Machine Learning Nanodegree

Capstone project SMS Spam Detection

S. Revathi

February 21th 2019

Proposal: SMS Spam Detection

I. Definition

This section defines problem statement as well as an overview of the message spam detection . The metrics used for evaluating the model and the function set are also described below.

1.1. Project Overview

This project is to explore the results of applying machine learning techniques to SMS spam detection. Short Message Service junk mail (sometimes known as mobile cell phone junk mail) is any junk message introduced to a mobile smart phone as text messaging through the SMS. The dataset for this challenge originates from the UCI Machine Learning Repository. More detail about dataset can be observed on UCI dataset authentic website. This dataset has been accrued from unfastened or loose for• research sources at the Internet. The collection is composed of simply one text report, wherein• every line has the right magnificence observed by using the uncooked message.

1.2. Problem statement

Here, we'd try to analysis exceptional techniques to become aware of junk mail messages. We will use the one of a kind approach, based on word remember and term-frequency inverse file-frequency (tf-idf) remodel to classify the messages. Following steps are required if you want to obtain the goal:

- Download and pre-process the SMS Spam Collection v.1 dataset.

- Test and discover first-rate technique (phrase depend or tf-idf• vectorizer) to categorise the messages. Selection of method and splitting the dataset into education and

- testing information. Initialize numerous classifier and educate it.

- Evaluate the classifiers and locating exceptional the model for a• dataset.

1.3. Metrics

Accuracy is the first metric to be checked whilst the algorithms are evaluated, is the sum of authentic positives and the actual poor outputs divided by using the statistics size.

The Scikit-examine library gives a convenience file whilst working on category issues to provide you a short idea of the accuracy of a model the use of a number of matrices, one among them is F1_score which paintings with all the model

Accuracy is defined as the number of true positives (TP) plus the number of true negative (TN) over the total number of sample (N).

Accuracy=(TP+TN)/N

Precision (P) is defined as the number of true positives (TP) over the number of true positives plus the number of false positives (FP) whereas Recall (R) is defined as the number of true positives (TP) over the number of true positives plus the number of false negatives (FN).

P=TP/(TP+FP) R=TP/(TP+FN)

These quantities are also related to the F1 score, which is defined as the harmonic mean of precision and recall. I found F1_score as appropriate to use as report metric in order to have a good idea of how the algorithm is behaving

F1=2*(P*R)/(P+R)

II. Analysis Below

Describes how the statistics was accumulated, which features were selected, and which algorithms had been explored. Finally, I outline the benchmark used to assess the performance of the trading method.

2.1. Data exploration

The dataset used for this project is SMS Spam Collection v.1 dataset originates from the UCI Machine Learning Repository. This dataset has been amassed from free or unfastened for studies resources on the Internet. The collection is composed of just one textual content file, where each line has the precise elegance observed through the raw message.

Number of ham messages in data set: 4825

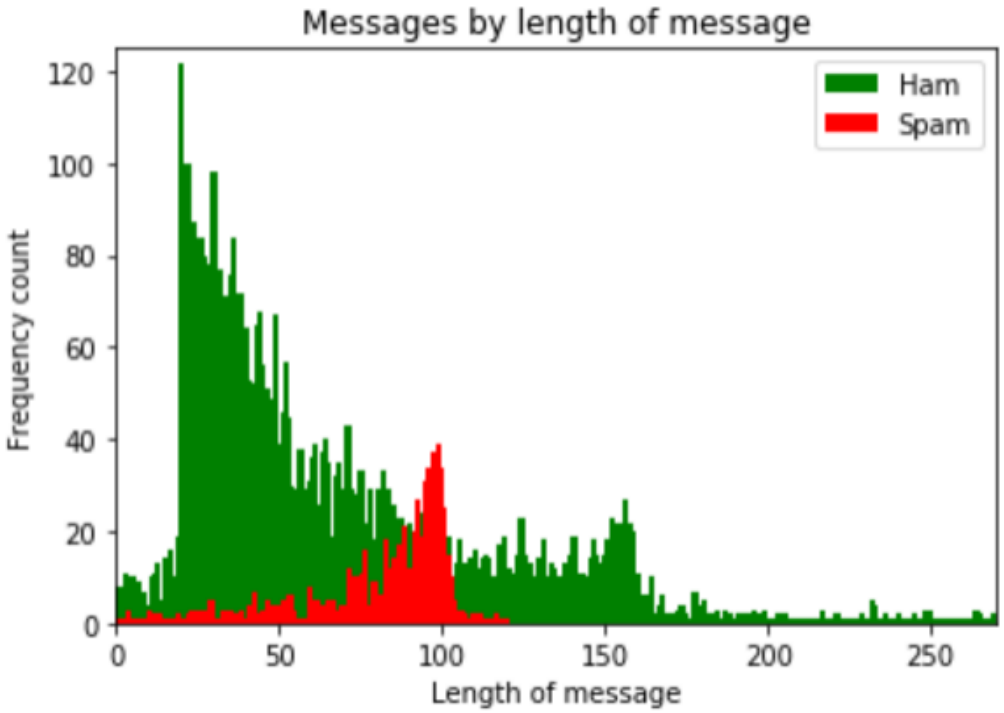
Number of Spam messages in data set: 747

