1. TWITTER API – CONFIGURED AND TESTED
2. PULL TWEETS – VIA TWEEPY USING TWITTER API
3. DATA PREPROCESSING – REMOVING SPECIAL CHARS, PUNCTUATION MARKS, ETC.
4. COMPUTING SENTIMENTS – USING PYTHON TEXTBLOB, nltk sentiment vader.
5. In the last slide to visualize our sentiments result we plot a **donut chart** and a **word cloud.**

**Textblob-**

Offers a simple api to access its methods to perform a range of nlp tasks, like, sentiment analysis.

Tweet = “it is a good utube channel”

Tweet.sentiment

Sentiment (polarity=0.8, subjectivity=0.75)

* Polarity – lies in range of [-1,1]
  + 1 means a positive and -1 means a negative statement
  + In our example polarity is 0.8, which means that the statement is positive.
* Subjectivity – lies in range of [0,1]
  + 1 refers to personal opinion, emotion and a judgement (subjective) and 0 refers to factual information (objective).
  + In our example subjective of 0.75 means it’s a public opinion and not a factual information.
  + Factual statement as I live in india.

**Nltk –**

Sentiment\_scores(tweet)

Neg:0.165, neu:0.588, pos:0.247, compound:0.527

* Vader is basically valence aware dictionary and sentiment reasoner, which is specifically trained to sentiments in social media, so it is quite accurate
  + Vader not only tells about the positivity /negativity score but also tells how positive or negative a sentiment is
* Compound is a metric that is normalized value of all ratings and lies in the range -1 to 1 and based on these ranges, we establish whether the final sentiment is Positive, negative and a neutral.
* Negative (-1, -0.05), neutral (-0.05,0.05), positive (0.05,1)

1. **Get recent public tweets on a keyboard –**

We are using **tweepy paginator**, which allows us to make multiple calls to twitter api. With a single api call, we can get a maximum of 100 results.

So, we are making 10 back-to-back calls, so we get 1000 tweets.

Towards the end we our defining the empty tweet list where we are storing our tweets on #apple.

Then, we define this dataframe tweet\_list\_df where we put all the tweets from the twist\_list.

Now, going further, we will keep adding columns to this dataframe, so we have a single view of our complete analysis at any point in time.

1. **Data pre-processing –**

First, we are defining preprocess\_tweet function and then we are calling this function on our tweets with this next cell.

All cleaned tweets we are storing in this cleaned\_tweet list.

1. **Generate sentiment labels –**

As discussed above, we are calling a textblob and a nltk vader sentiment Intensity Analyzer here to compute sentiments and then from the vader compound value(comp), we are computing the sentiments with this logic and finally we concatenate all these number to our dataframe.

1. **Sentiment visualisation –**

Now, we will visualize this data and understand what the aggregated public sentiments are –

* First, we are creating these new data frames for all positive, negative and neutral.
* Then, we use this count value in column function to count how many positive/negative and neutral. So, we can work on them separately.
* And then we call this count\_value function on our sentiment column within our dataframe.

1. **Word cloud –** now to get just what people are talking about. We may create word clouds that give us an overall idea of what is being talked about.
2. **We have our word cloud template stored in drive/folder that is needed to store the word clouds.**

Now we may create a word cloud on all the 1000 tweets to explore what people are talking about overall or on a specific tweet i.e positive/negative/neutral

People are writing more when they have a negative segmentation and talk least when their sentiment is neutral.