**Assignment 03 – Random Forest and Stacking -Updated**

There are two steps in this assignment: 1) Training and testing most of the algorithms over the given dataset, and 2) Using the best algorithms (classifiers) to generate a stacked model. You will also export the model in a pickle file, load it again and test it on one of the feature vectors from the dataset. You will submit both the ipynb and pdf files for the ensemble export. Total points = 75 (no extra points in this assignment).

You will use the bike-sharing dataset from assignment 2. Please start with a random.seed(last four digits of your GUIDg). Since this is a regression dataset, for the first part, you will follow the steps in assignment 2 (and some more) as shown below. The code is already with you from the Heroku app deployment section (MLModel.ipynb with regression algorithms, make sure to remove XGBRegressor from the list), and you can use that.

**I have intentionally misspelled some explanations in assignments. When you copy the text, make sure to spell check it. I will drop points for incorrect spelling and grammar (-5 points). Please also provide explanations/ reasoning for utilizing a command and describe the output. Failure to describe/ explain will result in lower points. Also, make sure to add a g#groupnumber (for example, g01) suffix to each variable, so we will know you are not just copy-pasting the given example code.**

# Load the dataset. (2)

Load the .csv file as a pandas data frame and display the first six points.

# Describe pandas Dataframe by using describe. (1)

You will need to make sure that categorical, DateTime, and continuous variables are correctly separated. ‘dteday’ is a ‘datetime’ variable (need to drop it for analysis). ‘holiday,' ‘workingday,' ‘weathersit,' ‘hum,' ‘mnth,' ‘cnt’ (dependent variable, y), ‘hr,' ‘yr’ are categorical variables (I might be missing some of them ).

Use Bike['season'] = Bike.season.astype('category') to convert a variable type.

# Show correlation heat plot of the entire dataset using matplotlib and sns, choose any color pallet (except blue) you like (experiment). (1)

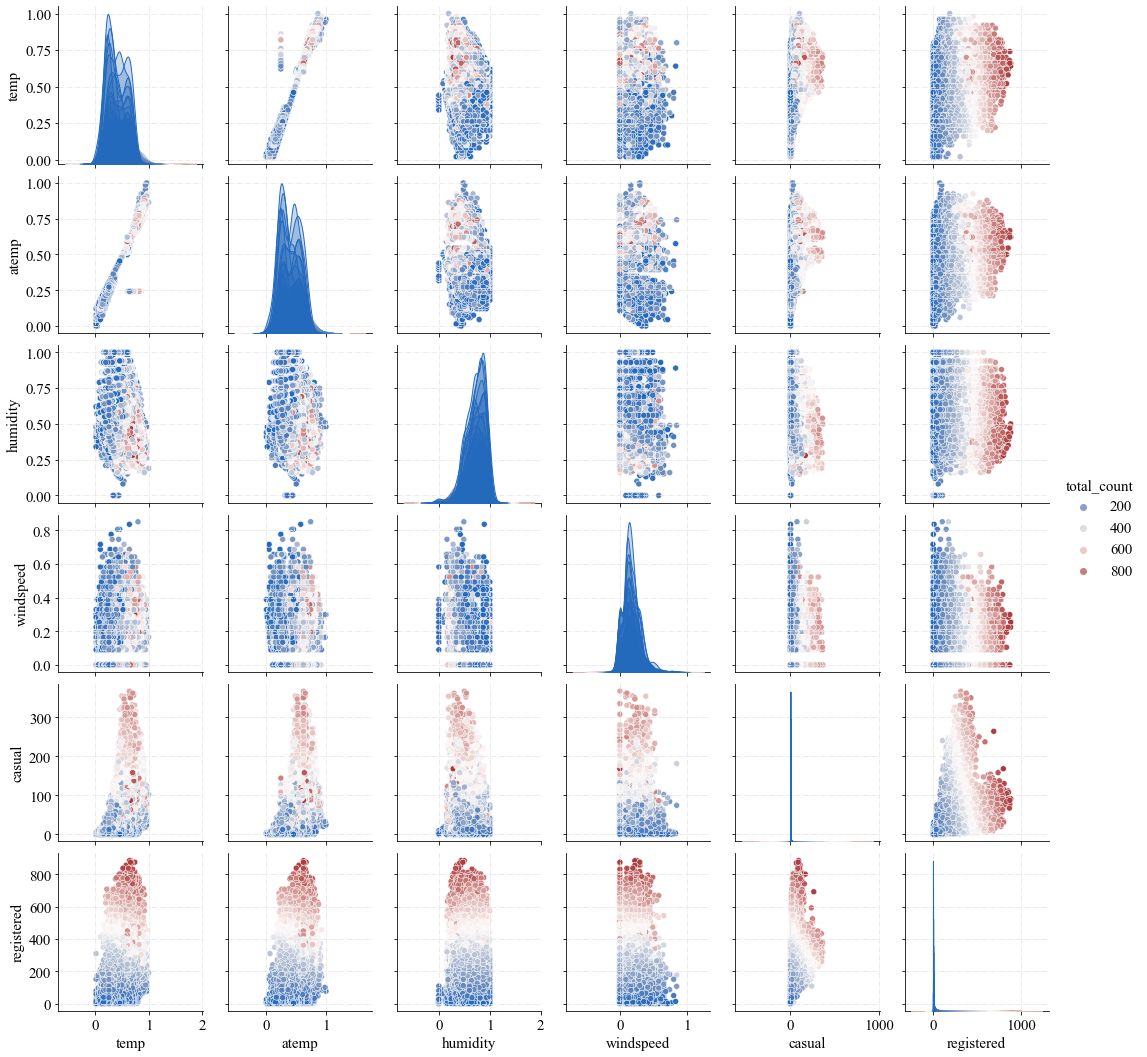
Only plot the continuous variables as corr plot by using

