

Model Development Phase Template

Date	10 th July 2024
Team ID	739754
Project Title	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	<pre>accuracy=model.score(X_test,Y_test) print("-----Decision Tree-----") print("Model accuracy\t\t",{accuracy}) print(f'Accuracy in Percentage\t{" {:.1%}".format(accuracy)}') print(classification_report(Y_test,Y_pred))</pre>	81%	<pre>confusion_matrix(y_test,ypred) array([[62, 13], [18, 76]])</pre>
Decision Tree		79%	<pre>confusion_matrix(y_test,ypred) array([[62, 13], [23, 71]])</pre>
KNN	<pre>accuracy=model.score(X_test,Y_test) print("-----KNearest Neighbor-----") print("Model accuracy\t\t",{accuracy}) print(f'Accuracy in Percentage\t{" {:.1%}".format(accuracy)}') print(classification_report(Y_test,Y_pred))</pre>	64%	<pre>confusion_matrix(y_test,ypred) array([[43, 32], [29, 65]])</pre>