Hit Song Prediction Report

ON

Hit Song Prediction based on Machine Learning algorithms & data visualisation.

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY,

IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING

By

Ankush Chimnani (71721088L)

Anish Dhamelia (71721097K)

Krishnakant Pathak (71721144E)

Mukesh Singh (71721280H)

UNDER THE GUIDANCE OF

Prof. Priyadarshini Patil

DEPARTMENT OF COMPUTER ENGINEERING

D. Y. PATIL COLLEGE OF ENGINEERING, AMBI

Sr.No.124 & 126, A/P Ambi, MIDC Road. Tal Maval,

Talegaon Dabhade, Pune - 410 506

ACADEMIC YEAR: 2018

DEPARTMENT OF COMPUTER ENGINEERING

D. Y. PATIL COLLEGE OF ENGINEERING, AMBI

Sr.No.124 & 126, A/P Ambi, MIDC Road. Tal Maval,

Talegaon Dabhade, Pune - 410 506

CERTIFICATE

This is to certify that the Hit Song Prediction based on Machine Learning algorithms and data visualisation.

Submitted by

Ankush Chimnani

Anish Dhamelia

Krishnakant Pathak

Mukesh Singh

Is a Bonafide work carried out by her under the supervision of **Prof. Priyadarshini Patil** and it is submitted towards the partial fulfilment of Bachelor of Engineering in Computer Engineering through the Savitribai Phule Pune University during the academic year 2019-20.

Prof. Priyadarshini Patil	Prof. Pradnya Randive& Prof. Meghna Solanki
Project Guide	Coordinator
Prof. Mininath Nighot	Dr. Abhay A. Pawar
HOD	Principal
Place:Pune	Date:

ACKNOWLEDGMENT

With immense pleasure, we are presenting the Hit Song Prediction report as part of the curriculum of the B. E. Computer Engineering. We wish to thank all the people who gave us an unending support right from the idea was conceived.

We express my sincere and profound thanks to our Head of the Department **Prof. Mininath Nighot** and my Seminar Guide **Prof. Priyadarshini Patil** for their guidance and motivation for completing my work, and we are also thankful to all those who directly or indirectly guided and helped me in preparation of this seminar.

Ankush Chimnani

Anish Dhamelia

Krishnakant Pathak

Mukesh Singh

ABSTRACT

Music is one of the finest forms of human expression and has existed since the beginning of civilization. This project is focused on predicting the songs that ruled their respective generations and give an idea about how human civilization has evolved and progressed with the passage of time. It makes use of the algorithms based on machine learning and data visualization to predict whether a particular song is hit or not. The term hit is a relative reference and therefore, certain parameters have been taken into consideration to predict a particular song as hit or not. The parameters which are used to predict the outcome for a particular song are based on the audio features of the song. Since the audio features are not perfectly balanced from the dataset perspective, the use of a general classification accuracy as the basis for algorithmic analysis might not be the best idea. Although, methods like confusion matrix, Area under the curve, and Receiver Operator Curve might be used for evaluation. Exploring the possibility of predicting hit songs is both interesting from a scientific point of view and something that could be beneficial to the music industry. In this research we raise the question if it is possible to classify a music track as a hit or a non-hit based on its audio features. We investigated which machine learning algorithms could be suited for a task like this. Four different models were built using various algorithms such as Support Vector Machine and Gaussian Naive Bayes. The obtained results do not indicate that it is possible to predict hit songs on our particular dataset. This stands in contrast to some previous research within this field.

Contents

List of Figures

List of Tables

1.INTRODUCTION

1.1 Motivation

Hit Song Science is a term coined by Mike McCready and trademarked by the company he co-founded, Polyphonic HMI. It concerns the possibility of predicting whether a song will be a hit, prior to its distribution using automated means such as machine learning software.

The scientific nature of Hit Song Science is a subject of debate in the music information retrieval (MIR) community. Early studies claimed that using machine learning techniques could capture some information from audio signals and lyrics that would explain[1] popularity. However, a larger-scale evaluation[2] contradicts the claims of “Hit Song Science”, i.e. that the popularity of a music title can be learned effectively from known audio features. Many other reasons, including the well-known "cumulative advantage" or preferential attachment effects deeply contradicts[3] the possibility of practical applications. Nevertheless, automatic prediction techniques are the basis of hit counselling businesses (HSS Technology). Recent work by Herriman’s et al. [4] has shown that audio features can indeed be used to outperform a random oracle when predicting top 10 versus top 30-40 hits.

Being able to predict whether a song can be a hit has important applications in the music industry. Although it is true that the popularity of a song can be greatly affected by external factors such as social and commercial influences, to which degree audio features computed from musical signals (whom we regard as internal factors) can predict song popularity is an interesting research question on its own. Motivated by the recent success of deep learning techniques, we attempt to extend previous work on hit song prediction by jointly learning the audio features and prediction models using machine learning. Specifically, we experiment with a logistic regression model, Naïve Bayes model and SVM that takes the particular song as the input for feature learning, that uses an external song dataset for supervised pre-training and auto-tagging, and the combination of these two models. We also consider the random forest model to characterize audio information in different scales.

The popularity of a song can be predicted by:

- Number of digital downloads
- Number of streams on a particular application
- Whether the song has been listed or verified by an agency such as Billboard
- Features of the song in terms of audio features such as timber, loudness.
- Analysis of social and commercial factors.
- Analysis of trends and patterns for a particular zone in time.

1.2 Project Idea

The term “hit” is of relative reference when it comes to music. It is based on certain non-predictable factors such as era in which a particular song, the social influence and the cultural history of a particular location. For a particular song to be hit or not, the timing of release also plays a vital role. For example, the Christmas folk songs might see a surge in their online streams on a particular platform during the Christmas season. But, certain measurable characteristics such as duration of the song, loudness, etc. can be used to predict the response for a particular song.

1.3 Problem Statement

The aim of this thesis is to investigate the question is it possible to predict Hit Songs with Machine Learning using Audio Features.

1.4 Goals and Objective

Some studies have tried to address the question whether or not it is possible to use machine learning techniques to predict hit songs. The purpose of this research is to further investigate this subject. To be able to analyse music tracks, a dataset is compiled based on data from Spotify’s API. The dataset consists of a number of audio features for each song.

Afterwards, a number of machine learning algorithms is applied to this dataset. The accuracies of the different models are then compared. Thus, the goal of this project is to examine if machine learning algorithms can be used to predict hit songs and with what accuracy this can be done.

1.5 Statement of Scope

The scope of this research is to investigate if it is possible to predict a hit song based on 13 audio features of a track. Four different machine learning techniques are used: Logistic Regression, K-Nearest Neighbours, Gaussian Naive Bayes and Support Vector Machine. The focus of our research is to investigate if these models can be applied on our dataset and to compare their performances. We are aware of the fact that there are different aspects that might affect if a track will become popular, e.g. social or economic. However, in this research we decided to focus only on the audio features of a music track. Because of time constraints we did not have the time to do any optimisation of the models. There was also a limitation in what metadata of the tracks we could obtain from the Spotify API. Unfortunately, the number of streams was not available from the API, which possibly could have provided value to the experiment. Another important parameter that was not accessible from the API was the popularity of an artist at a specific time.

1.6 Outcome

- The application can be used to predict a particular song would be hit or not. Also, it would also help in trend analysis for a particular era. Also, from a futuristic point of view, it would encourage further development of such applications based on other parameters.

2.LITERATURE SURVEY

2.1 Literature Survey

For proposed work to be better one following literature is analysed for existing systems working and critically evaluated on some evaluation method to find shortcomings from them.

