Scott Fiene

Abstract

Predictive analytics is all around us from predicting what a customer will buy to explicitly placing products next to one another to drive sales. Predictive analytics have other uses as well. The music industry is one industry that can benefit from Predictive analytics and the applications are many. One application could help recording companies better tell what songs will hit the charts compared to songs that will not.

predicting songs that top the charts

# Table of Contents

[Introduction 2](#_Toc59318156)

[Defining the CRISP-DM Methodology 2](#_Toc59318157)

[Business Understanding 3](#_Toc59318158)

[Data Understanding 3](#_Toc59318159)

[Data Preparation 3](#_Toc59318160)

[Modeling 3](#_Toc59318161)

[Evaluation 4](#_Toc59318162)

[Deployment 4](#_Toc59318163)

[CRISP-DM Methodology in Practice 4](#_Toc59318164)

[Business Understanding 4](#_Toc59318165)

[Data Understanding 5](#_Toc59318166)

[Spotify Dataset 5](#_Toc59318167)

[Grammy and Billboard Dataset 11](#_Toc59318168)

[Top 500 Greatest Songs of All Time Dataset 15](#_Toc59318169)

[Finishing Up Data Understanding 17](#_Toc59318170)

[Data Preparation 17](#_Toc59318171)

[Spotify Dataset 17](#_Toc59318172)

[Grammy and Billboard Dataset 19](#_Toc59318173)

[Top 500 Greatest Songs of All Time Dataset 22](#_Toc59318174)

[Further Cleaning and Merging 25](#_Toc59318175)

[Data Exploration 31](#_Toc59318176)

[Modeling 33](#_Toc59318177)

[Evaluation 39](#_Toc59318178)

[Deployment 39](#_Toc59318179)

[Conclusion 40](#_Toc59318180)

[References 41](#_Toc59318181)

# Introduction

Predictive analytics (PA) (Wikipedia, 2020) encompasses a variety of techniques to analyze historical and current data then make predictions off that data. It leverages techniques such as data mining, modeling, and machine learning to make those predictions of unknown events better than a human can, theoretically. There is quite a bit of steps to take before a model is ready to make predictions as we will explore further in the paper. PA has been implemented in grocery stores and department stores alike, the most famous being Target, for product placement and how much of certain things to stock. There are some industries that a lacking behind the retail industry, healthcare being one that comes to mind. The other is the music industry.

The music industry is notoriously famous for singers and songwriters becoming producers themselves (Dr. Dre and Jay-Z come to mind). It seems like the next logical career choice once their singer/songwriter career is over, they have the knowledge, know the industry and many have awards to show their skills. What about all those producers that do not have that track record, but instead love music that they became classically trained to be a music producer. There are a few benefits to note that could be obtained by applying PA to the music industry.

First and foremost, people could predict whether the recorded song would make it to the charts. Secondly, being able to predict top hits would lead to tours sooner. Thirdly, PA would lead to more innovations in the industry once more and more people jumped on the “model.” The main thing in common here is more money for all people involved and greater future opportunities.

Focusing more on the present, this is still very much in development with quite a but more work to be done. This paper will go into detail about the data obtained, cleaning performed, methods used, models chosen, questions answered, and the next steps needed to take.

# Defining the CRISP-DM Methodology

(Provost & Fawcett, 2013) (Wikipedia, 2020) (Data Science Project Management, 2020) (IBM, 2020)

The CRISP-DM methodology was followed in the creation of this paper. CRISP-DM stands for *Cross Industry Standard Process for Data Mining.* It was created as an effort to standardize the methods used in PA and data science in the 1990’s. It is the most widely used analytics model according to Wikipedia article about it. There are 6 steps and most of the are iterative. As you go through the process it might be necessary that you need to get more understanding of a certain problem to move on or when evaluating you find the model is terrible and must double back to tune it. The steps are:

* Business Understanding
* Data Understanding
* Data Preparation
* Modeling
* Evaluation
* Deployment

The CRISP-DM methodology was used in the development of this paper as well as the cleaning of data and modeling. The method is briefly explained below.

## Business Understanding

The business understanding phase focuses on understanding the problem to be solved and the objects and requirements of the project at hand. In this phase the analysts as they are the bridge between the business side and the analytics side. Some questions to as are:

* What do we want to do?
* How do we want to do it?
* What parts of this will require models?
* What is the plan to get this done?

## Data Understanding

The data understanding phase adds to the business understanding phase. The goal is to solve a business problem and the data is where the solutions comes from. Collecting, describing, and determining the strengths and weaknesses of the available data is essential. It is now that we may have to return to the previous phase to get further information

## Data Preparation

Once the initial steps are taken to obtain the data and understand it, the data preparation phase can begin. This is one of the, if not the most, important parts of the CRISP-DM methodology. The saying “garbage in, garbage out” still holds true no matter how much data you must throw at a problem. Cleaning data, aggregating measures, coercing the data into certain forms and numbers, normalizing or standardizing, and renaming columns are just a few of the cleaning techniques. Data always needs to be cleaned.

## Modeling

Modeling should only happen after the data is properly cleaned. Data preparation takes a large amount of time, but modeling can take quite a while too. Selecting a model, building the model, and assessing the model are a very iterative process and is often trial-and-error. The first run of the model should be at default settings so many models can be tested then the best one be adjusted to make it even better

## Evaluation

The evaluation phase of, as many think, does not focus on evaluating the model. Rather, it focuses on whether the model will meet the needs of the business. A very detail scrutiny of the model, the data, and the results happen in this phase as well. This scrutiny is the gatekeeper to deploying the model into production. Stakeholders, manager, and any other party of interest will give their scrutiny as well. This will often lead to further honing the business problem by going back to the start armed with everything in the process thus far.

## Deployment

This phase is the very last phase to happen. This will give the end user access to the model for use in the destined application. It is not just a deploy and be done phase, but a continual process of monitoring and continued maintenance to keep it up and running. Review and reporting on the final product also happen here.

# CRISP-DM Methodology in Practice

The brief explanation of the methodology above will serve to create a deeper understanding into the process behind this paper. In the business understanding phase, the problem will be described along with a high-level understanding of the expectation. In the data understanding, the data and all its aspects will be explained in detail. The data in all its beautiful detail will be explained in the data preparation phase and the modeling technique and method will be explained in the modeling phase. Assessment of the model in terms of the business problem is the topic for the evaluation phase and the deployment phase will discuss the next steps for the model.

## Business Understanding

As mentioned above, the music industry is notoriously famous for singers and songwriters becoming producers themselves (Dr. Dre and Jay-Z come to mind). It seems like the next logical career choice once their singer/songwriter career is over. They have the knowledge, know the industry and many have awards to show their skills. What about all those producers that do not have that track record, but instead love music that they became classically trained to be a music producer. There are a few benefits to note that could be obtained by applying PA to the music industry. First and foremost, people could predict whether the recorded song would make it to the charts. Secondly, being able to predict top hits would lead to tours sooner. Thirdly, PA would lead to more innovations in the industry once more and more people jumped on the “model.” The main thing in common here is more money for all people involved and greater future opportunities. The problem is how can music producers better determine whether the song will be a hit and top the charts and ultimately make greater income.

