Amazon Sentiment Analysis

Summer 2020

By Jad Mark Lahoud

Ryerson University

Table of Contents

Abstract: 5

Project overview: 6

Literature review: 7

Data set: 8

Overview: 8

Data exploration: 9

Data Visualization: 9

Approach: 12

1. Overview: 12
2. Data Collection: 12
3. Data exploration & visualisation: 12
4. Text Pre-processing: 13
5. Feature Extraction: 14
6. Classifiers: 14
7. Output: 14

Initial Code: 15

Model Evaluation and validation: 15

Conclusion 19

Bibliography 20

Abstract:

Sentiment analysis is often used to derive the emotion / opinion expressed in a text. It has wide applications, including analysis of product reviews, tweets, Facebook comments, discovering a brand's presence online and people's opinion on a subject online. Still, it is hard to implement sentiment analysis by machine learning because of the nature of the human language, such as metaphors, comparison, irony, jokes and exaggeration.

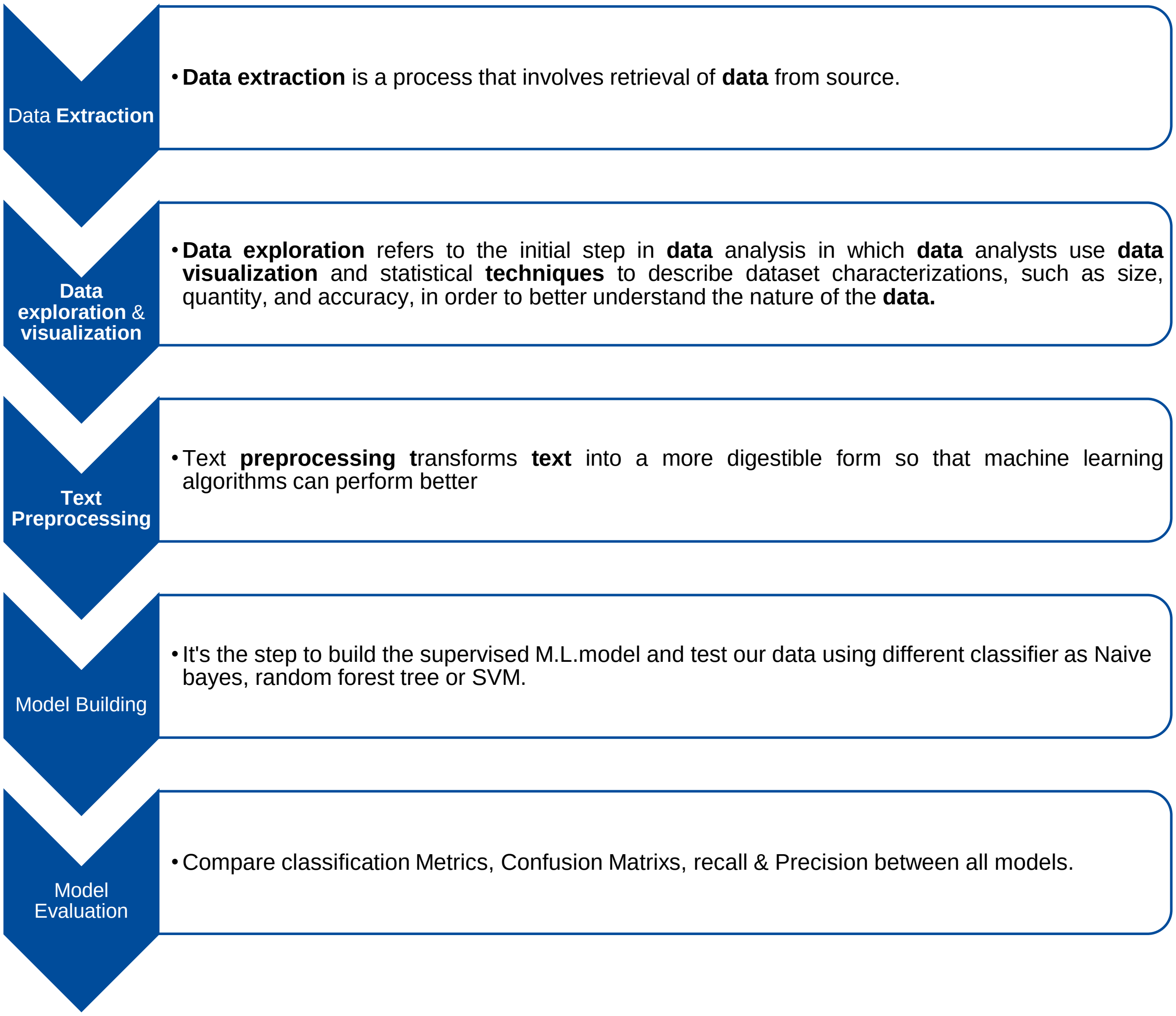
The objective of this project is to perform a sentiment analysis on Amazon unlocked cellphones reviews using Natural Language Processing (NLP) techniques and classify the reviews based on their emotional content as positive negative or neutral. The performance of the algorithm is then analyzed.

