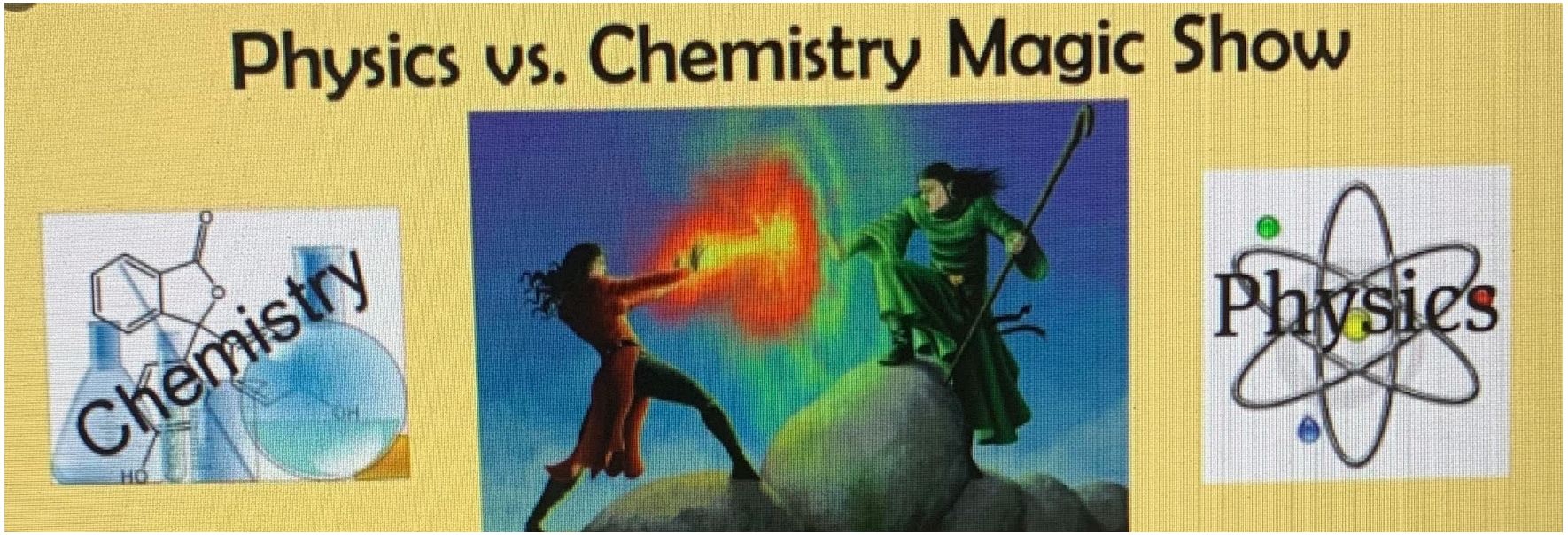


DSI Program

Project_3

By

Dereje Workneh

April 24, 2020

Project Goal: Classification of comments from two subreddits
Process:

- Data Collection: Reddit and Pushshift APIs
- Data Cleaning and EDA
- Preprocessing and Modeling
- Evaluation
- Conclusion and Recommendation

Data Source

- Url = https://api.pushshift.io/reddit/search/comment
- Reddit API:
- 100 posts per request & limited to 1,000 most recent posts total
- more difficult to download comments
- more cleaning of text for html tags, etc. may be required.
- **pushshift.io: (open data initiative to make social media data available for researchers and academic institutions)**
- 500 posts per request & no limit to requests
- and comments available searchable by various parameters

DATA COMPONENTS

- Title of the post
- Subreddit
- number of Comments

Problem to be solved

- What motivates readers up?
- What makes Reader engagement on Reddit
- What Post is Most Attractive

Data Collection

- Using the pushshift.io Reddit API, I downloaded:

- 20,000 submissions (10,000 each from /r/Physics and /r/Chemistry subreddits)

- 20,000 comments (10,000 each from /r/Physics and /r/Chemistry subreddits)

- I analyzed comments for this project, as Physics and chemistry submissions are mostly text

Data Cleaning / Preprocessing

- Cleaning:
- dropped duplicates (mod bot messages, etc.)

- re.sub() to remove: html, hyperlinks, punctuation, words with 2 or fewer letters, whitespace including line returns, non-standard characters (emoji)

- after cleaning: 14,426 -> 13,672 records

EDA: most frequent words

- **Physics:**

['book', 'doe', 'don', 'energy', 'field', 'force', 'good', 'ha', 'hole', 'just', 'know', 'light', 'like', 'look', 'lot', 'make', 'mass', 'mean', 'need', 'particle', 'people', 'physic', 'point', 'quantum', 'question', 'really', 'say', 'theory', 'thing', 'think', 'time', 'wa', 'want', 'way', 'work']

- **Chemistry:**

['acid', 'chemical', 'chemistry', 'did', 'doe', 'don', 'good', 'ha', 'just', 'know', 'lab', 'like', 'look', 'lot', 'make', 'need', 'organic', 'people', 'probably', 'reaction', 'really', 'right', 'solution', 'sure', 'thanks', 'thing', 'think', 'time', 'use', 'used', 'wa', 'want', 'water', 'way', 'work', 'year']

