MATHS 7107 Data Taming Assignment Final Report

Chang Dong

2023-04-24

summary

This Essay presents a project that mainly focuses on predicting the genre of the songs on Spotify using three machine-learning tools. This project involves importing data, cleaning data, Exploratory data analysis(EDA), PCA analysis, building three models, evaluating and comparing their performance, and making a fin al prediction using the best model. The model used in this project is Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), and Random Forest(RF). This essay discussed the influence of each categorical variable and numerical variable on our response variable. The result shows they all show some extent of influence on the response variable. And PCA dimension reduction shows little influence on our data set. Then we discussed the performance of the three models, and the result shows the best model is Random Forest which has the highest accuracy of 0.561 and the highest ACU of 0.850 under the cross validation mean result. As for each class, the "rock" class is almost the easiest to classify among the 6 classes, in Random Forest Model, the Sensitivity shows an almost 80% high score, and the ROC curve is an almost round right angle. And the Random Forest performs worst in "pop", compared with LDA and KNN worst in "r&b", which are all lower than 40%. Finally, the best model performs also stable in the test data set, which reaches similar accuracy and sensitivity in each class, and the most important features in the VIP are track_album_release_year, "danceability", "speechiness", and "tempo" accordingly. Overall, the model is efficient to make some predictions, especially in some classes.

1.Method

In this project, R studio(R 4.2.1) was used to build the whole work. And the pakages that we imported includes "skimr, tidyverse, tidymodels, themis, recipes, dials,kknn, vip,forcats,caret,MASS,discrim,yardstick,pROC". In this section, the methodology of this project will be introduced. Generally, we will introduce how we imported the 'spotify_songs_origin' data set, cleaned the data set, made some Exploratory Data Analysis, build the 3 models to make the prediction, and the evaluation of the 3 models.

1.1 Data import

The spotify_songs data set was obtained from the URL:'https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/2020-01-21/spotify_songs.csv'.

1.2 Data Cleanning Method

In the daat cleaning stage, we checked the data type of the 23 columns' varibales, and these varibles can be divided into 3 type, text, categorical data and numerical data. The text and categorical data can be distinguished by the unique data in this column, these variables of "track_id", "track_name", "track_artist", "track_album_id", "track_album_name" have more than 10,000 categories, so they should not be considered as the categorical variable. And these variables can not be used as the predictos in our model, because our model is not a language model, that can not process the characters, so these variables were removed from the data set.

After filtering these text, the rest of the character type variables like "playlist_name", "playlist_id", "playlist_genre", "playlist_subgenre" have less than 1000 categories, and it is reasonable to consider them as the categorical predictor. So these variables were converted into factor types. Finally, the rest of the variables should be the numerical variable, they are all countable or continuous. Especially, the year in the "track_album_release_date" variable was extracted and converted into numerical.

After convert the variable into the correct type, all the rows containing missing values were removed.

1.3 Exploratory Data Analysis(EDA) Method

After Cleaning data, these predictor candidates along with the response variable playlist_genre were piped into the Exploratory Data Analysis Stage. In this Stage, two different analysis according to the variable type was carried out.

1) For the Categorical variable, we checked the count of each category in the categorical variable(including the response variable, playlist_genre). And the chi^2 test was applied to the 3 variables, "playlist_name", "playlist_id", "playlist_genre", and "playlist_subgenre" with the response variable, which aims to check the correlation between them and the response variable. This can guide us on whether a variable is suitable for being a predictor or not.

2) For numerical variables, we did 3 things.

The first is the distribution of each numerical variable was inspected so that we can have a better understanding of this variable.

The second is the boxplot of this variable in the six response variable class, and this can help us judge if there is an obvious distribution difference between different genres. Such as if all the six genres of this variable have the same distribution, we can consider this variable has no obvious influence on the response variable. Using a boxplot can let us understand this matter more intuitively.

The last one is the ANOVA test which has the same function as the second one, but this is a statistical distribution differences test among more than 2 classes, so it has a better statistical interpretation than the second way. ANOVA test is a test to check the distribution differences between different classes, if they are different, we can believe that the variable shows a different distribution in a different class, so the variable has influence on the response variable. I built a standardized function containing the above 3 operations to process the 14 numerical variables using a for-loop. In addition, we also investigated the track popularity change over time.

