North American Journal of Engineering and Research

Est. 2020

Enhancing ML Model Performance through Feature Engineering and Model Selection

Pushkar Mehendale
Troy, MI, USA
pushkar.mehendale@yahoo.com

Abstract :

Feature engineering and model selection are crucial steps in the machine learning process. Feature engineering involves transforming raw data into informative features, while model selection entails choosing the optimal ML model for a specific task. Both processes significantly influence the accuracy and efficiency of ML models. This paper investigates the impact of feature engineering and model selection on ML model performance through an empirical analysis on various datasets and ML tasks. The findings suggest that the combination of feature engineering and model selection can lead to substantial improvements in prediction accuracy.

Keywords: Feature Engineering, Model Selection, Machine Learning, Prediction Accuracy, Empirical Analysis

I. INTRODUCTION

In the realm of machine learning, feature engineering and model selection stand as two pivotal steps that shape the success of any model. Feature engineering involves transforming raw data into a format that aligns with machine learning algorithms' requirements. This may entail removing redundant features, standardizing data, or crafting new features that hold greater relevance to the prediction task. Model selection, on the other hand, involves choosing the most suitable machine learning algorithm for a specific problem. Given the plethora of available algorithms, each with its distinct strengths and limitations, selecting the optimal one hinges on the data and the task at hand.

Both feature engineering and model selection are iterative processes that demand patience and experimentation. It is not uncommon to explore multiple approaches and assess their outcomes before arriving at the most effective solution. While it may be time-consuming, dedicating the necessary effort is crucial to achieve optimal results. The quality of feature engineering and model selection directly correlates with the performance of the resulting machine learning model.

Fortunately, there is an abundance of resources available to assist data scientists in mastering these techniques. Books, tutorials, and online courses provide comprehensive guidance, while software libraries can automate parts of the process. With the right resources and a commitment to learning, acquiring proficiency in feature engineering and model selection becomes attainable.

By investing time and effort in these critical steps, data scientists can unlock the full potential of their machine learning models. Enhanced performance translates into better decision-making, improved productivity, and increased profitability, ultimately propelling organizations toward success.

II. FEATURE ENGINEERING TECHNIQUES

Feature engineering transforms raw data into meaningful representations that enhance model performance. Several techniques can be employed that are explained in next sections.

A. Data Cleaning

Data cleaning, the initial and fundamental step in feature engineering, plays a pivotal role in enhancing the quality of a dataset. Its primary objective is to address inaccuracies, inconsistencies, and missing values, ensuring that the data is as accurate and complete as possible. This foundational step is crucial because the quality of the data directly influences the effectiveness of subsequent feature engineering and model training processes [2], [3].

Data cleaning encompasses various techniques aimed at improving data integrity. One common technique is imputation,

which involves estimating missing values based on available information. Imputation methods, such as mean, median, or mode imputation, can be employed to fill in missing values, minimizing the impact of missing data on subsequent analyses.

Another important aspect of data cleaning is outlier detection and correction. Outliers are extreme values that may deviate significantly from the rest of the data and can potentially distort the results of analyses. Outlier detection techniques, such as the interquartile range (IQR) method or z-score method, can be used to identify outliers. Once outliers are detected, they can be corrected or removed from the dataset to ensure that the data is representative of the underlying population.

Furthermore, data cleaning involves handling duplicate or redundant data. Duplicate data can arise due to various reasons, such as data entry errors or merging multiple datasets. Identifying and removing duplicate data is crucial to prevent skewing the results of analyses and ensure that each observation is unique and representative.

The process of data cleaning requires careful consideration and attention to detail. It is important to understand the context and domain of the data to determine the appropriate data cleaning techniques. Additionally, it is essential to validate the cleaned data to ensure that it meets the requirements and assumptions of the subsequent feature engineering and modeling processes.

By thoroughly cleaning the data, data scientists can improve the quality and reliability of their analyses. Cleaned data leads to more accurate and robust models, enabling better decision-making and insights.

B. Feature Creation

Feature creation is a crucial step in machine learning, aimed at transforming raw data into a form that is more suitable for modeling and analysis. It involves generating new features from the existing dataset to uncover hidden patterns and relationships that may not be immediately apparent.

