**Week 3**

* Descriptive report on Gun Violence in USA. This might help in understanding data Interpretation.

<https://www.kaggle.com/erikbruin/gun-violence-in-the-us-interactive-maps-and-eda/report>

* This is the sentiment140 dataset. It contains 1,600,000 tweets extracted using the twitter api . The tweets have been annotated (0 = negative, 4 = positive) and they can be used to detect sentiment.

Content

It contains the following 6 fields:

1. target: the polarity of the tweet (*0* = negative, *2* = neutral, *4* = positive)
2. ids: The id of the tweet ( *2087*)
3. date: the date of the tweet (*Sat May 16 23:58:44 UTC 2009*)
4. flag: The query (*lyx*). If there is no query, then this value is NO\_QUERY.
5. user: the user that tweeted (*robotickilldozr*)
6. text: the text of the tweet (*Lyx is cool*)

<https://www.kaggle.com/kazanova/sentiment140>

* import pandas\_profiling

pandas\_profiling.ProfileReport(df)

* 1. <https://github.com/pmathur5k10/Hinglish-Offensive-Text-Classification> Using Transfer Learning to re-train existing hate speech model to Hinglish (Hindi-English mixture). Can we do similarly for Ganglish (Gang-english mixture)?  
  2. <https://hackernoon.com/fighting-for-acceptance-on-social-media-tackling-offensive-tweets-in-hinglish-ad10e0eef2b9> more readable version of the paper  
  3. <https://data.world/thomasrdavidson/hate-speech-and-offensive-language> dataset used to create the first model  
  4. <https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/notebooks> - some possible approaches to train first model. Why transfer learning? I think it will reduce the number of tweets we have to label here yet give us hopefully better result.
* Word2Vec + LSTMOne of the best model for classification 1.6M tweets, taken here: <https://www.kaggle.com/paoloripamonti/twitter-sentiment-analysis>  
  But it requires labeled data. So when the Chicago data will be labeled, we can try this approach. The attempt to use trained weigths to label Chicago data as positive and negative seems not works good.<https://colab.research.google.com/drive/1fwCWig1ZOgoT59ov3fl6gJzoPEydblTr#scrollTo=vQb4235LmXcd>
* Link Prediction Algo

<https://colab.research.google.com/drive/10HXpaXWmLbSwtCvouBlguVFIRqEeN58P#scrollTo=iP6eLDXMl1Z4>

Key points from paper to consider/new learning:   
**What is the link prediction algo and we use it?**  
Social networks are dynamic in nature, as they grow over time through the addition of new users, creation of new relationships and ending of some old relationships. This dynamic change forms the base of link prediction algorithms. These algorithms involve trying to understand the process of these dynamic changes and try to replicate them. The algorithms try to use different features to predict new relationships with maximum accuracy.

<https://pdfs.semanticscholar.org/bb5e/9530cd210474a2b7ac5b263c8eea888a740e.pdf>

<https://medium.com/@vgnshiyer/link-prediction-in-a-social-network-df230c3d85e6>

**NLP**

I just started going through these lectures (<https://www.fast.ai/2019/07/08/fastai-nlp/>) from [fast.ai](http://fast.ai/) to understand NLP better, and I came across the concept of topic modelling using SVD (Single Value Decomposition) and NMF (Non-negative Matrix Factorization), which I tried to apply to the tweets in our dataset to extract important topics. <https://colab.research.google.com/drive/1AmNBdHb3y-q3XJty8SQEEC9bnNS6llnu> contains the results of the following:  
1) NMF with Count Vectorizer  
2) NMF with TF-IDF Vectorizer  
3) Truncated SVD with Count Vectorizer  
4) Truncated SVD with TF-IDF Vectorizer