# Unsupervised Natural Language Processing

Dialogue analysis of the popular American sitcom, Seinfeld
By: James Chase

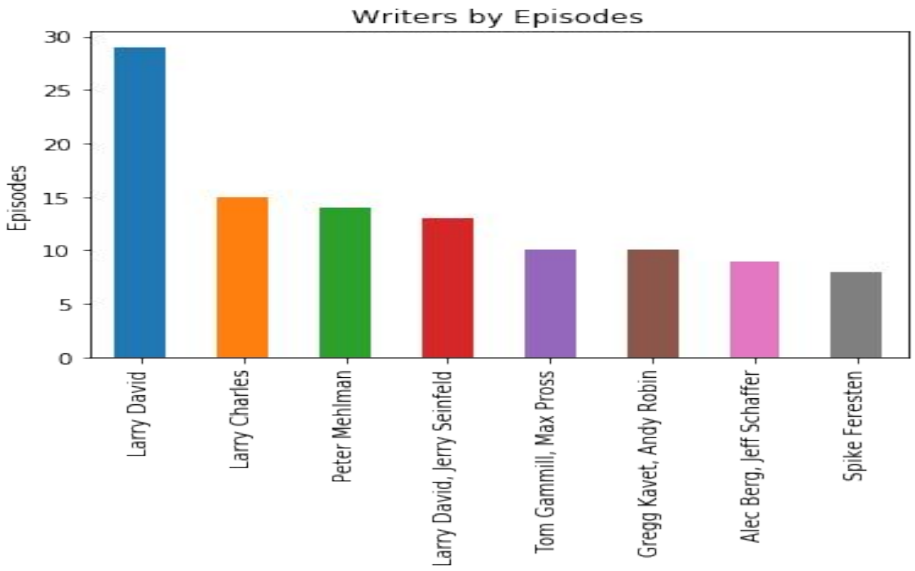
# Introduction

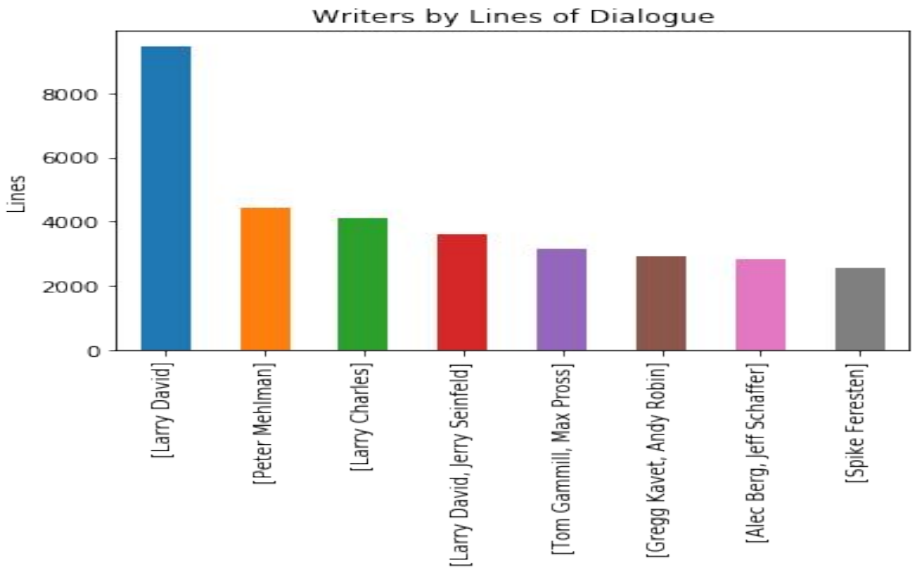
Seinfeld was an american sitcom often considered  one of the most influential sitcoms of all-time.  Often described as a show ‘about nothing,’ Seinfeld humorized the little details about everyday life.  I want to inspect how much ‘nothing’ this show is about.  Writers have the creative medium to push a show in a particular narrative direction.

