News Muse

-Let's Break the Fake

Ishrak Hayet
Sai Krishna Teja Damaraju
Madhu Peduri
Sushmitha Boddireddy

Breaking News

Team News Muse secures Nobel Prize for Technology

Breaking News

Team News Muse secures Nobel Prize for Technology

Fake News!!!

Introduction

- Fake News
  - Falls within the broader category of misinformation
  - Intended to mislead audience for gaining benefits
  - We consider only textual fake news
- Online News Popularity
  - Popularity - number of audience, activities around some news
  - Metadata plays a key role
    - News construct
    - When was the news published?
    - Did the news have any media content?
    - Etc.

Dataset 1

- Kaggle Fake News Dataset
  - Source: https://www.kaggle.com/c/fake-news/data
  - Instances: 18285 (After removing NaNs)
  - Attributes (total 5):
    - ID
    - Author
    - Title
    - Text
    - Label (fake/real)

Dataset 2

- UCI Online News Popularity Dataset
  - Source: https://archive.ics.uci.edu/ml/datasets/online+news+popularity
  - Instances: 39797 (no NaNs)
  - Attributes (total 61):
    - URL (1)
    - Timedelta (1)
    - Syntactical (6)
    - Topical (11)
    - Sentiment (16)
    - Keyword (10)
    - Media Frequency (4)
    - One-hot encoded publication day (8)
    - Self reference (3)
    - Target variable - number of social media shares (1)

Connecting the Datasets

- 61 features in UCI Online News Popularity Dataset
- News “title” and “text” in Kaggle Fake News Dataset
- Feature extractor [1] to extract 20 features from the “title” and “text of Kaggle Fake News Dataset
- These 20 features of Kaggle dataset are a subset of the 61 features in the UCI dataset - interplay analysis

Goals

- Detection of fake vs. real news
  - Classifier modeled on Kaggle Fake News Dataset
- Identification of popular vs. unpopular news
  - Classifier modeled on UCI Online News Popularity Dataset

Analysis on individual datasets

- (Real) News Recommender
  - Clustering applied to the Kaggle Fake News Dataset
- Kaggle news impact analysis
  - Regressor modeled on UCI Online News Popularity Dataset
  - Applied to Kaggle Fake News Dataset
  - Reliant on domain adaptation techniques

Analysis on interplay of both datasets

Individual Dataset Analysis (1)

Fake News Classifier

on

Kaggle Fake News Dataset

Fake News Classifier

- Classify news articles
  - Fake or Real
- Training Model - Labeled Kaggle Datasource

id	title	author	text	label
0	House Dem Aide: We Didn't Even See Comey's Letter Until Jason Chaffetz Tweeted It	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Letter Until Jason Chaffetz Tweeted It By Darrell Lucus o...	1

Fake News Classifier (continued..)

- Data Cleaning and Preprocessing
  - Remove NAN records and reindex
  - Drop insignificant columns (index and id)
- Feature Engineering
  - Stemming and Lemmatization
  - Remove special characters
  - Change to lowercase
  - Remove stop words
  - Bag of words using Countvectorizer
  - Term frequency and Inverse document frequency vectorizer

Fake News Classifier (continued..)

- Classification Algorithms
  - Multinomial Naive Bayesian
  - Passive Aggressive Classifier
- Observations
  - Best accuracy: 91% (CountVectorizer with PAC)

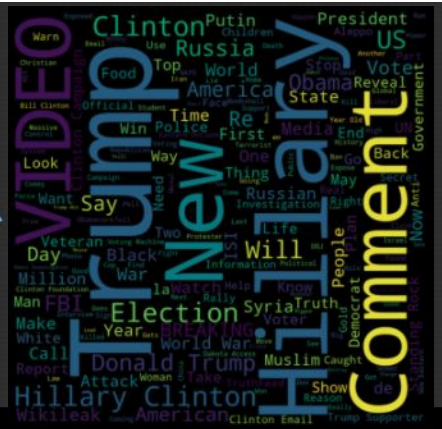
```
The achieved Accuracy: 0.910
Confusion Matrix
 [[1832  208]
 [ 122 1495]]
```

  - Passive Aggressive Classifier better than Multinomial Naive Bayesian
  - Countvectorizer proved to be better than TF-IDF in this case.
  - “Title” over “Text” trade off
    - Title - Under 30 seconds with 90.4 % accuracy
    - Text - 28 minutes with 91% accuracy
  - Author inclusion in the prediction model decreases the accuracy by 40 percent

Fake News Classifier (continued..)

