Predicting a song is a hit or not hit using its audio features with Naive Base and Logistic Regression.

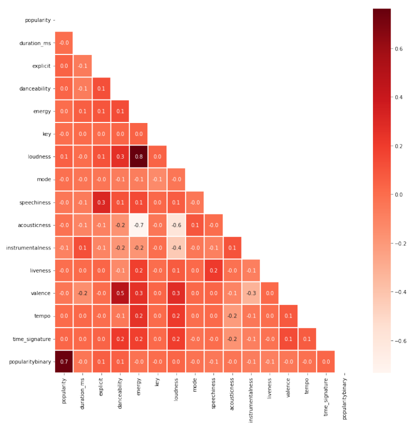
Ratna Pathak, City university of London.

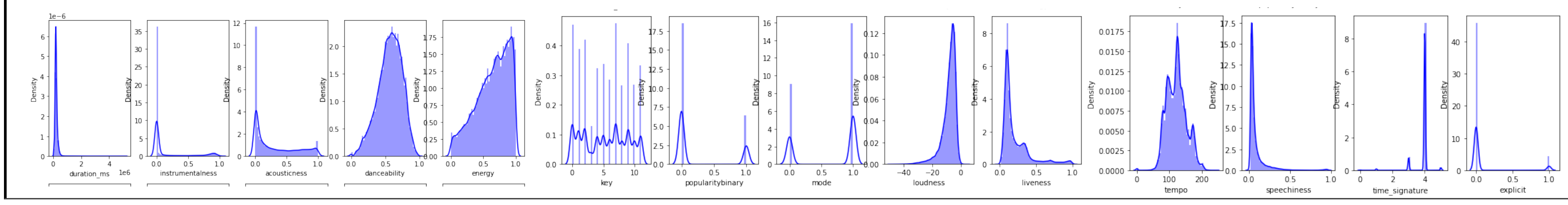
Description and motivation :

Being able to predict whether a song will be popular beforehand is an important research subject for the music industry. The main aim of this research is to solve this binary classification problem of predicting whether a song is a hit or not hit based on the audio features available for a song from Spotify using two Machine learning models: Naive Bayes and Logistic Regression. The performance of these two models is compared using performance metrics.

Exploratory Data Analysis :

- Dataset: Spotify data set consists of songs with different audio features available.
- The data was cleaned dropping the columns other than the audio features like track_id, artists, etc.
- While trying to find a pattern in the data, we observe the distribution of the data, all the features were plotted.
- The popularity of a song is given in percentage, to make it our binary predictor variable binary a threshold of 50 is taken in a way that if a song has popularity greater than 50 it will be considered a hit, else, not a hit. It was stored in the popularity binary column.
- A correlation heatmap was generated showing a 2-dimensional correlation matrix, using coloured cells to represent data from a monochromatic scale to find the correlation between the features so as to identify the feature having high correlation and can be reduced.
- Since our data have a variable scale, i.e, it does not follow normal distributions have normalized it.
- Since our predictor value is imbalanced, i.e. there are a lot of not-hit songs in comparison to hit-song, to make it even synthetic minority oversampling technique is applied to the data, now we have 1: 84633, 0: 84633, i.e same number of hit and non hit songs.
- The distribution of song features like danceability, energy, loudness and tempo is quite high. People like fast and loud music.
- The correlation between 'loudness' and 'energy' is 0.8 which is strong.





Logistic Regression

- Logistic Regression model is a supervised learning algorithm for classification. This model is therefore used for problems where the outcome can be classified into one of two categories. Here it is 1 for the hit song and 0 for the non-hit song.
- The predictor: popularity then deduce by setting a threshold value (50) is set.
- The logistic regression model can be extended to accommodate multiclass classification.
- The logistic regression provides a rule for classifying the test data with a cut-off on the predicted success probability.

Pros:

+ Logistic regression is simple to apply, train and understand.
+ Needs lesser time to train.
+ Easy to extend to a multi-class problem.
+ No or little hyperparameter optimization is required.

Cons:

-Possible underfitting of complex data-set.
- Sensitive to outliers.
-May not be accurate if the sample size is too small.

Naïve Bayes

- Naive Bayes algorithm is called 'naive' as it assumes that given any target, the features are independent. In our example, we have two classes (Hit song and not-hit song)
- Usually, Naive Bayes model performs surprisingly well despite the assumption of independence.
- It assumes prior probabilities from class distribution in the training data set, multinomial distribution for discrete features and gaussian distribution for continuous features but these can be adjusted to find the optimal model.
- A decision rule is used to pick the label, i.e, maximum a posteriori, it selects the most probable hypothesis.

Pros:

+ Easy to implement.
+ If the independence assumptions are true, it works better than most algorithms.
+ The probabilities needed to build the model are found in one scan, thus training is linear.
+ Robust to the noise in the dataset.

Cons:

- It's very rare that all the predictors will be independent in real life.
- If one attribute in test set is not present in the training set, NB assigns 0 probability.
-Difficult to work with continuous features.

Hypothesis Statement:

- Both models are expected to perform better than a random guess when predicting a song is hit or not. NB may not perform better than LR because of the assumption of independence of features.
- Naive Base is expected to have a lower train time.
- Although there are many external factors contributing to the popularity of the song, for our research, we are focusing on audio features only.

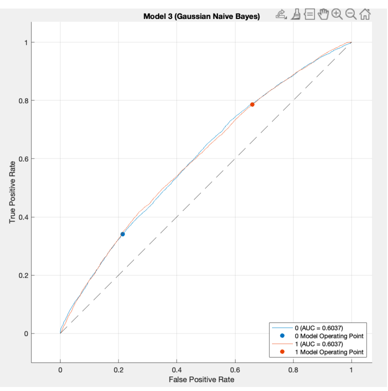
Experimental results, parameter choices and feature selection:

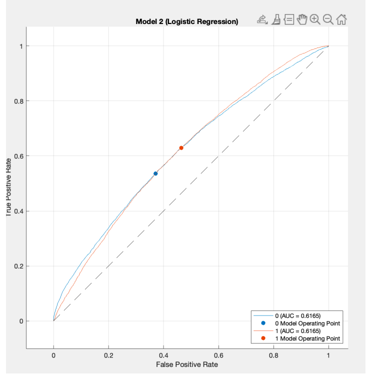
Logistic Regression:

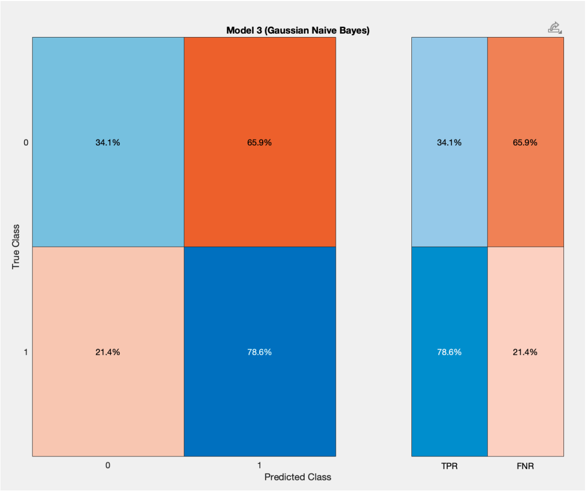
- The model was too complex as it had too many features.
- It was unable to predict hit songs initially so correcting the target class imbalance by under-sampling improved average recall and precision per fold but decreased average accuracy and AUC per fold.

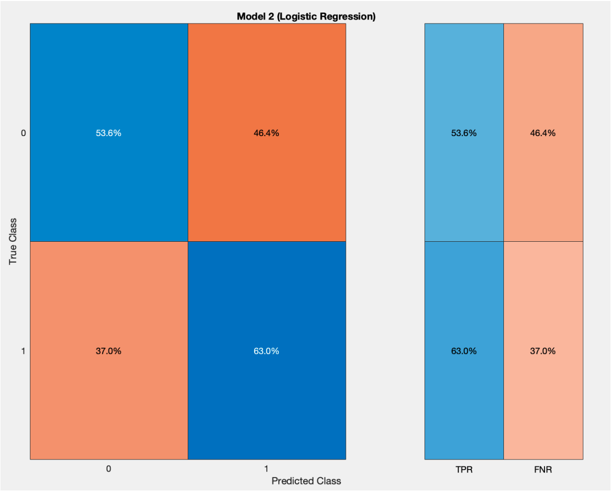
Naive Bayes

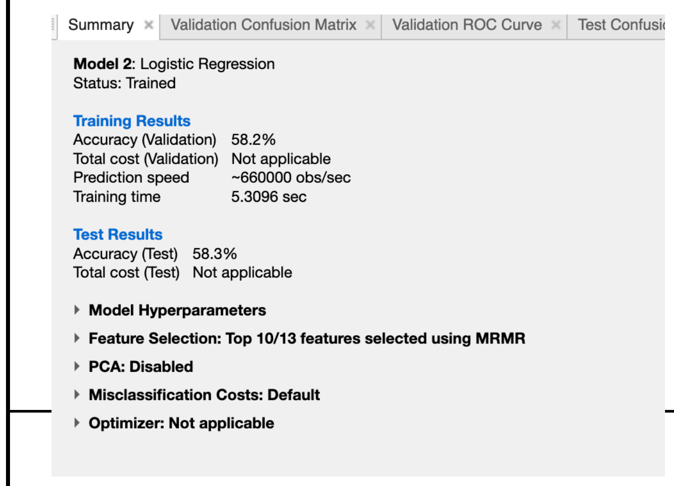
- Balancing target classes by under-sampling changed prior probabilities and increased recall and precision.
- The best model was identified as the model which used feature selection, and under-sampling to balance target classes and normalization.
- It performed better in predicting hit songs.









