BIFX 553 Midterm

J. Jedediah Smith

2/23/2023

Dr. Bindewald

1. **Performance Evaluation of Machine Learning Models**

Table

Description automatically generated with medium confidence

* 1. What is the number of true positive (TP) predictions?

90 Fragments

* 1. What is the number of true negative (TN) predictions?

70 Fragments

* 1. What is the number of false positive (FP) predictions?

30 Fragments

* 1. What is the number of false negative (FN) predictions?

10 Fragments

* 1. What is the number of positives (P) in the original data set?

100 Fragments

* 1. What is the number of negatives (N) in the original data set?

100 Fragments

* 1. What is the sensitivity (TPR) corresponding to these results?

TPR =TP/TP+FN

TPR = 90/90+10 = 0.90

* 1. What is the specificity (TNR) corresponding to these results?

TNR = TN/TN+FP

TNR = 70/70+30 = 0.70

* 1. What is the accuracy measure of these results?

ACC = TP+TN/TP+TN+FP+FN

ACC = 90+70/90+70+30+10 = 0.80

* 1. There are several ways to condense the quality of a prediction into a single number, one of them being Youden’s Index (or Youden’s J Score). It is defined as the sum of sensitivity and specificity minus one. Compute the J values corresponding to the given confusion matrix.

J = TPR+TNR-1

J = 0.90+0.70-1 = 0.60

1. **Data Preparation** 
   1. What quantities are reported in a structure investigation during data cleaning?

Count of data types, non-numeric data structure, and numeric data shape.

* 1. What is the focus of a data quality investigation?

Duplicated and missing values

* 1. List at least 3 potential steps of actual data cleaning that modifies raw data

Drop rows or columns with lots of missing data.

Drop rows or columns with lots of duplicated data.

Imputation, fill in missing values with best guesses.

* 1. List at least 3 quantities one can typically report during in the content investigation phase.

Discrete features.

Continuous features.

Correlation values.

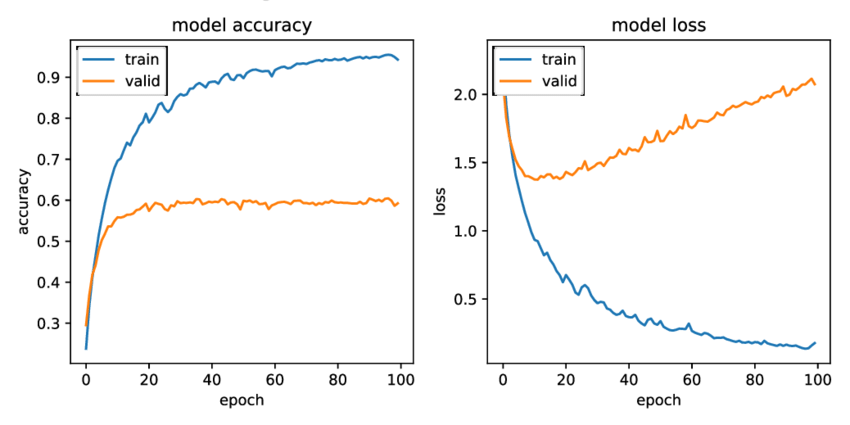
1. **Data Preparation in Pandas**
   1. A Python pandas DataFrame with variable name ‘df’ has a column with name ‘Assembly’ that indicates the name of the genome assembly as a text string. You want to filter out rows where this column has the value of ‘hg19’. Ensure that the row-indices are without gaps after the filtering. Provide the Python code.

df[df['Assembly'] != 'hg19'].reset\_index(drop=True)

* 1. The DataFrame has a second column called ‘Contigs’. You want to find the average value of this column but separately for each different value of column ‘Assembly’. Provide the Python code. (Hint: Consider Pandas methods groupby and aggregate.)

df.groupby(['Assembly']).mean()

1. **Overfitting**



* 1. Take a look at the left chart of above figure that depicts the accuracy of training data and validation data (aka test data) of a neural network machine learning model as a function of training rounds (also called epochs). In your own words, describe why this chart is an example of overfitting.

