# Budget Text Analysis

- Datatopian Visionaries

Akash Meghani,
Miguel Gaspar Utrera,
Naseeb Thapaliya,
Sultan Al Bogami,
Unnati Khivasara

Mentors: Dr. Soumya Mohanty
Jason Jones (Guilford County)

UNC GREENSBORO

# Hypothesis Testing:

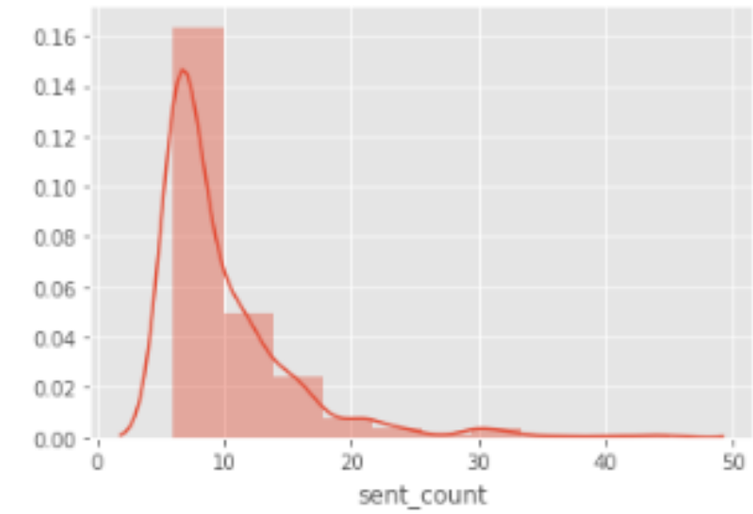
- H0 -> The sentiments remain same for service part from 2008 and 2020.

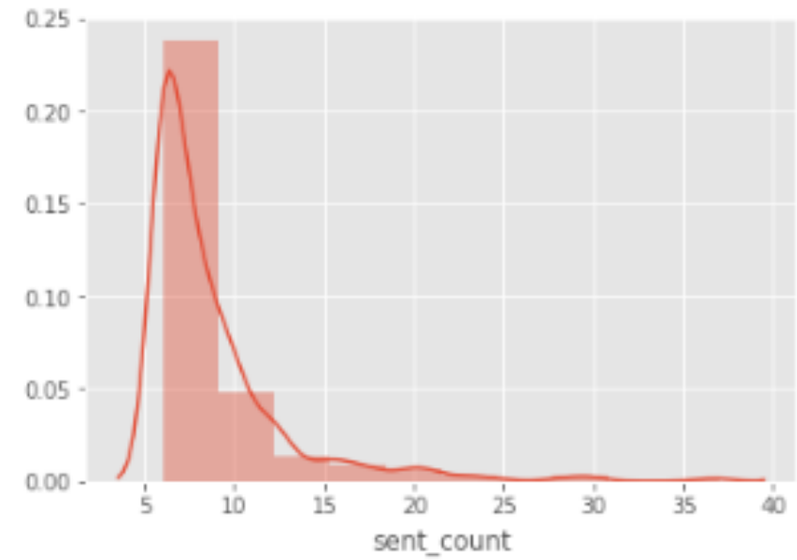
  H1 -> Sentiment changes for service part from 2008 to 2020.

- To prove this Hypothesis two sample is performed and p-value threshold is p = 0.05
- P-Value is greater than threshold (0.56) therefore we were failed to reject null hypothesis.

# Probability Distribution:

- I have concatenated Guilford county, Durham county, Durham city, charlotte city, Raleigh city :
- Took negative sentiment counts (at least more than 5 times).
- Took positive sentiment counts (at least more than 5 times):



# Machine Learning:

- Changed the whole data :

1) Parsed the pdf file.
2) Converted the string
3) Converted it into sentences
4) Data cleaning
5) Dropped the rows which are empty
6) Used Affin library from python to assign affin values
7) Assigned the sentiments accordingly

| | text | afinn_score | emotion |
|---|---|---|---|
| 0 | General revenues projected rebound from econom... | 0.0 | 1 |
| 1 | City continues face limitations balancing prio... | -1.0 | 0 |
| 2 | However City employees continue work hard prev... | -2.0 | 0 |
| 3 | Examples prior year reductions listed below | 0.0 | 1 |
| 4 | complete listing unfunded budget requests prov... | 0.0 | 1 |

# Machine Learning:

- X is text and Y is emotions.
- Used This vectorizer which breaks text into single words and bi-grams and then calculates the TF-IDF representation.
- Accuracy: 91.67

  RMSE: 0.28867

# Tasks - Sultan

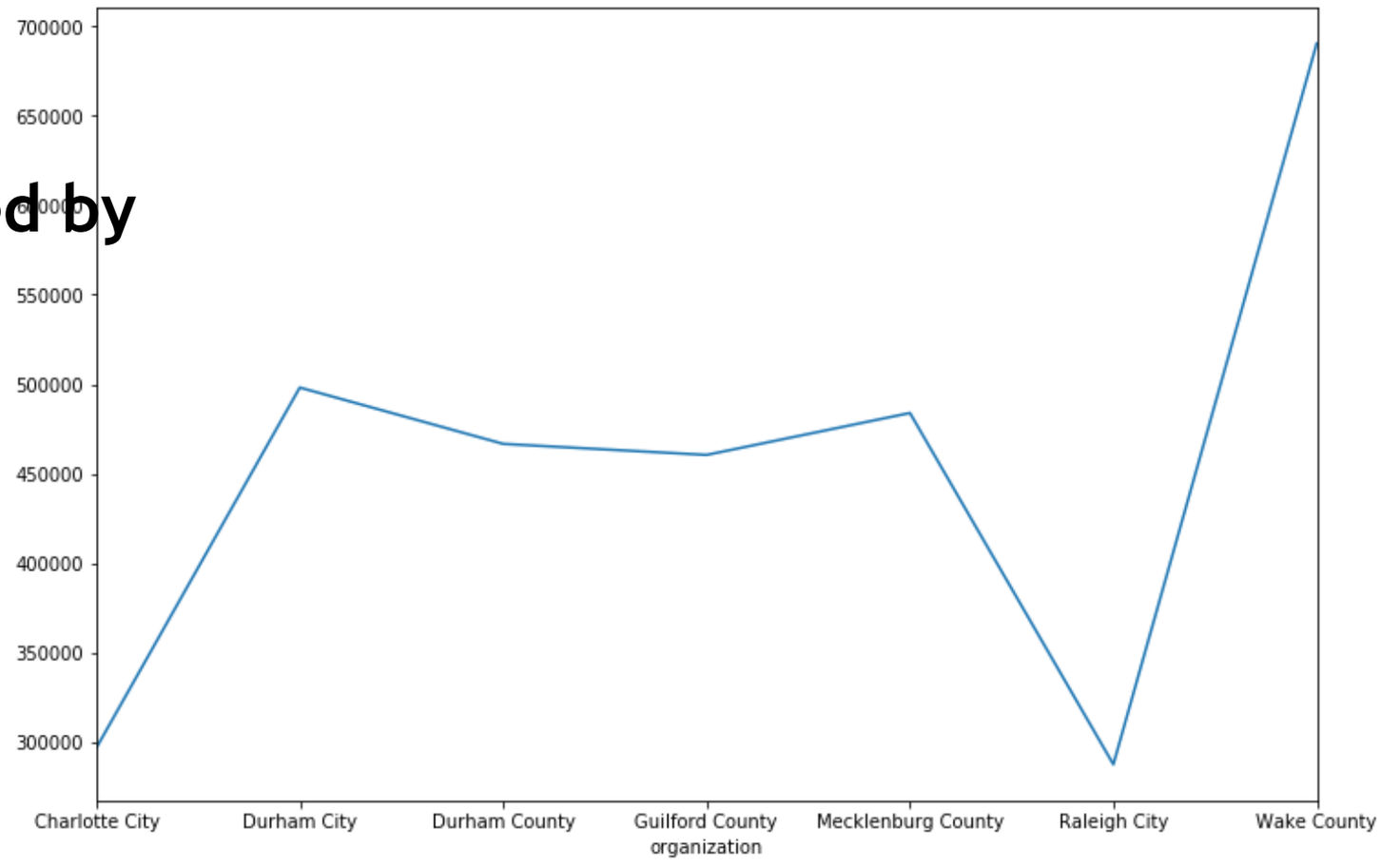
- Statistical Text Analysis:
  - Frequency distributions.
  - Mean, Variance, Standard deviations.
  - Hypothesis, and Hypothesis testing.

