Case 1: Energy Forecasting

Instructions:

- You can use either Python or R to work on this case.
- No sharing of work. You can work in your teams only
- You are expected to submit a report that summarizes the key steps in your implementation as a flow chart and also submit fully functional code.
- Deadline: 6/17/2016 11.59 PM. Late submissions lose 15% points per day.

The city of Boston has hired you to build a forecasting model to predict their energy usage. In order to do so, you are planning to use multi-linear regression to model Power usage as a function of multiple variables (temperature, day of week, month, weekday, hour of day etc.). You are expected to work on this in three parts:

Part 1: Algorithm implementation

1. Data wrangling and cleansing

The raw data you have been provided is real data collected in 2014 at the Jeremiah Burke school. See **RawData.csv.** You will need to clean the data and modify the format in the format given to you. (**sample format.csv)**

Note a couple of things:

1. The following are derived from the RawData.csv file

• kWh	Kwh is aggregated hourly. Sum of 12 observations (5 min intervals rolled up to hourly)
• month	1-12 => Jan-Dec – Derived from dates
• day	1-31 – Derived from dates
• year	Derived from dates
• hour	0-23 – Derived for each record corresponding to the hour of observation
• Day of Week	0-6 – Mon-Sun – Derived from dates
• Weekday	1- Yes 0- No – Derived from dates
• Peakhour	7AM-7PM – 1 ; 8PM-7AM – 0

2. You will have to get temperature data for Boston (KBOS) for the time period. For that review
   a. https://www.wunderground.com/weather/api/d/docs
   b. https://cran.r-project.org/web/packages/weatherData/weatherData.pdf
   c. https://straymarcs.net/2014/12/how-to-create-your-own-weather-forecast-program-using-python/

Dummy data was added to the sampleformat.csv file. You will need to get the actual hourly data from wunderground.com and merge it to the dataset. You should have a script that can take any input file in the format RawData.csv and convert it to the output format (sampleformat.csv). This should include the weather data download and merging part. No manual intervention should be required.

2. Multiple-Linear Regression

Implement a multiple linear regression using the cleansed data as the input (sample format.csv). Review the different concepts we discussed in class (feature selection, feature transformation etc.) and create the best possible implementation. Evaluate the performance metrics and create a model that produces the best possible Regression coefficients you can.

You should have a script that can take any input file in the format sample format.csv and outputs the regression coefficients into a text file. The script should also compute the RMS error, MAPE and MAE for your model and output to a text file. (See **RegressionOutputs.csv, PerformanceMetrics.csv** for formats)

Note: Temperature must be one of the features.

3. Forecast

You are given a sample **forecastData.csv** file with temperature data. You should write a script that does 2 things:

a. Convert the format to the format in forecastInput.csv
b. Use the regression inputs generated in step 2 and the forecastInput.csv to predict the power usage in KWH for each hour.

Part 2: Running your model (Verification)

On 15- June, you will receive 2 files for Validation

1. A raw data file from another school (similar to the raw data.csv file). Your code should take this file and the station code (for example KBOS) and run steps 1 & 2 above and generate the regression coefficients and performance metrics for this new file.
2. A new forecastData.csv file. You should use the script you generated in step 3 and predict the power usage (See **forecastOutput_Account_26435791004.csv** for the expected output format)

Submissions:

Report with flowchart and design of regression model and comments on evaluation metrics

Jermiah Burke school:

Outputs from:

1. Part 1: Data cleansing in the format **sampleformat.csv**
2. Part 2: Regression output files in the format **RegressionOutputs.csv, PerformanceMetrics.csv**
3. Part 3: Output from your forecast script in the format **forecastOutput_<Account No>.csv** (Example: forecastOutput_26435791004.csv)

Repeat this for the new school data provided on June 15th