Analyzing Consumer Sentiment of Future Trends in the Economy

Jack Boydell, David Richardson, Madeline Van Slyke, Alex Bamberger

Business Application/Background

Top performing organizations are “consumer centric” in the sense that they focus on understanding what the consumer values and work to satisfy their needs and desires.

“The purpose of business is to create and keep a customer.” - Peter Drucker

Smart Consumer Model

- belief that consumers understand their own notions of utility (preferences) and how to navigate the marketplace to achieve what they want
- learn from mistakes, skeptical
- utilize appropriate information
- **economically literate!**

Extension question: Can the idea of a “smart consumer” be extended to general economic indicators like unemployment and interest rates?

Question/Purpose

Overarching question: Do consumer' opinions and sentiment provide insight into being able to classify the change in unemployment and/or interest rates in a years time?

Areas of analysis:

- **Unemployment Rates:** the percentage of the labor force that is unemployed and actively seeking employment
- **Interest Rates (Federal Funds Rate):** the target rate that the Fed sets for depository institutions (banks) to lend to one another, acts as a benchmark that other interest rates follow in lock step

Ranges in our data:

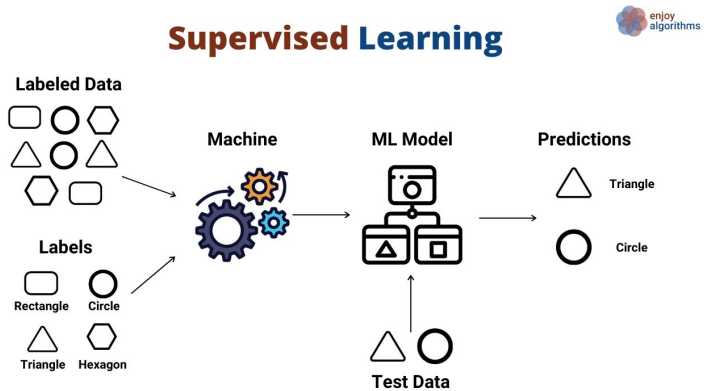
3.5% - 14.7 %

0.25% - 14 %

Machine Learning Overview

Machine Learning: the use of various algorithms to learn from and make predictions on data, programs learn and execute without explicitly programmed instructions

Supervised Learning: working with labeled data to predict or classify a designated target variable, two flavors: regression and classification



Classification: modeling the class of a categorical target variable with features, binary or multiclass

Data Sources

1. University of Michigan - Consumer Sentiment Index, Survey of Consumers

surveys of consumers
UNIVERSITY OF MICHIGAN

2. U.S. Bureau of Labor Statistics - Unemployment Rate Data
3. Federal Reserve Bank of St. Louis - Federal Funds/Interests Rates

Literature: mixed results about the predictive power of the Survey of Consumers, no approach taken from a Machine Learning (classification) perspective

