Data Collection and Preprocessing Phase

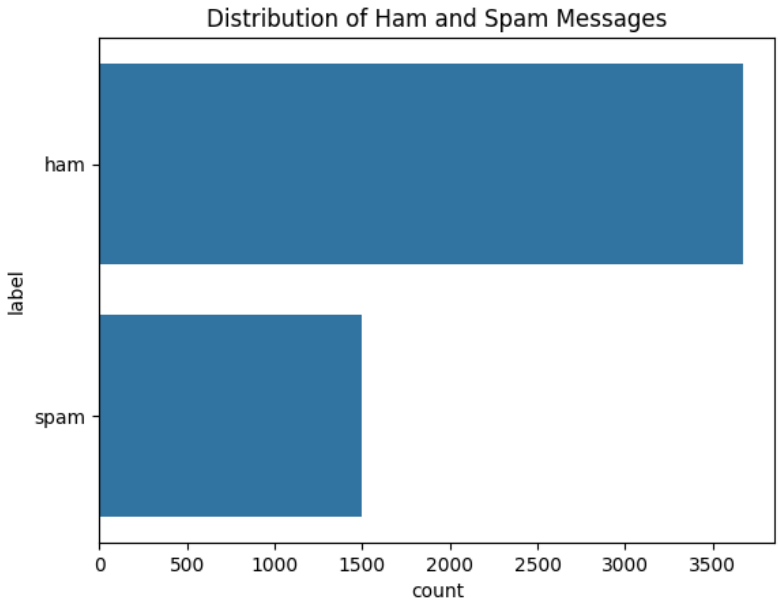
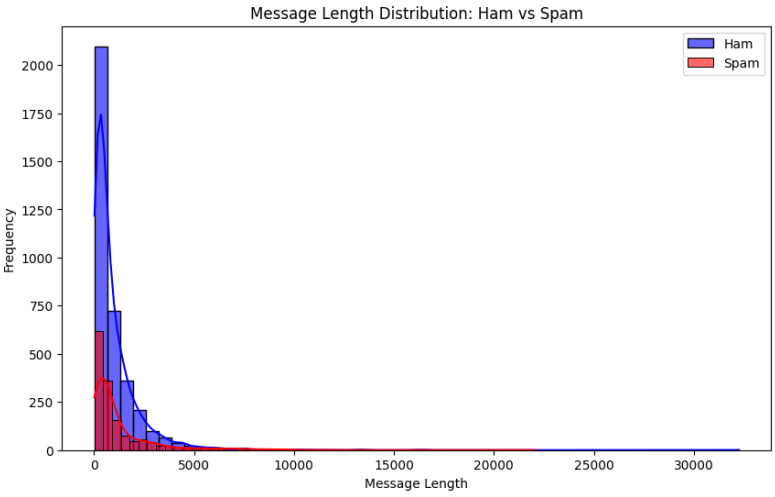
Date	4-10-2024
Team ID	LTVIP2024TMID25000
Project Title	SMS Spam Detection
Maximum Marks	6 Marks

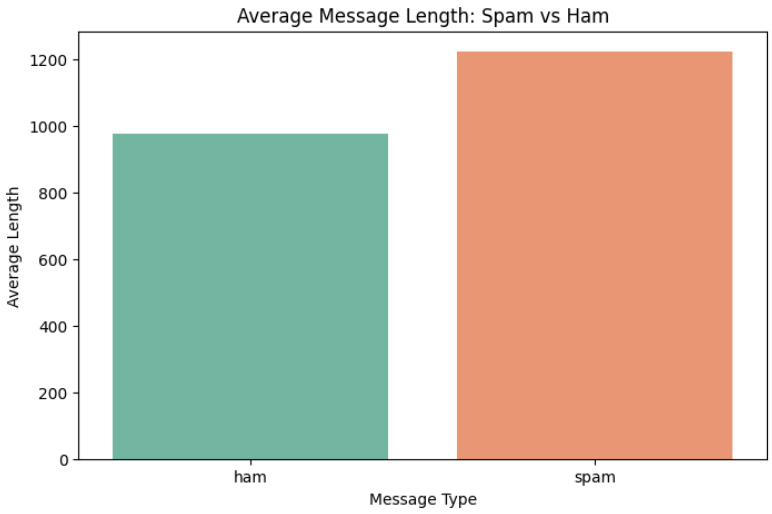
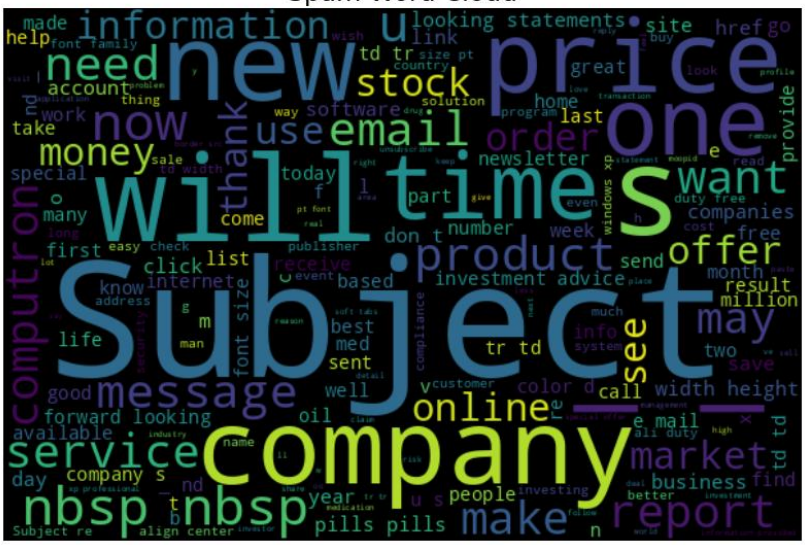
Preprocessing Template

