Department of Computer Science
2018/2019

One million Songs Dataset Project

Project Specification and Design

Big Data Group Project COMP530

Supervisor:
Dr. Michael Bane
Dr. Andrew Gargett (Hartree Centre)

By:
Al Jahdhami, Ismail - 201281702
Arinto, Carlos - 201386292
Yuefeng, Liang - 201350099

1. Introduction

Dealing with very high volumes of data is a recurring problem and has become a concern to many organizations and entities around the world. It varies from textual data to audio and visual and data can either be acquired through batch (already present) or streaming (real-time transmission) processes.

Notwithstanding, the musical field is a contributor to this emerging information factor, producing increasingly more audio files which are spread all over internet [1].

This project, titled "One Million Songs Dataset", intends to perform descriptive analysis on songs' metadata. Throughout this analysis we aim to answer research questions such as "Are there any patterns in how features' values behave throughout the years?" and "Can popularity be predicted in order to tell which characteristics make popular songs? ".

In order to answered these aims, some of the main challenges ahead (such a building a prediction model) will lie on cleaning the data. This task will be done by a step-by-step process (first the data format is converted, followed by ignoring all unused fields, etc..).

This project will be developed in Python coding language and run on Dawson, an HPC provided by the Hartree Centre, in order to increase its performance.

In order to work with the data, we will be using Pyspark, the Apache Spark Python library. This framework is capable of using the available distributed computing systems, which, combined with Dawson, will speed up the performance of the code.

In this document we will mention the main objectives and outcomes to be expected of his project, along with a description of the all the resources (hardware, software and files) to be used, along the risk assessment and background research done so far. On the design part of the project, we will provide a system overview, the procedures that will allow us to meet all the objectives and how the system will evaluated, ensuring it gives the expected outcomes. Lastly a Gantt chart will display all the milestones and tasks planned for the development of the project.

Also, this document represents a revised version of the initial project specification and design, and any changes made were done due to the tests run and advice taken from our supervisor as well as aspects learned from literature throughout the development of the project

2. Aim

The first aim of the project is to discover whether patterns in the data that could be of interest to a third party planning to invest in the musical industry can be found with the help. These patterns would be focused on behaviours (increasing, decreasing or constant values) that were consistent throughout the years (using line charts) and correlations between features (using correlation heatmaps and scatter plots), depicting how one's values may rise if the other rises as well, or in case of inverse correlation, gets lower.

As a second aim, we plan on building a system capable of using the song popularity score in this dataset, to draw conclusions about the data which will be valuable for musical-industry related workers and investors. In order to achieve this we will develop prediction models, compare and assess their accuracy and, provided they perform well, extract the feature importances to analyse which ones the investors should focus their attention on.

In the current state of the art, most projects who have worked on trend analysis of the "One Million Songs Dataset" have only investigated basic aspects about the exploration of the data (such as plotting number of songs per year)[2] predictions that are based on the use of all features [3]). We will go a step further on this analysis, achieving conclusions, later on specified in the *Objectives,* only obtainable by observing factors such as the fluctuation of values and selecting only the most relevant features to reduce the amount of information required while maintaining the accuracy of our predictions.

3. Objectives

3.1 Essential Objectives

a) A clean dataset that can with only the relevant columns, rows with no missing values and no symbols/noise;
b) Research on previous literature and use features that other people have considered worth analysing;
c) Determine whether popularity is a useful feature to model as well as the reasons why/why not;

d) Extract the most important features and determine if they alone can be used to predict popularity without any noticeable increases in the error score.
e) Based on the outcomes of both previous objectives, carry out our own analysis of patterns in the data (whether it means trying to replicate the discoveries made or attempt to find something new);
f) Build a big data pipeline to implement objectives 1, 2 and 3. This pipeline will need to perform appropriately with the scaling of the data.

3.2 Desirable Objectives

a) Develop classifier models and conclude if their performance would allow them to be used to predict the year.

4. Expected Outcomes (Deliverables)

4.1.Description of anticipated software

Essential:

- Data converted from an HDF5 file into a binary Parquet file;
- Cleaning of the data by checking if it is on the right format, excluding any irrelevant information and removing rows missing values
- Pyspark dataframe containing all the data in loaded from the binary file, ready to be used for analysis.
- Line charts, scatter plots, correlation heatmaps, density graphs and bar plots originated from the code, displaying the information gathered from the descriptive analysis.
- Various regression models ready to be compared on their performance predicting any desired feature.

Desirable:

1- Various classifier models to predict numeric variables than can be interpreted as discrete (such as the year) and their accuracy between each other and the regression models.

Description of anticipated documentation:

Essential:

1 - Project specification and design of the project (this document).

2 - User manual: directed at the visualisation of the conclusions taken regarding the mentioned *Objectives.*

Description of anticipated artefacts:

Essential:

1 - Diagrams that clearly summarize how the music tastes have changed over the years.

5. Resource requirements

Dataset:

The main resource of the project is one million songs' metadata, MSD, which is provided through a website from The Echo Nest [4]. We will need to download the subset of the whole data to our local machine to test some codes. In addition, we also need to programmatically query the data. As there is no genre labels in the main dataset, we will need to link the main dataset with another small dataset provided in a website, tagtraum industries, whose dataset has the song genre labels[5]. We will use PyTables package to manipulate data. PyTables is a package for managing hierarchical datasets and designed to efficiently and easily cope with extremely large amounts of data [6]. Then, we will convert the dataset to csv format. In the cleaning part, we will do the feature engineering in artist location column (converting the different ways in which locations were expressed into the corresponding country), so we need a list file included most cities names and countries names around the world.

