BLACK FRIDAY SALES PREDICTION

[21MCA4404]

Submitted By:

ARUN KUMAR A (ENG22MCA026)

In partial fulfilment for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

DEPARTMENT OF COMPUTER APPLICATIONS

DAYANANDA SAGAR UNIVERSITY

SCHOOL OF COMPUTER APPLICATION

KUDLUGATE, HOSUR ROAD, BANGALORE -560068

August 2024

ACKNOWLEDGEMENT

I wish to express my salutations to the beloved and highly esteemed institute **DAYANANDA SAGAR UNIVERSITY.** We thank **Dr**. **S. Senthil Kumar,** Dean & chairperson, School of Computer Application, Dayananda Sagar University, Bangalore, for his valuable suggestions and expert advice.

I am grateful to our guide and would like to express special thanks to **Prof. Alpa Patel** Assistant Professor, Department of Computer Applications, DSU for her constant support, necessary advice, and valuable guidance that helped me in completing the project/report work successfully.

I would like to thank all the Teaching and non-teaching staff of the Department of Computer Applications, DSU for their help. we would like to thank My parents and friends who have always supported me in every path of the Report work.

I perceive this opportunity as a milestone in my career development. we will strive to use the gained skills and knowledge in the best possible way and we will continue to work on my improvement in order to achieve the desired career objectives. we hope to continue the cooperation with all of you in the future.

ARUN KUMAR A [ENG22MCA026]

DAYANANDA SAGAR UNIVERSITY BANGALORE

School of Computer Application

Department of Computer Applications

KudluGate, Bangalore – 560068 Karnataka, India

BONAFIDE CERTIFICATE

This is to certify that the Project work titled **"BLACK FRIDAY SALES PREDICTION"** is carried out by ARUN KUMAR A **(ENG22MCA026)** a bonafide students of Master of Computer Applications at the **Dayananda Sagar University**, Bangalore in partial fulfilment for the award of degree in Master of Computer Applications, during the year **2023-2024.**

Guide	Chairman
Prof. Alpa Patel	**Dr. S. Senthil Kumar**
Assistant Professor	Dean & Chairperson
Dept. of Computer Applications,	Dept. of Computer Applications,
Dayananda Sagar University	Dayananda Sagar University

Project Viva Held on: -

Internal Examiner	External Examiner

ABSTRACT

This project involves the development of a Streamlit web application to analyze and predict Black Friday sales using various machine learning techniques. The application allows users to explore sales data, visualize trends, and predict purchases based on user inputs. By employing models such as linear regression, random forest, and decision trees, the app helps analyze the relationships among different attributes and predict future sales outcomes. This approach enhances current sales strategies and provides a framework for future sales predictions, ensuring the versatility and applicability of these models to other sales prediction scenarios. This project is valuable for ongoing retail analysis and strategic development, aiding retailers in better understanding their customers and improving overall sales performance.

The prediction model built will provide a prediction based on the age of the customer, city category, occupation, etc. The prediction model is implemented based on models like linear regression, Decision Tree Regressor, Random Forest Regressor

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	
	LIST OF FIGURES	
1	**INTRODUCTION**	1
	1.1 INTRODUCTION TO THE PROJECT	1
2	**LITERATURE SURVEY**	2
3	**SYSTEM CONFIGURATION**	3
	3.1 HARDWARE SPECIFICATION	3
	3.2 SOFTWARE SPECIFICATION	3
4	**SYSTEM ANALYSIS**	4
	4.1 Existing System	4
	4.2 Proposed System	4
5	**DATASET USED**	5
6	**DATA PRE-PROCESSING**	6
7	**FEATURE EXTRACTION**	7
8	**LIBRARY USED**	9
	8.1 Streamlit	9
	8.2 Joblib	9
	8.3 WordCloud	9
9	**MODEL EVALUATION AND SELECTION**	10
	9.1. Linear Regression Model	10
	9.2. Decision Tree Regressor	10
	9.3 Random Forest Regressor	11

CHAPTER NO	TITLE	PAGE NO
	9.4 Choice of Model	11
10	**USER INTERFACE**	12
	10.1 Front End	13
11	**CONCLUSION**	15
	FUTURE SCOPE	15
	REFERENCES	16
	APPENDICES	17
	A. SOURCE CODE	17
	B. SCREENSHOT	22

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO
12.1	HOME PAGE	22
12.2	Exploratory Data Analysis	22
12.3	EDA Gender Plotting	23
12.4	EDA City Plotting	23
12.5	User Input	24
12.6	Predicted Purchases	24

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION OT THE PROJECT

Black Friday, also known as Thanksgiving Day in the United States, occurs on the fourth Thursday of November and represents the busiest shopping day of the year. This sale period is crucial for the economy, with around 30 percent of annual retail sales happening during the holiday season, and certain retailers, such as jewelers, witnessing nearly 40 percent of their annual sales. Retailers often face challenges in meeting customer demands due to a lack of understanding of their preferences and needs. Prediction models are essential for gaining insights into customer behavior and optimizing sales strategies.

This project involves the development of a Streamlit web application to analyze and predict Black Friday sales using various machine learning techniques. The application allows users to explore sales data, visualize trends, and predict purchases based on user inputs. By employing models such as linear regression, random forest, and decision trees, the app helps analyze the relationships among different attributes and predict future sales outcomes. This approach enhances current sales strategies and provides a framework for future sales predictions, ensuring the versatility and applicability of these models to other sales prediction scenarios. This project is valuable for ongoing retail analysis and strategic development, aiding retailers in better understanding their customers and improving overall sales performance.

