Coronavirus Tweet Sentiment Analysis

Gaurav Yogeshwar

ABSTRACT

Coronavirus disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus. Most people who fall sick with COVID-19 will experience mild to moderate symptoms and recover without special treatment. However, some will become seriously ill and require medical attention. Its measures caused a disruption to millions of people across the world, not only affecting people's normal day-to-day activities but also their mental health.

The worldwide coronavirus pandemic has led to the establishment of worldwide curfews, quarantines and lockdown to mitigate further spread of the virus. During this time, it can be helpful to track the public's responses to these changes. This study aims to answer the following questions:

1. How do people feel during the crisis?
2. How does the general public sentiment change over time?
3. What are the topics that most contribute to this sentiment shift?

Keywords: Sentiment Analysis, Classified labels, Exploratory data analysis, Feature engineering, Logistic regression, Stochastic Gradient Descent Classifier, Accuracy.

PROBLEM STATEMENT

This project addresses the problem of sentiment analysis and to build a classification model to predict the sentiment of COVID-19.

That is, classifying tweets according to the sentiment expressed in them: extremely positive, positive, extremely negative, negative or neutral.

INTEGRAL METHODOLOGY

The entire Analysis is divided into the following phases: Dataset Description, Breakdown of Datasets, Examining the null values, Data Cleaning, pre-processing and Feature engineering followed by Exploratory Data Analysis by and applying different models. First, we collect the data from Alma's better dashboard. Thereafter we did basic data cleaning and data visualization. After visualizing the data set, we removed some unnecessary features and made it ready for analyzing the data set using different plots. Next, we conduct data modeling by using Bar plot graphs, violin plots, histogram, etc. After that we will tokenize the word and convert it into a bag of words so that it can be passed to the machine learning model. Finally, we build a classifier that can predict the sentiment of a tweet.

DATASET DESCRIPTION

Let's take a look at the data, which consists of Coronavirus Tweets.csv. There are 6 columns that give information about users and their tweets.

About Dataset Most regularly a dataset relates to the matter of the single database table, or the single factual information framework, where each segment of the table speaks to a specific variable, and each column compares to a given individual from the informational collection being referred to. In this project, I have analyzed all these various columns of the dataset.

Column	Explanation
Location	The place where it was tweeted, like cities , states & countries.
Tweet At	Date and time of tweet.
Original Tweet	Tweet of user.
Label	Sentiment of tweet.

The names and usernames have been given codes to avoid any privacy concerns.

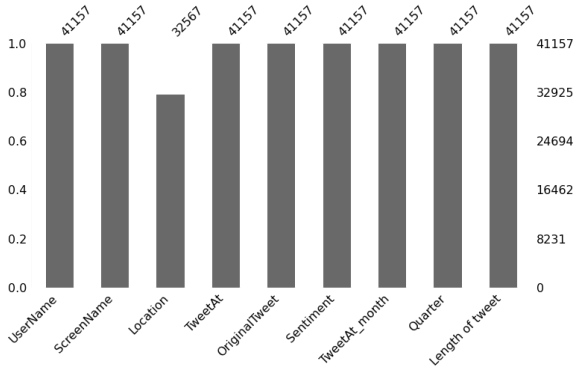
BREAKDOWN OF DATASETS

In order to go ahead for data visualization upon key factors we need to go for certain extra steps before proceeding to the main segment. In this part we are going with the following steps:

1. Mounting drive and installing a few libraries.
2. Importing Analytical necessary library classes for future analysis.
3. Reading the csv data file from Google drive, using latin encoding.
4. Visualizing all the columns of the respective Data frame.
5. Viewing all data information.
6. Checking the Unique values in the columns and duplicate rows in the dataframe.
7. Checking for null and missing values.
8. Combination of some feature to get new features
9. Get a quick statistical summary of the numeric columns.
10. Converting the data types to similar objects as the Analysis Demands.
11. Eradicating special characters from the dataset columns.

