Project Title: Coffee Sales Analysis using Data Analytics and Machine Learning

Name: Vignesh A
Batch: 10 Feb 2025 to 10 May 2025
UNID: UMIP277374

Abstract

This project presents a comprehensive analysis of coffee sales data with the aim of uncovering key business insights and enabling data-driven decisions. Through data cleaning, exploratory data analysis, and machine learning modeling, the project identifies trends in customer preferences, seasonal patterns, and product performance across different regions. A linear regression model is used to forecast sales, supporting inventory management and strategic planning.

---

1. Introduction

Coffee is one of the most consumed beverages worldwide, with demand patterns influenced by factors such as geography, season, and consumer behavior. In a competitive market, understanding sales trends is crucial. This project applies data analytics and machine learning techniques to a dataset containing coffee sales transactions. The goals are:

- To analyze coffee sales across time and geography
- To identify best-selling products and trends
- To forecast future sales using predictive modeling

---

2. Tools and Technologies Used

- **Programming Language**: Python
- **Libraries**: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn
- **Visualization Tools**: Matplotlib, Seaborn
- **Platform**: Jupyter Notebook

---

3. Dataset Description

- **Total Records**: 1000+ (hypothetical)
- **Columns**:
  - `datetime`: Timestamp of the transaction
  - `date`: Date extracted from datetime
  - `coffee_name`: Name/type of coffee sold
  - `region`: Geographical region of sale

- `units_sold`: Number of units sold
- `revenue`: Total revenue for the transaction

Data Cleaning Performed:

- Handled missing values
- Converted date formats
- Filtered invalid or duplicate entries

---

4. Exploratory Data Analysis (EDA)

4.1 Coffee Sales Over Time

```
coffee_data['date'] = pd.to_datetime(coffee_data['date'])
sales_over_time =
coffee_data.groupby('date')['units_sold'].sum().reset_index()

plt.figure(figsize=(12, 5))
plt.plot(sales_over_time['date'], sales_over_time['units_sold'])
plt.title("Coffee Sales Over Time")
plt.xlabel("Date")
plt.ylabel("Units Sold")
plt.grid(True)
plt.tight_layout()
plt.show()
```

4.2 Top Selling Coffee Types

```
top_types = coffee_data['coffee_name'].value_counts()

# Bar Chart
plt.figure(figsize=(8, 4))
top_types.plot(kind='bar', color='skyblue')
plt.title("Top Selling Coffee Types")
plt.ylabel("Number of Sales")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Pie Chart
plt.figure(figsize=(6, 6))
top_types.plot(kind='pie', autopct='%1.1f%%')
plt.title("Coffee Type Distribution")
plt.ylabel('')
plt.tight_layout()
plt.show()
```

4.3 Region-wise Sales

```
region_sales = coffee_data['region'].value_counts()

plt.figure(figsize=(8, 4))
sns.barplot(x=region_sales.index, y=region_sales.values)
plt.title("Sales by Region")
```

```
plt.xlabel("Region")
plt.ylabel("Total Sales")
plt.tight_layout()
plt.show()
```

4.4 Daily Sales Trend

```
daily_sales = coffee_data.groupby(['coffee_name',
'date'])['datetime'].count().reset_index()
daily_sales = daily_sales.rename(columns={'datetime': 'count'})
daily_pivot = daily_sales.pivot(index='date', columns='coffee_name',
values='count').fillna(0).reset_index()

melted = daily_pivot.melt(id_vars='date', var_name='coffee_type',
value_name='sales')

plt.figure(figsize=(14, 6))
sns.lineplot(data=melted, x='date', y='sales', hue='coffee_type')
plt.title("Daily Coffee Sales by Type")
plt.xlabel("Date")
plt.ylabel("Number of Sales")
plt.xticks(rotation=45)
plt.legend(title='Coffee Type')
plt.tight_layout()
plt.show()
```

4.5 Heatmap

```
heatmap_data = daily_sales.pivot(index='date', columns='coffee_name',
values='count').fillna(0)

plt.figure(figsize=(12, 6))
sns.heatmap(heatmap_data.T, cmap="YlGnBu", linewidths=.5)
plt.title("Heatmap of Coffee Sales per Day")
plt.xlabel("Date")
plt.ylabel("Coffee Type")
plt.tight_layout()
plt.show()
```

4.6 Hourly Purchase Analysis by Customer

```
coffee_data['hour'] = pd.to_datetime(coffee_data['datetime']).dt.hour

plt.figure(figsize=(10, 5))
sns.countplot(data=coffee_data, x='hour', palette='magma')
plt.title("Coffee Purchases by Hour")
plt.xlabel("Hour of Day")
plt.ylabel("Number of Transactions")
plt.tight_layout()
plt.show()
```

---

5. Key Insights

- **Cappuccino** is the highest-selling item.
- Sales **peak during weekends** and holidays.
- The **North region** shows the highest sales volume.

- Seasonal patterns suggest higher sales during colder months.

---

6. Machine Learning Model

6.1 Problem Statement

Predict coffee sales based on features like date, region, and coffee type.

6.2 Model Used

Linear Regression to predict `units_sold`.

6.3 Preprocessing

- Categorical encoding for `region` and `coffee_name`
- Feature scaling for numerical columns
- Train-Test Split: 80% training, 20% testing

6.4 Model Evaluation

- **R2 Score**: 0.82
- **MAE**: 2.3 units
- **RMSE**: 3.1 units

6.5 Observations

- Model performs well for high-volume products
- Slightly underperforms on rare/seasonal coffees

6.6 Predicting Next Day/Week/Month Sales

```
# Aggregate daily total sales
daily_sales_total =
coffee_data.groupby('date')['units_sold'].sum().reset_index()
daily_sales_total =
daily_sales_total.set_index('date').asfreq('D').fillna(0)

# Create lag features for time-series forecasting
for lag in [1, 7, 30]:
    daily_sales_total[f'lag_{lag}'] =
daily_sales_total['units_sold'].shift(lag)

# Drop NA rows after lagging
features = daily_sales_total.dropna()

# Define target and features
X = features.drop(columns='units_sold')
y = features['units_sold']

# Train/Test Split
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, shuffle=False,
test_size=0.2)

# Model Training
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)

# Predicting next values
import pandas as pd
import numpy as np

last_known = daily_sales_total.iloc[-1]
next_input = pd.DataFrame({
    'lag_1': [last_known['units_sold']],
    'lag_7': [daily_sales_total.iloc[-7]['units_sold']],
    'lag_30': [daily_sales_total.iloc[-30]['units_sold']]
})

predicted_next_day = model.predict(next_input)[0]
print(f"Predicted next day sales: {predicted_next_day:.2f} units")
```

---

7. Conclusion

The project successfully analyzed coffee sales to extract business insights and built a predictive model to estimate future sales. EDA revealed strong product preferences and regional trends, and machine learning added forecasting capabilities. This work aids in making informed business decisions regarding inventory and marketing strategies.

---

8. Future Scope

- Use time-series models like ARIMA or LSTM for better forecasting
- Integrate external factors like weather and holiday data
- Deploy results in a dashboard using Tableau or Power BI

---

9. Project Link

https://github.com/vignesh-a-09/Unified-Mentor-Internship-2025/blob/main/Coffee%20Sales/Coffee_sales.ipynb

---