Relationship Mapping Framework

Assignment 2: Relationship Mapping Framework

Group 2

April 25, 2025

1 Assignment Overview

Objective:
Develop a standalone system to identify, classify, and map relationships between entities in hoax call networks. The system is designed to perform the following:

- Develop edge representation models for at least 5 relationship types in hoax call networks.
- Create relationship extraction algorithms from communication data.
- Implement edge attribute schemas capturing relationship strength and characteristics.
- Build filtering mechanisms for different relationship types.
- Create a visualization system for multi-relationship networks.

Dataset Used:

- Call Detail Record (CDR) Datasets (e.g., from Kaggle) for telecom

2 Setup Instructions

Installation

1. Ensure you have Python 3 installed on your system.
2. Clone the repository to your local machine.
3. (Optional) Create and activate a virtual environment:

```
python -m venv venv
source venv/bin/activate  % For Unix/Linux/MacOS
venv\Scripts\activate     % For Windows
```

4. Install the required dependencies using:

```
pip install -r requirements.txt
```

Running the Application

1. Launch the Streamlit app by running:

```
streamlit run app.py
```

2. Use the interactive interface to input call statistics and view predictions.

3 Directory Structure

```
.
app.py                  % Streamlit application source code
CDR-Call-Details.csv    % Sample call detail records dataset
README.tex              % This README file (LaTeX source)
requirements.txt        % List of Python package dependencies
sample_output.png       % Screenshot/sample output of the app (optional)
```

4 Dependencies

The project relies on the following Python libraries:

- **streamlit** – Web app framework for interactive dashboards.
- **pandas** and **numpy** – For data manipulation and numerical computations.
- **scikit-learn** – For preprocessing, label encoding, scaling, and metrics.
- **xgboost** – For the XGBoost classification model.
- **seaborn** and **matplotlib** – For plotting static visualizations.
- **plotly** – For interactive bar chart visualization.

5 Features Implemented

- **Relationship Mapping Framework:** Models and extracts relationships in hoax call networks.
- **Edge Representation:** Designed for multiple relationship types with weighted attributes.
- **Filtering Mechanisms:** Allows filtering by relationship type.
- **Interactive Classification:** A Streamlit app to predict call behavior and assess hoax risk.
- **Mini Classification Report:** Displays precision, recall, and F1-scores per class.
- **Interactive Visualizations:** Includes an interactive bar chart (Plotly) for F1-scores and a seaborn heatmap for the confusion matrix.

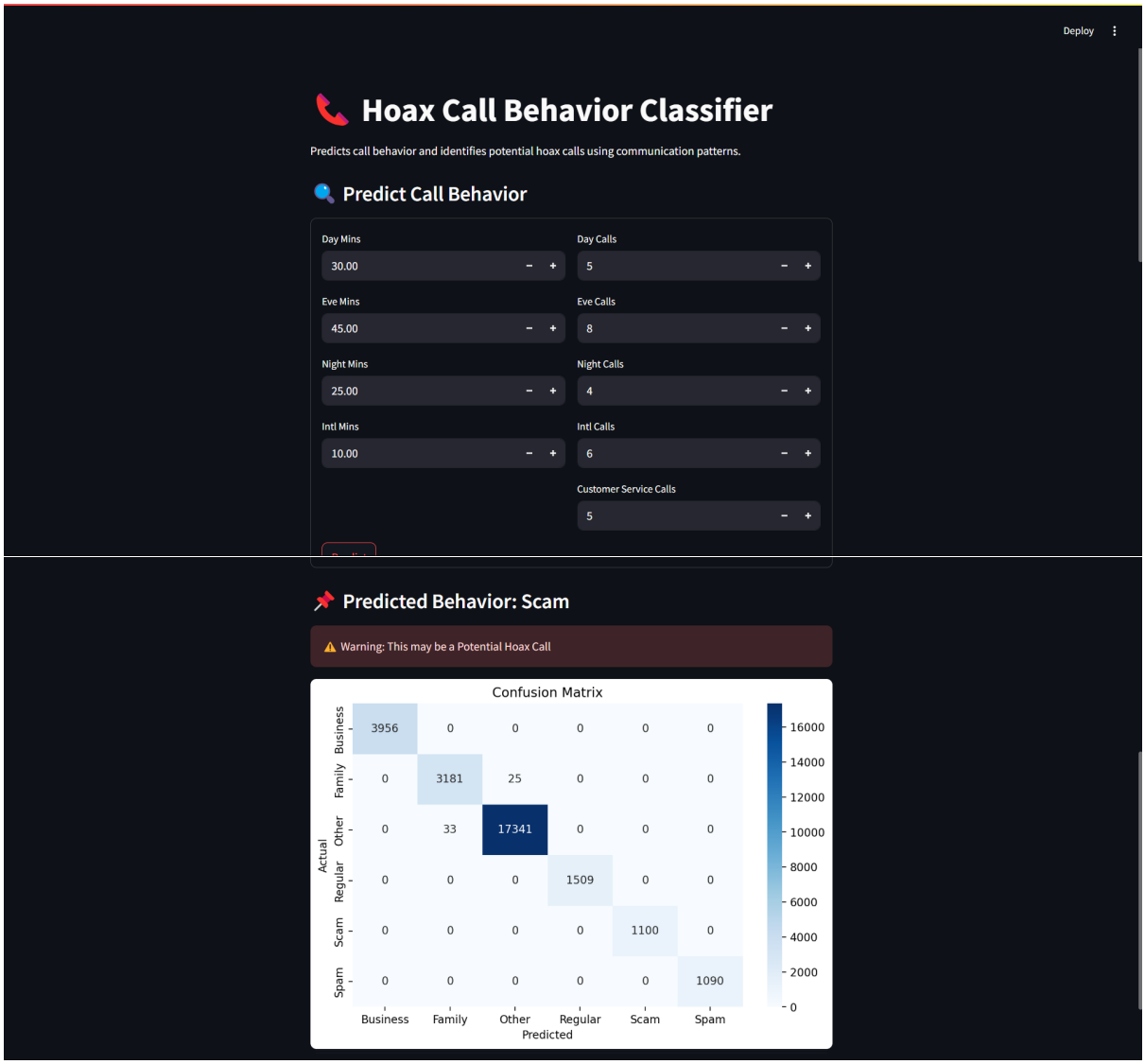
6 Instructions to Reproduce Results

1. Follow the setup instructions to install the necessary dependencies.
2. Run the Streamlit application using the command provided in the setup section.
3. Input sample call statistics through the web interface and view the output.
4. Explore the displayed classification report and interactive graphs.

7 Sample Outputs

Below are some sample outputs from the application.

Prediction Interface



```
Classification Report:
              precision    recall  f1-score   support

    Business       1.00      1.00      1.00      5999
      Family       0.93      0.95      0.94      4817
       Other       0.95      0.98      0.97     26071
     Regular       0.96      0.86      0.91      2252
        Scam       0.93      0.93      0.93      1640
        Spam       0.99      0.55      0.71      1574

    accuracy                           0.96     42353
   macro avg       0.96      0.88      0.91     42353
weighted avg       0.96      0.96      0.95     42353
```