
# Sentiment Analysis on Retail

# (Sample Experiment)

## Table of Contents

# 1. Introduction:

Sentiment Analysis on Retail deals with opinion of the customer on product whether he is positive or negative. The revenue of the business can be greatly improved by analyzing the opinions of the numerous customers. Their Negative opinions can be altered with the help of machine learning models thus predicting and improving the odds of the business success.

## 1.1 Problem Statement

In the era of digital competition to outstand and generating huge revenue in the domain of highly growing e-commerce business, constant monitoring of a customer mindset is very necessary to evaluate the degree of acceptance of one's product. This evaluation has to be very instant just after the product has reviewed.

## 1.2 Overview

After the keen observation of the above problem statement, one of the optimum solutions would be performing a sentimental analysis on the product reviews. Here we are using an Amazon mobile review dataset (you can find the dataset link in 2.2. External file) which contains model name, brand name, price (dollar $), reviews, ratings and review votes. The dataset contains 413841 records in which 345763 unique mobile model, 332 unique brands, the model prices which varies from 1.73$ to 2599$ and the mobile models are rated in between 1 to 5 star. From the dataset, we can observe that there are more number of mobile models received 5 star ratings and out of all the mobile brands 'Samsung' got more number of reviews on its models.

## 1.3 Analysis Flow:

The latest review on the product is given as an input to the Machine Learning model which is built to predict the sentiment through the Flask server. The Natural Language Processing is performed on the latest review and vectors are formed, these vectors help the Machine Learning model to predict the sentiment of the customer on product.

## 1.4 Pre-requisites:

- Basic Natural Language Processing knowledge.
- A brief knowledge NLTK and Gensim.
- Knowledge on Scikit-learn (Machine Learning models) (Random Forest).
- A basic knowledge on Matplotlib and Seaborn (Exploratory data analysis).
- A basic knowledge on Pandas, Numpy and Regular expressions.

# 2. Installation:

## 2.1 Software Requirements:

Use Anaconda prompt or command prompt to install the below packages.

### Data loading

*pip install pandas==0.23.4*

*pip install numpy==1.15.4*

### Cleansing

*pip install regex==2018.11.03*

*pip install BeautifulSoup==3.2.1*

*pip install nltk==3.3*

*pip install gensim==3.6.0*

### Exploratory Data Analysis

*pip install matplotlib==3.0.1*

*pip install seaborn==0.9.0*

*pip install wordcloud==1.5.0*

### Model

*pip install sklearn==0.0*

*pip install pickle5==0.0.4*

### API Call

*pip install Flask-RESTful==0.3.6*

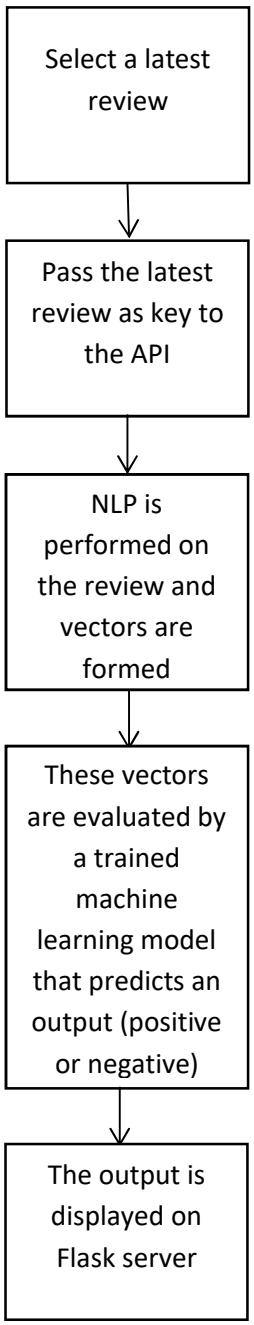
## 2.2 External Files:

- Amazon mobile review dataset with columns mobile name, mobile Brand, reviews, rating and review votes.
- Download link : "https://www.kaggle.com/PromptCloudHQ/amazon-reviews-unlocked-mobile-phones"

## 3. Configuration:

- Create a master folder with a name where the files related to this project can be stored.
- Create a folder inside the master folder with a name “amazon_sentiment_analysis” to store all the sentiment analysis code.
- Go to pickle_run.py and search for “pickle_model.pkl” and change the path location.

