Mathematics and Natural Sciences faculty

Machine Learning Methods

P160B124

Report no. 1

Report author:
Stud.
Marius Arlauskas
MGDMI-0

Supervisor:
Doc. Tomas Iešmantas

Kaunas, 2020

Task 1: linear regression

1. . Read the train bike.csv and test bike.csv datasets into you environment. How many features the data has? what are the types of those features? how many observations are there in the train and test datasets?

Asnwer: There are 7200 observations in train_bike dataset, and 14 variables. The Date column is datetime type; Seasons, Holiday and Functioning_Day are categorical, the rest of the variables are numerical, either integer, or real.

2. Select two features (except Date) and plot them against the Rented Bike Count (i.e. you should produce two different plots; don't forget to name the axes!). Discuss (based solely on the visual information) how useful this single feature is for the prediction of Rented Bike Count?

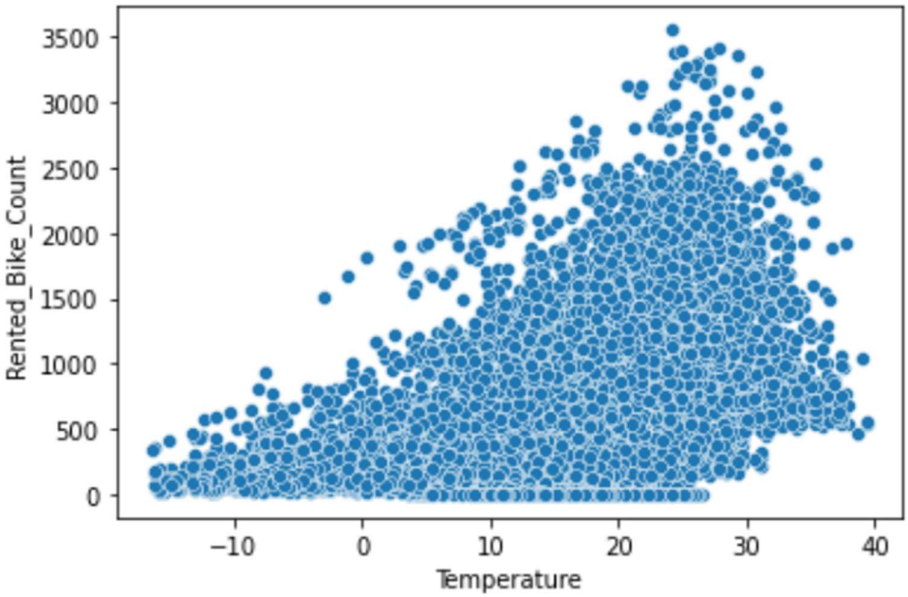

Figure 1 Dependence of Rented_Bike_Count and Temperature

The plot demonstrates positive correlation, as temperature increases, more bikes are rented, we could guess that the relationship is linaer, however because of the variance, it is hard to say for sure.

