جامعة أم القرى
UMM AL-QURA UNIVERSITY

Project Data Analysis 2

Name	ID
Amany Saeed Fallatah	44411940
Nahla Mohammad ALosaimi	44412006

COURSE PRESENTER

"DR. Omaima Fallatah "

Course Number: DS3114

Bachelor of science in Data Science

Kingdom of Saudi Arabia - Umm Al-Qura University

1. Introduction

Objective of the Analysis: Explain the goal of the project, which is to classify emails as spam or not spam using machine learning algorithms.

Dataset Description: Provide a brief description of the dataset used (emails.csv), including features like the email content (text) and the target variable (spam).

Problem Statement: Discuss the importance of spam detection in email systems, and why machine learning techniques are valuable for this task.

2. Data Preprocessing

Loading the Data: Explain how the data is loaded into a DataFrame using pandas and mention any data format requirements.

Exploratory Data Analysis (EDA): Summarize the results from:

Checking the shape of the dataset using df.shape.

Viewing column names to understand the features in the dataset.

Removing duplicates to clean the data.

Checking for missing values with df.isnull().sum() and confirming there are none.

Text Processing and Cleaning: Explain the text preprocessing function process:

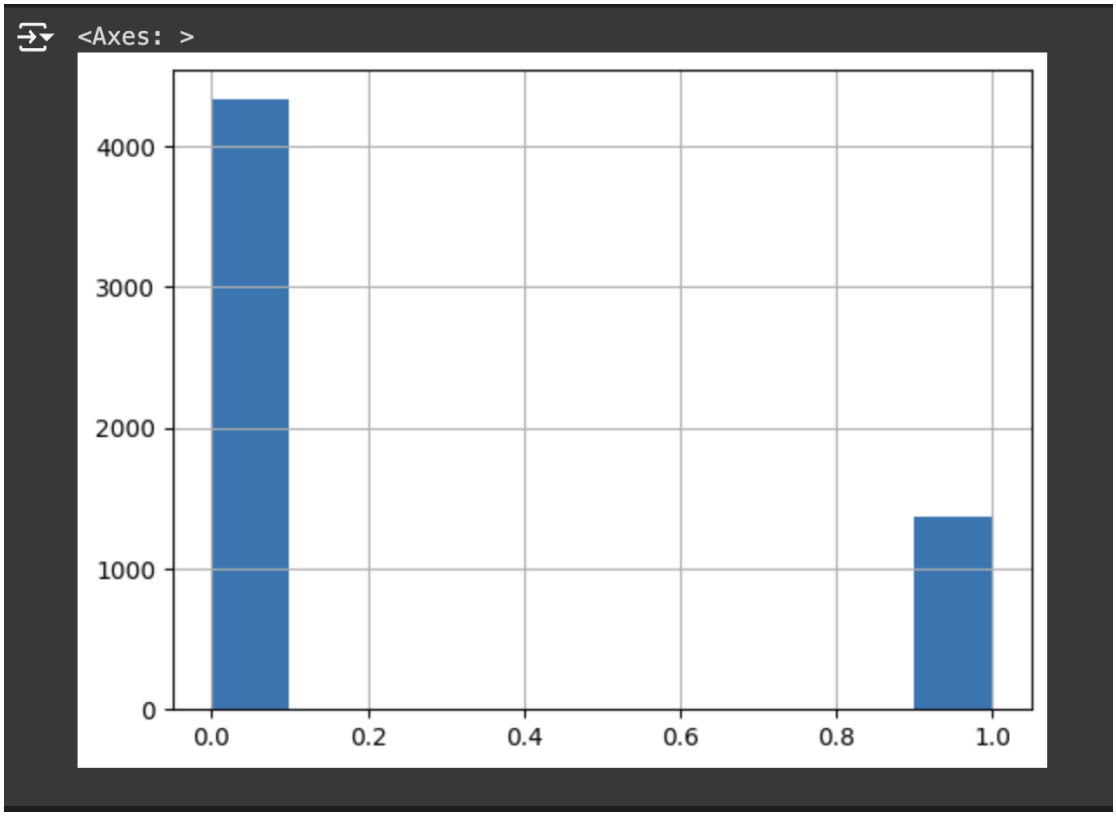
Tokenization: Splitting text into individual words.

Punctuation Removal: Removing punctuation from the text.

Stopword Removal: Removing common English stopwords to reduce noise.

Downloading Stopwords: Mention downloading the stopwords package using nltk.download("stopwords").

