FLIP ROBO

Email Spam Classifier

Submitted by:

SHIVAM N. GADEKAR

ACKNOWLEDGMENT

We would like to express our heartfelt gratitude to everyone who has contributed to the successful development of our email classifier based on machine learning. This innovative solution would not have been possible without the dedication and hard work. I would also like to thank to those for providing valuable feedback and helping us improve the system. We are proud to have created a powerful tool that simplifies email management and enhances productivity for individuals and businesses alike.

INTRODUCTION

- Business Problem Framing
    a. Spam emails can pose a significant problem for businesses. Some of the common problems associated with spam emails are:
    b. Productivity Loss: Employees spend a significant amount of time sorting through and deleting spam emails, which can result in decreased productivity and reduced efficiency.
    c. Security Risks: Spam emails can contain malware, viruses, or phishing scams that can compromise the security of a business's network and sensitive data.
    d. Brand Reputation Damage: Spam emails can damage the reputation of a business by sending emails with content that is inappropriate, offensive, or spammy.
- Conceptual Background of the Domain Problem

    Text processing: Knowledge of natural language processing (NLP) techniques such as text tokenization, feature extraction, and text classification is essential to effectively analyze email content.

    Email protocols and standards: Understanding of email protocols such as SMTP, IMAP, and POP, and email standards such as MIME and HTML, is important to effectively process and classify emails.

    A good understanding of the problem of spam emails and the challenges it poses to businesses.

- Review of Literature

Email spam classification has been a popular research topic for many years, with numerous studies exploring various algorithms and techniques for detecting and filtering unwanted emails. In this review, we aim to summarize the state of the art in email spam classification and highlight the contributions of previous studies.

Studies have used a variety of approaches for email spam classification, including supervised learning algorithms such as support vector machines (SVMs) and Naive Bayes, unsupervised learning algorithms such as clustering, and deep learning algorithms such as convolutional neural networks (CNNs). Most studies have used feature extraction techniques to extract relevant information from email content, such as keywords, text length, and presence of URLs.

- ## Motivation for the Problem Undertaken

  The motivation for this email spam classifier project is rooted in the recognition that traditional anti-spam filters and security software are often insufficient to protect against the increasingly sophisticated tactics used by spammers. By leveraging the power of machine learning algorithms, this project aims to develop a more effective solution that can accurately identify and filter spam emails.

Analytical Problem Framing

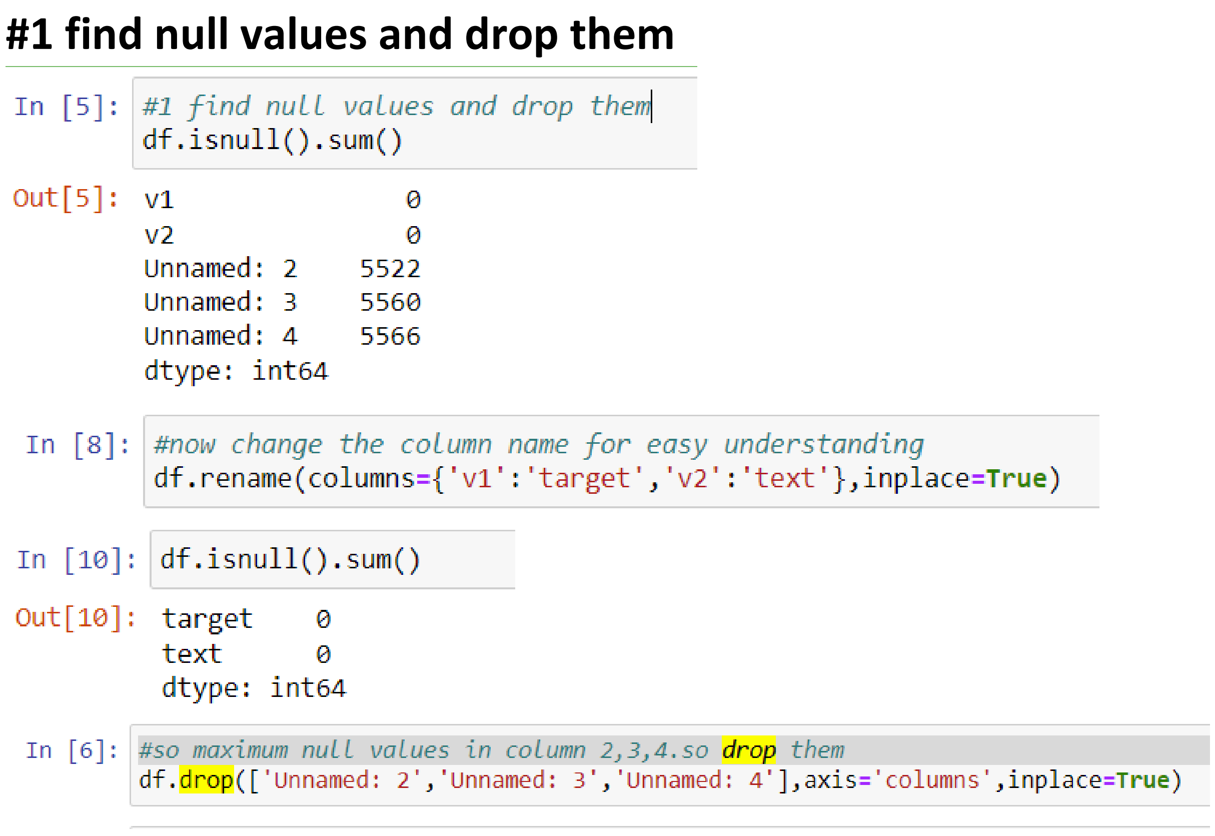
- Mathematical/ Analytical Modeling of the Problem

  For training we used three models GaussianNB(), MultinomialNB() and BernoulliNB().

