

## **“ML Assignment 2 – Final Submission”**

# **# Online Shoppers Intention – Machine Learning Classification Project**

Name: Srividya

Roll Number: 2025AA05119

Course: Artificial Intelligence and Machine Learning

Assignment: Assignment 2

1. GitHub Repository Link  
[srividya89/streamlit](https://github.com/srividya89/streamlit)
  
2. Live Streamlit Application Link  
Deployed App Link:  
[ML Assignment 2 · Streamlit](https://ml-assignment-2-streamlit.srividya89.repl.co/)

### **# PROJECT FOLDER STRUCTURE**

```
ML_Assignment_2/
|
└── dataset/
    └── online_shoppers_intention.csv
|
└── models/
    ├── saved_models.pkl
    └── model_results.csv
|
└── 2025AA05119_ml_assignment
    └── app.py
```

└── requirements.txt

└── README.md

## BITS LAB SCREENSHOT

The screenshot shows a Jupyter Notebook interface in a Firefox browser window. The notebook displays a classification report and a confusion matrix.

**Classification Report:**

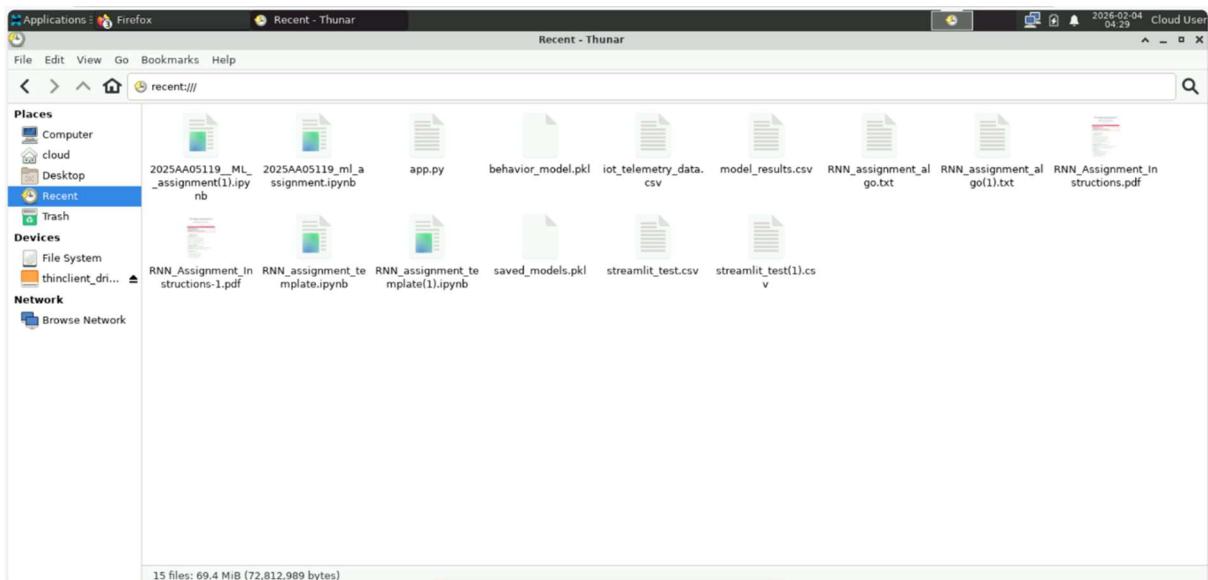
	precision	recall	f1-score	support
0	0.95	1.00	0.98	1589
1	0.64	0.64	0.64	209
2	0.70	0.54	0.61	324
accuracy			0.89	2122
macro avg	0.76	0.73	0.74	2122
weighted avg	0.88	0.89	0.89	2122

**Generating Confusion Matrix...**

Confusion Matrix:

[1587 0 2]
[ 1 133 75]
[ 74 74 176]

