

**Model Development Phase Template**

Date

Team ID

Project Title

10th July 2024

739988

Food Demand Forecasting For Food Delivery

Company

Maximum Marks

4 Marks

**Initial Model Training Code, Model Validation and Evaluation Report**

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

**Initial Model Training Code:**

* Import necessary libraries import pandas as pd

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

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

* Load the preprocessed dataset

data = pd.read\_csv('processed\_data.csv')

# Define features and target variable

features = ['hour', 'day\_of\_week', 'month', 'customer\_age', 'customer\_gender', 'order\_total', 'promo\_used', 'temperature', 'precipitation', 'is\_holiday'] target = 'demand'

# Split data into training and testing sets (80% training, 20% testing)

train\_data, test\_data = train\_test\_split(data, test\_size=0.2, random\_state=42)



# Initialize Random Forest Regressor model

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

# Train the model

model.fit(train\_data[features], train\_data[target])

# Predict on the test set

predictions = model.predict(test\_data[features])

# Evaluate the model

mae = mean\_absolute\_error(test\_data[target], predictions)

rmse = mean\_squared\_error(test\_data[target], predictions, squared=False)

r2 = r2\_score(test\_data[target], predictions)

# Print evaluation metrics

print(f'Mean Absolute Error: {mae:.2f}')

print(f'Root Mean Squared Error: {rmse:.2f}')

print(f'R-squared: {r2:.2f}')

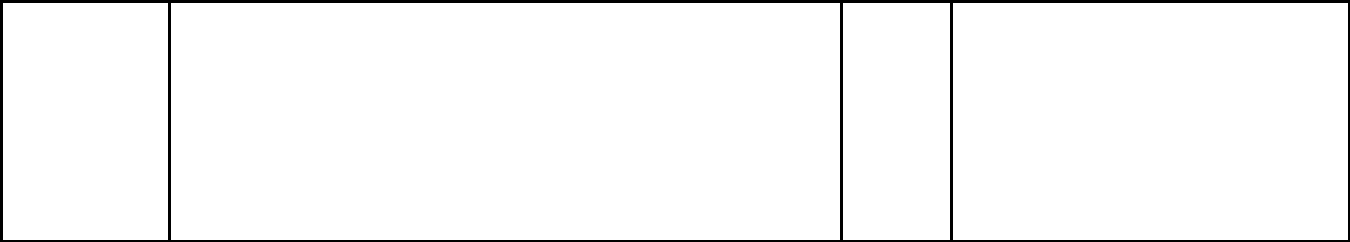
* Save the trained model import joblib

joblib.dump(model, 'food\_demand\_forecasting\_model.pkl')