CHAPTER 2

LITERATURE SURVEY

The research is carried out on the analysis and prediction of sales using various techniques. There are many methods proposed to do so by various researchers. In this section, we will summarize a few of the machine learning approaches.

C. M. Wu et al. [1] have proposed a prediction model to analyze the customer's past spending and predict the future spending of the customer. The dataset referred is Black Friday Sales Dataset from analyticsvidhya. They have machine learning models such as Linear Regression, MLK classifier, Deep learning model using Keras, Decision Tree, and Decision Tree with bagging, and XGBoost. The performance evaluation measure Root Mean Squared Error (RMSE) is used to evaluate the models used. Simple problems like regression can be solved by the use of simple models like linear regression instead of complex neural network models.

Odegua, Rising [2] have proposed a sales forecasting model. The machine learning models used for implementation are K-Nearest Neighbor, Random Forest, and Gradient Boosting. The dataset used for the experimentation is provided by Data Science Nigeria, as a part of competitions based on Machine Learning. The performance evaluation measures used are Mean Absolute Error (MAE). Random Forest outperformed the other algorithms with a MAE rate of 0.409178.

Singh, K et al [3] have analyzed and visually represented the sales data provided in the complex dataset from which we ample clarity about how it works, which helps the investors and owners of an organization to analyze and visualize the sales data, which will outcome in the form of a proper decision and generate revenue. The data visualization is based on different parameters and dimensions. The result of which will enable the end-user to make better decisions, ability to predict future sales, increase the production dependencies on the demand, and also regional sales can be calculated

Ramasubbareddy S. et al. [4] have applied machine learning algorithms to predict sales. The dataset for the experimentation purpose is taken from Kaggle, named as Black Friday Sales Dataset. The algorithms used for the implementation of the system are linear regression, Decision Tree, Random Forest. Root Mean Squared Error is used as the performance evaluation measure. As per RMSE lower the RMSE value better the prediction. As a result, based on the RMSE rate Rule-Based DT outperforms other machine learning techniques with a RMSE rate of 2291.

CHAPTER 3

SYSTEM CONFIGURATION

3.1 Hardware Specification

- Processor : Core i5 – 5th Gen
- RAM : 16GB
- System type : 64Bit Operating System
- Monitor : Dell 24 inch
- Keyboard : USB keyboard
- Mouse : USB Optical mouse

3.2 Software Specification

- Platform : Microsoft Windows 10
- Designing language : Python 3.9,Plotly,Joblib,WordCloud
- App framework : Streamlit App

CHAPTER 4

SYSTEM ANALYSIS

4.1 Existing System:

In the existing system, traditional methods are used for data analysis and sales prediction. These involve manual data exploration, basic statistical analysis, and limited predictive modeling. The process is often cumbersome, lacking a user-friendly interface, which makes it challenging for users to interact with the data efficiently. This system struggles to handle the complexity of Black Friday sales data and involves high risks of ambiguity and redundancy due to outdated manual entry processes.

Challenges in the Existing System:

- Limited ability to uncover hidden patterns and insights.
- Lack of advanced machine learning techniques for accurate sales predictions.
- Time-consuming manual processes for data exploration and analysis.
- Inefficiencies in handling large and complex datasets.
- Absence of an interactive user interface for ease of use.

4.2 Proposed System

The proposed system introduces a Streamlit-based web application that incorporates advanced data analysis techniques and machine learning models to enhance the accuracy and efficiency of Black Friday sales predictions. This system provides a seamless user interface where users can input data and receive predictions instantly.

Key Features of the Proposed System:

- User Interface: A user-friendly interface that allows easy interaction with the data, making the prediction process straightforward and accessible.
- Exploratory Data Analysis (EDA): Utilizes advanced EDA techniques to uncover hidden patterns and trends in the dataset.
- Machine Learning Models: Implements state-of-the-art models like Linear Regression, Decision Trees, and Random Forest for accurate forecasting.

CHAPTER 5

DATASET USED

This dataset consists of sales transactions from a Walmart retail store, providing a rich opportunity to enhance feature engineering skills and gain insights into shopping behaviors. It includes two primary datasets: Train and Test. The Train dataset is utilized for model training with a train-test split approach, allowing us to develop predictive models that forecast the Purchase Amount based on various features. The Test dataset serves as an input to evaluate these models by predicting purchase amounts. Sourced from Kaggle, this dataset captures diverse aspects of consumer purchases, such as product categories, user demographics, and shopping patterns, making it ideal for exploring advanced data analysis techniques and improving sales strategies.

- Training Set Shape: 550,068 records and 12 features
- Test Set Shape: 233,599 records and 11 features

Features:

- User_ID: Unique identifier for each user
- Product_ID: Unique identifier for each product
- Gender: Gender of the user
- Age: Age of the user, categorized into bins
- Occupation: Coded occupation of the user
- City_Category: Categorization of the user's city (A, B, or C)
- Stay_In_Current_City_Years: Number of years the user has lived in their current city
- Marital_Status: Marital status of the user (single or married)
- Product_Category_1: Primary product category
- Product_Category_2: Secondary product category
- Product_Category_3: Tertiary product category
- Purchase: Purchase amount

CHAPTER 6

DATA PRE-PROCESSING

Data pre-processing is a crucial phase in preparing the Walmart dataset for analysis and model development. The following steps were meticulously followed to ensure the data's quality and suitability for machine learning models:

Handling Missing Values:

Some entries in Product_Category_2 and Product_Category_3 were missing. We replaced these missing values with max value to retain the integrity of the dataset while acknowledging the absence of certain categories for some products.