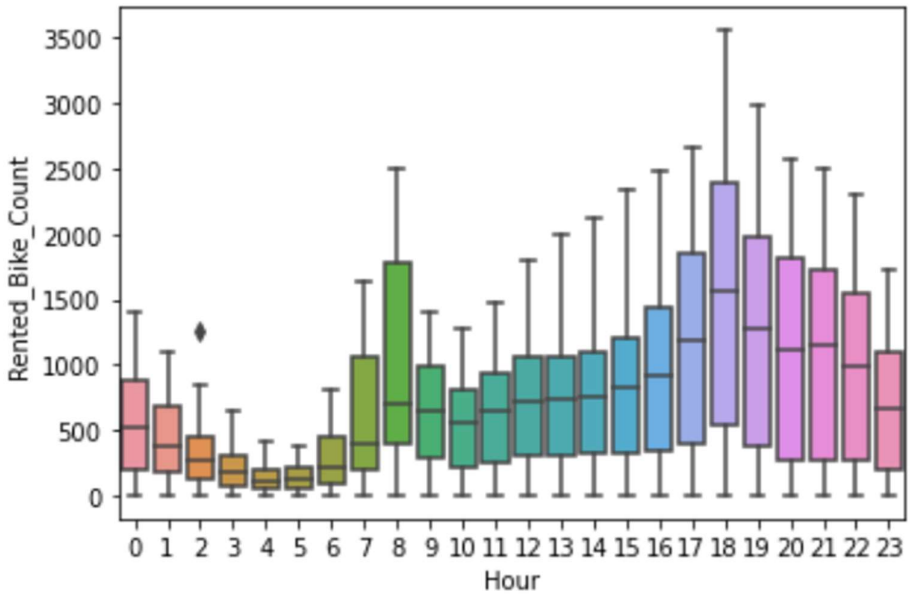

Figure 2 Dependence of Rented_Bike_Count and Hour

We can see in the figure two peaks: one at 8th hour, second at 18th hour, we could make a guess that at 8:00 people are leaving for work and are using bikes to commute, and at 18:00 they are either leaving work with bike as well as just renting bike for leisure after work. The relationship seems very reasonable, however, it is not linear.

3. Compute the matrix of correlations between variables using the function cor. Only between numerical values the correlation is possible.

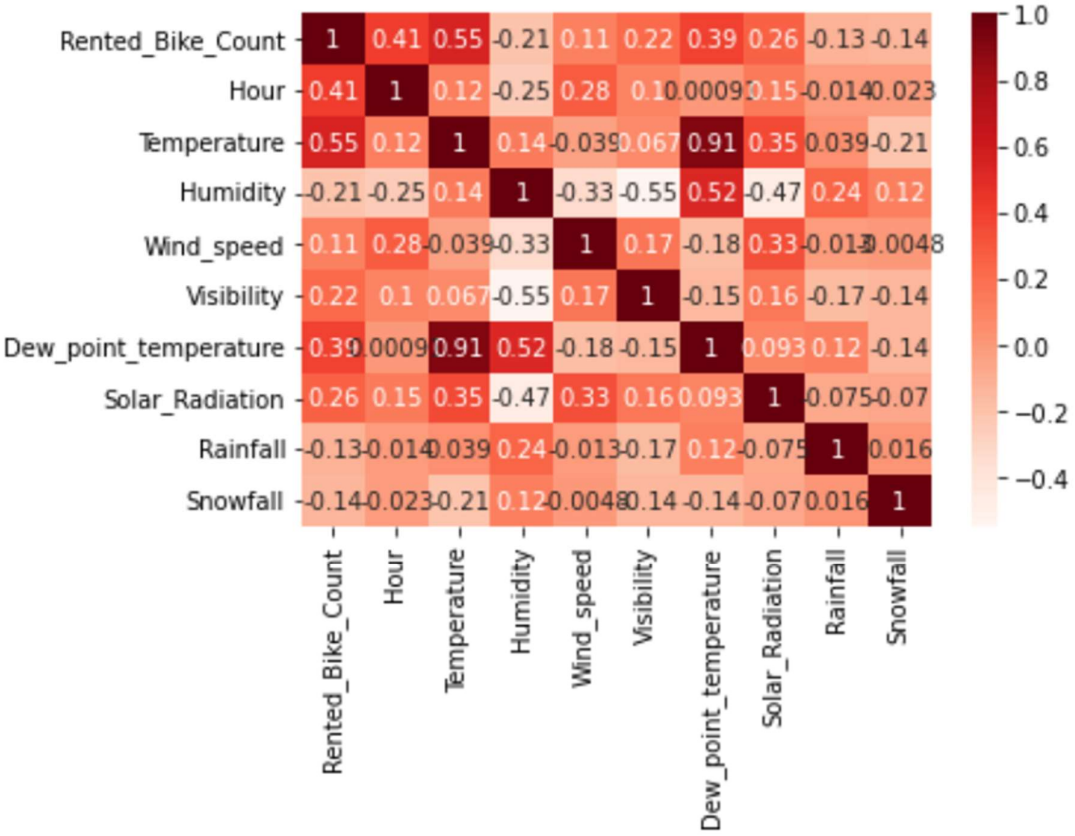

Figure 3 Visualised correlation matrix

All of the correlations with the target variable are fairly high, because of this we can say that there are no variables that are complete noise. However Temperature and dew point temperature are almost perfectly correlated. Let‘s investigate it further:

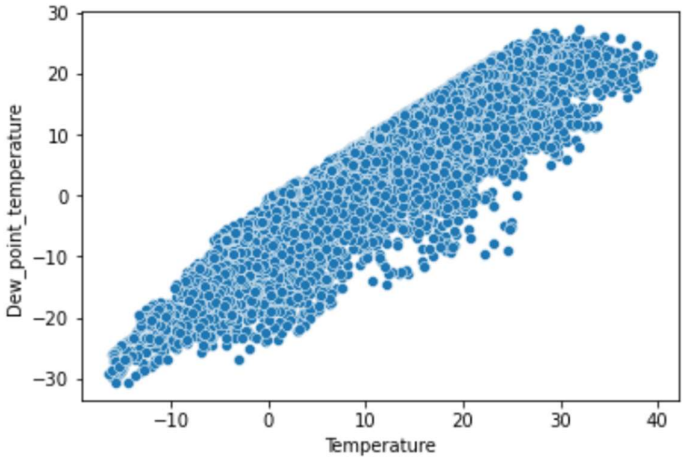

Figure 4 Temperature and dew_point_temperature dependence

The figure demonstrates almost perfect linear relationship. By dropping almost perfectly correlated fields we would only lose a little bit on information. However, linear regression requires no autocorrelation between variables, so I will drop the column completetly.

4. Predict the count of rented bikes for the testing dataset and plot the predictions. What problems do you see in your model based on this plot?

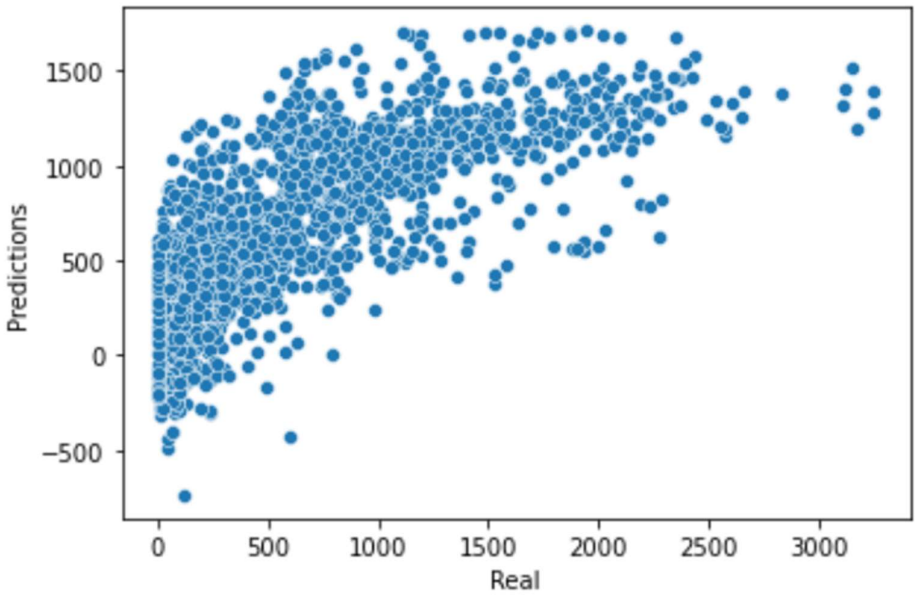

Figure 5 Linear regression predictions and test values plotted

R2=0.514

Rmse=428

There are some predictions that are negative, which cannot occur in reality.

5. Transform with log1p target variable then, train you linear regression model on the train dataset and predict counts of rented bikes for the test data set. Plot predicted and (transformed) true rented bike counts. Calculate the R2 measure of fit. Discuss the plot and calculated measure of fit:

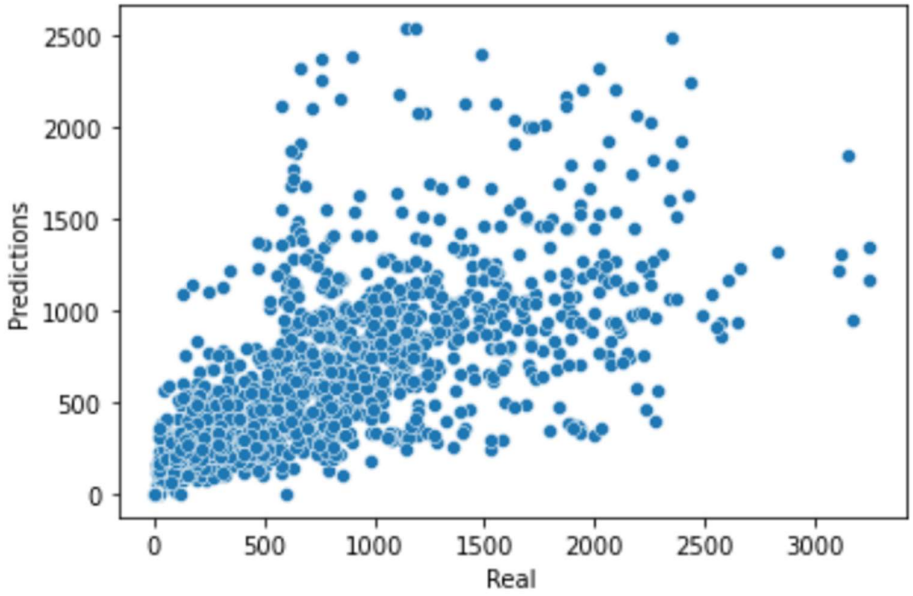

Figure 6 Linear regression predictions and test values plotted after transformation

R2 = 0.4436
Rmse = 458

The plot now seems better because there are no negative values. However it is still not that great of a predictions, and the R2 metric actually decreased.

Comment:

I did not measure R2 and rmse on transformed variables, I transformed train variable, then used expm1 function on the prediction to get the original format of the target variable, because it did not seem logical for me to use the transformed variables, since I just can multiply them by 0 and then the metrics would be perfect, but the predictions would be useless.

6. + 7.

Use interaction features and feature transformations then measure r2 metric.

	Names	r2	rmse
0	wind_temp	0.519330	425.841283
1	hour_vis	0.509899	429.998826
2	hum_temp	0.520679	425.243458
3	hour_temp	0.551898	411.161835
4	Base_model	0.512624	428.801374
5	MinMaxScaler	0.507263	431.153358
6	StandardScaler	0.511829	429.151326
7	Final_model	0.557514	408.577118

Figure 7 Table with names of the transforamations and prediction metrics of the model

Base_model is the model that was trained with no transformations, except categorical variables were encoded into binary variables. Only hour_temp interaction effect seemed to have any significance,

MinMaxScaler and StandarsScaler was applied to all features except target variable. Final model is just the model with hour_temp interaction feature + prediction transformation with the rule of if the prediction is negative, transform it to zero. MinMaxScaler only makes the coefficients more interpretable in terms of importance, and the StandardScaler is redundant, only because scikit-learn‘s linear regression‘s algorithm seems to have additional functionality for recentering features, otherwise the StandardScaler should have yielded at least some rmse or r2 gain.

Task 2: linear regression

1. Begin by reading the dataset music spotify.csv. Describe its structure, produce several examples of its entries.

Most of the features are numerical, and mostly are some sort of ratings that are trying to quantify concepts that are not really quantifiable in traditional sense. A few exceptions are duration_ms which is just the duration of the song, artist, song_title..

2. + 3.Create density plots of the variables:

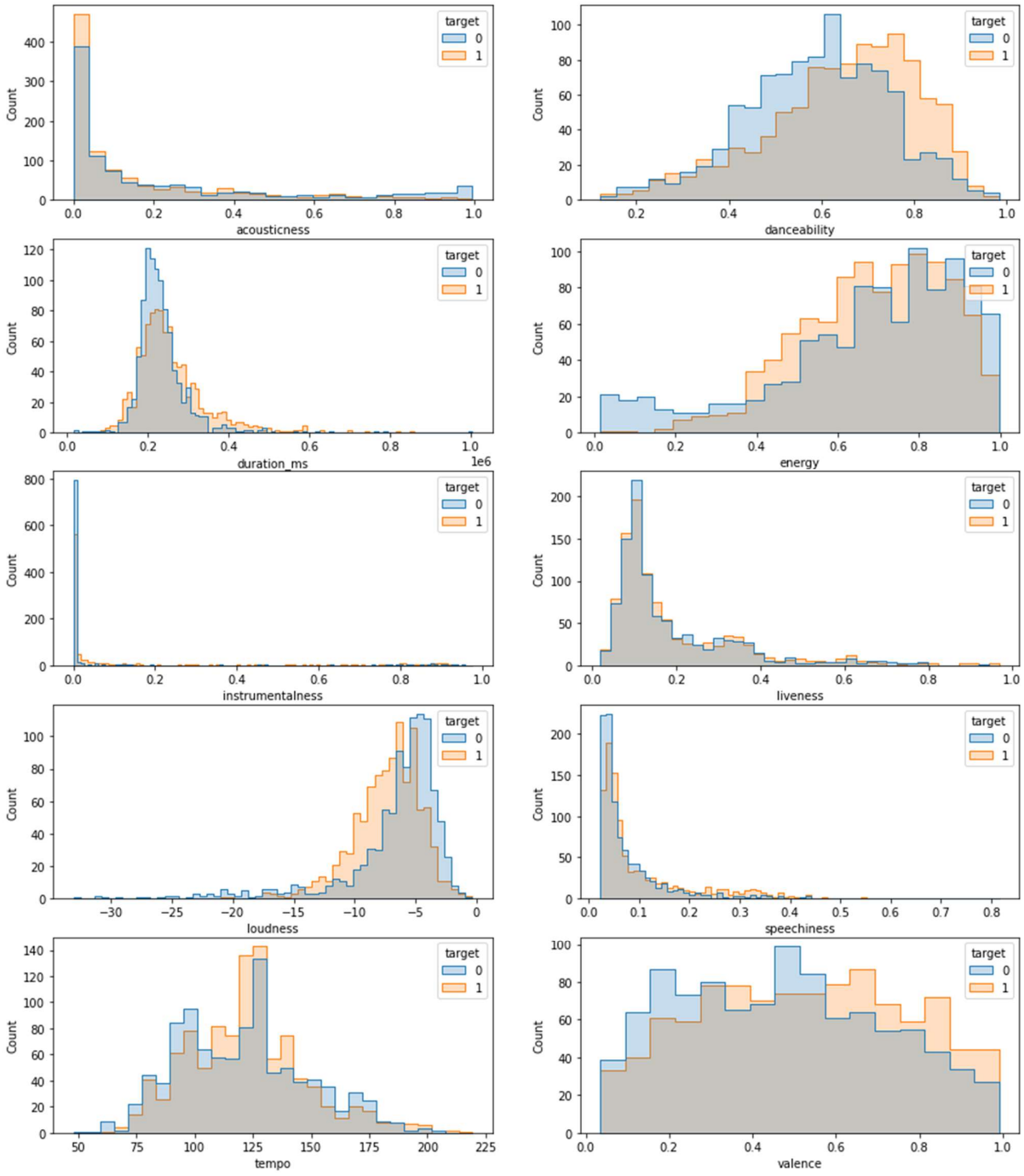

Figure 8 Distribution plots with target as hue

4. Create density plots of the variables liveness, loudness, speechiness, tempo, and valence, each versus the response variable target.

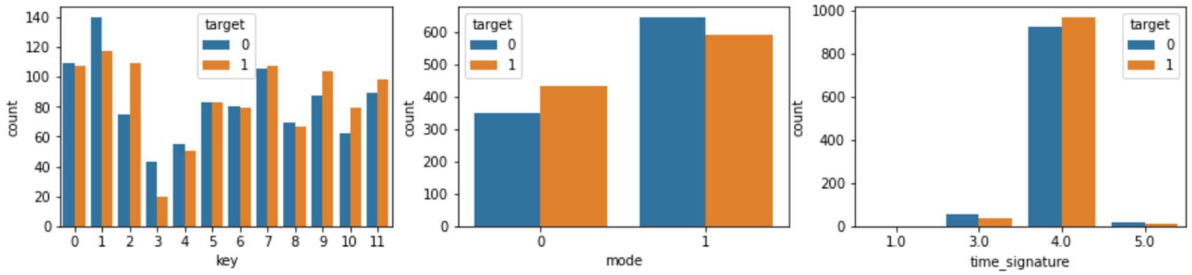

Figure 9 Barplots with respect to target

5. Examine the plots from parts (2) - (4). Based on these plots, which variables might be useful in differentiating which songs the user likes vs. those the user doesn't like? Briefly explain.

Danceability, loudness, and valence seems to be the candidates for the best predictors, because the distributions are not the same for different classes of target variable. As for the barplots, key seem to carry some information about the target, because of the difference in distributions when splitted by target

6. Split the data into training and testing subsets (splitting should be done based on target values). Fit logistic model to all variables except X, song title and artist. Using 0.5 probability threshold, report confusion table, accuracy for each class, as well as overall accuracy.

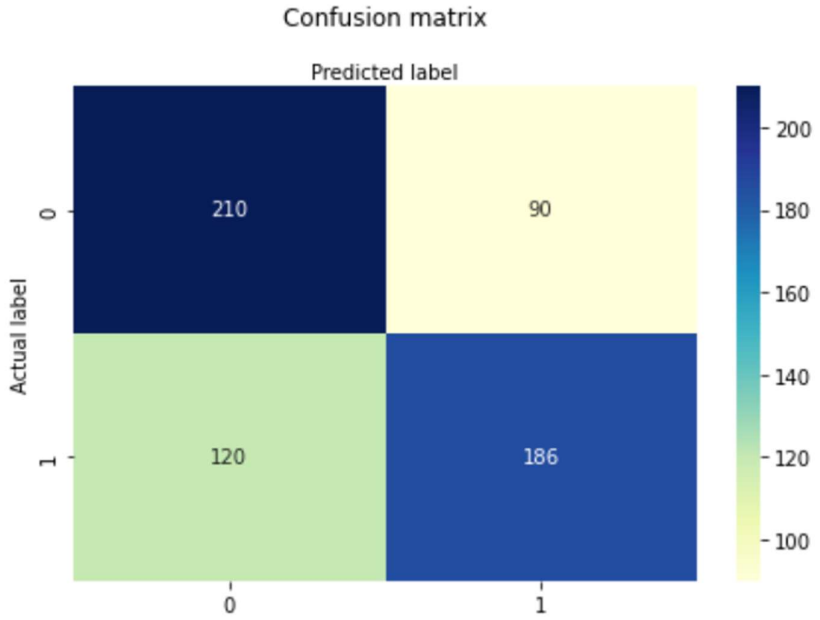

Figure 10 confusion matrix with threshold 0.5

Accuracy score = 0.653465

7. Select three probability threshold values (e.g. 0.3, 0.4, 0.5) and investigate accuracy results for each of these threshold values. Discuss, which is better: to make more mistakes by predicting if the song will be liked, or making more mistakes by predicting if the song will be disliked?

Which mistakes are more impactful depend on the goal, for example if we only recommend songs that will be liked, if we increase sensitivity to liked prediction mistakes we will decrease pool of songs that we can recommend and playlist may become repetitive, on the other hand if we make it not as sensitive, the user could dislike the platform since it recommends irrevelant songs, but in spotify‘s case, the song pool is huge and it negates the problem I mentioned earlier, the sensitivity to liked prediction mistakes is preffered. So we want to minimize False Positive outcomes.

