Podcast Recommendation Web App

A project report submitted in partial fulfillment of the requirements for the award of the degree of

B.Tech. in Computer Science

by

Aranya Patra (LCI2020034)

Anmol Sankadiya (LCI2020048)

Nikhil Verma (LCI2020044)

under the guidance of
Dr. Naveen Saini

विद्या ददाति विनयं विनयाद् याति पात्रताम्

Indian Institute of Information Technology, Lucknow
May 2023

Declaration of Authorship

We, **Aranya Patra, Anmol Sankadiya and Nikhil Verma**, declare that the work presented in "**Podcast Recommendation Web App**" is our own. We confirm that:

- This work was completed entirely while in candidature for B.Tech. degree at Indian Institute of Information Technology, Lucknow.

- Where we have consulted the published work of others, it is always cited.

- Wherever we have cited the work of others, the source is always indicated. Except for the aforementioned quotations, this work is solely our work.

- I have acknowledged all major sources of information.

Signed:

	(Aranya Patra)	(Anmol Sankadiya)	(Nikhil Verma)
Date:	24 Apr, 2023	24 Apr, 2023	24 Apr, 2023

CERTIFICATE

This is to certify that the work entitled “**Podcast Recommendation Web App**" submitted by **Aranya Patra, Anmol Sankadiya and Nikhil Verma** who got their name registered on **30 Jul 2020** for the award of B.Tech. degree at Indian Institute of Information Technology, Lucknow is absolutely based upon their own work under the supervision of **Dr. Naveen Saini**, Department of Computer Science, Indian Institute of Information Technology, Lucknow - 226 002, U.P., India and that neither this work nor any part of it has been submitted for any degree/diploma or any other academic award anywhere before.

Dr. Naveen Saini
Department of Computer Science
Indian Institute of Information Technology, Lucknow
Pin - 226 002, INDIA

Acknowledgements

We would like to express our deepest gratitude to everyone who has contributed to the completion of this project on **Podcast Recommendation Web App**.

First and foremost, we would like to thank our project advisor **Dr. Naveen Saini**, for their guidance, support, and valuable feedback throughout the project. Their expertise and insights have been invaluable in shaping the direction and scope of the project.

We are grateful to our peers and colleagues who provided feedback, suggestions, and insights on the project. Their constructive criticism and ideas have helped us to improve and refine our work.

We are honored and humbled to have had the opportunity to work on this project, and we hope that our work will be useful to podcast listeners worldwide. We believe that our work will help users discover new and exciting podcasts that they might not have otherwise found. We also hope that our project will inspire others to explore the field of machine learning.

Lucknow
May 2023

Aranya Patra
Anmol Sankadiya
Nikhil Verma

ABSTRACT

Finding a good podcast recommendation is a challenging task due to the vast amount of podcast content available online. There are thousands of podcasts in different genres, topics, and formats, making it difficult for users to find content that matches their interests. The current solutions for podcast recommendations mostly rely on user ratings, reviews, and subscriptions, which may not be accurate or up-to-date. Moreover, they often suffer from a cold start problem, where new users or podcasts with few ratings and reviews have limited or no recommendations.

To address these challenges, our project leverages machine learning algorithms to suggest podcasts based on various inputs such as existing podcasts in the dataset, genres, and keywords typed by the user. The dataset includes podcasts available on iTunes, providing a broad range of content across various genres and topics. By utilizing machine learning techniques, the application can offer personalized recommendations to users, improving the relevance of the suggested content.

The development of the project involves data collection, preprocessing, feature engineering, and model training for machine learning models, and training them using appropriate algorithms and hyperparameters. This process improves the accuracy and relevance of personalized podcast recommendations for users.

The end-result of this project is a web application that provides users with personalized and relevant podcast recommendations based on their inputs. The application aims to help users discover new podcasts that match their interests and preferences, thus improving their overall podcast listening experience. The project can also benefit businesses in the podcast industry by offering insights into user preferences and content consumption patterns, allowing them to optimize their content and marketing strategies.

Contents

Chapter 1

Introduction

1.1 Background

Podcasting has become an increasingly popular form of entertainment and education, with millions of podcasts available online. However, with the sheer volume of podcasts available, it can be difficult for listeners to find content that matches their interests and preferences. Current solutions for podcast recommendations often rely on user ratings and reviews, which may not always be accurate or up-to-date. To address this challenge, a podcast recommendation web app can be developed using machine learning algorithms. By analyzing the characteristics of existing podcasts and user preferences, the app can provide personalized recommendations to users.

1.2 Motivation

The motivation behind developing a podcast recommendation web app is to enhance the user experience by providing relevant and engaging content that matches their interests. Our project aims to improve the user experience by providing a personalized and engaging platform. By leveraging machine learning techniques, the app can analyze large amounts of data and provide accurate recommendations to users. This can potentially lead to increased user engagement, satisfaction, and loyalty, ultimately benefiting both listeners and podcast creators.

1.3 Problem Statement

Finding good podcast recommendations can be a challenging task for users due to the overwhelming number of options available. Traditional recommendation systems based on popularity or ratings can often lead to recommendations that do not match the user's interests. The lack of personalized and relevant recommendations can lead to a frustrating user experience and decreased engagement with podcast platforms. Therefore, there is a need for a podcast recommendation web app that utilizes machine learning to provide more accurate and tailored recommendations based on the user's preferences and specific podcast features.

