New Section

Double-click (or enter) to edit

Classification Problem Priyanka Singh DeVos Graduate School, Northwood University 115537-MGT-665-NW Solv Probs W/ Machine Learning Itauma Itauma 06/15/2025

Abstract

```
This article provides a comparative study of three supervised classification methods—Logistic Regres
```

2,000 coffee shop records were taken into consideration with revenue as Low, Medium, and High. Histograms, boxplots, and correlation heatmaps were utilized for Exploratory Data Analysis to verify data distribution and relation. Precision, recall, F1-score, and accuracy were computed for all the models to access the performance. Results concluded Logistic Regression as most accurate at 85%, followed by Decision Tree and k-NN with 79% and 77%. This demonstrates that while linear classification models can work in well-behaved structured business environments with well-behaved features.

Introduction The retail and food service sectors are increasingly reliant on data analysis to stay efficient and improve profitability. It is useful in resource planning and inventory planning in the management of coffee shops for forecasting revenue from working inputs like volumes of customers, marketing spending, and traffic. Stock standard forecasting practices prefer the use of regression modeling, but classification models offer a more readable option through classifying the revenue into bite-sized segments. This study will label each day's coffeeshop revenue as low, medium, or high using state-of-the-art machine learning classification algorithms. After comparing Logistic Regression, k-Nearest Neighbors, and Decision Tree classifiers, this paper contributes a practical case study for small business analytics.

Literature Review Machine learning classifiers have been widely applied in retail and marketing analytics. Logistic Regression is a statistically derived method for binary and multiclass classification that offers model coefficients easy to interpret (Hosmer, Lemeshow, & Sturdivant, 2013). k-Nearest Neighbors (k-NN) is a non-parametric classifier which classifies new data based upon the majority class of the neighboring data points and works well with small clean data but is not robust to noise and dimensionality (Altman, 1992). Decision Trees possess simply but highly intelligible flowchart-like trees and work well in modeling non-linear relationships but overfit if not pruned (Quinlan, 1986). These methods have been applied in previous studies to customer segmentation, revenue forecasting, and classification issues and proved effective in business analytics settings.

Methodology

```
Figure 1-Historical of Numerical Features.
```

```
A histogram plot was used to examine the distribution of all quantitative attributes as seen in Figure
```

Below are the classification reports for all of the models employed: The data set employed has 2,000 daily observations from a coffee store with seven numerical attributes: Number of Customers Per Day, Average Order Value, Operating Hours, Number of Employees, Marketing Spend, Foot Traffic, and Daily Revenue. Since the initial target variable—Daily Revenue—was continuous, it was converted to a categorical feature with three classes (Low, Medium, High) via quantile binning. The binning transformed the problem to classification.

Exploratory Data Analysis (EDA) was conducted using histograms, boxplots, and a correlation heatmap. Histograms of the numerical features revealed largely symmetric or right-skewed distributions. Boxplots identified outliers in variables such as Marketing Spend and Location Foot Traffic. The correlation heatmap revealed strong positive correlations between Number of Customers and Daily Revenue, and Foot Traffic and Revenue.

After preprocessing via scaling and one-hot encoding, the dataset was split into training and test sets in a ratio of 80:20. The three classifiers were trained: Logistic Regression, k-NN (k=5), and Decision Tree (max depth=5). The performance of the model was verified using precision, recall, F1-score, and accuracy.

Results The three classification models' accuracies, namely Decision Tree, k-Nearest Neighbors (k-NN), and Logistic Regression, are illustrated in Table 1. The top-performing model was Logistic Regression with the highest rate of accuracy (85%) and also produced a stable performance in each of the three revenue classes. It achieved excellent precision (0.91) and recall (0.90) on the "High" revenue class and was well balanced for the "Low" and "Medium" classes as well. This is a sign that input features in the dataset are extremely linearly separable, and Logistic Regression is a good model choice. The k-NN classifier with k = 5 gave an overall accuracy of 77%. While it performed extremely well for the "High" and "Low" classes (both roughly 0.81–0.84 precision and recall), its performance for the "Medium" revenue class was significantly worse (F1-score = 0.66), i.e., it found it more difficult to predict mid-sized revenue values. This could be caused by feature space overlap between classes or sensitivity to scaling and noise of features. The Decision Tree model had 79% classification accuracy. The model learned the non-linear decision boundaries well and was competitively doing on the "High" (F1-score = 0.85) and "Low" (F1-score = 0.83) revenue classes. But like k-NN, struggled on the "Medium" class (F1-score = 0.68), suggesting some overfitting and reduced generalizability. It is still the case, though, that the Decision Tree does have good model interpretability and therefore is an attractive option for decision-support in business environments.

