Assessment 2 Report

MODULE: MACHINE LEARNING WITH R

AUTHORS: FREDERIKA COOK, ZHEN CHEN, DURU DEMIRBAG, HANNAH MCAULEY

1.Introduction

1.1. Dataset Construction and Preprocessing

This project combines Spotify audio features with historical Billboard Hot 100 chart data to create a supervised learning dataset for predicting chart success.

Billboard chart data was sourced from the utdata/rwd-billboard-data GitHub repository, providing weekly rankings from the 1950s to 2025. Each row represents a track's chart position, title, artist, and weeks on chart. Spotify audio features were obtained from Kaggle, covering six decades (1960s–2019) and including attributes such as energy, danceability, tempo, loudness, and valence, along with artist and track metadata.

To merge the datasets, we developed a custom Python script that normalised artist and track names (e.g., lowercasing, removing punctuation and accents, and standardising terms like "feat."). Tracks were matched on these normalised values. Matched songs were labelled `on_billboard = 1`, while unmatched ones were assigned `0`. For matched entries, we also recorded peak position, weeks on chart, and the dates of first and last appearance.

The resulting dataset comprised 40,560 unique tracks: 20,030 hits and 20,530 non-hits, creating a near-balanced class distribution ideal for classification. However, the selection of non-hits—defined as tracks by artists with no Billboard history, from underrepresented genres, and limited to the U.S. market—may introduce bias. While this ensures strong contrast, it could limit generalisability beyond mainstream chart dynamics.

2.1 Data Integrity and Initial Exploration

The dataset contained no missing values or anomalies requiring imputation, providing a robust foundation for analysis. Initial inspection showed that some features were bounded between 0 and 1, while others displayed skewness or scale differences. These findings informed a structured EDA pipeline encompassing transformation, standardisation, encoding, and correlation analysis.

2.2 Distribution Assessment and Transformation Decisions

Histograms, QQ plots, and summary statistics were used to assess the distributions of numeric variables. Four features—`duration_ms`, `sections`, `chorus_hit`, and `speechiness`—exhibited heavy right skew and high kurtosis. Logarithmic transformations (*Figure 1*) improved their normality and stabilised variance.

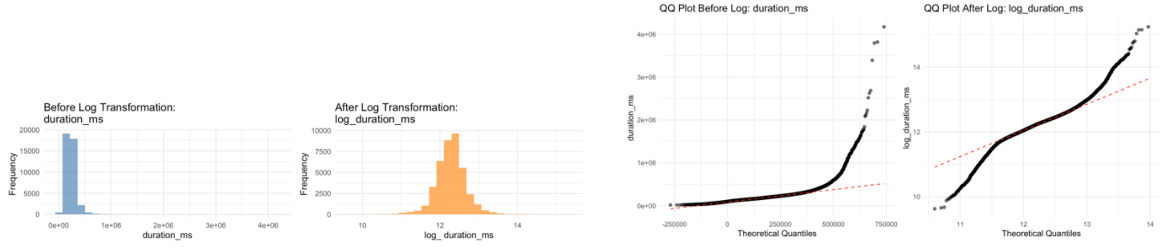


Figure 1: Histogram and QQ-Plots demonstrating `duration_ms` *before and after log-transformation.*

Instrumentalness showed a bimodal distribution reflecting vocal vs. instrumental tracks. As log transformation did not resolve this, it was retained in raw form, with categorisation considered during modelling. Features such as `acousticness` and valence also displayed multimodal patterns, likely tied to genre or emotional content. Binning was explored but ultimately deferred to tree-based models, which natively detect nonlinear splits.

2.3 Standardisation

Z-score standardisation was applied to variables with large numeric ranges—`log_duration_ms`, `log_sections`, `log_chorus_hit`, `tempo`, and `loudness`—to harmonise scales (*Figure 2*). This was especially beneficial for distance-based models like SVMs. However, standardisation reduces interpretability by detaching features from their original units.

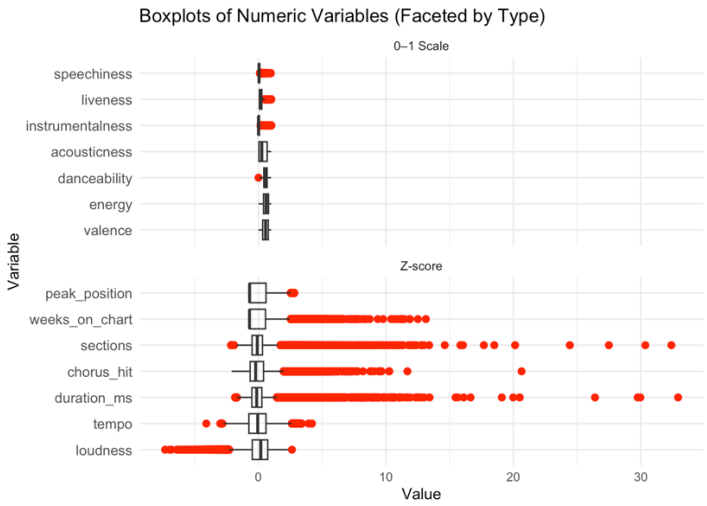


Figure 2: Boxplots of Numeric Variables, segmenting probabilistic features from standardised.

2.4 Encoding of Categorical and Cyclical Features

Discrete variables were encoded to align with modelling requirements. mode (major/minor) was retained as-is due to its binary format. `time_signature` was treated as a categorical factor, and although `key` is numeric (0–11) and cyclical, sine-cosine encoding offered no clear advantage. For simplicity and interpretability, `key` was also treated as a factor.

2.5 Correlation Structure and Feature Redundancy

Pearson correlations revealed strong collinearity between `energy` and `loudness` (r = 0.77) and between `duration_ms` and `sections` (r = 0.89) (*Figure 3*). To minimise redundancy, `energy` and `duration_ms` were retained for their interpretability, while `loudness` and `sections` were excluded from later models. A correlation network (*Figure 4*) showed clusters reflecting musical intensity (`energy`, `tempo`), mood and feel (`valence`, `danceability`, `acousticness`), structure (`duration_ms`, `chorus_hit`), and chart performance (`peak_position`, `weeks_on_chart`).

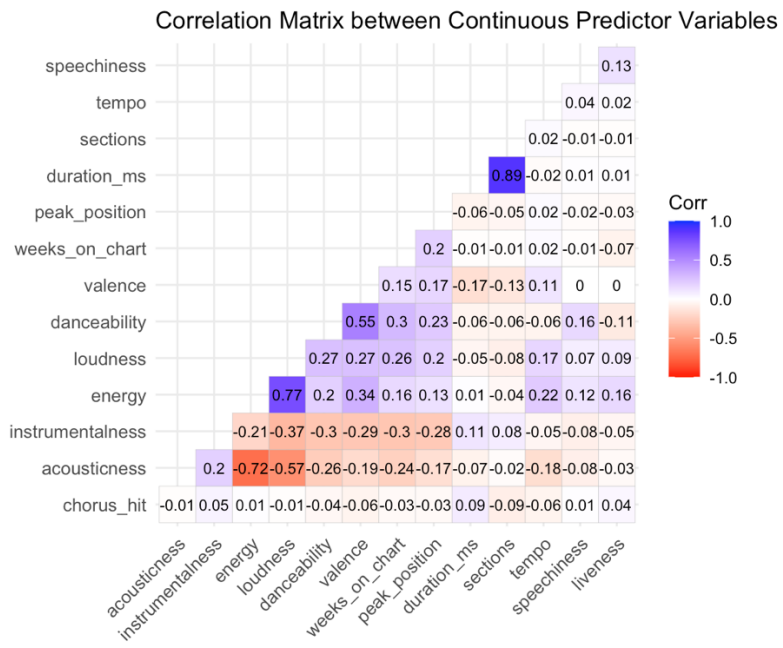


Figure 3: Correlation heatmap of continuous predictor variables.

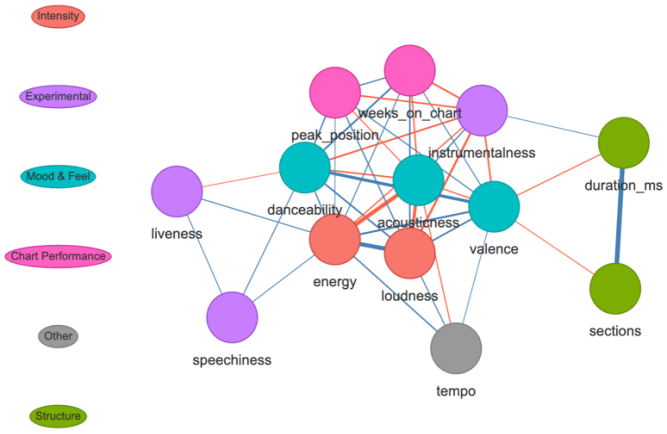


Figure 4: Correlation network graph (correlation threshold = 0.1)

2.6 Class-Based Differences and Feature Distributions

Violin plots (*Figure 5*) showed that Billboard hits tended to have higher `danceability`, `valence`, and `energy`, and lower `instrumentalness`, `speechiness`, and `acousticness`, reflecting a preference for upbeat, vocal-driven tracks. Charting songs also featured shorter `chorus_hit` times and fewer `sections`, suggesting that structural simplicity may aid listener retention. While `peak_position` and `weeks_on_chart` showed clear class separation, they were excluded from modelling due to risk of data leakage.

