Internship Project Report: Hit Predict:Predicting Billboard Hits Using Spotify Data

Symbiosis Institute of Geoinformatics (SIG)

Symbiosis International (Deemed University) 5th
Floor, Atur Centre, Gokhale Cross Road Model
Colony, Pune – 411016

SUBMITTED BY:

Piyusha More
(M.Sc. Data Science and Spatial Analytics)

CERTIFICATE

Certified that this thesis titled 'Hit Predict: Predicting Billboard Hits Using Spotify Data' is a bonafide work done by Miss. Piyusha More, at (Technocolabs Softwares) and Symbiosis Institute of Geoinformatics, under our supervision.

Supervisor, Internal

(Prof. Rajesh Dhumal)
Symbiosis Institute of Geoinformatics

Supervisor, External

K. Venu Gopal

(Mr. Venu Gopal
Kadamba)

Technocolabs Softwares

INDEX

ACKNOWLEDGEMENT

I am thankful to receive assistance from my mentors in making this project successful. I take this opportunity to express my deep regards and gratitude towards Mr. Rajesh Dhumal (professor/internal mentor) and Mr. Venu Gopal Kadamba (external mentor) for supporting me throughout the completion of the project.

I would also like to thank my teammates and parents for their support and encouragement during this project.

PREFACE

Billboard magazine publishes the Billboard Hot 100 every week, which is the music industry's benchmark record chart for songs. Songs are ranked based on sales, radio airplay, and online streaming on this chart. The Million Dataset is a publicly accessible database of audio characteristics and metadata for popular music files from today.

Introduction:

- The Billboard Hot 100 Chart continues to be one of the most reliable indicators of a song's success.
- It is the standard record chart for songs in the music industry, published weekly by Billboard magazine.
- This chart displays song rankings based on sales, radio airplay, and online streaming.
- The Million Dataset is a publicly accessible database of audio characteristics and metadata for popular music files from today.
- Data for 4000 songs was gathered from Billboard.com and the Million Song Dataset Songs.
- Spotify Web API was used for extracting audio features of collected songs from given two sites.
- Spotify Web API is an internet-based interface that allows programmes to retrieve and manipulate Spotify data.
- Every internet browser employs the HTTP protocol, which is used by the Web API.
- This Web API also gives users access to data about themselves, such as playlists and music saved in the Your Music library. The user grants such access through selected authorisation.
- To forecast which songs will become Billboard Hot 100 Hits, researchers used machine learning algorithms.
- Using machine-learning methods such as Logistic Regression, GDA, SVM, and Decision Trees, it was able to predict the Billboard success of a song with 75% accuracy.
- Ten audio features were extracted from the Spotify API like Danceability, Instrumentalness, Acousticness, Valence, Liveness, etc.

- Based on song features further data analysis and modeling was done.
- The meaning of each feature of a song is described below,

1. Danceability: The danceability of a song is determined by a combination of factors including beat strength, tempo stability, and overall tempo.
2. Instrumentalness: The number of vocals in the song is represented by this number.
3. Acousticness: The acoustic quality of a song is represented by this number. If the score is 1.0, the music is almost certainly acoustic.
4. Valence: A scale ranging from 0.0 to 1.0 that describes how positive a track is musically. Tracks with a high valence sound more positive (e.g. happy, cheerful), whereas tracks with a low valence sound more negative (e.g. sad, depressed, angry).
5. Energy: It is a perceptual indicator of activity and intensity. Fast, loud, and noisy are typical characteristics of energetic tracks.
6. Liveness: This number indicates how likely the song was recorded in front of a live audience. "A rating above 0.8 indicates that the track is likely to be active," according to the official documentation.
7. Speechiness: Speechiness detects whether or not a track contains spoken words.
8. Loudness: the quality of a sound that determines the strength of the auditory sensation it produces
9. Tempo: The lower the time between successive beats, the faster a piece must be played. A tempo of 60 beats per minute, for example, equals one beat per second, whereas a tempo of 120 beats per minute is twice as fast.
10. Mode: Mode column is the target column indicating whether song was billboard hit or not.
11. SpotifyID : The Spotify artist ID is a string of numbers and letters that identifies a particular artist.

12. Tracks: Name of songs

13. Artist: Name of particular artist sung a certain track/song

Project Objective:

The purpose of this project is to forecast whether or not a song will chart on the Billboard Hot 100 using Spotify API data by applying the following algorithms,

1. Logistic Regression
2. GDA
3. SVM
4. Random Forest

And create a Flask application of that model whose algorithm is giving higher accuracy and deploy that model on the Heroku platform.

LITERATURE REVIEW

To make the classification interval between two or more classes the greatest, the support vector machine approach is employed to identify the classification hyperplane between them (Dai, 2018). As SVM cannot handle the multi-classification problem more efficiently because it takes a long-time model building. Improved SVM have used using weighted Euclidean distance, the radial product kernel function, and SVM algorithm for less time model building, large data classification, and for higher accuracy which is one of the big advantages of SVM. Reproducing Kernel Hilbert Space is a method of mapping input vectors to higher-dimensional spaces that can be utilised in SVM classification (Dai, 2018).

From this, I understood random forests can produce higher accuracy and can process data faster in the case of large data containing a huge number of attributes because it proves that as more trees are added, random forests do not overfit. In this(Claesen *et al.*, 2014) The author of the paper worked on the development of conformal predictors, which produce region predictions with a pre-determined number of errors in the long run, but when used to make singleton predictions, conformal predictors based on random forest produced accuracy that was comparable to random forest accuracy. As a result, I realised that while conformal predictors are designed to forecast valid regions, they may also be used to predict singletons.(Claesen *et al.*, 2014).

The widely used Radial Basis Function (RBF) kernel in Support Vector Machine(SVM) is known to perform well on a large variety of issues(Thompson and Licklider, 2011). In this paper, a method is suggested in which using RBF kernels significantly lower the run-time complexity of models for many learning tasks.

SVM was created as a result of the best classification of linearly separable issues. The purpose of the support vector machine algorithm is to determine the classification line that will result in the biggest classification interval. To identify the best classification surface, the two categories of samples are accurately divided and extended to a spatial form. To improve the effect of the sample SVM classification, the attributes are linear transformed. The feature space's shape changes as well. The new feature space has a better linear classification hyperplane. In order to increase the performance of SVM classification, kernel function calculation may also effectively avoid the influence of some weak correlation or unnecessary features.

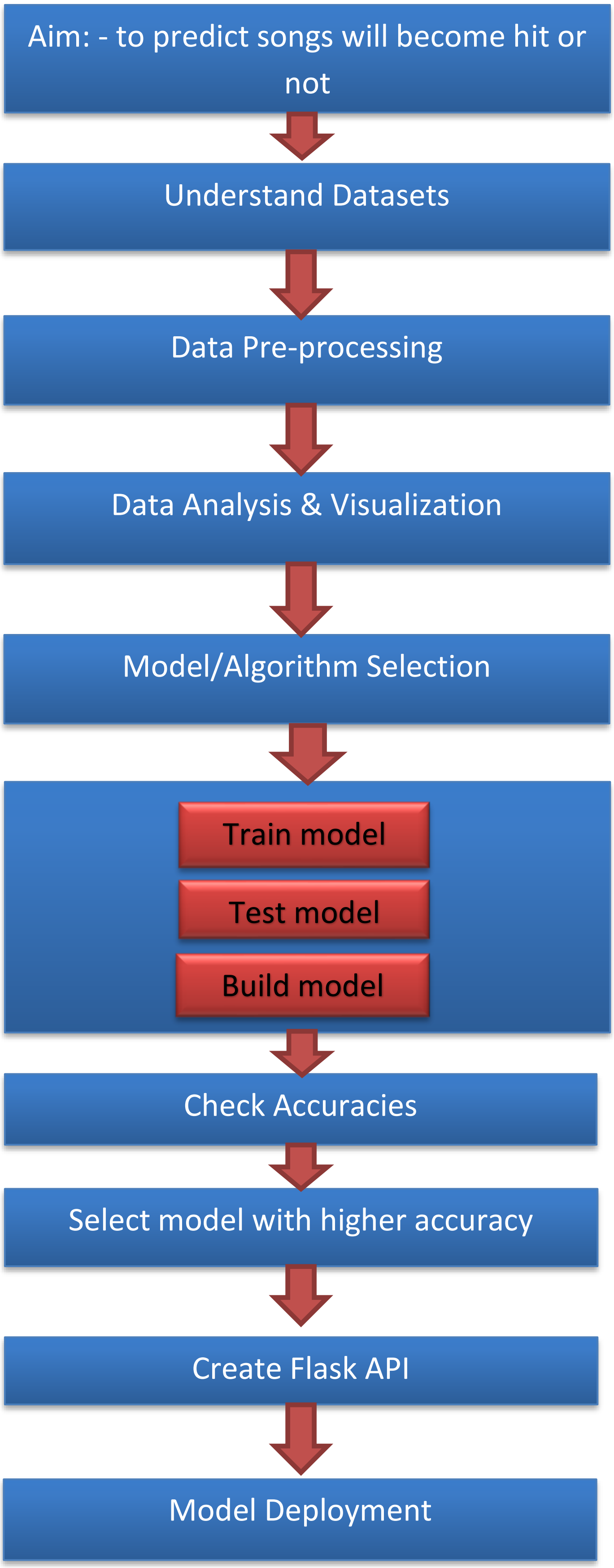
The multi-classification problem is incompatible with the support vector machine approach. The characteristics will be extracted into the feature space if the noise in the raw data is not appropriately dealt with. Similarly, multimedia applications generate vast amounts of high-dimensional data in the form of picture, sound, and video. To cope with the large-scale data problem in the paper, employ the strategy of decreasing feature disturbance. The training samples are separated into block learning by point - point voting classification under the condition that the classifier's performance is not lowered. By lowering the amount of training samples, the modelling time is cut in half.

