SMS Spam Detection Using Supervised Learning Techniques

Dinesh Adhithya

September 1, 2021

Introduction

Spamming is the act of sending unsolicited messages to a large number of people. Spam e-mails and text messages have led to fraud and loss of productivity among users.The need for robust spam classification algorithms are more than ever. In this work I try to use supervised machine learning techniques for spam detection.A baseline model using Naive Bayes and performed at an accuracy of 97% , to improve performance Logistic Regression was used and performed at an accuracy of 98% and Support vector machine performed with an accuracy of 99 % . The SVM out performs the other models with best recall hence reducing number of false positives. Hence ensuring spam being wrongly labelled as not spam is reduced as this could have harmful effects on an user.

Data set Description

We will consider the SMS Spam Collection data. The SMS Spam Collection is a public data set of SMS labelled messages, which have been collected for mobile phone spam research. The data set contains 5,572 real and non-encoded messages in English, tagged according to being legitimate (marked as ham) or (spam). The data set is available in plain text format. The data can be downloaded from this link.

Methods

Imbalance Data set

The SMS spam data set has 4825 messages labelled as "ham" ie. not spam and 747 messages labelled as "spam". Although we want our data set to have data points of both classes in equal numbers , but reducing the number of data points to 1494 data points makes our model not see a lot of the vocabulary hence we use the imbalance data set with keeping an eye on spam class's recall metric to reduce false positives.

Text Preprocessing

The SMS dataset has been cleaned using various functions implemented in nltk Loper and Bird (2002) package and the steps for cleaning and preprocessing the data set is listed below:

- The data stored in arrays with each element being a sentence which contains the SMS message , the sentences are then tokenized using split() function with white space as delimiter into words also called tokens.
- Sentences have words both in lower case and upper case , to maintain consistency all words are converted to lower case.
- The tokens are latter cleaned (noise removal) by removing all punctuation and special characters .
- The sentences are then Lemmatized , Text lemmatization is the process of eliminating redundant prefix or suffix of a word and extract the base word (lemma).
- All tokens which happen to be stop words are removed.

Feature Extraction

The text data after noise removal is now converted to numerical data by counting the frequency of occurrence of each word in the sentence . Then the term frequency is converted to Term Frequency-Inverse Document Frequency as TF-IDF normalizes the term frequency based on no. of documents a term appears in.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Figure 1: Figure showing TF-IDF , where tf is the term frequency , N is the number of documents and df is the no. of documents a token appears in.

This processed numerical data is passed onto machine learning models with optimal hyper parameter found using GridCV search , where the optimal optimizer and the range of n-grams to be included is found based on the best performing model . The best model is the one with highest f1 score.

Naive Bayes Classifier

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.Pedregosa et al. (2011)

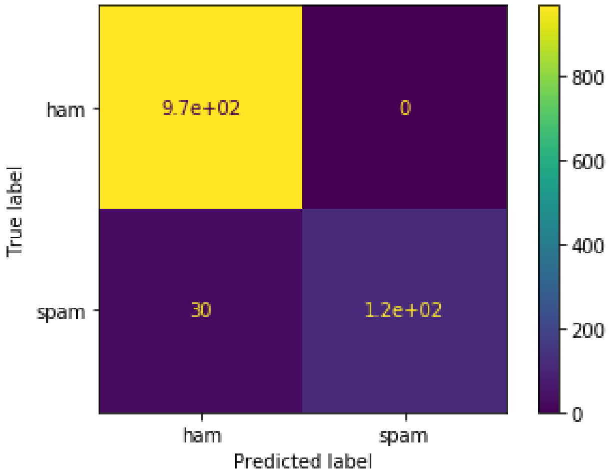

Figure 2: Figure showing confusion matrix of test data on Naive Bayes Classifier model

Logistic Regression Classifier

Logistic regression, is a linear model based on the logistic function for classification which predicts the probability of occurrence of data point in a particular class .

The model best performed with text input in bi-grams and tri-grams and the optimal solver was liblinear.

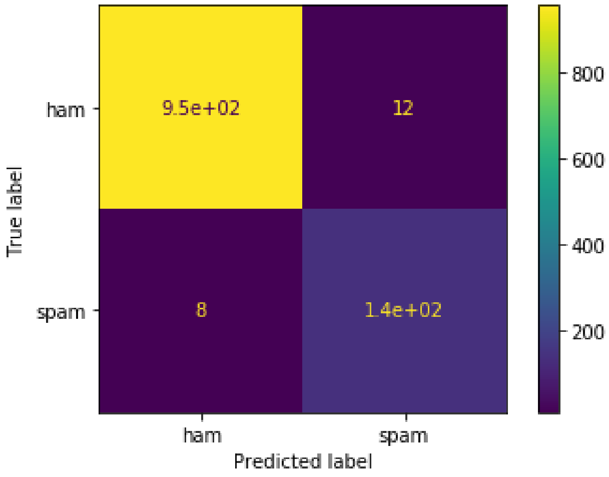

Figure 3: Figure showing confusion matrix of test data on logistic regression model

Support Vector Machine Classifier

A support vector machine constructs a hyper-plane which can be used for binary class classification , this hyper plane is chosen such that it maximizes the distances from the nearest points of the 2 classes and acts as a decision boundary.

