Qingyu Zhang / Andy

N-number: N19903322

Pascal Wallisch

Principles of Data Science (DS-UA-112)

11 May 2024

Capstone Project: Spotify Data Analysis

Introduction: Data cleaning, data transformation and dimension reduction

a. Data Cleaning: I use Pandas to store and slice the data and use NumPy array to train the model and do analysis. The entire dataset is store in a dataframe named "dataWhole". For each question, a new dataset named "dataQ#" (# is the number of the question) is created from "dataWhole" and only extracts necessary columns. Specific data cleaning steps will be done accordingly based on the following ideas to minimize the data and information lost: As I checked at first in the code, the whole dataset has no nan inside. It seems like we don't have missing data here. But when I checked the distribution of the data in **"popularity"** column (which is the outcome variable in many questions), I found a lot of strange **'0's** inside. **I will consider these outlier '0's as missing data and do row-wise removal to the dataset when analyzing the questions that involve column "popularity".** That is because when I searched the songs with '0' popularity on Spotify app and compare their number of plays with that of other songs with '40' or '50' popularity, I found quite a lot songs that are marked as '0' popularity actually have a lot of plays, even compared with '50' popularity songs. In addition, I've found that for many songs where "popularity" is labeled as 0, the "popularity" of other songs that belong to the same album or have the same author are also labeled as 0. This likely indicates that the dataset we are analyzing has **no record** of plays for these artists or albums, which is what led to the 0 popularity. (Note that I will only do row-wise removal based on whether its popularity is ‘0’ for the questions that involve analysis on “popularity”. For other questions I will keep all 52k songs information and do analysis because I don’t want to lose information.)

(Also note that I created a Data Cleaning Mode indicator named “**dataCleaningMode**” on top of the codes. When it is set to **1**, the code will **not treat** ‘0’s in “popularity” as missing data and won’t delete any row; when it is set to **2**, the code will **treat** ‘0’s in “popularity” as missing data and will do row-wise removal)

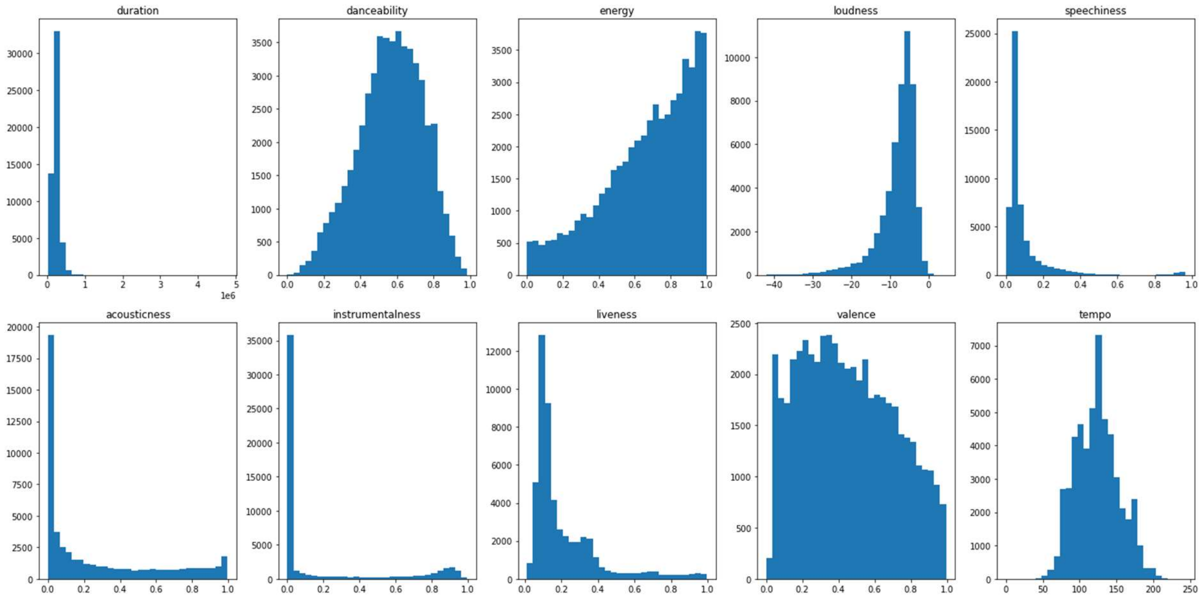
b. Dimension Reduction: In Question 8, I did Principal Component Analysis (**PCA**) on the 10 predictors / features required in Question 1. The data was **z-scored**, and the PCA was then fit with the transformed data. Eigenvalues were then graphed and analyzed, and I chose to use the **Kaiser criterion** and keep the principal components with eigenvalues greater than **1**.

c. Data Transformation: Data transformation was z-scored for use in PCA, and then rotated to graph the old 10 features in the new coordinate field. The rotated data was stored in a variable named “**rotatedPredictors**” and used in the following Question 9 and 10.

Questions 1 – 10:

1. “Consider the 10 song features duration, danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence and tempo. Is any of these features reasonably distributed normally? If so, which one?”

   I stored the names of the 10 features and enumerated to plot the distribution of each. Here is the result:

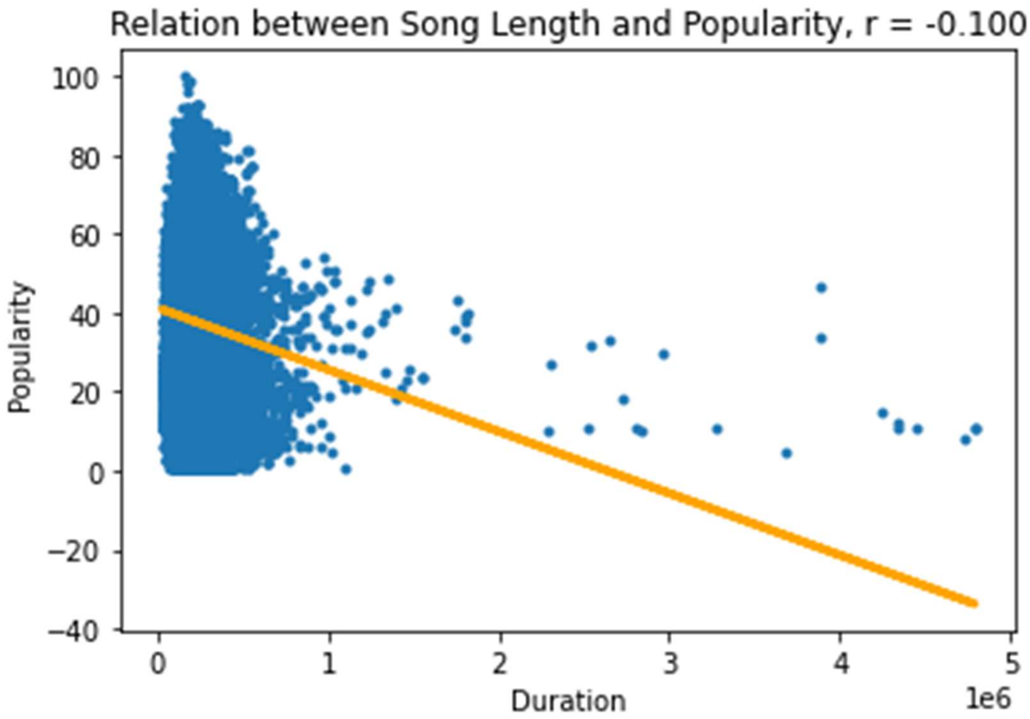


   As we can see from the 10 plots above, **“danceability”** and **“tempo”** are approximately **normally distributed.**

   (In addition, “loudness” seems to be a left-skewed distribution; “speechiness” and “liveness” seems to be right-skewed distribution.)

2. “Is there a relationship between song length and popularity of a song? If so, is the relationship positive or negative?”

   Song length is measured by **“duration”** in the dataset. Note that this question involves **“popularity”** column, so I did row-wise removal according to ‘0’s in “popularity” first. I extracted “duration” and “popularity” columns to do plotting and compute correlation coefficient. I got a correlation coefficient of **-0.1**, and graphed a scatter plot of the data as shown below.



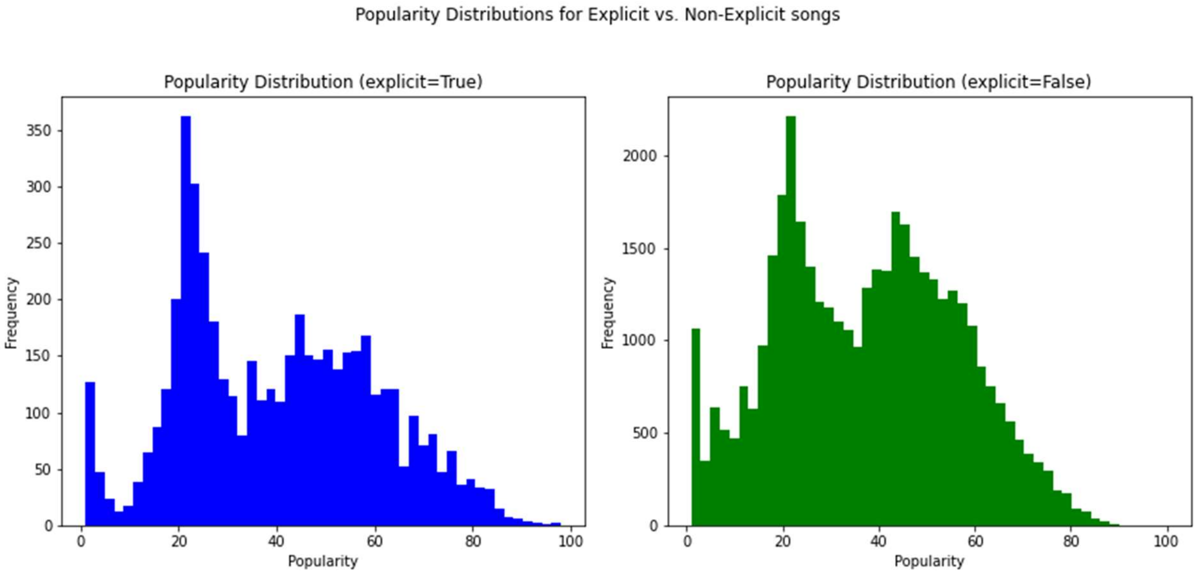
As we can see, there is a **weak** linear relationship between duration and popularity. And this relationship is **negative**.

3. "Are explicitly rated songs more popular than songs that are not explicit?"

This question involves analysis on **"popularity"**, therefore I do row-wise removal according to '0's in "popularity" first.

The **"explicit"** column has two kinds of values: **True and False.** Therefore, I divided all the remaining non-zero values in "popularity" column into two groups based on these Boolean values and stored in two variables. We want to know whether there is **systematic difference in popularity** between these two groups (explicit songs and non-explicit songs).

My **null hypothesis** is that whether a song is explicit or not does not affect the popularity of the song. To **select the right test** to test my null hypothesis I first check if the popularity is normally distributed:



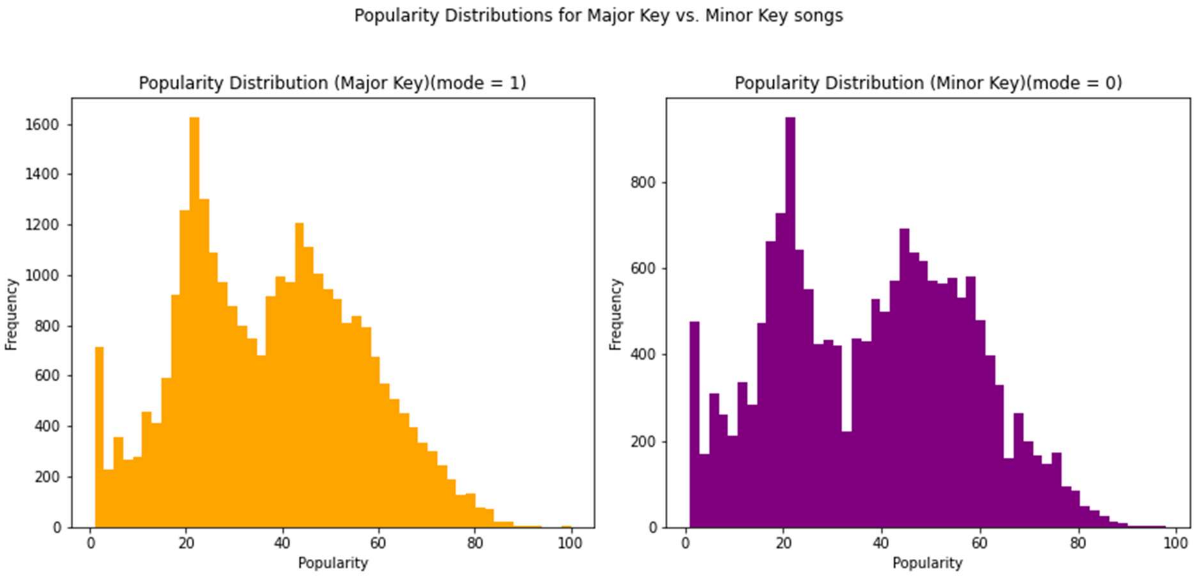
I would say they are **not normally distributed**. Besides, I think **popularity data doesn't reduce itself to means, and the data isn't categorical**. Therefore, I will use **Mann Whitney U test** between the popularity of explicit songs and the popularity of non-explicit songs. The median popularity of explicit song group is **39**, while the median popularity of non-explicit group is **38**. The U-test gives us a **p-value** of **5.925200292635755e-18**, which is very close to 0 and definitely smaller than threshold **alpha = 0.05**, which implies **statistical significance**.

Therefore, we **drop the null hypothesis** that whether a song is explicit or not does not affect the popularity of the song. The median difference between the two groups is **not by chance**. Since the median popularity of explicit group (39) is higher than that of non-explicit group (38), we can conclude that **explicitly rated songs are more popular than songs that are not explicit.**

4. “Are songs in major key more popular than songs in minor key?”

This question involves analysis on **“popularity”**, therefore I do row-wise removal according to ‘0’s in “popularity” first.

The **“mode”** column has two kinds of values: **1** (Major key) and **0** (Minor key). Therefore, similar to question 3, I divide the remaining non-zero values in “popularity” column into two groups based on whether the song is in major key or minor key. We want to know whether there is **systematic difference in popularity** between these two groups (Major key songs and Minor key songs). My **null hypothesis** is that whether a song is in major key or in minor key does not affect the popularity of the song. To **select the right test** to test my null hypothesis I first check if the popularity is normally distributed:



I would say they are **not normally distributed**. Besides, I think **popularity data doesn’t reduce itself to means, and the data isn’t categorical**. Therefore, I will use **Mann Whitney U test.** The median popularity of Major Key song group is **38**, while the median popularity of Minor Key song group is **39**. However, the U-test gives us a **p-value** of

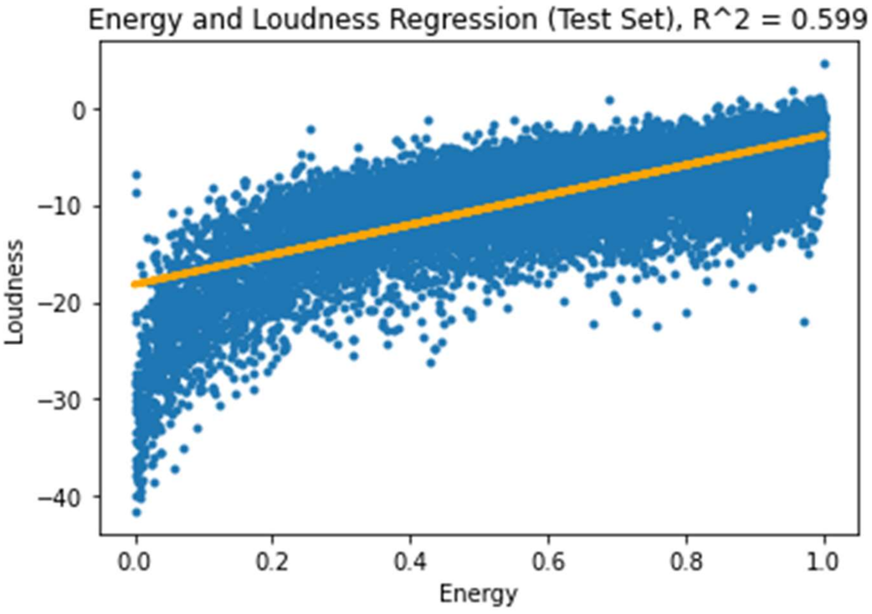
0.12671603536482307, which is **not smaller** than threshold **alpha = 0.05**, which implies **not statistically significant**.

Therefore, we **failed to drop the null hypothesis**. The observed difference in median popularity of the two sample groups is probably due to chance. We can conclude that **whether a song is in major key or in minor key does not affect the popularity of the song**.

5. "Energy is believed to largely reflect the "loudness" of a song. Can you substantiate (or refute) that this is the case?"

   To verify this statement, we do a **linear regression** model to **predict "loudness"** of songs from **"energy"**. Note that we will use **cross-validation** (train-test split) to **avoid overfit**.

   Since I use np.random.seed() to set my N-number as seed to **RNG**, the result of the cross-validation will be constant. I chose to use half of the dataset (train set) to train the linear regression model and use the other half of the dataset (test set) to evaluate the model (compute R-Square value) and plot the data points and the regression line (Figure below).