The dataset is publicly available on Kaggle (Amazon Reviews: Unlocked Mobile Phones, 2016) it consist of 400k reviews of unlocked mobile phones sold on Amazon.com.

Solution to the problem would be useful for a brand to gain a broader sense of their user's reviews towards a product through online reviews.

Project overview:

Sentiment Analysis has proved to be a key tool for companies in order to know their customers opinion and trends. By deriving high quality information from large volumes of text (reviews/comments), it is possible to understand how the market preferences evolve beyond sales statistics & KPI’s. For this reason, the goal of this capstone project is to perform an accurate sentiment analysis on Amazon unlocked mobile reviews. This analysis will help in determining whether the author’s intent is positive of negative. We will achieve the result in multiple steps. First by preprocessing the reviews and converting them to clean reviews by cleaning special characters, removing punctuation, stemming… Secondly the clean words will be converted into numerical representation. And finally, last step is to input those numerical representation of reviews to the Naive Bayes, random forest and Support vector machine (SVM) supervised machine learning model. The results and performance of all those classifiers will be compared in this project. The solution will help amazon to understand the general view toward their unlocked cellphone sales.



Data **Extraction**

- **Data extraction** is a process that involves retrieval of **data** from source.

Data exploration & **visualization**

- **Data exploration** refers to the initial step in **data** analysis in which **data** analysts use **data visualization** and statistical **techniques** to describe dataset characterizations, such as size, quantity, and accuracy, in order to better understand the nature of the **data.**

Text Preprocessing

- Text **preprocessing** **t**ransforms **text** into a more digestible form so that machine learning algorithms can perform better

Model Building

- It's the step to build the supervised M.L.model and test our data using different classifier as Naive bayes, random forest tree or SVM.

Model Evaluation

- Compare classification Metrics, Confusion Matrixs, recall & Precision between all models.

Literature review:

For this project I have referred to multiple articles ,books and websites to help me conduct the sentiment analysis on Amazon unlocked mobile reviews data set and to guide me step by step to successfully achieve the analysis.

1. Steven Bird, Ewan Klein, and Edward Loper (2009) Natural Language Processing with Python --- Analyzing Text with the Natural Language Toolkit, 2010 edn., : O'Reilly Media (Steven Bird)
   This book offers an introduction to NLP, a step by step guide on how to extract, pre-process, categorize, analyze and build features.

2. http://www.nltk.org/howto/ **NLTK** is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum. (How to use Natural language tool Kit, 2020). This website has a how to section and glossary that I will be referring to while doing the text pre-processing phase.

3. https://scikit-learn.org/stable/getting_started.html (Sklearn- Getting Started, 2020) The purpose of this guide is to illustrate some of the main features that scikit learn provides. It assumes a very basic working knowledge of machine learning practices (model fitting, predicting, cross-validation, etc.). I will be referring to this website on how to create my model and convert my text to numerical representation.

