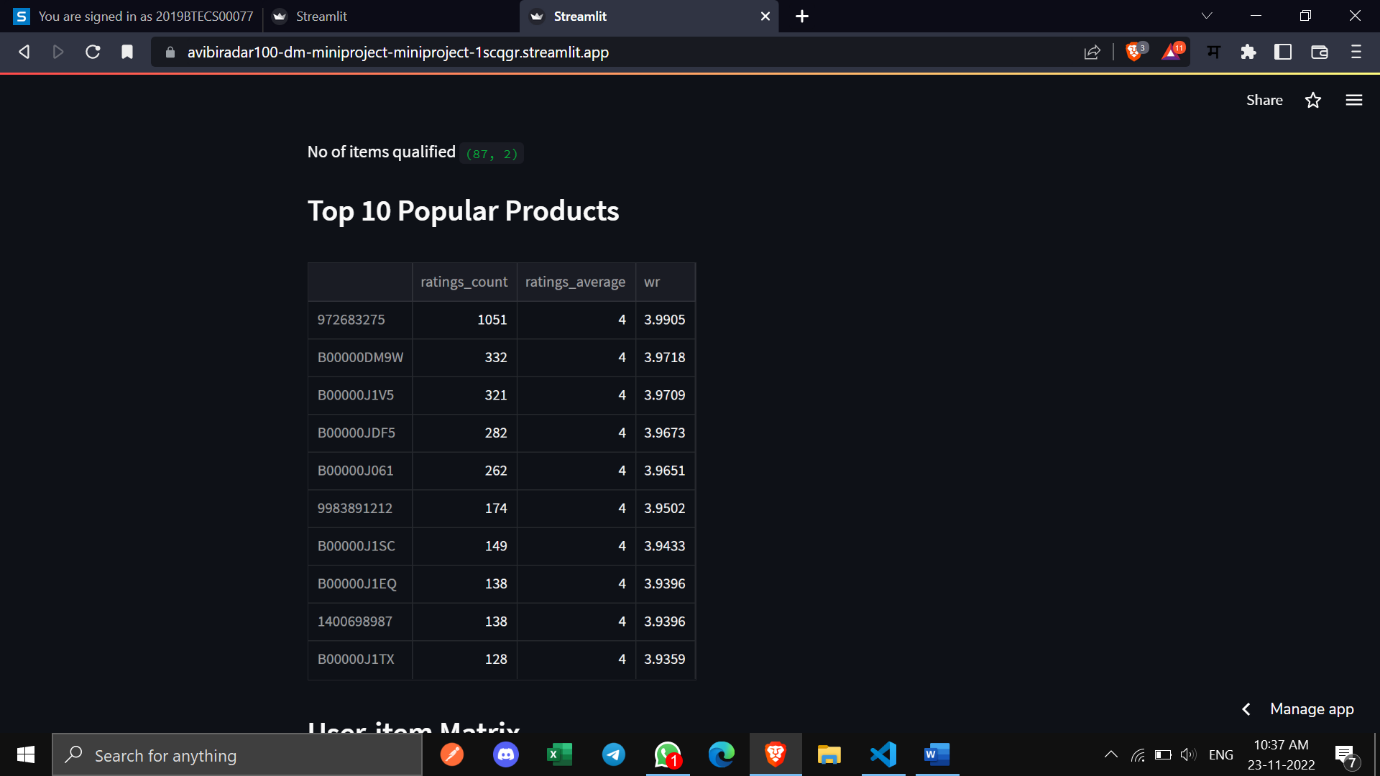
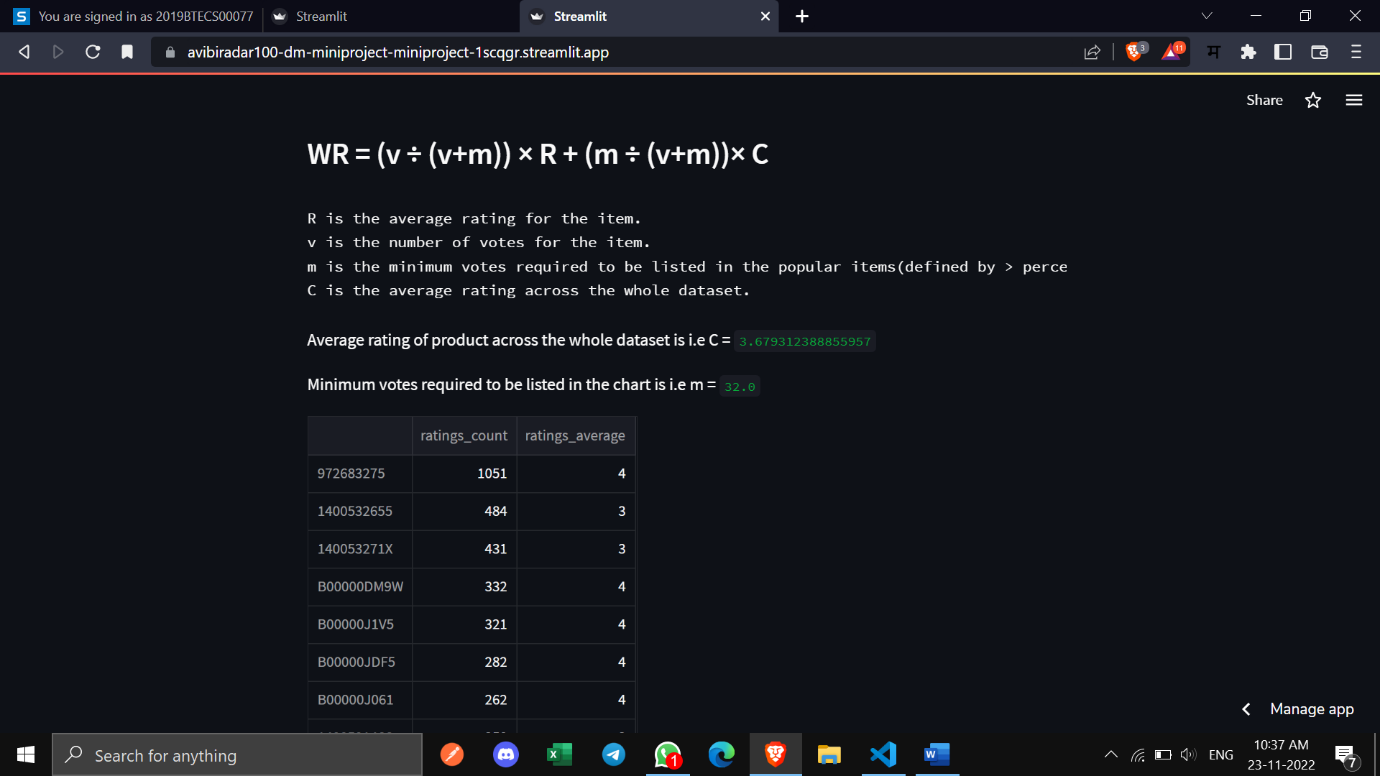
