Jonathan Henley

Professor Shamir

Senior Project 2

4/21/2017

Smart Raspi MP3 player

Abstract

Many people currently use a streaming music service or their phone to listen to music. The problem is that in order to use a streaming service, you first must have a device to use those services, and an internet connection. Using a phone to listen to music can take up storage space, and waste battery life. Using a Raspberry pi CPU and external touchscreen, I have developed a smart MP3 player in python that can play MP3 files that are stored on the raspberry pi. Which allows the user to not have to rely on their phone or other streaming services to listen to music. The player is enhanced by smart capabilities, so that the player learns the music preferences of its user based on its spacious, temporal, and physiological characteristics, and can choose songs and artists that are best fit to the specific preferences of the user. The smart features are based on machine learning using Weka technology, allowing the player to adjust and become more responsive to individual music preferences patterns. This eliminates the need for the user to create playlists or choose specific songs.

Introduction

The most popular way that people listen to music is by using a streaming service. Spotify(Stutz), and Pandora(Boulter and Beaupre) are some the biggest streaming services. In

order to use these services, an internet connection must be used. This means that data must be used, and data cost money. If you do not have an internet connection, then you are helpless when it comes to streaming music. Another issue with these services is that a device, typically a computer or a phone must be used in order to use their services. By doing so, will consume the phones battery life which could be used for other purposes or emergencies that could come up. One issue with Spotify is that to be able to select which songs you want to play, a subscription is required, which costs money, every month that it is used. With Pandora, a user creates channels or stations to listen to music. This is done by selecting different artists, then Pandora's algorithm will play songs from similar artists. The problem with Pandora is that it is not possible to request specific songs to play. While the IPod is currently one the most popular MP3 devices on the market, it held 78% of the music player market in 2011(Macale), they can be upwards of two or three hundred dollars to buy. An IPod comes with much more than normal MP3 capabilities, which is not always needed when someone wants to just listen to music. Physical music devices like the IPod used to be the most popular form of listening to music, but since 2008, IPod sales

have drastically decreased, as shown in the graph below,

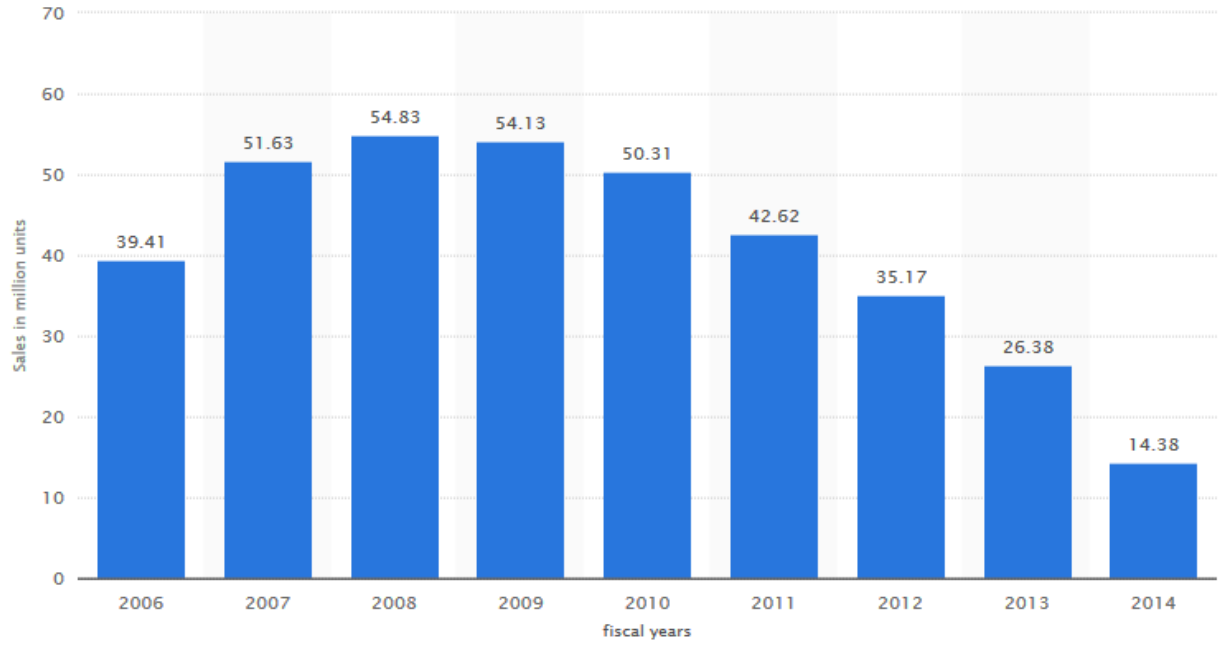

Fig A. IPod sales from 2006 to 2014 in million units.