Steps involved:

1.Examining Null Values



The most critical thing from which we can draw some observations is Dataset, however data comes with unexpected values too i.e. sometimes it may be Null or missing in other words the space might be blank.

Location columns have almost 20 percent nan values , but the good thing is that we don't need this feature for tweet sentiment.

Original tweet and sentiment, which is an essential feature for us, has no non-values.

2. Data Cleaning, Data Preprocessing and Feature Engineering.

It is the process of using domain knowledge to extract features from raw data via data mining technique.

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

There are Three general approaches:

- Extracting Information
- Combining Information
- Transforming Information

Extracting Information

Creating a new feature by extracting any hidden information from the given data, like month and quarter, and I created a new feature which is the length of the original tweet.

Combining Information

Creating a new feature by combining two or more features by some mathematical, logical or any other operation.

Transforming Information

Transform one type of data into different types of data that contain the same information. eg..

1. **One-hot encoding.**
2. **Ordinal / numeric encoding.**

Before performing any mathematical operation with the categorical data we have done a hot encoding on it.

Stop word

Stop words are the words in a stop list which are filtered out before or after processing of natural language data because they are insignificant (such as “the”, “to”, “an”, “in”).

Punctuation

Punctuation is the use of spacing, conventional signs, and certain typographical devices as aids to the understanding and correct reading of written text, whether read silently or aloud.

Stemming operations

Stemming operation bundles together words of the same root. e.g. stem operation bundles "response" and "respond" into a common "respon"

Extracting Features From Text Data (Feature Selection)

Most classic machine learning algorithms can’t take raw data.

Instead we need to perform a feature “extraction” from the raw text in order to pass numerical features to the machine learning algorithm so that our model can understand it.

There are two main method for Feature Extraction:

Count Vectorization

tf-idf (Term Frequency - Inverse Document Frequency)

Count Vectorization treats every word as a feature, with the frequency counts acting as a "strength" of the feature/word.

For large documents, matrices are stored as sparse matrices to save space, since so many values will be zero.

	Word 1 Count	Word 2 Count	...	Word M
Message 1	0	1	...	0
Message 2	0	0	...	0
...	1	2	...	0
Message N	0	1	...	1

Term frequency tf(t,d) is the raw count of a term in document:

★ The number of times that term t occurs in documents d

$$tf(t,d) = \frac{f_{t,d}}{\Sigma_{t' \in d} f_{t',d}}$$

where $f_{t,d}$ is the raw count of a term in a document, i.e., the number of times that term t occurs in document d.

An **inverse document frequency** factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

It is the logarithmically scaled inverse fraction of the document that contain the word n

$$idf(t,D) = log_e\left(\frac{N}{|\{d \in D : t \in d\}|}\right)$$

N: total number of documents in the corpus

$|\{d \in D : t \in d\}|$: number of documents where the term t appears.

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D)$$

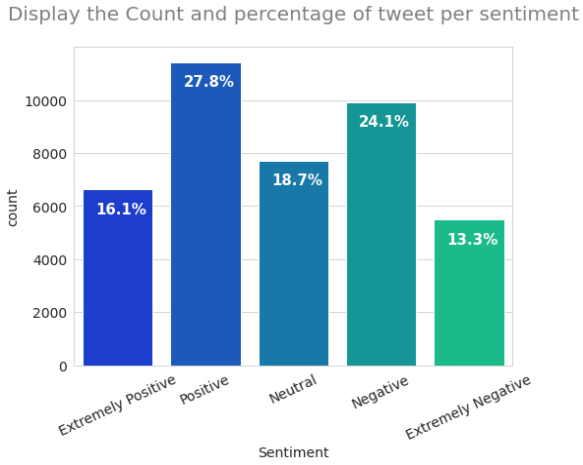
3. Exploratory Data Analysis

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypotheses and to check assumptions with the help of summary statistics and graphical representations.
First get a quick statistical summary of the numeric columns with .describe()
Username and screen name are unique identifiers.

Minimum length of tweet is 11, the length of a single tweet can not exceed 280, but here maximum length is 355.