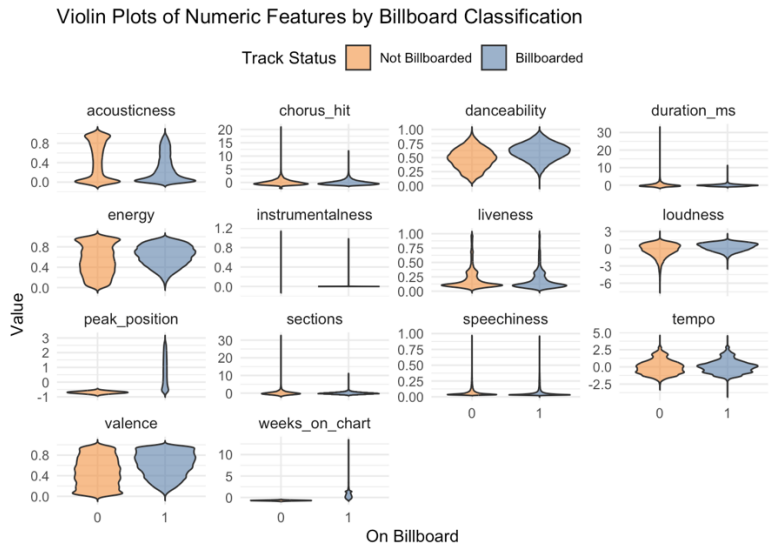


Figure 5: Violin plots of numeric features by Billboard classification.

2.7 Interaction Effects and Nonlinear Trends

Interactive scatterplots (Figure 6) revealed nuanced multivariate trends. `danceability` showed a weak positive association with `weeks_on_chart`, especially among top 10 hits. Diverging slopes in `chorus_hit` suggested that structural timing may differ between charting and non-charting tracks. Clustering in the `valence`–energy space pointed to emotional combinations conducive to hit potential. These observations supported the inclusion of interaction terms and the use of nonlinear models in the next phase.

2.8 Statistical Testing and Feature Relevance

Independent t-tests confirmed significant differences ($p < 0.001$) between hits and non-hits for key variables including `danceability`, `energy`, `valence`, `acousticness`, and `loudness`, validating their relevance for classification.

3. Final Feature Selection and Preprocessing Summary

A full breakdown of feature transformations, encodings, and exclusions is included in *Table 8.1.1*. These steps ensured standardised scaling, enhanced interpretability, and reduced multicollinearity.

4. Classification Techniques

To evaluate the predictive strength of the dataset, we applied a range of classification algorithms that span both interpretable and complex models. These included Logistic Regression as a baseline, alongside Random Forest, XGBoost, and Support Vector Machines (SVM), each capable of capturing non-linear relationships to varying degrees.

Logistic Regression offers simplicity and interpretability through a linear decision boundary but may struggle with complex feature interactions.

Random Forest enhances Decision Trees by averaging multiple models trained on different data subsets, improving generalisation while providing robust feature importance metrics, albeit with reduced interpretability.

XGBoost applies gradient boosting to iteratively correct errors, achieving high accuracy and modelling intricate patterns, though it requires more tuning and computational resources.

SVM constructs an optimal separating hyperplane and, using kernel functions, can capture non-linear relationships, though it can be computationally intensive and less interpretable.

By employing this diverse suite of models, we aimed to balance predictive accuracy, efficiency, and interpretability in identifying the most suitable approach.

5. Model Methodology and Results

5.1. Evaluation Metrics

To compare model performance, we used standard classification metrics:

- **Accuracy**: Overall proportion of correct predictions.
- **Precision**: Proportion of positive predictions that are correct; important when false positives are costly.
- **Recall (Sensitivity)**: Proportion of actual positives correctly identified; critical when false negatives carry a higher cost.
- **F1 Score**: Harmonic mean of precision and recall, offering a balanced assessment of both.
- **ROC AUC**: Measures overall class separability across thresholds; higher values indicate stronger discriminative ability.

5.2. Logistic Regression

Methodology

To evaluate the relationship between audio features and chart success, multiple logistic regression models were trained using the selected features. Models were iteratively refined by removing statistically insignificant variables, testing log and standardised transformations, and introducing interaction terms. Each model was assessed using accuracy, precision, recall, F1-score, and ROC AUC (*see Figures 8.1.1–8.1.6*).

Results

The baseline model (Model 1) achieved balanced performance (Accuracy = 73.18%, F1 = 0.75) using standardised features and categorical predictors. Log-transforming skewed variables (Model 2) had negligible impact, suggesting transformations were not beneficial. Removing insignificant variables and predictor levels (Model 3) slightly improved recall (0.82) and F1 (0.76). Excluding the entire key variable (Model 4) led to a marginal drop in accuracy (73.70%) but maintained high recall, justifying its exclusion.

Adding an interaction term between energy and valence (Model 5) produced the highest AUC (0.812), although other metrics remained stable. Reintroducing previously removed variables as standardised (Model 6) slightly decreased performance, confirming they did not meaningfully contribute.

All models exhibited strong discriminatory power, with AUC values ranging from 0.806 to 0.812. Model 5 was selected as the final logistic regression model for its optimal trade-off between interpretability and performance. A summary of model comparisons is presented in *Table 5.2.1.*

MODEL	KEY CHANGES	ACCURACY	RECALL	PRECISION	F1-SCORE	AUC
Model 1	Baseline	73.18%	0.81	0.71	0.75	0.806
Model 2	Log-transformed variables	73.12%	0.81	0.71	0.75	0.806
Model 3	Removed insignificant predictors	73.91%	0.82	0.71	0.76	0.81
Model 4	Removed 'key' variable	73.70%	0.82	0.71	0.76	0.808
Model 5	Added 'valence' x 'energy' interaction	73.91%	0.81	0.71	0.76	0.812
Model 6	Reintroduced standardised dropped predictors	73.08%	0.78	0.71	0.74	0.806

Table 5.2.1*: Performance comparison of logistic regression models.*

5.3. Random Forest

Methodology

We initially built Model 1 using all features and trained it with 200 trees (`ntree = 200`) and a node size of 1. The `mtry` parameter was tuned across values {3, 5, 6, 7, 9, 11} using 10-fold cross-validation (10-CV), with AUC as the selection criterion.

Model 2 was then constructed using a reduced feature set based on the importance rankings from Model 1. The same cross-validation and tuning procedure was applied. Model 3 further refined the feature set and repeated the process. All three models were evaluated on a held-out test set using AUC, accuracy, and confusion matrices. The best-performing model across these metrics was selected.

Results

Model 1 achieved the highest AUC and accuracy on the test set (*Table 8.2.1*), outperforming Models 2 and 3. Confusion matrices (*Figures 8.2.4, 8.2.8, and 8.2.12*) show that Model 1 made the most correct predictions across both classes.

To test model stability, we increased the number of trees from 200 to 500. However, the original model slightly outperformed the 500-tree version across all evaluation metrics: accuracy (0.7888 vs. 0.7874), Kappa (0.5775 vs. 0.5745), balanced accuracy (0.7887 vs. 0.7872), and AUC (*Table 8.2.2, Figure 8.2.1*). This confirmed that 200 trees were sufficient for optimal performance.

Model 1 was therefore selected as the final Random Forest model. Feature importance rankings (*Figure 8.2.5*) and partial dependence plots (PDPs; *Figure 8.2.13*) illustrate the contribution of each variable to model predictions.

5.4. XGBoost

Methodology

XGBoost was applied with an 80/20 stratified train-test split and standard preprocessing: identifier removal, one-hot encoding (`key`, `time_signature`), and standardisation of numerical features (`duration_ms`, `tempo`, `loudness`).

A baseline model with default hyperparameters (`n_estimators=100`, `learning_rate=0.1`, `max_depth=6`) achieved strong performance (ROC AUC = 0.86; generalisation gap = 3.6%; Table 8.4.1, Figure 8.4.2). Grid search tuning (5-fold CV) marginally improved AUC (up to 86.8%) but introduced overfitting, especially with deeper trees and more estimators (gap = 14.1%). Regularisation parameters (`alpha`, `lambda`, `gamma`) were included to address this, though gains remained modest.

Threshold tuning (Figure 8.4.3) improved the precision–recall trade-off, with ~0.45 yielding optimal F1. Probability calibration via `CalibratedClassifierCV` (sigmoid) further aligned predicted probabilities with observed outcomes (Figure 8.4.4).