The more streaming services gained popularity. The less popular physical devices like the IPod became.

There are several musical type products created for the Raspberry Pi that almost fix the issue of a non-streaming music player. One such product is the PiMusicBox(Wilson, 2016). It is software that can be used with a Raspberry Pi that can use multiple different music programs, like Pandora and Spotify, or it can use the local storage of the Raspberry Pi to play music from. The issue is that it still either has to use a streaming service, or a separate device to connect to the Raspberry Pi to play and select the songs. While it is also possible to play songs from a phones local storage, this will use up the phones battery. Another problem with Spotify or Pandora, is that the user still needs to set up what kind of music they want to listen to. They have to choose what kind of songs or genre style they wish to hear, or with Spotify they still need to

create their own playlists unless they want random songs. There needs to be a simple, inexpensive solution to playing customizable music without your phone or a streaming service. That will allow the user to effortlessly listen to songs specifically tailored to them, without intervention.

Hardware

To create the Smart Raspi MP3 player, a Raspberry Pi Model 3+ was selected, and a 5-inch touch screen is used to control the application. A Raspberry Pi is essentially a small Linux computer that can do most things a regular computer can do. It has four USB inputs to connect keyboards and monitors to, an Ethernet port, camera capabilities, and audio to. It has a relatively low cost, and there is substantial documentation to how to use it. The Raspberry Pi requires a 5V, 2A supply for power in order to turn on, and the touch screen requires a 5-12V supply. The Touch screen can connect to the Raspberry Pi either by a VGA cable or a HDMI cable. Several modules were used in order to capture data from the user. Those include a DHT22 temperature senor that reads the surrounding temperature of the module. A MTK3339 GPS board to get the users longitude and latitude location, and a ADXL 345 accelerometer in order to capture motion of the player. If the device is played indoors it is possible that the GPS can have trouble getting a connection for the location. If the device cannot get a fixed location signal then the program will read the location as 0. A signal amplifier was used in order to get better location reading.

Implementation

To create the music player, the chosen language to use was Python because of its simplicity and the amount of documentation that is available for it. Within Raspberry Pi there are several libraries that are used extensively. The first of which is the Pygame

library(pygame.mixer, 2016), specifically the mixer class of the library. This library is actually used for creating games within the Python language. It has a large amount of functions that handle sounds, which is used for playing, stopping, and pausing the song files. The other library that is used heavily within the project is tkinter(Shipman, 2015). Tkinter is used to create the GUI of the application, which stands for a Graphical User Interface of the touchscreen. There are several other libraries that could be used to create the GUI. Qt Designer for instance is a nice tool that is a drag and drop style integrated development environment(IDE) that can be used within python. It is not as straight forward as using tkinter, and requires software installation, whereas tkinter is already included in the Raspberry Pi and ready to go. Along with the libraries used for the sensors, temperature(Dicola, 2014), accelerometer(Dicola, 2016), GPS (Martijn, 2016). The main library that was used for the smart capabilities of the device was the Weka Wrapper(Reutemann, 2014). This wrapper is used in the application to execute Weka functionality in the Python language for the machine learning aspect.

Instead of streaming the music and needing a connection to the device, all the music files are stored locally on the Raspberry Pi. There is an 8G mini SD card that attaches to the SD card that contains the operating system, the program, and all the song files for the project. This allows the user the ability to swap out cards and share music files or playlists with other people. Or the option to connecting to the device through their laptop and downloading songs to the device. Now the user is not tied down by having an internet connection to play their music, as long as they have the files locally on the Raspberry Pi.

