|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated  **Discovering opportunities at LA**  ***Data analytics*** | Abstract  The City of Los Angeles faces a big hiring challenge: 1/3 of its 50,000 workers are eligible to retire by July of 2020. The city has partnered with Kaggle to create a competition to improve the job bulletins that will fill all those open positions.  Yun Wang, Bhimasen  **Data Analytics and Intelligence** |

# Data Science for Good: City of Los Angeles

**Help the City of Los Angeles to structure and analyze its job descriptions**

The City of Los Angeles faces a big hiring challenge: 1/3 of its 50,000 workers are eligible to retire by July of 2020. The city has partnered with Kaggle to create a competition to improve the job bulletins that will fill all those open positions.

# Problem Statement

The content, tone, and format of job bulletins can influence the quality of the applicant pool. Overly-specific job requirements may discourage diversity. The Los Angeles Mayor’s Office wants to reimagine the city’s job bulletins by using text analysis to identify needed improvements.

**The goal is to convert a folder full of plain-text job postings into a single structured CSV file and then to use this data to:**

**(1) identify language that can negatively bias the pool of applicants;**

**(2) improve the diversity and quality of the applicant pool; and/or**

**(3) make it easier to determine which promotions are available to employees in each job class.**

# Data Description

The job bulletins will be provided as a folder of plain-text files, one for each job classification.

Job Bulletins [Folder]

* 683 plain-text job postings

Instructions and Additional Documents [Folder]

* Job Bulletins with Annotations
* Annotation Descriptions.docx
* City Job Paths
* PDFs
* Description of promotions in job bulletins.docx
* Job\_titles.csv
* Kaggle\_data\_dictionary.csv
* Sample job class export template.csv

Do you think companies can find better candidates by improving their job postings?

# Data processing and analysis

**Convert process**

**Plain text job postings**

**CSV file**

**Identify language**

**Improve diversity and quality**

**Job promotions**

c

# Convert plain text files to one single CSV file.

# Sentiment Analysis - Natural Language Processing (NLP)

Sentiment Analysis, or Opinion Mining, is a sub-field of Natural Language Processing (NLP) that tries to identify and extract opinions within a given text. The aim of sentiment analysis is to gauge the attitude, sentiments, evaluations, attitudes and emotions of a speaker/writer based on the computational treatment of subjectivity in a text.

Businesses today are heavily dependent on data. Majority of this data however, is unstructured text coming from sources like emails, chats, social media, surveys, articles, and documents. The micro-blogging content coming from Twitter and Facebook poses serious challenges, not only because of the amount of data involved, but also because of the kind of language used in them to express sentiments, i.e., short forms, memes and emoticons.

Sentiment Analysis is also useful for practitioners and researchers, especially in fields like sociology, marketing, advertising, psychology, economics, and political science, which rely a lot on human-computer interaction data.

Interesting example using Sentiment analysis

The 2016 US Presidential Elections were important for many reasons.

Some of the interesting outcomes that emerged from the analysis were:

The tweets that mentioned ‘@realDonaldTrump’ were greater than those mentioning ‘@HillaryClinton’, indicating the majority were tweeting about Trump.

For both candidates, negative tweets outnumbered the positive ones.

The Positive to Negative Tweet ratio was better for Trump than for Clinton.

TextBlob Sentiment: Calculating Polarity and Subjectivity

For example:

from textblob import TextBlob

TextBlob("not a very great calculation").sentiment

## Sentiment(polarity=-0.3076923076923077, subjectivity=0.5769230769230769)

VADER(Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labelled according to their semantic orientation as either positive or negative.

Example for VADER:

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

analyser = SentimentIntensityAnalyzer()

sentiment\_analyzer\_scores("The phone is super cool.")

The phone is super cool----------------- {'neg': 0.0, 'neu': 0.326, 'pos': 0.674, 'compound': 0.7351}

The Positive, Negative and Neutral scores represent the proportion of text that falls in these categories. This means our sentence was rated as 67% Positive, 33% Neutral and 0% Negative. Hence all these should add up to 1.

