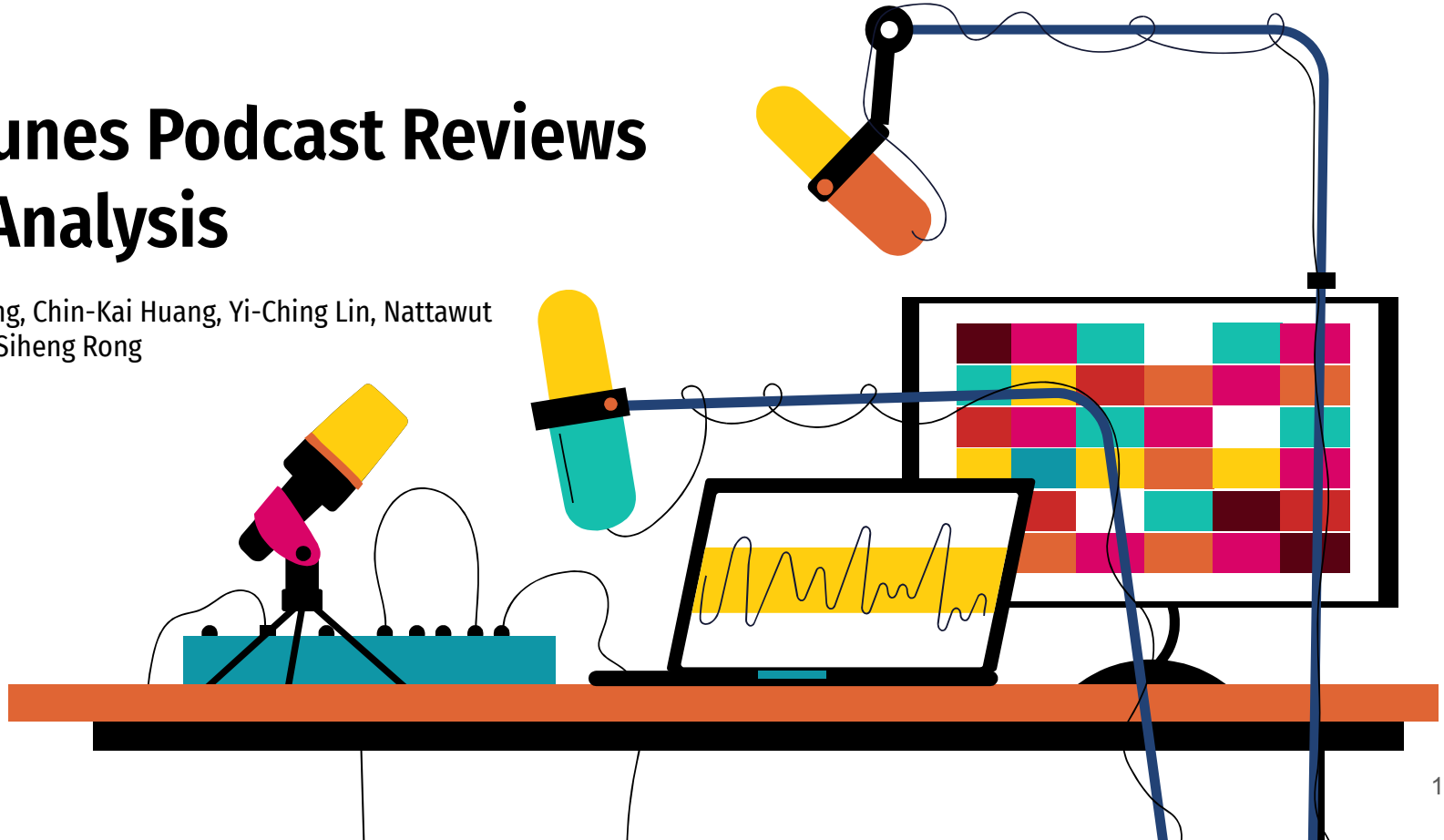
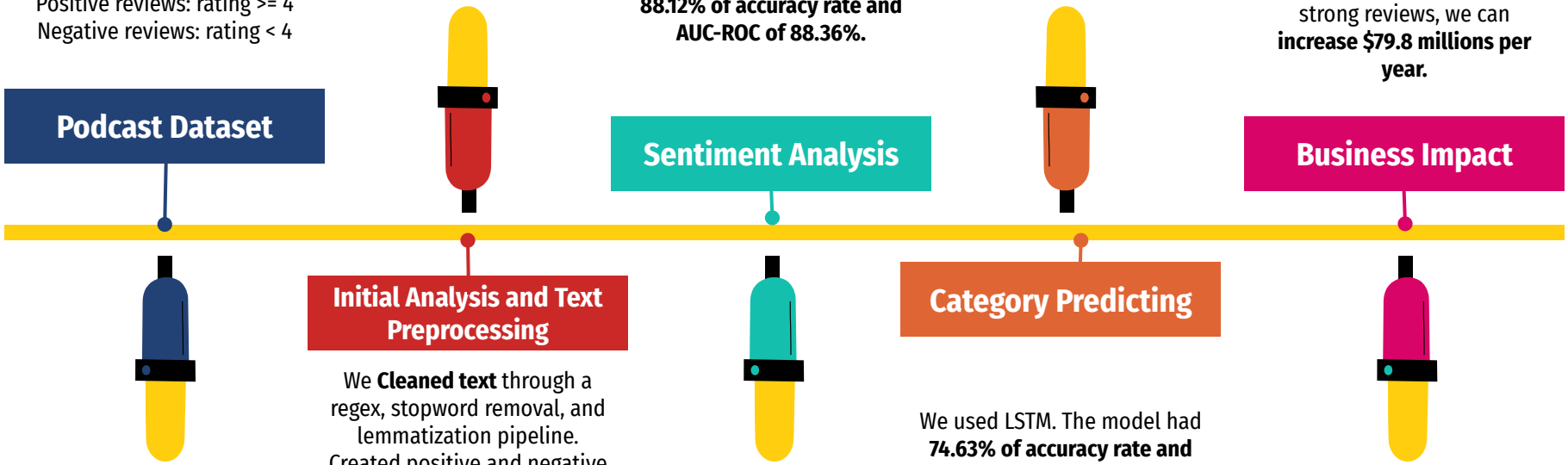
US iTunes Podcast Reviews Text Analysis

