**BIA 6303: Predictive Models  
Homework Assignment #1**

All of Homework Assignment #1 should be done using Python and Jupyter notebooks. You will submit your .html versions of the notebooks. You can either upload the html files individually to Canvas. Alternatively, you can put everything in a zip folder. Please be sure to name your zip folder as follows: Name\_HW1. For example, if I were to submit a zip folder, it would be “XuanPham\_HW1.zip.”

**Notebook 1: Warm up on Pre-processing using Pandas, Matplotlib (and a little Scikit Learn).**

Consider the data collected for a set of cars. This data is available in .csv format in the file auto.csv. Use **pandas** (as well as other libraries as mentioned) to perform the following tasks. You can check the Pandas documentation (http://pandas.pydata.org/pandas-docs/stable/) as well as many on-line blogs for assistance.

Label each section using markup to state what you are doing and what you see.

Your output for the above should look nice and professional AND well annotated. No pages and pages of junk output. Use heads, tails, and contain output so it looks good. A large part of the grade for this section is professionalism in a Juypter notebook.

1. Read in AUTO.csv file into a data frame and call it **Auto**.
2. Replace any missing values with the median value of the variable (feature).
3. Explore the general characteristics of the data, by computing the means and standard deviations of the numerical attributes, as well as the distributions of “drive type” and “fuel type” (both are categorical). You can do this for each individual variable separately or a function to compute for the entire data frame (or both).
4. Use **z-score normalization** to standardize the values of the **Weight** attribute. Show the results side-by-side with the original Weight attribute. [Do not change the original Weight attribute in the table.] Alternatively you may choose to use a [0,1] Min-Max Normalization.You can create this by a manual calculation or use StandardScaler() in sklearn.preprocessing (https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html).
5. Convert the categorical variables into dummies. Note that this requires converting each categorical attribute into multiple attributes (dummies) and assigning binary values corresponding to the presence or not presence of the attribute value in the original record.
6. Perform basic correlation analysis (a correlation matrix) among the attributes. Perform both numeric and visual correlation outputs.
7. Using the new converted dataset, perform a Principal Component Analysis (PCA). How much variation is captured in 3 Principal Components? Here is the documentation for PCA: <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>
8. Using Matplotlib library, create a scatter plot of the (non-normalized) Weight attribute relative to MPG. Be sure that your plot contains appropriate labels for the axes. (You can alternatively explore the seaborn package if you want: <https://seaborn.pydata.org/>).
9. Using the **hist** function in Matplotlib (or Seaborn or both), create histograms for (non-normalized) Luggage (using 6 bins) and MPG (using 7 bins).
10. Perform a cross-tabulation of the two "fuel type" variables versus the two "Drive type" variables. This requires the aggregation of the occurrences of each genre separately for each gender. You can use whatever appropriate data structure you which to store the results, but you can display it as as a 2 x 2 table with entries representing the counts. Then, use **Matplotlib** to create a bar chart graph to visualize of the relationships between these sets of variables (comparing Regular and Premium customer across the two drive types). Your chart should contain appropriate labels for axes.
11. Create a new data frame where fuel is “regular” AND MPG is over 21. Call it Auto\_sub.

**Notebook 2- Practice with Scikit Learn**

**You can refer to my example and the sklearn documentation:** [**http://scikit-learn.org/stable/index.html**](http://scikit-learn.org/stable/index.html)**. In fact, feel free to replicate my Jupyter notebook as needed.**

We are going to perform a basic classification process like you did in ADM but do it with scikit learn.

Label each section using markup to state what you are doing and what you see.

Your output for the above should look nice and professional AND well annotated. No pages and pages of junk output. Use heads, tails, and contain output so it looks good. A large part of the grade for this section is professionalism in a Juypter notebook.

1. Read in the **Churn Calls dataset** (5,000 observations).
2. Call the data frame **Churn**.
3. Set the target variable as Churn (a yes/no variable).
4. Perform some EDA so you get a feel for the data. Make comments on what you see and how it might affect your analysis.
5. Transform the data so factors are dummied and missing values (if any) are fixed.
6. Create a training and test (or validation) set. You can use any split: 90/10; 80/20; 70/30; 60/40; etc.
7. Perform a basic decision tree classifier. Alter at least 2 default arguments. Show your classification report and confusion matrix.
8. Cross validate your decision tree using CV= 10.
9. Perform a KNN=3 classifier. Show your classification report and confusion matrix.
10. Cross Validate your KNN. Comment on what this tells you.
11. Perform another classifier of your choice. Example: logistic regression, random forest, Naïve Bayes, etc. Show your classification report and confusion matrix. Check the scikit learn documentation and Raschka textbook for code examples to deploy your chosen classification model.
12. Which classifier perform the best? Justify your answer.