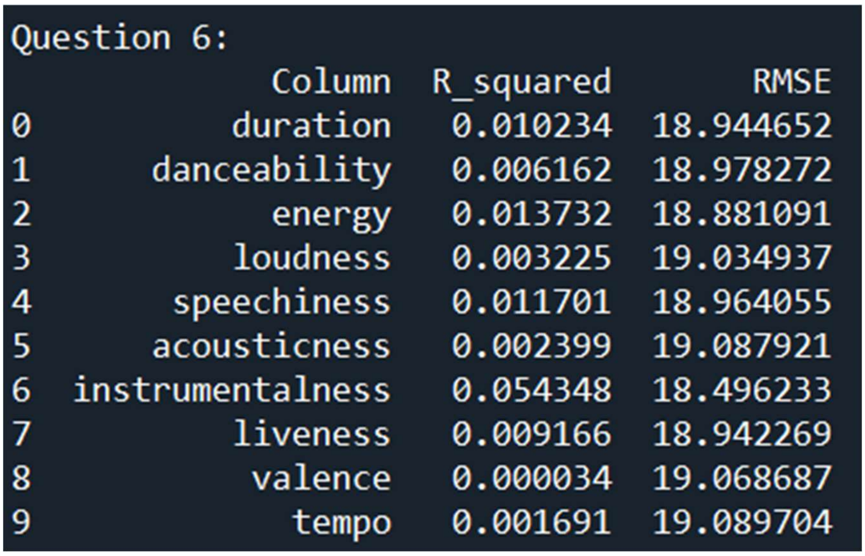
The **Pearson correlation** between "energy" and "loudness" is **0.775**, which indicates **strong positive linear relationship**. And the **R-Square** computed on the test set is **0.5986581999495897**. This means that the predictor "energy" and our model can **account for about 60% of the variance** in the outcome variable "loudness". This is a **pretty good R-Square** value and indicates that **energy can truly largely reflect the "loudness" of a song**.

6. "Which of the 10 individual (single) song features from question 1 predicts popularity best? How good is this "best" model?"

   This question involves analysis on **"popularity"**, therefore I do row-wise removal according to '0's in "popularity" first.

   To find which of the 10 individual song features from Question 1 is the best predicator for "popularity", we will do **10 linear regression** (use each of the 10 features one by one to predict "popularity") and record the corresponding **R-Square** value and **RMSE** of each in order to **evaluate each model**.

   Note that we will use **cross-validation** (train-test split) to **avoid overfit,** just same as before (train each of the 10 models on train sets, and evaluate each of the models on test sets).



```
Question 6:
             Column  R_squared       RMSE
0          duration   0.010234  18.944652
1      danceability   0.006162  18.978272
2            energy   0.013732  18.881091
3          loudness   0.003225  19.034937
4       speechiness   0.011701  18.964055
5      acousticness   0.002399  19.087921
6  instrumentalness   0.054348  18.496233
7          liveness   0.009166  18.942269
8           valence   0.000034  19.068687
9             tempo   0.001691  19.089704
```

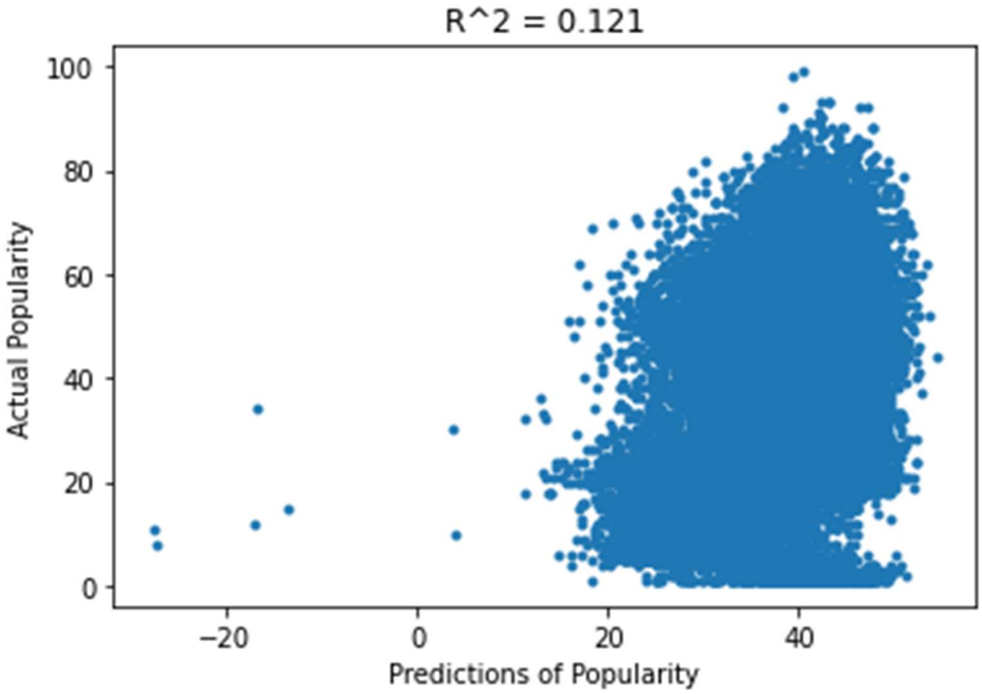
According to the table above, among the 10 features, the feature with the **highest R-Square** value is **“instrumentalness”** with an **R-Square** of **0.0543**, and a **RMSE** of **18.4962**. Therefore, we can say that “instrumentalness” is the “best” predicator for “popularity” among these 10 features.

Although “instrumentalness” is the “best” predicator in terms of R-Square compared with other 9 features, its R-Square is only **0.0543**, which is quite low and means that “instrumentalness” and the linear regression model only account for about **5.4% of the variance** in the outcome variable “popularity”. Under this circumstance, this “best” model is actually bad. **“Instrumentalness” cannot predict “popularity” well, at least not well in the linear regression model.** These 10 features all cannot predict “popularity” well in the linear regression model. I think that other **non-linear models** might use these features to predict “popularity” better, such as Random Forest Regression model.

7. “Building a model that uses *all* of the song features from question 1, how well can you predict popularity now? How much (if at all) is this model improved compared to the best model in question 6). How do you account for this?”

This question involves analysis on **“popularity”**, therefore I do row-wise removal according to ‘0’s in “popularity” first.