En-Ning Chiang, Chin-Kai Huang, Yi-Ching Lin, Nattawut Kananusorn, Siheng Rong



Executive Summary



Business Scope and Solutions

Business Scope

As of 2022, podcast listeners as a group have grown 29.5% in the last three years. There are currently 120 million podcast listeners in the U.S., and industry experts predict there will be over 160 million podcast listeners by 2023.

With the increase in demand, the market in this industry become more competitive. Therefore, our goal is to provide a better review system to help US iTunes keep its leadership in podcast streaming industry.

Solution 1

Sentiment Analysis Model

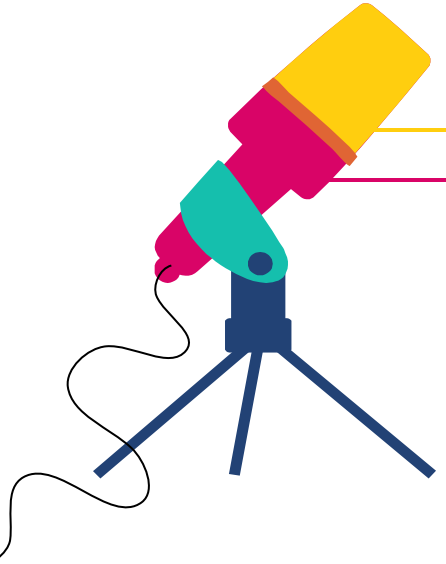
Aiming to determine positive/negative sentiment from reviews. Model scores can be integrated into podcasts' rating system, delivering a more robust representation of viewer sentiment and recommend podcast to listeners.

Solution 2

Category Predicting Model

Aiming to predict the podcast category of each reviews. By being able to accurately predict the category, we can determine the "quality" of a review, seeing if a review is more general or specific to a podcast.

