Sentiment Analysis using Unsupervised Learning techniques on Twitter data

Nontawat Pattanajak
Machine Learning Engineer Nano-degree Program
7 June 2020

Section 1: Definition

1.1 Project Overview

The purpose of this project is to investigate potential models to understand contents from messages on Twitter, which is classified into 3 groups such as positive, negative, and neutral. The study by Wenbin Zhang and Steven Skiena in 2010 [6] shows that there is a positive link between social media message and the trend of stock market price. Developing a Machine Learning model could benefit this work, which is called Sentiment Analysis.

1.2 Problem Statements

To develop a Machine Learning model for Sentiment Analysis, it is unlikely to label data from twitter messages in practice. Unsupervised Learning, which is one of the most well-known methods in Machine Learning, is considered to develop the model.

1.3 Metrics

To evaluate the performance for each Machine Learning model, accuracy, precision, recall, and f-score are considered to use.

Section 2: Analysis

2.1 Dataset

The dataset used in this project is developed by Sanders Analytics. There are 5,113 tweets related to 4 companies such as Apple, Google, Microsoft, and Twitter. The dataset can be found from this link [3]. The data is labelled into 4 groups such as irrelevant, negative, neutral, and positive.

However, when developing the Machine Learning model in this project, 3 classes are considered to use such as negative, positive, and neutral – there are totally 3,424 tweets. The number of twitter messages for each class is presented in table below.

Table 1: Dataset Breakdown

Label	Company				Total
	Apple	Google	Microsoft	Twitter	
irrelevant	139	479	500	571	1689
negative	**316**	**57**	**132**	**67**	**572**
neutral	**523**	**579**	**641**	**590**	**2333**
positive	**164**	**202**	**91**	**62**	**519**
Total	1142	1317	1364	1290	5113

Even though Unsupervised Learning method does not require labelled data for building the Machine Learning models, Sanders Analytics data (labelled data) is still considered to use in this

project. This is because it can be used to evaluate the performance of the models in terms of accuracy, precision, and re-call. Without the labelled data, it is impossible to do.

2.2 Unsupervised Learning Techniques

Unsupervised Learning is a method of classifying data which if the data is the similar characteristic, it will be classified into the same group. In this project, 5 techniques are introduced to use such as KMean, MiniBatchKMean, SpectralClustering, AgglomerativeClustering, and MeanShift.

(1) KMean: This technique clusters data by separating the data into groups by finding the similar characteristics. It requires developers to provide number of groups for the classification. The algorithm starts by randomly select the centroids of each data group. Then, the distance between centroid and each data point will be calculated to find the error. Then, the centroids will be moved until the error is the lowest value.

(2) MiniBatchKMean: This technique is the advantage of KMean. They have the similar concept. The exception is that MiniBatchKMean uses mini-batches to decrease the computational time [1]. Therefore, Mini-Batch size is another input required from developers. In general, MiniBatchKMean will provides the faster technique but its quality might be decrease [4].

(3) SpectralClustering: This technique starts by running a low-dimension embedding of the affinity matrix between samples, following by clustering technique – Kmean. This technique is suitable for small number of clusters [4].

(4) AgglomerativeClustering: It is one type of the most useful methods in hierarchical clustering. It starts by finding its own cluster. Then, clusters are merged together. The conditions to consider to merge the clusters are (1) Ward – to minimize the difference of squared sum, and (2) Maximum/Average/Single linkage – to minimize the maximum distance between data pairs of clusters [4].

(5) MeanShift: MeanShit technique is another clustering method. It can find the centroid by finding blobs in a smooth density of samples. The algorithm sets the number of clusters by itself and it stops the process of clustering when the change of centroid is small [4].

In addition to these, PCA is also considered to reduce the highly dimensional numbers of data [5] which may improve the model accuracy.

Principle Component Analysis (PCA): In the real-world problem of Sentiment Analysis, it might to work with a large number of vocabularies. This means that the size of training data trends to increase. It may lead to generate the lower accuracy of Clustering methods – the Clustering methods have to consider many dimensionalities to group the data together. PCA is considered to overcome this problem by providing the smaller dimensionality of data while it still provides the similar characteristics of the data. Transforming training data with PCA before feeding the data into classification or clustering model is a suggestion.

2.3 Benchmark

According to a study by Kishori K. Pawar and R. R. Deshmukh in 2015 [2], the state-of-the-art model for Sanders Analytics dataset is using Artificial Neural Networks which provides the accuracy at 88.62 percent.

Section 3: Methodology

3.1 Data Preprocessing

In this process, the original data is prepared before feeding into Clustering models. There are 4 steps such as loading data, splitting data, cleaning data, and feature extraction.

(1) Loading data: The original data is in csv file format. Pandas library is considered to use its built-in functions to load and filter basic information.

(2) Splitting data: The data is separated into 2 groups such as training and testing data with ratio 80:20. A built-in function from sk-learn is considered to do in this process.