Hardware:

Due to the size of the dataset which is about 280 GB, it is required to process the data using high-performance computing resources that are available at Hartree, making use of multiple nodes to speed up the large-scale analysis of the process. For this project, Hartree provides access to Dawson which has the following Hardware Specification on the cluster:

- 3 Name Node servers with 16 cores and 128GB RAM

- 36 Data Nodes with 24 cores and 64GB RAM

The cluster will provide the required RAM, Hard disk storage and processing power to deal with the task.

Software:

We used the Hadoop Distributed File System (HDFS), which is a distributed file system of Hadoop. HDFS provides the distributed storage capabilities to process large data sets and high throughput access to application data, which is suitable for applications that have large data sets [7]. Instead of using MapReduce, Apache Spark will be used due to its increased performance and usability. Jupyter Notebook will be used as the code programmatically and data visualisation tool. The final product will be available for consulting online, where the results will be gathered for the visualisation according to different planned objectives.

6. Research

Music has had an important role in our culture throughout human history. The music industry may interested in what makes up a "popular" song, which is to say, a song whose value translates directly into revenue. Therefore, being able to analyse the trends or find out some patterns in popular music would provide a significant capability to the industry; a key requirement for this capability to be realised is collecting enough music metadata to carry out the appropriate modelling. The Million Songs dataset more than adequately satisfies this requirement.

The Million Song Dataset (MSD) is an open resource for researchers, which we need to download metadata from. The reason why we choose MSD [4]:

• large-scale, open source

• freely available

• easy to get started

In addition, because there is no song genre labels in the MSD and we want to analysis the correlation between the song popularity and song type, we will associate the main dataset with the another dataset, which has the song genre labels.

6.1. Background literature

Pham et al (2015), found that predicting song popularity is important in keeping businesses competitive within the growing music industry. Their research was based on collecting some features to predict popularity using different classification and regression algorithms. [8]

In addition, Mohamed Nasreldin et al (2018) , from University of Texas, determine whether the popularity of a specific song is predictable. Their study aimed to find answers to three key questions:

- Find out the characteristics for hit songs
- What could be the largest impact on song's success
- Can old songs be used to predict the popularity of new songs

They found the important features are related to Artist information (familiarities, hotness and artist identifications) and technical information like tempo, mode, and loudness.

Otherwise, they found that there are missing values for songs hotness features so they filled by connecting to Top 100 songs website and see if the songs appeared once and they consider it as a hit. They used certain model to train their data and found the Xgboost has the highest accuracy value [9].

We found both literature are useful for our project as they used the features that helped to find and predict the popularity of the songs.

6.2. Get familiarized on how to use the recommended big data technologies

For this project, we need to look at performing analysis on large, disparate dataset. Research has been made into the big data stack to see what technologies will allow us to perform large scale data processing on these resources.

Hadoop + Spark

We need to look at options for processing the data. Firstly, Apache Hadoop is a cluster computing framework that enables distributed processing of large data on disk using a map-reduce model and allowing high-throughput access to the data using the Hadoop distributed file system (HDFS) for file management. In a typical Hadoop task, MapReduce is used for the data processing, however, there are limitations to this as it persists the full dataset to the distributed file system after each MapReduce job, which means this is a very slow process.

Apache Spark also uses the HDFS for data management, but it introduced an in-memory caching and instead passes data directly without writing to persistent storage. Spark also speeds up data processing as it keep a JVM running on each node and it use DAG-based task scheduling mechanism, which is better than MapReduce's iterative execution mechanism [10]. Therefore, we prefer the Apache Spark rather than MapReduce which will be used to process the dataset. To use spark, team need to be trained on this as we are not very familiar on how to use it, so we will get online tutorials on how to use this technology. In our project, we will use spark dataframe in the cleaning data and analysis part.

Jupyter Notebook

As for data visualisation technique, there are various options to perform. We choose to use Jupyter Notebook as our data visualisation tool. The Jupyter notebook combines two components [11]:

A web application: a browser-based tool for interactive authoring of documents which combine explanatory text, mathematics, computations and their rich media output. Notebook documents: a representation of all content visible in the web application, including inputs and

outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.

After starting the Jupyter Notebook serves and input the dataset, we will type code on the interface. In the analysis part, we will do the general trends analysis of songs features, so we will plot time series graphs. While in the correlation analysis part, we will use scatter plot to show two features' correlation.

6.3. local Tools

For this project, we will use Pycharm software to type python code and download the Anaconda to use Jupyter Notebook locally. Thus, our team members who are not familiar with the python need to download these softwares and learn to use them. Also, we need MobaXterm to access Dawson via ssh and then, all our code will be put into Jupyter and run in it.

7. Risk Assessment

For any success of a project it is mandatory to identify the risks that might the project encounter so it can be mitigated earlier.

Lose connection to Hartree Resources:

This could occur if we are unable to connect to Hartree machines due to internet or problem at hartree resources hardware or Software.

Personal Computer Failures:

As we are using our local machines to login to hartree machines we could have an issue with our laptops that couldn't boot or has hardware failure

Preprocessing and cleaning of data:

The data might need more time to be cleaned. we might got lots missing values and it could affect our analysis.

Schedule Risk:

There might be some delay to accomplish any schedule task, which might affect other tasks and delivery of the project

Technical Risk:

This could happen if there is some technical requirement that the team is not aware of that relate to programming or solving some issues in the project

Poor team dynamics:

Miscommunication or any problem with the team member, which may force them to leave the country which might affect the work related.

Changing of Scope:

Change of Hartree representative could bring new ideas and thoughts which might affect the project schedule and resources.

