**Deliverables**

Task1: Email Identifier

**Prepare a write-up that answers the following:**

**1. Describe your approach/methodology to solve above mentioned tasks.**

1. **Understand the problem statement.**

* Here we need to identify the eReceipt emails from other emails like shipping updates/ marketing emails/refunds/return/ cancelation emails.
* It’s a pure ML classification problem with vectorization of text features using NLP techniques.
* Here I’m using a Binary classification problem as we have two types of email datasets (eReceipt, Shipping)

1. **Acquire the Data.**

* I have downloaded two HTML files (eReceipt and Shipping) and taken my emails of the same eReceipt and shipping update to create a data set as below image.

**A screen shot of a computer

Description automatically generated**

1. **Exploratory Data Analysis, Preprocessing, and Feature Engineering.**

* Converted the above HTML files to text and remove tags.
* Creating class labels (1 – eReceipt, 0 – Shipping) and other word-based features like email\_subject, email\_subject\_length, email\_body\_length, order\_keyword, shipping\_keyword, delivery\_keyword, email\_text\_length, email\_text\_words

**A screenshot of a white table

Description automatically generated**

* Preprocess text data to remove stop words, punctuations, and Special characters.

1. **Word Cloud – to understand the most frequent words in text columns.**

* Email Text - word cloud   
  A close-up of words

  Description automatically generated
* Email Subject – Word Cloud

A close-up of numbers and letters

Description automatically generated

1. **Standardization of Numerical features to maintain the same scale.**

**A white table with black text

Description automatically generated**

1. **Apply NLP techniques to text features.**

* There are various NLP techniques like BoW(Bag of words), Tf-IDF, W2V, Weighted W2V
* I utilized Tf-IDF \* Weighted W2v- to give importance to the most frequent and less frequent words, and multiplying with W2V will also cover semantic meaning.

1. **Stratified train and test split**

* I have split the data set into training data (70%) and testing data (30%)

1. **Building different machine learning models**

* Trained different machine learning models on training data to perform classification tasks - KNN, Naïve Bayes, Logistic Regression, Liner SVM, Decision Tree, Random Forest, GBDT
* Tuned hyperparameter on each algorithm to find better accuracy.

A screenshot of a computer program

Description automatically generated

1. **Model evaluation**

* Since it’s a classification problem, we can evaluate using various parameters like Accuracy, Confusion Matrix, Log-loss, Precision, Recall, etc.
* I have utilized the log-loss parameter to find train and test log loss.

**A graph with a green line

Description automatically generatedA blue squares with white text

Description automatically generated**

1. **Champion Model**

* Based on the log-loss error, I choose the Champaign model among all trained classification models.
* We know log-loss would vary from 0 to infinity. Less the log-loss error better the model
* I have also used a confusion matrix with precision and Recall.   
    
  A close-up of a train log loss

  Description automatically generated

**Note: Please refer to the attached Jupiter Notebook for better understanding.**

1. **Description of Machine Learning/Deep Learning algorithms/models and frameworks you would choose.**

Among these NLP techniques like - Bow (Bag of words), Tf-Idf, W2v, weighted W2v   
I utilized Tf-IDF \* Weighted W2V as it will give more importance to the frequency of words and semantic meaning.

I have tried 4 machine learning classification models as below:

* Random Model – this is the worst model or dumb model, which sets the baseline for other models where we can expect better performance than other models.
* Logistic Regression – tried logistic regression model with hyperparameter tunning (L2 regressor)
* Linear SVM - I also tried an SVM classifier with hyperparameter tuning.
* XGBoost

**Note: Please refer to the attached Jupiter Notebook for better understanding.**

1. **How would you evaluate the proposed solution?**

* I have tried various ML classification models trained on training data with 10-fold cross-validation to tune hyperparameters.
* Based on the log-loss error, I chose the Champaign model among all trained classification models.
* We know log-loss would vary from 0 to infinity. Less the log-loss error better the model
* I have also used a confusion matrix with precision and Recall.

**Note: Please refer to the attached Jupiter Notebook for better understanding.**

1. **Diagrams/reference links etc. (code samples are a bonus)**

Please refer to the attached Jupiter notebook (Email\_Identifier.ipynb) containing the Python script.