Appendix

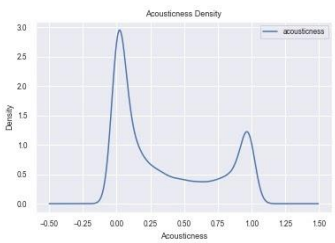
Glossary (Definitions from Google's ML Glossary: Referenced Later)

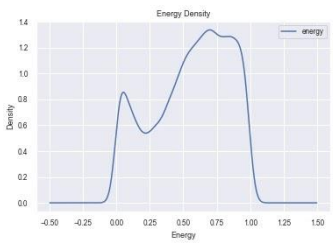
- **accuracy.** The fraction of predictions that a classification model got right. In multi-class classification, accuracy is defined as follows: the number of correct predictions divided by the total number of predictions.
- **attribute.** Synonym for features
- **binary classification.** A type of classification task that outputs one of two mutually exclusive classes.
- **categorical data.** Features having a discrete set of possible values.
- **class.** One of a set of enumerated target values for a label.
- **classification model.** A type of machine learning model for distinguishing among two or more discrete classes.
- **confusion matrix.** An NxN table that summarizes how successful a classification model's predictions were; that is, the correlation between the label and the model's classification. One axis of a confusion matrix is the label that the model predicted, and the other axis is the actual label. N represents the number of classes.
- **continuous feature.** A floating-point feature with an infinite range of possible values.
- **cross-validation.** A mechanism for estimating how well a model will generalize to new data by testing the model against one or more non-overlapping data subsets withheld from the training set.
- **dataset.** A collection of examples.
- **decision tree.** A model represented as a sequence of branching statements.
- **discrete feature.** A feature with a finite set of possible values.
- **ensemble.** A merger of the predictions of multiple models.
- **feature.** An input variable used in making predictions.
- **generalisation.** Refers to your model's ability to make correct predictions on new, previously unseen data as opposed to the data used to train the model.
- **gradient descent.** A technique to minimize loss by computing the gradients of loss with respect to the model's parameters, conditioned on training data. Informally, gradient descent iteratively adjusts parameters, gradually finding the best combination of weights and bias to minimize loss.
- **hyperparameter.** The "knobs" that you tweak during successive runs of training a model.
- **interpretability.** The ability to explain or to present an ML model's reasoning in understandable terms to a human.
- **linear model.** A model that assigns one weight per feature to make predictions.
- **logistic regression.** A classification model that uses a sigmoid function to convert a linear model's raw prediction (y') into a value between 0 and 1.
- **log-odds.** The logarithm of the odds of some event.
- **metric.** A number that you care about. May or may not be directly optimized in a machine-learning system. A metric that your system tries to optimize is called an objective.
- **model.** The representation of what a machine learning system has learned from the training data.
- **model training.** The process of determining the best model.
- **multinomial classification.** Classification problems that distinguish between more than two classes.
- **multinomial logistic regression.** Using logistic regression in multi-class classification problems.
- **noise.** Broadly speaking, anything that obscures the signal in a dataset. Noise can be introduced into data in a variety of ways.
- **objective function.** The mathematical formula or metric that a model aims to optimize.
- **overfitting.** Creating a model that matches the training data so closely that the model fails to make correct predictions on new data.
- **performance.** A measurement of either how efficiently a piece of software runs or how correct the model is
- **positive (class).** In binary classification, the two possible classes are labeled as positive and negative. The positive outcome is the thing we're testing for.
- **precision.** A metric for classification models. Precision identifies the frequency with which a model was correct when predicting the positive class.
- **prediction.** A model's output when provided with an input example.
- **preprocessing.** Processing data before it's used to train a model.
- **random forest.** An ensemble approach to finding the decision tree that best fits the training data by creating many decision trees and then determining the "average" one. The "random" part of the term refers to building each of the decision trees from a random selection of features; the "forest" refers to the set of decision trees.
- **recall.** A metric for classification models that determines how many, out of all the possible positive labels, were identified correctly.
- **regularisation.** The penalty on a model's complexity. Regularisation helps prevent overfitting.