[1] "Herremans, D. & Bergmans, T.", Hit song prediction on early adopter data and audio features."ISMIR 2017

In this study, a large dataset of social listening behaviour gathered through the Last.FM API was created. The dataset also contained hit/non-hit information from the Belgian Dance charts Ultratop 50, and audio data gathered from EchoNest. In future research, it would be interesting to perform parameter tuning of the models and look at an embedded model that combines both audio and social listening data to further enhance hit prediction accuracy. In order to advance this field, it would also be beneficial to have an open dataset with fixed hit definitions for benchmarking.

[2] "Yang, L.-C., Chou, S.-Y., Liu, J.-Y., Yang, Y.-H., & Chen, Y.-A", "Revisiting the problem of audio-based hit song prediction using convolutional neural networks2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)".

In this paper, we have introduced state-of-the-art deep learning techniques to the audio-based hit song prediction problem. Instead of aiming at classifying hits from non-hits, we formulate it as a regression problem. Evaluations on the listening data of Taiwanese users of a streaming company called KKBOX confirms the superiority of deep structures over shallow structures in predicting song popularity.

[3] "Dorien Herremansa*, David Martens and Kenneth S¨orensena", Dance hit songs prediction (JIMR 2014).

This research proves that popularity of dance songs can be learnt from the analysis of music signals. Previous less successful results in this field speculate that their results could be due to features that are not informative enough [Pachet and Roy, 2008]. The positive results from this paper could indeed, be due to the use of more advanced temporal features. A second cause might be the use of "recent" songs only, which eliminates the fact that hit music evolves over time.

[4] "Holly Silk, Raul Santos-Rodriguez, Cedric Mesnage, Tijl De Bie", DATA SCIENCE FOR THE DETECTION OF EMERGING MUSIC STYLES (ISMIR 2016).

We presented a website which allowed intuitive exploration of this vast amount of data, focusing on the four key features of artists, genres, locations, and fans. Looking forward, we wish to incorporate more advance data mining methodologies into the site. For example, we would like to automatically discover communities within the Twitter network of fans, or deploy novelty detection algorithms on the audio data we have collected, hoping it would help us to discover emerging music styles.

[5] "Vinitha S, Sweetlin S, Vinusha H and Sajini S", DISEASE PREDICTION USING MACHINE LEARNING OVER BIG DATA (CSEIJ 2018).

In this paper, it bid a Machine learning Decision tree map algorithm by using structured and unstructured data from hospital. It also uses Map Reduce algorithm for partitioning the data. To the highest of gen, none of the current work attentive on together data types in the zone of remedial big data analytics. Compared to several typical calculating algorithms, the scheming accuracy of our proposed algorithm reaches 94.8% with an regular speed which is quicker than that of the CNN-based unimodal disease risk prediction (CNN-UDRP) algorithm and produces report. The report consists of possibility of occurrences of diseases.

[6] "Dhwaani Parikh, Vineet Menon" Machine Learning Applied to Cervical Cancer Data (MECS 2019)

It is found that all the models are good, that is k-nearest neighbour, decision tree and random forest. K-nearest neighbour seems to be the best model with higher accuracy of the model, Higher AUC which as compared Decision tree and random forest which is very low. Precision recall and f1 score is also high for nearest-neighbour model. The f1 score for nearest-neighbour is 0.94 which is high when compared to decision tree and random forest.

[7] "R Dhanaraj, B Logan"Automatic hit songs prediction

Our results indicate that for the features used, lyric based features are slightly more effective than audio-based features at distinguishing hits. Combining features does not significantly improve performance. Analysis of the best lyric-based system shows that the absence rather than the presence of certain semantic information in the lyrics mean a song is more likely to be a hit.

[8] "Sebastian Raschka", Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning, University of Wisconsin–Madison Department of Statistics(2018).

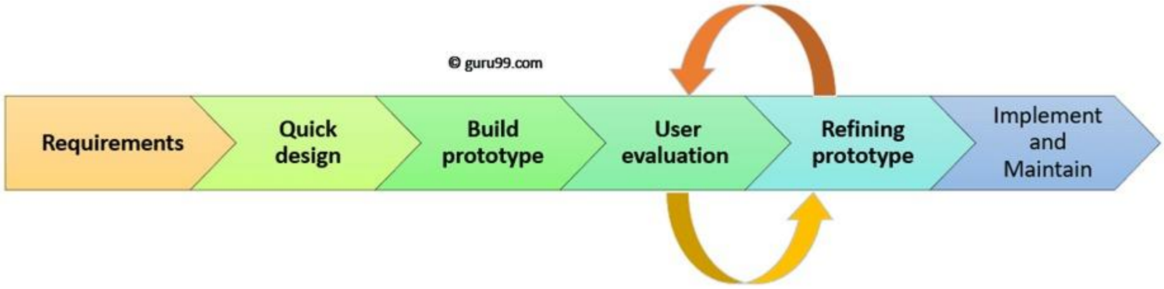
Since "a picture is worth a thousand words," I want to conclude this series on model evaluation, model selection, and algorithm selection with a diagram (Figure 23) that summarizes my personal recommendations based on the concepts and literature that was reviewed. It should be stressed that parametric tests for comparing model performances usually violate one or more independent assumptions (the models are not independent because the same training set was used, and the estimated generalization performances are not independent because the same test set was used.). In an ideal world, we would have access to the data generating distribution or at least an almost infinite pool of new data. However, in most practical applications, the size of the dataset is limited; hence, we can use one of the statistical tests discussed in this article as a heuristic to aid our decision making.

3.PROJECT PLAN

3.1 Development Methodology

Our project follows the prototype model. The steps involved are:

- Requirement gathering and analysis
- Quick Design
- Build a Prototype
- Initial user evaluation
- Refining prototype
- Implement product and maintain



Implementation.

- Integration and System Testing.
- Operation and Maintenance.

3.3 Software Requirements

The software requirements for the project are:

- Operating System: Linux, Widows, Mac OS
- Front End: Anaconda, Spyder
- Programming Language: Python 3, R(for visualization)
- Latest available NVIDIA drivers
- Database operations: MySQL workbench

3.4 Hardware Requirement

Table 2 shows our project Risk analysis:

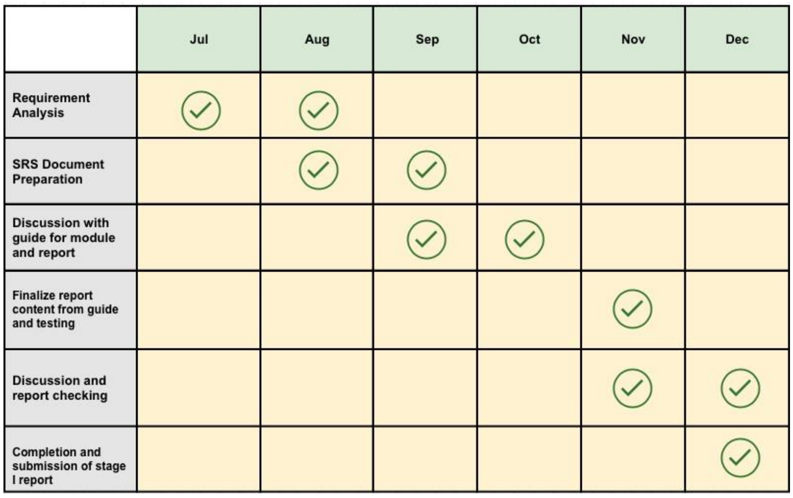
Sr. No.	Parameter	Minimum Requirements
1	System	Intel 2.5 GHz Intel Core i5 and above
2	RAM	4 GB and above
3	Hard Disk	500 GB
4	GPU	NVIDIA GeForce, AMD