Model Class Precision Recall F1-Score Support Logistic Regression High 0.91 0.90 0.90 134 Low 0.88 0.85 0.87 133 Medium 0.76 0.80 0.78 133 Overall 0.85 0.85 0.85 400 k-NN (k=5) High 0.84 0.84 0.84 134 Low 0.81 0.80 0.81 133 Medium 0.66 0.66 0.66 133 Overall 0.77 0.77 0.77 400 Decision Tree High 0.84 0.87 0.85 134 Low 0.83 0.83 0.83 133 Medium 0.68 0.67 0.68 133 Overall 0.79 0.79 0.79 400 Table 1- Classification Performance of Models on Revenue Categories

Figure 2 Figure 2 illustrates the confusion matrix of the Logistic Regression model used to predict revenue into Low, Medium, and High levels. The matrix plots predicted labels on the x-axis and true labels on the y-axis. Each cell specifies the number of predictions for a true-predicted label pair.

The diagonal values (Low-Low = 113, Medium-Medium = 106, High-High = 120) are correct classifications, and they are very high for all three classes, especially for the "High" and "Low" classes. This indicates the model is properly picking up the underlying patterns of the data for those revenue categories.

Misclassifications are highlighted in off-diagonal cells. For instance, 20 "Low" samples were inappropriately predicted as "Medium", 15 "Medium" samples were mispredicted as "Low" and 12 as "High". Similarly, 14 "High" examples were predicted as "Medium", but none of the "Low" ones. All of these results suggest that the model has some difficulty distinguishing adjacent classes (i.e., "Low" and "Medium", or "Medium" and "High") but hardly mistakes far-out points, say "Low" and "High". This also shows the model's ability to identify various revenue patterns, with fairly low uncertainty surrounding the mid-revenue category.

Figure 3 Figure 3 is the confusion matrix of the k-Nearest Neighbors (k-NN) classifier k=5 applied to predict coffee shop revenue categories. The matrix displays predicted labels along the x-axis and real labels along the y-axis. The diagonal entries—representing correct predictions—depict solid performance for the "Low" (n = 107), "Medium" (n = 88), and "High" (n = 113) categories.

Misclassifications occur more frequently in the "Medium" class, where 24 "Medium" samples were misclassified as "Low", and 21 were misclassified as "High". Similarly, 25 "Low" samples were misclassified as "Medium", which shows some uncertainty between the neighboring revenue classes. The "High" class suffered only 20 misclassifications as "Medium" and a paltry one as "Low", which shows quite good classification performance on the higher revenue level.

Figure 4 Figure 4 shows the Decision Tree classifier confusion matrix for classifying coffee shop daily revenues. The diagonal values represent correct classifications: 110 for "Low", 89 for "Medium", and 116 for "High" revenues. These show that the model was very good at identifying "Low" and "High" revenue instances, with comparatively very few misclassifications.