- Machine Learning:
  - Corpora similarity: Using ML, find methods to quantify corpora similarity.
    - Approach:
      - Divide the data set into two sets. First set = all budget documents – Guilford County budget documents. Second set =  Guilford County budget documents.
      - Create vectors.
      - Compute cosine similarity.
      - Visualize.

    - Progress:
      - Almost finished.

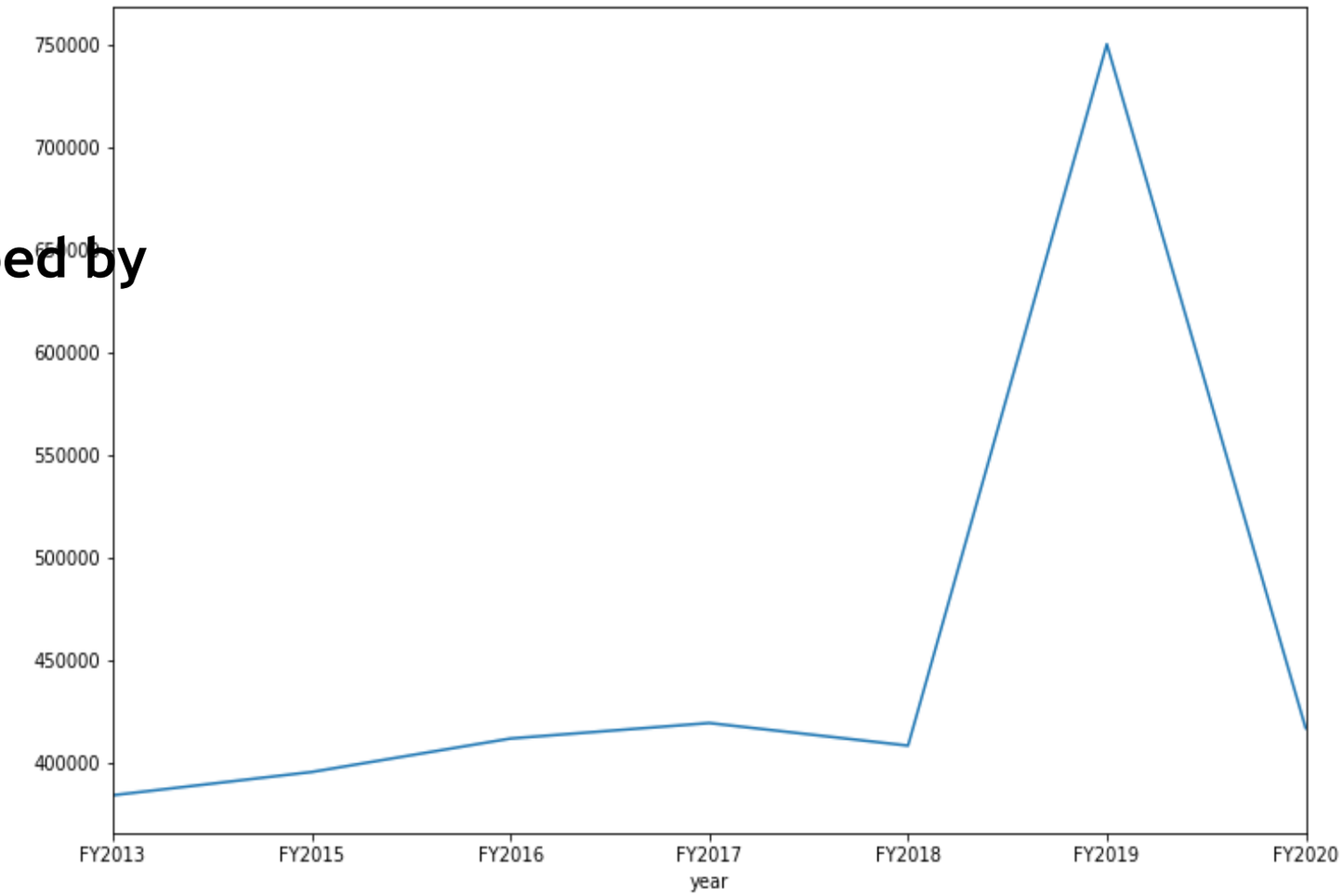
# Tasks - Sultan

**Count of words grouped by organizations.**



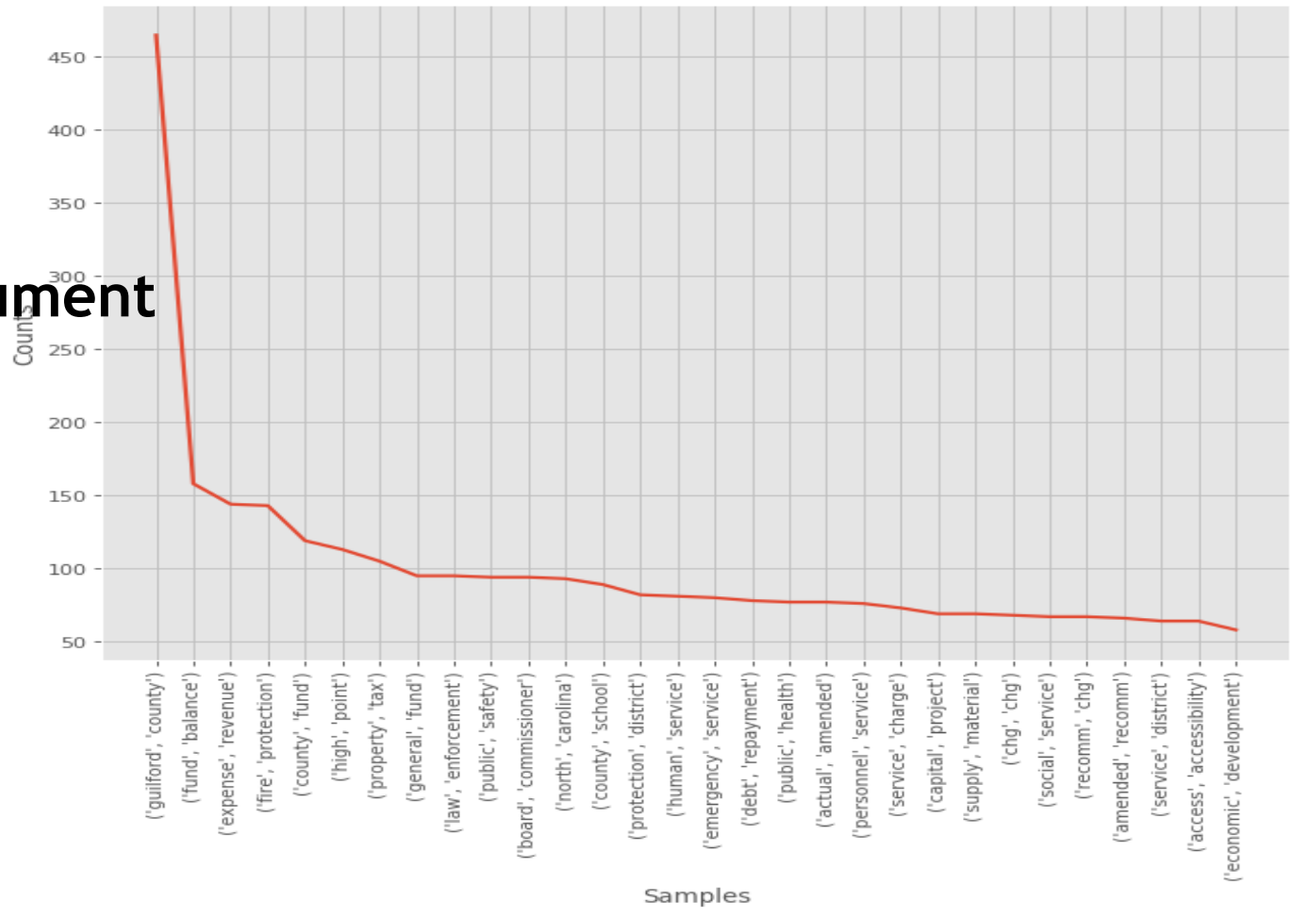
# Tasks - Sultan

**Count of words grouped by year.**



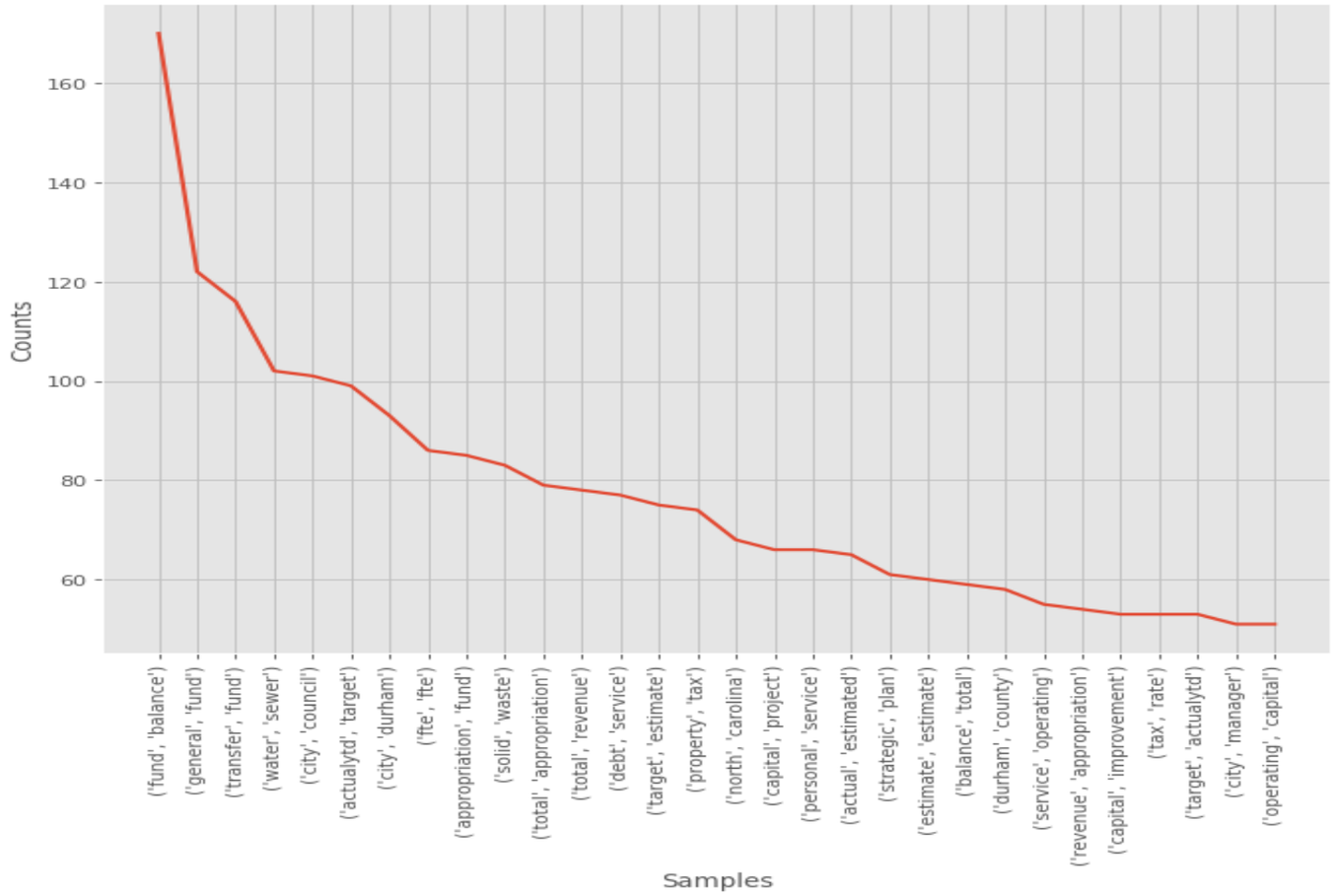
