Task 1 - Exploratory Data Analysis (EDA)

Using the dataset specified in the **Dataset** section at page 6, conduct an EDA and create an interactive notebook (.ipynb file) in **Python** that can be used as a presentation to explain the findings of your analysis. It should contain appropriate visualisations and explanations to assist readers in understanding how these elaborations are arrived at and their implications.

Deliverable

1. Jupyter Notebook in **Python**: a `.ipynb` file named **`eda.ipynb`**. (do adhere to the naming requirement)

Evaluation

In the submitted notebook, you are required to

1. Outline the steps taken in the EDA process
2. Explain the purpose of each step
3. Explain the conclusions drawn from each step
4. Explain the interpretation of the various statistics generated and how they impact your analysis
5. Generate clear, meaningful, and understandable visualizations that support your findings
6. Organise the notebook so that is it clear and easy to understand

Please note that your submission will be heavily penalised for any of the following conditions:

1. `.ipynb` missing in the submitted repository
2. `.ipynb` cannot be opened on Jupyter Notebook
3. Explanations missing or unclear in the submitted Jupyter Notebook

Task 2: End-to-end Machine Learning Pipeline

Design and create a machine learning pipeline (MLP) in Python scripts (`.py` files) that will ingest and process the entailed dataset, subsequently, feeding it into the machine learning algorithm(s) of your choice.

Do not develop your MLP in an interactive notebook.

The pipeline should be easily configurable to enable easy experimentation of different algorithms and parameters as well as ways of processing data. You can consider the usage of a config file, environment variables, or command line parameters.

Within the pipeline, data (provided in the Dataset section, Page 6) must be fetched/imported using SQLite, or any similar packages.

Deliverables

1. A folder named `**src**` containing Python modules/classes in `.py` format.
2. An executable bash script `**run.sh**` at the base folder of your submission to run the aforementioned modules/classes/scripts. DO NOT install your dependencies in the `run.sh`; this will be taken care of automatically when we assess the assignment if you have created your `requirements.txt` correctly.
3. A `**requirements.txt**` file in the base folder of your submission.
4. A `**README.md**` file that sufficiently explains the pipeline design and its usage. You are required to explain the thought process behind your submitted pipeline in the README. The README is expected to contain the following:
    a. Full name (as in NRIC) and email address (stated in your application form).
    b. Overview of the submitted folder and the folder structure.
    c. Instructions for executing the pipeline and modifying any parameters.
    d. Description of logical steps/flow of the pipeline. If you find it useful, please feel free to include suitable visualisation aids (eg, flow charts) within the README.
    e. Overview of key findings from the EDA conducted in Task 1 and the choices made in the pipeline based on these findings, particularly any feature engineering. Please keep the details of the EDA in the `.ipynb`. The information in the `README.md` should be a quick summary of the details from `.ipynb`.
    f. Describe how the features in the dataset are processed (summarised in a table).
    g. Explanation of your choice of models for each machine learning task.
    h. Evaluation of the models developed. Any metrics used in the evaluation should also be explained.
    i. Other considerations for deploying the models developed.

Evaluation

The submitted MLP, including the `README.md`, will be used to assess your understanding of machine learning models/algorithms as well as your ability to design and develop a machine learning pipeline. Specifically, you will be assessed on

1. Appropriate data preprocessing and feature engineering
2. Appropriate use and optimization of algorithms/models
3. Appropriate explanation for the choice of algorithms/models
4. Appropriate use of evaluation metrics
5. Appropriate explanation for the choice of evaluation metrics
6. Understanding of the different components in the machine learning pipeline

In your submitted Python scripts (`.py` files), you will be assessed on the quality of your code in terms of reusability, readability, and self-explanatory.

Please note that your submission will be penalised for any of the following conditions:

1. Incorrect format for `requirements.txt`
2. `run.sh` fails upon execution
3. Poorly structured `README.md`
4. Disorganised code that fails to make use of functions and/or classes for reusability
5. MLP not submitted in Python scripts (`.py` files), including MLP built using Jupyter Notebooks.

Note for Windows users

DO NOT submit a Windows batch (`*.bat`) script in replacement of the bash script. Use either 'Windows Subsystem for Linux (WSL)' or 'Git Bash'/'cygwin' for the creation of the bash script.

Problem Statement

Objectives

In 2023, Singapore faced a significant challenge with telecommunication fraud. Despite telecommunications operators successfully blocking approximately 300 million scam calls, as reported by the Infocomm Media Development Authority (IMDA), Singapore still saw over 46,000 scam cases, marking an eight-year rise in such incidents. This resulted in a substantial financial loss of S$651.8 million. The continuous increase in scam activities underscores the urgent need for more robust preventive measures.

As a machine learning engineer at a leading telecommunications firm, you are tasked with developing machine learning models to classify phone calls as 'scam' or 'not scam'. You will utilise historical call data to develop models that identify and learn from patterns associated with fraudulent activities, thereby enhancing the security of phone communications.

Upon detecting a call as a potential scam, your models will facilitate the automatic triggering of an SMS warning to the user immediately. This proactive approach is designed to strengthen individual defences against fraud, decrease the incidence of scam cases, and ultimately mitigate the financial losses associated with these crimes.

In your submission, you are expected to evaluate **at least three suitable models** for predicting whether a phone call is a scam call.

Dataset

The dataset provided contains information that your telecommunications firm has collected regarding scam calls. Do note that there could be synthetic features in the dataset. Therefore, you would need to state and verify any assumptions that you make.

You can query the datasets using the following URL:
https://techassessment.blob.core.windows.net/aiap17-assessment-data/calls.db

Instructions for setting up SQLite and querying the database

The dataset can be accessed through the `calls.db`. You may find either of the following packages, `SQLite` or `SQLAlchemy`, useful for accessing this database.

You should place the `calls.db` file in a `data` folder. Your machine learning pipeline should retrieve the dataset using the relative path `data/calls.db`.

DO NOT upload the `calls.db` onto your GitHub repository.

List of Attributes

Attribute	Description
ID	A unique identifier for each call.
Call Duration	Total duration of the call in seconds.
Call Frequency	Frequency of calls from the same caller number to all other numbers in the last 24 hours.
Financial Loss	Amount of money reported stolen by the receiver.
Flagged by Carrier	Whether the caller number was flagged by the carrier as suspicious before.
Is International	Whether the call is an international call.
Previous Contact Count	Number of times the caller has called the same receiver before.
Country Prefix	Country code where the call is originating from.
Call Type	Medium used for the call.
Timestamp	The date and time when the call was made.
Device Battery	The battery status of the caller's device.
Scam Call	Indicating if the call is a Scam or Not Scam.