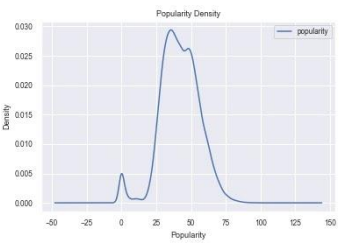
- **ridge regularisation.** A type of regularisation that penalizes weights in proportion to the sum of the squares of the weights.
- **sigmoid function.** A function that maps logistic or multinomial regression output (log odds) to probabilities, returning a value between 0 and 1.
- **supervised machine learning.** Training a model from input data and its corresponding labels.
- **target.** In supervised learning, the "answer" or "result" portion of an example.
- **test set.** The subset of the dataset that you use to test your model after the model has gone through initial vetting by the validation set.
- **training.** The process of determining the ideal parameters comprising a model.
- **train/training set.** The subset of the dataset used to train a model.
- **validation.** A process used, as part of training, to evaluate the quality of a machine learning model using the validation set.
- **validation set.** A subset of the dataset—disjoint from the training set—used in validation.
- **weight.** A coefficient for a feature in a linear model, or an edge in a deep network.

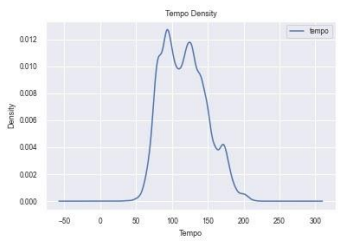
Intermediate Results and Implementation Choices

- In the EDA, a maximum number of 4000 observations was set for each of the 6 genres we decided to retain in our dataset.
    - This was due to the dataset being too large for the algorithms to run in a reasonable amount of time.
    - 4000 was decided upon as it maximised our dataset, whilst making sure the algorithms didn't take an excessive amount of computation time.
    - We set a consistent limit across all of the models, so that we wouldn't lose the fairly even distribution of the target classes, in turn guaranteeing they are exactly evenly distributed.
- k=10 was decided for the k-fold cross-validation so that we could reduce bias in our model training and obtain cross-validation errors that would shed more light on which of the two models would be more appropriate.
- Both models were chosen for their lack of assumptions made about distribution of classes or attributes, as many of the attributes displayed irregular distributions in the kernel density estimation graphs produced in the EDA (examples below).









Multiple Multinomial Logistic Regression (MLR)

- Trained and tested using the functions *mnrfit* and *mnrval*.
- Errors calculated through the use of the *crossval* function.
- Using VIF to filter out multicollinear attributes was tested with 2 common VIF thresholds:
    - *VIF threshold = 2.5:* The features 'energy' and 'loudness', with VIFs of 5.7674 and 2.9558 respectively, were filtered out of the dataset.
    - *VIF threshold = 5:* The feature 'energy', with a VIF of 5.7674, was filtered out of the dataset.
    - The classification errors for these two cases are shown in the table on the right.

	Train Error	Test Error	10-fold CV Error
VIF threshold = 2.5	33.15%	34.72%	33.25%
VIF threshold = 5	32.34%	33.63%	32.48%

- VIFs were calculated as the diagonal elements of the inverse of the correlation matrix (Belsley, Kuh and Elsch, 1980, p. 93)
- Final model fit on a dataset that wasn't subject to filtering of attributes based on VIFs, due to decreased performance metrics when attributes were removed.

Random Forest (RF)

- Model hyperparameters optimised using the *fitcensemble* function with the 'Method' option set to 'Bag' in order to specify bootstrap aggregation, which is a defining feature of RFs.
- The '*OptimizeHyperparameters*' option was set to optimise the number of learning cycles, the minimum leaf size, and the number of variables to sample for each split in a tree.
- Once the hyperparameters had been optimised, they were assigned to variables for use in our model fitting.
- Fit the model using the 10-fold partition as well as the test set, both using *fitcensemble*.
    - Optimised hyperparameters (minimum leaf size and the number of variables to sample for each split) are specified within the *templateTree* function preceding *fitcensemble*, which creates a tree learner template.