Encoding Categorical Variables:

Categorical variables such as Gender, Age, City_Category, and Marital_Status were converted into numerical formats. For instance, we encoded Gender as 0 for female and 1 for male, and City_Category as 0 for A, 1 for B, and 2 for C.

Scaling Numerical Features:

Numerical features such as Age and Stay_In_Current_City_Years were scaled to standardize the data. This step ensured that each feature contributed equally to the model's performance by preventing features with larger scales from dominating the training process. Standardizing the features helped in achieving a balanced and stable training process, leading to better model performance.

Creating Dictionaries for Categorical Mappings:

We created dictionaries to map the categorical values to their corresponding numerical encodings. These mappings facilitated easier interpretation and manipulation of the data during the modeling phase. For example, we used dictionaries to map Gender to 0 and 1, and City_Category to 0, 1, and 2.

CHAPTER 7

FEATURE EXTRACTION

Feature selection is crucial in Black Friday sales prediction as it helps in identifying the key factors that influence purchasing behavior. By selecting the most informative features, the model can focus on relevant aspects such as user demographics, product categories, and shopping patterns, which directly impact sales predictions. Moreover, feature selection aids in reducing the dimensionality of the dataset, making the model more computationally efficient and less prone to overfitting.

Feature selection techniques such as manual selection or automated methods may have been employed. Manual selection involves domain expertise to handpick features based on their relevance and significance to sales prediction. Alternatively, automated methods such as univariate feature selection, recursive feature elimination, or feature importance ranking based on tree-based models may have been utilized to objectively identify the most important features.

The features selected for Black Friday sales prediction in the provided code include a combination of factors that directly influence purchasing behavior.

These features encompass:

- **User_ID and Product_ID**:

  These identifiers uniquely represent each user and product, helping track individual transactions and analyze user-specific purchasing patterns.

- **Gender**:

  This feature indicates the sex of the user, which can influence purchasing preferences and behavior.

- **Age**:

  The age of the user, categorized into bins, provides insights into the purchasing habits of different age groups.

- **Occupation**:

  This masked feature indicates the user's occupation, which can affect disposable income and purchasing power.

- **City_Category**:

  This feature categorizes the city into A, B, or C, reflecting urbanization levels and potentially influencing consumer behavior.

- **Stay_In_Current_City_Years**:

  The number of years a user has stayed in their current city, which may correlate with purchasing patterns and brand loyalty.

- **Marital_Status**:

  The marital status of the user, which can impact purchasing decisions and preferences.

- **Product_Category_1, Product_Category_2, Product_Category_3**:

  These features represent the primary and additional categories to which a product may belong, helping to understand preferences for different product types.

- **Purchase**:

  The purchase amount, which is the target variable for prediction, representing the total spending on Black Friday.

By carefully selecting and extracting these features, the project aims to enhance the predictive power of the machine learning models. This approach not only aids in improving the accuracy of current sales predictions but also provides a robust framework for analyzing future sales scenarios. The versatility of these features ensures their applicability to other sales prediction tasks, making this project valuable for ongoing retail analysis and strategy development.

CHAPTER 8

LIBRARY USED

8.1 Streamlit:

Streamlit is an open-source Python library that allows developers to create interactive web applications with minimal effort. It is designed for data scientists and machine learning engineers to quickly build and share applications that visualize data and machine learning models. Streamlit's simplicity lies in its ability to turn Python scripts into interactive apps without the need for extensive front-end development.

- st.title(): Sets the title of the web app.
- st.write(): Displays text, data, and other elements.
- st.line_chart(): Plots a line chart of the provided data.

8.2 Joblib:

Joblib is a Python library used for efficiently serializing and deserializing large Python objects, such as NumPy arrays and machine learning models. It is particularly useful for saving and loading machine learning models and other large data structures, as it provides optimized performance for both I/O operations and memory usage.

- dump(): Serializes and saves the object to a file.
- load(): Deserializes and loads the object from the file.

8.3 WordCloud:

WordCloud is a Python library used to generate visually appealing word clouds from text data. A word cloud displays words in varying sizes, with more frequent words appearing larger and bolder. This visualization technique helps to quickly identify the most common terms and themes within a text dataset.

- WordCloud(): Initializes the word cloud object with customization options.
- generate(): Creates a word cloud from the provided text.
- plt.imshow(): Displays the generated word cloud image.

CHAPTER 9

MODEL EVALUATION AND SELECTION

we evaluate the performance of several regression models used for predicting Black Friday sales: Linear Regression, Decision Tree Regressor, and Random Forest Regressor. The evaluation is based on Root Mean Squared Error (RMSE) and R^2 score.

9.1. Linear Regression Model:

- Root Mean Squared Error (RMSE): 4701.99
  The Linear Regression model assumes a linear relationship between the input features and the target variable. While its RMSE indicates a higher average prediction error compared to other models, it provides a baseline for model performance due to its simplicity and interpretability.
- R^2 Score: 0.110
  With an R^2 score of 0.110, Linear Regression explains only about 11% of the variance in sales data. This suggests that the model has limited explanatory power for the complexity of the Black Friday sales data.

Despite its simplicity, the Linear Regression model offers easy interpretability, making it useful for understanding the relationship between features and the target variable. However, its performance in terms of RMSE and R^2 score indicates that it may not be the best model for capturing complex patterns in the sales data.

9.2. Decision Tree Regressor:

- Root Mean Squared Error (RMSE): 3306.32
  The Decision Tree Regressor, capable of capturing non-linear relationships, demonstrates a lower RMSE compared to Linear Regression. This suggests that it provides more accurate predictions by fitting the data more closely.
- R^2 Score: 0.560
  The R^2 score of 0.560 indicates that the Decision Tree Regressor explains approximately 56% of the variance in the sales data, representing a significant improvement over Linear Regression.

