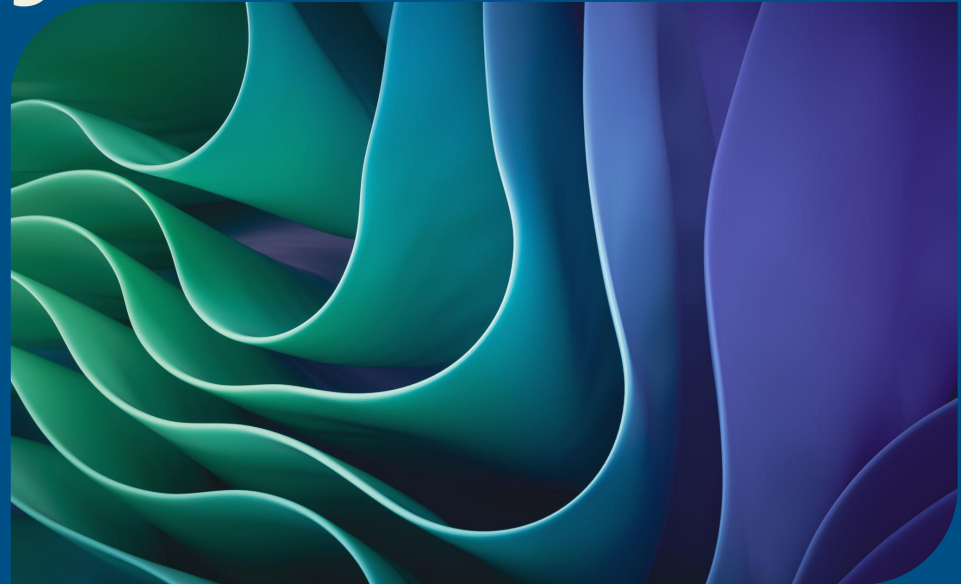


Capstone Project Data Analysis

Tyler Singh
Ella Johnston
Divya Bengali



1. **Data Loading and Exploration**
2. **Data Cleaning and Preprocessing**
3. **Statistical Analysis**
4. **Data Visualization**
5. **Machine Learning Implementation**
6. **Business Insights and Recommendations**

1. **Data Loading and Exploration**
2. **Data Cleaning and Preprocessing**
3. **Statistical Analysis**
4. **Data Visualization**
5. **Machine Learning Implementation**
6. **Business Insights and Recommendations**

Purpose

- Ensures that we understand the dataset before cleaning, analysis, or machine learning
- About getting familiar with the data

Key Actions

- Used **pandas** to import the CSV
- For key findings:
 - `pd.read_csv('sales_data.csv')`
 - `df.shape`
 - `df.columns.tolist()`
 - `df.dtypes`
 - `df.isnull/duplicated`
 - `df.describe`

Key Findings

- Dataset shape
- Columns
- Data Types
- Missing Values Per Column
- Unique Values
- Count, Mean, Standard Deviation, Min, Max

1. Data Loading and Exploration
2. **Data Cleaning and Preprocessing**
3. Statistical Analysis
4. Data Visualization
5. Machine Learning Implementation
6. Business Insights and Recommendations

```
# Converting date columns to datetime objects
df['registration_date'] = pd.to_datetime(df['registration_date'])
df['purchase_date'] = pd.to_datetime(df['purchase_date'])

# Create derived features

# Derived Feature 2: Recalculate Customer Lifespan in Days (Fixing 'customer_lifespan' error)
df['customer_lifespan_days'] = (df['purchase_date'] - df['registration_date']).dt.days

# Outlier detection and handling

# Numerical columns to check for outliers
numerical_cols = ['income', 'revenue', 'customer_lifespan_days']

for col in numerical_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
```

```
# Handle missing values before encoding
print("--- Handling Missing Values (Imputation) ---")

# Numerical Imputation (using mean)
df['age'].fillna(df['age'].mean(), inplace=True)
df['income'].fillna(df['income'].mean(), inplace=True)
df['revenue'].fillna(df['price'] * df['quantity'], inplace=True)

# Categorical Imputation (using mode)
# Use .mode()[0] to get the most frequent value
for col in ['gender', 'location']:
    df[col].fillna(df[col].mode()[0], inplace=True)

print(df.isnull().sum())
print(df.head())

--- Handling Missing Values (Imputation) ---
customer_id      0
age              0
gender           0
location         0
income           0
registration_date 0
purchase_date    0
product_category 0
brand            0
price            0
quantity         0
revenue          0
purchase_frequency 0
avg_order_value  0
customer_lifespan 0
customer_lifespan_days 0
dtype: int64
```

1. Data Loading and Exploration
2. Data Cleaning and Preprocessing
3. **Statistical Analysis**
4. Data Visualization
5. Machine Learning Implementation
6. Business Insights and Recommendations