Data Visualization:



lustration showing the number of spam and not spam

Data Visualization:

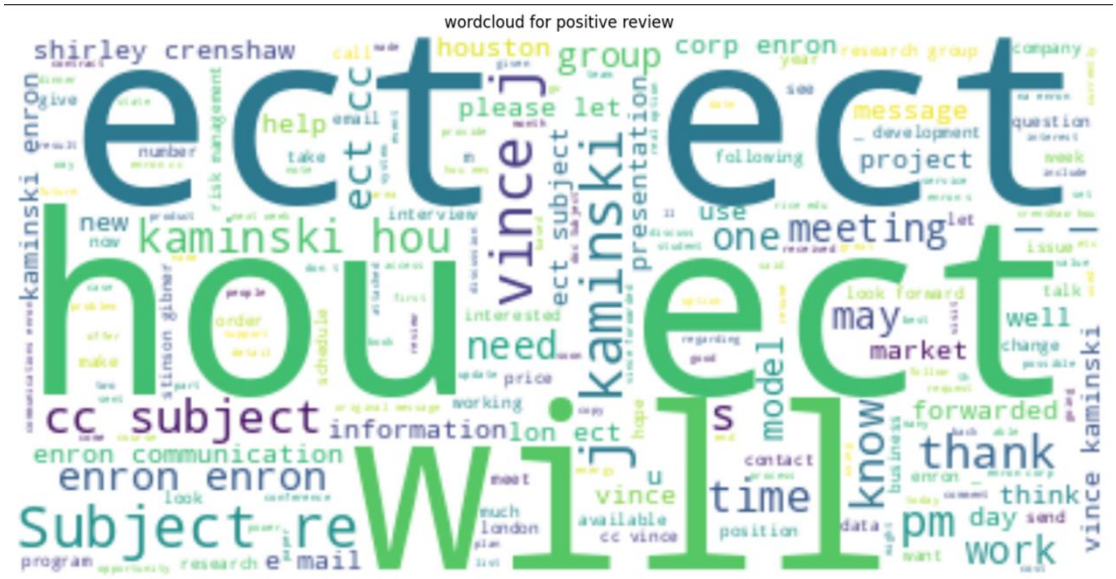
To generate a word cloud from the text available in the text column of messages classified as "spam." A word cloud is a visual representation of text where the most frequently occurring words are displayed larger in the image, such as "subject," د"money," etc

Importing Libraries:

-WordCloud: to generate the word cloud

-matplotlib.pyplot: to display the images and plots

Data Visualization:



The code generates a Word Cloud from the text found in messages classified as "not spam" using the Python library WordCloud. A Word Cloud is a visual tool that displays the most frequently occurring words in larger sizes, allowing you to quickly see the prominent words in the data. For example, in the Word Cloud provided, words like "ect", "hou", "subject", and "will" appear as the most repeated words in the "not spam" messages

Importing Libraries:

- WordCloud: This is used to create the word cloud visualization.
- matplotlib.pyplot: This is used to display the word cloud as an image.

3. Feature Extraction :

Bag of Words (BoW) Approach Using CountVectorizer:

Explain how CountVectorizer is used to convert text data into a numerical format suitable for machine learning.

Describe the analyzer=process argument that customizes the tokenization process.

Shape of the Transformed Data:

Show the shape of the feature matrix x using x.shape.

4. Splitting Data for Training and Testing

Train-Test Split:

Explain splitting the data into training and testing sets using train_test_split, with an 80-20 split.

Shape of the Data: •

Display the shape of xtrain, xtest, ytrain, and ytest.

5. Model Training and Evaluation: Naive Bayes Classifier

Training the Model:

Discuss training a MultinomialNB classifier on the training set (xtrain, ytrain).

Training Set Evaluation:

Present evaluation metrics on the training set:

Accuracy, Precision, Recall, F1-Score: Explain each metric and why it is important for classification tasks.

Confusion Matrix: Visualize and describe the confusion matrix, which shows true positives, false positives, true negatives, and false negatives.

Testing Set Evaluation:

Present similar metrics for the testing set, and compare them to the training set results to assess overfitting or underfitting.

6. Model Training and Evaluation: Logistic Regression

Training the Model:

Explain training a LogisticRegression model on the training set (xtrain, ytrain).

Training Set Evaluation:

Present evaluation metrics (accuracy, precision, recall, F1-score) and the confusion matrix for the training set.

Testing Set Evaluation:

Present evaluation metrics and the confusion matrix for the testing set.

Comparing Naive Bayes and Logistic Regression:

Compare the performance of the two models and discuss any trade-offs.

7. Conclusion and Recommendations

Summary of Findings:

Summarize the results from both models and discuss which model performed better on the test data.

Highlight any interesting observations from the confusion matrix or the classification reports.

Challenges and Limitations:

Discuss any issues encountered, such as class imbalance, data quality, or overfitting.

Future Work:

Suggest potential improvements, such as using more advanced models (e.g., Support Vector Machines, Random Forests), using different feature extraction techniques, or applying hyperparameter tuning.

Recommendations:

Recommend the best approach based on the findings and propose how the model could be deployed in a real-world email filtering system.

We must point out that logical registration is better than NB.