One technique used in feature creation is polynomial features. This technique captures the interactions between variables by creating new features that are polynomial terms of the original variables. For instance, a quadratic polynomial feature would include terms like x^2, xy, and y^2, where x and y are the original variables. This technique is particularly useful when there is a non-linear relationship between variables, as it allows the model to capture more complex interactions.

Another technique is domain-specific transformations, which involves applying knowledge from the specific domain or industry to create relevant features. For example, in time series data, features like rolling averages and lagged values can be crucial. Rolling averages smooth out fluctuations in the data and help identify trends, while lagged values capture the dependence of the current value on its past values.

The process of feature creation is iterative, and it often involves experimenting with different techniques and combinations to find the optimal set of features for a particular modeling task. By enhancing the model's ability to capture complex relationships in the data, feature creation significantly improves the predictive performance of machine learning models.

C. Feature Selection

Feature selection is a crucial step in machine learning, aiming to reduce the dimensionality of a dataset by retaining only the most relevant features. It offers several benefits, including improved model interpretability and reduced overfitting. By selecting a subset of features that are highly informative and discriminative, feature selection helps in simplifying the model and making it more interpretable. This is particularly advantageous in scenarios where the dataset contains a large number of features, making it challenging to understand the model's behavior and decision-making process.

Moreover, feature selection helps in mitigating the problem of overfitting. Overfitting occurs when a model learns the specific details of the training data too closely, leading to poor generalization performance on unseen data. By removing irrelevant and redundant features, feature selection prevents the model from capturing spurious patterns in the data and promotes better generalization. This results in models that are more robust and reliable in making predictions on new data.

Several methods are commonly employed for feature selection, each with its own strengths and weaknesses. Recursive Feature Elimination (RFE) is a sequential backward selection method that iteratively removes features based on their importance, as determined by a ranking criterion such as information gain or correlation with the target variable [5]. Principal Component Analysis (PCA) is a dimensionality reduction technique that identifies the principal components, which are linear combinations of the original features that capture most of the variance. SelectKBest is a filter method that selects the top k features based on a statistical measure, such as the chi-square test or mutual information [4].

The choice of feature selection method depends on various factors, including the type of dataset, the learning algorithm used, and the desired level of interpretability. It is often beneficial to experiment with different methods to find the one that best suits the specific problem at hand. By carefully selecting a subset of informative and relevant features, feature selection can significantly enhance the performance and interpretability of machine learning models.

D. Feature Scaling

Feature scaling is a critical step in machine learning that involves transforming raw data into a consistent and comparable scale. It is especially crucial for specific algorithms, including support vector machines (SVM) and k-nearest neighbors (KNN), which are sensitive to the scale of the data [6]. Feature scaling helps normalize the data, making it easier for these algorithms to process and interpret.

One commonly used feature scaling technique is Min-Max Scaling. This method scales the features to a range between 0 and 1 by subtracting the minimum value from each data point and dividing the result by the difference between the maximum and minimum values. Min-Max Scaling ensures that all

features are within the same range, allowing for easier comparison and interpretation.

Another popular feature scaling technique is StandardScaler. In contrast to Min-Max Scaling, StandardScaler scales the features to have a mean of 0 and a standard deviation of 1. This transformation assumes that the data is normally distributed, which is not always the case in real-world scenarios. However, StandardScaler often performs well in practice and is widely used in machine learning.

Applying feature scaling to data offers several benefits. Firstly, it helps to mitigate the impact of outliers, as extreme values are brought closer to the mean. Secondly, it improves the convergence of machine learning algorithms, making them more efficient in finding optimal solutions. Thirdly, feature scaling enhances the interpretability of the model, as the features are now on a comparable scale and their coefficients can be directly compared.

By incorporating feature scaling as a preprocessing step, machine learning models can focus on the relationships between features, rather than being influenced by the scale of the data. This leads to improved performance, enhanced interpretability, and more robust models.

E. Handling Categorical Features

Categorical features are a common challenge in machine learning, as most algorithms require numeric data for processing. Handling categorical features effectively is crucial to ensure accurate and meaningful results from models.