Moreover, we can measure the likelihood of risks, so we can mitigate their consequences that might affect the project performance. Below is the assessment of the risks that were identified:

	Risk Identified	**Risk assessment Risk impact = Likelihood X Consequence**	**Risk Mitigation**
1	Lose connection to Hartree Resources	3 X2=6	Do all the work in the local machines first and do the backup of the code in github
2	Personal Computer Failures	3 X2=6	Try to fix the Laptop and work with other team member and should save the work in github

3	Preprocessing and cleaning of data	4 X2=8	delete the rows which has empty value or try to fill it with mean if it is numerical value
4	Hardware and software failure	3 X 4=12	Implement daily Backup and replicate of the code in the local pc and to google drive
5	Schedule risk	3 X 3 =9	Adjust the time and try to fit it with another tasks schedule
6	Technical Risk	3 X 3 =9	Seek for help from Hartree or who has experience on that
7	Team poor Dynamics	2 X 3 =6	Understand of team's behaviors and issues
8	Changing of scope	2 X 4=8	Be prepared to defend against any changes that could affect the project

8. Design:

8.1. System Overview

Considering that the full "One Million Song Dataset" contains around 280Gb of data, it would become too computationally expensive to perform any task on it. On the first stage of the project, personal computers will be the main tool to work on a much smaller subset (1.9Gb). When the complete dataset is used, Hartree Center's HPC's will provide the computational power necessary to operate the data within reasonable timestamps.

The data will be stored firstly stored in a binary file in the group's local machines and later one migrated to Dawson's HDFS.

After the preprocessing the cleaning stages, the data will be plotted in different ways using Pyspark methods and subjected to training and testing phases in the existent prediction models.

Lastly, the outcomes of this project will be mainly focused on ensuring that the full data pipeline (extraction, loading and transformation of the data) was built, whether the detection of patterns trough descriptive analysis is possible or not, how well song popularity can be predicted and how to reduce the necessary amount of features required to accurately do so.

8.2. Data Sources

In this project we will use the million song dataset provided by by The Echo Nest. We will, however, start with the subset(1%, 1.8 GB compressed)and do our full cleaning analysis and predictions, and only then moving into the full dataset for scalability testing.

The data was provided in HDF5 format and contains information in the form of metadata related to songs and their release date, artist, which are in each file.

Each song in the dataset consists of 54 features categorized by audio analysis(tempo, duration, mode, loudness, key, time signature, section start), artist information (artist familiarity, artist popularity, artist name, artist location), and song related features (releases,

title, year, song hotness). It's worth noting, however, that not all of these contained valid (column with only 0's) or useful (ID's) information.

This source was sufficient to produce our analysis and prediction outcomes.

Also, in order for us to do more analysis on the genre type we must download an other subset we will need to link the main dataset with another dataset provided in a website, tagtraum industries, which has the genre type field, then link it with the original subset by using TrackID field and generate a new CSV file and this could be done using oracle database to make this link.

The dataset will be loaded into Dawson by Hartree engineers where we can have access and start working on it. Moreover, the data that is provided is open source which is used for research purpose

8.3 Pre-process Design

8.3.1.Data cleaning and pre-processing

In the sample dataset and subset provided, there are columns such as "danceability" which don't bring any valuable information (as all values are zeroes), rows which contain missing data and values that contain symbols such as punctuation signs and noise such as the existence of a "b' " before some of the song and artist names.

This was dealt using the following procedures:

Column filtering

Not all the features presented relevant values to be taken into account. Some of them only contained zeroes (danceability and energy) while others just contained codes that created only in order to assign an ID to features such as artists, songs, etc…

For that reason, these columns were discarded, reducing the number of columns and therefore the computational time necessary to perform any sort of analysis or prediction or even further cleaning on the dataset.

Missing values (Numeric and Categorical):

This was done by simply removing any row which contained a missing value. This method was preferred to filling the values with the mean/median (for numeric fields) or the word "missing" (for categorical fields) since these will strongly bias the outcomes towards the values used for the filling and since still kept ⅛ of the original size, a sample significant enough to perform the analysis and prediction.

Symbols and noise:

Some values in the data included symbols like "Δ" or random prefixes/suffixes in them such as "b'" in the song name "b' Nothing Else Matters".

In order for these to be removed, a list containing all the known symbols was created, and any character presented on that list would be deleted. As for the noise, the team visually identified the prefix/suffix "words" that were involved in these occurrences and proceeded to delete them as well.

Feature engineering:

There was also an occurrence of a column which presented it's values in many forms, The feature "location" had it's values in the form of countries (ex: Sweden), cities (ex: Barcelona), USA states (ex: Colorado) and country initials (Ex: CA). In order to perform any analysis on this feature, and to test if it could be useful to predict popularity, the team gathered files which contained countries, initials, USA states, etc.. and mapped them to the corresponding country names. This way, once they column value was read, it would successfully be converted a standardized format of representing a country.

8.4. Data Storage and Access

The dataset was stored at Dawson Cluster initially. Then the data will be loaded into HDFS to be analysed by python code and Spark to perform data cleaning and doing analysis on the data.

After processing the data into Hadoop and got the cleaned data as CSV file. We will load the CSV file to oracle database and link it with the subset data which has the genre type and produce a new CSV file that has the genre type column and process it again into HDFS for further analysis. Furthermore, we convert the CSV file to parquet file to make the system perform faster.

Loading the Dataset:

In order to access the dataset which resides in Dawson cluster Python programming language libraries are used. For instance will use the file "HDF5_getters.py", written by Thierry Bertin-Mahieux at Columbia University [4]. This file makes use of the python libraries numpy (Numerical Python) and tables (PyTables/Python Tables), which deal with a hierarchical format such as HDF5. Also, we used matplotlib/pylab for visualize the data in Jupiter. Moreover, data will be transformed to CSV files then be loaded into as HDFS files in Hadoop.