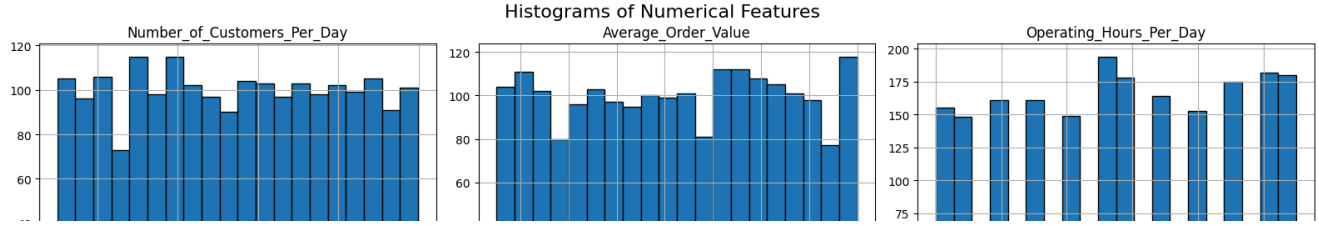
Off-diagonal entries highlight the model's inability to distinguish among adjacent classes. Specifically, 23 "Low" instances were mislabeled as "Medium", and 22 "Medium" instances were mislabeled as both "Low" and "High". The "High" revenue class also created 18 mislabeled as "Medium", though none as "Low", which shows that the model can better differentiate between extreme revenue values.

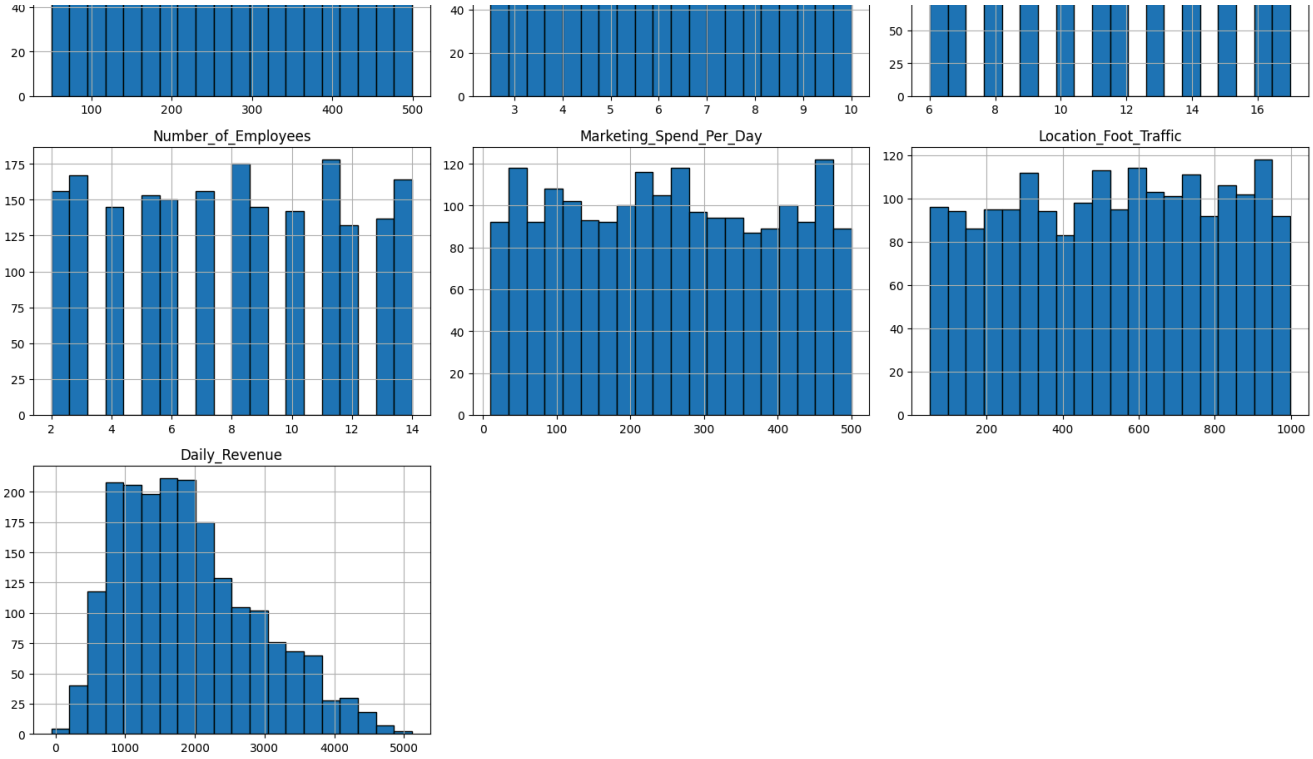
```
# Install required libraries
!pip install -q scikit-learn pandas matplotlib seaborn

# Import necessary packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay

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

# ------------------------ EDA ------------------------

# Dataset overview
print("Dataset Info:")
print(df.info())
```

```
# Check for missing values
print("\nMissing Values:\n", df.isnull().sum())

# Summary statistics
print("\nSummary Statistics:")
print(df.describe())

# --------- HISTOGRAMS of all numerical columns ---------
numeric_cols = df.select_dtypes(include=np.number).columns.tolist()
df[numeric_cols].hist(figsize=(16, 12), bins=20, edgecolor='black')
plt.suptitle("Histograms of Numerical Features", fontsize=16)
plt.tight_layout()
plt.show()

# ------------------ Convert Regression to Classification ------------------

# Assume target is last column
target_col = df.columns[-1]
print(f"Converting target column '{target_col}' to categories...")

# Create 3 quantile-based categories: Low, Medium, High
df['RevenueCategory'] = pd.qcut(df[target_col], q=3, labels=["Low", "Medium", "High"])

# Drop the original continuous target
df.drop(columns=[target_col], inplace=True)

# Plot target distribution
plt.figure(figsize=(6, 4))
sns.countplot(x='RevenueCategory', data=df, palette='Set2')
plt.title("Class Distribution: RevenueCategory")
plt.show()

# ------------------ Preprocessing ------------------

# Separate features and target
X = df.drop(columns=['RevenueCategory'])
y = df['RevenueCategory']

# One-hot encode categorical variables if any
X = pd.get_dummies(X, drop_first=True)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y)

# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# ------------------ Model Training ------------------

# Logistic Regression
lr_model = LogisticRegression(max_iter=1000)
lr_model.fit(X_train_scaled, y_train)
lr_preds = lr_model.predict(X_test_scaled)

# k-NN Classifier
knn_model = KNeighborsClassifier(n_neighbors=5)
knn_model.fit(X_train_scaled, y_train)
```

