CSC 4780/6780
Fall 2022
Homework 12

November 13, 2022

This homework is due at 11:59 pm on Sunday, Nov 20. It must be uploaded to iCollege by then. No credit will be given for late submissions. A solution will be released by noon on Monday, Nov 21.

it is always a good idea to get this done and turned in early. You can turn it in as many times as you like – iCollege will only keep the last submission. If, for some reason, you are unable to upload your solution, email it to me before the deadline.

Incidentally, I rarely check my iCollege mail, but I check my `dhillegass@gsu.edu` email all the time. Send messages there.

Be sure to rename your solution directory to match your name.

1 Classifying tumors

In the US, about 288,000 cases of breast cancer will be diagnosed this year. The tumors are either malignant (bad) or benign (not bad). The University of Wisconsin has released a dataset with 30 metrics for 570 actual tumors and whether they were malignant or benign.

You will use this dataset to develop a system that predicts whether a tumor is malignant or benign based on these 30 metrics.

Here is where the data is from: `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29`

You will use a support vector classifier with a non-linear kernel to do this.

2 Training

Create a program called `breast_train.py` that reads in `train_breast.csv`. Separate the inputs and target (`diagnosis`) into different numpy arrays.

Make the output boolean: True if the tumor is malignant.

Make a `StandardScaler` and fit it to the data. Use it to standardize the training data.

Use `GridSearchCV` to find the kernel and C that give the best F1 score for `sklearn.svm.SVC`:

- Try the following kernels: Linear, Radial Basis, Polynomial, and Sigmoid.
- Try the following values for: 0.5, 1.0, 2.0, 3.0, 4.0.

Create a new `SVC` with the best parameters. Fit it to all the training data. Print the amount of time fitting took.

Write out the `StandardScaler` and the `SVC` to `classifier.pkl`

Print out the accuracy and confusion matrix for the training data.

It should look something like this when it runs:

```
> python3 breast_train.py
X shape = (516, 30), y shape=(516,)
Best parameters = {'C': 3.0, 'kernel': 'rbf'}
Fitting took 0.003047 seconds with d=30 input.
Accuracy on training data = 98.84%
Confusion on training data:
[[320   0]
 [  6 190]]
```

3 Testing

Create a program called `breast_test.py` that reads in `test_breast.csv`. Separate the 30 inputs and the 1 output into different numpy arrays.

Make the output boolean: True if the tumor is malignant.

Read in `StandardScaler` and `SVC` from `classifier.py`.

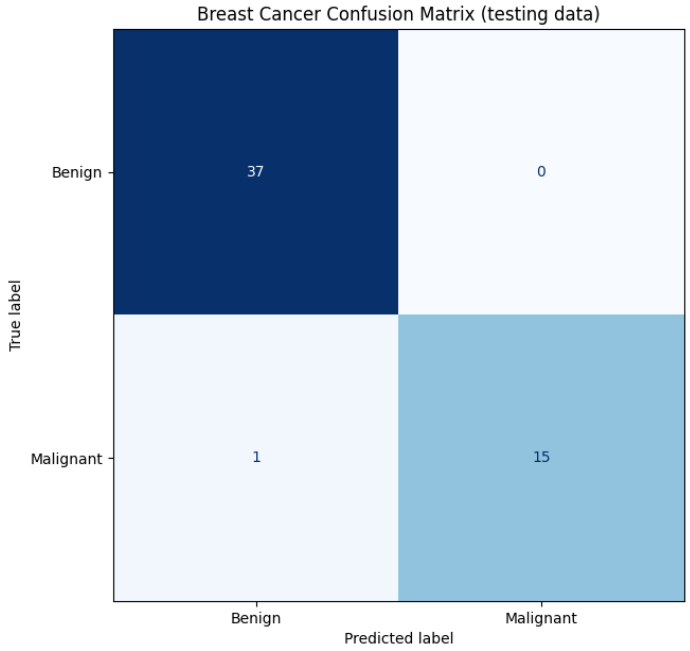
(Do not refit the `StandardScaler`! We are translating/scaling the test data exactly like we did the training data.)

Print out the accuracy and confusion matrix for the testing data. Also make a nice confusion matrix diagram called `text_confusion.png`.

It should look something like this when it runs:

```
> python3 breast_test.py
X shape = (53, 30), y shape=(53,)
Accuracy on testing data = 98.11%
Confusion on testing data:
[[37  0]
 [ 1 15]]
Wrote test_confusion.png
```

And `test_confusion.png` should look something like this:



4 Reduce dimension with PCA

Maybe the model takes too long to fit. (Not in this case, but that is a common reason that PCA is used.)

Let's use principal component analysis to reduce the dimension of the input from 30 to 11. This will make it easier for the classifier, but the change may damage our accuracy a little.

Duplicate `breast_train.py` and `breast_test.py` to `breast_train_pca.py` and `breast_test_pca.py` respectively.

In `breast_train_pca.py`, compute a W matrix that will reduce the 30 dimensional input to 11 dimensions using principal component analysis. Train the SVC classifier using only the 11-dimensional data. Write the scaler, W, and the SVC classifier out to `pca_classifier.pkl`.

Do *not* use sklearn's PCA. Do it explicitly using matrix operations and `numpy.linalg.svd`.

When `breast_train_pca.py` runs:

```
> python3 breast_train_pca.py
X shape = (516, 30), y shape=(516,)
Best parameters = {'C': 2.0, 'kernel': 'rbf'}
Fitting took 0.002278 seconds with d=11 input.
Accuracy on training data = 98.45%
Confusion on training data:
[[318   2]
 [  6 190]]
```

Notice that the fit happens a little faster.

In `breast_test_pca.py`, read the scaler, the W matrix, and the SVC classifier from `pca_classifier.pkl`. Use it to classify the training data. Print the accuracy and confusion matrix. Write out the confusion matrix diagram as `test_confusion_pca.png`.

When `breast_train_pca.py` runs:

```
> python3 breast_test_pca.py
X shape = (53, 30), y shape=(53,)
Accuracy on testing data = 94.34%
Confusion on testing data:
[[36  1]
 [ 2 14]]
Wrote test_confusion_pca.png
```

Notice that the accuracy is a little lower.

5 Criteria for success

If your name is Fred Jones, you will turn in a zip file called `HW12_Jones_Fred.zip` of a directory called `HW12_Jones_Fred`. It will contain:

- `breast_train.py`
- `breast_test.py`
- `classifier.pkl`
- `test_confusion.png`
- `breast_train_pca.py`

- `breast_test_pca.py`
- `pca_classifier.pkl`
- `test_confusion_pca.png`
- `train_breast.csv`
- `test_breast.csv`

Be sure to format your python code with black before you submit it.

We would run your code like this:

```
cd HW12_Jones_Fred
python3 breast_train.py
python3 breast_test.py
python3 breast_train_pca.py
python3 breast_test_pca.py
```

Do this work by yourself. Stackoverflow is OK. A hint from another student is OK. Looking at another student's code is *not* OK.

The template files for the python programs have import statements. Do not use any frameworks not in those import statements.