**MODEL**

**CLASSIFICATION REPORT**



**F1**

**SCO**

**RE**

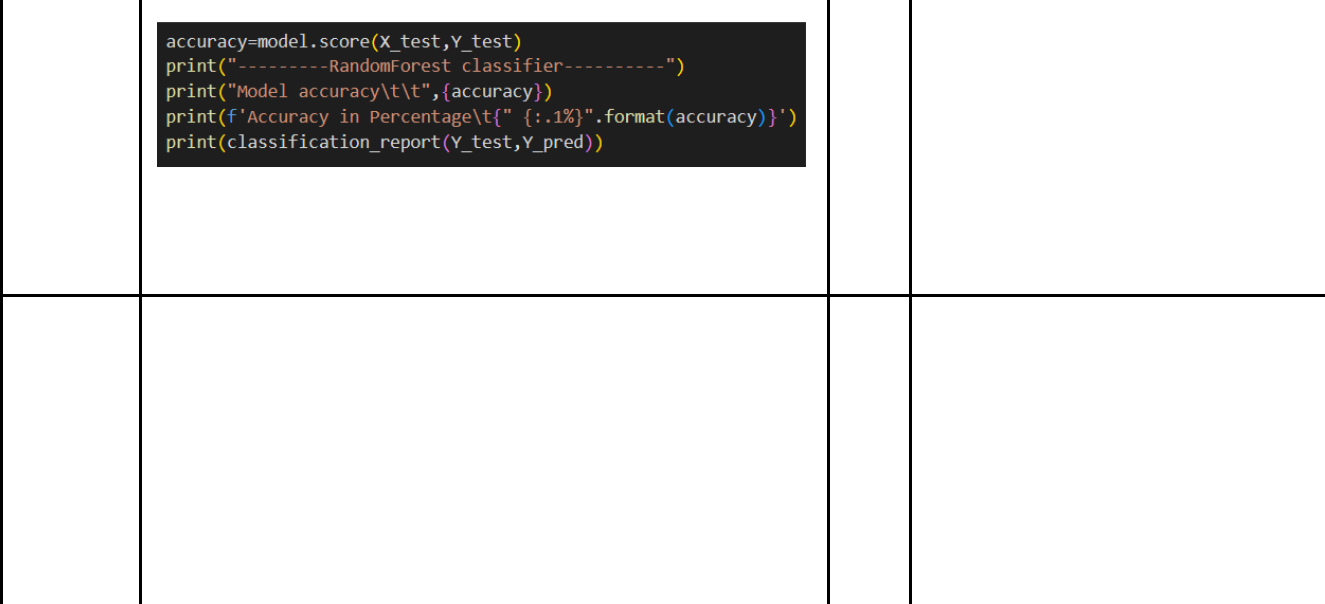
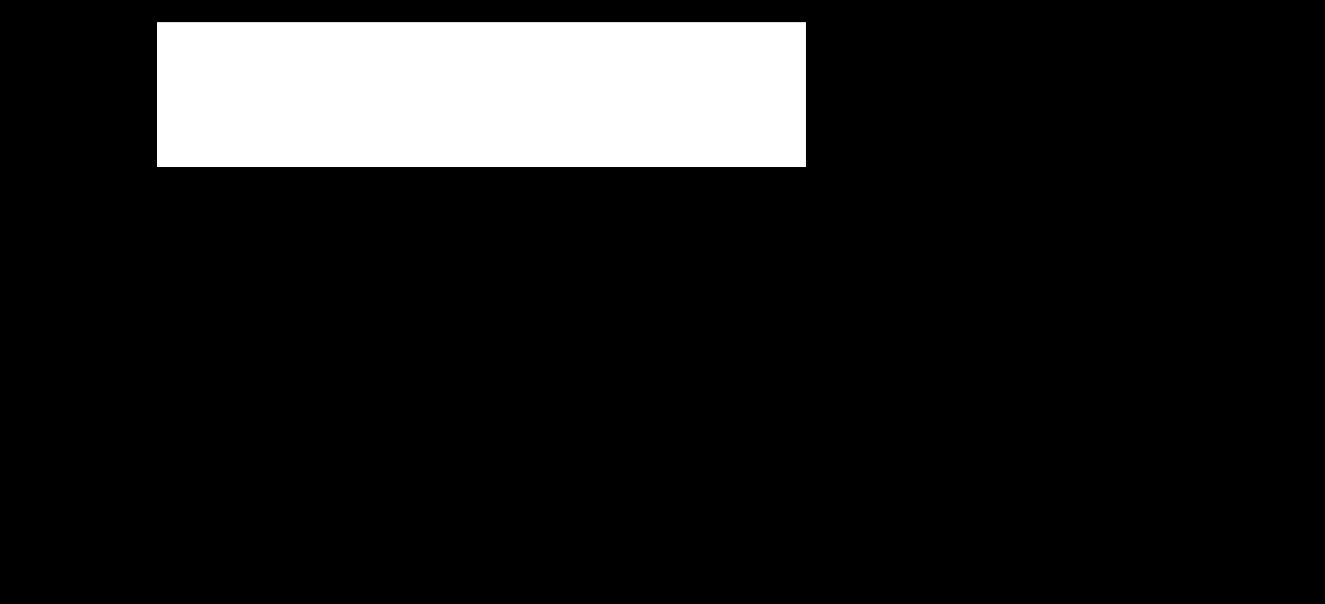
**CONCLUSION**



