MGT - 415

Data Science in Practice

Spotify:

Predicting the success factors of a song

De Becker, Sophie

Di, Yao

Gallese, Mattia

Martiriggiano, Giacomo

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

May 11th, 2020

Table of Contents

1. Introduction

The aim of this data science project is to investigate what determines the popularity of a song. The context of this problem was chosen to be the scenario where we want to release a song and we need to, prior to releasing it, already have an idea of whether it will be successful or not. The US Spotify charts were used, in this context we are thus assuming to be releasing the song in the US. Initially we conduct a traditional binary classification problem to decide if a song is successful or not in general. Further investigations can allow us to determine key characteristics that factor in making a song successful by treating the problem as a regression, also allowing us to take seasonality into consideration.

2. Business Context

In this chapter, the business relevance that sustain the importance of applying analytical tools in the audio industry is presented focusing on the value added to all the stakeholders. After understanding how important it is for a company to implement analytics, the strategy defined for the analysis is explained.

2.1 Business Environment & Main Stakeholders

Music represents an important entertainment tool and cultural heritage for people since the early human history. During time, it signed historical periods, and evolved according to the social trends distinguishing the society. This last characteristic is what inspired our project and drawn our attention on this industry. Indeed, nowadays, with the increasing digitalization, the global music industry made a big step evolving to a completely new level. Innovative companies like Spotify kick started the era of streaming in the audio industry completely transforming the way in which it works for all the stakeholders.

The global music industry in 2018 reached a record value of $19.1 billion that has never been seen in the whole past decade. As said above, this growth is mainly driven by the new trend of digital music streaming. As a proof of this, $11.2 billion over the total value comes from digital channels (59% of the total) such as Spotify (McIntyre, H. ;2019, April 02).

This growth of the whole industry, and this fast evolution of the typical business model, have an impact on all the stakeholders involved in the sector. In figure 2.1 it is possible to observe a graph showing the relation between the different entities interested in the production of a song.

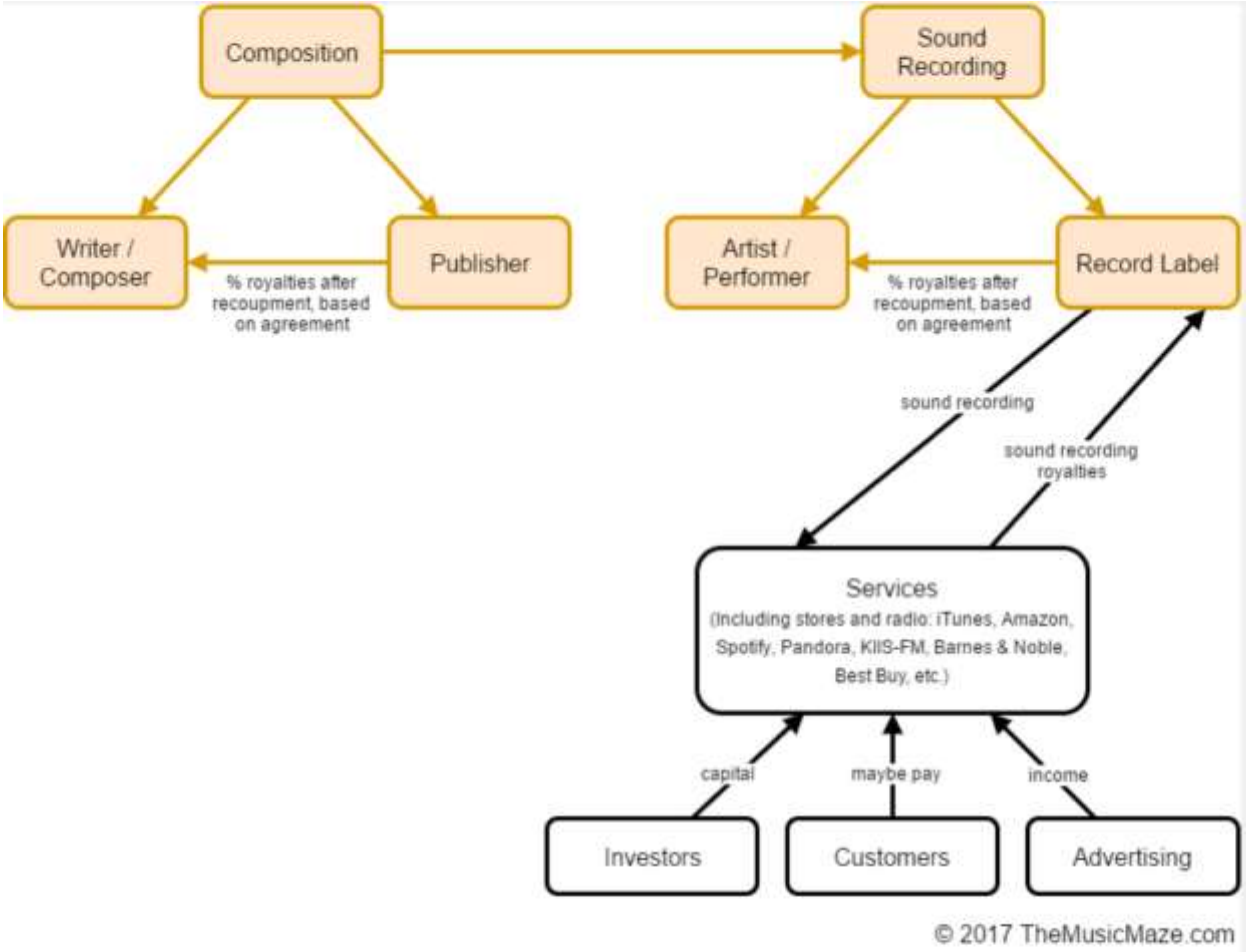

Composition
Sound Recording
Writer / Composer
Publisher
% royalties after recoupment, based on agreement
Artist / Performer
Record Label
% royalties after recoupment, based on agreement
sound recording
sound recording royalties
Services
(Including stores and radio: iTunes, Amazon, Spotify, Pandora, KIIS-FM, Barnes & Noble, Best Buy, etc.)
capital
maybe pay
income
Investors
Customers
Advertising
© 2017 TheMusicMaze.com

Figure 2.1: Music Industry Stakeholders

The focus in this analysis is placed on the sound recording side of the business since our team has been hired by a record label to design a model which help them in recognize which song is worth producing based on its audio characteristics.