Consumer Sentiment ~ Unemployment

Model: Logistic Regression Classifier

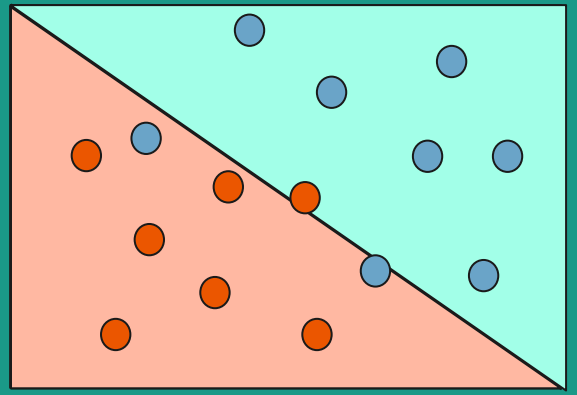
Feature Engineering

twelve_month_net_change
-0.5
-0.4
-0.5
-0.3
-0.4

year_NC
1
1
1
1
1

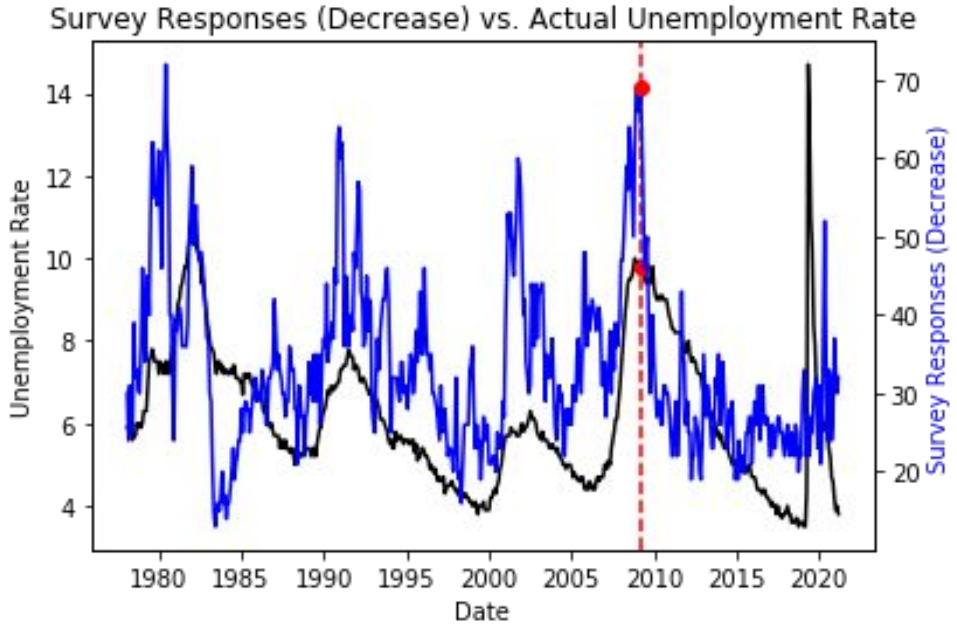
Understanding Logistic Regression:

- creates a linear decision boundary that divides data into two distinct groups, above and below the line



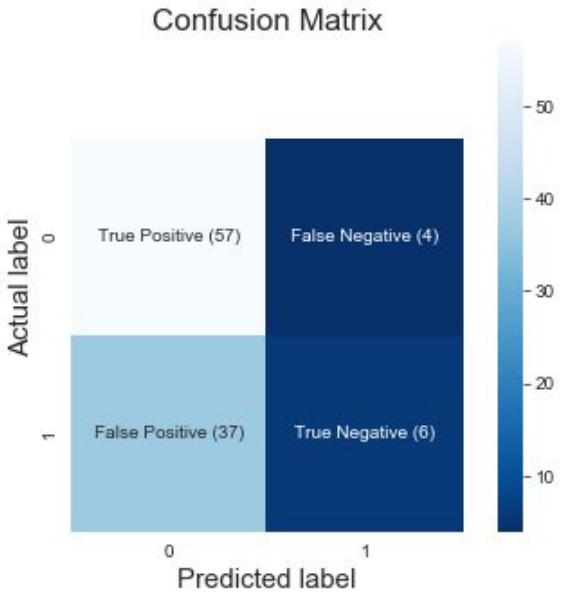
Exploratory Data Analysis

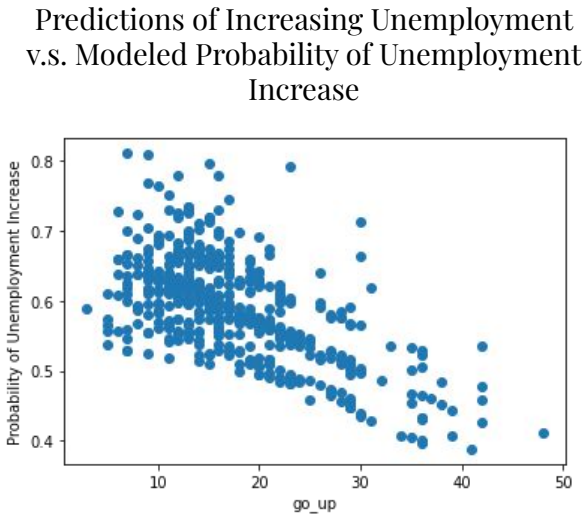
Time series comparison of survey counts for “go down”/decreasing vs. unemployment rate:



Modeling/Results ~ Unemployment Rates

- overall low classification accuracy (~.6)
- promising results in correctly classifying the an increasing/no change rate (see recall)





Performance Metrics:

Accuracy score = 0.606

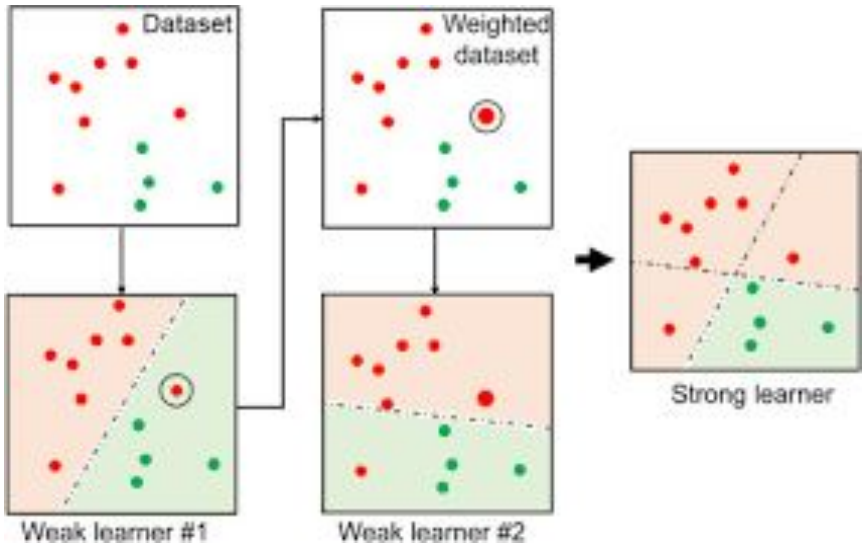
Precision = 0.61

Recall = 0.93

Adaptive Boosting Ensemble Method

Using an adaptive boosting classifier with the base estimator being the logistic regression model discussed previously.

Understanding AdaBoostClassifier()



Recall Logistic Regression metrics:

Accuracy score = 0.606

Precision = 0.61

Recall = 0.93

Comparing accuracy scores to Logistic Regression by itself we find no huge improvement in accuracy or in the confidence interval… dataset is not huge!

Consumer Sentiment ~ Interest Rates

Model: Linear Support Vector Machine (SVM)

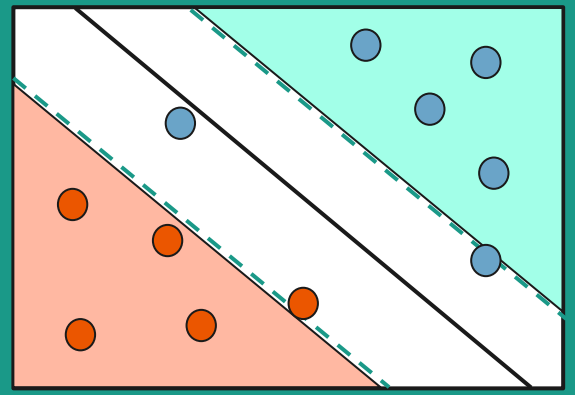
Understanding Linear SVM:

- creates linear decision boundary
- maximizes margin between classes
- tries to minimize misclassification

Feature Engineering

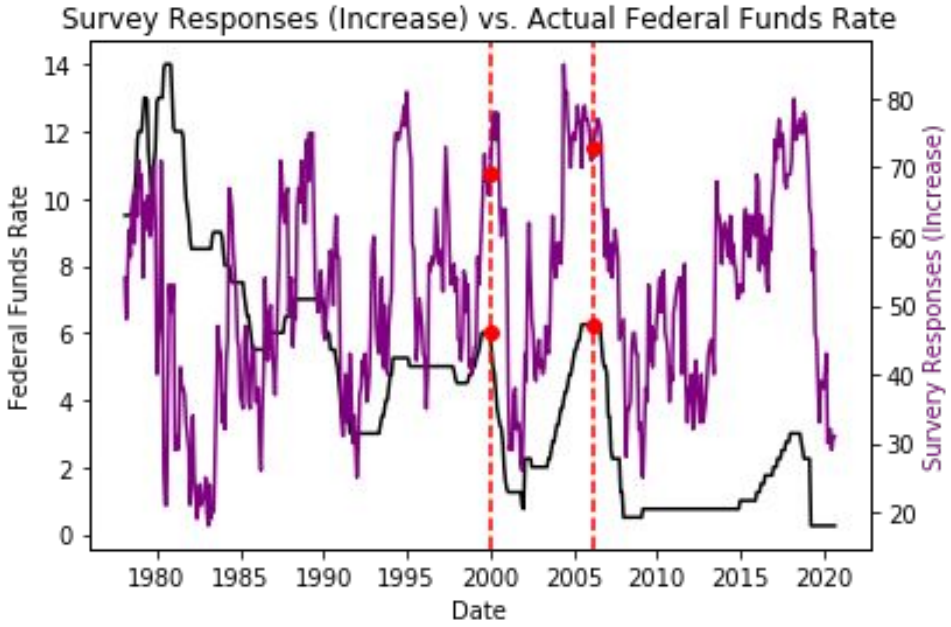
fed_funds_rate	one_month_net_change
12.00	0.00
12.52	-0.52
13.00	-0.48
13.00	0.00
12.94	0.06