Hypothesis Testing

```
# Hypothesis testing:
print(f"Null Hypothesis (H0): There is no difference in mean transaction revenue between Male and Female customers.")
print(f"Alternative Hypothesis (HA): There is a significant difference in mean transaction revenue between Male and Female customers.")

male_rev = df[df['gender'] == 'Male']['revenue']
female_rev = df[df['gender'] == 'Female']['revenue']
t_stat, p_value = stats.ttest_ind(male_rev, female_rev, equal_var=False)

print(f"\nTwo-Sample T-Test Statistic: {t_stat:.4f}")
print(f"P-value: {p_value:.4f}")

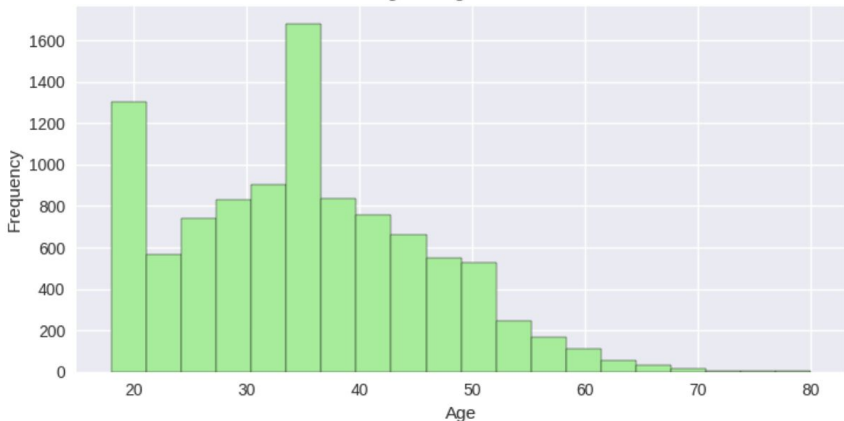
if p_value < ALPHA:
    print(f"Conclusion: Reject H0. The difference in mean transaction revenue between Male and Female customers is statistically significant (p < {ALPHA}).")
else:
    print(f"Conclusion: Fail to Reject H0. There is no statistically significant difference in mean transaction revenue (p >= {ALPHA}).")
```

Null Hypothesis (H0): There is no difference in mean transaction revenue between Male and Female customers.
Alternative Hypothesis (HA): There is a significant difference in mean transaction revenue between Male and Female customers.

Two-Sample T-Test Statistic: 0.5940
P-value: 0.5525
Conclusion: Fail to Reject H0. There is no statistically significant difference in mean transaction revenue (p >= 0.05).

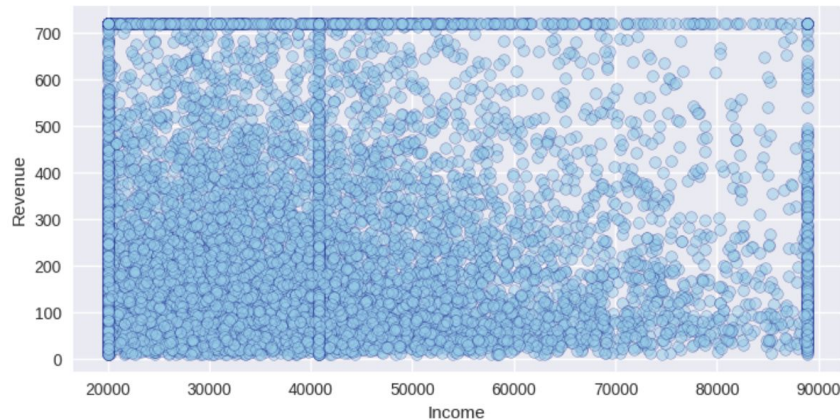
1. Data Loading and Exploration
2. Data Cleaning and Preprocessing
3. Statistical Analysis
4. **Data Visualization**
5. Machine Learning Implementation
6. Business Insights and Recommendations

Histogram: Age Distribution

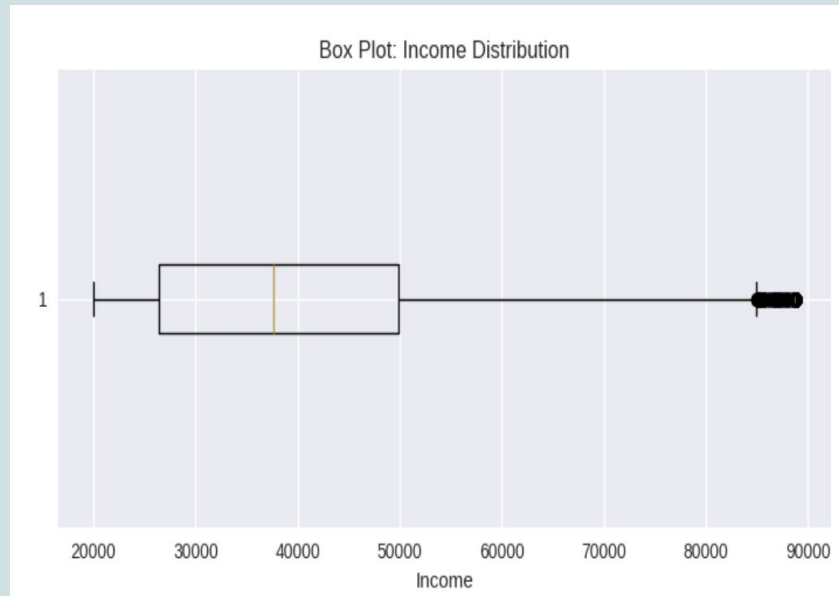


```
axes[0].hist(df['age'], bins=20,
color='lightgreen',
edgecolor='black')
axes[0].set_title("Histogram: Age
Distribution")
axes[0].set_xlabel("Age")
axes[0].set_ylabel("Frequency")
```