- The remaining hyperparameter, the number of learning cycles, is used as an option within *fitcensemble.*
- Testing and training error calculated using the *loss* function.
- 10-fold cross-validation error calculated using the *kfoldLoss* function

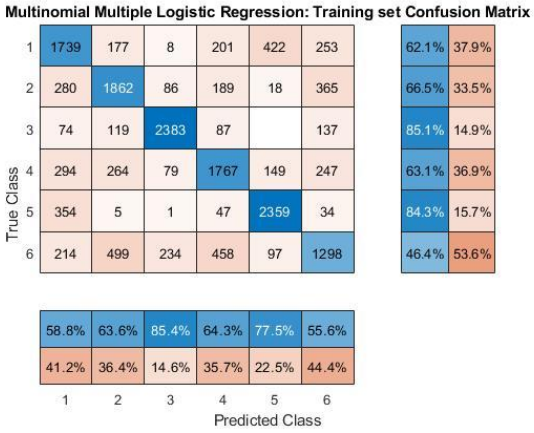
MLR & RF

- Confusion matrices are computed using the *confusionchart* function.
- Due to the task involving multinomial classification, we had to use both the micro-averages and macro-averages of precision, recall and F1 scores.

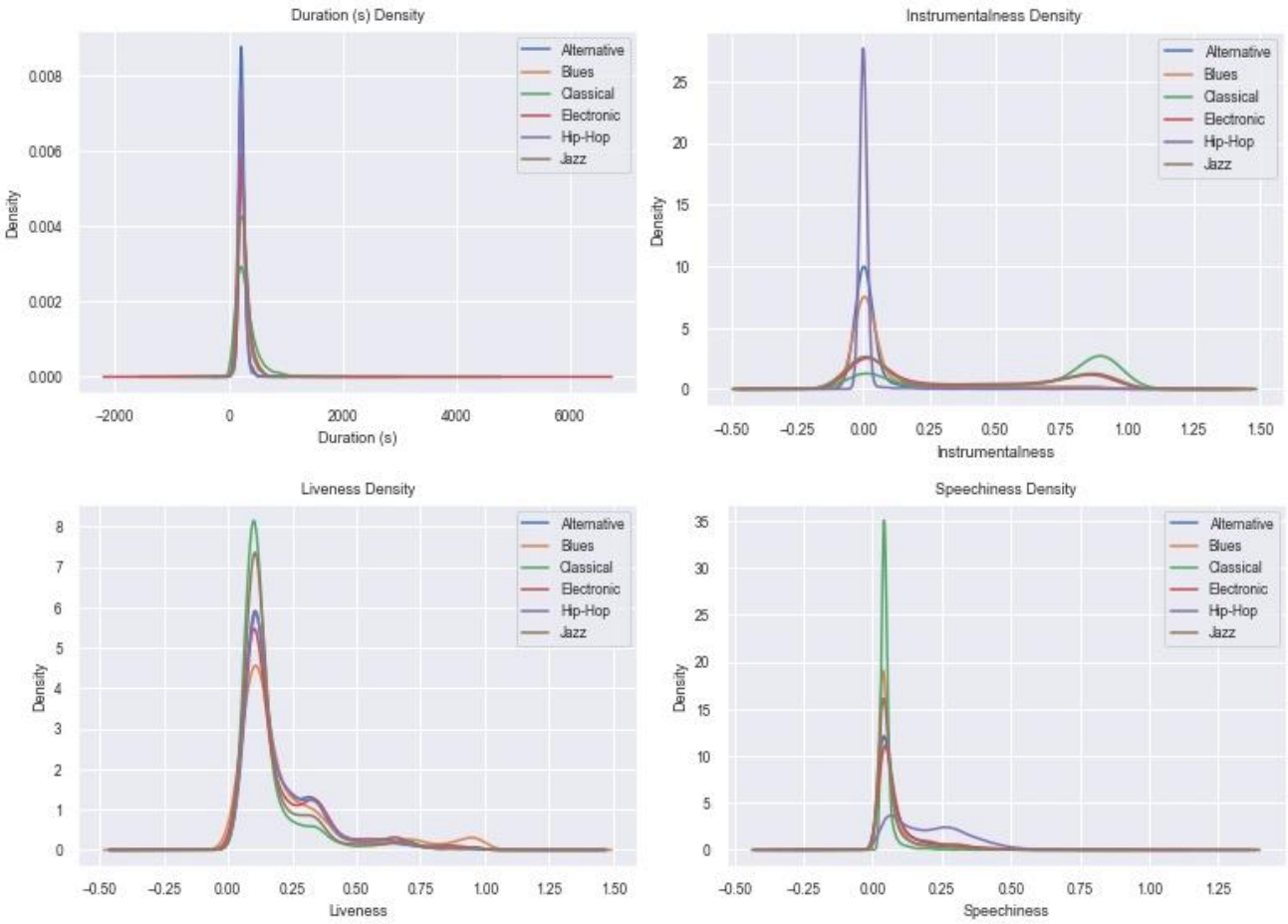
Performance Tables & Graphs (excluded from the poster and previous sections of the appendix)

