This document describes all jupyter notebooks in their logical order i.e. the order in which they were created. functions.py is a collection of all custom functions that were reused across the project. Approach and results have been discussed in further details in KYC_project_report.pdf. All processed data files including vectorizers, trained models and tweets data can be found on my S3 bucket – 's3:\\apandey-capstone-data\' with the same names as referred to in notebooks. If you wish to download raw datasets as well, please follow the mentioned links for each dataset in this document.

1. data_collection.ipynb: This file reads datasets in their raw format and does an initial exploration of datasets.
    a. Sentiment 140 – 1.6 million observations of tweets labelled to denote polarity. Explored various columns of the dataset along with the tweet text column. Transformed existing labels to denote a binary classification problem where 1 indicates grievance and 0 indicates non grievance i.e. positive or neutral tweets. Extracted just text and label column, every other column was dropped. Raw data can be downloaded at http://help.sentiment140.com/for-students/
    b. Consumer complaints - Consumer complaints against financial products. Explored features, dropped null rows and created a label column with value 1 for all observations as the dataset comprises of complaints only. Dropped every column except for 'Consumer complaint narrative' i.e. complaint text and labels. . Raw data can be downloaded at https://www.kaggle.com/cfpb/us-consumer-finance-complaints
    c. Amazon Reviews – json file Amazon reviews spanning across 6 categories. Extracted review content ('Content') and rating ('Overall') from each category, stored in temporary Dataframe and compiled all observations together into a master DataFrame. Raw Dataset can be downloaded at http://sifaka.cs.uiuc.edu/~wang296/Data/index.html

2. Data_preprocessing.ipynb - Preprocessing like checking for nulls and duplicates and ensuring all datasets comprise of features that follow a standard project convention.
    a. Dropped duplicates from all dataset.
    b. Created labels for amazon reviews where observations with ratings lower than or equal to 2 were labelled as grievances i.e. 1 and every other observation was labelled as non-grievance i.e. 0.
    c. Renamed text column in each dataset to 'text'. By the end of this file there are only 2 columns in each dataset – 'text' and 'label'.

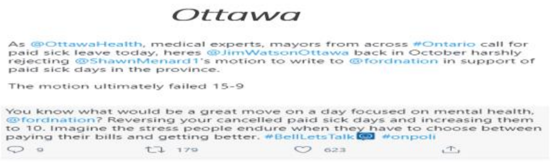
3. Data_preprocessing2.ipynb – Compiles observations from all datasets into one master dataset and performs cleaning to make dataset ready for training classification models.
    a. Created a dataset by 50,000 observations by extracting certain number of observations from each dataset.
    b. Data Cleaning – Data Cleaning steps are as below:

        i. Remove user handles - e.g. @broskiii OH SNAP YOU WORK......
        ii. Remove hashtags - e.g. #at&amp;t is complete fail
        iii. Remove urls.
        iv. Address contractions e.g. 'didn't' by expanding contractions via contraction map.
        v. Replace punctuations with whitespace.
        vi. Remove numbers.
        vii. Trim repeated letters in a word to successive repetitions e.g. ‘I am soooooooooo hungryyyyyyyyyyyyyyyyyy’ - ‘I am soo hungryy’.

4. Vectorization_fit_model.ipynb – Vectorizes training dataset created in previous step ‘Data_preprocessing2’ and fits intial unoptimized models on vectorized data.
    a. Split data into train and test.
    b. Created a custom tokenizer and vectorized data using CountVectorizer with min_df of 0.01 i.e. 1% and default n_gram range.
    c. Fit Logistic regression model on vectorized data and evaluated model performance by accuracy and top predictor words.
    d. Noticed that top predictor words were not generalized and biased towards product names from consumer complaints dataset. Created a list that comprised names of financial products with most mentions.
    e. Modified tokenizer to filter names of products and refit Countvcetorizer with an n_gram range of (1, 3).
    f. Refit Logistic Regression model and verified top predictor words again. Top predictor words were much more generic than earlier.
    g. Fit Decision Tree, Random Forest, Naïve Bayes, SVM Linear and SVM rbf.
    h. Compared models against each other on different metrics like Accuracy, Precision, Recall and F1 score.
    i. Models that stood out in comparison were Logistic Regression and random forest.
5. prototype_hyperparameter_tuning_sagemaker.ipynb – This notebook creates a bigger dataset from 3 datasets. Establishes baseline for both Logistic Regression and Random forest and prototypes Gridsearch code to be run on AWS Sagemaker.
    a. Created a dataset of 150,000 observations from 3 datasets.
    b. Fitted logistic regression and random forest models to establish baseline, for each model, that comprised of model metrics and top predictor words.
    c. Created parameter grid for each model and prototyped code on a sample, of 100 observations from master dataset, to tune hyperparameters to improve model’s Recall.
6. sagemaker_notebook_final.ipynb – This notebook takes the prototype code from previous step to tune hyperparameters for Logistic Regression and Random Forest. This notebook displays the results as run on aws sagemaker instance and no changes were made to it on local instance.