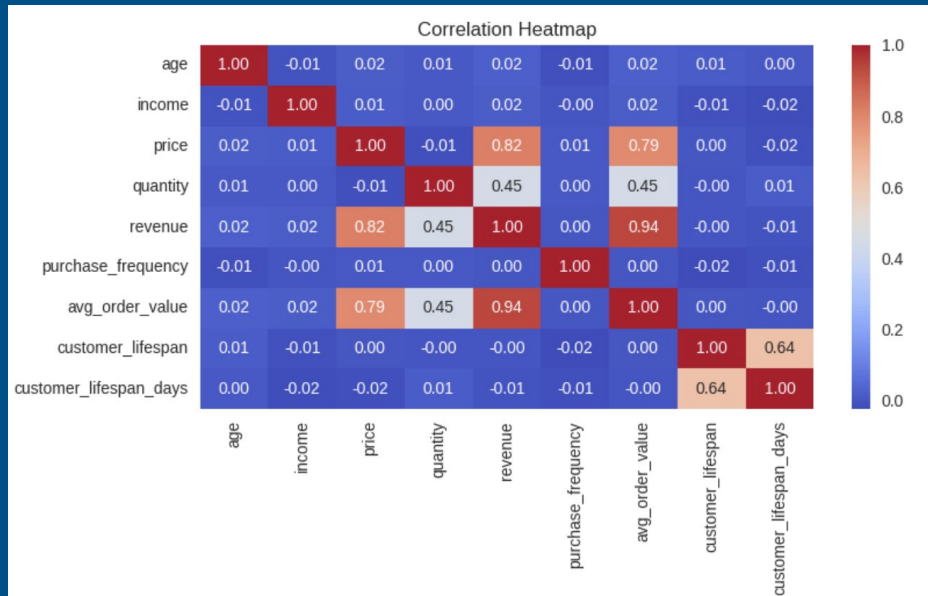
Scatter Plot: Income vs Revenue



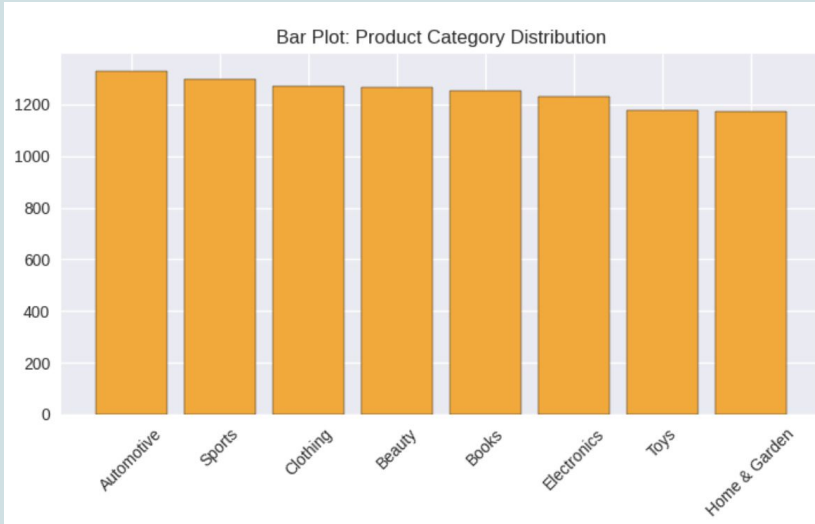
```
axes[1].scatter(df['income'],
df['revenue'], alpha=0.5,
c='skyblue', edgecolors='navy')
axes[1].set_title("Scatter Plot:
Income vs Revenue")
axes[1].set_xlabel("Income")
axes[1].set_ylabel("Revenue")
```



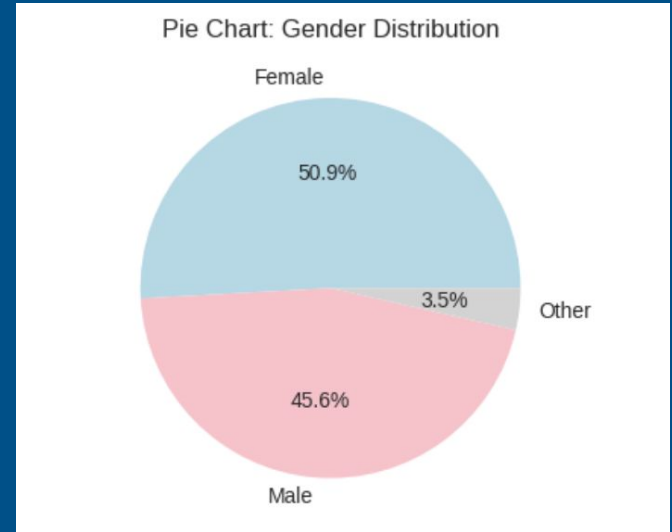
```
axes[2].boxplot(df['income'],
vert=False)
axes[2].set_title("Box Plot:
Income Distribution")
axes[2].set_xlabel("Income")
```



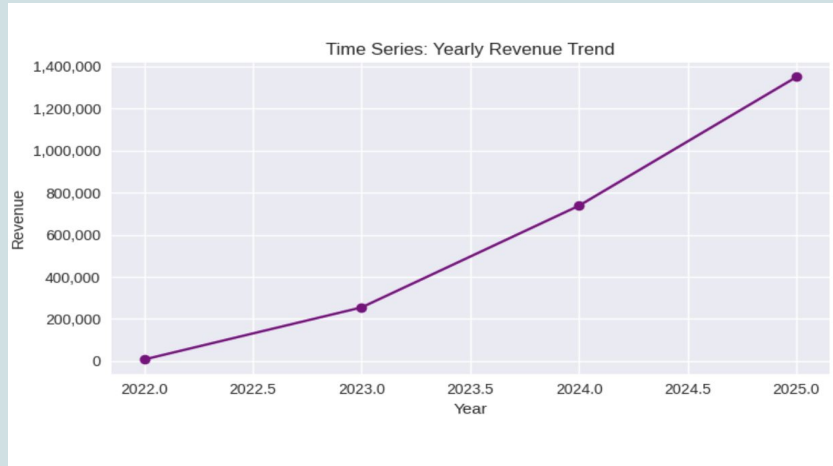
```
numeric_df =
df.select_dtypes(include=['int64','float64'])
sns.heatmap(numeric_df.corr(), annot=True,
cmap="coolwarm", fmt=".2f", ax=axes[3])
axes[3].set_title("Correlation Heatmap")
```