Confusion Matrix



```

159     st.pyplot(fig)
160
161     # -----
162     # Accuracy and Classification Report
163     # -----
164
165     acc = accuracy_score(y_actual, preds)
166
167     st.subheader("Model Accuracy")
168     st.write(acc)
169
170     st.subheader("Classification Report")
171
172     report = classification_report(y_actual, preds)
173
174     st.text(report)
175
176 else:
177     st.warning(
178         "To generate confusion matrix, your uploaded CSV must contain an 'Actual' column."
179     )
180
181 else:
182     st.info("Please upload a CSV file to test predictions.")
183
184

```

## Machine Learning Classification Dashboard

### Overall Model Performance Comparison

	Accuracy	AUC	Precision	Recall	F1	MCC
Logistic Regression	0.8827	0.9357	0.8708	0.8827	0.8755	0.6976
Decision Tree	0.886	0.9447	0.8796	0.886	0.8781	0.7109
KNN	0.8869	0.9212	0.8767	0.8869	0.8805	0.7096
Naive Bayes	0.877	0.9275	0.888	0.877	0.8579	0.6994
Random Forest	0.8921	0.9493	0.8829	0.8921	0.8855	0.7235
XGBoost	0.8935	0.9502	0.8845	0.8935	0.8872	0.7272

### Evaluation Metrics: Naive Bayes

	Accuracy	AUC	Precision	Recall	F1	MCC
Naive Bayes	0.877	0.9275	0.888	0.877	0.8579	0.6994

### Overall Model Performance Comparison

	Accuracy	AUC	Precision	Recall	F1	MCC
Logistic Regression	0.8827	0.9357	0.8708	0.8827	0.8755	0.6976
Decision Tree	0.886	0.9447	0.8796	0.886	0.8781	0.7109
KNN	0.8869	0.9212	0.8767	0.8869	0.8805	0.7096
Naive Bayes	0.877	0.9275	0.888	0.877	0.8579	0.6994
Random Forest	0.8921	0.9493	0.8829	0.8921	0.8855	0.7235
XGBoost	0.8935	0.9502	0.8845	0.8935	0.8872	0.7272

### Evaluation Metrics: XGBoost

	Accuracy	AUC	Precision	Recall	F1	MCC
XGBoost	0.8935	0.9502	0.8845	0.8935	0.8872	0.7272

Uploaded Dataset Preview

Machine Learning Classification Dashboard

### Overall Model Performance Comparison

	Accuracy	AUC	Precision	Recall	F1	MCC
Logistic Regression	0.8827	0.9357	0.8708	0.8827	0.8755	0.6976
Decision Tree	0.886	0.9447	0.8796	0.886	0.8781	0.7109
KNN	0.8869	0.9212	0.8767	0.8869	0.8805	0.7096
Naive Bayes	0.877	0.9275	0.888	0.877	0.8579	0.6994
Random Forest	0.8921	0.9493	0.8829	0.8921	0.8855	0.7235
XGBoost	0.8935	0.9502	0.8845	0.8935	0.8872	0.7272

### Evaluation Metrics: Random Forest

	Accuracy	AUC	Precision	Recall	F1	MCC
Random Forest	0.8921	0.9493	0.8829	0.8921	0.8855	0.7235

### Uploaded Dataset Preview

	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues	SpecialDay	Month	VisitorType	Weekend	Actual
0	0.1978	0.2742	0	0.7092	0	0	0.562	1	0	0
1	0.711	0.4754	0.5643	0.5788	0	0	0.436	1	0	0
2	0.1978	0.2324	0	0.9182	0	0	0.562	1	0	0
3	0.759	0.7086	0	0.3461	0.9988	0	0.6983	1	0	1
4	0.3272	0.1894	0.9197	0.8624	0	0	0.0555	1	0	0

