Assignment 1 – Supervised Learning

Michael Lukacsko  
mlukacsko3@gatect.edu

# Introducing the data

## Breast Cancer Dataset

The breast cancer dataset (Dua, D. and Graff, C. 2019) is a widely used collection of patient records collected from the University of Wisconsin. The dataset contains information on 569 patients and is composed of 357 benign cases and 212 malignant cases. Each patient record in the dataset includes various features such as the patient's age, tumor size, and the presence of certain biomarkers, as well as a binary classification for diagnosis (benign or malignant).

## Wine Origin Dataset

The data in the wine dataset (C.Blake, 1998) is the result of a chemical analysis of wines produces in a specific area of Italy but derived from three different cultivators. Each of the 178 samples include 13 features such as alcohol content, acidity, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue and OD280/OD315 of diluted wines, and a target variable indicating the wine cultivator (1,2,3).

What is immediately noticeable in the wine dataset is that, unlike the binary target class value in the breast cancer dataset, the target class of this dataset is more complex because of its multi class target represented by a 1, 2, or a 3. However, because both datasets are well balanced, they provide a more realistic representation of the population and can lead to better model performance. This common denominator helps ensure that the models being tested are not biased towards a specific class, as all classes are represented equally.

# preprocessing of the data

Preprocessing data for machine learning is an important step because it ensures that the data is in a suitable format for the model to learn from and can help improve the model's performance. Because both datasets are well balanced, there was very little preprocessing that needed to be done. As such, I converted the ‘B’ and ‘M’ target value in the breast cancer dataset to a 0 or 1. Moreover, I dropped the ‘ID’ column as it is not needed. The wine origin dataset did not need to be cleaned up. Next, scaling of the data is performed on both datasets. This preprocessing step is vital because all features will have similar ranges, which can help prevent certain features from dominating the model. The sklearn.preprocessing.scale package was used to standardize the data by removing the mean and scaling to unit variance. To complete preprocessing, each of the datasets were split into training/test data using sklearn.model\_selection.train\_test\_split. 80% of the data is used for training and the remaining 20% for test data.

# classifier 1: decision tree

## Wine Origin Dataset

Starting with a vanilla decision tree and no hyperparameter turning, the decision tree classifier runs with an accuracy of 85.25%

### Hyperparameter Tuning

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedThe first parameter selected as part of hyperparameter tuning was maximum depth. Using a max\_depth range of 1-20 to pre-prune, it can be seen in the validation curve plotted below (left image) that the performance of the model as a function of the maximum depth of the tree shows the growth of the decision tree to saturation once the maximum depth exceeds 10. That is, the training score has reached a point at which increasing the depth of the tree no longer improves the performance of the model and starts to cause overfitting of the data.

1. Decision tree hyperparameter tuning: Wine Dataset – Left: Pre-Pruning Max Depth. Right: Post Pruning Cost Complexity (CCP).

The second hyperparameter used to tune the decision tree model is cost complexity post pruning. Testing the decision tree with a ccp\_alpha range between 0.0001 and 0.01 is shown in figure 1 above (right image). It is clear that the greatest accuracy is achieved between alpha=0.0001 and 0.0023. Moreover, even though our training accuracy has decreased to 0.95, this model will now be more generalized, and it will perform better on unseen data.

Chart, line chart

Description automatically generated Lastly, using GridSearchCV to perform an exhaustive search over a specified parameter grid. The learning curve plotted below in figure 2 is a visual representation of this.

1. Learning Curve: Wine Data Set

The accuracy of this model increases 2% to 87.25%. However, there is a large gap between the training and cross-validation scores. This could be attributed to the model having a high variance and low bias, which causes overfitting.

## Breast Cancer Dataset

Again, a vanilla decision tree is used to determine a baseline accuracy. The decision tree model using the breast cancer dataset, without any tuning, produces as accuracy score of 90.35%.

### Hyperparameter Tuning

Using the same maximum depth range, 1-25, figure 3 below shows that the decision tree reaches saturation with a max depth of 7. Moreover, like the learning curve generated for the wine origin dataset, there is overfitting. This is evident by the large gap between the training and cross validation scores. The second hyperparameter, ccp\_alpha, also shows a high bias and overfitting of the data. Using the same range, 0.0001 to 0.01, note that the training score decreases while the cross-validation score increases slightly. It is possible that more data and/or removing less important features would help with the Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedbias and overfitting.

1. Decision Tree Breast Cancer Dataset – Left: Pre-Pruning Max Depth. Right: Post Pruning Cost Complexity (CCP).

Chart, line chart

Description automatically generatedThe learning curve below is plotted after GridSearchCV determined the best parameters to use. Like the learning curve in figure 2, there is a large gap between cross validation and training scores, however, it also shows an interesting difference as compared to the learning curve in figure 2. Here, the training and cross validation scores move away from each other in the last 20% of training examples.

1. Learning Curve: Breast Cancer Dataset

# CLASSIFIER 2: Neural Network (NN)

## Wine Origin Dataset

Using a NN classifier as the second model, the initial NN performed quite well with an accuracy of 97.22%.

### Hyperparameter Tuning

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedTo tune this NN classifier, hidden layer size and alpha are used. Starting with an alpha of -10, this parameter is tested up to a range of 10. The results figure below on the left show that when alpha=1, there is a significant drop off in accuracy. The second hyperparameter, hidden layer size, is tested with a size of 2 to 30. In general, a hidden layer with a large number of units can capture complex relationships between the inputs and outputs, leading to improved model accuracy. However, a large hidden layer size can also lead to overfitting. The right plot in figure 5 below shows this well – The model performs quite well as hidden layer size increases from 2 to about 10 but, after that, the model begins to increase its variance, reduce bias, and overfit the data. In contrast to the alpha parameter, accuracy increases as the units get larger.

1. Neural Network hyperparameter tuning: Wine Origin Dataset – Left: Alpha. Right: Hidden Layer Size

Chart, line chart

Description automatically generatedChart

Description automatically generatedFinally, a learning curve and loss curve are plotted in figure 6 below. Using GridSearchCV to determine optimal alpha and hidden layer size, it would appear that the neural network classifier is a good model to use for this dataset. Very little overfitting is present, and the model performs optimally after around 60% of the training examples are used. Moreover, the loss curve, also plotted below, shows that, over 200 epochs, this model has a very good learning rate and achieves low error values.

1. Neural Network, Wine Origin Dataset. Loss curve, left. Learning Curve, right

## Breast Cancer Dataset

The NN classifier fitted with the breast cancer dataset performs slightly under the wine origin dataset. An accuracy of 96.49% is achieved.

### Hyperparameter Turning

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedTuning alpha and hidden layer size using the breast cancer dataset are plotted in figure 7 below. As can be seen in the plot on the left, the tuning of alpha, using the same range as that used in the wine origin dataset, has a similar effect. By increasing alpha, the model is restricted in a way that reduces the weight of the model. This results in a reduction of overfitting, which can be seen when alpha is equal to about 1. Equally important to note is that an alpha value greater than one causes underfitting of the data and accuracy drops significantly, resulting in a high bias and low variance. The second hyperparameter, plotted on the right in figure 7, is hidden layer size. This parameter was chosen because it has a significant impact on model performance. As it relates to the breast cancer dataset, a hidden layer size less than 4 causes this classifier to be too simple to capture the true relationship between the features and the target. After a value of 5 is reached, the model performs quite well until a value of about 25 where the gap between training and cross-validation scores begin to increase because of overfitting.

1. Neural Network hyperparameter tuning: Breast Cancer Dataset – Left: Alpha. Right: Hidden Layer Size

Chart, line chart

Description automatically generatedChart

Description automatically generatedPlotting the loss curve and learning curve show some similarities with respect to the loss curve, however, the learning curve looks very different than it did using the wine dataset. Here, the learning curve scores do not converge, suggesting that overfitting is occurring because of the model being too complex and fitting the training data too well, but failing to generalize well to new data.

1. Neural Network – Breast Cancer Dataset. Loss Curve, Left. Learning Curve, Right

# classifier 3: boosting

## Wine Origin Dataset

Using a vanilla AdaBoost classifier and the wine origin dataset, this classifier predicts the class value with an accuracy of 89.00%.

### Hyperparameter Tuning

The first hyperparameter used to tune this classifier is number of estimators. Here, more estimators refer to the rounds of trees the model will fit. A higher number of estimators results in a more complex model, and, as can be seen in the plot below, is a validation curve with very high variance thus causing the data to be overfit. The second hyperparameter, learning rate, is another key parameter used to tune this boosting classifier. The validation curve below is different than any of the other validation curves previously evaluated as there is no significant gap between the training and cross-validation scores. As shown in validation curve below, the ideal learning rate at which there accuracy is highest. Also, important to note is the closeness of the training and cross-validation scores. Variance and bias are seemingly mitigated as this point, and the result is a well fit model.

