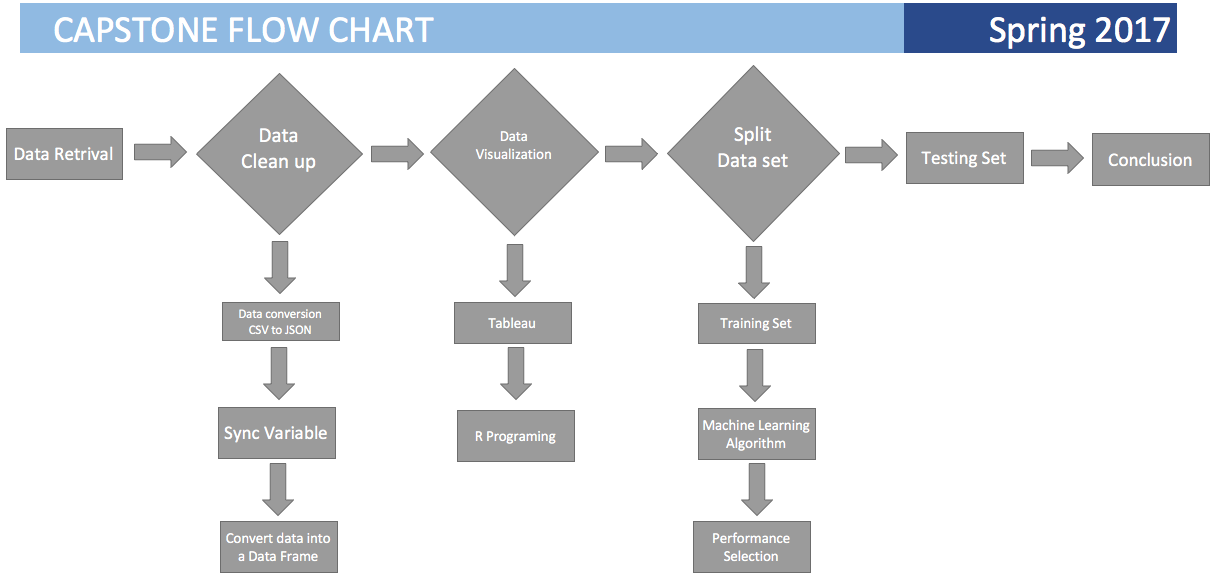
**Rhonald Reside  
DS-670  
Assignment 5**

**Method (Algorithm Overview and Implementation in Zeppelin)**

**FlowchartMethod**

**Data Retrieval and Combining**

The flowchart above is the suggested approach I planned on to complete the capstone. The data set I was assigned was the weather data. The source of the data set is from CityPulse. The website contains a collection of smart city data sets from Aarhus, Denmark. On the CityPulse Smart City website, the weather data set is available in two types of format. One format is in .tar format and the other format is in JSON format. The data is separated by two sets of dates. The first range of the weather data set is February 2014 to June of 2014. The second set of the weather data set is in August 2014 to September 2014.

I was first interested in working with the .tar format file. Since this is my first exposure to .tar, I was really interested in seeing its data structure and content. At first I tried to research what programs or tools can read this .tar file. Unfortunately, I was not successful to open the .tar file. From my research on how to use .tar, it seemed to be just a proprietary file only readable to certain machines. As much as I wanted to work with .tar, I had to abandon this format. It was difficult for me to find a package in R or Python that can work .tar.

**Data Clean Up**

The alternative data files that the weather data set came in was JSON file format. Weather data set came with seven different types of files (i.e., variable). It came in with Dew Point, which was in degrees Celsius. Humidity was the next variable which came in percentage. Pressure was the next variable which came in the measurement of mBar. Temperature was the next variable which was measured in degrees Celsius. Wind direction was the next set of variables which was came in the measurement of degrees. The next variable is wind speed, which was measured in kilometers per hour. The final variable was visibility

I decided to use excel to first look at the six files of variable. I figured it was easier for me to clean and structure the data in excel. Once you opened each file, you discovered that each measurement was also time stamped. I spent about 3 hours to clean up and turn the six files in a data frame. My goal was to combine all the six variables into one csv file. This way I can use several tools to study the data so that I can come up with a good solid story. While I was creating the file in excel, it was simple for me to convert the Celsius into Fahrenheit. I used the formula Temperature in Celsius multiplied by nine fifths and adding 32 ( x .

Once I had all my variable converted into a data frame, I discovered that the time stamp for each variable was perfectly aligned. This made it very easy and clean to find a measure at a specific time. In other words, if I wanted a measurement on February 2, 2014 at 1:00 AM, I had an exact measurement of temperature, dew point, humidity, pressure, wind direction and wind speed and visibility. I used excel to resave the JSON file as a csv. I knew once I got to this point that it would be easy to move the data into different environment to study and find any patterns that would help my story for the data.

