Model Performance and Parameters

EBENEZER OLUBAYODE

April 16, 2024

Model Performance

Model Type	Accuracy
SVM	0.865
Logistic Regression	0.846
Decision Tree	0.868
Neural Network	0.837
k-Nearest Neighbors	0.844

Best Parameters For Each Model

Parameter	SVM	Logistic Regression	Decision Tree	Neural Network	k-NN
Cost	1 (2^0)	NA	NA	NA	NA
rbf_sigma	0.25 (2^{-2})	NA	NA	NA	NA
Penalty	NA	0.000339	NA	0.00000278	NA
cost_complexity	NA	NA	0.001	NA	NA
tree_depth	NA	NA	15	NA	NA
min_n	NA	NA	10	NA	NA
hidden_units	NA	NA	NA	6	NA
neighbors	NA	NA	NA	NA	26

Performance Analysis

Decision Tree exhibits the highest accuracy at 0.868. This suggests that for the given dataset and task, the Decision Tree model might capture the patterns most effectively, especially given its ability to model non-linear relationships and interactions between features.

SVM (Support Vector Machine) is a close second with an accuracy of 0.865. This model, which maximizes the margin between classes, is also performing quite well, indicating its robustness in handling high-dimensional data or data where classes are not easily separable by a simple hyperplane.

Logistic Regression shows an accuracy of 0.846. As a linear model, it might be slightly less capable of capturing complex patterns as effectively as

Decision Tree or SVM but still performs reasonably well, possibly due to a well-tuned penalty parameter of 0.000339 that helps in managing overfitting.

k-Nearest Neighbors (k-NN) has an accuracy of 0.844. This model's performance is slightly lower, which might be due to its dependency on the local neighborhood, which can affect its ability to generalize well on unseen data. Additionally, k-NN is sensitive to the distance metric used; disparities in feature scales or the inclusion of irrelevant features can distort the perceived similarities between instances which can leads to inaccurate predictions.

Neural Network has the lowest accuracy of 0.837. Neural networks' performance is highly contingent upon effective hyperparameter tuning, including the number of hidden units, learning rate, epochs, batch size, and the incorporation of various layers like dropout for regularization. Despite diligent tuning and the choice of using the best hyperparameters, including six hidden units, this might not still be optimal for the model. Neural networks depend heavily on a wide range of hyperparameters, such as learning rate, epochs, batch size, and dropout. However, the only parameter used for training this model was the number of hidden units. This approach might be too simplistic for the dataset, or there could be issues related to the non-tuning of other critical hyperparameters.

In summary, **Decision Tree** and **SVM** show the best out-of-sample performance, closely followed by **Logistic Regression** and **k-NN**. The **Neural Network** lags slightly behind in this particular case. The differences in performance can often be attributed to how each algorithm handles the underlying data structure and complexity, as well as how well their hyperparameters are tuned to prevent overfitting while maintaining good generalization on unseen data.

General Considerations

The inherent characteristics of dataset (such as non-linear separability, noise, and class overlap) can limit the effectiveness of certain models. Both k-NN and NN might struggle if the data doesn't suit their strengths.

And regarding evaluation metric: While accuracy is a straightforward metric, it might not fully capture the model's performance, especially in imbalanced datasets. Other metrics like precision, recall, F1-score, or ROC-AUC might provide more insights into where the models are lacking. Given these factors, even well-tuned models using k-NN and neural networks might not always achieve the highest accuracy compared to models like decision trees or SVMs, which might be better suited to the specific characteristics of the dataset or are more robust to certain types of data irregularities. It would be good to explore other metrics depending on the datasets and its interpretation and compare.