- Data Sources and their formats

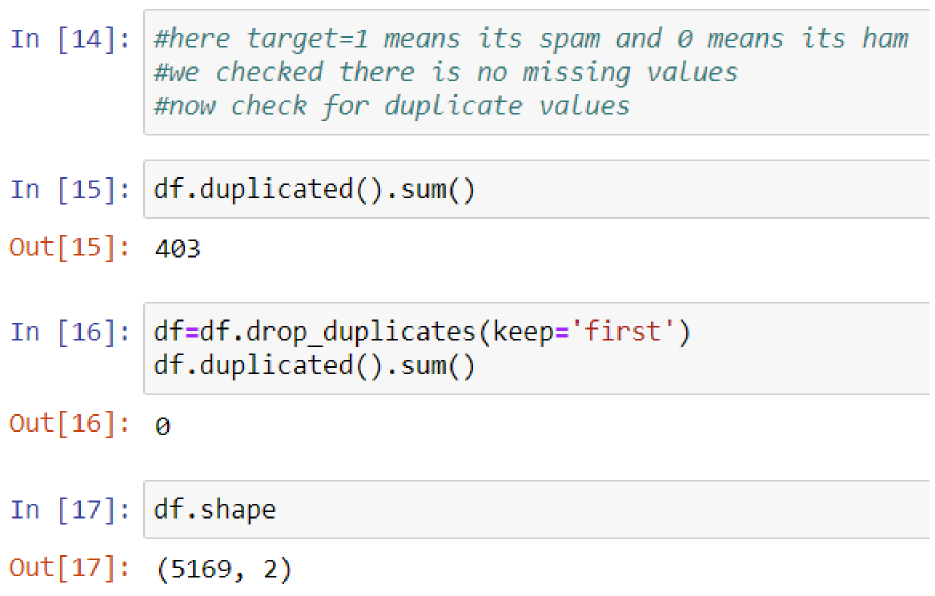
  Data available in .csv format with shape of (5572, 5).

- Data Preprocessing Done



#1 find null values and drop them

```
In [5]: #1 find null values and drop them
        df.isnull().sum()

Out[5]: v1                0
        v2                0
        Unnamed: 2     5522
        Unnamed: 3     5560
        Unnamed: 4     5566
        dtype: int64

In [8]: #now change the column name for easy understanding
        df.rename(columns={'v1':'target','v2':'text'},inplace=True)

In [10]: df.isnull().sum()

Out[10]: target    0
         text      0
         dtype: int64

In [6]: #so maximum null values in column 2,3,4.so drop them
        df.drop(['Unnamed: 2','Unnamed: 3','Unnamed: 4'],axis='columns',inplace=True)
```



```
In [14]: #here target=1 means its spam and 0 means its ham
         #we checked there is no missing values
         #now check for duplicate values

In [15]: df.duplicated().sum()

Out[15]: 403

In [16]: df=df.drop_duplicates(keep='first')
         df.duplicated().sum()

Out[16]: 0

In [17]: df.shape

Out[17]: (5169, 2)
```

- Data Inputs- Logic- Output Relationships

  Input provided was text based data with target feature categorised into two as Spam not not Spam.

- State the set of assumptions (if any) related to the problem under consideration

  1. Data is balanced
  2. Spam mails are manually classified by Human and not by machine learning algorithm.
  3. Data is true and genuine

- Hardware and Software Requirements and Tools Used

  Intel i3, 1000 GB, 6th generation and 4 GB Ram.

Model/s Development and Evaluation

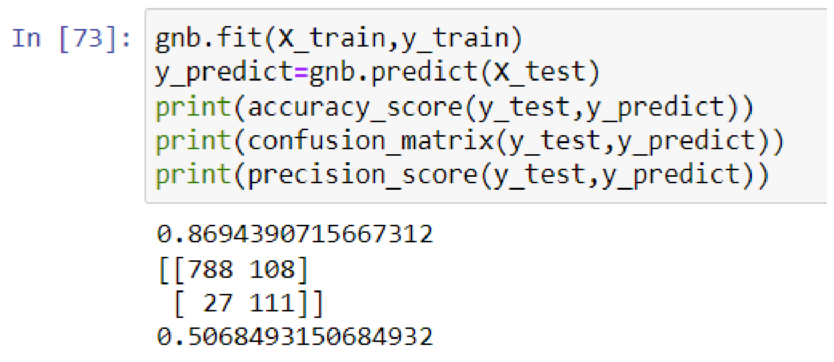
- Identification of possible problem-solving approaches (methods)

  First tried to convert data into machine understandable language and trainer on three models.

