Feature Engineering vs Feature Selection vs Hyperparameter Optimization in the Spotify Song Popularity Dataset

Alan Cueva Mora (d20125565@mytudublin.ie)
Brendan Tierney (brendan.tierney@tudublin.ie)
School of Computer Science

The Presenter

Alan Cueva Mora

Software developer with more than ten years of experience; Former Core DataBase Developer in Oracle and an active member of the MariaDB Community. Master's degree student in the Technological University Dublin.

- Data Science / Data Analytics
- Computer Science

Abstract

Research in Featuring Engineering has been part of the data pre-processing phase of machine learning projects for many years, but newer sometime forget its importance.

It can be challenging for new data scientist to understand its importance along with various approaches to find an optimized model. This work uses the Spotify Song Popularity dataset to compare and evaluate different techniques:

- Feature Engineering
- Feature Selection
- Hyperparameter Optimization.

Feature Engineering has a greater effect on model efficiency when compared to the alternative approaches.

Introduction

Feature Selection and **Hyperparameter Optimization** are two sophisticated machine learning techniques with a strong research background. For an early-stage data scientist, it is very easy to think that they are the best alternative in a machine learning task.

Feature Engineering is an important but labour-intensive take of machine learning. Most machine learning performance is heavily dependent on the representation of the feature vector. As a result, much of the actual effort in deploying machine learning algorithms goes into the design of pre-processing pipelines, data transformations, domain and metadata knowledge.

Kaggle competitions and ACM's KDD Cup have seen feature engineering play a very important part in several winning submissions. Additionally, the Kaggle Algorithmic Trading Challenge was won with an ensemble of models and feature engineering. The features engineered for these competitions were created manually by the data scientist, utilizing their domain knowledge.

What approach is better?

Relate Research

There are some common research topics in machine learning literature, one of them is about comparing different machine learning methods to solve specific tasks and identify the better scenarios for a method.

Another common machine learning research topic is to compare similar techniques, as is the case of feature selection and feature extraction. The objective of both methods is to reduce the feature space to improve data analysis. Feature selection performs the reduction by selecting a subset of features without transforming them, while feature extraction reduces dimensionality by computing a transformation of the original features to create other features that should be more significant.

In featuring engineering research, it is common to find comparisons between combinations of different engineered features and methods to identify which methods generally benefit from the same set of engineered features.

Considering that in every machine learning task the objective is to reduce the error, there is no reason not to compare completely different techniques, such as those proposed in this work.

Feature Engineering

Feature Engineering involves calculating new features, based on other features, and it is primarily a manual, time consuming task.

The Spotify Song Popularity dataset consists of 129172 rows and 17 independent variables of which three are strings and cannot be used in the machine learning task. For this type of data, feature engineering focuses on generating numerical variables from these.

Every new variable was evaluated using spearman correlation, follow this criteria:

- Very weak (0.00 - 0.19)
- Weak (0.20 - 0.39)
- Moderate (0.40 - 0.59)
- Strong (0.60 - 0.79)
- Very Strong (0.80 - 1.00)

All statistics generated for this work were statistically significant (p-value < 0.05).

Feature Engineering: New Features

The new features were generated from string features in the dataset (release date, artists and name):

- Numerical Release Date
  - year_rd = YEAR(release_date) + (DAY_OF_YEAR(release_date) / 365)
- Number of Artists
  - artists_n = LEN(EVAL(artists))
- Mean Artists Popularity
  - Dictionary of Mean Artist Popularity
  - Mean of MAP of artists of the song with an imputation on unknown artists.
- Name Length
  - name_len = LEN(name)
- Name Language
  - Language detection from the name of the song and a encoder

NOTE: Only the Numerical Release Date wasn't taken into consideration for its similarity with year.

Feature Engineering: Language Detection

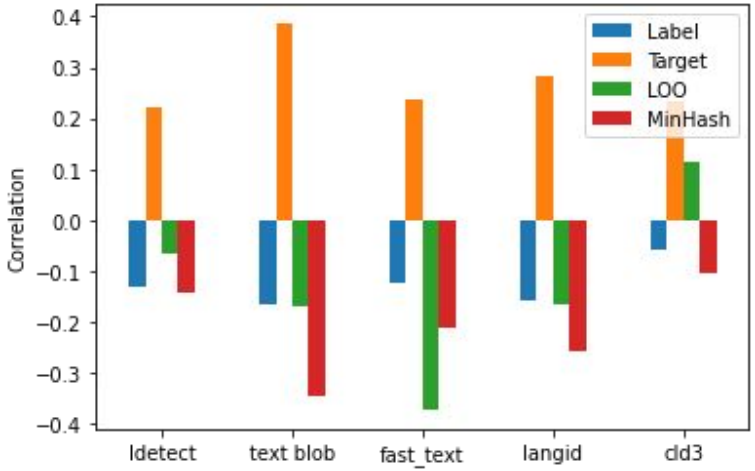
Five different language detect libraries were taken into consideration:

- LangDetect (46 languages detected)
- TextBlob (88 languages detected)
- FastText (120 languages detected)
- LangId (79 languages detected)
- CLD3 (97 languages detected)

This high cardinality variables were tested using different encoders:

- Label Encoder
- Target Encoder
- Leave One Out Encoder
- Min Hash Function

The combination of TextBlob and the Target Encoder produce the best performance.

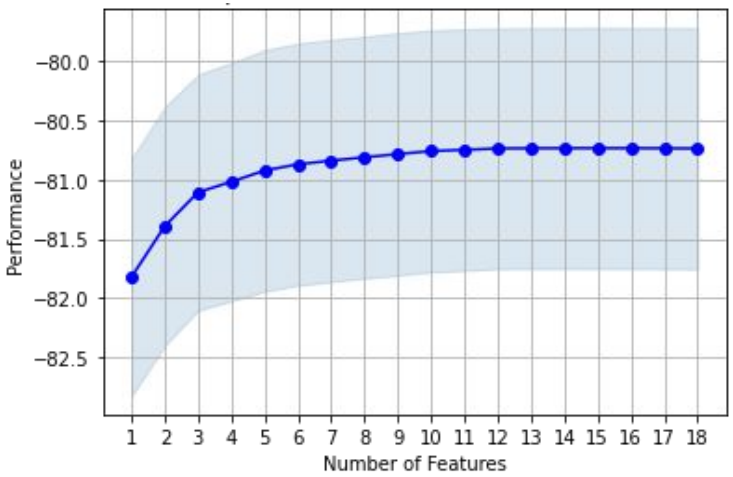


Libraries and Encoders Correlation Comparison

Feature Selection

In machine learning, feature selection entails selecting a subset of the available features in a dataset to use for model development. Among its advantages are generating better models and reducing computations cost. The techniques considered in this section are Least Absolute Shrinkage and Selection Operator (LASSO) and Sequential Forward Selection (SFS).

First, an SFS task was executed using all features to detect any negative performance contribution to the model.



Sequential Forward Selection using all features

Feature Selection: Continuation

- In order to get the number of features to consider based on evidence a LASSO task was run using different regularization parameter values. The recommended values are 0.1, 0.01 and 0.001 and the one that includes more variables (C=0.01) reduces the number of features from 18 to 14, excluding danceability, energy, liveness and speechiness.
- Considering the number of features proposed by LASSO, an SFS task was run using the same number, and it results with a different subset of features. The SFS task excludes variables mode, key, explicit and danceability. Only danceability was excluded by both processes.

Feature Selection: Conclusion

In order to choose the best subset, three regression tasks were run using CV and measuring the RMSE statistic.

- All non-string features 8.9846
- SFS subset 8.9845
- LASSO subset 8.9984

The SFS subset was best one, but the improvement is almost insignificant, so the advantage is to reduce the computational cost and omit features that do not contribute to the performance. It is important to mention that no new features were excluded by any methods.

Hyperparameter Optimization

Hyperparameter optimization consists of testing a set of hyperparameters of a model and identifying the optimal values for them. In this section, five methods were taken into consideration. They can be grouped in two parts: linear (1 and 2) and tree methods (3, 4 and 5).

1. Linear Regression (LR)
2. Ridge Regression (RR)
3. Decision Tree Regressor (DTR)
4. Extra Tree Regressor (ETR)
5. Random Forest Regressor (RFR)

The hyperparameter optimization task was performed using a CV grid search (K=3). Unfortunately, there are no hyperparameters for the Linear Regression, for the Ridge Regression there is one, the regularization strength, but this only improve the RMSE by 0.000004, so this step focused on the tree methods.

Hyperparameter Optimization: Continuation

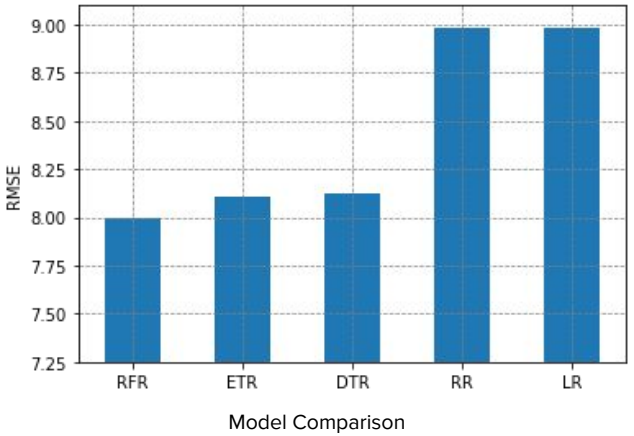
In the tree methods, one parameter directly influences the results. The max depth parameter which specifies how many levels of nodes the tree could have. When this parameter is set to None, the tree will expand the nodes until all leaves are pure or until all leaves contain less than two samples.

Limiting the tree was clearly a good option, not only because the train for the entire tree takes too much time, but the results are better. After an exhaustive evaluation the best values of max depth were 11 for DTR and 15 for ETR and RFR. Another parameter was the criterion which measures the quality of a split where the only options that worked were mean square error and mean squared error with Friedman's improvement score for potential splits, but the evaluation proves that this parameter doesn't affect the metrics.

In the specific case of DTR and ETR there was an option to add Bagging Regression (BR). The BR is an optimization to improve the stability and accuracy of the method, the Bagging split data and use it in different decision trees and ensemble the result. In both cases the BR was the best option.

In order to compare the RMSE metrics with the previous section of this work, five CV tasks (K=5) were run using the best parameters for each method. Figure 3 shows that the RFR model gets the best metrics.

The result of the hyperparameter optimization and the model comparison was a RMSE reduction from 8.98 to 7.99. Although the grid search task is automatic, it takes a lot of execution time which requires monitored because it can easily crash when computer resources are depleted.



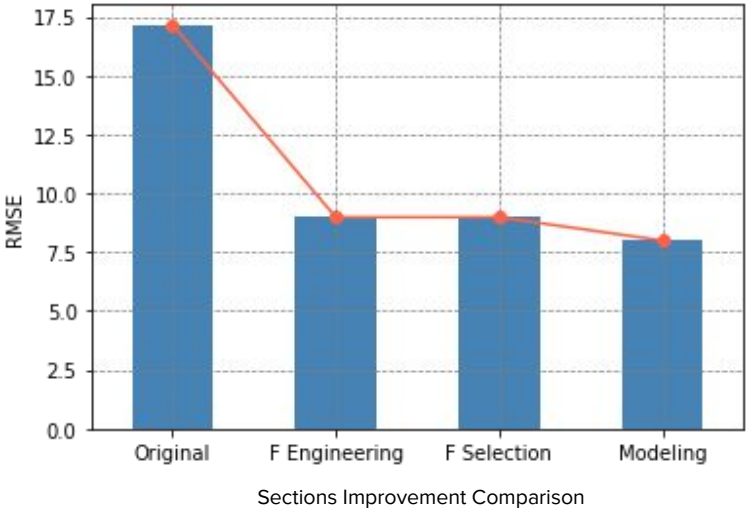
Model Comparison

Conclusion

The analysis performed over the Spotify Song Popularity dataset involves feature engineering, feature selection, hyperparameter optimization and model selection using CV to validate each step.

Feature engineering was by far the technique that generated the best reduction of the RMSE metric. Figure 4 shows how this technique reduced the error to almost half while the improvements produced by Feature Selection and Hyperparameter Optimization/Model Selection was almost not significant.

It can be concluded for this dataset, a well performed feature engineering task has a greater impact on the model performance than more sophisticated machine learning techniques, even when each step takes approximately the same time and resources its value is not the same.



Sections Improvement Comparison

Future Work

This experiment focused on using one particular dataset. Future work will look to expand to include more datasets from a variety of domains. This will be done to evaluate the effect of these tasks and to see if similar outcomes can be achieved.

Thanks!