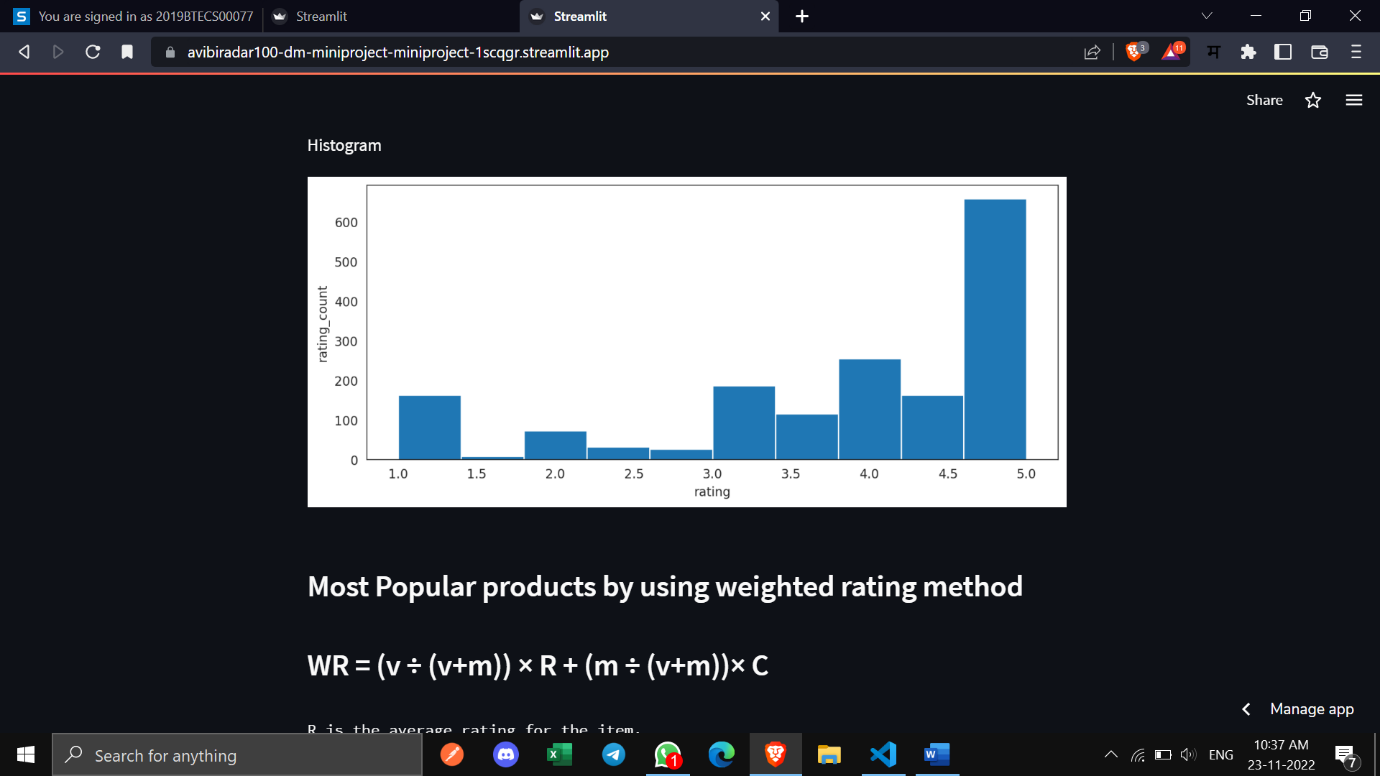
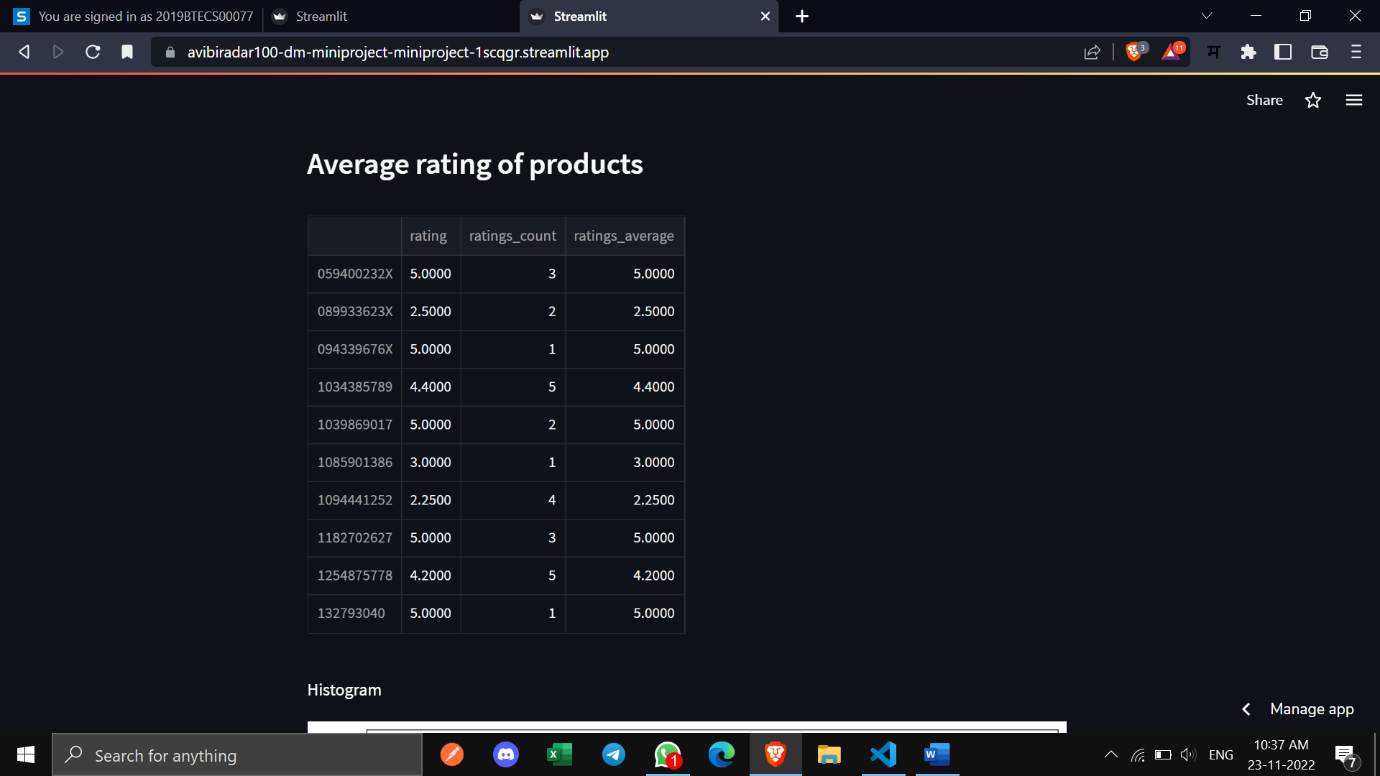
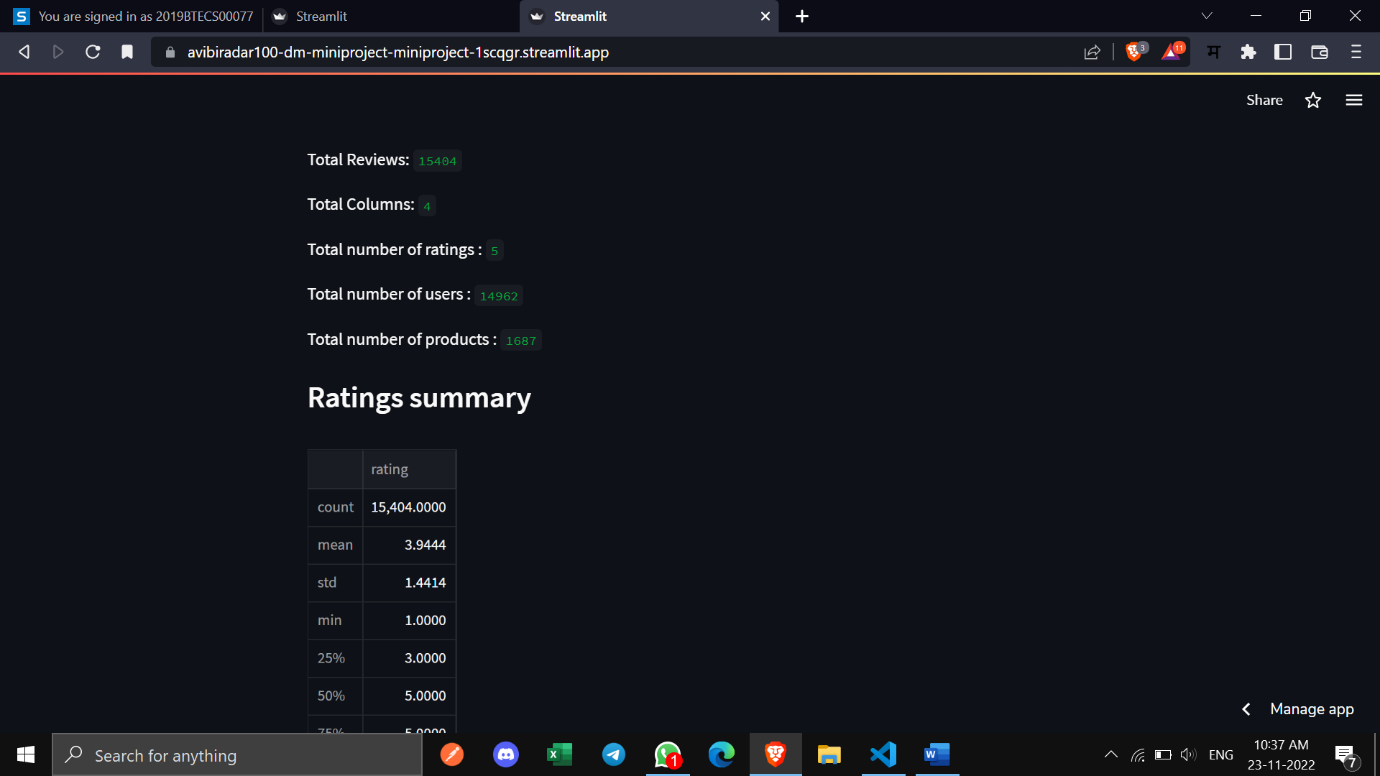