Terms impacting Fake news

```
[(-10.79985853025502, 'said mr'),
 (-10.394393422146857, 'breitbart news'),
 (-9.63670772044934, 'mr obama'),
 (-9.183776075102251, 'ms'),
 (-9.071157196994104, 'presid trump'),
 (-9.008099061026966, 'session'),
 (-8.980700086838851, 'breitbart'),
 (-8.954031839756691, 'sport'),
 (-8.802893210399425, 'islam state'),
 (-8.802893210399425, 'saturday')]
```



Terms impacting Real news

```
[(-4.661627643430595, 'trump'),
 (-4.673426122736554, 'clinton'),
 (-4.855485590680891, 'one'),
 (-4.856272063391502, 'peopl'),
 (-4.913569330764339, 'state'),
 (-4.99061554494066, 'would'),
 (-4.994020996949 72, 'us'),
 (-5.1119796274340334, 'hillari'),
 (-5.113674447341442, 'like'),
 (-5.141305898573496, 'time')]
```

Individual Dataset Analysis (2)

Popularity Classifier
on
UCI Online News Popularity Dataset

UCI news Popularity Dataset

Features

- Using the raw content of a new article, 60 features are derived and one target feature having the number of shares of that news article.
- We have 40k instances with 59 features (excluding URL and target) to be modeled for our data analysis.
- This set of features would cover below three main aspects of the raw content
  - Text aspect
  - Statistical aspect
  - General aspect
- There are both dependent and independent features in our dataset.

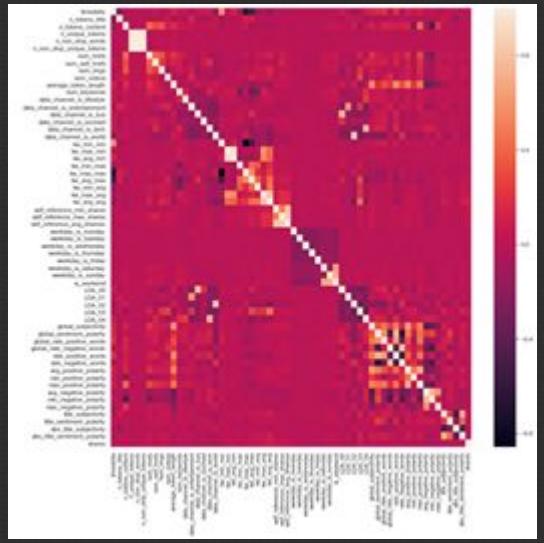
UCI news Popularity Dataset

Exploratory Analysis

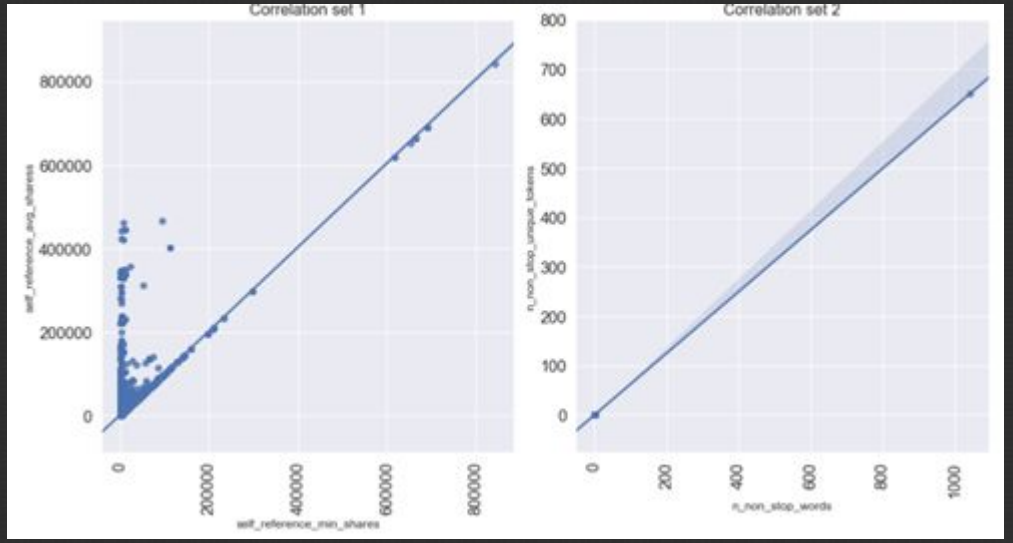
- We have two kinds of features – Binary and continuous
  - Binary – Features that are categorical and created by one hot encoding.
  - Continuous – Features that are derived from statistics (min, avg and max) of language processing aspects
- We have features that are dependent and independent. Our features form different sets. Each set of these features corresponds to an aspect of the news article.
- Features from the same set are dependent and features across the sets are independent.
- We also have the features that have negative values and skewed features.

