Internship Project Report

|  |  |
| --- | --- |
| Project Title | Furniture Sales Prediction – E-Commerce Analysis |
| Name | Sandhiya |
| Course | B.Sc. Data Science |
| Internship Domain | Data Analytics |
| Date | June 2025 |

# Table of Contents

1. 1. Abstract
2. 2. Company & Context
3. 3. Objectives & Project Definition
4. 4. Methodology & Design
5. 5. Implementation & Results
6. 6. Challenges & Solutions
7. 7. Learnings & Skill Growth
8. 8. Future Recommendations
9. 9. Conclusion
10. 10. Tech Stack
11. 11. Appendices

# 1. Abstract

This project involved analyzing an e-commerce furniture dataset to predict product sales categories (Low, Medium, High). Using data cleaning, feature engineering, and a classification model, the project aimed to extract business insights and enhance sales forecasting accuracy. The final model achieved an accuracy of 64%, highlighting the value of machine learning in real-world sales prediction.

# 2. Company & Context

This internship was conducted with Unified Mentor as part of a short one-month academic training program focused on practical applications of data analytics. The dataset used was inspired by AliExpress e-commerce listings to simulate real-world business data environments. The main objective was to explore patterns in sales behavior and apply data science techniques in a business-relevant context.

# 3. Objectives & Project Definition

- Analyze product attributes like price, discount, and shipping

- Classify products into Low, Medium, High sales categories

- Use regression and classification techniques

- Gain insights into key drivers of online furniture sales

# 4. Methodology & Design

- Data Cleaning: Converted price formats, handled missing values

- Feature Engineering: Created discount percentage & free shipping flags

- Modeling: Tried regression, then used Random Forest classification

- Evaluation: Used accuracy, precision, confusion matrix for performance analysis

# 5. Implementation & Results

- Classification accuracy achieved: 64%

- Best-performing class: Low sales (Precision: 0.73)

- Weakest-performing class: High sales (Precision: 0.20)

- Final model: Random Forest Classifier

# 6. Challenges & Solutions

- Skewed sales data: Very few high-selling products

➤ Solution: Transformed the target variable into 3 categories

- Poor regression results: R² < 0.01

➤ Solution: Switched to classification modeling approach

# 7. Learnings & Skill Growth

- Learned how to prepare and clean real-world messy datasets

- Gained experience with classification models and evaluation metrics

- Improved skills in Python, Scikit-learn, and EDA using Matplotlib

- Understood how to evaluate and tune model performance effectively

# 8. Future Recommendations

- Add NLP to extract features from product titles

- Balance dataset using SMOTE or under-sampling

- Include product rating, image quality, or brand as new features

- Try advanced models like XGBoost or Neural Networks

# 9. Conclusion

This project successfully demonstrated how classification models can help predict product sales levels in an e-commerce context. The Random Forest model delivered reasonable accuracy, and several opportunities for model improvement were identified. The project helped bridge academic learning with practical, hands-on experience.

# 10. Tech Stack

- Programming Language: Python

- Libraries: Pandas, Scikit-learn, Matplotlib, Seaborn

- Tools: Jupyter Notebook / Google Colab

# 11. Appendices

### *Key Code Snippets*

* Data Cleaning – Converting Price & Handling Missing Values

data['original\_price'] = data['original\_price'].str.replace('$', '').astype(float)

data.dropna(inplace=True)

data['original\_price'] = data['original\_price'].str.replace('$', '').astype(float)

data.dropna(inplace=True)

* Feature Engineering – Creating Discount Percentage

data['discount\_percent'] = ((data['original\_price'] - data['sale\_price']) / data['original\_price']) \* 100

data['free\_shipping'] = data['shipping\_fee'] == 0

data['discount\_percent'] = ((data['original\_price'] - data['sale\_price']) / data['original\_price']) \* 100

data['free\_shipping'] = data['shipping\_fee'] == 0

* Target Variable Transformation – Categorizing Sales

def categorize\_sales(units):

if units < 50:

return 'Low'

elif units < 150:

return 'Medium'

else:

return 'High'

data['sales\_category'] = data['units\_sold'].apply(categorize\_sales)

def categorize\_sales(units):

if units < 50:

return 'Low'

elif units < 150:

return 'Medium'

else:

return 'High'

data['sales\_category'] = data['units\_sold'].apply(categorize\_sales)

* Model Training – Random Forest Classifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = RandomForestClassifier()

model.fit(X\_train, y\_train)

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = RandomForestClassifier()

model.fit(X\_train, y\_train)

* Model Evaluation – Accuracy and Confusion Matrix

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

y\_pred = model.predict(X\_test)

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

y\_pred = model.predict(X\_test)

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))

### *Output Summaries*

- Output of data.head()

- Summary from data.info()

- Distribution of target labels

### *Model Evaluation Metrics*

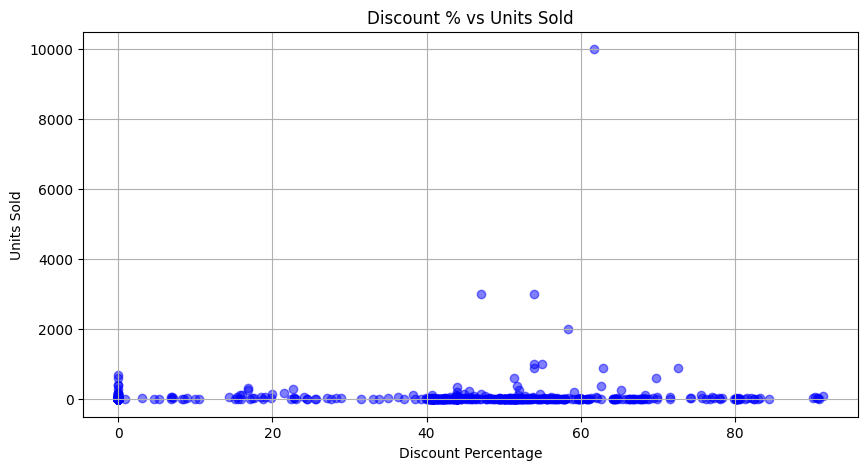
- Accuracy: 64%

- Confusion Matrix (text and visual)

- Classification Report (Precision, Recall, F1-Score)

### *EDA Screenshots*

- **Figure 1**: Scatter plot showing correlation between discount and sales



-**Figure 2**: Scatter plot showing how price affects number of units sold.”

A graph with green lines and numbers

AI-generated content may be incorrect.