Homework 8

CSC-432

Due: 4/11/13

Pima Indians Dataset

1. **Get the Data**

The Pima Indians Diabetes dataset from UCI at the following URL

http://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/

Use `pandas.read_csv` to load this data into a variable called `pima_data`. Make sure you use header=None (and know why you are doing so). Also, go ahead and name the features and the target using the names argument.

```
1. How many observations are included in this dataset?
2. Give me the means, std, min, and max of the features only.
3. What percentage of people in this dataset have diabetes?
```

2. **Feature Selection**

I have run some feature selection algorithms over the data to find relevant features for predicting the incidence of diabetes. Of the 8 features, $x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7$, I discovered that features 1, 4, 5, and 7 may help the most in explaining diabetes. I used PyFeast to do the feature selection, if you want to explore this yourself.

```
1. What are the selected features?
2. Use ‘pandas.scatter_matrix‘ to plot the selected features,
   differentiating between the outcome classes (diabetes and no
   diabetes) by color.
3. What do the plots suggest to you about the direction of
   correlation between each feature and the incidence of diabetes?
```

Hint: To differentiate between the classes in the plots, use the following code. It assumes that `outcome` is a `pandas.Series` that contains the `class` variable.

```
class_color = list(outcome.apply(lambda x : "red" if x else "blue"))
```

Explain what this step does in a code comment.

Pass this to `pandas.scatter_matrix` using the color keyword argument.

3. **Data Pre-processing**

Before we train a machine learning model, we often must do some data pre-processing, as you have seen in class.

1. Use the ‘pandas.cut‘ function to quantize x_7, as we did in class. Why might you want to do this?
2. ‘Standardize‘ the rest of features using ‘scipy.stats.zscore‘. What does this standardization do to each variable?

4. **Fit a Model to the Data**

I have included two files on github in the notebooks repository – logit.py and perceptron.py. You will want to import the following to answer this question.

```
from logit import logit_loglikelihood, logit_score, logit_func
from perceptron import Perceptron
```

I am assuming that these files are in the same directory as your homework script or otherwise in `sys.path`.

1. Fit a Logit maximum likelihood model to the data as we did in class.

Hint: Do not forget to add a constant to your right-hand side variables. You can use `scipy.optimize.fmin_bfgs` to find the maximum likelihood parameters. You should pass `logit_score` to the keyword argument `fprime`. You can use zeros as the starting values for the optimization.

2. Fit a Perceptron model to the data as we did in class. Train the Perceptron for 500 iterations.
3. Compare the accuracy of the two. What percentage of the observations were correctly predicted? Assume that anything over .5 is a 1.

Hint: For the Logit model, recall that the model is

$$y = g(X\beta) + \epsilon$$

where g is the logit function (`logit_func`). This is exactly the same as finding $\hat{y}$ for the least-squares model, then calling `logit_func` on the result.

5. **Training, Validation, and Testing**

To be sure that we have not over- or under-fit a Perceptron or Neural Network model, we will have to check.

1. Split the data into training, validation, and testing sets, using one of the methods discussed in class and train either the Perceptron or Multi-Layer Perceptron. You will not have to do this for a Logit model, because the likelihood if convex and thus the results are from the unique global extremum.

2. Train the classifiers to achieve the best results that you can. You may use the Logit, the Perceptron, or the Multi-Layer Perceptron. You should be able to achieve at least 75% accuracy. You may work in groups of up to 3 people for this problem. You will take five minutes as a group to present your results in class on Thursday. You may use any features that help. You can clean the data in any way you want. Ie., you do not have to standardize. You do not have to quantize x_7.