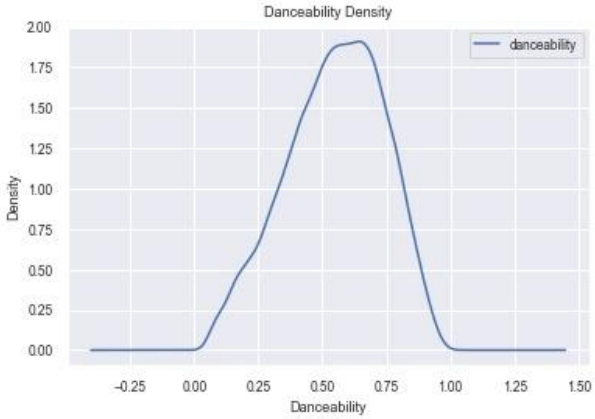
Computation time (seconds):

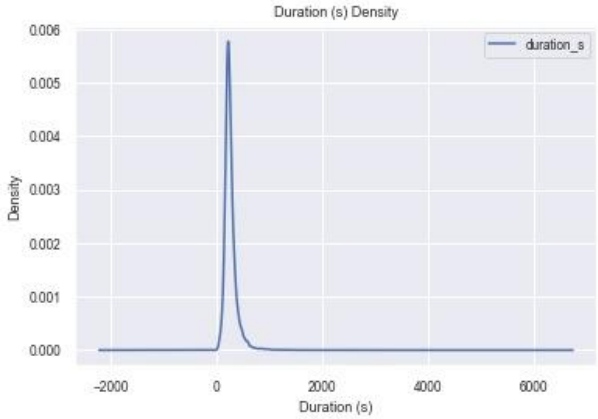
	Training		Testing		Total	
	Train	10-fold	Test	10-fold	Train/Test	10-fold
MLR	38.68	319.07	1.11	43.93	39.79	363
RF	43.72	371.37	4.4	17.02	48.12	388.40

