LYrec

The future of bad song recommendations

THE GOAL

Create a content based recommender which recommends songs purely based on lyrics

Questions

- Can we accurately compute lyrical similarity?
- Is lyrical similarity a good predictor of user preference?

Challenge 1- Finding data

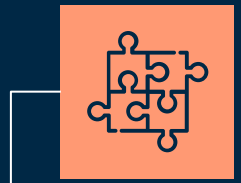
1. Most datasets do not have lyrics due to copyright concerns
2. Most user-song preference datasets were anonymous with respect to both users and songs
3. Matching between datasets by song title is very difficult
4. Many datasets were too big >100GB or too small <.25GB

Our Data

01

BIGML dataset

A dataset containing ~600,000 songs with titles and lyrics



02

Million Songs Dataset

A dataset containing ~1,000,000 songs with songID's and song names but no lyrics

03

Echo Nest Taste Dataset

A dataset containing ~120,000 SongID/UserID/TimesPlayed triplets

Challenge 2- cleaning the data

Text Encodings

We wanted our data in UTF-8 encoding. Some of the data in the dataset was clearly misencoded. We used the FTFY python library to help us reencode

Stopwords

Stopwords such as “the”, “is”, and “and” mess up our models by introducing noise. We use NLTK to remove the stopwords.

Languages

Some of our data was in other languages. For the scope of this project we only care about english lyrics, so if a song used unknown characters, it was ignored

Bag of Words Model

Our first recommender system

01

What is Bag of Words?

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1

Advantages/issues Bag of Words

BoW is good for simple analysis

1. Bag of Words finds direct similarity between songs
2. Fast and easy to write
3. Accuracy was increased as we removed stopwords and did preprocessing

Disadvantages of BoW

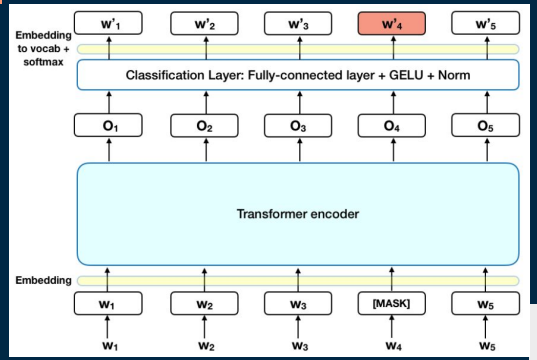
4. Does not consider context
5. Does not consider synonyms
6. Has trouble with stopwords, It, on, and, etc.
7.

BERT Model

The better recommender system

02

What is BERT?



Bidirectional Encoder Representations from Transformers

BERT is a language model produced by Google in 2018 which takes into account words context and position in order to encode it

Not This

This is not a state of the art language model

Why did we select BERT

BERT can differentiate context! Consider the lines:

- A panda eats shoots and leaves
- A cowboy eats, shoots, and leaves

Although these two lines would be almost the same in bag of words BERT can use context to differentiate between different meanings each word might have.

Other Advantages of BERT

1. BERT comes ready trained on a huge amount of data ~800 Million lines of text
2. BERT is open source and free
3. BERT is the current gold standard in the field of NLP

Deployment/challenges

Tools

Challenges

1. **Trained model on Google Colab CPU**
2. **Had to compute similarity on the fly**
3. **Document embeddings take 8GB and we could not host easily**
4. **Hosing a recommender system, e is very difficult**

Demo!

Disclaimer: we did not censor lyrics at all. Viewer discretion advised

Analysis - Christmas Songs

Jingle Bells

- Too many versions to work well
- All jingle bell versions were correctly identified as similar!
- Could be used to find all versions

Here Comes Santa Claus

- Copyrighted
- Lots of interesting christmas songs identified as related

Feliz Navidad

- Finds a mix between spanish and english christmas songs
- Strong relationship with “we wish you a merry christmas”

Analysis – Do lyrics encode other information?

Justin Bieber, Baby:

- Backstreet Boys, I'll Never Find Someone Like you
- Glee, Baby
- Rick Astley, Everytime

Taylor Swift, Never Getting Back Together:

- George Strait, I Don't Want to Talk It Over Anymore
- Don Williams, I Want You Back Again
- Don Henley, The Last Worthless Evening

Analysis – Do people care about lyrics?