1.4 Objectives

The objective of the podcast recommendation web app is to provide personalized and relevant recommendations to users based on their listening history and preferences. The app will use machine learning algorithms and natural language processing to analyze user data and generate recommendations that match their interests. The aim is to enhance the user experience by providing a convenient and efficient way to discover new podcasts and help them stay engaged with their favorite content. The app will also allow new users to search for highly rated podcasts by choosing their favourite genres.

1.5 Contributions

Our web application will contribute to solving the problem of overwhelming choice and low discoverability of podcasts by providing personalized recommendations to users based on their listening history and preferences. This will enhance the listening experience for users and encourage them to explore and discover new podcasts that align with their interests.

Moreover, our app will use machine learning algorithms for better recommendation accuracy and will continuously learn and adapt to user feedback. This will provide a seamless and engaging user experience.

Additionally, our web application will automate the recommendation process, reducing the need for manual curation and increasing the efficiency of the recommendation engine. This will allow us to provide faster and more reliable recommendations to users, keeping them engaged with the platform and improving overall satisfaction.

Chapter 2

Literature Review

The field of podcast recommendation systems has seen significant development in recent years, with researchers exploring various techniques to enhance listener engagement and satisfaction. *Reddy et al. (2021)* [3] proposed a model for predicting engagement in podcasts based on language usage in podcast transcripts. The model leverages topic modeling and sentiment analysis techniques to extract features related to language usage and outperformed several baselines in predicting engagement. This study highlights the importance of understanding listener engagement in improving podcast recommendation systems.

In a similar vein, *Yang et al. (2018)*[6] investigated user interactions with podcast recommendations delivered via smart speakers. The authors found that users preferred shorter recommendations that were specific to their interests, and that the personality of the voice assistant affected their engagement with the recommendations. This study suggests that voice-based podcast recommendations have the potential to improve user engagement and satisfaction and provides insights into optimizing the delivery of recommendations via smart speakers.

The paper by *Hariri et al. (2012)*[1] proposes a context-aware music recommendation system that considers not only the user's preferences but also the contextual information related to the user's situation. The authors used latent Dirichlet allocation (LDA) to model the topics and sequential patterns of music listening behavior and demonstrated that the context-aware recommendation approach significantly outperforms non-contextual approaches. This study emphasizes the importance of considering contextual information in recommendation systems and provides a framework for incorporating such information into the recommendation process.

Schedl et al. (2015)[4] work explores how sensor data and social media data can be used to make music recommendations that are personalized to a listener's mood, activity level, and social context. The paper proposes a system that uses data from sensors such as accelerometers and heart rate monitors, as well as social media data from sources like Twitter, to infer a listener's current context and make personalized music recommendations based on that context. This study highlights the potential for recommendation systems to incorporate diverse sources of data beyond just user listening history to create more personalized and engaging experiences for listeners.

Finally, in *Xing et al. (2016)*[5] the authors propose a content-based recommendation system for podcast audio-items that utilizes natural language processing techniques. The model extracts features such as topics, sentiment, and named entities from podcast transcripts to recommend relevant episodes to users. The study highlights the importance of incorporating non-audio features in podcast recommendation systems and shows that the proposed model outperforms several baselines in terms of recommendation accuracy. This study provides insights into the use of natural language processing techniques for podcast recommendation and emphasizes the potential of content-based approaches for personalized podcast recommendation.

In summary, the reviewed papers provide valuable insights into various techniques for improving podcast and music recommendation systems. The studies emphasize the importance of understanding listener engagement and considering contextual information in the recommendation process. The use of deep learning techniques and diverse sources of data also have the potential to enhance the personalization and accuracy of recommendation systems. These insights can inform the development of more effective and engaging podcast recommendation systems in the future.

Chapter 3

Methodology

,

3.1 Data Collection

We needed to collect a dataset of podcast episodes that includes information such as episode title, description, audio features, and metadata such as genre, tags, and duration. This dataset can be obtained from various sources such as podcast hosting platforms or web scraping.

To develop the model, we gathered titles and descriptions for a large number of podcasts. We obtained a list of thousands of podcast titles from a dataset available on Github, filtered to include only English language podcasts and removed duplicates. We scraped the iTunes website using Python's Beautiful Soup library to obtain the titles and descriptions for each podcast. Despite the time-intensive nature of API queries and web scraping, this approach allowed us to gather the necessary data for our recommendation algorithm.

3.2 Data Pre-processing

The raw data collected needs to be pre-processed before being used to train the recommendation system. This includes text cleaning, feature extraction, and data normalization.

We then pre-processed and cleaned the text data by removing mixed alphanumeric words like season/episode number, sponsorship statements, date, and links. Stopwords were also removed (such as "the", "a", "an",

”in”) and PorterStemmer in the nltk library was used for stemming. combine them

3.3 Similarity Metrics

Content-based filtering relies on analyzing the textual content of items to determine their similarity to a user’s preferences. One popular technique to measure this similarity is cosine similarity. Cosine similarity measures the angle between two vectors projected onto an n-dimensional vector space, where each dimension corresponds to a different term in the document. The cosine of the angle between two vectors is a measure of their similarity, with a cosine value of 1 indicating that the two vectors are identical and a cosine value of 0 indicating that the two vectors are completely dissimilar.