Model accuracy is very high with the training data, but drops quite dramatically when evaluating the test (validation) data. A large disparity like this indicates that the model is far better at predicting the training data than the test data, textbook overfitting. Ideally, we want similar performance with both groups.

* 1. Take a look at the right chart of above figure that depicts the value of the loss function of a neural network machine learning model as a function of training rounds (also called epochs). In your own words, explain why the loss function values of the training data are getting close to zero and the loss function of the validation set is eventually increasing. [Remember, the loss function is what the neural network is actually trying to minimize during training (and most loss functions cannot return values less than zero), the improved accuracy is just a ‘byproduct’ of that.]

Training loss will trend towards zero as the model establishes relationships between values in the training data. This could be patterns present across the both training and test data or coincidences only found in the training data. Test (validation) loss will trend away from zero when there are too many training only coincidences. Big gaps and opposing trends can be indicative of overfitting. Ideally, we want similar trends with minimal gaps.

* 1. In your own words, list at least two approaches for reducing the problem of overfitting in machine learning.

Train with a larger more varied subset of data. If the training data used does not accurately capture the trends and patterns of the test data, a model will often struggle with overfitting. It is often recommended to train on 80% of your data and save the remaining 20% for testing.

Select for relevant features. If the data contains a large amount of different variables, some may be redundant or functionally unimportant to your analysis. Removing these can decrease noise and improve model fit.

1. **XGBoost**

* 1. In your own words – what are key differences between XGBoost and a Random Forest machine learning algorithm?

Both utilize decision trees, however XGBoost makes them one at a time. Subsequent trees are improvements or adjustments of previous ones. Random Forest creates individual trees that are not connected in such a manner.

* 1. In your own words, what is the meaning of the parameter max\_depth in XGBoost?

It helps limit complexity by determining the levels of depth and maximum number of nodes each branch can have.

* 1. In your own words, what is the meaning of the parameter learning\_rate in XGBoost?

It determines how much changes are weighted between the iterations of decision trees generated.

* 1. List 2 possible ways to reduce overfitting in XGBoost. If mentioning parameter names also mention if their values should be increased or decreased.

Decreasing max\_depth can help reduce overfitting by limiting tree complexity and ignoring finer details that may only be present in the training data. But do not set it to 0 because that represents infinity or no limit.

Decreasing subsample can help reduce overfitting by instructing each tree to only utilize a certain percent of the training data. This way, each tree ends up using a slightly different set of training data.

* 1. For a large dataset, list 2 possible ways to increase the prediction performance of XGBoost based on changed parameters

Adjust the learning\_rate and max\_depth. Bigger will probably be better for a large data set. Setting these values to too small can make it difficult for the model to establish meaningful relationships. But since it is tedious to rerun the cluster to find optimal values, I would recommend using grid search to test multiple combinations of values at once.

1. **Clustering**

* 1. What is the difference between clustering and classification?

Clustering will group unlabeled data together based on likeness. Classification is just the regular sorting of labeled data into particular categories.

* 1. List at least two shortcomings that some cluster algorithms exhibit

With K-Means, results are based on how random initial clusters were.

With DBSCAN, may struggle to handle datasets that have varied densities.

* 1. What approach can one use to identify a “reasonable” amount of clusters in data?

Use the elbow method! Just graph the sum of squares for each potential cluster value and select a value before the big elbow jump on the graph.

* 1. In your own words, how does the DBSCAN clustering algorithm work?

It clusters things based on gaps between data points. If points are a certain distance from each other, they get clustered together. Points too far away from any other clusters are just left as outliers.

* 1. What is the most important parameter in the Python implementation of DBSCAN?

The most important parameter is ‘eps.’ It determines how much space can be between points in order for them to get grouped together.