OMNC_cat
0
-1
-1
0
1



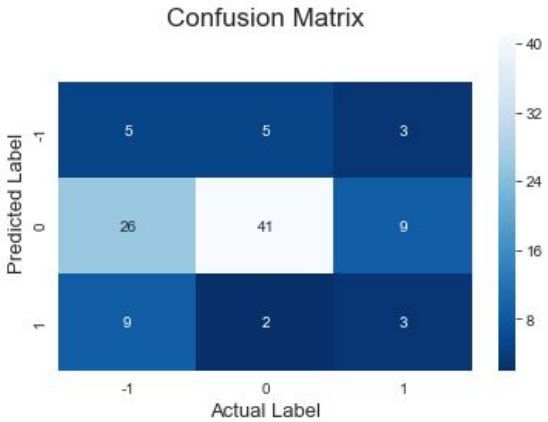
Exploratory Data Analysis

Time series comparison of survey counts for “go up”/increasing vs. federal funds rate:



Modeling/Results ~ Interest Rates

- initial accuracy ~.7 with logistic regression
- we discovered the model was predicting all observations as stagnant, dominant class
- after balancing the class weights, SVM gave an accuracy ~0.55
- better at classifying stagnant interest rates than increasing or decreasing interest rates



*no change ("0" class) highlighted below

	precision	recall	f1-score
-1	0.12	0.38	0.19
0	0.85	0.54	0.66
1	0.20	0.21	0.21
accuracy			0.48
macro avg	0.39	0.38	0.35
weighted avg	0.67	0.48	0.54

Performance Metrics:

Accuracy score = 0.553

Avg. Precision = 0.39

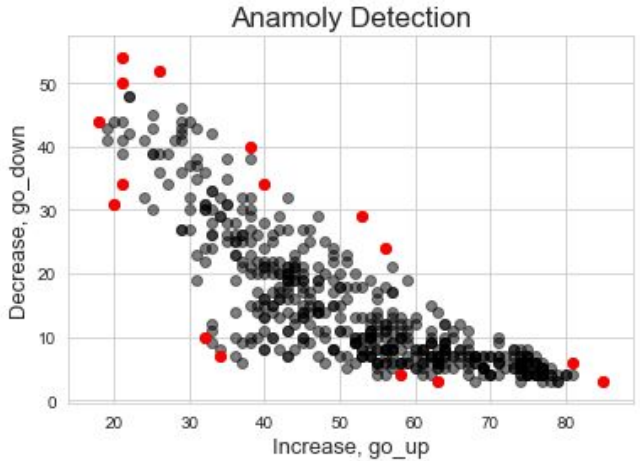
Avg. Recall = 0.38

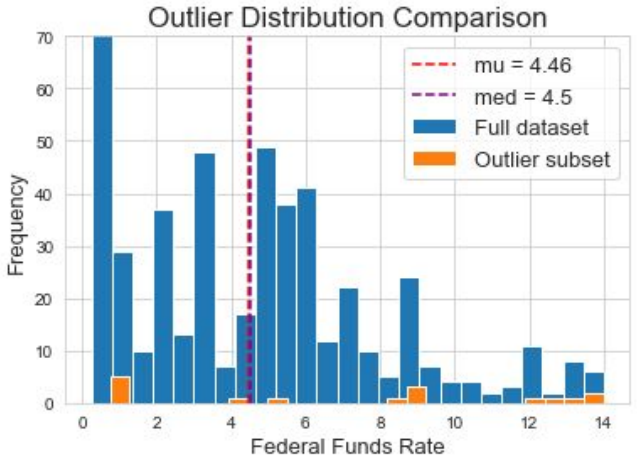
Anomaly/Outlier Detection - OneClassSVM

An example of unsupervised learning using one class support vector machine to identify anomalies in survey responses.

