Project 3:
NLP Classifiers

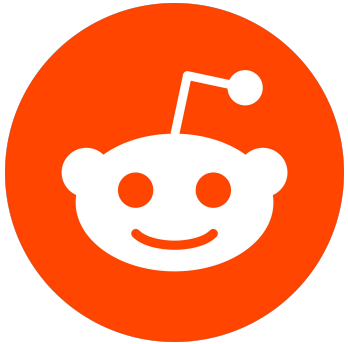
Using Reddit's API to predict post content from ***r/audioengineering*** *and* ***r/livesound***

Thomas Ludlow, DSI-US-6
December 21, 2018

trouble@reddit.com

How can we use predictive modeling to best predict which subreddit a post came from?

Reddit's Web API



- Returns data in ***.json*** dictionary format
- Allows iterated requests
  - Provides ***after*** value for reference
  - Limits data to ~1,000 posts at a time
- Accessible with Python's ***requests*** library

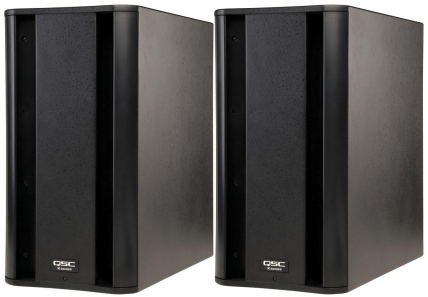
Selected subreddits:

r/audioengineering

- **123,000** subscribers
- “Products, practices and stories about the profession or hobby of recording, editing, or producing audio.”
- API yield: **924 posts**

r/livesound

- **32,300** subscribers
- “A place for audio humans.”
- API yield: **980 posts**



← pair of subs

Exploratory Data Analysis (EDA)

- Convert dictionary data into Pandas DataFrames
- Remove blanks, duplicates, and ***HTML*** formatting
- Create single index
- Add our binary classes:
  - ***is_ls = 1*** (livesound)
  - ***is_ls = 0*** (audioengineering)

Goal: Predict whether a post came from r/livesound

Pre-Processing

Prepare data for Natural Language Processing (NLP) models

Tokenizing

- ***nltk*** library's ***RegexpTokenizer***
- Uses Regular Expressions to parse string into multiple tokens

Stemming

- Reduces token words to stems by removing suffixes

Lemmatizing

- Reduces tokens with a lighter touch than stemming

Vectorizing

CountVectorizer

- Converts string into matrix of counts of each token
- Set ***stop_words*** to remove common English words
- Set ***ngram_range*** to add multi-word combinations

TfidfVectorizer (TF-IDF)

- “Term Frequency - Inverse Document Frequency”
- Converts strings to matrix with float values
- ***stop_words, ngram_range***

Model Selection

Use ***sklearn***'s ***GridSearchCV*** to optimize over a range of models and hyperparameters

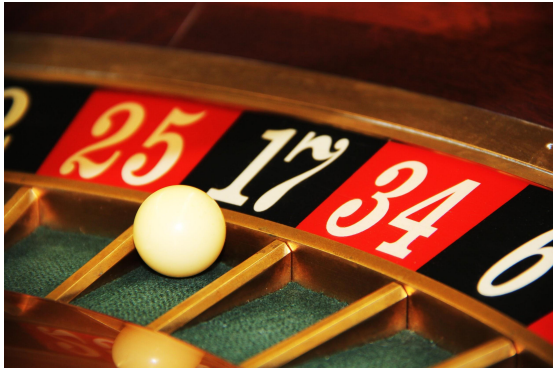
Pipeline groups models with pre-processors

Create indexed ***train_test_split*** vectors to evaluate performance

1. **Multinomial Naive-Bayes**
   - Lemmatized
2. **Random Forest**
   - Lemmatized
3. **Gradient Boost**
   - Decision Tree
   - Lemmatized
4. **Logistic Regression**
   - TF-IDF

Multinomial Naive-Bayes

- Used to classify vectors based on probabilities of the training data variables
- Pros: Quick to build, reliable predictions
- Cons: Tough to interpret,
  unreliable predicted probabilities



Random Forest / Gradient Boost

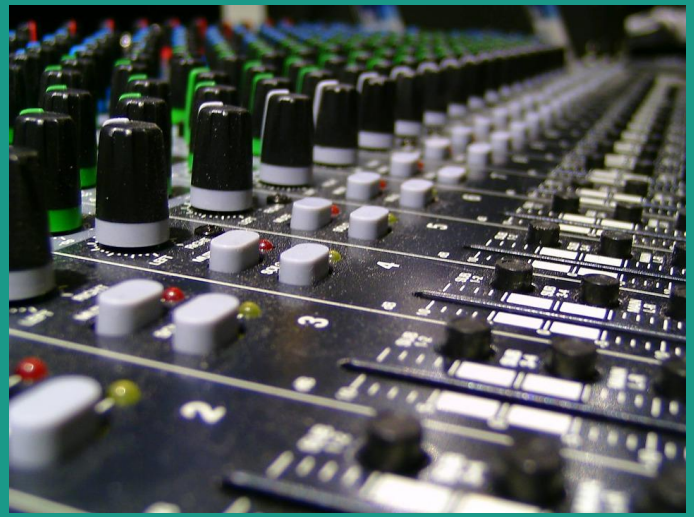
- Built using Decision Tree models
- Trees use binary nodes to subdivide training data for predictions
- **Random Forest**
  - Generates numerous trees with random replacement sampling, and a combined predictive model (***bagging*** method)
- **Gradient Boost**
  - Iterates a single model, adjusting to the previous run's residuals (***boosting*** method)

