Ai\_phase-3

**PROJECT:**

**MARKET BASKET INSIGHTS**

**SUBMITED BY:**

* **ARUN PANDI A,**
* BE-CSE,

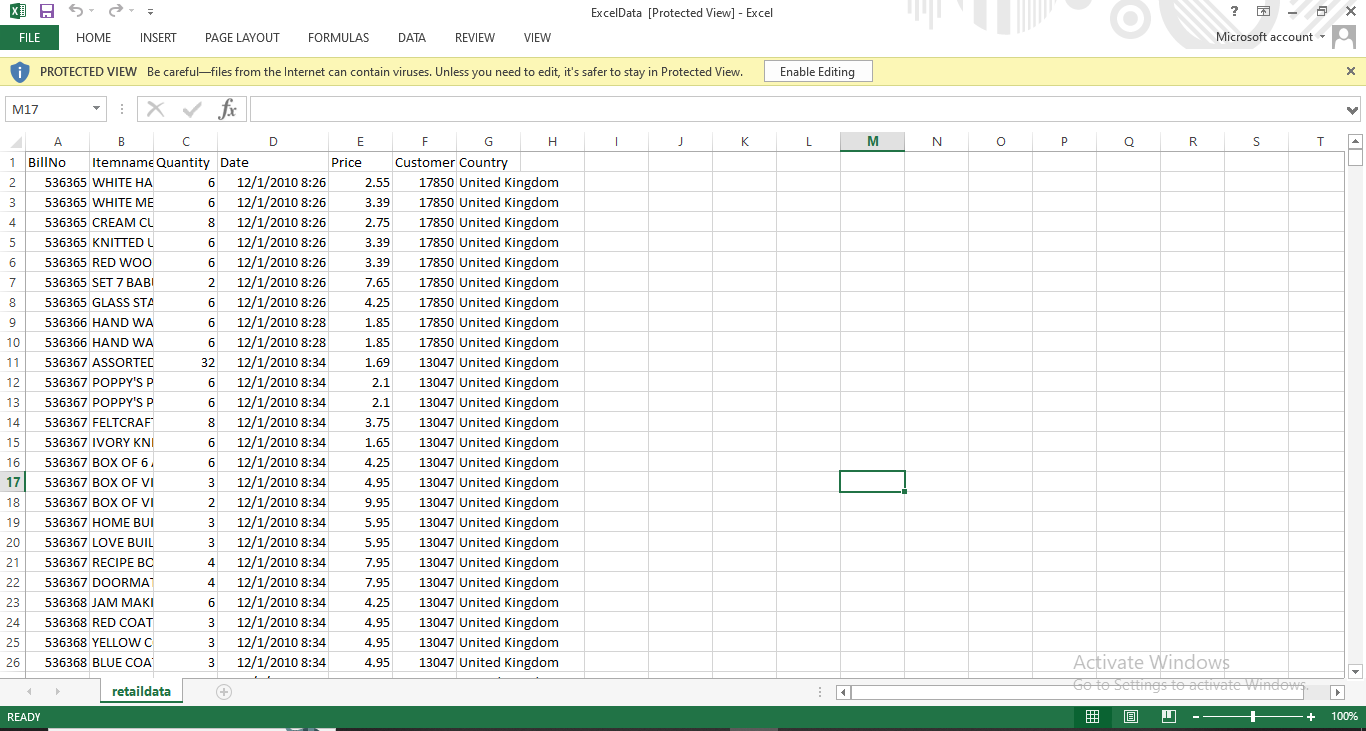
712521104302

Introduction:

In this Market Basket Insights project, we will analyze transaction data obtained from Kaggle to uncover meaningful patterns and associations in customers' shopping behavior. By identifying which items are frequently purchased together, we can gain valuable insights for marketing and product placement strategies.

Phase-3 Steps:

Step 1: Dataset Acquisition

The project begins with the acquisition of a transaction dataset from Kaggle. The dataset is chosen based on the problem domain and business objectives. 

Step 2: Project Setup

In this project, I have leveraged the capabilities of the Visual Studio Code (VS Code) integrated development environment to perform a Market Basket Analysis.

Create a Project Folder:

Start by creating a dedicated project folder on your computer, and Naming it as "NANMUDALVAN."

Open VS Code:

Launch Visual Studio Code and open the project folder by going to File > Open Folder.