Table 2: Hardware Requirements

3.5 Project Schedule

Major Tasks in the Project stages are:

- Task 1: Requirement analysis
- Task 2: SRS document preparation
- Task 3: Discussion with guide for modules and report
- Task 4: Finalize report content from guide and testing
- Task 5: Completion and submission of stage I report



	Jul	Aug	Sep	Oct	Nov	Dec
Requirement Analysis	✓	✓				
SRS Document Preparation		✓	✓			
Discussion with guide for module and report			✓	✓		
Finalize report content from guide and testing					✓	
Discussion and report checking					✓	✓
Completion and submission of stage I report						✓

Figure 2: Project Plan

4.SOFTWARE REQUIREMENT SPECIFICATION

4.1 Purpose and Scope of Document

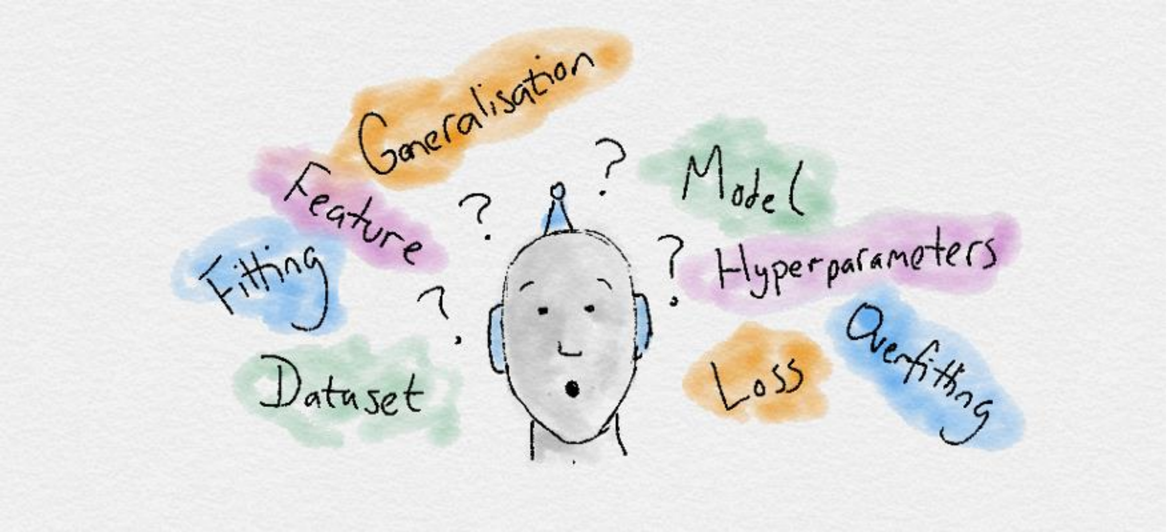
Machine learning:

Machine learning tools wants to give programs the ability to learn and adapt (Shalev-Schwartz & Ben-David 2014). Tom Mitchell provided a formal definition that says:” A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E’’ (Mitchell et al. 1997). In other words, a program is said to learn if the performance of a certain task advances with training.

. In the past 50 years, there has been an explosion of data. This mass of data is useless unless we analyse it and find the patterns hidden within. Machine learning techniques are used to automatically find the valuable underlying patterns within complex data that we would otherwise struggle to discover. The hidden patterns and knowledge about a problem can be used to predict future events and perform all kinds of complex decision making.

Most of us are unaware that we already interact with Machine Learning every single day. Every time we Google something, listen to a song or even take a photo, Machine Learning is becoming part of the engine behind it, constantly learning and improving from every interaction. It’s also behind world-changing advances like detecting cancer, creating new drugs and self-driving cars.

Terminology: Terms which used in machine learning frequently are listed below.

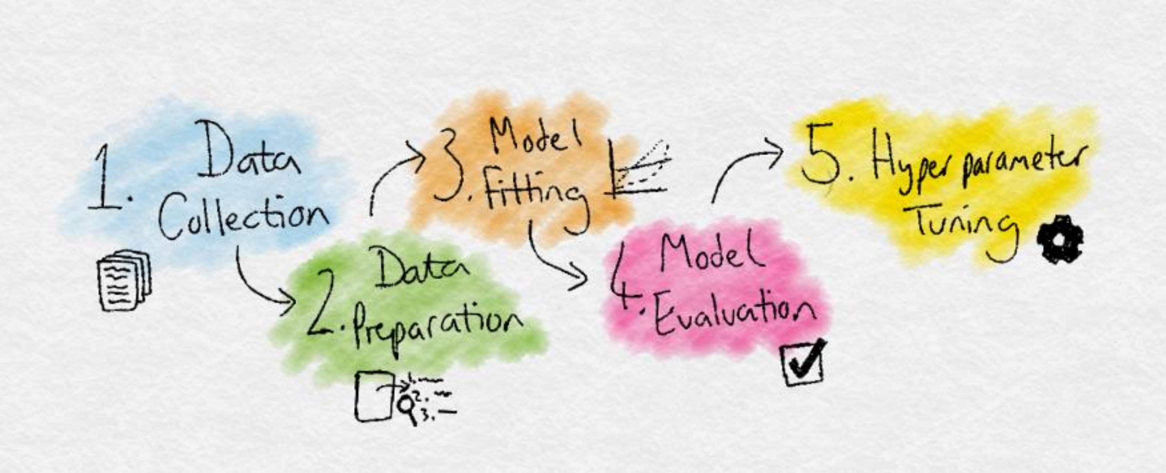


- **Dataset**: A set of data examples, that contain features important to solving the problem.

- **Features**: Important pieces of data that help us understand a problem. These are fed in to a Machine Learning algorithm to help it learn.

- **Model**: The representation (internal model) of a phenomenon that a Machine Learning algorithm has learnt. It learns this from the data it is shown during training. The model is the output you get after training an algorithm. For example, a decision tree algorithm would be trained and produce a decision tree model.

Process:



1. **Data Collection:** Collect the data that the algorithm will learn from.

2. **Data Preparation:** Format and engineer the data into the optimal format, extracting important features and performing dimensionality reduction.

3. **Training:** Also known as the fitting stage, this is where the Machine Learning algorithm actually learns by showing it the data that has been collected and prepared.

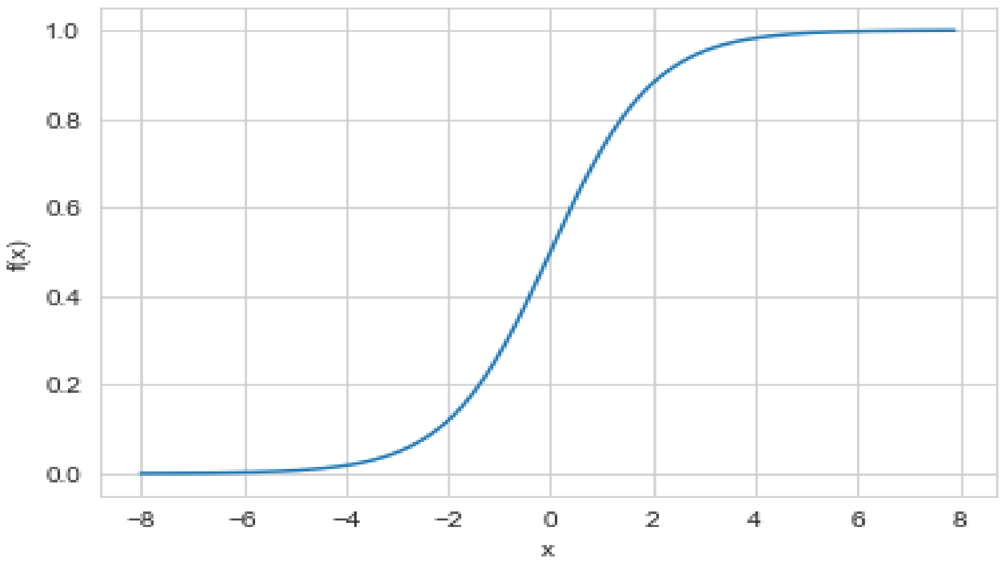
4. **Evaluation:** Test the model to see how well it performs.