1.4 Model Implement Method

Before this Stae, in order to reduce the dimension of 14 variables space, I tried to apply PCA(finally not applied, reason will be explianed in next part), and reduced the data set size to 1,000 songs per genre due to computing power. Then the reduced data set was divided into training and testing set according stratified sampling method to keeps all the class balanced in both training and testing set. After that, two methods was applied to process the training set, bootstrap and cross-validation. Bootstrap is a sampling method with replacement, and we do this process 10 times to get 10 data set. And bootstrap data set was used to tune the best (hyper-)parameter. While the cross-validation(cv) is spliting method, the 5 validation set account for 20% of the total training set size, and these 5 set are not overlapped. Thes 5 cv data sets were used for the next stage

1.5 Model Evaluation Method

In this stage, the cv data sets were used to train the 3 models in the best hyper-parameter and find the best one of the 3 models. Mean AUC and accuracy, detailed ROC curve and sensitivity in six classes were discussed in this part. After selecting the best model, we used the test data to make a prediction and compare the prediction with the ground truth, and evaluate the final performance of the best model.

2.Result

In this part, we mainly focus on the result of the Exploratory Data Analysis(EDA) , in EDA we will answer the 3 questions that Spotify founders care about through our results. Besides, I tried to use PCA to recude the dimension. And finally in this part, we will also introduce the result of our modeling.

2.1 Exploratory Data Analysis(EDA)

1)Categorical Variable

playlist_id: This variable has 471 unique categories, and the distribution of the counts of the 471 categories shows in Fig1. It is a left-skewed distribution from the histogram, and most of the categories are less than 100 counts with minor exceeding this value. The minimum counts is around 0 and the maximum counts is around 250

playlist_name: This variable with 449 unique categories has the same property like the playlist_id, similar distribution shape shows in Fig2. Because each playlist_name roughly coressponds to a specific playlist_id, so they shows the similar pattern.

playlist_subgenre: This variable has 24 classes. According to the bar chart of Fig3, these classes are not balanced but not very imbalanced, because most of the counts are within the range of 800-1750. The highest value is around 1750 counts which corresponds to the "progressive electro house" subgenre.

playlist_genre (Response variable): As the response variable, playlist_genre has 6 classes, which are "EDM", "Latin", "pop", "r&b", "rap", and "rock". From the Fig4. we can see that they are also imbalanced, but most of the counts are within the range of 4000-6000, the highest courts are around 6000 referring to the "edm" genre.

Chi^2-Test: This Test aims to check the correlation between two categorical variables. Now that we have 3 pairs, which are playlist_genre (Response variable) versus playlist_id, playlist_name, and playlist_subgenre respectively. From the test result, we observed that these categorical variables have a very strong correlation with the response variable because the chisq_test p-value is all close to 0. To build a model we need to care about the highly correlated variables, because these may not be a real predictor, but they may contain sufficient information about the response variable. Such as the subgenre contains all the information of genres, we can absolutely predict a true label of genre just using subgenre. So as the playlist_id and playlist_name. So we should not choose these three variables as the predictors. Finaly we removed the 3 categorical predictors from our variables.

2)Numerical Variable

Due to the length of the article, I cannot introduce all the variables one by one. I will briefly introduce the overall situation.

I build histograms for all the numerical variables to check their distributions. Their distribution varies a lot. Except for the "mode" which shows a binary distribution and "key" shows a discrete distribution, all the other variables show a continuous distribution including right-skewed("speechiness", "acousticness","instrumentalness", "liveness"), left-skewed ("enegry","loudness","track_album_release_year") and nearly symmetric ("track_popularity", "danceability", "valence", "tempo", "duration_ms").

As for the boxplot, most of the numerical variables show different patterns in different genres. This means these variables have some extent of influence on the response variable.

Lastly, for the statistical proof that variables have an influence on the response variable, we observed that all these variables in the ANOVA tests show great statistical significance to the response variable. Because the p-value is all less than 0.05 for the multiclass comparison(though in some variables, some classes have a similar distribution pattern and the p-value is relatively big). We can not absolutely trust this statistical method to believe these variables should be the predictors, but at least we can know that they show influences on the response variables, so we can not reject these variables. Finally, all 14 variables were included in our

final predictors. And the Details can be referred to in the Appendix "EDA-Numerical variable" part fig 5. -fig 32. and the ANOVA summary for each variable.

2.2 The 3 questions that Spotify founders care about

Does the popularity of songs differ between genres?

From Fig6 we can see that, the popularity distribution in the six genres shows different patterns, they all have different mean values and box boundaries. More details about their difference can be referred to the ANOVA summary for the "track_popularity" variable. We can not accept the assumption that the popularity of songs is the same among the six genres (p-value $<$ 2e-16). But from the detailed report of the ANONA, we observed that "pop-latin" have a similar distribution pattern in popularity, so as the"rock-r&b", because we have more than 90% possibility to trust they are similar. Instead, the rest of the pairs show weak similarity in the ANONA report. So we can believe tha the popularity of songs differ between genres.

Is there a difference in speechiness for each genre?

The same check way as the last question. From Fig18 we can see that, the popularity distribution in the six genres shows different patterns, they all have different mean values and box boundaries. More details about their difference can be referred to the ANOVA summary for the "speechiness" variable. We can not accept the assumption that the popularity of songs is the same among the six genres (p-value $<$ 2e-16). But from the detailed report of the ANONA, we observed that all the pairs show weak similarity in the ANONA report(pairs p-value $\sim$ 0). So we can believe that there is a difference in speechiness for each genre.