The Decision Tree Regressor captures more complex relationships in the data and shows better performance metrics. However, decision trees can be prone to overfitting, which may affect generalizability.

9.3 Random Forest Regressor:

- Root Mean Squared Error (RMSE): 3052.02

  The Random Forest Regressor, an ensemble method combining multiple decision trees, achieves the lowest RMSE among the models evaluated. This suggests it provides the most accurate predictions with the least error.

- R^2 Score: 0.725

  With an R^2 score of 0.725, the Random Forest Regressor explains about 73% of the variance in the sales data, indicating the best performance in capturing the underlying patterns.

The Random Forest Regressor demonstrates the highest accuracy and best fit for the data, making it the most effective model for predicting Black Friday sales in this context. Its ability to handle complex relationships and interactions between features contributes to its superior performance.

9.4 Choice of Model

- RMSE: The Random Forest Regressor has the lowest RMSE, indicating the highest accuracy in predictions.
- R^2 Score: The Random Forest Regressor also has the highest R^2 score, reflecting the best ability to explain the variance in the target variable.

The Random Forest Regressor is selected as the primary model for predicting Black Friday sales. It not only provides the most accurate predictions but also offers a robust approach to capturing complex patterns in the data. While other models like Linear Regression and Decision Tree Regressor offer useful insights and interpretations, the Random Forest Regressor's performance metrics make it the preferred choice for this task.

CHAPTER 10

USER INTERFACE

Streamlit is an open-source Python library that allows developers to create interactive web applications for data science and machine learning projects. Unlike traditional web frameworks, Streamlit is designed to be straightforward and user-friendly, providing a quick way to deploy machine learning models and data visualizations. It focuses on simplicity and ease of use, making it ideal for creating interactive dashboards and applications.

Key Components of the Streamlit Web Application for Black Friday Sales Prediction:

- **Streamlit App Setup:**
  - The application is structured with different pages for home, exploratory data analysis (EDA), and machine learning (ML) model prediction.
  - The app uses Streamlit’s sidebar for navigation, where users can switch between different functionalities of the app.
- **Data Loading and Preprocessing:**
  - Data is loaded and preprocessed using Streamlit's caching capabilities to improve performance.
  - Preprocessing steps include encoding categorical variables and scaling numerical features to prepare them for model input.
- **Web Pages:**
  - The **Home Page** displays a word cloud generated from Thanksgiving greetings in multiple languages, allowing users to explore various ways to say "Happy Thanksgiving."
  - The **EDA Page** provides visualizations and summaries of the Black Friday sales data, including distributions of key variables and relationships between features.
  - The **ML Page** allows users to input various features related to sales and make predictions using a pre-trained machine learning model.

- **Prediction:**
  - User inputs from the ML Page are processed and transformed into a format suitable for the machine learning model.
  - The pre-trained model is used to make predictions, which are then displayed on the same page with relevant results.
- **Output Rendering:**
  - Predictions and results are presented interactively, with Streamlit updating the user interface dynamically based on user inputs.
  - Key metrics and visualizations are displayed in an easy-to-read format, making the insights from the predictions accessible.
- **Additional Features:**
  - **Visualization Integration:** Various visualizations, including pie charts, bar charts, and histograms, are integrated to provide a comprehensive view of the data.
  - **Interactive Elements:** Users can interact with the application by selecting different features and observing how the predictions change in real-time.

10.1 FRONT END

The front end of the Streamlit application focuses on user experience and interaction for the Black Friday sales prediction model.

HTML and CSS Considerations:

- **Page Layout:**
  - **Home Page**: Features a word cloud visualization created with the WordCloud library, offering a visually engaging way to explore Thanksgiving greetings.
  - **EDA Page**: Includes various plots and summaries that help users understand the sales data and its distributions.

- **ML Page**: Provides input fields for users to enter sales-related features and view prediction results.

➢ **Styling and Design:**

- The app uses Streamlit’s built-in styling capabilities and external CSS for a polished look.
- Custom styling includes setting colors, fonts, and layout properties to enhance the visual appeal and usability of the application.

➢ **Animation and Interactivity:**

- Streamlit supports basic interactivity through widgets like sliders, selectors, and input fields, allowing users to interact with the application easily.
- Advanced animations and effects are not directly supported in Streamlit but can be simulated using interactive elements and dynamic updates.

CHAPTER 11

CONCLUSION

we developed a predictive model for Black Friday sales using various machine learning techniques. By meticulously executing data collection, preprocessing, exploratory data analysis (EDA), and model evaluation, we created a tool capable of forecasting sales with notable accuracy. Among the models tested, the Random Forest Regressor demonstrated the best performance, providing a robust prediction of sales trends. The integration of this model into a user-friendly Streamlit application allows for an interactive experience, enabling users to engage directly with the predictions. This project has not only delivered valuable insights into consumer purchasing patterns but also showcased the practical application of machine learning in sales forecasting.

11.1Future Scope:

Model Enhancement:

- Advanced Models: Exploring more advanced machine learning models and techniques, such as Gradient Boosting Machines (GBMs) or Neural Networks, may provide better performance and insights.
- Hyperparameter Tuning: Implementing techniques like Grid Search or Random Search for hyperparameter tuning could optimize model performance further.

Data Integration:

- Real-Time Data: Integrating real-time data sources, such as live sales feeds or social media sentiment analysis, could enhance the model's predictive capabilities and provide more up-to-date insights.
- External Data: Incorporating external factors like economic indicators or competitor sales data might offer a more comprehensive view of sales trends

CHAPTER 12

REFERENCES

[1] C. M. Wu, P. Patil and S. Gunaseelan, "Comparison of Different Machine Learning Algorithms for Multiple Regression on Black Friday Sales Data," 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), 2018, pp. 16-20, doi: 10.1109/ICSESS.2018.8663760.

