W205: Final Project

Tim Hurt and Yannie Lee

Understanding Job Growth and Sentiment

Research Objective

Jobs have been a hot topic in politics lately, so we wanted to dive deeper how job growth and shrinkage affect the population’s view (ie sentiment)

What we need:

(1) Understanding of areas that have been hit hard by job losses

(2) A way to understand how people in those areas are feeling

Data Sources

We will utilizing two main sources:

- For Job Changes in an area, we will be looking at annual payroll changes in County business patterns (1994 to 2014) (link here)
- To understand the population's view, we will be analyzing free Twitter data

We linked the two datasets by Zip Code. Since Twitter does not allow users to pull by zip code, we mapped zip codes to lat long

Although CBP data is only available up to 2014, we will be looking at trends and identifying areas that have lost the most jobs as of 2014.

Data Cleansing

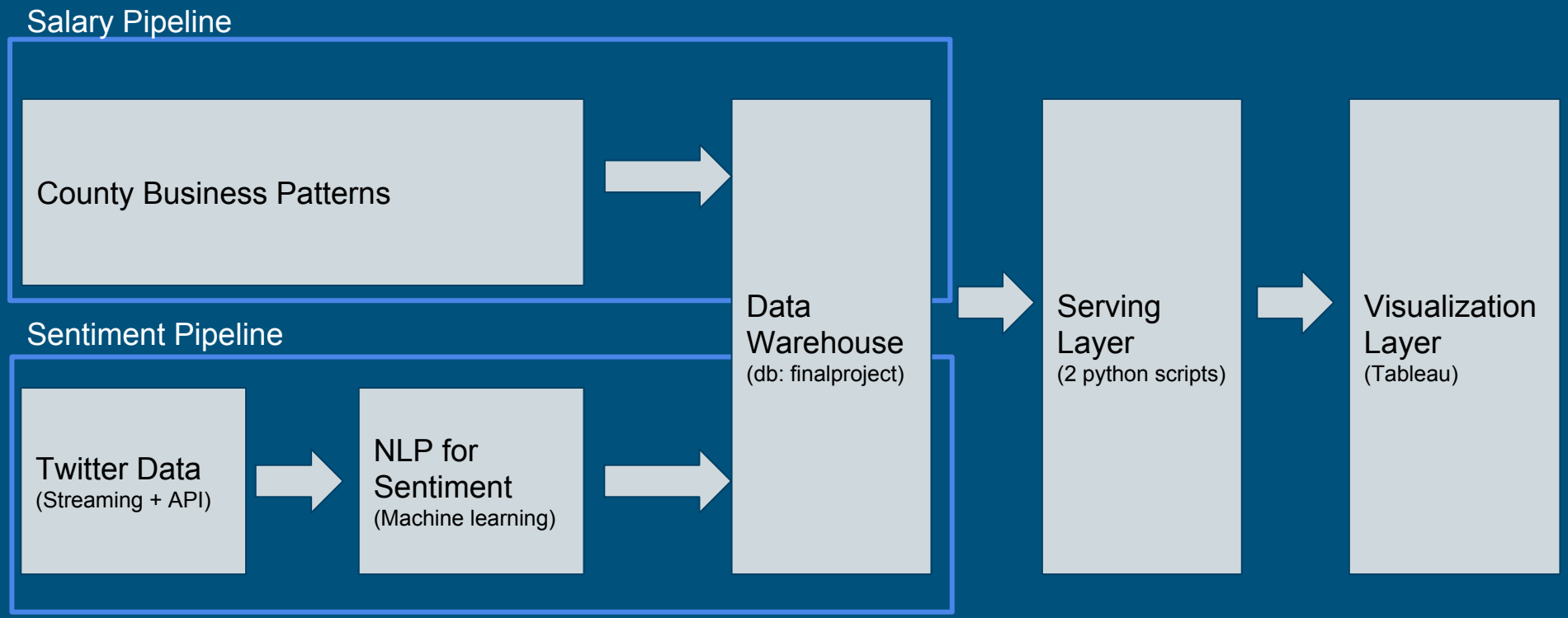
Salary Data:

- Created a composite primary by appending the year to each row
- Got rid of erroneous/missing/non-existent zip codes (99999)
- Ignored zip codes with missing years or missing payroll data