How to Operate

The main operating screen that the user will spend most of their time in is shown in Figure B. This screens contains much of the functionality of the application. Along the top left

side of the screen there are four buttons labeled “Up”, “Down”, “Left” and “Right”. These buttons are used to traverse between the music folders. The “Right” button will open the contents of the selected folder shown in the Music List box. The “Left” button will open the previous folders content that was shown before the “Right” button was pressed. The Down button is used to increment the index of the selection in the list box down by one. The Up button is used to decrement the index of the selection in the list box up by one. Below the navigation button is the volume control. The default volume is set to zero, on power up of the device. The user can slide the bar towards the right and left to increase and decrease the volume respectively. The max volume output is set at the value of one when the bar is slid to the right all the way.

At the top of the screen the device shows what current artist and song is being played. In the middle of the screen below the artist and song display is a list box that shows the music folders and songs. The Library and Playlists folder are initially shown as the first items in the list. The Library folder contains every artist and their songs within the folder. The user’s created playlists are shown under the Playlists folder. Below the list box is the pause, and play buttons. The pause button will pause a song that is currently being played. When the button is pressed again when a song is paused, the song will resume playing. The Play button starts playing the song that is currently selected in the list box. It also will add the played song to a database that stores the songs played by the user in the past hour. After an hour has past the database is then reset. On the bottom of the screen there is a button called “Get me a song!”. This button will get a predicted song based on the user’s data when pressed. It can take several seconds in order to start playing the song.

On the right side of the screen are several buttons. Starting from the top is the Add Playlist, Delete, Add To, Song Up, and Song Down. The Add Playlist button is used to create a

playlist. The button will bring up a screen that will prompt the user to enter in the name of the playlist they want to create, as shown in Figure C. The user can exit out of the screen without creating a playlist by hitting the Cancel button, or create a playlist that will be stored under the Playlists folder by hitting the Okay button. The Delete button will remove the and delete a folder selected, or it will delete a song that is selected from a playlist or the music library. The Add To button adds a song that is currently selected by the user to a created playlist. The screen shown in Figure D. is opened to select the playlist that the user wants to add the selected song to. The Up and Down buttons are used to select which playlist, and the Add button will add the song to the playlist and return the user to the main screen. The user can change the order of the songs within a playlist, by choosing the playlist they wish to edit and pressing the Song Up and Song Down button to change the order.

Machine Learning Capability

The machine learning aspect of this device uses Weka functions in order to make its prediction of songs. When a user selects to play a song manually, the application stores the song, artist, day, time of day, the current user's location, movement, and the temperature. That data is then associated to the song and is stored into a CSV database file on the device. This database file is then converted to a ARFF file format for Weka to use. When the user wants to play a predicted song, then the user presses the "Get me a song!" button. The program then gets the users data similar to how it is collected when a song is selected manually, except the song and artist is not known. The application builds a classifier using a machine learning algorithm with Weka with the data in the database file in order to create a trained model. The program then runs the user data with the trained model in order to get a predicted song.

Once the device gets a predicted song, it will then check to see if the predicted song has been played by the user in the past hour by checking the hour database. If it has then a random song by the predicted artist will then be chosen to play. If the predicted song has not been played in the past hour the predicted song will begin to play.

Weka Testing and Results

In order to determine the best algorithm to use in order to have the most accurate song prediction several algorithms were tested using the Weka GUI application on Windows. The following algorithms were tested; J48, Decision Table, Random Forest, Random Tree, Bayes Network, LMT and the Hoeffding Tree. To test the algorithms a sample of songs had been played at different days, times, two different locations, when the device while at standstill and when it was moving.

The first test that was preformed consisted of 150 songs that were played and 93 of those songs were distinct. The results of this test can be seen in Figure E. With this limited amount of played songs, the highest accuracy of the predictions across all the algorithms were only 6.6% Further testing was conducted with a sample of 450 played songs with 117 that were distinct as shown in Figure F. Here it can be seen that the predictions have become much more accurate with the highest percentage being 76.2% from the Random Forest and Random Tree algorithm. Figure G. shows the last test that was performed in order to find the most accurate algorithm. In this test 918 songs were played with 117 being distinct again. The most accurate percentage is 88.67% which belongs to both the Random Forest and Random Tree. Based upon the three tests the Random Forest and Random Tree performed the best and gave the same results. Therefore, the Random Forest algorithm was chosen in order to predict a song for the user. The one downside that can be seen from these test results is that until the user plays at least a couple

hundred songs, the device may not give a very accurate prediction for a song. On the other side, if the user continues to play music the device can become very accurate in its predictions.