Focusing on the Record Label, its profit is composed by different parts that needs to be added to get to the total profit. On the revenue side, the Record Label receives an up-front fee that varies from $250 to $10000 based on different factors as the followings: size of the Record Label, artist's success, number of songs recorded, and Record Label experience and success (Rudsenske, S.; 2018, August 11). This component of the profit is not directly affected by the optimization done through the model, but, on the long term, if a Record Label publishes more successful songs, it will be able to receive higher up-front fees. A second component of the Record Label’s revenue is represented by the royalties that Spotify or other platforms pay based on the number of streams that a song has. Usually, these platforms pay a royalty corresponding to the 52% of the revenue generated by each stream/play of a song. This component of the revenue is directly affected by the implementation of the model as increasing the chance of producing a successful song results in an increase in the number of streams the song has and in the royalty for the Record Label. On the cost side, other than the fixed recording costs, the Record Label must pay a royalty to the artists that corresponds to 15% to 50% of his revenue from the digital platforms. Since the company which committed us the project is Sony, which is one of the three biggest Record Labels in the market, it is assumed that it pays a royalty of 15% to his artists (Sisario, B.; 2018, September 6). To summarize, the profit of the Record Label can be expressed as follows:

$$Profit = Fixed\ Fee + 0.52 * Revenue\ by\ Streams - Recording\ Costs - 0.15 * Royalty$$

After understanding the benefit that the model can bring to the Record Label, it is important to understand also the positive results it will indirectly have on the whole ecosystem.

Considering the artists and the digital platforms, they will also see an increase in profit thanks to the application of the model from the Record Label as their revenue streams are also highly dependent on songs’ success. Moreover, regarding the digital platforms, it is important to underline that they have the power of making deals with the Record Labels to provide them the data they collect.

2.2 Model Design Strategy

At this point, based on the knowledge acquired on the global audio industry, and on the typical business model of a Record Label company as Sony, a strategy to build a model is designed with the scope of identifying if a song will be successful based on its musical features, and to understand the impact main features have on the number of streams a song will have once published.

As a first step, the data on the top 200 global charts have been collected as explained in chapter 3. Spotify is selected as a source of data since it is currently the largest player in the industry among the digital music streaming platforms with a 36% market share (Watson, A. 2019, December 9). Moreover, the company accorded us the permission to use their data for our model.

After collecting the data and exploring the main characteristics of the database (chapter 4), a classification model is designed in order to predict if a song will be successful (chapter 5). With this goal in mind, three variants of the model are tested. In particular, the definition of "success" is different in the three iterations. For each iteration, different supervised learning model are implemented, and based on the metrics specified in chapter 5 the best one is selected.

After realizing that predicting in a reliable way the success of a song is very difficult for the various non observable variables which have an impact on it, a regression model is designed to estimate the impact that each audio feature has on the number of streams a song can reach (chapter 6). Also, in this case, different models are tested to identify the most appropriate one based on the metrics specified in the relative chapter

3. Data Collection

Before beginning the data analysis, we needed to collect a list of relevant songs as well as corresponding information/features for each song. We focused on recent data (2018 and 2019 Spotify charts) since song trends are highly dependent on culture at a given time, thus we cannot directly take music trends from very far in the past and assume that they would still apply today.

The data collection process required a sequence of steps listed below:

1. Go to Spotify 200 charts and select US region. The page can be found using following link: https://spotifycharts.com/regional/us/weekly/latest
2. Select dates of the week of the chart needed and download to csv. The data sets were downloaded manually for each week in 2018 and 2019 (52x2 csv files)
3. Each CSV file of 2018 and 2019 was renamed with an indicator 1-52 marking the week of the year it represented

Example:

regional-us-weekly-2018-1	✓	3/18/2020 6:59 PM	Microsoft Excel C...	20 KB
regional-us-weekly-2018-2	✓	3/18/2020 10:16 AM	Microsoft Excel C...	20 KB

Figure 3.1: CSV Files

4. A Python Jupyter notebook was used to concatenate the 52 weeks of each year respectively and iterate though each song to populate with features provided by Spotify. The features were retrieved using Spotify API and the GET technique (see Jupyter notebook for full detailed code).

The table below presents all the features extracted using Spotify Web API with a small summary taken from the website ("Get Audio Features for a Track | Spotify for Developers," n.d.).

Table 3.1: Features Description

Feature	Description
danceability	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
energy	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity.
key	The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation . E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1.
mode	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0
loudness	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track
speechiness	Speechiness detects the presence of spoken words in a track.
acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic
instrumentalness	Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content
liveness	Detects the presence of an audience in the recording

valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive
tempo	The overall estimated tempo of a track in beats per minute (BPM)
type	The object type: “audio_features”
id	The Spotify ID for the track
Uri	The Spotify URI for the track
Track_href	A link to the Web API endpoint providing full details of the track
analysis_url	An HTTP URL to access the full audio analysis of this track.
duration_ms	The duration of the track in milliseconds.
time_signature	The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure)

Note: In order to verify that we were allowed to use this data for a university project and for educational purposes, we reached out through numerous platforms until eventually the official verified Spotify account responded saying we had permission.

4. Data Exploration

4.1 Data verification – error handling

Since the data was manually collected it has the advantage of being more or less known to us. This means there shouldn't be too many mistakes or missing values assuming the company Spotify is a pretty professional well-established brand. Nonetheless certain preliminary data analysis was carried out to make sure the data was appropriate before applying model training.

- isnull() was used to verify there was no missing value
- the type of object (string, int) was verified to be consistent with the feature it described

4.1.1 Outlier detection

The outlier detection step can be found at the start of 'EDA with KPI 2', it was carried out before detailed analysis or modelling was conducted on/with the data. The 10 continuous variables: ['danceability', 'energy', 'loudness','speechiness', 'acousticness','instrumentalness', 'liveness','duration_ms','valence', 'tempo'] were used to apply Mahalanobis distance with a 1% p-level. The test assumes normal distribution of the continuous variable which isn't too unreasonable of an assumption. However, it isn't completely accurate since we do see some graphs are quite different from normal distribution, but nevertheless it is still useful to test if this method returns any outliers. Using chi-square test and comparing with MD squared values we conclude that there are no apparent outliers which suggests, or dataset is consistent.

4.2 General trends and pattern

The distributions for certain feature were plotted for years individually and for all the data combined. The graphs superimposing different years shows that there are already small differences in distribution between 2018 and 2019, reinforcing the idea that music trends evolve and confirming we were right to look at the two most recent years for our analysis. The overall distribution show features are often centered around single peaks, such as for 'duration', but it isn't always the case. The 'tempo' graph for example has 2 peaks of comparable height (proportion of songs). The entire set of features distribution is further explored in 4.2.2.

4.2.1 Initial KPI

The EDA for the first KPI was generated before redefining the success definition, it is left in the report for completeness, in case the information is of interest to the reader.