Cormat = DFName[['temp', 'atemp', 'humidity', windspeed', 'casual', 'registered', ‘total\_count']].corr()

And then using sns,heatmap on the cormat above.

# Show the Scatterplot matrix for the dataframe (avoid matplotlib and sns for this assignment). You can use [Scatterplot Matrix Plotly](https://plotly.com/python/splom/). Use the code for the second image that shows different colors for classes. In this case, you will get two colors since we have two classes. Also, experiment with visual aspects of the image (not a lot, but an excellent visual will always leave a better impression. you can change color, thickness, font, font size, font color, etc.). There is no need to explain the plots but save them in a pdf/SVG/png with either static export function or HTML export function from plotly [Interactive HTML Export Plotly](https://plotly.com/python/interactive-html-export/). (10)

When generating the scatter plot matrix, choose color=cnt. This will generate a scatterplot matrix with colors based on the distribution of the variable ‘cnt’. In regression Ensemble it will generate a gradient-based on the value of each dot fig = px.scatter\_matrix(bike\_data, color='total\_count') and it will look like



# Split the dataset into the Training set and Test set. Choose your preferred split and justify the rationale. (1)

Here you will need to be a bit creative and use only the columns that are independent variables. For example, drop DateTime column. You will also need to make sure your categorical columns are ready

# Perform classification routine by using LogisticRegression(), KNeighborsRegressor(), DecisionTreeRegressor(), RandomForestRegressor(), BaggingRegressor(), GradientBoostingRegressor(). Output the accuracy box plot as we have seen in the class. Remember to use the object oriented approach and develop a function (def…), this will be very helpful for the next assignment. (40)

# Select the best regressor for the level o classifier. Use light gradient boosting regressor (LGBMRegressor) regression as a second-level classifier. Similar to 5, generate the box plot and show the accuracy of each algorithm and stacked classifier. Also, show the confusion matrices of the above algorithms. (15)

# Export the Pickle model and import it back. Use the imported model to predict the y\_test from x\_test and report the accuracy. (3)

# The scatter plot of actual vs. predicted values. (2)

Make sure to have axis labels (both x and y) and title on the graph.