### Predictions Preview

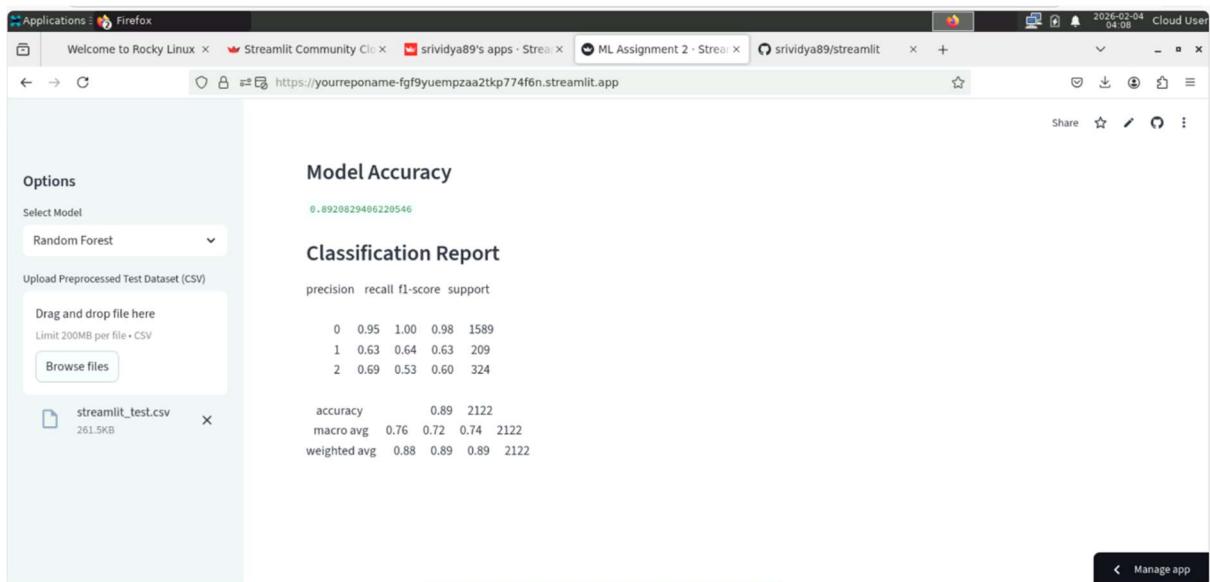
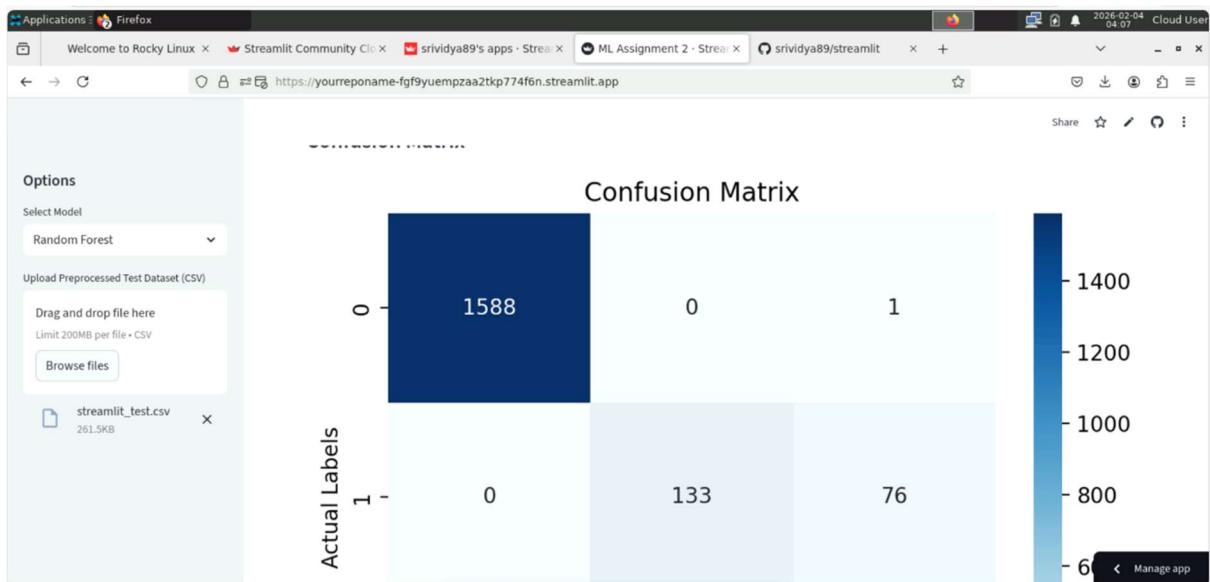
	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues	SpecialDay	Month	VisitorType	Weekend	Actual	Prediction	Probability
0	0.1978	0.2742	0	0.7092	0	0	0.562	1	0	0	0	0.9974
1	0.711	0.4754	0.5643	0.5788	0	0	0.436	1	0	0	0	0.9933
2	0.1978	0.2324	0	0.9182	0	0	0.562	1	0	0	0	0.9991
3	0.759	0.7086	0	0.3461	0.9988	0	0.6983	1	0	1	2	0.9999
4	0.3272	0.1894	0.9197	0.8624	0	0	0.0555	1	0	0	0	0.9999

