MANIPAL UNIVERSITY
JAIPUR

INSPIRED BY LIFE

## School of Computing and Information Technology

**Department of Computer Science Engineering**

**B. TECH– MINOR PROJECT REPORT**

# CS1779

# Mini Project Course

**Supervisor - Jayakrishna R**

*A Project Report*

*on*

# Million Song Dataset Predictive Analysis using Regression Techniques

*carried out as part of the course CS1779*

*Submitted by*

***Yash Chaudhry***

***149105420***

***VII B.Tech CSE – C***

***Akul Rastogi***

***149105416***

***VII B.Tech CSE - A***

*in partial fulfilment for the completion of course*

**CS1779 Mini Project**

In

**Computer Science and Engineering**

**Department of Computer Science and Engineering,**
**School of Computing and IT,**
**Manipal University Jaipur,**

***November, 2017***

# CERTIFICATE

This is to certify that the project entitled **Million Song Dataset Predictive Analysis using Regression techniques** is a bonafide work carried out as part of the course ***CS1779 Mini Project***, under my guidance by ***Yash Chaudhry & Akul Rastogi,*** students of ***B.Tech CSE VIIth sem*** at the Department of Computer Science and Engineering, Manipal University Jaipur, during the academic semester _***VIIth***__, in partial fulfilment of the requirements for the completion of course 1779 in Computer Science and Engineering, at MUJ, Jaipur.

Place:

Date: Signature of the Instructor (s)

# DECLARATION

I hereby declare that the project entitled **Million Song Dataset Predictive Analysis using Regression techniques** *submitted* as part of the partial course requirements for the course ***CS1779 Mini Project***, for the completion of the course in Computer Science and Engineering at Manipal University Jaipur during the ***November 2017 VII$^{th}$*** semester, has been carried out by me. I declare that the project has not formed the basis for the award of any degree, associate ship, fellowship or any other similar titles elsewhere.

Further, I declare that I will not share, re-submit or publish the code, idea, framework and/or any publication that may arise out of this work for academic or profit purposes without obtaining the prior written consent of the Course Faculty Mentor and Course Instructor.

| | |
|---|---|
| Signature of the Student: | Signature of the Student: |
| Place: | Place: |
| Date: | Date: |

# Table of Contents:

# 1. Introduction

Music is an art form and cultural activity whose medium is sound organized in time which becomes an integral part of our lives. We listen to it when waking up, while in transit, at work, and with our friends. For many, music is like a constant companion. It can bring us joy and motivate us, accompany us through difficult times, and alleviate our worries.

Music is much more than mere entertainment. It has been a feature of every known human society—anthropologists and sociologists have yet to find a single culture throughout the course of human history that has not had music. It's extraordinary to think that a simple vibration unseen by the human eye can facilitate a deeply rich emotional experience, alter perception and consciousness, and induce ecstatic states of being.

Learning to predict and categorize this music, using data science prediction algorithms and principles, can really help us to gain an insight into what lies in the heart of the music and its listener while bringing us closer to question the very aspect of its evolution and improvisation through the human era by understanding its constituents impact over time.

## 1.1) Motivation

As digital content distribution booms, people now have and can access music collections on an unprecedented scale. Commercial music libraries easily exceed 15 million songs, which is a huge number and which vastly exceeds the listening capability of any single person! Even one million songs would take more than seven years of non-stop listening.

Now think, what if we wanted to categorize all of the music ever produced o create a library; which could be used by a music recommendation system, how cumbersome that task could become?  Here Data Analytics, a booming industrial requirement, could play a crucial role in determining/predicting the release of the song through which it could be classified.

# 2. Literature Review

## 2.1) Importance and usage of Million Song Dataset

Million song dataset, a freely-available collection of audio features and metadata for a million contemporary popular music tracks, was created with an intent to help new researchers get started in the field of Machine Learning as it provides a huge reference dataset for evaluating their research findings. Attractive features of the Million Song Database include the range of existing resources to which it is linked, and the fact that it is the largest current research dataset in the field. It also encourages research on algorithms that can scale to commercial size of data. Illustration in the paper of year prediction has been produced as an example application, a task that until now had been difficult to study owing to the absence of a large set of suitable data.

Many people dismiss the idea of this research area being a very practical and important academic domain but they fail to analyze the underlying problem with the current scenario. How can we develop scalable algorithms without the largescale datasets to try them on? Collecting the actual music for a dataset of more than a few hundred CDs becomes something of a challenge. It is simply a lot of work to manage all of the details for this amount of data. Finding happiness in results diagnosed from algorithms that only work for a mere hundred or thousands of a data unit is just an illusion. A large dataset helps reveal problems with algorithm scaling that may not be so obvious or pressing when tested on small sets, but which are critical to real world deployment. Sometimes we can even miss relatively rare phenomena's because we failed to look at the whole picture, we were only looking at a square peg of it. And of course, having a single, multipurpose, freely-available dataset greatly promotes direct comparisons and interchange of ideas and results.

| Dataset | No of Songs |
|---|---|
| RWC50 | 465 |
| CAL500 | 502 |
| GZTAN | 1000 |
| USPOP | 8752 |
| SWAT10K | 10,870 |
| OMRAS2 | 50,000 |
| MusiCLEF | 200,000 |
| MSD | 1,000,000 |

**Table 2.1- Depicting MSD stands out as the largest dataset currently available for research.**

## 2.2) Outcome of Literature Review

After evaluating closely, the aspect of the creation of this very data set, its usage and importance in the field of not just music but in the academic domain as well, one can easily comprehend and deduce that there is a lot of scope and need for the development of algorithms that work on a larger scale i.e. commercial levels and even psychologically it challenges people to introspect their ideas and notion on understanding that relevancy can be found even in the most tiniest, ever present, simple ideas and notions like music and art. Larger data set development hence becomes an important requirement if one really wants to predict or produce accurate findings in any given field or domain.

Some of the specific usage for the MSD can be Metadata analysis, artist recognition from the audio, automatic music tagging, recommendations and similarity findings, cover song recognition and mood prediction based on lyrics.

## 2.3) Problem Statement

Predict the release year of a song using the attributes provided in the dataset.

## 2.4) Research Objective

1) Understanding the importance of creation of large dataset.
2) Analyzing the constituents of MSD dataset.
3) Understanding the recent trend/liking towards music by people.
4) Finding correlations among the constituent for accurate predictability.
5) Developing a prediction algorithm that scales to commercial size.

# 3. Methodology and Framework

## 3.1) System Architecture

| | |
|---|---|
| Processor: | Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz x 4  2.40GHz |
| Memory: | 4.00GB |
| System Type: | 64-bit Operating System, x64-based processor |

## 3.2) Design Methodology, Algorithm and Techniques

1) Python 2.7 with Spark, Numpy, Pyplot APIs, produced in Jupyter Notebook
2) Baseline – Average Label Model
3) Linear Regression Model via Gradient Descent
4) Linear Regression Model via Stochastic Gradient Descent
5) Grid Search to find better regularization parameters

## 3.3) Technology Selection Criteria

➢ Spark

- **Speed** - Run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.
- **Generality** - Combine SQL, streaming, and complex analytics.
- **Runs Everywhere** - Spark runs on Hadoop, Mesos, standalone, or in the cloud. It can access diverse data sources including HDFS, Cassandra, HBase, and S3.

## ➢ Python

- **Hottest IT company requirement in scripting languages.**
- **Broad Standard Library -** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- It is a **general-purpose** language, which means it can be used to **build just about anything.**

## ➢ Numpy &Pyplot

- ***NumPy*** is the **fundamental package** for **scientific computing with *Python***
- It contains useful **linear algebra, Fourier transform, and random number capabilities.**
- It adds **support for large, multi-dimensional arrays and matrics**.
- ***Pylpot*** *functions m*akes **matplotlib library work like MATLAB** (a multi-paradigm numerical computing environment)
- Each **pyplot function makes some change to a figure**: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

## ➢ Jupyter Notebook

- ➢ **Language of choice** - The Notebook has support for over 40 programming languages, including Python & R.
- ➢ **Share notebooks -** Notebooks can be shared with others using email, Dropbox, GitHub and the Jupyter Notebook Viewer.
- ➢ **Interactive output** - Your code can produce rich, interactive output: HTML, images, videos.
- ➢ **Big data integration** - Leverage big data tools, such as Apache Spark, from Python, R

# 4. Work Done

## 4.1) Implementation phase

**Bottom Up Design:**

1) Perform Data Analysis on MSD Dataset to analyze attributes
2) Find relevant subset dataset from 280GB dataset for evaluation
3) Parse MSD Dataset to evaluate integrity of data
4) Develop RMSE evaluation model for testing accuracy of prediction
5) Categorize dataset into training, validation and testing dataset
6) Develop a baseline average model for comparison and evaluation
7) Implement Linear Regression model via Gradient Descent model algorithm
8) Implement MLlibs Linear Regression model via Stochastic Gradient Descent
9) Implement Grid Search to find better regularization parameters
10) Compare and evaluate best model for prediction
11) Visualize all of the above steps with relevant graphs
12) Produce heatmap for finding variance in attributes of the MSD dataset and to find relevant iteration number and accurate regularization parameters
13) Develop test case models for modular unit testing
14) Finally list out available algorithms for finding better prediction results

# Sequence Diagram

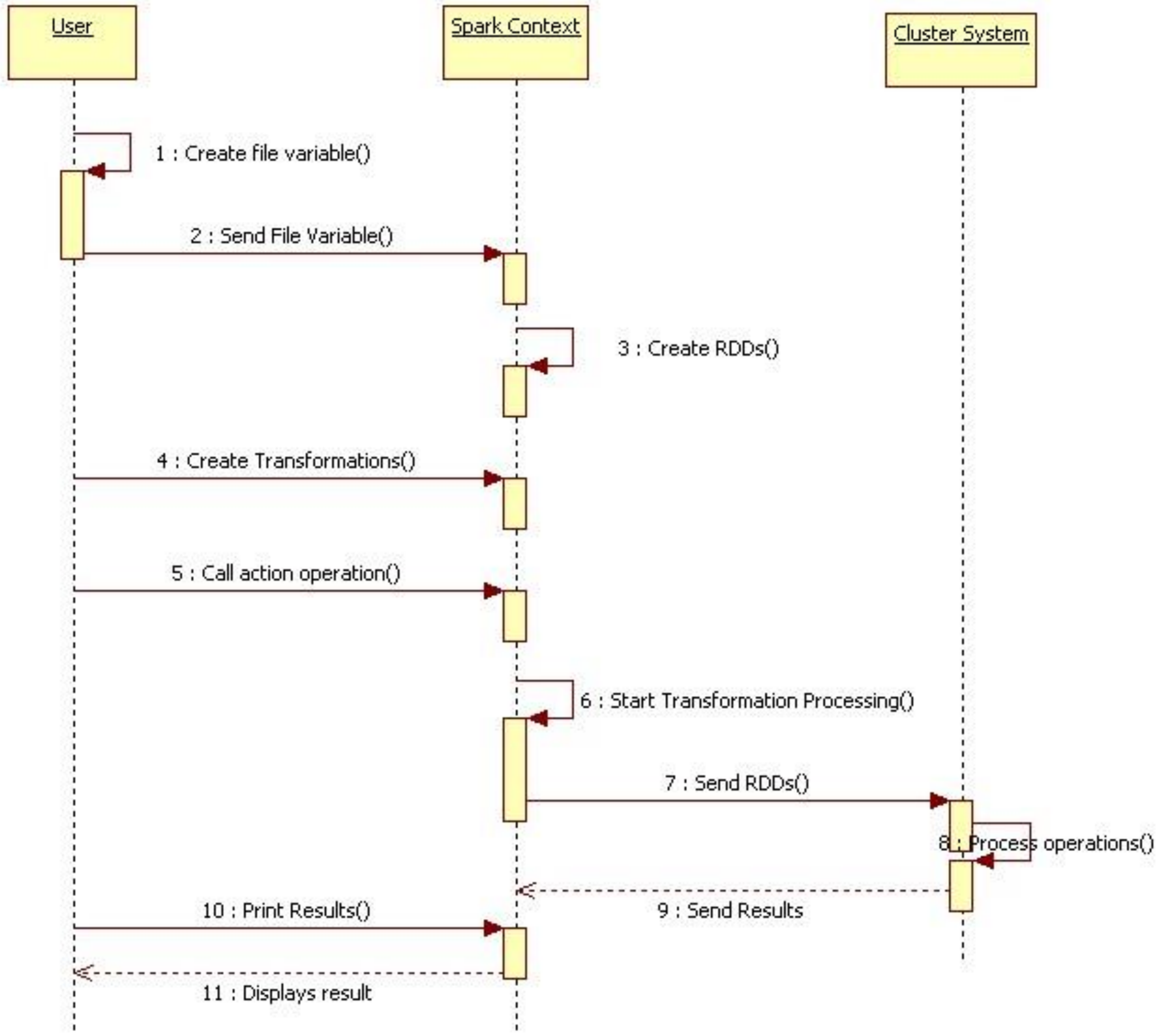


**Figure 4.1.1 Sequence Diagram depicts the working of Spark to process operations.**

# <u>Flowchart</u>

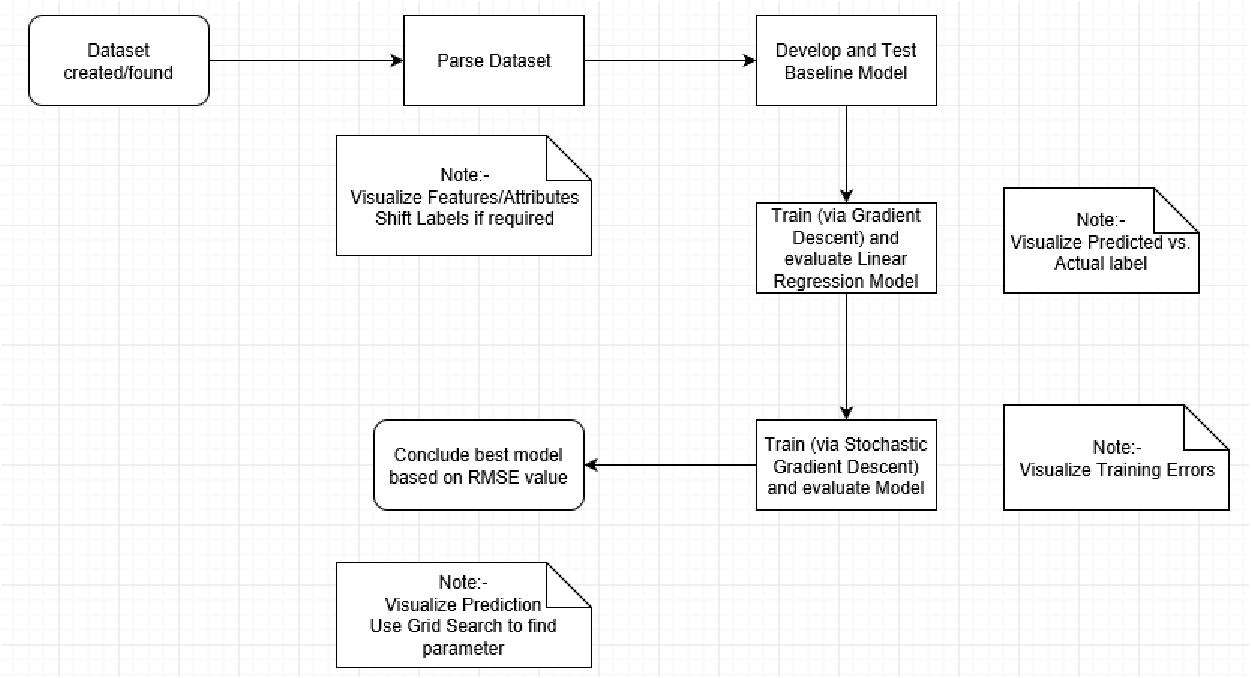

Dataset created/found
Parse Dataset
Develop and Test Baseline Model
Note:- Visualize Features/Attributes Shift Labels if required
Train (via Gradient Descent) and evaluate Linear Regression Model
Note:- Visualize Predicted vs. Actual label
Conclude best model based on RMSE value
Train (via Stochastic Gradient Descent) and evaluate Model
Note:- Visualize Training Errors
Note:- Visualize Prediction Use Grid Search to find parameter

**Figure 4.1.2 Flowchart Diagram depicts the procedure of solving the problem.**

## 4.2) Results

1) Baseline Model – Average Label

Baseline Validation RMSE = 21.586452

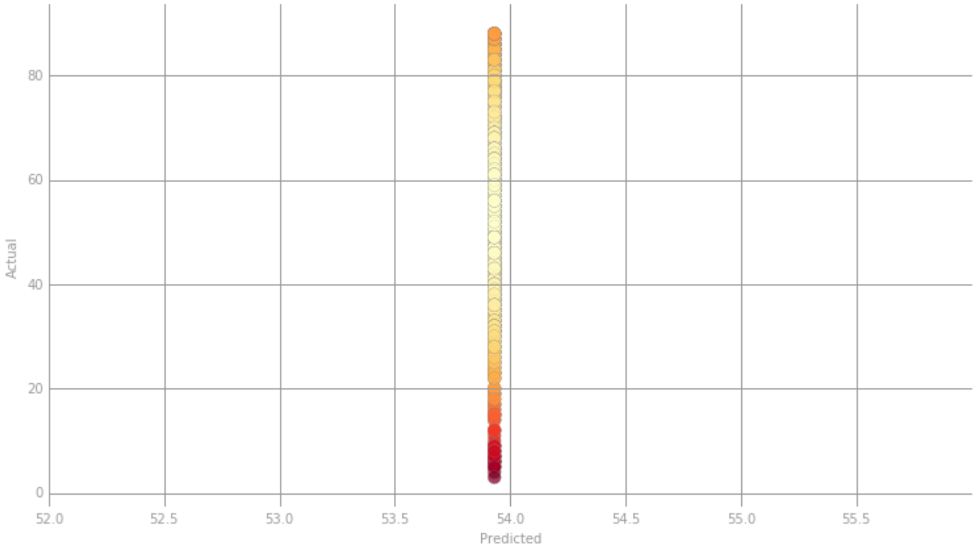


**Figure 4.2.1 - Average Label Model Prediction year vs Actual year of dataset**

**Inference :-**

- This scatter plot uses Baseline Prediction Model (Average Trained Year).
- Also, the points in the scatter plots are color-coded, ranging from light yellow when they are true and predicted values are equal to bright red when they drastically differ.

2) Linear Regression via Gradient Descent

Validation GD RMSE = 19.192

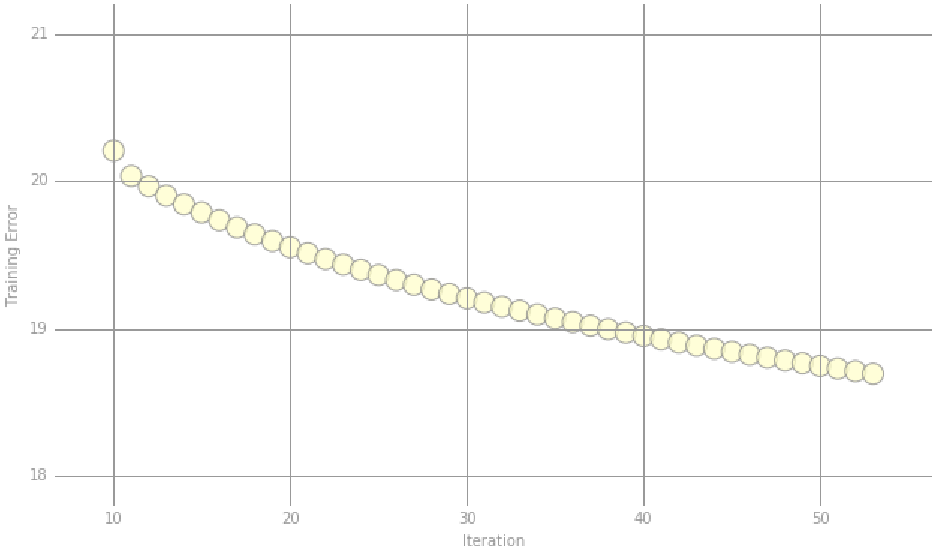


**Figure 4.2.2 - Training Error reduction analysis using Gradient Descent**

### Inference :-

- This scatter plot shows Training Error as a function of Iteration.
- Errors reduce as the number of iterations are increased to calculate more appropriate or accurate weights for Gradient Descent algorithm.

3) Linear Regression via Stochastic Gradient Descent

Validation RMSE GSD = 19.873

***Performance reduced due to ineffective regularization parameter setup.**

4) Grid Search on Stochastic Gradient Descent

Validation RMSE GSD with Grid Search = 17.483

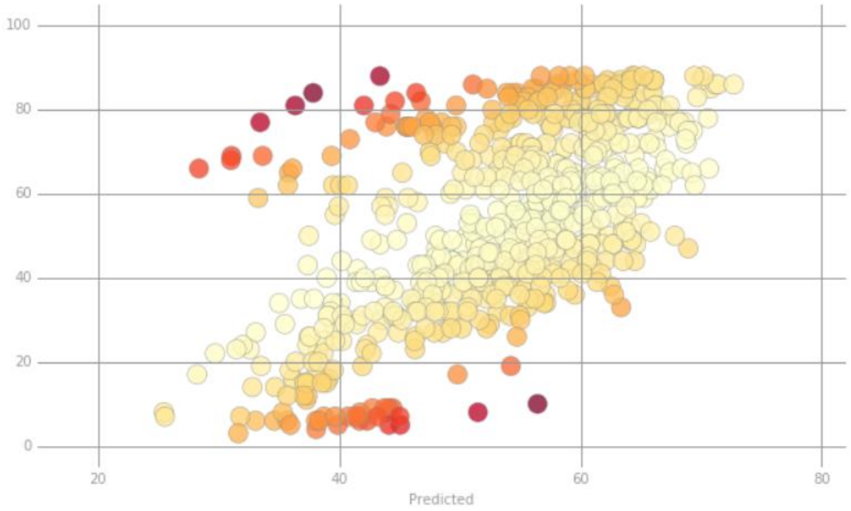


**Figure 4.2.3 - Grid Search GSD Model Prediction year vs Actual year of dataset**

**Inference :-**

- Stochastic Gradient Descent without good regularization parameter decision doesn't perform so well in comparison to Vanilla Gradient Descent .
- With Grid Search implementation on SGD we are significantly able to reduce the Root Mean Square Error.
- A color-coded scatter plot visualizing the predicted value from this model against true label.

5) Heatmap (Iteration no. & regularization parameter)