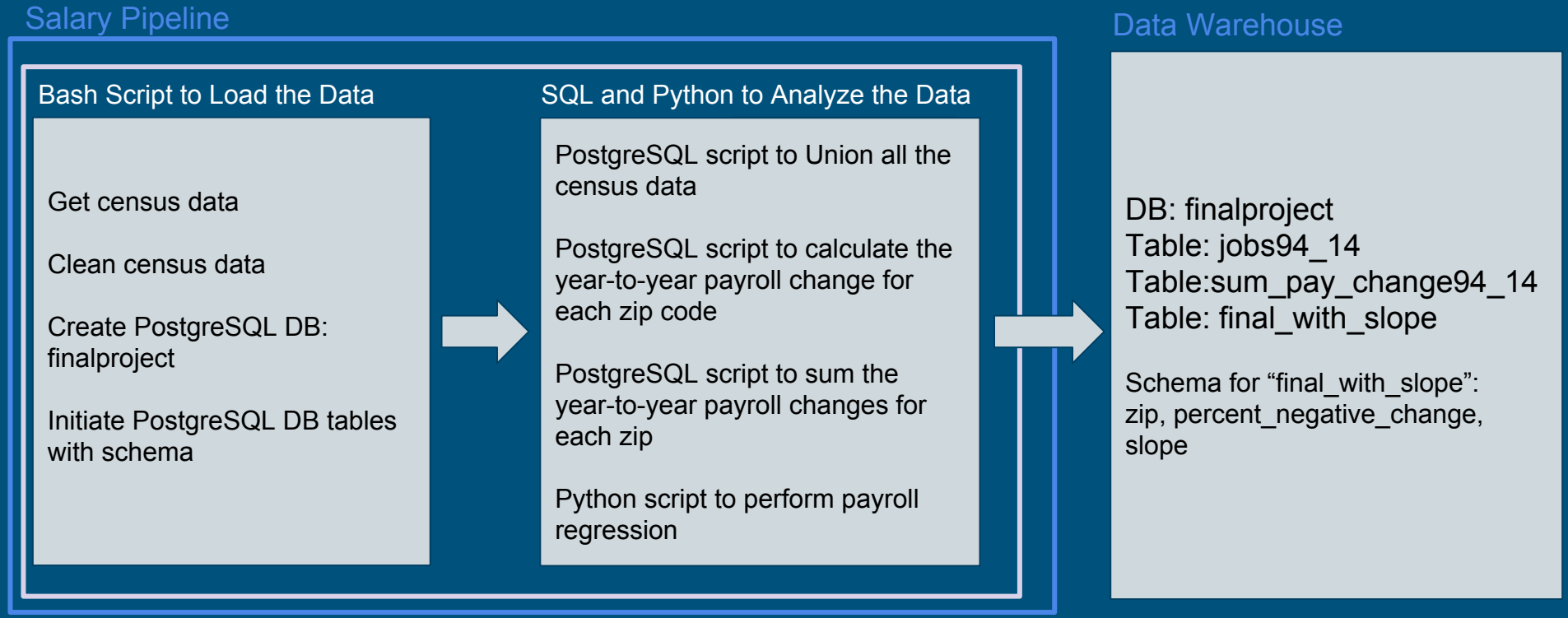
Zip Code to Lat Long Mapping

- Cleaned data to be readily used by script (remove fields that weren't used, rounding of lat/long coordinates, removing zip codes that had a NaN zip code)