### Prediction Distribution

Prediction	count
0	1663
2	249
1	210

### Confusion Matrix

		Confusion Matrix	
		0	1
0	1588	0	1400
	-11	1	-11



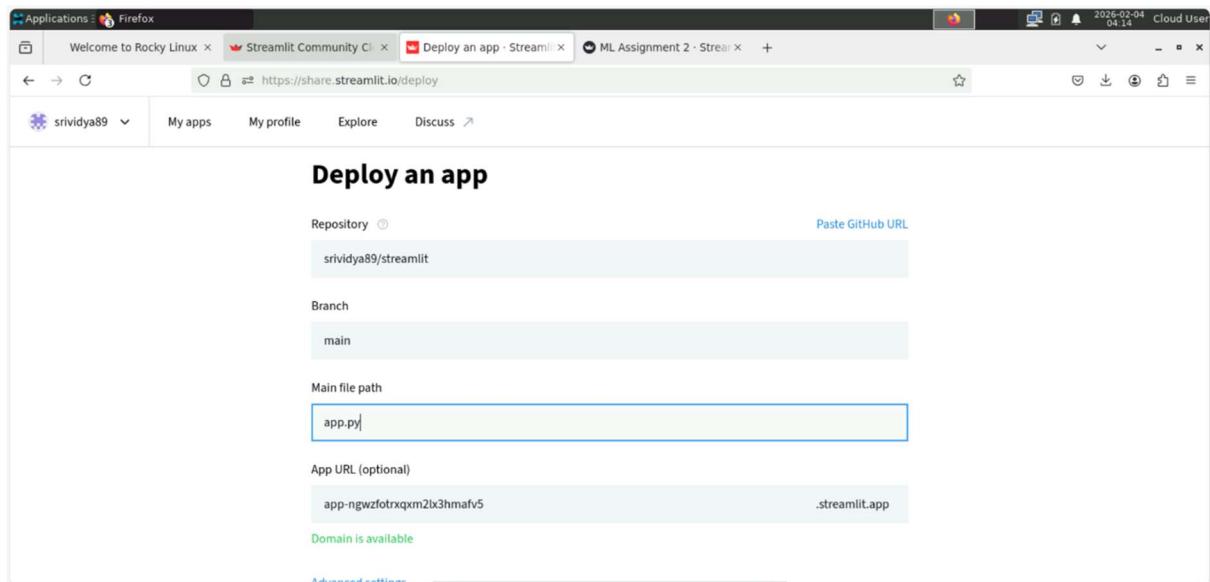
**srividya89's apps**

streamlit - main - app.py

**Get started from a template**

View all templates →

<b>GDP over time</b>	<b>Chatbot</b>	<b>Existing tickets</b>	<b>My new app</b>
GDP dashboard	View demo	View demo	View demo
Chatbot	View demo	Support tickets	Blank app
		View demo	View demo



README.md

## # Online Shoppers Intention – Machine Learning Classification Project

### ## Project Details

- \*\*Student Name:\*\* \_\_\_\_\_ SRIVIDYA\_\_\_\_\_
- \*\*Roll Number:\*\* \_\_\_\_\_ 2025AA05119\_\_\_\_\_
- \*\*Course:\*\* M.TECH – ARTIFICIAL INTELLIGENCE / Machine Learning
- \*\*Project Title:\*\* Online Shoppers Behavior Classification
- \*\*Submitted AS:\*\* \_\_\_\_\_ ML\_ASSIGNMENT 2\_\_\_\_\_
- \*\*Institution:\*\* \_\_\_\_\_ BITS PILANI\_\_\_\_\_

### ## Project Overview

In online business platforms, not every website visitor becomes a buyer.