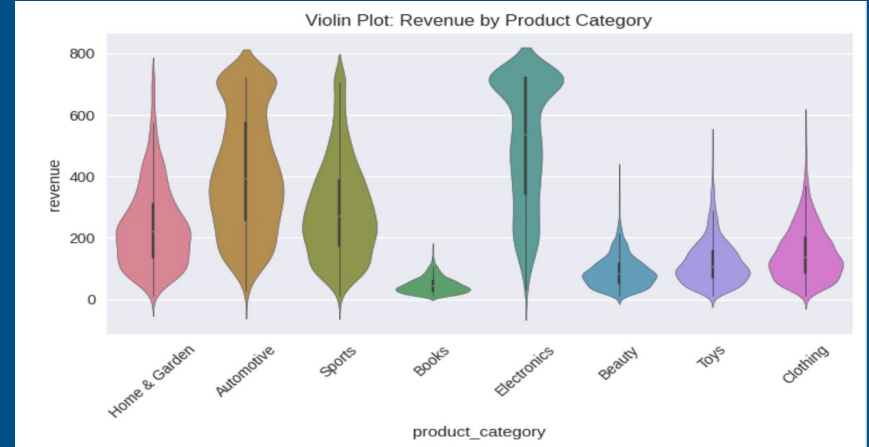
```
category_counts =
df['product_category'].value_counts()
axes[4].bar(category_counts.index,
category_counts.values, color= 'orange',
edgecolor='black')
axes[4].set_title("Bar Plot: Product
Category Distribution")
axes[4].tick_params(axis= 'x', rotation=45)
```



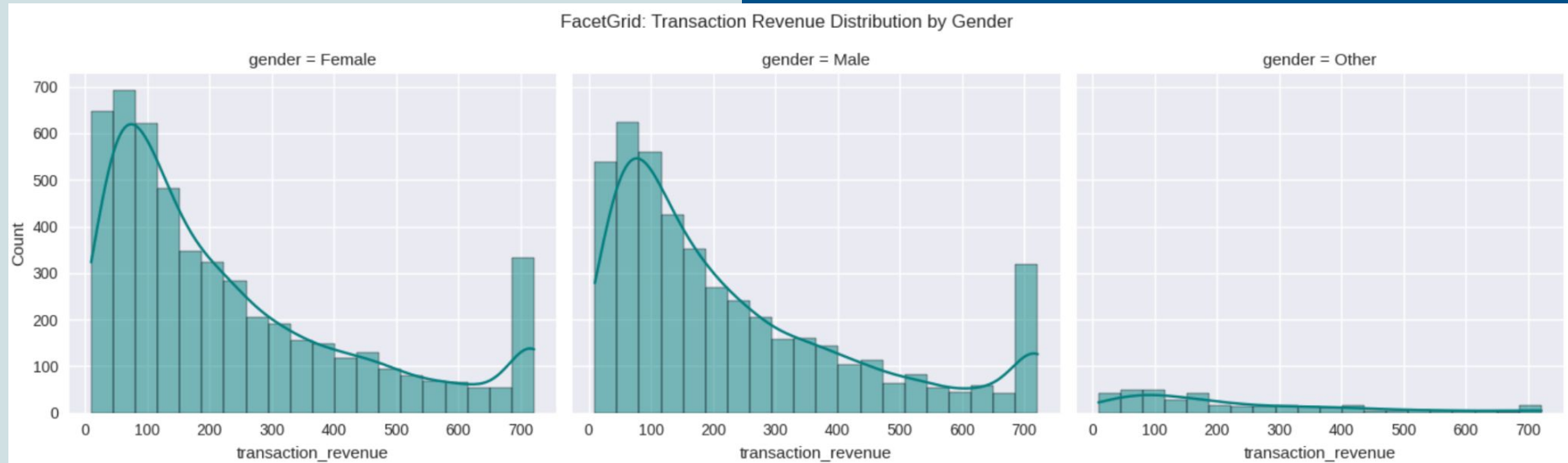
```
gender_counts = df['gender'].value_counts()
axes[5].pie(gender_counts,
labels=gender_counts.index, autopct= '%1.1f%%'
,
colors=[ 'lightblue', 'pink', 'lightgrey' ])
axes[5].set_title("Pie Chart: Gender
Distribution")
```



```
df['purchase_date'] = pd.to_datetime(df['purchase_date'])
yearly_revenue =
df.groupby(df['purchase_date'].dt.year)['revenue'].sum()
axes[6].plot(yearly_revenue.index, yearly_revenue.values,
marker='o', color='purple')
axes[6].set_title("Time Series: Yearly Revenue Trend")
axes[6].set_xlabel("Year")
axes[6].set_ylabel("Revenue")
axes[6].set_yticklabels([f'{int(label):,}' for label in
axes[6].get_yticks()])
```



```
category_counts =
df['product_category'].value_counts()
axes[4].bar(category_counts.index,
category_counts.values, color='orange',
edgecolor='black')
axes[4].set_title("Bar Plot: Product
Category Distribution")
axes[4].tick_params(axis='x', rotation=45)
```



```
g = sns.FacetGrid(df, col="gender", height=4, aspect=1.2)
g.map(sns.histplot, "revenue", bins=20, kde=True, color="teal")
g.fig.suptitle("FacetGrid: Revenue Distribution by Gender", y=1.05)
```

