GENRE-BOT

USING NATURAL LANGUAGE PROCESSING IN MUSIC MARKETING

PROVIDING THE RIGHT CONTENT TO THE RIGHT LISTENERS

- Listeners have music genre affinities, either self-identified or observed from listening habits
- Social media aggregation tools, such as Sprout, can provide us with a wealth of user-generated text content
- Top terms from that content tell us what's trending in each genre
- We can mine those terms for marketing and music recommendation opportunities

sproutsocial

CROSS-GENRE MARKETING

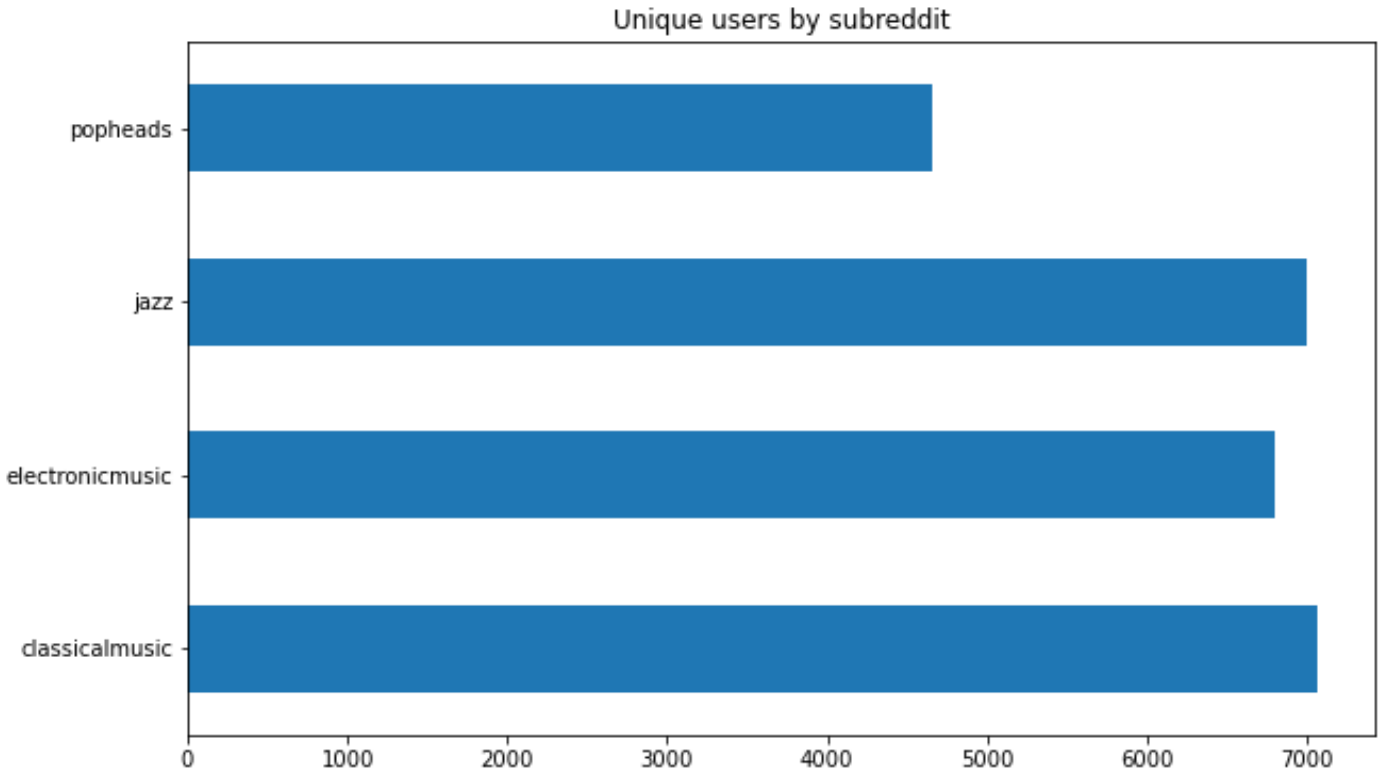
DATA OVERLOAD

- Sprout sends me 1000s of user-generated documents a day from a variety of social media sources
- It's too much for a human to analyze!
- So….what do we do?