UCI news Popularity Dataset

Exploratory Analysis



Correlation Heat map



Regression plots between sample features

UCI news Popularity Dataset

Feature Engineering

- In our exploratory data analysis, we found two issues with our data - Presence of negative values and Skewed features.
- For negative features, we added the (-1 * minimum) of negative value to the feature to shift the negative values range to positive range.
- We used min max scaler to normalize the data. This normalization technique would reduce the skewness nature of the features.
- We can see the range of the features is 0 to 1, after normalization

UCI news Popularity Dataset

Feature Engineering

- Our target feature is continuous variable, number of shares of the news article.
- We use 1400 as our threshold for classification. This threshold was provided by the authors of the dataset from their research.
- News articles shared less than 1400 times are Not-popular articles and articles shared more than 1400 times are popular articles.
- We divided our dataset in to 70-30 split for training and test sets.

UCI news Popularity Dataset

Classification

- Multinomial Naïve Bayes:
  - Our data set has features that are independent and few are dependent features.
  - Most of our features are counts generated from text analysis. These features follow multinomial distribution.

```
Times for Training, Prediction: 0.07815, 0.00450
Accuracy for Training, Test sets: 0.62930, 0.63032
-----------------------------------------------------------------------
```

- We used accuracy as our metrics, and we can see 63% of accuracy for our multinomial classification.

UCI news Popularity Dataset

Classification

- Support vector machine (SVM):
  - SVM works well in High dimensional space. It works good for both linear and non-linear separable feature sets.
  - We have a high dimensional dataset (60) and a greater number of statistical features (min, max and avg) of text are involved which are non-linear in nature.

```
Times for Training, Prediction: 232.36762, 63.84094
Accuracy for Training, Test sets: 0.61542, 0.61308
--------------------------------------------------------------------------------
```

- We can see 61% of accuracy for our SVM classification. However, one disadvantage is its training time. We can observe that training time is considerably high compared to other classifiers.

UCI news Popularity Dataset

Classification

- Random forest:
  - All our previous classifiers are parametric both Multinomial bayes and Support vector machines.
  - Random forest classifier is a non-parametric and it can work on all types of data including skewed data.
  - Another useful feature of random classifier is that it provides a metric for feature importance like SVM.

```
------------------------------------------------------------------------
Times for Training, Prediction: 5.90694, 0.03211
Accuracy for Training, Test sets: 0.70404, 0.66050
------------------------------------------------------------------------
```

- We can see 70% of accuracy for our random forest classification. However, we can see difference between Training and test accuracies. This suggest about the possibility of overfitting while training.

