Predicting Year of Song from Audio Features

UC Irvine CS 178: Machine Learning and Data Mining

Tianqi Yang, Jiaxiang Wang, Wanjing Zhang, Zezhong Zhang

ABSTRACT

In this project, we are going to predict the song's year by using a dataset of extracted features of each song from Million Song Dataset(MSD)[18]. We collect approximately half million of songs which is a subset of MSD written for standardized tests to train different models including five machine learning algorithms which are Principal Component Analysis(PCA), T-distributed stochastic neighbor embedding(T-sne), Linear Regression, Support Vector Machine(SVM), Neural Network(NN) with 75% of the dataset as training data and 25% of the dataset as test data[12].

1.INTRODUCTION

There are many ways to categorize music, such as genre, artists, target listeners, and cultural context. In addition, different musical eras can also be regarded as an important standard to summarize a music feature. Certain music has a very obvious mark of their time, and people have a special taste for certain types of music as a result.

The **goal** of this project is to create a machine learning project which can predict the song's year by using the song's audio features[2]. The motivation for this project is that we all love music and we want to see how successfully the machine learning algorithm can predict based on purely subjective human judgment[19]. There are songs that do not have the mark of the time they really belong to. What we try to do, is satisfying the need of people, who have particular

favor for a type of music from a certain musical era, for example, music in the late 80s. We can give out the list of music that has the common characteristics belonging to that period no matter it was created at that time or not[4].

2.DATASET

For this project, we are planning to use the Million Song Dataset(MSD). MSD is a freely-available collection of audio features and metadata for a million contemporary popular music tracks by LabROSA at Columbia University and EchoNest. This was narrowed down to songs released between 1922 to 2011. Due to the size of the full dataset, we opted to use UCI Machine Learning Repository Dataset[21] which is a subset of MSD. The purpose here is to predict the year of the song was released in, based on its audio features.

2.1 Labels

The original dataset has released year as the label for each song. We convert this to release decade since we are trying to predict the decade a song was released in, not the exact year[6]. Each song owns 90 attributes. The first 12 attributes are the average timbre of each song, so we labeled as "TimbreAvg1, TimbreAvg2,…, TimbreAvg9". The rest attributes which are from the 13^{th} are the covariance labeled as "TimbreCovariance1, TimbreCovariance2,…, TimbreCovariance78".

2.2 Features

The Million Song Dataset contains both metadata and audio data for each song. In fact, one of our main challenges was deciding which features to use, since each song had more data than we could reasonably train on if we included all the audio data. The features extracted from

the 'timbre' features from The Echo Nest API. We take the average and covariance overall 'segments', each segment being described by a 12-dimensional timbre vector."[2].

3 MACHINE LEARNING ALGORITHM

To Predict a song's released timeframe, we used five machine learning algorithms which are Principal Component Analysis(PCA)[10], T-distributed stochastic neighbor embedding(T-sne)[3], Linear Regression, Support Vector Machine(SVM), Neural Network(NN)[5]. First, we are using the histogram to show the how the data is distributed and we found that the number of data samples is not uniform across release decades. There are not many samples of songs released before 1950.

We visualized our data by using Principal Component Analysis to reduce the dimensionality of the original 90-feature data into 2 dimensions[16]. Since we have reduced a large number of feature numbers, the PCA[20] result was not markable in giving credible patterns to our data. To improve the visualization, we further applied a new model, t-Distributed Stochastic Neighbor Embedding. By using t-SNE[8], we hoped that it could generate a more normalized pattern for us to read the pattern.

To make a primary prediction based on the features in the data, we used the Linear Regression model introduced in our course, CS 178. In order to get a horizontal comparison of features, we generated a random index number and implemented the corresponding feature to the regression, feeding the regression degrees [1, 2, 3, 5, 8, 13]. We also plotted an exhaustive feature comparison dot plots and histograms between every pair of 4 random-chosen features from year 1975 and 2010 to visualize the distinction between the songs in these 2 years.[11]

For the **Neural Network[1]**, the data is most likely skewed on the late 90s and early 2000s. The Neural Network can still perform very well with Stochastic Gradient Descent on MSD. We used a neural network regularization, with an alpha of 1e-5, 0.0001 and 10. Also, 4 hidden layers, each with 100 hidden nodes. The L2 regularization function was applied to the cost function to avoid over-fitting.

$$J(W) = \sum_{i=1}^{N}(y - \hat{y}) + ||\alpha_1 W_1 + \alpha_2 W_2||_2$$

Where W1 is the weight matrix mapping the features to the hidden layer and W2 is the weight matrix mapping the output of the hidden layer to the final output.

4.CODING

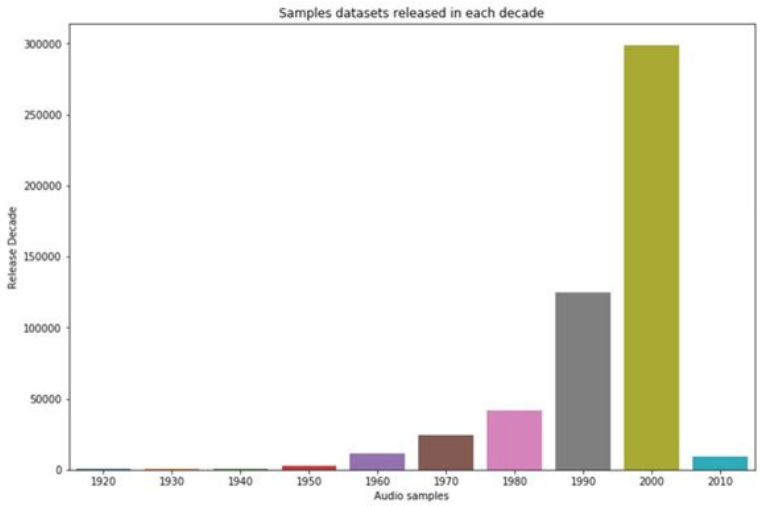
In the Linear Regression Algorithm, we split the data into 90% training data and 10% testing data. In this section, we used a pre-implemented smaller database extracted from our original one. We did a horizontal comparison among the features by feeding the model a random feature (range from 1-90) every time[15]. After visualizing the random features, we discovered great difficulty in making regression to the data and predict the year number for the correlation not being linearly distributed. Based on the regression, we deduced the MSE differences between training data and testing data with corresponding algorithm by feeding the model different regression degree [1, 2, 3, 5, 8, 13] so that we could eventually optimize the degree into a certain range[13].

In the Neural Network(NN) Algorithm[7], we split the data into 75% training data and 25% testing data. However, the MSD is too large, so we randomly select 10000 training data from training dataset and 1000 testing data from testing dataset. We use the histogram to show how data is distributed. Observing the distribution of classes in the training labels. Heavily biased towards late 90's/early 2000s, but it isn't entirely skewed. So we try to define a classifier. We call the multi-layer Perceptron classifier from sklearn library on the Stochastic Gradient Descent with an alpha of 1e-5. These parameters can be tuned. Currently, this neural network has 4 hidden layers, each with 100 hidden nodes. Then we fit the training examples into Neural Network MLPClassifier[17] and call predict function on the testing examples. After calling the prediction function, on the histogram, the result shows that the predicting year is 2004. It’s approximately correct. Then I tried to use different regularization values and different training example size to see if the predicting changes.

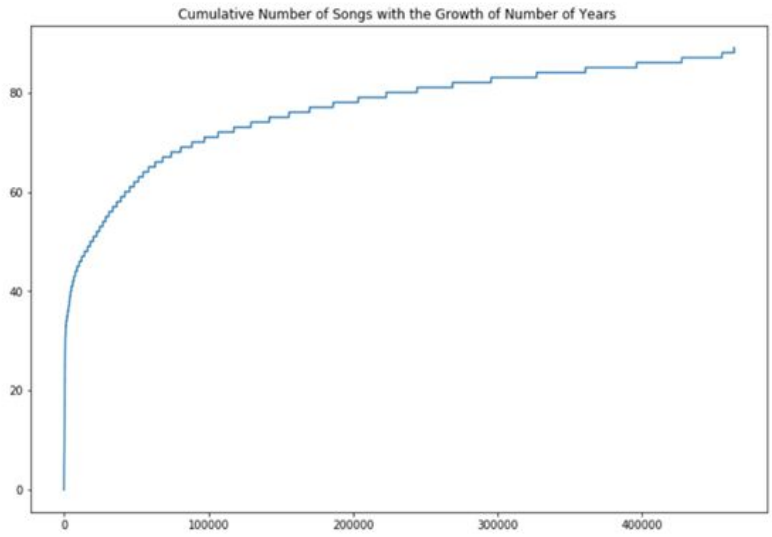
In the Support Vector Machine(SVM) Algorithm, we use the same method as we used on Neural Network Algorithm but we call the SVM from sklearn instead of MLPClassifier.