To use cosine similarity for content-based filtering, we first represent each item and user profile as a vector of features (e.g., words in the item description). Then, we calculate the cosine similarity between each item and the user profile. Items with higher cosine similarity values are considered more similar to the user’s preferences and are recommended accordingly.

Other similarity metrics that can be used for content-based filtering include Jaccard similarity and TF-IDF. Jaccard similarity measures the similarity between two sets by calculating the size of their intersection divided by the size of their union. TF-IDF (term frequency-inverse document frequency) measures the importance of each term in a document by weighting its frequency by the inverse of its frequency in the entire corpus. This can be used to compare the similarity between documents based on their content.

3.4 Building Models

Recommendation algorithms are an essential part of our podcast recommendation web app, as they help to personalize recommendations for each user. We can choose from several different types of recommendation algorithms, including content-based filtering, collaborative filtering, and hybrid approaches.

To create feature vectors for each podcast, we have used different vectorizers such as Tfidf, Count Vectorizer, Word2Vec, and GloVe, all of which are commonly used in natural language processing. Tfidf weighs the importance of words in a document based on their frequency, Count Vectorizer creates a bag-of-words representation, and Word2Vec and GloVe create word embeddings that capture the meaning of words in a low-dimensional vector.

In addition to vectorizers, we have also applied BERT encoding to the podcast data to create contextualized embeddings that capture the meaning of the text in its context.

To group similar podcasts together, we have applied clustering algorithms to the feature vectors.

1. K Means is a clustering algorithm that groups data points into K clusters, where K is a user-specified parameter.

2. Agglomerative Clustering is a bottom-up hierarchical clustering algorithm that starts with each data point as its own cluster and then merges clusters together until there is only one cluster remaining.

3. Affinity Propagation is a clustering algorithm that does not require a user-specified parameter for the number of clusters. Instead, it uses a message-passing algorithm to determine the number of clusters and the data points that belong to each cluster.

4. Hierarchical Clustering is a top-down hierarchical clustering algorithm that starts with all data points in a single cluster and then recursively splits the clusters into smaller clusters until each data point is in its own cluster.

By applying clustering algorithms to the feature vectors of the podcasts, you can group similar podcasts together and make recommendations based on the user's listening history and preferences.

3.5 Testing and Evaluation

We have used various evaluation metrics such as similarity between podcasts, feedback ratings, and diversity metrics that can help us test and evaluate the performance of our podcast recommendation web app. These metrics will enable us to determine the accuracy and effectiveness of our recommendation system.

After defining the evaluation metrics, we trained the recommendation system using pre-processed data and the chosen recommendation algorithm. The system undergo testing using the evaluation metrics to evaluate its performance. Based on the evaluation results, we made adjustments to the system parameters as required to enhance its performance.

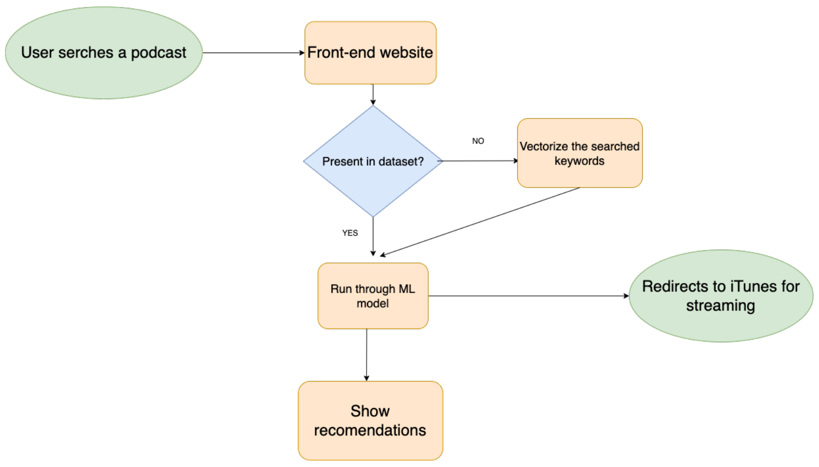
During the testing phase, we evaluated the similarity between recommended podcasts to ensure that the recommendations are relevant and useful to the users. We also considered feedback ratings to measure user satisfaction with the recommendations provided. Additionally, we will assess diversity metrics to ensure that the recommended podcasts cover various topics and genres.

By training and testing our podcast recommendation web app and adjusting system parameters based on the evaluation results, we enhanced its performance and provide users with accurate and relevant podcast recommendations.

3.6 Deployment

Once the recommendation system is developed and tested, the next step is to deploy it on a web application so that users can access it. For our project, we used the Streamlit library to build a web application that allows users to input their preferences and receive personalized podcast recommendations.

Here is the workflow diagram of the working of our web app-



Once the web application is deployed, it's important to continuously monitor its performance and make improvements based on user feedback. This can include analyzing user behavior to determine if the recommendations are meeting their needs, identifying areas of the system that may need improvement, and incorporating new data and feedback to improve the system's accuracy and relevance.