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

This dataset is tab-separated values (TSV) report. There are general 5572 entries in the dataset and has two column “Class” and “Text” where each row represent distinctive message and Class comprise unique categories ham and junk mail. Dataset does no longer require any sort of cleaning, wrangling and there's no null fee in any column.

2.2 Exploratory Visualization

The 2 categories ham and spam, it is feasible to take benefit of the textual content due to the fact it is one of a kind in every of the categories. There are a whole lot of phrases common words within the identical magnificence of messages. Each category of messages has a few similar form of key phrases i.E. In spam message phrases like Free, Winner, win, won, award, congrats, congratulation, decided on, urgent, and Cash it's miles easier to classify them. Another way of looking messages is word rely, counting words in every message and plotting graph with the increasing wide variety of word duration. Spam messages are plotted in crimson and ham messages are plotted in green. But due to the fact that all of the junk mail messages are overlapping on ham messages it might be very difficult to identify junk mail/ham messages solely on the premise of duration. Thus classify on the idea of phrase count could no longer be beneficial.



2.3 Algorithms and Techniques

Text class issues can be each supervised and unsupervised. As we defined in advance, our problem context made us pick out an approach based on supervised learning, wherein we desired to classify our text and check whether it's miles unsolicited mail or ham. The class troubles offer a huge style of gadget learning algorithms to clear up a hassle, it's not possible to recognise which approach and which

algorithm goes to work the high-quality way at the dataset beforehand. That approach we want to the usage of trial and blunders.

Techniques which we're going to us are primarily based on phrase rely and term-frequency inverse documentfrequency (tf-idf) transform. Some of the maximum important gadget mastering algorithms are blanketed in this mission are Naive Bayes, Decision Tree, AdaBoost, KNearest Neighbours and Random Forest.

- Naïve Bayes(Multi-NB) one of the most effective class algorithms that exist. Naive Bayes doesn't require a ton of information and is understood to be very speedy.
- Decision Trees (DTs) are a non-parametric supervised getting to know technique used for type and regression. The goal is to create a version that predicts the cost of a target variable by means of getting to know simple choice policies inferred from the statistics features.
- AdaBoost (AdaBoost) classifier is a meta-estimator (in our case it's miles Decision Tree) that starts off evolved by becoming a classifier on the authentic dataset and then suits additional copies of the classifier at the identical dataset but in which the weights of incorrectly classified times are adjusted such that subsequent classifiers consciousness more on hard cases.
- Random Forest (RF) is simple to interpret and is nonparametric,• meaning we don't need to worry about tuning a group of parameters which include while the use of a Support Vector Machine. They are often praised due to the fact they work "out of the box".
- KNearest Neighbor (KNN) a simple and regularly effective• category or regression set of rules. K-NN is a kind of example based mastering, or lazy mastering, wherein the function is only approximated domestically and all computation is deferred until category.