One aspect that could be investigated when considering songs at individual weeks was seasonality. Averages of different feature between weeks 1-52 are shown on plots in section ‘Exporing possible time patterns’ of the Jupyter Notebook. We can see seasonality is a factor which may be crucial in determining the success of a particular song at a particular week. This type of information wasn’t included in our classification problem where our goal is to input a song and predict if at some point it reached top 20, but we see week number must be a determining factor in regression problems.

4.2.2 Selected success definition

Continuous variables:

In this section the graphs and figures relate to the final selected success definition (KPI 2) for top or not explained in section 5.2

One of the main observations from this EDA is the very small perceptible difference in distribution peaks for top and not top songs. This already indicates that the classification problem will be difficult to implement, it will be difficult to obtain high performance scores. The similarity in distributions can be seen in the graphs below for ‘loudness’, ‘danceability’ and ‘liveness’. When multiple peaks are present, for example with ‘tempo’ or ‘liveness’ even showing similar secondary peak for both top and not top songs. Although we do see for the top the frequency distribution has a higher bin in the center.

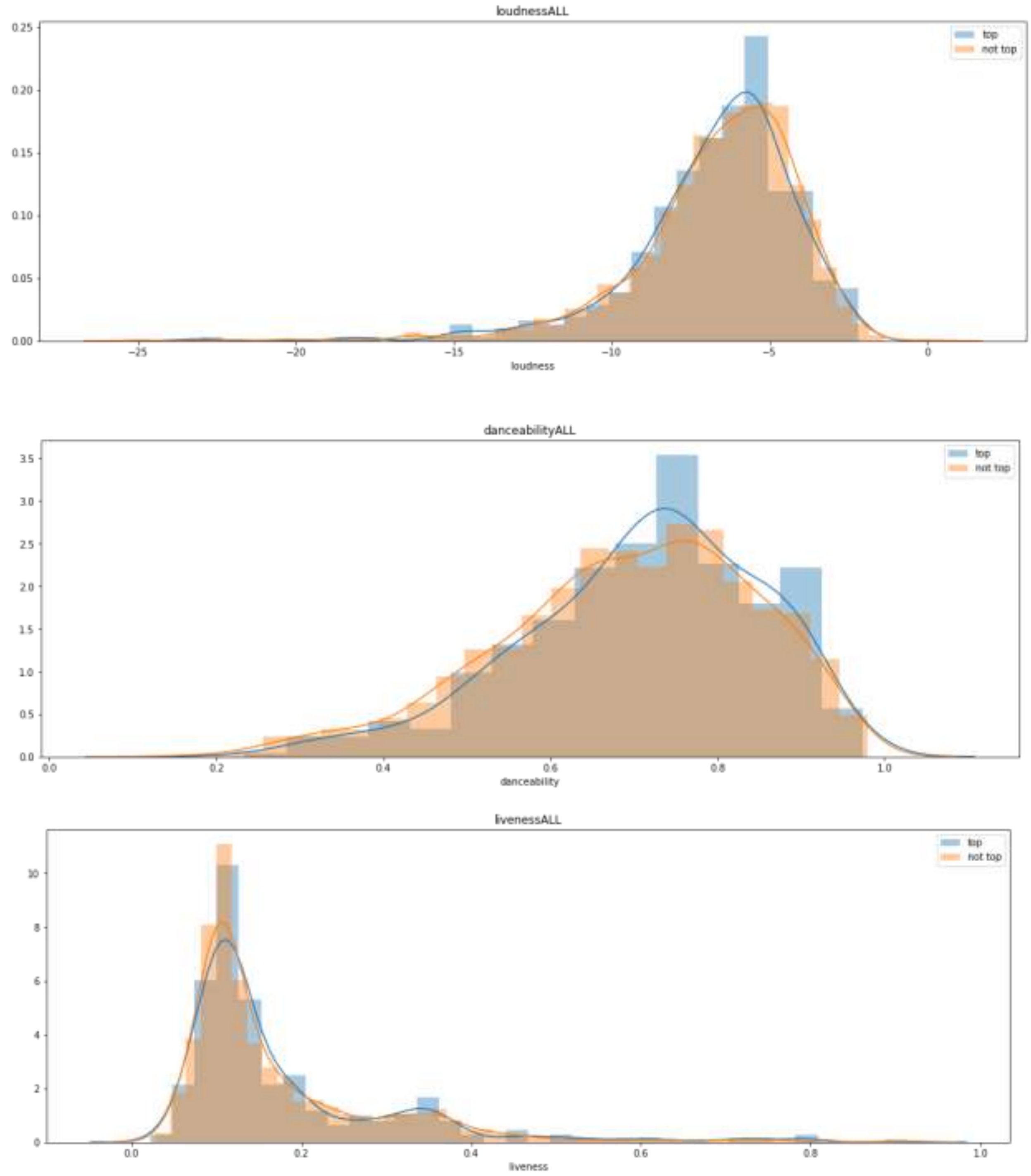


Figure 4.1: Probability Distribution of 3 Continuous Features amongst Songs using sns.distplot with Kernel Density Estimate

When considering certain other features, we can see some small variations as seen in the figures below for ‘valence’ and ‘energy’ where there is a slight translation of the distribution center, nevertheless it isn’t as striking as the clear distinction between peaks which can be sometimes observed in sample classification problems. The two distributions still are mostly overlapping each other.