1. Data Loading and Exploration
2. Data Cleaning and Preprocessing
3. Statistical Analysis
4. Data Visualization
5. **Machine Learning Implementation**
6. Business Insights and Recommendations

[illegible]

Median transaction revenue: 165.06

Numeric features: ['age', 'income', 'price', 'quantity', 'customer_lifespan_days', 'registration_year', 'registration_month', 'registration_dayofweek', 'purchase_price']

```
Categorical features: ['gender', 'location', 'product category', 'brand']
```

```
Total features used: 46
```

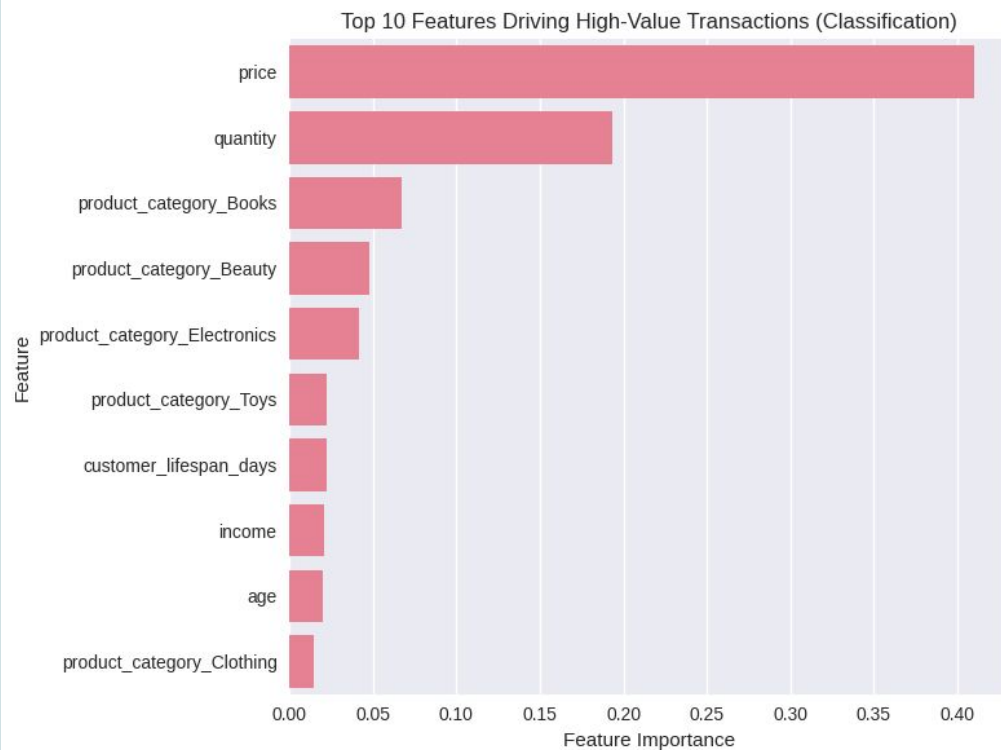
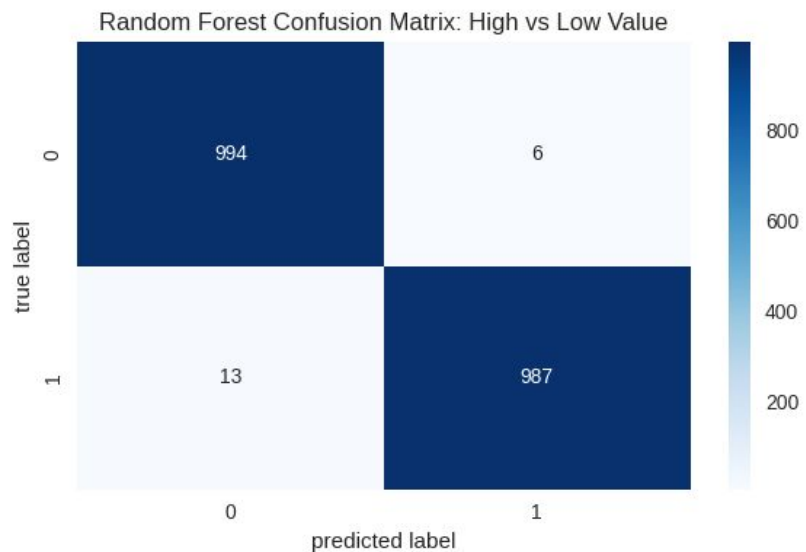
	age	income	price	quantity	customer_lifespan_days	registration_year	registration_month	registration_dayofweek	purchase_year	purchase_month	...
0	40.000000	33889.0	58.75	2	75.0	2025	4	4	2025	7	...
1	30.000000	20000.0	177.15	2	190.0	2025	2	6	2025	8	...
2	34.999457	58321.0	133.91	2	111.0	2025	5	6	2025	9	...
3	39.000000	39590.0	16.46	6	296.0	2022	11	5	2023	9	...
4	42.000000	23078.0	94.95	1	41.0	2025	1	0	2025	2	...

