KGISL INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SPAM DETECTION

INTRODUCTION

Spam is a significant threat to the stability and security of these platforms in the digital age, as electronic communication has grown widespread. Traditional rule-based techniques to spam detection frequently fall short of adjusting to spammers' dynamic tactics. Text cleaning, tokenization, and vectorization are examples of data preprocessing operations that prepare data for the construction of predictive models employing various classifiers.

ABSTRACT

Spam is a continuous danger to the integrity and security of digital communication systems, notwithstanding their exponential growth. The present research focuses on the development and improvement of spam detection algorithms using machine learning approaches. Various models and feature extraction strategies are investigated using a diversified dataset that includes both genuine and spam cases. The research goes into the complexities of data preprocessing, such as text cleaning, tokenization, and vectorization. To build prediction models, a variety of classifiers such as Nave Bayes, Random Forest, KNeighbors, and Support Vector Machines are used. Comprehensive metrics are used to measure the performance of these models, providing insights into the strengths and limitations of each strategy.

OBJECTIVE

Spam detection's major goal is to automatically recognise and filter out uninvited and potentially damaging messages from lawful communication channels. Spam detection uses powerful algorithms and machine learning approaches to discriminate between legitimate material and unwelcome messages such as phishing attempts, viruses, and ads. This procedure improves the overall security and dependability of electronic communication platforms, resulting in a safer and more simplified user experience. In the face of developing spamming strategies, the ultimate goal is to reduce the impact of spam, protect users from potential risks, and maintain the integrity of digital communication channels.

INNOVATION

The spam is detected easily by the user.

TECHNIQUES/TOOLS

- Visual Studio Code
- Machine Learning
- Python

MODULES

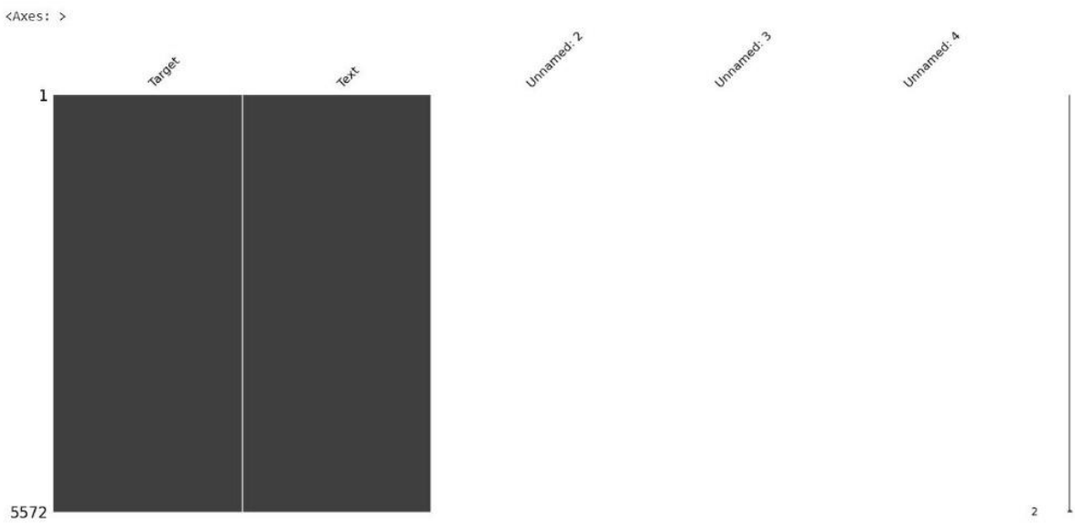
i. Data Exploration
ii. Data Preprocessing
iii. Vectorization
iv. Setting of model

DATA EXPLORATION

Explores the data to understand its structure and characteristics.

i. **Features Engineering** : Enhance the dataset by creating new features

ii. **Outlier Detection** : Identify and address outliers in the data



	Target	Text
0	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives around here though
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u. U have won the £750 Pound prize. 2 claim is easy, call 087187272008 NOW1! Only 10p per minute. BT-national-rate.
5568	ham	Will ü b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other suggestions?
5570	ham	The guy did some bitching but I acted like i'd be interested in buying something else next week and he gave it to us for free
5571	ham	Rofl. Its true to its name