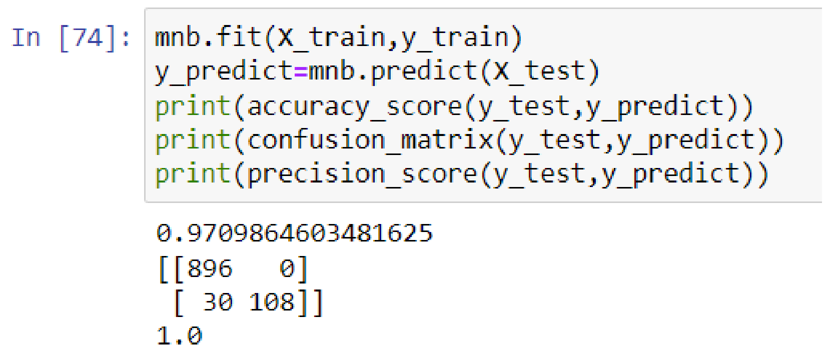
- Testing of Identified Approaches (Algorithms)

  1. GaussianNB(),
  2. MultinomialNB() and
  3. BernoulliNB().

- Run and Evaluate selected models



```
In [73]: gnb.fit(X_train,y_train)
         y_predict=gnb.predict(X_test)
         print(accuracy_score(y_test,y_predict))
         print(confusion_matrix(y_test,y_predict))
         print(precision_score(y_test,y_predict))

         0.8694390715667312
         [[788 108]
          [ 27 111]]
         0.5068493150684932
```



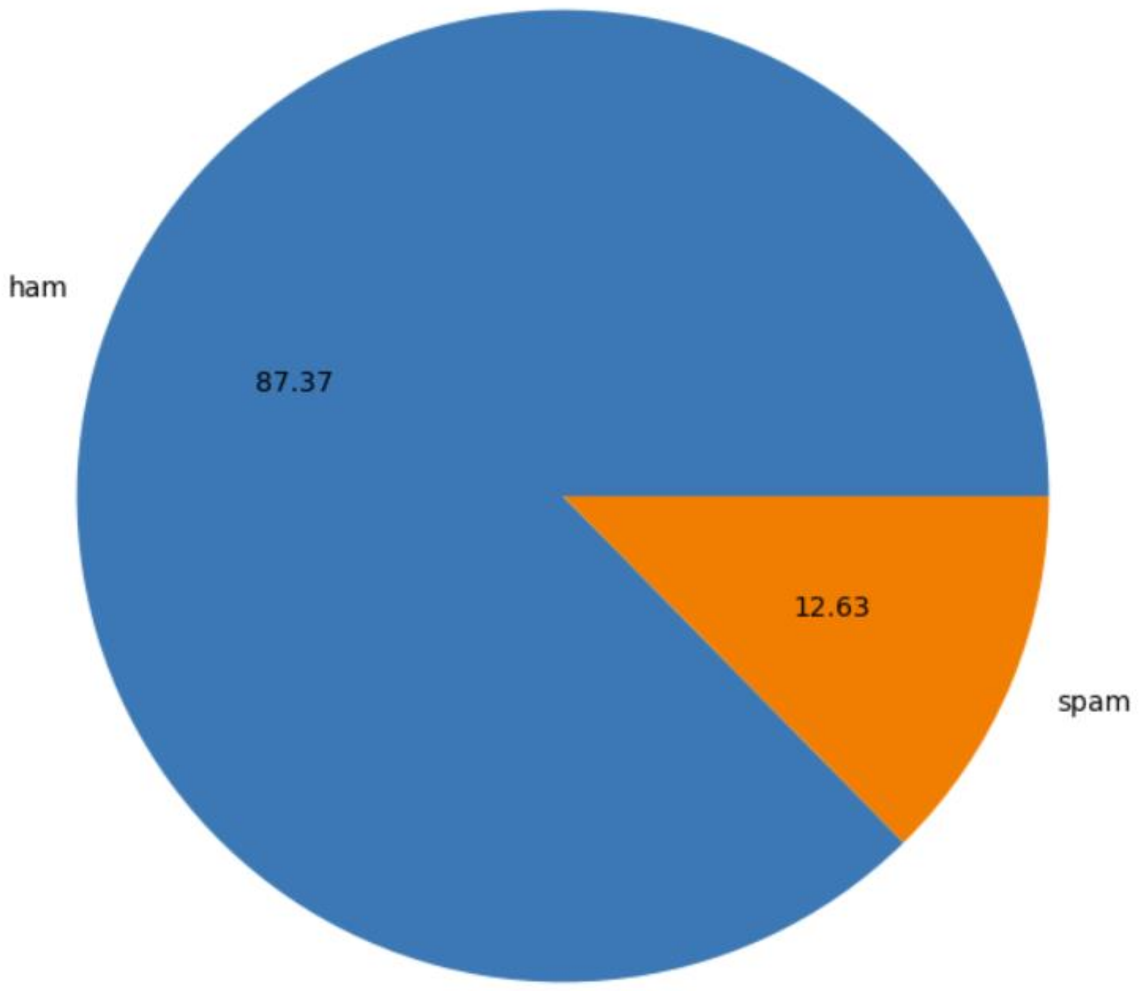
```
In [74]: mnb.fit(X_train,y_train)
         y_predict=mnb.predict(X_test)
         print(accuracy_score(y_test,y_predict))
         print(confusion_matrix(y_test,y_predict))
         print(precision_score(y_test,y_predict))

         0.9709864603481625
         [[896   0]
          [ 30 108]]
         1.0
```