Chapter 4

Implementation and Results

We implemented a podcast recommendation system using a content-based filtering approach. The system was developed using Python and its various libraries such as pandas, numpy, sklearn, and NLTK. The following are the main steps involved in the implementation of the system:

Data Collection: We collected podcast data from various sources such as Apple Podcasts.

Data Pre-processing: We pre-processed the data by removing stop words, stemming, and applying various vectorization techniques such as TF-IDF, Count Vectorizer, Word2Vec, GloVe, and BERT encoding.

Recommendation Algorithm: We used a content-based filtering approach to generate personalized recommendations for the user. We applied different clustering algorithms such as K-Means, Agglomerative Clustering, Affinity Propagation, and Hierarchical Clustering to cluster similar podcasts based on the user's preferences.

Web Application: We developed a web application using Streamlit to provide an interactive interface for the user to input their preferences and receive personalized podcast recommendations.

Evaluation Metrics: To evaluate the performance of our podcast recommendation system, we used the following evaluation metrics:

Cosine Similarity: We used cosine similarity to measure the similarity between the recommended podcasts and the test podcast. We calculated the cosine similarity between the test podcast and the recommended podcasts, and the higher the value, the more similar the recommended podcast is to the test podcast.

Feedback Rating: We asked our peers to rate the recommended podcasts on a scale of 1 to 5 based on their relevance to the test podcast. The higher the rating, the more relevant the podcast is to the test podcast.

Diversity: We evaluated the diversity of the recommended podcasts based on whether the recommendations were expected or unexpected provided in the paper kaminskas et al. (2016)[2]. Expected recommendations are those that are in the same genre as the test podcast, while unexpected recommendations are those that are in different genres.

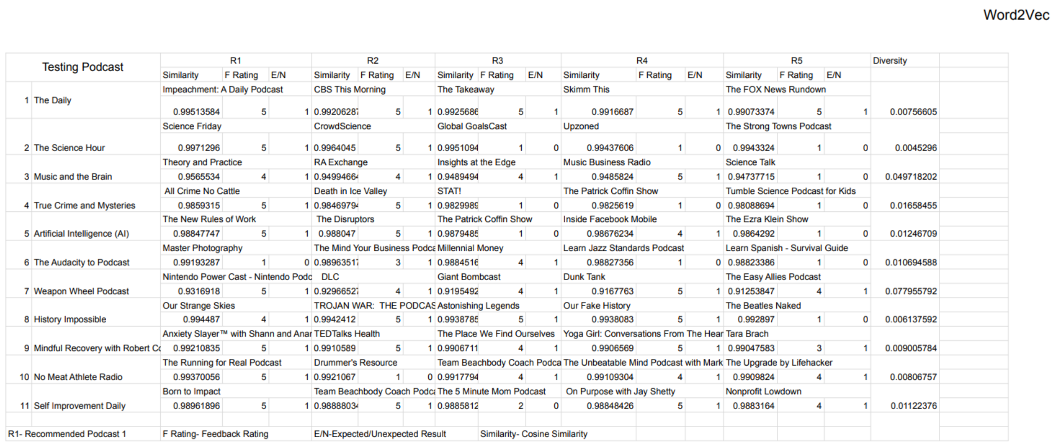
We evaluated the performance of two best performing algorithms: word2vec + cosine similarity and Kmeans + tfidf + Cosine Similarity. Both algorithms performed well in terms of cosine similarity, feedback rating, and diversity.

Cosine Similarity: Both algorithms achieved high cosine similarity scores, with word2vec achieving a higher score than Kmeans.

Feedback Rating: Both algorithms received high feedback ratings, with Kmeans receiving a slightly higher rating than word2vec.

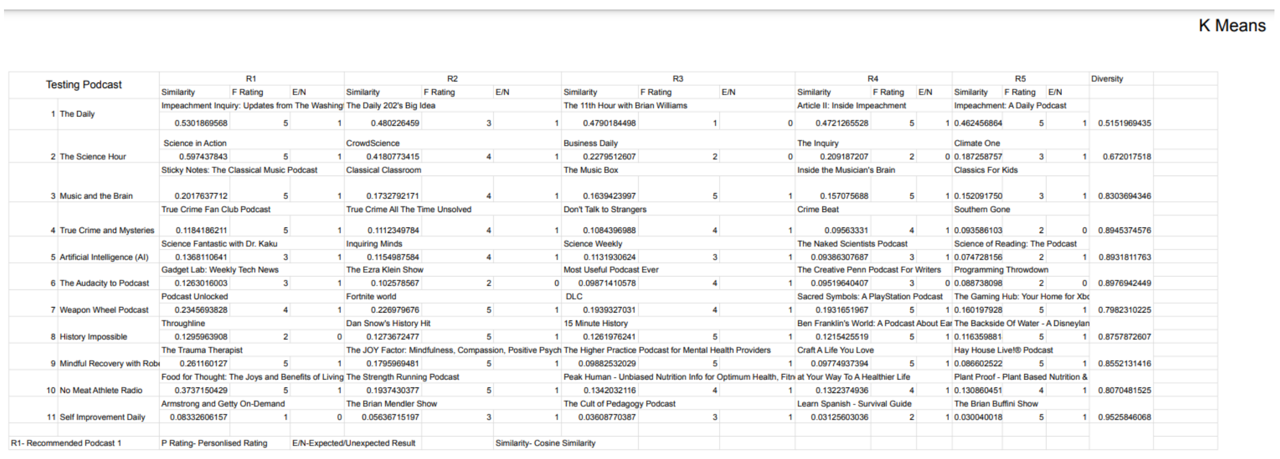
Diversity: Both algorithms provided a mix of expected and unexpected recommendations, indicating a good level of diversity in the recommendations.