I. Display the Count and percentage of tweet per sentiment

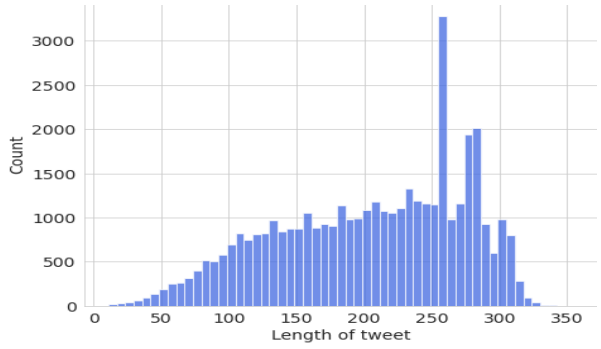
There's more tweets with a positive sentiment than a negative.



II. Histogram for Tweet Length

As histogram shows that the length of the tweets is negatively skewed.

The tallest tower is near 255.

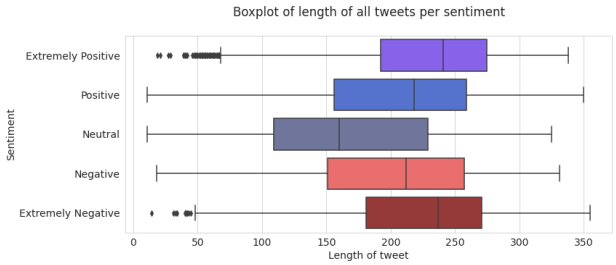


The length of :

Neutral sentiment tweets are positively skewed, rest are negatively skewed.

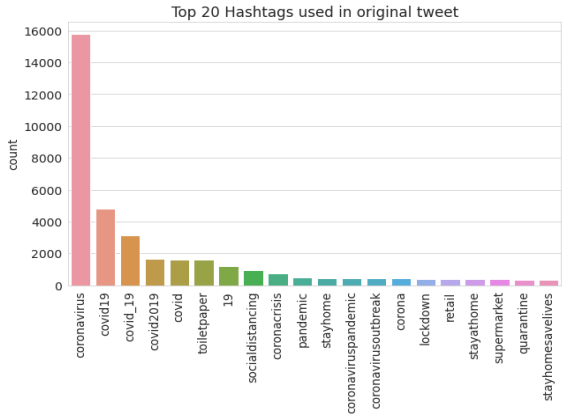
Extremely positive and extremely negative sentiments tweets follow the same distribution.

Positive and Extremely negative sentiments tweets follow the same distribution.



III. Hashtag

On Twitter, adding a "#" to the beginning of an unbroken word or phrase creates a hashtag.



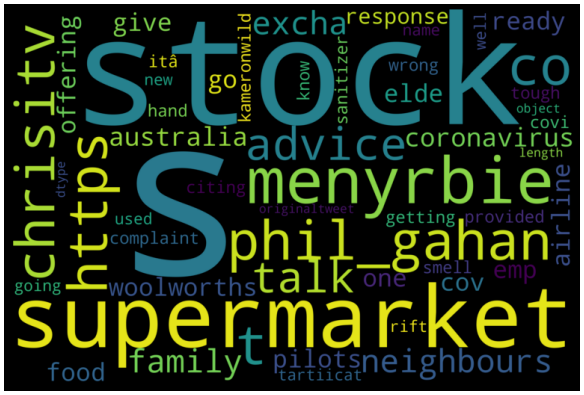
When you use a hashtag in a Tweet, it becomes linked to all of the other Tweets that include it. Including a hashtag gives your Tweet context and allows people to easily follow topics that they're interested in, it also helps in finding trending topics in twitter.

Most of the hashtags are about coronavirus outbreak and pandemic, Social distancing, lockdown , staying at home etc.

Due to the lockdown, people are also facing problems due to the closure of supermarkets, shortage of food, and running out of toilet paper.

IV. Word of cloud

let's describe a poster/hashtag for our storyline/data, which helps the audience to see some repetitive words of our story



The stock market has collapsed due to the pandemic, so people are talking about it.

