Sunday, October 6th

Music Taste Prediction

Lab 1 - FRTN65, Modelling & learning from data

Eric Portela

Outline of my approach

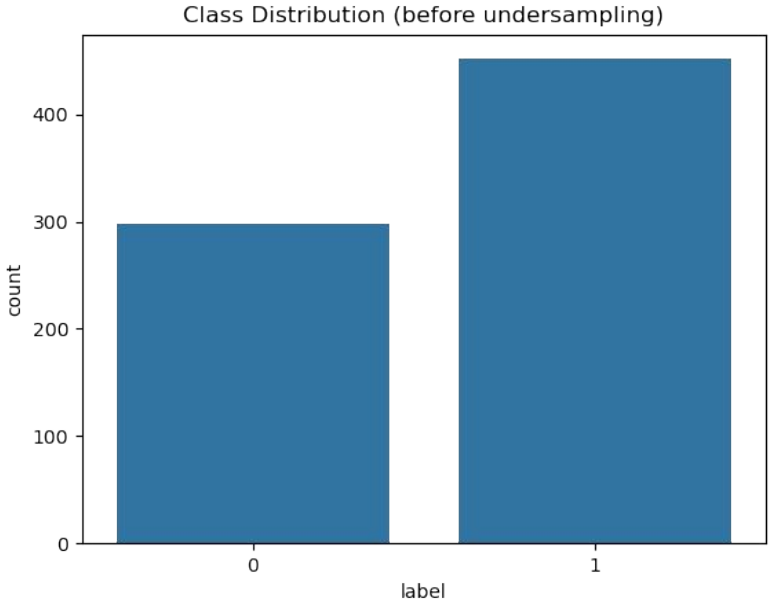
1. Exploratory data analysis (EDA) / Data preprocessing
    a. Get an understanding of the data
    b. Check for class imbalance
    c. Find potential redundancy through pairwise correlation matrix
    d. Histogram plot for applying suitable scaling method
2. Split data into features (X) and target variable (y)
3. Train 4 different models with different range of flexibility and interpretability
    a. Applied a 10-fold Grid Search to search each model's hyperparameter space
    b. KNN
    c. Multiple Logistic Regression
    d. Random Forest
    e. SVC

EDA - Data Overview (***songs_to_classify*** **dataset**)

- 750 songs, no missing values in dataset
- Slight class imbalance (298 NO, 452 YES)
- High pairwise correlation between a set of three features
  - acousticness/energy: -0.78
  - acousticness/loudness: -0.70
  - energy/loudness: 0.83
- Scaling of features based on the distribution of each feature
  - Applied 3 different scaling methods
    - linear scaling
    - z-scaling
    - log-scaling

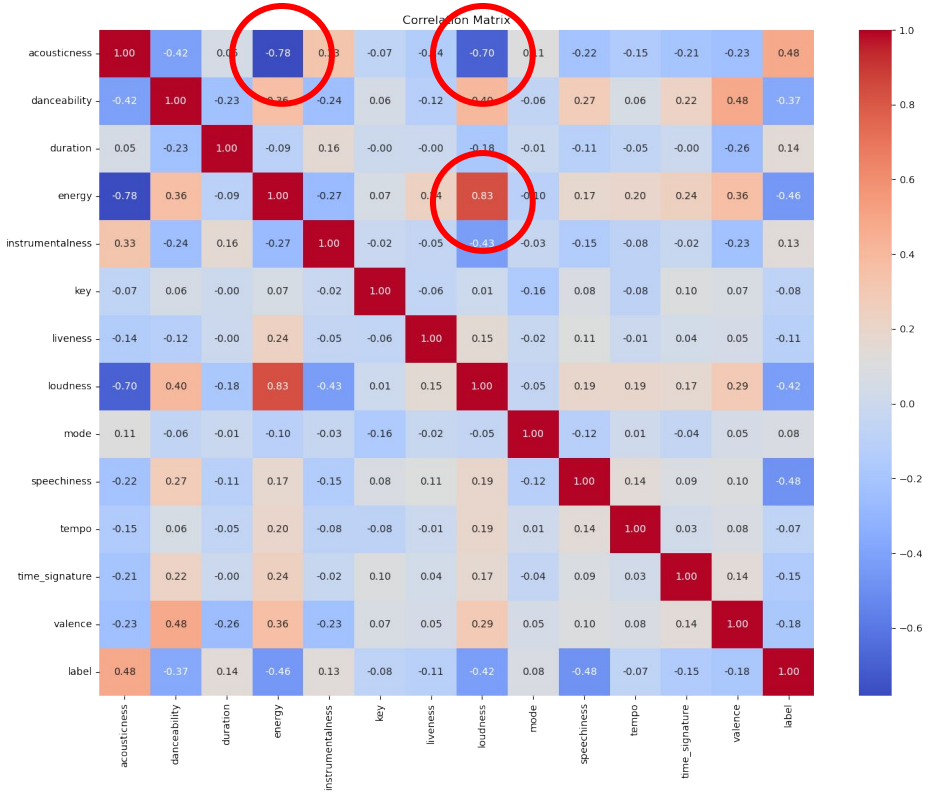
EDA - Imbalanced class data + pairwise plot

- Roughly 40% NO, 60% YES
- Addressed using undersampling of the majority class (to reach 50/50)



EDA - Correlation matrix

- Pairwise plot showed high correlation between 3 sets of features
  - acousticness/energy: -0.78
  - acousticness/loudness: -0.70
  - energy/loudness: 0.83
- I decided to interpret ‘high’ as ≥ 0.7 or ≤ -0.7
- I decided to drop ‘energy’ and ‘loudness’ to reduce potential redundancy for the models and/or multicollinearity



Choice of models - Interpretability vs flexibility

- I decided to train
  - KNN
  - Logistic Regression
  - Random Forest
  - SVC
- When choosing the models I sought not only to seek models that could give me reasonable predictions, but also try to understand the ‘black box’ or the estimate *f* if you will (*inference*)
- Hence, I choose the models based on this, where both KNN and SVC can be seen as less interpretable (more flexible) while the Multiple Logistic Regression and Random Forest are less flexible (more interpretable)

Model training

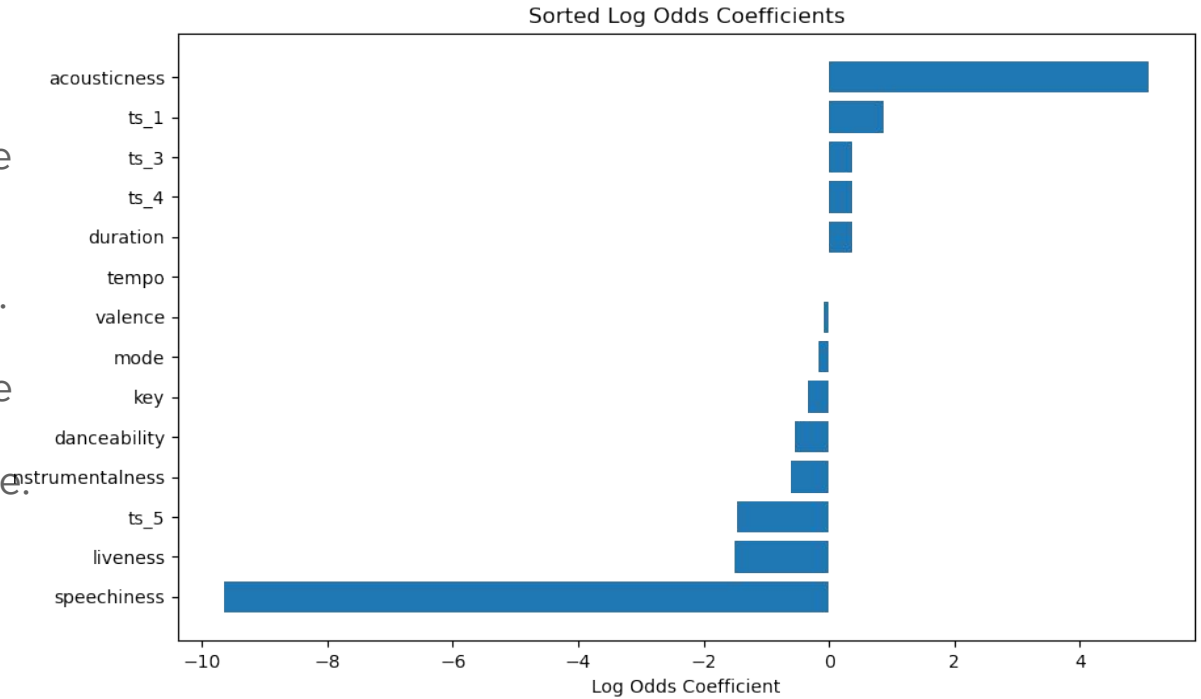
- I split the data into an 80-20 ratio
  - 80% for k-fold cross validation
  - 20% for test data to perform the predictions and print classification report
- I choose to train and tune the models using k-fold Cross Validation (k=10) with Grid Search
- **Why k-fold CV**: To reduce bias and variance, reduce variability in terms of a single train-test split.
- **Grid Search** is a greedy algorithm for hyperparameter tuning, given a parameter grid that defines the hyperparameter space you want to search within