Here is the table for the results.



Word2Vec

Testing Podcast		R1			R2			R3			R4			R5			Diversity
		Similarity	F Rating	E/N	Similarity	F Rating	E/N	Similarity	F Rating	E/N	Similarity	F Rating	E/N	Similarity	F Rating	E/N	
		Impeachment: A Daily Podcast			CBS This Morning			The Takeaway			Skimm This			The FOX News Rundown			
1	The Daily	0.99513584	5	1	0.99206287	5	1	0.99256866	5	1	0.9916687	5	1	0.99073374	5	1	0.00756605
		Science Friday			CrowdScience			Global GoalsCast			Upzoned			The Strong Towns Podcast			
2	The Science Hour	0.9971296	5	1	0.9964045	5	1	0.9951094	1	0	0.99437606	1	0	0.9943324	1	0	0.0045296
		Theory and Practice			RA Exchange			Insights at the Edge			Music Business Radio			Science Talk			
3	Music and the Brain	0.9565534	4	1	0.9499466	4	1	0.9489494	4	1	0.9485824	5	1	0.94737715	1	0	0.049718202
		All Crime No Cattle			Death in Ice Valley			STAT!			The Patrick Coffin Show			Tumble Science Podcast for Kids			
4	True Crime and Mysteries	0.9859315	5	1	0.9846979	5	1	0.9829989	1	0	0.9825619	1	0	0.98088694	1	0	0.01658455
		The New Rules of Work			The Disruptors			The Patrick Coffin Show			Inside Facebook Mobile			The Ezra Klein Show			
5	Artificial Intelligence (AI)	0.98847747	5	1	0.988047	5	1	0.9879485	1	0	0.98676234	4	1	0.9864292	1	0	0.01246709
		Master Photography			The Mind Your Business Podca			Millennial Money			Learn Jazz Standards Podcast			Learn Spanish - Survival Guide			
6	The Audacity to Podcast	0.99193287	1	0	0.9896351	3	1	0.9884516	4	1	0.98827356	1	0	0.98823386	1	0	0.010694588
		Nintendo Power Cast - Nintendo Podc			DLC			Giant Bombcast			Dunk Tank			The Easy Allies Podcast			
7	Weapon Wheel Podcast	0.9316918	5	1	0.9296652	4	1	0.9195492	4	1	0.9167763	5	1	0.91253847	4	1	0.077955792
		Our Strange Skies			TROJAN WAR: THE PODCAS			Astonishing Legends			Our Fake History			The Beatles Naked			
8	History Impossible	0.994487	4	1	0.9942412	5	1	0.9938785	5	1	0.9938083	5	1	0.992897	1	0	0.006137592
		Anxiety Slayer™ with Shann and Anar			TEDTalks Health			The Place We Find Ourselves			Yoga Girl: Conversations From The Hear			Tara Brach			
9	Mindful Recovery with Robert Co	0.99210835	5	1	0.9910589	5	1	0.9906711	4	1	0.9906569	5	1	0.99047583	3	1	0.009005784
		The Running for Real Podcast			Drummer's Resource			Team Beachbody Coach Podca			The Unbeatable Mind Podcast with Mark			The Upgrade by Lifehacker			
10	No Meat Athlete Radio	0.99370056	5	1	0.9921067	1	0	0.9917794	4	1	0.99109304	4	1	0.9909824	4	1	0.00806757
		Born to Impact			Team Beachbody Coach Podca			The 5 Minute Mom Podcast			On Purpose with Jay Shetty			Nonprofit Lowdown			
11	Self Improvement Daily	0.98961896	5	1	0.9888803	5	1	0.9885812	2	0	0.98848426	5	1	0.9883164	4	1	0.01122376
R1- Recommended Podcast 1		F Rating- Feedback Rating			E/N-Expected/Unexpected Result			Similarity- Cosine Similarity									