**Objective:  To explore the  unique dialogue among the different Seinfeld episodes by different writers and writer tandems and to illustrate if this show is about more than nothing.**

# Data Collection

Dialogue data for all 180 episodes was found in text form hosted on kaggle.com which was scraped from the fan website, seinology.com.  53 writers were involved with the majority of episodes written by 8 writers or writer combinations.  These top 8 were responsible for 33088 lines of dialogue in 108 episodes:





# Strategy

- Text Cleaning
- Text Vectorization:
    - Doc2Vec
    - Tf-idf Vectorization
- Clustering:
    - K Means
    - Spectral Clustering
    - Affinity Propagation
- Prediction:
    - Logarithmic Regression
        - Clusters as a feature

# Cleaning and Doc2Vec / Tf-idf Vectorizer

Text was cleaned by lowercasing all text, removing punctuation, and removing stopwords.  Stopwords are words that provide no meaning or context in the text - they are often used words (the, a , is,  etc.) that we want to ignore.  I also added  eighteen additional stopwords that showed up very frequently in writer and cluster word clouds that were heavily used in Seinfeld(gonna, jerry, hey, etc.) with the remaining words changed to lemmas.

The remaining lemmas were passed through the Doc2Vec algorithm which vectorized each episode.  Unlike Word2Vec, Doc2Vec also includes paragraph embeddings.

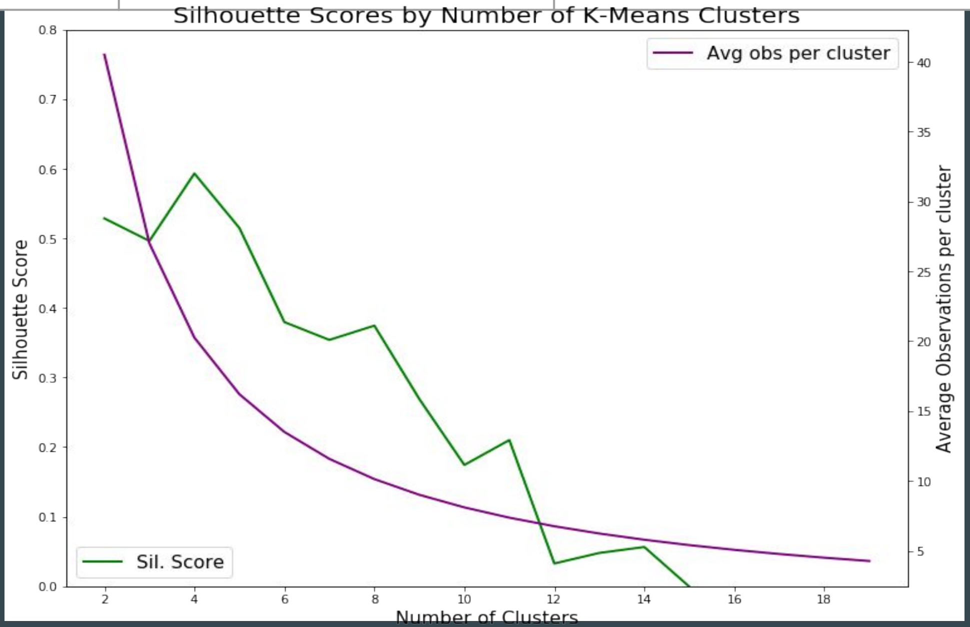
In addition, the lemmas per episode were run through the tf-idf vectorizer algorithm.

# Doc2Vec versus Tf-idf Vectorizer

Silhouette Scores:

| | K Means | Affinity Propagation | Spectral Clustering |
|---|---|---|---|
| Doc2Vec | **0.593** | 0.263 | 0.502 |
| Tf-idf | 0.415 | **0.540** | 0.162 |

Silhouette scores evaluate how similar a point is to its own cluster.  A higher value means a tighter cluster. Both K-Means performed better when running silhouette scores. With Doc2Vec edging out Tf-idf.