## Data Understanding

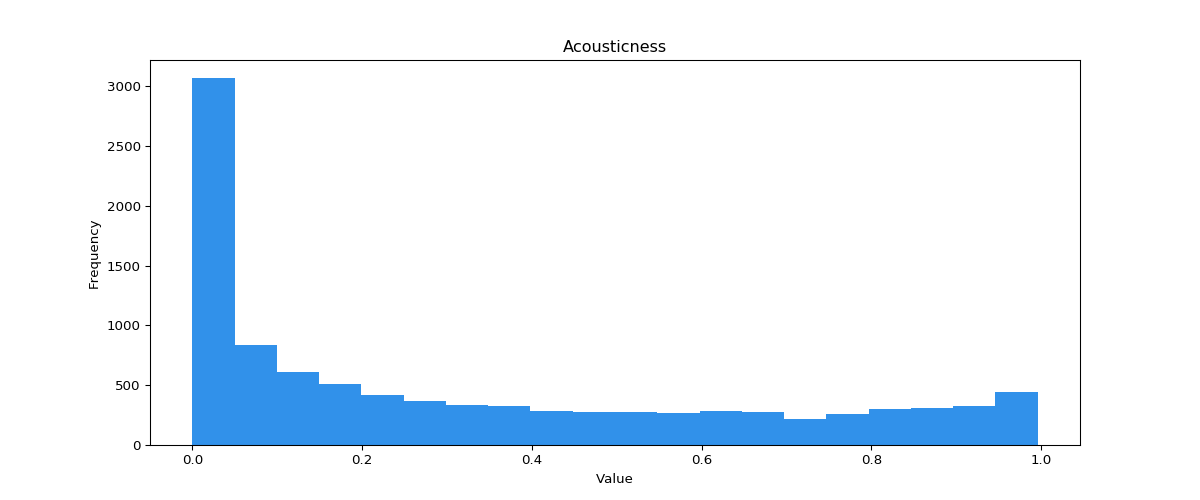
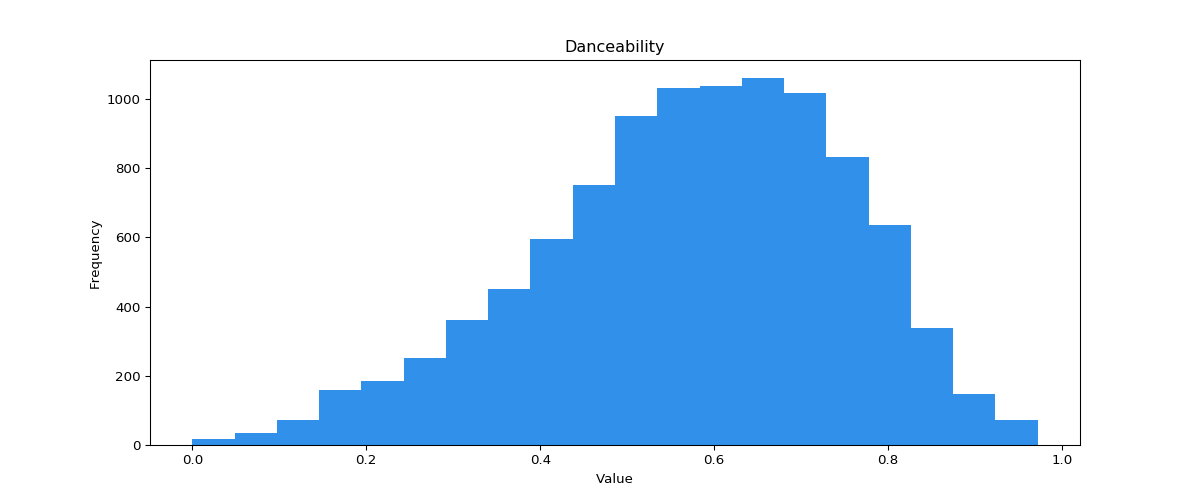
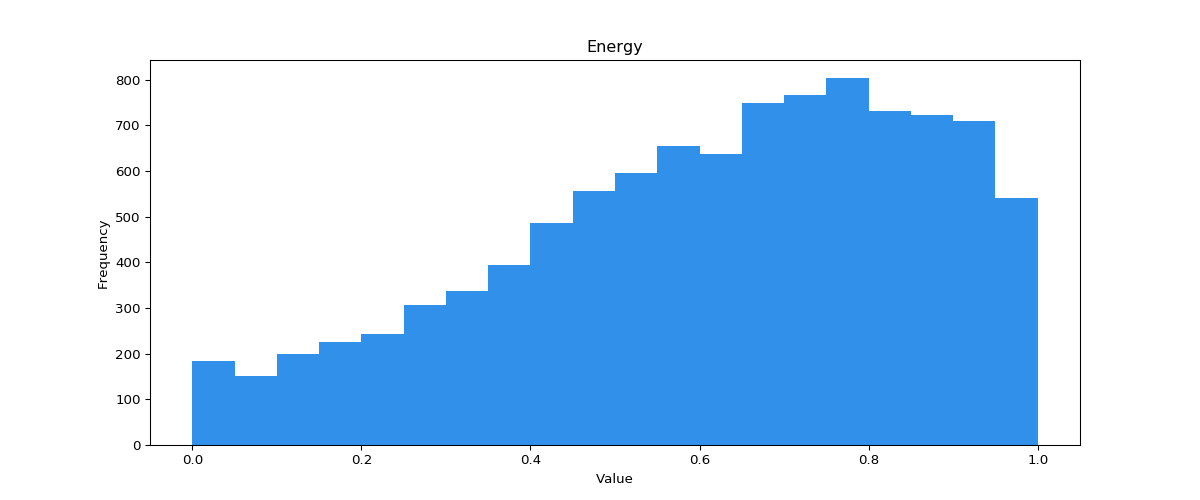
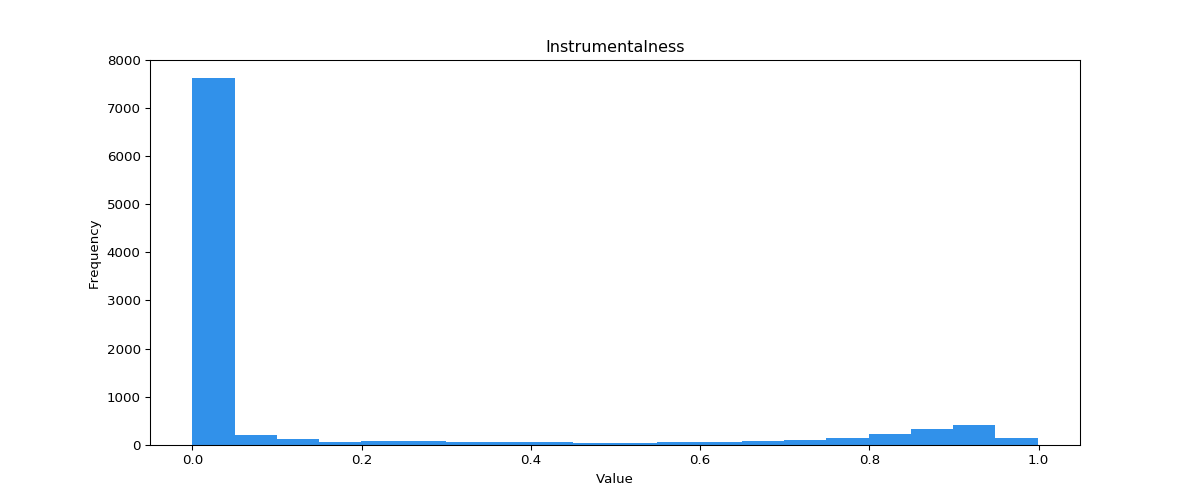
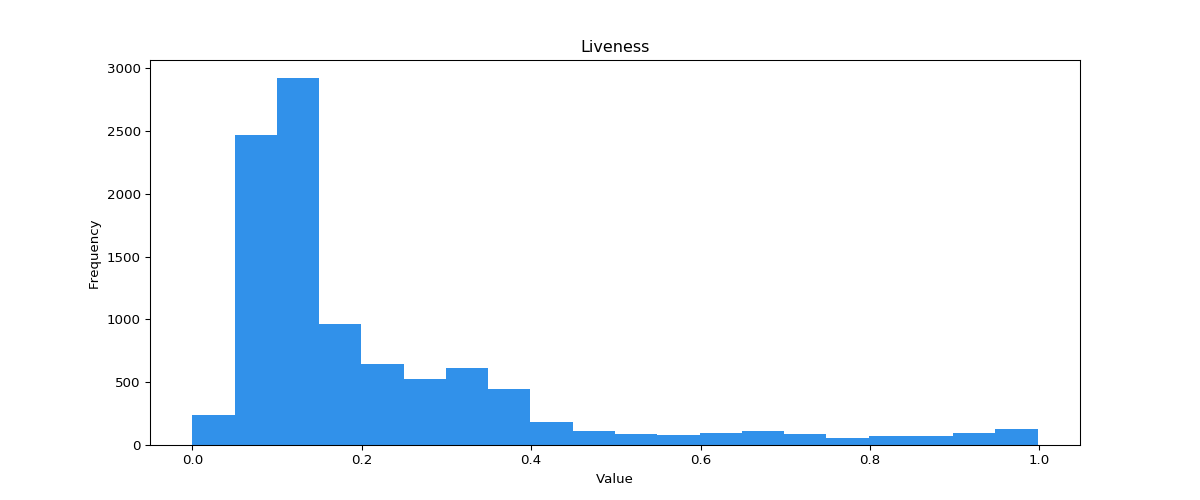
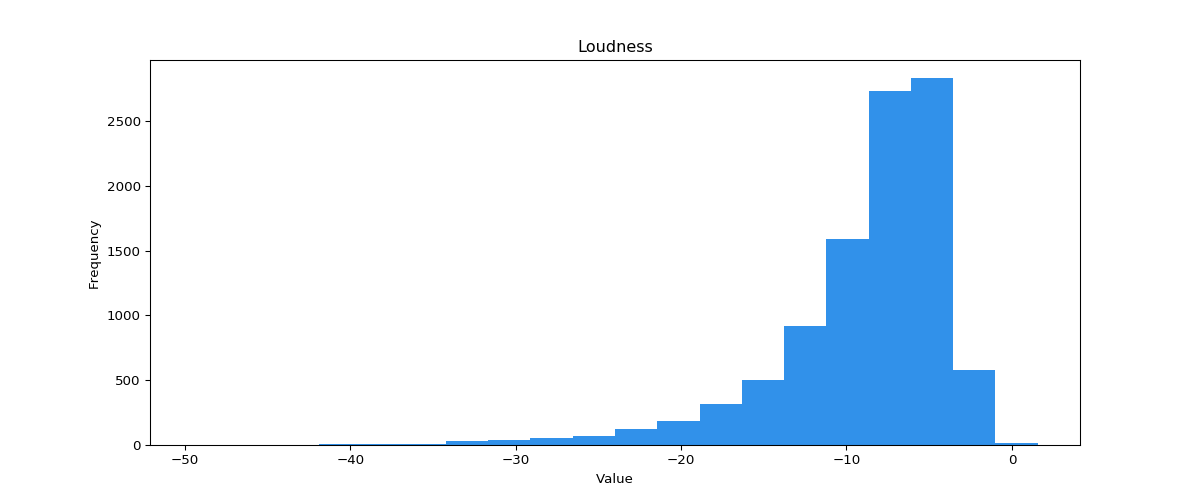
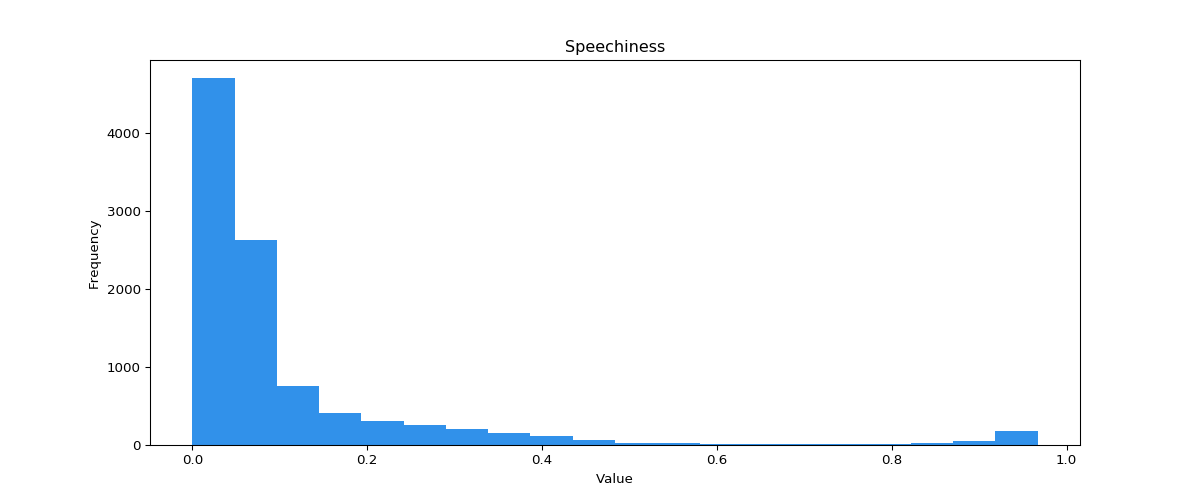
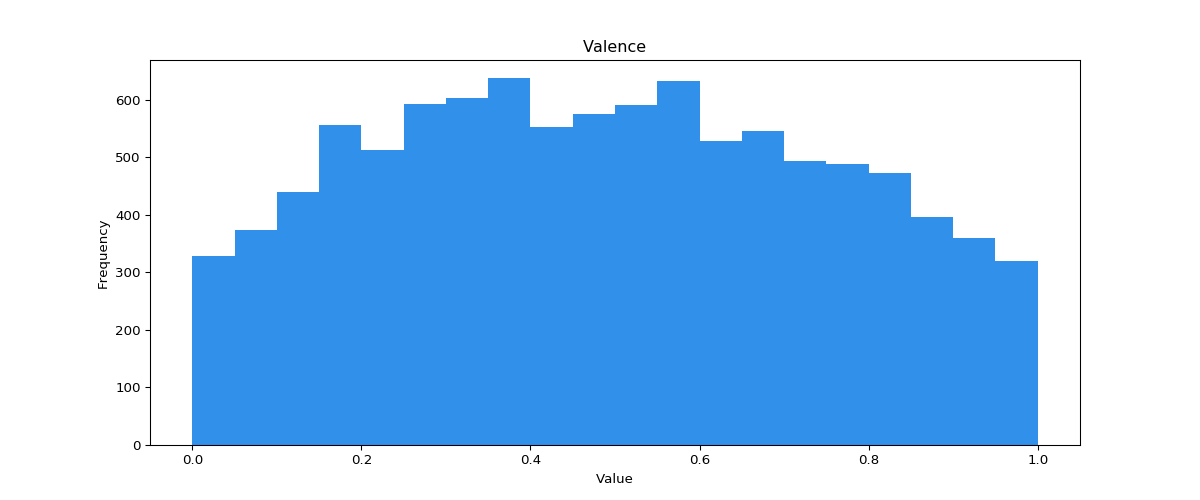
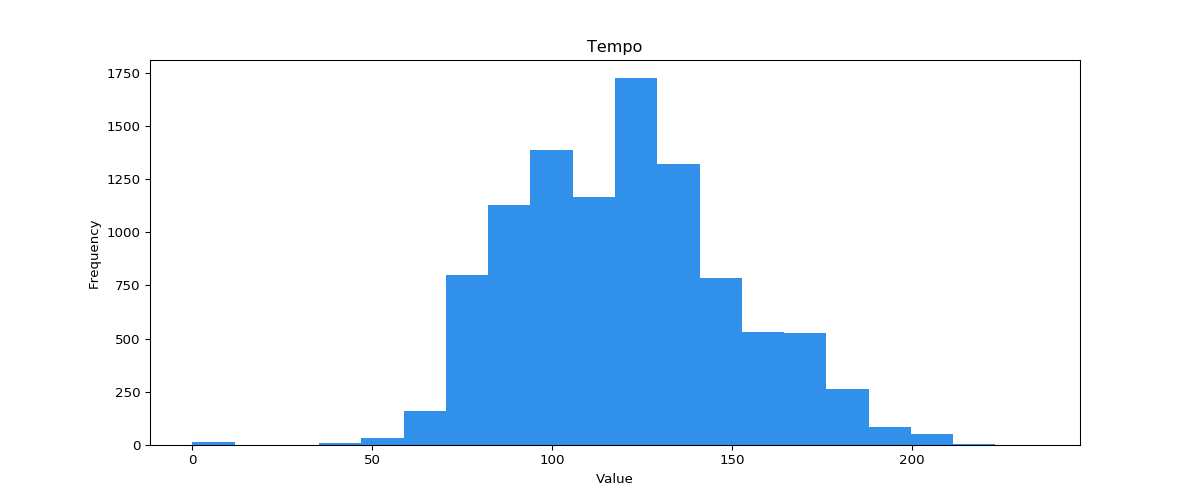
Data is needed to address the business problem properly. There are 4 data sets used that were obtained from the Kaggle website. They are:

1. Spotify Dataset (Ay, 2020)
2. Billboard Dataset (DeFoe, 2020)
   * Contains Billboard songs and Grammy songs
3. 500 Greatest Songs of All Time Dataset (Hany, 2020)

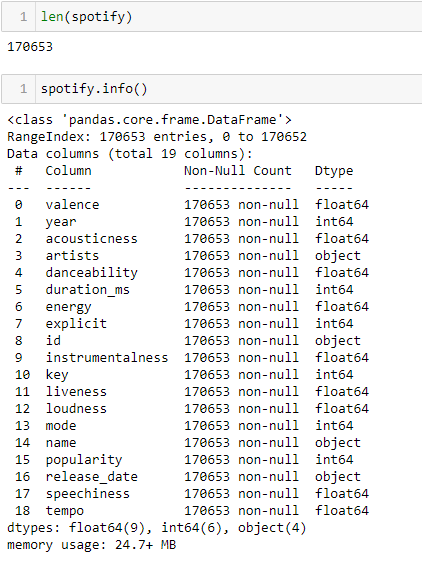
### Spotify Dataset

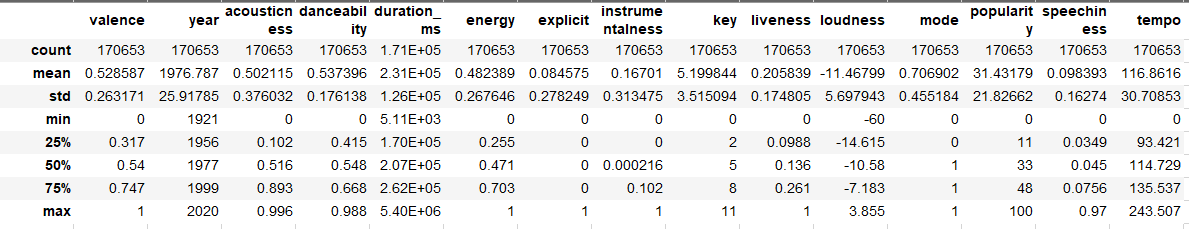
(Spotify, 2020) (Ay, 2020)

The first dataset is the main data the other datasets are joined to. It contains all aspects of a song as defined by Spotify API. The audio features included are:

* Duration\_ms
  + The duration of the song in milliseconds
    - The mean duration is 207,467 ms
* Key (Wikipedia, 2020) (MasterClass, 2020)
  + The estimated octave of the song mapped to an integer value know as pitch class
    - A pitch class is the set of all pitches that are whole number or octaves apart.
    - An octave in terms of music is the distance between one note the next note with the same name and in physics it is the distance between one note and another note that is double the frequency.
    - 0 is mapped to Octave C
    - 1 is mapped to Octave C#
    - 2 is mapped to Octave D
    - 3 is mapped to Octave D#
    - 4 is mapped to Octave E
    - 5 is mapped to Octave F
    - 6 is mapped to Octave F#
    - 7 is mapped to Octave G
    - 8 is mapped to Octave G#
    - 9 is mapped to Octave A
    - 10 is mapped to Octave A#
    - 11 is mapped to Octave B
    - The median key is 5 or Octave F
* Mode (Wikipedia, 2020)
  + Indicates the modality of a song, major and minor
    - The difference between the modes is major tends to be brighter and happier, whereas minor is the opposite
    - Major is mapped to 1
    - Minor is mapped to 0
* Acousticness
  + A measure from 0.0 to 1.0 determining whether the track is acoustic or not
    - 1.0 represents acoustic
    - Distribution – mean 0.5 and median 0.52
* Danceability
  + How suitable a song is for dancing and is based on tempo, rhythm, stability, beat strength, and overall regularity
    - 1.0 is most danceable
    - Distribution – mean 0.53 and median 0.55
* Energy
  + A measure of intensity and activity. Energetic songs feel fast, loud, and noisy. Dynamic range, perceived loudness, timbe, onset rate, and general entropy contribute to energy.
    - 1.0 is most energetic
    - Distribution – mean is 0.48 and the median is 0.47
* Instrumentalness
  + Whether a song has no vocals.
    - 1.0 is very likely to contain no vocals
    - Distribution – mean is 0.17 and the median is 0.00022
* Liveness
  + Whether there is an audience present.
    - 0.8 and greater have a high probability of being live
    - Distribution – mean is 0.21 and the median is 0.14
* Loudness
  + Loudness of the track in decibels (dB)
    - Loudness is the psychological correlate of physical strength and is a ratio between two number rather than a measurement itself
    - Ranges from -60 to 0
    - Going the full range is an increase of 1,000,000 time the power
    - Distribution – mean is -11.47 and the median is -10.58
* Speechiness
  + The presence of spoken words in a track
    - 0.66 to 1.0 are mostly words like a talk show
    - 0.33 to 0.66 contain both music and speech
    - 0.0 to 0.33 are just music
    - Distribution – mean is 0.98 and the median is 0.045
* Valence
  + Musical positiveness of the song
    - High valence sound happier and lower valence sounds more sad
    - Distribution – mean is 0.52 and the median is 0.54
* Tempo
  + Estimated in beats per minute (BPM) and is the speed or pace of a given song
    - Distribution – mean is 116.86 and the median is 114.73
* Artist
  + Name of the artist
* Name
  + Name of the song
* Explicit
  + Whether the song is explicit or not
    - 0 is mapped to not explicit
    - 1 is mapped to explicit
* Year
  + Year the song was released

The total length of the data set is 170,653 rows, 15 columns, and there are no missing values. There are a total of 9 float numbers, 6 integer numbers and 4 objects (strings, id, dates). The initial exploration of the data revealed that there is some cleaning to be done. The artists are wrapped in brackets and quotes (an example [‘artist’]), this will need to be removed to join the others dataset to this. The ID field is not very useful in this case and will need to be removed and there are 2 date fields representing the same information so one can be removed.





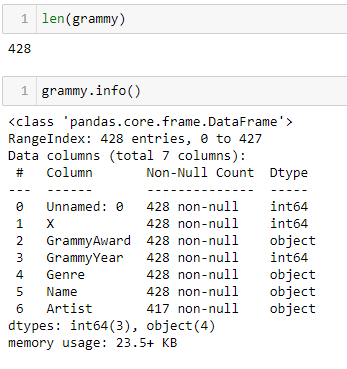
### Grammy and Billboard Dataset

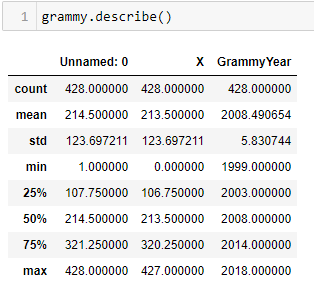
(DeFoe, 2020)

The next dataset is the Grammy songs from 1999 to 2019 and contains songs that have made it to the Grammy’s and receive a reward. The features are:

* Unnamed: 0
  + Looks to be an ID column
* X
  + Look to be an ID column
* Grammy Award
  + The type of award the was presented
    - Record of the year, song of the year, etc.
* Grammy Year
  + The year the award was given
* Genre
  + The genre that the award was given in
* Name
  + Name of the of the song
* Artist
  + Name of the artist

The Grammy dataset has 428 rows of data and 7 columns. There are 11 rows which have no value in the artists column. The dataset has 3 integer features and 4 object features.

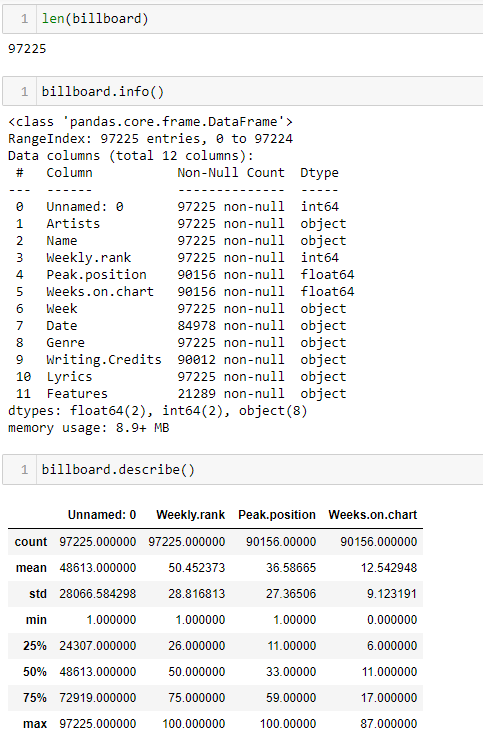


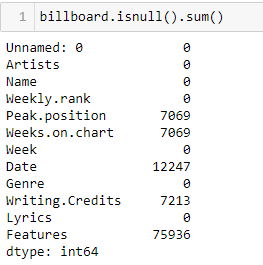


The Billboard dataset is a list of songs that were on the Billboard hot 100 songs chart. The features are:

* Artist
  + Name of the artist
* Name
  + Title of the song
* Weekly.rank
  + The rank that song made in each week
* Peak.position
  + The topmost position the song hit on the charts
* Weeks.on.chart
  + The number of weeks the song spent on the charts
* Week
  + The week in question for the above Weekly.rank feature
* Date
  + The date of the song was release
* Genre
  + The type of genre the song is considered to have
* Writing.Credits
  + The people who wrote the song
* Lyrics
  + A very incomplete field that was supposed to house the lyrics of each song
* Features
  + The artists that are featured in the song

The length of the billboard dataset is 97,225 rows of data in 12 columns consisting of 2 floats, 2 integers, and 8 objects. There is a significant amount of null that will need to be dealt with.





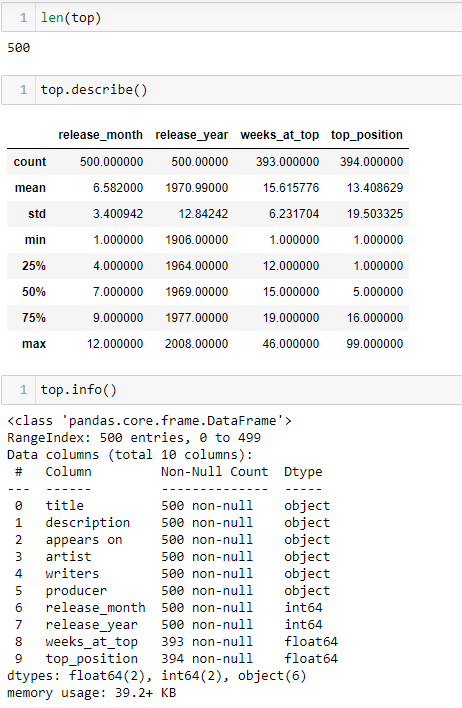
### Top 500 Greatest Songs of All Time Dataset

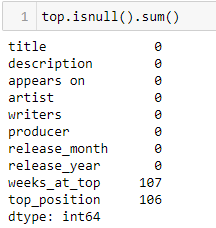
(Hany, 2020)

The final dataset used is from the Rolling Stones definitive list of the 500 greatest songs of all time. The features are:

* Title
  + The name of the song
* Description
  + Description of the song
* Appears On
  + The album of the artist that the song appeared on
* Artist
  + Name of the artist
* Writers
  + Name of the people who wrote the song
* Producer
  + Name of the person who produced the song
* Released
  + The month and year when the song was released
* Streak
  + How long the song spent on the charts
    - The mean is 15.62 and the median is 15 weeks at the top
* Position
  + The highest position achieved by the song
    - The mean is 13.41 and the median is 5 as the top position

The length of this dataset is 500 rows with 10 rows consisting of 2 floats, 2 integers, and 6 objects. There are some null values in a couple of the columns.





### Finishing Up Data Understanding

In total there are 4 datasets that need to be cleaned and coerced into a position that can be joined to one another. This first task is to get the ‘Artist’ and ‘Name’ columns into the same format across all datasets. There are some special characters in these columns that will need to be removed as well. The next column change only refers to the Top 500 dataset and that is to change the ‘weeks\_at\_top’ to ‘Weeks.on.chart’ and ‘top\_position’ to ‘Peak.position’ so that they can be joined to the billboard data set. Some aggregation of the various measure will be helpful to make the datasets more normalized. Finally, there are some columns that need to drop all together as they make no sense.

## Data Preparation

This phase will focus on the proper cleaning and joining of the dataset to get them ready for modelling. The columns will need to be in the same format as well as the values contained within. The null values need to go so they will not throw off the calculations and aggregation of some measure columns will be done to return only distinct values for each row and feature.

### Spotify Dataset

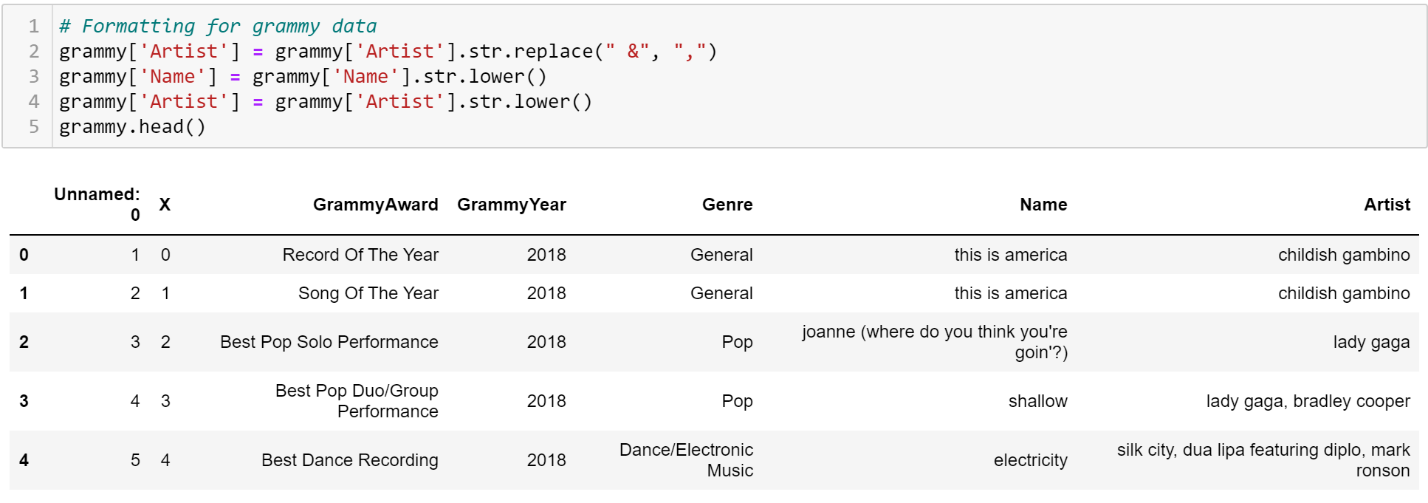
One of the things that came up in the exploration phase was that the artist columns was wrapped in brackets and quotes on both ends ([‘artist’]). That is easy to get rid of in python with the ***str.strip()*** and the ***str.replace()*** methods. The strip method was called to take the brackets off because the served no purpose and the replace method was called to replace all single quotes with nothing and all “&” signs with a comma. The names of the ‘name’ and ‘artist’ columns were changed to titles in place and the contents of both columns had the ***str.lower()*** method called on them to turn the contents into all lower case.

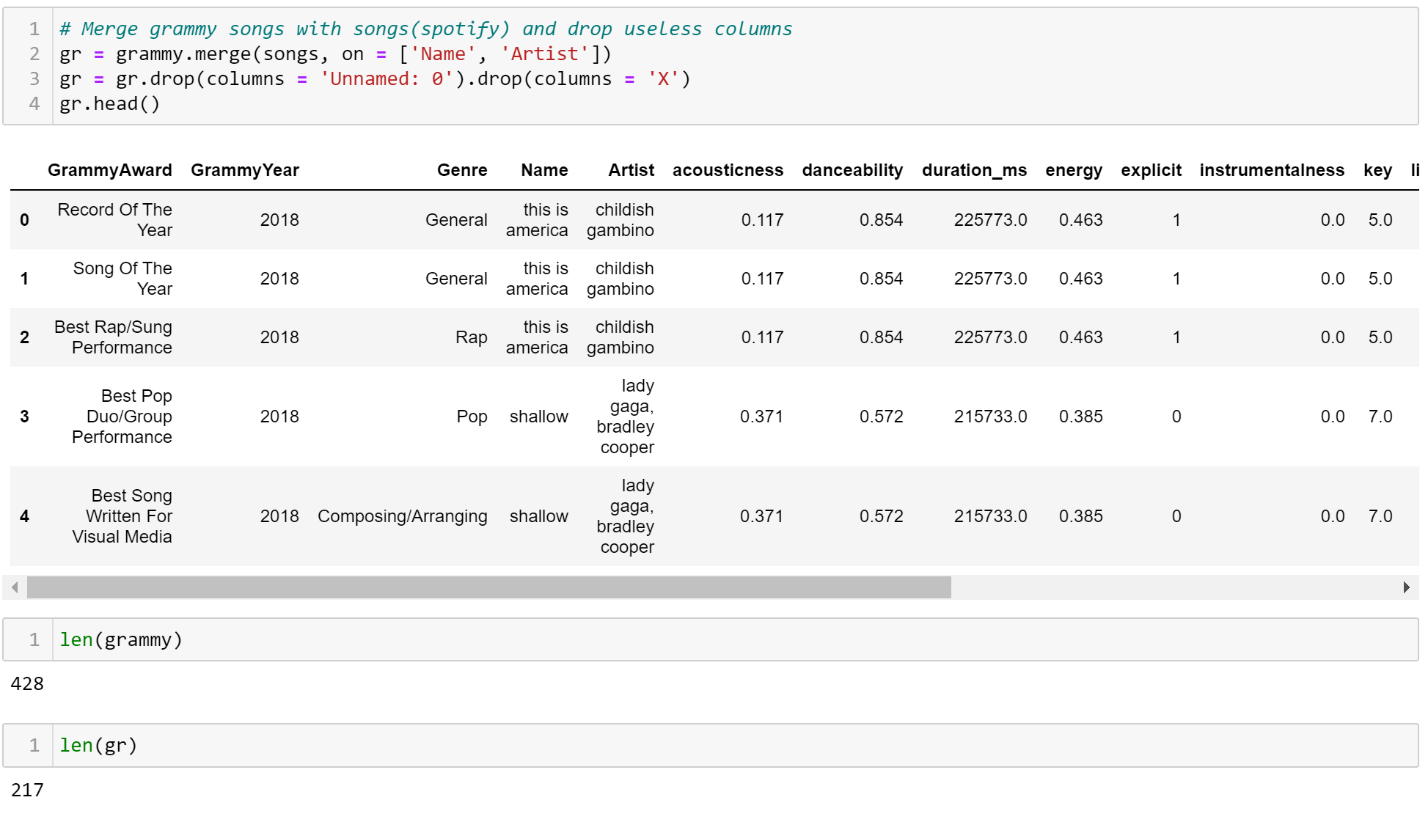


The next task was to aggregate all the measures to make a more coherent table and to get rid of any duplicate values. The ***groupby()*** and ***agg()*** methods were called on the Spotify object and then stored as another object called songs. This grouped the data by ‘Name’ and ‘Artist’ and then took a dictionary of key: value pairs as the arguments. This would find each instance where ‘Name’ and ‘Artist’ is the same then apply the specific aggregation supplied. This is to ensure that there are no duplicates in the data so that it will not fan when joined with other datasets. This also reduced the number of rows from 170,653 to 157,303 or about 13,000 values.

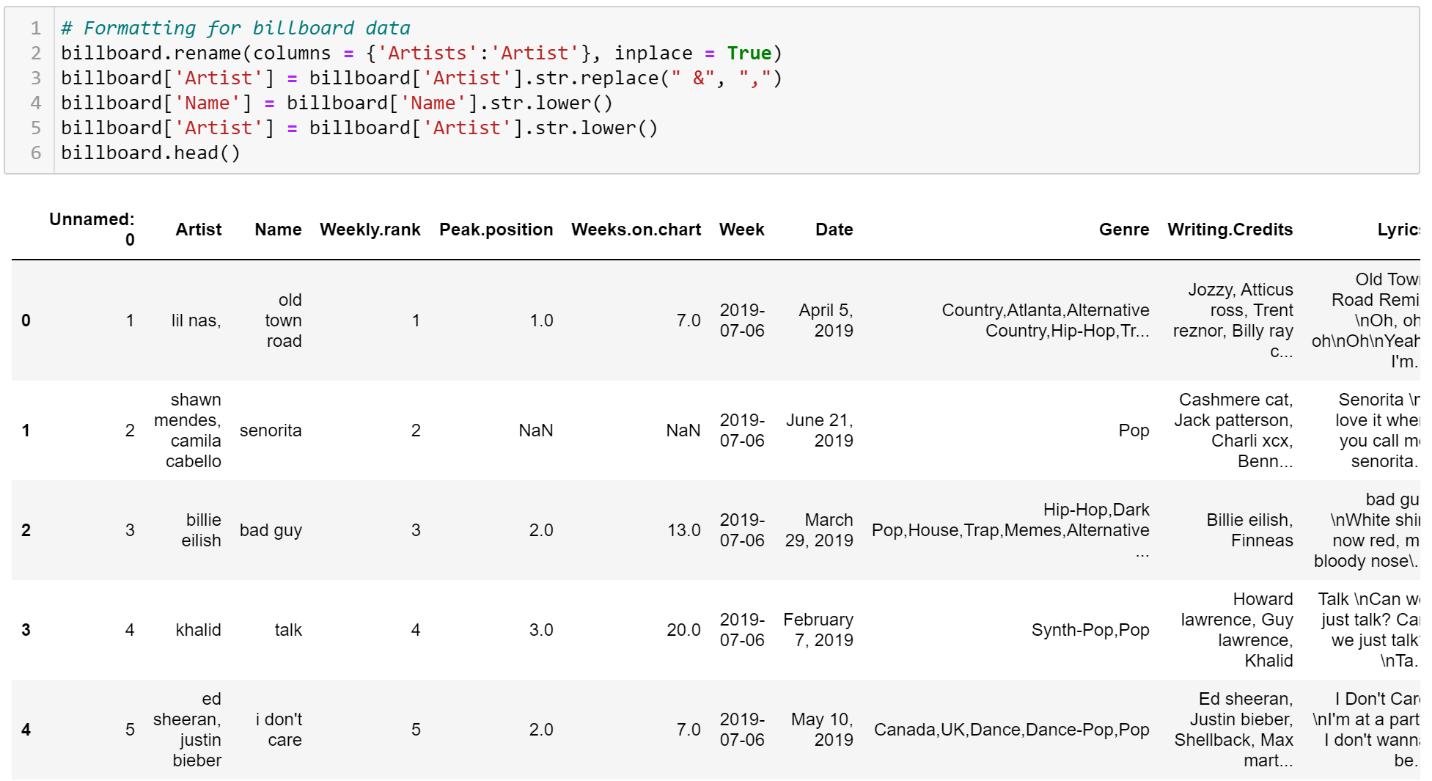
### Grammy and Billboard Dataset

A similar cleaning method was applied to the Grammy and Billboard datasets. For the Grammy data the “&” character was replaced with a comma. Then the ***str.lower()*** method was called on both ‘Name’ and ‘Artist’ to make it lower case



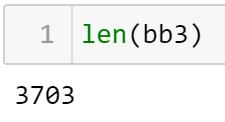
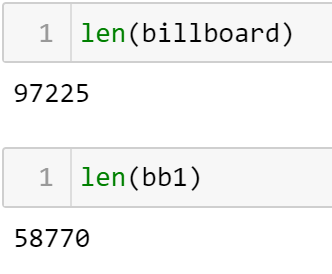
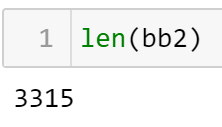
This put the structure of the Grammy data into the same as the Spotify data for merging. The next step was to merge the data, so the ***merge()*** method was called in the Grammy variable and merged with songs on ‘Name’ and ‘Artist.’ Then the unnecessary columns were drop from the merged dataset. The merge dropped the length of the Grammy data from 428 to 217, but now it contains the song attributes of the songs that have received Grammy’s.

For the Billboard dataset the same methods used previously were applied here also. ‘Name’ was already capitalized here, but ‘artist’ still needed capitalization. The “&” symbol was replaced with a comma and then ‘Artist’ and ‘Name’ values were lower cased.

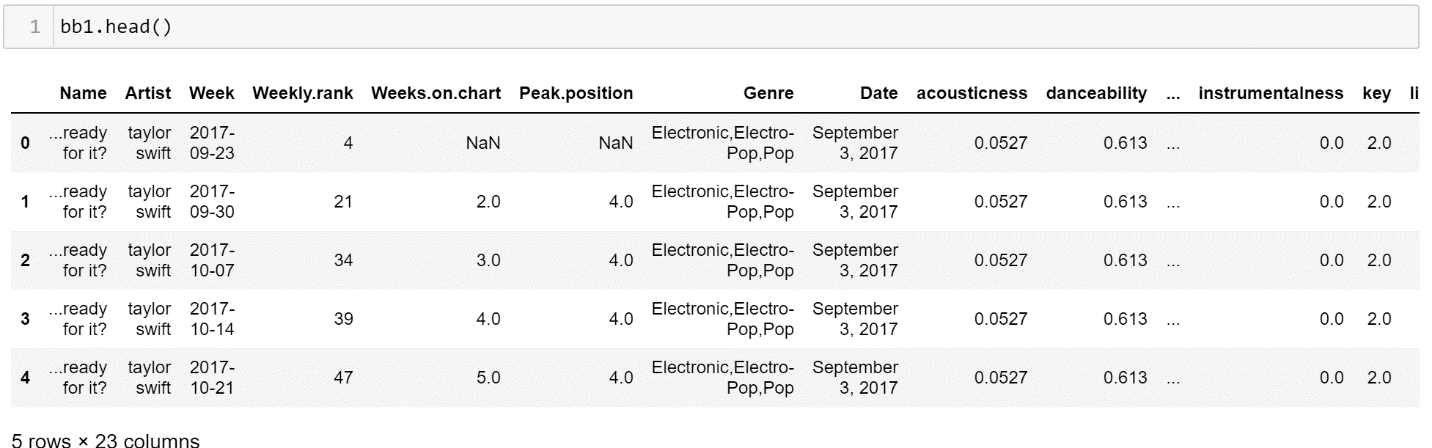


Once the initial changes were made, billboard variable was treated similarly to the Spotify variable. Here bb1 was created by calling the ***groupby()*** method on billboard and grouping on ‘Name.’ ‘Artist,’ Week,’, and ‘Weekly.rank.’ The ***agg()*** method was called on the remained columns to get rid of duplicates, then bb1 was merged with songs.  The bb2 variable was created by grouping on ‘Name’ and ‘Artist’ and aggregating only ‘Week.on.chart’ and ‘Peak.position’ along with dropping the NA values by calling the ***dropna()*** method. The variable bb3 is made the same way as bb2 except bb3 only shows the aspects of the song and not the position or how many weeks the song spent on the chart. The bb1 variable is considered a bridge table to obtain bb2 and bb3.

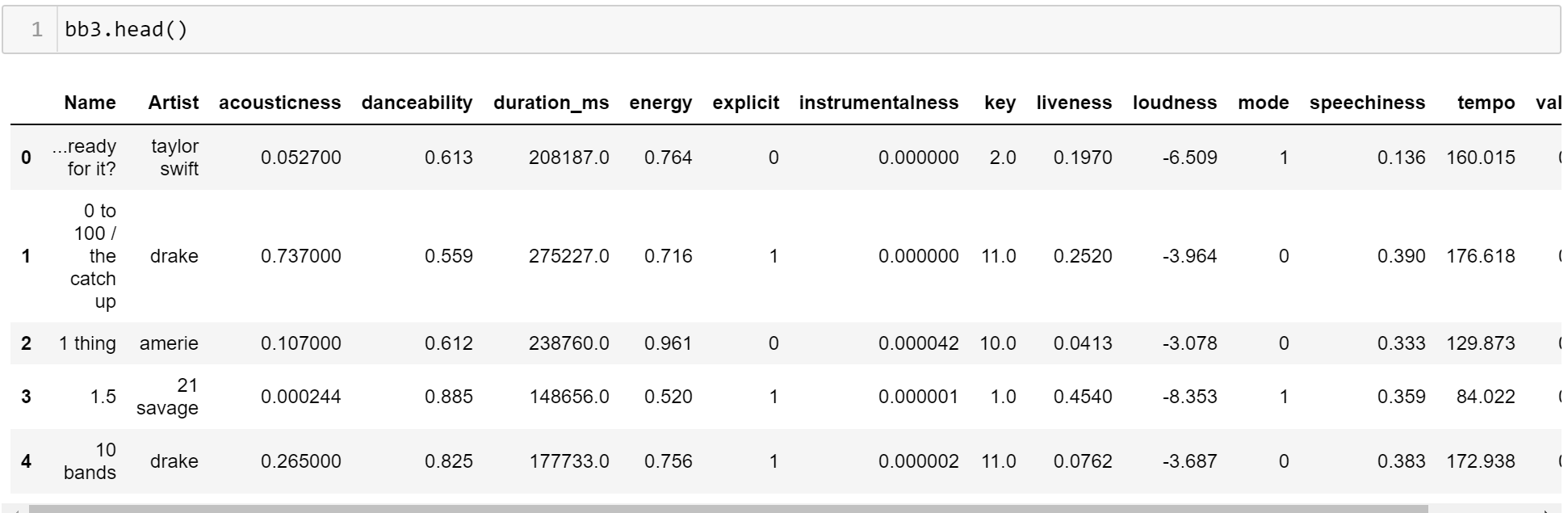
The creation of bb1 reduced the number of rows from billboard. Billboard had 97,225 rows and bb1 only contains 58,770. The length of bb2 and b33 are 3,315 and 3,703, respectively.



The finished product of bb1, bb2, and bb3 look like the following in order.

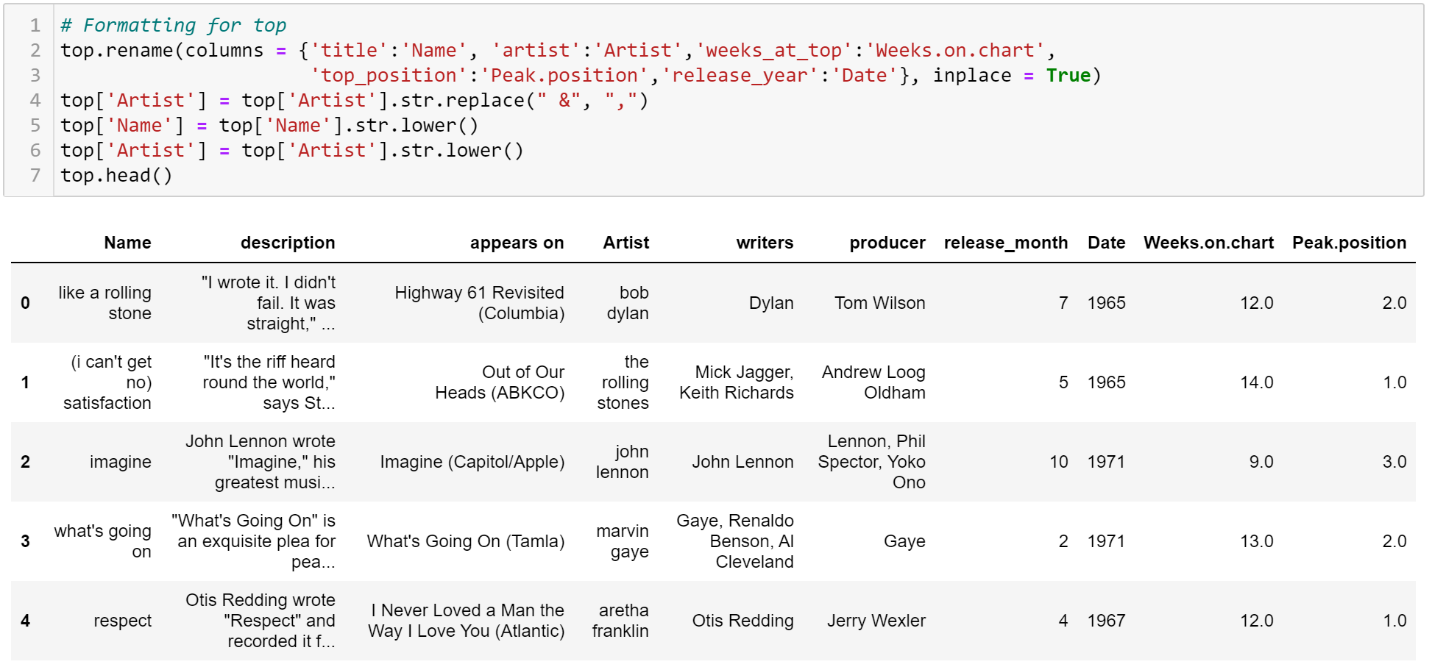




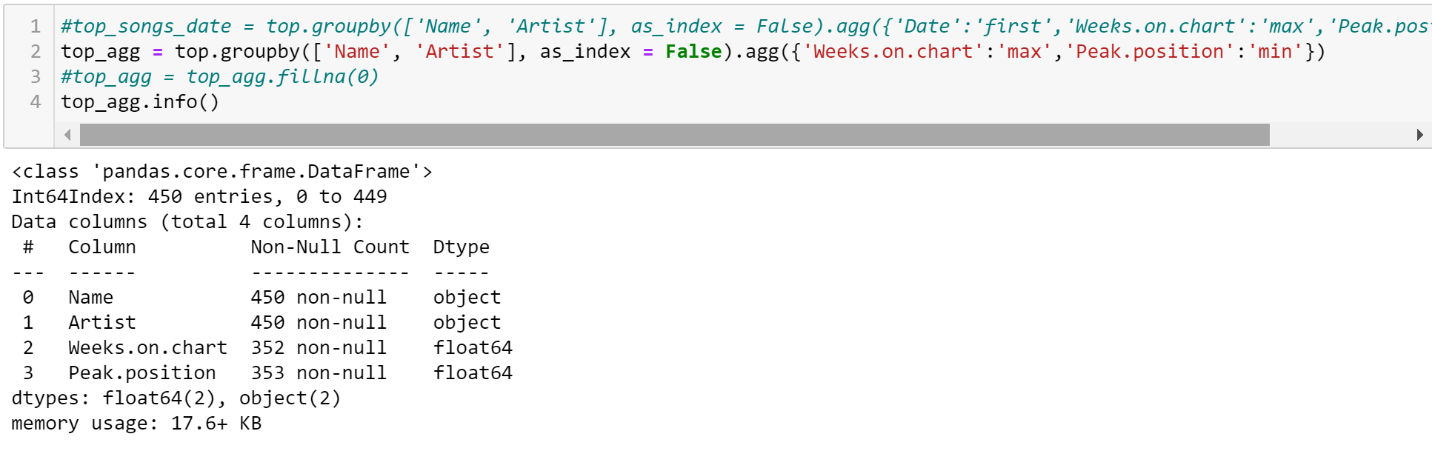


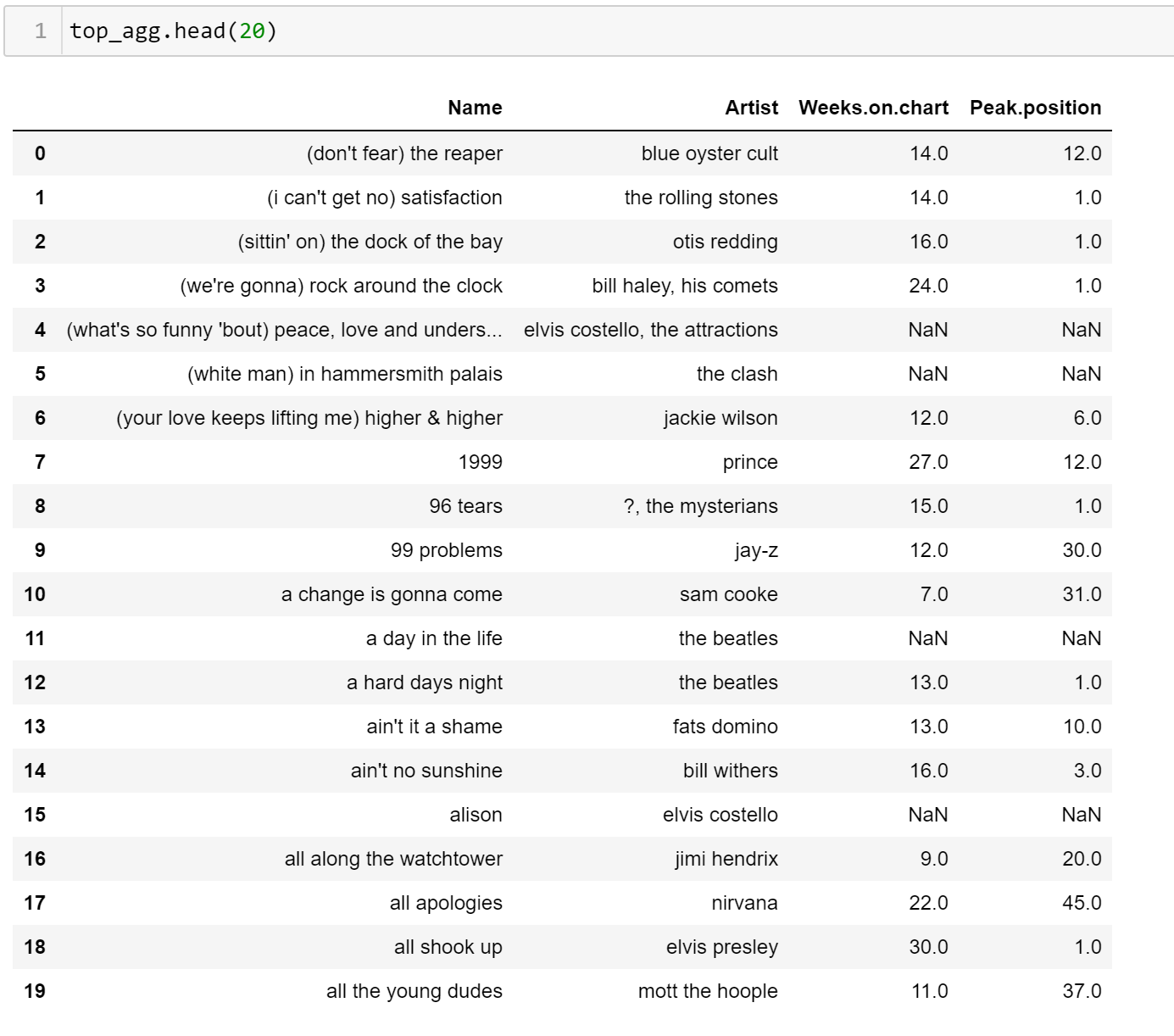
### Top 500 Greatest Songs of All Time Dataset

Since the datasets are all being joined together the same cleaning was performed again. The columns were renamed to reflect the other datasets column names, “&” was replaced with a comma, and the values of ‘Name’ and ‘Artist’ were lower cased.

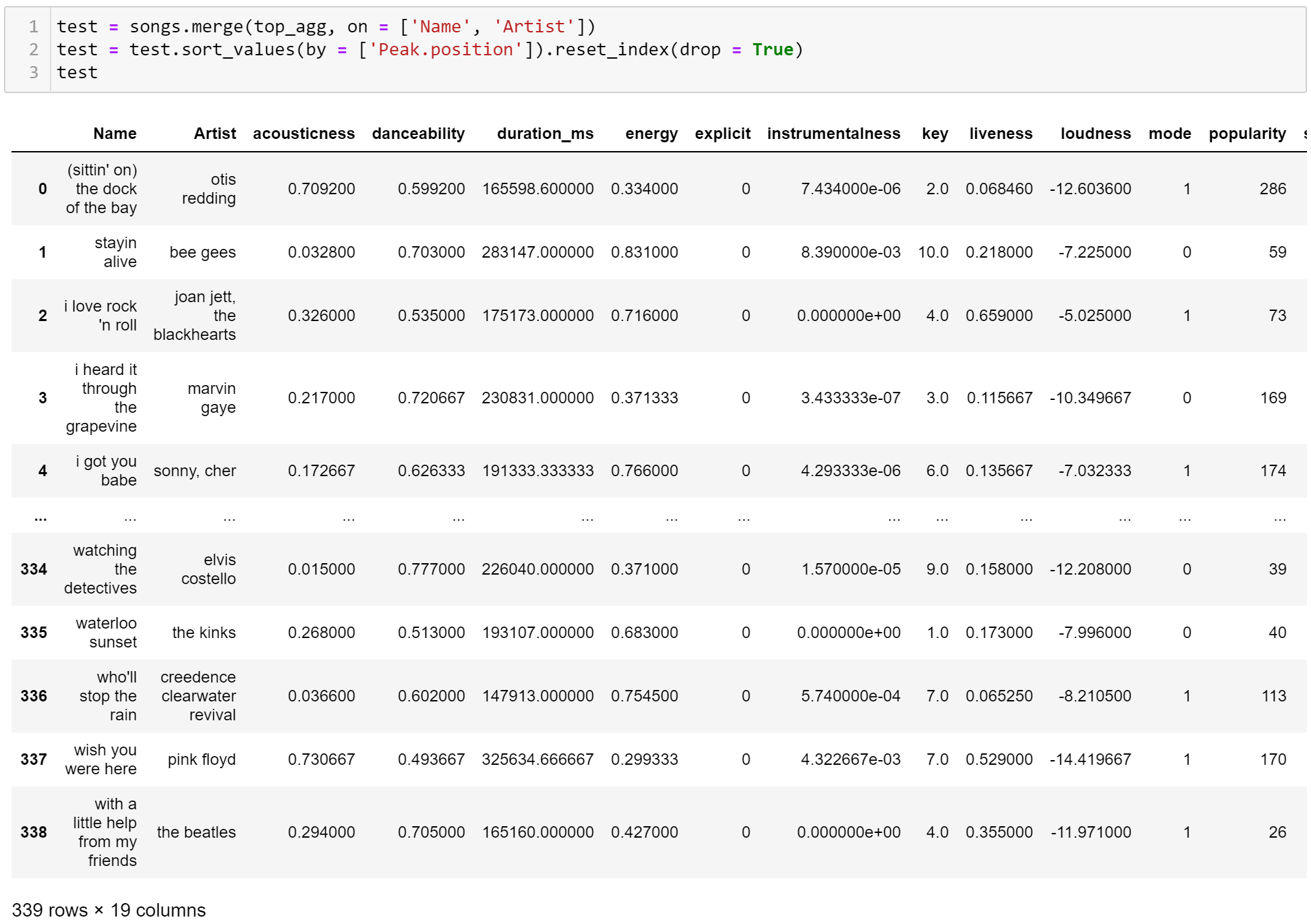


The top variable has some unneeded columns. To remove those columns ***groupby()*** method was called on top and the ***agg()*** method was used to the remaining measures. This produced some NA values which will be dealt with later by using the ***fillna()*** method.





When merging songs with top\_agg it produced and out of order dataset, so to remedy this the ***sort\_values()*** method was called on the variable and sorted by ‘Peak.position’ then the  ***reset\_index()*** method



### Further Cleaning and Merging

After all the above was cleaned and moved into the modelling phase, I had realized that I need more concise datasets. The below code is showing the new variable that were created by further merging other datasets together with ones that were products of datasets already merged. Top\_songs, top\_grammy, top\_bb1, top\_bb2, and top\_bb3 are used in the modeling process. The below code show that the NA values in top\_songs were all filled with 0, unnecessary columns were dropped, and the average of both x and y versions of ‘Weeks.on.chart’ and ‘Peak.position. There was 1 NA value in the ‘Weeks.on.chart’ and ‘Peak.position’ columns in top\_grammy, top\_bb2, and top\_bb3 the values were imputed accordingly with 20 and 76.

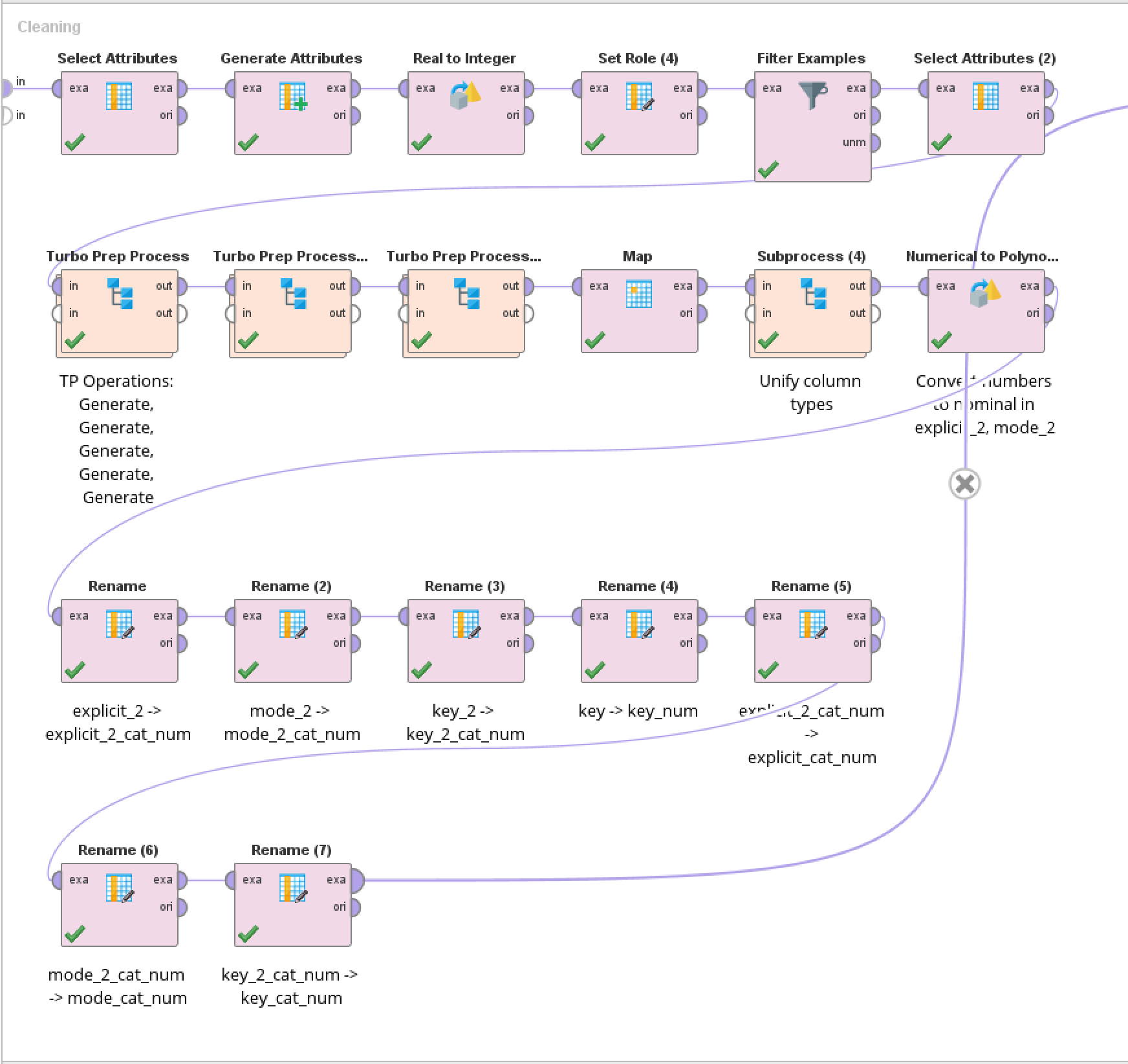
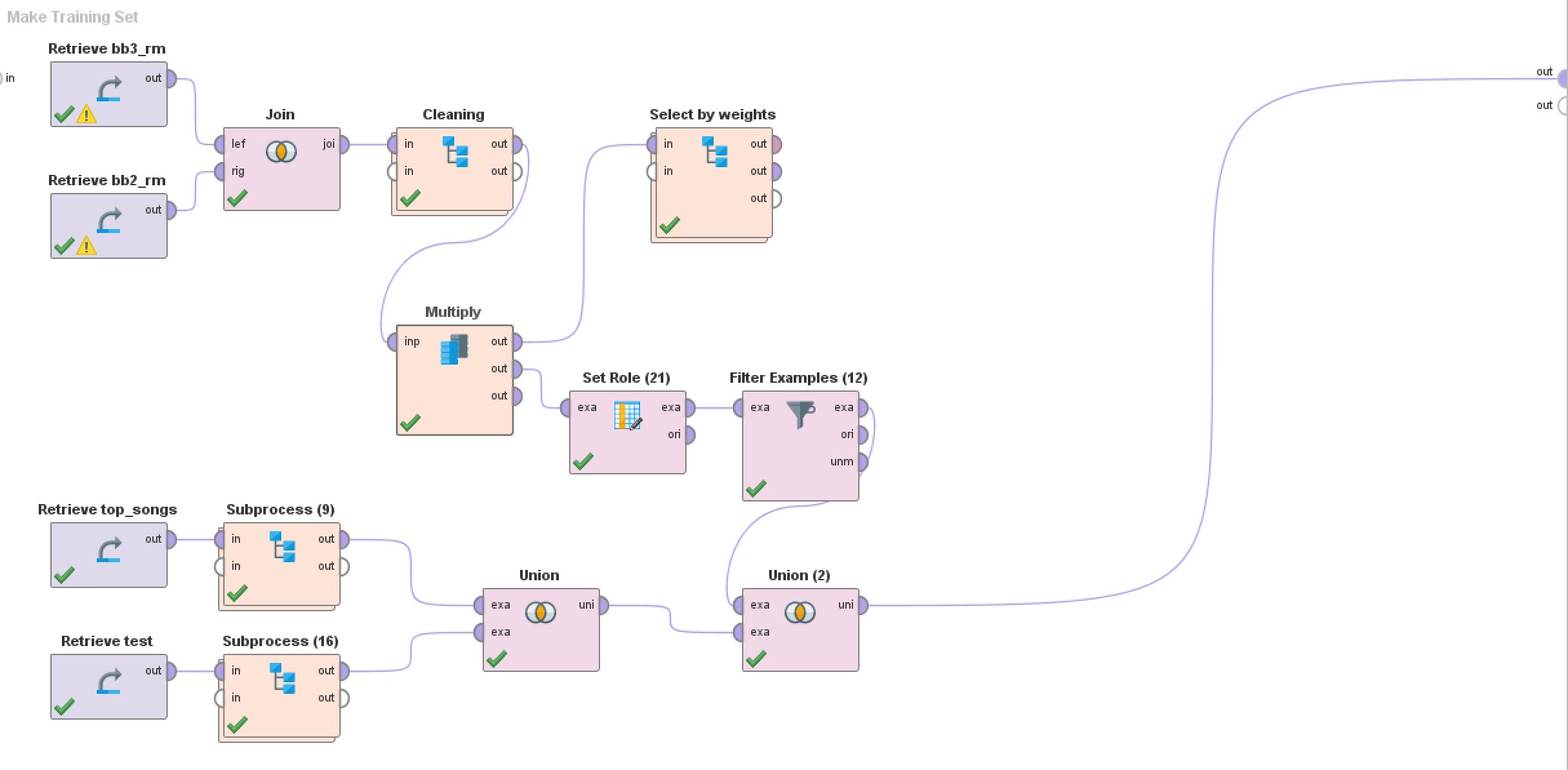
Once everything has been merged the ***str.title()*** method was called on ‘Name’ and ‘Artist’ to return it to title case and where necessary loudness was normalized by dividing the values by 60 and adding 1.

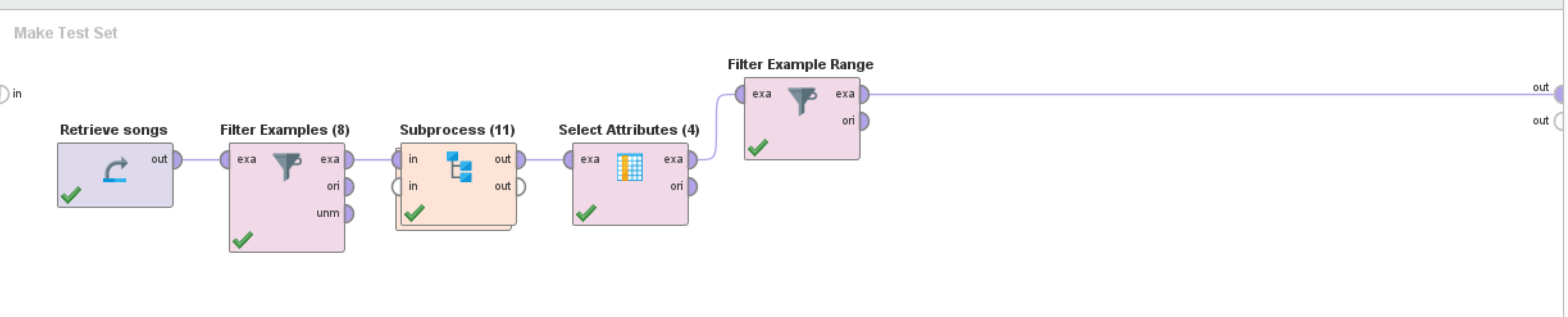


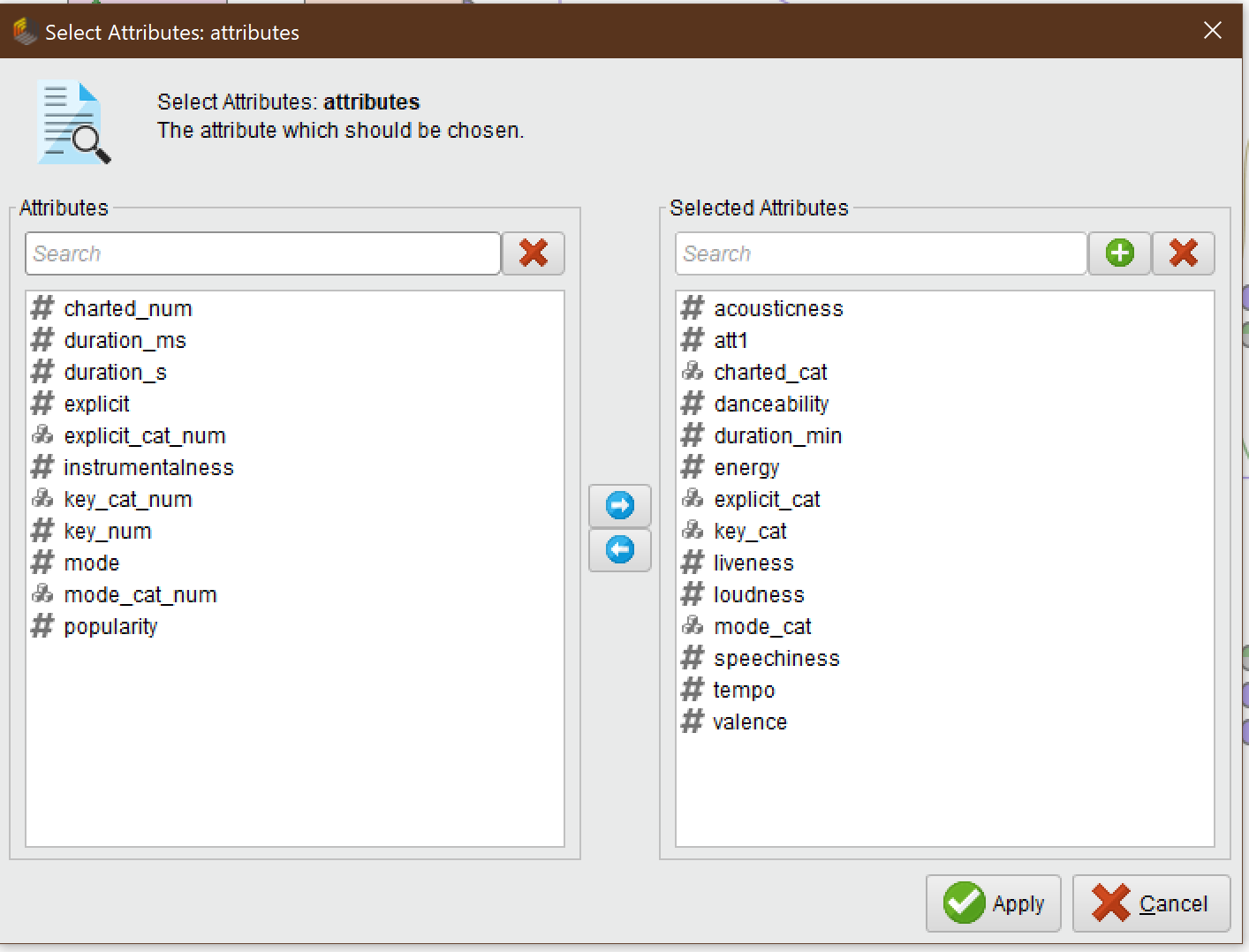
This is the end of the data preparation phase in python. All the above have been exports as csv files and are imported into RapidMiner to do slightly more difficult cleaning. Bb3, bb2, top\_songs, test, and songs were all imported into RapidMiner.

Starting with the inner most process. This subprocess is called cleaning and is used multiple time across the whole process. This process creates several new variables:

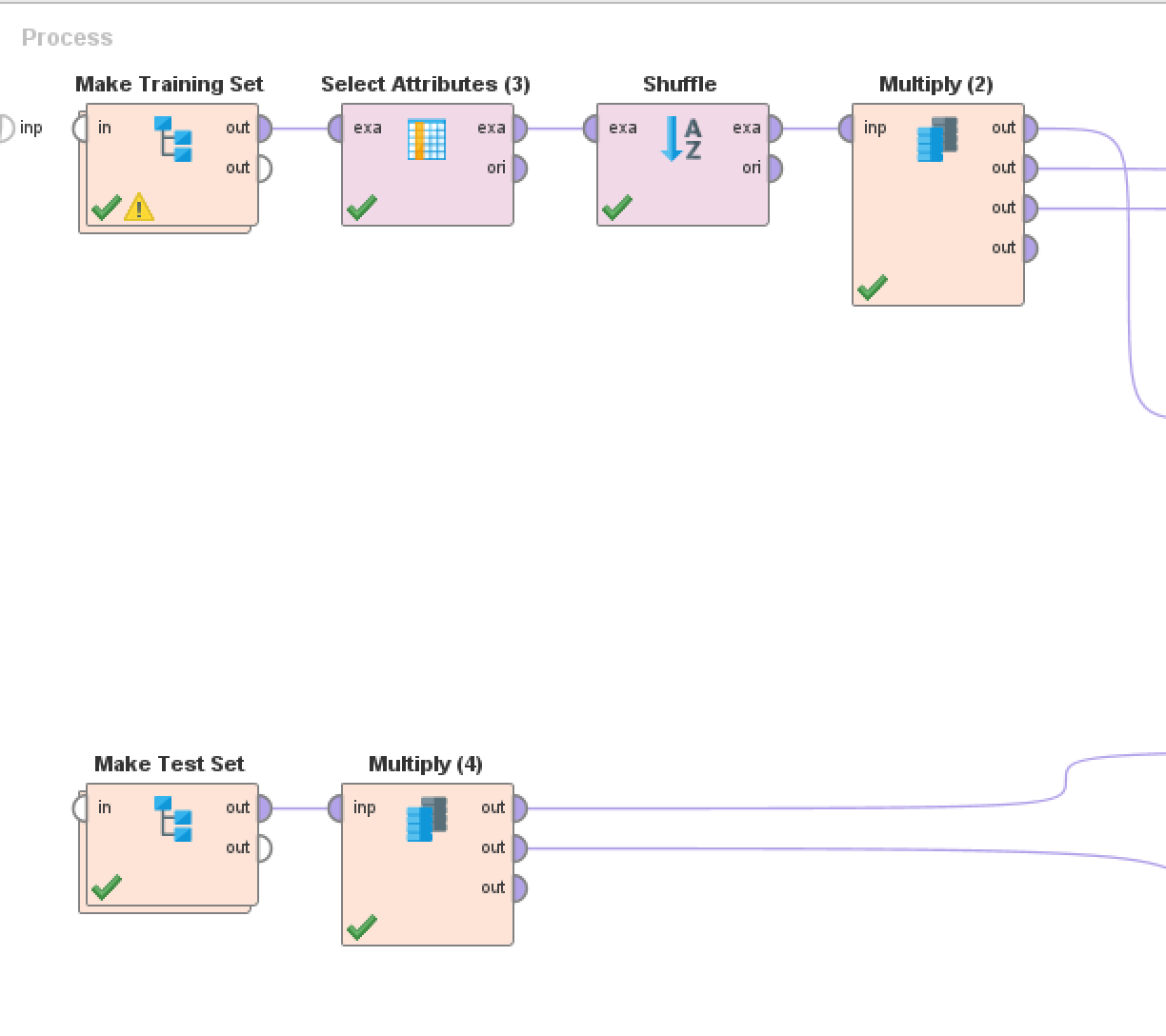
* Charted\_cat and charted\_num
  + This is the label and is what is predicted by the model
  + Charted\_cat values are Yes and No and charted\_num values are 1 and 0
  + Both are coded as a categorial variable
* Duration\_s and duration\_min
  + Created from the duration\_ms variable by dividing by 1000 then again by 60 respectively
* Explicit\_cat and explicit\_cat\_num
  + Determined from the explicit variable
  + Explicit\_cat values are Yes and No and explcit\_cat\_num values are 1 and 0
  + Both are categorial variable
* Key\_cat and key\_cat\_num
  + Calculated from the key variable
  + Key\_cat has values code as the octave they reflect (i.e. octave g and octave g#)
  + Key\_cat\_num has the values coded as an integer from 0 to 11
  + Both are categorical veriables
* Mode\_cat and mode\_cat\_num
  + Calculated from the mode variable
  + Mode\_cat has a major and minor value and mode\_cat\_num has a 0 and 1 value
  + Both are categorical variables

The cleaning process also renames the variables to be more uniform. Zooming out a step from the cleaning process is the make training set process. This step joins together bb2 and bb3 then cleans it via the above process, the att1 and charted\_cat is set as ID and label, respectively, and then the “No” values are filtered out. Top\_songs and test are also cleaned via the same process and have their labels set and then union together. Both merged datasets are further union together creating a dataset with 1,066 rows consisting of “Yes” and “No” values.

Making the test set follows a similar process with a couple added steps. The process loads the songs csv and filters data down to a date range of 1950-2000. It is then cleaned as above. This results in about 70,000 rows but then uses a range to only return 1000 rows. Combining both process looks like the following. Noticing that the make training set process and the make test set process are the base. The attributes selected are below, then the results are shuffled so that the “Yes” and “No” variables are more spread out before modelling.



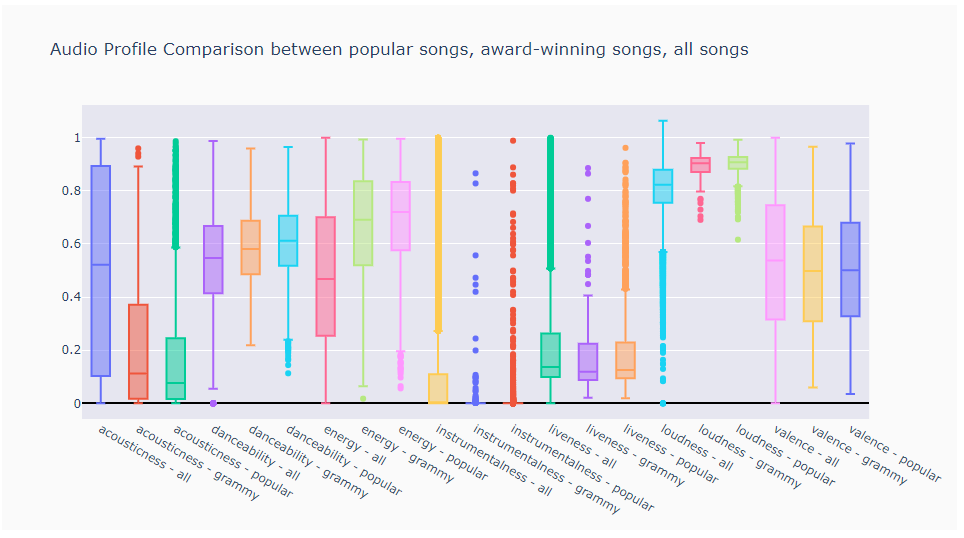
The below picture is top level overview of the whole merging and joining process in RapidMiner

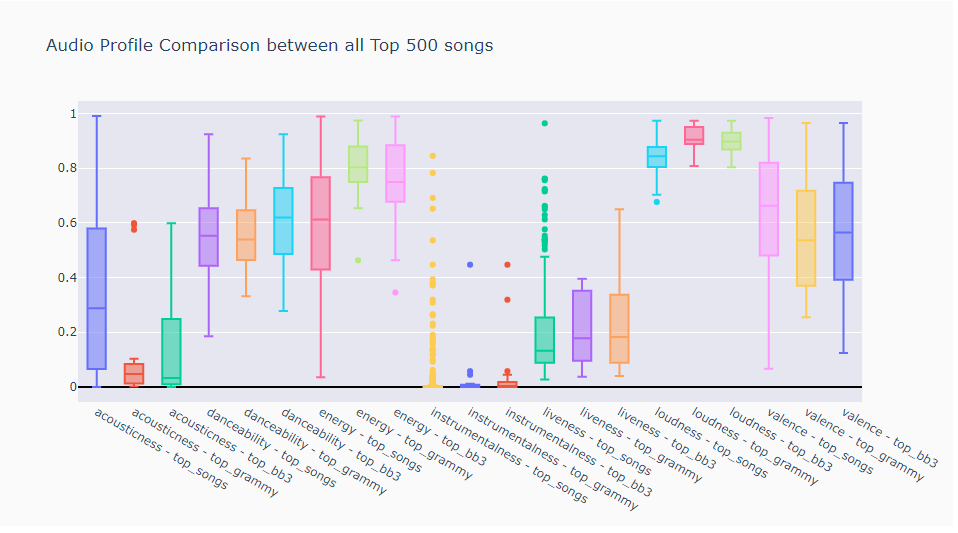


### Data Exploration

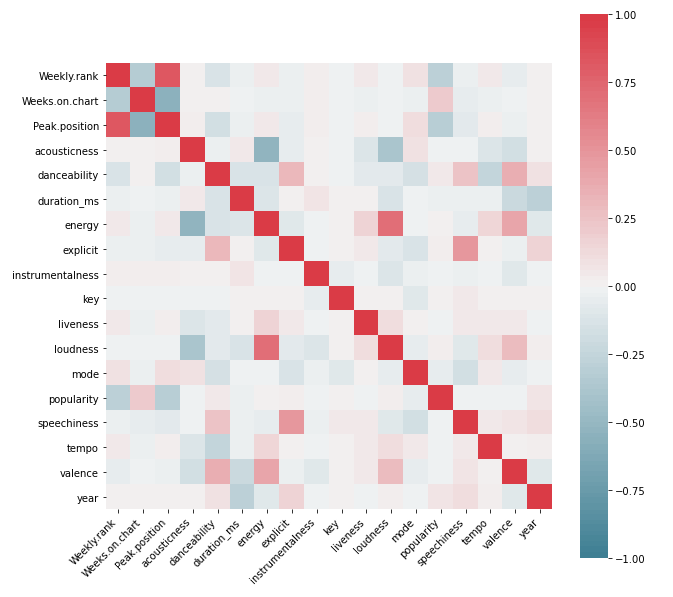
The cleaning is all done, and the data is at a spot that allows for some further data exploration. A boxplot is a great way to show the spread of multiple variables simultaneously. A boxplot is a graphical depiction of groups of numerical data that will also show outliers for each plotted variable. The graph displays all the song aspect from the Spotify data, Grammy data, and billboard data.