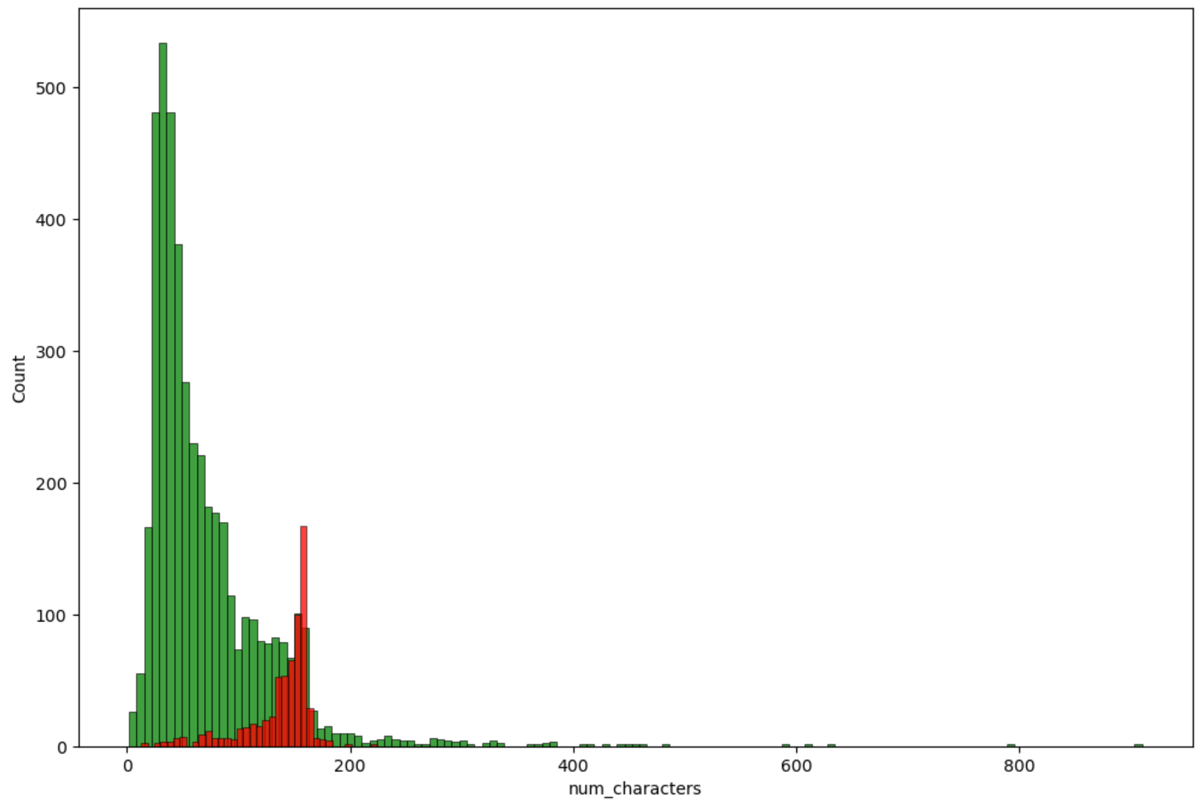
```
In [75]: bnb.fit(X_train,y_train)
         y_predict=bnb.predict(X_test)
         print(accuracy_score(y_test,y_predict))
         print(confusion_matrix(y_test,y_predict))
         print(precision_score(y_test,y_predict))

         0.9835589941972921
         [[895   1]
          [ 16 122]]
         0.991869918699187
```