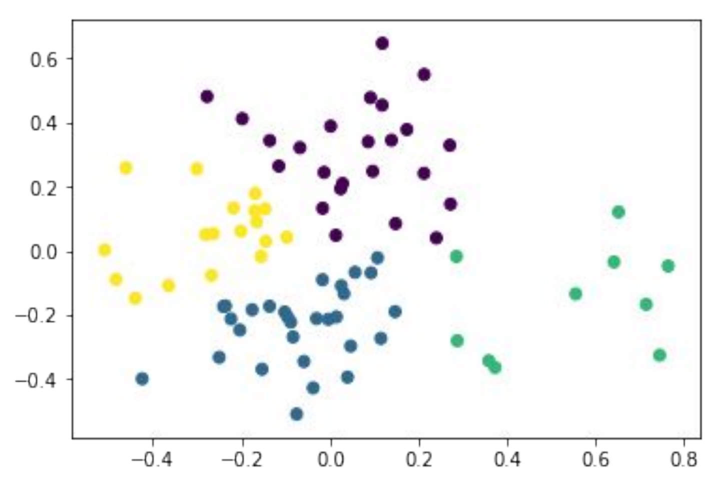
**Highest Silhouette score for K Means on Doc2Vec is with 4 clusters**

# Doc2Vec Clusters

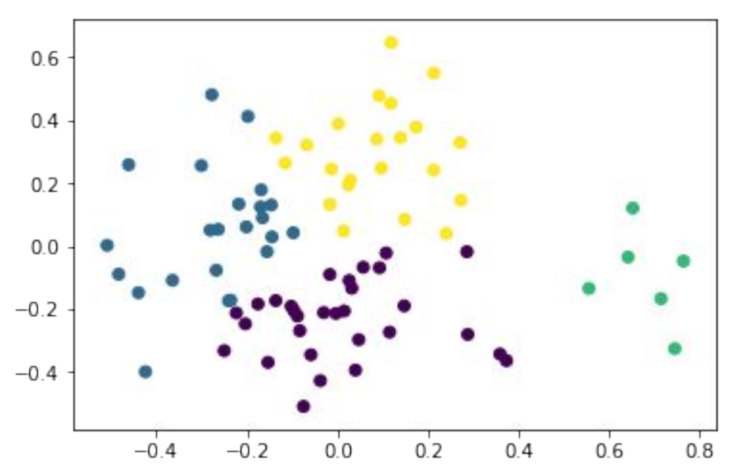
Our Doc2Vec vector data initially had 30 features before we reduced it down to 2 features via PCA reduction.  Visually, K-Means and Spectral Clustering are similar, while Affinity Propagation estimates 8 clusters.

With only 4 clusters, it is unlikely that we're clustering around writers. We can evaluate how writers are clustering using a cross-tab. Further, we can inspect  the word clouds of particular clusters to see how the algorithms are clustering.  Let's compare Spectral Clustering and K-Means.