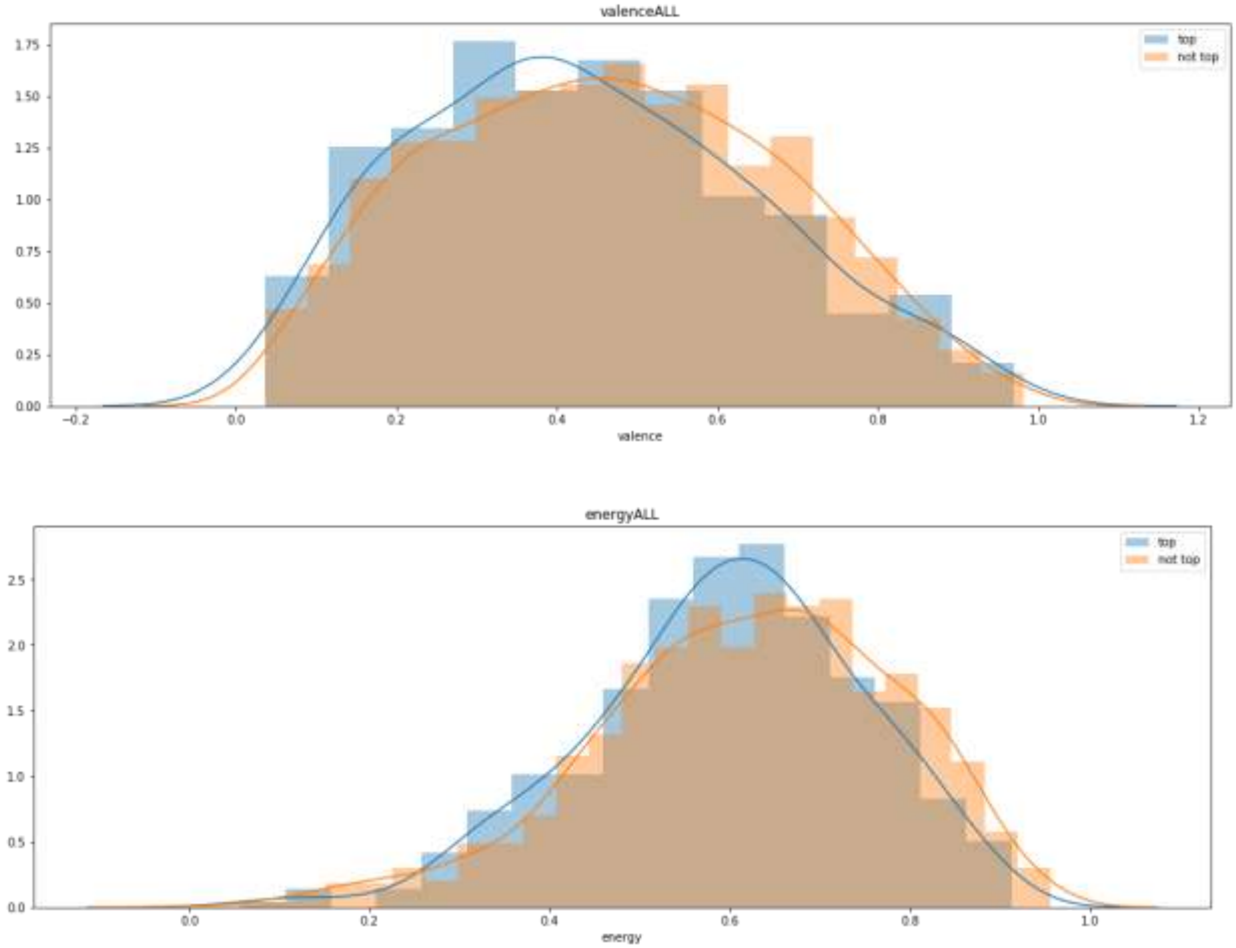


Figure 4.2: Probability Distribution of 2 Continuous Features amongst Songs using sns.distplot with Kernel Density Estimate. Showing Slight Difference between Top and Not Top

The general distribution of continuous features can be seen at the beginning of ‘EDA with KPI 2’ section of the Jupter Notebook in the figure titled ‘Distribution of different features for dfallnondup dataframe’. Most of the features have single peaks where most songs are concentrated, sometimes the curve appears symmetric (ex:‘duration’), other times it appears more skewed (ex:‘energy’). As mentioned before, we see certain features have multiple peaks (‘tempo’ and ‘liveness’), this isn’t surprising for the music industry. Indeed, this can be due to the fact there are multiple dominating music styles with comparable or non-comparable popularity which causes multiple peaks and not a one-size fits all most common music type. In general, we see the assumption of normality doesn’t always hold, in addition to mulitple peaks there are also distribution which behave in a more exponential decrease way such as the one for ‘speechiness’.

In order to attempt to gain more understanding of possible pattern between features and best rating of a given songs scatter plots were generated with features associated with a probability ['speechiness', 'acousticness','liveness', ‘instrumentalness’]. Foreach best position (200 to 1) the average of this feature was plotted. Another data agglomeration method was used where for every distinct measure of a feature we plot the average best position of songs recorded for that feature. The graphs didn’t result in any clear indication of observed pattern. The random appearance of the cloud of point reinforced the idea that it would be hard to discern between a top vs not top song.

Discrete Variable:

In the EDA part, the key and mode features has been analyzed. They are the most important composition elements sine they can be useful for describing and classifying music. Indeed, they provides useful insights into a song since it is a descriptive quality and can invoke several different emotional responses.

It is interesting to note that the most common key in data set is C#, which is also known as D♭. Analyzing only the non-top songs, the key C# is significantly the most used key. While, considering only the top songs, it can be observed as the gap among the other keys is drastically reduced, specially between C# and C.

All the keys can be in major mode or minor mode, and the combination of different keys and modes can arouse several emotions. The C# in major mode arouses an emotion of innocently happy to the listeners, while C# in minor mode arouses an emption of innocently sad and love-sick.

When looking at the impact we see that the proportion of top songs for the key C is highest, the distribution tends for songs in C appears to lean more towards the top 20 than for the other keys. The analysis reveals the impact of mode on ratings to not appear very significant.

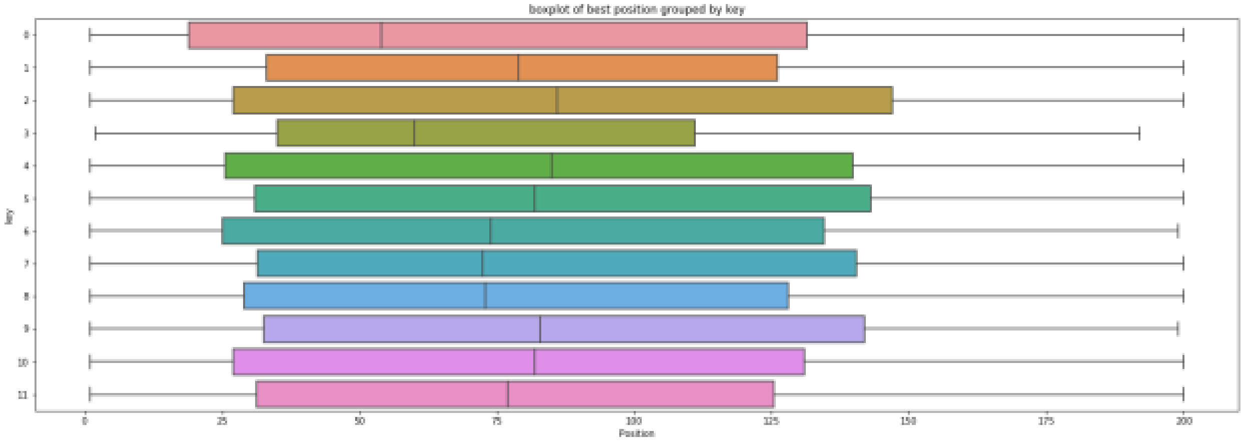


Figure 4.3: Box Plot for Best Rankings of a Song for Different Key (Standard Pitch Class Notation i.e 0=C) of the Song

