Case Study Rubric: Comparing Model Performance in Identifying Spam Emails

Due: TBD
Submission format: Upload PDF and GitHub repository link to Canvas

General Description:

Submit to Canvas a PDF document and a link to your GitHub repository.

Why am I doing this?

This study is an opportunity to showcase your ability to clean data, engineer features, build and tune machine learning models, and critically evaluate their performance. It mirrors tasks common in professional data science and prototyping environments.

What am I going to do?

You will combine your technical skills in data preprocessing, model building, evaluation, and conceptual analysis to compare the performance of three machine learning models: K-Nearest Neighbor (KNN), Logistic Regression, and Decision Trees, in identifying spam emails.
Your final deliverable will include:

- A **written portion** submitted as a PDF (including figures and references)
- A **GitHub repository** containing your complete code and any necessary data

How will I know I have succeeded?

You will meet expectations when you carefully follow the criteria detailed below.

Category	**Details**
Formatting	Submit each required component: ● **Written Portion**: Submit a PDF file of your report, including an executive summary, methodology, analysis, results, discussion, and references. ● **GitHub Repository**: Upload all code and necessary data files. Repository must be titled "SpamModel-[FirstNameLastName]". ● **References**: Include a references page at the end of the written portion, formatted in **IEEE citation style**.
Written Portion	Your written report should clearly explain your process, thought process, and interpretation:

	• **Executive Summary**: Summarize the problem, your hypothesis, and your approach in a concise paragraph. • **Methodology**: Describe the data cleaning, preprocessing steps, and modeling approach, including a simple graphic outlining the workflow. • **Analysis**: Present exploratory data analysis (EDA) including insights about common words, spam/ham proportions, average email lengths, and text distributions. • **Results**: Discuss model performance, including accuracy and precision scores for each model. Clearly indicate if any model achieved at least 95% accuracy. • **Reflection**: In a short paragraph, discuss challenges faced (e.g., balancing the dataset, parameter tuning) and what you would do differently in the future.
Code	Your code must include the following components: • **Data Cleaning**: Remove duplicates and nulls, rename columns, map spam/ham labels. • **Text Preprocessing**: Lowercasing, tokenization, stopword removal, punctuation removal, stemming. • **Model Building**: Build and train KNN, Logistic Regression, and Decision Tree models. • **Model Evaluation**: Use 80/20 train/test split and 10-fold cross-validation. Calculate and report **accuracy** and **precision** for each model. • **Parameter Tuning**: Attempt tuning hyperparameters such as K for KNN and max_features for TF-IDF. • **Documentation**: Code must be clearly commented and organized into readable scripts or Jupyter notebooks.
References	• Include any external sources consulted beyond those given, properly cited in **IEEE format** at the end of the PDF.

Additional Notes:

- Models should be compared based on both **accuracy** and **precision.**
- Visualizations (such as word clouds, histograms of email lengths, or confusion matrices) are encouraged but not required.
- Your analysis should address whether your hypothesis about KNN was correct based on your findings.