In general, there are trends between each of the song aspects. It is noticeable that there is a clear difference between all songs and the Grammy and popular songs. Acousticness has a huge interquartile range and from that we can deduce that there are not very many acoustic songs that make it on the top charts. All the other song aspects seem pretty matched up in comparison.



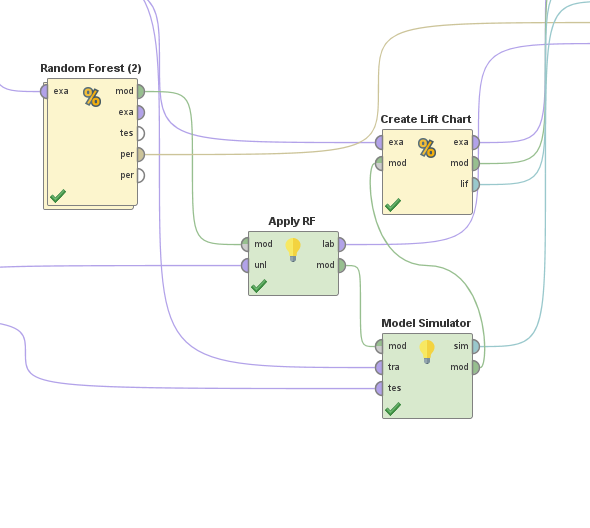
I played around with this charts style and was able to uncover something else quite interesting. This chart below is the same type of boxplot, but the only difference being that it uses all the datasets merged with the top 500 songs of all time data. You can see acousticness still has quite a large IQR. What is quite interesting here, is that there are virtually no instrumental songs on the top charts. The case can be made that there IS no instrumental song on the top charts because technically they are all outliers. For the final product, the outliers were not removed.

I produced a couple heatmap as well. Something interesting I discovered on the below heatmap. The top left corner of the heat map is where all the action is. Peak.position has a high positive correlation with Weekly.rank, but a high negative correlation with Weeks.on.chart. I took this to mean that a high weekly rank would lead to a high overall position while the higher the position the lower the number of weeks the song will spend on the chart.



## Modeling

Although I do enjoy the data preparation phase because it is so thrilling to see it all come together in clean and tidy way, the modeling phase has its positive aspect as well. In this part I will explain more about the model that was created to predict whether the song charted or not. RapidMiner was solely used for this portion.

As you can see in the picture below, I chose to build a random forest model. What is not pictured are the many iterations done before coming to this conclusion. Deep learning, generalized linear model, and naïve bayes were all tried, but random forest prevailed. A 10-fold cross validation method was used for training. Cross validation holds out one subset of data for each fold and runs the model 10 times. It then averages the measure of fitness in prediction and returns and accurate model the generalizes well to unknown data.

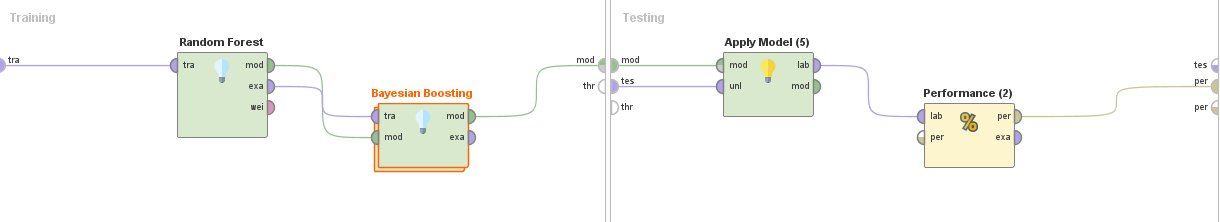
There are some performance metrics that will be used in determining model accuracy that should be defined before explaining the model. Accuracy will be used in the context of the whole and not just by itself. This is an imbalanced dataset and accuracy could be potentially misleading. The equation for accuracy is

Recall is the true positive rate of the model and this is the calculation

Precision is the positive predictive value the tells the proportion of true positives. The calculation is

The area under the curve, or AUC, if the probability that a classifier will fall under the curve. Baseline AUC is 0.5 means random guesses and AUC 1.0 is a perfect prediction.

Now that those are defined, we can move on to assessing the model. In the random forest cross validation operator there is a training and test side. The training side holds the model and then the testing side applies the model and calculates performance. There is model booster in here for greater accuracy that will be explained in a moment. Without the model booster and default settings the accuracy of the model is 83.87%, precision is 83.98%, recall is 92.34% and the AUC is 0.92. All these numbers translate to a decent model, but I was able to squeeze out a little bit better accuracy.

Tweaking the model slightly, I increased the max depth from 10 to 12. This made the model better slightly. Accuracy is 85.15%, precision is 84.68%, recall is 93.66, and the AUC is 0.929. This is probably a good stopping point, but I wanted to see if there was a way to further increase the accuracy without overfitting.

In RapidMiner there is an operator called Bayesian boosting (RapidMiner, 2020). It is an operator that requires a learner model inside it. Bayesian boosting works by reweighting the training set for each iteration so previous patters discovered are “sampled out.” The inner learner is applied several times and then combined globally. This operator has a couple parameters as well. The rescale label prior parameter is selected meaning that all classes are equally probably for each iteration. Allow marginal skews parameter is selected as well meaning that marginal weights/probabilities of the subsets are changed and makes the two classes equally likely to happen. The learner inside the booster is a default deep learning model.

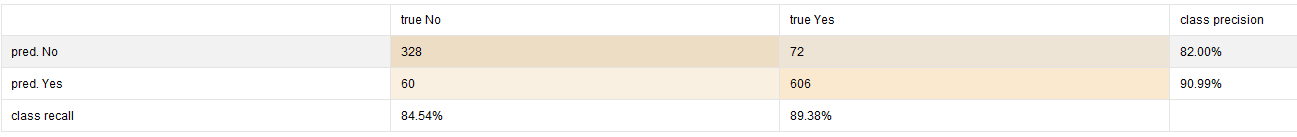
With the small tweak to the random forest model and the booster activated the model accuracy becomes 87.62%, precision is 91.04%, recall is 89.38%, and the AUC is 0.933.

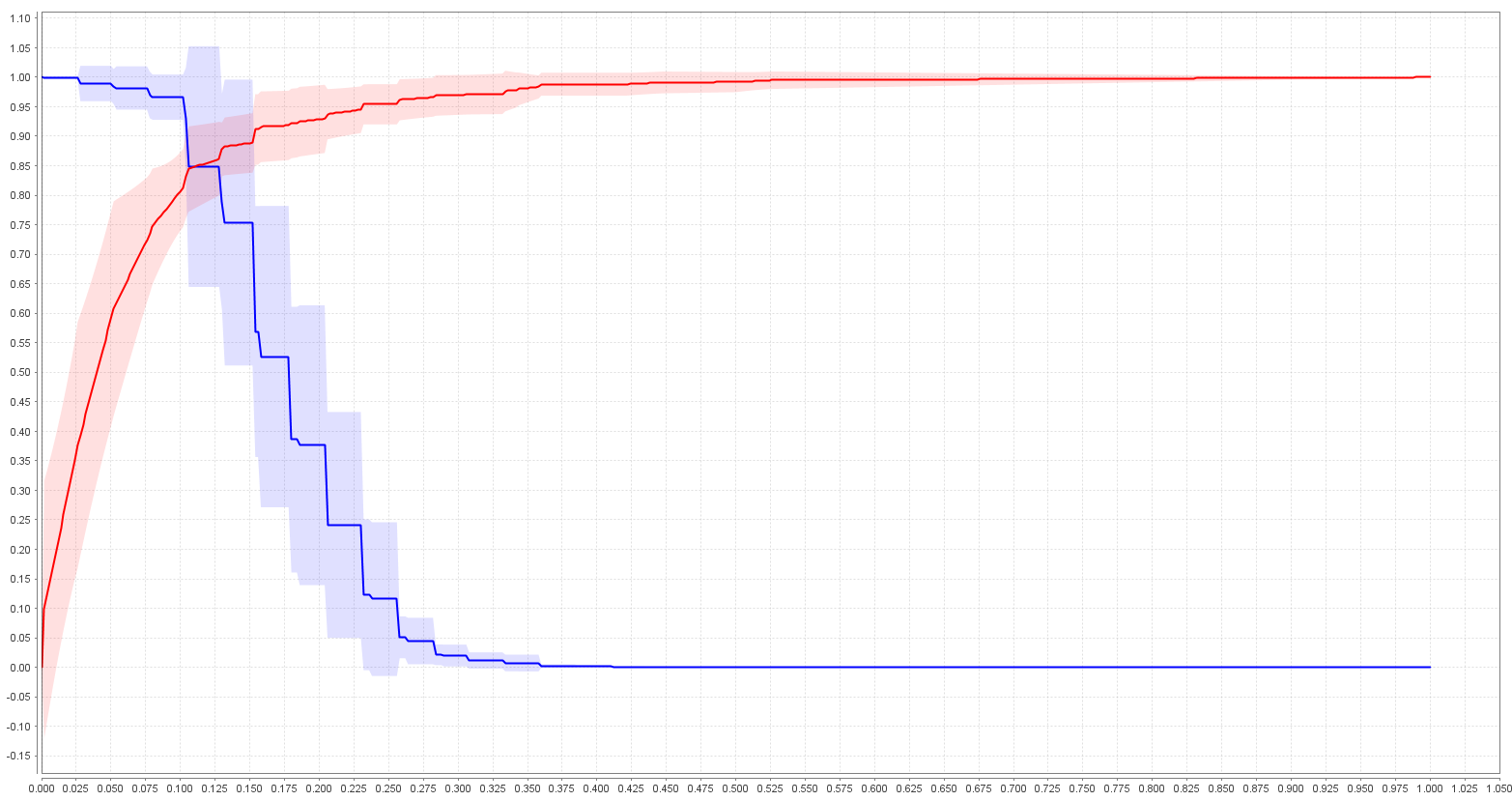


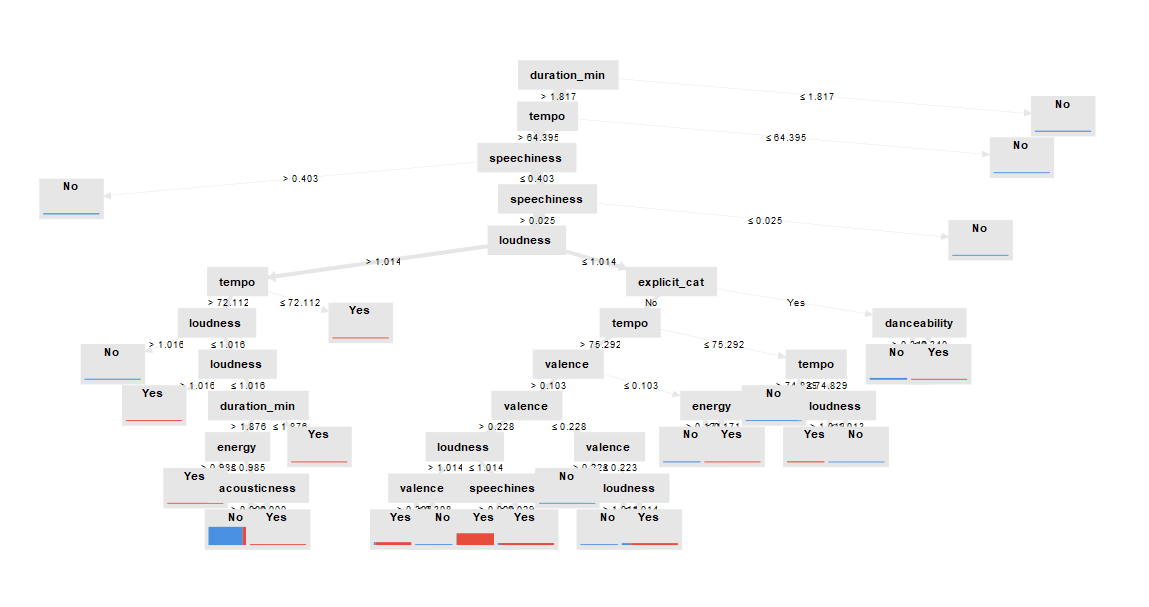


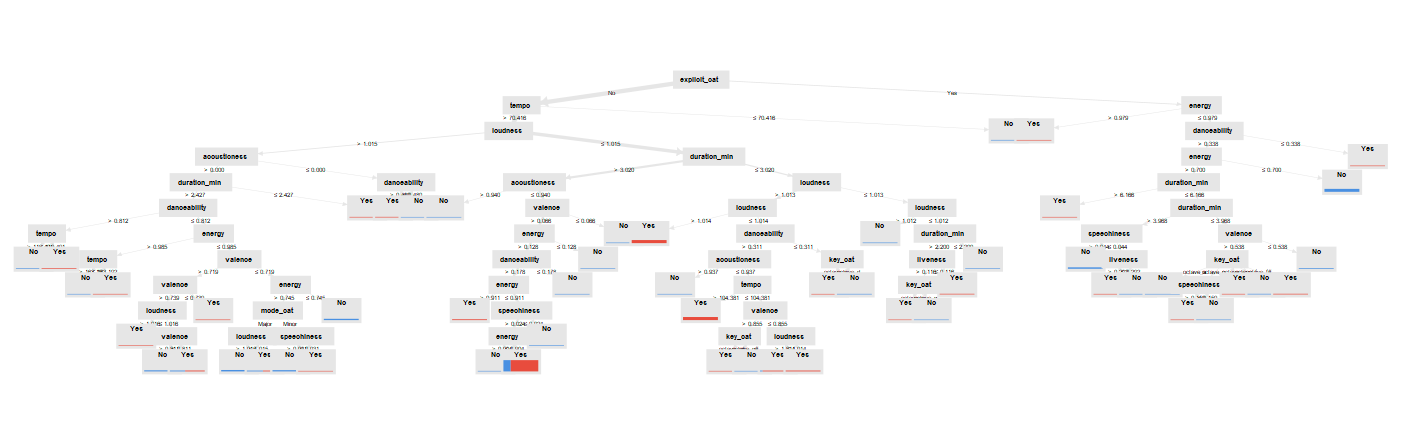




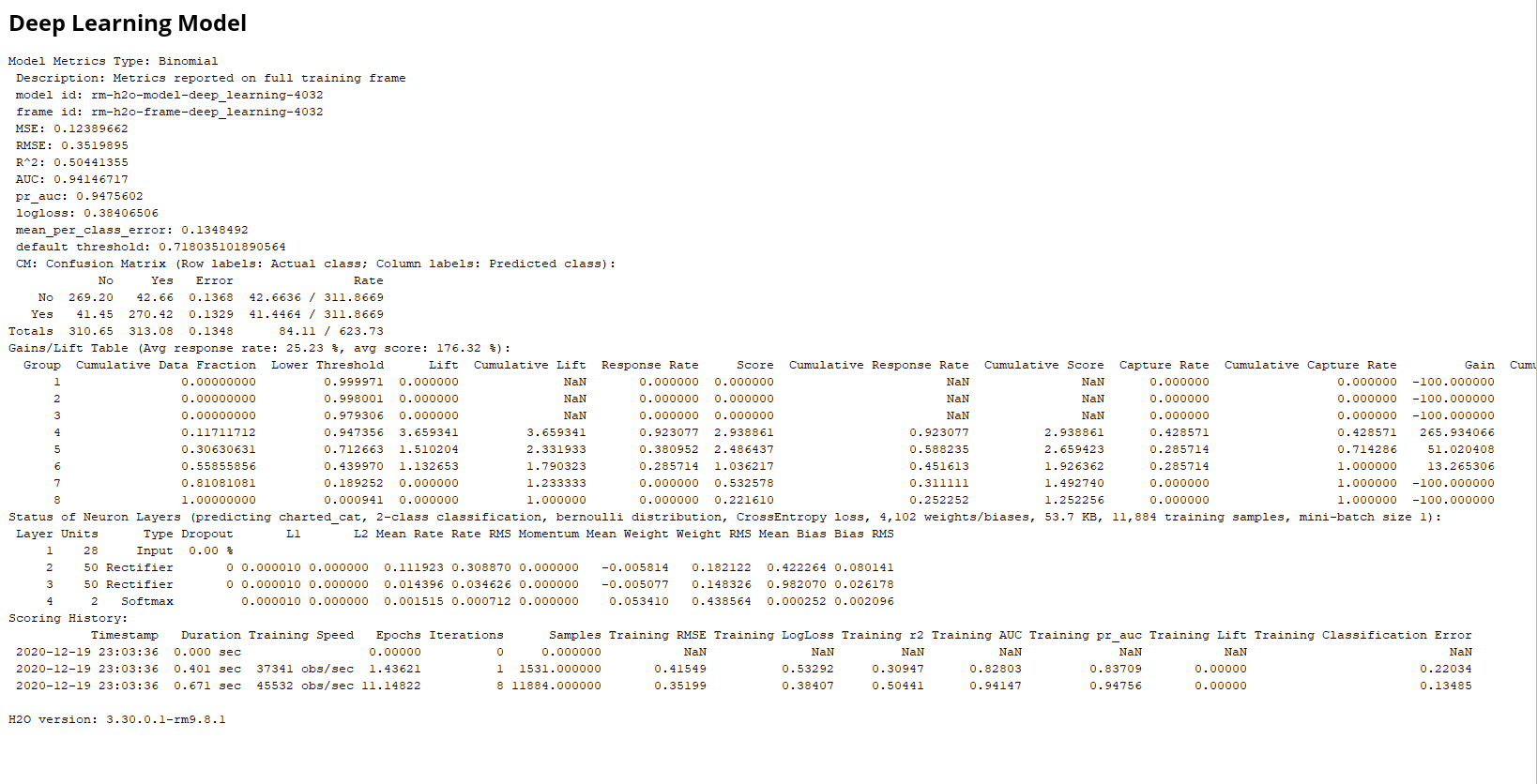




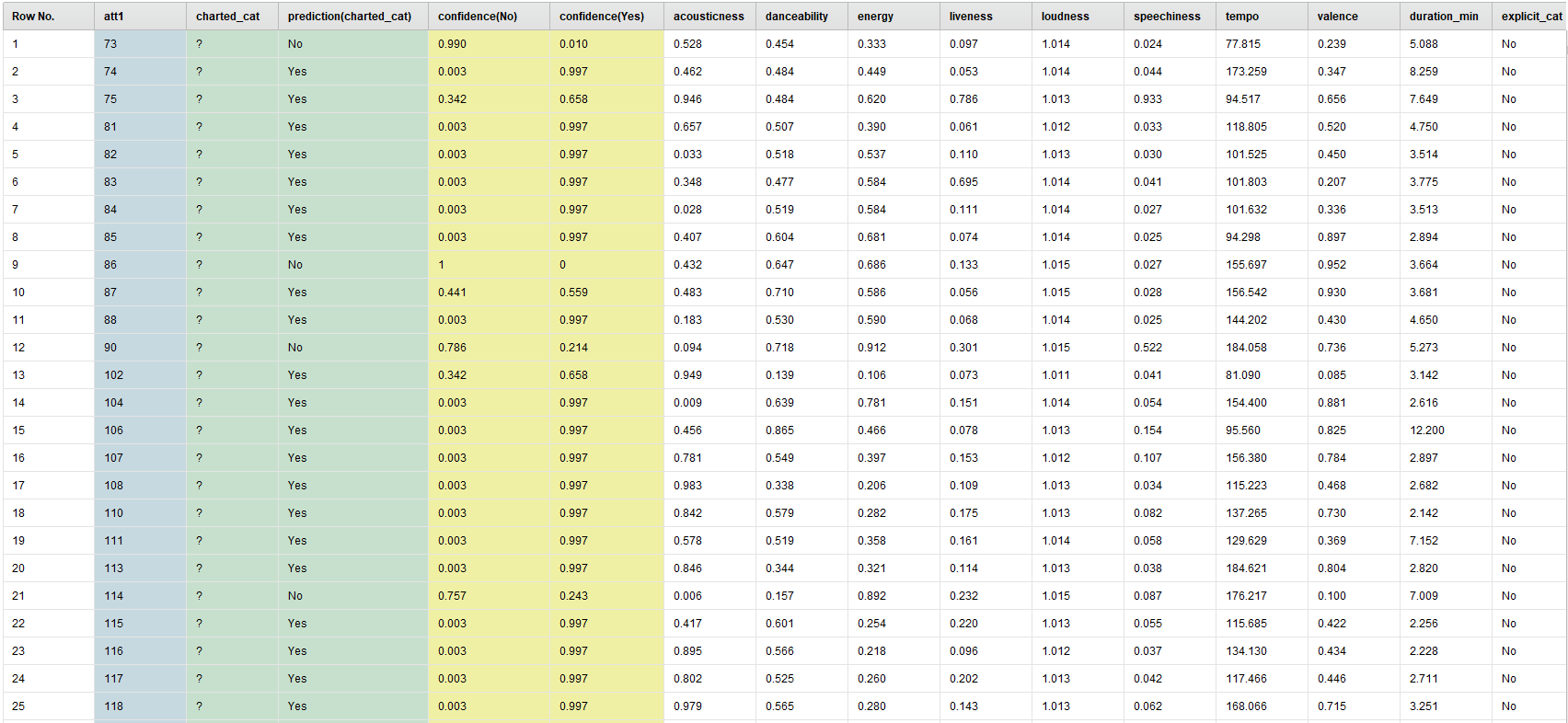




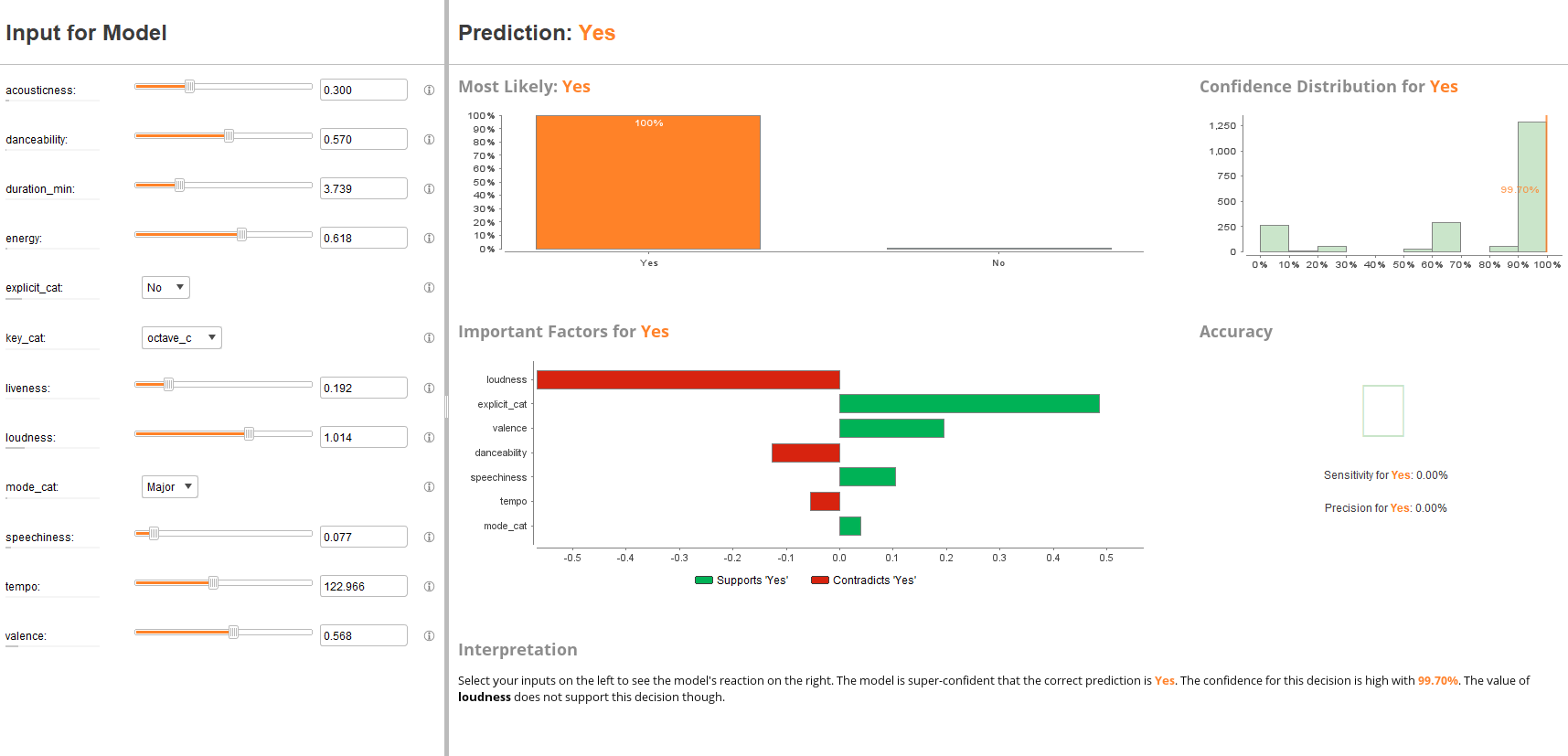
The tree models produced were very complex. The two pictures above are some of the smaller ones. Below is one of the deep learning models applied in the Bayesian boosting operator.



This is a small sample of the 2,000 total predictions with a confidence rating for each prediction.



RapidMiner also has a model simulator operator that is like what the app might look like. You input the values and the simulator tells you in real time Yes or No.



## Evaluation

Producers use prior experience, knowledge, and some gut feeling when it comes to determining if an artist’s song will make it to the charts. The purpose of this model would be to take the guess work out of it. Producers would simply analyze the song in real time and input the numbers into a clean app that would give a yes or no answer if the song is expected to chart or not. A side benefit would also be increase profits in through the various avenues and royalties.

This phase was in a way combined with the modeling phase during model assessment and looked decent. After scrutiny and model testing, this model seems to do a fairly good job at predicting the class variable. It also has decent performance metrics. It is safe to say that we can move this model into production and get some use out of it. One of the issues to address here would be the type of genre the studio tends to produce. This would require going back to business understanding and restarting the process focusing on that genre. As is stands currently, this is a general-purpose model and a specialized model based off the genre

## Deployment

The model is now in deployment and is predicting songs that chart and making the producers job easier. The model will need continual maintenance and updating since the music industry is changing all the time, but in time the model and algorithm will get better with more and more data put into it.

# Conclusion

Predictive analytics (PA) encompasses a variety of techniques to analyze historical and current data then make predictions off that data. It leverages techniques such as data mining, modeling, and machine learning to make those predictions of unknown events better than a human can, theoretically. With the amount of data prevalent in the world today, it is only a matter of time until machine learning takes over all industries. The music industry would be easy to implement a predictive model in. Spotify already has song analysis. If that could be implemented to analyze in real time in recording studios, it would make the model that much easier to use and make the producers job that much easier.

This paper was a look into what a model used to predict chart topping songs would look like, but there is still a large amount of work to be done. As mention above, this is a general-purpose model and not a specialized one. This would be a model used in studios that produce just as much rap as they do country and rock music. To me, however, this is not the case as most recording studios are know for the genre of music they produce. So, the next logical step would be to see if this generalized model could also specialize in certain genres. Genre might not even matter to the model at all. Another next step would be to start building out the app. Decisions would need to be made on how to package the final product, the best method for distribution, and of course cost of the product.

# References

Ay, Y. E. (2020). *Spotify Dataset 1921-2020*. Retrieved from Kaggle: https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks

Data Science Project Management. (2020). *CRISP-DM*. Retrieved from Data Science Project Management: https://www.datascience-pm.com/crisp-dm-2/

DeFoe, D. (2020). *Data on Songs from Billboard 1999-2019*. Retrieved from Kaggle: https://www.kaggle.com/danield2255/data-on-songs-from-billboard-19992019

E-Home Recording Studio. (2020, 12 20). *The Musicians Guide to Understanding Decibels*. Retrieved from E-Home Recording Studio: https://ehomerecordingstudio.com/decibels/#:~:text=In%20music%2C%20decibels%20are%20a%20measurement%20of%20Sound,a%20comparison%20to%20a%20different%20number%3A%200%20SPL.

Hany, O. (2020). *500 Greatest Songs of All Time*. Retrieved from Kaggle: https://www.kaggle.com/omarhanyy/500-greatest-songs-of-all-time

IBM. (2020). *CRISP-DM Help Overview*. Retrieved from IBM Knowledge Center: https://www.ibm.com/support/knowledgecenter/SS3RA7\_sub/modeler\_crispdm\_ddita/clementine/crisp\_help/crisp\_overview.html

MasterClass. (2020, November 8). *Music 101: What is an Octave*. Retrieved from MasterClass: https://www.masterclass.com/articles/music-101-what-is-an-octave#what-is-an-octave

Provost, F., & Fawcett, T. (2013). *Data Science for Business.* O'Reilly Media.

RapidMiner. (2020, 12 20). *Bayesian Boosting*. Retrieved from RapidMiner Documentation: https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/ensembles/bayesian\_boosting.html#:~:text=%20Bayesian%20Boosting%20%28RapidMiner%20Studio%20Core%29%20%201,which%20can%20now%20be%20applied%20on...%20More%20

Spotify. (2020, 12 20). *Web API*. Retrieved from Spotify for Developers: https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/

Wikipedia. (2020). *Cross-Industry Standard Process for Data Mining*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Cross-industry\_standard\_process\_for\_data\_mining

Wikipedia. (2020). *Pitch Class*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Pitch\_class

Wikipedia. (2020). *Predictive Analytics*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Predictive\_analytics

Wikipedia. (n.d.). *Pitch Class*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Pitch\_class