A screenshot of a cell phone

Description automatically generated

We use compound score in our project.

VADER performs very well with emojis, slangs and acronyms in sentences.

# Analysis in diversity and quality.

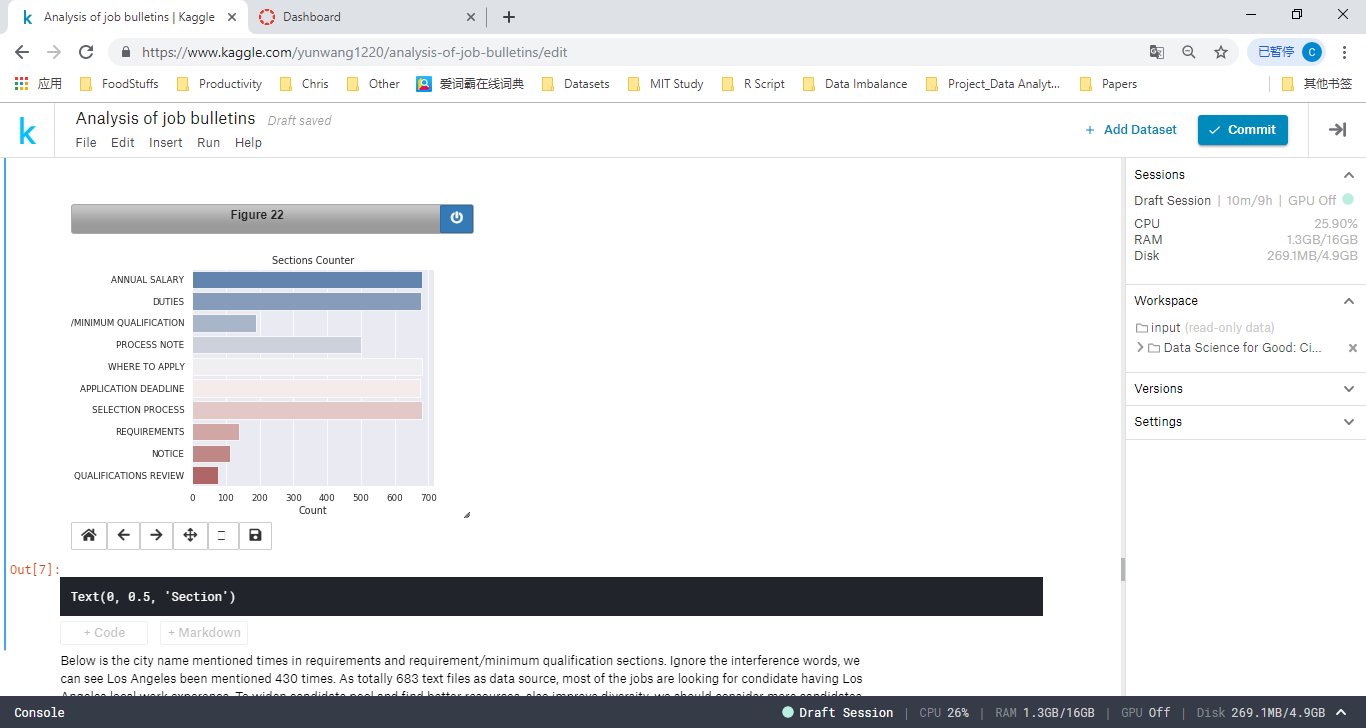
**Diversity & Quality Improvement**

During our experiments, we cannot find any information about diversity. Since every bulletin has one sentence below, we consider that there is no problems on the job description.