K Means

	Testing Podcast	R1			R2			R3			R4			R5			Diversity
		Similarity	F Rating	E/N	Similarity	F Rating	E/N	Similarity	F Rating	E/N	Similarity	F Rating	E/N	Similarity	F Rating	E/N	
1	The Daily	Impeachment Inquiry: Updates from The Washing			The Daily 202's Big Idea			The 11th Hour with Brian Williams			Article II: Inside Impeachment			Impeachment: A Daily Podcast			
		0.5301869568	5	1	0.480226459	3	1	0.4790184498	1	0	0.4721265528	5	1	0.462456864	5	1	0.5151969435
2	The Science Hour	Science in Action			CrowdScience			Business Daily			The Inquiry			Climate One			
		0.597437843	5	1	0.4180773415	4	1	0.2279512607	2	0	0.209187207	2	0	0.187258757	3	1	0.672017518
3	Music and the Brain	Sticky Notes: The Classical Music Podcast			Classical Classroom			The Music Box			Inside the Musician's Brain			Classics For Kids			
		0.2017637712	5	1	0.1732792171	4	1	0.1639423997	5	1	0.157075688	5	1	0.152091750	3	1	0.8303694346
4	True Crime and Mysteries	True Crime Fan Club Podcast			True Crime All The Time Unsolved			Don't Talk to Strangers			Crime Beat			Southern Gone			
		0.1184186211	5	1	0.1112349784	4	1	0.1084396988	4	1	0.09563331	4	1	0.093586103	2	0	0.8945374576
5	Artificial Intelligence (AI)	Science Fantastic with Dr. Kaku			Inquiring Minds			Science Weekly			The Naked Scientists Podcast			Science of Reading: The Podcast			
		0.1368110641	3	1	0.1154987584	4	1	0.1131930624	3	1	0.09386307687	3	1	0.074728156	2	1	0.8931811763
6	The Audacity to Podcast	Gadget Lab: Weekly Tech News			The Ezra Klein Show			Most Useful Podcast Ever			The Creative Penn Podcast For Writers			Programming Throwdown			
		0.1263016003	3	1	0.102578567	2	0	0.09871410578	4	1	0.09519640407	3	0	0.088738098	2	0	0.8976942449
7	Weapon Wheel Podcast	Podcast Unlocked			Fortnite world			DLC			Sacred Symbols: A PlayStation Podcast			The Gaming Hub: Your Home for Xbo			
		0.2345693828	4	1	0.226979676	5	1	0.1939327031	4	1	0.1931651967	5	1	0.160197928	5	1	0.7982310225
8	History Impossible	Throughline			Dan Snow's History Hit			15 Minute History			Ben Franklin's World: A Podcast About Ear			The Backside Of Water - A Disneylan			
		0.1295963908	2	0	0.1273672477	5	1	0.1261976241	5	1	0.1215425519	5	1	0.116359881	5	1	0.8757872607
9	Mindful Recovery with Robe	The Trauma Therapist			The JOY Factor: Mindfulness, Compassion, Positive Psych			The Higher Practice Podcast for Mental Health Providers			Craft A Life You Love			Hay House Live!® Podcast			
		0.261160127	5	1	0.1795969481	5	1	0.09882532029	5	1	0.09774937394	5	1	0.086602522	5	1	0.8552131416
10	No Meat Athlete Radio	Food for Thought: The Joys and Benefits of Living			The Strength Running Podcast			Peak Human - Unbiased Nutrition Info for Optimum Health, Fitn			at Your Way To A Healthier Life			Plant Proof - Plant Based Nutrition &			
		0.3737150429	5	1	0.1937430377	5	1	0.1342032116	4	1	0.1322374936	4	1	0.130860451	4	1	0.8070481525
11	Self Improvement Daily	Armstrong and Getty On-Demand			The Brian Mendler Show			The Cult of Pedagogy Podcast			Learn Spanish - Survival Guide			The Brian Buffini Show			
		0.08332606157	1	0	0.05636715197	3	1	0.03608770387	3	1	0.03125603036	2	1	0.030040018	5	1	0.9525846068
R1- Recommended Podcast 1		P Rating- Personlised Rating		E/N-Expected/Unexpected Result			Similarity- Cosine Similarity										

Overall, our podcast recommendation system performed well in terms of providing relevant and diverse recommendations to users. Both word2vec + cosine similarity and Kmeans + tfidf + cosine algorithms are effective in generating personalized recommendations based on the user's preferences.

Chapter 5

Conclusion and Future Work

5.1 Conlusion

In summary, our project successfully developed a recommendation model that recommend podcasts based on user's preferences. It clusters genres effectively and provides highly rated podcast recommendations based on those genres. The model consistently delivers sensible recommendations for over 10 podcasts without any significant deterioration. Overall, it outperformed other models tested and provides relevant and accurate recommendations for users based on their chosen genres.

The web app's use of content-based filtering and clustering algorithms, along with continuous monitoring and evaluation metrics, makes it a useful tool for users seeking personalized recommendations. With the ever-growing data and advancements in algorithms, we can expect even more accurate and personalized recommendations for podcast listeners in the future. Our project demonstrates the potential of using machine learning algorithms to enhance the user experience in the podcast industry.

5.2 Future Work

There are several opportunities for future work to further improve the podcast recommendation web app. One possible direction is to collect more podcast data, including podcast transcriptions, to improve the accuracy of the recommendations. By including transcriptions, the system can analyze the content of the podcast episodes and provide more granular recommendations based on specific topics.

Another area for improvement is to recommend specific podcast episodes instead of just the podcast itself. This can be achieved by analyzing the content of each episode and providing granular recommendations based on the user's preferences.

Additionally, creating a smaller or larger look back window for episode titles and descriptions can improve the relevance of the recommendations. A smaller look back window can provide more recent recommendations, while a larger window can provide a more comprehensive view of the podcast's content.

Another possible improvement is to consider sub-genres for specificity. By including sub-genres, the system can provide more targeted recommendations based on the user's specific interests.

Implementing more computationally heavy algorithms like BERT can also improve the accuracy of the recommendations. BERT is a pre-trained language model that can understand the context of words in a sentence and can provide more nuanced recommendations based on the user's preferences.