Feature importance analysis (Figures 8.4.5–6) consistently highlighted `speechiness`, `valence`, `duration_ms`, `instrumentalness`, `acousticness`, and `danceability`. RFECV (Figure 8.4.7) confirmed that performance was retained with a reduced feature set. SHAP plots (Figures 8.4.8–10) revealed nuanced non-linear effects, such as the curved interaction between `energy` and `valence`.

Dimensionality reduction (PCA, t-SNE) and KMeans clustering provided exploratory insights but limited classification utility due to feature overlap.

Results

Despite small gains in AUC and F1 from tuning, the baseline XGBoost model was selected for its strong generalisability and stable performance: accuracy = 78.1%, ROC AUC = 86.2%, precision = 74.0%, recall = 84.8%, and F1 = 79.0% (*Table 8.4.1*). Its low generalisation gap (3.6%) confirmed its robustness to unseen data.

5.5. Support Vector Machine

An SVM model was trained using the selected features: `danceability`, `energy`, `key`, `mode`, `speechiness`, `acousticness`, `instrumentalness`, `liveness`, `valence`, `duration_ms`, `time_signature`, and `chorus_hit`. As `key` and `time_signature` are categorical, and `mode` is binary, one-hot encoding was applied prior to training. All features were scaled to the [0, 1] range.

A radial basis function (RBF) kernel was chosen due to the non-linear separability of the data. Model training used a 70/30 train-test split, with 3-fold cross-validation for tuning the hyperparameters `C` and `gamma`. Despite the computational cost, this produced a well-performing baseline (Model 1), with accuracy and F1-score both at 0.76 (*Table 8.4.1*). The model prioritised recall for hits (class 1), capturing most Billboard tracks, though at the cost of slightly reduced precision (72%). Non-hit predictions (class 0) showed higher precision (83%), indicating greater conservatism in labelling non-hits.

Figure 8.4.1 shows the trade-off between precision and recall as the decision threshold varies. The F1-score peaked around 0.40–0.45, with accuracy remaining stable—suggesting an optimal threshold below the default of 0.5.

Model 2 was retrained with 10-fold cross-validation, using refined `C` and `gamma` ranges. Re-evaluation at the updated threshold of 0.4 achieved a more balanced performance across both classes, despite a marginal drop in accuracy from 0.77 to 0.76 (*Tables 8.4.2–8.4.3*). This threshold was therefore preferred for its fairer precision–recall trade-off.

As shown in *Figure 8.4.2*, the ROC curve for the final model yields an AUC of 0.84, indicating strong class separability. The chosen threshold (0.4), marked in red, achieves a high true positive rate while keeping false positives relatively low—validating the model's effectiveness and supporting its threshold adjustment.

6. Result Comparison

All models were evaluated using cross-validation to ensure fair comparison despite differing train-test splits. As shown in Table 8.6.1, Random Forest and XGBoost delivered the strongest results, each achieving 78% accuracy and an AUC of 0.86. SVM followed closely with 76% accuracy and AUC = 0.84, while Logistic Regression, though less powerful (73% accuracy, AUC = 0.81), offered high interpretability and performed respectably given its simplicity.

Feature influence was consistent across models. `instrumentalness` was the strongest predictor: tracks with low values (e.g., <0.1)—indicating vocal content—were more likely to chart. Similarly, lower `acousticness` and moderate `danceability` (~0.5–0.7) were associated with higher success, while extremely high danceability (>0.8) appeared counterproductive.

`energy` showed a U-shaped pattern: tracks with very low (<0.2) or very high (>0.8) energy were more likely to be hits, compared to mid-energy songs. `speechiness` was beneficial up to a threshold (~0.6), aligning with stylistic elements like rap or spoken-word intros.

Optimal `duration_ms` clustered around 2–3 minutes, reflecting current streaming-era norms. `valence` demonstrated that both high (happy) and low (melancholic) songs could succeed, with emotional intensity appearing more important than mood direction.

Less impactful features included `tempo`, `liveness`, and `chorus_hit`. Charting tracks typically had tempos between 50–100 BPM and introduced choruses later (~100+ seconds), possibly indicating build-up or delayed gratification. Tracks with either very few or many structural `sections` also performed slightly better, hinting at a preference for simplicity or standout structure.

Finally, `mode`, `key`, and `time_signature` showed negligible influence. Removing them reduced accuracy by <1%, confirming their minimal contribution to prediction.

7. Conclusion

This project explored whether a song's audio features could predict its appearance on the Billboard Hot 100, using machine learning models trained on a dataset spanning six decades of music. Tree-based classifiers—particularly XGBoost and Random Forest—outperformed linear models, achieving top performance (Accuracy = 78%, AUC = 0.86). These results highlight the non-linear nature of hit prediction, where complex interactions between audio traits play a central role.

Even simpler models like Logistic Regression performed well above chance (>70% accuracy), suggesting that core attributes of hit songs—such as structure, energy, and vocal presence—remain consistent over time. SHAP and PDP analyses consistently highlighted `instrumentalness`, `acousticness`, `valence`, `duration`, and `energy` as the most predictive features, while `key`, `mode`, and `time_signature` had negligible impact. Emotional intensity—whether upbeat or melancholic—emerged as a recurring theme in successful tracks, reinforcing the role of music as a conduit for emotional connection.

However, the findings should be interpreted in light of certain limitations. The selection of non-hit tracks may introduce label bias, particularly if they represent niche or experimental sounds. Future studies could address this by matching on genre, popularity, or release year to better isolate the "hit" effect. Additionally, the dataset spans a 60-year period, during which musical tastes and production norms evolved significantly. Incorporating temporal features or training decade-specific models could enhance the relevance and precision of future predictions.

Ultimately, while chart success depends on many external factors—marketing, timing, cultural resonance—this study demonstrates that machine learning can uncover meaningful and persistent audio patterns linked to commercial success. These insights not only deepen our understanding of what makes a song a hit, but may also offer practical guidance to artists, producers, and marketers seeking to craft music that resonates with wide audiences.

8. APPENDIX

8.1. Exploratory Analysis Figures and Tables

ACTION	FEATURES	REASON
Dropped	`track, artist, track_norm, artist_norm, target, uri, first_charted, last_charted, peak_position, weeks_on_chart`	Removed to avoid data leakage, identity bias, or redundancy.
Log-Transformed & Standardise	`duration_ms, sections, chorus_hit (→ log_duration_ms, log_sections, log_chorus_hit)`	Addressed skewness and scaled for comparability.
Standardised	`log_duration_ms, log_sections, log_chorus_hit, tempo, loudness`	Ensured consistent scaling for models sensitive to magnitude differences.
Categorical Encoding	`key, time_signature`	Treated as categorical factors due to their discrete and cyclical nature.
Binary Retained	`mode`	Already binary; kept in original format.
Retained (Original Scale)	`danceability, energy, valence, acousticness`	Preserved due to interpretability and well-behaved distributions.

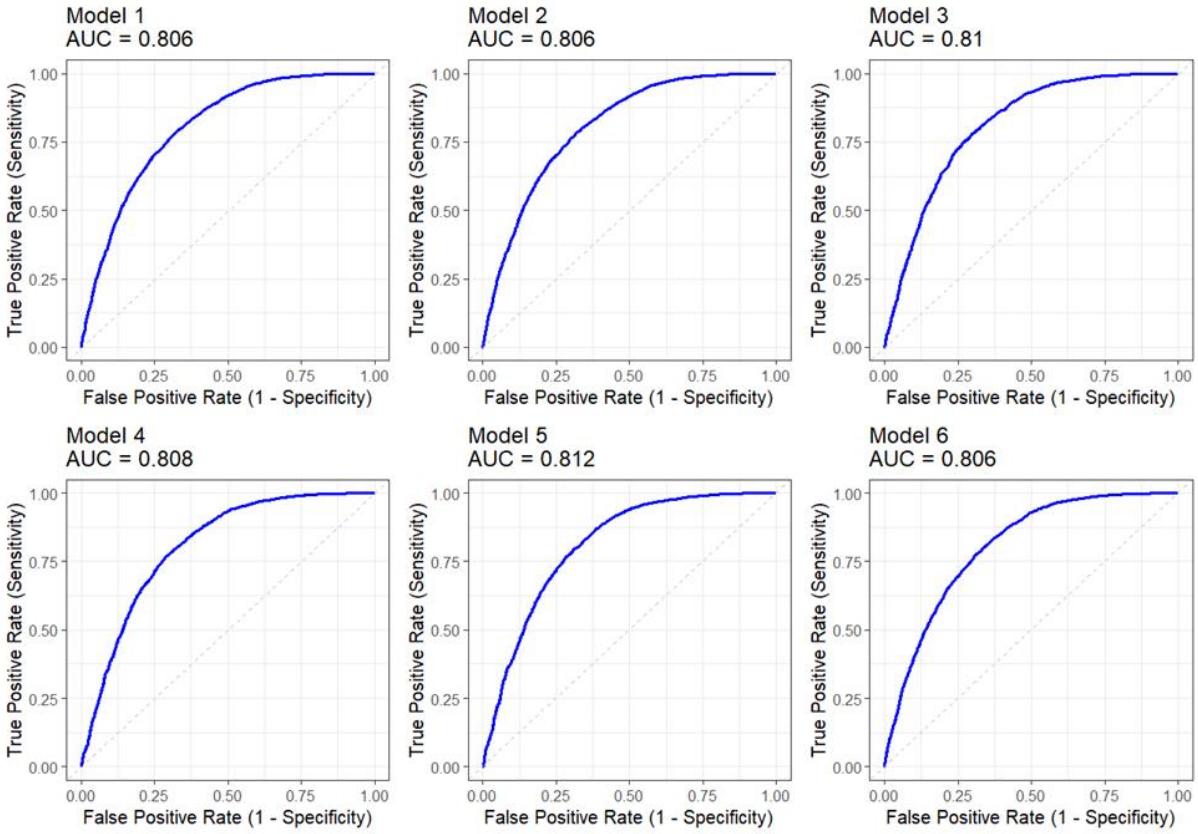
Table 8.1.1: The following preprocessing steps were applied to prepare the dataset for modelling. These transformations ensured fair scaling, improved interpretability, and reduced multicollinearity—especially important for linear classifiers like Logistic Regression and SVM.