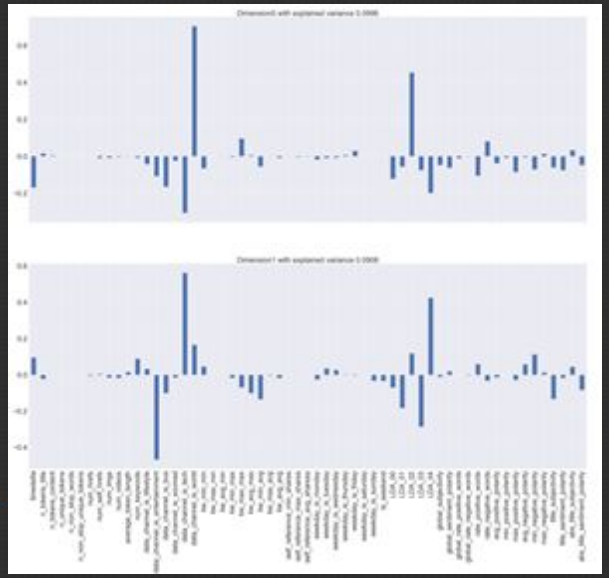
UCI news Popularity Dataset

Dimensionality Reduction

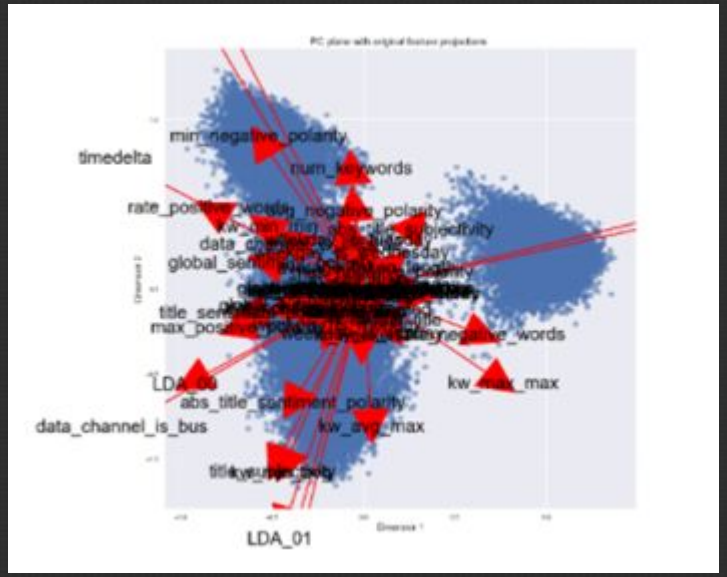
- Principal component analysis (PCA): PCA projects high dimensional data on to a lesser dimensional space by maximizing variance.
- This results in obtaining new set of less features with same essence and maximized variance along their axes.
- We have tried PCA analysis with 6 dimensions but observed that last four dimensions has less explained variance.
- If we reduce dimensions from 6 to 2, we have most of the variance of the original features.

UCI news Popularity Dataset

Dimensionality Reduction



Original features along projected dimensions



Scatter plot of reduced dimensions

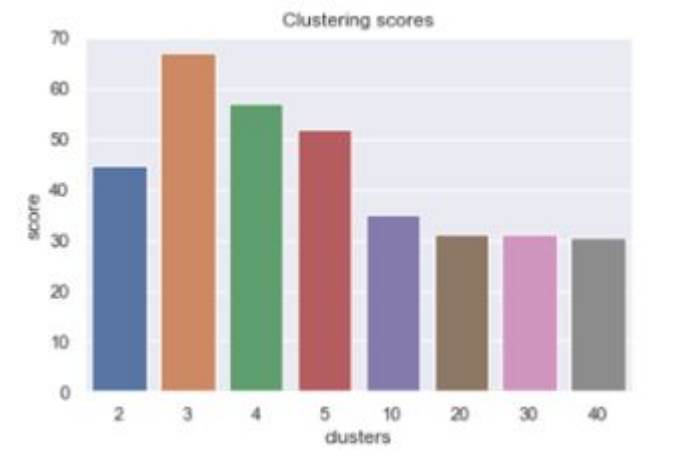
UCI news Popularity Dataset

Clustering

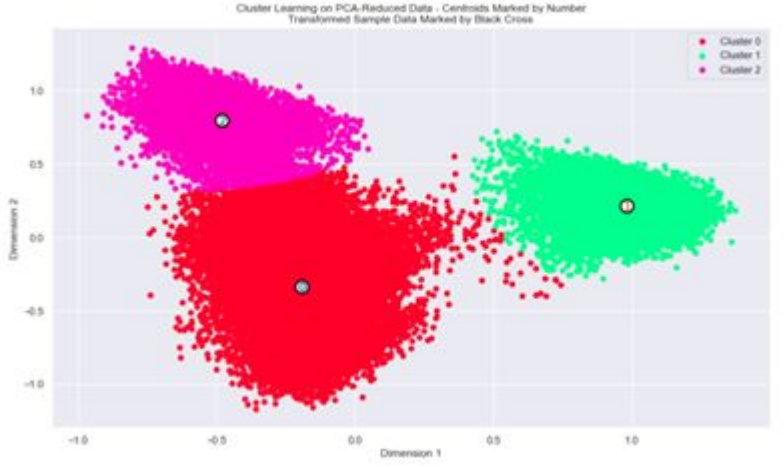
- We have implemented Gaussian mixture clustering on the reduced data to verify the hidden clusters given by PCA.
- We performed clustering algorithm for different number of clusters. We can observe that algorithm has better accuracy when the number of clusters is 3.
- We used silhouette score as our metric to measure the performance of this clustering.