5 rows x 46 columns

```
['purchase_year', 'purchase_month', 'purchase_dayofweek']
```

[illegible]

... Logistic Regression Performance:					
	precision	recall	f1-score	support	
0	0.95	0.95	0.95	1000	
1	0.95	0.95	0.95	1000	
accuracy			0.95	2000	
macro avg	0.95	0.95	0.95	2000	
weighted avg	0.95	0.95	0.95	2000	
Random forest performance:					
	precision	recall	f1-score	support	
0	0.99	0.99	0.99	1000	
1	0.99	0.99	0.99	1000	
accuracy			0.99	2000	
macro avg	0.99	0.99	0.99	2000	
weighted avg	0.99	0.99	0.99	2000	



Regression Model Performance

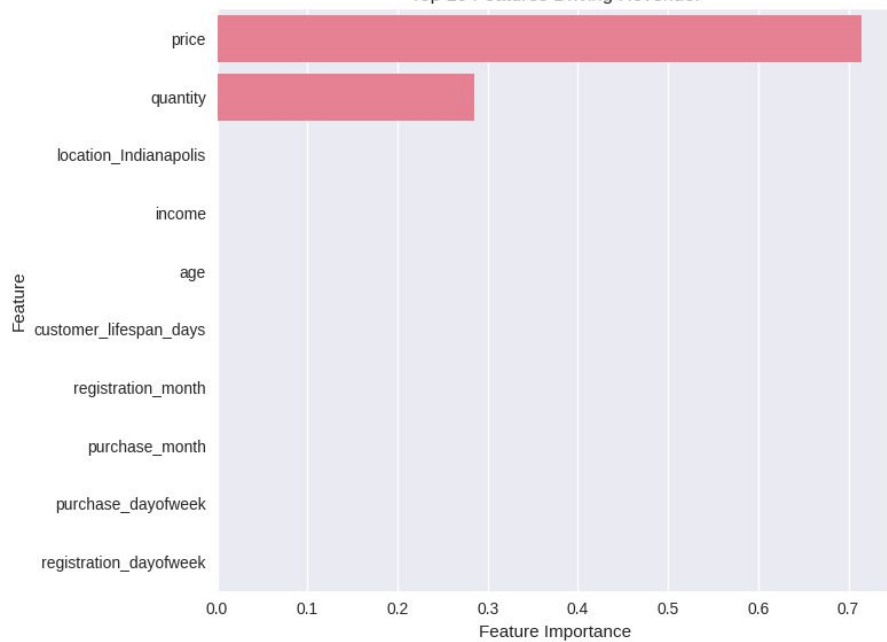
— Random Forest Regressor —

MAE : 0.49

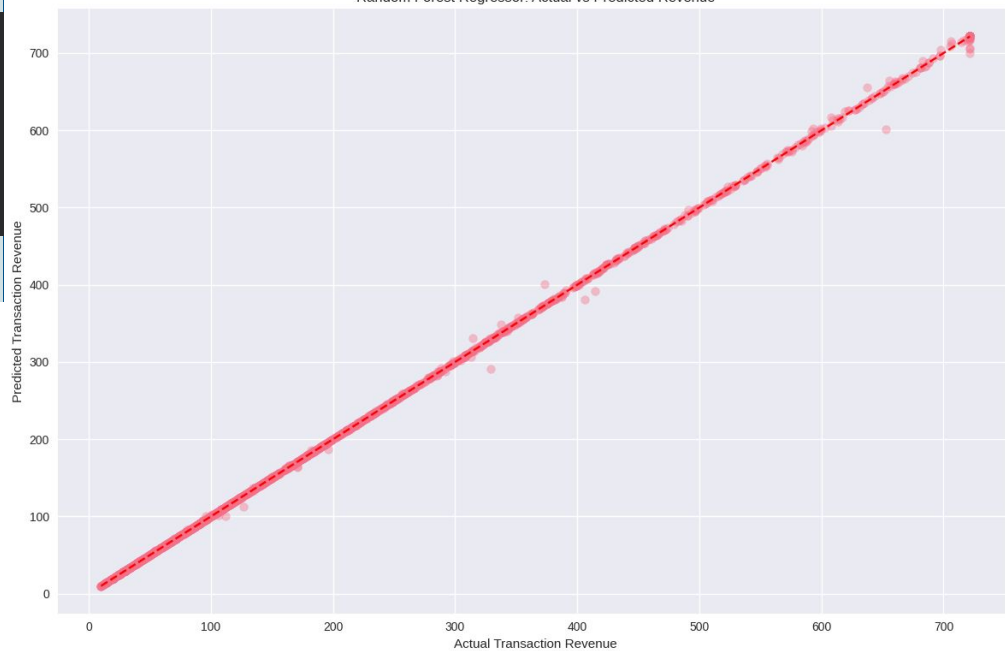
RMSE: 2.19

R^2 : 1.000

Top 10 Features Driving Revenue:



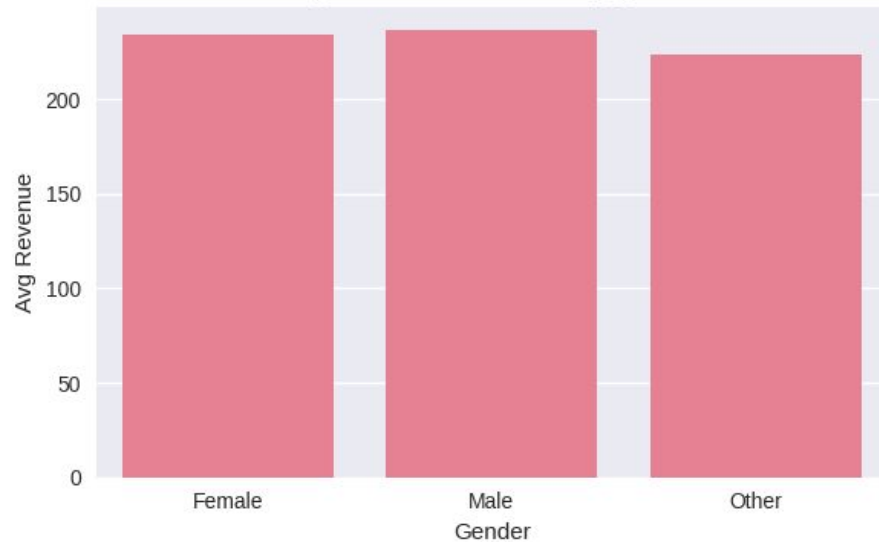
Random Forest Regressor: Actual vs Predicted Revenue



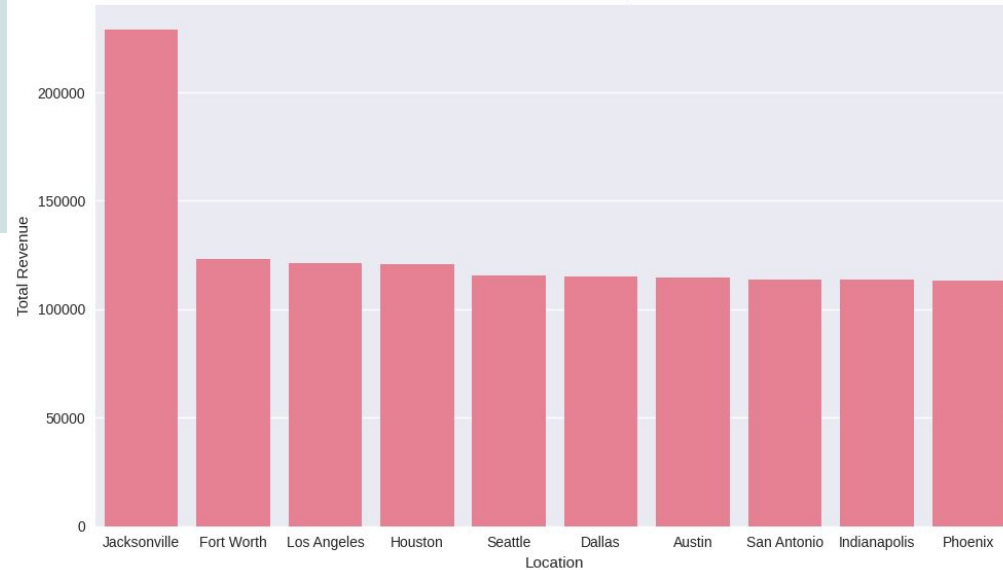
1. Data Loading and Exploration
2. Data Cleaning and Preprocessing
3. Statistical Analysis
4. Data Visualization
5. Machine Learning Implementation
6. **Business Insights and Recommendations**

Revenue by product category			
	count	mean	sum
product_category			
Electronics	1232	508.685718	626700.805
Automotive	1331	412.623186	549201.460
Sports	1299	295.435300	383770.455
Home & Garden	1171	238.727502	279549.905
Clothing	1273	151.674611	193081.780
Toys	1176	121.413112	142781.820
Beauty	1267	91.620560	116083.250
Books	1251	46.526938	58205.200
Revenue by gender:			
	count	mean	sum
gender			
Female	5088	234.202297	1191621.285
Male	4558	236.660244	1078697.390
Other	354	223.322034	79056.000

