ASL Inidividual Assignment 01

Joshua Damm

2024-12-04

Data Preparation

The data were split into 75% training and 25% test data. Later, I am asked to compare the performance of the XGBoost ensemble with the Gaussian Process Regression (GPR). Since we are asked to only consider numerical features in the GPR, I am also only going to look at numerical features in the XGBoost.

Question 1

For the random forest (RF) model without gradient boosting, the number of trees could not be specified in the initial parameter grid. I tried out different values yielding similar RMSE values as random forests rely on a large ensemble of decision trees, and the number of trees determines how robust the predictions are. Therefore, tuning this parameter is often less critical. The performance of the RF model (lower cross-validated RMSE) initially increased with higher numbers of included predictor, reached its best performance at 10 predictors used for splitting. However, 5 predictors yielded a comparable RMSE and should therefore be preferred due to parsimony. After that, the performance went down again, suggesting overfitting for higher numbers of predictors.

For the gradient boosted machine (GBM) model, The hyperparameters interact with each other so that clear separable trends are less clear to extract. For shrinkage parameters (learning rates) of 0.001 the number of boosting iterations lead to increased performance. For a shrinkage of 0.01, the RMSE goes up again after 1000 iterations. For a shrinkage of 0.1 the RMSE is increasing proportionally with the number of boosting iterations, suggesting overfitting. There is no clear trend in the amount of interaction depth.

For the XGBoost, the interaction depth of the threes also does not play a crucial role in cross-validated model performance. The number of boosting iterations is generally leading to a better model performance for the small shrinkage parameter (slow learning rate). For the moderate learning rate of 0.01 however, the performance slightly decreases again after a number of 1000 boosting iterations, suggesting overfitting for higher boosting iterations. The learning rate of 0.1 is too high and the model overfits on the training data.

Question 2

After plugging in the optimal hyperparameters in the XGBoost model we make predictions based on the test data. After that we compute the Root Mean Squared Error (RMSE) between our predictions and the target observations on the test data. The RMSE is appr. 22.

Question 3

I chose to pick gain as my importance measure. It measures the average improvement in the model's accuracy based on the loss function brought by a feature when it is used for splitting. This measure focuses on the quality of the splits and not just their quantity (frequency). Using gain also brings limitations like its sensitivity to outliers and noise which sometimes leads to assigning a high importance to features that are only predictive for a small subset of the data. Therefore, other importance measures should also be included in the final decision based on context.

Question 4

Based on gain, AgeAFQT is the most relevevant indicator /obvious driver of predictive performance, followed by a more nuanced predictive relevance by PermInc, HOMECog_Pct88, and DOB_Yr_Child.

Question 5

5a

I standardized predictor variables (e.g., subtracting the mean and dividing by the standard deviation) while fitting the GPR as recommended. It ensures that the kernel hyperparameters (e.g., length scales) are comparable across dimensions, improving numerical stability and optimization during training. Unfortunately, for the linear and neural network kernel, not all model parameters reached convergence also with higher repeats (up to 25) and different tolerance values. The neural network kernel shows the best model evidence (lowest negative log marginal likelihood) and shows therefore the best fit to the training data.

5b

One advantage of using the bayesian approach of selecting a model via model evidence vs. the more frequentist approach of cross-validation (CV) is that it is more computationally efficient. The evidence of the model is the marginal likelihood, which computes the likelihood of the data given the model in a single evaluation by integrating over all possible parameter values and considering the whole dataset. In CV, the data is being split many times to evaluate the optimal parameters by considering multiple subsets of the data, which is more computationally costly, especially for more complex models like tree ensembles and for large complex data sets.

A second advantage could be the robustness of model evidence selection to overfitting. The marginal likelihood (model evidence) integrates over all possible model parameters, including the likelihood of the data given the model and prior information about the parameters (potentially inferred from the data), therefore balancing model fit and complexity. This also reduces the risk of overfitting through penalizing model parameters constellations that are overly complex. CV, on the other hand, optimizes the parameters based on data splitting, leading to variability in the model performance criterion (e.g. RMSE as the loss function) based on the splits that were run. In tree ensembles, for instance, greedy algorithms make splits on their influence on the reduction of the loss function, which can lead to overfitting based on noise in the data, or only considering highly influetial variables in earlier splits in a greedy fashion, as opposed to considering all possible ways of splitting and evaluating the data.

Question 6

6a

I used the best performing neural network kernel to make predictions based on the test set. The MSE was appr. 497.90.

6b

The latent predictive distribution includes the “true” noiseless values of the target variable at given input locations. It represents the model’s belief about the true, underlying relationship of the predictors and the response. The observed predictive distribution, however, represents the actual distribution of observed values given the predictor values.

- posterior variance captures the model’s uncertainty in the latent predictions, derived from the posterior covariance matrix
- After adding the noise variance to the posterior variance we obtained the variance of the observed posterior predictive distribution
- We construct the 95% CI for the predictive mean of the observed predictive distribution
- **Approximately 97.4% of the test data points fall within 95% probability mass of the observed predictive distribution**

Question 7

The first model is more complex because of its kernel (covariance function). Given that both models share the same noise variance β^{-1} of 1, only the models’ complexity is determined by their respective kernels. The first kernel implies that for two arbitrary inputs the covariance is scaled by a factor of 10, whereas the covariance is only scaled by a factor of two for the second kernel. A larger kernel scaling factor allows for more flexibility when fitting the data, whereas a smaller scaling factor corresponds to a smaller prior variance for the covariance function, which makes the model fit less flexible, i.e. more constrained. The first model shows therefore higher flexibility, but also increased risk of overfitting. On the other hand, the higher flexibilty also improves the capacity to capture more complex relationships in the data. The second model shows less flexibility and might therefore generalize better to noisy data with the price of a risk for underfitting.

Question 8

8a

The XGBoost model shows a better predictive performance with an RMSE of appr. 22.00 vs. a RMSE of appr. 22.31 for the GPR model.

8b

Two advantages of XGBoost over GPR

1) XGBoost is omptimized for large (high-dimensional) data sets by using efficient memory usage and parallelization. GPR scales poorly with increasing number of data points due to the inversion of increasingly large covariance matrices in more complex kernels (like the squared exponential).

2) Due to its tree-based structure XGBoost is able to deal with complex non-linear relationships and interaction in the data well. It is therefore also well suitable for heterogenous data structures with potential missing values. GPR, however, requires careful kernel selection to tune the model to complex data structures and convergence issues can occur frequently, especially with larger and more complex data sets. We saw the convergence issues in this assignment as well, even though we only selected numeric predictor variables, as for categorical predictors, different kernel functions would have had to be selected.

Two advantages of GPR over XGBoost

1) GPR provides a probabilistic framework by including uncertainty information, e.g. in the form of credible intervals in the predictive distribution. XGBoost does not include this.

2) GPR's reliance on kernel enables more model interpretability than XGBoost. One can explicitely choose to model a relationship in the data (linear, periodic, an arbitrarily complex smooth function) and investigate the model fit. This helps to come closer to understanding the data generating process. XGBoost does not have a similar feature and relies on post-hoc analyses like feature importances.