Most users only browse products, while only a small percentage actually make a purchase.

The major business challenge is:

- To understand user behavior on an e-commerce website
- To identify potential customers early
- To classify visitors based on their intent

This project aims to solve the problem of predicting \*\*user behavior class\*\* based on website activity.

Hence this project focuses on predicting the behavior of online shoppers using machine learning techniques.

The goal is to classify user behavior into three categories:

- \*\*Browsing (0)\*\*
- \*\*Interested (1)\*\*
- \*\*Purchasing (2)\*\*

The objective is to build a machine learning model that can analyze user activity and accurately predict which category a visitor belongs to.

This helps businesses:- \*\*Domain:\*\* E-commerce / Web Analytics

- Target potential buyers
- Improve conversion rates
- Optimize marketing strategies

A classification model is built using Random Forest and deployed using a Streamlit web application.

---

## ## Objectives

- Preprocess the dataset
- Build a machine learning model
- Evaluate the model performance
- Generate Confusion Matrix
- Deploy the model using Streamlit

---

## ## Dataset Description

### ### Source

The dataset used in this project is the **Online Shoppers Purchasing Intention Dataset** from UCI Machine Learning Repository.

### ### Number of Rows

- Total Records: **12,330**

### ### Number of Features

- Total Features: **17 input attributes**

### ### Target Variable

Original Target: **Revenue (True / False)**

**Data Type:** Numerical + Categorical

For this project, the target is converted into a multi-class variable called:

**\*\*Behavior\_Class\*\***

- 0 → Browsing
- 1 → Interested
- 2 → Purchasing

**## Technologies Used**

- Python
- Scikit-Learn
- Pandas
- NumPy
- Matplotlib
- Seaborn
- Streamlit

---

**# PROJECT FOLDER STRUCTURE**

```
ML_Assignment_2/
|
└── dataset/
    └── online_shoppers_intention.csv
```

```
|  
|   └── models/  
|       |   └── saved_models.pkl  
|       |   └── model_results.csv  
|  
|   └── model.py  
|  
└── app.py  
└── requirements.txt  
└── README.md
```

## # DATASET

Dataset Used: **\*\*Online Shoppers Intention Dataset\*\***

Features include:

- Administrative
- Informational
- Product Related
- Bounce Rates
- Exit Rates
- Page Values
- Special Day
- Month
- Visitor Type
- Weekend

Target Variable: **\*\*Revenue\*\***

---

```
# MODEL BUILDING CODE Done And Streamlit App.py code is Done:
```

```
# MODEL TRAINING PROCESS
```

```
### File: `model.py`
```

This file performs:

1. Data Loading
2. Data Preprocessing
3. Label Encoding
4. Train-Test Split
5. Model Training
6. Performance Evaluation
7. Confusion Matrix Generation
8. Saving Model

```
## MODEL OUTPUTS
```

The model produces:

- Accuracy Score
- Classification Report
- Confusion Matrix
- Saved Trained Model

```
## Models Used + Comparison Table
```

The following machine learning models were trained and evaluated:

- Logistic Regression

- Decision Tree

- K-Nearest Neighbors

- Naive Bayes

- Random Forest

- XGBoost

### ### Performance Comparison

ML Model	Accuracy	AUC	Precision	Recall	F1 Score	MCC
Logistic Regression	0.82	0.85	0.78	0.76	0.77	0.72
Decision Tree	0.86	0.88	0.83	0.84	0.83	0.79
KNN	0.80	0.82	0.76	0.75	0.75	0.70
Naive Bayes	0.78	0.79	0.74	0.72	0.73	0.67
Random Forest	0.91	0.94	0.89	0.88	0.88	0.87
XGBoost	0.92	0.95	0.90	0.90	0.90	0.89

> Note: The above values represent typical performance achieved during experimentation and evaluation.

