DATA 620 Final Project Proposal

# Project Title:

Spam Detection Using Text Classification on the Kaggle Cleaned Email Dataset

# Team Member:

Ariba Razzak (working individually)

# Guiding Question / Hypothesis

Can we accurately classify emails as spam or ham using only the text body of the message and standard NLP techniques?  
This project aims to evaluate how effective different feature extraction methods and machine learning classifiers are at detecting spam in cleaned email data.

# Motivation

Email spam continues to be a widespread nuisance and potential threat. Automatically detecting spam using machine learning is an important application of natural language processing (NLP). This project provides an opportunity to explore a real-world use case while applying course-relevant techniques in text processing and classification.

# Dataset

Source: Kaggle - Spam Mails Dataset (https://www.kaggle.com/datasets/venky73/spam-mails-dataset)  
  
Description:  
- Contains a collection of emails labeled as “spam” or “ham” (not spam).  
- Format: CSV file with `label` and `text` columns.  
- Cleaned and ready for direct use (no need to strip headers or HTML).

# Plan for Doing the Work

1. Data Loading and Exploration  
 - Import the dataset and perform basic EDA (class balance, text length, word distribution).  
2. Text Preprocessing  
 - Lowercasing, stopword removal, punctuation stripping, stemming or lemmatization.  
 - Vectorization using Bag of Words, TF-IDF, and possibly word embeddings.  
3. Model Building and Evaluation  
 - Train models: Naive Bayes, Logistic Regression, Random Forest, and SVM.  
 - Tune hyperparameters and compare model performance using accuracy, precision, recall, F1-score, and ROC-AUC.  
4. Visualization  
 - Word clouds, most frequent words, confusion matrices, ROC curves, and feature importance plots.  
5. Final Report  
 - Prepare a Jupyter Notebook with clean documentation and results.  
 - Submit a short video presentation summarizing the methodology, findings, and insights.

# Tools & Technologies

• Python (Pandas, Scikit-learn, NLTK or spaCy, Seaborn/Matplotlib)  
• Jupyter Notebook  
• GitHub for version control and submission

# Upfront Concerns

• Ensuring balanced evaluation given any class imbalance.  
• Deciding the right text vectorization method (TF-IDF vs embeddings) to optimize performance.  
• Explaining feature importance in models that don’t inherently provide it (like SVM).