**Project Synopsis**

**Natural Language Processing:**

**Sentiment Analysis:** Perform Sentiment Analysis on tweets that you can fetch from Twitter API or use any dataset for the same.

* **What is Sentiment Analysis?**

Sentiment Analysis is the key to determine what each entry of a particular set of data indicates i.e. to determine the underlying emotion/attitude/opinion of a particular entry (review, comment, tweets, etc.).

* **How do we go about it?**

Considering the tweets, to put it in simple words, each tweet is comprised of a number of words, some words may carry a positive or negative emotion with them. Our job is to calculate the overall effect/sentiment about what the tweet is indicating – is the tone negative or positive? Is it appreciation or condemnation?

A basic approach can be:

Each positive word will have a positive score or a greater score/weight (if comparison is made between two positive words), similarly negative words might have a lesser score/weight or a negative score. To calculate the overall effect we will have to find the sum of the scores of all the words in the tweet and the result will be our answer!

**Learning the weights = Training a Classifier**

*However errors are prone to occur*:

For example: If you are evaluating on the basis of single words,

If a tweet contains the word ‘GOOD’, you will assign it a positive score!

But what if the tweet says ‘NOT GOOD’ ?

This creates a more complex scenario.

Predicted Label

|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Positive | True Positive | False  Negative |
| Negative | False Positive | True Negative |

True Label

Such Complex Scenarios need more Complex Models but on the brighter side, complex models have less Bias than the simple models!

To put it in a graph:

Single Word

Multiple words

Bias of model

Hence more complex the model, less the bias.

After this step, we calculate the probability of the correctness of the output label based upon the input sentence given by P(y|x). This probability can be calculated using the **NAÏVE BAYES THEOREM.**

**The classifier we now built is now known as NAÏVE BAYES CLASSIFIER.**

* **How do we program it?**

We divide our dataset into Train data and Test data.

* As the name suggests that the training data is used for experimenting or rather training a model or build the model.
* The test data is used to test/evaluate the model to check it’s validation.
* **Coding it!**

**Libraries :** NLTK (Natural Language Toolkit), pandas, Textblob.

Installing tweepy using pip!

**ALGORITHM:**

**Section A: Preparing The Test Set**

* Step A.1: Getting the authentication credentials
* Step A.2: Authenticating our Python script
* Step A.3: Creating the function to build the Test set

**Section B: Preparing The Training Set**

**Section C: Pre-processing Tweets in The Data Sets**

**Section D: Naive Bayes Classifier**

* Step D.1: Building the vocabulary
* Step D.2: Matching tweets against our vocabulary
* Step D.3: Building our feature vector
* Step D.4: Training the classifier

**Section E: Testing The Model**