Actually above all machine learning algorithms are used for predicts spam message among our data set

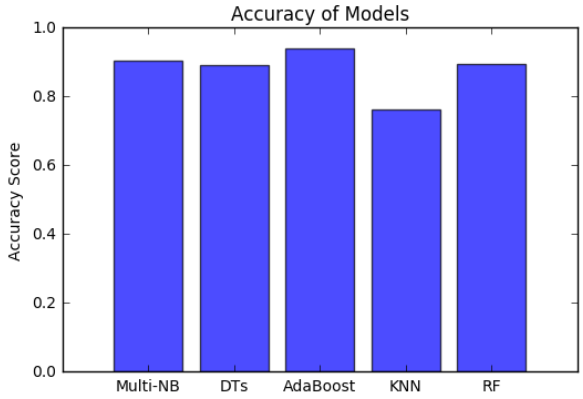
```
: # loop to call function for each model
  clf = [A,B,C,D,E]
  pred_val = [0,0,0,0,0]

  for a in range(0,5):
      train_classifier(clf[a], X_train, y_train)
      y_pred = predict_labels(clf[a],X_test)
      pred_val[a] = f1_score(y_test, y_pred,  average='binary')
      print pred_val[a]
```

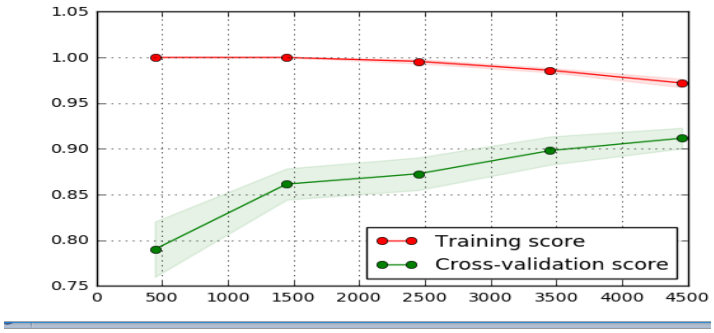
```
0.904411764706
0.890459363958
0.940350877193
0.763485477178
0.896296296296
```

Predict the f1-score of each algorithm and then we can determine that Ada Boost classifier has a high f1-score so that buy using the Ada Boost classifier we can easily predict the spam sms below we it as clear with visualization

```
# ploating data for F1 Score
y_pos = np.arange(len(objects))
y_val = [ x for x in pred_val]
plt.bar(y_pos,y_val, align='center', alpha=0.7)
plt.xticks(y_pos, objects)
plt.ylabel('Accuracy Score')
plt.title('Accuracy of Models')
plt.show()
```



For predict the sms spam we can train and test the score train score can be represented as red color and test score can be represented as green color



Machine Learning algorithms. The available public and private leader board score in the Kaggle competition can be used to benchmark the performance of my algorithm. Also, it is possible to explore how the proposed model perform compared to existing models. The result shows that Naïve Bayes work better on the dataset with an accuracy score of 0.70.

III. Methodology

As precise before dataset does now not require any sort of cleaning. Column “Class” incorporate express value ham and junk mail due to the fact classifier require numeric value ham is replaced with 1 and spam is replaced with 0. Similarly new columns are delivered "Count". Count contain a number of phrases in a given message.

	Class	Text	Count
0	0	Go until jurong point, crazy.. Available only ...	111
1	0	Ok lar... Joking wif u oni...	29
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	0	U dun say so early hor... U c already then say...	49
4	0	Nah I don't think he goes to usf, he lives aro...	61

3.2 Implementation

3.2.1 Data processing

To teach our classifier we want to use time period frequency– inverse report frequency (tf–idf), it's far a numerical statistic that is supposed to reflect how vital a word is to a record in a corpus. It is used as a weighting thing in statistics retrieval, text mining, and person modelling.

$tf.idf(t,d) = tf(t,d) \times idf(t)$

The tf-idf cost increases proportionally to the wide variety of times a phrase seems within the file however is offset via the frequency of the word in the corpus, which allows to regulate for the fact that some words appear extra regularly in popular.