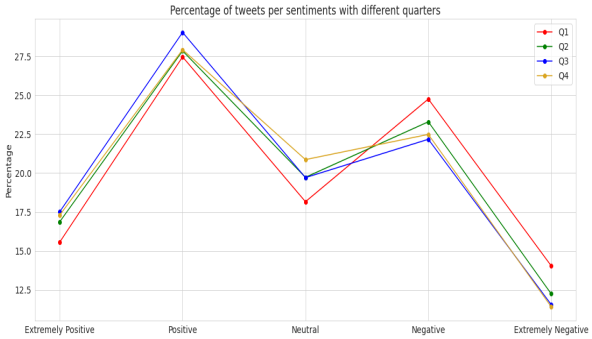
Having trouble buying from the supermarket

V. Tweets per Quarter

First quarter has the highest percentage of negative and extreme negative tweets.

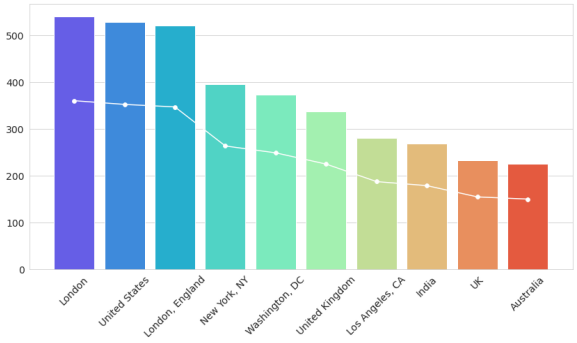
Third quarter has the highest percentage of positive and extreme positive tweets.

Fourth quarter has the highest percentage of Neutral tweets.



VI. Number of tweets per location

As the location suggests, most of the places are from English speaking countries or countries where people understand English, such as the UK, USA, India, Canada, Australia etc., and among these most of them are also from the US and UK.



4. Split the data set into training and testing parts

We split the dataset by 20% test size taking into account the stratification,

5. Fitting different models

For modeling we tried various classification algorithms like:

- ➔ Logistic Regression
- ➔ Linear SVC
- ➔ Multinomial NB Classifier
- ➔ SGD Classifier
- ➔ Decision Tree
- ➔ Random Forest

And the performance of logistic regression in that is the best, with 0.61 accuracy

We can see that since there are five levels the accuracy is not good so we will convert it to binary class.

I divided the sentiment feature into two parts, negative and non negative sentiments, negative means overall negative sentiment.

After retraining all the models, we found that now the accuracy has increased from **61 to 86.**

6. Tuning the hyperparameters for better accuracy

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting with SGDclassifier.

Model performance is best with

- ➢ L1 (lasso) regularization.
- ➢ Maximum iteration is 100.

After retraining all the models, we found that now the accuracy has increased from **86 to 87.5.**

7. Evaluation Metrics

Evaluation metrics are used to measure the quality of the statistical or machine learning model, there are many different types of evaluation metrics available to test a model. Such as

Accuracy , Precision, Recall , F1-Score, ROC-AUC etc.

8. Lime explanation for tweet sentiment

We have applied LIME to help us provide an explanation not only to end users but also to ourselves about how a specific NLP model works.

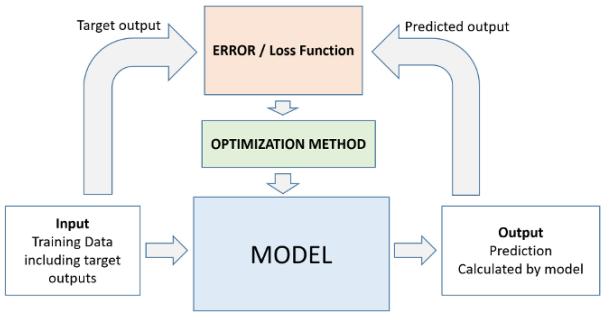
9. Model Persistence (Saving and Loading a Model)

Now we will save our model with the help of the joblib library so that we don't have to re-train the whole model to use it again later.