8.2. Logistic Regression Figures and Tables

Table 8.2.1. *Evaluation Metrics for Models 1 to 6*

Model	Training Accuracy (%)	Precision	Recall	F1-Score	AUC (see plots in Graphs 2.3.1-6)
Model 1	73.18	0.71	0.79	0.75	0.806
Model 2	73.12	0.71	0.79	0.75	0.806
Model 3	73.91	0.71	0.82	0.76	0.810
Model 4	73.40	0.70	0.82	0.76	0.808
Model 5	73.91	0.71	0.81	0.76	0.812
Model 6	73.08	0.71	0.78	0.75	0.806

Figures 8.2.1-6: ***ROC Curve Plots for Models 1 to 6 (with AUC Values)***



8.2. Random Forest Figures and Tables

Table 8.2.1: ***Model parameters used to achieve the best performance, accuracy on the test dataset, and AUC in the test dataset.***

	features	parameters	accuracy	AUC
Model 1	**All**	**Ntree=200** **Nodesize=1** **Mtry=5**	**0.7888**	**0.8663**
Model 2	**Remove key and time_signature**	**Ntree=200** **Nodesize=1** **Mtry=3**	**0.7727**	**0.8507**
Model 3	**Remove key, time_signature and mode**	**Ntree=200** **Nodesize=1** **Mtry=3**	**0.7711**	**0.8503**

Table 8.2.2: ***The performance of Model 1 with different tree numbers***

Model 1	Accuracy	AUC
Ntree=200	0.7888	0.8663
Ntree=500	0.7874	0.8681

Figure 8.2.1*: Result for ntree=200(left), ntree=500(right)*

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  3011  684
       yes 1029 3388

               Accuracy : 0.7888
                 95% CI : (0.7798, 0.7977)
    No Information Rate : 0.502
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5775

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7453
            Specificity : 0.8320
         Pos Pred Value : 0.8149
         Neg Pred Value : 0.7670
             Prevalence : 0.4980
         Detection Rate : 0.3712
   Detection Prevalence : 0.4555
      Balanced Accuracy : 0.7887

       'Positive' Class : no
```

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  2991  676
       yes 1049 3396

               Accuracy : 0.7874
                 95% CI : (0.7783, 0.7962)
    No Information Rate : 0.502
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5745

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7403
            Specificity : 0.8340
         Pos Pred Value : 0.8157
         Neg Pred Value : 0.7640
             Prevalence : 0.4980
         Detection Rate : 0.3687
   Detection Prevalence : 0.4520
      Balanced Accuracy : 0.7872

       'Positive' Class : no
```

Figure 8.2.2: Result for 10 CV

```
Random Forest

32448 samples
   15 predictor
    2 classes: 'no', 'yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 29203, 29204, 29203, 29203, 29203, 29203, ...
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
   3    0.8621736  0.7211643  0.8526129
   5    0.8650672  0.7379018  0.8351298
   6    0.8641460  0.7402668  0.8323092
   7    0.8637838  0.7403275  0.8331872
   9    0.8632231  0.7409945  0.8319329
  11    0.8631477  0.7417829  0.8295519

ROC was used to select the optimal model using the largest value.
The final value used for the model was mtry = 5.
```

Figure 8.2.3: Plot for ROC under different mtry for Model 1

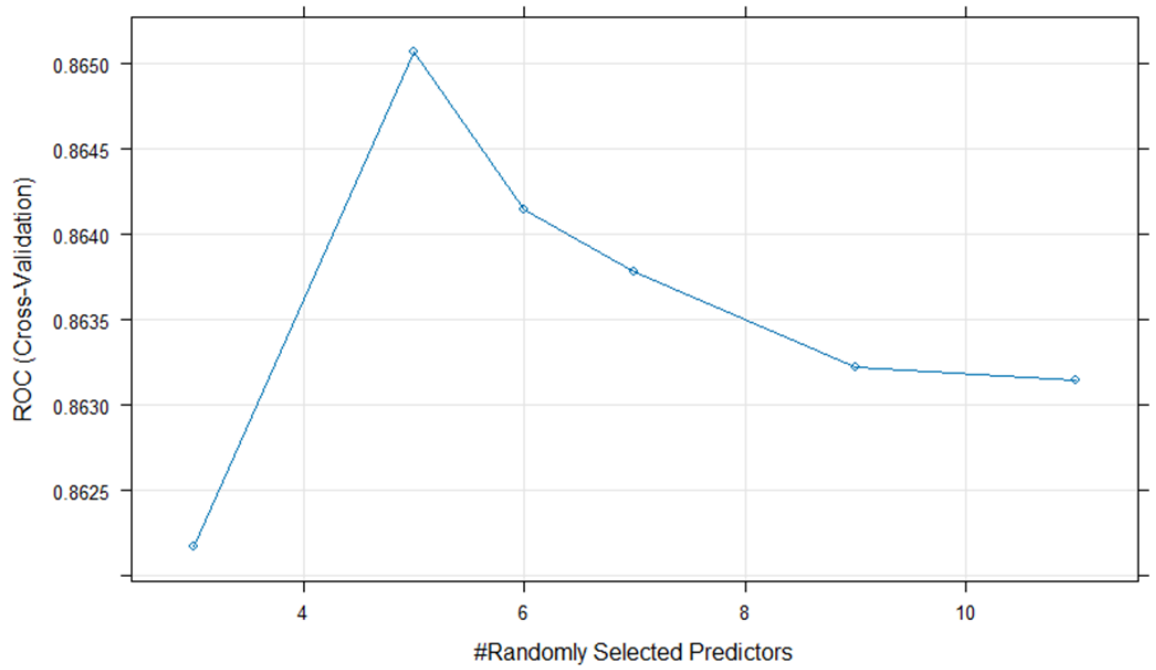


Figure 8.2.4: confusionMatrix and Accuracy for Model 1 when mtry=5

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  3011  684
       yes 1029 3388

               Accuracy : 0.7888
                 95% CI : (0.7798, 0.7977)
    No Information Rate : 0.502
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5775

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7453
            Specificity : 0.8320
         Pos Pred Value : 0.8149
         Neg Pred Value : 0.7670
             Prevalence : 0.4980
         Detection Rate : 0.3712
   Detection Prevalence : 0.4555
      Balanced Accuracy : 0.7887