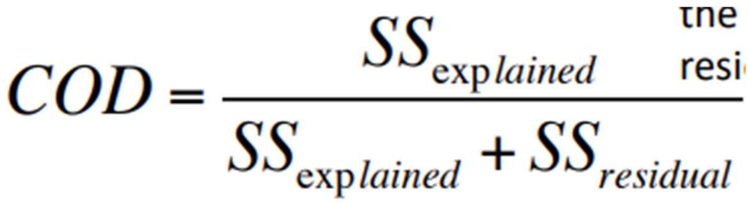
We will do a **multiple linear regression** model to use **all the 10 features** to predict **“popularity”**. Note that we will use **cross-validation** (train-test split) to **avoid overfit**, just same as before (train the model on the train set and evaluate the model on the test set). Here is plot of predicted popularity values (predictions of popularity from xTest) vs. actual popularity values (yTest):



R-Square value of this multiple linear regression model is **0.12124633789887307**. This means that the 10 features and the multiple linear regression model accounted for about **12.1% of the variance** in the outcome variable “popularity”. This suggests that this new model is **still not good** at predicting “popularity”.

This multiple linear regression model accounted for about **12.1% - 5.4% = 6.7% more variance** in the outcome variable “popularity” than the best model in Question 6.

Compared with the best model in Question 6, our new multiple linear regression model didn’t improve a lot. And I think this improvement is probably due to the fact that **adding more predictors to the linear regression model will always increase the R-Square value** (explain more data variance). Even if the newly added predictors do not contribute much to the predictive power of the model, they usually explain a small portion of the previously unexplained variation, leading to a decrease in $SS_{residual}$, therefore leading to an increase in R-Square (COD).



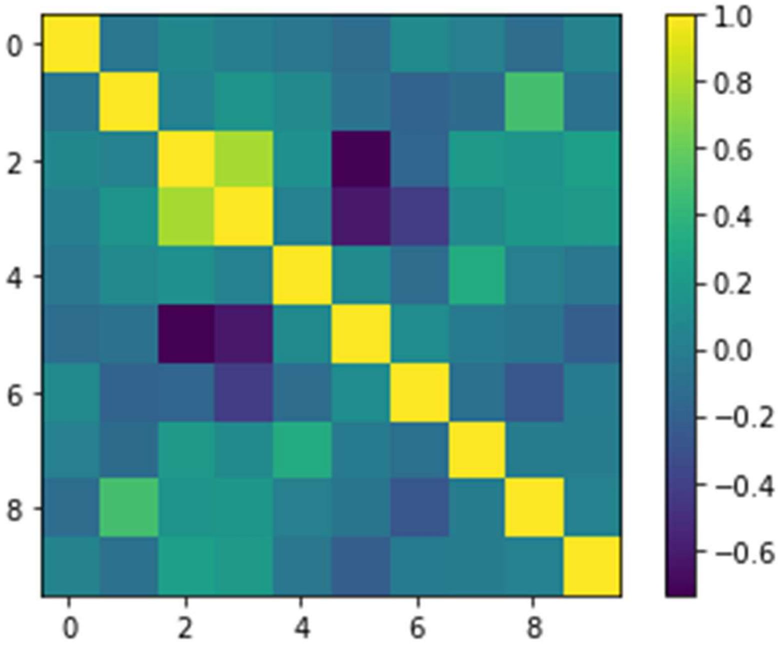
$$COD = \frac{SS_{\exp lained}}{SS_{\exp lained} + SS_{residual}}$$

Linear models probably are not good choices for situations in Question 6 and 7. If there exist relationships, they might be **non-linear**.

8. “When considering the 10 song features above, how many meaningful principal components can you extract? What proportion of the variance do these principal components account for?”

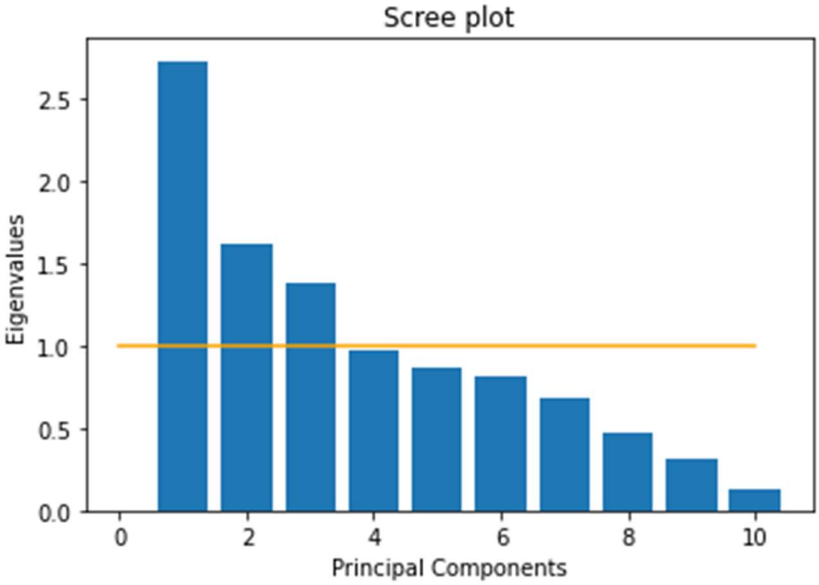
We will do a Principal Component Analysis for the 10 features in Question 1.

