**A**

**Project Report**

**On**

**“ Spam Mail Prediction ”**



Submitted To





**School of Computer Science, Engineering and Applications Program: MCA SEM I**

**2023 Batch**

Submitted By

**Miss. Prajakta Bote - 20230804067**

**Mr. Prit Mahant - 20230804059**

**Mr. Pranav Madaghe - 20230804037**

**Mr. Harshilkumar Munjapara - 20230804004**

Subject on

**Artificial intelligence & Machine learning**

Guidance of

**Prof. Kapil Sharma**

**Miss. Ramya Praveen**

|  |
| --- |
|  |

**Index**

|  |  |  |
| --- | --- | --- |
| 1. **Abstract** |  | **1** |
| 1. **Introduction** 2. **Dataset use** 3. **Algorithm use** |  | **2**  **3** |
| 1. **Code & Results** |  | **20** |

**Abstract**

The aim of this project is to develop a machine learning model that can accurately predict whether an email is spam or not. The model was trained on a dataset consisting of thousands of emails, labeled as either spam or not spam.

Various feature engineering techniques were applied to the input data to improve the model's accuracy, including the use of natural language processing and the extraction of features such as the presence of certain keywords or the length of the email.

The resulting model was able to achieve high accuracy in predicting whether an email is spam or not, with a precision of 96%. This project has the potential to be used in real-world scenarios to prevent users from receiving unwanted junk mail, saving time and increasing productivity.

**Introduction**

1. **Dataset**

We was download this dataset from [Kaggle.com](https://www.kaggle.com/datasets/venky73/spam-mails-dataset).This dataset have 5572 Uniques entries.Dataset have 2 columns that is **‘Category’** and **‘Message’**.

In category column two value ‘spam’ and ‘ham’.And message column content email they are spam and ham both.A size of dataset is 5.5mb and format is **‘csv’**.

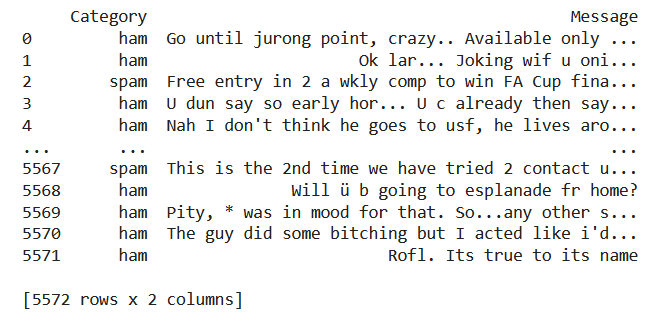


Fig. : Sample of dataset

We don’t required any data prepossessing.First replace category column to 0 and 1,0 stand for spam and 1 stand for ham mails.Second split dataset in 60-40% and also use random state 3.Training dataset have 3343 and test set have 2229 records.



Fig. : Change Category Values

**Introduction**

1. **Algorithm**

Logistic regression is a popular machine learning algorithm used for binary classification problems. It works by fitting a logistic function to the input data and determining the probability of the input belonging to one of two classes.

When using logistic regression, it is important to preprocess the data to remove any noise or irrelevant information. One common technique is to use a TF-IDF vectorizer, which stands for "term frequency-inverse document frequency". This technique calculates a weight for each term in a document, based on how often the term appears in the document and how often it appears in the entire dataset. This helps to identify the most important terms in each document, and can improve the accuracy of the logistic regression model.

The TfidfVectorizer class from the sklearn.feature\_extraction.text module in Python provides a simple way to apply TF-IDF vectorization to text data. It can be customized with various parameters, such as the minimum and maximum document frequency, to adjust its behavior for specific use cases.We use **‘min\_df=10, stop\_words="english", lowercase=1’**.