Data Preprocessing

➢CountVectorizer:

- Baseline logistic regression model train/test scores:
- 0.9230 / 0.8063

➢Tf-idf:

- Baseline logistic regression model train/test scores:
- 0.8845 / 0.8087

Data Preprocessing

- **Stop Words:**
- Baseline logistic regression model using standard English stop words:
  - train/test scores: 0.9230 / 0.8063,
  - using additional stop words: 0.9231 / 0.8060

- Baseline random forest model using standard English stop words:

features with highest feature importance values included “wa”, “don”, “ha”, “isn”

- Adding these to stop words didn’t have much effect on this model - before train/test scores: 0.9847 / 0.7760, after: 0.9847 / 0.7760

Data Preprocessing

- **n-grams: (1 - 3):**

- CountVectorizer & Logistic regression:
- top 20 features all 1-grams except ‘just need’, train/test scores: 0.9277 / 0.8031

- Tf-idfVectorizer & Logistic regression:
- top 20 features all 1-grams except ‘sound like’, train/test scores: 0.8856 / 0.8084

Models

➢**Logistic regression:**

- Gridsearch best params: C = 1.0, penalty: l2 (ridge)
- Train / test scores: 0.8002 / 0.8092

➢**Random forest:**

- Gridsearch best params: max depth: None, n_estimators: 30
- Train / test scores: 0.0.7696 / 0.7706

➢**Multinomial naive Bayes:**

- Gridsearch best params: alpha: 0.5
- Train / test scores: 0.8202 / 0.8210

Conclusions

➢ **Physics vs Chemistry :**

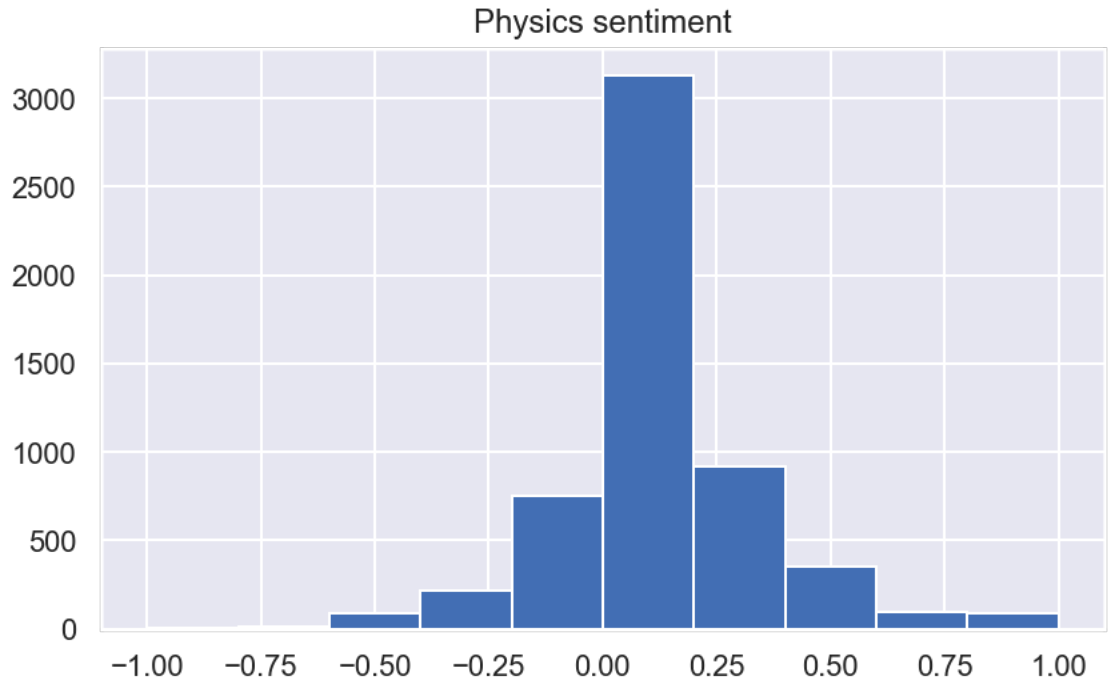
✓The differences outweigh the similarities for NLP and classification modeling Best scoring model: **Multinomial naive Bayes**, Train / test scores: 0.8202 / 0.8210