Lastly, trying collaborative filtering can be a promising approach for the podcast recommendation web app. Collaborative filtering is a technique that uses the preferences of similar users to generate personalized recommendations. By incorporating this method, the system can provide even more personalized recommendations based on the user's listening history and preferences.

Appendix A

Appendix A

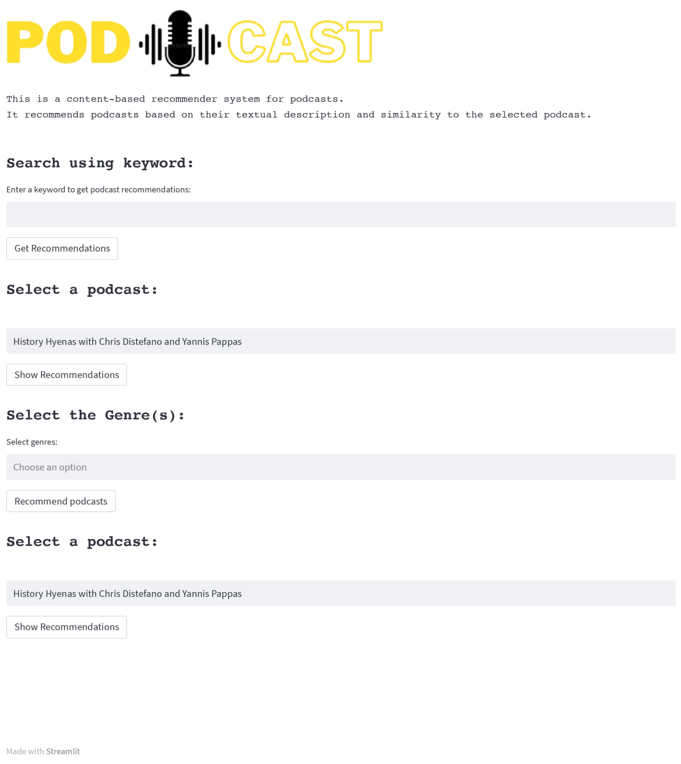
Code Snippets:

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import pairwise_distances


# Convert the preprocessed descriptions into numerical features
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(df['text'])

# Train a KMeans clustering model
kmeans = KMeans(n_clusters=100, random_state=42)
kmeans.fit(X)


# Assign each podcast episode to a cluster
cluster_labels = kmeans.predict(X)


# Recommend episodes from the same cluster as a given podcast name
def recommend_episodes(podcast_name, n_recommendations=5):
    # Find the indices of all episodes of the given podcast name
    podcast_episodes = df.index[df['title'] == podcast_name].tolist()

    # Find the cluster labels of all episodes of the given podcast name
    podcast_clusters = [cluster_labels[episode_id] for episode_id in podcast_episodes]

    # Find the indices of all episodes in the same clusters as the given podcast
    cluster_episodes = []
    for cluster_label in set(podcast_clusters):
        cluster_indices = np.where(cluster_labels == cluster_label)[0]
        cluster_episodes.extend(cluster_indices)

    # Remove the indices of the episodes of the given podcast
    for episode_id in podcast_episodes:
        if episode_id in cluster_episodes:
            cluster_episodes.remove(episode_id)

    # Calculate cosine similarity between each recommended podcast and the give
    similarities = []
    for episode_id in cluster_episodes:
        similarity = pairwise_distances(X[episode_id], X[df.index[df['title'] =
        similarities.append(similarity[0][0])

    # Choose a random set of recommendations from the same clusters, sorted by
    recommendations = [x for _, x in sorted(zip(similarities, cluster_episodes)
    recommendations = recommendations[:min(n_recommendations, len(recommendatio

    return recommendations


import numpy as np
import pandas as pd
import nltk
from gensim.models import Word2Vec
from sklearn.metrics.pairwise import cosine_similarity

# define a custom tokenizer to tokenize the text
class MyTokenizer:
    def __init__(self):
        pass

    def fit(self, X, y=None):
        return self

    def transform(self, X):
        transformed_X = []
        for document in X:
            tokenized_doc = []
            for sent in nltk.sent_tokenize(document):
                tokenized_doc += nltk.word_tokenize(sent)
            transformed_X.append(np.array(tokenized_doc))
        return np.array(transformed_X)

    def fit_transform(self, X, y=None):
        return self.transform(X)

# define a mean embedding vectorizer
class MeanEmbeddingVectorizer(object):
    def __init__(self, word2vec):
        self.word2vec = word2vec
        # if a text is empty we should return a vector of zeros
        # with the same dimensionality as all the other vectors
        self.dim = len(word2vec.wv.vectors[0])

    def fit(self, X, y=None):
        return self

    def transform(self, X):
        X = MyTokenizer().fit_transform(X)

        return np.array([
            np.mean([self.word2vec.wv[w] for w in words if w in self.word2vec.wv]
                    or [np.zeros(self.dim)], axis=0)
            for words in X
        ])

    def fit_transform(self, X, y=None):
        return self.transform(X)

# load the podcasts data
podcasts_df = pd.read_csv("podcasts.csv")

# train the Word2Vec model
text_list = list(podcasts_df.text)
tokenized_text = [nltk.word_tokenize(i) for i in text_list]
w2v_model = Word2Vec(tokenized_text, sg=1)