Data set:

Overview:
The Amazon unlocked cellphone mobile review dataset is publicly available on Kaggle. The data comes in a Comma separate value file. The data set is six attributes and 413,840 records. For this capstone I will be using google Collab and python to analyze the data. The data schema is:

Attribute	Attribute Description	Attribute Type
Product Name	The name of the Product. Example :Apple iPhone 6 - Unlocked (Silver) 16GB	Object
Brand Name	Name of the parent company. e.g. Samsung, Apple	Object
Price	Price of the product. (Max: 2598, Min: 1.73, Mean: 226.86)	Float64
Rating	Rating of the product ranging between 1-5 (Max:5 , Min:1 , Mean:3.81)	Int64
Reviews	Description of the user experience	Object
Review Votes	Number of people voted the review (Min: 0, Max: 645, Mean: 1.50)	Float64

Summary statistics of numerical features:

	Price	Rating	Review Votes
Count	407,907.00	413,840.00	401,544.00
Mean	226.86	3.81	1.50
std	273.00	1.54	9.16
Min	1.73	1.00	0.00
25%	79.99	3.00	0.00
50%	144.71	5.00	0.00
75%	269.99	5.00	1.00
Max	2,598.00	5.00	645.00

The chart above gives us a clear view of the statistical summary of the numerical features of the dataset.

The prices of the products go from a low of $1.73 to a mac of $2,598. The average price is $226.86 with an average rating of 3.81 stars.

Data exploration:

All queries and codes for the following calculation will be shared on my GitHub account:

Total number of reviews:413,840

Total number of brands: 385

Total number of unique products: 4,410

In reference to the rating 1 to 5, we consider that a rating below 3 is negative, above is positive and equal to 3 as neutral.

Percentage of reviews with neutral sentiment: 7.68%

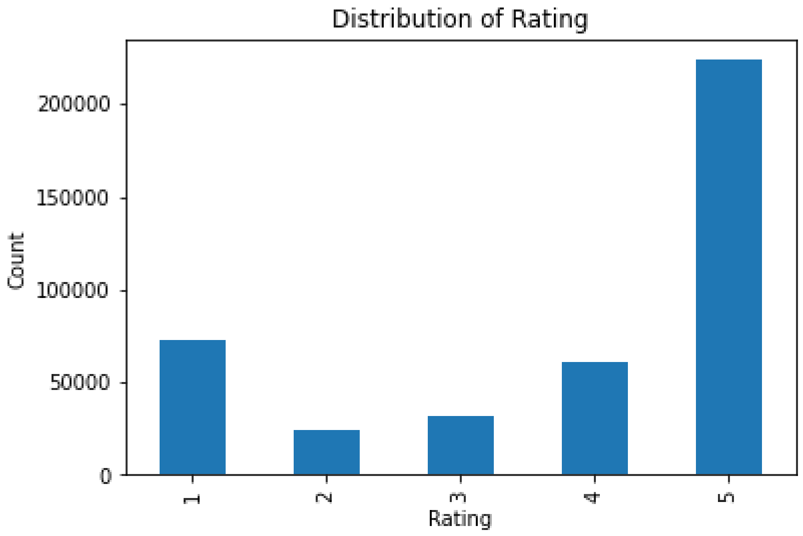
Percentage of reviews with positive sentiment: 68.86%

Percentage of reviews with negative sentiment: 23.45%

Data Visualization:

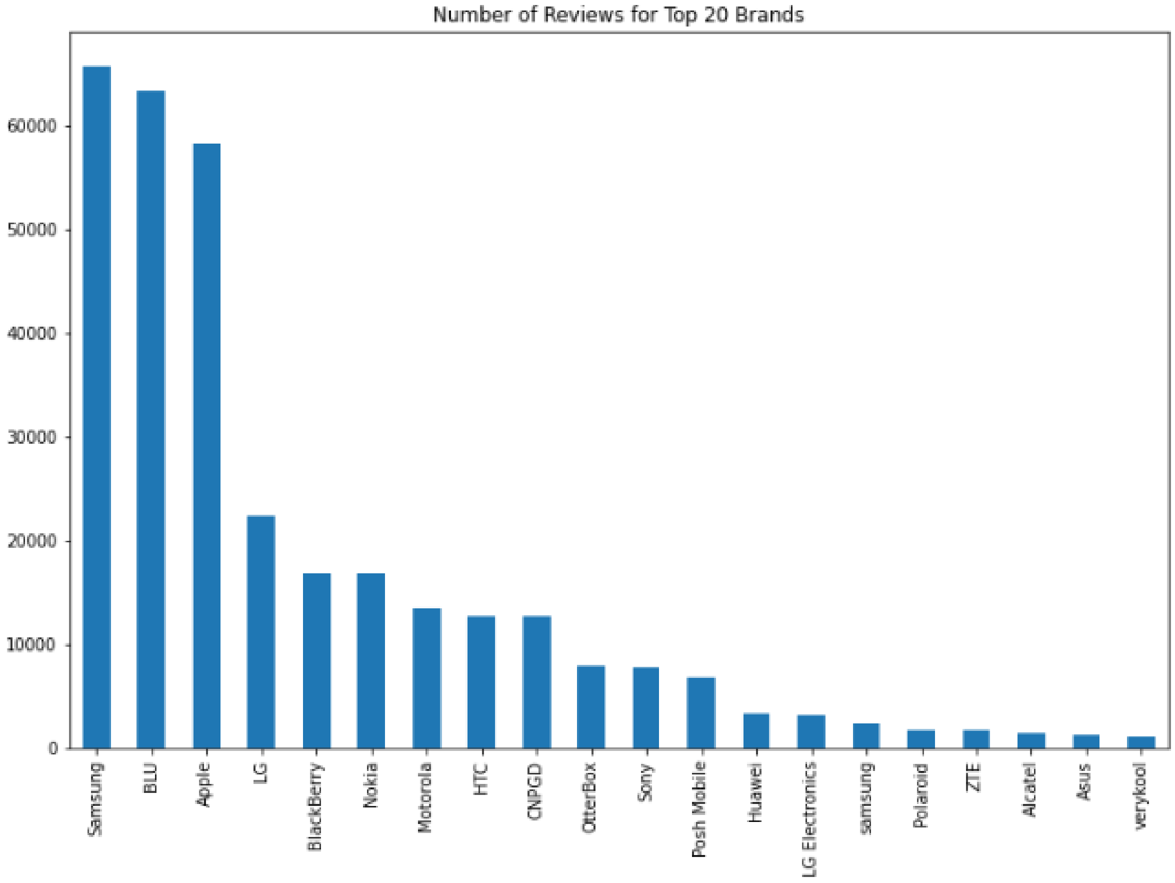
Distribution of Rating:

Rating	# of reviews
1	72,351
2	24,729
3	31,766
4	61,393
5	223,606



The highest count of reviews is for the 5-star rating and the lowest is for the 2-star rating.