SUBREDDIT SAMPLING

- Data imported from music subreddits such as **classicalmusic** and **jazz** via the PushShift API
- Posts sampled proportionally across the lifetime of the subreddit to get complete picture
- 10,000 posts per subreddit used for training
- Text content for each document is a cleaned version of the title + the post text

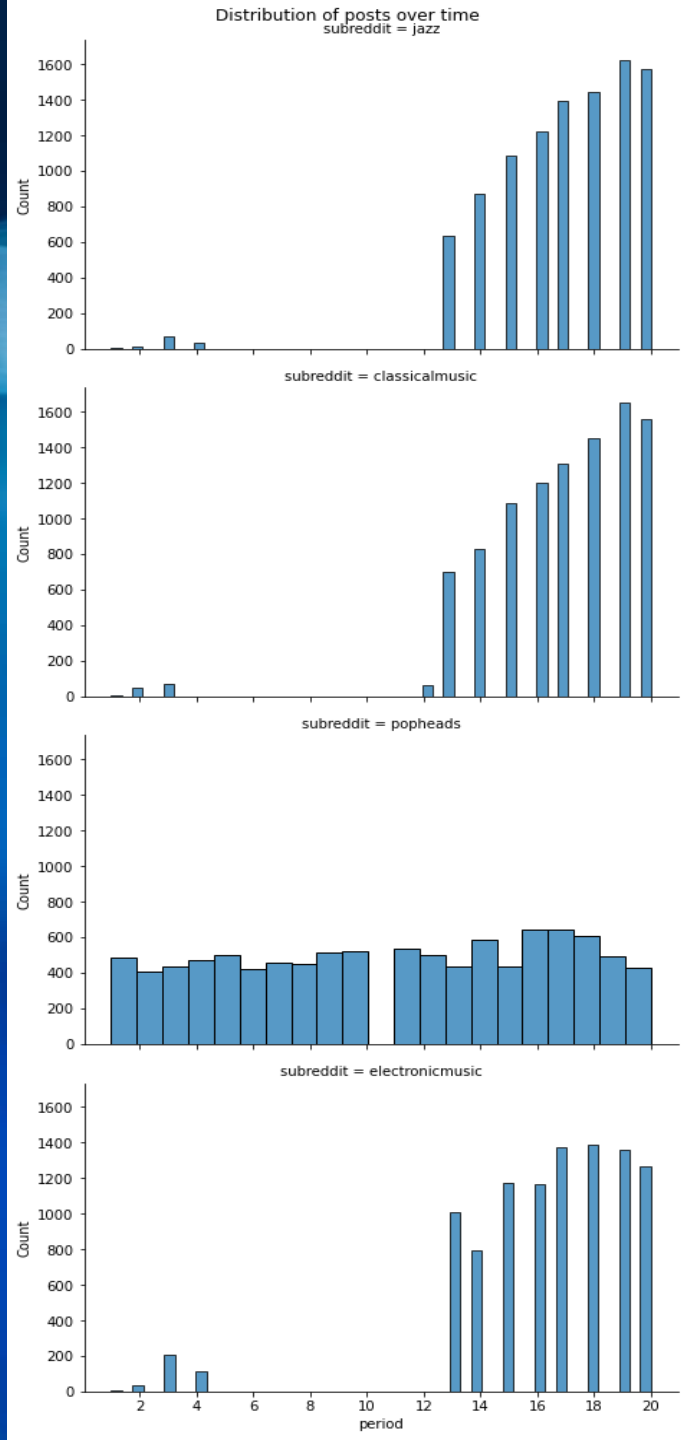
PLENTY OF POSTERS



- Unique post authors are mostly well-distributed per genre
- **Popheads** looks a little suspect – fewer authors, probably over-represented writing styles

A LIFETIME OF POSTS

- **Jazz, classicalmusic** and **electronicmusic** all look very similar, including the gap in the middle
- **Popheads** sticks out here too
- So we probably won't use **popheads** to train



