PROJECT REPORT ON
EMAIL SPAM CLASSIFIER

Submitted by

AKASH PANDEY

Introduction :-

Business Problem Framing: -

Spam Detector is used to detect unwanted, malicious and virus infected texts and helps to separate them from the nonspam texts. It uses a binary type of classification containing the labels such as '**ham'** (nonspam) and **spam**. Application of this can be seen in Google Mail (GMAIL) where it segregates the spam emails in order to prevent them from getting into the user's inbox.

Conceptual Background of the Domain Problem:

This is a UK forum in which cell phone users make public claims about SMS spam messages, most of them without reporting the very spam message received. The identification of the text of spam messages in the claims is a very hard and time-consuming task, and it involved carefully scanning hundreds of web pages.

Review of Literature:

A collection of 5573 rows SMS spam messages was manually extracted from the Grumbletext Web site. The files contain one message per line. Each line is composed by two columns: v1 contains the label (ham or spam) and v2 contains the raw text.

Motivation for the Problem Undertaken:

Our goal is to build a SMS spam classifier which can used to classify ham and spam so that it can be controlled and minimizes the spam SMS. So, we fit the supervised machine learning model and predict the test data.

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem:

We use Statistical techniques and analytics modeling in our projects, such as:

- describe() : use to calculate the statistical values that are mean, standard deviation, quantile deviation, minimum and maximum

values. ○ corr(): use to calculate the relation between feature variable with the target variable.

- skew(): use to check whether the skewness is present in the continuous data or not.

Data Source and their formats:

The data set of the Email Spam Classifier as show in the fig:

```
data=pd.read_excel('C:\\Users\\Dell\\Desktop\\spam_dataset.xlsx')
data.head()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

There are 5572 rows and 5 columns. ‘v1’ column is our target variable and others are feature variable. In three of the columns there is shown NaN.

Data Pre-processing

There are 99% of null values in the three columns, so we drop them. Add one new column having length of the SMS. Also remove the duplicate rows from the dataset. Apply feature engineering technique on v2. Convert object data into numerical data.

Data inputs-Logic-Output Relationships

The ‘v1’ column is our target variable. In this problem v2 column have a poor correlation with the v1 whereas v2_length is better correlated with the v1.

Hardware and Software Requirements and Tools used

Hardware:

- Memory 16GB minimum
- Hard Drive SSD is preferred 500GB
- Processor intel i5 minimum

- Operating system Windows 10

Software:

- Jupyter notebook (Python)

Libraries: pandas (used to create the data and read the data) numpy (used with the mathematical function) seaborn (used to create a different types of graphs) matplotlib (used to plot the graph) stopwords (used to remove the unnecessary words) train_test_split (used to split the data into train and test data) accuracy_score (used to calculate accuracy score for train and test) classification_report (to display precision, f1 score) confusion_matrix (form the matrix) roc_curve (used to plot the area under curve)

Model/s Development and Evaluation

Identification of possible problem:

We approach to both statistical and analytical problem

- Plot a bar graph for nominal data and distribution graph for continuous data
- describe () use to calculate mean, standard deviation, minimum, maximum and quantile deviation
- corr() used to calculate the correlation of input variable with the output variable.
- skew() used to calculate the skewness of the data

❖ plot a boxplot to check an ouliers

Testing of Identified Approaches:

Here we work on the classification problem so the machine learning models are:

- Logistic Regression
- K Neighbors Classification
- Random Forest Classification
- Classification

Run and Evaluate selected models:

➢ Logistic Regression

```
x_train,x_test,y_train,y_test = train_test_split(x,y, test_size=0.30,random_state=15)
```

```
lr.fit(x_train,y_train)
y_pred1 = lr.predict(x_test)
accuracy = accuracy_score(y_test,y_pred1)*100
print("accuracy score:",accuracy)
```

```
accuracy score: 87.13259307642063
```

```
cm= confusion_matrix(y_test,y_pred1)
print(cm)
```

```
[[1289   40]
 [ 157   45]]
```

```
clr=classification_report(y_test,y_pred1)
print(clr)
```

```
              precision    recall  f1-score   support

         0.0       0.89      0.97      0.93      1329
         1.0       0.53      0.22      0.31       202

    accuracy                           0.87      1531
   macro avg       0.71      0.60      0.62      1531
weighted avg       0.84      0.87      0.85      1531
```

The logistic regression of precision is 89, recall is 97 and f1-score is 93. The sum of true negative and false negative is 197 and the area under the curve is 59.

➢ K Neighbors Classification

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.30,random_state=7)
```

```
knn.fit(x_train,y_train)
y_pred2 = knn.predict(x_test)
accuracy = accuracy_score(y_test,y_pred2)*100
print("accuracy score:",accuracy)
```

```
accuracy score: 93.59895493141738
```