# Tasks - Sultan

**Most Frequent bigrams in Guilford County budget document From 2020**



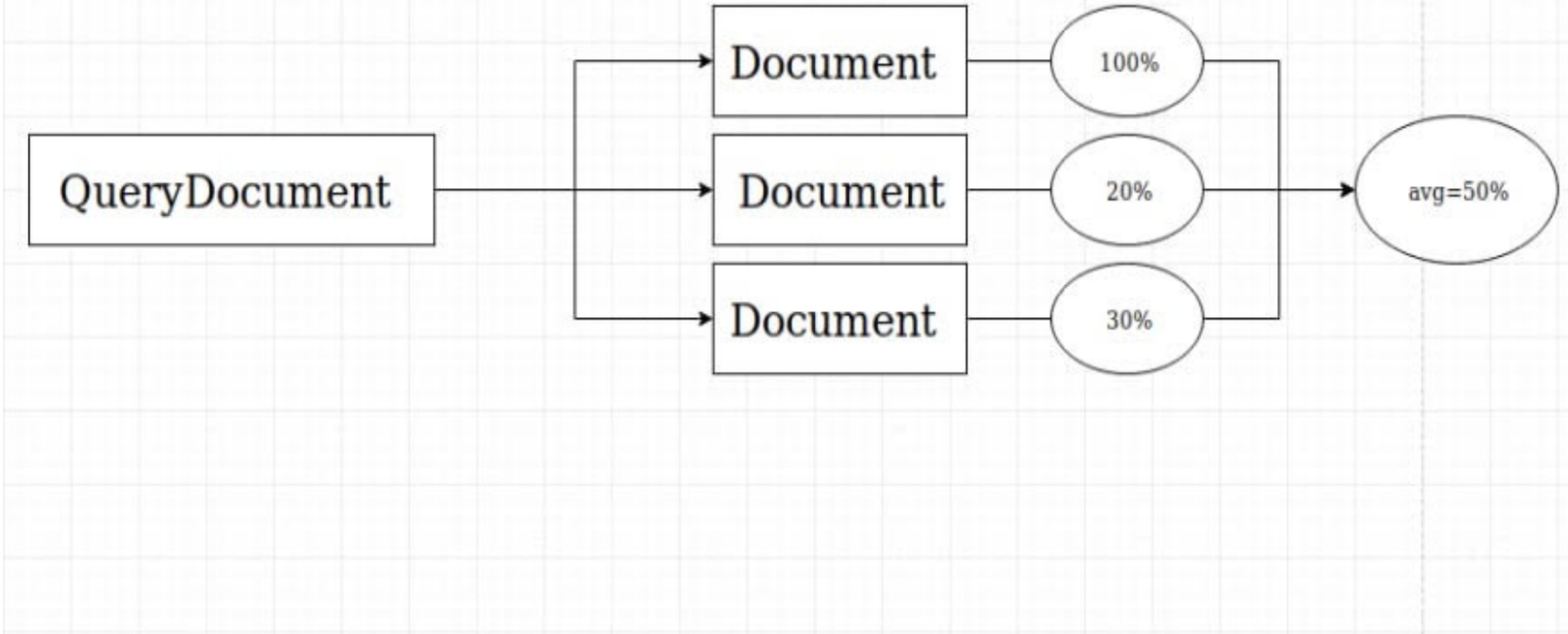
# Tasks - Sultan

**Most Frequent bigrams in Durham budget document From 2020**



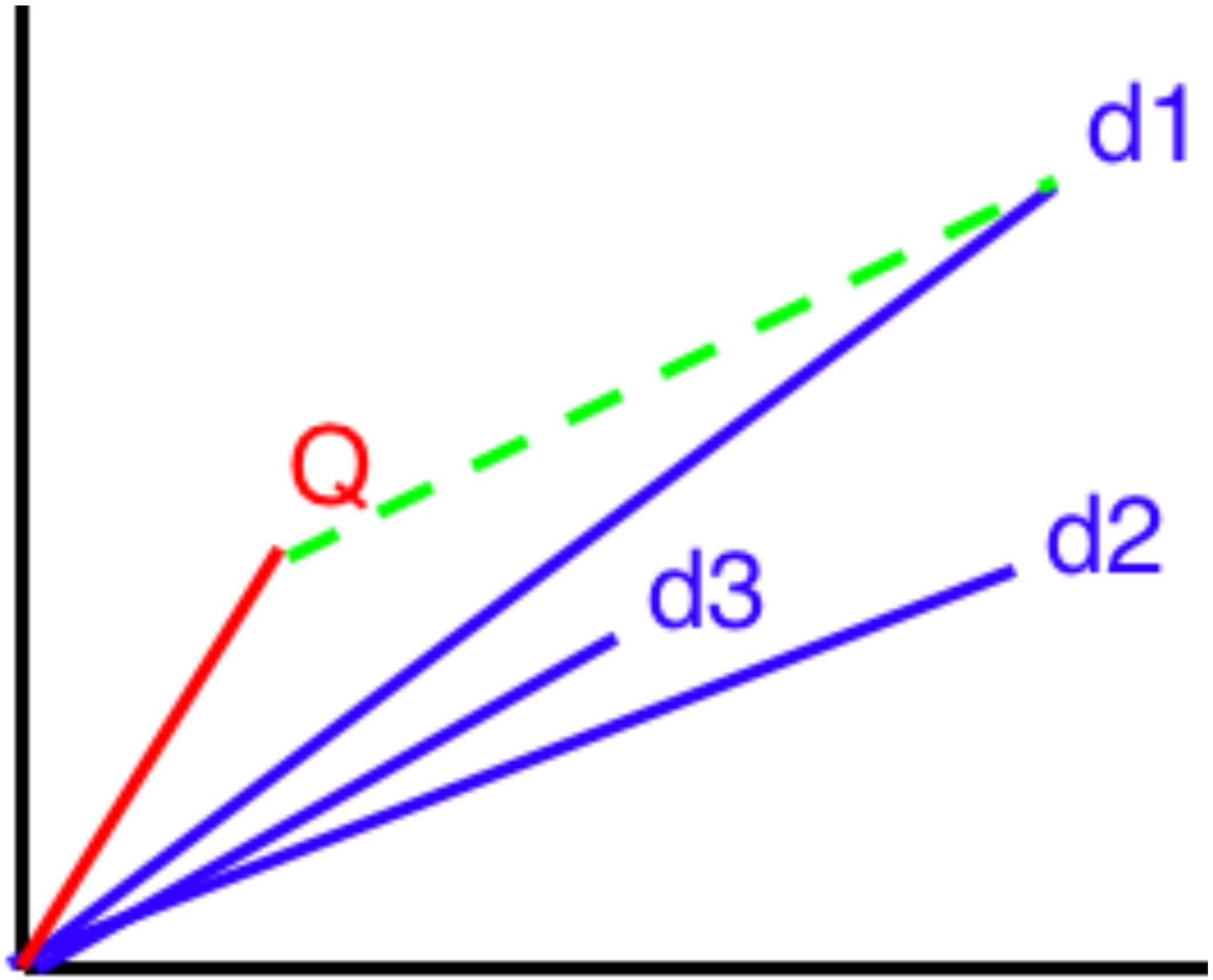
# Tasks - Sultan

**IS Guilford County talking about the same things as the other organizations?