# **MLG Apprentice Capstone project - Humor detection**

## Swapnil P

## **Things accomplished:**

So far, I have been able to

* Scrape and clean the data required for project. Augment it with an additional humor corpus found online.
* Create a baseline model and evaluate different models for accuracy and recall metrics

A short description of all scripts is provided below.

## 00\_Scrape\_Following.ipynb

This script will scrape the twitter users that my favorite funny twitter acct(@AbbieEvansXO) herself follows. This is my hack to get a mostly correct list of funny accounts. (Otherwise, I'd have had to individually find them & then create a list of accounts to scrape - may be later)

## 01\_Scrape\_Tweets\_From\_Funny\_Accounts.ipynb

This script will scrape tweets from the 'following' list of twitter accounts that was output of 00\_Scrape\_Following.ipynb. A csv is created for tweets from every account and saved as username\_date.csv

## 02\_Append\_All\_Tweets\_Together.ipynb

This simple script will append all the individual csv files created by 01\_Scrape\_Tweets\_From\_Funny\_Accounts.ipynb. Those will be cleaned (using preprocessor library). Also, we filter out tweets that have RT (retweet) count < 50 (assumption here is that RT>=50 is mostly going to be a funny tweet). Final O/P saved in Tweets\_Combined.csv

## 03\_Read\_All\_Data.ipynb

This script reads the following inputs.

1. Our processed tweets i.e.  (O/P of 02\_Append\_All\_Tweets\_together.ipynb)
2. Readily available humor corpus(positive as well as negative class) - One-liners, Proverbs, Wikipedia sentences, Reuters news headlines. Found at [*https://github.com/CrowdTruth/Short-Text-Corpus-For-Humor-Detection*](https://github.com/CrowdTruth/Short-Text-Corpus-For-Humor-Detection)

It creates a final dataframe with (sentence, class) columns that we will use later for modelling.

## 04\_Baseline\_Models\_With\_Word2Vec\_Dimensions.ipynb

This script will create Word2Vec dimensions for every row in our dataset and then run a few baseline models.

1. First, we create word vectors for every row, then just average those word vectors to create sentence vectors.
2. Apply baseline models using following scoring metric and plot them for comparison

   -accuracy

   -recall (positive class i.e. jokes important not to miss)