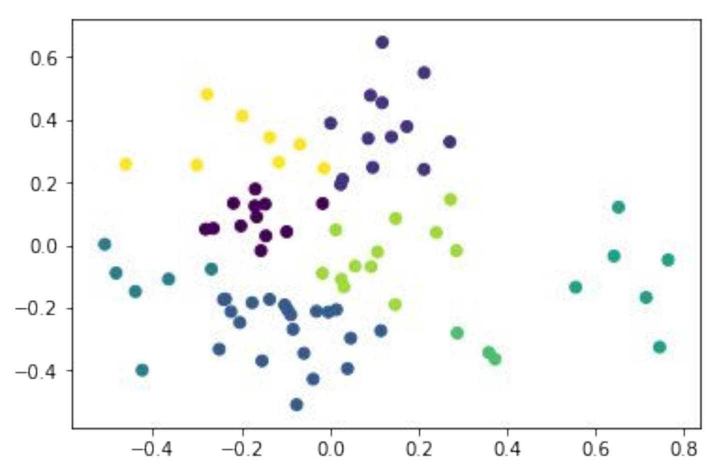
K-Means clusters



Spectral Clustering clusters



Affinity Propagation clusters



K-Means

| Clusters | Alec Berg/Jeff Schaffer | Gregg Kavet/ Andy Robin | Larry Charles | Larry David | Larry David/ Jerry Seinfeld | Peter Mehlman | Spike Feresten | Tom Gammill/Max Pross |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 8 | 9 | 1 | 4 | 0 | 0 |
| **1** | 3 | 2 | 2 | 4 | 2 | 3 | 5 | 8 |
| **2** | 0 | 0 | 0 | 4 | 5 | 1 | 0 | 0 |
| **3** | 1 | 2 | 4 | 5 | 1 | 3 | 1 | 1 |

Spectral Clustering

| Clusters | Alec Berg/Jeff Schaffer | Gregg Kavet/ Andy Robin | Larry Charles | Larry David | Larry David/ Jerry Seinfeld | Peter Mehlman | Spike Feresten | Tom Gammill/Max Pross |
|---|---|---|---|---|---|---|---|---|
| **0** | 2 | 2 | 2 | 7 | 3 | 3 | 4 | 7 |
| **1** | 2 | 2 | 5 | 5 | 1 | 4 | 2 | 2 |
| **2** | 0 | 0 | 0 | 1 | 4 | 1 | 0 | 0 |
| **3** | 1 | 1 | 7 | 9 | 1 | 3 | 0 | 0 |

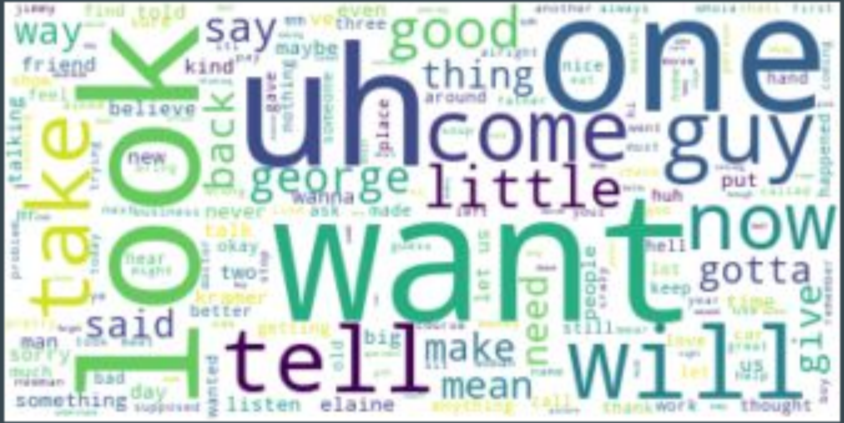
We can see that there does not appear to be a distinct writer to cluster relationship.

Both algorithms cluster Larry David and Larry Charles with high frequency in the same cluster.

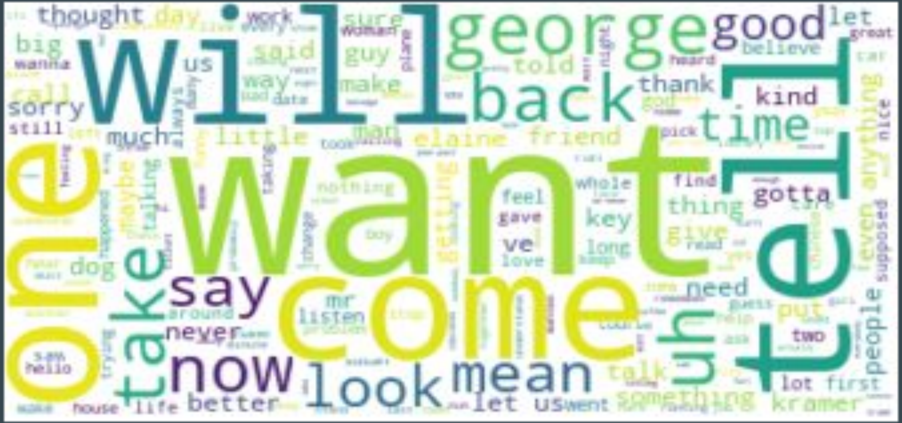
We can look at the cluster word clouds to see how the clusters may be clustering. We can also look at Larry Charles and Larry David Word Clouds.