Dataset Information and Data Schema



Dataset Size

The original dataset is 2 millions US iTunes podcast reviews, updated monthly. Due to the limitation of memory capacity, we randomly sampled 50k reviews from the original dataset.

Dataset Balance

50K reviews include 15,073 unique podcasts, 28,249 positive reviews (rating >= 4), and 21,751 negative reviews (rating < 4).

Category Schema	**Podcast ID**	**Category**				
Podcast Schema	**Podcast ID**	**iTunes ID**	**Slug**	**iTunes url**	**Title**	
Reviews Schema	**Podcast ID**	**Title**	**Content**	**Rating**	**Author ID**	**Created Date**

Initial Analysis

Text Preprocessing

Rating Label

Define sentiment based on ratings: rating >= 4 is **Positive**, rating < 4 is **Negative**

Category Combination

We put similar categories into one new category: We categorize

society/ religion/ government/ history/ education/ kids as **“society”**;
tv/ leisure/ sports/ music/ fiction/ arts as **“entertainment”**;
science/ technology/ health/ crime as **“others”**.

Final categories: Society, Entertainment, Comedy, News, Business, News, Others

Text Preprocessing

Stopword Removal

We use different ways to remove the most common words for different models, including nltk, Spacy

Regex Cleaning

We perform different regex cleaning for different models to remove high frequent and irrelevant words

Lemmatization

To group together inflected forms of words, we use lemmatization for more meaningful word context and form for all models

Combine Text Info

Combined the information from 2 text columns as model input, including review titles, and reviews

Word Frequency

We use nltk word count, count vectorization and TF-IDF to find high frequency words to draw quick insights of regex target

Word Embeddings

We use different word embedding approaches for different models, including Glove and Word2Vec