MYRIAD OF MODELS

- **15** models trained on **classicalmusic** and **jazz** subreddits
- All but **2** achieved our minimum desired accuracy of **80%**
- Stemming/lemmatization
- Count vs TFIDF vectorization
- Logistic regression, K-nearest neighbors, decision trees + tree ensembles, boosted models, stacked models
- Collected **train accuracy**, **test accuracy**, **precision, recall, F1,** and **time required to fit**

MYRIAD OF MODELS

	model	fit_time	train_acc	test_acc	recall	precision	f1
0	CountVectorizer + LogisticRegression	0.874005	0.975254	0.931795	0.912048	0.949415	0.930356
1	TfidfVectorizer + LogisticRegression	1.495226	0.956996	0.93681	0.917671	0.954071	0.935517
2	TfidfVectorizer + tuned LogisticRegression	4.055915	0.955257	0.938014	0.918474	0.955704	0.936719
3	TFIDF + MultinomialNB	1.296156	0.941413	0.927984	0.925703	0.92981	0.927752
4	TfidfVectorizer + Bagging	28.872257	0.992376	0.906319	0.884739	0.924465	0.904166
5	TfidfVectorizer + Random Forest	2.628158	0.91446	0.90331	0.841365	0.960128	0.896832
6	TfidfVectorizer + ExtraTrees	1.858499	0.92175	0.911735	0.89759	0.923554	0.910387
7	TfidfVectorizer + AdaBoost	7.38828	0.900482	0.895286	0.832129	0.952206	0.888127
8	TfidfVectorizer + GradientBoost	2.421923	0.933119	0.924574	0.883534	0.96238	0.921273
9	TfidfVectorizer + LogReg/MNB/Grad Stack	15.633441	0.952782	0.937613	0.926908	0.947066	0.936878
10	TfidfVectorizer + MNB/Grad stack	4.415673	0.945693	0.931996	0.922892	0.939877	0.931307
11	TfidfVectorizer + Logreg/Grad stack	14.953498	0.957665	0.937212	0.92249	0.950352	0.936214
12	TfidfVectorizer + Logreg/MNB stack	10.692293	0.957263	0.941424	0.931727	0.950041	0.940795

PICKING A WINNER

- We mainly care about **test accuracy**, overfitting delta from **train to test**, and **time required to fit**
- Weighted sum of these 3 scaled factors – 4:2:1 ratio
- Overall scores calculated, and the winner is…

