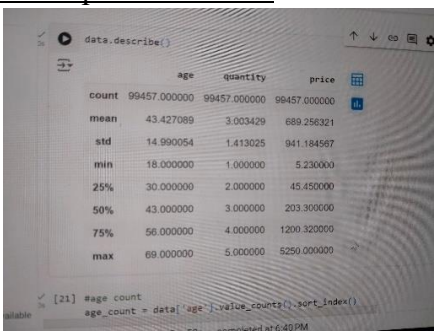
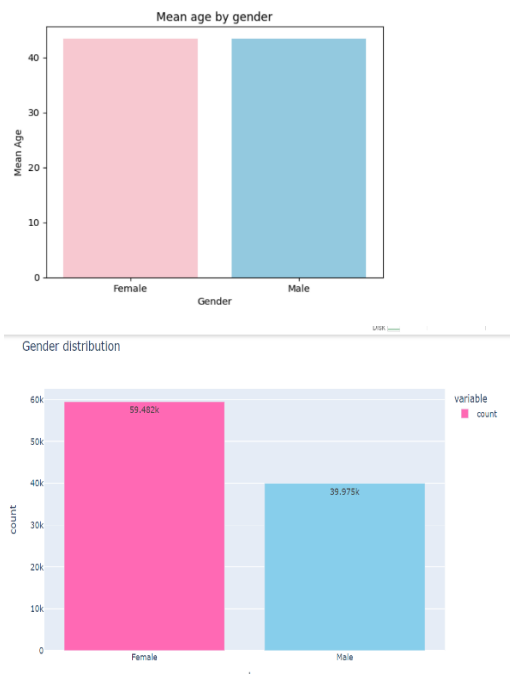
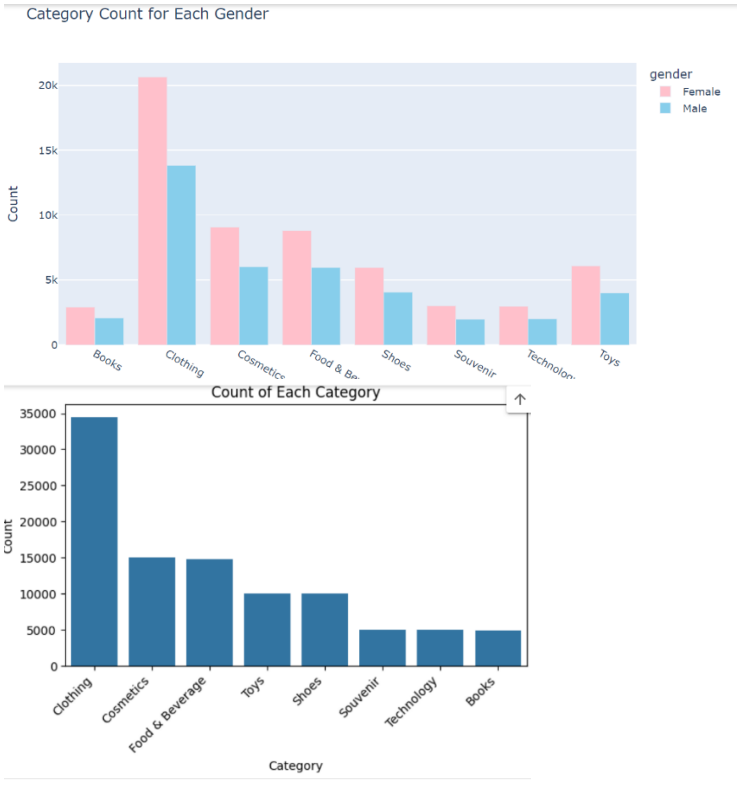
Data Collection and Preprocessing Phase

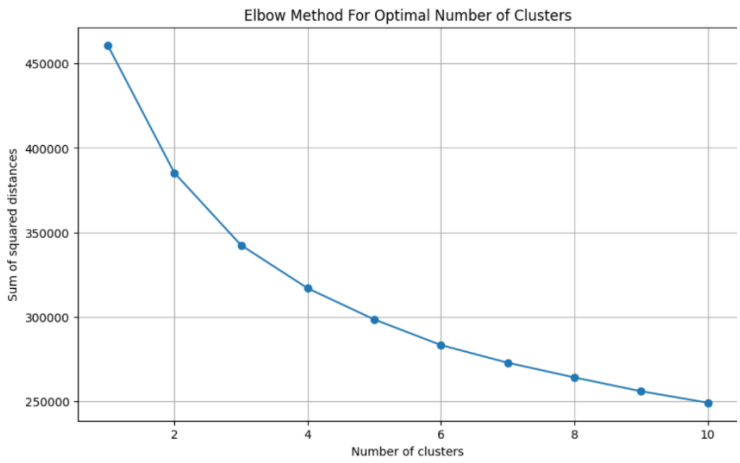
Date	06-07-2024
Team ID	739936
Project Title	Customer Shopping Segmentation by using machine learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing Report

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

Section	**Description**
Data Overview	Dimension: 99457rows × 10 columns Descriptive statistics: 
Univariate Analysis	