(3) Cleaning data: The data is cleaned in 2 parts – cleaning training data (x) and labelled data (y).

   a. *Training Data (x):* This is the message from twitter data. The data is cleaned by removing punctuation, un-alphabetic words, stop words, and short words.
   b. *Labelled Data (y):* The labelled data in string words are converted into integer such as 'positive' is 2, 'negative' is 0, and 'neutral' is 1.

(4) Feature Extraction: This step is to convert list of vocabulary data into integer matrix which can be used to train Clustering models. `CountVectorizer` which is a built-in function from sk-learn, plays a major rule to process the data.

3.2 Implementation

There are 5 Unsupervised Learning Technique is considered to develop a Machine Learning model for Sentiment Analysis. This is to find the model that can provide the highest accuracy to improve the model for the next step. The table below shows 5 Unsupervised Learning techniques and their input parameters.

Table 2: Input Parameters for 5 Machine Learning models

Model	Model Name	Input Parameters
1	Kmean	n_clusters=3, random_state=0
2	MiniBatchKMeans	n_clusters=3, random_state=0, batch_size=6
3	SpectralClustering	n_clusters=3, assign_labels="discretize", random_state=0
4	AgglomerativeClustering	n_clusters=3
5	MeanShift	bandwidth=3

3.3 Refinement

The model that can provide the highest accuracy from the previous step is considered to do the improvement with PCA. The assumption is that lower dimensionality of data could provide the simpler data for training a Machine Learning model and it results in better model's performance. Therefore, PCA is considered to use by varying the number of dimensionalities from 100 to 2600 and step size is 100.

3.4 Method to evaluate model performance

The process of evaluating model performance can be classified into 3 steps.

(1) Selecting the best model performance from 5 models by considering accuracy, precision, recall, and f-score for each model

(2) Finding number of component analysis for PCA for the selected model which aims to improve the selected model's accuracy

(3) The evaluation of improved model by considering accuracy, precision, recall, and f-score for the selected model by showing matrix analysis.

Section 4: Results

4.1 Performance of 5 Machine Learning models

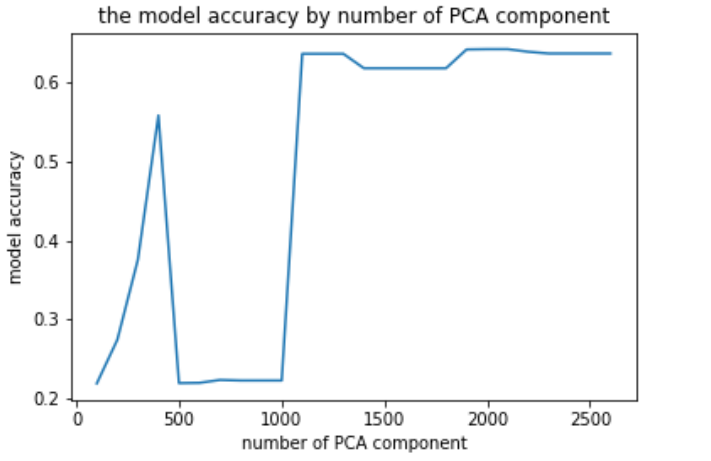
5 models are evaluated by 5 criteria such as accuracy of training data, accuracy of test data, recall, precision, and f-score. It observes that MiniBatchKMeans model provides the highest accuracy for both training and test data at 52.68 and 51.32 percent respectively. This model is selected to be improved for the next step. The table below shows performance of 5 models.

Table 3: The mode performance of 5 Machine Learning models

Evaluation	Kmean	MiniBatchKMeans	SpectralClustering	AgglomerativeClustering	MeanShift
f_score (test data)	0.384182	0.49636	0.073788	0.312678	0.077294
Precision (test data)	0.41394	0.490482	0.694899	0.50239	0.695537
re_call (test data)	0.360584	0.515328	0.20438	0.278832	0.20146
testing_accuracy	0.360584	**0.515328**	0.20438	0.278832	0.20146
training_accuracy	0.385907	**0.526835**	0.159182	0.315078	0.121942

4.2 The Result of finding number of components for PCA

MiniBatchKMeans model is considered to be improved with PCA. After running the experiments to find the most suitable number of components. It found that number of components at 2000 and 2100 provides the highest training accuracy at 64.15 percent.




```
 n_component:  100 accuracy:  0.2194231471339905
n_component:  200 accuracy:  0.27455275648046734
n component:  300 accuracy:  0.37568455640744797
 n_component:  400 accuracy:  0.5578678349762687
n_component:  500 accuracy:  0.21978824388462942
n_component:  600 accuracy:  0.22015334063526834
n_component:  700 accuracy:  0.22380430814165753
 n_component:  800 accuracy:  0.2230741146403797
 n component:  900 accuracy:  0.2230741146403797
n component:  1000 accuracy:  0.2230741146403797
n_component:  1100 accuracy:  0.6356334428623586
n_component:  1200 accuracy:  0.6356334428623586
n component:  1300 accuracy:  0.6356334428623586
n component:  1400 accuracy:  0.6173786053304126
n_component:  1500 accuracy:  0.6173786053304126
n_component:  1600 accuracy:  0.6173786053304126
n_component:  1700 accuracy:  0.6173786053304126
n_component:  1800 accuracy:  0.6173786053304126
n_component:  1900 accuracy:  0.6411098941219423
n_component:  2000 accuracy:  0.6414749908725812
n_component:  2100 accuracy:  0.6414749908725812
n_component:  2200 accuracy:  0.6381891201168309
n_component:  2300 accuracy:  0.6359985396129975
n_component:  2400 accuracy:  0.6359985396129975
n component:  2500 accuracy:  0.6359985396129975
n_component:  2600 accuracy:  0.6359985396129975
```