5. **Tuning:** Fine tune the model to maximise it's performance.

Machine learning Algorithms:

Logistic Regression:

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, Online transactions Fraud or not Fraud, Tumor Malignant or Benign. Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

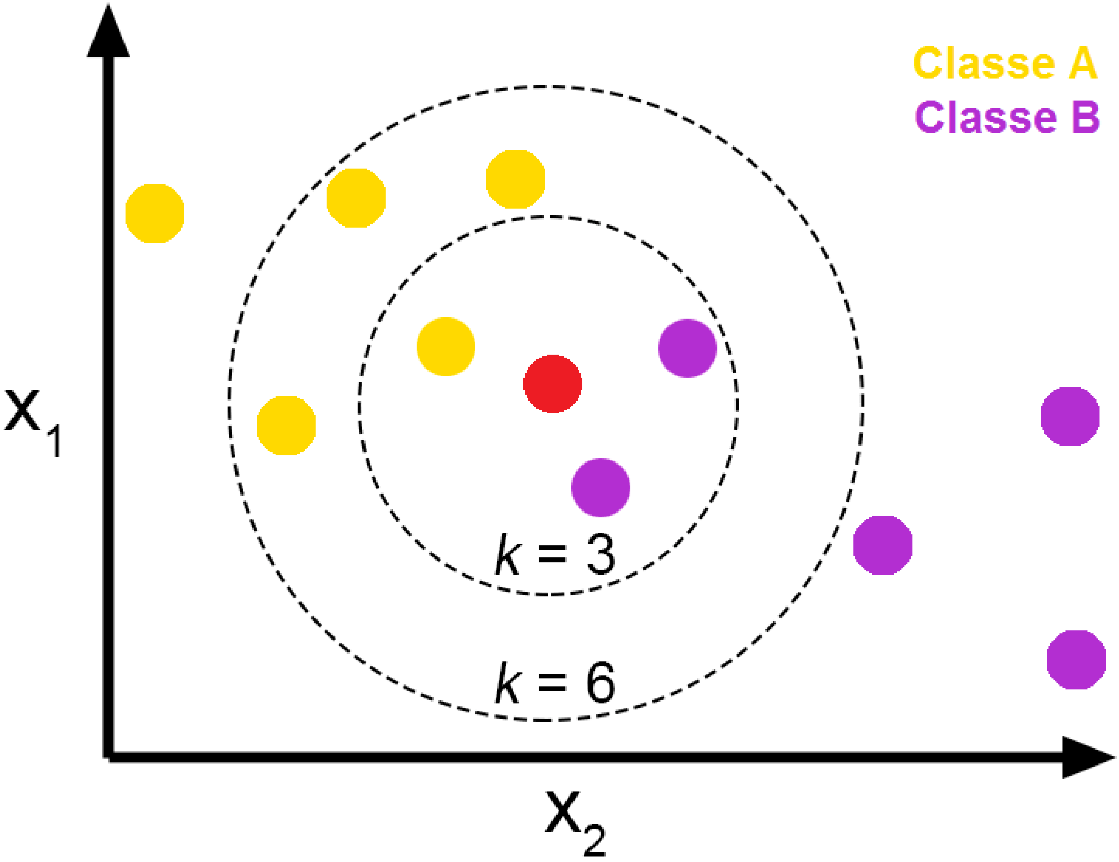


Logistic regression is a mathematical model which can be used to describe the relationship between one or more independent variables and one dependent binary variable (Klein Baum, Klein & Pryor n.d.).

This model is therefore used for problems where the outcome can be classified in one of two categories. When applying the estimated logistic model to new cases of a test dataset, it provides a prediction of success probability, which is a number between 0 and 1. The logistic regression provides a rule for classifying the test data with a cut-off on the predicted success probability (Ledolter 2013).

K-Nearest Neighbours:

KNN (K — Nearest Neighbors) is one of many (supervised learning) algorithms used in data mining and machine learning, it's a classifier algorithm where the learning is based "how similar" is a data (a vector) from other.



K-Nearest Neighbours classification implements learning based on the K nearest neighbours of each query point. This method is a type of non-generalizing learning since it only stores instances of the training data, it does not create an internal model for generalizing the data. The choice of the value of k is dependent on the dataset: if k is set to a low value, the point is assigned to a class with only a few neighbours, and if k is a high value, then the effect of noise is suppressed and the classification is of a more generalized form in Scikit module of python.

Gaussian Naive Bayes:

Bayes' theorem finds many uses in the probability theory and statistics. There's a micro chance that you have never heard about this theorem in your life. Turns out that this theorem has found its way into the world of machine learning, to form one of the highly decorated algorithms. In this article, we will learn all about the Naive Bayes Algorithm, along with its variations for different purposes in machine learning.

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

Above equation gives the basic representation of the Bayes' theorem. Here A and B are two events and**,**

P(A|B): The conditional probability that event A occurs, given that B has occurred. This is also known as the posterior probability.

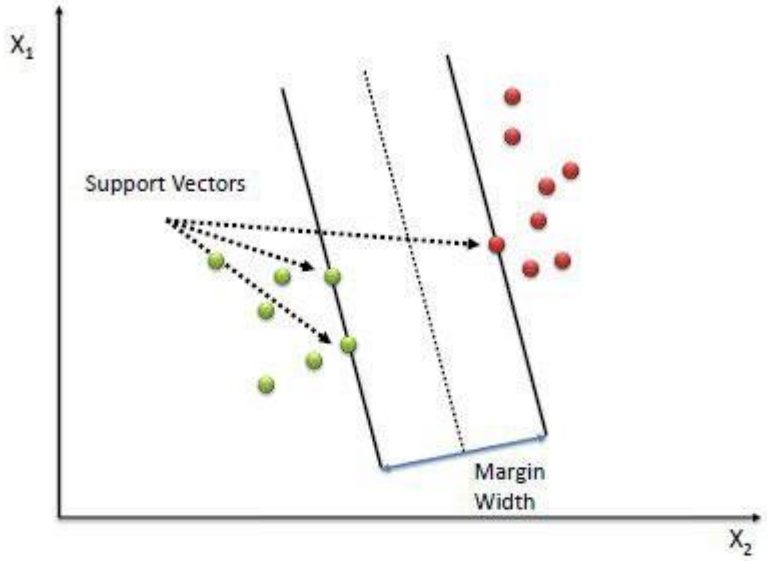
P(A) and P(B): probability of A and B without regard of each other.

P(B|A): the conditional probability that event B occurs, given that A has occurred.

Naive Bayes methods are a set of supervised learning algorithms (Learn 2017). Bayes' theorem is used in these methods with the naive assumption that each feature is independent of every other feature. This is called conditional independence (Zhang 2004). The Naive Bayes methods must be applied to discrete variables but can be extended to continuous variables, most commonly by assuming a Gaussian distribution. This extension is called Gaussian Naive Bayes.