Model I - KNN

- **Motivation**: To train a model with high flexibility and low interpretability. Furthermore it does not make any assumptions in terms of a predefined form for the decision boundary, because it’s a non-parametric model.
- **Presteps**:
  - Performed a **PCA** in order to reduce the dimensionality of the data because it is known that KNN performs worse on high dimensionality data
  - First **three** components seemed most relevant in terms of carrying the most explained variance, hence I choose these for model training

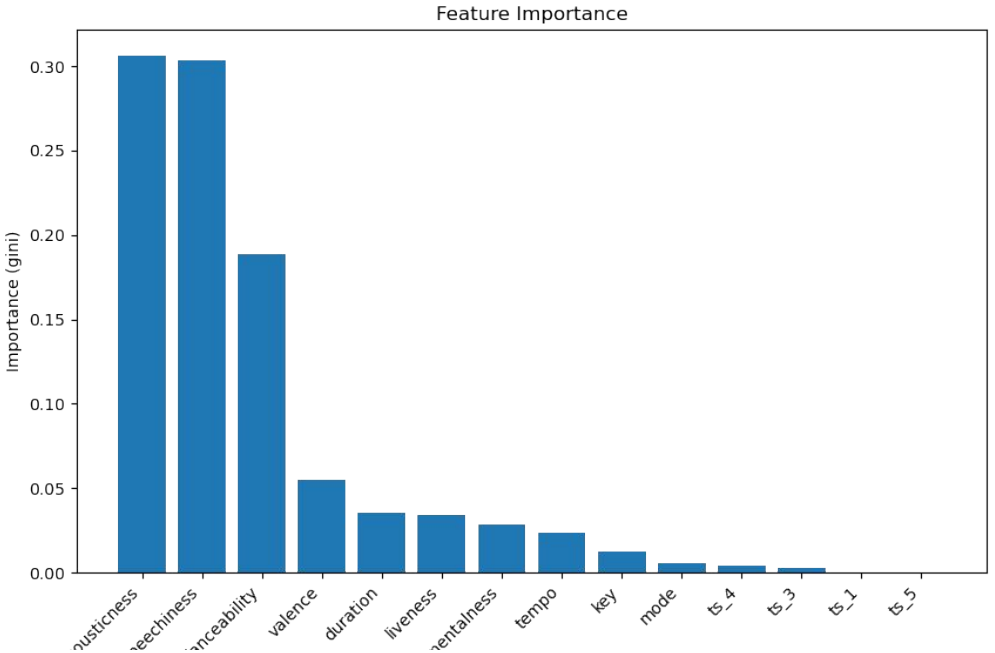
Model II - Logistic Regression

- **Motivation**: To train a model with high interpretability.
- The model provides information to understand the relationship between each feature to the target variable through the logarithmic odds. Essentially this explains how much one unit increase of the specific feature affects the outcome of the target variable.



Model III - Random Forest Classifier

Motivation: To train a tree-based model, which is highly interpretable (feature importance plot) has non-linear decision boundaries.



Model IV - SVC using gamma

- **Motivation**: To train a model with high flexibility, can be instantiated using either linear or non-linear decision boundaries.
- The kernel and the gamma value were one of the hyperparameters tuned for the SVC.

Model Comparison

KNN

	precision	recall	f1-score	support
0	0.77	0.68	0.72	69
1	0.63	0.73	0.67	51
accuracy			0.70	120
macro avg	0.70	0.70	0.70	120
weighted avg	0.71	0.70	0.70	120

Logistic Regression

	precision	recall	f1-score	support
0	0.83	0.87	0.85	69
1	0.81	0.76	0.79	51
accuracy			0.82	120
macro avg	0.82	0.82	0.82	120
weighted avg	0.82	0.82	0.82	120

Random Forest Classifier

	precision	recall	f1-score	support
0	0.84	0.83	0.83	69
1	0.77	0.78	0.78	51
accuracy			0.81	120
macro avg	0.80	0.81	0.80	120
weighted avg	0.81	0.81	0.81	120

SVC

	precision	recall	f1-score	support
0	0.79	0.75	0.77	69
1	0.69	0.73	0.70	51
accuracy			0.74	120
macro avg	0.74	0.74	0.74	120
weighted avg	0.74	0.74	0.74	120

Conclusions

- In terms of the F1-score (the harmonic mean between precision and recall) it seems that both the Logistic Regression and Random Forest performs the best in terms of predicting Andreas music taste
- In terms of the logarithmic odds it seems that Andreas prefers acoustic songs over speechy ones
- The feature importance plot showed (Random Forest) showed that acousticness, spechiness and danceabillity were very important features in terms of predicting the target variable
- No solid conclusion can be drawn regarding what type of decision boundary is more suitable, as the SVC with rbf also performed quite well