Word Cloud

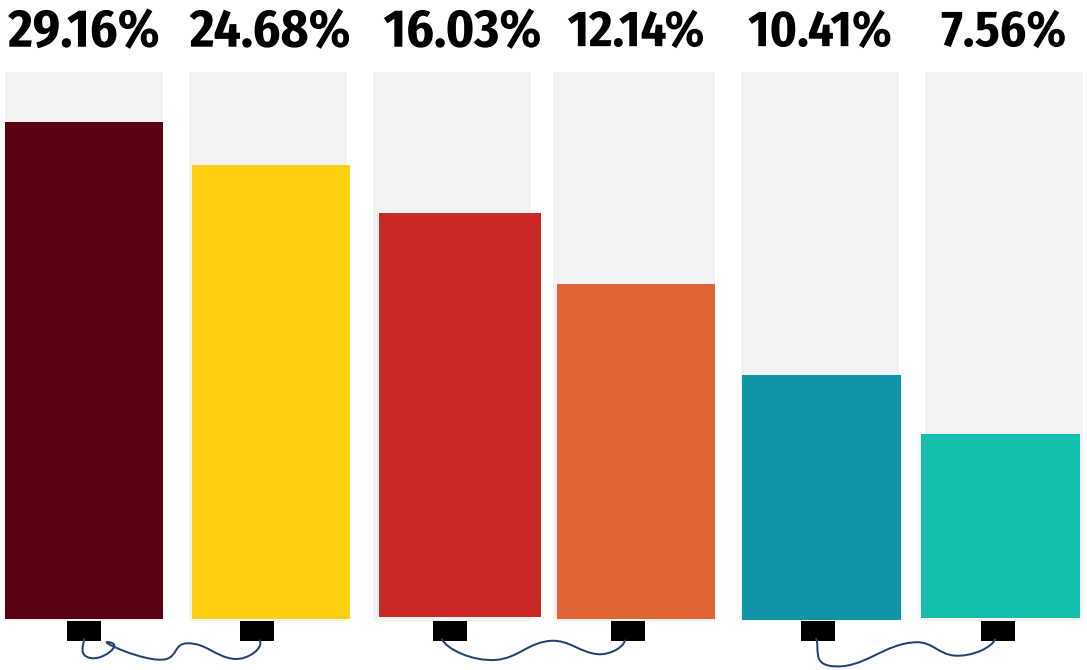
Word Phrase Cloud For Positive

Word Phrase Cloud for Negative

Exploratory Category Analysis

We calculate how many percentage of each category in the dataset

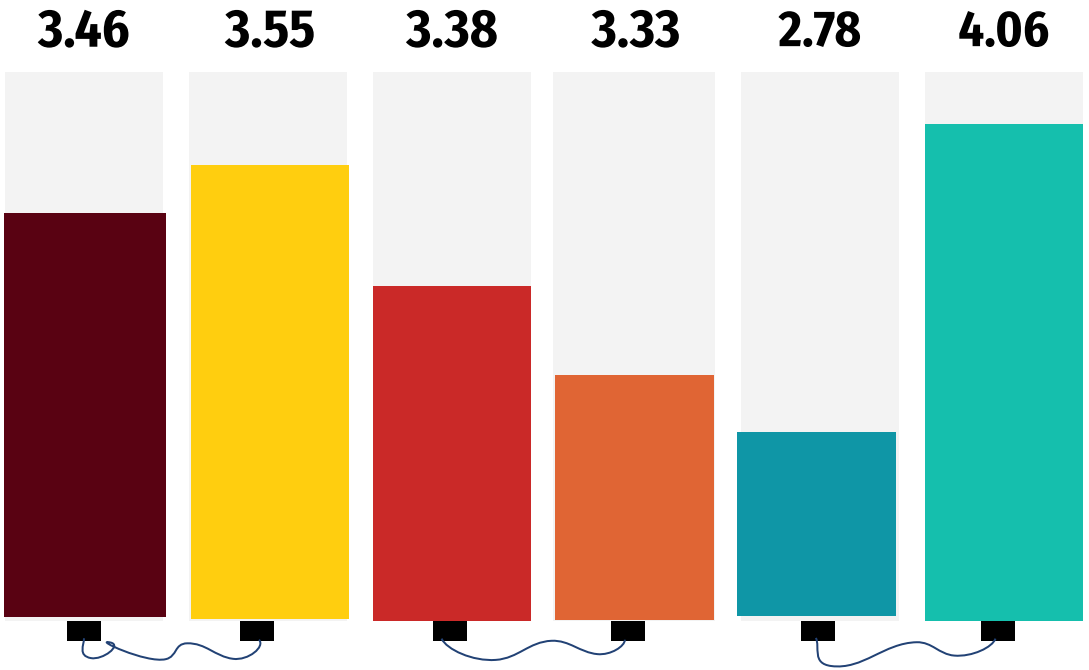
- Society
- Entertainment
- Comedy
- Others
- News
- Business



Exploratory Category - Rating Analysis

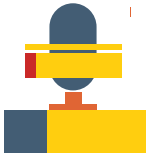
We calculated the average rating of each category.

- Society
- Entertainment
- Comedy
- Others
- News
- Business



Sentiment Analysis

Sentiment Analysis Data Preprocessing



Remove strange sign (e.g. ‘\n\n’, N) and unuseful words from trails & errors word count results

Verb: listen, get, show, make, think, use, know, want, hear, come, listen, would, go, say

Noun: podcast, episode, people, year, minute, one, time, story, people, host, guy

Others: really, lot)

Check the balance of a positive and negative review between training and testing data (test_size=0.2, random_state=20)