UCI news Popularity Dataset

Clustering



Scores for different clusters

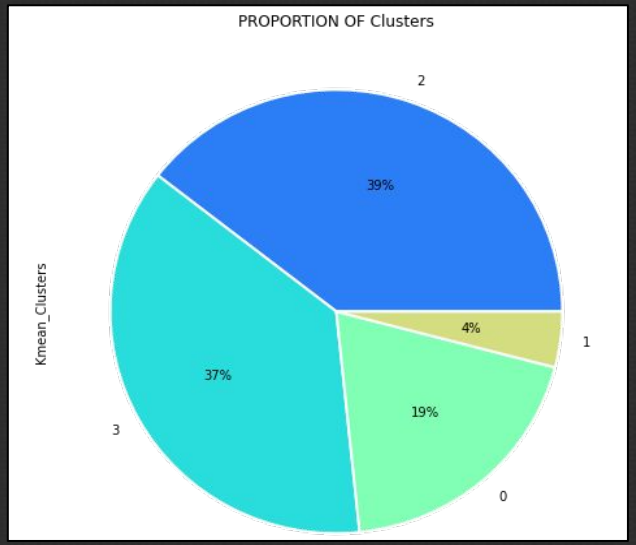


Decision boundary of predicted clusters

Interplay Analysis on Datasets (1)

(Real) News Recommender
(Clustering)

Clustering analysis



- K-means clustering
- Used Feature extraction algorithm on Kaggle fake news dataset.
- Performed clustering on the meta dataset.
- Chosen five features on which we have performed clustering.
- We have used the Kneed algorithm which determines elbow value from the graph to perform K-means clustering.

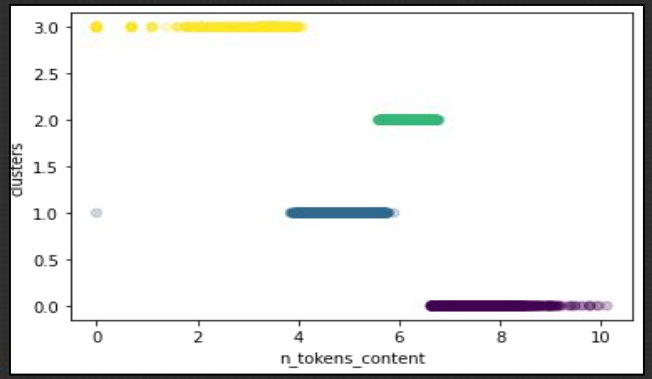
Recommender

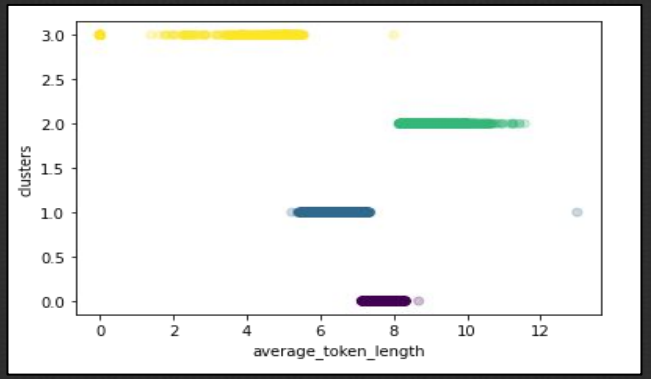
- Predicted the cluster number for the articles in the dataset.
- Calculated number of articles in each cluster.
- Filtered the real news.
- Recommended 5 articles.

	id	author	title
2802	8106	Karen Zraick and Sandra Stevenson	Trump, Price, Pruitt: Your Wednesday Evening B...
2102	6056	John Hayward	Trump Agenda on Offense: 7 Stories in 24 Hours...
5667	16283	John Hayward	Study: Unemployment Fuels National Drug Epidem...
7215	20785	Ann Coulter	Ann Coulter: How to Provide Universal Health C...
7006	20185	Nate Cohn	Is Traditional Polling Underselling Donald Tru...