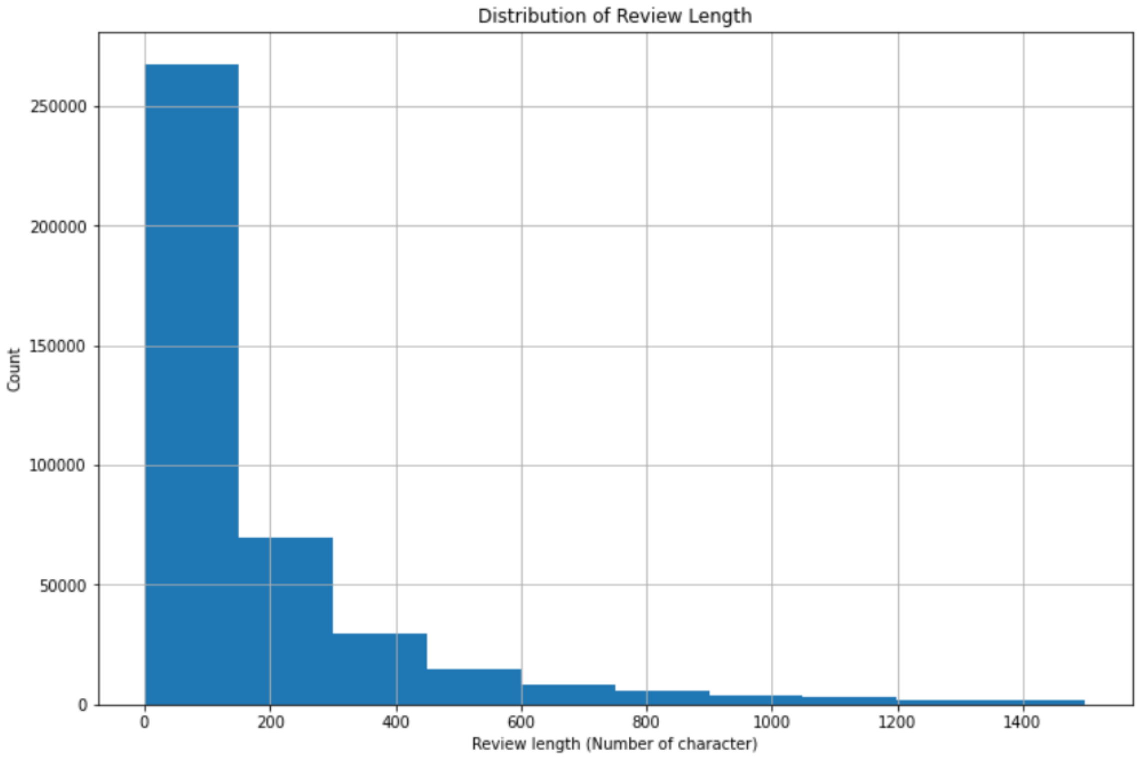
Number of Reviews for Top 20 Brands



The highest number of reviews is for Samsung and the least number of reviews are for the products Very Kool.

Distribution of Review length:

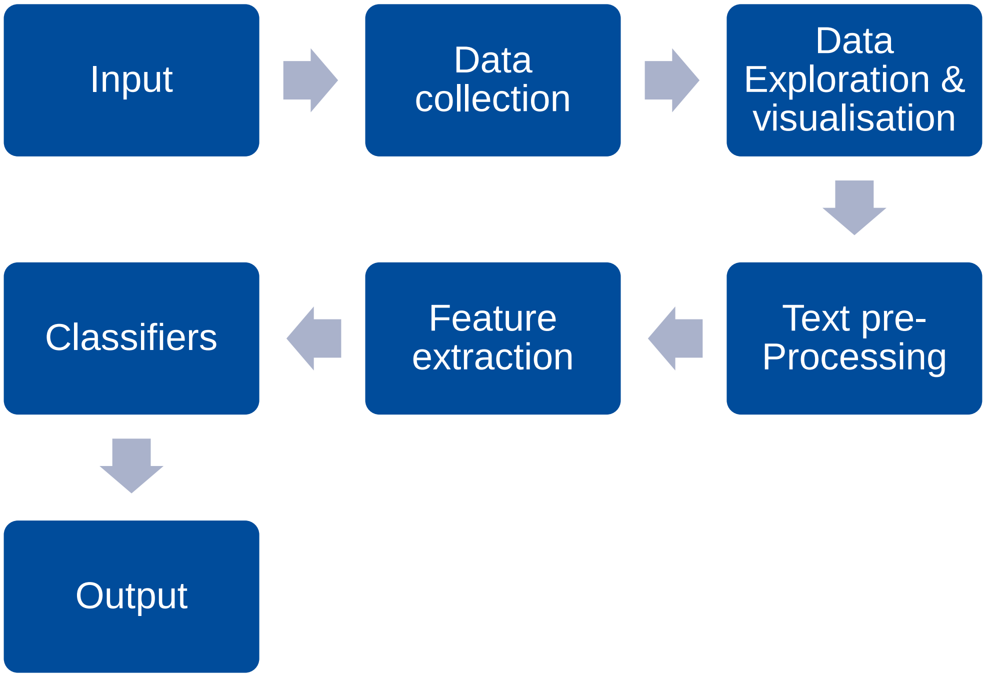
The following chart shows the number of reviews falls exponentially with the increase of review length.