The optimal solution for SVM was found with input text in bi-grams and tri-grams and with a linear kernel.

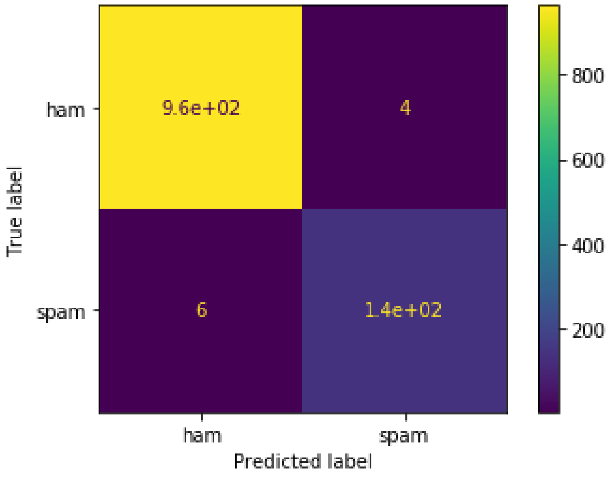

Figure 4: Figure showing confusion matrix of test data on Support Vector Machine Classifier model

Evaluation Criteria

In evaluating our model's performance , we want to classify whether an SMS is spam or not. We measured our model's performance using precision , recall and F1- score. Considering spam messages to be negative and not spam messages to be positive , we want to reduce the number of false positives as this can have detrimental effects on an SMS receiver. So both accuracy and recall needs to be taken care of , hence we use F1 score which is the geometric mean of accuracy and recall. Since our data points of class "spam" are large in number we also used balanced accuracy to circumvent the imbalance in class of data points. A macro-average will compute the metric independently for each class and then take the average (hence treating all classes equally), whereas a micro-average will aggregate the contributions of all classes to compute the average metric. Hence micro average also turns to be a good parameter to judge our model given the imbalance in the data set towards one class.

Metric	Formula
True positive rate, recall	$\frac{\text{TP}}{\text{TP+FN}}$
False positive rate	$\frac{\text{FP}}{\text{FP+TN}}$
Precision	$\frac{\text{TP}}{\text{TP+FP}}$
Accuracy	$\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$
F-measure	$\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$

Figure 5: Figure showing mathematical formula's of metrics discussed above . TP happens to be true positives , FP are false positives , TN are true negatives and FN are false negatives.

Analysis and Results

The support vector machine model outperforms the other two in terms of accuracy and recall and the tables below show the performance of the 3 models based on metrics discussed above. Results show that SVM is a good model to work with an imbalanced data set .

Table 1: Table showing performance of Naive Bayes Classifier

metric	precision	recall	f1-score	support
ham	0.97	1.00	0.98	966
spam	1.00	0.80	0.89	149
accuracy	-	-	0.97	1115
macro avg	0.98	0.90	0.94	1115
weighted avg	0.97	0.97	0.97	1115

Table 2: Table showing performance of Logistic Regression

metric	precision	recall	f1-score	support
ham	0.99	0.99	0.99	966
spam	0.93	0.95	0.94	149
accuracy	-	-	0.98	1115
macro avg	0.96	0.97	0.97	1115
weighted avg	0.98	0.98	0.98	1115

Table 3: Table showing performance of Support Vector Machine

metric	precision	recall	f1-score	support
ham	0.99	1.00	1.00	966
spam	1.00	0.94	0.97	149
accuracy	-	-	0.99	1115
macro avg	1.00	0.97	0.98	1115
weighted avg	0.99	0.99	0.99	1115

Discussions and Conclusion

The fact that words such as SMS , customer are closely associated with spam , the model mislabels them as spam (see Table 4). The messages wrongly labelled as spam , the model hasn't been able to understand the context of the message and hence models which take context into account such as LSTM , GRU models can be used to improve the performance of the spam classifier.

Table 4: Table showing the mislabelled messages by SVM Model

SMS message	Actual label
customer place call	ham
th gower mate am! r u man good wale ill b back â‘morrow c u wk wa msg 4 â– random	ham
babe: u want dont u baby im nasty thing 4 filthyguys fancy rude time sexy bitch go slo n hard txt xxx slo(4msgs)	spam
yup next stop	ham
hello darling today would love chat dont tell look like sexy	spam
realize 40 year we'll thousand old lady running around tattoo	spam
sorry missed call let's talk time i'm 07090201529	spam
burger king - wanna play footy top stadium get 2 burger king 1st sept go large super coca-cola walk winner	spam
customer place call	ham
email alertfrom: jeri stewartsize: 2kbsubject: low-cost prescripiton drvgsto listen email call 123	spam

The above models were implemented in jupyter notebooks Kluyver et al. (2016) using the scikit-learn Pedregosa et al. (2011) package in python Van Rossum and Drake (2009) language. The GitHub repository Kowsari et al. (2019) had served as a good reference for this work. Stackexchange has also served as a good resource for debugging LaTex and Python scripts. All the codes used for this work can be found at the GitHub repository. Any changes you wish to suggest can be reported on GitHub itself.

References

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. In Loizides, F. and Schmidt, B., editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. E., and Brown, D. E. (2019). Text classification algorithms: A survey. *Information*, 10(4).

Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics.*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual.* CreateSpace, Scotts Valley, CA.