1. Chart, line chart

   Description automatically generatedChart, line chart

   Description automatically generatedAdaBoost hyperparameter tuning. Wine Origin Dataset: Number of Estimators, Left. Learning Rate, Right

Chart, line chart

Description automatically generatedThe final plot below shows a learning curve for the AdaBoost classifier after determining the optimal hyperparameters using GridSearchCV. Because of the modest gap seen between training and cross-validation scores, it could be said that there is clearly some overfitting. However, accuracy of this classifier is increased to 92.35%, which is better than the 89% accuracy our vanilla AdaBoost classifier started with.

1. Boosting Learning Curve: Wine Origin Dataset

## Breast Cancer Dataset

This AdaBoost classifier performs, with a starting accuracy of 98.25%, significantly better on the breast cancer dataset as compared to the wine origin dataset.

### Hyperparameter Tuning

Using the same parameters as descried above, number of estimators and learning rate, what immediately stands out is the difference between plots when using number of estimators. The plot in figure 11 below shows the stating score plateauing at around 25 estimators. There is still a significant amount of bias in the plot, and, as a result, a large gap between the cross-validation and training scores. The learning rate plot below, however, is very similar to the corresponding wine origin plot above. A learning rate between 0 and 2.5 causes a decrease in model performance but recovers where the learning rate is greater than 2.5.

1. Chart, line chart

   Description automatically generatedChart, line chart

   Description automatically generatedAdaboost Hyperparameter Turning. Breast Cancer Dataset: Number of Estimators, Left. Learning Rate, Right

Examining the learning curve in figure 12 below, the training and cross-validation scores do not converge at any time using this set of data. Despite using GridSearch to determine the ideal parameter values for the number of estimators and learning rate, this model would most likely benefit from adding more training data.

Chart, line chart

Description automatically generated

1. Boosting Learning Curve – Breast Cancer Dataset

# classifier 4: support vector machine (svm)

## Wine Origin Dataset

Implementing a Support Vector Classifier (SVC) with no tuning yields a baseline accuracy of 92.50%.

### Hyperparameter Tuning

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedThe first hyperparameter used for tuning this SVC is “C”. The” C” parameter, or regularization parameter, controls the trade-off between achieving a low training error and a low testing error. As can be seen plotted on the left in figure 13, smaller value of “C” means a wider margin, hence a lower accuracy. As the value of “C” increases, this classifier has a narrower margin. This hyperparameter s value ultimately affects the model's ability to classify the training data correctly. As such, it can be observed that a smaller value of “C” results in a higher misclassification rate on the training data, while the larger values of “C” result in lower misclassification rates on the training data. Next, gamma is tuned with different kernel values (RBF, Sigmoid, and Poly). Because gamma affects kernels differently, it was applicable to test the same range of gamma values with varying kernels. Visually, how the kernels interpret the value of gamma differently can be seen in validation curve below (right plot). The validation curve of the RBF kernel and Sigmoid kernel are similar in that scoring is the same up to a value between 10-3 and 10-2. After which overfitting begins to occur for both of these kernels as well. The Poly kernel begins to overfit the data around a gamma of 10-1. What is common with all three kernels tested is that the lower values of gamma produce lower training and cross-validation scores. This is most likely attributed to the underlying relationship between the features and the target variable being non-linear. As a result, the model requires a higher gamma to produce a more complex decision boundary to capture the relationships between features and target values.

1. SVC Hyperparameter Tuning. Wine Origin Dataset: “C”, left. Gamma, right.

# Classifier 5: K-nearest neighbors

# References

1. Dua, D. and Graff, C. 2019. Breast Cancer Wisconsin (Diagnostic) Data Set, Retrieved 2023-01-24 from https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data.
2. C.Blake. Sept 21, 1998. Wine recognition data, Retrieved 2023-01-24 from https://archive.ics.uci.edu/ml/datasets/wine

# Appendices

You may optionally move certain information to appendices at the end of your paper, after the reference list. If you have multiple appendices, you should create a section with a *Heading 1* of “Appendices.” Each appendix should begin with a descriptive *Heading 2;* appendices can thus be referenced in the body text using their heading number and description, e.g. “Appendix 5.1: Survey responses.” If you have only one appendix, you can label it with the word “Appendix” followed by a descriptive title, e.g., “Appendix: Survey responses.”

These appendices do not count against the page limit, but they should not contain any information *required* to answer the question in full. The body text should be sufficient to answer the question, and the appendices should be included only for you to reference or to give additional context. If you decide to move content to an appendix, be sure to summarize the content and note it in relevant place in the body text, e.g., “The raw data can be viewed in *Appendix 5.1: Survey responses.*”