One approach to handling categorical features is One-Hot Encoding. This technique creates a new binary column for each unique category within a feature. For example, if a feature represents the color of a car, One-Hot Encoding would create three binary columns for "red," "blue," and "green." The presence of a category is indicated by a value of 1, while its absence is denoted by a value of 0. This approach is straightforward and preserves the relationship between different categories. However, it can lead to a large number of columns, especially when there are many unique categories [7].

Another technique for handling categorical features is Label Encoding. Unlike One-Hot Encoding, Label Encoding assigns a unique integer to each category. For example, the color "red" might be assigned the integer 1, "blue" the integer 2, and "green" the integer 3. This approach is more compact than One-Hot Encoding, as it requires only one column to represent the feature. However, it assumes that the categories have an inherent order, which may not always be the case.

More advanced methods for handling categorical features include Target Encoding. This technique replaces the categories with the mean of the target variable for each category. For example, if the target variable is the price of a car, Target Encoding would replace the color "red" with the average price of cars that are red. This approach can capture more complex relationships between features and the target variable, leading to improved model performance. However, it requires more computational resources and can be sensitive to outliers in the data.

III. Model Selection Methods

Choosing the right model is crucial for achieving high performance in machine learning tasks. Several model selection strategies are discussed in next sections.

A. Cross-Validation

Cross-validation is a robust and commonly employed technique for assessing the generalization capability of machine learning models. At its core, it entails partitioning the available dataset into k subsets, termed folds. The crux of the method lies in training the model on k-1 folds and evaluating its performance on the remaining fold. This iterative process is repeated k times, with each fold serving as the validation set precisely once. This systematic approach guarantees that the entire dataset is utilized for both training and validation, leading to a thorough evaluation of the model's behavior.

Cross-validation plays a pivotal role in mitigating overfitting, a phenomenon where a model exhibits exceptional performance on the training data but falters when confronted with unseen data [1]. By gauging the model's performance on multiple, independent subsets of the dataset, cross-validation provides a more reliable estimate of its generalization ability. Furthermore, it facilitates the selection of the most suitable model for a given problem.

The utility of cross-validation extends beyond model evaluation. It also serves as a valuable tool for comparing different models, tuning hyperparameters, and assessing the relevance of feature sets. By systematically evaluating various configurations and options, data scientists gain valuable insights that guide their decision-making during the model selection and optimization process.

B. Hyperparameter Tuning

Hyperparameter tuning plays a pivotal role in machine learning, as it involves optimizing the parameters that govern the learning process of a model to enhance its performance. There are various methods for hyperparameter tuning, each with its own strengths and weaknesses.

One commonly used method is Grid Search, which exhaustively searches over a specified parameter grid. This approach ensures that all possible combinations of hyperparameters are evaluated, but it can be computationally expensive, especially for models with a large number of hyperparameters.

Another popular method is Random Search, which samples parameter combinations randomly from a specified range. While Random Search is less computationally intensive than Grid Search, it may not be as effective in finding the optimal hyperparameters.

Bayesian Optimization is a more sophisticated approach to hyperparameter tuning. It builds a probabilistic model of the objective function, which represents the performance of the model as a function of the hyperparameters. Bayesian Optimization then uses this model to select the most promising

hyperparameters for evaluation, making it an efficient and effective method for optimizing model performance.

Effective hyperparameter tuning can significantly improve the accuracy and robustness of a machine learning model. By optimizing the hyperparameters, it is possible to reduce overfitting, improve generalization, and enhance the overall performance of the model on unseen data.

C. Ensemble Methods

Ensemble methods are a powerful tool for improving the performance of machine learning models. By combining the predictions of multiple models, ensemble methods can reduce variance, improve accuracy, and make more robust predictions. Three common ensemble methods are Bagging, Boosting, and Stacking.

Bagging (Bootstrap Aggregating) involves training multiple instances of the same model on different subsets of the data. The predictions from these models are then averaged to produce a final prediction. Bagging reduces variance by averaging out the errors of the individual models. It is particularly effective when the base models are unstable, meaning that they are sensitive to small changes in the data.

Boosting is another ensemble method that involves training multiple models sequentially. Each model is trained on a weighted version of the data, with the weights adjusted based on the performance of the previous models. The goal of boosting is to create a sequence of models that are increasingly accurate. Boosting is particularly effective when the base models are correlated, meaning that they make similar errors.