Approach:

To conduct the sentiment analysis for the amazon unlocked mobile review our approach will be the following:

1. **Overview:**



2. **Data Collection:**

Data collection is the process of gathering and measuring data information or any variables of interest in a standardized and established manner that enables the collector to answer or test hypothesis and evaluate outcomes of the particular collection. (techopedia, 2018).

For our capstone project we have chose the unlocked mobile phones amazon reviews that contains 400,000 reviews publicly available on Kaggle. The data set has the following attributes Product name, brand name, price, rating, reviews and review votes.

3. **Data exploration & visualisation:**

Data exploration is the initial step in data analysis, where users explore a large data set to uncover initial patterns, characteristics and point of interest. This process helps to create a broad picture of important trends and major points to conduct the data analysis. (data exploration, 2020)

For our capstone project, our data exploration is based on statistical calculation mean, std, Min , Max, and quartile 25 50 & 75 for all the numerical attributes of the dataset. Those calculation helps us to employ the correct analysis and effectively present the results.

Data visualization is the graphical representation of information and data. By using visual elements like charts and graphs, data visualization provide an accessible way to see and understand trends, outliers, and patterns in data. (Data visualization, 2019).

For our capstone project, we have created two charts distribution of ratings and numbers of reviews for the top 20 brands. The two charts helped us to understand more in depth the data set.

4. Text Pre-processing:

Text preprocessing is traditionally an important step for natural language processing (NLP) tasks. It transforms text into a more digestible form so that machine learning algorithms can perform better. (NLP Text Preprocessing: A Practical Guide and Template, 2019)

In general, there is three main components for text pre-processing:

- **Tokenization**: is the process of breaking up a given text into units called tokens
- **Normalization**: is the process that converts a list of words, tokens, to a more uniform sequence. In general, the two types of normalization are **Stemming** and **Lemmatization**.
- **Noise removal**: is an important step in text pre-processing. It is devoted to stripping text of formatting.

For this capstone project I will start by using only the data that has a positive (4-5 start rating) and negative review (1-2 star rating) and drop neutral reviews (3 star raiting). Before the text pre-processing, I will encode positive reviews as 1 and negative reviews as 0. After classifying positive and negative reviews, we need to find a word embedding to convert a text into a numerical representation to fit in the machine learning algorithms.

I will be using the Natural language tool kit. **NLTK** has a prebuild functions for natural language processing and computational linguistics. The following steps will be completed in the analysis:

- Remove html tags
- Remove non-character such as digits and symbol
- Convert all text to lower case
- Remove stop words such as “the” “and”
- Convert to root words by stemming
- Remove the word “**amazon**” from text because it would not add any value to our sentiment analysis.

5. Feature Extraction:

One common approach of word embedding is frequency-based embedding such as **Bag of word model**. Bag of word model learns a vocabulary list from a given corpus and represents each document based on some counting methods of words.

For this capstone project I will be using the scikit-learn **SKLearn** library. The feature will help us conduct our sentiment analysis is **TF-IDF** Term Frequency Inverse Document Frequency which is intended to reflect how important a word is to a document in a collection or corpus. Also included in the library are the feature of **Word2vec** creates vectors that are distributed numerical representations of word features, features such as the context of individual words.

6. Classifiers:

After we have numerical representation of the text data, we can input those representation in a supervised learning algorithm. For this project we are going to use the following algorithms:

1. **Naïve Bayes**: text classifier is based on the Bayes' Theorem with a strong assumption that a feature will always be independent of other features. This is done in order to predict the category of a given sample. Naïve Bayes calculates the probability of each category using Bayes theorem. The category having the highest probability becomes the output. (Naive Bayes Classifier From Scratch in Python, 2019)
2. **Random forest**: To use this classifier we need to use word2vec from our **SKlearn** library. The random forest operates by building up multiple decision trees at the time of training and it will output the class that is the mode of the classes or classification of each tree. (Using word2vec to Analyze News Headlines and Predict Article Success, 2019)
3. **Support Vector Machine**: This algorithm is mostly used in classification problems where each data item is plotted as a point in n-dimensional space where n represents number of features we have, with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding what differentiate the two classes .

