Kadin Matotek TP Project

1. **Read and process csv file**
    - Requirement: Read csv file
    - Implementation: iterate through emails
    - Remove all commas and 0/1 after email
    - Use array list and add each line (each email)
2. **Compute features for each email**
    - Requirement: Word Count, “URL” Count, “NUMBER” Count, I Count, “sex” count, hyperlink count, “you” count. Possibly more?
    - Implementation: Create a class to represent an email, get features for each email
    - Thinking about using a map / hashmap not sure which one yet
    - Possibly represent each feature with min, max, mean and median
3. **Email representation**
    - Requirement: “raw text (email)” + the features using a map / hashmap
    - Implementation: Email class with attributes for features and a label for known class (spam/not spam)
4. **Save features and summaries**
    - Requirement: Save email feature data and summary features in separate csv file
    - Implementation: Output file, need a refresher on this
    - Should be able to get read by a spreadsheet or statistics package
5. **Display features and summaries**
    - Requirement: Display individual email features and summary features
    - Implementation: Through GUI If I can figure it out
6. **Distance Metric/Computation**
    - Requirement: Compute euclidean distance between two emails based on features
    - Implementation: Create methods for computation. Calculate squared differences. Square root accumulate value
7. **Classification**
    - Requirement: Classify emails as spam or not spam based on distance from models
    - Implementation: Compare distance of an email to spam and not spam models, then classify.
    - Get email summary of spam model, and email summary of not spam model, then compare which is more similar.

UML

Email
-email id: int -rawText: String -features: Map<String,Int>
+ getEmailId(): int + getRawText(): String + getFeautres(): Map<String,int>

ReadFile
-emailList: ArrayList<Email> -spamModels: Map<String,Int> -notSpamModels: Map<String,Int>
+ReadFile() +getEmailList(): ArrayList<Email> +getSpamModel(): Map<String, Int> +getNotSpamModel(): Map<String, Int> +CalculateDistance(email: Email, model: Map<String,Int>): double +CalculateDistanceBetweenEmails(email: Email, email2: email): double +calculateSummaryData(emailGroup: List<Email>): Map<String,Int> +ClassifyEmail(email: Email: boolean +toString(): String

Calendar:
Week 1 (9/4) - Finish Draft, Figure how to use scanner to read file. Experiment with grabbing some features for each email, and putting them in a Map.
Week 2 (9/11) - Create and complete Email class. Probably the easiest part of implementation.
Week 3 (9/18) Start Read File Class. Figure how to read a file, assign each line to an Email Object, and create an ArrayList<Email> of them.
Week 4 (9/25) Figure which data I will be using for each data set, Create functions to both calculate the distance between other emails and the data set.
Week 5 (10/1) Should Be fully working by now, Try learning how to Implement data visually. Possible through GUI, Maybe export to excel?
Week 6 (10/8) Start working on showing data visually
Week 7 (10/15) Hopefully data can be shown visually by now, improve how it looks
Week 8 (10/22) Review everything, get opinion from peers