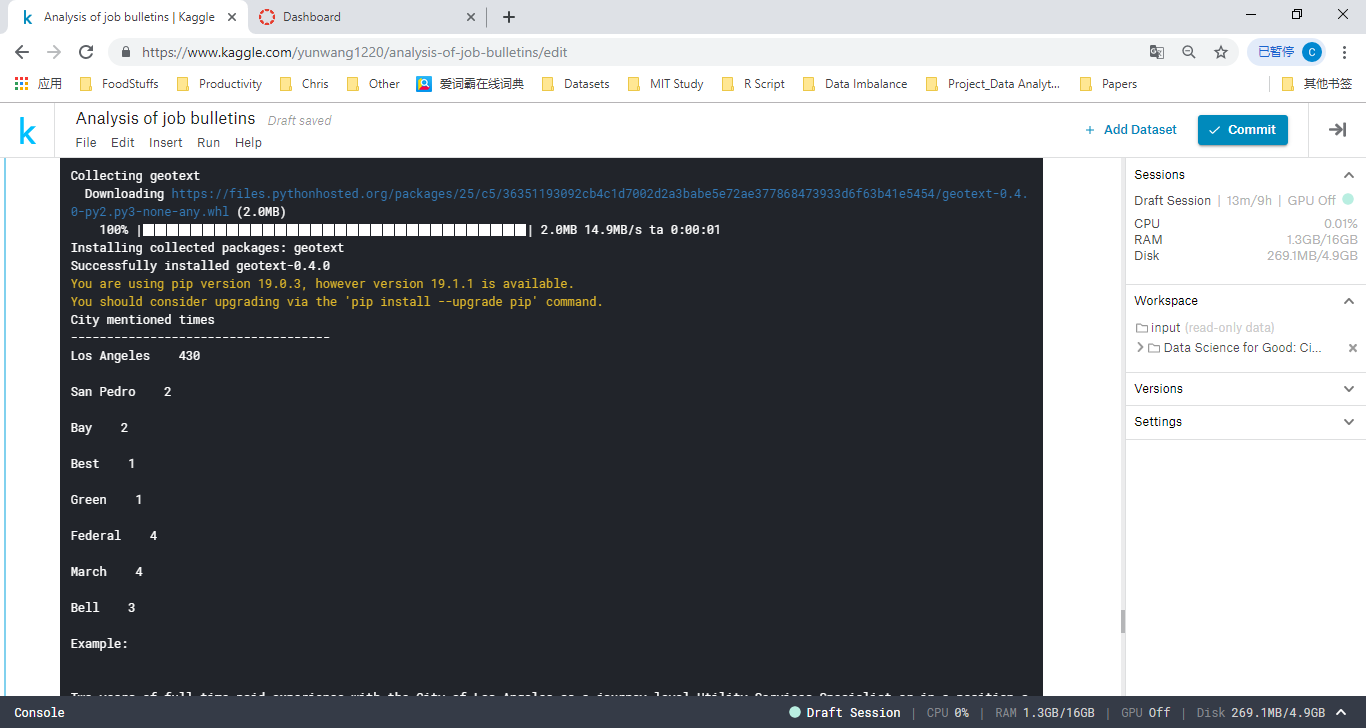
The City of Los Angeles does not discriminate on the basis of race, religion, national origin, sex, age, marital status, sexual orientation, gender identity, gender expression, disability, creed, color, ancestry, medical condition (cancer), or Acquired Immune Deficiency Syndrome.

Every text file has many sections, but seems the format is not strictly followed. We can see requirements and requirement/minimum quanlification are missing in lots of job bulletin files.

Develop a format and keep all the files aligned would improve the quality of the job bulletins.



The city name mentioned times in requirements and requirement/minimum qualification sections. Ignore the interference words, we can see Los Angeles been mentioned 430 times (totally 683 text files as data source).