Support Vector Machines:

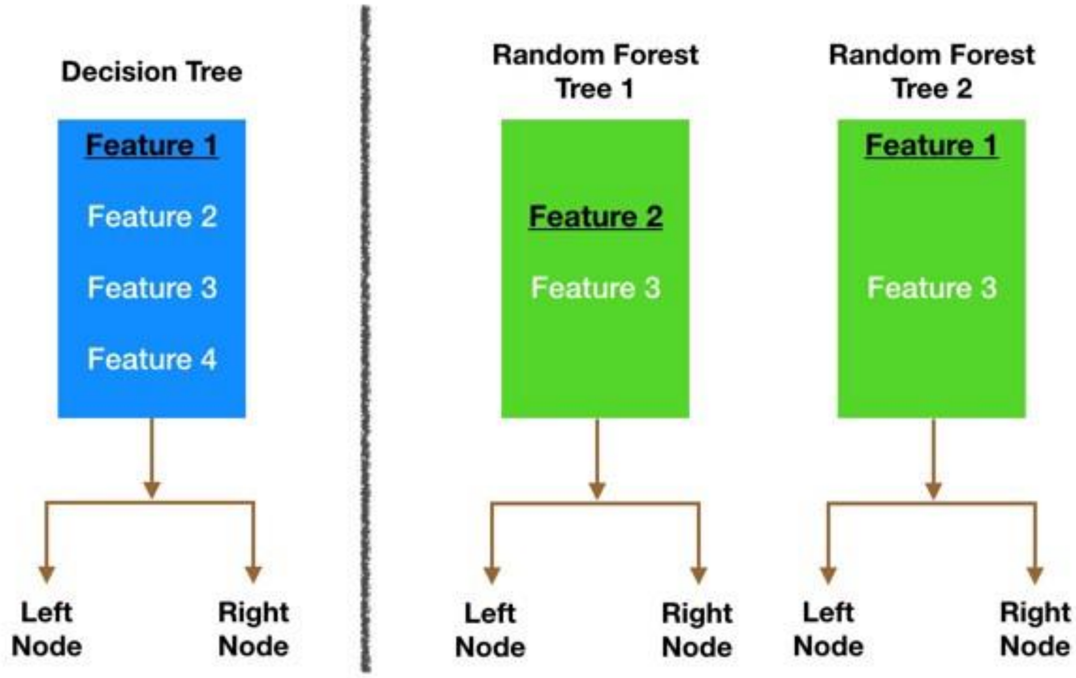
Support Vector Machine are perhaps one of the most popular and talked about machine learning algorithms. They were extremely popular around the time they were developed in the 1990s and continue to be the go-to method for a high performing algorithm with little tuning. In this blog we will be mapping the various concepts of SVC.



A Support Vector Machine is a discriminative classifier formally defined by a separating hyperplane. In other words, given labelled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In a two-dimensional space this hyperplane is a line dividing a plane in two parts with one class on each side of the line. In a 3-dimensional space, this separator is instead a plane separating the different classes. The Support Vector Machine attempts to find a separating hyperplane with a margin that is as large as possible from the nearest data points (OpenCV 2017).

Random Forest:

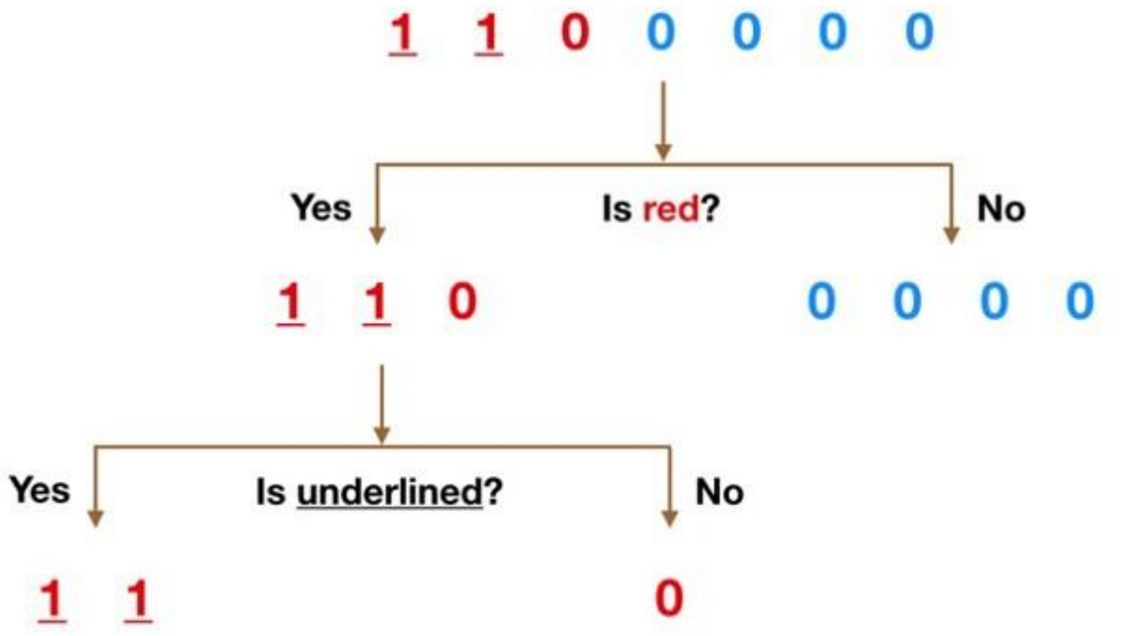
The random forest is a classification algorithm consisting of many decisions trees**.** It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.



Random forest algorithm is a supervised classification algorithm. As the name suggest, this algorithm creates the forest with a number of trees. In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results. For creation of decision trees, C4.5 algorithm is used which is an extension of ID3 algorithm.

Decision Trees:

Decision trees as they are the building blocks of the random forest model. Fortunately, they are pretty intuitive. I'd be willing to bet that most people have used a decision tree, knowingly or not, at some point in their lives.



Imagine that our dataset consists of the numbers at the top of the figure to the left. We have two 1s and five 0s (1s and 0s are our classes) and desire to separate the classes using their features. The features are color (red vs. blue) and whether the observation is underlined or not. So how can we do this?

Color seems like a pretty obvious feature to split by as all but one of the 0s are blue. So we can use the question, “Is it red?” to split our first node. You can think of a node in a tree as the point where the path splits into two — observations that meet the criteria go down the Yes branch and ones that don’t go down the No branch.

The No branch (the blues) is all 0s now so we are done there, but our Yes branch can still be split further. Now we can use the second feature and ask, “Is it underlined?” to make a second split.

The two 1s that are underlined go down the Yes subbranch and the 0 that is not underlined goes down the right subbranch and we are all done. Our decision tree was able to use the two features to split up the data perfectly.

4.2 Overview of responsibilities of Developer

- Develop technical and functional specifications for projects.
- Assist in determining time and cost estimates for assigned projects.
- Develop new applications or make enhancements according to project needs.
- Utilize programming principles, tools, and techniques to write application codes.
- Plan, coordinate and execute project activities to ensure timely completion.
- Ensure project deliverables meet business requirements.
- Prepare test cases and strategies for unit testing and integration testing.
- Perform code reviews to identify basic technical and logical errors.
- Resolve application development issues in a timely manner - Manage project risks, and milestones.
- Develop best practices to improve productivity.