A random forest is made up of (x1, y1), (x2, y2),..., (xm, ym) training sets. The percentage of valid predictions given for x by decision trees is then used to calculate the conformance score of a new example (x, y). Conformal predictors were created to offer region predictions with a fixed error rate. However, we can ignore the nature of conformal predictors and force them to always generate a specific prediction in order to compare them to bare predictions output by traditional machine learning methods. We can anticipate the label with the highest p-value after assigning a p-value to each label for each item. Forced point prediction is the term for this. We make a random prediction if several labels have the same p-value (this is known as a tie). As a result, we can add the framework of conformal prediction to the equation.

In this paper the result (Claesen *et al.*, 2014)was discussed that the framework added of conformal prediction to the equation. We can provide valid region predictions and complement each prediction with confidence using the random forest algorithm without losing accuracy and while benefiting from conformal predictions: we can produce valid region predictions and complement each prediction with confidence. The results of comparing the accuracy of different conformal predictors in forced point prediction were consistent with the efficiency comparison: cp-rf and cp-rf-kNN significantly outperformed other predictors on some mass spectrometry datasets and were at least as good as the benchmarks on all data sets.

For dichotomous outcomes(Musa, 2013), logistic regression (LR) is a multivariable technique. It's a typical statistical categorization method that's especially useful for illness state (healthy/diseased), decision making (yes/no), and mortality models (dead, living). In practical disciplines such as medicine, biology, and epidemiology, it is commonly employed in binary classification problems. Because of its simplicity and interpretability, it has been widely used. Logistic regression necessitates specific data requirements, such as low or no collinearity among the independent variables and linearity of the independent variables with the logit model. The purpose of this research paper is to create a standard, thorough comparison of SVM and LR on multiple data sets. The results reveal that the SVM and LR perform equally well for balanced and imbalanced data across all performance measures. Support vector machines, on the other hand, may be superior for severely unbalanced data sets.

Project Flowchart


Aim: - to predict songs will become hit or not
Understand Datasets
Data Pre-processing
Data Analysis & Visualization
Model/Algorithm Selection
Train model
Test model
Build model
Check Accuracies
Select model with higher accuracy
Create Flask API
Model Deployment

METHODOLOGY:

Data Description:

Data for 4000 songs was gathered from Billboard.com and the Million Song Dataset Songs were from 1990 to 2018, in CSV format, and ten audio features were taken via Spotify Web API, as shown in the table below. Where the mode characteristic is used as a target variable, with the values 0 and 1 indicating whether the song was a Billboard hit or not. Spotify's API was used to extract audio features. Except for loudness, which is measured in dB, Spotify assigns a rating between 0 and 1 to each song.

Audio Features	
Danceability	Liveness
Instrumentalness	Speechiness
Acousticness	Loudness
Valence	Tempo
Energy	Mode

Table-1

Data Pre-processing:

Data pre-processing entails converting raw data into well-formed data sets in order to use data mining methods. Raw data is frequently incomplete and formatted inconsistently. The success of every project involving data analytics is directly proportional to the quality of data preparation. Data validation and data imputation are both part of the pre-processing process. The purpose of data validation is to determine whether the data is comprehensive and accurate. The purpose of data imputation is to rectify errors and fill in missing numbers, which can be done manually or automatically by BPA programming.

Both database-driven and rules-based systems benefit from data preparation. Data preparation is essential in machine learning (ML) procedures because it ensures that big datasets are organised in a way that learning algorithms can understand and comprehend the data they contain.

The tool python was used for data pre-processing and cleaning purposes. In python there are different libraries but, in this project, we mainly used pandas and NumPy for data cleaning processing. I used these libraries to remove the unwanted fields and to merge the billboard datasets and million song datasets.

Data Cleaning:

Filling in empty values or eliminating rows with missing data, smoothing noisy data, or correcting discrepancies in the data are all examples of data cleansing methods. Human error can lead to data inconsistencies. To avoid giving that data item an advantage, duplicate values should be deleted via deduplication.

Data integration is the process of combining data with various representations and resolving data conflicts.

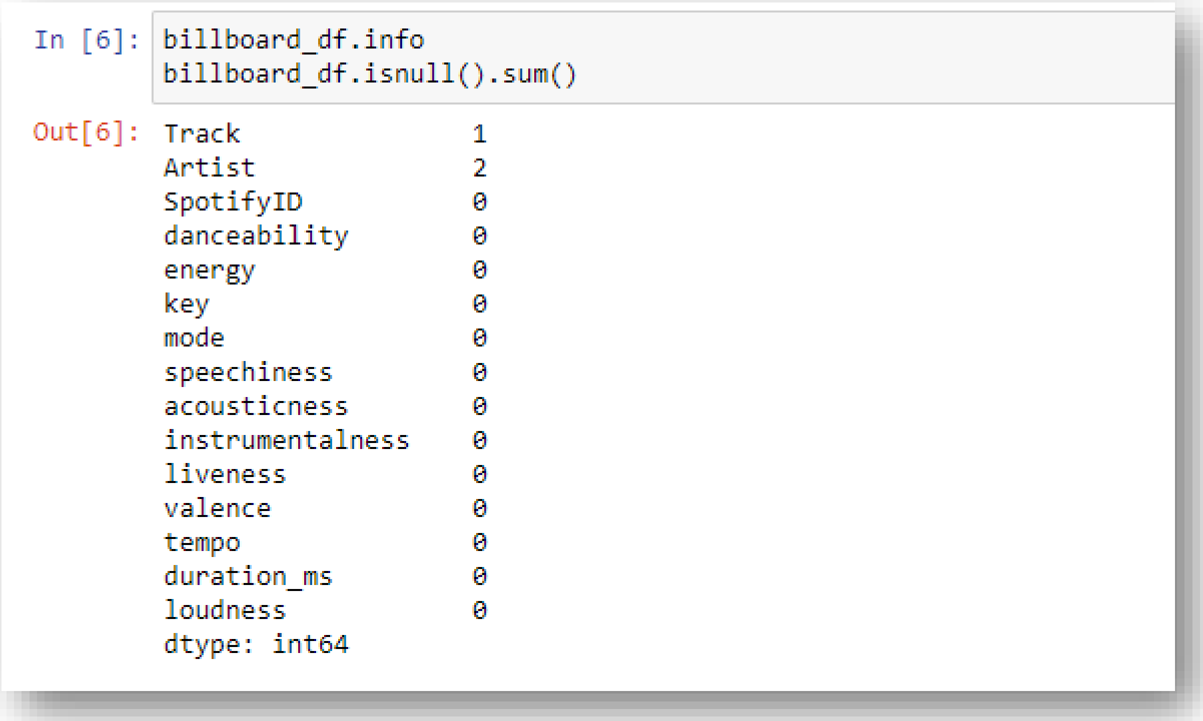
Data normalisation and generalisation are two steps in the data transformation process. Normalization is a method of ensuring that no data is duplicated, that everything is saved in one location, and that all dependencies are logical.

Databases can become slower, more expensive to access, and more difficult to effectively store when the volume of data is large. In a data warehouse, data reduction is used to display a simplified version of the data.

Discretization of data: To replace raw values with interval levels, data could be discretized. By dividing the range of attribute intervals, this step reduces the number of possible values for a continuous attribute.

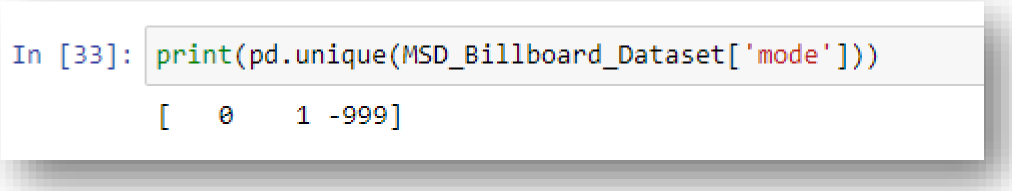
Sampling of Data: Occasionally, a dataset is too large or complex to work with due to time, storage, or memory constraints. If a portion of the dataset has roughly the same properties as the original, sampling techniques can be used to pick and work with just that subset.

1. I found missing values in the billboard dataset, in Track and Artist features which are as follows, and dropped those columns using dropna() function.

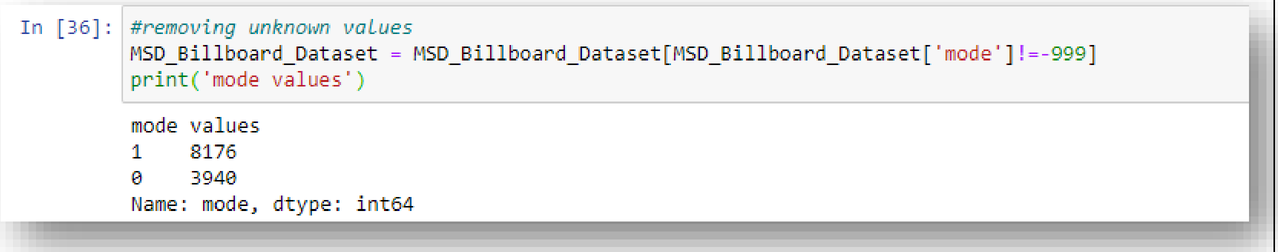


```
In [6]: billboard_df.info
        billboard_df.isnull().sum()

Out[6]: Track               1
        Artist              2
        SpotifyID           0
        danceability        0
        energy              0
        key                 0
        mode                0
        speechiness         0
        acousticness        0
        instrumentalness    0
        liveness            0
        valence             0
        tempo               0
        duration_ms         0
        loudness            0
        dtype: int64
```

2. The target variable mode should consist of 0's and 1's but the column was also having unknown values of -999.

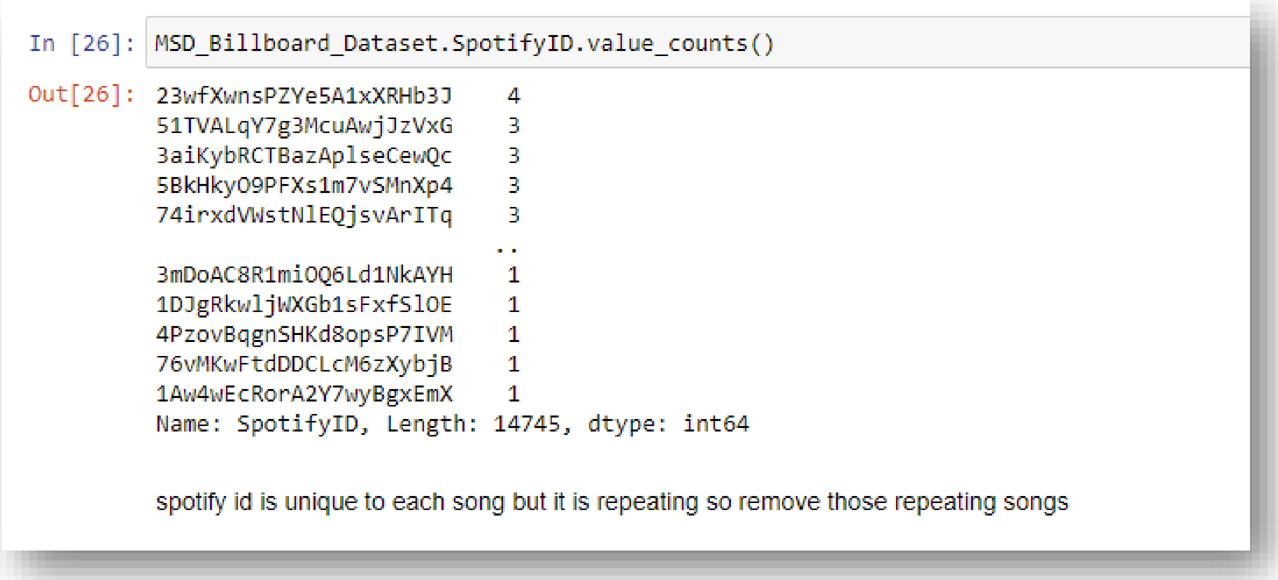


```
In [33]: print(pd.unique(MSD_Billboard_Dataset['mode']))

         [   0    1 -999]
```



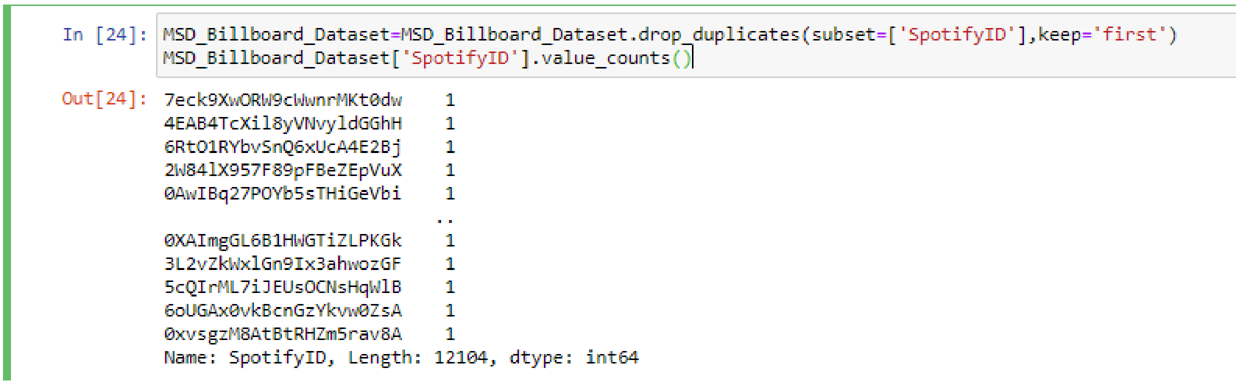
```
In [36]: #removing unknown values
         MSD_Billboard_Dataset = MSD_Billboard_Dataset[MSD_Billboard_Dataset['mode']!=-999]
         print('mode values')

         mode values
         1    8176
         0    3940
         Name: mode, dtype: int64
```

3. Each song has a unique Spotify Id but the Spotify Id column consisted of repeated values of Spotify Id's which were making the dataset inconsistent



```
In [26]: MSD_Billboard_Dataset.SpotifyID.value_counts()

Out[26]: 23wfXwnsPZYe5A1xXRHb3J    4
         51TVALqY7g3McuAwjJzVxG    3
         3aiKybRCTBazAplseCewQc    3
         5BkHkyO9PFXs1m7vSMnXp4    3
         74irxdVWstNlEQjsvArITq    3
                                  ..
         3mDoAC8R1miOQ6Ld1NkAYH    1
         1DJgRkwljWXGb1sFxfSlOE    1
         4PzovBqgnSHKd8opsP7IVM    1
         76vMKwFtdDDCLcM6zXybjB    1
         1Aw4wEcRorA2Y7wyBgxEmX    1
         Name: SpotifyID, Length: 14745, dtype: int64
```

spotify id is unique to each song but it is repeating so remove those repeating songs