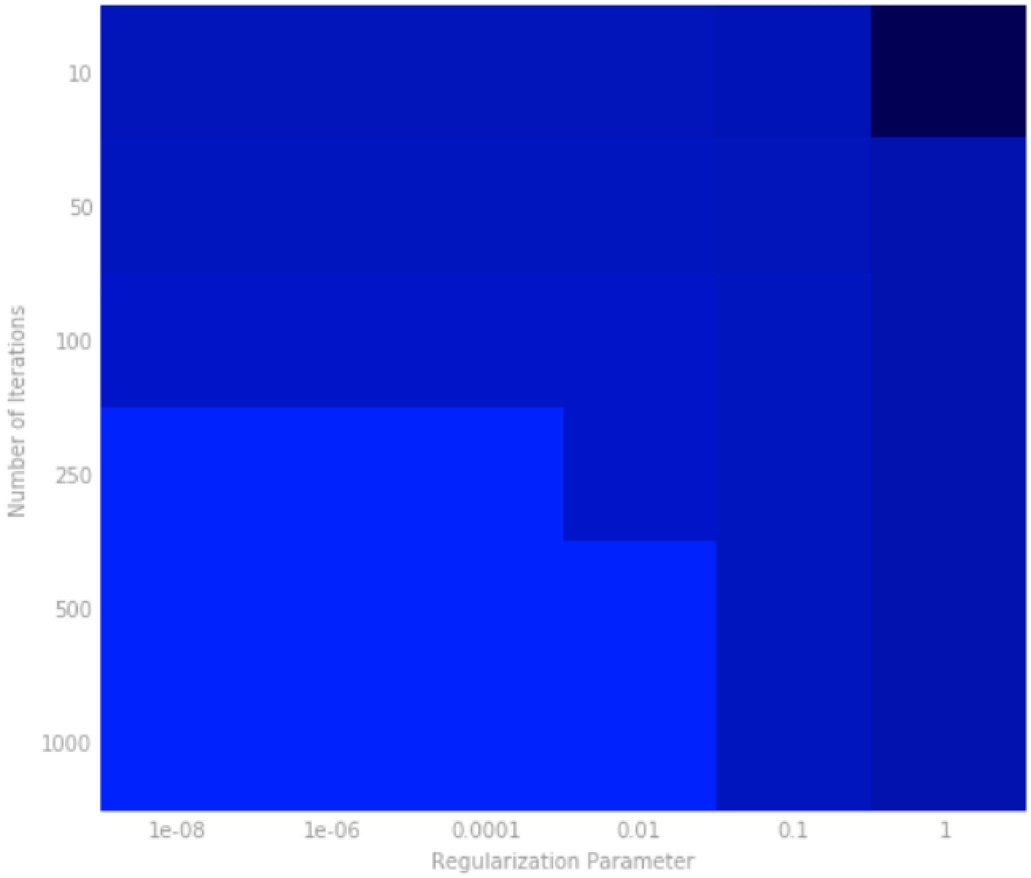


**Figure 4.2.4 - Brighter heat map depicting best results parameters**

## Inference :-

- The plot represents a heat map where the brighter colors correspond to lower RMSE values for regularization parameter against No. of Iterations.

6) Data Analysis using R programming

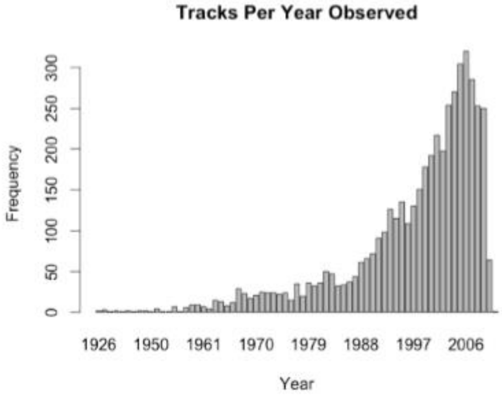
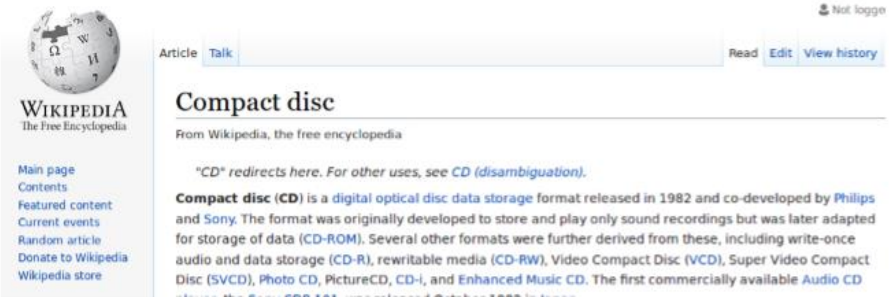


*Figure 1 - Shows No. Of songs and yearly distribution.*

**What do we find?**

- We observe that the count of frequency actually starts taking off from 1970.
- On further investigation we found that from below reference that the trend could be explained due to the invention of the CDs by Philips and Sony.



## Compact disc

From Wikipedia, the free encyclopedia

*"CD" redirects here. For other uses, see CD (disambiguation).*

**Compact disc** (**CD**) is a digital optical disc data storage format released in 1982 and co-developed by Philips and Sony. The format was originally developed to store and play only sound recordings but was later adapted for storage of data (CD-ROM). Several other formats were further derived from these, including write-once audio and data storage (CD-R), rewritable media (CD-RW), Video Compact Disc (VCD), Super Video Compact Disc (SVCD), Photo CD, PictureCD, CD-i, and Enhanced Music CD. The first commercially available Audio CD

**Figure 4.2.5 - Data Analysis Reporting**

## Inference :-

- A simple histogram depicting Frequency of songs in an year.
- Frequency starts taking off from 1970 and surging in 1980s
- Compact Disk released in 1982 developed by Sony and Philips? Could this have a hand in music surge?

| **Statistics** | **Value** |
|---|---|
| Size | 463715 |
| Mean | 1998.386 |
| Std Dev | 10.940 |
| Min | 1922 |
| 1st Quartile | 1994 |
| Median | 2002 |
| 3rd Quartile | 2006 |
| Max | 2011 |