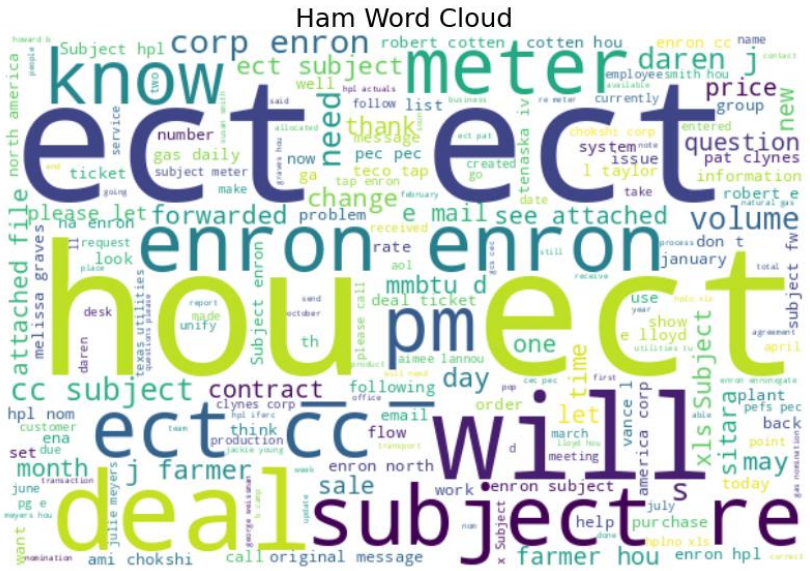
Dataset variables will be statistically analyzed to identify patterns, trends, and potential anomalies in the text data. Python will be employed for preprocessing tasks such as text cleaning, tokenization, and vectorization to prepare the data for modeling. **Tfidf Vectorization** will be used to convert the text data into numerical features that can be processed by the machine learning model. Data cleaning will address noise such as special characters, links, and numbers, ensuring high-quality input for subsequent analysis and prediction, thus forming a solid foundation for accurate spam detection.

Section	Description
Data Overview	Dimension: 5171 Rows x 5 Columns Descriptive statistics:

	Unnamed: 0	label	text	label_num
0	605	ham	Subject: enron methanol ; meter # : 988291\r\n...	0
1	2349	ham	Subject: hpl nom for january 9 , 2001\r\n(see...	0
2	3624	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...	0
3	4685	spam	Subject: photoshop , windows , office . cheap ...	1
4	2030	ham	Subject: re : indian springs\r\nthis deal is t...	0

Univariate Analysis	 

Bivariate Analysis	
Multivariate Analysis	Spam Word Cloud 

	
Outliers and Anomalies	Addressed noise like irrelevant special characters, numbers, and excessive spaces.
Data Preprocessing Code Screenshots	
Loading Data	`data=pd.read_csv("/content/drive/MyDrive/spam_ham_dataset.csv",encoding="latin1")` `# Assuming 'data' is the dataset we inspected earlier` `df = data.copy()` `#checking the head of dataset` `df.head()`

Handling Missing Data	```# Dropping unnecessary index column df.drop(columns=['Unnamed: 0'], inplace=True) # Checking for any missing data print(df.isnull().sum())``` ```label 0 text 0 label_num 0 dtype: int64```
Data Transformation	```# Function to clean the text def clean_text(text): text = text.lower() # convert to lowercase text = re.sub('\[.*?\]', '', text) # remove text in square brackets text = re.sub('https?://\S+|www\.\S+', '', text) # remove links text = re.sub('<.*?>+', '', text) # remove HTML tags text = re.sub('[%s]' % re.escape(string.punctuation), '', text) # remove punctuation text = re.sub('\n', '', text) # remove newline characters text = re.sub('\w*\d\w*', '', text) # remove words containing numbers return text # Apply the cleaning function to the 'text' column df['cleaned_text'] = df['text'].apply(clean_text) # Display the cleaned text df[['text', 'cleaned_text']].head()```
Feature Engineering	Attached the code in the final submission.
Save Processed Data	-