[2] Odegua, Rising. (2020). Applied Machine Learning for Supermarket Sales Prediction.

[3] K. Singh and R. Wajgi, "Data analysis and visualization of sales data," 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave), 2016, pp. 1-6, doi: 10.1109/STARTUP.2016.7583967.

[4] Ramasubbareddy S., Srinivas T.A.S., Govinda K., Swetha E. (2021) Sales Analysis on Back Friday Using Machine Learning Techniques. In: Satapathy S., Bhateja V., Janakiramaiah B., Chen YW. (eds) Intelligent System Design. Advances in Intelligent Systems and Computing, vol 1171. Springer, Singapore. https://doi.org/10.1007/978-981-15-5400-1_32

APPENDICES

A.SOURCE CODE

```
# Core Pkgs
import streamlit as st
import streamlit.components.v1 as stc
from home_page import run_home_page
from eda_app import run_eda
from ml_app import run_ml

html_temp = """
    <div style="background-color:#3872fb;padding:10px;border-radius:10px">
    <h1 style="color:white;text-align:center;">Black Friday Sales App</h1>
    <h4 style="color:white;text-align:center;">Happy Thanksgiving</h4>
    </div>
    """


def main():
    stc.html(html_temp)
    menu = ["Home", "EDA", "ML", "About"]
    choice = st.sidebar.selectbox("Menu", menu)

    if choice == "Home":
        run_home_page()
    elif choice == "EDA":
        run_eda()
    elif choice == "ML":
        run_ml()
    else:
        st.subheader("About")
        st.info("Built with Streamlit")
        st.text(" MCA")
        st.text("ARUN KUMAR A")

if __name__ == '__main__':
    main()
```

```
import streamlit as st
import pandas as pd
# Data Viz Pkgs
import matplotlib.pyplot as plt
import matplotlib
matplotlib.use('Agg')
import seaborn as sns
import plotly.express as px

@st.cache_data
def load_data(data):
df = pd.read_csv(data)
 return df

def count_plot(dataframe, column_name, title=None, hue=None):
    '''
 Function to plot seaborn count plot
Input: Dataframe name that has to be plotted, column_name that has to be
plotted, title for the graph
Output: Plot the data as a count plot
    '''
base_color = sns.color_palette()[0]
sns.countplot(data=dataframe, x=column_name, hue=hue)
plt.title(title)

def run_eda():
st.subheader("EDA")
 submenu = st.sidebar.selectbox("Submenu", ["EDA", "Plots"])
 df = load_data("data/BlackFriday.csv")

if submenu == "EDA":
 st.subheader("Exploratory Data")
 st.dataframe(df.head())

c1, c2 = st.columns(2)

with st.expander("Descriptive Summary"):
 st.dataframe(df.describe())
```

```
with c1:
with st.expander("Gender Distribution"):
st.dataframe(df['Gender'].value_counts())
with c2:
with st.expander("Age Distribution"):
 st.dataframe(df['Age'].value_counts())
elif submenu == "Plots":
 st.subheader("Plotting")
 col1, col2 = st.columns(2)
 with col1:
 with st.expander("Pie Chart (Gender)"):
 gen_df = df['Gender'].value_counts().to_frame()
 gen_df = gen_df.reset_index()
 gen_df.columns = ['Gender Type', 'Counts']
 p01 = px.pie(gen_df, names='Gender Type', values='Counts')
 st.plotly_chart(p01, use_container_width=True)
  with st.expander("City"):
 city_df = df['City_Category'].value_counts().to_frame()
city_df = city_df.reset_index()
city_df.columns = ['Category', 'Counts']
p01 = px.pie(city_df, names='Category', values='Counts')
st.plotly_chart(p01, use_container_width=True)
 with col2:
with st.expander("Bar Chart(Gender)"):
fig = plt.figure()
 sns.countplot(df['Gender'])
 st.pyplot(fig)

 with st.expander("Plot of Occupation"):
 fig = plt.figure()
sns.countplot(df['Occupation'])
 st.pyplot(fig)

with st.expander("Age"):
 age_df = df['Age'].value_counts().to_frame()
age_df = age_df.reset_index()
age_df.columns = ['Age Range', 'Counts']
 p01 = px.bar(age_df, x='Age Range', y='Counts')
```

```
st.plotly_chart(p01, use_container_width=True)

with st.expander("Gender vs Marital Status"):
marital_df = df.groupby(['Gender',
'Marital_Status']).size().to_frame().reset_index()
marital_df.rename(columns={0: 'Counts'}, inplace=True)
 po2 = px.bar(marital_df, x='Marital_Status', y='Counts', color='Gender')
st.plotly_chart(po2)

# Core Pkgs
import streamlit as st

# Utils
import numpy as np
import joblib
import os

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

def scale_data(x):
 X = scaler.fit_transform(x)
return X

age_dict = {'0-17': 1, '55+': 7, '26-35': 3, '46-50': 5, '51-55': 6, '36-45': 4, '18-
25': 2}
gender_dict = {"Female": 0, "Male": 1}
marital_status_dict = {"Single": 0, "Married": 1}
city_dict = {'A': 0, 'B': 1, 'C': 2}

def get_value(val, my_dict):
 for key, value in my_dict.items():
 if val == key:
return value

# Load ML Models
@st.cache_data
def load_model(model_file):
```

```
loaded_model = joblib.load(open(os.path.join(model_file), "rb"))
return loaded_model

def run_ml():
st.subheader("Black Friday Sales Predictor")
col1, col2 = st.columns(2)

with col1:
 gender = st.radio("Gender", ("Female", "Male"))
 age = st.number_input("Age", 1, 75)
 occupation = st.number_input("Occupation", 1, 20)
 city_category = st.selectbox("City Category", ["A", "B", "C"])
stay_in_current_city = st.number_input("No of Years of Stay in Current
City", 1, 10)

with col2:
 marital_status = st.radio("Marital Status", ("Single", "Married"))
product_category_1 = st.number_input("Product 1", 1, 20)
 product_category_2 = st.number_input("Product 2", 1, 20)
product_category_3 = st.number_input("Product 3", 1, 20)

selected_options = {
'Gender': gender, 'Age': age, 'Occupation': occupation,
'City_Category': city_category, 'Stay_In_Current_City_Years':
stay_in_current_city,
'Marital_Status': marital_status, 'Product_Category_1': product_category_1,
'Product_Category_2': product_category_2, 'Product_Category_3':
product_category_3
    }

gender_en = get_value(gender, gender_dict)
city_category_en = get_value(city_category, city_dict)
marital_status_en = get_value(marital_status, marital_status_dict)
single_sample = [
gender_en, age, occupation, city_category_en, stay_in_current_city,
marital_status_en,
        product_category_1, product_category_2, product_category_3
    ]
```