**Table 4.2.6 - Statistical Analysis of MSD dataset subset**

## Inference :-

- Number crunching statistics to know what we are dealing with. Some of them :-
- Size of Dataset
- Standard Deviation
- Quartile Knowledge

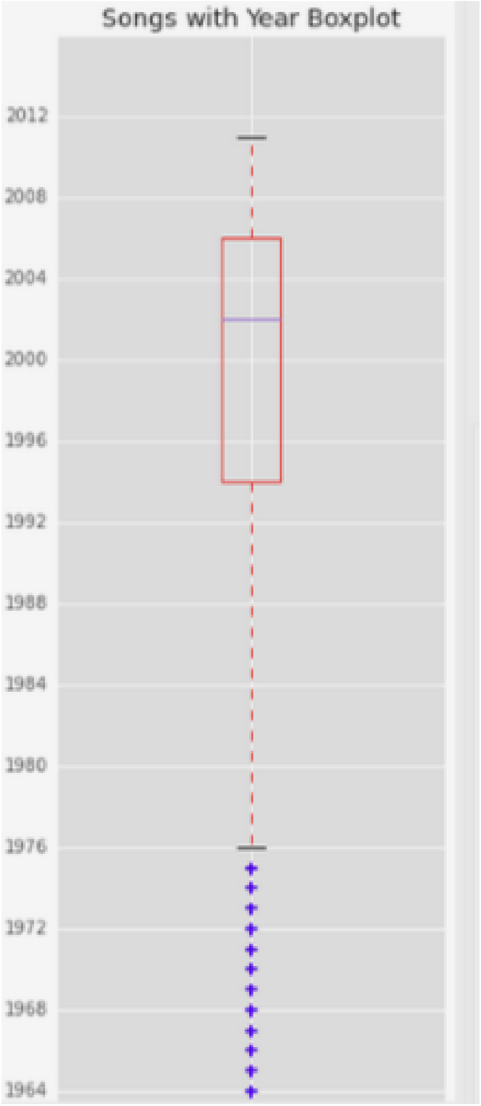
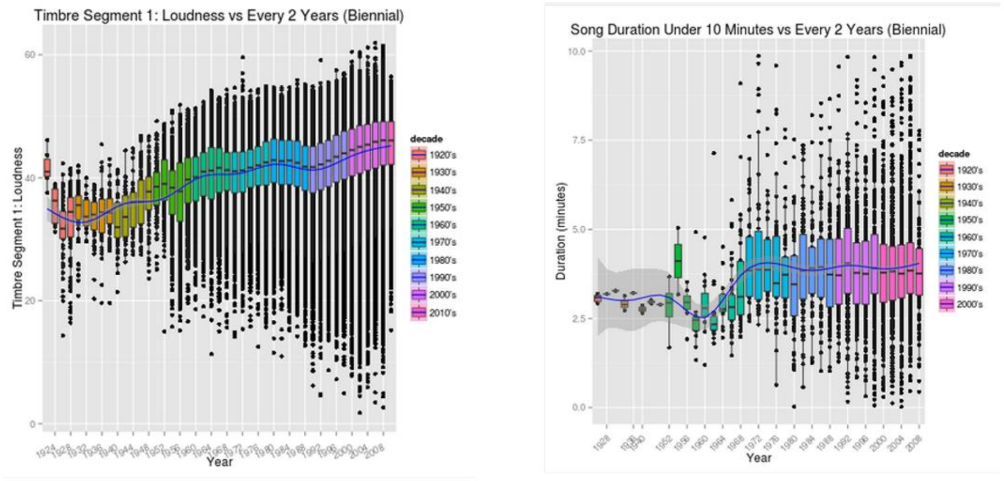


**Figure 4.2.7 - Year Boxplot for statistical analysis**

**Observations:**

- We can observe that we get a negative skew, a left skewed graph from histogram.
- Central mass of distribution lies between the mid 1990 to mid 2000s
- Song proportion from 1922 to 1976 is barely 5% of sample data.
- Peak count song year comes out to be 2007.
- 2007 was the release year of 1st Generation iPhone