Machine Learning Workflow

Whether we're solving a regression problem or a classification problem, the workflow for training a model is exactly the same:

- ➢ We initialize a model with random parameters (weights & biases).
- ➢ We pass some inputs into the model to obtain predictions.
- ➢ We compare the model's predictions with the actual targets using the loss function.
- ➢ We use an optimization technique (like least squares, gradient descent etc.) to reduce the loss by adjusting the weights & biases of the model
- ➢ We repeat steps 1 to 4 till the predictions from the model are good enough.



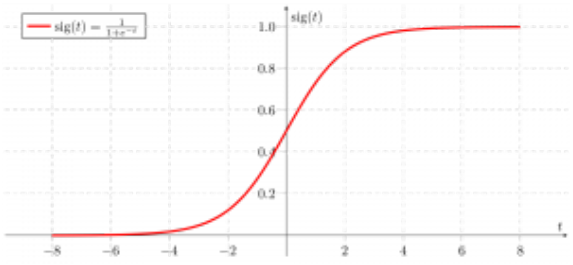
Algorithms

I. Logistic Regression

Logistic Regression is actually a classification algorithm that was given the name regression due to the fact that the mathematical formulation is very similar to linear regression.

The function used in Logistic Regression is sigmoid function or the logistic function given by:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



We need to put predicted $\hat{y}$ in place of x. Just like linear regression

$$\hat{y} = \sigma\left(\sum_{i=0}^{n} \beta_i x_i\right) = \sigma\left(\beta_0 x_0 + \ldots\ldots + \beta_n x_n\right)$$

$$\hat{y} = \frac{1}{1 + exp\left(-\sum_{i=0}^{n} \beta_i x_i\right)}$$

By rearenging it

$$ln\left(\frac{\hat{y}}{1-\hat{y}}\right) = \sum_{i=0}^{n} \beta_i x_i$$

LHS is log of odds, so

- Positive β indicates an increase in liklihood of belonging to 1 class with increases in associated x features.
- Negative indicate an decrease in likelihood of belonging to 1 class with increase in associated x feature

The optimization algorithm used is: Maximum Log Likelihood, we seek to minimize the following log loss

$$J(\mathbf{x}) = -\frac{1}{m}\sum_{j=1}^{m} y^j \log\left(\hat{y}^j\right) + (1-y^j)\log\left(1-\hat{y}^j\right)$$

$$J(\mathbf{x}) = -\frac{1}{m}\sum_{j=1}^{m}\left(y^j \log\left(\frac{1}{1+e^{-\sum_{i=0}^{n}\beta_i x_i^j}}\right) + (1-y^j)\log\left(1-\frac{1}{1+e^{-\sum_{i=0}^{n}\beta_i x_i^j}}\right)\right)$$

Here m is the number of training examples.

II. Regularization

Regularization seek to solve few common model issues by:

- Minimizing model complexity.
- Penalizing the loss function.
- Reducing model overfitting and variance.
- Requires some additional bias.
- Requires a search for optimal penalty hyperparameters.

There are three main type of Regularization;

1. L1 Regularization (LASSO)
2. L2 Regularization (Ridge)
3. Combining L1 and L2 (Elastic Net)

L1 regularization adds a penalty equal to the absolute value of magnitude of coefficents.

Penalty term is , $\lambda \sum_{i=1}^{n} |\beta_i|$

- ➔ It Limits the size of the coefficients.
- ➔ Can yield a sparse model where some coefficient can zero.

L2 regularization adds a penalty equal to the square value of magnitude of coefficents.

- ➔ All coefficients are shrunk by the same factor.
- ➔ Does not necessarily eliminate coefficients.