## 4. Design:



# 5. Development:

## 5.1 Phase-1 (DATA LOADING):

In phase-1, the required python packages have been installed and the Amazon mobile review dataset is loaded into the Pandas Dataframe which is used to train the Machine Learning model to perform sentiment analysis.

## 5.2 Phase-2 (DATA CLEANSING):

In phase-2, in order to perform sentiment analysis the required data should be in proper format. To remove the unnecessary tags, emoji's and stopwords each review in a dataset undergoes tokenization which converts the entire review into tokens (words). Now, Natural language processing techniques like stemming (Stemming is the process of converting the words of a sentence to its stem word) and lemmatization (Lemmatization is the process of converting the words of a sentence to its dictionary form) is performed. The Word2Vec package in Gensim will assign the vectors to the combined sentence (addition of vectors of each word).

## 5.3 Phase-3 (EXPLORATORY DATA ANALYSIS):

In phase-3, in statistics, exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. From the Bar plot on 'ratings vs rating_count' (Fig 1) more mobile models have got 5 star rating, the Bar plot on 'brands vs review_count' (Fig 2) Samsung brand got more reviews.

## 5.4 Phase-4 (MODEL BUILDING):

In phase-4, the entire data is divided into train and test data. Where, training data is to train the Machine learning model and testing data is to evaluate the trained model. The sentiment analysis on the mobile review dataset is performed by using machine learning algorithm 'Random Forest'. **Random Forest** Classifier is supervised classification/regression ensemble algorithm, Random forest classifier creates a set of decision trees from randomly selected subset of training set. Scikit-learn package in python provides Random Forest algorithm. By using Random Forest algorithm, we are predicting target variable (positive sentiment or negative sentiment of product) from the given review on the product.

## 5.5 Phase-5 (MODEL EVALUATION):

In phase-5, Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. The trained Random forest algorithm model is evaluated by using few evaluation techniques like Accuracy score, AUC_ROC_SCORE, Classification report, Confusion matrix.

## 5.6 Phase-6 (DEPLOYMENT):

In phase-6, the machine learning model is deployed on the Flask server by using Pickle. Pickle is a python package which is used for serializing and de-serializing a Python object structure. The

Random forest model which is trained by using training dataset is dumped into the pickle object by using dump () function. The further predictions of sentiment analysis are done by using pickle object. Firstly, the latest review is received from the Flask API undergoes sentiment analysis and predicts the result by using pickle object and sentiment is displayed by using Flask server.

# 6. Implementation:

## 6.1 Request :

**Objective:**

Perform a sentiment analysis on product reviews and Find the customer sentiment (either positive or negative) on product by its reviews.

**Process** :

- Receives latest review on product from the Flask server.
- Example: *127.0.0.1:5000/<latest review>*
- Performs NLP techniques.
- By using Word2Vec package each words in a sentence will be converted into a vector
- The machine learning model which is dumped into pickle predicts the sentiment (either positive or negative) on the latest review.
- The predicted sentiment is displayed in Flask server.

**Input** : Latest review on product

**Output** : Sentiment of the customer on product

## 6.2 Python Files:


run.py


sentiment_analysis_mobile_reviews.py


predict_review.py


pickle_run.py

- Ctrl+r(run window will pop up)
- cd (your path to amazon_sentiment_analysis)
- python sentiment_analysis_mobile_reviews.py
- search for pickle file and change the path to the pickle file in pickle_run.py
- cd (your path to master folder)
- python run.py
- You will get a link, copy that and paste in your browser.
- Add '/' '<Latest review>' Example: http://127.5.500x/latest_review