Stacking is an ensemble method that involves training a meta-model to combine the predictions of several base models. The base models are typically trained on different subsets of the data or using different algorithms. The meta-model is then trained to predict the final output based on the predictions of the base models. Stacking can be more accurate than Bagging or Boosting when the base models are diverse, meaning that they make different errors.

Ensemble methods have been shown to outperform single models significantly in a variety of tasks, including classification, regression, and clustering. They are particularly useful when the data is noisy, high-dimensional, or nonlinear.

D. Model Evaluation Metrics

Evaluating the performance of machine learning models is a crucial step in the development and deployment process. To do this effectively, it's essential to select appropriate evaluation metrics that provide insights into different aspects of the model's behavior. Some commonly used metrics include accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) [2], [7].

Accuracy measures the overall correctness of the model's predictions, but it can be misleading in certain scenarios. For example, a model that always predicts the majority class will have high accuracy even if it fails to identify the minority class correctly. Precision, on the other hand, measures the proportion of positive predictions that are correct, providing information about the model's ability to avoid false positives. Recall, also known as sensitivity or the true positive rate, measures the proportion of actual positive instances that are correctly identified, giving insights into the model's ability to detect true positives.

The F1-score combines precision and recall into a single metric, offering a balanced evaluation of the model's performance. It considers both false positives and false negatives, making it suitable for scenarios where both types of errors are equally important. Additionally, the AUC-ROC curve provides a comprehensive view of the model's performance across all possible classification thresholds. It measures the ability of the model to distinguish between positive and negative instances and is particularly useful when the dataset exhibits imbalanced class distributions.

Selecting a suitable set of evaluation metrics is crucial for assessing the model's performance comprehensively. By considering multiple metrics, data scientists can gain a deeper understanding of the model's strengths and weaknesses, enabling them to make informed decisions about model selection and tuning.

IV. CONCLUSION

Feature engineering and model selection are indispensable for building high-performing machine learning models. Our analysis highlights the effectiveness of various techniques in enhancing model accuracy and efficiency. By systematically applying data cleaning, feature creation, selection, and scaling, we can significantly improve the quality of input data, leading to better model performance. Furthermore, strategic model selection and hyperparameter tuning ensure that the chosen models are well-suited to the task, maximizing their predictive power.

Future work will explore automated feature engineering and model selection methods to further streamline the model development process. Advances in AutoML and hyperparameter optimization techniques hold promise for making these processes more efficient and accessible, allowing practitioners to focus on higher-level aspects of machine learning model development. Additionally, exploring the integration of domain knowledge into feature engineering processes can further enhance the relevance and predictive power of the generated features.

REFERENCES

[1] Dietterich, Thomas G.. “Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms.” *Neural Computation* 10 (1998): 1895-1923.

[2] Saito, Takaya and Marc Rehmsmeier. “The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets.” *PLoS ONE* 10 (2015).

[3] Heaton, Jeff. “An empirical analysis of feature engineering for predictive modeling.” *SoutheastCon* (2016): 1-6.

[4] Nargesian, Fatemeh, Horst Samulowitz, Udayan Khurana, Elias Boutros Khalil and Deepak S. Turaga. "Learning Feature Engineering for Classification." *International Joint Conference on Artificial Intelligence* (2017).

[5] Li, Jundong, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. "Feature Selection: A Data Perspective." *ACM Computing Surveys* (2017) 50 (6): 94.

[6] Uddin, Muhammad Fahim, JeongKyu Lee, Syed Sajjad Hussain Rizvi and Samir E. Hamada. "Proposing Enhanced Feature Engineering and a Selection Model for Machine Learning Processes." *Applied Sciences* 8 (2018): 646.

[7] Roe, Kenneth D, Vibhu Jawa, Xiaohan Tanner Zhang, Christopher G. Chute, Jeremy A. Epstein, Jordan K. Matelsky, Ilya Shpitser and Casey Overby Taylor. "Feature engineering with clinical expert knowledge: A case study assessment of machine learning model complexity and performance." *PLoS ONE* 15 (2020).