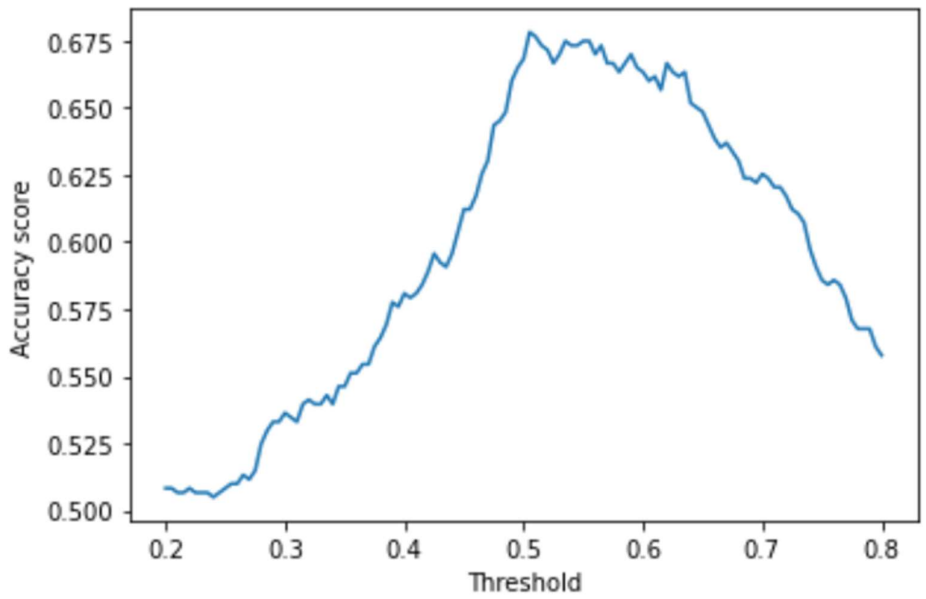

Figure 11 Accuracy score relationship with threshold

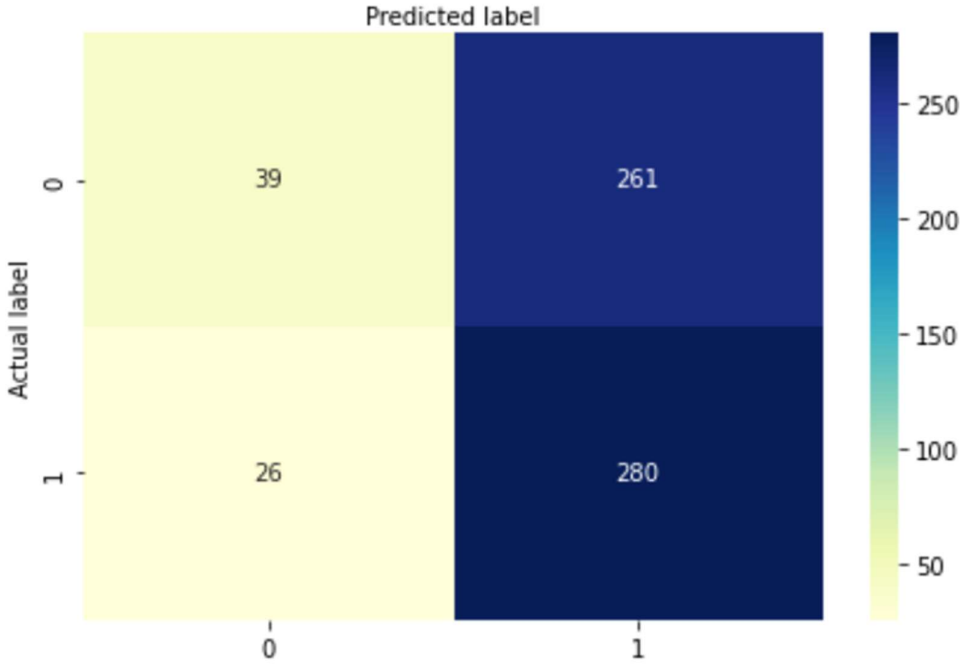

Figure 12 Confusion matrix for threshold=0.35

Accuracy score = 0.526

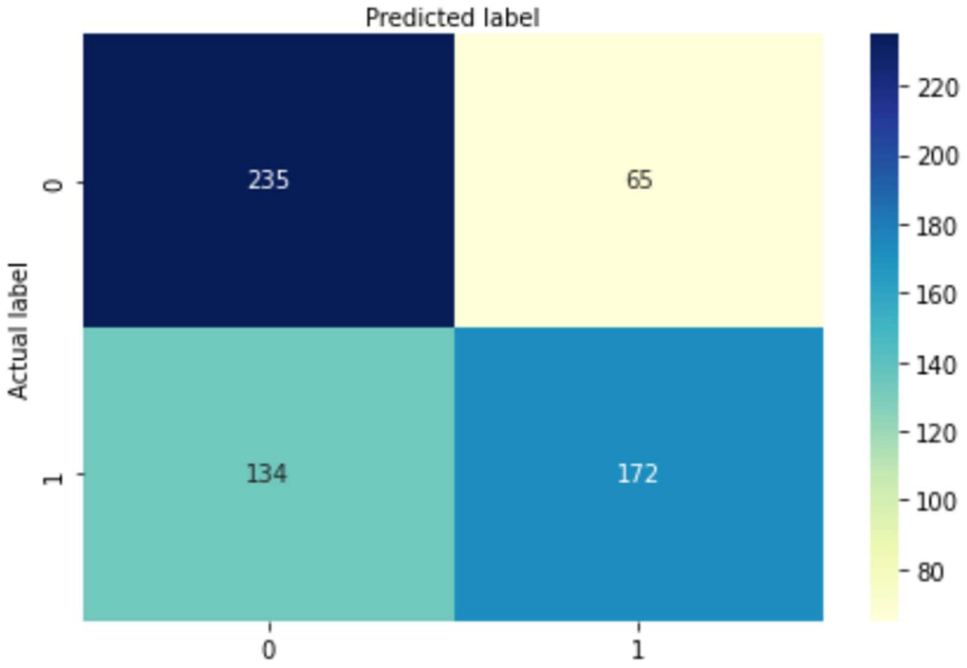

Figure 13 Confusion matrix for threshold=0.55

Accuracy score = 0.672

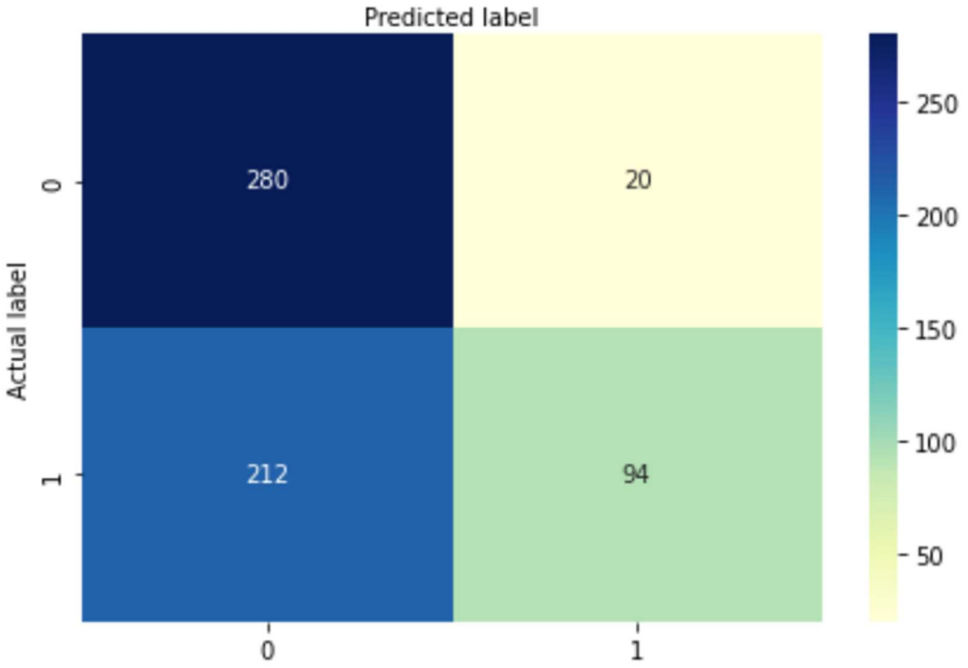

Figure 14 Confusion matrix for threshold=0.65

Accuracy score = 0.617

8. In (5) you discussed how well different variables might differentiate between two classes. Some variables seemed more usefull than others. Perhaps you could, based on discussion in (5) reduce the number of parameters and still have the same (or even higher) accuracy?

	Dropped_feature	Accuracy	Accuracy gain
0	danceability	0.691419	0.014851
1	time_signature	0.684818	0.008251
2	mode	0.681518	0.004950
3	base_model	0.676568	0.000000
4	duration_ms	0.669967	-0.006601
5	energy	0.668317	-0.008251
6	instrumentalness	0.666667	-0.009901
7	tempo	0.666667	-0.009901
8	key	0.658416	-0.018152
9	loudness	0.651815	-0.024752
10	acousticness	0.650165	-0.026403
11	valence	0.645215	-0.031353
12	liveness	0.638614	-0.037954
13	speechiness	0.605611	-0.070957

Figure 15 table with dropped features and accuracy gain, with threshold=0.5

Based on this table, I dropped danceability, time_signature, mode and the final accuracy score was equal to 0.67, which mean we only lost 0.007 of overall accuracy.