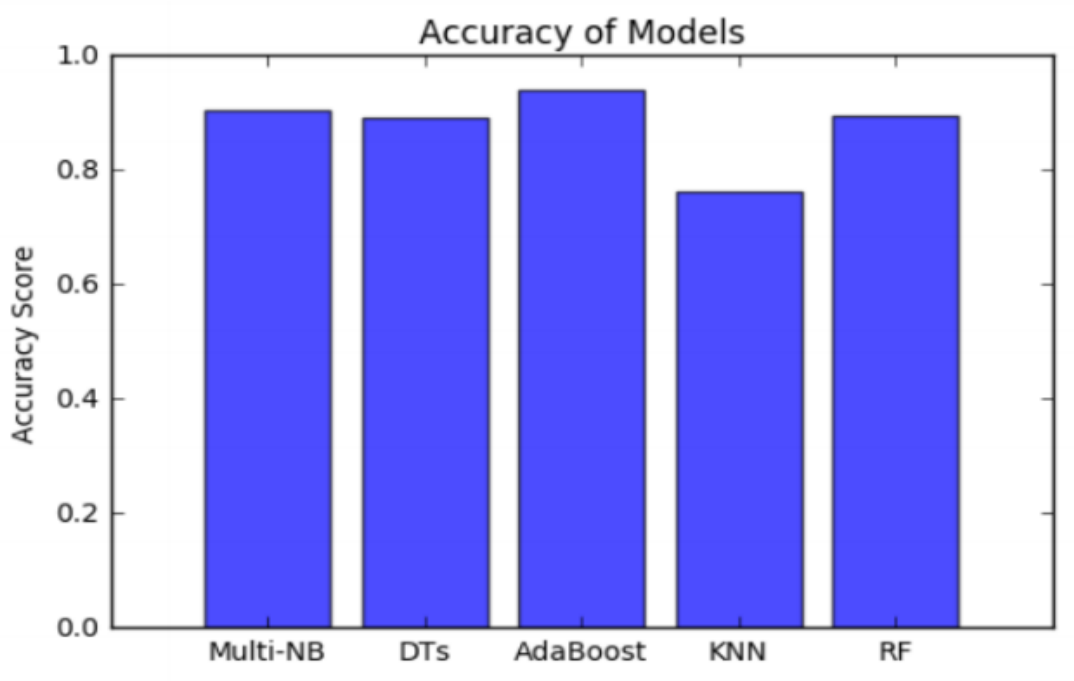
3.2.2 Classifier and Training

Before we begin classifying, we first cut up the information into a training and checking out set. Features are saved in X and goal is stored in y. Data is divided into schooling and trying out with a ratio of 4:1 that's eighty% for education and 20% for trying out. The classifiers are initialised for Naïve Bayes, Decision Tree, AdaBoost, Random Forest, and KNearest Neighbor and the training for the version starts, then the checking out outputs the very last predictions and compares them with the check labels to degree the accuracy the use of the accuracy_score. Training model turns into quite simple with none because of TfidfVectorizer from Scikitanalyze library, because it converts all of the text facts into numeric statistics and simplifies the method of training and checking out for text data.

3.2.3 Visualization

The accuracy of the models after schooling them with model tuning and without model tuning. Bar graph has been plotted to reveal accuracy of various models. The bar graph on left indicates the accuracy of models without tuning and version on right indicates the accuracy of fashions with model

tuning.



3.2.4 Results

Clearly, AdaBoost is operating pleasant whilst as compared to all of the different model, after that Random Forest, Decision Tree, then Naïve Bayes and KNearest Neighbor. Since all of the fashions are operating thoroughly even without any form of version tuning or refinement there may be possibilities that models are over-fitting information for that reason archiving excessive accuracy.

3.3 Refinement

As stated in advance the tf-idf cost will increase proportionally to the wide variety of instances a phrase appears inside the report however is offset by using the frequency of the word inside the corpus, which facilitates to regulate for the truth that some phrases seem extra regularly in popular. Words like I, me, my, it, the, he, she does now not offer a whole lot facts about ham/spam however are very excessive in frequency as a consequence decreases the accuracy. To remove those sort of phrase from characteristic information, herbal language toolkit (nltk) can be beneficial, the nltk feature for stopwords may be used to put off popular phrases from feature facts and provide more significant facts for the classifier to educate on and for this reason increases the an accuracy of version.