Sentiment Analysis Model Architecture

Best Deep Learning Model: LSTM

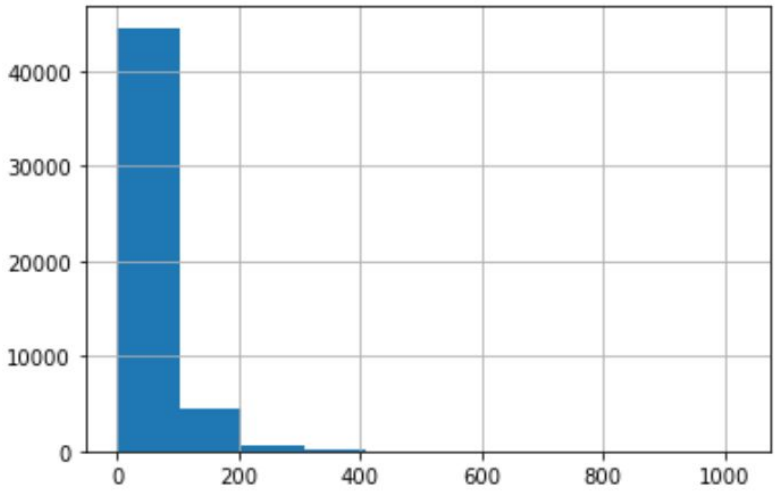
LSTM

- Trained our own Word2Vec embedding (300 dimensions) with a LSTM layer.
- LSTM networks were designed specifically to overcome the long-term dependency problem faced by recurrent neural networks RNNs.
- The model predicts the probability of one review to be positive.

Model Hyperparameters

- **Max Length** = 400
- **Layers:** Embedding (Using Trained Word2Vec Model), Masking, LSTM (32 units), Dense (16 Units, Sigmoid), Dense (2 Unit, SoftMax)

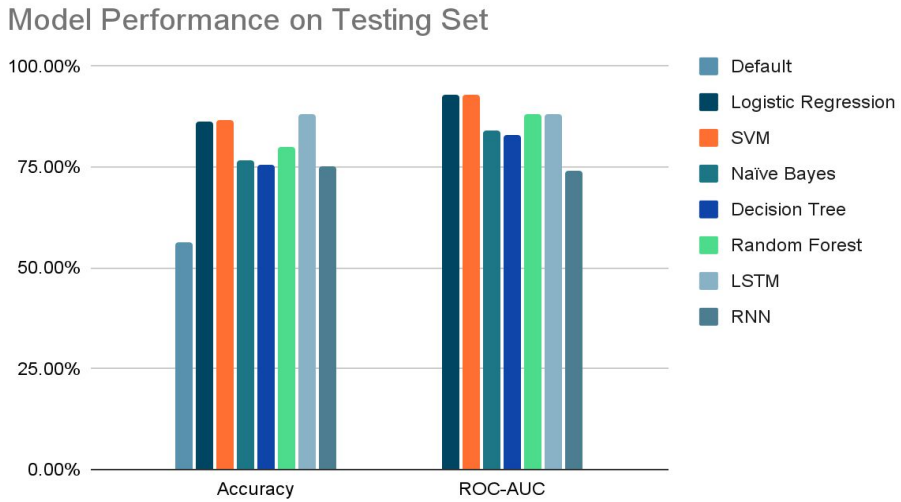
Histogram of Review Length after Preprocessing



Most documents length was less than 200, so we could build the model with max length lower than the actual one (1023)

Sentiment Analysis Model Performance

- Our ML models perform significantly better than the default rate, with a lift of **31.58%**
- LSTM have the best performance, with **a 88.08% accuracy and 87.86% of ROC-AUC.**
- The model has **a good balance between recall and precision** which indicate that there is no bias between a positive and negative prediction.
- Our final recommendation would be to use LSTM to predict a review sentiment to **recalculate the rating and customize podcast recommendations.** The automated analytics metric could be generated and then provide new insights into audience feedback to podcasters.



Confusion Matrix	Predicted Positive	Predicted Negative
Actual Positive	5066	595
Actual Negative	597	3742

Sentiment Analysis Business Use Cases

With the sentiment score as part of the rating system, like weighted average, the rating may be more reliable for listeners.

Improved Rating System

Comprehensive Analytics

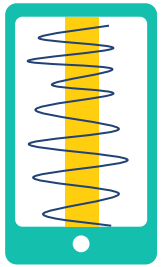
iTunes has already created analytical tools for podcasters, which is iTunes Podcasts Connect. The sentiment score can tell podcasters more about the feelings of their audience.

Based on the sentiment score, iTunes Podcast can recommend relevant podcasts or shows from same podcasters based on listeners' real sentiment.