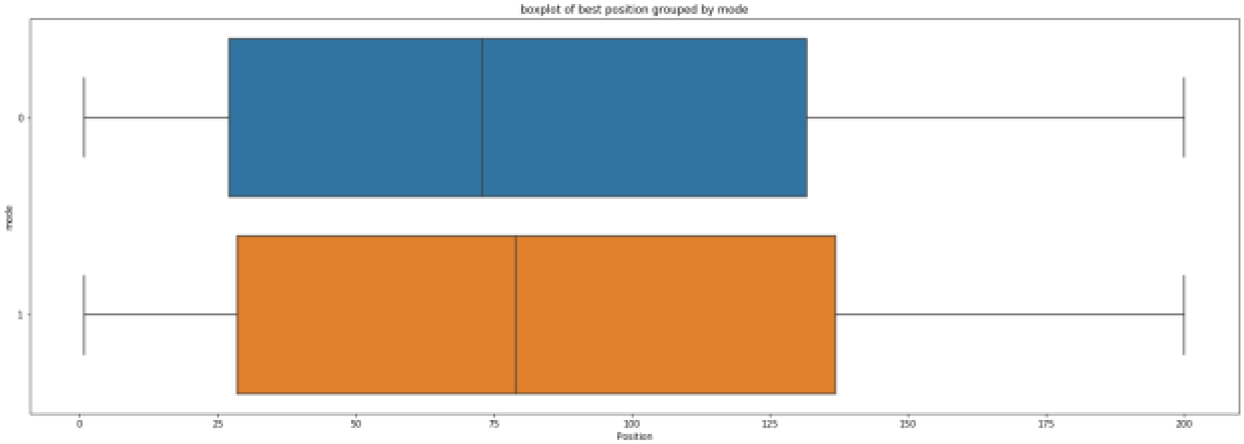


Figure 4.4: Boxplot for Best Ranking of Songs for Different Modes. 1:Major and 0: Minor

5. Classification Problem

In the following chapter, the process to design an appropriate classification problem for the business context is described.

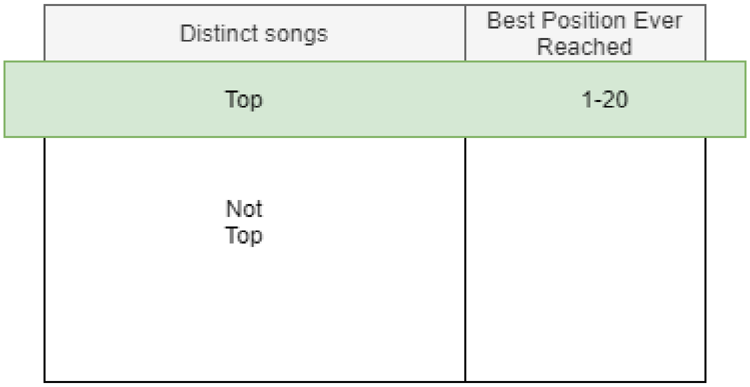
5.1 Initial Idea

The first measure of success, or KPI 1 is defined as follows: all instance of a song in a weekly chart is treated as a distinct entity. A song is considered a hit if it is in the top 20 of 200 so top 10%. This means that a certain song may be a hit one week and not a hit a subsequent week. Results remain in the Jupyter Notebook, however we quickly decided this may not be the best classification criteria since it is counting a song multiple time if it appears multiple times in the ranking. This means that the same song could be successful one week and not successful another. For a classification problem, we would prefer an absolute criterion that classifies the song in general as successful or not regardless of the week. This leads us to the KPI that was chosen (5.2).

5.2 Chosen success definition

The second success definition, the one chosen for the classification analysis, is as follows:

We keep the best (closer to 1 out of 200) position ever reached by a song and drop duplicates. If a song every reached top 20 it is classified as a hit or 'top' song in Jupyter notebook.



Distinct songs	Best Position Ever Reached
Top	1-20
Not Top	

Figure 5.1: Schema of Success Definition in Dataset

The KPI chosen (KPI 2) seemed more appropriate since it defined a song's success as a whole for the classification problem (not week dependent).

5.3 Model Training and Selection Method

Below is a schema of the high-level approach to train and subsequently choose the best model. In order to solve a classification problem, we considered models appropriate to this application: Random forest, SVM and Logistic regression. All three can be used for a binary classification problem which is our scenario (top (1) or not (0)). Below is an overall map of the steps taken to arrive at the selected 'best' model.

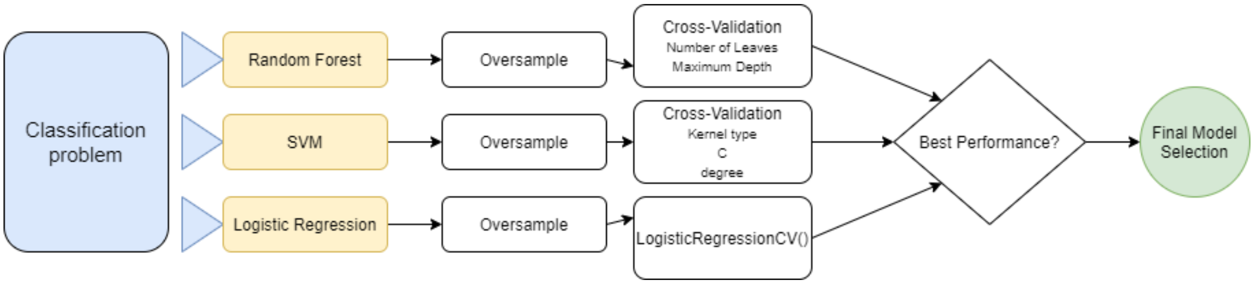


Figure 5.2: Process Flow Chart of Method Used to Identify Best Version of Classification Model

Oversampling: oversampling was clearly necessary in our case since the classification is very unbalanced with a lot more 'not tops' (0) than 'tops' (1).