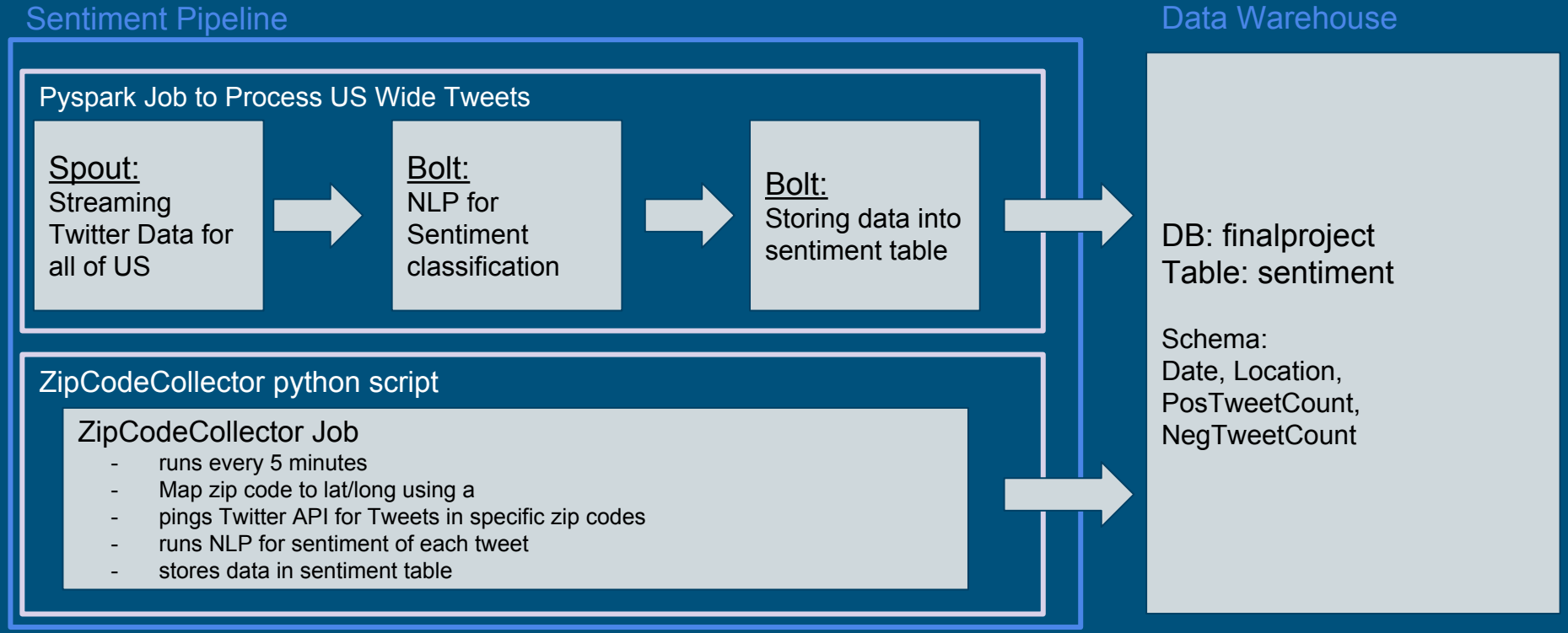
High Level Infrastructure Details



Salary Pipeline Details



Sentiment Pipeline Details



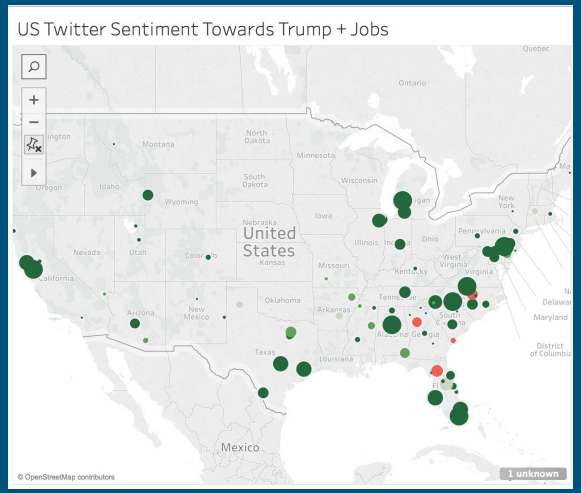
Serving Layer

- The serving layer allows the user to have pre-written scripts to access the data in the data warehouse. There are 2 prewritten scripts:
  - ExtractZipCodeData.py:
    - called with 2 parameters: (1)“path_to_save_export”, (2)“zipcodes”
  - ExtractZipCodeDataDateFilter.py
    - called with 3 parameters: (1) “path_to_save_export”, (2)“zipcodes”, (3) “start_date”,”up_to_not_including_end_date”
  - For both scripts, “zipcodes” can be: “all”, “bottom25”, “top25”, “top25bottom25”, “baseline”, or just a list of zips
    - If a list of zips is passed, it should be in the format: “94105”,”90025”,”45328”. Note that the baseline for the US will also be returned
- The scripts create a csv that can be saved to a github repo for easy local access