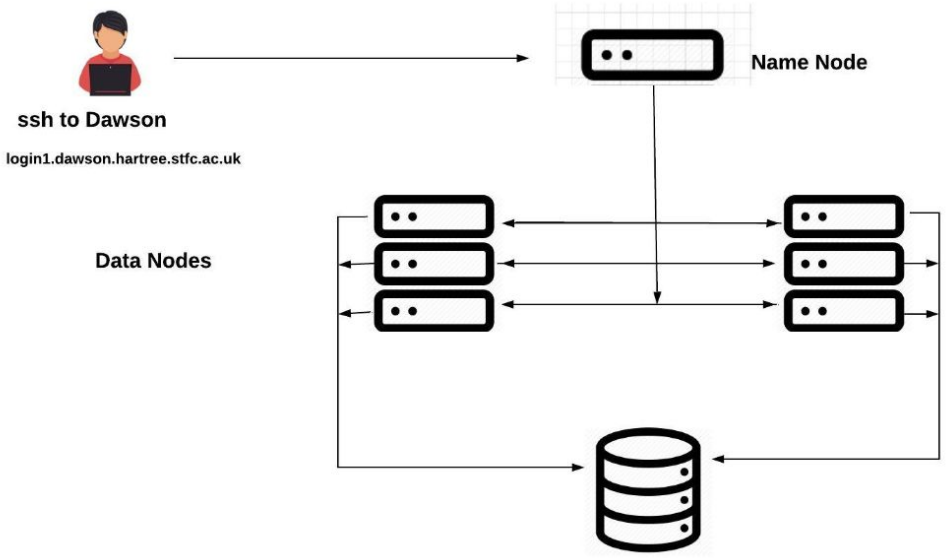


Figure 1: Process on how to access the data

ER Diagram:

A database ER diagram visualizes how entities within a database relate to each other and the attributes of each entity. For this project there was a need to create a database in order to

combine 2 dataset together and could analyze the popularity of genre type in various locations.

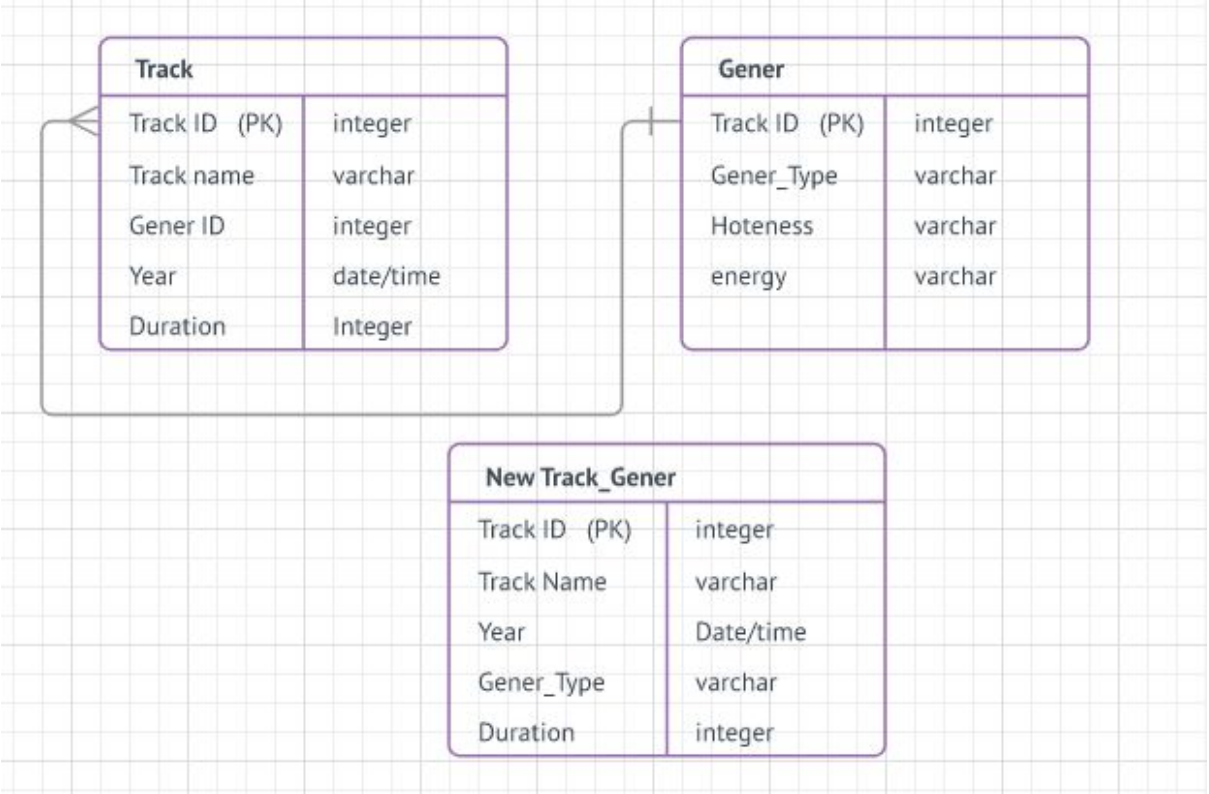


Figure 2 : ER diagram of the database

8.6. Analysis and Performance Score

8.6.1.1 Data cleanse

The process will start developing and using a python algorithm which will convert the data from HDF5 format into CSV so that it can be uploaded into spark dataframe and allow the following cleaning methods. Next, all the unused parameter fields will be removed from the dataset. After that another algorithm will remove any symbols and punctuation from the existing textual fields.

As for the artist location column, we will convert the different ways in which locations were expressed into the corresponding country by using a list file included most cities name and countries name.

Lastly, a final algorithm will remove any data point/instance which contains missing values. The dataset will then be ready to be uploaded and worked on.

8.6.1.2 Prediction Models

To achieve one of the objectives which consisted on determining whether popularity is a useful feature to model, we will try various regression models.

These would allow us to check how accurately "song popularity" could be predicted, as well as extracting the main features that are most related to it.

