AIML421_A3

kakauchad_300212228

2022-10-05

Contents

1 AIML421 Assignment 3

This project attempts to develop a model to predict music genres of unseen data, based on training over 50,000 rows of data, with 19 features, including the class label (i.e. the song's genre).

1.1 Initial data analysis

1.1.1 Data features and class label

The training data consists a total of 50,000 rows of data, with 19 features. The target feature is 'genre' and the classes are balanced - 5000 instances of each of the 10 possible classes. The dataset has seven categorical features 'artist_name', 'track_name', 'track_id', 'key', 'mode', 'time_signature', and the class label 'genre'.

Of the categorical features, 'key' and 'mode' refer to musical attributes (i.e. 'key' is the predominant note that the song is played, 'mode' indicates if the song is in the major or minor key) and could provide some indication of musical genre.

The dataset has 12 numerical features 'instance_id', 'popularity', 'acousticness', 'danceability', 'duration_ms', 'energy', 'instrumentalness', 'liveness', 'loudness', 'speechiness', 'tempo', and 'valence'.

Of the numerical features:

- 'instance_id' is an arbitrary value for identification only
- 'popularity' has integer values ranging from 0 - 96, with median 41
- 'acoustic' has float values ranging 0-1, with mean 0.35
- 'danceability' has float values, ranging 0-1, with mean 0.57
- 'duration_ms' has float values, ranging from -1 to 4.8e06, with mean 1.5e05
- 'energy' has float values, ranging from 0-1, with mean 0.62
- 'instrumentalness' has float values, ranging from 0-1, with mean 0.09
- 'liveness' has float values, ranging from 0-1, with mean 0.26
- 'loudness' has float values, ranging from -38.4 to 3.
- 'speechness' has float values, ranging from 0-1, with mean 0.17
- 'valence' has float values, ranging from 0-1, with mean 0.49

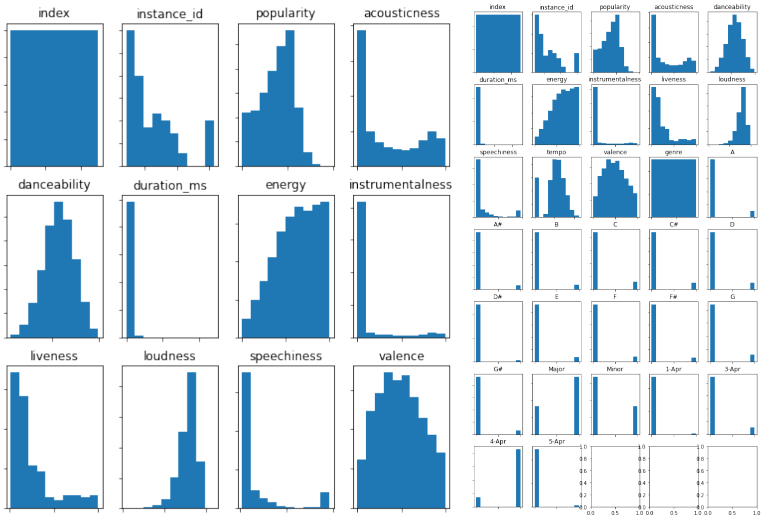

Figure 1: Histograms of raw numeric features and encoded features

1.1.2 Insights from histograms

We can see from the histograms (Figure 1) that ‘acousticness’, ‘duration_ms’, ‘instrumentalness’, and ‘speechiness’ are heavily weighted toward low values, with more than three-quarters of the instances in the lowest bins. Most of these features provide little discriminatory power because they are essentially the same across all of the different classes, so make good candidates for removal from the model.

‘danceability’, ‘popularity’, and ‘valence’ have distributions that appear normal. ‘loudness’ and ‘energy’ are have heavier left tails, and ‘liveness’ has a heavier right tail. Although these features are individually imbalanced, they may have useful interactions that help predict music genre.

We can also plot histograms of encoded categoric features, although the encoded categoric features (‘key’, ‘mode’, ‘time_signature’) don’t appear to offer much additional value.

1.1.3 Handling missing and invalid values

We see that ‘tempo’ has a question mark as a value and there are some songs with a ‘duration_ms’ of ‘-1 ms’, which is an invalid value. We can replace these invalid values or remove the instances that have these invalid values. Checking how many times these values appear to determine how we can treat them, we see that ‘tempo’ has 7461 occurrences of ‘?’, and ‘duration_ms’ has around 10,000 occurrences of ‘-1’.

We have a range of options for treating the invalid values: remove the missing row with the invalid value, impute some value to replace the invalid value, or replace the value with some arbitrary value.

Since ‘duration_ms’ has a heavily skewed distribution (i.e. mostly very low values) so it is unlikely that this feature will contribute much to the predictive power of the model. If we impute values in this feature it may mask the predictive power of other features, so we will remove those instances with invalid values in ‘duration_ms’.

