

IE6700 Data Management for Analytics Milestone (Application in Python)

Group 5

Student 1: Aliya LNU

Student 2: Nixon Lobo

6315389069 (Tel of Student 1)

8573209201 (Tel of Student 2)

Lnu.ali@northeastern.edu

Lobo.ni@northeastern.edu

Percentage of Effort Contributed by Student1: 50%

Percentage of Effort Contributed by Student2: 50%

Signature of Student 1: Aliya LNU

Signature of Student 2: Nixon Lobo

Submission Date: 04/07/2024

```
[3]: #!pip install mysql-connector-python
```

Requirement already satisfied: mysql-connector-python in c:\users\ie6700\anaconda3\lib\site-packages (8.3.0)

```
[2]: import mysql.connector
```

```
# Define the connection parameters
connection_params = {
    'host': 'localhost',
    'user': 'root',
    'passwd': 'IE6700_Vm%24',
    'database': 'project'
}

try:
    # Connect to the MySQL server
    connection = mysql.connector.connect(**connection_params)

    if connection.is_connected():
        print("Connected to MySQL server")

        # Execute a simple query to test the connection
        cursor = connection.cursor()
        cursor.execute("SELECT VERSION()")
        db_version = cursor.fetchone()
        print("MySQL Server version:", db_version)

        # Close the cursor and connection
        cursor.close()
        connection.close()
        print("Connection closed")
    else:
        print("Connection failed")
except mysql.connector.Error as error:
    print("Error while connecting to MySQL", error)
```

Connected to MySQL server
MySQL Server version: ('8.0.36',)

```
[2]: import mysql.connector
import pandas as pd
```

```
# Connect to the MySQL database
connection = mysql.connector.connect(
    host='localhost',
    user='root',
    passwd='IE6700_Vm%24',
    database='project'
)

# Get a list of all tables in the database
cursor = connection.cursor()
cursor.execute("SHOW TABLES")
tables = cursor.fetchall()

# Initialize an empty dictionary to store DataFrames for each table
dfs = {}

# Iterate over each table and load its data into a DataFrame
for table in tables:
    table_name = table[0]
```

```
# Close the cursor and connection
cursor.close()
connection.close()
```

```
# Access the DataFrames for each table using dfs[table_name]
# For example:
# print(dfs['feedback'].head())
```

```
C:\Users\IE6700\AppData\Local\Temp\ipykernel_10816\737901593.py:24: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or data
base string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
df = pd.read_sql(query, connection)
```

```
[3]: import mysql.connector
import matplotlib.pyplot as plt

# Define the connection parameters
connection_params = {
    'host': 'localhost',
    'user': 'root',
    'passwd': 'IE6700_YmK24',
    'database': 'project'
}

try:
    # Connect to the MySQL server
    connection = mysql.connector.connect(**connection_params)

    if connection.is_connected():
        print("Connected to MySQL database")

        # Define and execute the query to retrieve the count of users in each city
        query = "SELECT City, COUNT(*) AS NumUsers FROM user_data GROUP BY City"
        cursor = connection.cursor()
        cursor.execute(query)
        data = cursor.fetchall()

        # Extract cities and corresponding user counts
        cities = [row[0] for row in data]
        user_counts = [row[1] for row in data]

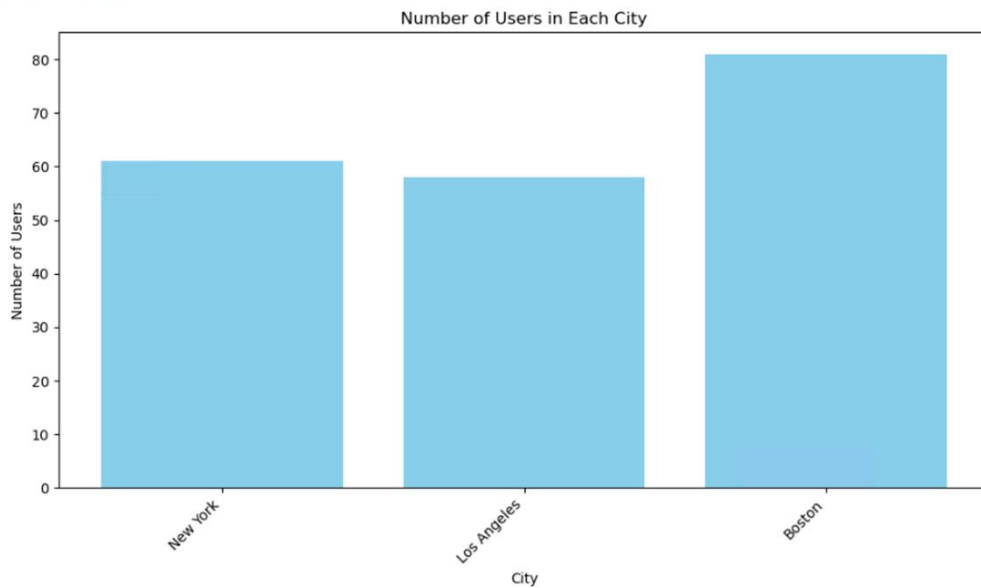
        # Create a bar chart
        plt.figure(figsize=(10, 6))
        plt.bar(cities, user_counts, color='skyblue')
        plt.xlabel('City')
        plt.ylabel('Number of Users')
        plt.title('Number of Users in Each City')
        plt.xticks(rotation=45, ha='right')
        plt.tight_layout()
        plt.show()

        # Close the cursor and connection
        cursor.close()
        connection.close()
        print("Connection closed")

    else:
        print("Connection failed")

except mysql.connector.Error as error:
    print("Error while connecting to MySQL", error)
```