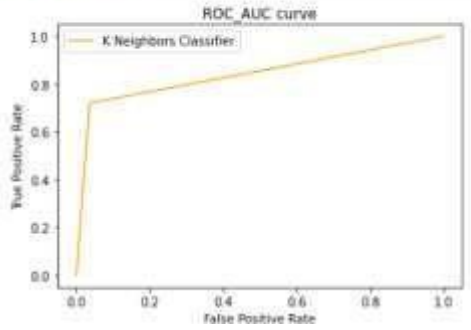
```
cm= confusion_matrix(y_test,y_pred2)
print(cm)
```

```
[[1307   49]
 [  49  126]]
```

```
clr=classification_report(y_test,y_pred2)
print(clr)
```

```
              precision    recall  f1-score   support

         0.0       0.96      0.96      0.96      1356
         1.0       0.72      0.72      0.72       175

    accuracy                           0.94      1531
   macro avg       0.84      0.84      0.84      1531
weighted avg       0.94      0.94      0.94      1531
```

```
fpr,tpr,thresholds = roc_curve(y_test,y_pred2)

plt.plot(fpr,tpr,color='orange',label='K Neighbors Classifier')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC_AUC curve')
plt.legend()
plt.show()

auc_score = roc_auc_score(y_test,y_pred2)*100
print("AUC_score",auc_score)
```




```
AUC_score 84.19321533923303
```

The K Neighbors classification of precision is 96, recall is 96 and f1-score is 96. The sum of true negative and false negative is 98 and the area under the curve is 84.

- Random Forest Classification

```
x_train,x_test,y_train,y_test = train_test_split(x,y, test_size=0.30,random_state=14)
```

```
rfc.fit(x_train,y_train)
y_pred3 = rfc.predict(x_test)
accuracy = accuracy_score(y_test,y_pred3)*100
print("accuracy score:",accuracy)
```

```
accuracy score: 93.20705421293273
```

```
cm= confusion_matrix(y_test,y_pred3)
print(cm)
```

```
[[1304   47]
 [  57  123]]
```

```
clr=classification_report(y_test,y_pred3)
print(clr)
```

```
              precision    recall  f1-score   support

         0.0       0.96      0.97      0.96      1351
         1.0       0.72      0.68      0.70       180

    accuracy                           0.93      1531
   macro avg       0.84      0.82      0.83      1531
weighted avg       0.93      0.93      0.93      1531
```

The Random Forest classification of precision is 96, recall is 97 and f1-score is 96. The sum of true negative and false negative is 104 and the area under the curve is 82.

- MultinomialNB Classification

```
x_train,x_test,y_train,y_test = train_test_split(x,y, test_size=0.30,random_state=22)
```

```
mb.fit(x_train,y_train)
y_pred4 = mb.predict(x_test)
accuracy = accuracy_score(y_test,y_pred4)*100
print("accuracy score:",accuracy)
```

```
accuracy score: 64.85956890920967
```

```
cm= confusion_matrix(y_test,y_pred4)
print(cm)
```

```
[[871 458]
 [ 80 122]]
```

```
clr=classification_report(y_test,y_pred4)
print(clr)
```

```
              precision    recall  f1-score   support

         0.0       0.92      0.66      0.76      1329
         1.0       0.21      0.60      0.31       202

    accuracy                           0.65      1531
   macro avg       0.56      0.63      0.54      1531
weighted avg       0.82      0.65      0.70      1531
```

The MultinomialNB classification of precision is 89, recall is 90 and f1-score is 89. The sum of true negative and false negative is 538 and the area under the curve is 62.

The **K Neighbors Classification** gives **better** precision score, recall and f1score. The total of True Negative and False Negative in the confusion matrix

is less in the same model and area under the curve is also higher for the testing data.

Visualisation:

On visualising the continuous data we see the data is right skewed and the target variable is imbalance because ham is in a large number and spam is small in number. The column v2 is of object type.

Interpretation of the Results:

On our analysis basis we go through various models and then we conclude better model on the basis of various classifications. Our data is imbalance so we do not consider accuracy score for model testing. Here we go with the precision, f1-score, confusion matrix and area under the curve. That will predict the test data on training the train data.

Conclusion

Key Findings and Conclusions of the Study:

On study that data we see that there are approx. 5000 missing values in each of the three columns so we drop them and also see duplicate data in 403 rows. The ham (nonspam) values are very high in numbers and spam are less in number. On checking the relation we see that in every columns have not that much good relation.

Learning Outcomes of the Study in respect of Data Science:

Here we first clean the data by dropping the columns from dataset whom having huge null values and the rows having duplicacy. Removing the unnecessary word from the text message. Visualise the continuous data and here we see the data is skewed. The target variable ‘v2’ which is imbalanced. In analysis we do describe the statistical values, correlation, outliers and skewness. Fit some

classification models and find the better one i.e. K Neighbors Model. Calculate confusion metrics, classification report and ROC curve.