In [110]:
```
# anamoly detection using OneClassSVM - interest rates
cols = ['go_up', 'stay_same', 'go_down', 'dk_na', 'relative']
an_detec = OneClassSVM(kernel = 'rbf', gamma = 0.001, nu = 0.03) # percentage to consider outliers
an_detec.fit(interest_rates[cols])
```

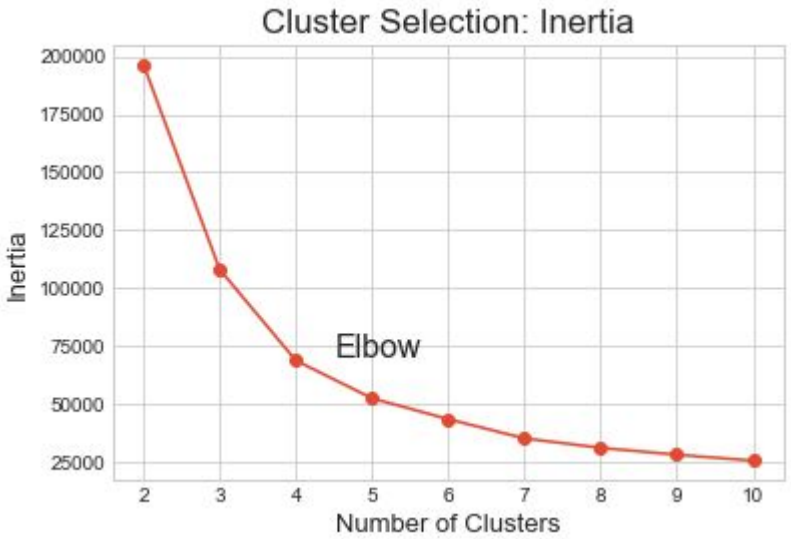
Out[110]:
```
OneClassSVM(cache_size=200, coef0=0.0, degree=3, gamma=0.001, kernel='rbf',
            max_iter=-1, nu=0.03, random_state=None, shrinking=True, tol=0.001,
            verbose=False)
```

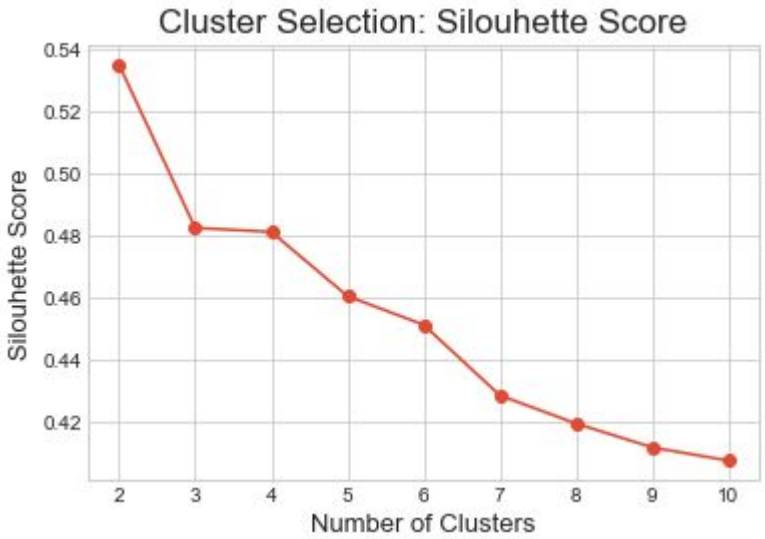




Anomaly/Outlier Detection - KMeans Clustering

Utilizing a KMeans clustering algorithm to identify potential anomalies with a euclidean distance percentile set to 97 (3% of the data will be considered outliers). This allows for comparison to the previous OneClassSVM me[illegible]

