Homework 7 BANA 6620

**Question 1: Logistic Regression**

You should use the LogisticRegression() function of the linear\_model package of sklearn for this question. Your solution might be modeled after the Titanic survival example from the previous lecture. Use the NBA dataset that we have used before in class, and read-in the dataset as a dataframe.

* (a) Define a “good rebounder” as someone who averages more than 8 rebounds per game. (i.e. The corresponding value in the column TRB is larger than 8.) Construct a new column in your dataframe that corresponds to “good rebounder”.
* We want to predict whether an NBA player is a good rebounder based on various predictor variables.
* Use minutes played (MP), position (Pos), points scored per game (PS/G), assists (AST), and steals (STL) as predictor variables.
* (b) Before constructing your model, we want to reduce the number of position categories from 7 to 5. If a player has position (Pos) “PF-C”, modify it to “PF”. Also modify any position “SG-SF” to “SG”. The five distinct positions afterwards should be “PG”, “SG”, “SF”, “PF”, and “C”.
* (c) Remember any categorical data should be represented using indicator variables.
* (d) Use 300 randomly selected rows to train your logistic regression model.
* (e) Form a prediction on the remaining rows of data.
* (f) Print out and interpret the coefficients of your logistic regression model.
* (g) If you named your logistic regression reg, then reg.score(xTest,yTest) computes what percentage of outcomes (in this case “good rebounder” or “not good rebounder” were correctly identified. Use this function to report the accuracy of your model on the test data set.
* (h) Construct 3 hypothetical individuals with the following characteristics:
  + Person 1: SG, 15MP, 10PS/G, 4AST, 2STL,
  + Person2: C, 30MP, 5PS/G, 5AST, 4STL
  + Person3: SF, 35MP, 20PS/G, 3AST, 1.8STL
* (i) Compute the probability that each of the above individuals is a good rebounder.
  + You might first need to vertically stack the data in a 2-D array.
  + Use reg.predict\_proba to help predict these probabilities.

**Question 2: Seaborn**  
a. Use Seaborn to construct a joint plot of “fare” and “age” from the Titanic dataset.

b. Select a stock ticker of your choice that appears on Yahoo Finance and has at least 1 year of historical data. For example, if I chose AAPL, then I should navigate to: <https://finance.yahoo.com/quote/AAPL/history?p=AAPL>

Click on “Historical Data”. On the next page, click on “Download Data”, which will download a .csv file with one year of stock history.

You should choose some stock other than AAPL.

Use Seaborn to plot the adjusted close value of your ticker symbol on the y-axis and the days from the start of the dataset along the x-axis. Your plot should include a Lowess curve.

Below are some syntax examples that could be useful. To make use of these, you may need to adapt the below examples somewhat, and perhaps use a *for* loop. Suppose you call your dataframe aaplData.

aaplData["Date"] = pd.to\_datetime(aaplData["Date"]) #treat column like dates/time

(aaplData["Date"][15]-aaplData["Date"][4]) #compute the time difference between row 15 and row 4 of the date column

(aaplData["Date"][15]-aaplData["Date"][4]).days #compute the time difference, but the output is the integer number of days, rather than a Timedelta object

(aaplData["Date"][5]-aaplData["Date"].min()).days #days elapsed from the “minimum” (earliest) day in the dataset, to the date in row 5

c. For your given stock, compute the slope, intercept, correlation coefficient, p value of slope, and standard error of slope between days from start of dataset (along the x-axis) and stock adjusted close price (along the y-axis). The linregress function of the stats package from scipy may be helpful.