**



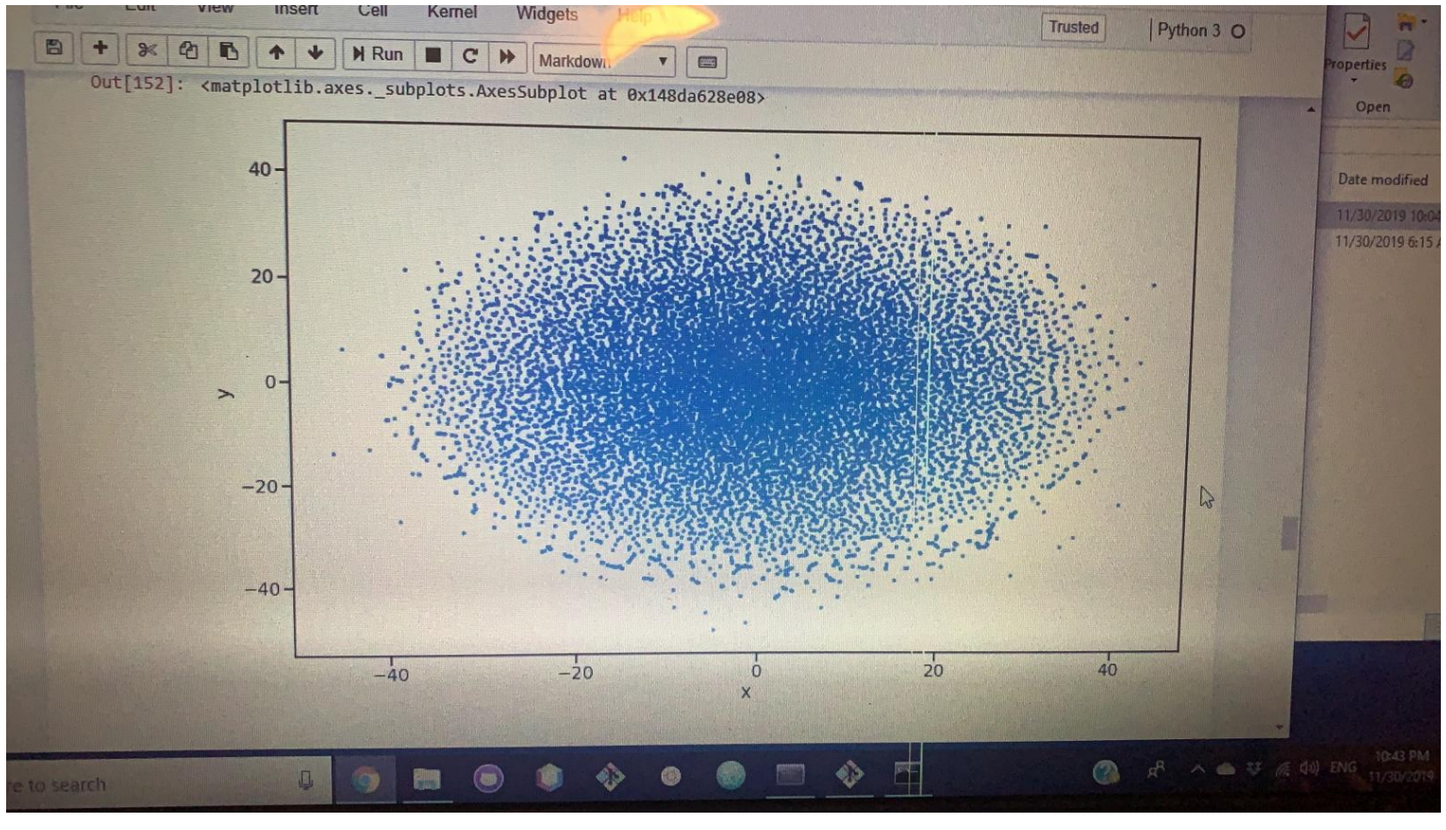
- Q = Query Document.
- D = budget documents
- Each line represents a doc.

- Task: Convert to vectors, and compute cosine similarity.

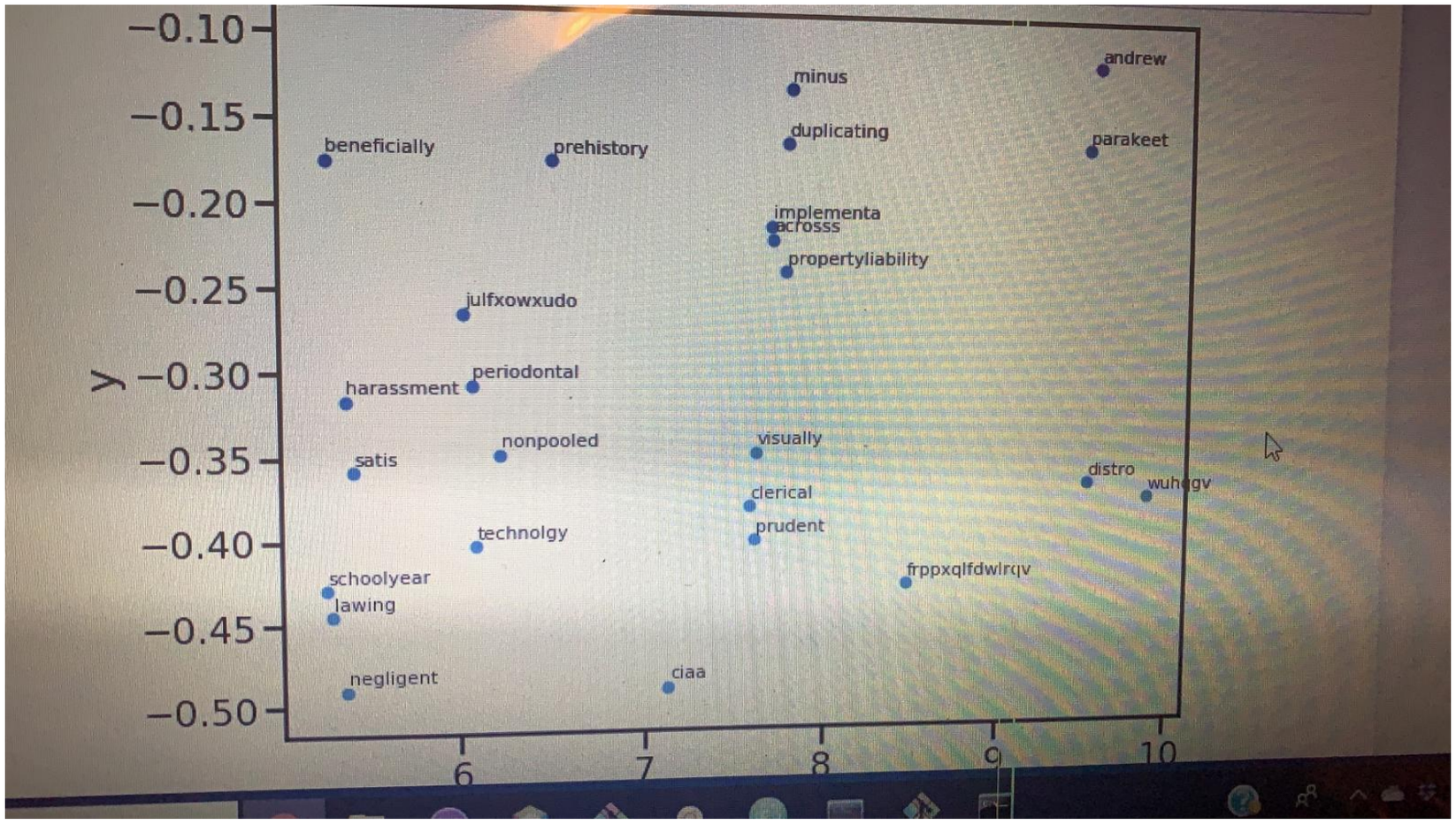


# Tasks – Sultan

- Cluster of words from 2013-2020 documents combined.



