Introduction to Data Analysis and Machine Learning with Python

Homework 6

Mon Apr 20 2024
Due: April 29 2024

Read the data from the file “TopRight_20230803.txt” that we have used before into a pandas dataframe. Copy the data in the “TimeStamp” and “Pressure” columns into numpy arrays called X and y. This can be done by copying the columns into a pandas series and using numpy.array() to convert the series into an array. We will use X (the TimeStamp) as input value and y (the Pressure) as target value for an AdaBoostRegressor. Use the numpy.reshape(-1,1) to change the data in the X array into the 2D array with shape (n,1) that the regressor fit function expects for the input.

Set up an AdaBoostRegressor like this:

```
regr = AdaBoostRegressor(DecisionTreeRegressor(max_depth=4), n_estimators=300)
```

and train the regressor using X as input and y as target (using fit()). Use the predict() function to predict a new value y2 for each X input.

1. Plot original y vs X and y2 vs X as scatter plots. Do the predicted values follow the target values closely?
   a. Yes
   b. No
   c. Can’t tell

2. Plot the difference d between all corresponding target values y and predicted values y2. What is the shape of this distribution? Approximately,
   a. Uniform
   b. Exponential
   c. Gaussian

3. What is the standard deviation of this distribution? Approximately,
   a. 1
   b. 3
   c. 30
   d. 300
   e. 3000

4. Increase the max_depth of the DecisionTreeRegressor to 10. Repeat the training and prediction, and replot the X vs y scatterplot and the plot of the y2 - y difference (note that the training step may take several minutes). Does the fit improve? What is the standard deviation now?
   a. 1
   b. 3
   c. 30
   d. 300
   e. 3000

5. Read the file “data_HW6.txt”. This contains 20000 entries, with features x1,x2,x3 and class “sig” or “bkg”. Set up an MLP regressor like so:

   ```
   mlp = MLPClassifier(solver='adam', hidden_layer_sizes=(5, 5), max_iter=10000)
   ```

   Using only feature x3, split the data set into 70% training data and 30% test data, fit the MLP regressor to the data and predict the class for each test object. Use metrics.accuracy_score() to check the accuracy of the prediction. Which accuracy do you get using only feature x3? Approximately,

   a. 1%
   b. 25%
   c. 50%
   d. 75%
   e. 99%

6. Now use all 3 features x1, x2, x3 and retrain the same regressor. Does the classification improve? What is the accuracy now?

   a. 1%
   b. 25%
   c. 50%
   d. 75%
   e. 99%