

Project Title: PREDICTING CUSTOMER
SATISFACTION USING XGBOOST AND FLASK

NAME: Dare Onaolapo Oladipo

Project Objective:

The primary goal of this project is to build a machine learning model that predicts **customer satisfaction levels** (e.g., Satisfied, Neutral, Unsatisfied) based on various behavioral and demographic attributes, and deploy this model as a simple web application using **Flask**.

Dataset Overview:

The dataset contains historical customer behavior, including:

- ✓ Total Spend
- ✓ Items Purchased
- ✓ Average Rating
- ✓ Days Since Last Purchase
- ✓ Discount Applied
- ✓ Gender
- ✓ Membership Level
- ✓ Satisfaction Level (Target)

The target variable, **Satisfaction Level**, is a categorical variable with classes such as Satisfied, Neutral, and Unsatisfied.

Preprocessing Steps:

Label Encoding:

- Satisfaction Level was encoded using `LabelEncoder` from `scikit-learn` and saved as `label_encoder.pkl` for future use.
- Gender was manually encoded as binary (Male = 1, Female = 0).
- Membership was mapped manually as (Silver = 0, Gold = 1, Platinum = 2).

Scaling:

Features were standardized using `StandardScaler` to ensure that all input features were on the same scale.

The scaler was saved as `scaler.pkl`.

Train-Test Split:

The dataset was split into training and testing sets using an 70/30 ratio.

Model Building:

The model used is an **XGBoost Classifier**, chosen for its high performance in classification problems.

RandomizedSearchCV was applied for hyperparameter tuning with parameters like:

`n_estimators, max_depth, learning_rate, subsample, etc.`

The best model was selected based on cross-validated accuracy and saved as `best_xgb_model.pkl`.

Model Performance:

After training and tuning, the model achieved the following performance on the test set:

Accuracy: ~100%

Classification Report:

Precision:100%

Recall: 100%,

F1-score:100%

Confusion Matrix: Visualized to analyze prediction distribution.

Web Application (Flask):

A web application was created using Flask to allow users to input customer data and receive a satisfaction prediction.

User Inputs (via HTML form):

Total Spend, Average Rating, Item Purchased, Days Since Last Purchase, Discount Applied, Gender, Membership Level

Backend:

The Flask app loads the trained model, scaler, and label encoder.

Inputs are scaled and passed to the model for prediction.

The output is converted back to a readable satisfaction label and displayed.

Key Files:

File	Description
app.py	Flask backend code
scaler.pkl	Saved StandardScaler
label_encoder.pkl	Saved LabelEncoder for satisfaction labels
best_xgb_model.pkl	Trained and tuned XGBoost model
templates/index.html	Input form for customer features
templates/result.html	Displays the predicted satisfaction level

Final Outcome:

The final solution is a fully functioning **ML-powered web app** that enables non-technical users to predict a customer's satisfaction level in real time by entering a few key data points.