# K-Means Clusters

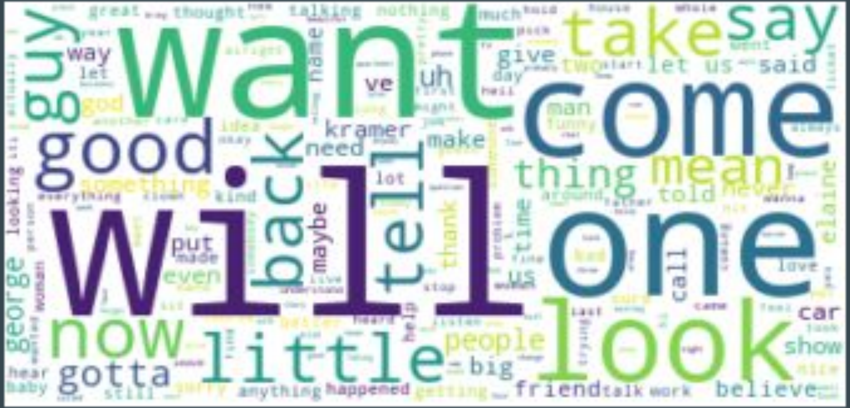
## Cluster 1



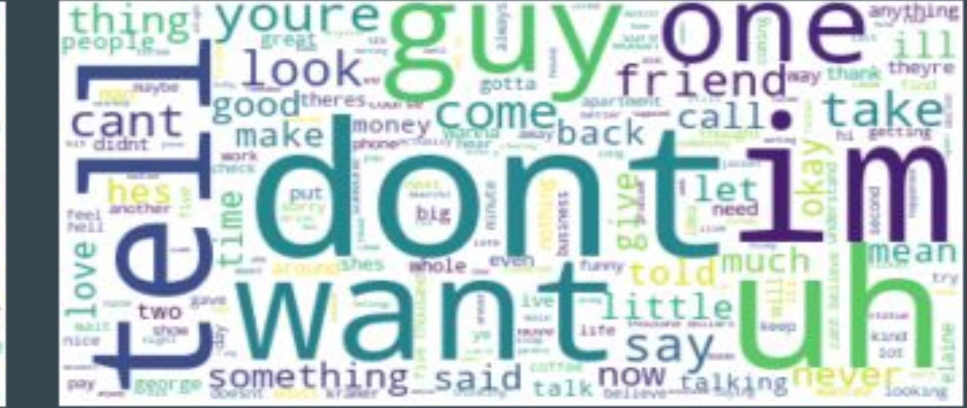
## Cluster 3



## Cluster 2



## Cluster 4



# K-Means Clusters and Dialogue Difficulty

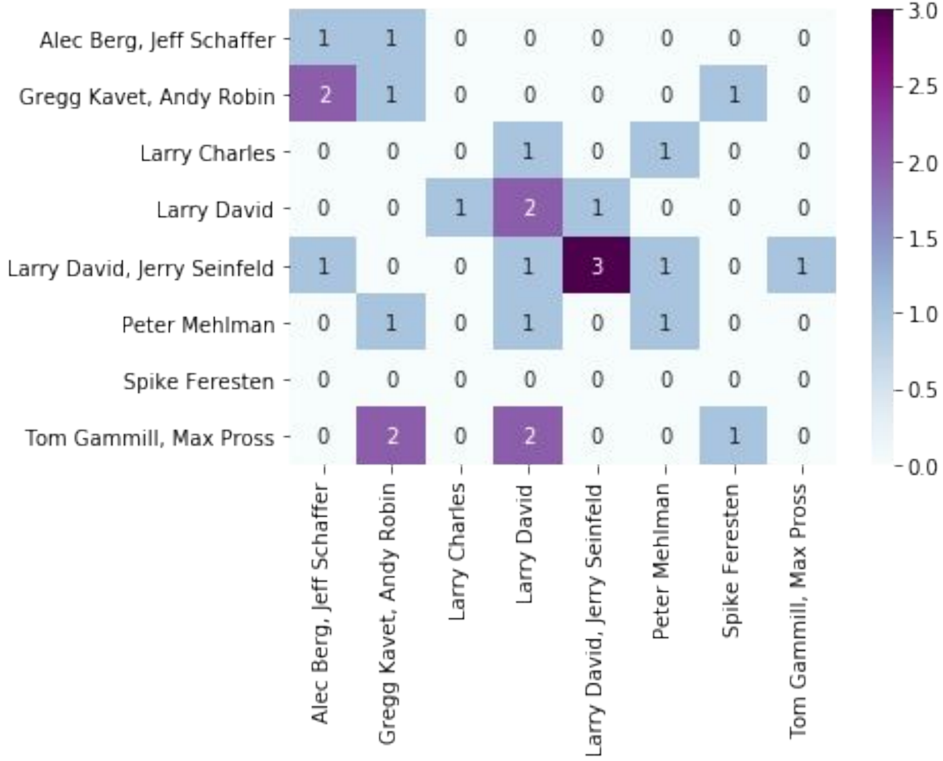
The clusters are not clustering the same words all the time which is good, the clusters are working properly; however, we can see some of the same words appear in each of the four clusters such as 'want', 'will', 'tell', 'come', etc.  The clusters don't tell us much either - there's nothing unique about them.

That's the inherent difficulty of trying to run NLP on a sitcom, especially one 'about nothing.'  Seinfeld lacks season-long story arcs for example, which could help the algorithm pick distinct features.  Harder still, Seinfeld's comedy much of the time comes from character delivery and inflection which we cannot vectorize to a point where to computer could process that information.

So far we are having a difficult time illustrating Seinfeld isn't about nothing.  Next we will try to pass our data through a supervised model, where knowing out output variable (Writers) will show if Seinfeld dialogue has distinctive enough features for prediction.