       'Positive' Class : no
```

Figure 8.2.5: The importance of features based on result of Model 1

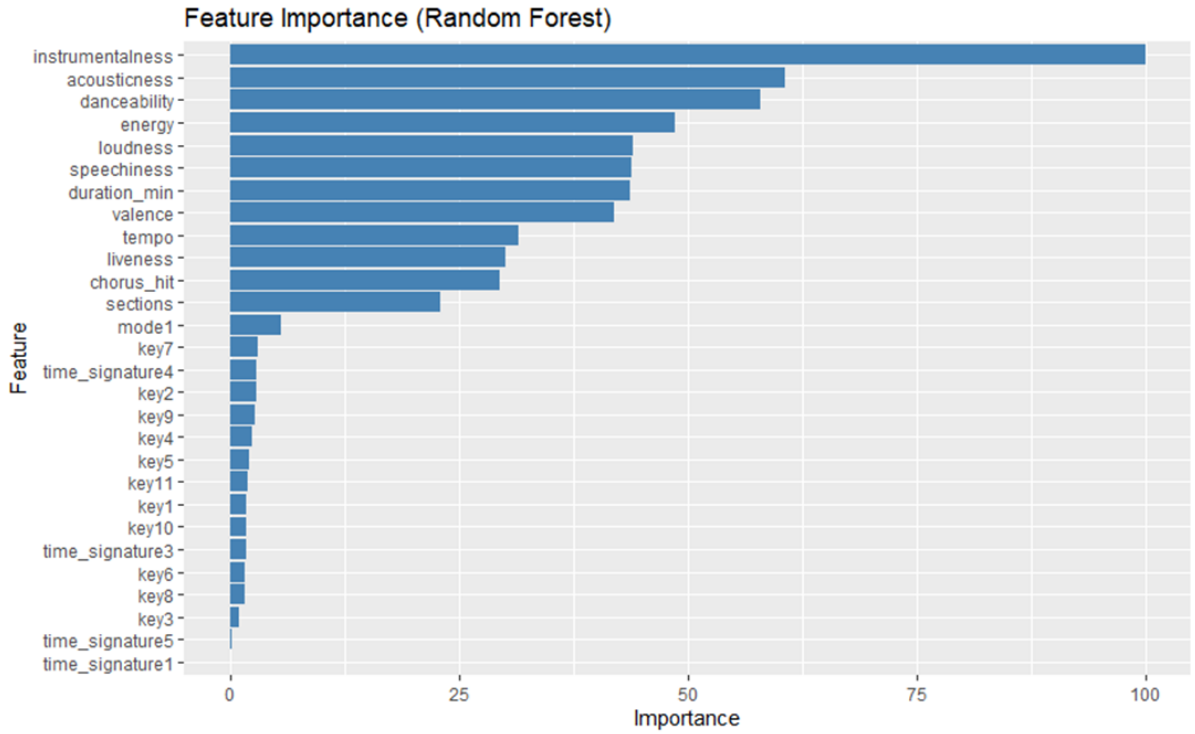


Figure 8.2.6: 10 CV for Model 2

```
> print(sd_tuned)
Random Forest

32448 samples
   12 predictor
    2 classes: 'no', 'yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 29203, 29204, 29203, 29203, 29203, 29203, ...
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
   3    0.8544633  0.7203760  0.8313693
   5    0.8529501  0.7195876  0.8286120
   6    0.8530941  0.7232868  0.8286750
   7    0.8518063  0.7215282  0.8279854
   9    0.8515194  0.7240752  0.8264180
  11    0.8515482  0.7237720  0.8246639

ROC was used to select the optimal model using the largest value.
The final value used for the model was mtry = 3.
```

Figure 8.2.7: Plot for ROC under different mtry for Model 2

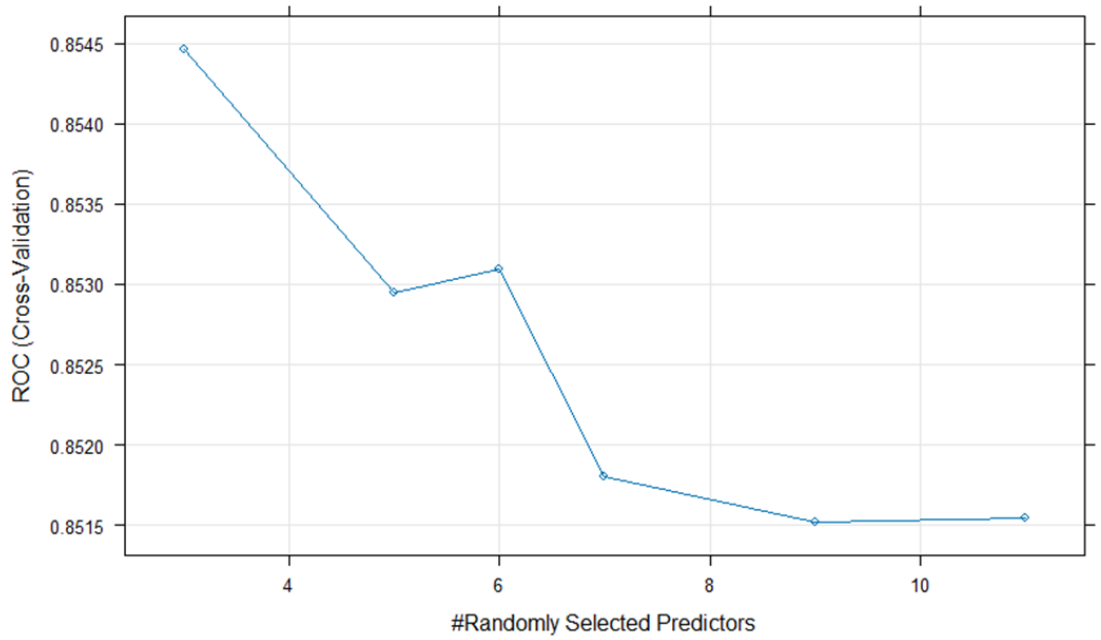


Figure 8.2.8: confusionMatrix for Model 2 when mtry=3

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  2892  696
       yes 1148 3376

               Accuracy : 0.7727
                 95% CI : (0.7634, 0.7818)
    No Information Rate : 0.502
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5452

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7158
            Specificity : 0.8291
         Pos Pred Value : 0.8060
         Neg Pred Value : 0.7462
             Prevalence : 0.4980
         Detection Rate : 0.3565
   Detection Prevalence : 0.4423
      Balanced Accuracy : 0.7725

       'Positive' Class : no
```

Figure 8.2.9: The importance of features based on result of Model 2

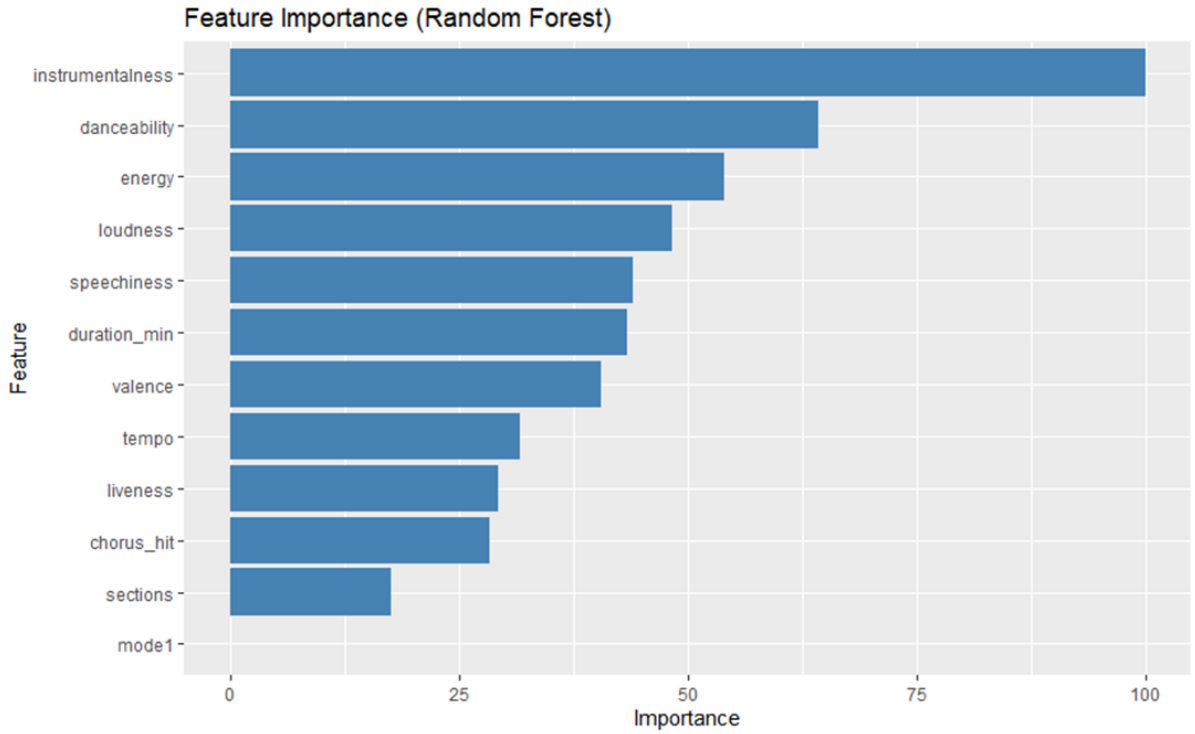


Figure 8.2.10: 10 CV on the Model 3

```
Random Forest

32448 samples
   11 predictor
    2 classes: 'no', 'yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 29203, 29204, 29203, 29203, 29203, 29203, ...
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
   3    0.8519221  0.7174045  0.8318707
   5    0.8509228  0.7206792  0.8293016
   6    0.8507859  0.7189206  0.8283610
   7    0.8504261  0.7199515  0.8278605
   9    0.8499606  0.7200121  0.8266068
  11    0.8487758  0.7199515  0.8242883

ROC was used to select the optimal model using the largest value.
The final value used for the model was mtry = 3.
```