Application Screens

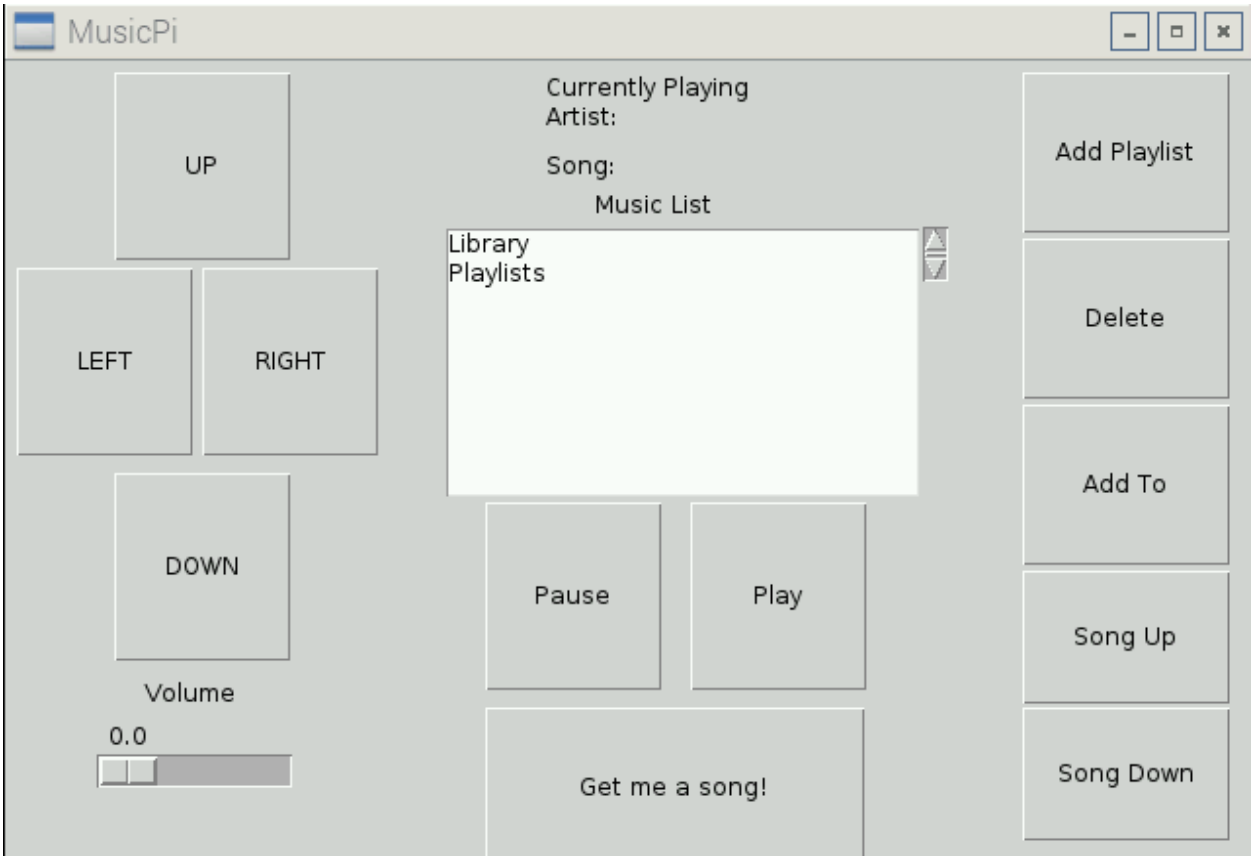

Figure B. Main screen of mp3 player