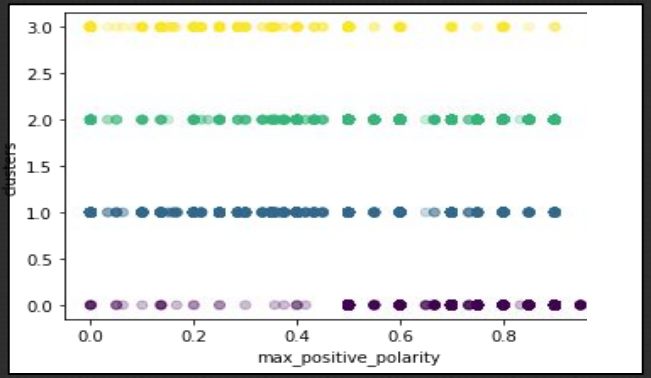
Screenshot of recommended news articles

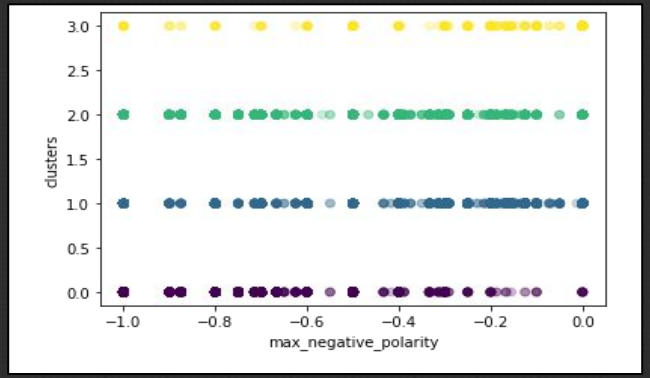
Features which are affecting clusters more

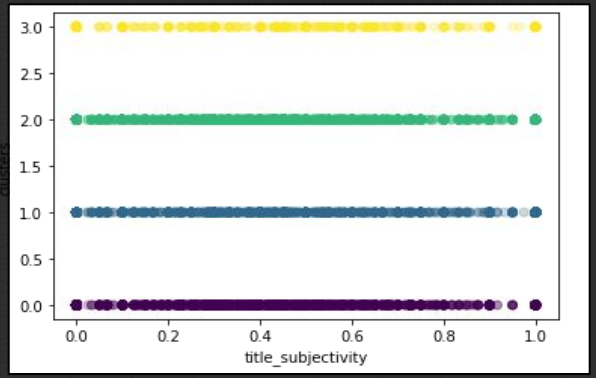




Features which are affecting clusters less







Interplay Analysis on Datasets (2)

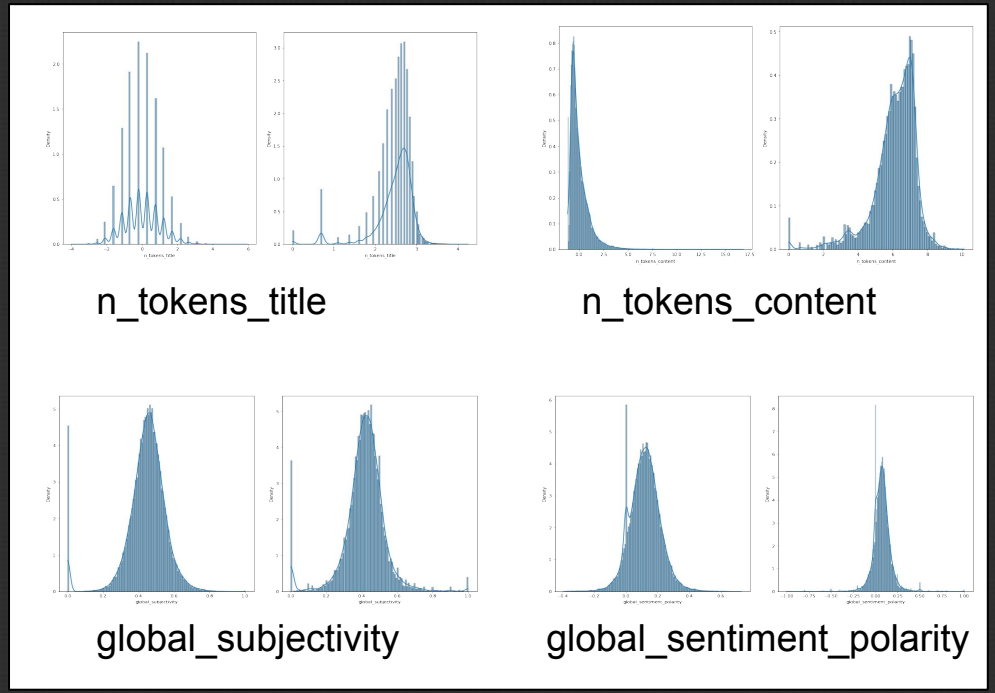
News Impact Analysis
(Regression)