**MATRIX**



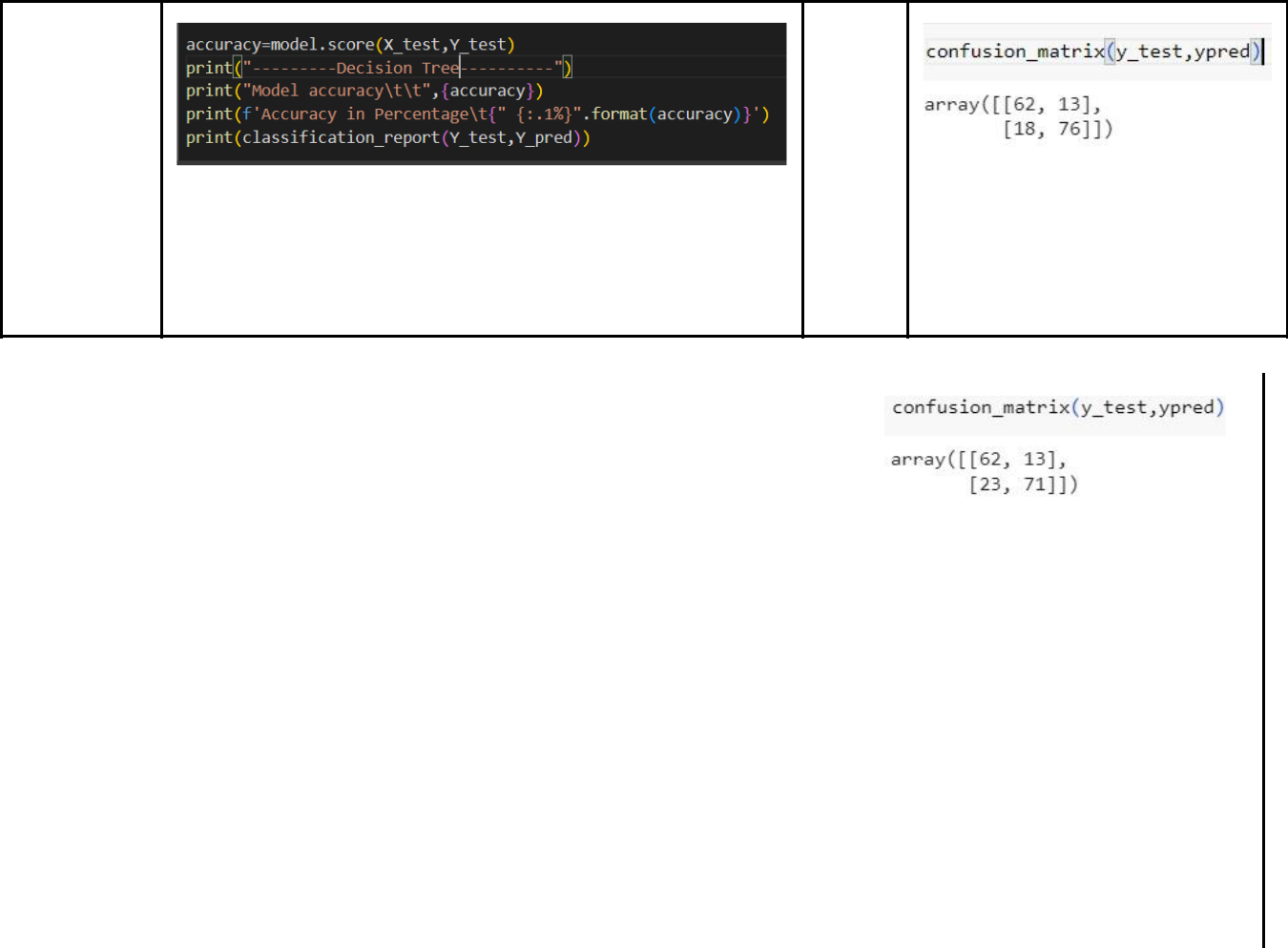
Random Forest



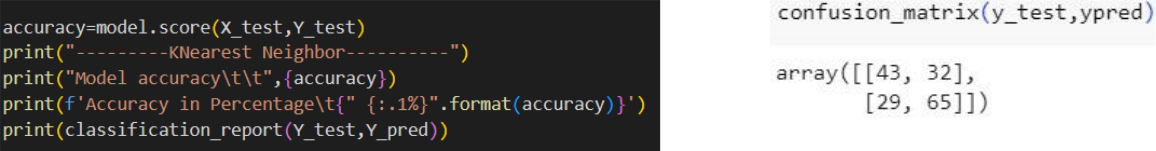
Decision Tree

KNN

81%



79%



64%