Connected to MySQL database



Connection closed

```
[4]: import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

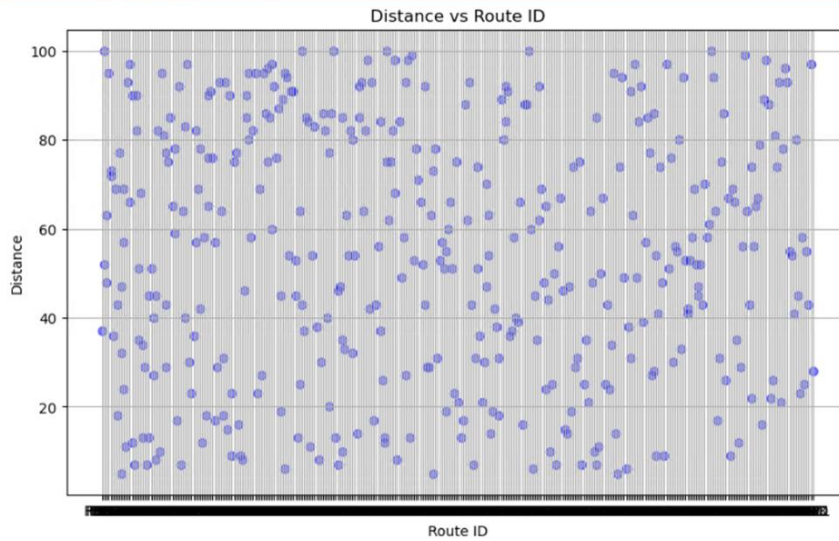
# Connect to the MySQL database
connection = mysql.connector.connect(
    host='localhost',
    user='root',
    passwd='IE6700_Vm%24',
    database='project'
)

# Define the SQL query to retrieve data
query = "SELECT RouteID, Distance FROM route"

# Execute the query and fetch the results into a DataFrame
df = pd.read_sql(query, connection)

# Close the database connection
connection.close()

# Create the scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(df['RouteID'], df['Distance'], color='blue', alpha=0.5)
plt.title('Distance vs Route ID')
plt.xlabel('Route ID')
plt.ylabel('Distance')
plt.grid(True)
plt.show()
```



```
[5]: import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

# Connect to the MySQL database
connection = mysql.connector.connect(
    host='localhost',
    user='root',
    passwd='IE6700_Vm%24',
    database='project'
)

# Define the SQL query to retrieve product prices
query = "SELECT Price FROM products"

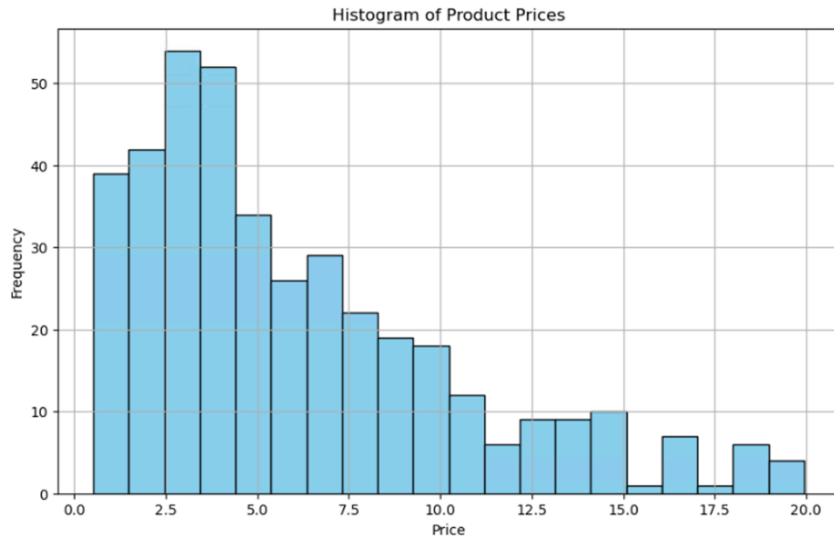
# Execute the query and fetch the results into a DataFrame
df = pd.read_sql(query, connection)

# Close the database connection
connection.close()

# Create the histogram
plt.figure(figsize=(10, 6))
plt.hist(df['Price'], bins=20, color='skyblue', edgecolor='black')
plt.title('Histogram of Product Prices')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```

C:\Users\IE6700\AppData\Local\Temp\ipykernel_10816\870975281.py:17: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or data