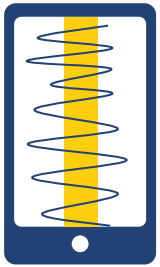
Recommendation System

Category Predicting Using Reviews

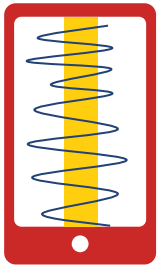
Category Predicting Data Preprocessing



Add information from podcast title columns to model input



Used Regex to remove high-frequency words based on Count Vectorization result



Used pre-trained word embedding from Spacy to vectorize text corpus, obtained 500 features

Category Predicting Model Architecture

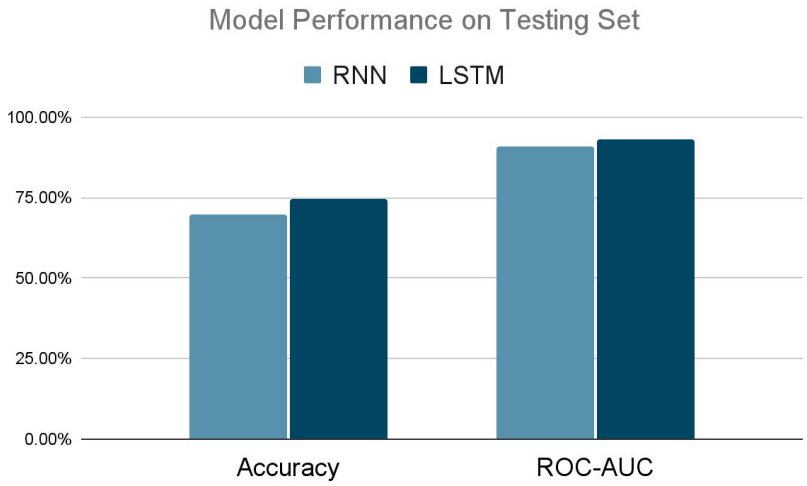
RNN Hyperparameters

- Vocab Size: 56,954
- Input Length: 500
- Layers: Embedding Masking (Using Pre-Trained GLOVE Model), SimpleRNN (64 units), Dense (16 Units, Sigmoid), Dense (1 Unit, SoftMax)

LSTM Hyperparameters

- Vocab Size: 56,954
- Input Length: 500
- Layers: Embedding Masking (Using Pre-Trained GLOVE Model) , LSTM (32 units), Dense (16 Units, Sigmoid), Dense (6 Units, SoftMax)

Category Predicting Model Performance



Form LSTM Model

Category	Recall	Precision
Society	0.79	0.73
Entertainment	0.73	0.75
Comedy	0.75	0.71
News	0.79	0.87
Business	0.65	0.74
Others	0.70	0.74

- LSTM model performs better than the RNN model with 74.63% accuracy on testing set and 93.19% of ROC-AUC score.
- The model has good balances between recall and precision for all six categories.

Category Predicting Business Use Cases

Review Quality Rating

Our model can predict the category a podcast might be associated with and therefore, we can use these predictions **to determine the quality of a podcast rating.**

For example, if a single review simply said, “This podcast is the worst ever”, our model is likely not able to determine which category that review belongs to. However, if the review comes with “He completely embarrassed himself in a BBC interview with Andrew Neil and it really exposed the weakness of his show.”, our model is likely to determine the review belongs to “News” and the podcasters should improve his interview skills.

As a result, we can quickly find higher quality reviews and promote them as the tops in the review section.

Audience

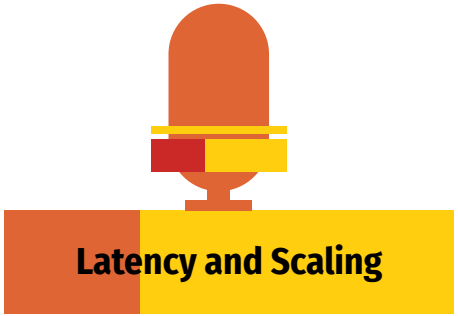
Audience just needs to look at the top reviews to get specific understanding about the podcast, which can improve user experience in the iTunes website or APP and thus attract more customers.

Podcaster

Podcasters can get higher quality feedbacks for their future improvement by focusing on the higher quality reviews. This can attract more podcasters for iTunes.