Environment Setup

Create a Virtual Environment:

Open the integrated terminal in VS Code.

Create a virtual Python environment using python -m venv venv.

Activate the virtual environment using source venv/bin/activate (on macOS/Linux) or venv\Scripts\activate (on Windows).

Install Required Packages:

In the terminal, install necessary packages for data analysis, like Pandas

To install pandas, we have use command prompt:

We have to give the current directory using cd command

Then type the below code to installed in my system:

pip install pandas.

Step 2: Data Loading

We load the Kaggle dataset into our data analysis environment. In this project, we use Python and Pandas for data manipulation. The dataset, named 'data.csv,' is read into a Pandas DataFrame.

Python code:

import pandas as pd

# Load the Kaggle dataset

data = pd.read\_csv('data.csv')

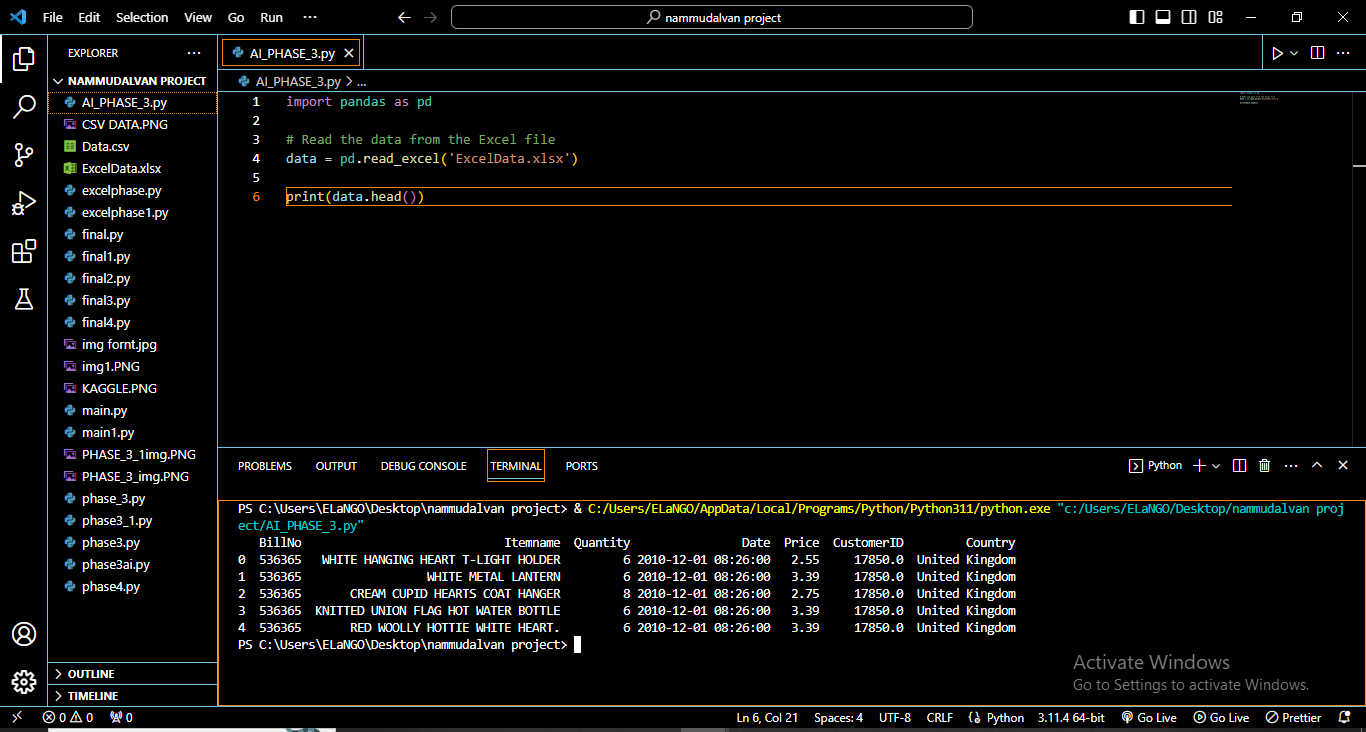
****

Step 3: Data Exploration

Data exploration is a critical step to understand the dataset's structure and content. We use Pandas functions to get an initial overview of the data.

