Aprendizagem Automática 23/24

First Home Assignment

Grupo 33

Autores: Guilherme Cepeda – 62931 horas: 29

Guilherme Rosário – 62543 horas: 8

Marco Viana – 62550 horas: 8

Objective 1

The primary objective of this work was to create the best regression model to predict the variable **critical_temp**. Before building a model, the priority was to understand the type of data contained in the dataset and assess whether it was necessary to process the data or remove any variables. We realized that the data was already cleaned, and all the 81 columns were numerical. Based on the correlation function we could understand how much each attribute correlates with the target variable, the 3 more correlated attributes are: *wtd_std_ThermalConductivity; range_ThermalConductivity and std_ThermalConductivity.* This dataset was collected with the goal of predicting the **critical_temp** variable values using the 81 features from superconductors.

The algorithms we chose to create the regression model were the CART (Classification and Regression Trees), **using the mean squared error (MSE) as the impurity metric** and the Linear Regression Model since it's a simpler model to test. Since it's a ***decision tree-based approach***, we did not need to perform variable scaling. We started by splitting the dataset into two parts: 75% of the data was used to train and evaluate the models, and the remaining 25% was used to create the **independent validation set (IVS).** This separation is crucial when we don't have dedicated data to test the model, as the IVS simulates the model's performance in real-world scenarios where it would be used.

To train a good model that fits the data well and can predict our variable of interest (critical_temp), we tried to understand which hyperparameters could be useful and better suited to our case. To test different models and have a basis for comparison, we selected the stopping criteria of **maximum depth (max_depth) and the minimum number of samples per leaf node (min_samples_leaf).**

We opted for the k-fold cross-validation method to assess the model quality. The dataset was divided into ten partitions (k = 10), with each fold serving as a testing set while the remaining data was used for training. Various standard performance metrics were employed to evaluate the models, with a primary focus on the Pearson correlation coefficient, which measures the linear relationship between actual and predicted values. This was complemented by an examination of mean absolute error (MAE), root mean squared error (RMSE), and maximum error (ME). These error-related metrics are of paramount importance in industry-related applications, where precision in predicting values associated with a specific outcome is a key objective.

As mentioned earlier, the trained models differed in the choice of hyperparameters and their tuning. To determine which model was most likely to have superior performance in predicting the IVS data and, consequently, a more generalized application, we conducted an iteration where we slightly varied the hyperparameter values and compared the scores between the training set and the test set. The main goal of this approach was to select models among all hyperparameter values that did not exhibit overfitting or underfitting. Thus, we varied max_depth from 1 to 28 in increments of 3, resulting in 10 different models,

and we also varied min_samples_leaf from 1 to 55 in increments of 6, generating another 10 models. And finally, just one iteration of the Linear Regression model generates only one model.

Model	Val hyperparameter	RVE	RMSE	Correlation Score	ME	MAE
	max_depth 1	0.53	23.408	0.728	175.525	17.463
	max_depth 4	0.731	17.716	0.855	162.537	12.248
	max_depth 7	0.806	15.031	0.898	184.35	9.707
	max_depth 10	0.846	13.401	0.92	184.35	8.025
	max_depth 13	0.868	12.399	0.932	184.35	6.956
	max_depth 16	0.871	12.267	0.934	184.35	6.504
	max_depth 19	0.865	12.52	0.932	184.35	6.465
	max_depth 22	0.867	12.461	0.932	184.35	6.453
	max_depth 25	0.874	12.095	0.936	168	6.333
Decision Tree Regressor	max_depth 28	0.87	12.318	0.934	184.35	6.386
	min_samples_leaf 1	0.867	12.454	0.933	184.35	6.44
	min_samples_leaf 7	0.881	11.785	0.939	171.275	6.63
	min_samples_leaf 13	0.873	12.161	0.935	158.576	6.962
	min_samples_leaf 19	0.869	12.335	0.932	180.828	7.246
	min_samples_leaf 25	0.868	12.382	0.932	160.969	7.387
	min_samples_leaf 31	0.863	12.623	0.929	175.614	7.623
	min_samples_leaf 37	0.857	12.885	0.926	177.73	7.826
	min_samples_leaf 43	0.852	13.112	0.923	166.095	8.025
	min_samples_leaf 49	0.85	13.197	0.922	177.25	8.153
	min_samples_leaf 55	0.845	13.435	0.919	173.516	8.357

Table 1 – Summary of performance statistics of the 20 models

Model	Val hyperparameter	RVE	RMSE	Correlation Score	ME	MAE
Linear Regression	None	0.733	17.632	0.856	176.057	13.354

Table 2- Linear Regression Model performance statistics

After analyzing Table 1, we can conclude that between the values **max_depth = 7 and max_depth = 13**, the correlation values of the test set begin to stabilize. There is no further benefit in increasing this hyperparameter beyond that point. However, between the range of **min_samples_leaf = 13 to min_samples_leaf = 55** the Correlation score value starts to decrease proving that there is no further benefit in increasing the hyperparameter.

Considering that these 21 models have already been evaluated based on Pearson correlation scores, Table 1 and 2 includes various other performance statistics of the models that can now be used for our final selection. Looking at the RVE and RMSE values exhibit very slight variations. However, the model with **min_samples_leaf = 13** stands out as it has the lowest maximum error, which is a value we aim to minimize. Finally, the hyperparameter that instills the most confidence in producing the best model based on the tested conditions is **min_samples_leaf = 13**. A new model with this hyperparameter was trained using the training dataset and then tested with the IVS, resulting in the statistics included in Table 3.