4.3 Use-case Diagram

A use case diagram graphically depicts interaction between the system and the users of the system. In terms of software engineering, a use case is a set of steps defining interaction between an actor (user) and the system. Use case diagram for the system is as shown

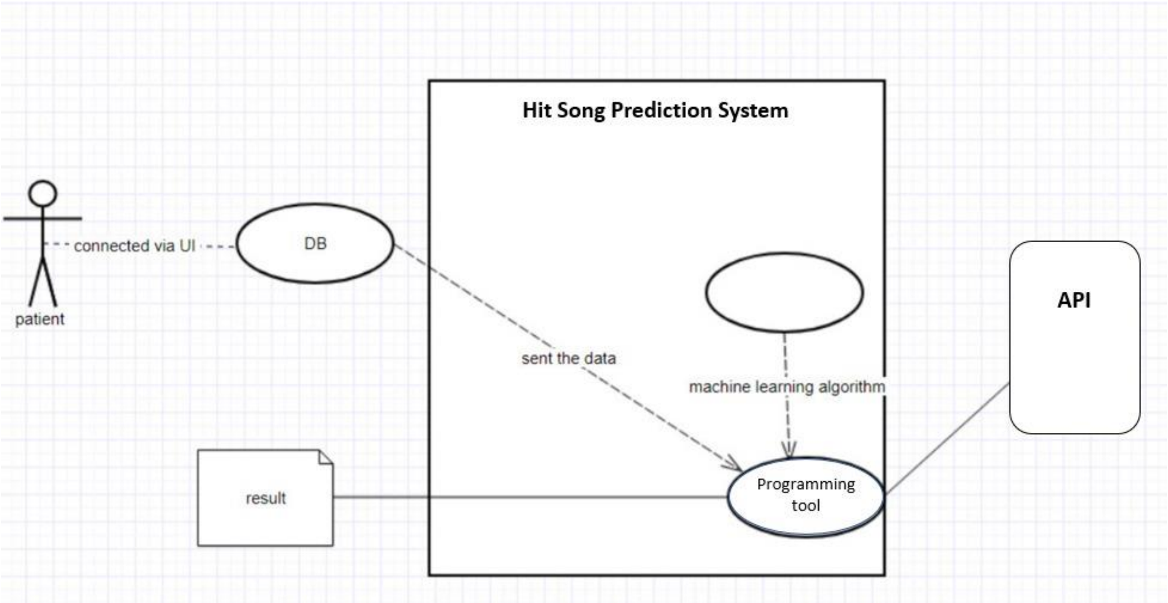


Figure 3: Use-case Diagram

4.4 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So, deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

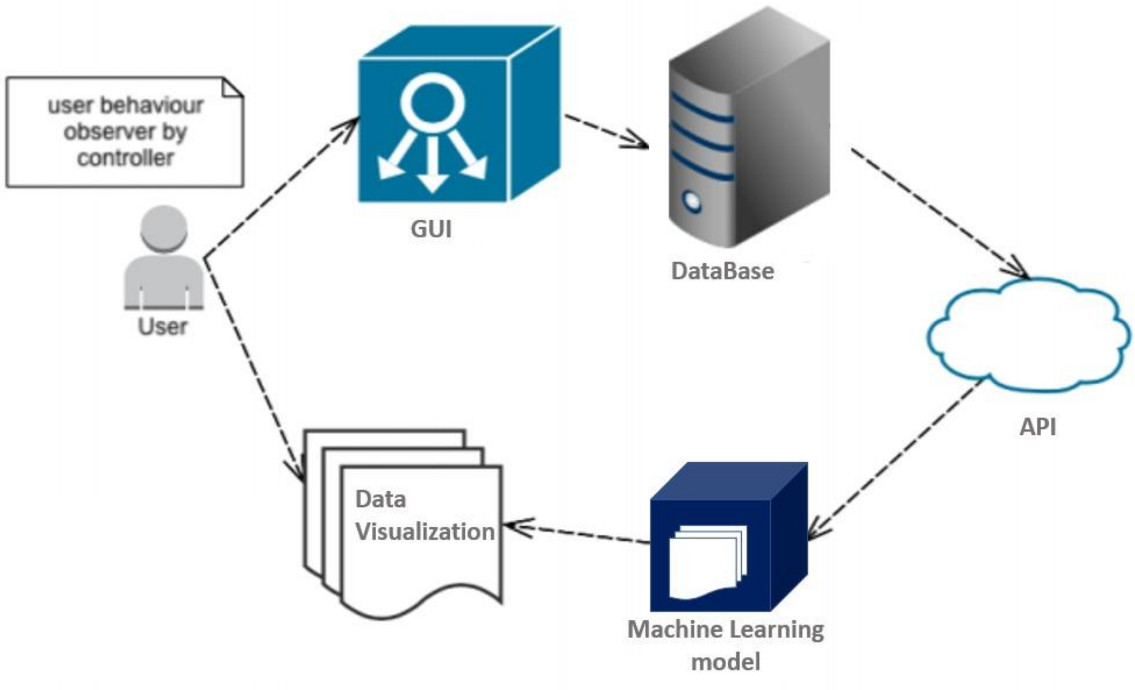


Figure 4: Deployment Diagram

5.DETAILED DESIGN

5.1 Introduction

The dataset used in this research consists of both hit songs and non-hit songs. The hit songs were retrieved from the Billboard Hot 100, between the years 2016 and 2018. The hit dataset consists of 287 tracks (duplicates were removed). The non-hits songs were obtained by gathering 25 songs from 13 different genres, resulting in a set of 322 tracks, after removing the tracks that had appeared on the Billboard Hot 100 between 2016 to 2018. The total dataset resulted in a size of 609 tracks. To obtain the audio features of the tracks we used Spotify's open Web API. To retrieve the data from the Spotify API a lightweight Python library was used, called Spotify. In most cases, the pre-processing part of the workflow is cleaning the data, for instance remove null values and values that differ a lot from the rest. To avoid this issue, we decided not to use complete datasets, but instead compile our own datasets directly from Billboard and Spotify. In this way, we could make sure our datasets only contained the metadata that were relevant for our project. We also made sure to only select the audio features which values were in the same range, and did not contain any null values. The only thing we had to make sure of was that there were no duplicates in the dataset, since each datapoint should be unique for the classification. but instead compile our own datasets directly from Billboard and Spotify. In this way, we could make sure our datasets only contained the metadata that were relevant for our project. We also made sure to only select the audio features which values were in the same range, and did not contain any null values. The only thing we had to make sure of was that there were no duplicates in the dataset, since each datapoint should be unique for the classification.

The audio features of hit songs differ over the decades. We think this is an appropriate thing to keep in mind when we are selecting what data to be used for model building. To visualize this, we created diagrams in which we plotted how the different features for song tracks has changed over time. The data is based on the Billboard Hot 100 tracks from 1951 until today, and the audio features are fetched from the Spotify API.