Logistic Regression

- Regression classifier predicting probability of **1** or **0** value for target
- **Easily interpretable coefficients**

Model Evaluation

- Obtain newer subreddit posts via API
  - **r/audioengineering: 79** new posts
  - **r/livesound: 48** new posts
- Tested models individually and using ***sklearn ensemble*** techniques
- Model to **baseline accuracy**: **51.4%**



Model Accuracy Scores

1. **Multinomial Naive-Bayes**
   Train/Test: **83.4%** New Posts: **84.3%**
2. **Random Forest**
   Train/Test: **77.3%** New Posts: **76.4%**
3. **Gradient Boost**
   Train/Test: **80.5%** New Posts: **81.9%**
4. **Linear Regression**
   Train/Test: **81.3%** New Posts: **85.0%**

Voting Classifier Accuracy

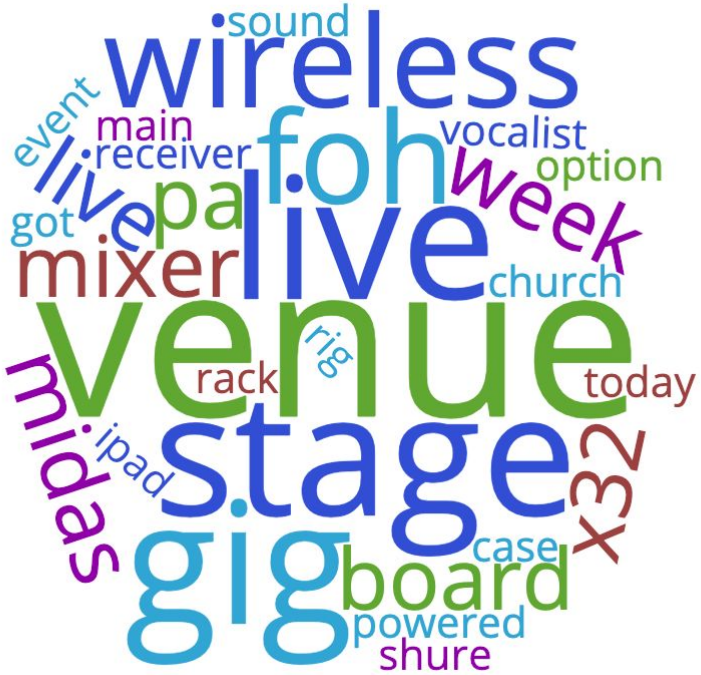
Takes an even vote of all 4 models

Train/Test: **83.0%**

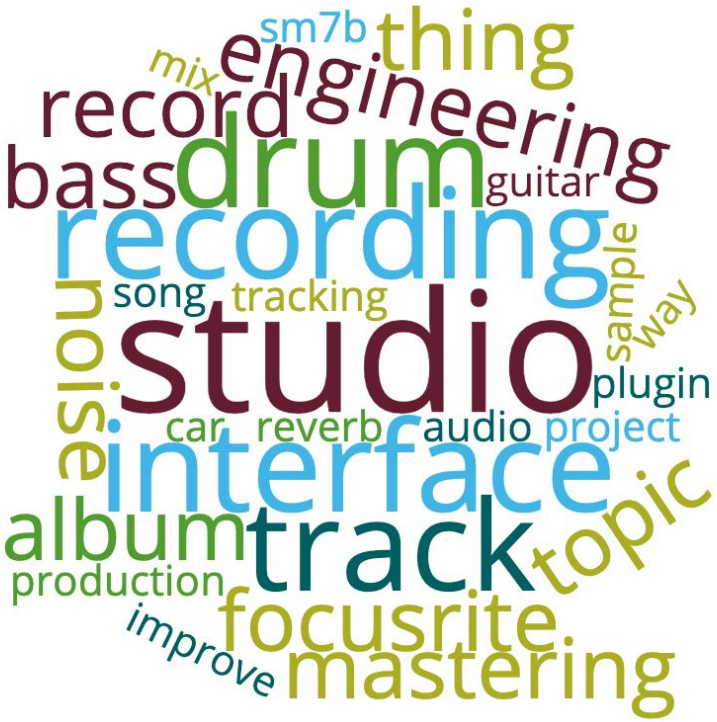
New Data: **88.98%**

Key terms for each sub:

r/livesound



r/audioengineering



Recommendations for Reddit

- Highest-performing model is **Logistic Regression**, and coefficients allow us to understand the data easily
- Use ensemble modeling methods like ***sklearn***'s ***VotingClassifier*** to improve accuracy
- Develop multiple models for targeted purposes
  - **Random Forest** had the lowest accuracy, but it has **91.7%** specificity
- Use models to forecast and budget for manual administrator review

Questions?

Thank you!