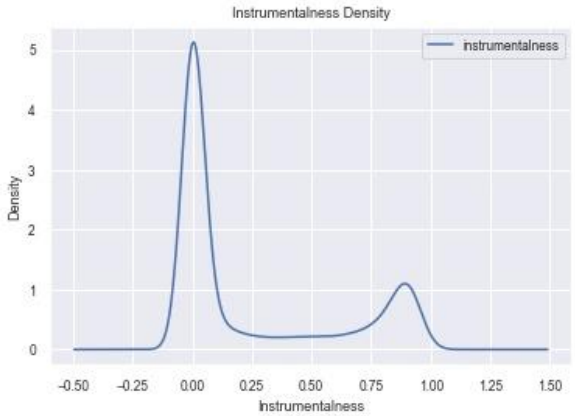


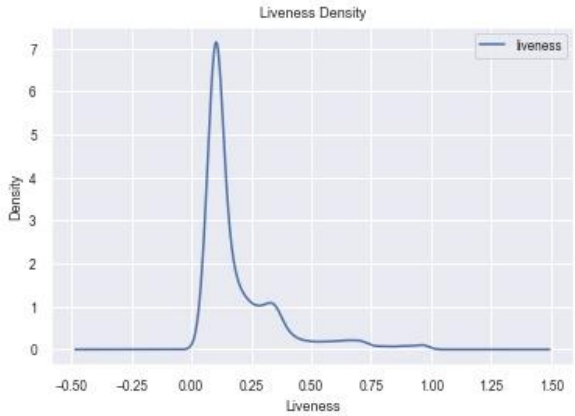
Kernel Density Estimation Graphs

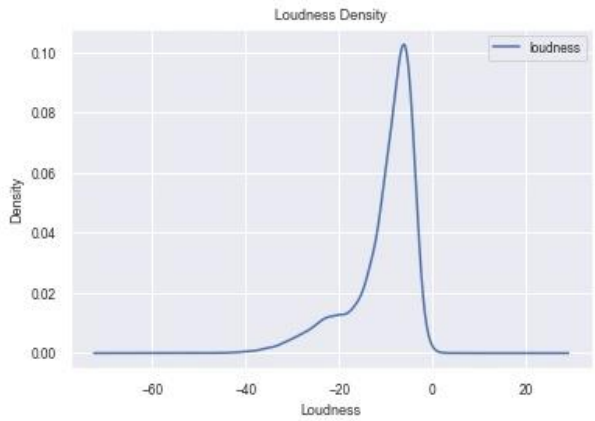


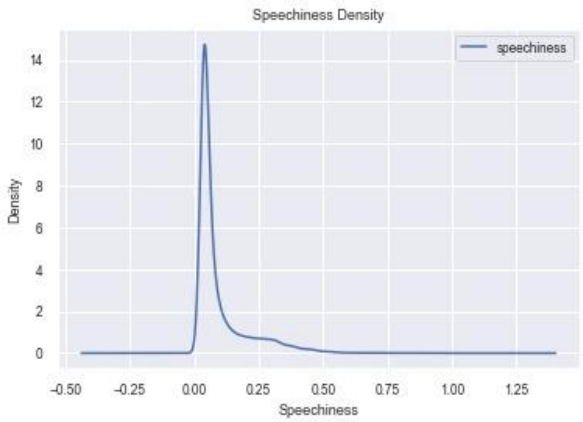


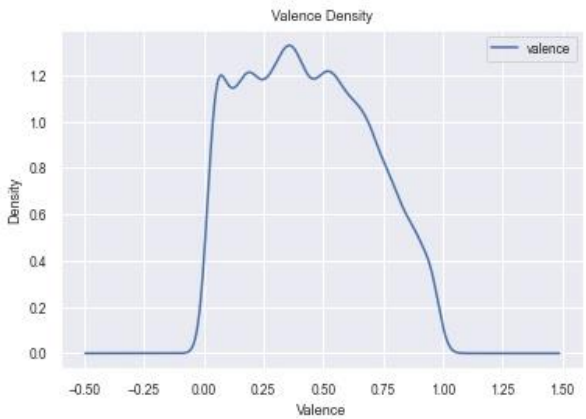












Appendix References:

Belsley, D. A., Kuh, E. and Elsch, R. E. (1980) *Regression diagnostics: Identifying influential data and sources of collinearity.* Nashville, TN: John Wiley & Sons.

Machine learning glossary, Google Developers. Available at: https://developers.google.com/machine-learning/glossary (Accessed: December 15, 2021).