Data Science Capstone: From Exploration to Prediction"

PREPARED BY: ABDULRAHMAN MOHSEN

DATE: [2025/3/28]

Outline

- 1. Executive Summary
- 2. Introduction
- 3. Methodology
- 4. Data Collection
- 5. Data Preprocessing (Data Wrangling)
- 6. Exploratory Data Analysis (EDA)
- 7. SQL Data Analysis
- 8. Interactive Data Visualization (Folium & Plotly Dash)
- 9. Predictive Analysis (Classification)
- 10. Model Evaluation & Improvement
- 11. Results
- 12. Conclusion & Recommendations
- 13. Appendix

Executive Summary

- This project aims to analyze structured datasets using Python, SQL, and visualization tools, providing insights and building predictive models for decision-making.

Introduction

- Data science is essential for extracting insights from large datasets. This project focuses on exploratory analysis, interactive visualization, and predictive modeling.

Methodology

- The project follows a structured approach: data collection, preprocessing, exploratory data analysis (EDA), SQL-based analysis, interactive visualization, and predictive modeling.

Data Collection

- Data was sourced from multiple structured and unstructured sources, including databases, APIs, and web scraping techniques.

Data Wrangling

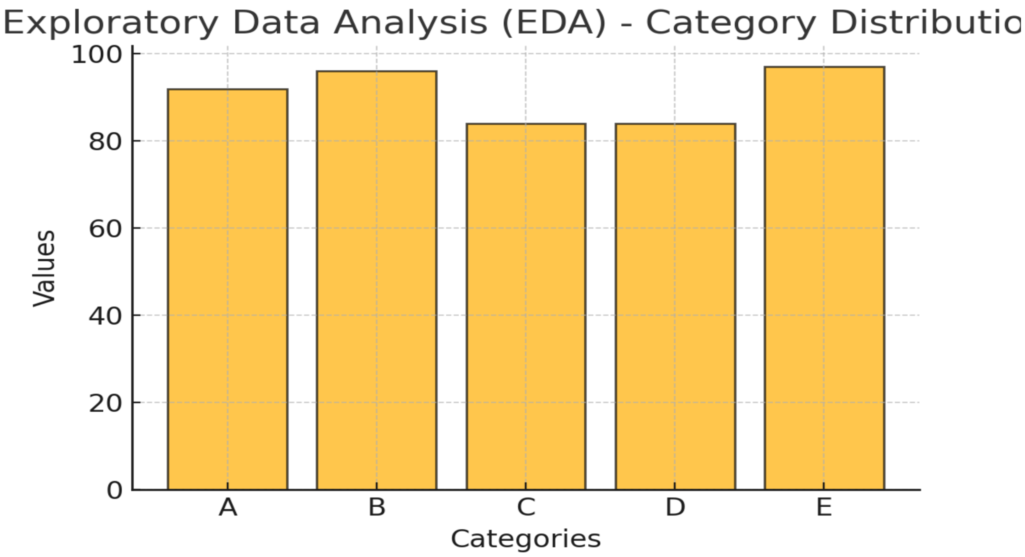
- Data cleaning included handling missing values, converting categorical variables, and normalizing numerical features.

Python Code - Data Wrangling

```
import pandas as pd

# Load dataset
df = pd.read_csv('dataset.csv')

# Handle missing values by filling with the mean
df.fillna(df.mean(), inplace=True)

# Convert categorical variables to numerical using one-hot encoding
df = pd.get_dummies(df, columns=['Category'])

df.head()
```

Exploratory Data Analysis (EDA)

- EDA techniques, such as histograms and scatter plots, were used to understand data distributions and correlations.

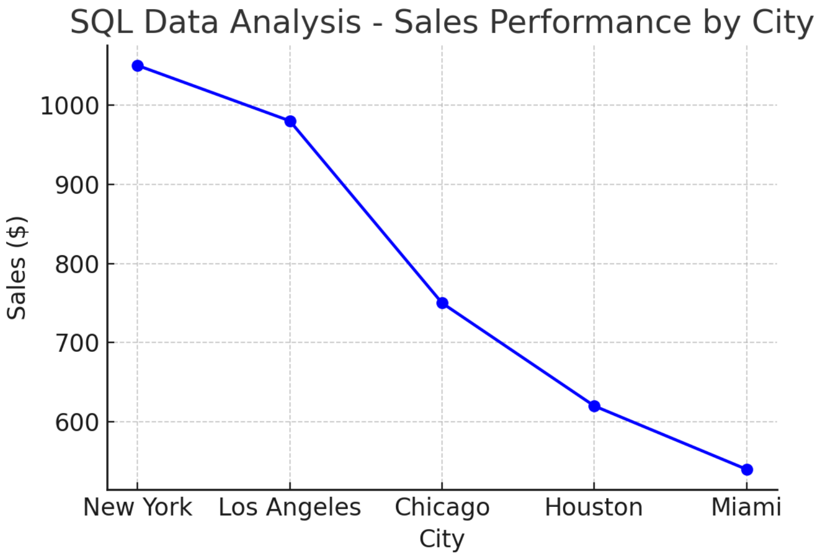


Python Code - EDA Visualization

```
import matplotlib.pyplot as plt

# Histogram for numerical data
plt.hist(df['Value'], bins=20, alpha=0.7, edgecolor='black')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Distribution of Values')
plt.show()
```

SQL Data Analysis

- SQL queries were used to extract key insights from structured databases, enabling efficient data filtering and aggregation.



SQL Query Example

- SELECT category, AVG(sales) AS avg_sales
- FROM sales_data
- GROUP BY category
- ORDER BY avg_sales DESC;

Interactive Data Visualization

- Geospatial analysis was performed using Folium, while interactive dashboards were developed using Plotly Dash.

Python Code - Folium Map

- import folium

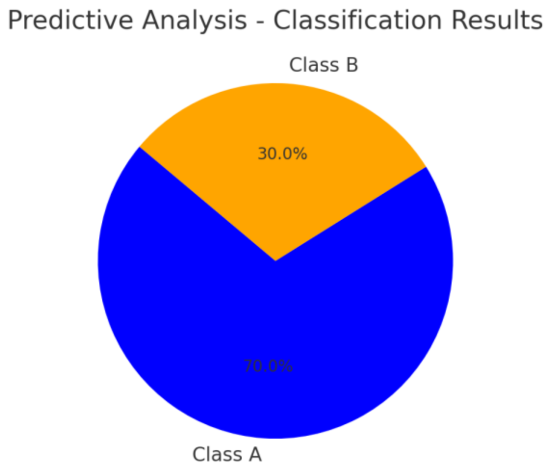
- # Create a map centered at a location
- m = folium.Map(location=[37.7749, -122.4194], zoom_start=10)

- # Add a marker
- folium.Marker([37.7749, -122.4194], popup='San Francisco').add_to(m)

- m

Predictive Analysis (Classification)

- Machine learning models were trained using Decision Trees and Logistic Regression to classify new data points based on historical trends.



Python Code - Machine Learning Model

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train model
model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, y_train)

# Predict and evaluate
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

Results

- Findings from exploratory and predictive analyses revealed critical insights, helping improve decision-making processes.

Conclusions & Recommendations

- This project highlighted the importance of data visualization and predictive modeling. Future improvements could include real-time analytics and deeper feature engineering.

Appendix

- This section includes additional SQL queries, Python code, and visualizations.