5572 rows × 2 columns

DATA PREPROCESSING

Prepare the data for analysis and modelling.

i.Cleaning Text : Remove irrelevant characters and format the text.

ii.Tokenization : Break down text into smaller units, typically words.

iii.Removing Stopwords : Eliminate common and less informative words

```
The First 5 After Tokenize Texts:
['go', 'until', 'jurong', 'point', 'crazy', 'available', 'only', 'in', 'bugis', 'n', 'great', 'world', 'la', 'e', 'buffet', 'cine', 'there', 'got
['ok', 'lar', 'joking', 'wif', 'u', 'oni']
['free', 'entry', 'in', 'a', 'wkly', 'comp', 'to', 'win', 'fa', 'cup', 'final', 'tkts', 'st', 'may', 'text', 'fa', 'to', 'to', 'receive', 'entry'
['u', 'dun', 'say', 'so', 'early', 'hor', 'u', 'c', 'already', 'then', 'say']
['nah', 'i', 'don', 't', 'think', 'he', 'goes', 'to', 'usf', 'he', 'lives', 'around', 'here', 'though']
['freemsg', 'hey', 'there', 'darling', 'it', 's', 'been', 'week', 's', 'now', 'and', 'no', 'word', 'back', 'i', 'd', 'like', 'some', 'fun', 'you'
<ipython-input-18-9599fe73282a>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.loc[:,'Tokenize_text'] = df.apply(lambda row: nltk.word_tokenize(row['Clean_text']),axis=1)
```


```
The First 5 After Tokenize Texts:
['go', 'until', 'jurong', 'point', 'crazy', 'available', 'only', 'in', 'bugis', 'n', 'great', 'world', 'la', 'e', 'buffet', 'cine', 'there', 'got
['ok', 'lar', 'joking', 'wif', 'u', 'oni']
['free', 'entry', 'in', 'a', 'wkly', 'comp', 'to', 'win', 'fa', 'cup', 'final', 'tkts', 'st', 'may', 'text', 'fa', 'to', 'to', 'receive', 'entry'
['u', 'dun', 'say', 'so', 'early', 'hor', 'u', 'c', 'already', 'then', 'say']
['nah', 'i', 'don', 't', 'think', 'he', 'goes', 'to', 'usf', 'he', 'lives', 'around', 'here', 'though']
['freemsg', 'hey', 'there', 'darling', 'it', 's', 'been', 'week', 's', 'now', 'and', 'no', 'word', 'back', 'i', 'd', 'like', 'some', 'fun', 'you'
<ipython-input-18-9599fe73282a>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.loc[:,'Tokenize_text'] = df.apply(lambda row: nltk.word_tokenize(row['Clean_text']),axis=1)
```


VECTORIZATION

Convert the processed text data into numerical vectors using techniques like TF-IDF.

```
The First 5 lines in corpus :
go jurong point crazy available bugis n great world la e buffet cine get amore wat
ok lar joke wif u oni
free entry wkly comp win fa cup final tkts st may text fa receive entry question std txt rate c apply
u dun say early hor u c already say
nah think go usf live around though
```