7. Output:

The output of all classifier will be in a summary chart. I will be Comparing the classification Metrics, confusion Matrix, recall & Precision between all models

Initial Code:

I have shared the initial code and dataset on GitHub under my account laouds91.

https://github.com/laouds91

Data set is too big to be uploaded on GitHub, please find below link to dataset:

https://www.kaggle.com/PromptCloudHQ/amazon-reviews-unlocked-mobile-phones

Model Evaluation and validation:

The goal of this project is to classify the reviews into positive and negative sentiment. There are two main steps involved. First, we need to find a word embedding to convert a text into a numerical representation. Second, we fit the numerical representation of text to machine learning algorithms or deep machine learning.

For this project I used the Bag of Word BOW model, that learns the vocabulary list from a given corpus and represent each document on some counting methods of words.

After text preprocessing and having clean reviews I input my data in three machine learning algorithms. For the following model I will be comparing the following:

Accuracy on validation score that is the score to evaluate the models performance.

TN / True Negative: when a case was negative and predicted negative

TP / True Positive: when a case was positive and predicted positive

FN / False Negative: when a case was positive but predicted negative

FP / False Positive: when a case was negative but predicted positive

Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class it is defined as the ratio of true positives to the sum of true and false positives.

Precision – Accuracy of positive predictions.

Precision = TP/(TP + FP)

Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives

Recall: Fraction of positives that were correctly identified.

Recall = TP/(TP+FN)

The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

1. **Naïve Bayes:**

Naive Bayes text classifier is based on the Bayes's Theorem, which helps us compute the conditional probabilities of occurrence of two events based on the probabilities of occurrence of each individual event, encoding those probabilities is extremely useful.

```
Accuracy on validation set: 0.9184

AUC score: 0.8790

Classification report:
             precision    recall  f1-score   support

          0       0.87      0.80      0.83       778
          1       0.93      0.96      0.95      2311

avg / total       0.92      0.92      0.92      3089

Confusion Matrix :
 [[ 622  156]
 [  96 2215]]
```

2. **TFIDF with Logistic regression:**

Term Frequency (TF)

The number of times a word appears in a document divded by the total number of words in the document. Every document has its own term frequency.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

Inverse Data Frequency (IDF)

The log of the number of documents divided by the number of documents that contain the word w. Inverse data frequency determines the weight of rare words across all documents in the corpus.

$$idf(w) = log(\frac{N}{df_t})$$

Accuracy on validation set: 0.9310
AUC score: 0.8985

Classification report:

```
             precision    recall  f1-score   support

          0       0.89      0.83      0.86       778
          1       0.94      0.96      0.95      2311

avg / total       0.93      0.93      0.93      3089
```

```
Confusion Matrix:
 [[ 648  130]
 [  83 2228]]
```

I have implemented TFIDF Vectorizer with logistic regression and investigated the following results:

- Top 10 features with smallest coefficients:

['not' 'return' 'disappointed' 'waste' 'horrible' 'worst' 'poor' 'slow' 'stopped' 'doesn']

- Top 10 features with largest coefficients:

['great' 'love' 'excellent' 'perfect' 'good' 'easy' 'best' 'far' 'amazing' 'awesome']

This visualization helps us to justify a bit about how logistic regression learn classifying positive and negative sentiment in this setting.

3. Random Forest Classifier:

Random Forest is a tree-based machine learning algorithm that leverages the power of multiple decision trees for making decisions. As the name suggests, it is a “forest” of trees. Each node in the decision tree works on a random subset of features to calculate the output. The random forest then combines the output of individual decision trees to generate the final output.