Context

- A quick retrospect
  - 61 features in UCI Online News Popularity Dataset
  - News “title” and “text” in Kaggle Fake News Dataset
  - Feature extractor [1] to extract 20 features from the “title” and “text of Kaggle Fake News Dataset
  - These 20 features of Kaggle dataset are a subset of the 61 features in the UCI dataset - interplay analysis

Distribution Differences (1)

- 4 sample comparisons shown out of 20 common feature distribution comparisons
- For every pair of comparison, left - UCI Dataset, right - extracted Kaggle features
- Some similar and some dissimilar distributions



Jensen Shannon Divergence (JSD)

- Concrete metric to compute the differences between the two distributions
- Jensen Shannon Divergence internally uses Kullback-Leibler divergence
- But, JSD is symmetric (should be fast)
- JSD value range [0, 1]; 0 being similar and 1 being dissimilar distributions
- Some features had very low (very similar) JSD and some had very high (very dissimilar) JSD between UCI and Kaggle datasets

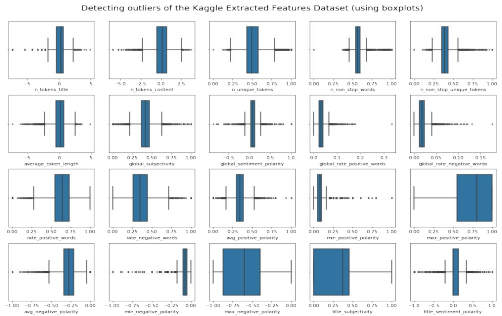
Hypothesis

- Outliers were skewing the datasets - can outlier removal from the datasets improve the distribution divergence score (bring it closer to 0)

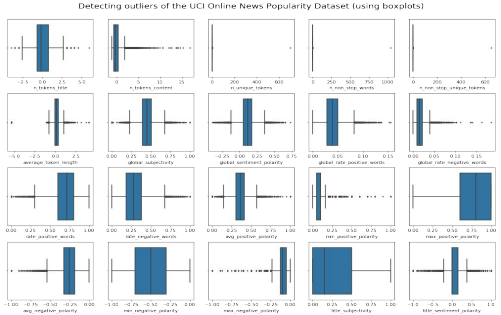
- Can domain adaptation help?

Outliers

- Outlier threshold 1% and 99%
- Average JSD before and after outlier removal: 0.35 and 0.40
- No improvement in divergence
- But, achieving a sense of smoothing (all feature JSD are close to 0.5) (no bias)



UCI Dataset



Kaggle dataset

Domain Adaptation

- Falls under the broader category of transfer learning
- We intend to train a regression model on the source domain (UCI dataset) but test it on the target domain (Kaggle dataset) : diff domains, same task
- During training - more emphasis on the source instances that look like they came from the target domain
- Source instance reweighting: ws = $(p_i t/(1-p_i t))$ [2]
- We get $p_i t$ using a probabilistic meta-classifier that classifies the source and target instances (logistic regression)

Regression Models

- Linear and Non-linear (SVR) are used
- Training is done on the reweighted source instances (UCI dataset)
- Testing is done on the target instances (Kaggle dataset)
- Evaluation:
  - Linear Regression: RMSE 8858
  - Non-linear Regression (SVR - rbf kernel): RMSE 8909

Deployment

AWS Sagemaker and React

Deployment Training

- Scikit learn estimator in a docker container for Sagemaker
- Model training happens in the backend using a t4.medium instance
- Model deployed through AWS Sagemaker using the Scikit learn estimator
- Deployed model is exposed as an HTTP endpoint

Deployment Testing

- Frontend interface is built with react
- Deployed model is invoked from react using aws-sdk
- Model invocation follows the following workflow:
  - **model_fn:** retrieves the saved model using joblib
  - **input_fn:** json input parsing formatting based on model requirements
  - **predict_fn:** connecting the retrieved model and formatted input
  - **output_fn:** formatting the prediction output into json

References

[1] K. Fernandes, P. Vinagre and P. Cortez. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.

[2] Transfer learning: domain adaptation by instance-reweighting - https://johanndejong.wordpress.com/2017/10/15/transfer-learning-domain-adaptation-by-instance-reweighting/