Group 4:

Name: Babar Ali

Sap Id:65731

Name: Wahid Modassar

Sap Id:65292

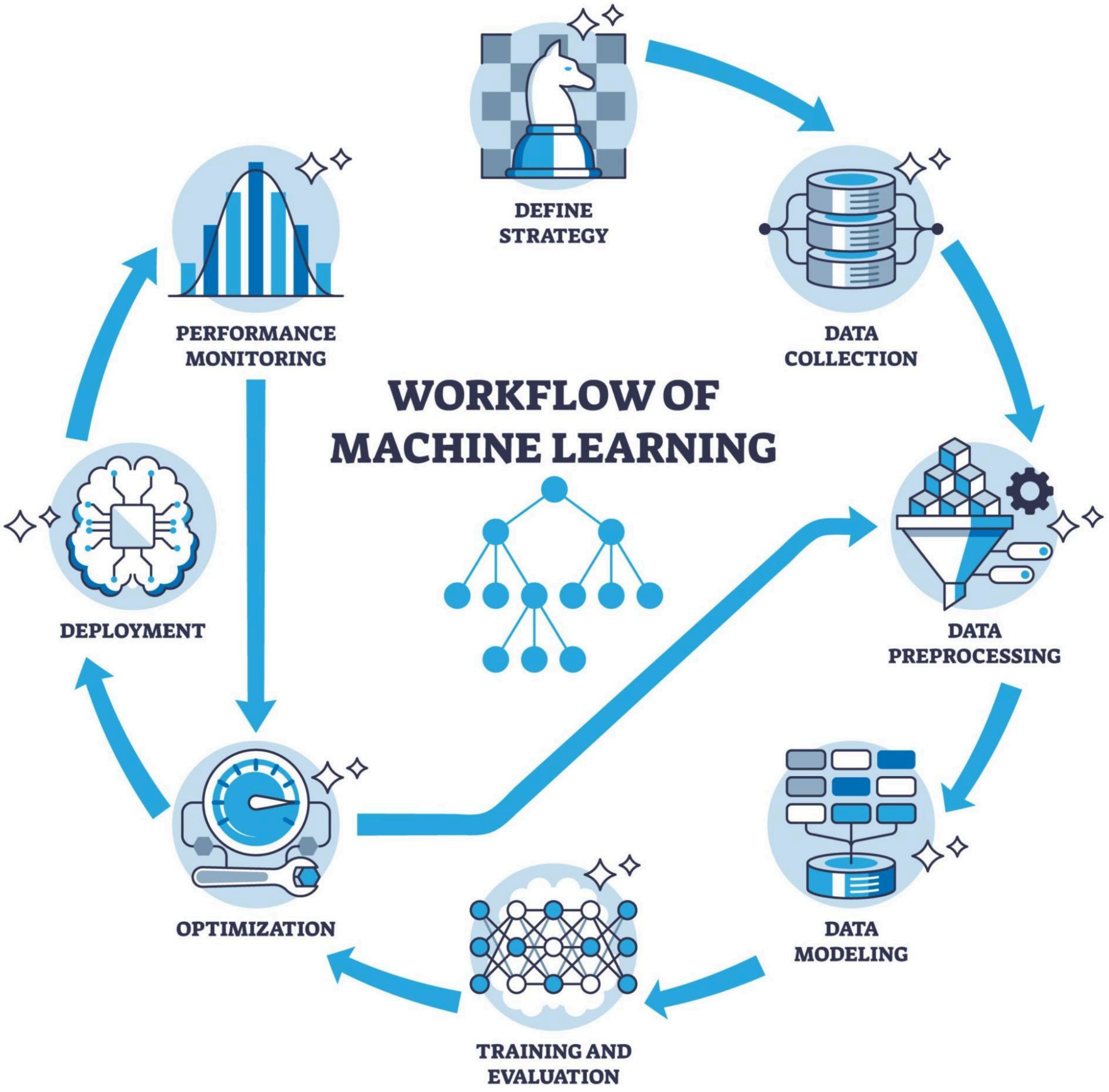
Submitted to: Sir Muhammad Junaid Khan

Technical Documentation: Email Spam Detection

1. Workflow Overview

The system follows a linear data processing pipeline designed to transform raw text into actionable predictions:

1. **Ingestion:** Data is fetched remotely from the GitHub repository (spam.csv).
2. **Cleaning:** The raw Message column is processed into a cleaned_text format using Natural Language Processing (NLP) techniques.
3. **Transformation:** Text is vectorized into a TF-IDF (Term Frequency-Inverse Document Frequency) matrix to represent words numerically.
4. **Training:** The Multinomial Naive Bayes model fits to the training set vectors.
5. **Validation:** The model predicts on the test set to generate performance metrics.
6. **Serialization:** The trained model and vectorizer are saved to disk (.pkl files) for future use.



Getty Images

2. Key Functions

Function Name	Description	Inputs	Outputs
preprocess_text	Cleans raw text for analysis.	text (str): Raw email string.	cleaned_text (str): Lowercase, tokenized string without stopwords/punctua

			tion.
train_test_split	Splits data for validation.	Features (X), Target (y).	X_train, X_test, y_train, y_test.
model.fit	Trains the classifier.	X_train, y_train.	Trained Model Object.
model.predict	Predicts class for new data.	X_test (Vectors).	Array of predictions (0 or 1).

3. Inputs and Outputs

Inputs

- **Dataset:** spam.csv (Accessed via pd.read_csv from GitHub URL).
  - *Category:* The label (ham/spam).
  - *Message:* The content of the email.
- **User Input:** Single raw string (e.g., "Free offer! Click now!") for testing predictions.

Outputs

- **Console Output:**
  - Accuracy score (float).
  - Classification Report (Precision, Recall, F1-score).
  - Prediction result for the test string.
- **Files:**
  - spam_detector_model.pkl: The serialized trained model.
  - tfidf_vectorizer.pkl: The serialized vectorizer vocabulary (required to transform new inputs).

4. Libraries and Dependencies

- **pandas:** Used for creating the DataFrame, handling CSV data, and managing columns.
- **nltk (Natural Language Toolkit):**
  - stopwords: List of common English words to ignore (e.g., "the", "is").
  - word_tokenize: Splits sentences into individual words.
- **re (Regular Expressions):** Used for removing URLs, HTML tags, and punctuation characters.
- **scikit-learn:**
  - TfidfVectorizer: Converts text to numerical features.
  - train_test_split: Splits the dataset into training and testing subsets.
  - MultinomialNB: The classification algorithm.
  - metrics: Calculates accuracy and generates the classification report.

- **joblib:** Used to save and load the trained model objects.

5. Execution Steps

To run this project, follow these steps in a Python environment (Jupyter Notebook or Google Colab):

1. Install Dependencies:
   Run the command !pip install pandas nltk scikit-learn numpy to ensure all required libraries are installed in the environment.
2. Load Data:
   Execute the cell that defines the GITHUB_RAW_URL and runs pd.read_csv(). This fetches the data directly from the repository without requiring manual file uploads.
3. Preprocess:
   Run the cell containing the preprocess_text function definition and application. This will create the cleaned_text column in the DataFrame.
4. Train:
   Execute the vectorization and model training cell. This fits the MultinomialNB model to the training data.
5. Evaluate:
   Run the evaluation cell to see the Accuracy score and the detailed Classification Report.
6. Test:
   (Optional) Modify the new_email variable string with your own text and run the prediction cell to test specific messages.
7. Save:
   Run the final cell to export the .pkl files (spam_detector_model.pkl and tfidf_vectorizer.pkl) for deployment or later use.