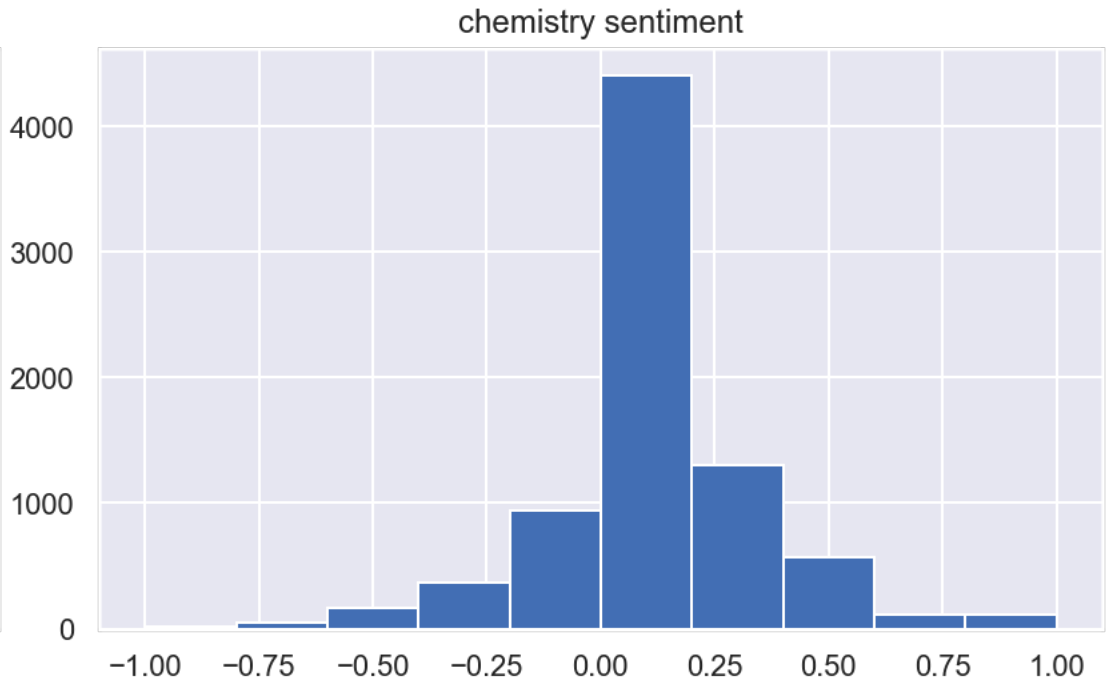
Next Steps

- Model accuracy score
- Engineering additional Features
- Grid search features
- Collection of posts by time
- Potential improvements: collect more training data, do more data cleaning and preprocessing (remove more stop words i.e. numbers, stem/lemmatize i.e. -ing verbs), more intensive gridsearching to optimize models, try more models (boosting, SVM)

Additional Work: Sentiment Analysis

- sentiment analysis with TextBlob.sentiment.polarity





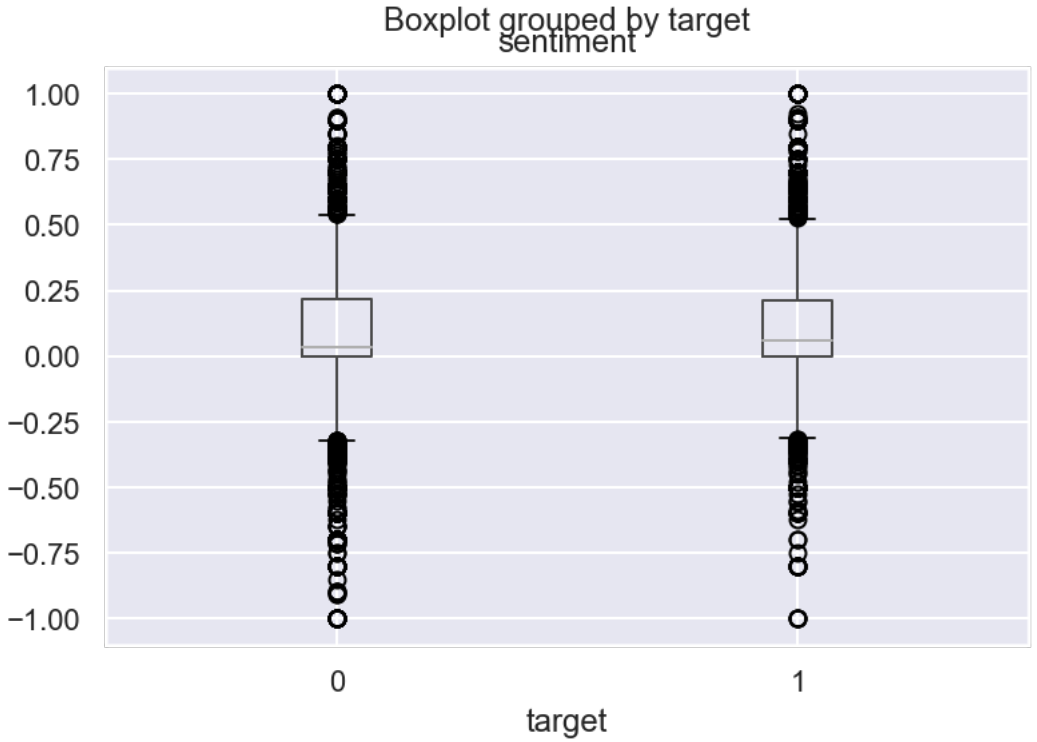
Additional Work: Sentiment Analysis

➢**mean sentiment:**

- Physics: 0.107
- Chemistry: 0.097

➢**median sentiment**:

- Physics: 0.057
- Chemistry: 0.031
- Physics have more comments 0.1 and above



THANK YOU !

TIME TO QUESTIONS AND COMMENTS