Avg Transaction Revenue by gender



Total Transaction Revenue by location



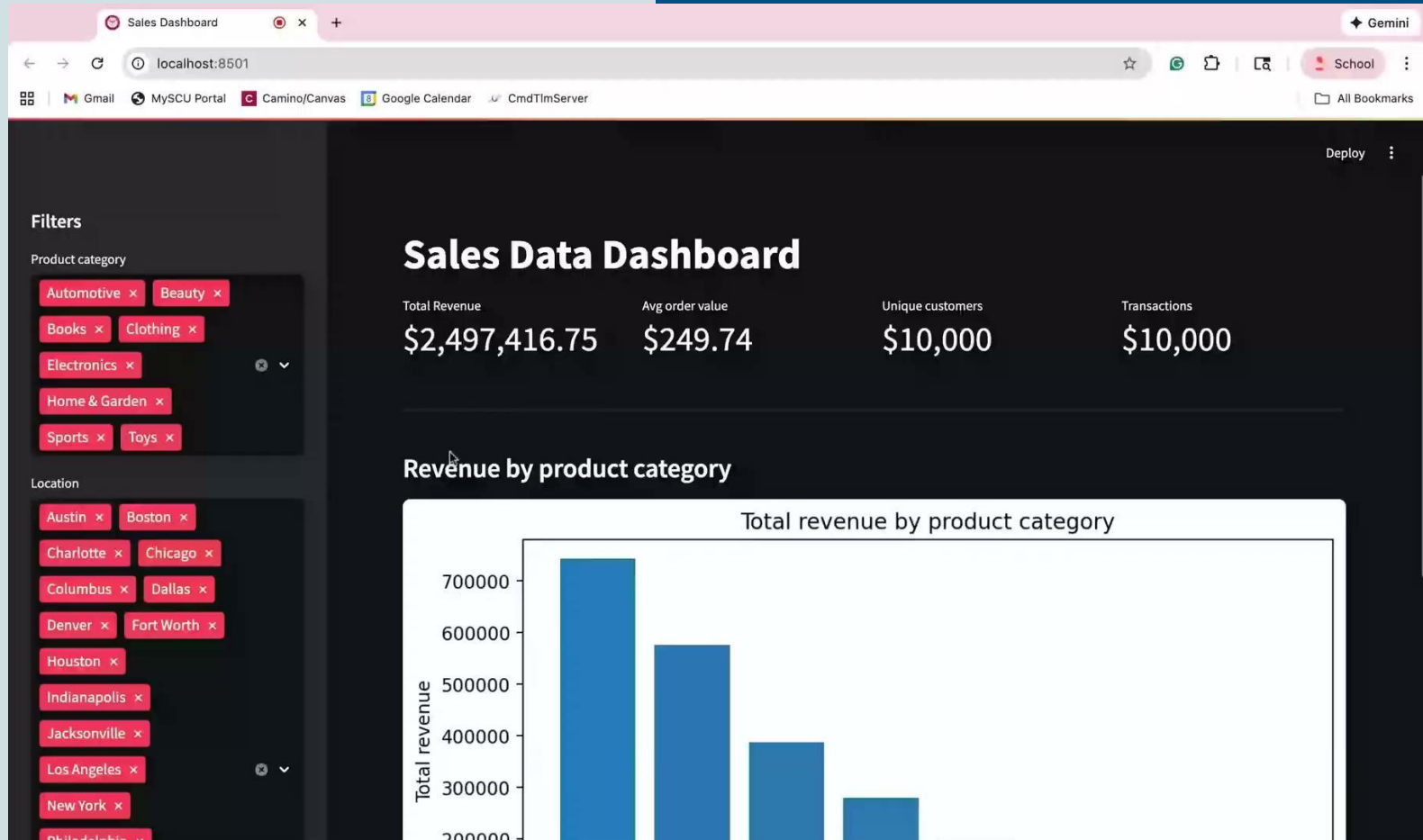
Total revenue: \$2,349,374.67

Avg order value: \$234.94

Total number of unique customers: \$10,000.00

Total number of transactions: \$10,000.00

Repeat purchase rate: \$0.00



Conclusion

Findings

1. **Value is all about Income and Time:** Customer Income and Customer Lifespan Days are the features that matter most.
2. **Sales Range is Highly Volatile (95% Prediction Interval):** The Prediction Interval for any single future transaction is between \$0.00 and \$494.04, with a strong average of \$187.35.
3. **Gender Doesn't Predict Spend:** Average spend between male and female customers is not statistically different.

Recommendations & Impact

1. Launch a '90-Day Lifespan Accelerator' Program

- Focus on locking in new customers early.
- Use automated emails and targeted promotions to entice new users

2. Introduce Income-Based 'Elite Bundles'

- Create tiered, high-price product bundles that package premium items together at a total cost well above our current average transaction price.

3. Align Inventory with High-Value Locations

- Prioritize inventory and fulfillment for the best-selling, high-revenue products specifically in those markets.

IMPACT:

1. **Sales Boost:** 0% increase in the low end of our average transaction revenue range
2. **Retention:** 5% drop in customer churn during the first 90 days
3. **Smarter Spend:** Move budget away from unnecessary gender-specific marketing and put it into location-specific targeting

Capstone Project: Sales Data Analysis and Customer Insights

Overview

This project provides a comprehensive data science analysis of a sales and customer dataset (sales_data.csv). The primary goal is to establish data integrity, perform rigorous statistical analysis, implement an initial machine learning model to predict customer value, and derive actionable, data-driven business recommendations.

The analysis is documented step-by-step in the Jupyter Notebook Capstone_Project (2).ipynb.

Key Findings (Summary from Analysis)

The statistical and predictive modeling revealed critical insights into customer value and transaction predictability:

1. **High-Value Drivers are Tenure and Income:** The machine learning model identified **Customer Income** and **Customer Lifespan Days (tenure)** as the most critical features for predicting high-value customers.
2. **Volatile Sales Range:** The 95% Prediction Interval for a single sale is **\$0.00 to \$494.04** (Mean: \$187.35\$). This high variability confirms that high-value transactions are inconsistent, requiring standardized, high-tier product offerings.
3. **Gender Neutrality in Spending:** A Two-Sample T-Test confirmed **no statistically significant difference** in the mean transaction revenue between male and female customers, allowing marketing focus to shift to income and location.

Strategic Recommendations

The derived recommendations are focused on increasing customer lifespan and formalizing high-value sales:

1. **Launch a '90-Day Lifespan Accelerator' Program** to lock in customer engagement early.
2. **Introduce Income-Based 'Elite Bundles'** to target high-income customers and standardize high-tier revenue.
3. **Align Inventory with High-Value Locations** (New York, Boston, San Francisco) to optimize fulfillment in high-yield areas.

Repository Structure

└─ Capstone_Project (2).ipynb # Primary analysis notebook

← [README.md](#)

Requirements.txt below

```
requirements.txt

This file lists the required Python packages for the Capstone Project analysis.
Pandas: pip install pandas
Numpy: pip install numpy
Scipy: pip install scipy
Sklearn: pip install scikit-learn
Matplotlib: pip install matplotlib
Seaborn: pip install seaborn
```

Thank You!