5.RESULT

After importing the data, we use 2 graphs to show the statistics of the dataset. **Figure#1** shows the spreading of the sample datasets released in each decade from 1920 to 2010. The **Figure#2** graph shows the cumulative number of songs with the growth of the number of Years. The x-axis for the third graph is the number of the sample and y-axis is accumulative of the increasing of the year. From these 2 graphs, we can have a basic idea of the characteristics of the dataset and how the data spread.



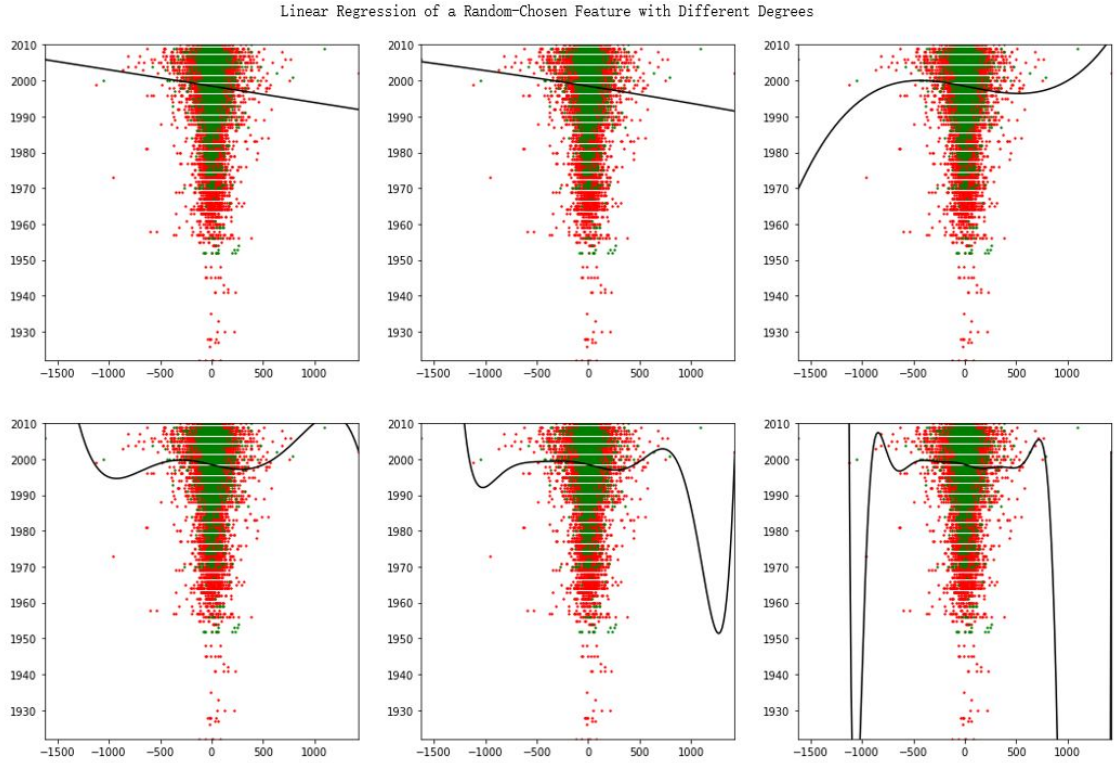
[Figure#1]



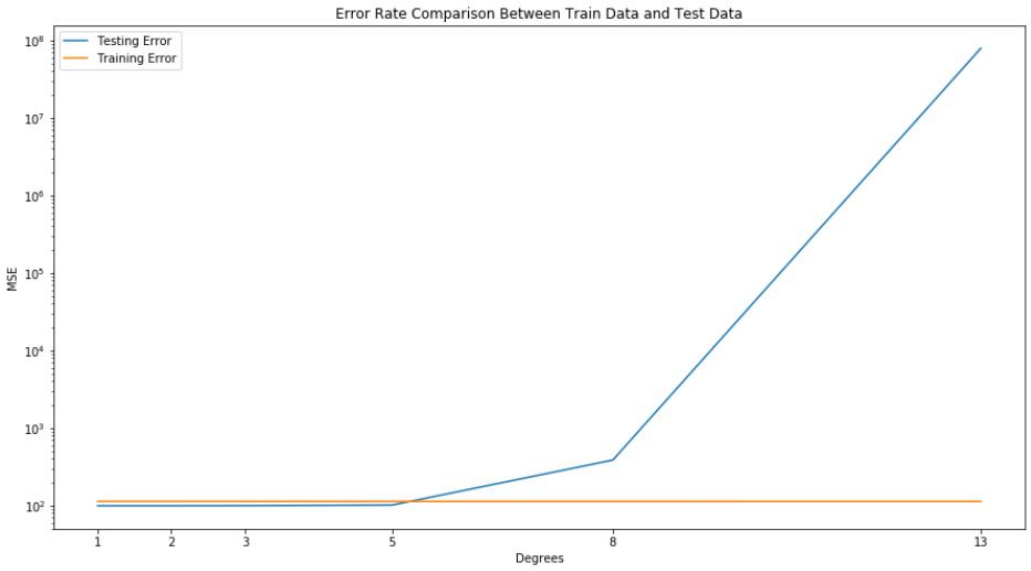
\

[Figure#2]

The Linear Regression model suggests a optimized regression degree range (1-5) by showing the MSE difference between training data and testing data[14].

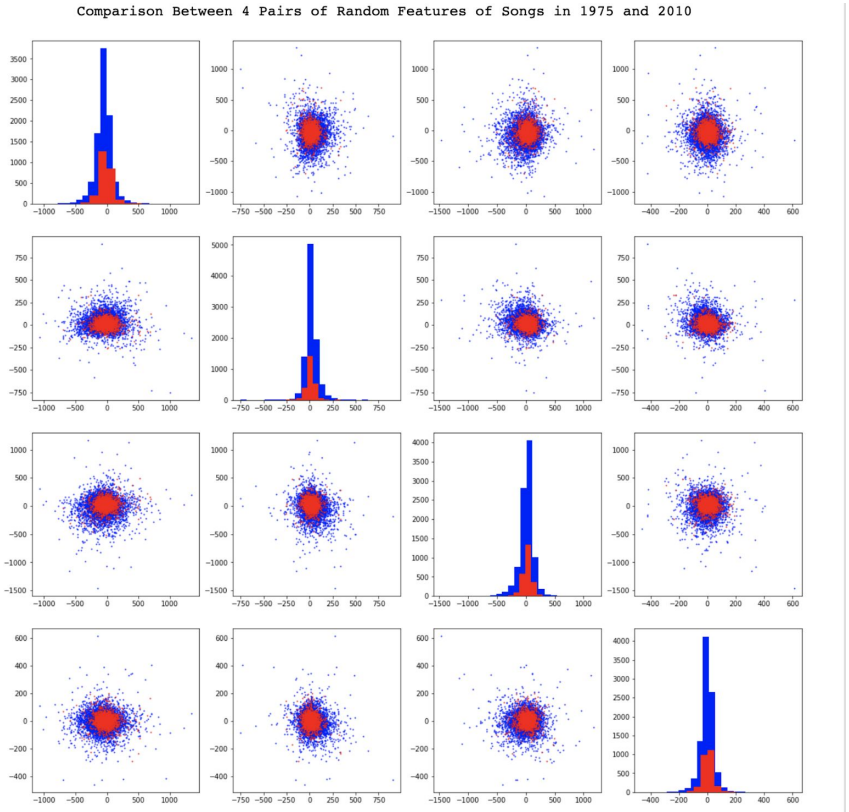


[Figure#3]



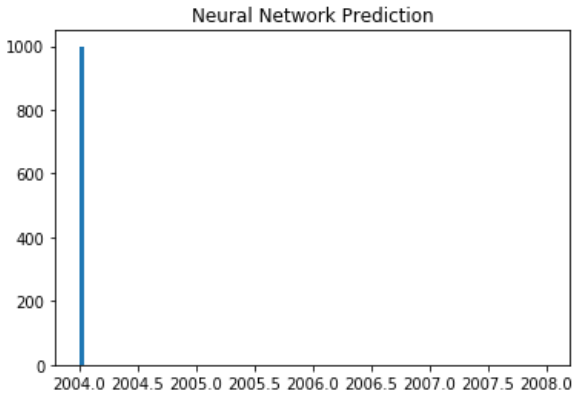
[Figure#4]

While the multi-dimensional comparison plots imply small distinctions between songs in 1975 and 2010 given 4 random features.



