Spam Classifier in Machine Learning

1 Problem Statement

Every day we get dozens of email notifications and most of them are spam. You are probably familiar with what spam emails are already; spam email filtering is an essential feature for email services such as Gmail, Yahoo Mail, and Outlook. Spam emails can be annoying for users, but they bring more issues and risks with them. For example, a spam email can be designed to solicit credit card numbers or bank account information, which can be used for credit card fraud or money laundering. A spam email can also be used to obtain personal data, such as a social security number or user IDs and passwords, which then can be used for identity theft and various other crimes.

2 Project Objective

The main objective for this project we will build a spam classifier through machine learning models in which we will check the prediction through different models and check the performance metrics of each machine learning models.

Models that we have used in this project are given below:

- Random Forest
- Support Vector Machine (SVM)
- Multinomial Naive Bayes
- Logistic Regression

3 Dataset

All experiments are performing on the following dataset provided by UCI Machine Learning:

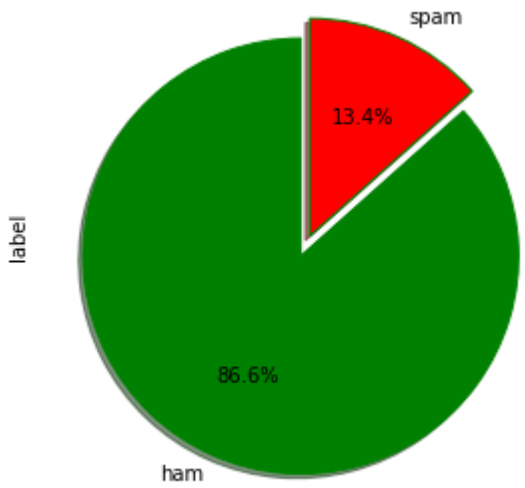
1. Spam Email Data Set: Dataset is about to spam email text and label. It will used for supervised binary classification tasks. Total numbers of instances in the data file is 5572 with 2 attributes. Label attribute in which spam and ham email as a target.

4 Exploratory Data Analysis

Exploratory data analysis is performed to gain different useful information and hidden insights from dataset. In this section different statistical techniques have been used to gain insights and then being visualized into appropriate charts and plots.

4.1 Target Distribution

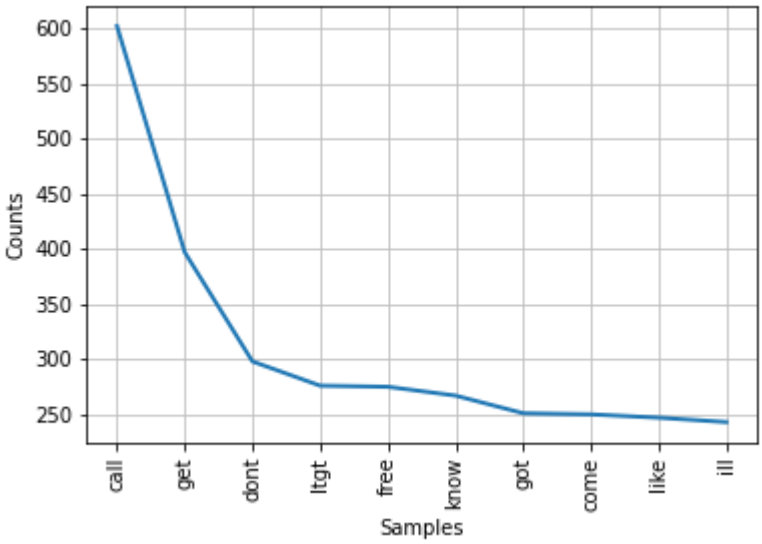
This analysis is about the distribution of target feature. The bar chart representing this information is shown below.



From above chart it is clear that in most of the email are not spam. The percentage of spam is 86.6% and denied is 13.4%.