Implementation Roadmap

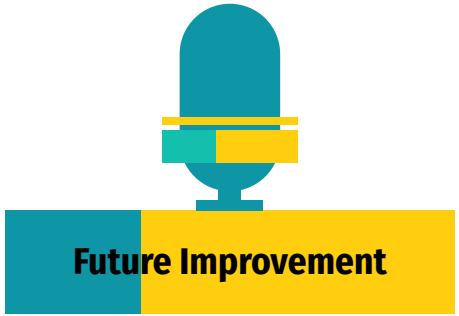
Implementation Roadmap



Latency and Scaling

For production, all text preprocessing models should be pre-trained, so they integrate seamlessly into a feature engineering pipeline.

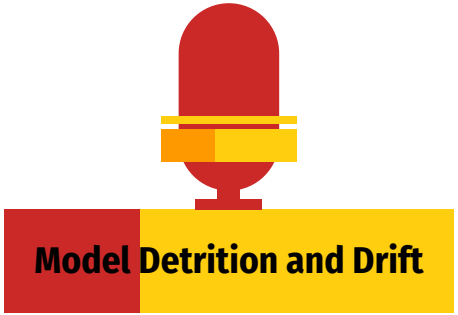
We successfully classified 50,000 reviews in 107 seconds for category prediction and 126 seconds* for sentiment analysis. This can be scaled for real-time scoring using **server-side cursors**.



Future Improvement

One improvement we can do in the future is to add one column for podcast description (Slug is not enough). We can use cosine similarity on podcast description to better categorize podcast.

Our sentiment analysis can generate reports that can be used as a basis for future recommendations for audience response to podcasts.



Model Detrition and Drift

The number of podcast reviews for each category varies largely. We recommend collecting more data on these categories and rebuilding the model with these new observations.

* running time varies depending on the device capacity.

Business Impact

ROI: Sentiment Analysis Model

Approaches

1. Upgrade current rating system with weighted sentiment scoring and improve podcast recommendation system
2. Aggregate sentiment monitoring tools to iTunes Podcast Connect platform for podcaster analytics

Earning & Savings

iTunes Podcast has 28.5 million listeners monthly while Spotify is estimated to reach 32.5 million podcast listeners per month. The upgraded rating system can help listeners find better podcasts and may attract 300k new users, creating revenue of $4.5 million (CPM $15).
More comprehensive and robust analytics information can attract new 10k podcasters to join Apple Podcasters Program with revenue of about $200k.

Expenditures

Engineering team $160K per month. Technical Costs $40K.

ROI per year

$(4.5+0.2) * 12 - $0.2 * 12 =

$ 54 million

ROI: Category Predicting Model

Approaches

1. Determine quality of podcast reviews to create a new dimension that feeds into the rating system
2. Provide high-quality reviews on top of the review section, so audience will have more information about the rating of podcasts.

Earning & Savings

Increased traffic from promoting strong reviews will increase $2.25 million monthly (150k increase in impression from dedicated users base) (CPM $15).

More robust feedback can attract new 5k podcasters to join Apple Podcasters Program with revenue of about $100k.

Expenditures

Engineering team $160K per month. Technical Costs $40K.

ROI per year

$(2.25+0.1) * 12 - $0.2 * 12 =

$ 25.8 million

Final Return on Investment

$ 54 million

Sentiment Analysis Model

Add weighted scoring features to recommendation system

$ 25.8 million

Category Predicting Model

Promote strong reviews by determining high-quality reviews

Return on Investment Around

$ 79.8 million

Conclusion

New Business Opportunity

Advanced NLP techniques can create business opportunities by sentiment analysis and category predicting. These models can also improve podcast rating system and recommendations.

Model Summary

Sentiment analysis: Our best model is LSTM which has 88.% of accuracy rate and has 88.36% of AUROC. Category predicting: Our best model is LSTM which has 74.63% of accuracy rate and has 93.19% of AUROC.

Future Improvement

To increase accuracy rate, we can use different vectorization approaches and different machine learning algorithms. We can also include more information about podcast such as description and podcaster to perform more comprehensive analysis.

Thank you!