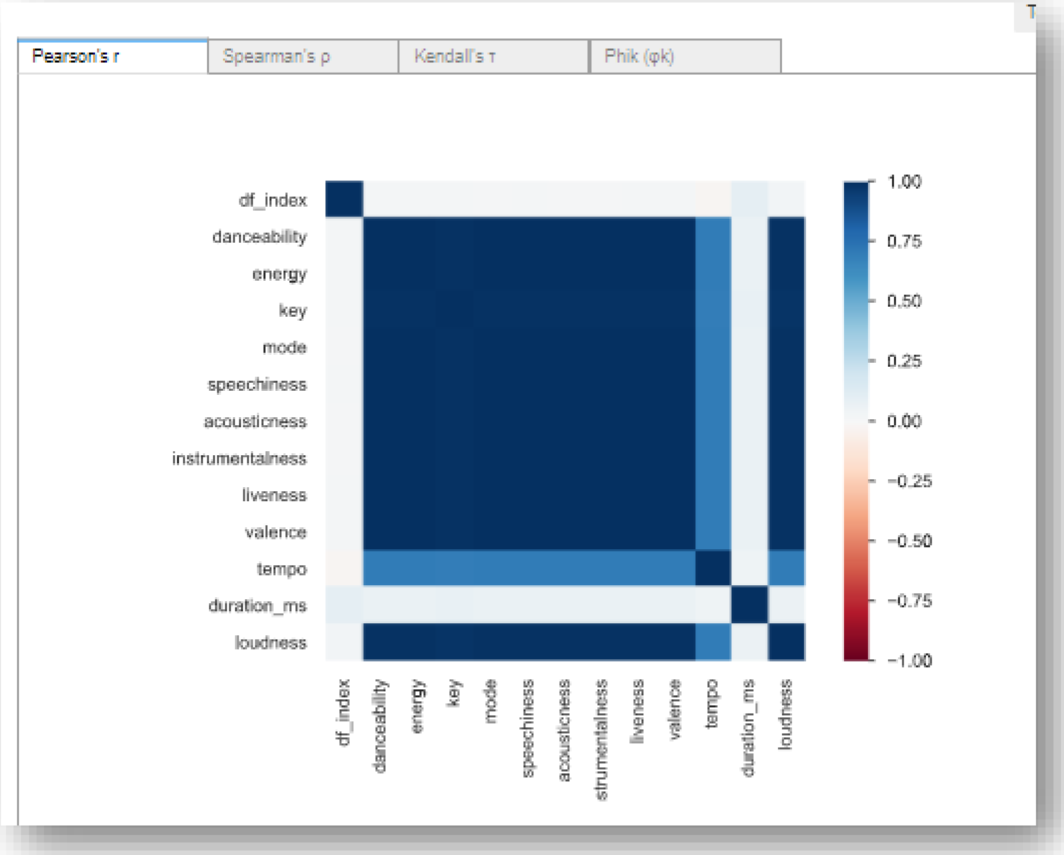


```
In [24]: MSD_Billboard_Dataset=MSD_Billboard_Dataset.drop_duplicates(subset=['SpotifyID'],keep='first')
         MSD_Billboard_Dataset['SpotifyID'].value_counts()

Out[24]: 7eck9XwORW9cWwnrMKt0dw    1
         4EAB4TcXil8yVNvyldGGhH    1
         6RtO1RYbvSnQ6xUcA4E2Bj    1
         2W84lX957F89pFBeZEpVuX    1
         0AwIBq27POYb5sTHiGeVbi    1
                                  ..
         0XAImgGL6B1HWGTiZLPKGk    1
         3L2vZkWxlGn9Ix3ahwozGF    1
         5cQIrML7iJEUsOCNsHqWlB    1
         6oUGAx0vkBcnGzYkvw0ZsA    1
         0xvsgzM8AtBtRHZm5rav8A    1
         Name: SpotifyID, Length: 12104, dtype: int64
```

Data Analysis and Visualization:

We acquire data at many times across processes and transactions today, which offers enormous potential to improve the way we work. This data analysis, on the other hand, can only provide value to the business if it's used to generate insights on how to improve your products and services. Data analysis is the science of analysing a piece of data in order to develop conclusions about the information in order to make judgments or simply to broaden one's knowledge on a variety of topics. It entails putting data via operations. This is done in order to gain exact conclusions that will aid us in achieving our goals, such as operations that cannot be pre-defined since data collecting may disclose special difficulties.

According to that I tried to understand the dataset and gather knowledge from the study, by looking at the billboard dataset and million songs dataset I came to understand that merging both datasets should be beneficial. As a result, I combined the two datasets and began looking for correlations between features using Pearson's correlation, The Pearson's correlation coefficient is derived as the product of the standard deviations of each data sample divided by the covariance of the two variables. It is the process of normalising the covariance between two variables in order to provide a score that can be understood. In short, a linear link between two variables is measured by the strength and direction of the relationship. The values are always between -1 (strong negative relationship) and +1 (strong positive association) (strong positive relationship). where I discovered that all of the features are exactly connected, as seen in the diagram below.

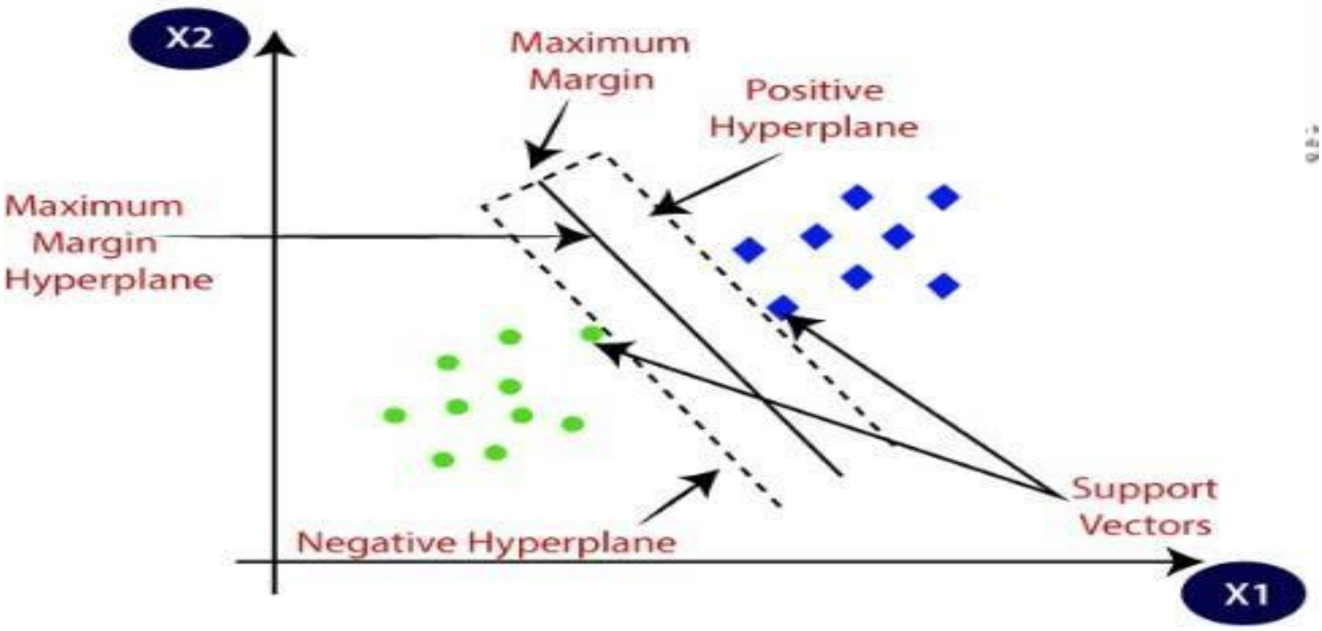


After discovering the link, I discovered that each song is sung by an artist who adds his own musical elements to the mix. So, the Artist column was dropped from the dataset, and the remaining other columns except for the mode column were kept as independent variables because all features of the song are necessary to know what background song features it used and how much. Also “mode” column name was replaced to “hit” because the mode is also one of the functions used in python due to which errors were getting generated.

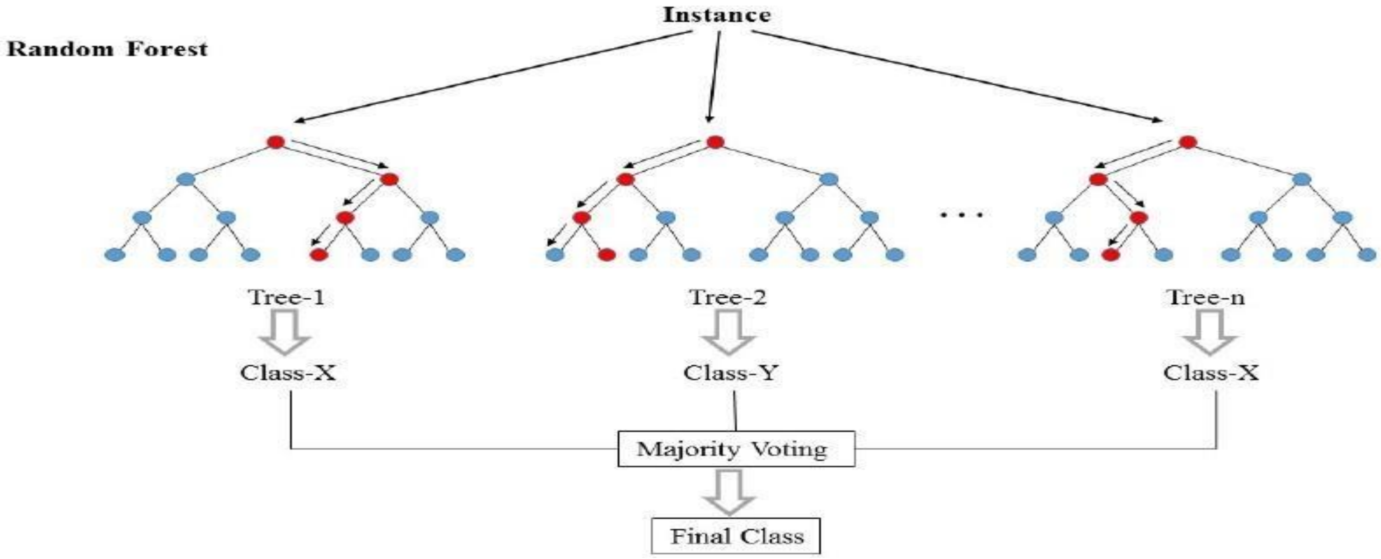
Build and Train Model:

A model is created by learning and generalising from training data, then using that knowledge to new data it has never seen before to make predictions and achieve its goal. In this scenario, I need to forecast whether or not a song will become a Billboard success. For that purpose, the dependent features of songs mentioned in table 1 have been used and mode which is used as a target.

1. **Support Vector Machine: -**
   It's a supervised machine learning technique that's used to solve two different problems: classification and regression. This approach is also commonly utilised in the classification of different sorts of issue statements.
   In order to do so, we must plot the data in n-dimensional space. Hyperplanes can also be created with the help of a suitably SVM classifier. We must also determine the best margin for the dataset we are using depending on the margins.



2. **Random Forest Model:**
   Both regression and classification problems can be solved with this approach. It's a sequence of trees with different selections. It helps to increase accuracy by lowering overfitting in the Decision tree. Rather of depending on a single decision tree, it collects predictions from each tree and forecasts the ultimate output based on the majority of votes. It helps to increase accuracy by lowering overfitting in the Decision tree.

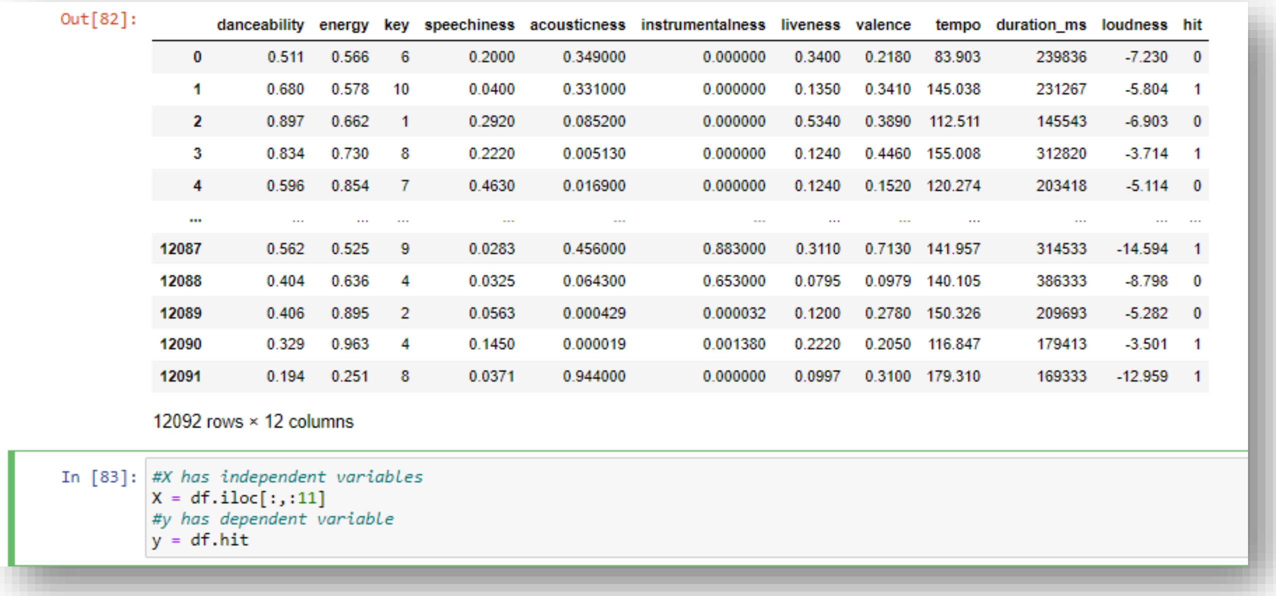


3. **Logistic Regression Model: -**
   Based on a collection of independent variables, the logistic regression approach is used to estimate discrete values. By fitting data to a logit function, it assists you in predicting the likelihood of an event occurring. As a result, logistic regression is another name for it. Its output value falls between 0 and 1 because it predicts probability. Logistic regression is similar to linear regression; however, it is employed when the dependent variable is something other than a number. It's termed regression, but it actually does classification based on the regression, sorting the dependent variable into one of two groups.

4. **Gaussian discriminant analysis model: -**
   When we have a classification issue with continuous random variables as input features, we can apply GDA, a generative learning technique in which

we assume p(x|y) is distributed according to a multivariate normal distribution and p(y) is distributed according to the Bernoulli distribution. According to towards data science one of article, if a binary classification issue in which we wish to learn to discriminate between two classes, class A (y=1) and class B (y=0), based on particular attributes. Now we'll take all of the label A samples and try to learn the features so that we can create a model for class A. Then we take all of the instances labelled B and try to understand their characteristics in order to create a distinct model for class B. Finally, when classifying a new element, we compare it to each model to discover which one best fits. It's termed Generative Learning Algorithms because we try to model p(x|y) and p(y) instead of the p(y|x) we performed earlier.

The dataset got split into 75/25 into training/testing sets and separated the features as dependent and independent variables. As follows,

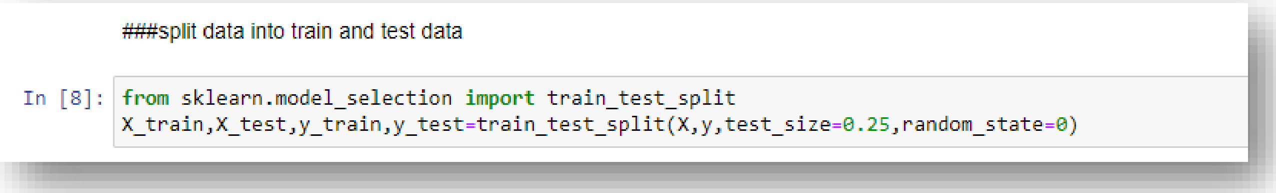


Out[82]:

	danceability	energy	key	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	loudness	hit
0	0.511	0.566	6	0.2000	0.349000	0.000000	0.3400	0.2180	83.903	239836	-7.230	0
1	0.680	0.578	10	0.0400	0.331000	0.000000	0.1350	0.3410	145.038	231267	-5.804	1
2	0.897	0.662	1	0.2920	0.085200	0.000000	0.5340	0.3890	112.511	145543	-6.903	0
3	0.834	0.730	8	0.2220	0.005130	0.000000	0.1240	0.4460	155.008	312820	-3.714	1
4	0.596	0.854	7	0.4630	0.016900	0.000000	0.1240	0.1520	120.274	203418	-5.114	0
...	...	...	...	...	...	...	...	...	...	...	...	...
12087	0.562	0.525	9	0.0283	0.456000	0.883000	0.3110	0.7130	141.957	314533	-14.594	1
12088	0.404	0.636	4	0.0325	0.064300	0.653000	0.0795	0.0979	140.105	386333	-8.798	0
12089	0.406	0.895	2	0.0563	0.000429	0.000032	0.1200	0.2780	150.326	209693	-5.282	0
12090	0.329	0.963	4	0.1450	0.000019	0.001380	0.2220	0.2050	116.847	179413	-3.501	1
12091	0.194	0.251	8	0.0371	0.944000	0.000000	0.0997	0.3100	179.310	169333	-12.959	1

12092 rows × 12 columns

In [83]:
```
#X has independent variables
X = df.iloc[:,:11]
#y has dependent variable
y = df.hit
```



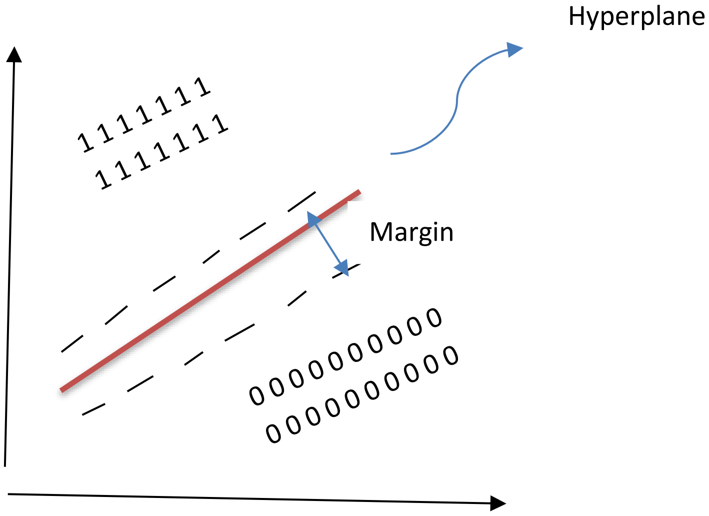
###split data into train and test data

In [8]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=0)
```

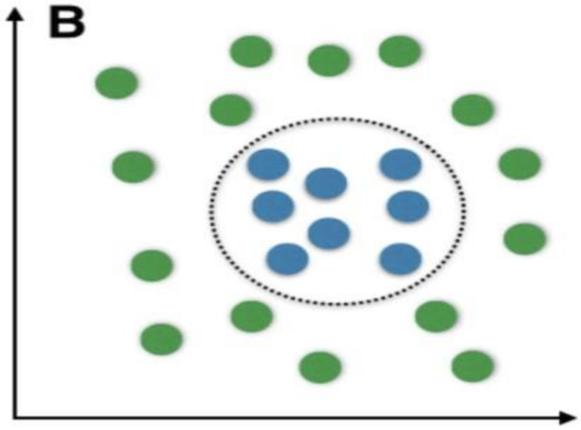
Firstly, I applied Logistic Regression Algorithm and Gaussian Discriminant Analysis Algorithm (GDA) but the accuracies were low as compared to the other two algorithms, the next algorithm I applied to my model was SVM (Support Vector Machine).

For two-group classification issues, SVM is a supervised machine learning model that uses a classification method. It takes these data points and generates the hyperplane, also known as a decision boundary, where anything that falls on either side of the plane is treated as a 0 or 1 based on the input attributes.

SVM can solve two types of problems, linearly separable problems, and non-linearly separable problems. If data is linearly separable, apart from classifying two separate classes by creating a hyperplane It also constructs two margin lines, each of which has a certain distance between them so that the points of both categories can be easily separated linearly. After creating hyperplane, SVM also creates two parallel planes one in dotted lines on 1's side and another dotted line on 0's side and both the lines are parallel to the hyperplane that is actually created.



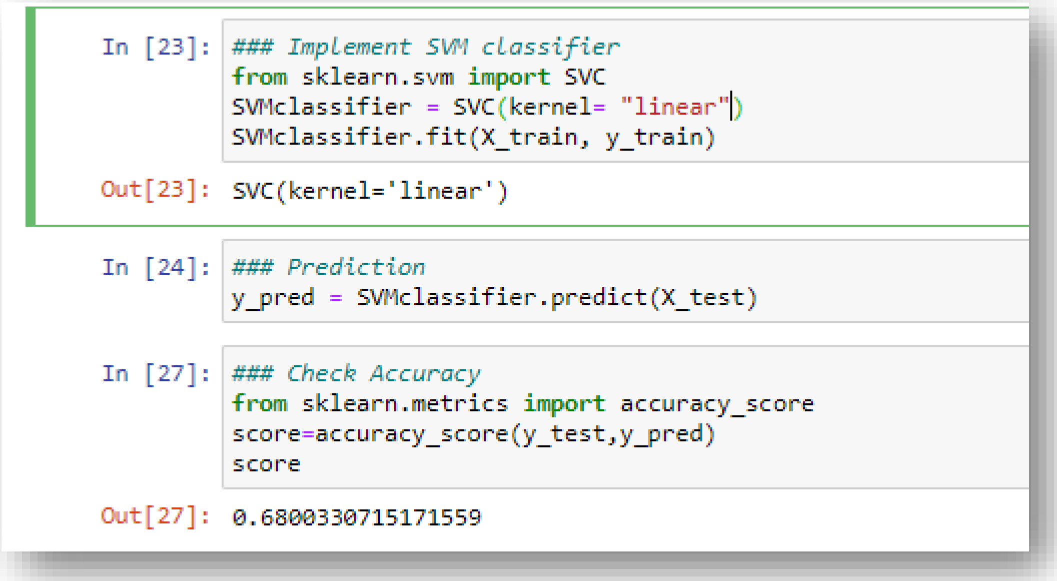
When a dotted hyperplane is created parallel to the main hyperplane it makes sure that the line passes through one of the nearest 1's points similarly when another dotted line is created it passes through one of the nearest 0's points and from there particular hyperplane is created and this is the institution behind SVM.



Non-linear Data

To solve the problem of non-linearly separable problems SVM uses a technique called SVM kernels. The main aim of SVM kernels is that it transforms lower dimensions to higher dimensions, for example, 2D to 3D. because after converting or transforming dimensions to higher dimensions it will be easy to classify particular points by hyperplane itself. Kernels just create some more features based on particular kernel formula.

To check whether the dataset is linearly separable or non-linear, I used the SVC model's linear kernel, where I understand that if the accuracy of the model is below 50% then that dataset is non-linear and if the model gives higher accuracy by simply using linear kernel the dataset is linearly separable. In my case, the dataset is linearly separable so I had no need to use other kernels as follows.

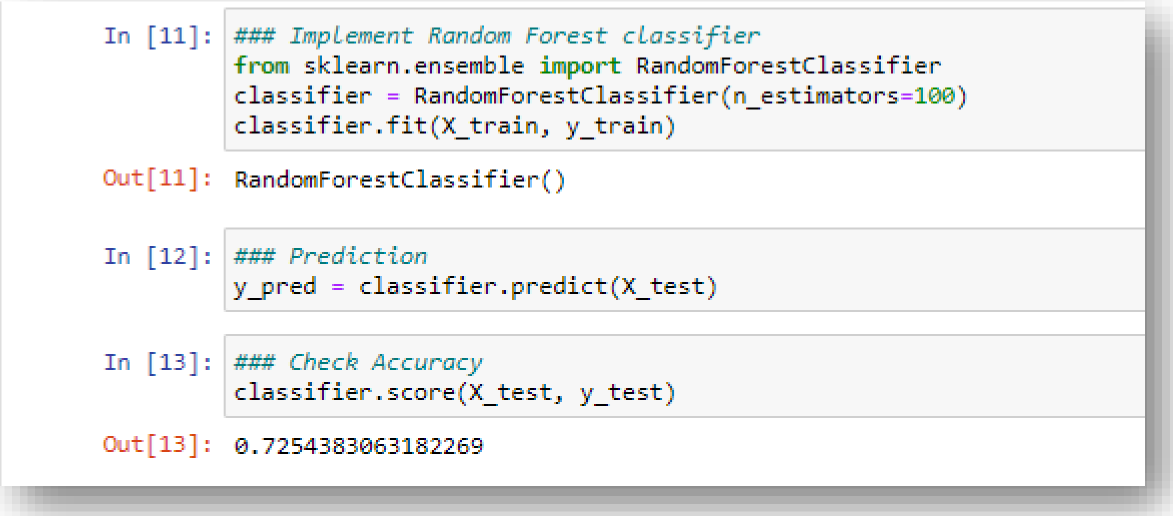


```
In [23]: ### Implement SVM classifier
         from sklearn.svm import SVC
         SVMclassifier = SVC(kernel= "linear")
         SVMclassifier.fit(X_train, y_train)

Out[23]: SVC(kernel='linear')

In [24]: ### Prediction
         y_pred = SVMclassifier.predict(X_test)

In [27]: ### Check Accuracy
         from sklearn.metrics import accuracy_score
         score=accuracy_score(y_test,y_pred)
         score

Out[27]: 0.6800330715171559
```

Lastly, I applied a random forest algorithm to my dataset. Random forest is a supervised machine learning technique that may be used to address problems like regression and classification. It's an ensemble learning technique which is used to come up with infusions to complicated problems by integrating many classifiers.

The name random forest itself tells that Bagging or bootstrap aggregation are used to train the algorithm's resulting forest. In ML algorithms, bagging is one of the methods in ensemble learning which is used to get better accuracy on a model. Decision trees are elementary constituents of random forest.

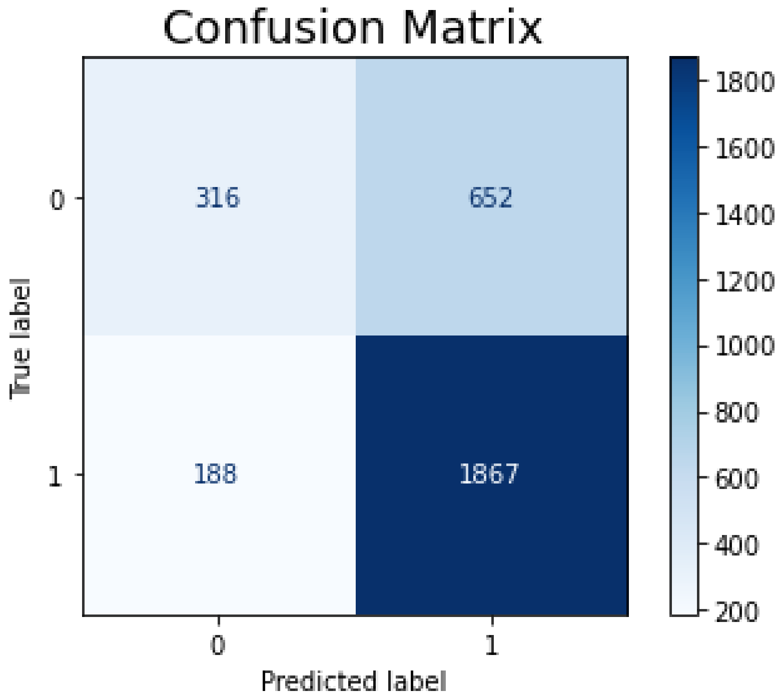
There are many features of random forest, for that one should be using it like, it is more accurate than any other decision tree algorithm, it reduces overfitting in decision trees. During boosting, at the node's splitting point, a subset of characteristics is chosen at random. In this model, the hyperparameter n_estimators are just the number of trees the algorithm creates before taking the averages of forecasts, as illustrated below,