4.2 Email Text Distribution

This analysis is about the distribution of email text. The bar chart representing this information is shown below.



From above chart it is clear that in most word that used in an email text is call and then get.

5 Data Preprocessing

We preprocess the text in any classifier, we just need to complete some primary operations before. First, we have to preprocess our whole data. The preprocessing of data contains the following steps.

Step-1: remove stop words: Removing stop words means we have to remove helping verbs, like, is, am, and are.

Step-2: Lowercase alphabets: Lower case alphabets mean we have to convert all upper-case alphabets into the lower case, for instance, „WORLD" change into the world.

Step-3: Remove punctuation; Removing punctuation means we have to remove all the punctuation's characters from our text like we have to remove comma, parenthesis, examination sign, underscore sign, hash tag etc.

Step-4: Feature Extraction: The last important step is to get the features. so, we will use the Tfidf-Vectorizer method from sklearn library.

6 Splitting data

Every dataset for Machine Learning model must be split into two separate sets – training set and test set. Usually, the dataset is split into 70:30 ratio or 80:20 ratio. This means that you take either 70% or 80% of the data for training the model while leaving out the rest 30% or 20%. The splitting process varies according to the shape and size of the dataset in question. We split the 80% data for training and 20% for testing.

We split the dataset 80% for training and 20% for testing and we can see the shape of training and testing as shown in fig below:

```
#split data into training and testing set
x_train, x_test, y_train, y_test = train_test_split(df['preprocess_text'], df['label'], test_size=0.2, random_state=0)
```

7 Evaluation Metrics

State of the art evaluation metrics for supervised binary classification problems are given below:

- **Confusion Matrix**

In machine learning algorithm, confusion matrix is a performance measurement. A confusion matrix is a summarized table of the number of correct and incorrect predictions (or actual and predicted values).

	Actual Positive	***Actual Negative***
Predicted Positive	➢ **TP**	➢ **FP**
Predicted Negative	➢ **FN**	➢ **TN**

$$where\ TP = True\ Positive$$
$$FP = False\ Positive$$
$$TN = True\ Negative$$

$$FN = False\ Negetive$$

True Positive (TP): where the model correctly predicts the positive class.
True Negative (TN): where the model correctly predicts the negative class.
False Positive (FP): where the model incorrectly predicts the positive class.
False Negative (FN): where the model incorrectly predicts the negative class.

- **Accuracy**

Accuracy represents the correctly predict both the positives and negatives out of all the predictions.

Measure to evaluate how accurate model's performance is:

$$\frac{TP + TN}{TP + FP + FN + FP}$$

- **Precision**

Precision represents the correctly predict the positives out of all the positive prediction it made.

Measure to evaluate how accurate model's performance is:

$$\frac{TP}{TP + FP}$$

- **Recall**

Recall represents the correctly predict the positives out of actual positives.

Measure to evaluate how accurate model's performance is:

$$\frac{TP}{TP + FN}$$

- **F_1**

F1 is a combination of both precision and recall.

Provides information of both sides TN and TP

$$2 * \frac{Precision * Recall}{Precision + Recall}$$

8 Model Performance Summary

This section involves into the performance of all machine learning models and conducts a systematic comparative analysis to determine which model is the best. Table depicts the result of all models on testing data. It shows that SVM attains the highest Accuracy on testing data and also high other metrics as compare to others.

Metrics	Logistic Regression	SVM	MultinomialNB	Random Forest
accuracy	0.960538	0.98296	0.965919	0.980269
precision	0.970561	0.990246	0.980866	0.988741
recall	0.867703	0.940625	0.88125	0.93125
f1-score	0.910147	0.963514	0.922871	0.957394
time detection	0.339	0.293	0.284	0.308

9 Conclusion

Firstly we preprocess the text before fit to the model and then check the results of four models and split the data file for 80% training and 20% testing, check all the models on testing data. SVM giving best results as compare to others models. So, we choose a SVM model for final prediction.