Grouping a Large Dataset of News Documents and predicting its' Classes


dreamstime

1. Frame problem at hand : Predict Classes of Large Dataset New Documents

- Unsupervised Learning -> Large Dataset News Documents -> features -> predict Classes
- Potential Stakeholders
  - Politicians ???
  - Companies??
- Potential Added Value via Project Implementation
  - **Time saved on manual labor**
  - Metric/s
    - **Time**
  - What happens if No Project Implementation
- Current Base Model to measure Potential Added Value??

1.1 Initial Evaluation of Potential Value of Project if Implemented

- Why should my data science team do this project instead of others?
    - Politics are significant in every nation -> Politicians will find said data product very useful in saving precious time
        - Metric to optimize for said project?
            - Most Valuable resource -> Time
- What is the outcome of this step?
    - Politicians able to save precious time

1.2 Determine current approach/ create Baseline Model

- Why do it?
  - Chosen ML model > Baseline Model → ADDED Potential Value?
  - ADDED Potential Value > Cost of Time investment?

Delving into the Data Science Process:

2. Collect the raw data needed for the problem

Regarding the chosen dataset:

- Contains:
  - 18828 Newsgroup documents(messages)
  - 20 different Newsgroups
  - Each message
    - File format
      - Text
- The chosen data set can be found: http://qwone.com/~jason/20Newsgroups/
- much appreciation for Ken Lang for the collection of the data & previous Kaggle contributors of analysis

3. Process & Explore the Data before In-Depth Analysis

Original Data

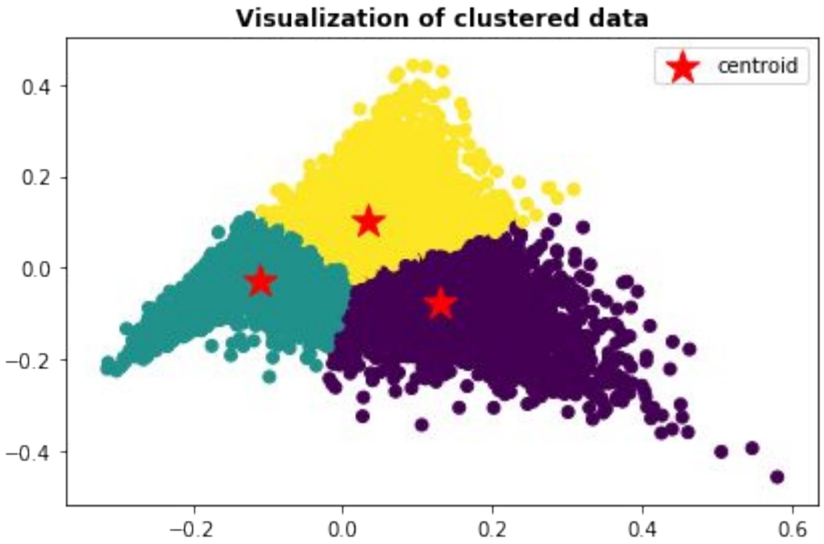
- List of 18828 News Documents
- List of 18828 pathnames of News Documents
- List of 20 Newsgroups each New Document belongs to