- Key Metrics for success in solving problem under consideration

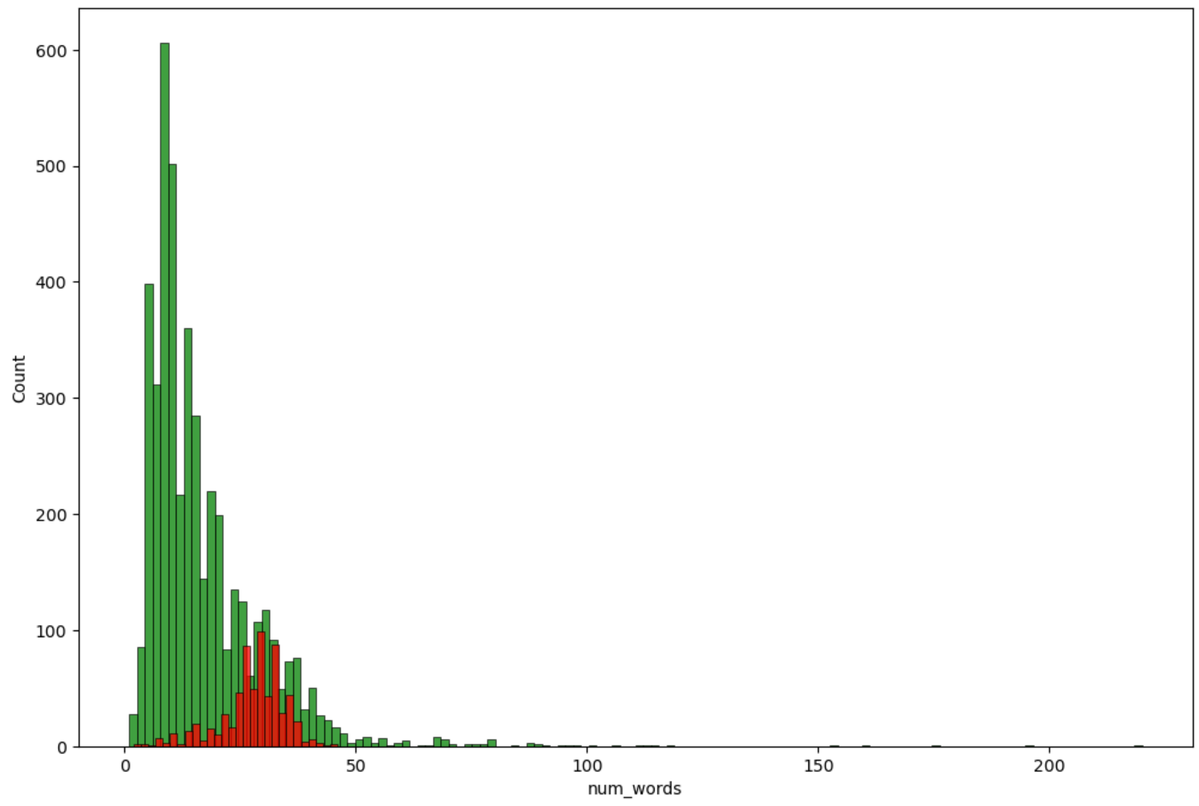
```
For  KN
Accuracy -  0.9052224371373307
Precision -  1.0
For  NB
Accuracy -  0.9709864603481625
Precision -  1.0
For  DT
Accuracy -  0.9274661508704062
Precision -  0.8118811881188119
For  LR
Accuracy -  0.9584139264990329
Precision -  0.9702970297029703
For  RF
Accuracy -  0.9748549323017408
Precision -  0.9827586206896551
For  AdaBoost
Accuracy -  0.960348162475822
Precision -  0.9292035398230089
For  BgC
Accuracy -  0.9574468085106383
Precision -  0.8671875
For  ETC
Accuracy -  0.9748549323017408
Precision -  0.9745762711864406
For  GBDT
Accuracy -  0.9477756286266924
Precision -  0.92
For  xgb
Accuracy -  0.971953578336557
Precision -  0.943089430894309
```

- Visualizations..

