STEPS

1. Retrieving Data:
   1. Getting Election Tweets:
      1. Used Tweepy to rehydrate 14 million twitter ID’s relating to tweets from the 2016 election, over a 4 month period from 7/8/16-11/8/16
      2. Stored 60 gb .json file in Blob storage
      3. Used Azure Data Factory to transfer data to Azure Data Lake Storage
   2. Getting County Level voting results/census data:
      1. Downloaded from Datasoft as a .csv
   3. Getting state level Voter registration data:
      1. Downloaded as .csv from US census
   4. Getting labeled tweet data for contextual sentiment:
      1. Downloaded .tsv from semeval-2013 with number of tweet id’s, index range for contextual word, and sentiment derived from tweet given the chosen words context
      2. Used custom python script to connect to twitter api and retrieve actual tweet text for training custom contextual sentiment model
2. Formatting Data:
   1. Election Tweets (done in DB):
      1. Get ADL mounted to read tweet .json in DB
      2. Connect azure cog. Serv. Api
      3. Import modules
      4. Convert .json to spark .rdd
      5. Create two diff. csv for diff. use case and quicker load time
         1. Tweet text & sentiment.csv
            1. Select columns used from original .json

Id, text, lang, retweet\_count, screen\_name

* + - * 1. Other formatting of rdd, such as replacing “ with \”
        2. Save rdd as parquet for quick read back into DB after cluster is restarted
        3. Read parquet as rdd
        4. Convert rdd to pandas df
        5. Add columns for sentiment analysis and use Azure or Vader to add sentiment score and label ‘pos’ ‘neu’ or ‘neg’
        6. Write as .csv to connected ADL
      1. Tweet location.csv
         1. Select columns used from original .json

Id, coordinates, state, city

* + - * 1. Save rdd as parquet for quick read back into DB after cluster is restarted
        2. Read parquet as rdd
        3. Convert rdd to pandas df
        4. Write as .csv to connected ADL
      1. Merge.csv
         1. Join the location and tweet .csv on id and keep location data, id, and sentiment (no text)
      2. Text.csv (just text and id for experimental contextual sentiment)(using local jupyter)
         1. Import modules
         2. Create df removing columns from tweet text & sentiment .csv
         3. Save as pickle for quick loading when virtual env. Restarted
         4. Use spacy to tokenize each tweet
         5. Use spacy to find POS, contextual dependency and key words in tweets
         6. Add columns with POS, context dep., Trump subject, Clinton subject
         7. Add contextual sentiment
  1. County Voting results (done in excel):
     1. Took existing data which had county voters and population, combined with voter registration by state data
        1. Used proprietary algorithm which takes county # votes and population and uses number of registered voters at state level to estimate number of registered voters at county level (see mathematical algorithm @ bottom)
        2. Calculate voter turnout based on this registered voters count
        3. Add turnout bucket column that classifies % turnout range for fill map visual

1. Training Model:
2. Visualizing Data:
   1. Notes:
      1. Unfortunately can’t use ArcGis for map visuals because can’t be used in Iframe in website
      2. Use Chiclet Slicer for cross visual filtering
   2. Power BI visualizations:
      1. County level voter turnout fill map:
      2. Election twitter activity heatmap:
      3. Piechart for EACH candidate, show percentage positive/negative in context to them:
      4. County level twitter sentiment in regards to candidate fill map:
      5. Word Cloud for each candidate that shows most used words with Clinton/Trump as the subject:
      6. Line Chart comparing candidates avg. sentiment with them as subject over time:
      7. Timeline storyteller that shows top 3 most retweeted tweets by each candidate:

TOOLS

* Blob Storage: Used for data storage of rehydrated tweets initially
* Azure data factory: used to transfer dataset from Blob to ADL
* Azure Data Lake Storage: Used for tweet data stored long term, used for initial access by Databricks
* Azure Cognitive Services: Azure ML models for different use cases
  + Sentiment Analysis API: Used API limit for free usage to score tweets, uses trained model
* Power BI: Used for visualization and exporting visual for public on website
* Twitter API: Created administrative app with api key/secret to rehydrate tweets with Tweepy or via direct calls from custom python application; make twitter API calls to pull tweets json
* Python: Used within Databricks, Notebooks, and VS for all Big Data processing
  + Numpy: Not used rn
  + Pandas: Used for loading data in python and processing
  + Spacy: Text Analysis Library but with more customizability than NLTK
    - Used for text tokenization and dependency parsing
    - Trained model on my specific kind of tweet text for better identification of key words
  + NLTK: Text Analysis Library
    - Vader Sentiment: Rule based sentiment analysis; not ML model, Used for sentiment tagging on tweets score -1 to 1
* Databricks: Spark clusters used for big data processing of 14 millions tweets to be passed for sentiment analysis and other formatting
  + 128 gb driver and 32 gb worker (1-8) nodes
* Jupyter Notebooks:
  + Azure Notebooks: Primarily used for testing due to restrictions in processing power, and max data capacity
  + Local Notebooks: Used to lower costs for smaller data file processing; basically anything that didn’t have the full 60 gb rehydrated tweets .json
* Tweepy: 3rd party twitter api rehydrating application

DATA

* Election Tweets:
  + 2016 election Tweet ID’s for tweets containing ‘Trump’, ‘Clinton’ etc.
    - <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi%3A10.7910%2FDVN%2FPDI7IN>
    - Sampled around 14 million ID’s over a 4 month period from 7/8-11/8
* County level voting results and census data:
  + <https://data.opendatasoft.com/explore/dataset/usa-2016-presidential-election-by-county%40public/table/?disjunctive.state>
* Voting registration data by state:
  + <https://www.census.gov/data/tables/time-series/demo/voting-and-registration/p20-580.html>
* Labeled Twitter data for contextual sentiment model:
  + <https://www.cs.york.ac.uk/semeval-2013/task2/index.php%3Fid=data.html>

Proprietary Mathematical Formulaic Proof: