Bank Term Deposit Predictor - Project Report

1. Project Overview

This project involves the development and deployment of a machine learning model to predict whether a client will subscribe to a term deposit based on historical marketing data from a Portuguese banking institution. The solution includes a data pipeline for preprocessing and training, and a web application interface using Streamlit for end-user interaction.

2. Dataset

- **Source:** UCI Machine Learning Repository - Bank Marketing Dataset
- **Records:** 41,188
- **Features:** 20 (including both categorical and numerical)
- **Target Variable:** y (whether the client subscribed to a term deposit: ‘yes’ or ‘no’)

3. Tools & Technologies

- **Programming Language:** Python 3.10
- **Libraries:** pandas, scikit-learn, joblib, streamlit, xgboost, seaborn, matplotlib, imbalanced-learn
- **Deployment:** Streamlit Community Cloud

4. Data Preprocessing

- Removed leading/trailing spaces from column names.
- Identified categorical features:
    - `job`, `marital`, `education`, `default`, `housing`, `loan`, `contact`, `month`, `day_of_week`, `poutcome`
- Applied `OneHotEncoder` to categorical features using `ColumnTransformer`.
- Retained all other columns using the `remainder='passthrough'` option.

5. Model Building

- Constructed a pipeline with two main steps:
    1. **Preprocessing:** One-hot encoding of categorical columns
    2. **Classification:** Random Forest Classifier with `class_weight='balanced'`
- Used scikit-learn version **1.2.2** to ensure model compatibility during deployment
- Saved the trained pipeline using `joblib`

6. Streamlit Web App

- A user interface was created using Streamlit to:
    - Display model information

    - Collect user inputs for all model features
    - Predict the likelihood of a term deposit subscription
    - Display prediction results in a user-friendly format

7. Deployment

The model was deployed on **Streamlit Community Cloud (**https://bank-term-deposit-predictor-p6z5scte7dhhugd3haiapu.streamlit.app/**)**

- A custom `requirements.txt` was created with exact package versions:

  ```
  numpy
  matplotlib
  seaborn
  imbalanced-learn
  xgboost
  streamlit==1.35.0
  pandas==1.5.3
  scikit-learn==1.2.2
  joblib==1.2.0
  ```

- Deployment issue related to model loading (`_RemainderColsList`) was resolved by matching the scikit-learn version used during training and deployment.

8. Results & Observations

- The trained pipeline was successfully loaded and tested locally and in the deployed app.
- The app provides accurate predictions and is easy to use.

9. Conclusion

The project demonstrates the complete lifecycle of a machine learning project—from data preprocessing and model training to building an interactive web application and deploying it online. Matching library versions during training and deployment was key to resolving compatibility issues.

10. Future Improvements

- Integrate model evaluation metrics (accuracy, precision, recall, F1-score).
- Implement logging and error handling in the Streamlit app.
- Add model explanation tools such as SHAP or LIME.
- Explore hyperparameter tuning for model optimization.

Author: Michael Odumanye
Project: Azubi Africa TMP