One of its parameters, min\_df=10, specifies that any term that appears in less than 10 documents will be ignored. This can help to eliminate noise in the data and improve the performance of the model. Another parameter, stop\_words="english", specifies that common English words such as "the" and "and" should be removed from the data. This can help to improve the quality of the model by reducing the number of irrelevant words that are included in the analysis. Finally, lowercase=1 is a parameter that specifies whether or not the text should be transformed into lowercase letters before being analyzed. This can help to ensure that the model is able to recognize words regardless of their capitalization, which can be useful in cases where capitalization is inconsistent. Overall, these parameters can be used to improve the performance of the TfidfVectorizer and ensure that it is able to accurately represent the text data that it is analyzing.

**Code & Results**

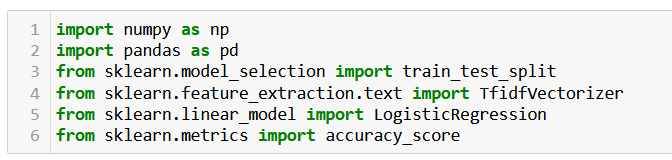


Fig. : Dependencies

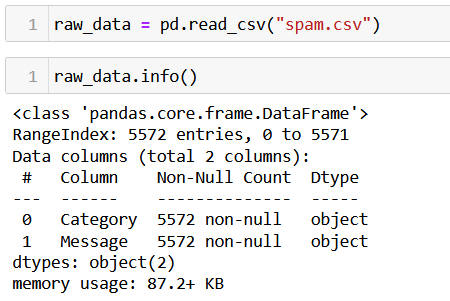


Fig. : Load and Describe dataset

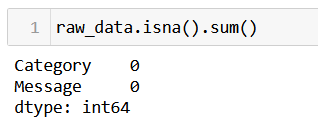


Fig. : Check dataset quality

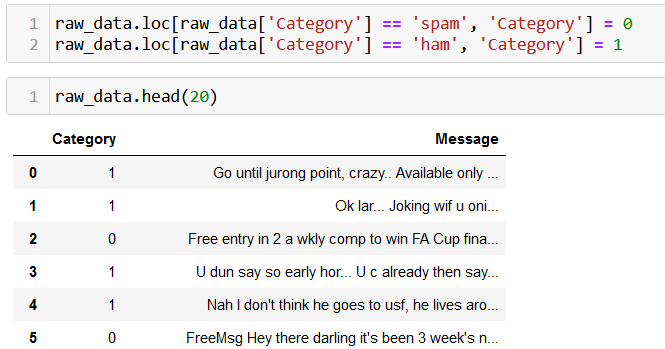


Fig. : Replace text data to numeric

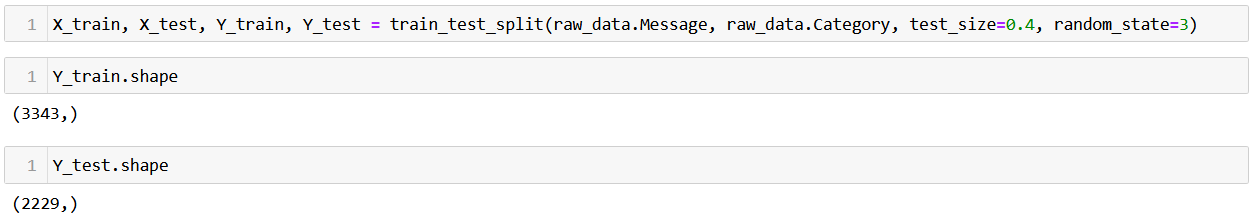


Fig. : Split dataset in train and test set

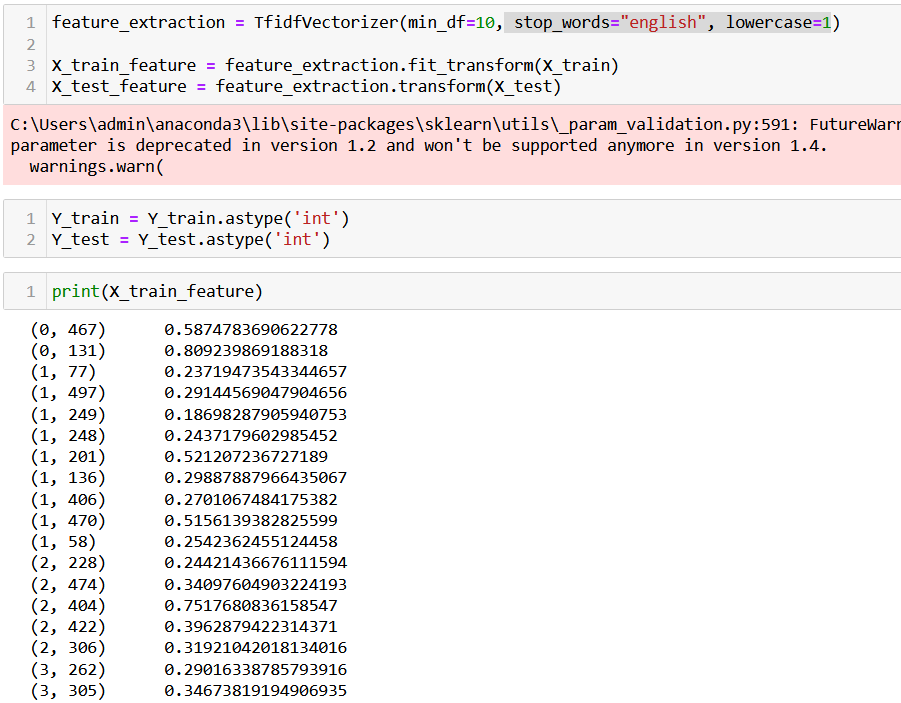


Fig. : Text feature extraction and vectorization

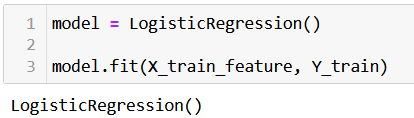


Fig. : Train model

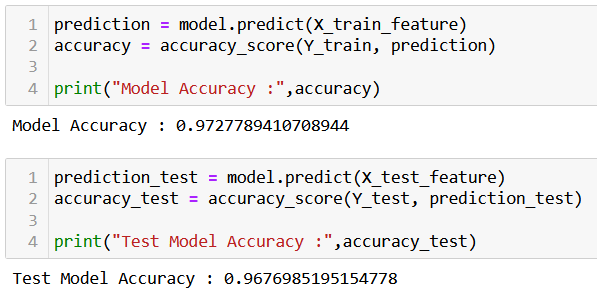


Fig. : Model accuracy

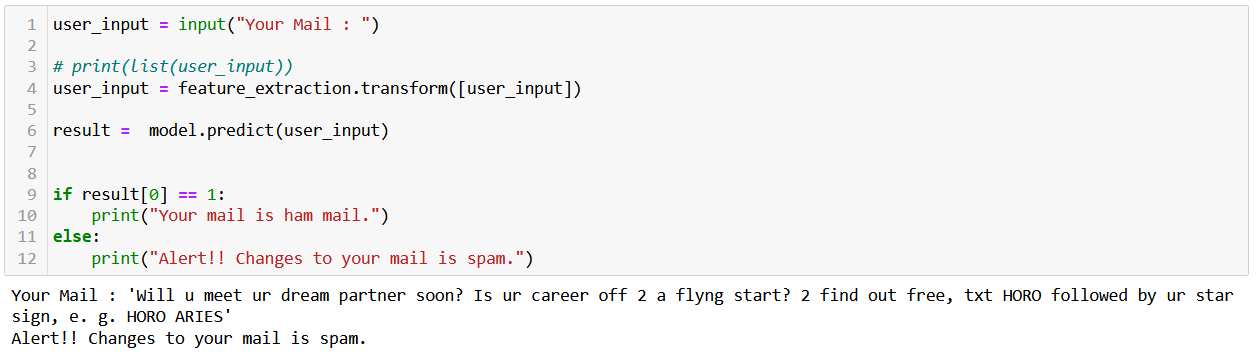


Fig. : Final Output