A range of different regression models will be tested models, including: Random Forest, Decision Tree, Gradient-Boosted and Linear regression. A ratio of 70% of the data for training and 30% for testing will be established for all the models, as it often performs well (information provided by the team's supervisor, coincident with the one taught in our lectures). In order to compare them, the team will look at the "RSME" value of each, which translates into the standard deviation of the prediction errors [12]. This means that the aim will be to get a value of RSME as low as possible.

After selecting the best model and ensuring its efficiency, the team will proceed to the "feature selection". This step will reveal which features contributed the most to the prediction of the "song popularity".

To obtain these results, all the "feature importances" will be extracted from the regressor. Based on those scores, the best features will be selected. The term "best" accounts for all values above 10% (0.1) which is significant considering 13 features will be "fed" to the model.

8.6.1.3 General descriptive analysis of features over time

The initial step for this analysis, will be to create groups that are all numeric values. Then, we will calculate the mean of each group in each year and standard deviation and plot the trend

change of these features into graphs. Finally, we would find out some patterns in these graphs.

8.6.1.4 Genre and Artist Location of most popular songs in the past 20 years

Initially, an algorithm will pick all the songs between 1990 and 2010 as we found the data over these years is about 80% of the sample dataset. It is available to use these data to analysis.

Over that subset, the songs will be rated by popularity, by ordering them regarding the field “hotness”. Then, we define "popular songs" as those with songs popularity score ranking at top 20% of all songs. The reason why we choose 20% is that we found in some literatures, the researchers used this value but did not mention the reason, thus we think this may be an available value [12].

After this, an algorithm will count the number of times each genre and locations appears in this “ popular” subset. Finally, we will plot the correlation between the song popularity and the song genre or artist location in line chart. We want to find out if popular music is related to the type of music or artist location.

8.6.1.5 Features Correlation

Firstly, we will make an algorithm to extract all the numeric values of features and put them into lists respectively. Then, these lists will be reorganize to a pandas dataframe, which is is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns) [13]. We will use a method only in pandas to calculate the correlation matrix of all these features and plot the matrix to a heatmap. Finally, we could find out which features are most related to the popularity and visualize the relationships using a method in Seaborn library [14].

8.6.1.6 - Performance Score

Regarding the general performance of the code, the team will attempt to improve the speed at which the data is read, by converting the HDF5 file into a binary Parquet file instead of a CSV, and comparing the timing results. Something similar will be done for the cleaning of the data. As Pyspark takes advantage of distributed computing, the removal of rows with

missing values, and the process of accessing each value and removing any symbols / noise, will be done using Pyspark methods (on the dataframe) instead of the PyTables library and compare both executions times once more.

As for the prediction models, the RSME value has a range of 1 (as popularity scores are comprised between 0 and 1). Therefore, a value of 0.2 or below will mean that the average distance between the predicted values and the expected prediction was on the top 20% of possible outcomes, and will be seen as a good performance prediction method.

8.8. System Evaluation

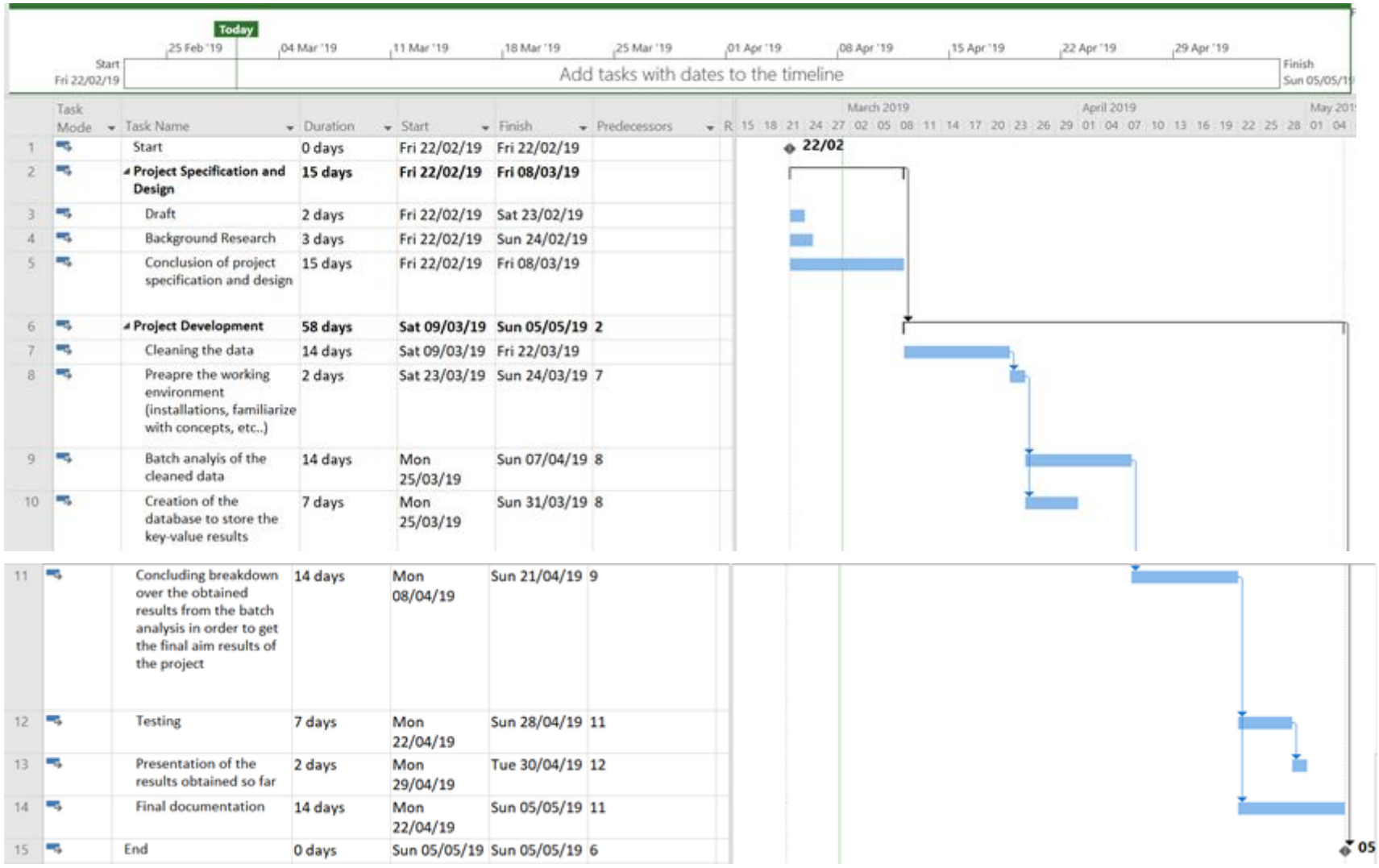
The system evaluation will assess whether or not the system can:

- ➢ Successfully convert HDF5 data into a binary Parquet file .
- ➢ Load all the data from the binary file into a Pyspark dataframe, ready to be used in the code.
- ➢ Produce the plots mentioned so far allowing the the observation and better understanding of the data we working with as well as any visible patterns.
- ➢ Use the models to achieve the prediction error/accuracy of the tested features.
- ➢ Extract the feature importance scores, map them to the corresponding feature name and rank them from highest to lowest.

The system will firstly be evaluated by the members responsible for its development and then shared with the supervisor, module coordinator and any other personnel involved in the project.

Their opinion shall determine if the aim of the project was fully fulfilled, partially fulfilled, or quite not yet satisfied, and should include constructive criticism both on the positive and negative aspects of the achieved results.

9. Plan



	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		Start	0 days	Fri 22/02/19	Fri 22/02/19	
2		**Project Specification and Design**	**15 days**	**Fri 22/02/19**	**Fri 08/03/19**	
3		Draft	2 days	Fri 22/02/19	Sat 23/02/19	
4		Background Research	3 days	Fri 22/02/19	Sun 24/02/19	
5		Conclusion of project specification and design	15 days	Fri 22/02/19	Fri 08/03/19	
6		**Project Development**	**58 days**	**Sat 09/03/19**	**Sun 05/05/19**	2
7		Cleaning the data	14 days	Sat 09/03/19	Fri 22/03/19	
8		Preapre the working environment (installations, familiarize with concepts, etc..)	2 days	Sat 23/03/19	Sun 24/03/19	7
9		Batch analyis of the cleaned data	14 days	Mon 25/03/19	Sun 07/04/19	8
10		Creation of the database to store the key-value results	7 days	Mon 25/03/19	Sun 31/03/19	8
11		Concluding breakdown over the obtained results from the batch analysis in order to get the final aim results of the project	14 days	Mon 08/04/19	Sun 21/04/19	9
12		Testing	7 days	Mon 22/04/19	Sun 28/04/19	11
13		Presentation of the results obtained so far	2 days	Mon 29/04/19	Tue 30/04/19	12
14		Final documentation	14 days	Mon 22/04/19	Sun 05/05/19	11
15		End	0 days	Sun 05/05/19	Sun 05/05/19	6

Milestones:

1. Project Specification and Design
2. Project Development
3. Testing

10. Bibliography

[1] Darren Heitner (2018), Big Data Is Revolutionizing the Music Industry. Here Are the Lessons for Your Business. Available online: https://www.inc.com/darren-heitner/big-data-is-revolutionizing-music-industry-here-are-lessons-for-your-business.html [Accessed: 23/03/2019].

[2] Unspecified author (2017), 'Music_Trends_Analysis', Available online: https://github.com/ys2843/million-song-dataset-analysis/tree/master/Music_Trends_Analysis/output [Accessed: 26/03/2019].

[3] Raul Soutelo (2018), 'Music-genre-classification-with-the-Million-Song-Dataset' Available online: https://github.com/raulsoutelo/Music-genre-classification-with-the-Million-Song-Dataset [Accessed: 26/03/2019].

[4] Millions Songs Dataset: [https://labrosa.ee.columbia.edu/millionsong/], [access on:20 February 2019]

[5] tagtraum genre annotations for the Million Song Dataset: [http://www.tagtraum.com/msd_genre_datasets.html], [access on: 1 March 2019]

[6] Pytables: [https://www.pytables.org/], [access on : 1 March 2019]

[7] HDFS: [https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html], [access on: 24 February 2019]

[8] Pham, J et al. (2015), 'Predicting Song Popularity', (online) Available at: http://cs229.stanford.edu/proj2015/140_report.pdf , [Accessed on: 01 March 2019]

[9] Mohamed, N et al. (2018), ‘Song Popularity Predictor’, (online) Available at: https://towardsdatascience.com/song-popularity-predictor-1ef69735e380 , [Accessed on: 24 February 2019].

[10] Apache spark: [https://spark.apache.org/docs/latest], (access on:23 February 2019)

[11] Jupyter Notebook: [https://jupyter-notebook.readthedocs.io/en/stable/notebook.html], (access on 24 Feb 2019)

[12] Popular music definition:[https://github.com/AsTimeGoesBy111/Spotify-Music-Data-Analysis/], [access on: 24 February 2019]

[13] Pandas. Dataframe:[https://www.geeksforgeeks.org/python-pandas-dataframe/],(access on: 6 March 2019)

[14] Seaborn library:[https://seaborn.pydata.org/index.html], [Accessed on: 27 February 2019]

User Document

1 - Disclaimer

This user document assumes that the reader has previous work experience with running code on a computer as well as the know-how to install the required libraries

2 - Downloading the code

The first step will be to download the code from the open source repository in Git-Hub.
The repo is available at: https://github.com/Carlos-Tiago/OneMillionSongsAnalysis. The user can either clone the repository to a selected location on its local machine or manually downloading the files.