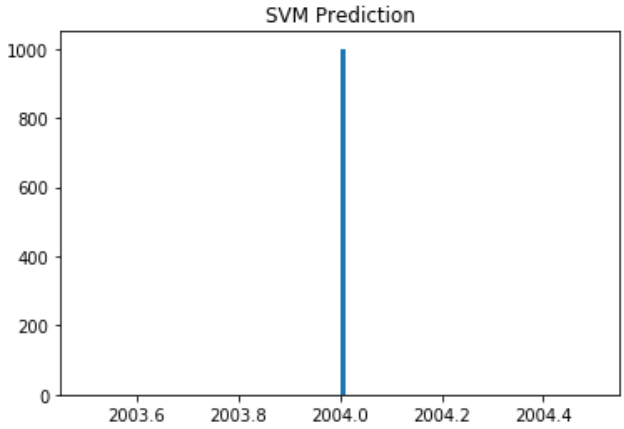
[Figure#5]

On the Neural Network Algorithm[9], we only get 7.5% accuracy on the prediction function. We try different regularization values and enlarge the training dataset. However, the result shows most likely the same. The prediction histogram is on the figure 6.



[Figure#6]

On the Support Vector Machine Algorithm, we get the same accuracy as we get in NN(7.5%). The prediction histogram for SVM is on the figure#7



[Figure#7]

6.CONCLUSION

As in the projects, we first observed the data to evaluate the results of models. Without a huge dataset, models can generate a good prediction direction that we believe is acceptable for the purpose of our application, as to grab the feeling of years of music. The task itself is a vague task as human beings can find it extremely hard to tell, and most likely not care which year exactly a song is composed. On the contrary, a user of this application may care much more about whether a group of songs that our model generates share the same feelings, i.e., share the same features.

7.References

[1] T. Li et al. "*Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network*" in IMECS, 2010.

[2] H. Bahuleyan. *Music Genre Classification using Machine Learning Techniques* in arXiv, 2018.

[3] "*Predict Release Timeframe from Audio Features*." Kaggle

[4] UCI Machine Learning Repository: Data Sets

[5] I. Simon et al. *Learning a Latent Space of Multitrack Measures*

[6] S. Dai et al. *Music Style Transfer:* A Position Paper in arXiv, 2018.

[7] L. Gatys et al. *A Neural Algorithm of Artistic Style* in arXic, 2018

[8] M. Abadi et al. *TensorFlow*: Large-scale machine learning on heterogeneous systems, 2015.

[9] T. Li et al. "Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network" in IMECS, 2010.

[10] "*Year Prediction - Million Songs Database*." song_year

[11] R. Chen et al. "*Isolating Sources of Disentanglement in VAEs*" in arXiv, 2018.

[12] "*How to Handle Imbalanced Classes in Machine Learning*." EliteDataScience, 25 Jan. 2019

[13] Albon, Chris. "*Handling Imbalanced Classes With Downsampling*." Chris Albon, 20 Dec.2017

[14] "*PANDAS-Questions and Answers*." National Institute of Mental Health, U.S. Department of Health and Human Services

[15] "*Loading CSV Data in Python with Pandas*." PythonHow

[16] Seif, George, and George Seif. "*5 Quick and Easy Data Visualizations in Python with Code.*" Towards Data Science, Towards Data Science, 1 Mar. 2018

[17] "*Sklearn.neural_network.MLPClassifier*." Scikit

[18] "*Getting the Dataset*." Getting the Dataset | Million Song Dataset

[19] "*Journal of Machine Learning Research*." Journal of Machine Learning Research

[20] "*Statistical Data Visualization*." Seaborn

8. CONTRIBUTIONS

- Wanjing - Pre-processing data, visualizing data with PCA and t-SNE, result references
- Zezong - Classical algorithms and experiments, poster lead.
- Tianqi Yang - Neural network, support vector machine, report lead, result discussions.
- Jiaxiang - Analyzing statistical data of Million Song dataset, future work.