Accuracy on validation set: 0.9226

AUC score: 0.8878

Classification report:

```
             precision    recall  f1-score   support

          0       0.87      0.82      0.84       778
          1       0.94      0.96      0.95      2311

avg / total       0.92      0.92      0.92      3089
```

Confusion Matrix:

```
 [[ 636  142]
 [  97 2214]]
```

Conclusion

This project has applied three different machine learning algorithms Naïve Bayes, TFIDF and Random forest on the Amazon reviews unlocked mobile dataset. The results from this exercise showed that in term of accuracy the TFIDF with logistic regression approach achieves better results than the naive Bayes or Random forest approach when the data set was used as a training and testing data set. We can further improve the final model by using more training data to give better performance and accuracy.

Bibliography

A Comprehensive Guide to Understand and Implement Text Classification in Python. (2018, april 23). Retrieved from analyticsvidhya: https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/

Amazon Reviews: Unlocked Mobile Phones. (2016, December). Retrieved from Kaggle: https://www.kaggle.com/

Buildwithpython. (2020, March 4). *Sentiment Analysis Python - 3 - Cleaning Text for Natural Language Processing (NLP)*. Retrieved from youtube: https://www.youtube.com/watch?v=lcgqP8g6i84

data exploration. (2020, January 10). Retrieved from sisense: https://www.sisense.com/glossary/data-exploration/

Data visualization. (2019, April 26). Retrieved from Tableau: https://www.tableau.com/learn/articles/data-visualization

Edureka. (2018, October 2018). *Natural Language Processing (NLP) & Text Mining Tutorial Using NLTK | NLP Training | Edureka*. Retrieved from youtube: https://www.youtube.com/watch?v=05ONoGfmKvA

Edureka, A. Q. (2018, August 8). *Edureka*. Retrieved from youtube: https://www.youtube.com/watch?v=O_B7XLfx0ic&t=309s

How to use Natural language tool Kit. (2020, June 09). Retrieved from NLTK- Natural language tool Kit: http://www.nltk.org/howto/

Mudduluru, S. (2018, December 5). *Sentiment Analysis to classify Amazon Product Reviews Using Supervised Classification Algorithms*. Retrieved from youtube: https://www.youtube.com/watch?v=VXt9SQx5eM0&t=147s

Naive Bayes Classifier From Scratch in Python. (2019, October 25). Retrieved from machinelearningmastery: https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/

NLP Text Preprocessing: A Practical Guide and Template. (2019, August 30). Retrieved from Towardsdatascience: https://towardsdatascience.com/nlp-text-preprocessing-a-practical-guide-and-template-d80874676e79

python, B. (2020, May 14). *Sentiment Analysis Python - 10 - Positive or Negative Sentiments | NLTK*. Retrieved from youtube: https://www.youtube.com/watch?v=HtYweBOCp7A

python, B. w. (2020, MArch 4). *Sentiment Analysis Python - 4 - Tokenization and Stop Words (NLP)*. Retrieved from youtube: https://www.youtube.com/watch?v=KrEhmADXTr8 sentdex. (2015, May 1). *Natural Language Processing With Python and NLTK p.1 Tokenizing words and Sentences*. Retrieved from sentdex: https://www.youtube.com/watch?v=FLZvOKSCkxY

Sklearn- Getting Started. (2020, June 9). Retrieved from scikit-learn: https://scikit-learn.org/stable/

Steven Bird, E. K.--.-A. (2009). *Natural Language Processing with Python.* O'Reilly Media.

techopedia. (2018, February 9). Retrieved from techopedia.com: https://www.techopedia.com/definition/30318/data-collection

tv, m. l. (2018, july 15). *https://www.youtube.com/watch?v=h-Tpb_blwb0*. Retrieved from youtube: https://www.youtube.com/watch?v=h-Tpb_blwb0

Understanding the Classification report through sklearn. (2018, july 07). Retrieved from muthu.co: https://muthu.co/understanding-the-classification-report-in-sklearn/

Using word2vec to Analyze News Headlines and Predict Article Success. (2019, March 3). Retrieved from Towards data Science: https://towardsdatascience.com/using-word2vec-to-analyze-news-headlines-and-predict-article-success-cdeda5f14751