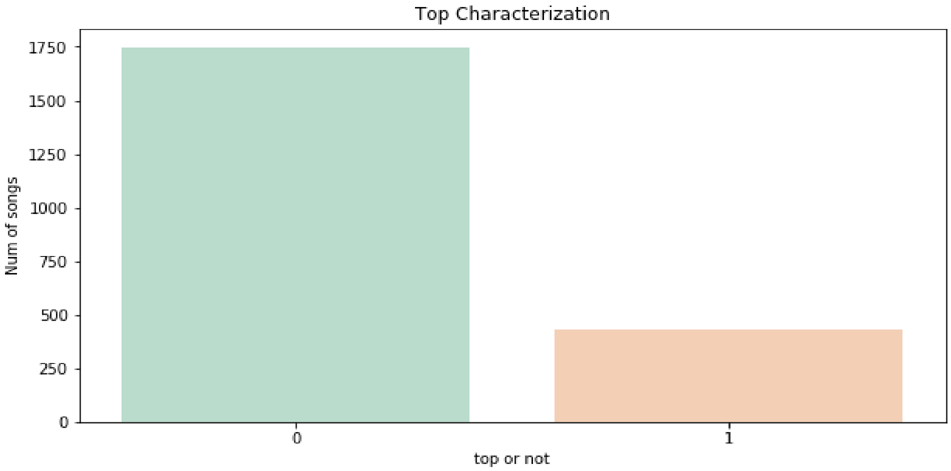


Figure 5.3: Representation of Uneven Distribution of Top and Not Top Songs after Duplicates of Songs in Different Weeks

Train-test splits: In all three models the training set was 80% of the total data set and the split was done using 'stratify'. The oversampling was only done on the training data since we still want our test data to represent a realistic distribution of tops and not tops. The Oversampling was done using *sample()* random sampling with replacement set to ‘True’. One drawback of this is that during cross validation we are tuning considering this new oversampled set which may not give the most accurate results since the test set used within training set during cross validation don’t have same distribution as final test set.

Cross Validation: The cross validation was carried out by varying different parameters and using 5-fold cross validation on the training sets to select best parameters for respective models using f1 score as an indicator of performance.

Below is a more detailed schema of the way training split, data modification and testing were carried out during the model creation/training process

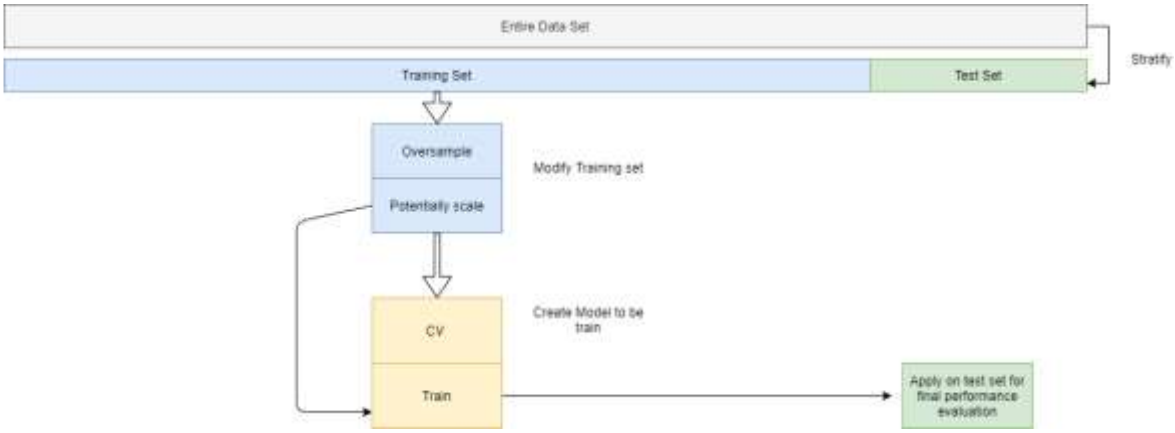


Figure 5.4: Schema of Train-test Split and Cross-Validation Process

It is possible that more optimal method could be used since the parameters in this case are tuned using a balanced train and test set but the test set itself keeps the unbalanced distribution. Ideally, cross-validation could be ameliorated by having unbalanced cv test sets.

5.3.1 Random Forest

Model: RandomForestClassifier()

Tuned Parameters during CV: 1) maximum depth of trees and 2) minimum number of leaves.

Result:

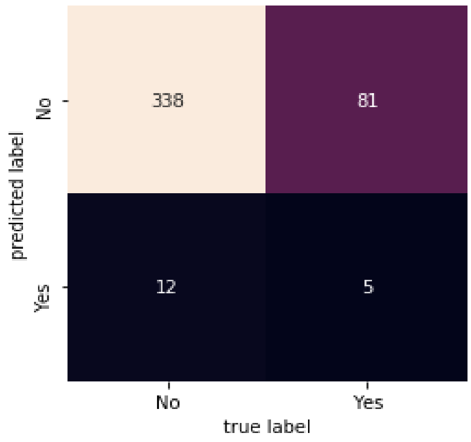


Figure 5.5: Confusion Matrix for Random Forest

F1 score: 0.097

5.3.2 SVM

Model: svm.SVC()

Tuned Parameters during CV: 1) Kernel Types 2) degrees or C

In SVM it is often best to scale data to avoid according too much importance to a variable just because of its scale. Standard scaler was used in this case to scale the features before doing the train-test split.

Result:

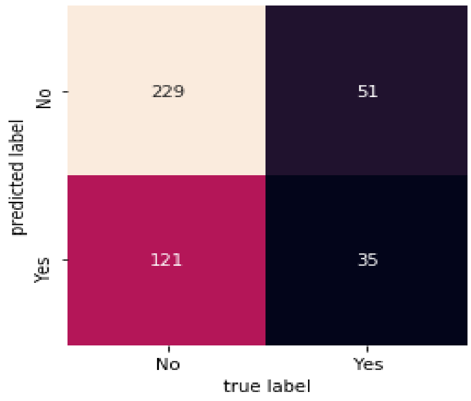


Figure 5.6: Confusion Matrix for SVM

F1 score: 0.289

5.3.3 Logistic Regression

Model: Logistic Regression

Tuned Parameters during CV: Penalty Cs

In this part, we take advantage of the binary convergence property of the logistic function to achieve the binary classification on top and non-top music. We select the trial hyper parameters by uniformly mapping on the logarithm axis. Then we applied the five-fold cross-validation to find the model with maximum average f1 score.