```
df = pd.read_sql(query, connection)
```



```
[6]: import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

# Connect to the MySQL database
connection = mysql.connector.connect(
    host='localhost',
    user='root',
    passwd='IE6700_Vm%24',
    database='project'
)

# Define the SQL query to retrieve product categories and their counts
query = "SELECT Category, COUNT(*) AS Count FROM products GROUP BY Category"

# Execute the query and fetch the results into a DataFrame
df = pd.read_sql(query, connection)

# Close the database connection
connection.close()

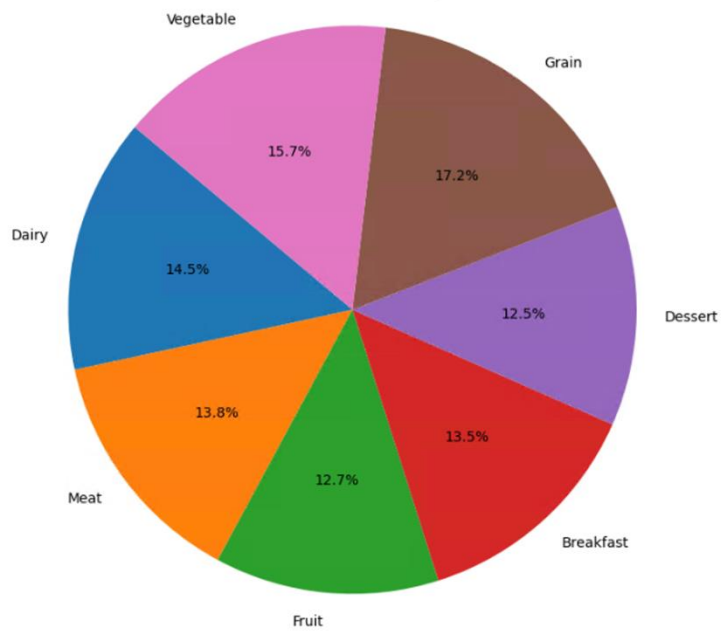
# Create the pie chart
plt.figure(figsize=(8, 8))
plt.pie(df['Count'], labels=df['Category'], autopct='%1.1f%%', startangle=140)
plt.title('Pie Chart of Product Categories')
plt.axis('equal')
plt.show()
```

C:\Users\IE6700\AppData\Local\Temp\ipykernel_10816\2541091634.py:17: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.

```
df = pd.read_sql(query, connection)
```

```
df = pd.read_sql(query, connection)
```

Pie Chart of Product Categories



```
[7]: import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

# Connect to the MySQL database
connection = mysql.connector.connect(
    host='localhost',
    user='root',
    passwd='IE6700_Vm%24',
    database='project'
)

# Define the SQL query to retrieve product prices
query = "SELECT Price FROM products"

# Execute the query and fetch the results into a DataFrame
df = pd.read_sql(query, connection)

# Close the database connection
connection.close()

# Create the box plot
plt.figure(figsize=(8, 6))
plt.boxplot(df['Price'], vert=False)
plt.title('Box Plot of Product Prices')
```

```

user='root',
passwd='IE6700_Vm%24',
database='project'
)

# Define the SQL query to retrieve product prices
query = "SELECT Price FROM products"

# Execute the query and fetch the results into a DataFrame
df = pd.read_sql(query, connection)

# Close the database connection
connection.close()

# Create the box plot
plt.figure(figsize=(8, 6))
plt.boxplot(df['Price'], vert=False)
plt.title('Box Plot of Product Prices')
plt.xlabel('Price')
plt.show()

```

C:\Users\IE6700\AppData\Local\Temp\ipykernel_10816\3270895359.py:17: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
df = pd.read_sql(query, connection)