Clustering Dataset

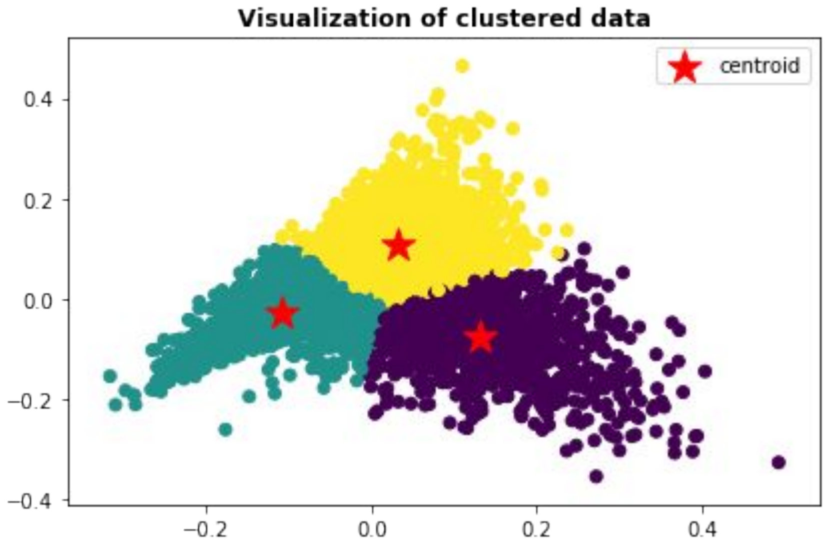
- 5000 features
- 18828 rows

K-Means Clustering

K-Means Training dataset



K-Means Test dataset



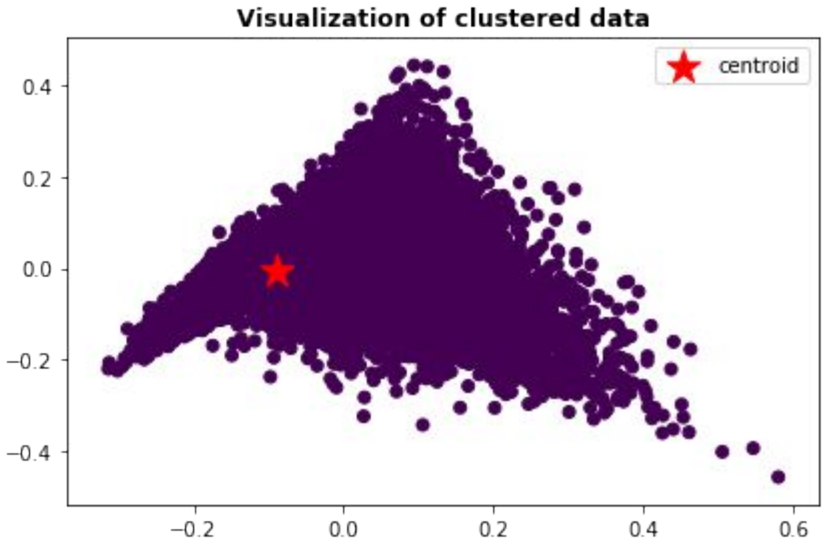
- **Somewhat radially symmetrical isotropic true clusters**
  - **-> somewhat captures underlying patterns**

K-Means Clustering Evaluation

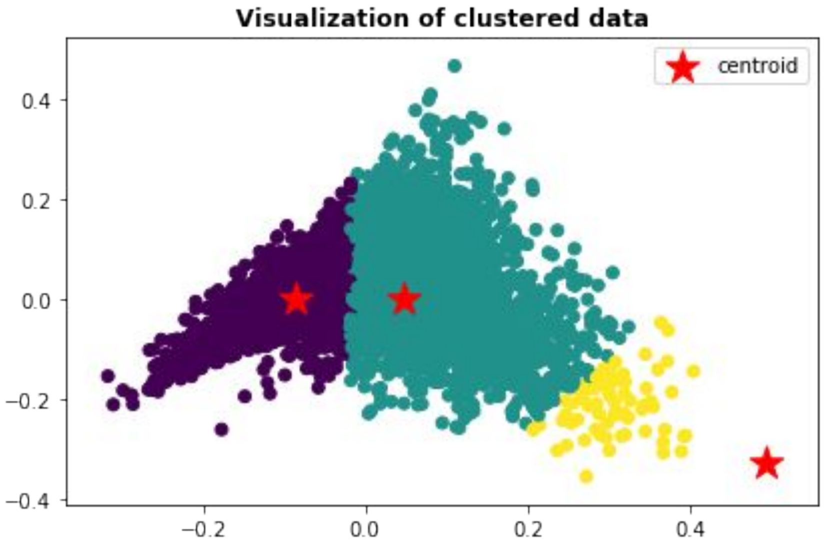
- ARI
  - **0.04 -**
    - **-> relation datapoint pairs ground truth & new solution -> close perfect randomness**
- Similarity Silhoutte Coefficient
  - **.007**
  - **.007**
  - **.006**
  - **.007**
    - **-> consistency coefficients of subsets**
    - **-> samples very close to neighboring clusters**

Mean Shift Clustering

Mean Shift Training dataset



Mean Shift Test dataset



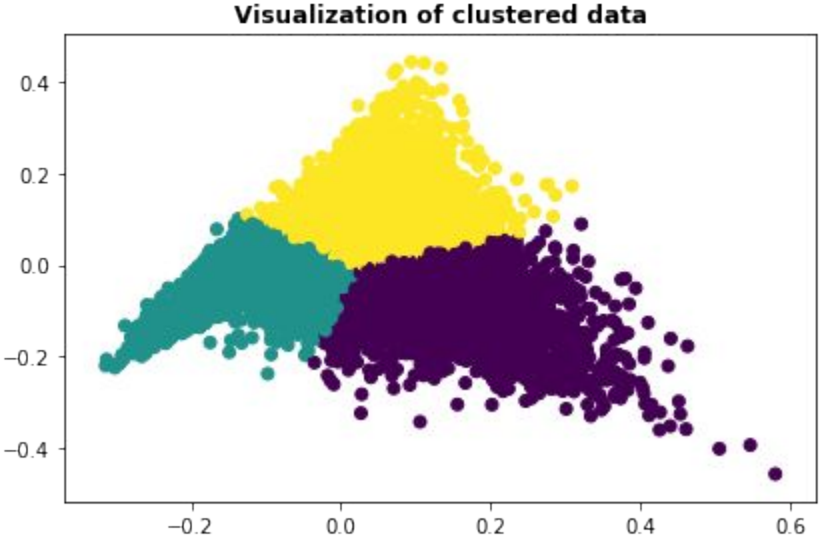
- Somewhat radially symmetric isotropic shape
  - **-> Somewhat captures underlying data patterns**