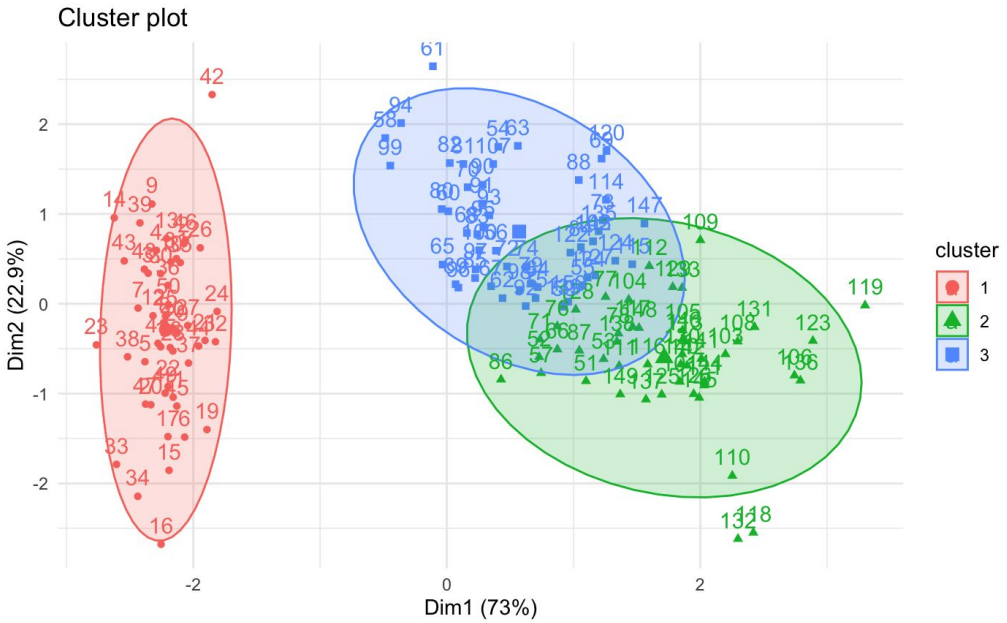





```
an_model = KMeans(n_clusters = 4)
```

KMeans Clustering - Silhouette Score

Silhouette score is the mean silhouette coefficient over all instances.



Silhouette coefficient = (b - a) / max(a,b)

a = the mean distance to the other instances in the same cluster

b = the mean next nearest-cluster distances

Varies between -1 and 1:

- 1: instance is well inside its own cluster
- 0: instance is close to a cluster boundary (ex 147)
- -1: instance may be assigned to the wrong cluster (blue/green overlap)

KMeans Clustering - Filtering/Results

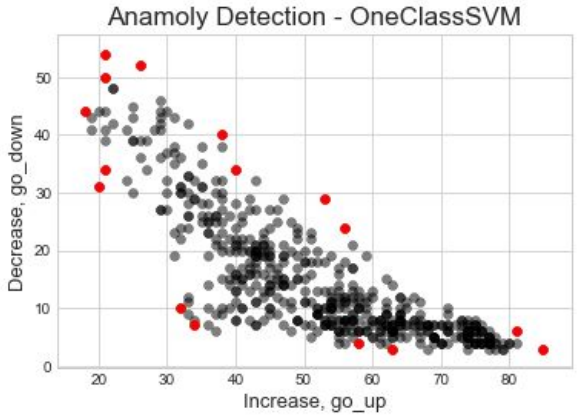
Of the 16 total outliers for each algorithm (3% of data), OneClassSVM and KMeans algorithms shared no similarities.

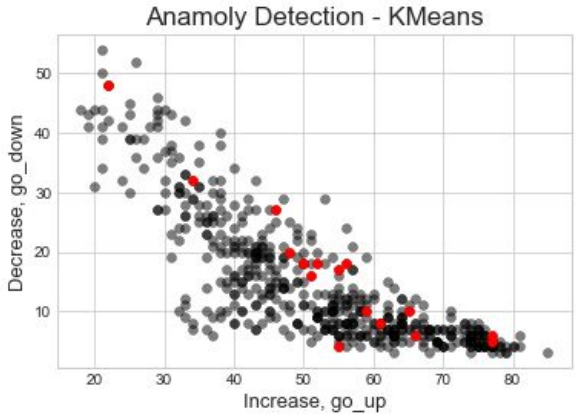
Code: select cluster centroids and calculate distances of each point to its cluster's center, outliers are those with a distance in the 97th or higher percentile

```
centroids = an_model.cluster_centers_
points = np.empty((0,len(X[0])), float)
distances = np.empty((0,len(X[0])), float)

for i,v in enumerate(centroids):
    distances = np.append(distances, cdist([v], X[clusters == i], 'euclidean'))
    points = np.append(points, X[clusters == i], axis=0)

percentile = 97
outliers = X[np.where(distances > np.percentile(distances, [percentile]))]
```





Review/Conclusion

- overall low classification accuracy
- models based on only consumer survey responses should not be used on their own
- could be used in tandem with other sources of information
- utilized effective anomaly detection, comparing different algorithms

Thank you! Questions?