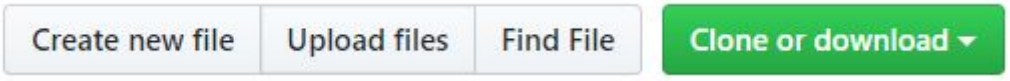


Fig1- Option to clone or download the content of the repo

3 - Running the code

In order to run the code, first the user must assure the installation of the following software:

- ➔ An IDE of choice (running the code on the terminal is also possible provided the user has the knowledge to do so)
- ➔ Python v3.0 programming language or above.

Then, to ensure that all the necessary libraries are present, the user can open the terminal / command prompt and type the following commands:

Getting numpy (array operations)

→ *sudo pip install numpy*

Getting the Spark library for Python (loading and working with the data)

→ *sudo pip install pyspark*

Getting matplotlib (plots)

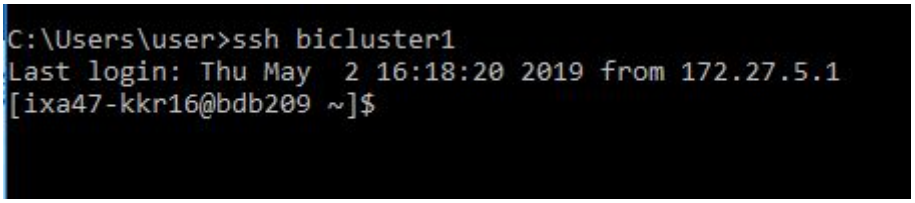
→ *sudo pip install matplotlib*

Note: Due to the command “sudo”, the users password may be requested (as in to provide administrator rights and ensure that everything can be installed in the standard locations). If the user does not have ownership of the machine, just add “ *--user* “ to the end of each command. This will create a “local” installation of the libraries to the current session.

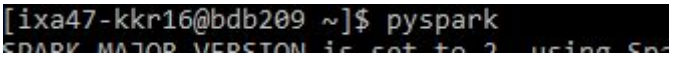
Then to run the code in the same execution environment:

1) First, we need to get the private key from Hartree then we apply it to our machines to be able to login to Dawson