The first step is to check whether there is correlation among these 10 features. We will run and show the **correlation matrix** of these 10 features / predictors:



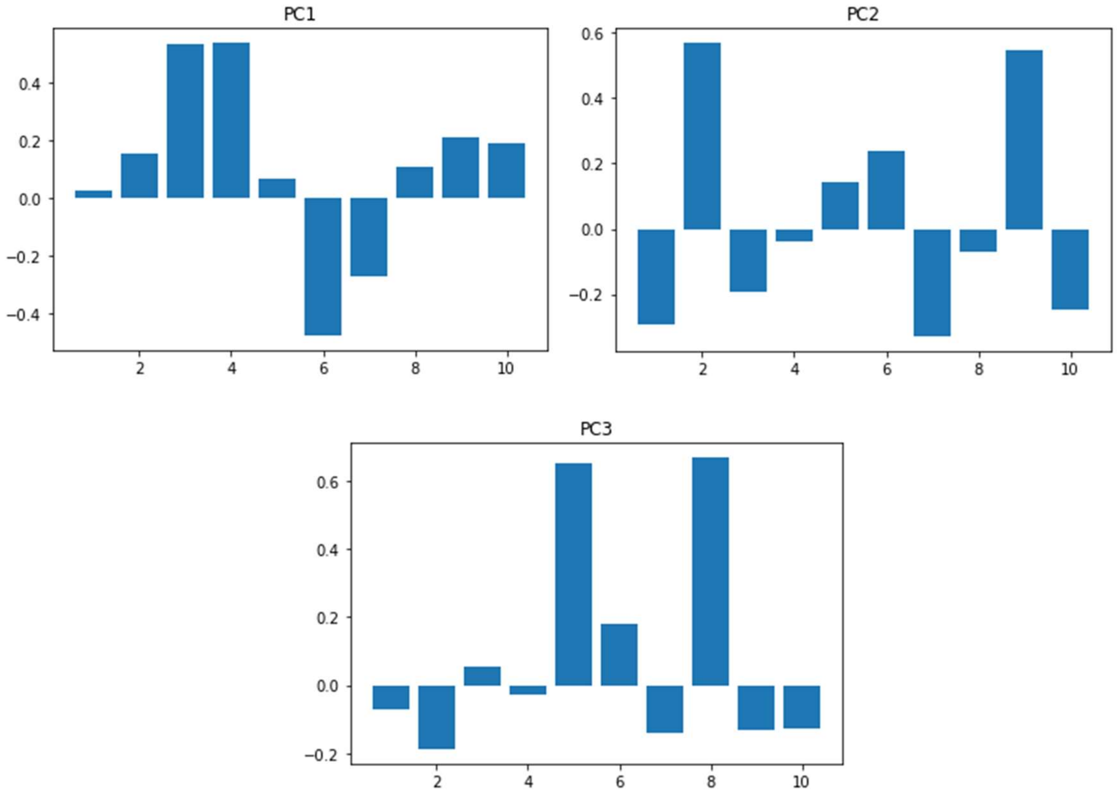
From here we see that there are multiple correlations within, which creates incentive to perform a dimension reduction like PCA.

We first **normalize the data (z-score)** before doing PCA. We calculate and plot the **eigenvalues** of our principal components:

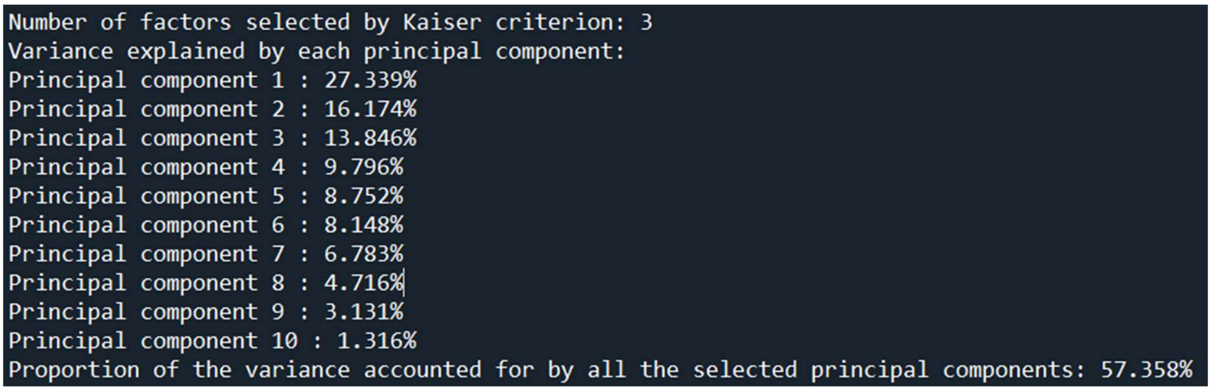


By **Kaiser criterion**, we extracted **3 principal components** with eigenvalue greater than **1**.

There are the **loadings**:



The **transformed / rotated data** is returned by the function and stored in a variable named **"rotatedPredictors"** for later use in Question 9 and 10.



```
Number of factors selected by Kaiser criterion: 3
Variance explained by each principal component:
Principal component 1 : 27.339%
Principal component 2 : 16.174%
Principal component 3 : 13.846%
Principal component 4 : 9.796%
Principal component 5 : 8.752%
Principal component 6 : 8.148%
Principal component 7 : 6.783%
Principal component 8 : 4.716%
Principal component 9 : 3.131%
Principal component 10 : 1.316%
Proportion of the variance accounted for by all the selected principal components: 57.358%
```

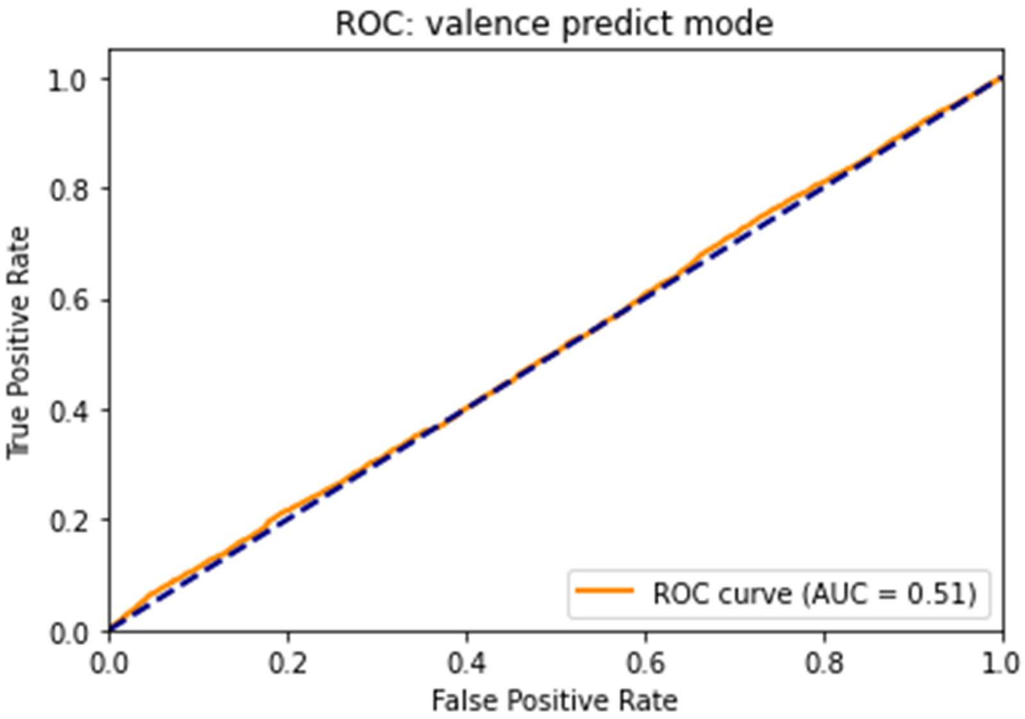
According to my calculation, these **3 selected principal components** together **account for** about **57.358%** of the **variance**.