# Confusion matrices on Test Group:

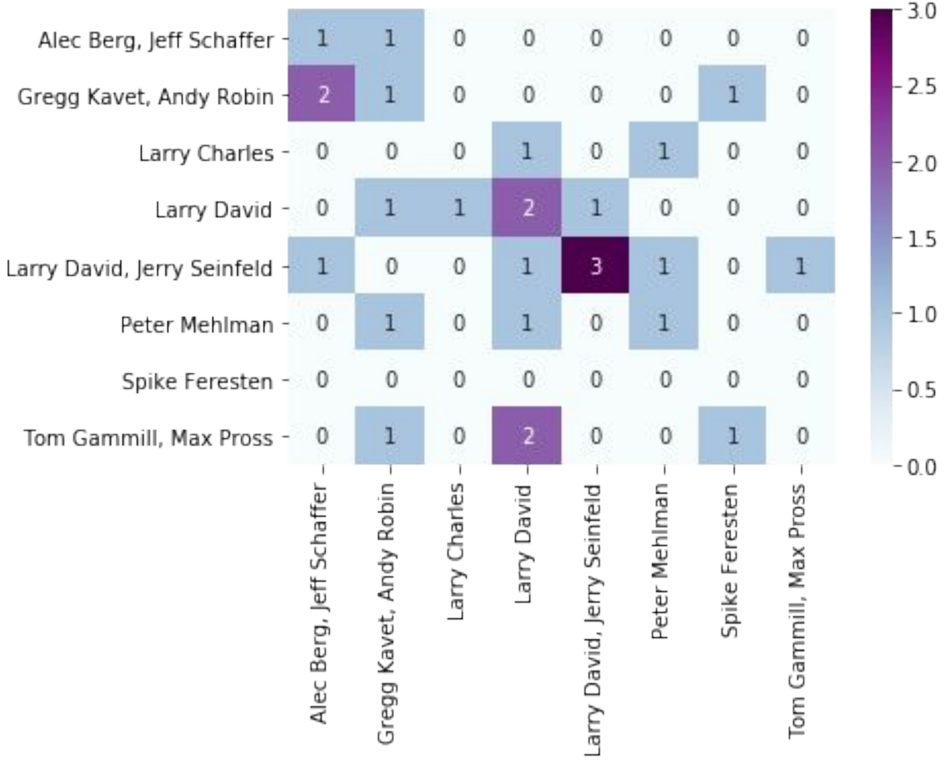
## K-Means Logistic Regression without clusters



| | Alec Berg, Jeff Schaffer | Gregg Kavet, Andy Robin | Larry Charles | Larry David | Larry David, Jerry Seinfeld | Peter Mehlman | Spike Feresten | Tom Gammill, Max Pross |
|---|---|---|---|---|---|---|---|---|
| Alec Berg, Jeff Schaffer | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Gregg Kavet, Andy Robin | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| Larry Charles | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| Larry David | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 |
| Larry David, Jerry Seinfeld | 1 | 0 | 0 | 1 | 3 | 1 | 0 | 1 |
| Peter Mehlman | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| Spike Feresten | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Tom Gammill, Max Pross | 0 | 2 | 0 | 2 | 0 | 0 | 1 | 0 |

Accuracy Score = 29.6%

## K-Means Logistic Regression with clusters as a feature



| | Alec Berg, Jeff Schaffer | Gregg Kavet, Andy Robin | Larry Charles | Larry David | Larry David, Jerry Seinfeld | Peter Mehlman | Spike Feresten | Tom Gammill, Max Pross |
|---|---|---|---|---|---|---|---|---|
| Alec Berg, Jeff Schaffer | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Gregg Kavet, Andy Robin | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| Larry Charles | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| Larry David | 0 | 1 | 1 | 2 | 1 | 0 | 0 | 0 |
| Larry David, Jerry Seinfeld | 1 | 0 | 0 | 1 | 3 | 1 | 0 | 1 |
| Peter Mehlman | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| Spike Feresten | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Tom Gammill, Max Pross | 0 | 1 | 0 | 2 | 0 | 0 | 1 | 0 |

Accuracy Score = 29.6%

# Supervised Results & Discussion

It makes sense that both logistic regression models with clusters as a feature were nearly identical after the first model scored so poorly.  If a supervised model, one which knows the output, can't make accurate predictions how can we expect our clustering algorithms to cluster anything meaningful?

Simply, the dialogue in Seinfeld was too similar episode to episode for me create meaningful clusters or a predictive model to gain insights into writer habits.  Without being able to cluster anything meaningful or create a good model, I haven't been able to show Seinfeld isn't about nothing!

A better approach may have been to compare writers over several sitcoms.  More unique dialogue may be achieved this way.  Clustering would work better over several sitcoms as well as unique features would be more easily found.

# To better predict/cluster

- To further analyze these writers further, one could add dialogue from other shows they've contributed to find unique features
- Can explore more text vectorization techniques
- Would be easier to try to predict around who spoke each line.  With four main characters (Jerry, George, Elaine, Kramer) the model would have more lines of dialogue per outcome variable.  Each character may have more unique features than writers of the episodes