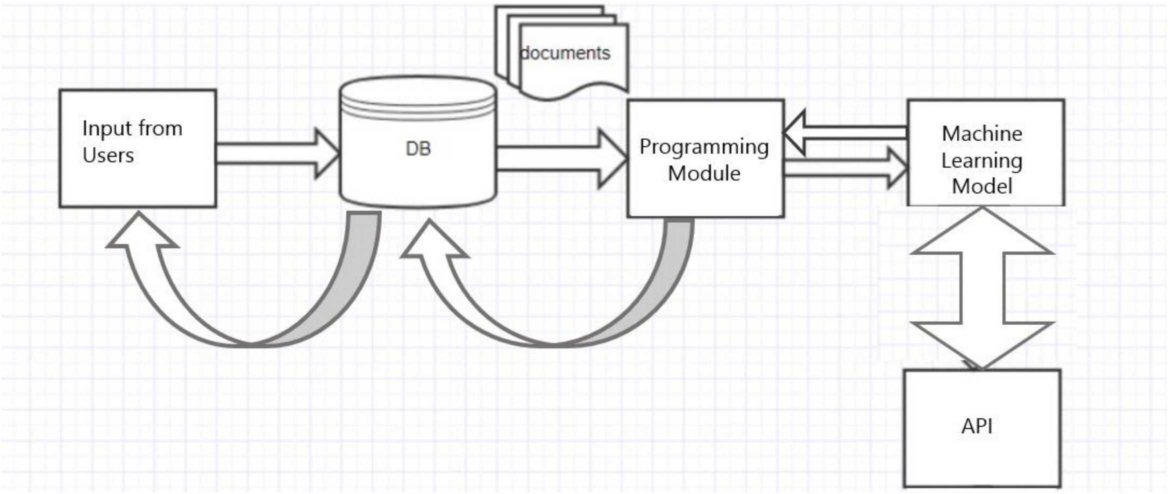


Figure 7: Hit Song Prediction

5.5 Architectural Design

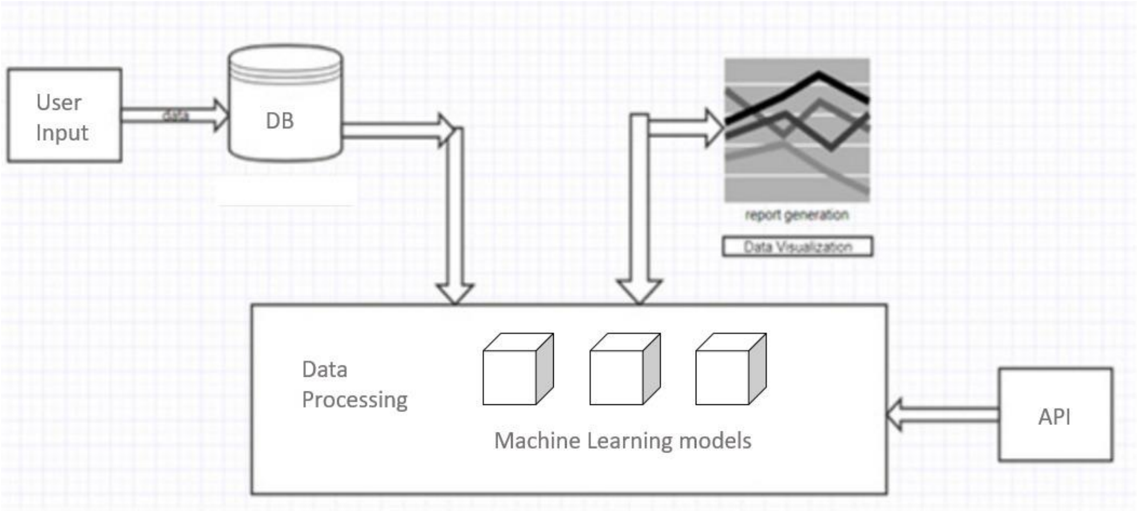


Figure 9: Architecture Diagram

[1] User Input

The user input provides an interface to interact with the application. The user input is connected to the database which also provides the task of validation of credentials of the user, along with the input given to the user.

[2] Database:

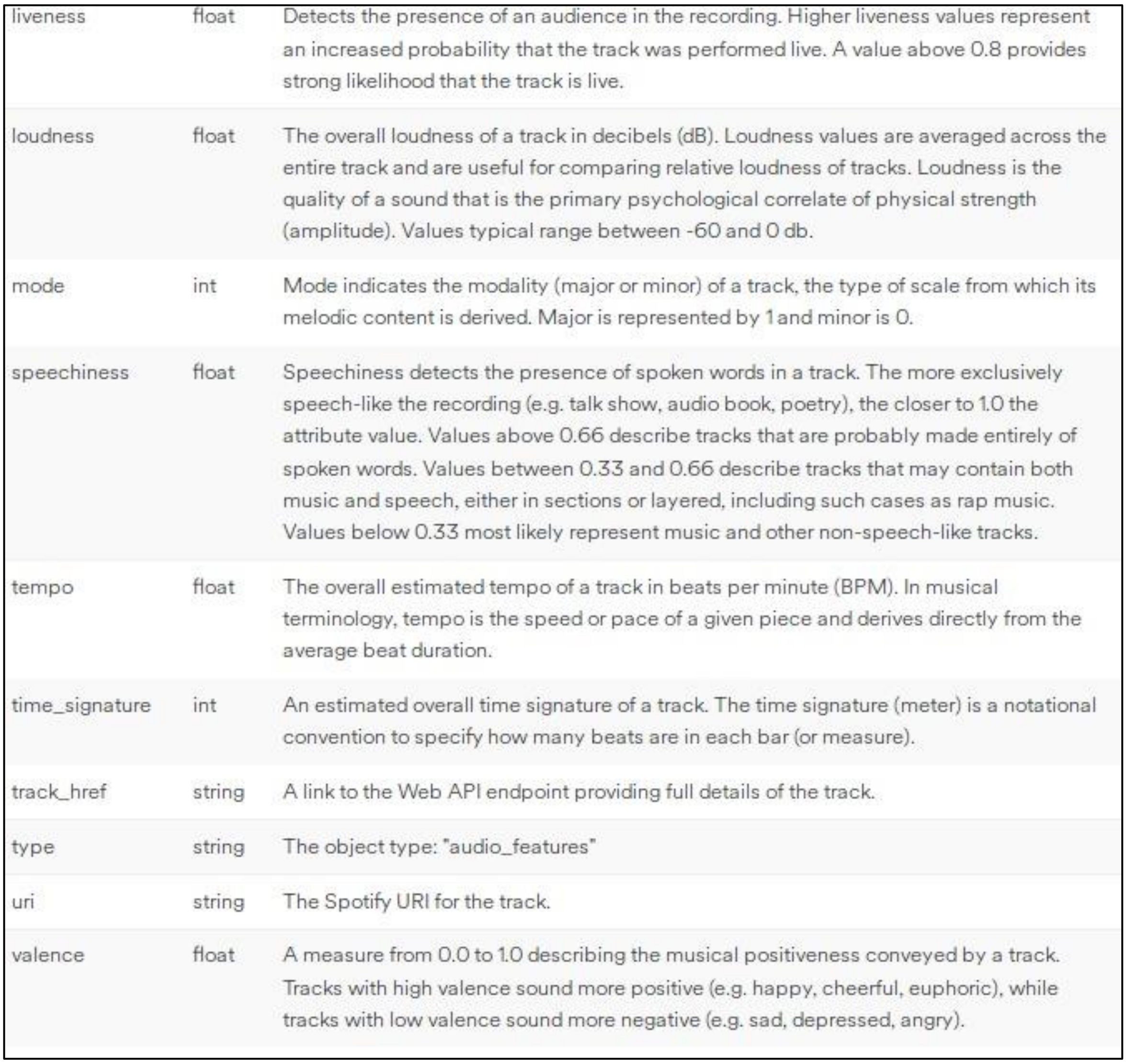
The database is used for managing the data entered by the user and also, to manage user specific data. The results obtained from the machine learning models are also fed into the database, so as to fetch the results from the database to the user, along with the use of these results for future trend analysis.