B.FINAL OUTPUT

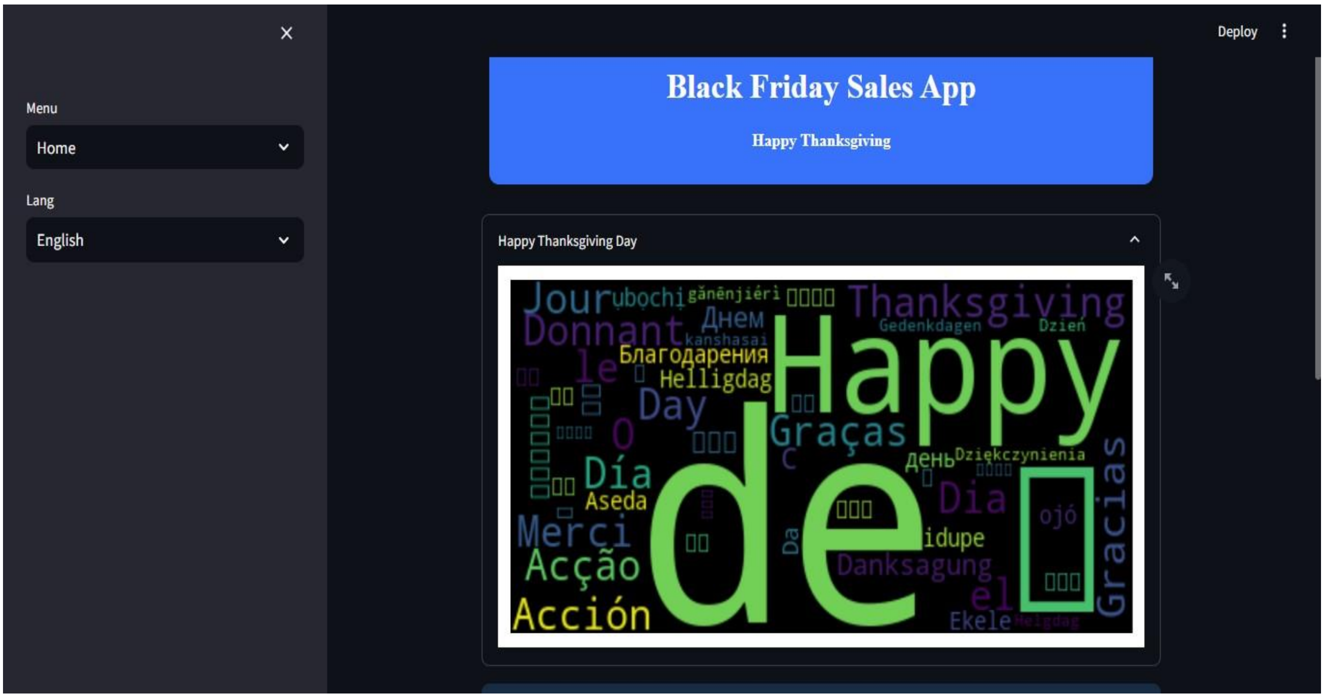

Fig 12.1 Home page

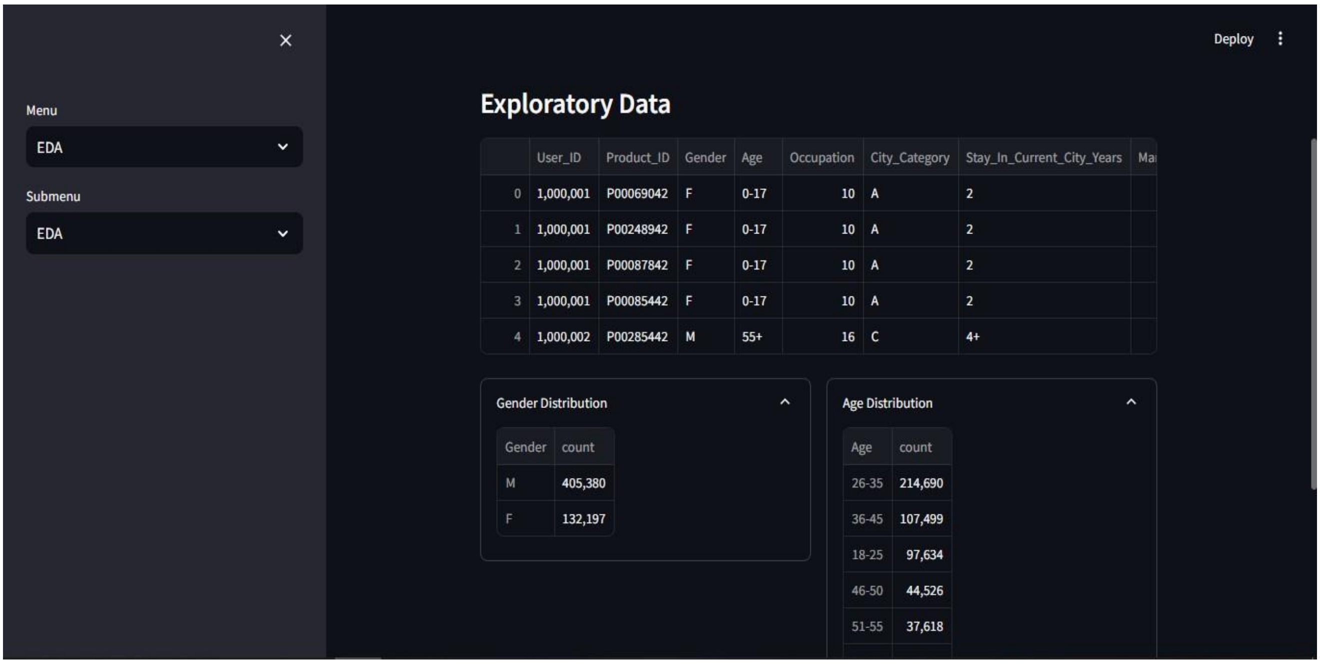