Appendix

Sentiment Analysis Model Hyperparameters Tests

Architecture	C	solver	penalty	max_iter	Accuracy / ROC-AUC
Logistic Regression with Count Vectorizer (unigram)	0.1	liblinear	l2	60	0.85 / 0.92
Logistic Regression with Count Vectorizer (bigram)	0.1	liblinear	l2	60	0.76 / 0.87
Logistic Regression with Count Vectorizer (trigram)	0.1	liblinear	l2	60	0.60 / 0.72
Logistic Regression with TF-IDF Vectorizer (unigram)	1.0	sag	l2	90	0.86 / 0.93
Logistic Regression with TFIDF Vectorizer (bigram)	1.0	sag	l2	90	0.76 / 0.88
Logistic Regression with TFIDF Vectorizer (trigram)	1.0	sag	l2	90	0.58 / 0.72
Logistic Regression with Word2Vec (spacy)	1000.0	newton-cg	l2	90	0.83 / 0.90

Sentiment Analysis Model Hyperparameters Tests

Architecture	splitter	min_samples_split	min_samples_left	max_depth	criterion	Accuracy / ROC-AUC
Decision Tree with Count Vectorizer (unigram)	random	38	27	50	entropy	0.76 / 0.83
Decision Tree with Count Vectorizer (bigram)	random	38	27	50	entropy	0.61 / 0.67
Decision Tree with Count Vectorizer (trigram)	random	38	27	50	entropy	0.58 / 0.57
Decision Tree with TFIDF Vectorizer (unigram)	best	10	35	38	entropy	0.75 / 0.82
Decision Tree with TFIDF Vectorizer (bigram)	best	10	35	38	entropy	0.61 /0.65
Decision Tree with TFIDF Vectorizer (trigram)	best	10	35	38	entropy	0.58 / 0.56
Decision Tree with Word2Vec (spacy)	random	4	43	36	gini	0.74 / 0.81

Sentiment Analysis Model Hyperparameters Tests

Architecture	n_estimators	min_samples_split	min_samples_left	max_depth	criterion	Accuracy / ROC-AUC
Random Forest with Count Vectorizer (unigram)	150	5	1	10	gini	0.60 / 0.90
Random Forest with Count Vectorizer (bigram)	150	5	1	10	gini	0.57 / 0.77
Random Forest with Count Vectorizer (trigram)	150	5	1	10	gini	0.57 / 0.58
Random Forest with TFIDF Vectorizer (unigram)	100	5	1	10	log_loss	0.61 / 0.88
Random Forest with TFIDF Vectorizer (bigram)	100	5	1	10	log_loss	0.57 / 0.73
Random Forest with TFIDF Vectorizer (trigram)	100	5	1	10	log_loss	057 / 0.57
Random Forest with Word2Vec (spacy)	150	5	4	10	gini	0.80 / 0.88

Sentiment Analysis Model Hyperparameters Tests

Architecture	Parameters	Accuracy / ROC-AUC
Naive Bayes with Count vectorizer (unigram)	Default	0.64 / 0.69
Naive Bayes with TFIDF Vectorizer (unigram)	Default	0.64 / 0.69
Naive Bayes with Word2Vec (spacy)	Default	0.77 / 0.84
Support Vector Machine with Count vectorizer (unigram)	Default	0.64 / 0.67
Support Vector Machine with TFIDF Vectorizer (unigram)	Default	0.87 / 0.93
Support Vector Machine with Word2Vec (spacy)	Default	0.83 / 0.91

** Naive Bayes and Support Vector Machine do not fine tune the optimal parameters since the limitation of computation power.

Sentiment Analysis Model Hyperparameters Tests

Architecture	Max Doc Length	Epochs	Vocab Size	Word Embedding	Accuracy / ROC-AUC
RNN 5 layers (Embedding, Masking, SimpleRNN(64), Dense(16), Dense(2))	400	3	53567	Pretrained Glove (100,)	0.75 / 0.74
LSTM 5 layers (Embedding, Masking, LSTM(32), Dense(16), Dense(2))	400	5	53567	Pretrained Glove (100,)	0.87 / 0.87
LSTM 5 layers (Embedding, Masking, LSTM(32), Dense(16), Dense(2))	400	2	48699	Trained Word2Vec (300,)	0.88 / 0.88