```
In [11]: ### Implement Random Forest classifier
         from sklearn.ensemble import RandomForestClassifier
         classifier = RandomForestClassifier(n_estimators=100)
         classifier.fit(X_train, y_train)

Out[11]: RandomForestClassifier()

In [12]: ### Prediction
         y_pred = classifier.predict(X_test)

In [13]: ### Check Accuracy
         classifier.score(X_test, y_test)

Out[13]: 0.7254383063182269
```

RESULT:

As previously written, it was able to predict ~75% of billboard success. And among all the algorithms it was seen that the Random forest model achieves higher accuracy near to 75% that is ~73%



A confusion matrix is a table that describes how well a classification model (or classifier) performed on a set of test data for which the true values are known. Although the confusion matrix is straightforward to comprehend, the terminology used to describe it can be perplexing.

True Positive: The number of times the classifier correctly predicts the positive class as positive is referred to as true positive (TP = 316).

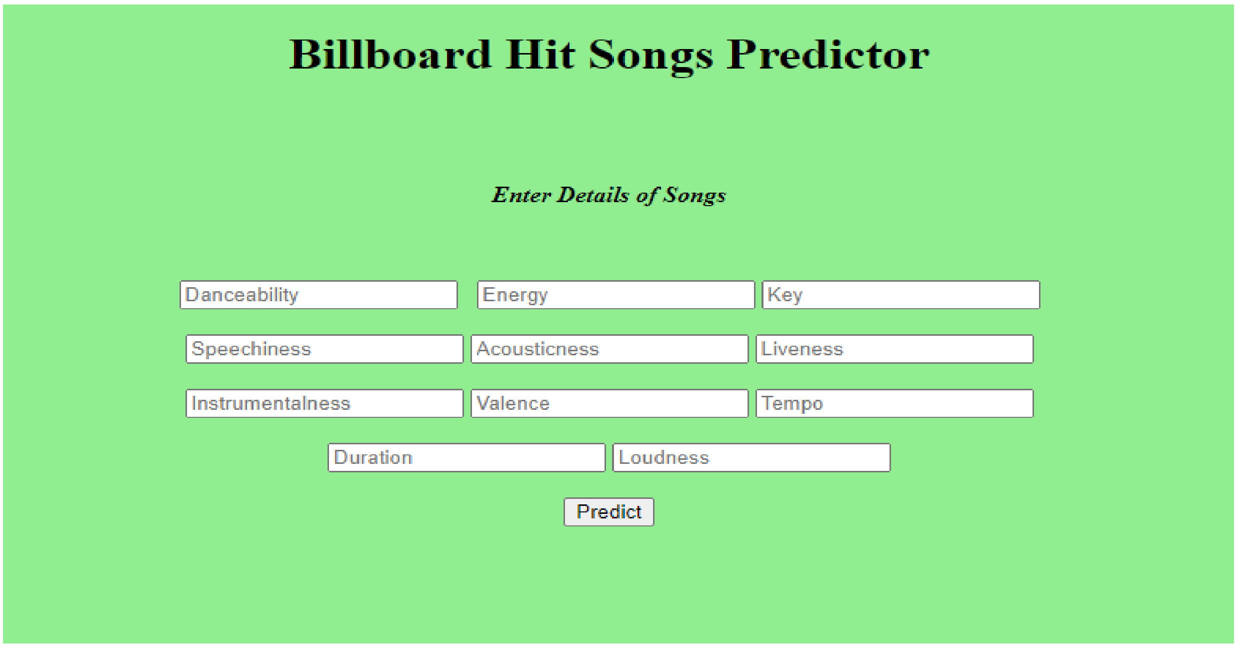
True Negative: The number of times the classifier has successfully predicted the negative class as negative(TN= 1867).

False Positive: This term refers to the amount of times a classifier wrongly predicts a negative class as a positive. (Error Type 1)(FP = 652)

False Negative: The number of times the classifier gets it wrong and forecasts the positive class as negative. (Error Type 2)(FN = 188)

Looking at the confusion matrix prepared by the random forest model, we can easily say that actual hit songs are 2055 and actual non-hit songs are 968 and the model predicted 504 non-hit songs and 2519 hit songs.

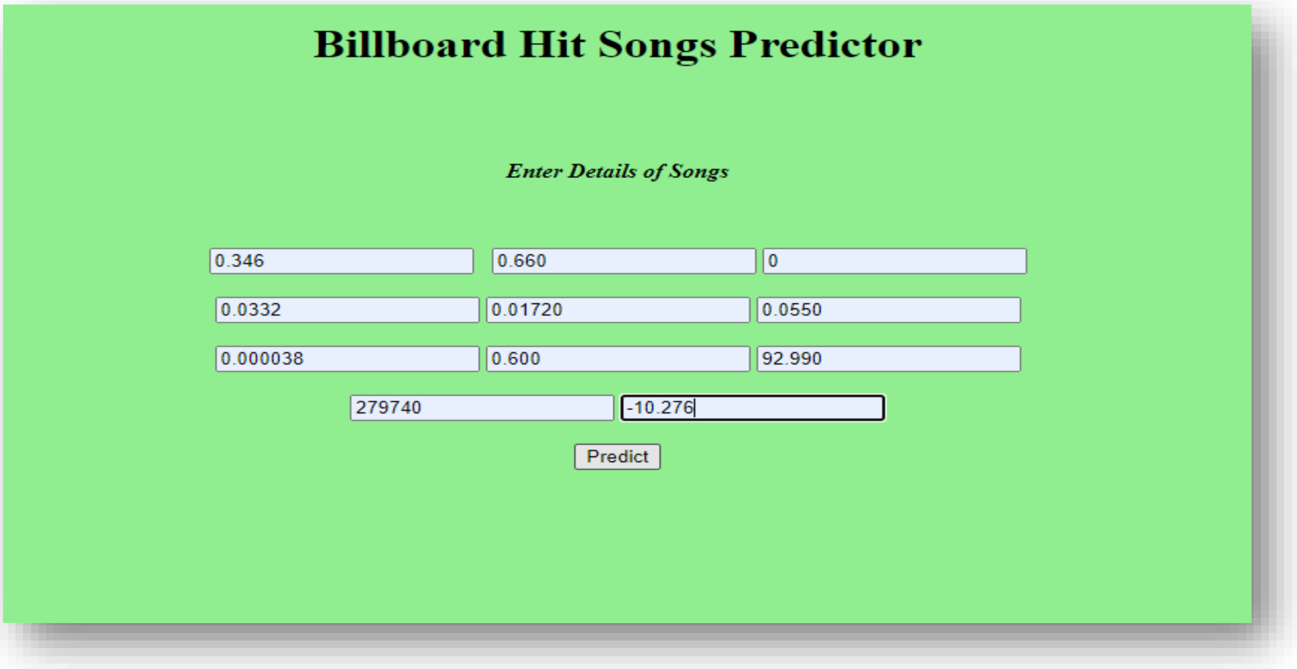
After modeling, I created a flask application which is as follows,



Billboard Hit Songs Predictor

Enter Details of Songs

And after entering the data, the following interface was displayed…



Billboard Hit Songs Predictor

Enter Details of Songs

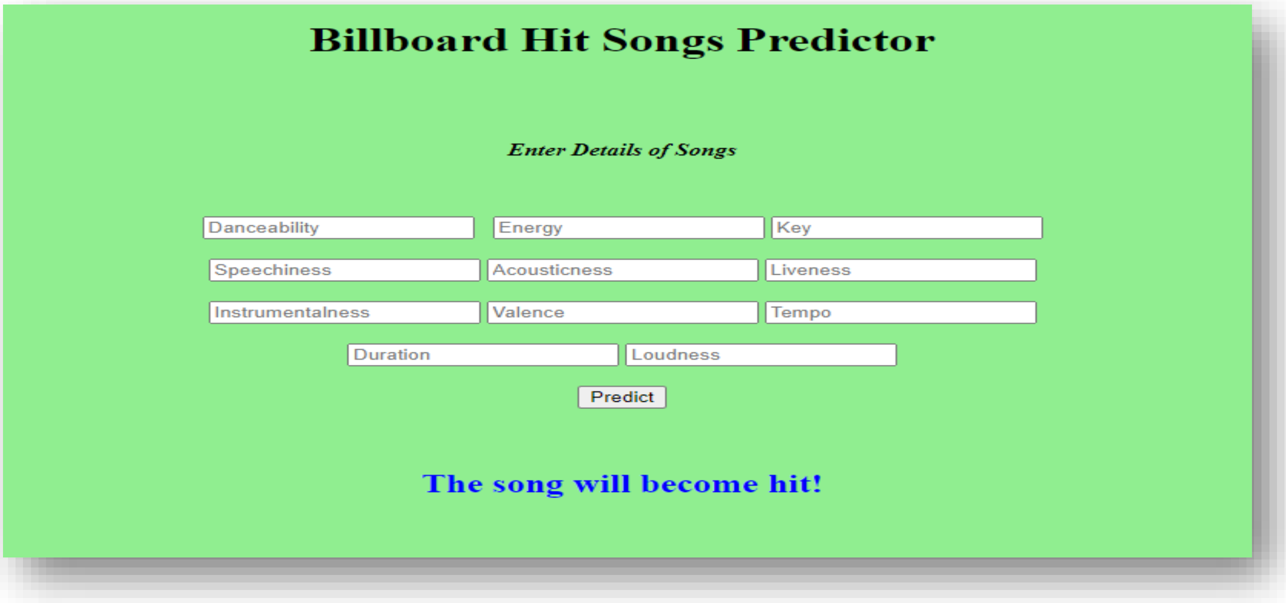
0.346 | 0.660 | 0

0.0332 | 0.01720 | 0.0550

0.000038 | 0.600 | 92.990

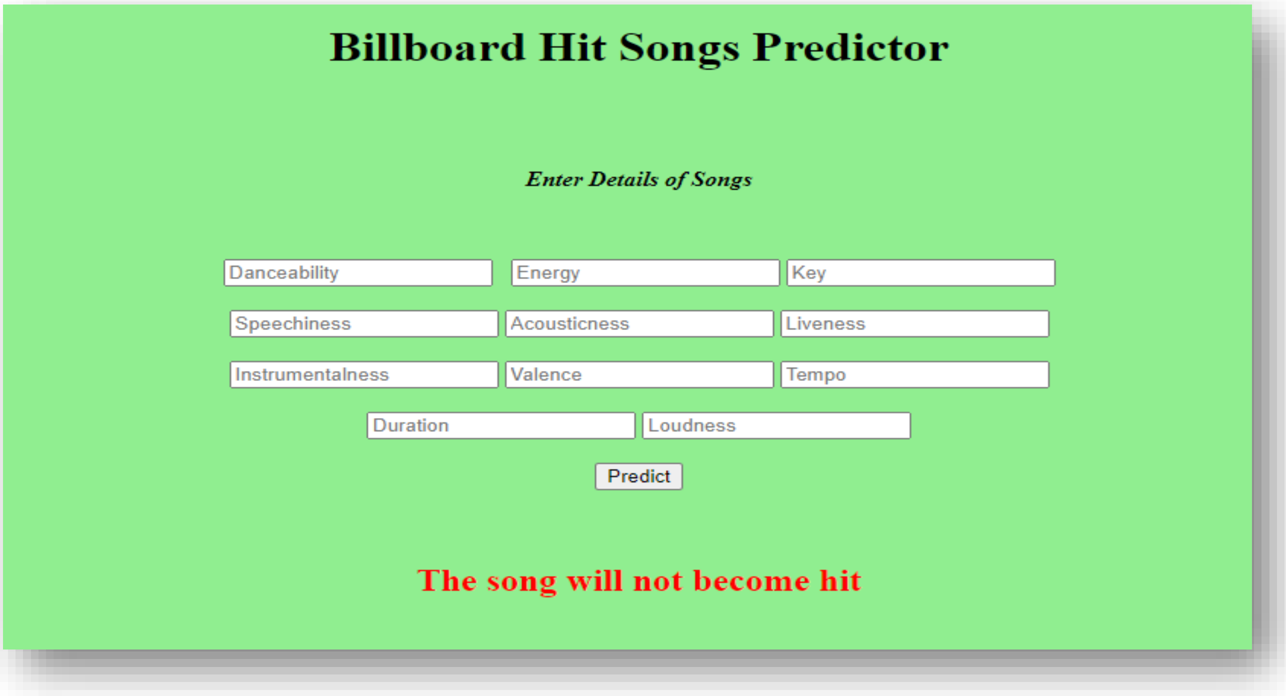
279740 | -10.276

The prediction was whether a song will become a hit or not which is as follows,


Billboard Hit Songs Predictor

Enter Details of Songs

The song will become hit!


Billboard Hit Songs Predictor

Enter Details of Songs

The song will not become hit

CONCLUSION

Billboard hits are nothing new, but I attempted a little part of myself to see if I could apply data science techniques to this sector. How useful will it be if I apply a specific set of algorithms to the dataset? I was told at first that I would get an accuracy of around 75%, but I didn't know which method I would use to achieve that. for that purpose, I started my data analysis and gone through all four algorithms which were assigned to us. While working on those algorithms I originated that random forest is one of the most powerful and widely used algorithms which enables organizations to solve classification and regression problems effectively. It is not only easy and flexible to use but also makes accurate predictions in decision making and also an ideal algorithm which enables organizations and developers in solving overfitting problems in datasets.