**Labeling of Dataset**

My next step after my conversion from CSV to JSON, was to understand my data set and see if I can discover any patterns or gaps in the data. I decided to use Tableau for this step. Even though the beauty of Tableau is that you blend your data from any format, the steps I took in excel to convert the data into a csv file made working with the data easier in Tableau. I was able to create a database in Tableau just for the weather data.

I find it easiest to visualize my data after it was loaded into Tableau. I created line graphs, heat maps and averages to help me build a story from the data. I discovered a few things after this step using Tableau. First, I found that there was a data gap in from June 8, 2014 to August 1, 2014. I could see this once I created a line graph with temperature, dew point and humidity. It was a very interesting visual find. I suspect that since this this data gap occurred throughout all the variable, there was most likely the possibility of data storage issue or they all ran out of battery at the same time.

The next discovery I found was the warmest day and the coldest day in the data set. I found that August 2, 2014 was the warmest day measured at 80.60° F with a humidity of 37 and a dew point of 11. The coldest day occurred March 11, 2014 with a humidity of 84 and a dew point of -4. I decided to create a heat map that represented the high temperature all the days. I also did the same for the cold and created heat map of the coldest temperature day

Next I loaded all the data in R. By doing so, I found one of the variables to be incomplete. The visibility variable only ranged from February 2014 to June 2014. I therefore am deciding not to use this variable. At this point it would not matter because visibility variable does not really affect relative humidity.

**Splitting Data Set**

I feel needed to apply complex statistic or machine learning to the project. R programming language made this step easier. I decided to first split my data into two data sets. I created a training data set and a test data set. This is an important step because you can run your statistics on your training data set. I hoped at this point that the statistics I have a very low error rate so that I can beat my competitive article. Just as a reminder, my competitive article has an error rate lower then 5%. The other point I would like to mention is that no matter how low your error rate is on your training data set, you expect your testing data set error rate to be slightly higher or at least very close to your training data set error rate. If your error rate for your testing data set is lower, then your training data set is not a good representation of your total data set.

I plan to use logistic regression as part of my analysis. Even though we can see there is a relationship with temperature, dew point and humidity when graphing in Tableau, I would like to see a statistical number that show this correlation. So when in R programming, I use the library ISLR for my statistical analysis. I plan to use temperature as my Y-Variable, also known as my dependent variable. I label my variables as dew point and humidity; we would also know this as independent variables.

From the ISLR library, I use the glm() function for the logistic regression model. I first use a binary response to tell me if my temperature is a hot day or a cold day. I use 65° as the decision variable. If the temperature is above 65°, then my algorithm will lable the temperature hot. If it is lower then 65°, then the temperature is labeled cold. By labeling my temperature inputs, the glm() function will create a confusion matrix. A confusion matrix is a table used to describe the classification performance of my model of my training data set. In this case, it shows us the accuracy of our classifier of 65°.

The generalized linear model us useful for predicting an outcome from a binary response from a data set. It is sometimes called discriminant function analysis because its assumptions are less restrictive. Our formula for the data set is

logistic\_model = glm ( Yresults ~ tmpm\_Fahrenheit + Dewptm + hum + Pressurem + wdir ,   
 data = data1 ,   
 family = binomial   
 )

In the formula above, we are calling our function a logistic\_model. Inside our glm() function, we have our independent variable and dependent variables, our data set, and binary response. Again, Yresults is our temperature. Our independent variable is the rest of our variables , , ,, and (i.e. temperature, dew point, humidity, pressure and wind direction.) Data is our training data set. Family equals binomial is our binary response to 65°.

We fit our data to model the glm() function. As an outout, we see the variable in the usual way and a receive a binomial error distribution. The estimators of the coefficient show us weight influences in a positive manner while displacement is negative. The goal is to calculate the predicted probability of temperature for the specified values of our predictors, i.e. dew point, humidity, pressure and wind direction.

**Conclusion**

I predict that there will be a strong correlation with humidity and dew point and weather. My hypothesis is that our output will agree with our competitive article. I think the glm() function will have lower error rate then our competitive article therefore making our model a better choice for relative humidity correlation. Our competitive article believes that you can lower your error rate when fitting the data without the help of computers. I believe outherwise, that it is very necessary to use computers to predict. Using computers will reduce your error.

**Pseudo Code**

Summarize our method with use of code.

1. Download data
2. Clean and structure data. Converting JSON file to CSV
3. Visual analysis of data using Tableau and R programing. Graph the variables, i.e. temperature, dew point and humidity. Create heat maps of the cold temperature and warm temperature.
4. Use R programing to create statistics with the weather data. This can confirm the visual findings found from Tableau. We can confirm our warmest temperature, our coldest temperature, and finally the relationship between temperature, dew point and humidity.
5. Split the weather data set in R. I would be creating a training data set and testing data set. Splitting our data would be important when we cross validate and evaluate our glm model.
6. Fit our data into the glm() function.
   1. Binary response of 65. We classify our temperature if its hot or cold.
   2. Output from the glm() function will include:
      1. Error rate
      2. Z-value
      3. Weight of variable
7. Evaluate the model. Understand the output from the glm() function
8. Apply model to the test data set.