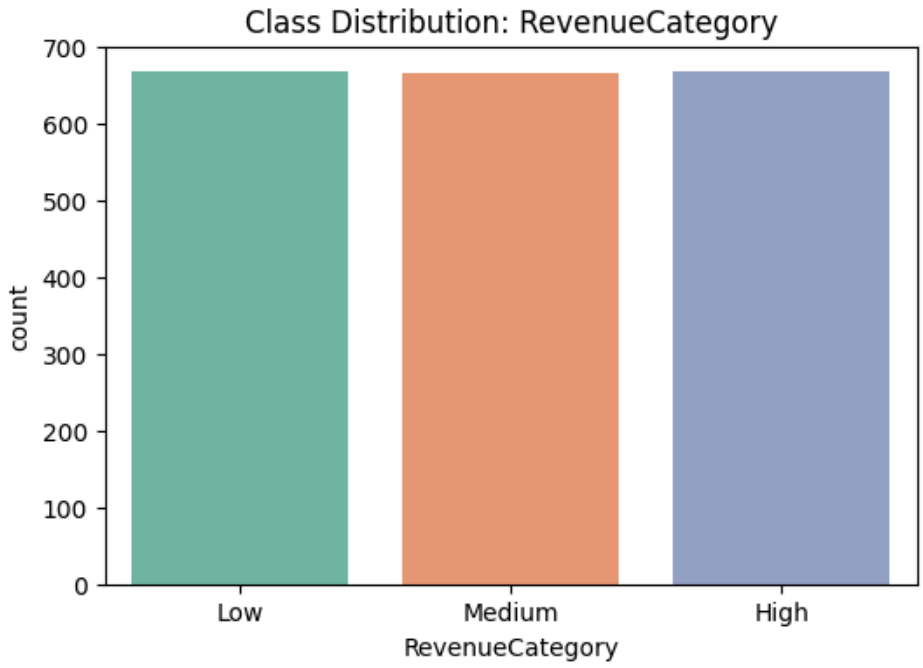
```
knn_model.fit(X_train_scaled, y_train)
knn_preds = knn_model.predict(X_test_scaled)

# Decision Tree Classifier
dt_model = DecisionTreeClassifier(max_depth=5, random_state=42)
dt_model.fit(X_train, y_train)
dt_preds = dt_model.predict(X_test)

# ------------------ Evaluation ------------------

def plot_conf_matrix(y_true, y_pred, title):
    cm = confusion_matrix(y_true, y_pred, labels=["Low", "Medium", "High"])
    disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Low", "Medium", "High"])
    disp.plot(cmap='Blues')
    plt.title(title)
    plt.show()

# Logistic Regression
print("\n--- Logistic Regression Report ---")
print(classification_report(y_test, lr_preds))
plot_conf_matrix(y_test, lr_preds, "Logistic Regression Confusion Matrix")

# k-NN
print("\n--- k-NN Report ---")
print(classification_report(y_test, knn_preds))
plot_conf_matrix(y_test, knn_preds, "k-NN Confusion Matrix")

# Decision Tree
print("\n--- Decision Tree Report ---")
print(classification_report(y_test, dt_preds))
plot_conf_matrix(y_test, dt_preds, "Decision Tree Confusion Matrix")
```