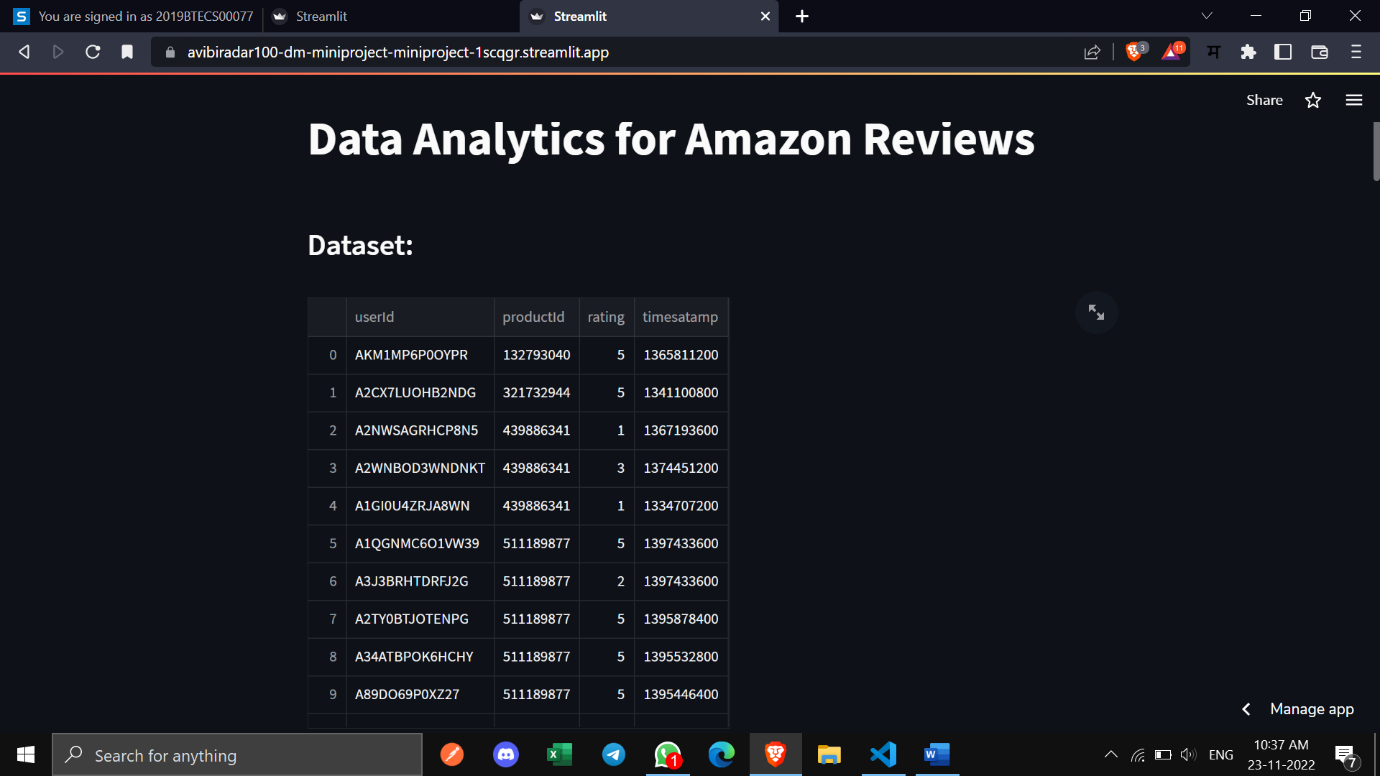
st.subheader("User-item Matrix")

# Matrix with row per 'user' and column per 'item'

pivot\_df = df.pivot(index = 'userId', columns ='productId', values = 'rating').fillna(0)

st.write(pivot\_df.head(5).transpose())

**Output:**

****