*Music loudness increased over the years.*

*Song Duration increased over the years.*

**Figure 4.2.8 - Super Scatter Boxplot Lineplot for trend analysis**

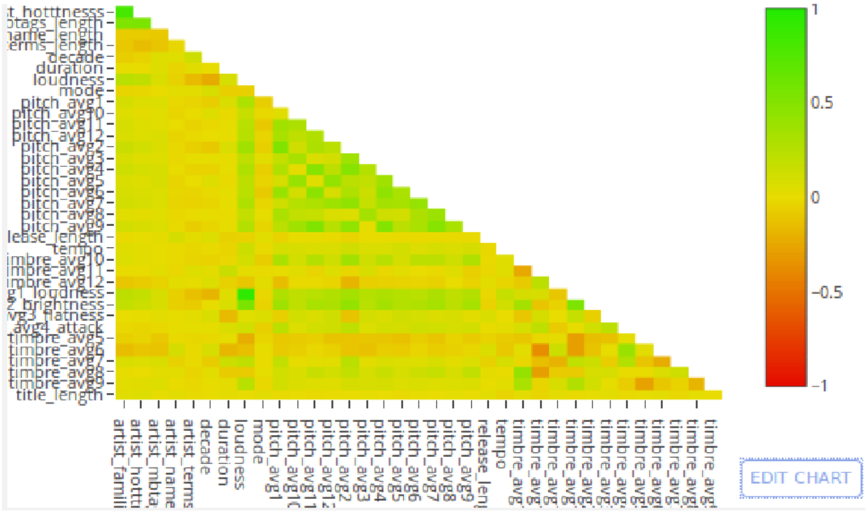


**Figure 4.2.9 - Heatmap for visualizing Correlations among attributes**

## Highly correlated features:

- Loudness vs. Familiarity (Louder the better?)
- Hotness vs. Familiarity (More familiar more hot?)

## 4.3) Individual Contribution

**Akul Rastogi**

- R Tutorials
- Trend Deduction in Data Analysis
- Raw data parsing and integrity check
- Unit testing modules
- Baseline Model Implementation
- Gradient Descent Implementation

**Yash Chaudhry**

- Internshala Data Analytics Course
- Statistical Analysis in Data Analysis
- Pyplot Visualizations
- Training, validation & test data set definition
- Stochastic Gradient Descent Implementation
- Grid Search Implementation

# 5. Conclusion and Future

Clearly Stochastic Gradient Descent with Grid Search integrated wins the race by providing the best possible/accurate prediction of the year release of a music in all of the models implemented by providing the least Root Mean Square Error.

By analyzing the attributes of the MSD dataset we can easily conclude that one could even produce a music/cover song, following the trends of the recent year, that might become a hit of tomorrow. So, anyone could become a Rockstar now? Even AI are analyzing the recent trends and developing music album to please the ears of humans.

Million Song Dataset is actually becoming the natural choice for researchers wanting to try out ideas and algorithms on a data that is standardized, easily obtained, and relevant to both academia and industry. Researchers at LabROSA in 2011 hoped for this very reality which is actually coming true every single day as people find their interest driving them towards data science techniques applications and usage.

Finally, as the dataset keeps growing everyday there is likely a better chance that we will actually get to see or find even a far bigger picture which might have been missing for so long which can in itself pave a way for far more innovative ideas, phenomena, trends, strategies and algorithms.

## 5.1) Proposed Work Plan for future

Implementation of following techniques can produce comparable or even better results:

1) 2 Way Interaction Model – Examine Variable combination which affects response
2) Gaussian Process Model Neural network - Avoiding local minima in neural networks
3) Nonlinear Conjugate Gradient - Very successful in regression - O(n)
4) L-BFGS - Uses Hessian approximation
5) Levenberg-Marquardt Algorithm - Best optimization algorithm but O(n^3)

# 6. References

[1] Million Song Dataset, official website by Thierry Bertin-Mahieux, available at: https://labrosa.ee.columbia.edu/millionsong/

[2] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, Paul Lamere THE MILLION SONG DATASET 12th International Society for Music Information Retrieval Conference (ISMIR 2011) http://ismir2011.ismir.net/papers/OS6-1.pdf

[3] Wikipedia https://en.wikipedia.org/wiki/Gradient_descent

[4] Sebastian Ruder http://ruder.io/optimizing-gradient-descent/

[5] Hackerearth http://blog.hackerearth.com/gradient-descent-algorithm-linear-regression

[6] Stackoverflow https://stackoverflow.com/questions/33621399/understanding-gradient-of-gradient-descent-algorithm-in-numpy

[7] Analytics Vidhya https://www.analyticsvidhya.com/blog/2017/03/introduction-to-gradient-descent-algorithm-along-its-variants/

[8] LearnR https://github.com/rstudio/learnr

[9] Anthony D Jospeh Spark Tutorial Berkeley https://www.edx.org/course/big-data-analysis-apache-spark-uc-berkeleyx-cs110x

[10] Evanz Machine Learning tutorial https://github.com/EvanZ/myvagrant

[11] Ujjwalkarn Machine Learning & R Tutoial https://github.com/ujjwalkarn/DataScienceR

[12] Python Pyplot Tutorial https://matplotlib.org/users/pyplot_tutorial.html

[13] Matt Nedrich https://github.com/mattnedrich/GradientDescentExample

[14] Spark Documentation https://spark.apache.org/docs/1.1.1/api/java/org/apache/spark/mllib/regression/LinearRegressionWithSGD.html

# 7. Appendix

**Research Paper Implementation -**

- Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, Paul Lamere THE MILLION SONG DATASET 12th International Society for Music Information Retrieval Conference (ISMIR 2011) http://ismir2011.ismir.net/papers/OS6-1.pdf