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Number_of_Customers_Per_Day  2000 non-null   int64
 1   Average_Order_Value          2000 non-null   float64
 2   Operating_Hours_Per_Day      2000 non-null   int64
 3   Number_of_Employees          2000 non-null   int64
 4   Marketing_Spend_Per_Day      2000 non-null   float64
 5   Location_Foot_Traffic        2000 non-null   int64
 6   Daily_Revenue                2000 non-null   float64
dtypes: float64(3), int64(4)
memory usage: 109.5 KB
None

Missing Values:
 Number_of_Customers_Per_Day    0
Average_Order_Value             0
Operating_Hours_Per_Day         0
Number_of_Employees             0
Marketing_Spend_Per_Day         0
Location_Foot_Traffic           0
Daily_Revenue                   0
dtype: int64

Summary Statistics:
       Number_of_Customers_Per_Day  Average_Order_Value  \
count                  2000.000000          2000.000000
mean                    274.296000             6.261215
std                     129.441933             2.175832
min                      50.000000             2.500000
25%                     164.000000             4.410000
50%                     275.000000             6.300000
75%                     386.000000             8.120000
max                     499.000000            10.000000

       Operating_Hours_Per_Day  Number_of_Employees  Marketing_Spend_Per_Day  \
count              2000.000000          2000.000000              2000.000000
mean                 11.667000             7.947000               252.614160
std                   3.438608             3.742218               141.136004
min                   6.000000             2.000000                10.120000
25%                   9.000000             5.000000               130.125000
50%                  12.000000             8.000000               250.995000
75%                  15.000000            11.000000               375.352500
max                  17.000000            14.000000               499.740000