Result:

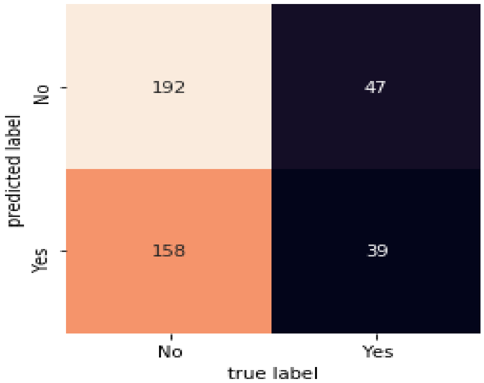


Figure 5.7: Confusion Matrix for Logistic Regression

F1 score: 0.276

5.4 Final Selection of Classification Model

Using F1 score as measure of performance the best classification model appears to be SVM with F1 of 0.289. It isn't very different from logistic regression classifier with F1 of 0.276. If we had a specific cost associated to missed opportunity of a song and lost money invested in non-top song, we may more accurately choose which model is the most advantageous by seeing which one minimizes cost:

$$P(model\ predicts\ it's\ a\ hit, but\ it\ isn't) \ * \ cost\ lost\ investment$$
$$+ \ P(\ model\ predicts\ it's\ not\ a\ hit\ but\ it\ is) * cost\ of\ lost\ success$$

It is also important to note that the performance scores change from one run of the code to the next and SVM and logistic regression being very close the best performing one will likely change for different runs of the same code.

1. Regression Model

The following chapter presents the process to design an appropriate regression model. As our classification models are time-independent and incapable to predict cardinal variables, we introduce the time-dependent regression algorithms as the supplementary. The regression models are applied to address two types of problems: predicting the future performance of the songs already launched on the market and forecasting the streaming times which is correlated to the singer's profit.

6.1 Position Prediction – Use Regression to Predict the Top Songs

6.1.1 Problem Definition

From the business point of view, it will also be valuable if the company could predict the trend of the songs which have already been launched. If the prediction is accurate enough, the customer could adjust their strategy agilely based on the future performance of their songs. The potential actions could be investing more on the positive trending song and relocate the resource assigned for the songs which will become out of fashion, which could generate greater profit for the company and the stakeholders.

As the classification object "Top" is based on the value "Position", one of the possible solutions is to change the classification problem to a regression problem on "Position". The classification will be done based on the predicted "Position" value. This approach will also avoid the data unbalance problem as we observed in the previous section

Different from the approach in our classification model, as the position is a time-dependent value, the model will be week-based. Some dynamic features will be introduced to improve the accuracy of the time series approach.

6.1.2 Feature Engineering

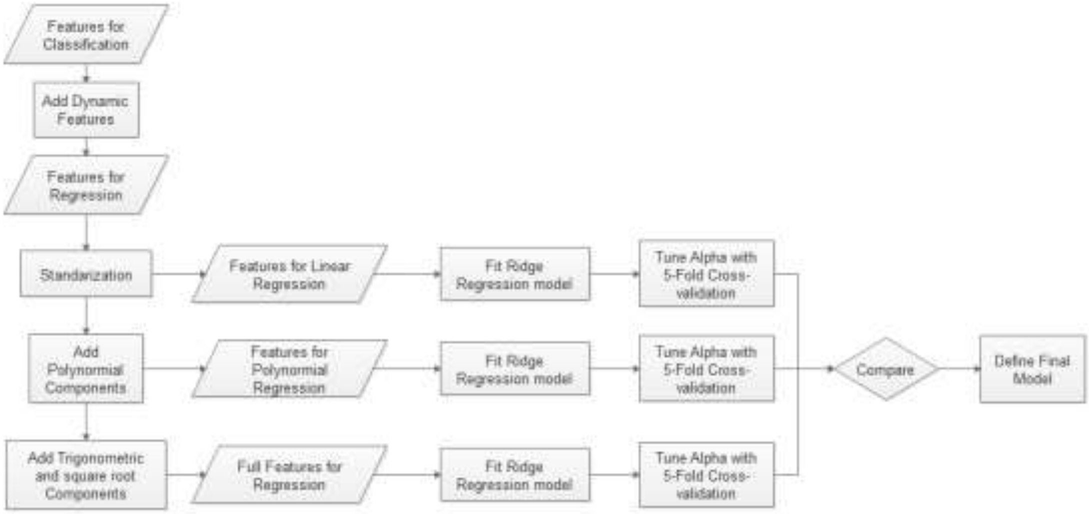

Features for Classification
Add Dynamic Features
Features for Regression
Standarization
Features for Linear Regression
Fit Ridge Regression model
Tune Alpha with 5-Fold Cross-validation
Add Polynormial Components
Features for Polynormial Regression
Fit Ridge Regression model
Tune Alpha with 5-Fold Cross-validation
Compare
Define Final Model
Add Trigonometric and square root Components
Full Features for Regression
Fit Ridge Regression model
Tune Alpha with 5-Fold Cross-validation

Figure 6.1 Flow Diagram for the Regression Process

Our feature engineering process includes two main parts:

1. Select and adding the dynamic features
2. Mathematical feature augmentation, which will be detailed in the model selection section.

Besides keeping all the music quality features we used in out classifier model, we try to understand which kinds of time-series data could be introduced to our regressor.

Normally, the Spotify users may try to listen the songs which has already been popular. Therefore, we introduce the new features recording the stream in the last 5 appearances in the list and the position in the last list.

However, if the popular song stays too long on the top lists, the user may consider it "out-of-fashion" and streams less. In order to take this case into consideration, we introduced two other features:

1. **week-count**: count how many times the song has been in the top 200 list
2. **week-prev**: when was the last time the song appeared in the top 200 list.
3. **week**: current week of the estimation, working together with the week-prev, we could observe if the song has felt out of the list in the past period.

6.1.3 Model Selection

The Ridge Regression is chosen as the main regression model applied. The advantage of the Ridge Regression model could be concluded into two points: the penalty component in the model prevents overfitting as well as the algorithm consumes reasonable amount of calculation power.

We would like to compare the performance of the three models.

1. Linear Regression:

   As Ridge Regression is based on linear regression, we just need to standardize the features to avoid the bias led by dimensional unbalance.

2. Polynomial Regression:

   Currently, there is no theoretical proof or evidence defines the relationship between position and our selected features. Our first assumption based on linear relationship is not convincing enough. In order to avoid the functional form misspecification, we need to introduce more complex model for validation purpose. The polynomial relationship is a natural guess.

   Instead of changing the fitting model, we would like to take the advantages of Ridge Regression. Therefore, we chose to augment the original features to introduce the polynomial terms.

3. Full Features Regression (linear + polynomial + trigonometric + square root):

   The full feature regression model is selected with the similar consideration and modelling methodology as the Polynomial Regression model. Besides the polynomial terms, we also add the trigonometric components and the square root of the original features.

   For each model, 5-fold cross-validation is applied to tune the hyperparameter Alpha in the Ridge model. The model with maximum average F1-score is chosen. Classification is performed based on the predicted position from the selected model.

6.1.4 Results

By classifying with regression on ‘Position’ parameter, we observed our most accurate model (Accuracy Score = 0.92) with the highest F1 score (F1 score = 0.77).

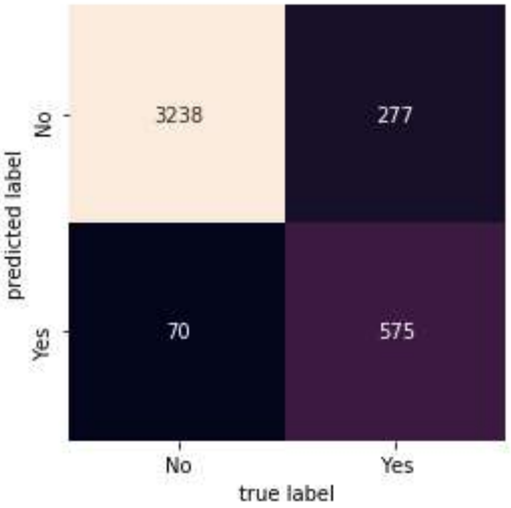


Figure 6.2: Confusion Matrix for Classification with Regression Approach

The following reasons may explain why the regression-based classifier has better performance:

1. The regression model includes the dynamic parameters which could reflect the current trend of the music.
2. Comparing to the classification, the regression model assigned different values on the songs close or far from the separation boundary between top and non-top songs. The soft boundary generated with the regression-based model is more comparable to the real-world model.

6.2 Stream Prediction – Find the Decisive Factor of Earning

6.2.1 Problem Definition

After building up a reasonable model in the previous section to analysis which song could hits the top 20, we would like to understand how much the singer could benefit from their top-ranked songs. Based on the current business model of Spotify, streaming number is the decisive factor for the singer's profit (CNBC, 2019). The income of the singer from Spotify is correlated to the number of plays rather than the weekly ranking of their songs’. Therefore, the streaming number could be a better measure for the success of a song according to the financial interest of the singer.

6.2.2 Modelling

The modelling process for stream prediction is similar as that for position prediction. We are using the Ridge Regression model, and archiving linear, polynomial and full features regression with feature augmentation. The only difference is that the dependent variable is changing from ‘Position’ to ‘Stream’.

6.2.3 Results

The best regression result we could achieve is with the polynomial model with the evaluation criteria of maximum average R2 score of the regression. Even with the best model, we could only achieve 0.67 for the R2 score, which means only 67% of the variance in the Stream data could be explained by the model we found.

7. Business Insights

Why classification model doesn’t enable us to produce highly precise conclusions: it is important to recognize that a lot of not-measured factors are involved in making a song popular. It isn’t possible to expect a perfect recipe, if it was then everyone would already be using it. There is a lot of factors that are up to luck and can’t be translated into mathematical numbers. Sometimes when a song is included in a tv-show or movie which is very successful it suddenly gets a huge boost and becomes very popular. An example can be ‘Bella Ciao’ which became very popular recently after its inclusion in the Hit Netflix TV show Casa de Papel. It can also be pure serendipity, for example the song “shallow” initially made recently popular by the movie “a star is born” shot back up in the charts because a woman sang it in the UK tube in a Youtube video that went viral (Cachero, 2020).

It is also important to notice that even if a song is classified a hit and another not a hit it doesn’t necessarily guarantee it is better to produce the hit. Indeed, a song may never reach top 20 but stay in top 50 for may many weeks and as a result bring in more money than a song who reaches top 1 once and is just a one-hit wonder. We always need to remain critical when using a certain measure and recognize its limitations in measuring true success of a song

The reason that it may be hard to classify ‘top’ vs ‘not top’ song also stems from the fact that we are looking at the top 200 songs which already imply they are all somewhat successful and not complete flops. It may be easier, if we had access to a large database, to classify songs at the top 10% among the entire population of songs released in a given time frame.

8. Conclusion

In this report, it was designed a model for Sony to identify which song is worth producing based on its audio characteristics. This model would allow the company to increase the number of streaming and profits. To build the model, the team selected Spotify as a source of data, because it is the most popular audio streaming platform.

In the first step, the criteria for defining a song as successful were settled. A song was considered a hit if it reached the top 20 positions at least once in 2018 or 2019. At this stage, the team faced its first challenge since the success of a song was previously based on weekly period. However, this caused many issues because a song could appear many times in the ranking. Three different classification models were developed, and their performance were compared based on accuracy. However, no model performs extremely well since it is very hard to model the industry through the classification analysis.

Therefore, the problem was tackled from another point of view, the team decided to predict which songs, among the ones already launched on the market, would be the most successful based on the analysis of historical data. In this way, Sony could generate greater profits by relocating resource on potential successful songs. To implement this process, the ridge regression model was applied.

The model has high margin of improvement in order to be more reliable. In future, the team will implement new techniques to increase the accuracy level of the model. First of all, it is missing the link between the singer and the estimation. The estimation accuracy may improve significantly if the singer’s characters is included into the database. Secondly, it could be put more weight on the time-based factors since the selection of the song may depends on the season and holiday period. Last but not the least, the team could also try to improve the mathematical model for the estimation, e.g. introducing neural network which allows us to identify the possible links we may have missed in our causal analysis.

REFERENCES

McIntyre, H. (2019, April 02). The Global Music Industry Hit $19 Billion In Sales In 2018, Rising By almost 10%. Retrieved May 01, 2020, from: https://www.forbes.com/sites/hughmcintyre/2019/04/02/the-global-music-industry-hits-19-billion-in-sales-in-2018-jumping-by-almost-10/

Rudsenske, S. (2018, August 11). How Does A Producer Get Paid, Retrieved May 02, 2020, from https://www.musiccontracts.com/blog/2018/4/23/first-steps-to-setting-up-a-label-sjk55

Sisario, B. (2018, September 6). A New Spotify Initiative Makes the Big Record Labels Nervous. Retrieved May 02, 2020, from https://www.nytimes.com/2018/09/06/business/media/spotify-music-industry-record-labels.html

Watson, A. (2019, December 9). Music streaming market share. Retrieved from https://www.statista.com/statistics/653926/music-streaming-service-subscriber-share/

Get Audio Features for a Track | Spotify for Developers. (n.d.). Retrieved May 5, 2020, from https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/

CNBC. (2019, 07 31). Spotify Revenue Beats but Adds Fewer-than-Expected Paid Subscribers. Retrieved 05 09, 2020, from CNBC: https://www.cnbc.com/2019/07/31/spotify-revenue-beats-but-adds-fewer-than-expected-paid-subscribers.html

Cachero, P. (2020, February 20). A woman was randomly challenged to sing “Shallow” in a train station, and she blew people away. Retrieved from https://www.insider.com/woman-finish-lyrics-challenged-sing-shallow-subway-station-viral-2020-2