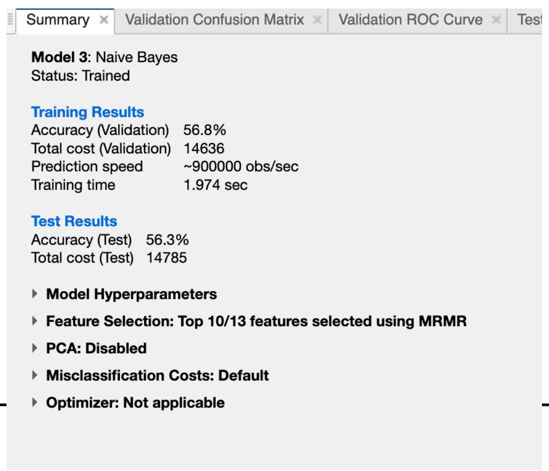


Model 2: Logistic Regression
Status: Trained

Training Results
Accuracy (Validation) 58.2%
Total cost (Validation) Not applicable
Prediction speed ~660000 obs/sec
Training time 5.3096 sec

Test Results
Accuracy (Test) 58.3%
Total cost (Test) Not applicable

- **Model Hyperparameters**
- **Feature Selection: Top 10/13 features selected using MRMR**
- **PCA: Disabled**
- **Misclassification Costs: Default**
- **Optimizer: Not applicable**



Model 3: Naive Bayes
Status: Trained

Training Results
Accuracy (Validation) 56.8%
Total cost (Validation) 14636
Prediction speed ~900000 obs/sec
Training time 1.974 sec

Test Results
Accuracy (Test) 56.3%
Total cost (Test) 14785

- **Model Hyperparameters**
- **Feature Selection: Top 10/13 features selected using MRMR**
- **PCA: Disabled**
- **Misclassification Costs: Default**
- **Optimizer: Not applicable**

Methodology:

1. Data is split into a 80: 20 ratio i.e, 80% data for training and 20% data for testing set.
2. The test data remains unseen to models until its ready for testing.
3. Use of the Hold-out method to partition the data so that the training set after modelling can be cross-validated.
4. Try to improve the models using parameter optimization, wherever possible.
5. Optimise models are done by feature selection and correcting target class imbalance.
6. Evaluate which is optimal model based on performance metrics.
7. Measure and compare the predictive performance of the optimized models. By recording train and test times.

Analysis and Evaluation of results:

- The initial application of LR on the data set without removing the imbalance provided the accuracy test as 93.3% which is quite high, and 68.5 % for naive base. Logistic Regression predicted the non-hit song greatly but due to the imbalance,i.e, presence of a lesser example of hit songs, the prediction of hit songs was close to null.
- The application of the Synthetic Minority Oversampling Technique(SMOT) and feature selection (MRMR) improved the prediction of the hit-song class greatly but affected the overall performance of both models reducing it to 58.3% in the case of Logistic Regression model and 56.8% in the case of Naive Base model. As predicting hit song was an important parameter, the second improved model was kept for analysis.
- The final performance of both model is comparable showing not much difference with the accuracy test lying in the range of 50s.
- Comparing the ROC of both the model gives similar result , Logistic Regression performing slightly better.
- The training and test time for both the models were as expected, where Naive Base took lesse time, but the accuracy of validation was more for Logistic Regression.(as shown in the table)
- Comparing the Confusion matrix of test of both the model, Naive base performed better in predicting the hit-song class, which is an important facter to be considered.(as shown in the figure)

Lessons Learned :

- Optimizing for both models involves feature engineering and feature selection.
- Normalization may not always improve model performance
- Prediction class imbalance leads to inefficient model performance in terms of precision and recall.

Future Work:

- Outliers should be detected for all the features before proceeding to model.
- Explore the effects of using variants of the NB classifier.
- Deciding the threshold value for the prediction can be done more on the informed analysis of the trends of populer songs in place of applying basic threshold.

References:

1.Pachet, F. & Roy, P. (2008), 'Hit Song Science Is Not Yet a Science.' and Pachet, F. (2012), 'Hit song science';

2.Pham, J., Kyauk, E. & Park, E. (2016), 'Predicting song popularity'.

3.https://uk.mathworks.com/help/stats/classificationlearner-app.html.

4.Christopher M Bishop, "Pattern Recognition and Machine Learning", chapter 4.

5.James Pham, Edric Kyauk and Edwin Park: "Predicting Song Popularity "