Homework Coding Guidelines

In this class, we focus on learning machine learning classifiers and implementing them in Python. It is important that your Python code is clear and easy for us to understand for grading. We would like to require that you follow the following guidelines:

- code should be commented. It should be clear, for example, that a particular piece of code is for question 1, part 1, this code is for question 2 part 2, etc.

- variables should have descriptive names like ”default_probability” not just ”x” and we want to see commented statements like

  ```
  # here we compute default probability and print
  default_probability = ........
  print("default probability is ",  default_probability)
  ```

  as opposed to something like this:

  ```
  x = .....
  print(’result is ", x)
  ```

- functions should have descriptive names that reflect what is being computed For functions, explain the expected input and output in docstrings.

- place code for each question into its own file. Do not create a separate file for each question subpart. If questions use the same code, put the shared code in a separate module (e.g. ”helper.py”) and import this module in your files

- specify the question sub-part in the Python comments for the code that it covers (when possible and appropriate), such as Question 3, Part 2.
- ”print” statements must explain what is being printed
- output data from Python must match your tables
- all computations must be done in Python (not R, Excel, etc.)
- your Python scripts must be submitted together with your data files. Your scripts must assume that data is in the same directory.
- do not use hard-coded path names so we do not need to change to our local path when grading. Use relative path names only. No drop-box or google doc references! You can use something like this

```python
import os
here=os.path.abspath(__file__)
input_dir=os.path.abspath(os.path.join(here,os.pardir))
ticker_file = os.path.join(input_dir, ticker_name + ".csv")
```

If we cannot follow your code, we will not give you credit for it! Use the following ”mental” guideline: if you look at your own code a year from now, would you be able to understand it?