# convert the text into mean word embeddings using Word2Vec
mean_embedding_vectorizer = MeanEmbeddingVectorizer(w2v_model)
mean_embedded = mean_embedding_vectorizer.fit_transform(podcasts_df['text'])

# compute the cosine similarity matrix
w2v_cosine_sim = cosine_similarity(mean_embedded)

# define a function to recommend podcasts based on cosine similarity
def recommend_podcasts_w2v(title, cosine_sim=w2v_cosine_sim, n=5):
    # get the index of the podcast that matches the title
    indices = pd.Series(podcasts_df.index, index=podcasts_df['title'])
    idx = indices[title]
```

Appendix B

Appendix B

Web App Interface



This is a content-based recommender system for podcasts.
It recommends podcasts based on their textual description and similarity to the selected podcast.

Search using keyword:

Enter a keyword to get podcast recommendations:

Select a podcast:

History Hyenas with Chris Distefano and Yannis Pappas

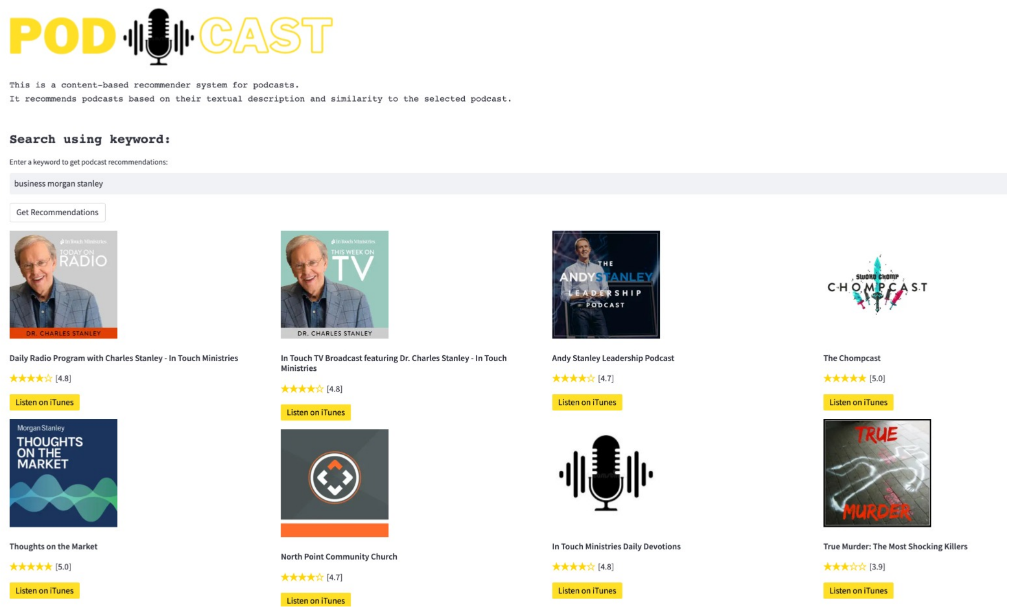
Select the Genre(s):

Select genres:

Choose an option

Select a podcast:

History Hyenas with Chris Distefano and Yannis Pappas



```
This is a content-based recommender system for podcasts.
It recommends podcasts based on their textual description and similarity to the selected podcast.
```

Search using keyword:

Enter a keyword to get podcast recommendations:

business morgan stanley

Get Recommendations

Daily Radio Program with Charles Stanley - In Touch Ministries
★★★★☆ [4.8]
Listen on iTunes

In Touch TV Broadcast featuring Dr. Charles Stanley - In Touch Ministries
★★★★☆ [4.8]
Listen on iTunes

Andy Stanley Leadership Podcast
★★★★☆ [4.7]
Listen on iTunes

The Chompcast
★★★★★ [5.0]
Listen on iTunes

Thoughts on the Market
★★★★★ [5.0]
Listen on iTunes

North Point Community Church
★★★★☆ [4.7]
Listen on iTunes

In Touch Ministries Daily Devotions
★★★★☆ [4.8]
Listen on iTunes

True Murder: The Most Shocking Killers
★★★☆☆ [3.9]
Listen on iTunes

Bibliography

[1] N. Hariri, B. Mobasher, and R. Burke. Context-aware music recommendation based on latenttopic sequential patterns. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 131–138, 2012.

[2] M. Kaminskas and D. Bridge. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 7 (1):1–42, 2016.

[3] S. Reddy, M. Lazarova, Y. Yu, and R. Jones. Modeling language usage and listener engagement in podcasts. *arXiv preprint arXiv:2106.06605*, 2021.

[4] M. Schedl. Listener-aware music recommendation from sensor and social media data. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part III 15*, pages 213–217. Springer, 2015.

[5] Z. Xing, M. Parandehgheibi, F. Xiao, N. Kulkarni, and C. Pouliot. Content-based recommendation for podcast audio-items using natural language processing techniques. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 2378–2383. IEEE, 2016.

[6] L. Yang, M. Sobolev, C. Tsangouri, and D. Estrin. Understanding user interactions with podcast recommendations delivered via voice. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 190–194, 2018.