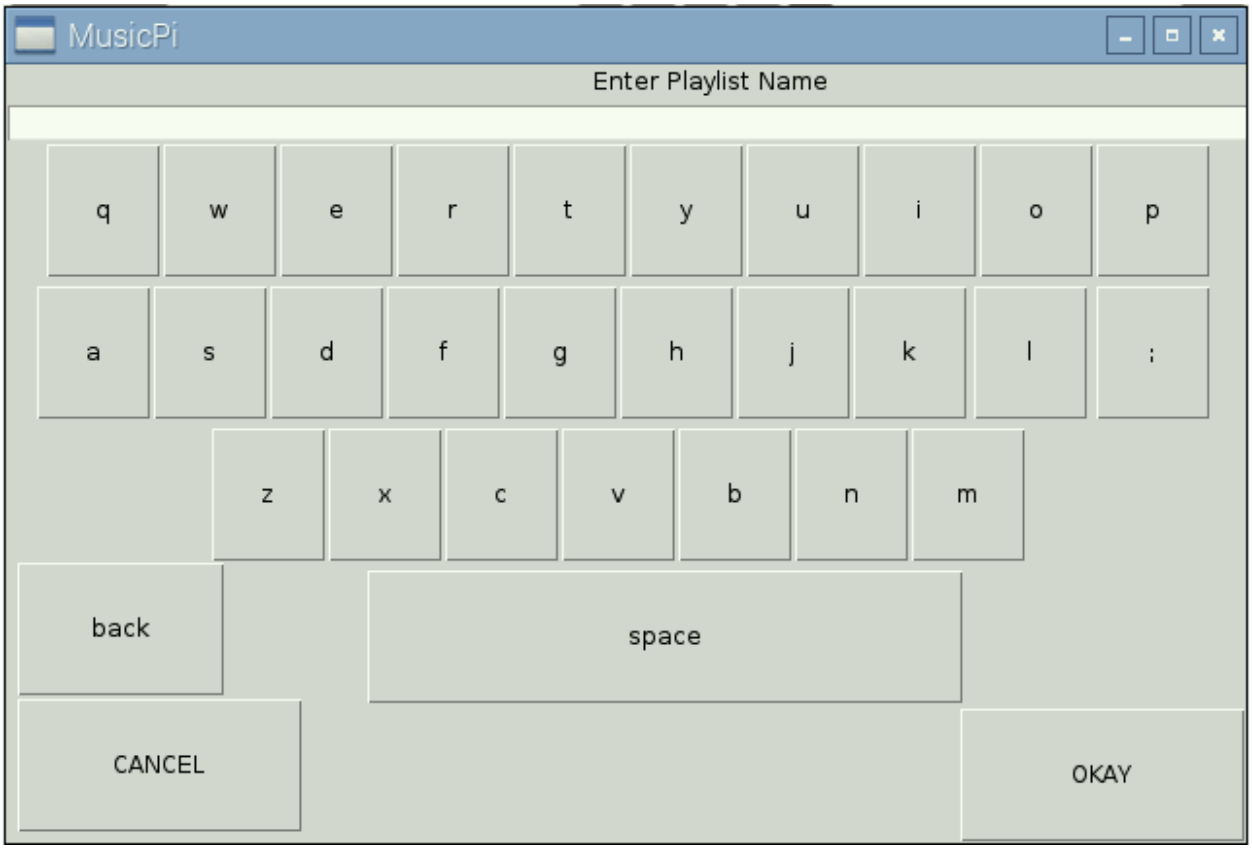

Figure C. Keyboard input for adding a playlist name.

