Sentiment Analysis of Nine Android App Reviews

1- Introduction:

Data analysis and analytics has enabled to reveal the hidden facts and patterns about the big data. A large volume, variety and velocity of data is called big data. Businesses are having huge amount data on their e-commerce websites which is exceeding rapidly every day and that results in increasing the capacity of the IT departments of the businesses. The data contains other two V's which plays an important role for the businesses which are Veracity and Value. E-commerce data consist of the reviews of the customers which can be used to determine the behavior of the consumers. The behaviors are considered as the most reliable source to make decisions on the future trends of the data. In the E-commerce dataset could vary from structured to unstructured dataset. The relevant data is filtered and the useful information is fetched from insights of the data. The insights focused on the e-commerce are future trends planning and forecasting, marketing opportunities and getting patterns of consumer behavior etc.

Sentiment Analysis is the process of 'computationally' determining whether a piece of writing is positive, negative or neutral. It's also known as opinion mining, deriving the opinion or attitude of a speaker.
Why sentiment analysis?
Business: In marketing field companies use it to develop their strategies, to understand customers' feelings towards products or brand, how people respond to their campaigns or product launches and why consumers don't buy some products.
Politics: In the political field, it is used to keep track of political view, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well.
Public Actions: Sentiment analysis also is used to monitor and analyze social phenomena, for the spotting of potentially dangerous situations and determining the general mood of the blogosphere.

2- Literature Review:

NLTK is an open source natural language processing (NLP) platform available for Python. It is capable of textual tokenization, parsing, classification, stemming, tagging, semantic reasoning and other computational linguistics. NLTK comes with an inbuilt sentiment analyzer module – nltk.sentiment.vader—that can analyse a piece of text and classify the sentences under positive, negative and neutral polarity of sentiments. I have used this in my project as train set data is not heavy and reviews result after analyzer are satisfactory. The advantage of this approach is that sentences containing negated positive words (e.g. "not happy", "not good") will still receive a negative sentence sentiment (thanks to the heuristics to flip the sentiment of the word following a negation). Some simpler sentiment analysis tools will just take the average of the sentiments of the words and would miss subtle details like this. The disadvantage of this approach is that Out of Vocab (OOV) words that the sentiment analysis tool has not seen before will not be classified as positive/negative (e.g. typos). At a macro level, the Python sentiment analysis compound score recognized the negativity of the overall data set. Like I stated in the introduction to Vader, if you need quick and dirty, this is your tool. Sentiment analysis tools can provide quick and easy results that classify blobs of text into positive, neutral, negative, and more. Punching compound scores into lists could be very useful for looking at unit-based, role-based, or any other micro analysis apart from the aggregate.

Apart from Vader, one can create one's own classification model using Naïve's Bayes Classifier. In the machine learning context, Naïve's Bayes Classifier is a probabilistic classifier based on Bayes' theorem that constructs a classification model out of training data. This classifier learns to classify the reviews to positive or negative using the supervised learning mechanism. The learning process starts by feeding in sample data that aids the classifier to construct a model to classify these reviews. In classification tasks you need a big data set in order to make reliable estimations of the probability of each class. You can use Naïve Bayes classification algorithm with a small data set but precision and recall will keep very low. It is easy and fast to predict class of test data set. It also perform well in multi class prediction. When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data. It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as "Zero Frequency". To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.

On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously.Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Further, I am also mentioning a link detail where VADER has been compared to more than seven other sentiment analyzing systems and declared VADER is the best for social reviews sentiment analysis. One problem that is often overlooked is how to calculate probabilities, for Naive Bayes, when working with real valued features. Often, people attempt to either discretize the feature (which leads to questions on the number of discrete values to have) or they attempt to fit a normal curve. Potential solution is to estimate a none normal distribution. In addition to the answers provided above, I would only add that naive Bayes performs less well than other methods such as support vector machines. To get very picky, one could also raise the point that any probabilities calculated are not mathematically accurate. However, for machine learning purposes, we are only interested in relative probabilities.

3- Description of the Data:

Data has been chosen from Amazon website. Data consist of two portions, training set data and test set data. Both portions have 20,000 reviews about android applications of different mobile companies. Both datasets have different type of reviews and product IDs. For both datasets, each review includes three information; 1- Semantic Value, 2- Product ID, 3- Review text. Semantics values ranges from 1-3. 1 if review is negative, 2- if review is neutral and 3 if review is positive. In test dataset, I have to filter reviews of 9 product IDs that belongs to 3 companies; 3 products per company. Based on the reviews, I have to recommend which company is best suited to be invested in.