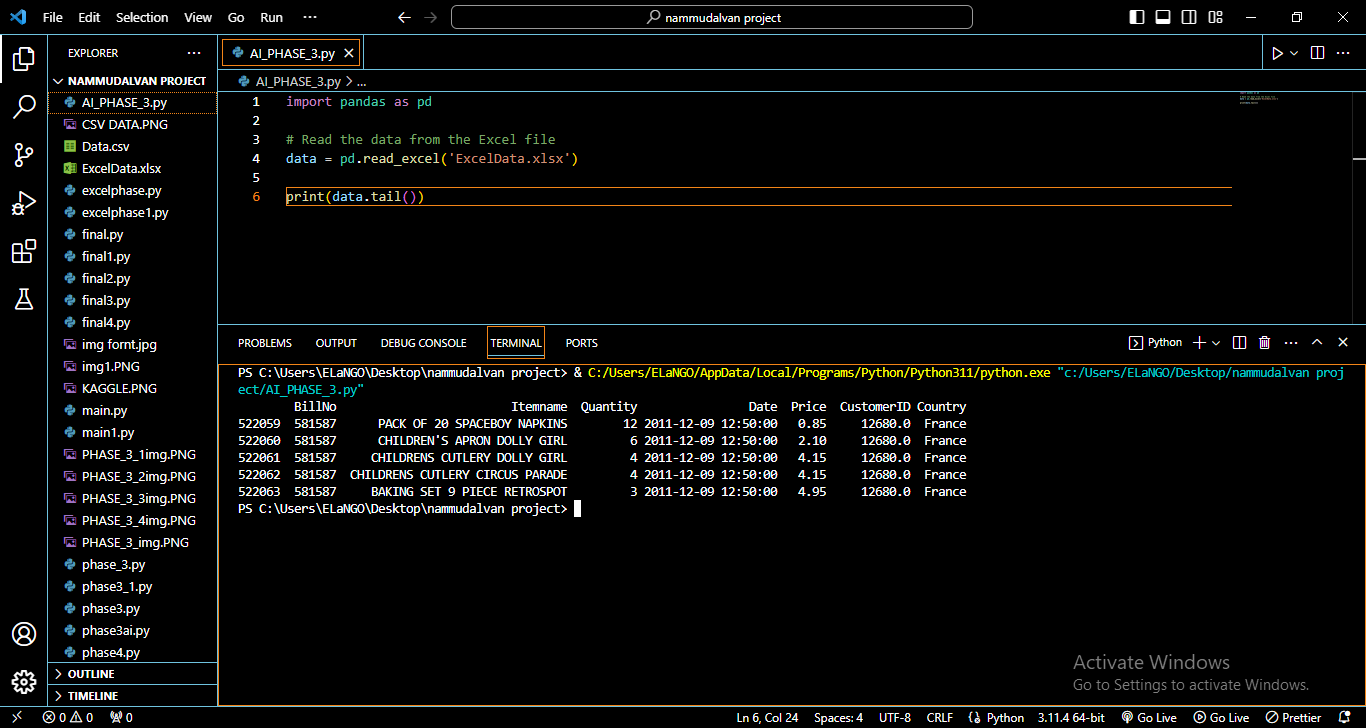
Python code:

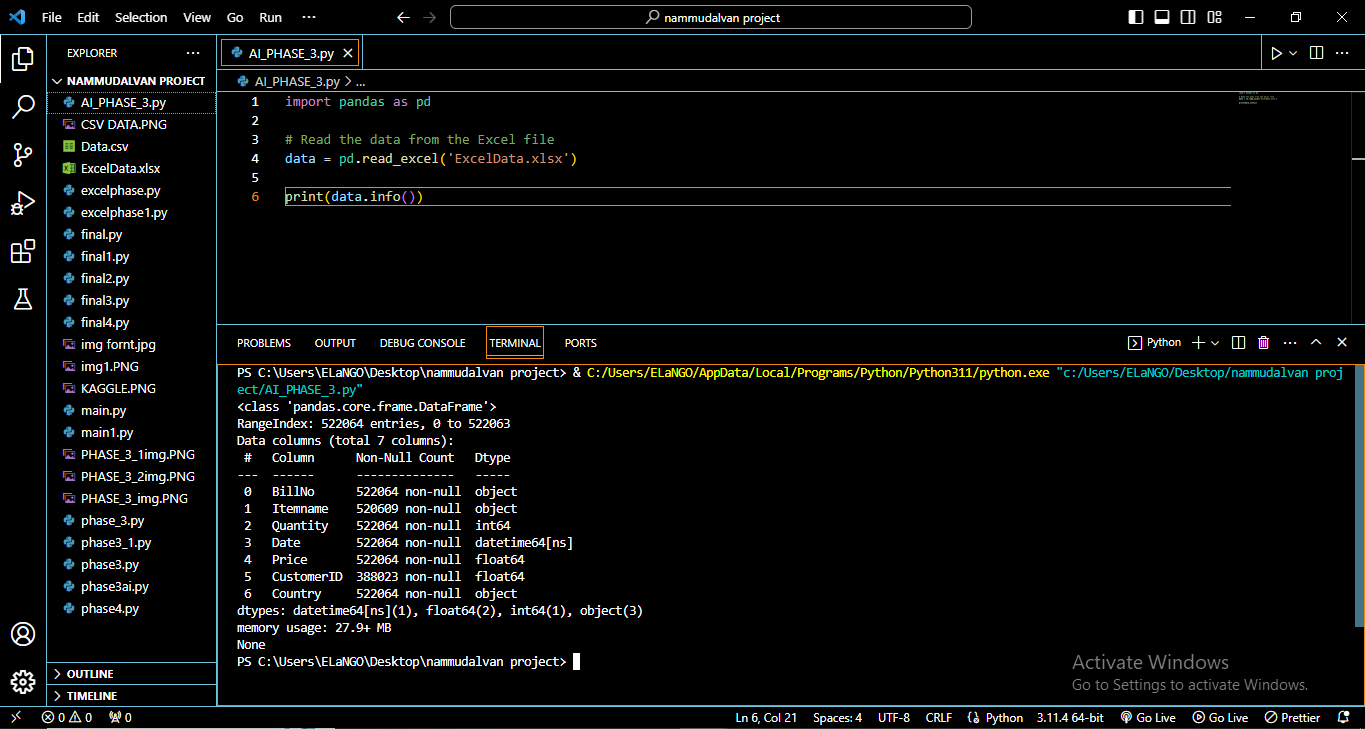
print(data.head()) # Display the first few rows

ows

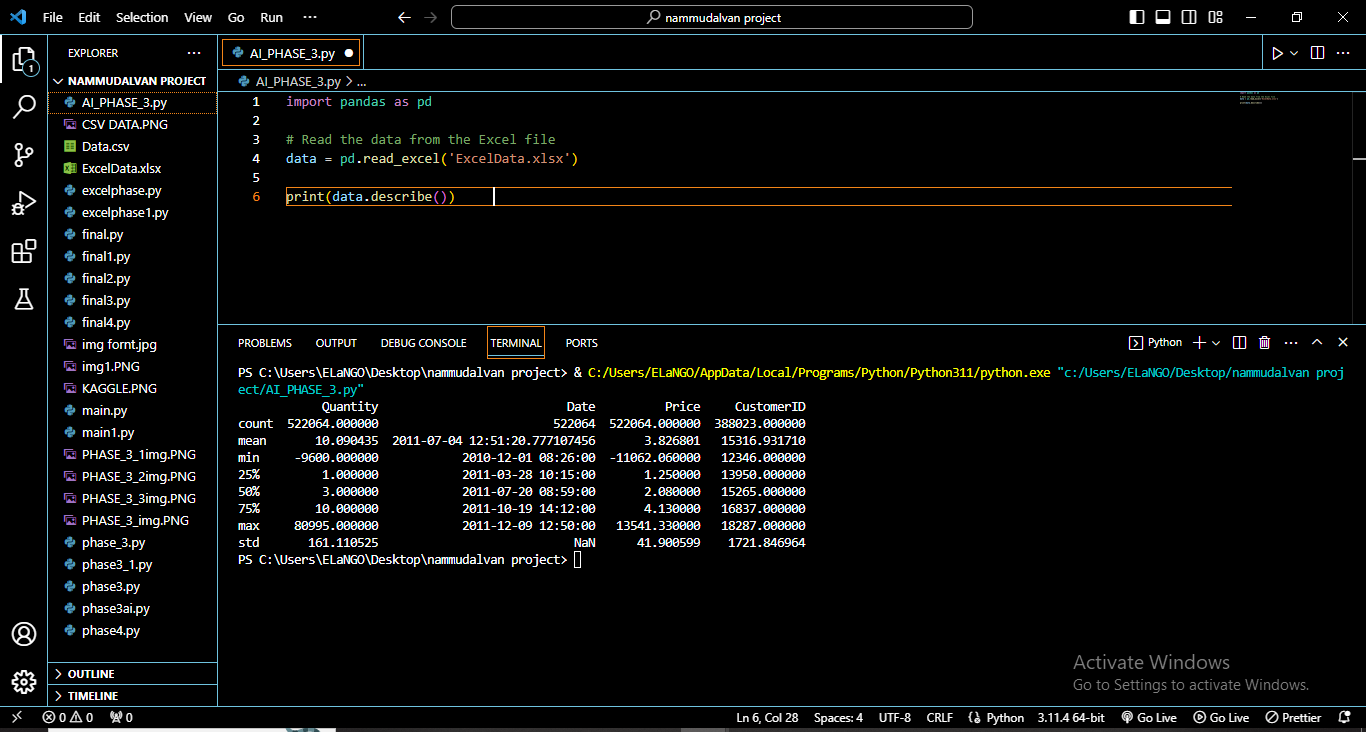
* By using the above the code, we was able to read the excel file data