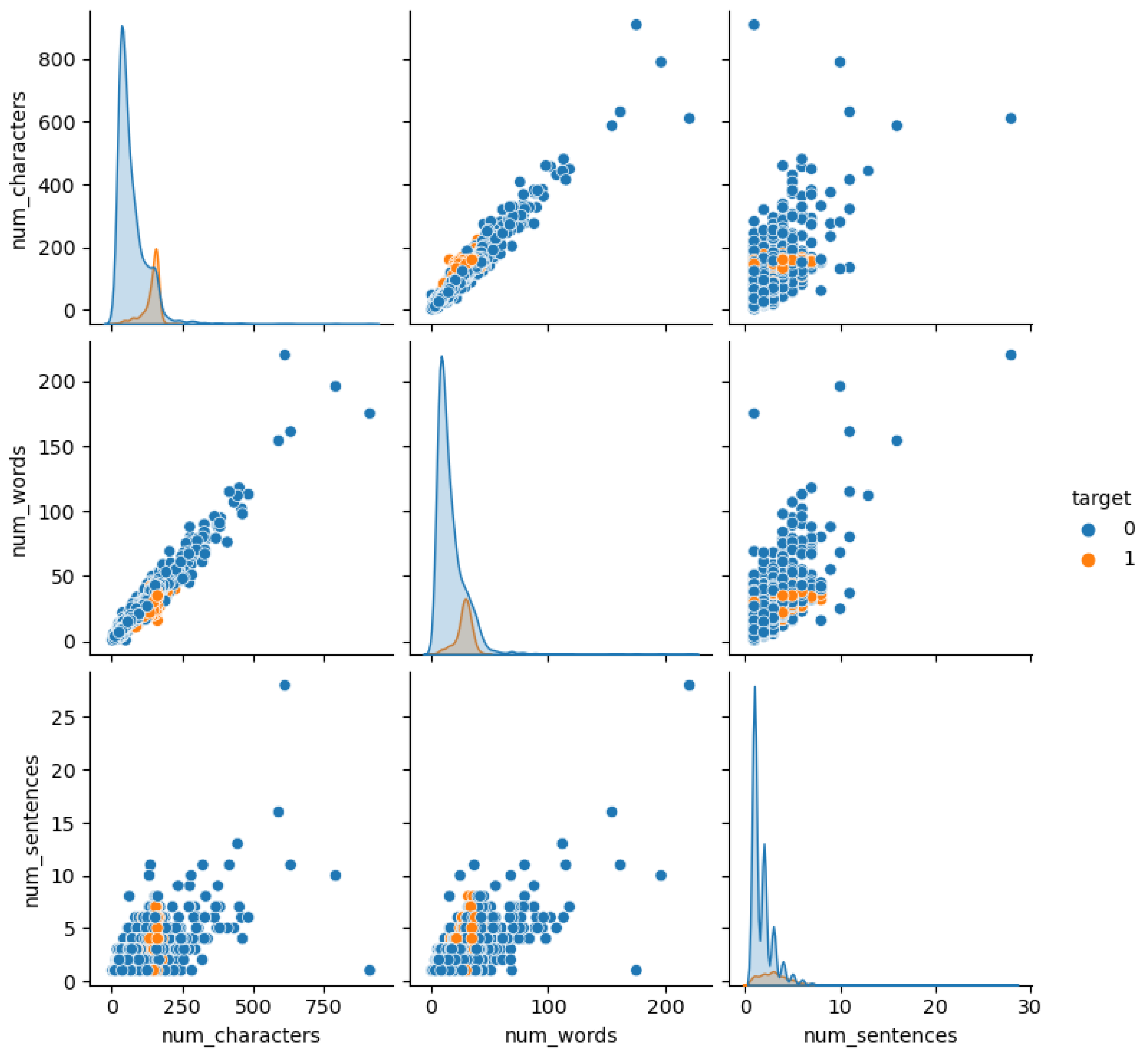




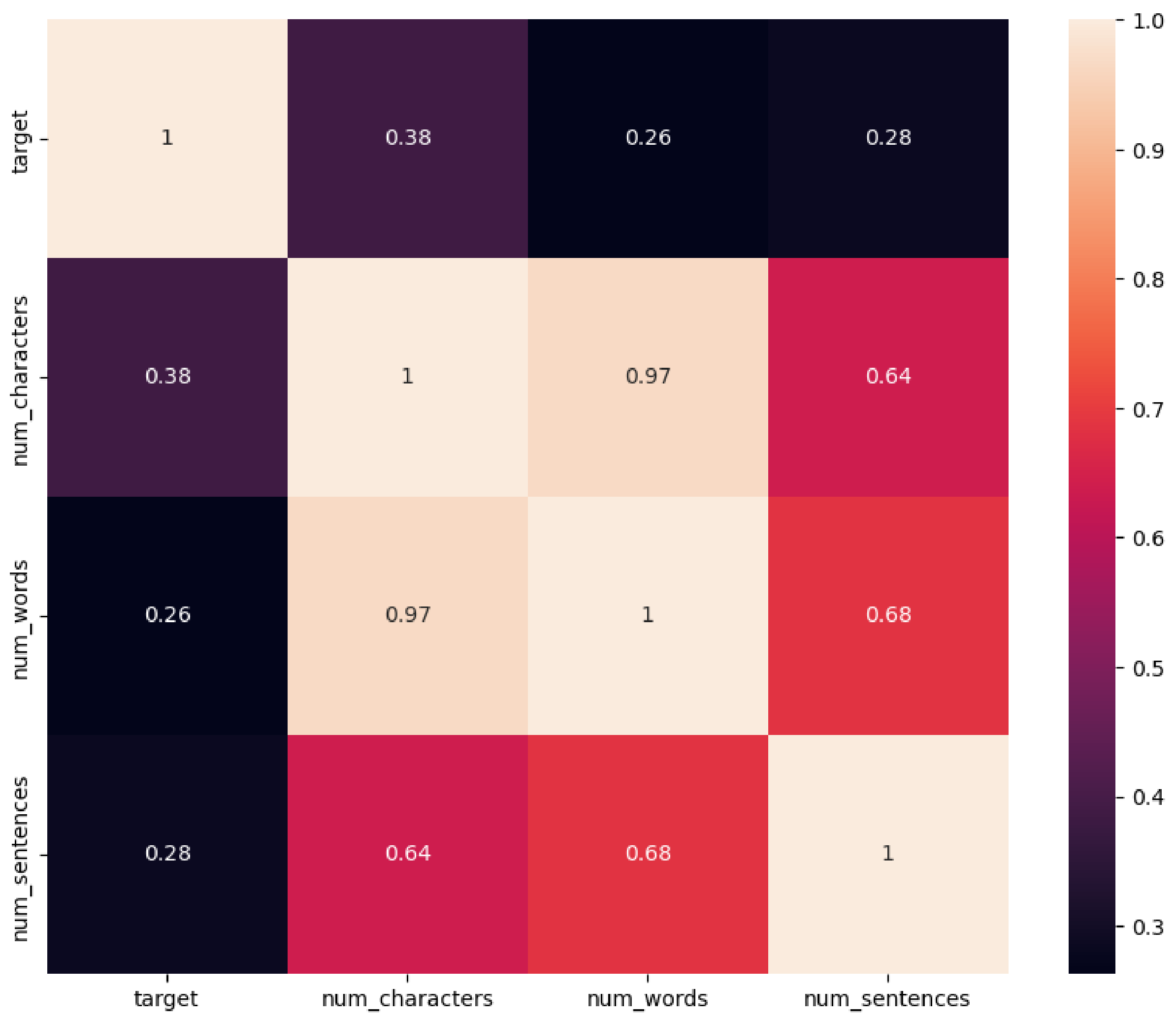
Count
500
400
300
200
100
0
0
200
400
600
800
num_characters