Fig 12.2 Exploratory Data Analysis

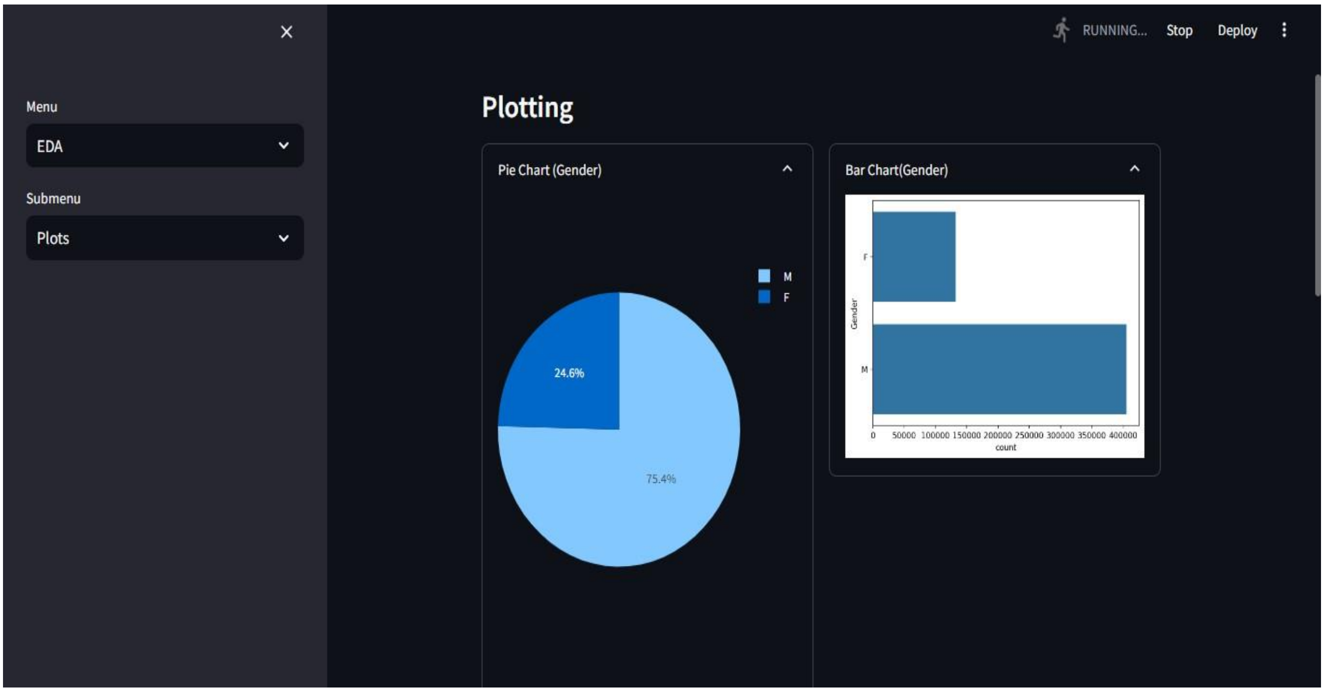

Fig 12.3 EDA Gender Plotting

Fig 12.4 EDA City Plotting