4- Methodology:

The data chosen consist of different types of data from nominal to the text format. In order to extract the useful information out of the data Python development environment has been used to deduce useful results in the form of different graphs patterns. Initially the random and unordered data has to be made in a form where it can be used. The cleaning of data has been done and a dataframe has been made. By looking insights to the data following tasks has been done in cleaning the data and making it in a usable form:

- Loading datasets in .txt format
- Extracting reviews and semantics values
- Creating a dataframe
- Applying semantic analyzer
- Creating binary label for supervised learning

After creating dataframe and arranging semantics values & reviews in two different columns, sentiment analyzer algorithm has been applied. It gives four useful values for each review; polarity value, positive value, negative value and a neutral value. These four values have been further used as features for creating a **binary label**. I have applied five different supervised learning classifiers to train the given training dataset, the results of which are mentioned below:

1- Logistic Regression

The results are not promising for training data set.

```
CLASSIFIER : LogisticRegression

Train Set Accuracy: 0.765120463860
Train Set ROC: 0.6946836826242029
Train Set Recall: 0.78780807103420
Train Set Precision: 0.93444504455
```

2- Neural Networks

The results are not promising for training data set.

```
CLASSIFIER : Neural Networks

Train Set Accuracy: 0.766669999000
Train Set ROC: 0.6921625958463619
Train Set Recall: 0.80104463437796
Train Set Precision: 0.91115311909
```

3- Random Forrest (Specialized)

The results are best for the training dataset and this classifier has been used for test dataset prediction.

```
CLASSIFIER : RandomForrest

Train Set Accuracy: 0.962061381585
Train Set ROC: 0.9677760422072779
Train Set Recall: 0.95712705744583
Train Set Precision: 0.99324871725
```

4- K- Nearest Neighbor

The results are not promising for training dataset.

```
CLASSIFIER : K-Nearest Neighbor

Train Set Accuracy: 0.769619114265
Train Set ROC: 0.69824294184374
Train Set Recall: 0.80124003542958
Train Set Precision: 0.91608155549
```

5- Support Vector Machine (SVM)

The results are not promising for training dataset.

```
CLASSIFIER : Supporrt Vector Machi
nes

Train Set Accuracy: 0.764420673797
Train Set ROC: 0.6969507578619044
Train Set Recall: 0.78312307261003
Train Set Precision: 0.94295166081
```

As mentioned above, I have calculated accuracy, ROC value, Recall and Precision for each classifier to make sure the best performed classifier is to be used for test dataset.

For test dataset, raw data includes many android application products from vaious companies. But task given requires to evaluate three companies based on the reviews of three android products from each company and recommend the most suitable company for investment. So I have first filtered out all reviews related to these nine android products and made dataframe of each company separately. Each data frame includes semantic value and review text itself. I applied the same sentiment analyzer on the test dataset and generated four columns of polarity value, positive value, negative value and neutral value against each review.

Next, I have applied same selected classifier (Random Forrest) to predict binary label for each company. Once I got binary labels for three android products of all three companies, I have just calculated the percentage of positive reviews. Since task requires to nominate the best company for investment, it's a non-statistical problem. I only sorted the best company on number of positive reviews ignoring neutral and negative reviews.

5- Results:

Feature Engineering:

Feature analysis part involves extracting the actual meaning, importance, correlations, dependencies and relations of the individual features with others. By knowing such information of each feature, data relevant features and information will be pushed to the predictive model to infer the logical information from the given data.

Sentiment Analysis:

The first attempt to understand this data is to measure score of the sentiment of each review, positive or negative. Further polarity, positive, negative, neutral count will be focused.

Neutral/Negative/Positive Score: Indicates the potency of these classes between 0 and 1. **Polarity Score**: Measures the difference between the Positive/Neutral/Negative values, where a positive number closer to 1 indicates overwhelming positivity and a negative number closer to -1 indicates overwhelming negativity.

From the results, It has been seen that most of the reviews got the positive sentiments same as the binary label. But on the contrast to the binary label, it has also been seen that, each positive review has high polarity value while low negative and positive values. However positive values are bit higher than negative values. The reviews with positive sentiment polarity score have increasing occurrence as the rating goes high. But for negative and neutral polarity score, the binary label is 0.

With ADD_1 company has highest percentage of positive reviews for its products, it is recommended as the most suitable company to be invested in.

6- Conclusions:

With the advancement of technological change, business processes specially e-commerce has been shifted from manual to internet-based solutions in order to increase the recurrent revenue. This paper focuses on the analysis of sales data of an e-commerce store in order to identify the hidden patterns of purchase history and to predict the scenarios where user preferred to recommend or not recommend the products. We found that ADD_1 has three androiid products B004NWLM8K, B004Q1NH4U, B004LPBTAA that have most positive reviews and are popular. Similarly, if a customer is giving low rating to a product and never recommended a certain item; actually, the customer is having complaints about that particular product. It turns out that, random forrest technique has the highest accuracy and precision value in predicting whether the product will have positive review or not based on customers reviews. With Vader Sentiment anyway, it was difficult to understand the exact meaning of the positive, neutral, and negative breakout values. The compound score is useful, but more research is needed.

References:

https://www.researchgate.net/publication/275828927_VADER_A_Parsimonious_Rule-based_Model_for_Sentiment_Analysis_of_Social_Media_Text