CSC 4780/6780
Fall 2022
Homework 07

October 2, 2022

This homework is due at 11:59 pm on Sunday, Oct 9. 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, Oct 10.

it is always a good idea to get this done and turn it 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.

1 Logistic Regression

Many data science problems require us to give a "Yes" or "No" answer. One of the common methods we use is logistic regression. In this exercise you are going to split a data set into a training set and testing set. You will fit the model to the training set and test it with the test set. You will toy with the threshold to get different levels of recall and precision.

1.1 Explore the data

You are given `framingham.csv` which is a real data set from a famous study on coronary heart disease: `https://en.wikipedia.org/wiki/Framingham_Heart_Study`

Create a program called `chd_explore.py` that:

- Reads `framingham.csv` into a pandas data frame.
- Uses `pandas_profiling.ProfileReport` to make a report.
- Save the report to `data_report.html`

Open `data_report.html` in a browser and look at the data. You are trying to predict `TenYearCHD`, whether the person will have coronary heart disease event in the next 10 years.

1.2 Split the data

Create another program called `chd_split.py` that

- Reads `farmingham.csv` into a data frame.
- (You know from the report that some of the rows are missing values.) Drops rows with missing values.
- Uses `sklearn.model_selection.train_test_split` to divide the dataframe into two dataframes: the test data frame will be a randomly chosen 20% of the rows, the train data frame will have the rest.
- Save the data frames into `test.csv` and `train.csv`.

1.3 Fit the model to the training data

Create another program called `chd_train.py` that

- Reads `train.csv` into a data frame.
- Divides it into two numpy arrays: Y is `TenYearCHD`, X is the other columns.
- Uses `sklearn.preprocessing.StandardScaler` to standardize the columns of X.
- Uses `sklearn.linear_model.LogisticRegression` to fit the data.
- Prints out the accuracy of the model on the training data.
- Saves the scaler and the logistic regression model to a single pickle file called `classifier.pkl`.

1.4 Test the model on the testing data

Create another program called `chd_test.py` that

- Configure a logger.
- Reads `test.csv` into a data frame.
- Divide it into numpy arrays X and Y, as above.
- Loads the scaler and logistic regression model from `classifier.pkl`.
- Apply the scaler to X so that it is scaled exactly as the training data was.
- Print the accuracy of the model on the testing data.
- Use `sklearn.metrics.confusion_matrix` to print a confusion matrix.

- Try 40 thresholds between 0 and 1. For each one, use the logger to print:
  - The threshold
  - The accuracy
  - The recall score
  - The precision score
  - The F1 score

  Like this: `INFO@14:49:15: Threshold=0.220 Accuracy=0.776 Recall=0.50 Precision=0.32 F1 = 0.393`
- Make another confusion matrix using the threshold that gave you the best F1 score.
- Create a graph of the recall and precision vs. threshold as `threshold.png`

2 Softmax

The softmax function makes several different kinds of multi-class classifiers possible. We need to understand softmax.

Make a LaTeX document called `Softmax.tex`. In that file, answer the following questions. Turn in `Softmax.pdf`

2.1 Compute a softmax

What is the softmax of the vector $[5, 3, 0, -1]$? (Hint: the answer is also a vector.)

2.2 6780 Students only: Compute the Jacobian of the softmax

This one is for the graduate students! Undergrads should read it, but chuckle and move on.

If you have a function $f : R^n \rightarrow R^m$, we can think of that as m functions $f_1 : R^n \rightarrow R$, $f_1 2 R^n \rightarrow R$, ..., $f_m : R^n \rightarrow R$.

The Jacobian of f, then, is the matrix

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

What is the jacobian of the softmax function at $[5, 3, 0, -1]$?

3 Criteria for success

If your name is Fred Jones, you will turn in a zip file called `HW06_Jones_Fred.zip` of a directory called `HW06_Jones_Fred`. It will contain:

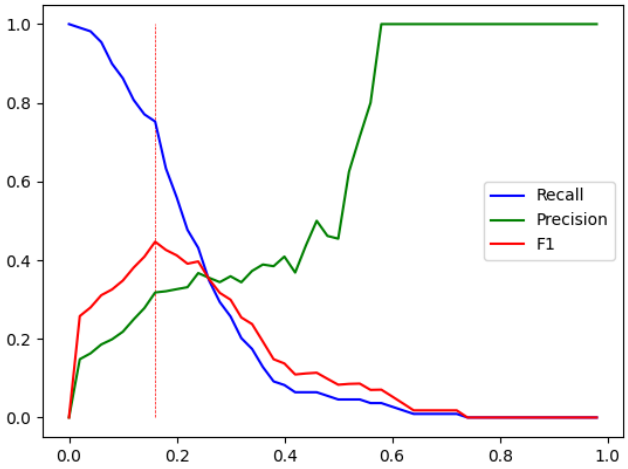
- `chd_explore.py`
- `chd_split.py`
- `chd_train.py`
- `chd_test.py`
- `framingham.csv`
- `train.csv`
- `test.csv`
- `classifier.pkl`
- `threshold.png`
- `data_report.html`
- `Softmax.pdf`

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

We will run your code like this:

```
cd HW07_Jones_Fred
python3 chd_explore.py
python3 chd_split.py
python3 chd_train.py
python3 chd_test.py
```

`threshold.png` should look something like this:



(But, of course, your test/train split is random and probably different from mine. So yours will look a little different.)

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.

4 Reading

You should have read through the cross validation section of Chapter 11 : *Machine Learning for Classification.*

Here's a good video on Logistic Regression: `https://youtu.be/yIYKR4sgzI8`

Here's a video from the same guy on Cross Validation `https://youtu.be/fSytzGwwBVw`