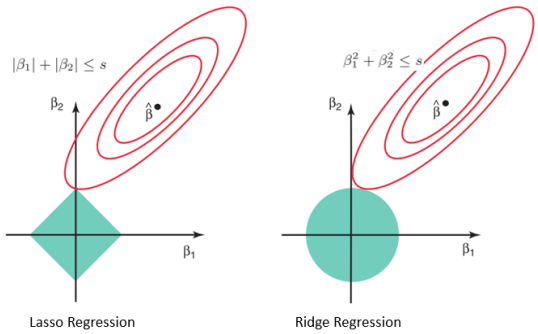
Penalty term is , $\lambda \sum_{i=1}^{n} \beta_i^{\,2}$

Elastic net combine L1 and L2 with the addition of an alpha parameter deciding the ratio between them:

$$\lambda\left(\frac{1-\alpha}{2}\sum_{i=1}^{n}\beta_i^{\,2} + \alpha\sum_{i=1}^{n}|\beta_i|\right)$$

λ decide the strength of the penalty.
α lies between 0 to 1, it decides the ratio of L1 to L2 penalty.

L1 constraints the sum of absolute values.
L2 constraints the sum of squared values.

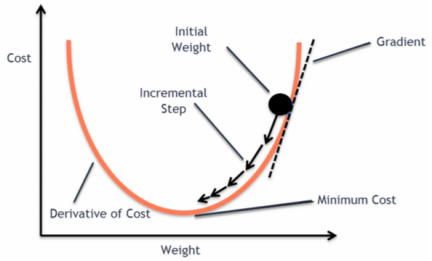


For the case of two features,

- ➔ Lasso Penalty $|\beta_1| + |\beta_2| \le s$
- ➔ Ridge Penalty $\beta_1^{\,2} + \beta_2^{\,2} \le s$

III. Information on SGD Classifiers

First of all let's talk about Gradient descent in general.



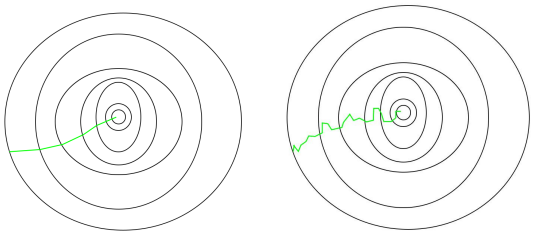
In a nutshell gradient descent is used to minimize a cost function. There are three well known types of gradient descent:

- ➔ Batch gradient descent
- ➔ Stochastic gradient descent
- ➔ Mini-batch gradient descent

Batch gradient descent computes the gradient using the whole dataset to find the minimum located in its basin of attraction.

Stochastic gradient descent (SGD) computes the gradient using a single sample.

Mini-batch gradient descent finally takes the best of both worlds and performs an update for every mini-batch of n training examples.



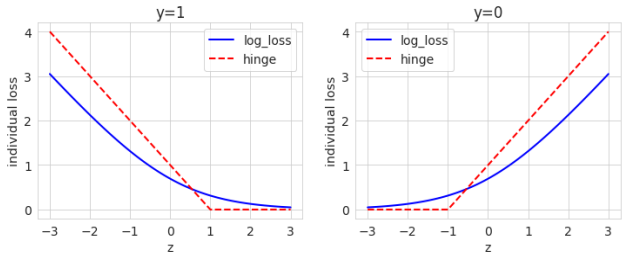
Left fig is path taken by batch gradient descent
Right fig is path taken by Stochastic gradient descent

Loss Functions

In machine learning it is common to formulate the classification task as a minimization problem over a given loss function.

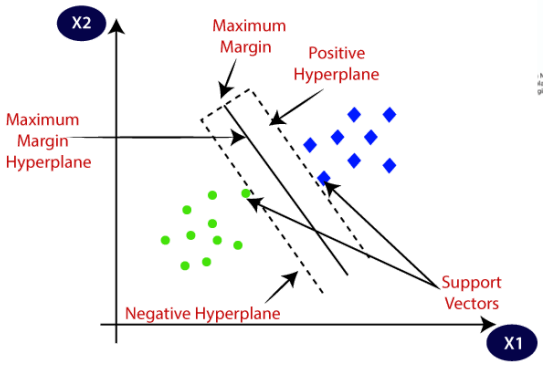
$$L(\hat{y}, y) = \frac{-1}{m}\left[\sum_{j=1}^{m}\left(y^{(j)} log(\hat{y}^{(j)}) + (1 - y^{(j)}) log(1 - \hat{y}^{(j)})\right)\right]$$

In hinge loss we replace the individual loss $-log\left(\hat{y}^{(j)}\right)$ with $max\left(0, 1 - \theta^{T} \cdot x^{(j)}\right)$, and $-log\left(1 - \hat{y}^{(j)}\right)$ with $max\left(0, 1 + \theta^{T} \cdot x^{(j)}\right)$.