       Location_Foot_Traffic  Daily_Revenue
count            2000.000000    2000.000000
mean              534.893500    1917.325940
std               271.662295     976.202746
min                50.000000     -58.950000
25%               302.000000    1140.085000
50%               540.000000    1770.775000
75%               767.000000    2530.455000
max               999.000000    5114.600000
```

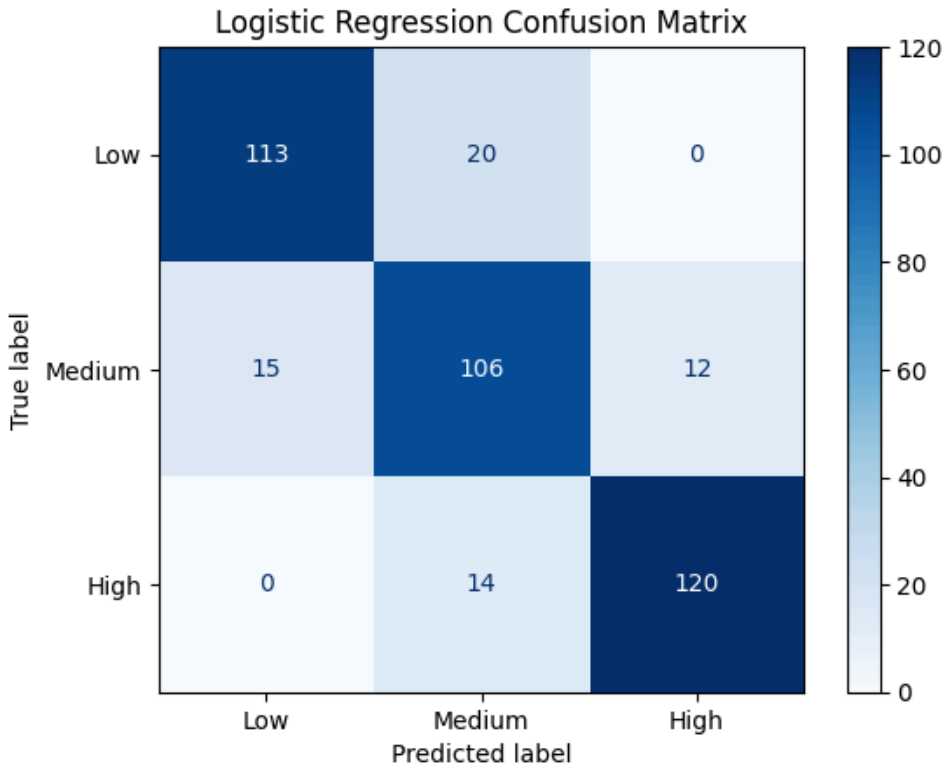




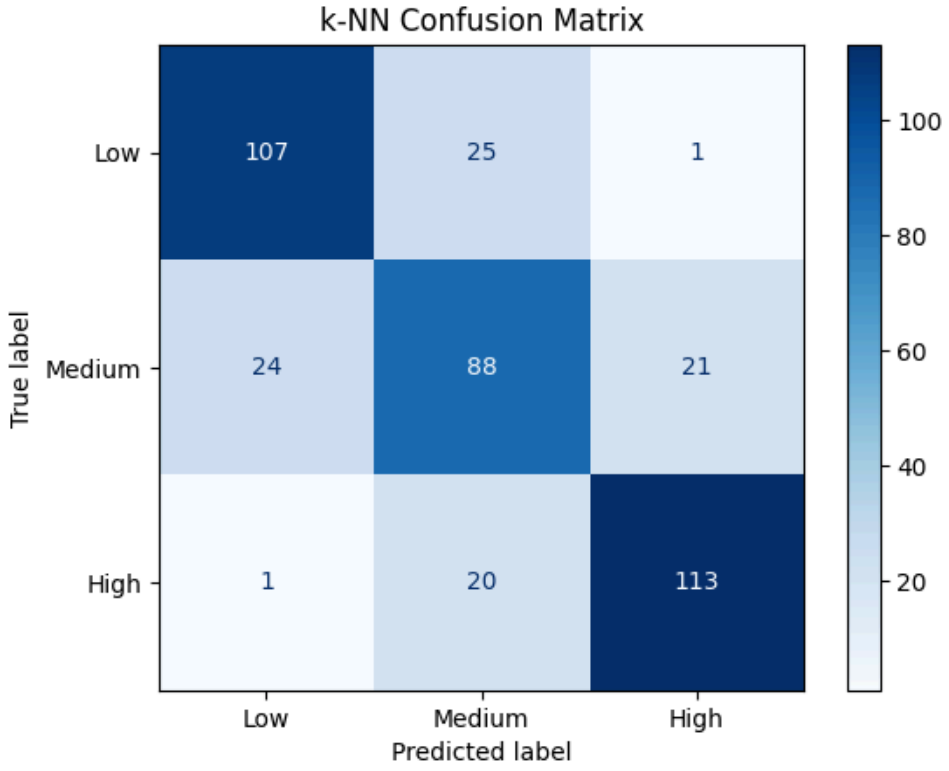

