Guide to Tackle Machine Learning Competitions

Dr. Belkacem Sami

May 13, 2023

1   Read and Understand the Problem

Before starting any competition, it's essential to read and understand the problem statement. It's also essential to refer to the documentation of the dataset to understand the features, labels, and other information related to the problem.

2   Preview the Data and Define the performance Measure

After understanding the problem, it's crucial to preview the data and the columns. It helps to know the size of the dataset, the number of features, and the type of data. We can also define the performance measure at this stage.

3   Read Literature and Previous Solutions

Reading literature, forums, discussions, and previous solutions related to the current problem can help us get ideas to efficiently approach the problem. The purpose is to save time and prevent us from reinventing the wheel.

4   Data Preparation

Data preparation involves data cleaning, data augmentation, data visualization, and feature engineering. It's essential to clean the data and design good features, as it can go a long way in improving the performance of our model.

4.1   Data Cleaning

Data cleaning involves handling missing data, removing duplicates, handling outliers, checking for data compatibility, and other preprocessing steps.

4.2   Data Augmentation

Data augmentation involves generating new data by applying transformations like rotations, translations, and scaling. It enhances the robustness of the model.

4.3 Data Visualization + Time Features

Data visualization can help us understand the patterns and trends in the data. It's also essential to add and visualize time features if applicable to the problem.

4.4 Feature Engineering

Feature engineering involves selecting the most relevant features and creating new ones from the existing features. It improves the performance of the model.

5 Select the Model

If we have a large dataset and care about speed, then we can choose a machine learning algorithm based on speed or ease of use. If we care about accuracy, we should try a bunch of different models and select the best one by cross-validation. Ensemble learning methods such as XGBoost or LightGBM are suitable for tabular data, and deep learning methods are suitable for non-tabular data.

6 Train and Fine-tune the Model

Train the model using the training data and fine-tune the model to select the best hyperparameters by considering (1) cross-validation methods, (2) parameters according to the type of the problem, and (3) hyperparameter tuning methods such as Grid search, Randomized search, and Bayesian optimization.

7 Plot the Learning Curve

Plotting the learning curve can help us detect overfitting or underfitting in the model. We can adjust the model accordingly to improve its performance.

8 Evaluate the Model

After fine-tuning the model, it's essential to evaluate it, validate it, and compare it with baselines or other models. We can evaluate the model using various evaluation metrics such as accuracy, precision, recall, F1 score, AUC-ROC score, MSE, RMSE, etc., depending on the type of problem. It's also crucial to validate the model on unseen data to ensure that it generalizes well to new data.

9 Check the Type of Predictions

We must check the type of the predicted values to ensure that it meets the problem's requirements, e.g. float, integer, Boolean, no negative values, etc.

10 Explain the Machine Learning Model

Finally, we can explain the machine learning model by using techniques like feature importance, partial dependence plots, SHAP values, linear model coefficients, and so on. Explaining the model can help us gain insights into the model's behavior and improve its performance. It can also help us gain the trust of stakeholders and make informed decisions based on the model's predictions.

Note

- Reiterate through steps 4 to 8 to improve the model's performance.

- Avoid data leakage, it happens when the training data contains information about the target, but similar data will not be available when the model is used for prediction.

- Consider transfer, weakly, self, and semi-supervised learning for a gain of data and learning time.