IV. Linear Support Vector Classifier

In the SVM algorithm, the goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future.



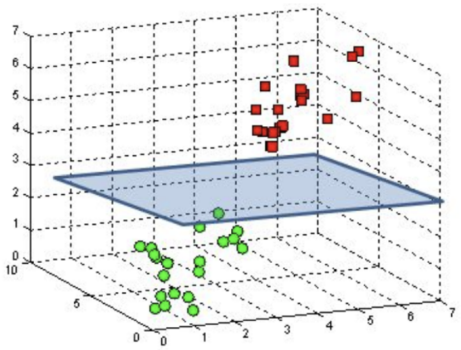
This best decision boundary is called a hyperplane which maximizes the margin and Define as,

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n = 0$$

In SVM we use the optimization algorithm as:

$$\min_{\xi, w, b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$
$$\text{s.t} \quad y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi_i,$$
$$\xi_i \geq 0;\ i = 1, \ldots, m.$$

Where C is cost parameter and ξ_i are slack variables.



We use hinge loss to deal with the noise when the data isn't linearly separable.
Kernel functions can be used to map data to higher dimensions when there is inherent non linearity.

Model performance:

Model can be evaluated by various metrics such as:

I. Confusion Matrix-

The confusion matrix is a table that summarizes how successful the classification modelis at predicting examples belonging to various classes. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label.

II. Accuracy

Accuracy is given by the number of correctly classified examples divided by the total number of classified examples. In terms of the confusion matrix, it is given by:

$$\frac{TP+TN}{TP+TN+FP+FN}$$

III. Precision, Recall and F1 Score-

Accuracy can be a useful measure if we have a similar balance in the dataset , but it does not hold good for imbalanced data.

Precision is the ratio of correct positive predictions to the overall number of positive predictions :

$$\text{precision} = \frac{TP}{TP+FP}$$

Recall or sensitivity of the model is the ratio of correct positive predictions to the overall number of positive examples in the set:

$$\text{recall} = \frac{TP}{TP+FN}$$

There is typically trade of between precision and recall, F1 score is define as harmonic mean of precision and recall

$$\text{F} = \frac{2 \times precision \times recall}{precision + recall}$$

The harmonic mean (instead of normal mean) allows the entire harmonic mean to zero if either precision or recall ends up being zero.

IV. Area under ROC Curve(AUC)-

There can be a trade-off between True Positive and False Positive. Receiver Operator Characteristic (ROC) curves use a combination of the true positive rate (the proportion of positive examples predicted correctly, defined exactly as recall) and false positive rate (the proportion of negative examples predicted incorrectly) to build up a summary picture of the classification performance.

Hyper parameter tuning

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of

learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

Grid Search CV-Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model's performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

Conclusion:

That's it! We reached the end of our exercise.

Starting with loading the data so far we have done EDA , null values treatment, encoding of categorical columns, feature selection, split the dataset for training and testing purposes, text preprocessing, converting text into bags of words and then model building.

In all of these models our accuracy revolves in the range of 60 to 61%.

So we convert target label sentiment from multiclass to binary classification, and suddenly accuracy reached 85 %.

Then after Hyperparmeter tuning we get accuracy of 86.5 percent.

So the accuracy of our best model is 86.5% which can be said to be good for this large dataset.

References-

- An Introduction to Statistical Learning: With Applications in R
- GeeksforGeeks
- Analytics Vidhya
- Towards Data Science
- Almabetter notes