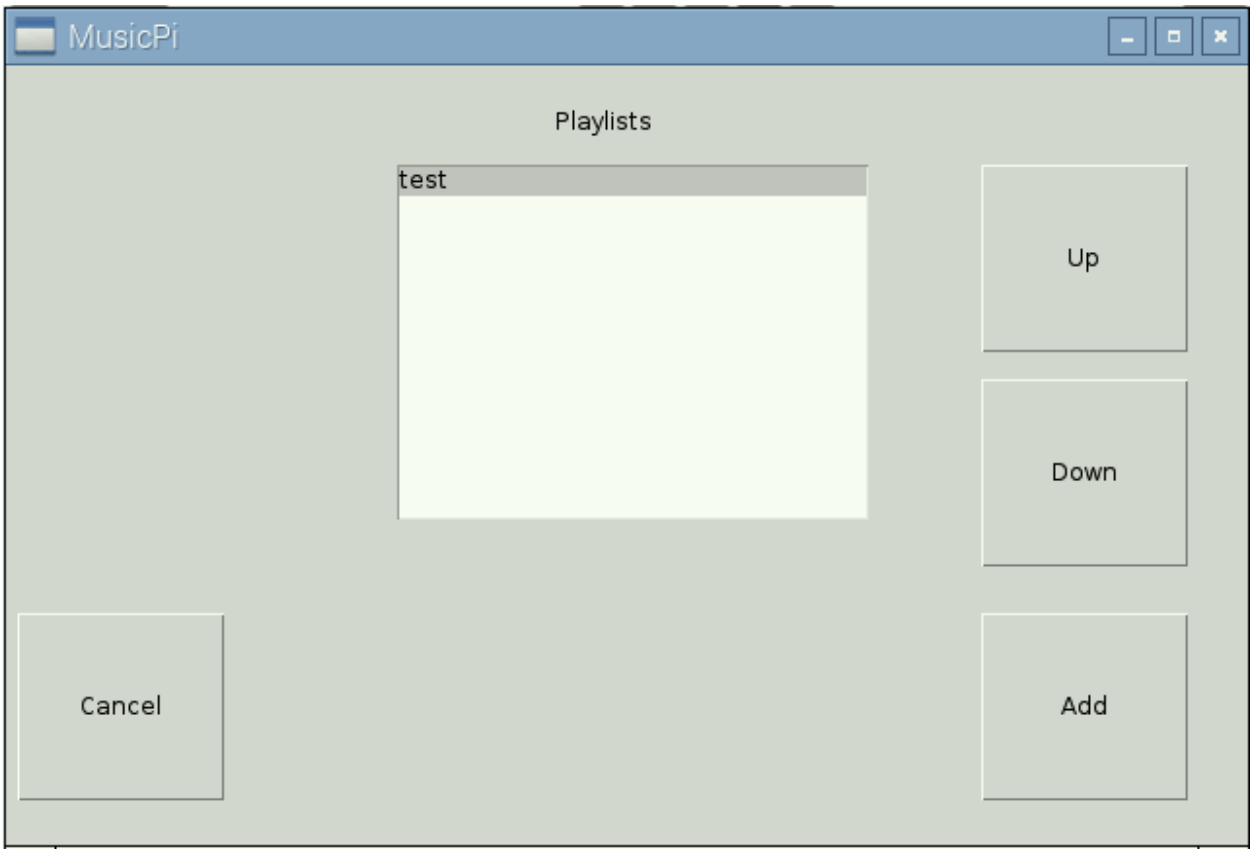

Figure D. Playlist selection for song

Weka Prediction Graphs

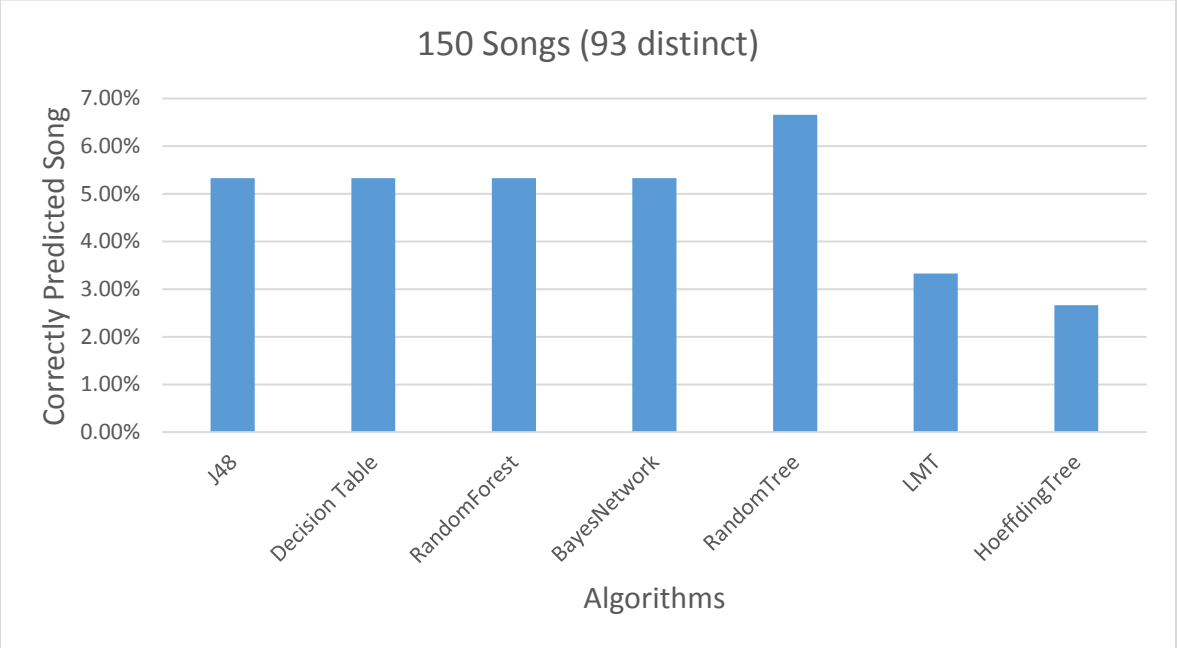

Figure E. Correct song prediction percentage of algorithms for 150 songs

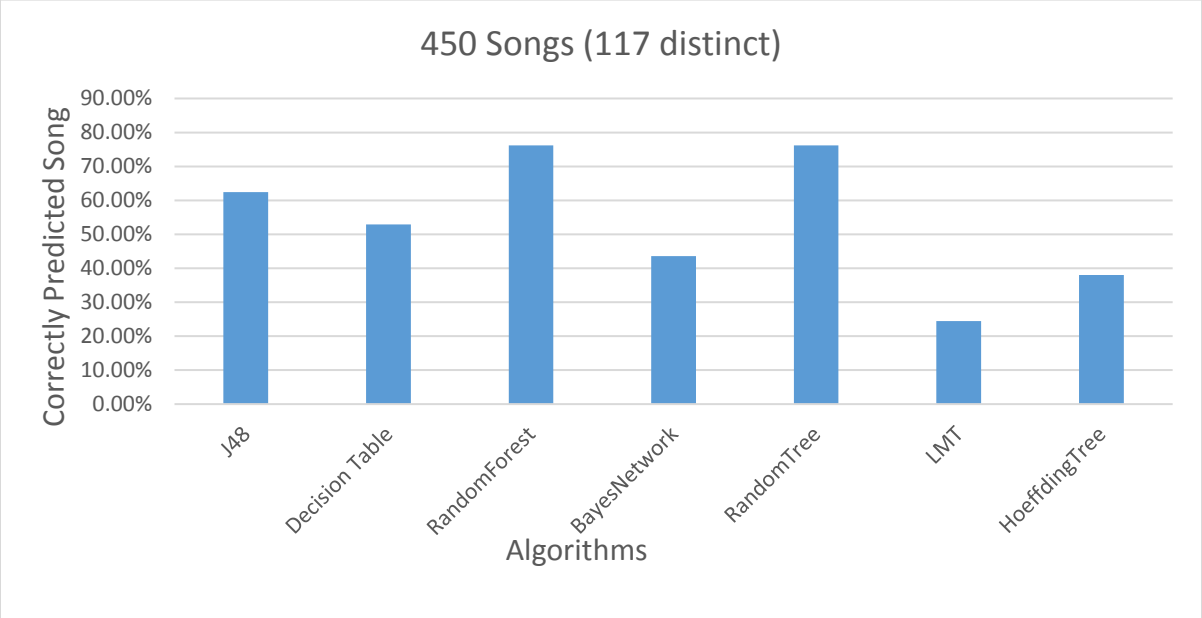

Figure F. Correct song prediction percentage of algorithms for 450 songs

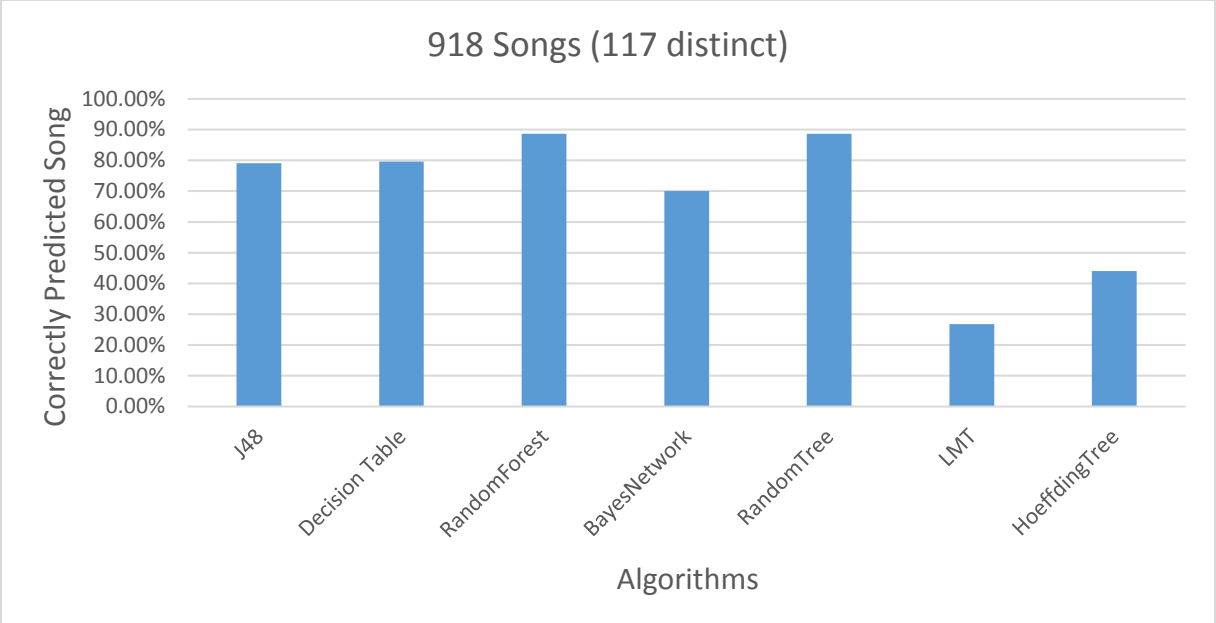

Figure G. Correct song prediction percentage of algorithms for 918 songs

Conclusion

The Smart Raspi provides the user with a simple way to listen to their music, while at the same time giving them a music experience unlike any other streaming service or device through the use of the smart capabilities of the device. The more songs the user plays the better the device gets. Eventually the user will no longer have to worry about what kind of songs they want to listen too, or playlists to create, because the device will play them a song based on the user's data collected. Since the device uses the Raspberry Pi technology, the player can be embedded in to any machine being controlled by a