SETTING OF MODEL

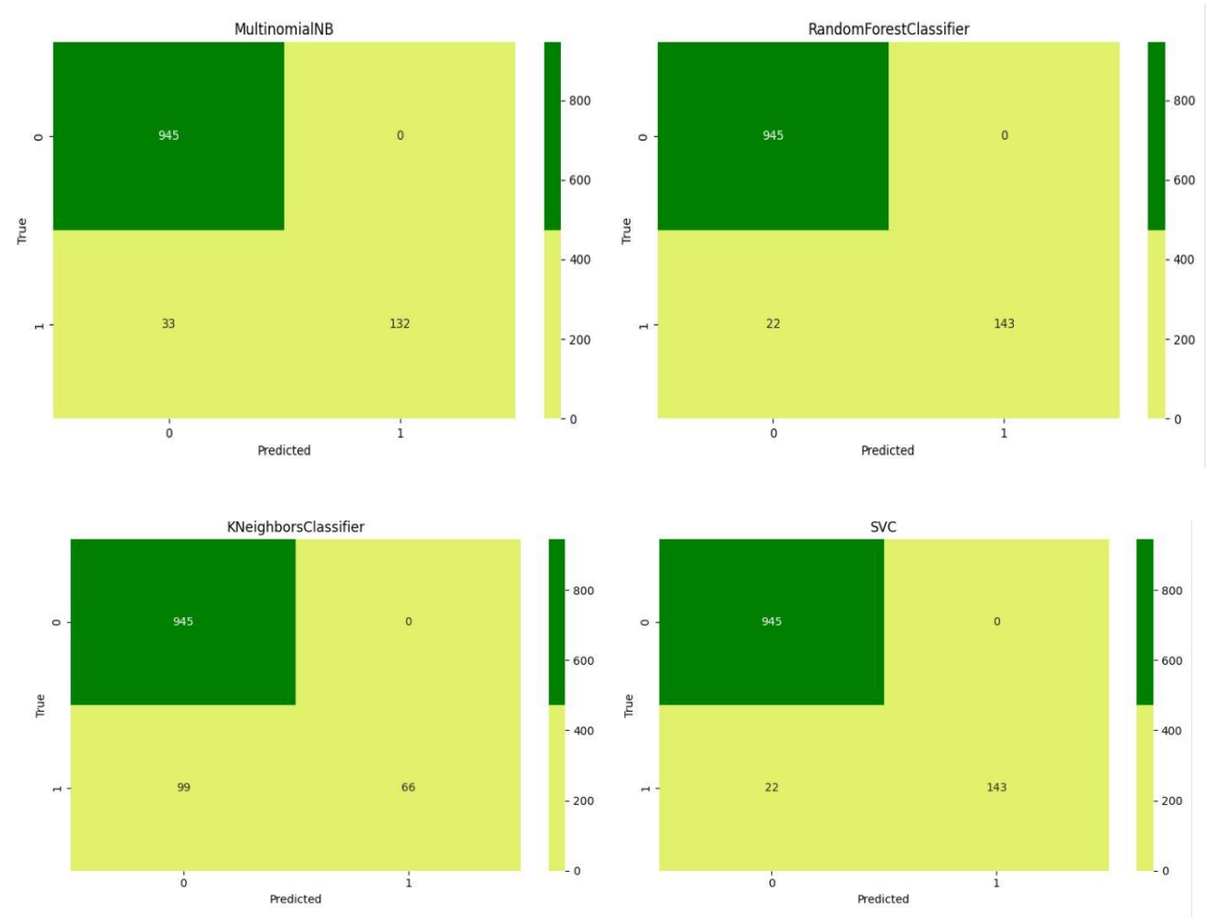
Model Building:

Construct models to predict or classify based on the features.

Utilize various classifiers like Naïve Bayes, RandomForest, KNeighbors and Support Vector Machines(SVM)

Evaluating models:

Assess the performance of the models using appropriate metrics and techniques



	Precision	Recall	F1score	Accuracy on Testset	Accuracy on Trainset
NaiveBayes	1.000000	0.800000	0.888889	0.970270	0.973856
RandomForest	1.000000	0.866667	0.928571	0.980180	1.000000
KNeighbours	1.000000	0.400000	0.571429	0.910811	0.926076
SVC	1.000000	0.866667	0.928571	0.980180	0.997070

PROJECT LINK

GOOGLE COLAB- https://shorturl.at/eglX6

VIDEO LINK - https://shorturl.at/lxNR3

CONCLUSION

Finally, this study emphasises the significance of machine learning in strengthening spam detection approaches. We discovered the potential of varied classifiers to distinguish between genuine and harmful content by digging into the complexities of data exploration, preprocessing, and model creation. The evaluation measures provide a more detailed understanding of each model's strengths and flaws, directing future improvements in the field.

PROJECT TEAM DETAILS

STUDENT DETAILS

SOWMIYA P (711720104092)

ROSITHA A (711720104071)

PRIYADHARSINI M (711720104064)

PAVITHRA B (711720104061)

FACULTY COORDINATOR

Ms. SUGANTHI A

ASSISTANT PROFESSOR