1.1.4 Feature combinations

We can examine pairs of features to check for correlation to identify if highly correlated features that may present options for removal. We run scatter plots across all features but note only three interesting insights:
1. ‘tempo’ feature has strong predictive power for some genres (indicated by strong colour banding in combination with multiple features in Figure 2)
2. ‘popularity’ feature has some weaker predictive power but still exhibits some clustering of genres, when combined with some features (see Figure 3)
3. some feature pairs appear to have positive correlation (e.g. ‘energy’ & ‘loudness’, ‘danceability’ & ‘valence’), some negatively featured (e.g. ‘acousticness’ & ‘loudness’, and ‘acousticness’ & ‘energy’) (see Figure 4)

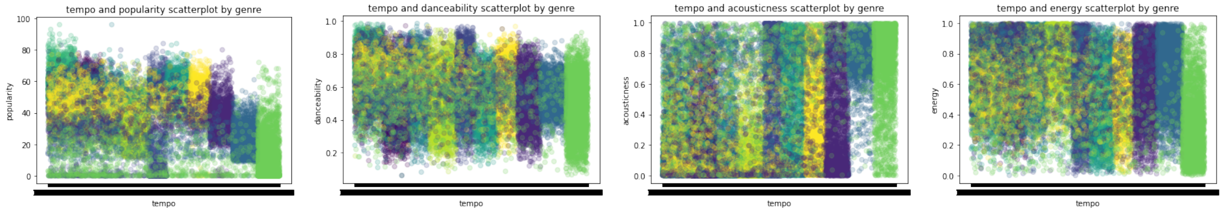

Figure 2: Scatterplots showing clear banding in ’tempo’ feature

1.1.5 Insights from the initial analysis

The key insights from the initial analysis are that ‘tempo’ looks to be a good predictor of some music genres; ‘popularity’ is lesser but still a reasonable predictor of music genre; ‘loudness’ is positively correlated

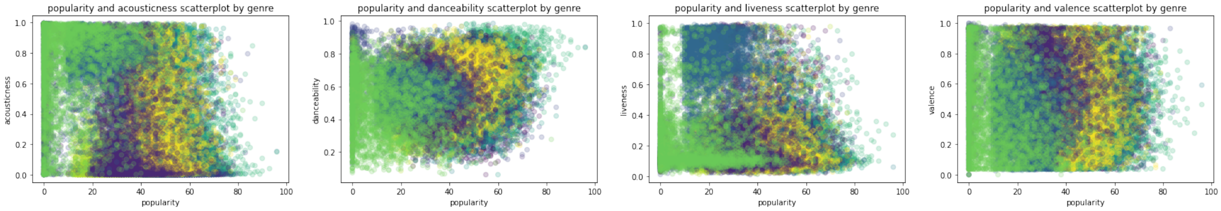

Figure 3: Scatterplots showing some genre clustering by 'population' feature

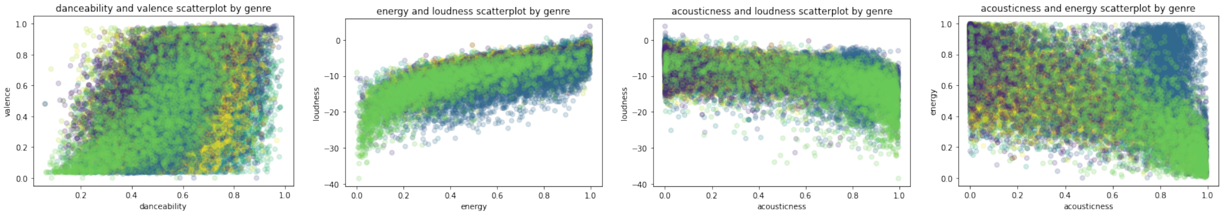

Figure 4: Scatterplots showing correlation between features

with 'energy' and negatively correlated with 'acousticness' so is potentially redundant against two other features; most categorical features have heavily unbalanced distributions and are unlikely to offer much predictive power. With a dataset of 50,000 features, we can either remove the instances with invalid features (i.e. 10,000 in 'duration_ms' and 7,500 in 'tempo') and still have sufficient instances to establish a good predictive model. Alternatively we could consider removing the 'duration_ms' feature since it has a heavily biased distribution, but we should likely retain 'tempo' because it has shown clear banding by genre, so this is a better candidate for imputing invalid values.

2 Developing and testing machine learning systems

Before selecting any particular model, I scaled the dataset (excluding 'artist_name', 'track_name', 'track_id', and the label 'genre'), split the data into train and test sets (50:50) and then ran a Logistic Regression model, using 'elasticnet' as the penalty. Logistic Regression is a model that allows each feature to contribute to the prediction, but I introduced a penalty for more complex models, encouraging a simpler model.

I did a bit of work reducing the number of features in the dataset, by manually removing the features identified in the initial data analysis and retaining just

In designing my system I had just finished working up a neural network using the tensorflow.keras library so I decided to try applying that to this system (not a great decision, but more on that in the lessons learnt part). I chose a neural network that took a couple

2.1