# Tasks - Sultan



# THE QUESTION ?

*“ Does a topic model for one year can identify the latent semantic structure that persists over time in this budget text domain ?*

# Tasks

- Train LDA Model on the budget texts from 2019.
- Grab Topic distributions for every budget texts using the LDA Model
- Use Topic Distributions directly as feature vectors in supervised classification models (Logistic Regression, SVM, etc) and get F1-score.
- Use the same 2019 LDA model to get topic distributions from 2018 and 2020 (**the LDA model did not see this data!**)
- Run supervised classification models again on the 2018 and 2020 vectors and see if this generalizes.

# Converting Topics to Feature Vectors for Machine Learning

In [108]:
```
train_vecs = []
for i in range(len(GC_df)):
    top_topics = lda_model.get_document_topics(corpus[i], minimum_probability=0.0)
    topic_vec = [top_topics[i][1] for i in range(10)]
    topic_vec.extend([GC_df.iloc[i].sent_count]) # counts of reviews for restaurant
    topic_vec.extend([len(GC_df.iloc[i].word)]) # length review
    train_vecs.append(topic_vec)
```

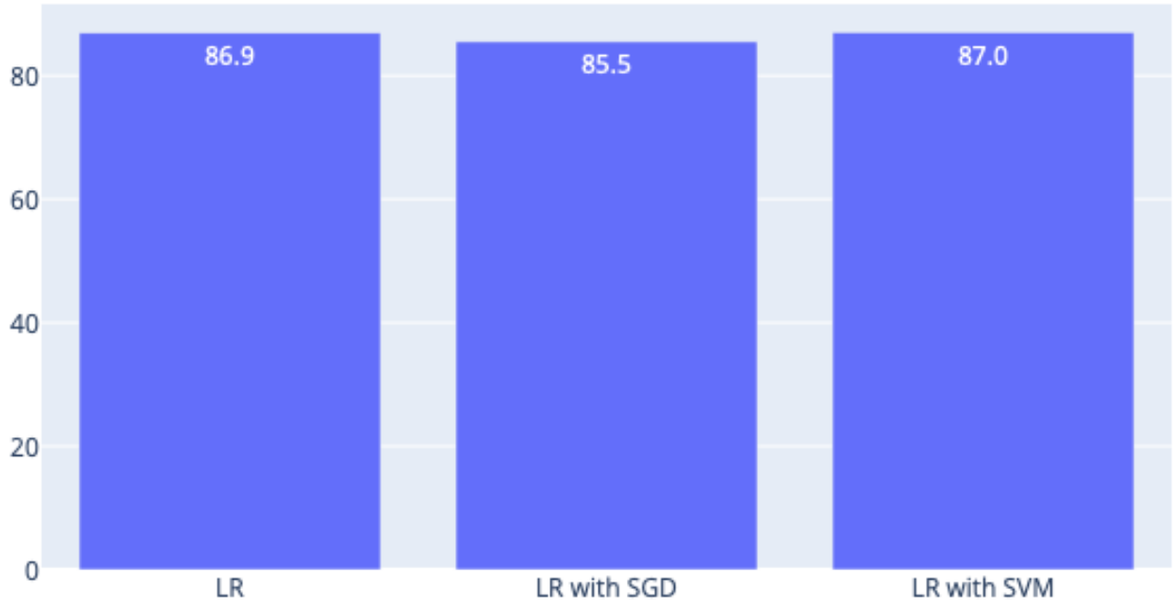
In [109]:
```
train_vecs[2]
```

Out[109]:
```
[0.04846649,
 0.042821117,
 0.03781131,
 0.0386842,
 0.055064,
 0.050130684,
 0.043984495,
 0.087888956,
 0.54818475,
 0.046964042,
 36,
 4]
```

# Supervised Classification (Training Data Result)

- X = [train_vecs];
- Y = [predicted_labels];
- Result:

```
Logistic Regression Val f1: 0.869 +- 0.003
Logisitic Regression SGD Val f1: 0.855 +- 0.008
SVM Huber Val f1: 0.870 +- 0.003
```



# Supervised Classification (Testing on Unseen Data

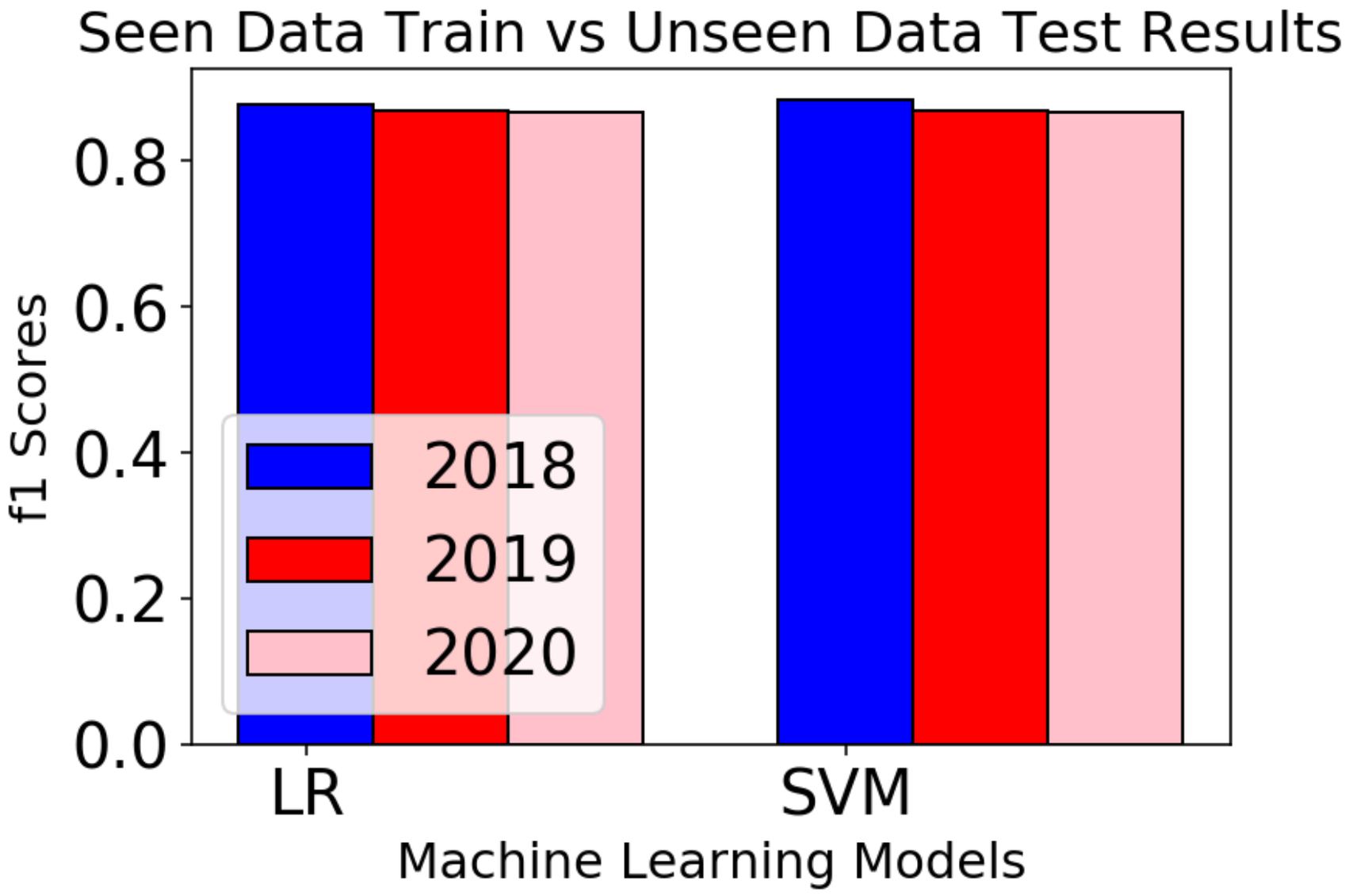
- For 2018:

```
0.877561103199744
0.883026010151702
```