7. Post_sagemaker_evaluation – Evaluates optimized grids and fits, for each model, from sagemaker.
    a. Compares optimized Logistic Regression and Random forest fits with their baseline fits on accuracy, Precsion, Recall, F1 score and predictor words.
    b. Logistic Regression showed 1% improvement in recall and random forest showed no improvement at all.
    c. Although random forest did better on both Recall and F1 score, top predictor words for the model indicated that the model was biased towards product names from consumer complaints and amazon reviews dataset.
    d. Created a new dataset that comprised of observations only from tweet dataset, Sentiment 140, to see if Random Forest was able to generalize, when trained on this dataset, while still maintaining the performance observed on mixed dataset.
    e. Random forest showed top split words as being more generic but performed poorly on all metrics and hence was ruled out.

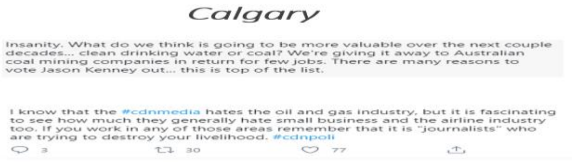
8. Topic_modelling – This notebook utilized optimized Logistic Regression fit, from sagemaker, to predict labels for tweets from City Of Toronto and performed Topic modelling on tweets labelled as grievances to extract topic/subject of grievances.
    a. Used twint to pull tweets for the city of Toronto with a retweet filter of 50.
    b. Applied cleaning functions, used for Training dataset, and vectorized twitter data using same custom tokenizer and predicted labels with optimized Logistic Regression fit.
    c. Mapped predicted labels to the original, not vectorized, Twitter dataset and extracted all observations with 1 label to create a grievance DataFrame.
    d. Evaluated LDA topic modelling performance with both Countvectorizer and tf-idf vectorization. Topic modelling performed better with CountVectorizer.
    e. Mapped modelled topics back to the grievance Dataframe so each row in the DataFrame is represented by one of the modelled topics, dominant topic.
    f. Evaluated entity recognition with spacy as an alternative to LDA.
    g. Evaluation indicated that Topic modelling and spacy entity recognition must be used together and are not alternatives to each other as each serves a different purpose.
9. Explore_cities.ipynb – This notebook applies all the steps above on twitter data from Canadian cities, Ottawa and Calgary, to highlight grievances for each city. Results do speak to the nature of each city. Grievances from Ottawa are mostly about government policies and those from Calgary revolve around Mining and Oil & Gas industries. Below are some original tweets for modelled topics for each city.


Ottawa

As @OttawaHealth, medical experts, mayors from across #Ontario call for paid sick leave today, heres @JimWatsonOttawa back in October harshly rejecting @ShawnMenard1's motion to write to @fordnation in support of paid sick days in the province.

The motion ultimately failed 15-9

You know what would be a great move on a day focused on mental health, @fordnation? Reversing your cancelled paid sick days and increasing them to 10. Imagine the stress people endure when they have to choose between paying their bills and getting better. #BellLetsTalk #onpoli


Calgary

Insanity. What do we think is going to be more valuable over the next couple decades... clean drinking water or coal? We're giving it away to Australian coal mining companies in return for few jobs. There are many reasons to vote Jason Kenney out... this is top of the list.

I know that the #cdnmedia hates the oil and gas industry, but it is fascinating to see how much they generally hate small business and the airline industry too. If you work in any of those areas remember that it is "journalists" who are trying to destroy your livelihood. #cdnpoli