9. "Can you predict whether a song is in major or minor key from valence? If so, how good is this prediction? If not, is there a better predictor?"

I used **logistic regression classification** to predict the **binary outcome** of **major or minor key** from **"valence"**. I use **ROC curve** and **AUC score** to evaluate how good are the predictions of my models.

Note that we will use **cross-validation** (train-test split) to **avoid overfit.** I fit the logistic regression model with train set and do evaluations (ROC curve and AUC) with test set.

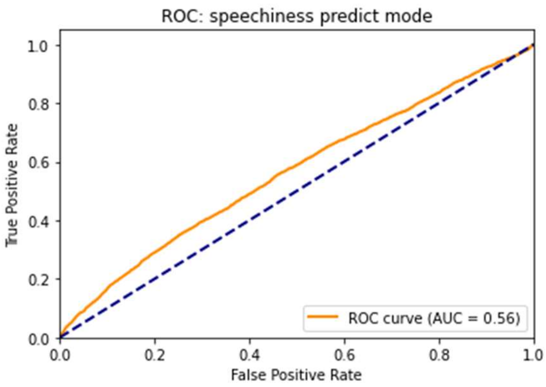
Here is the ROC curve and AUC score of the logistic model to predict **"mode"** (major or minor key) from **"valence"**:

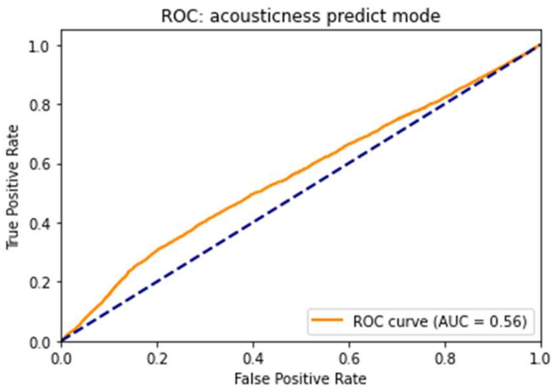


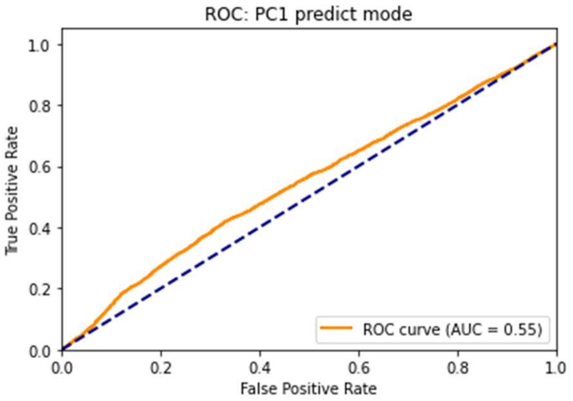
As we can see, the **ROC curve** is close to the **diagonal line** and the **AUC score** is only about **0.51**. This means that our "valence predict mode" logistic model is almost same as **"random guessing by chance"**. Therefore, this model is **pretty bad** and we can conclude that we **cannot** predict whether a song is in major or minor key from **valence**, at least in terms of logistic regression model.

Find better predictors:

Several other 9 features and some principal components are better predictors compared with “valence”. I will show **three** predictors with **highest AUC score**:







As we can see, **“speechiness”, “acousticness”, and principal component 1** are all better predictors for “**mode**” (major or minor key) **compared with “valence”**. But according to their **AUC score (0.56 and 0.55),** these three predictions are **still not very good**. The predictions of these three models are just a bit better than random guessing by chance.

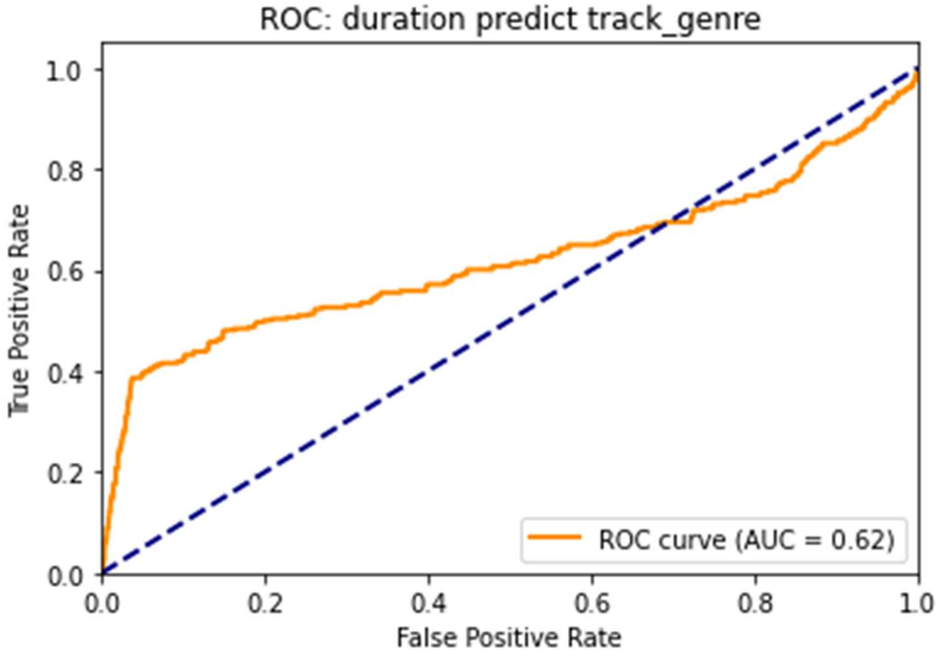
10. “Which is a better predictor of whether a song is classical music – duration or the principal components you extracted in question 8?”