[3] Data Processing:

The data processing module has different machine learning models which work on the data that is fetched by the API. The accuracy of the different models is compared and one with the highest accuracy is used.

[3] API:

The API is used to fetch the data that is used by machine learning algorithms for classification purposes. In our project, we have used the API provided by Spotify. The description of the audio features provided by Spotify are:



liveness	float	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
loudness	float	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.
mode	int	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
speechiness	float	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
tempo	float	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
time_signature	int	An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).
track_href	string	A link to the Web API endpoint providing full details of the track.
type	string	The object type: "audio_features"
uri	string	The Spotify URI for the track.
valence	float	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

5.6 Component Design

The purpose of the component diagram can be summarized as

- Visualize the components of a system.
- Construct executables by using forward and reverse engineering.
- Describe the organization and relationships of the components.

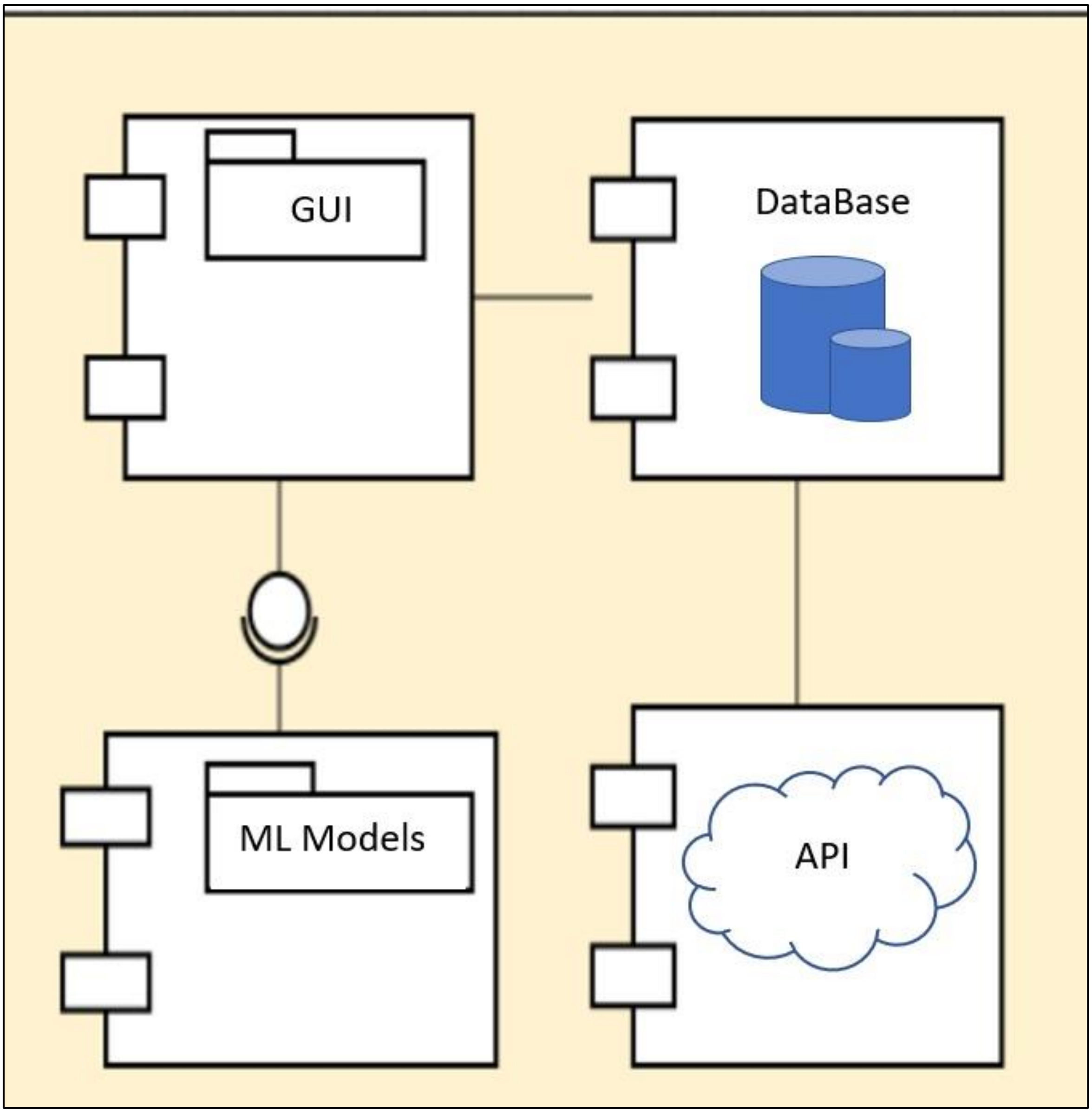


Figure 10: Component Diagram

Component diagrams are used during the implementation phase of an application. However, it is prepared well in advance to visualize the implementation details.

6 SUMMARY AND CONCLUSION

6.1 Conclusion

Before going into any results, it should be taken into consideration that it makes no sense to use a general classification ‘accuracy’ here, because the classes are not balanced (see Figure 1). If you want to use accuracy, it should be class specific. This is a common mistake, but very important to keep in mind. We therefore use Receiver Operator Curve (ROC), Area Under the Curve (AUC) and Confusion Matrices to properly evaluate the models. This is a common mistake, but very important to keep in mind. We therefore use Receiver Operator Curve (ROC), Area Under the Curve (AUC) and Confusion Matrices to properly evaluate the models. After initial analysis, it was found out that random forest model, performed better than the other models. But, since the parameter selection can vary, therefore model selection becomes imperious. Therefore, the same dataset is trained on different datasets and the output is predicted. The model with the highest accuracy is taken into consideration.

References

[1] "Herremans, D. & Bergmans, T."

Hit song prediction on early adopter data and audio features."ISMIR 2017".

[2] "Yang, L.-C., Chou, S.-Y., Liu, J.-Y., Yang, Y.-H., & Chen, Y.-A". "Revisiting the problem of audio-based hit song prediction using convolutional neural networks IEEE International Conference on Acoustics, Speech and Signal Processing "ICASSP 2017".

[3] "Dorien Herremansa, David Martens and Kenneth S¨orensena" Dance hit songs prediction " JIMR 2014".

[4] "Holly Silk, Raul Santos-Rodriguez, Cedric Mesnage, Tijl De Bie" DATA SCIENCE FOR THE DETECTION OF EMERGING MUSIC STYLES (ISMIR 2016).

[5] "Vinitha S, Sweetlin S, Vinusha H and Sajini S"DISEASE PREDICTION USING MACHINE LEARNING OVER BIG DATA (CSEIJ 2018).

[6] "Dhwaani Parikh ,Vineet Menon" Machine Learning Applied to Cervical Cancer Data

(MECS 2019).

[7] "R Dhanaraj, B Logan" Automatic hit songs prediction (GS 2017).

[8] "Sebastian Raschka" Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning,University of Wisconsin–Madison Department of Statistics(2018).

[9] "Rahul Chourasiya1,Vaibhav Patel2, Anurag Shrivastava3" CLASSIFICATION OF CYBER ATTACK USING MACHINE LEARNING TECHNIQUE AT MICROSOFT AZURE CLOUD (IRJEAS 2018). [4] "Holly Silk, Raul Santos-Rodriguez, Cedric Mesnage, Tijl De Bie" DATA SCIENCE FOR THE DETECTION OF EMERGING MUSIC STYLES (ISMIR 2016).

[10] "MINNA REIMAN, PHILIPPA ÖRNELL" Predicting Hit Songs with Machine Learning School of Electrical Engineering and Computer Science(2018)