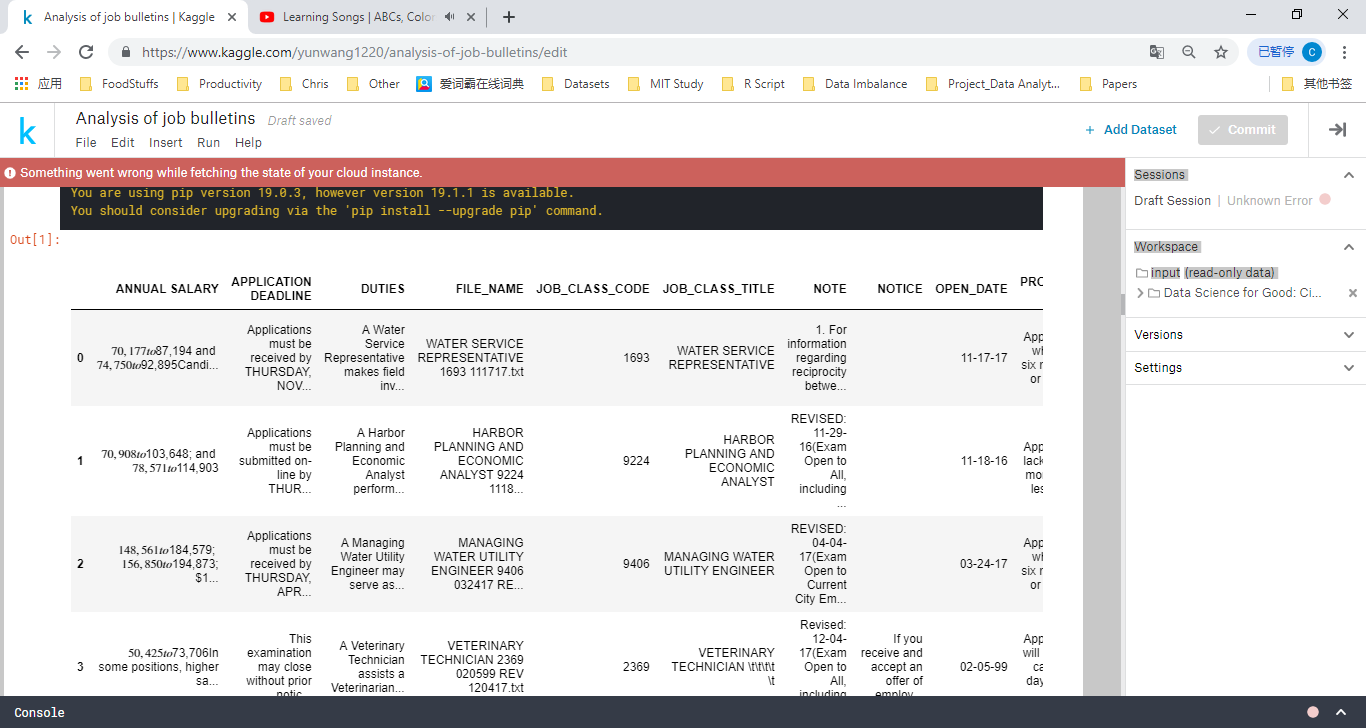


# Identify job promotions

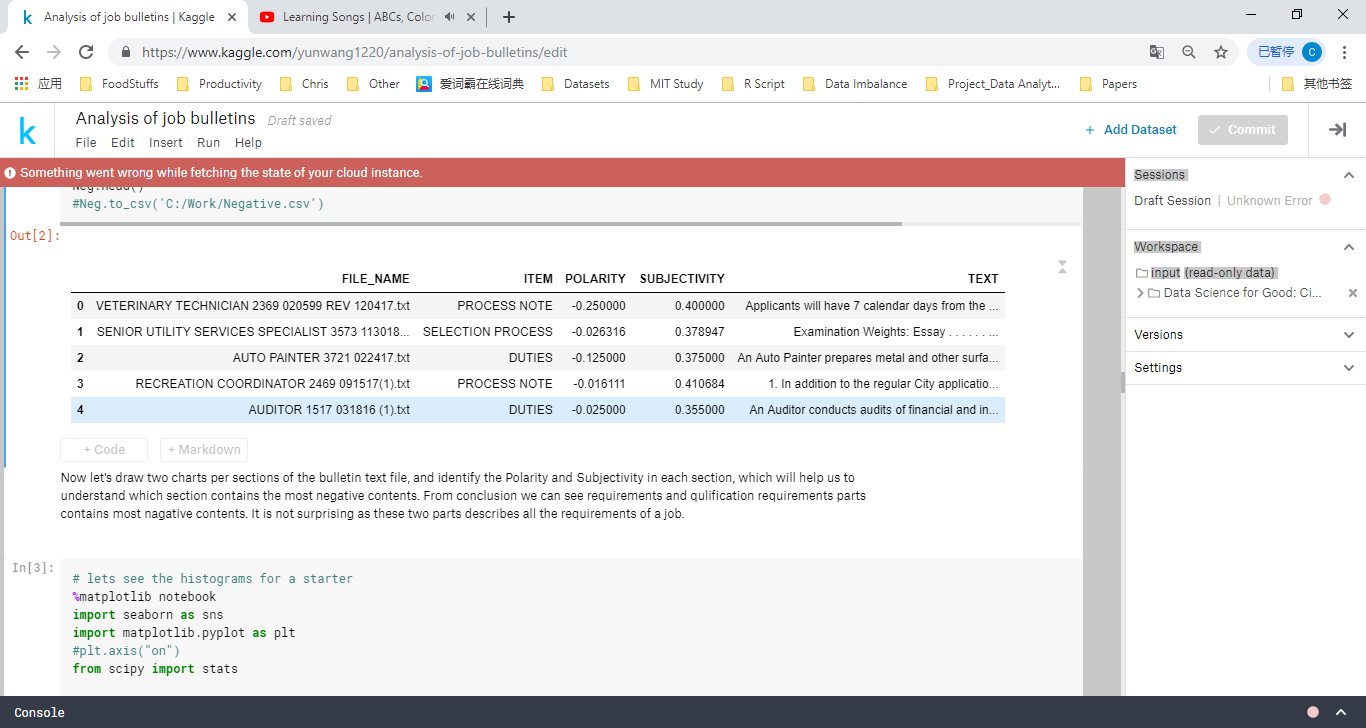
We analyzed the [REQUIREMENTS/MINIMUM QUALIFICATIONS] part to get the promotion information as all the promotion can be found in this section. If there is some promotion information, we will extract them out and draw a line to current job. Also, if we found other promotion on other bulletins, we combined them together to show the career path.

# Appendix

**CSV file output:**



**get all the negative sections**



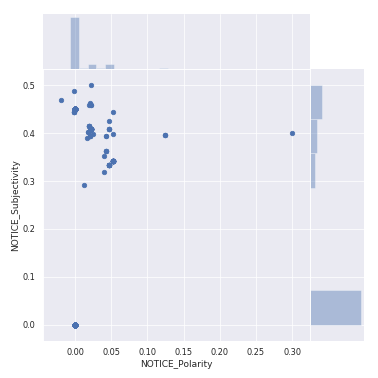
**identify the Polarity and Subjectivity in each section**

From conclusion we can see requirements and qualification requirements parts contains most negative contents.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

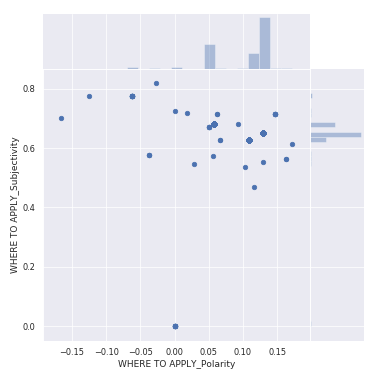
Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

This chart shows the amount of all the sections having negative contents.

A screenshot of a cell phone

Description automatically generated

Use word cloud to check the most negative words in "REQUIREMENTS" and "REQUIREMENT/MINIMUM QUALIFICATION" section.

A screenshot of a cell phone

Description automatically generated

the most frequent negative words used in all bulletins. we use vaderSentiment to get the most negative 50 words. Most of them are coming from "SELECTION PROCESS". May it's good to only mention what kind of skills will be selected, not lack of some skills will be disqualified.

A screenshot of a cell phone

Description automatically generated

Diversity and Quality Improvement

Promotion