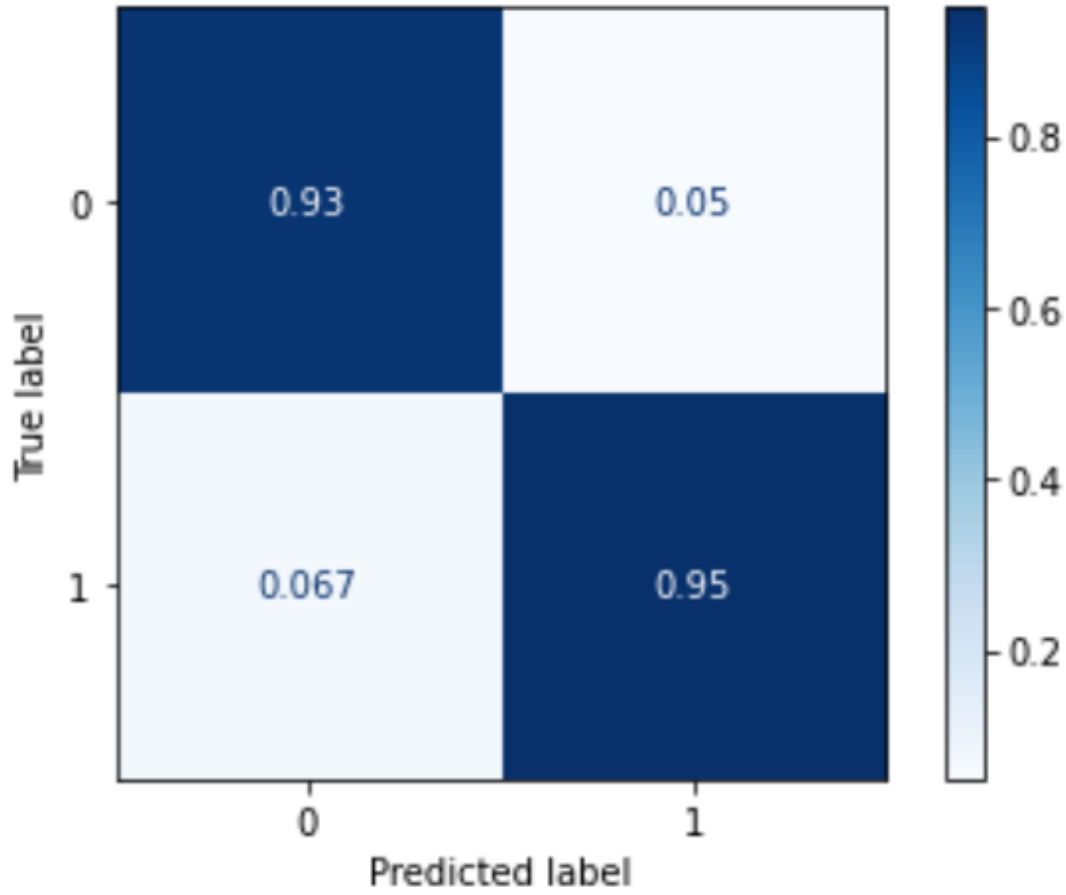
THE VERY LAST ONE

TFIDF + STACK OF LOGISTIC REGRESSION + MULTINOMIAL NAÏVE BAYES

```
tfidf = TfidfVectorizer(max_df = 0.8, max_features = 4000, stop_words = 'english')
X_vec_train = tfidf.fit_transform(X_train)
X_vec_test = tfidf.transform(X_test)

logreg = LogisticRegression(C=1, l1_ratio=0.1, max_iter=10000, penalty='elasticnet',
                    solver='saga')
mnb = MultinomialNB()

l1_estimators = [('logreg', logreg), ('mnb', mnb)]

stack = StackingClassifier(l1_estimators)
```

HOW'D WE DO?

- **Train accuracy** of **95.7%**
- **Test accuracy** of **94.1%**
- **Recall** of **93.2%**
- **Precision** of **95.0%**
- **F1 score** of **94.1%**
- **Final fit time** of **10.1 seconds**
- **True = jazz**



HOW'D WE DO?

- Misclassified **122** classical posts as jazz
  - `Steps to take to become a singer`
  - `soundtrack transcribing to notes or synthesia help`
  - `Sambo Godly Talk`
- Misclassified **170** jazz posts as classical
  - `The Clash Walk Evil Talk The Clash Walk Evil Talk`
  - `Looking for upbeat peppy music like the Ren Stimpy theme song`
  - `Pain`

TOP 10s

JAZZ WORDS

- jazz 2753
- like 936
- music 719
- know 637
- just 620
- ve 515
- really 437
- looking 420
- play 372
- new 363

JAZZ BIGRAMS

- **miles davis** 136
- don know 92
- **big band** 90
- jazz music 75
- does know 69
- new jazz 67
- **john coltrane** 60
- jazz albums 58
- **charlie parker** 55
- feel like 54

CLASSICAL WORDS

- music 1824
- classical 1108
- like 803
- piece 676
- pieces 639
- piano 552
- know 545
- just 510
- ve 426
- help 355

CLASSICAL BIGRAMS

- classical music 600
- don know 84
- **piano concerto** 81
- sheet music 55
- **th century** 51
- feel like 51
- does know 50
- ve heard 49
- thanks advance 46
- classical pieces 45

WHAT'S NEXT?

- Stopwords are our friends and we need to make more
- Experiments in clustering and PCA…outlook is murky, supervised seems the way to go
- Multi-class problem – there are more than 2 genres in the world…outlook is promising!

THANK YOU

GABRIELMANGIANTE@GMAIL.COM