- For 2020:

```
0.8663699340718182
0.866575145453356
```

# Supervised Classification (On Test Data)



# Hypothesis Testing

- H0(null hypothesis) -> The ML models are similar and perform for all the year .
- H1 -> The ML models are truly different and perform differently.
- Condition for Hypothesis taken such that p-value threshold is p = 0.05

```
chi-squared: 10.861150070126227
p-value: 0.0009820269000594094
```

- Hence, the null hypothesis was rejected, as the models were completely different.

```
]: wd_counts2.most_common(20)
```

```
]: [('fy', 3442),
    ('city', 1603),
    ('fund', 1448),
    ('durham', 1220),
    ('services', 1115),
    ('program', 1086),
    ('department', 694),
    ('budget', 675),
    ('revenues', 668),
    ('community', 650),
    ('development', 633),
    ('management', 593),
    ('service', 564),
    ('total', 563),
    ('public', 538),
    ('water', 537),
    ('general', 524),
    ('fte', 509),
    ('funds', 500),
    ('capital', 487)]
```

```
wd_counts1.most_common(20)
```

```
[('fy', 1661),
 ('city', 820),
 ('fund', 712),
 ('durham', 635),
 ('services', 541),
 ('program', 538),
 ('budget', 369),
 ('department', 357),
 ('community', 356),
 ('revenues', 332),
 ('management', 304),
 ('development', 292),
 ('service', 282),
 ('total', 282),
 ('public', 264),
 ('general', 259),
 ('water', 250),
 ('funds', 245),
 ('capital', 241),
 ('projects', 240)]
```

# The budget documents are roughly 55% similar

# Hypothesis Testing

H0 : The sentiments for Charlotte Document 2008 and 2020 are same

H1 : The sentiments for Charlotte Document 2008 and 2020 are not same

p-value = 0.28

Result : Accept Null Hypothesis

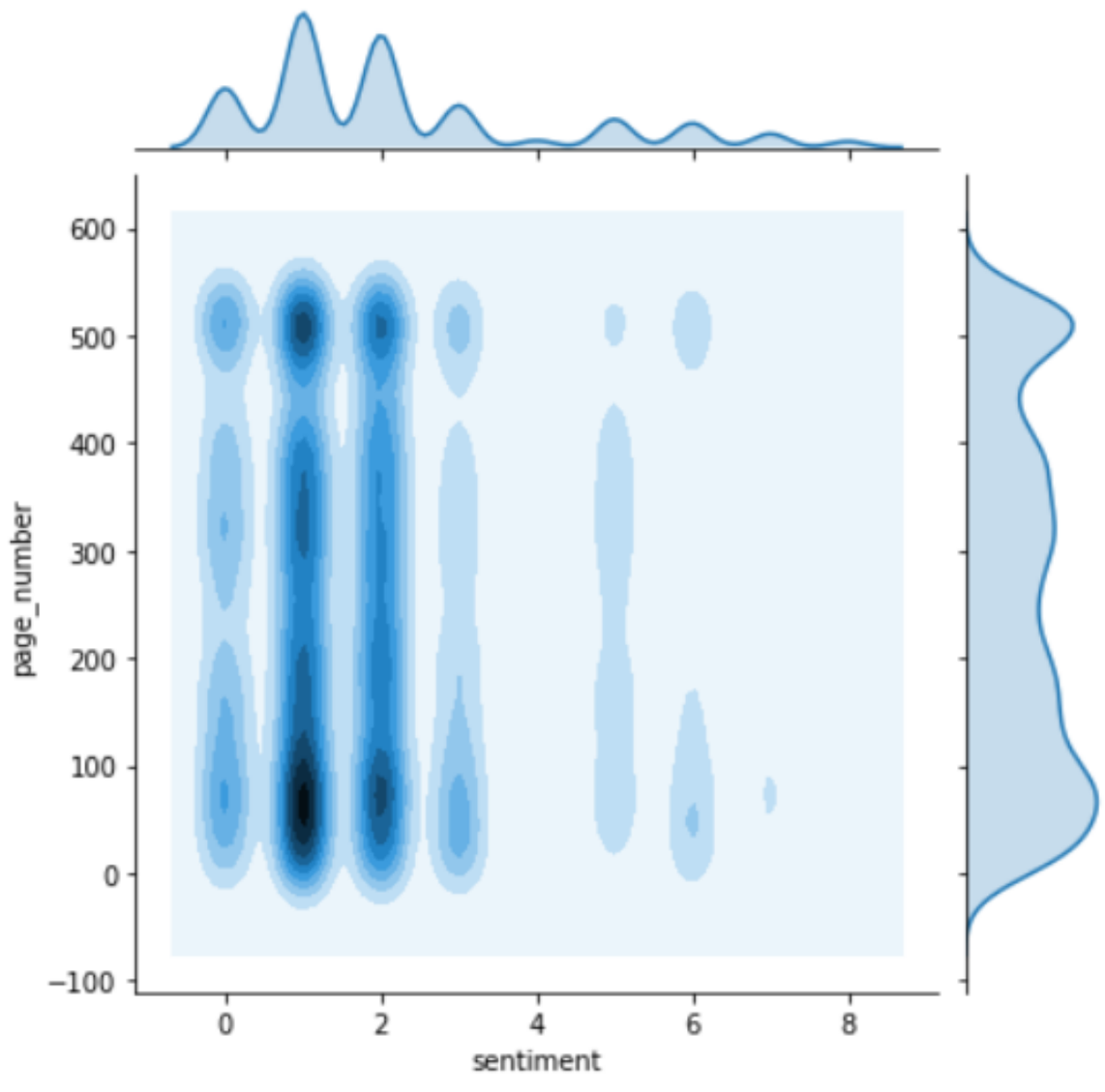
H0 : The sentiments for Raleigh Document 2014 and 2015 are same