Raspberry Pi if the program is loaded on it, and if there are music files on the device. A few examples of applications that could use embedded systems are thermostats, printers, calculators, refrigerator. Any user could be using any of those machines that uses an embedded system and enjoy their music at the same time. The MP3 really shines in its ability to be portable to many applications and the unique tailored experience of each device to its user.

Works Cited

pygame.mixer. Pygame, Web. 9 July 2016. <http://www.pygame.org/docs/ref/mixer.html>.

Global Apple iPod sales from 2006 to 2014 (in million units) . Statista, Web. 9 July 2016. <http://www.statista.com/statistics/276307/global-apple-ipod-sales-since-fiscal-year-2006/>.

Boulter, Jeffrey R., and Todd M. Beaupre. "Internet radio and broadcast method." U.S. Patent No. 7,711,838. 4 May 2010.

Braam, Martijn. GPSD-py3. March 6th, 2016. Github Repository https://github.com/MartijnBraam/gpsd-py3

Dicola, Tony. Adafruit Python ADXL345. April 19th, 2016. Github Repository https://github.com/adafruit/Adafruit_Python_ADXL345

Dicola, Tony. Adafruit Python DHT. June 1st, 2014. Github Repository https://github.com/adafruit/Adafruit_Python_DHT

Macale, Sherilynn. 4th October,2011. "Apple has sold 300M iPods, currently holds 78% of the music player market" <http://thenextweb.com/apple/2011/10/04/apple-has-sold-300m-ipods-currently-holds-78-of-the-music-player-market/>.

Shipman, John W. *Tkinter 8.5 reference: a GUI for Python*. New Mexico Tech, 31 Dec. 2013. Web. 9 July 2016. <http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html>.

Stutz, Colin. “Spotify Tops Pandora As World’s Most Popular Music Streaming App” 1st December 2015. http://www.billboard.com/articles/news/6784774/spotify-pandora-most-popular-music-streaming-app-worldwide

Reutemann, Peter. Python Weka Wrapper. March 30th 2014. Github Repository https://github.com/fracpete/python-weka-wrapper

Wilson, Adam. *Pi MusicBox*. Ed. Web. 9 July 2016. <http://www.pimusicbox.com/ >.

I have neither given nor received unauthorized aid in completing this work, nor have I presented someone else’s work as my own.