Figure 8.2.11: Plot for ROC under different mtry for Model 3

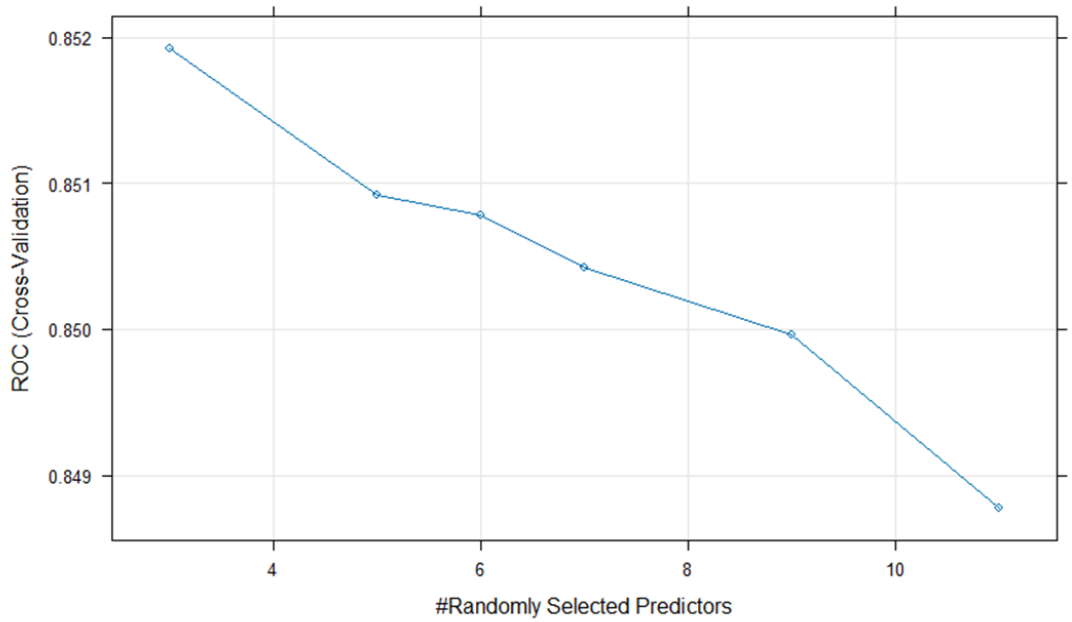
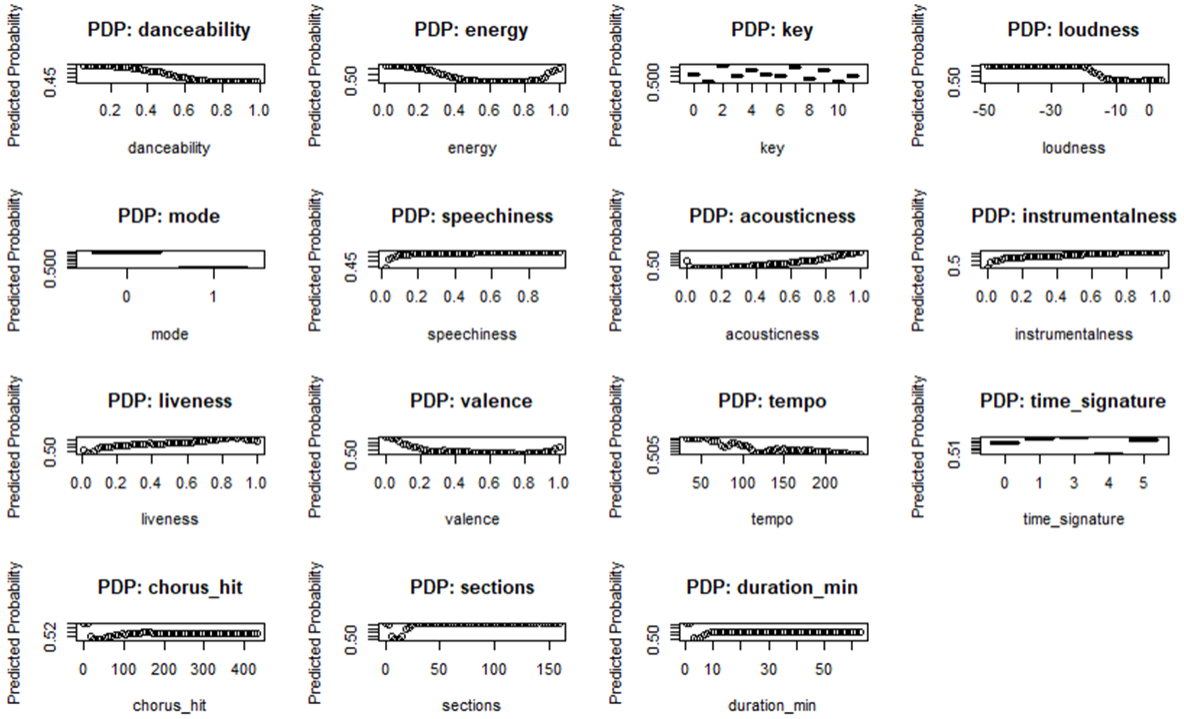


Figure 8.2.12: confusionMatrix for Model 3 when mtry=3

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  2870  687
       yes 1170 3385

               Accuracy : 0.7711
                 95% CI : (0.7618, 0.7802)
    No Information Rate : 0.502
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5419

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7104
            Specificity : 0.8313
         Pos Pred Value : 0.8069
         Neg Pred Value : 0.7431
             Prevalence : 0.4980
         Detection Rate : 0.3538
   Detection Prevalence : 0.4385
      Balanced Accuracy : 0.7708