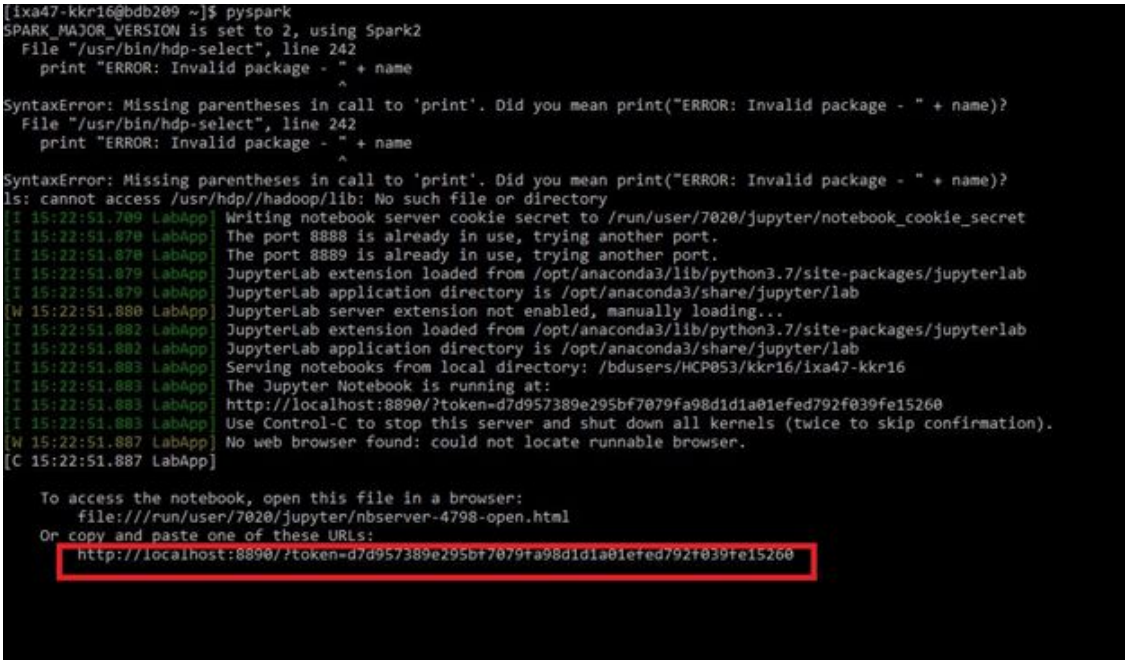
2) Login to Dawson by ssh bicluster



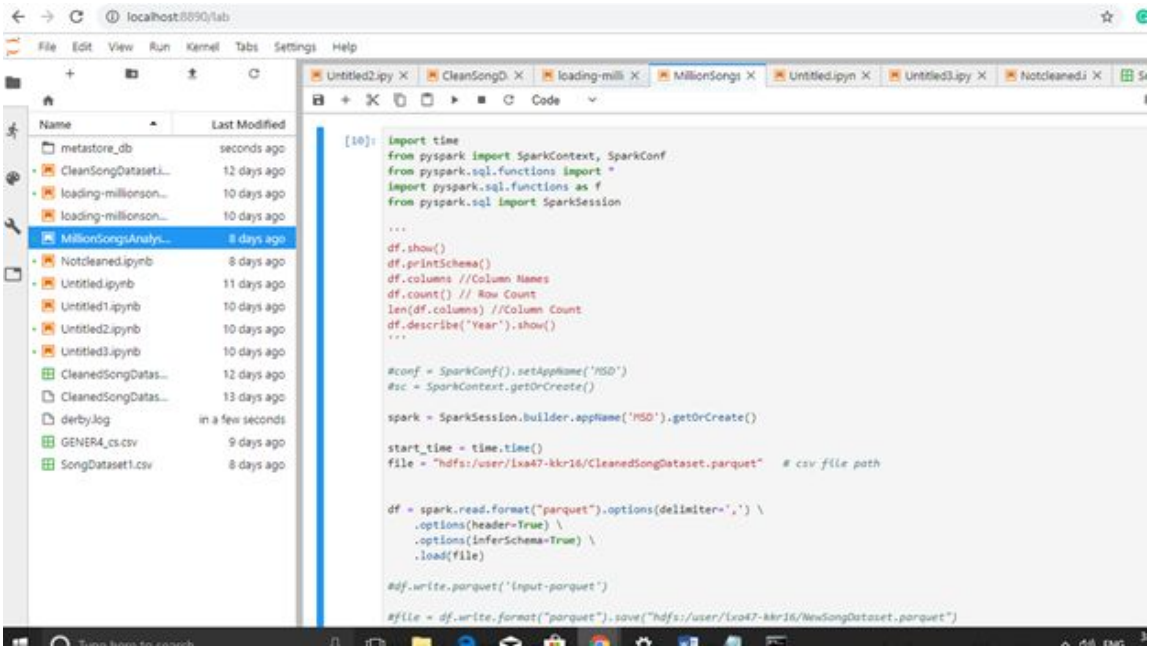
3) Write pyspark



4) User could take the link as highlighted below and paste it into the browser



5) Now you got the Jupyter Notebook just need to create a new notebook and paste the code there and press run



6) In order to access the CSV file that are in the Hadoop file system we need to check the location of the file first by applying the below code

```
[ixa47-kkr16@bdb209 ~]$ hdfs dfs -ls
Found 13 items
drwx------   - ixa47-kkr16 kkr16          0 2019-05-01 07:00 .Trash
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-05-02 16:41 CleanedData.csv
-rw-r--r--   3 ixa47-kkr16 kkr16     380970 2019-04-30 15:34 CleanedSongDataset.csv
-rw-r--r--   3 ixa47-kkr16 kkr16     212195 2019-04-30 16:31 CleanedSongDataset.parquet
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-05-01 13:58 GENER4.parquet
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-05-01 14:02 GENER41.parquet
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-05-01 14:03 GENER42.parquet
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-05-01 14:05 GENER43.parquet
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-05-01 14:17 GENER44.parquet
-rw-r--r--   3 ixa47-kkr16 kkr16     132148 2019-05-01 11:49 GENER4_cs.csv
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-04-30 20:03 NewSongDataset.parquet
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-04-30 18:29 SongDataset.csv
-rw-r--r--   3 ixa47-kkr16 kkr16    2444428 2019-05-02 16:19 SongDataset1.csv
[ixa47-kkr16@bdb209 ~]$
```

7) Put the CSV file into HDFS system using the below command as shown in the screen shot then you can list the file as shown in the screenshot below

```
[ixa47-kkr16@bdb209 ~]$ hdfs dfs -put GENER4_cs123.csv /user/ixa47-kkr16/
[ixa47-kkr16@bdb209 ~]$ hdfs dfs -ls
Found 13 items
drwx------   - ixa47-kkr16 kkr16          0 2019-05-10 15:47 .Trash
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-05-02 16:41 CleanedData.csv
-rw-r--r--   3 ixa47-kkr16 kkr16     380970 2019-04-30 15:34 CleanedSongDataset.csv
-rw-r--r--   3 ixa47-kkr16 kkr16     212195 2019-04-30 16:31 CleanedSongDataset.parquet
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-05-01 13:58 GENER4.parquet
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-05-01 14:02 GENER41.parquet
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-05-01 14:03 GENER42.parquet
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-05-01 14:05 GENER43.parquet
-rw-r--r--   3 ixa47-kkr16 kkr16     132148 2019-05-10 15:50 GENER4_cs123.csv
drwxr-xr-x   - ixa47-kkr16 kkr16          0 2019-04-30 18:29 SongDataset.csv
-rw-r--r--   3 ixa47-kkr16 kkr16    2444428 2019-05-02 16:19 SongDataset1.csv
```

8) Then you could run the code in Jupyter Notebook.

4. Explanations of error messages and troubleshooting guides

Some errors may be happened in running the cleaning code:

(1) FileNotFoundError: [Errno 2] No such file or directory

Solution:The system did not find the data file, you need to find out the correct path where the data saved.

Some errors may occured in starting spark session:

(1) IllegalArgumentException: "Error while instantiating 'org.apache.spark.sql.hive.HiveSessionStateBuilder':"

Solution: The systems failed to start a spark session. Users need to shutdown the kernel, deleted the metastore_db directory, then restart the kernel again.

(2) AnalysisException: 'Path does not exist: hdfs://bdb1a03.dawson.hartree.stfc.ac.uk:8020/user/yxl13-kkr16/NumSongDataset.parquet;'

Solution:The systems failed to find the file. Maybe the file name is wrong, or maybe you did not move the file to Dawson HDFS system.

5. Information to contact the developer of the system if an undocumented question arises

Yuefeng, Liang email: liangyf1013@gmail.com

AL-Jahdhami, Ismail email: aljahdhami@gmail.com

Arinto, Carlos email: carlostiago.arinto@gmail.com

Statement of where source code and dataset

Source code are in group 1 file exchange. You could find the sample dataset in Dawson HDFS
the command to view the dataset is
hdfs dfs -ls /hdfs/data/HCP053/songs/MillionSongSubset/data