Mean Shift Clustering Evaluation

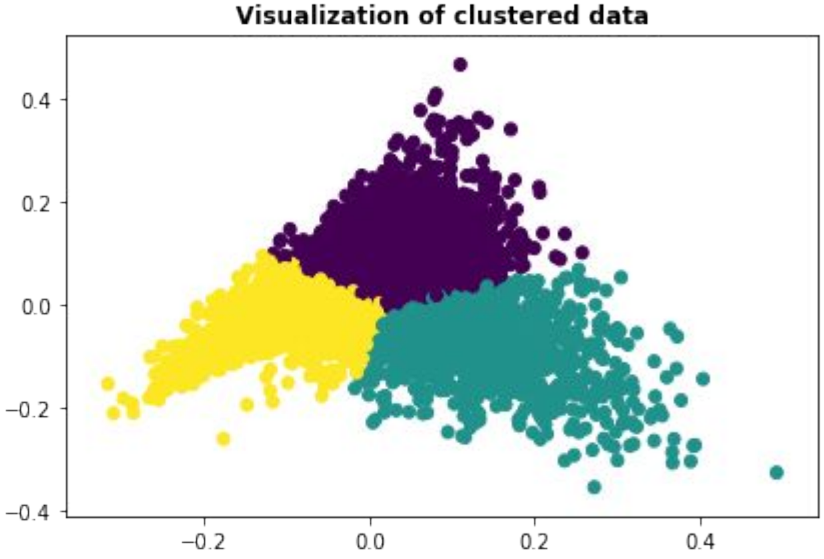
- ARI
  - **.0004**
    - **-> relation datapoint pairs ground truth & new solution -> close perfect randomness**
- Similarity Silhoutte Coefficient
  - **-.06**
  - **-.05**
  - **-.06**
  - **-.06**
    - **Consistency of coefficients between subsets**
    - **Samples assigned to WRONG clusters**

Spectral Clustering

Spectral Training dataset



Spectral Test dataset



- **Somewhat radially symmetric isotropic shape**
  - **-> Somewhat captures underlying data patterns**

Spectral Clustering Evaluation

- ARI
  - **.03**
    - **-> relation datapoint pairs ground truth & new solution -> close perfect randomness**
- Similarity Silhoutte Coefficient
  - **.007**
  - **.007**
  - **.006**
  - **.007**
    - **-> consistency coefficients of subsets**
    - **-> samples very close to neighboring clusters**

Clustering Algorithms Evaluation

WORST in Capturing data patterns:

- MeanShift
  - Least true cluster shape
  - ARI
    - **Ground truth vs new solution most close to perfect randomness**
  - Similarity Silhoutte Coefficient
    - **Negative**
      - **-> samples assigned to wrong clusters**

BEST in Capturing data patterns:

- K-Means vs Spectral
  - **K-Means**
    - **Slightly better ARI evaluation score**

4. In-Depth Analysis

Classification -> Multi Classification

- 20 different classes
  - 3 Classes below in count
    - 1 Class significantly below
      - Class imbalanced
        - Multi Classificaition -> most common ML performance metrics:
          - Average accuracy
          - F1 score
          - Log- loss
          - Mathews Correlation Coefficient
            - Log-loss symmetric -> does not consider class imbalances
            - Average accuracy ? No
            - Mathews very high performance but binary so -> F1 score micro

Training vs Test Accuracy F1- micro Score on ML models:

	Training data set Acc Score	Test data set Acc Score
Random Forest Classifier	**.992**	**.691**
Logistic Regression	**.991**	**.812**
Multinomial Naive Bayes	**.858**	**.802**

- **RFC ml model overfitting immensely -> captures alot of noise**
- **LR ml model overfitting -> still captures noise**
- **MNB ml model not overfitting + not underfitting**
- **Decision Threshold = .5**

5. Communicate Results of analysis (Potential Data Product)

Uncovered Insights for Proposal Implementation

- **Multiclass Multinomial Naive Bayes best + solid ml model performer**
  - **Closest training and test F1- micro scores with training being tad bit higher-**
    - **-> .858 vs .802 not overfitting**
  - **Training F1 - micro score decent**
    - **.858**
      - **-> not underfitting**
  - **Initial Decision Threshold = .5**
    - **.802 > .5 -> positive + F1 score -> positive w uneven class**
  - **For even closer + higher training and test mean accuracy scores;**
    - **dimension reduction on 5000 features?**
    - **experimenting with NLP & Neural Network features**
    - **tuning parameters**

Moving Forward:

- Aim to make data science project **MOST CREDIBLE:**
  - **Need to implement couple ESSENTIAL CRUCIAL STEPS:**
    - **Determine current approach/ create baseline model**
    - **ML monitoring & feedback**
    - **ML model to Baseline model evaluation via A/B Testing**

Link to Corresponding Jupyter Notebook:

https://github.com/pman117/Data_Science_Portfolio/blob/master/End_to_End_Data_Products/Grouping_and_Classifying_Large_Dataset_News_Documents/Grouping_and_Classifying_Large_Dataset_News_Documents.ipynb

Link to Corresponding folder containing entire project:

https://github.com/pman117/Data_Science_Portfolio/tree/master/End_to_End_Data_Products/Grouping_and_Classifying_Large_Dataset_News_Documents