Methodology:

- Compare BERT, Bag of Words, Baseline Model - RMSE
- Use models to predict % of listens per user - “listen agnostic”

$$\frac{\text{Listens for Song}}{\text{Total User Listens}}$$

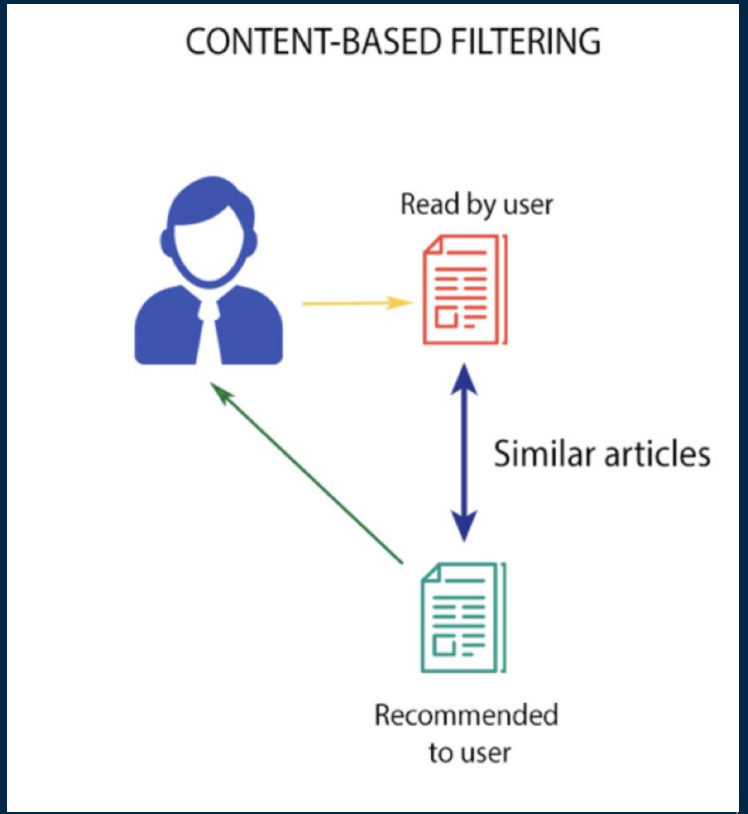
Piecing the data together

- Connecting song titles, artists, lyrics, and "listens" in different tables
- Required many table joins on song titles & artist names

	track_id	song_id	artist	song	user_id
0	TRMHWSF128F934D22D	0	Eliza Doolittle	Go Home	0
1	TRMHWSF128F934D22D	0	Eliza Doolittle	Go Home	1
2	TRMHWSF128F934D22D	0	Eliza Doolittle	Go Home	2
3	TRMHWSF128F934D22D	0	Eliza Doolittle	Go Home	3
4	TRMHWSF128F934D22D	0	Eliza Doolittle	Go Home	4
...	...	...	...	...	...
4824	TRYIHNA128F934D221	33	Eliza Doolittle	Moneybox	542
4825	TRYIHNA128F934D221	33	Eliza Doolittle	Moneybox	544
4826	TRYIHNA128F934D221	33	Eliza Doolittle	Moneybox	770
4827	TRYIHNA128F934D221	33	Eliza Doolittle	Moneybox	2760
4828	TRYIHNA128F934D221	33	Eliza Doolittle	Moneybox	2854

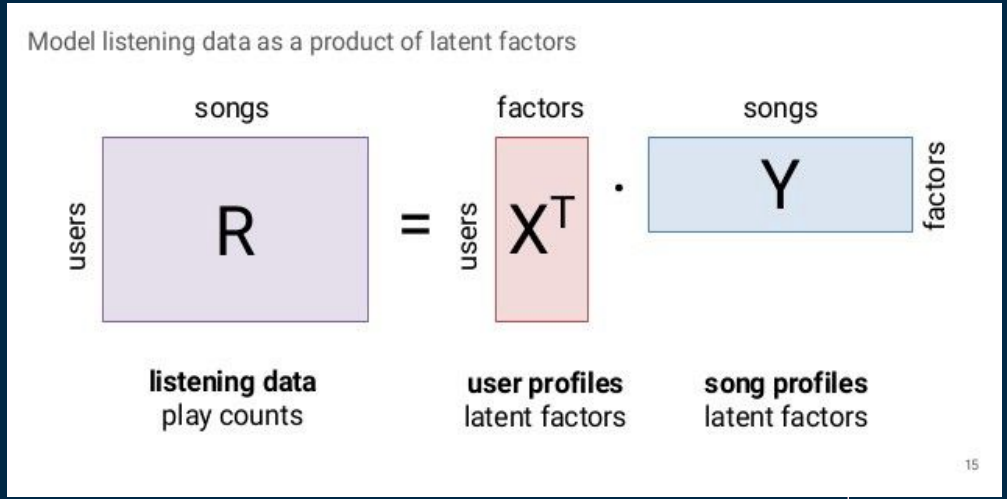
Content Based Filtering

- Generate latent features for songs using BERT, BOW
- Generate User Profiles based on weighted average of latent song reps



Making Predictions

- Tested models have latent features - BERT, BOW, MF
- Matrix Product yields predictions for entry (u,s)



Analysis - Do lyrics matter?

No. No one cares.

Bag of Words rmse: 5.293
BERT rmse: 5.294
Baseline rmse: 5.117

But I hoped...
baseline rmse: 0.7135563583690526
user-user rmse: 0.18755203340460716
bag of words rmse: 0.123785397123722 52
BERT rmse: 0.10575903232384942

Questions?