	
Bivaraite analysis	

Multivariate Analysis	

Outliers and Anomalies	-

Data Preprocessing Code Screenshots

Loading Data

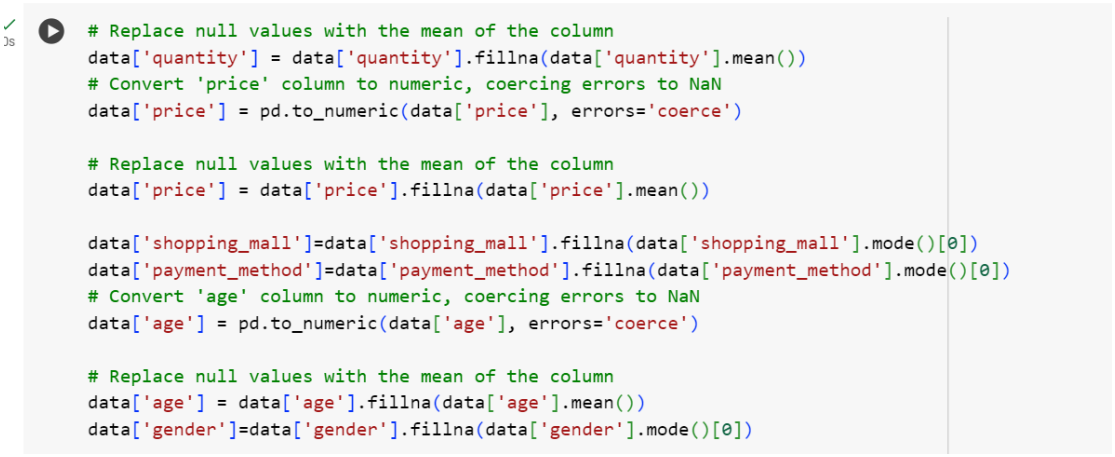
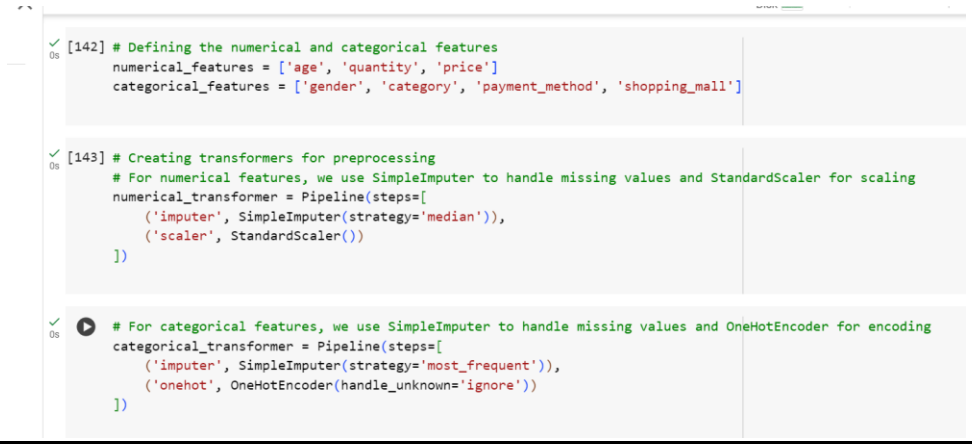
READ THE DATASET

```
data=pd.read_csv("/content/customer_shopping_data.csv")
data.shape
```

```
(99457, 10)
```

```
data.isnull()
```

invoice_no	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...

Handling Missing Data	 ``` # Replace null values with the mean of the column data['quantity'] = data['quantity'].fillna(data['quantity'].mean()) # Convert 'price' column to numeric, coercing errors to NaN data['price'] = pd.to_numeric(data['price'], errors='coerce') # Replace null values with the mean of the column data['price'] = data['price'].fillna(data['price'].mean()) data['shopping_mall']=data['shopping_mall'].fillna(data['shopping_mall'].mode()[0]) data['payment_method']=data['payment_method'].fillna(data['payment_method'].mode()[0]) # Convert 'age' column to numeric, coercing errors to NaN data['age'] = pd.to_numeric(data['age'], errors='coerce') # Replace null values with the mean of the column data['age'] = data['age'].fillna(data['age'].mean()) data['gender']=data['gender'].fillna(data['gender'].mode()[0]) ```
Data Transformation	 ``` [142] # Defining the numerical and categorical features numerical_features = ['age', 'quantity', 'price'] categorical_features = ['gender', 'category', 'payment_method', 'shopping_mall'] [143] # Creating transformers for preprocessing # For numerical features, we use SimpleImputer to handle missing values and StandardScaler for scaling numerical_transformer = Pipeline(steps=[('imputer', SimpleImputer(strategy='median')), ('scaler', StandardScaler())]) # For categorical features, we use SimpleImputer to handle missing values and OneHotEncoder for encoding categorical_transformer = Pipeline(steps=[('imputer', SimpleImputer(strategy='most_frequent')), ('onehot', OneHotEncoder(handle_unknown='ignore'))]) ```
Feature Engineering	Attached the codes in final submission.
Save Processed Data	-