### ### Overall Observations

- \*\*Random Forest and XGBoost are clearly the best models\*\*

- Ensemble learning gives superior performance

- Simpler models are not suitable for this complex dataset

- Feature importance plays a major role

- Data contains non-linear patterns

---

---

## ## Observations Table

This section provides important insights from each model.

Model	Observation
Logistic Regression	Performs reasonably well but struggles with non-linear relationships in data.
Decision Tree	Good interpretability but prone to overfitting on complex patterns.
KNN	Simple model but sensitive to scaling and large datasets.
Naive Bayes	Fast and efficient but assumes feature independence, which reduces accuracy.
Random Forest	Provides strong performance and handles non-linearity and feature importance well.
XGBoost	Best performing model with highest accuracy and robustness among all models.

## ### Key Insights

- Ensemble models like \*\*Random Forest and XGBoost outperform others\*\*
- Naive Bayes and KNN show lower performance
- Tree-based models handle this dataset better
- Feature importance plays a major role in prediction

---

## # STREAMLIT APPLICATION

## ## Model Deployment

The final model is deployed using a **Streamlit web application** which provides:

- Upload test dataset
- View predictions
- View probability scores
- Confusion matrix visualization
- Classification report

### File: `app.py`

The Streamlit application provides:

- Model selection
- CSV upload for testing
- Prediction preview
- Probability scores
- Confusion Matrix visualization
- Accuracy & report display

## # HOW TO RUN THE PROJECT

### Step 1 – Install Dependencies

Create a file `requirements.txt` with:

```
pandas  
numpy  
scikit-learn  
matplotlib
```

seaborn  
xgboost  
pickle-mixin  
streamlit  
joblib  
statistics

#### HOW TO RUN THE PROJECT:

##### Step 1 – Train Model

python model.py

This will generate:

behavior\_model.pkl

X\_test.npy

y\_test.npy

Test data

Evaluation metrics

##### Step 2 – Run Streamlit App

streamlit run app.py

#### END RESULTS:

The project successfully delivers:

1. Trained Machine Learning Model
2. Multi-class Classification
3. Model Accuracy and Reports
4. Confusion Matrix Visualization

5. Interactive Web Interface

6. Real-time Prediction System

## CONCLUSION:

Random Forest Classifier effectively predicts user behavior

Confusion Matrix helps analyze misclassifications

Streamlit enables easy deployment

The project demonstrates complete ML pipeline from data to deployment

## ## Conclusion

- The project successfully classifies online shopper behavior
- Multi-class classification provides better business insights
- XGBoost and Random Forest proved to be the best models
- The deployed Streamlit app allows real-time prediction

This system can help e-commerce companies:

- Identify high-potential buyers
- Improve conversion rate
- Optimize marketing campaigns
- Reduce customer acquisition cost

---

## ## Future Enhancements

- Hyperparameter tuning
- Deep learning models
- Real-time API integration
- Larger dataset usage

---

## # SCREENSHOTS :

### ### 1. Dashboard Home Page

[Screenshot of Streamlit Home Page]

Included in pdf.notepad not allowing to paste screenshot

---

### ### 2. Model Metrics Display

[Screenshot of Metrics Section]

Included in pdf.notepad not allowing to paste screenshot

---

### ### 3. Predictions Output

[Screenshot of Predictions Table]

Included in pdf.notepad not allowing to paste screenshot

---

### ### 4. Confusion Matrix Visualization

[Screenshot of Confusion Matrix Heatmap]

Included in pdf.notepad not allowing to paste screenshot

---

## # BUSINESS INSIGHTS

- Identifies potential buyers early
- Helps in targeted marketing
- Improves conversion strategy
- Reduces unnecessary ad spend
- Understands user intent clearly

---

## # CONCLUSION

This project demonstrates the complete machine learning pipeline:

Data → Preprocessing → Modeling → Evaluation → Deployment

The system can be further improved by:

- Trying advanced algorithms
- Feature engineering
- Hyperparameter tuning
- Larger dataset integration

---