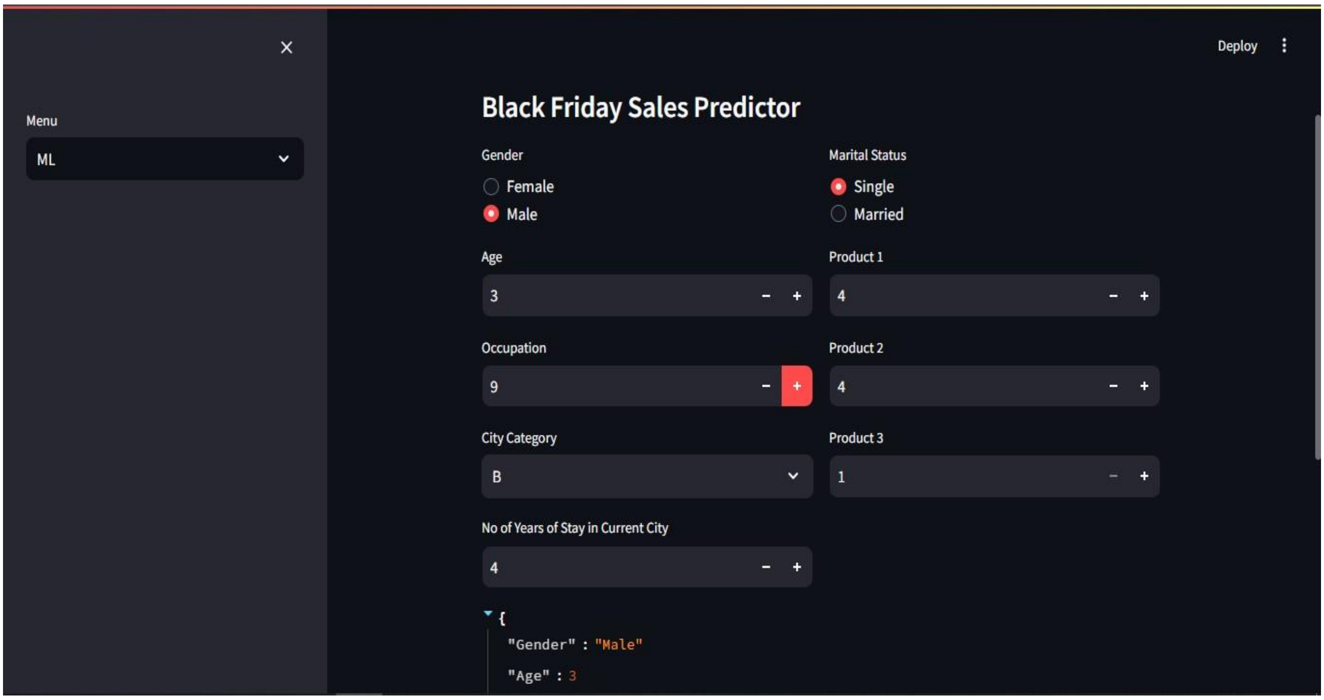

Fig 12.5 User Input

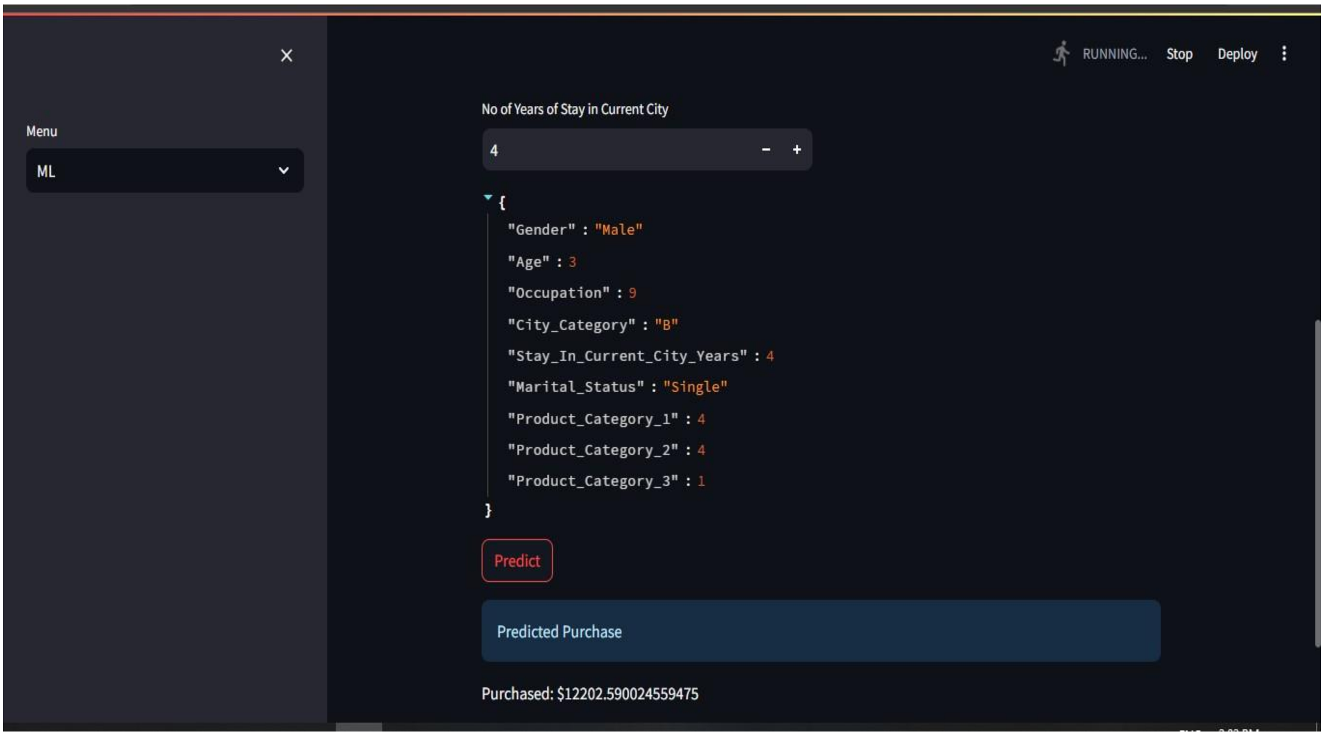

Fig 12.6 Predicted Purchase