Task 1 Report

Our program measures the likelihood of an email being a phishing attempt by looking at three different aspects of the email: the text content of an email, any URLs contained in the text of an email, and the email ID.

When looking at the text content of an email, the program uses three different machine learning algorithms to analyze likelihood of phishing. The machine learning algorithms are all implemented using Python's sklearn library. The three algorithms were trained on data which was marked as either a phishing attempt or not a phishing attempt, and were then operationalized with an sklearn Pipeline object to be able to make predictions on real data. When the text of an email is processed, each machine learning algorithm classifies it as either a phishing attempt (1) or a legitimate email (0). The votes of all three algorithms are added to the final score, so the highest contribution to the score from this section of the program (3) comes when all three algorithms vote that the email is a phishing attempt, and the lowest score (0) comes when all three vote that the email is legitimate. The algorithms may not always agree, so if only two of them classify the email as a phishing attempt, then this portion of the program will contribute a score of 2 to the final score, for example.

The program also checks any URLs to see if they contain suspicious keywords such as 'update,' 'login', or 'verify', or if the URL starts with a number. If the program detects a suspicious URL, it adds 1 to the score.

The program finally checks the email ID. If the email ID contains a number, then the program marks it as suspicious and adds 1 to the final score.

The sum of the three parts of the program comprises the final score. Since the first part has a maximum of 3, and the last two parts can contribute either 1 or 0, the final score of the email is a scale from 0 to 5, with 0 meaning the lowest probability of a phishing attempt and 5 meaning the highest probability of a phishing attempt.

The scopes of improving our security scanner for phishing attempts include the fact that the scanner will become more accurate when it is able to train its machine learning algorithms on a larger dataset. Also, whenever the machine learning algorithms make a mistake, the users could mark the classification as a mistake, and the algorithms could be retrained using this feedback in real time. In this way, the machine learning algorithms would actually become more accurate as time went on and more users used this program.