4.3 The performance result of the improved model

The improved model, MiniBatchKMeans with PCA (component 2000), is evaluated the performance by comparing the information before improving. It found that after applying PCA, the model accuracy increases for both training and test data to be 64.18 and 60.88 percent respectively. F-Score of test data which shows the balance between precision and recall is observed at 0.52.

Table 4: The performance comparison of a baseline and an improved model.

Evaluation	MiniBatchKmeans (baseline model)	MiniBatchKMeans with PCA (improved model)
f_score (test data)	0.49636	**0.520964**
Precision (test data)	0.490482	0.4617
re_call (test data)	0.515328	0.608759
testing_accuracy	0.515328	**0.608759**
training_accuracy	0.526835	0.64184

In addition to this, there is analysis by considering the performance for each class (positive, negative, and neutral) of testing data. It found that the model seems to be able to classify 'neutral' data. On the other hands, it fails to classify 'negative' and 'positive' data – it observes low values of precision, re-call, and f-score.

Table 5: The model performance of test data by classes.

	precision	recall	f1-score	support
negative	0.00	0.00	0.00	138
neutral	0.65	0.89	0.75	448
positive	0.27	0.17	0.21	99
micro avg	0.61	0.61	0.61	685
macro avg	0.30	0.35	0.32	685
weighted avg	0.46	0.61	0.52	685

4.4 Discussion

In the first step, 5 Machine Learning models are developed and evaluated the performance to do sentiment analysis using un-labelled data. It found that MiniBatchKMeans technique provide the best model. Then, it is considered to improve the model performance by applying PCA to data before feeding into the MiniBatchKMeans model. The purpose of using PCA is to reduce the dimensionality of data which may be able to improve the model accuracy.

At the final result, it observes that the model developed by MiniBatchKMeans technique and PCA with component 2000 provides the highest accuracy at 64.18 and 60.88 percent for training and testing data. It also found that the model seems to be able to classify 'neutral' data but it is not successful to identify 'positive' and 'negative' data.

Section 5: Conclusion

Developing a Machine Learning model for Sentiment Analysis using Unsupervised Learning technique is run an experiment by considering 5 techniques such as KMean, MiniBatchKMean, SpectralClustering, AgglomerativeClustering, and MeanShift. It found that MiniBatchKMean technique provides the highest model accuracy at 52.68 and 51.53 percent for training and test data respectively. The model developed with this technique is improved by applying PCA for the data before feeding into the model. It also found that the model can be improved by 11.55 and 9.34 percent for training and test data respectively.

When comparing to the state-of-the-art model which is developed by Artificial Neural Network (ANN) by Kishori K. Pawar and R. R. Deshmukh in 2015 [2], the MiniBatchKMean model with PCA cannot beat the state-of-the-art model. The state-of-the-art model achieves the accuracy at 88.62 percent, while the model in this project achieves at 60.88 percent. This is because the model developed by ANN uses labelled data to train the model, while the model in this project uses un-labelled data. In order to beat the state-of-the-art model, the developers might re-consider data preprocessing by applying other techniques, which might provide relevant data to train Unsupervised Learning model.

References

[1] D. Sculley, "Web-Scale K-Means Clustering", https://www.eecs.tufts.edu/~dsculley/papers/fastkmeans.pdf [accessed 7 June 2020]

[2] Kishori K. Pawar and R. R. Deshmukh, "Twitter Sentiment Classification on Sanders Data using Hybrid Approach", https://www.researchgate.net/publication/281002808_Twitter_Sentiment_Classification_on_Sanders_Data_using_Hybrid_Approach [accessed 31 May 2020]

[3] Sanders Niek J., "Twitter Sentiment Corpus." Sanders Analytics. Sanders Analytics LLC., https://github.com/zfz/twitter_corpus [accessed 31 May 2020]

[4] SK-Learn, "Clustering", https://scikit-learn.org/stable/modules/clustering.html [accessed 7 June 2020]

[5] SK-Learn," Decomposing signals in components", https://scikit-learn.org/stable/modules/decomposition.html [accessed 7 June 2020]

[6] W. Zhang and S. Skiena, "Trading Strategies to Exploit Blog and News Sentiment", https://www.researchgate.net/publication/221297824_Trading_Strategies_to_Exploit_Blog_and_News_Sentiment [Accessed 7 June 2020]