How does track popularity change over time?

To answer this question, I made 3 plots (fig33-35.) and build 1 linear model. From Fig33 we can see that, if we divide the data into the five-year group, we can see that the mean track_popularity has a downward trend over time until 2010, but after this point, the mean track_popularity slightly goes up. But the distribution has not an apparent change over time.

If we just look at the mean value of each year in Fig34., we can have the same judgment as Fig33, the same rough downward trend until approximately 2010 from around 65 to 30, then rise after that to around 50 in 2020. The smooth trend line of Fig35. shows us the evidence that in 2020 it reached the same popularity level as 1960s after the same trend depicted in Fig33-34.

Finally, we built a linear model, which proves that there is a linear relationship between year and popularity and the coefficient is positive of 0.16009 unit/year, the positive value is due to a large amount of value after 2010. So we can have a basic judgment that, from 1960 to 2010, the popularity have a downward trend, but after that, it rise fast and almost recovered to the same mean level as 1960 in 2020.

2.3 PCA result

In order to reduce the dimension of 14 variables space, I tried to apply PCA. From Fig36. the Proportion of Total Variance Explained by PC, which stands for the accumulated interpretable percentage of each axis occupied of the newly transformed space. The curve in Fig36. shows there is no obvious influence after applying PCA, from the result we can see that the 95% accumulated interpretable percentage occurs at PC12, which is near PC14. Which means PCA is not suitable for this dataset. Finally, I didn't transform our data using PCA.

2.4 Modeling results

Before Molding I preprocessed the training set using "step_zv","step_normal", and "step_corr" which can remove zreo-variance column, normalization our data and reduce multi-linear-correlation respectively. This process was applied in test set later. We used bootstrap to tune 3 models, LDA, KNN, and Random Forest. Here, I didn't tune KNN and Random Forest Except for LDA(no need to tune). I will introduce the tuning process and result of the 3 model in below.

LDA

1)Modeling

LDA (Linear Discriminant Analysis) is a classic linear classification algorithm that classifies data by projecting data into a low-dimensional space and minimizing the ratio of intra-class variance and maximizing inter-class variance. Because there is no need to tune LDA, so I just applied the bootstrap data set to this model and using cross validation to evaluate the overall performance in the 5 validation set.

2)Evaluation

Overall, LDA reaches a mean accuracy of around 0.489, and mean Area Under the Curve(Receiver Operating Characteristic,ROC) is around 0.806 of the six classes in the 5 fold. From Fig38-43., we can see that the best performance of this model is in the "rock" Genre, which has the fullest curve with the highest Area under the curve. And "rock" class also has the highest sensitivity of around 0.645, and the lowest sensitivity is 0.317 at Class "r&b". Because this is multi-classification task, we more cared about the True positive rate, so sensitivity of each class matters more than specificity.

KNN

1)Modeling

KNN is a classification algorithm that determines one data point label by calculating the distance(Euclidean, Manhattan, and so on) between the point and their k nearest neighbor's (vote majority in k >= 2). KNN model was tuned in bootstrap data set using grid search parameter k in the range(1, 100) at 25 levels, the result shows the best k in accuracy mode is 89, whose accuracy is around 0.496. Then I use 89 to finalize our model knn_model_best.

2)Evaluation

Overall, KNN reaches a mean accuracy of around 0.509, which is slightly higher than LDA, and mean Area Under the Curve(Receiver Operating Characteristic,ROC) is around 0.820 of the six classes in the 5 fold. From Fig44-49., we can see that the best performance of this model is also in the "rock" Genre, which has the fullest curve with the highest Area under the curve. But "edm" class has the highest sensitivity of around 0.659, and the "rock" class is also not bad at around 0.629, and the lowest sensitivity is 0.364 also at Class "r&b" but relatively higher than LDA.

Random Forest

1)Modeling

Random Forest is an ensemble learning method, it uses multiple trees to minimize the total error to increase the overall accuracy. In Random forest, each tree will have limited depth and features(candidates to split), feature candidates are the mtry, and min_n controls the degree of bunching. Random Forest model was tuned in bootstrap data set using grid search parameter tree = 100(default), mtry and min_n at 5 levels. The result shows the best hyper parametr is tree = 100, mtry = 4, min_n = 11, and it can be refered to Fig37. The best mean accuracy is around 0.560 and the best mean AUC is around 0.85. Then I use the best hyperparameter to finalize our model rf_model_best.

2)Evaluation

Overall, Random forest reaches a mean accuracy of around 0.561, which is the highest among the 3, and the mean Area Under the Curve(Receiver Operating Characteristic, ROC) is around 0.850 of the six classes in the 5 fold, also the top of the 3. From Fig50-55., we can see that the best performance of this model is also in the "rock" Genre, which has the fullest curve(close to the right angle) with the highest Area under the curve. The highest sensitivity is 0.797 at the "rock" class, and the lowest class is slightly different than before, which is the "pop" class of 0.379, But this lowest value is still the largest of all the lowest values.

Best Model Selection

In Summary, among the 3 models, Random Forest performs the best no matter in mean accuracy, mean AUC, or in the ROC, the sensitivity of the six classes. So we chose random forest as the final best model to predict our test data. As for each class, "rock" class is almost easiest to classify among the 6 classes, in Random Forest Model, the Sensitivity shows a almost 80% high score, and the ROC curve is almost round right angle. And the Random Forest performs worst in "pop", compared with LDA and KNN worst in "r&b", which are all lower than 40%.

3.Discussion

In this section, we will introduce the Pros and Cons of the three models, and give the final prediction of test data in the best model we chose.

3.1 Pros and Cons

LDA

1) Pros LDA has good interpret ability and is fast to compute

2) Cons Low accuracy in our data set, and only have linear decision boundary, so it is not suitable for a complex data set.

KNN

1) Pros Easy to understand in mathematics, and almost no computation time in training.

2) Cons Low accuracy in our data set, and cost a lot computation in a lot of testing, especially in grid search to select the best k.

Random Forest

1) Pros Very easy to understand intuitively, relatively high accuracy among the 3 models, especially in some class can reach around 80% sensitivity. Cna show the imporatance of rach features.

2) Cons Cost a lot of computation when training especially when needing to search for a good hyperparameter.

3.2 Final Prediction Performance

In the final prediction stage, the whole training set was used to fit the model. After training this model, we can refer to Fig 56. The importance rank of features outcome by this model of this data set that the most important four features are the highest "track_album_release_year", followed by "danceability", then "speechiness", and last the "tempo". These are all the features our founders interested in. Thus the four features perform relatively high importance in our prediction.

After that, the test data was piped into this final model to make a prediction. The final accuracy in the test set is around 55.2%, which means 55.2% of data points can be correctly classified into the true class. Although this is not very high, it is still far beyond the probability of random guessing of 16.7%. So the model has some predictive effect.

As for some specific classes, like "rock", the True positive rate is around 80%, so if we predicted a song that belongs to "rock", we'll be more confident to trust it belongs to that class. But for the "latin", "pop", if the model gives these predictions, we may not have enough confidence to trust it belongs to that class, we may even exclude that it falls into this class. Finally, the Multi-class area under the curve is 0.843, which means this model has a not good performance in six classes classification problem, but it doesn't stand for it is effective for all classes, it's an overall performance score.

Besides, we noticed that the accuracy and sensitivity of the six classes have little difference from the cross-validation stage, which means our model has a good generalization ability, which also means our model has good stability when meeting a new data set.

3.3 Improvement Recomendation

The result of this model can give some useful information to our funders, we have already know the most importance features in our dataset, and we have already know that different model performs good and bad in different classes. We can have 3 improvement to your project in the future work.

1) Focus more about the features are more important, like we can give a higher weight to these features when doing KNN, which means it can take up more proportion when calculating geometric distance.
2) Using Bagging Method to combine several base learners(KNN, LDA, Random Forest) to co-make decisions, because the three model performs good and bad in different classes, they may not make mistake at the same time. Thus, it can help improve our total performance.
3) Exploring other powerful machine learning or deep learning algorithms to improve our model.

4. Conclusion

In conclusion, this paper shows a project that aims to predict the genre of a Spotify song using there different machine algorithms to build models. EDA result shows all categorical and numerical predictors have correlations with the response variable; the popularity of songs differ between genres, and there is a difference in speeches for each genre, and track popularity changes over time with a downward trend first until 2010 then a rise to almost the same level as in 1960. This paper shows that Random Forest performs the best among the three models, with the highest accuracy of 0.561 and the highest ACU of 0.850 under the cross validation mean result. And the three models perform differently in different classes, "rock" class is almost the easiest to classify among the 6 classes of all classifiers, and in Random Forest Model, the Sensitivity shows an almost 80% high score. Instead, the Random Forest performs worst in "pop", compared with LDA and KNN worst in "r&b", which are all lower than 40%. And the best model Random Forest performs also stable in the test set. The VIP plot shows the most important four features are also what our funder is interested in. Based on our observations, this paper also gives the Improvement Recommendation directions to improve our prediction performance through 3 ways 1) Raise the importance(weight) of what our funders are interested in;2) Use the bagging method to lower our error; 3) Exploring other machine learning or deep learning algorithm to improve our model. This paper shows machine learning algorithm is powerful to solve real-world problems.