**PROJECT**

**Sentiment Analysis on Reviews From Amazon**

**What is sentiment Analysis?**

Sentiment Analysis is the process of ‘computationally’ determining whether a piece of writing or data is positive, negative or neutral.

In marketing companies like Amazon, it is used to develop strategies, to understand customers’ opinion towards products or brand, how people respond to their products and why consumers don’t buy some products and many other related information.With the huge amount of reviews, creates an opportunity to see how people reacts to the product. I will predict the sentiment of the reviews using machine learning tools.

The dataset I will be using is from <http://deepyeti.ucsd.edu/jianmo/amazon/categoryFilesSmall/Cell_Phones_and_Accessories_5.json.gz>

**Quick look at the Data**

Cleaning the data and data processing was performed on ‘reviews.json’ file and importing the data as pandas DataFrame.

I have taken only those columns which were required further down the Analysis such as ‘overall’ and 'reviewText’. Dropping ‘style’, ‘vote’ and ‘image’ columns which are not required down for analysis.

And, also calculated the count of all the products ratings 1-5 from the overall column in the dataframe.

# **Split into Train/Test**

Here I have explored the dataset and then split data into training set and test sets. Goal is to finally train a sentiment analysis classifier

Since the majority of reviews are positive (5 stars), I opted to stratified split on the reviews score to ensure that I have balanced data during training the classifier.

Once Split is done, Checking whether the train or test sets were stratified proportionately.

# **Sentiment Analysis**

I have build classifiers to determine reviews sentiment using feature.

Segregated the ratings from 1-5 into positive, neutral, and negative.

I have used CountVectorizer to performs:

Text preprocessing: Tokenization breaking down the sentences into words

Stopwords : filtering the, in, are and many other words

Occurrence counting : building dictionary of features

Feature Vector : converts the dictionary of text into a feature vector

I see higher average count values on words that carry very little meaning, this will shorter documents that have lower average counts with same frequencies, as a result, we will use TfidfTransformer to reduce this redundancy:

Term Frequencies (Tf) divides the number of occurrences for each word by total number of words.

Term Frequencies times Inverse Document Frequency (Tfidf) decreases the weight of each word.

## **Methods**

Software used for this project is scikit-learn, an open source machine learning software package in Python. The classification models selected for categorization are: Naive Bayesian, Logistic Regression, and Support Vector Machine

**Fine tuning the Support Vector Machine Classifier**

I will run a Grid Search of the best parameters on a grid of possible values and also run the grid search with LinearSVC classifier pipeline, parameters and cpu core maximization

Then I will fit the grid search to our training data set and use final classifier (after fine-tuning) to test some arbitrary reviews.

Finally we will test the accuracy of our final classifier (after fine-tuning)

**Support Vector Machines** is best for classification by measuring extreme values to differentiate the worst cases so that it can easily classify between Positive, Neutral and Negative.

After testing with few arbitrary reviews results are performing correctly with Positive, Neutral, Negative results.

After running the grid search, our Support Vector Machine Classifier has improved its accuracy level.

## **Detailed Performance Analysis of Support Vector Machine Classifier**

## I have analyzed the best mean score of the grid search (classifier, parameters, CPU core), best estimator and the best parameter

* Precision: determines how many objects selected were correct
* Recall: tells you how many of the objects that should have been selected were actually selected
* F1 score measures the weights of recall and precision (1 means precision and recall are equally important, 0 otherwise)
* Support is the number of occurrences of each class

Finally, the reviews in this dataset are majorly positively rated according to the results achieved.