# 7. Visualization

## 7.1 Exploratory Data Analysis:

### 7.1.1 Bar Plot (Rating vs Rating_count)

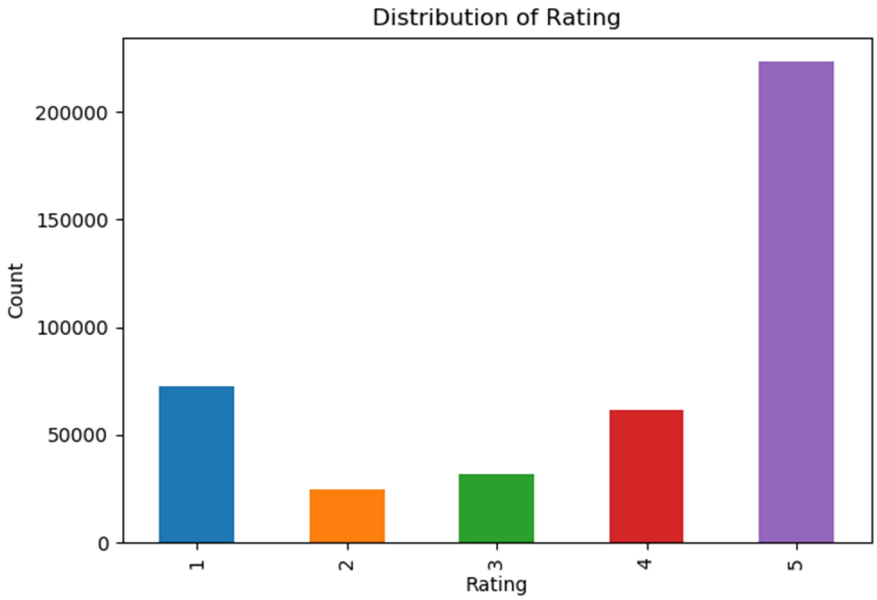



Fig 1: Bar Plot (Rating vs Rating_count)

### 7.1.2 Bar Plot (Brands vs Review_count)

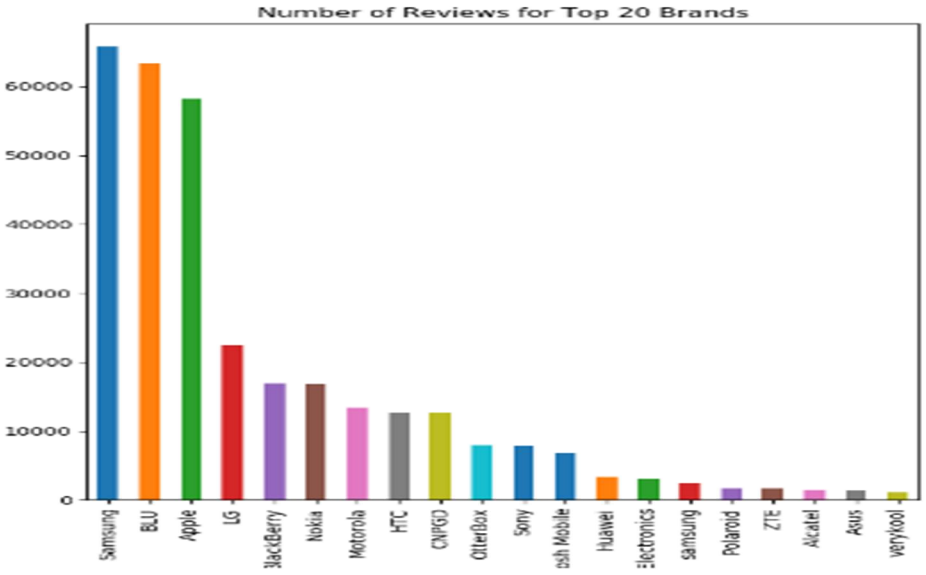


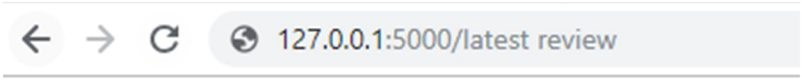

Fig 2: Bar Plot (Brands vs Review_count)

## 7.2 Execution:

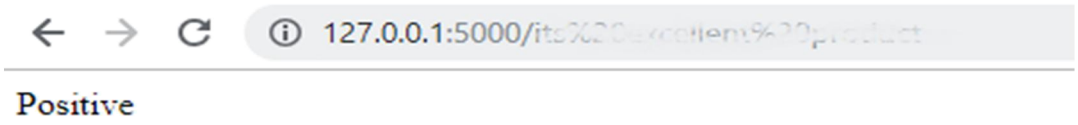
### 7.2.1 Step 1:

```
(      ) C:\Users\                              \amazon-mobile_review_sentiment-analysis>python ru
n.py
C:\Users\sagarvarma.gurajala\AppData\Local\conda\conda\envs\1stclass\lib\site-packages\gensim\uti
ls.py:1212: UserWarning: detected Windows; aliasing chunkize to chunkize_serial
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
 * Serving Flask app "run" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
C:\Users\                  \AppData\Local\conda\conda\envs\       \lib\site-packages\gensim\uti
ls.py:1212: UserWarning: detected Windows; aliasing chunkize to chunkize_serial
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
 * Debugger is active!
 * Debugger PIN: 891-836-460
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

### 7.2.2 Step 2:




### 7.2.3 Step 3:




Positive