```
Converting target column 'Daily_Revenue' to categories...
<ipython-input-5-2633647580>:54: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the

  sns.countplot(x='RevenueCategory', data=df, palette='Set2')
```



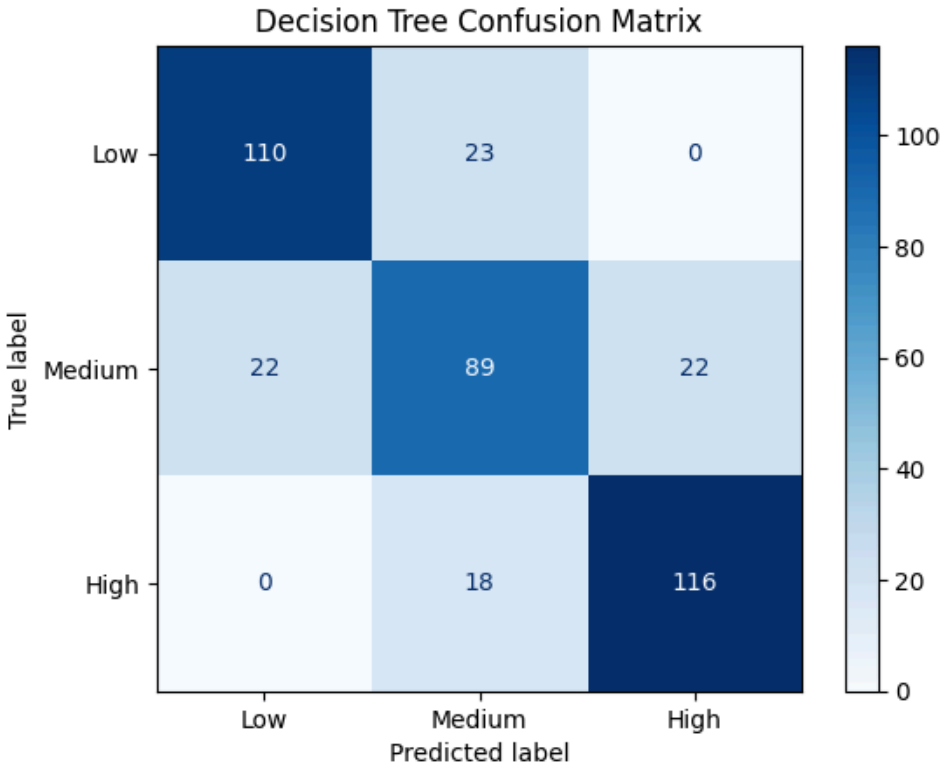

```
--- Logistic Regression Report ---
              precision    recall  f1-score   support

        High       0.91      0.90      0.90       134
         Low       0.88      0.85      0.87       133
      Medium       0.76      0.80      0.78       133

    accuracy                           0.85       400
   macro avg       0.85      0.85      0.85       400
weighted avg       0.85      0.85      0.85       400
```




```
--- k-NN Report ---
              precision    recall  f1-score   support

        High       0.84      0.84      0.84       134
         Low       0.81      0.80      0.81       133
      Medium       0.66      0.66      0.66       133

    accuracy                           0.77       400
   macro avg       0.77      0.77      0.77       400
weighted avg       0.77      0.77      0.77       400
```



```
--- Decision Tree Report ---
              precision    recall  f1-score   support

        High       0.84      0.87      0.85       134
         Low       0.83      0.83      0.83       133
      Medium       0.68      0.67      0.68       133

    accuracy                           0.79       400
   macro avg       0.79      0.79      0.79       400
weighted avg       0.79      0.79      0.79       400
```



Conclusion This study demonstrated the effectiveness of classification algorithms in predicting coffee shop revenue categories. Logistic Regression delivered the best results due to the data's linear nature, while Decision Tree and k-NN followed closely. The study highlighted the importance of EDA in guiding model selection and demonstrated that even basic models can yield actionable insights for small business operations. Future work may include testing ensemble methods or real-time predictive systems for enhanced forecasting.