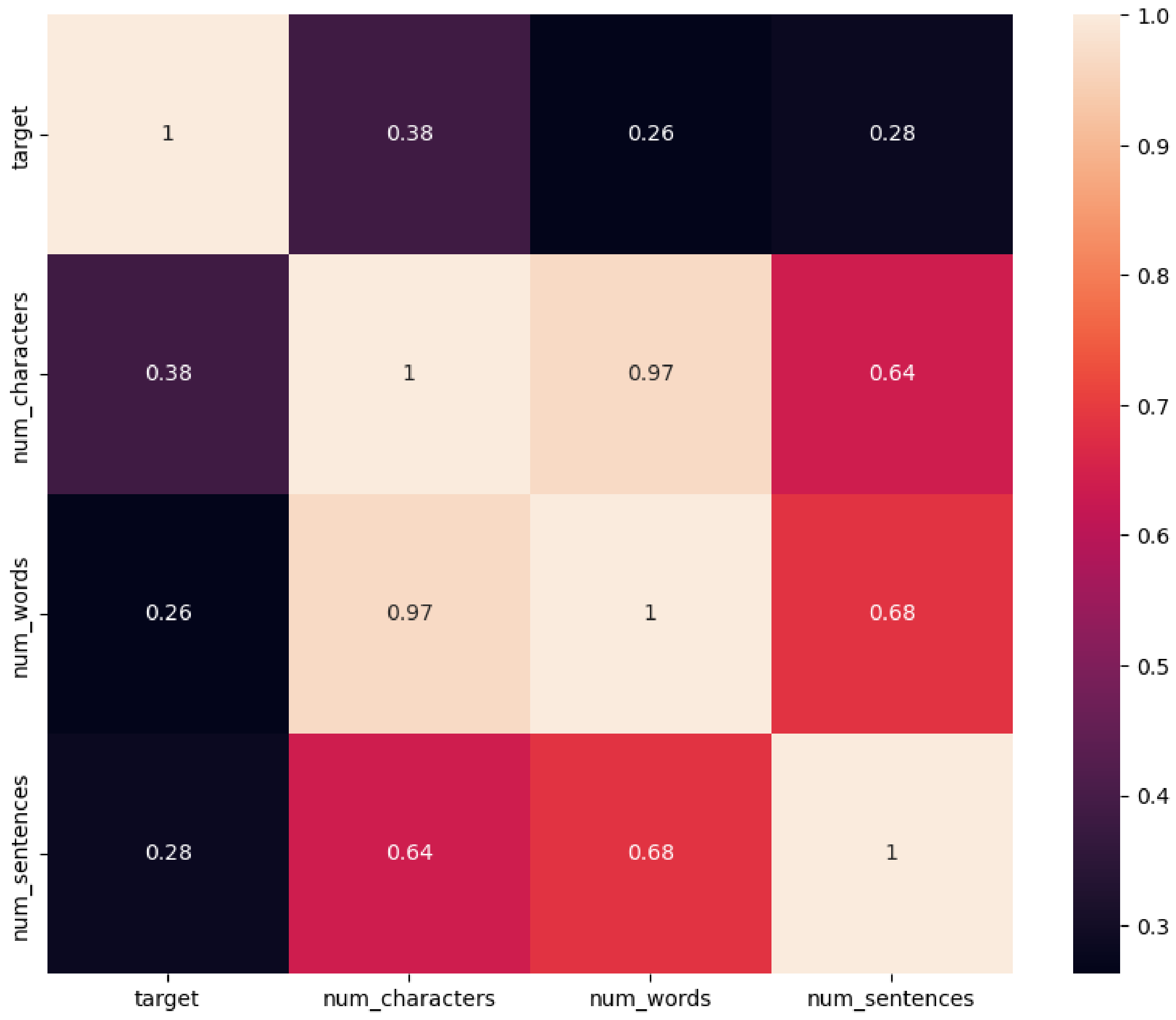


Count
600
500
400
300
200
100
0
0
50
100
150
200
num_words

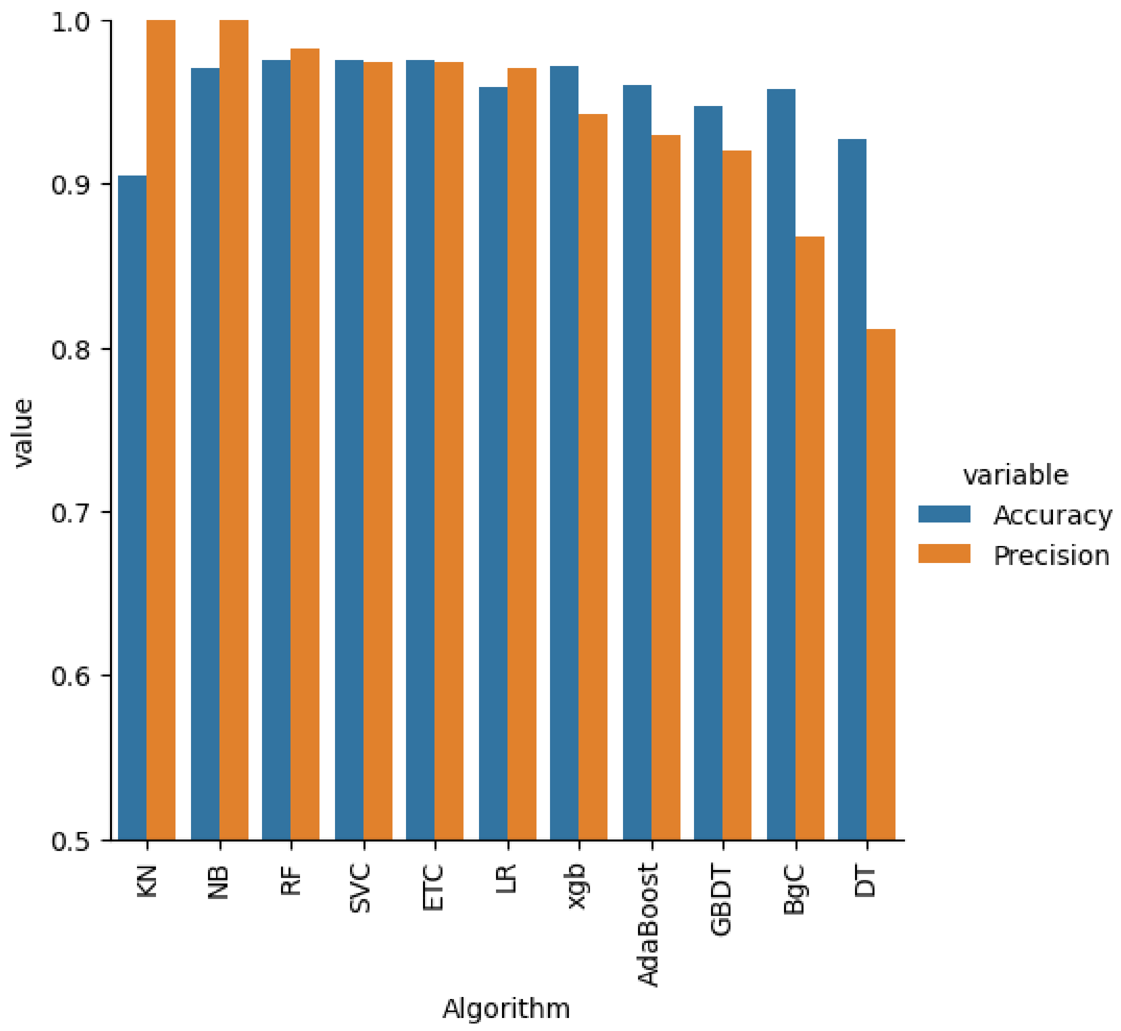


num_characters
num_words
num_sentences
target
0
1


target
num_characters
num_words
num_sentences
1
0.38
0.26
0.28
0.38
1
0.97
0.64
0.26
0.97
1
0.68
0.28
0.64
0.68
1
target
num_characters
num_words
num_sentences
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3



- Interpretation of the Results


value
1.0
0.9
0.8
0.7
0.6
0.5
KN
NB
RF
SVC
ETC
LR
xgb
AdaBoost
GBDT
BgC
DT
Algorithm
variable
Accuracy
Precision

CONCLUSION

- ### Key Findings and Conclusions of the Study

There are some words that email spammer usually use, need to work and filter those words.

- ### Learning Outcomes of the Study in respect of Data Science

The email spam classifier project provides several important learning outcomes for individuals interested in data science. These include:

Understanding of Machine Learning Algorithms: By building a model for email spam classification, individuals will gain a deeper understanding of machine learning algorithms and how they can be applied to real-world problems.

Feature Selection and Engineering: This project provides an opportunity to explore and apply various feature selection techniques, including text processing and feature scaling, that can improve the accuracy of the model.

Data Preprocessing: The process of preparing and cleaning the data is a critical step in the modeling process, and this project provides hands-on experience in working with real-world datasets.

- Limitations of this work and Scope for Future Work

  Despite its usefulness, the email spam classifier model has several limitations that need to be considered in order to improve its accuracy and effectiveness. These limitations include:

  Data Quality: The quality of the data used in the model can greatly impact its performance. For example, if the training data is not representative of the real-world distribution of spam and non-spam emails, the model may not perform well on new data.

  Feature Relevance: The choice of features used in the model can greatly impact its performance. For example, if the model is trained on irrelevant or misleading features, its accuracy may be low.

  Model Over-fitting: Over-fitting occurs when the model becomes too complex and fits the training data too closely, causing poor performance on new data. This can be mitigated by using regularization techniques or by using a more appropriate model.

  Changing Email Trends: The spam landscape is constantly changing, with new tactics and techniques being used by spammers. This means that the model must be regularly updated to account for these changes in order to maintain its accuracy.