       'Positive' Class : no
```

Figure 8.2.13: PDPs plot for Model 1 when ntree =200



8.4. XGBoost Figures and Tables

Table 8.4.1: Model performance comparison table, highlighting ideal values for each evaluation metric

Model Performance Comparison:

	Model	Train Accuracy	Test Accuracy	Accuracy Gap	ROC AUC	Precision	Recall	F1 Score
5	Regularised Model	86.6%	78.8%	7.7%	86.8%	75.3%	84.1%	79.4%
2	ROC Tuned Model 2	91.3%	78.6%	12.6%	86.8%	75.1%	83.8%	79.3%
1	ROC Tuned Model	89.6%	78.5%	11.1%	86.4%	75.0%	83.6%	79.1%
4	Accuracy Tuned Model	92.5%	78.4%	14.1%	86.7%	75.0%	83.3%	79.0%
7	Custom Threshold Model	–	78.3%	–	86.0%	73.5%	86.7%	79.5%
8	Calibrated + Custom Threshold	–	78.2%	–	86.2%	75.9%	80.7%	78.2%
0	Baseline Model	81.7%	78.1%	3.6%	86.2%	74.0%	84.8%	79.0%
3	Log-Transformed Model	81.7%	78.1%	3.6%	86.2%	74.0%	84.8%	79.0%
6	Reduced Feature Model	82.9%	78.1%	4.8%	86.0%	74.4%	84.0%	78.9%
11	Truncated SVD Model	–	76.1%	–	84.2%	71.6%	84.3%	77.5%
9	PCA XGBoost	–	75.0%	–	81.9%	69.8%	85.5%	76.9%
10	Kernel PCA Model	–	74.4%	–	82.1%	69.8%	83.7%	76.1%

Figure 8.4.2: ROC Curve for Baseline Model

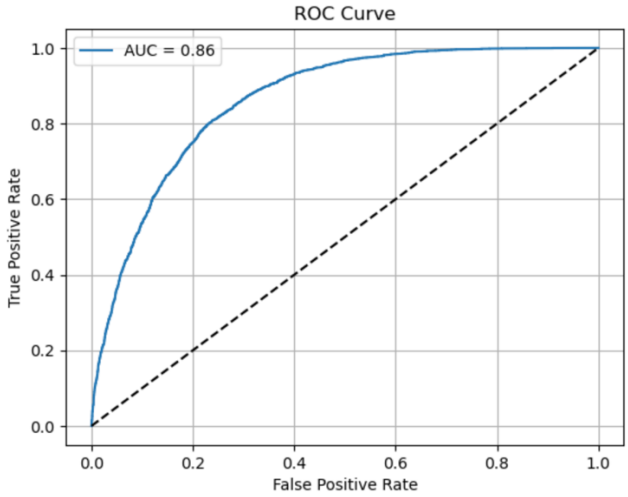


Figure 8.4.3: Precision-Recall vs Threshold plot to fine-tune the threshold

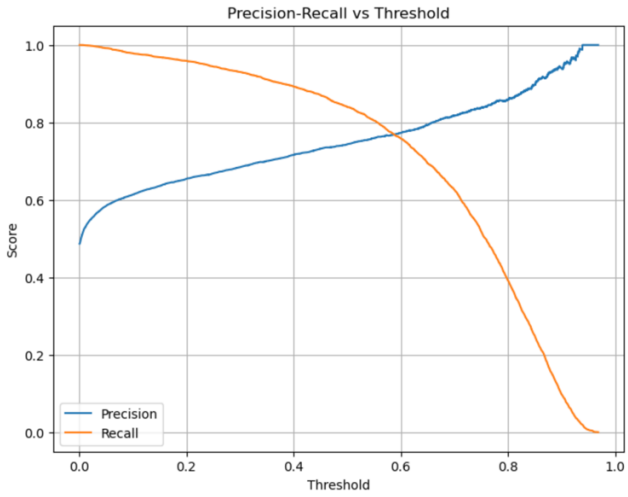


Figure 8.4.4: Calibration curve based on CV=10

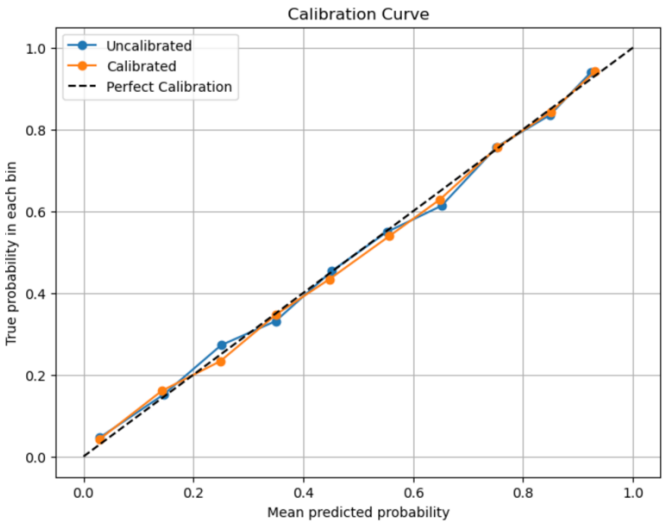


Figure 8.4.5: Feature importance performed on reduced feature-set

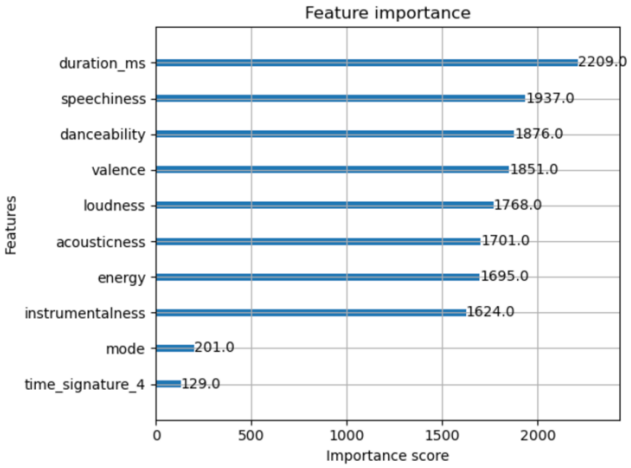


Figure 8.4.6: Results of permutation importance conducted on validation set

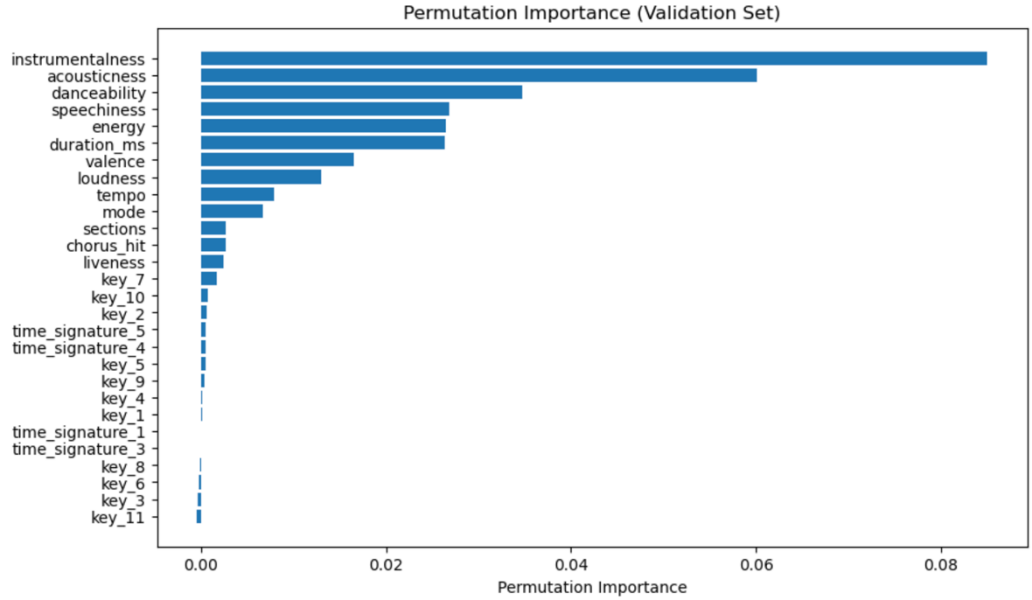


Figure 8.4.7: REFC plot to assess the benefit of number of features vs. accuracy

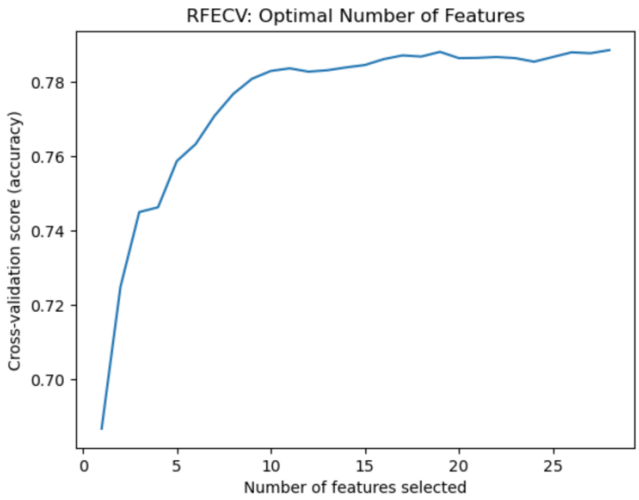