Best Model hyperparameter	RVE	RMSE	Correlation Score	ME	MAE
min_samples_leaf 13	0.877	12.048	0.937	109.493	6.773

Table 3 – Best Regression model performance statistics with IVS

We can conclude that the selected model demonstrates the potential to predict the variable of interest. This is supported by its performance with the IVS data, which closely aligns with the model's performance during the testing phase and even reduces the ME. As a result, the model doesn't seem to suffer from overfitting or underfitting, showcasing effective generalization to new data.

Objective 2

The second objective of this work was to create the best binary classification model assuming as positive all instances with values of **critical_temp >= 80.0 (represents 17%) and as negatives all remaining cases,** the correlations between the features and the target variable are similar from those on the previous objective**.**

In pursuit of our objective, we introduced a new column in the dataframe, which we named **critical_temp_high**. This column contains positive instances with critical temperatures greater than or equal to 80.0, while all other instances are marked as negatives. Subsequently, we removed the earlier **critical_temp** column from the dataframe since it was no longer needed for this particular objective.

Just as in the previous objective, we used the CART algorithm, but this time to create a classification model, utilizing the **Gini impurity metric to build a Decision Tree** and we also used **the Logistic Regression Model** to compare results. Initially, we divided the dataset into two parts: 75% for training and evaluating the models, and the remaining 25% to create the IVS. Based on the hyperparameters used in objective 1, we decided to keep the same stopping criterion (max_depth and min_samples_leaf) and added the C hyperparameter in the Logistic Regression model. We again employed the K-Fold cross-validation method for model validation, with k=10. The performance of the models trained from the cross-validation subsets was assessed using various statistics, including the Matthews Correlation Coefficient (MCC) and Recall, which represents the ability of a model to predict all positive cases. Given the nature of the problem, Recall takes precedence over Precision, as it is preferable to detect all positive cases, even if it means some false positives, rather than leaving cases undetected.

Model	Val hyperparameter	Accuracy	Precision	Recall	F1-Score	Matthews Correlation Coeficient
	max_depth 1	0.833	0	0	0	0
	max_depth 4	0.879	0.671	0.531	0.593	0.528
	max_depth 7	0.911	0.735	0.729	0.732	0.679
	max_depth 10	0.926	0.776	0.782	0.779	0.735
	max_depth 13	0.929	0.78	0.799	0.789	0.747
	max_depth 16	0.929	0.785	0.795	0.79	0.748
	max_depth 19	0.929	0.789	0.782	0.786	0.743
	max_depth 22	0.929	0.789	0.786	0.787	0.745
	max_depth 25	0.928	0.79	0.777	0.784	0.741
Decision Tree Classifier	max_depth 28	0.93	0.788	0.79	0.789	0.747
	min_samples_leaf 1	0.926	0.778	0.779	0.779	0.735
	min_samples_leaf 7	0.927	0.78	0.787	0.783	0.74
	min_samples_leaf 13	0.924	0.781	0.757	0.769	0.724
	min_samples_leaf 19	0.92	0.769	0.741	0.755	0.707
	min_samples_leaf 25	0.918	0.772	0.718	0.744	0.696
	min_samples_leaf 31	0.916	0.768	0.713	0.739	0.69
	min_samples_leaf 37	0.916	0.774	0.703	0.737	0.689
	min_samples_leaf 43	0.913	0.758	0.704	0.73	0.679
	min_samples_leaf 49	0.912	0.757	0.695	0.725	0.674
	min_samples_leaf 55	0.914	0.763	0.7	0.73	0.68

Table 4 – Summary of the performance statistics of the 20 models, Decision tree Classifier

After analyzing Table 4, we can conclude that between the values of max_depth = 13 and max_depth = 28, the MCC values for the test set start to stabilize, and there is no significant gain in increasing this hyperparameter beyond that value. However, between min_samples_split = 25 and min_samples_split = 55, the MCC values for the test set start to stabilize, and there is no gain in increasing the hyperparameter value.

Model	Val hyperparameter	Accuracy	Precision	Recall	F1-Score	Matthews Correlation Coeficient
	C 0.001	0.861	0.736	0.261	0.386	0.384
	C 0.002	0.864	0.731	0.287	0.413	0.401
	C 0.005	0.87	0.729	0.35	0.473	0.445
	C 0.01	0.873	0.705	0.406	0.515	0.47
	C 0.02	0.875	0.695	0.442	0.54	0.488
Logistic Regression	C 0.05	0.875	0.665	0.498	0.57	0.505
	C 0.1	0.885	0.679	0.59	0.631	0.566
	C 1	0.89	0.689	0.622	0.654	0.59
	C 10	0.892	0.691	0.639	0.664	0.601
	C 100	0.891	0.685	0.636	0.66	0.596

Table 5 – Summary of the performance statistics of the 10 models, Logistic Regression

Tables 4 and 5 summarize all the performance statistics of the 30 models, with particular emphasis on the hyperparameter **max_depth = 13** as it exhibits the best **recall/F1 score** among all the models. As previously mentioned, recall plays a crucial role in classification related problems, where false negatives can have severe consequences. Therefore, **max_depth = 13** was chosen as the hyperparameter to train the final model with the training dataset for testing with the IVS (Table 6).

Best Model hyperparameter	Accuracy	Precision	Recall	F1-Score	Matthews Correlation Coeficient
max_depth 13	0.935	0.827	0.785	0.806	0.767

Table 6 – Best binary classification model performance statistics with the IVS

In conclusion, based on the performance statistics, which did not vary significantly between the model's testing phase and the test with the IVS, it can be inferred that the chosen model has the potential to be used and perform well when applied to real-world data.