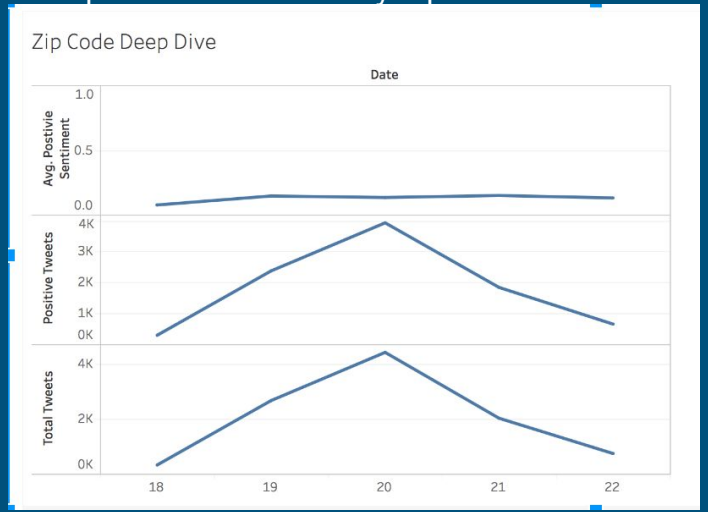
Visualization Layer

- We have created two views in tableau, both can be accessed via a web based interface for easy consumption. However, the tableau public application is needed for further analysis, like filtering to specific zip codes.

National View



Deeper Dive of Trends by Zip Code



Link to tableau public: https://public.tableau.com/profile/yannie#!/

Known limitations

- Search Term - Search Terms (“Trump” or “Jobs”) might not have been ideal, try different terms
- NLP Algo - We used a simplistic NLP algorithmn that couldn’t take distinguish between more nuanced parts of speech, like job offers, sarcastic tone
- Limited Data - Free version of Twitter API will only stream data for a short period of time, so we are only able to analyze a sample of data
- Geo Filter - Twitter data does not allow us to query specifically by zip code or country, so we have used a geobox for the US and lat/long + 5 mile radio for the zip code queries
- Limited supplemental data - Our data answers our research question directly but we do not have a lot of additional peripheral data to allow analysts to look at the data in different ways. For example, having information like population in the zip code could help us weigh the results

Proposed Roadmap

- Try collecting data with different search terms to make sure we are collecting data that will help us answer our research question
- Pay for Twitter data so we can gather more data and not be limited
- Use a more sophisticated NLP algorithm to get a broader range of emotions
- Supplement our data so we can perform more sophisticated analysis:
    - Extract more user data from Tweet
        - Would have to change structure of the way data is collected now
        - But could still be store in the same database
    - Perhaps even store the Tweet itself
- Collect data for all zip codes, over a longer period of time

Scaling the system

Salary Pipeline

- Data is only available once a year, so this shouldn't really be a concern
  - Average size of file is: 16MB/year
  - Tables for processing: 45MB/year
  - => 61MB/year (0.059GB/Year)

Sentiment Pipeline

- Since we are storing aggregated data, data storage isn't a real issue for the solution as it is currently designed. Estimate for pulling data for all zip codes:
  - Average row of data: 6.7271632278e-8 (GB)
  - Rows/Day: 34,000, Rows/Year: 12,410,000 (34000 total zips in US)
  - Storage needs/year: 1.04 GB
- To increase the processing capacity for US wide tweets, add more spouts/bolts (we are already using a pyspark, a "scale out" system, so this is easy to do.
- We can also start multiple instances of ZipCodeCollector.py to collect data for more zip codes if we run into constraints

Overall:

We need 2GB/year to store the data. With Postgres DB limits of:

Maximum Database Size: Unlimited
Maximum Table Size: 32 TB
Maximum Row Size: 1.6 TB
Maximum Field Size: 1 GB
Maximum Rows per Table: Unlimited
Maximum Columns per Table: 250 - 1600 depending on column types
Maximum Indexes per Table: Unlimited

we don't foresee any issues with storage based on the system's current design

Questions?