Figure 8.4.8: SHAP value plot for reduced feature-set

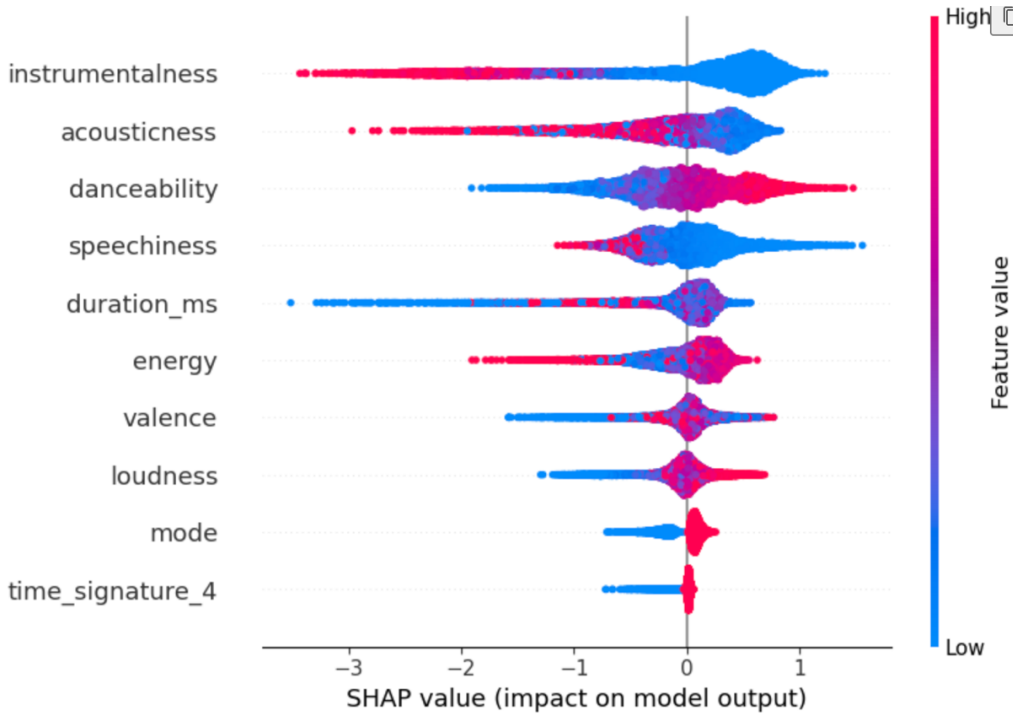


Figure 8.4.9: SHAP feature interaction heatmap

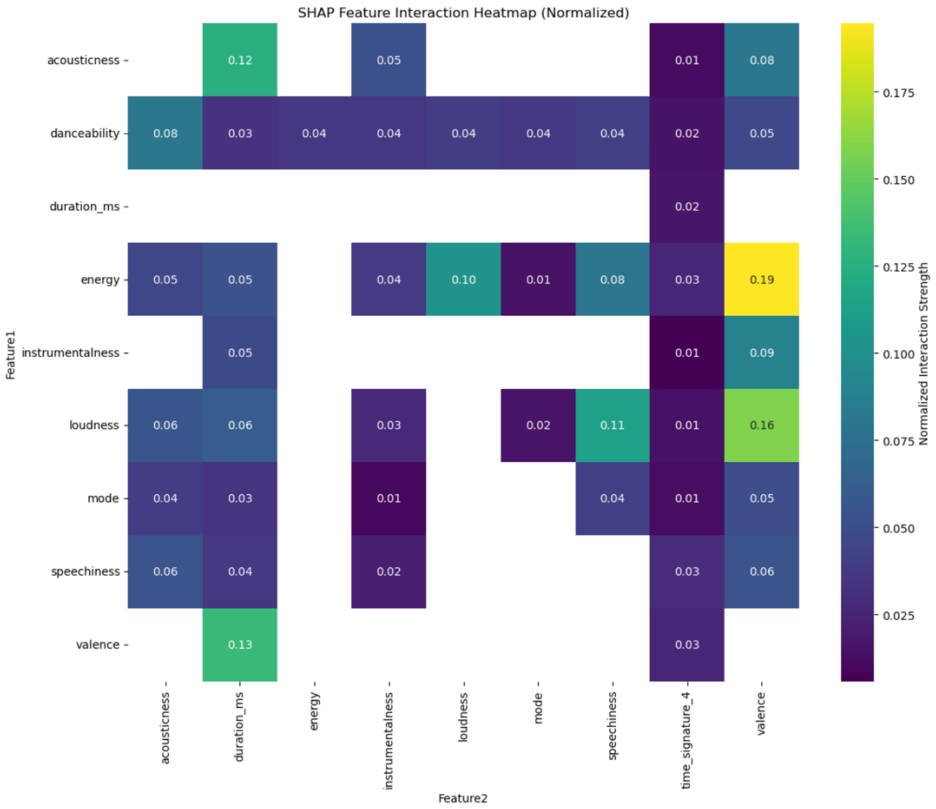
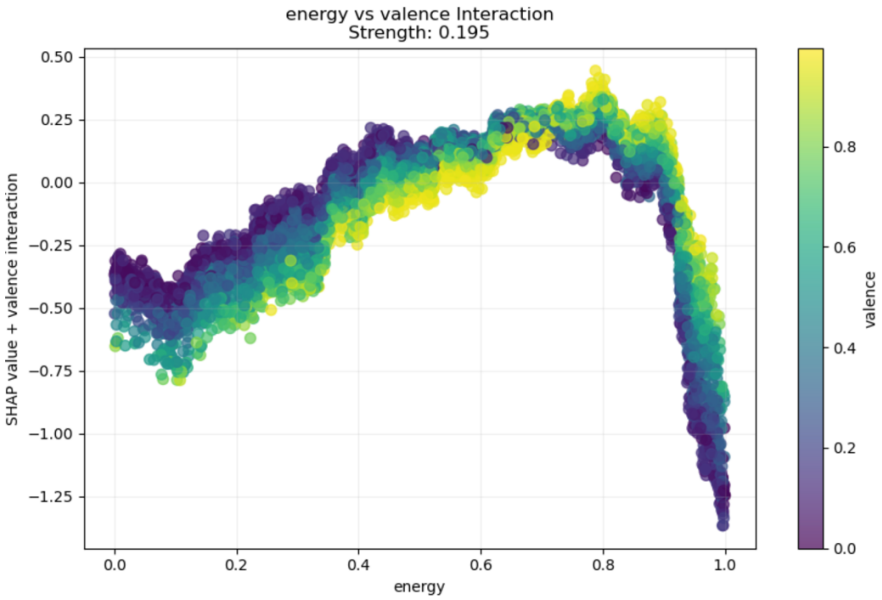


Figure 8.4.10: SHAP value interaction plot showing energy x valence



8.4. SVM Figures and Tables

Table 8.4.1: ***Confusion Matrix for Model 1***

	precision	recall	f1-score	support
0	0.83	0.67	0.74	6169
1	0.72	0.86	0.78	5999
accuracy			0.76	12168
macro avg	0.77	0.76	0.76	12168
weighted avg	0.77	0.76	0.76	12168

Figure 8.4.1*: Threshold vs Performance Metrics Analysis*

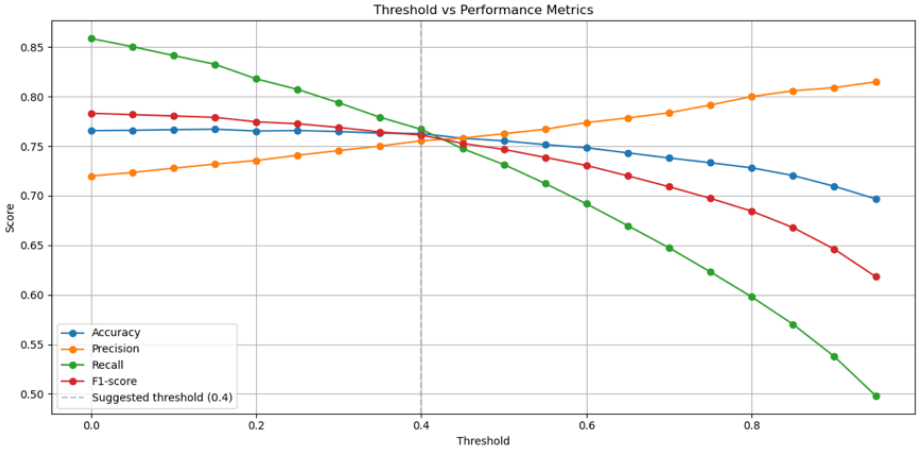


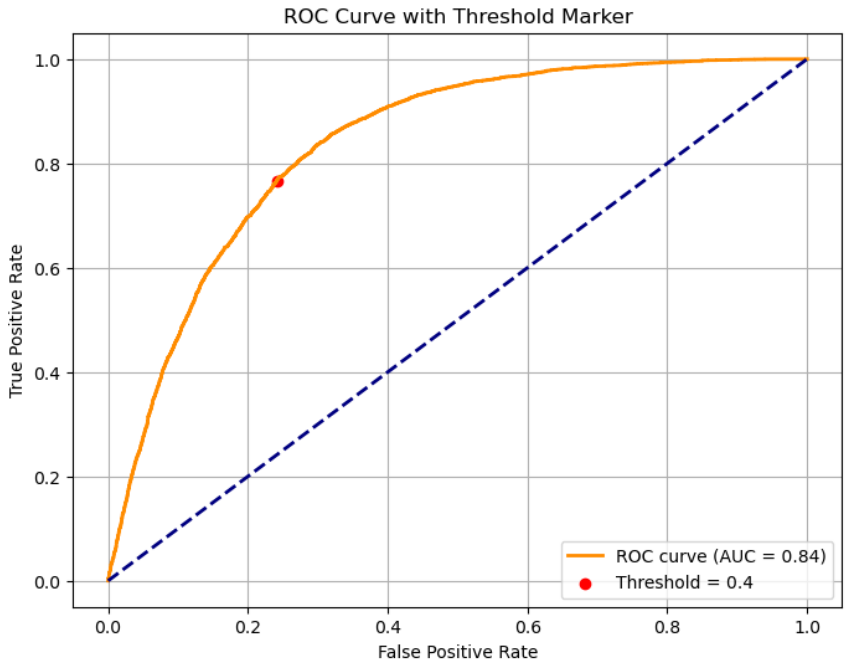
Table 8.4.2*: Confusion Matrix for Model 2 (Threshold is default)*

	precision	recall	f1-score	support
0	0.83	0.68	0.74	6169
1	0.72	0.86	0.78	5999
accuracy			0.77	12168
macro avg	0.78	0.77	0.76	12168
weighted avg	0.78	0.77	0.76	12168

Table 8.4.3*: Confusion Matrix for Model 2 (Threshold is 0.40*

	precision	recall	f1-score	support
0	0.77	0.76	0.76	6169
1	0.75	0.77	0.76	5999
accuracy			0.76	12168
macro avg	0.76	0.76	0.76	12168
weighted avg	0.76	0.76	0.76	12168

Figure 8.4.2: *ROC Curve with Threshold Marker*



8.5. Table 6.5.1: Variables Included in Each Method

	Logistic Regression	Random Forest	SVM	XGBoost
danceability	✅	✅	✅	✅
energy	✅	✅	✅	✅
key	✅	✅	✅	✅
loudness	✅	✅		✅
mode	✅	✅	✅	✅
speechiness	✅	✅	✅	✅
acousticness	✅	✅	✅	✅
instrumentalness	✅	✅	✅	✅
liveness	✅	✅	✅	✅
valence	✅	✅	✅	✅
tempo	✅	✅		✅
duration_ms	✅	✅	✅	✅
time_signature	✅	✅	✅	✅
chorus_hit	✅	✅	✅	✅
sections	✅	✅		✅

8.6. Table 8.6.1: Metrics for Each Method's Best Model

	Logistic Regression	Random Forest	SVM	XGBoost
AUC	0.812	0.8663	0.84	0.862
Accuracy	0.7391	0.7888	0.76	0.781