IV. Results

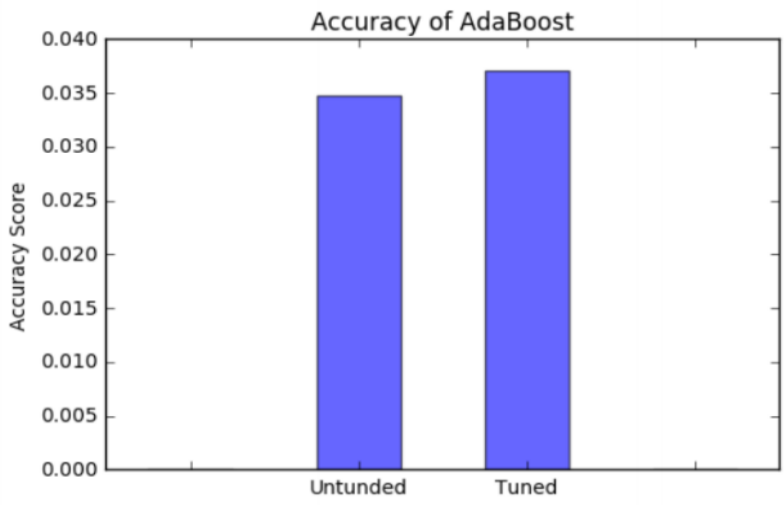
Yes, my model is well generalize to unseen data.

Yes, it is sensitive to small changes in the data because Ada Boost can be sensitive to outliers / label noise because it is fitting a classification model (an additive model) to an exponential loss function, and the exponential loss function is sensitive to outliers/label noise.

Present We cannot trust this model completely because our data is old data so it is works well on my data but now technology was developed so number of spam messages was increased

4.1 Model Evaluation and Validation

The accuracy of the fashions after schooling them with stop words and model tuning. The bar graph has been plotted to reveal accuracy score of various models. Clearly the winner after refinement is AdaBoost with an accuracy of 0.9370 which higher than untuned version zero.9347 and runner-up version is Naïve Bayes which have worked thoroughly after refinement compared to the result before refinement. The bar graph on left suggests accuracy after model tuning and bar graph on right show accuracy of AdaBoost with a tuned and unturned model. No explicit tuning required for AdaBoost as all the model has been tuned using stop word earlier than test_train_split.



4.2 Justification

The result of this model is very satisfactory, comparing to the benchmark. All the final fashions labored higher than benchmark despite half of the dataset, it's far feasible to mention that the end result is honest and the venture is finished underneath delight the precision of 93%. Only thing negative is the time it takes for training, perhaps 10 to 15 seconds but the output has accurate labels and that is the most important.

V. Conclusion

5.1 Reflection

In this project, we tried to evaluation exclusive strategies to identify unsolicited mail messages. We used the exceptional technique, primarily based on phrase remember and time period-frequency inverse file-frequency (tf-idf) remodel to classify the messages. Since check data have very excessive meaning for human and very hard for the system to understand, the biggest challenge became to evaluation the take a look at facts and convert it into a few significant numeric data without traumatic the relation among categories. Best approach to transform the check data into significant numeric information is tf-idf vectorizer. This become the maximum interesting part of the whole mission to convert large quantity of textual content information into numeric records. Best end result become generated the use of tf-idf vectorizer with AdaBoost and Naïve Bayes classifier, both completed accuracy approx. Ninety three%, that is a great end result.

5.2 Improvement

Finding the precise device studying version isn't always the end of the paintings, it is feasible to save and load the version the usage of Scikit - learn's Pickle, a neural networks can achieve some great results.

This consideration must be taken in account for improvement of model

VI. References

- https://www.kaggle.com/uciml/sms-spam-collection-dataset
- http://www.nltk.org
- http://scikit-learn.org/stable/modules/classes.htm