The values in the **“track_genre”** column are different strings rather than binary numerical labels. Therefore, I made a copy of “dataWhole” (my variable that stored the whole dataset) and converted the values in its “track_genre” column into **binary numerical labels** (**“classical”** songs are labeled as **1** and all the **other genres** songs are labeled as **0**).

I used **logistic regression classification** to predict the **binary outcome** of **“classical” or not** from **“duration”**. I use **ROC curve** and **AUC score** to evaluate how good are the predictions of my models.

Note that we will use **cross-validation** (train-test split) to **avoid overfit.** I fit the logistic regression model with train set and do evaluations (ROC curve and AUC) with test set.

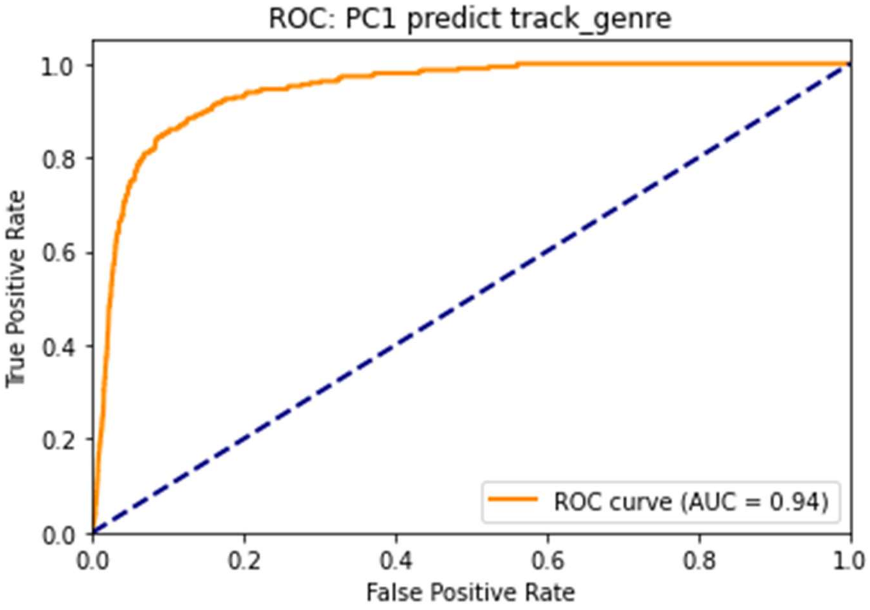
Here is the ROC curve and AUC score of the logistic model to predict whether a song is **classical music or not** from “**duration**”:



As we can see, the **AUC score** of this model is about **0.62**, so I think it’s an “ok” model.

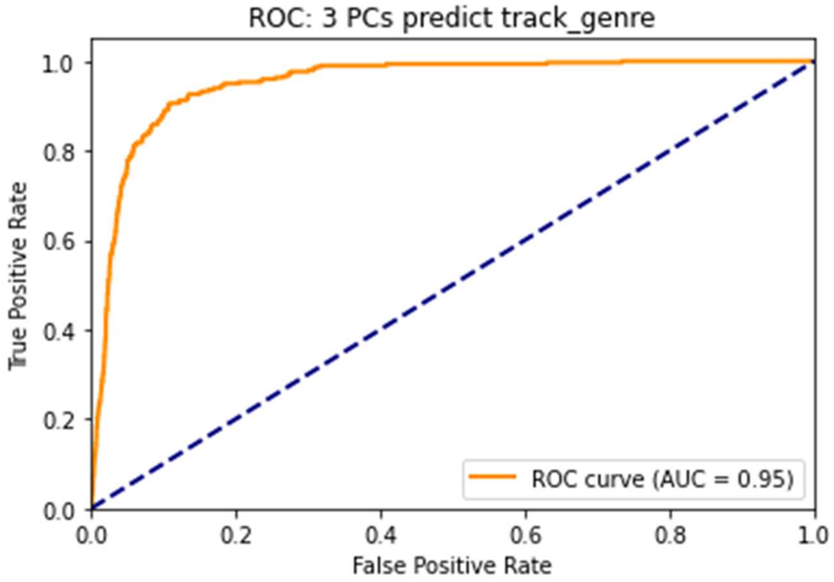
Note that at the top right part of the graph, the ROC curve has a part below the diagonal line, which means that this model predicts even worse than random guessing by chance at those choices of the threshold.

Let’s see the **ROC curve** and **AUC score** of the model to predict from **principal component 1**:



From the graph above, given the **ROC curve** shape and the **AUC score** of **0.94**, we can clearly see that this model is definitely **better** than the previous model (which use “duration” to predict whether a song is classical or not). If we fit and set the model correctly, the AUC score is usually **between 0.5 and 1**. Therefore, a score of **0.94** means that this model (which use PC1 to predict) is **pretty good**.

We probably also want to see the case when we use **all the 3 selected principal components** to predict whether a song is **classical or not**. Here is the result:



As we can see, this model also has a pretty good ROC curve and AUC score of **0.95**. But compared with our previous model which use the **principal component 1** only, the AUC score of this model only improved about **0.01**, which suggests that among these 3 PCs, **PC1** did most of the jobs. Adding PC2 and PC3 to the model seems to be “useless” and might have **higher risk of overfitting**. So, I would like to only use **principal component 1** to predict.

Conclusion for Question 10: Compared with “duration”, **principal component 1** that I extracted is **a better predictor** of whether a song is classical music.