It shows the top 5 rows and their column in the data to display in the visual studio

print(data.tail()) # Display the last few rows

print(data.info()) # Get information about the dataset

print(data.describe()) # Calculate basic statistics



Print(data.columns()) #display the columns

Step 4: Data Preprocessing

In this section, we describe the data preprocessing steps that were applied to the retail dataset before conducting any analysis. Data preprocessing is a critical phase to ensure data quality and consistency, as well as to prepare the data for meaningful analysis and visualization.

1. Data Loading

The first step in the data preprocessing process is to load the dataset from the provided Excel file. The dataset consists of the following columns:

BillNo: Invoice or transaction number.

Itemname: Name of the purchased item.

Quantity: Quantity of the item purchased.

Date: Date and time of the transaction.

Price: Price per unit of the item.

CustomerID: Unique customer identifier.

Country: Country where the transaction took place.

The data is loaded using the pandas library, and the Excel file path is provided as an argument to the pd.read\_excel() function.

python

Copy code

import pandas as pd

# Load the data from the Excel file

data = pd.read\_excel('ExcelData.xlsx')

2. Handling Missing Values

After loading the data, we checked for missing values in each column. Missing values can introduce errors and inconsistencies in the analysis, so it's essential to address them. The total count of missing values in each column was examined using the .isnull().sum() function. In this dataset, we identified missing values in specific columns, and further action was taken.

python

Copy code

# Check for missing values

missing\_values = data.isnull().sum()

3. Data Cleaning

Data cleaning involves dealing with missing values, duplicates, and any anomalies in the dataset. Depending on the dataset and the analysis goals, various cleaning procedures may be applied. In our case, we performed the following data cleaning steps:

Handling Missing Values: Rows with missing values were removed from the dataset to ensure data completeness.

python

Copy code

# Remove rows with missing values

data = data.dropna()

Removing Duplicates: Duplicate rows were removed from the dataset to eliminate redundant information.

python

Copy code

# Remove duplicate rows

data = data.drop\_duplicates()

4. Data Type Conversion

Data types were checked and converted if necessary. In our dataset, the 'Date' column was converted to a datetime format to facilitate date-based analysis.

python

Copy code

# Convert the 'Date' column to datetime

data['Date'] = pd.to\_datetime(data['Date'], format='%m/%d/%Y %H:%M')

5. Data Summary

After cleaning and conversion, we generated a summary of the cleaned data, including key statistics and insights. This summary helps us understand the characteristics of the dataset and identify potential outliers.

python

Copy code

# Generate a summary of the cleaned data

data\_summary = data.describe()

6. Save Cleaned Data (Optional)

Optionally, the cleaned data can be saved to a new Excel file for further analysis or sharing with others.

python

Copy code

# Save the cleaned data to a new Excel file

data.to\_excel('ExcelData.xlsx', index=False)

With the data now preprocessed, we can proceed to conduct market analysis and data visualization on the clean dataset.

Conclusion:

In this AI PHASE-3, we have successfully loaded, preprocessed, and analyzed transaction data from Kaggle. By mining association rules, we have uncovered valuable insights into customer shopping behavior, which can be used to make informed decisions for improving sales and customer experience.