The abbreviation API stands for Application Programming Interface. It a connection between computers or computer programmes is known as an application programming interface (API). It's a form of software interface that provides support for other programmes. An API specification is a document or standard that explains how to create and use a connection or interface. The model was created based on a random forest algorithm and that model was then converted into a Flask application.

Flask is a Python-based microweb framework. It is referred to as a microframework because it does not necessitate the usage of any specific tools or libraries. It doesn't have a database abstraction layer, form validation, or any other components that rely on third-party libraries to do typical tasks. Flask is an API in python that allows model developers to build up web applications. The created models can be deployed on many platforms, I referred to open-source platforms like Streamlit, Heroku, etc. I deployed my application on the Heroku platform by uploading my application-related files on GitHub and then connected my GitHub repository to Heroku.

GitHub is an online service that hosts Git-based software development and version control. It includes Git's distributed version control and source code management tools as well as its own. The. git/ subdirectory inside a project is referred to as a Git repository. This repository keeps track of all modifications made to your project's files, creating a history over time. A repository stores all of the files in your project, as well as the revision history for each file. Within the repository, you can discuss and manage your project's development. Heroku is a cloud Platform as a Service (PaaS) that runs on containers (PaaS). Heroku is a platform that allows developers to deploy, manage, and scale modern programmes. Our platform is attractive, adaptable, and simple to use, making it the quickest way for developers to get their products to market. Users of Heroku are charged based on the number of virtual machines required for their apps. The core technology for the Heroku platform and user-created applications is Amazon Web Services. It allows developers to construct applications quickly because it is so convenient. It was a wonderful experience in completing thing project I got to learn various things which are have already mentioned in this report.

REFERENCES

Claesen, M. *et al.* (2014) ‘Fast Prediction with SVM Models Containing RBF Kernels’. Available at: http://arxiv.org/abs/1403.0736.

Dai, H. (2018) ‘Research on SVM improved algorithm for large data classification’, *2018 IEEE 3rd International Conference on Big Data Analysis, ICBDA 2018*, (1), pp. 181–185. doi: 10.1109/ICBDA.2018.8367673.

Musa, A. B. (2013) ‘Comparative study on classification performance between support vector machine and logistic regression’, pp. 13–24. doi: 10.1007/s13042-012-0068-x.

Thompson, J. R. and Licklider, B. L. (2011) ‘Visualizing Urban forestry: Using concept maps to assess student performance in a learning-centered classroom’, *Journal of Forestry*, 109(7), pp. 402–408. doi: 10.1093/jof/109.7.402.

Websites and blogs used for information

1. https://www.billboard.com/charts/hot-100
2. http://millionsongdataset.com/pages/getting-dataset/
3. https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/
4. https://builtin.com/data-science/random-forest-algorithm
5. https://www.geeksforgeeks.org/