H1 : The sentiments for Raleigh Document 2014 and 2015 are not same

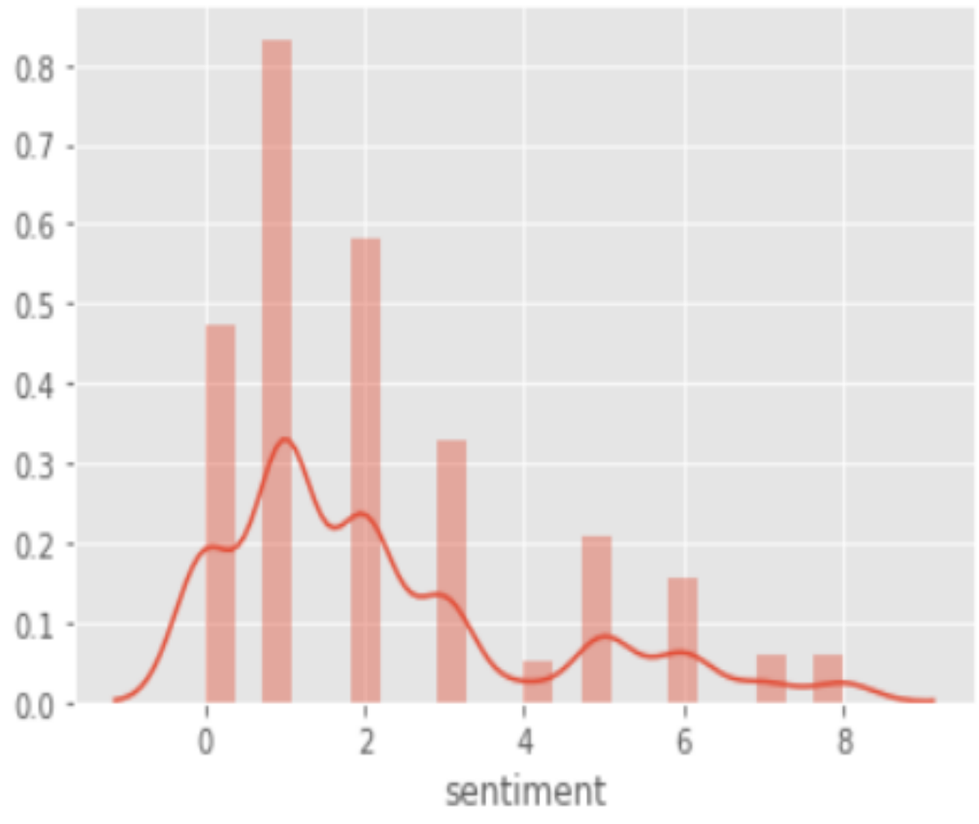
p-value = 0.98

Result : Accept Null Hypothesis

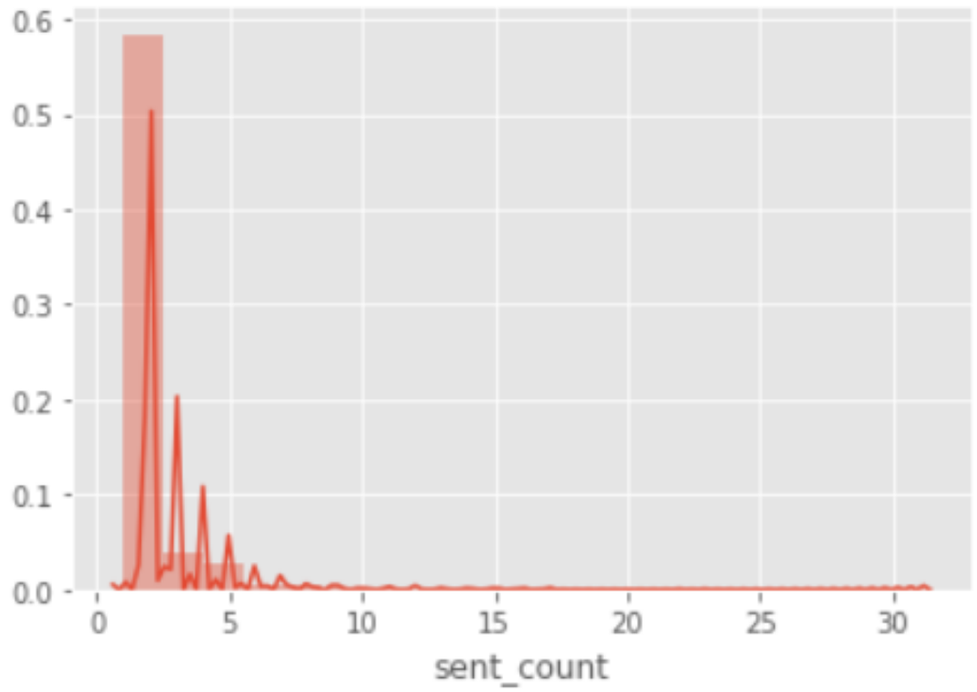
# Sentiments over Sections



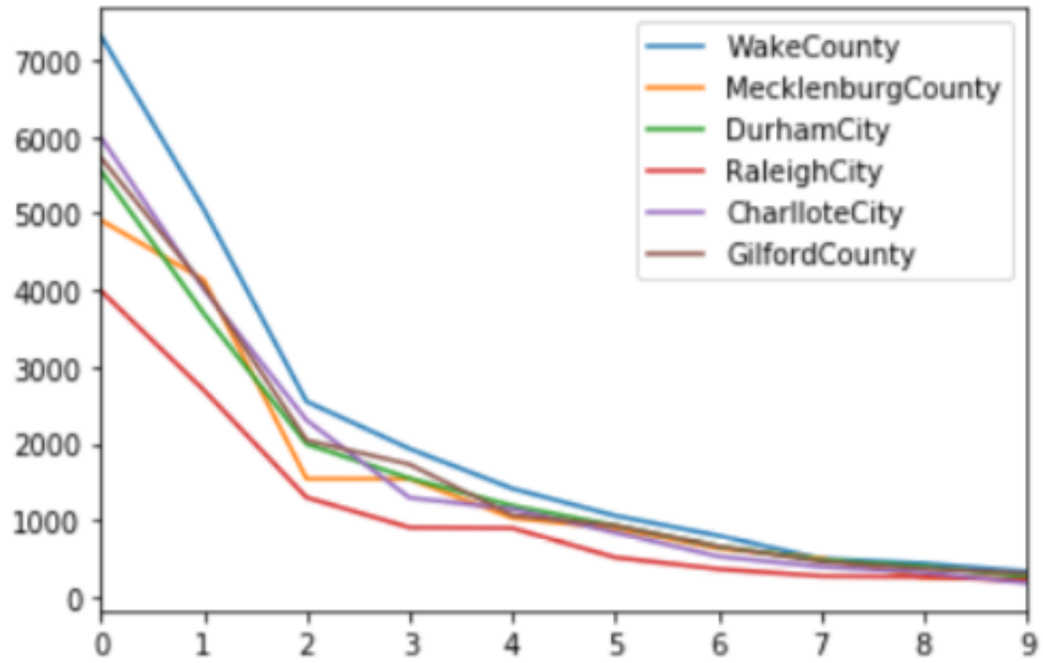
# Gillford County Budget Document of 2008 Sentiments Distribution



# Charlotte Sentiments for 2008 Budget Document



# Frequency Distribution of sentiment for Counties and cities



# Classification of Sentiments using Logistic Regression, Random Forest Classifier and Linear SVC

```
predicted = model.predict(tfidf_vector.transform(X_test))

#Calculate error between actual values and predicted values
mse = mean_squared_error(y_test, predicted)
rmse = np.sqrt(mse)
print ("RMSE :", rmse)
accuracy = accuracy_score(y_test, predicted)
print("Accuracy : %.2f%%" % (accuracy * 100.0))
```

```
RMSE : 0.5033222956847166
Accuracy : 74.67%
RMSE : 0.32659863237109044
Accuracy : 89.33%
RMSE : 0.32659863237109044
Accuracy : 89.33%
```