Homework 7 - K Nearest Neighbors (KNN)

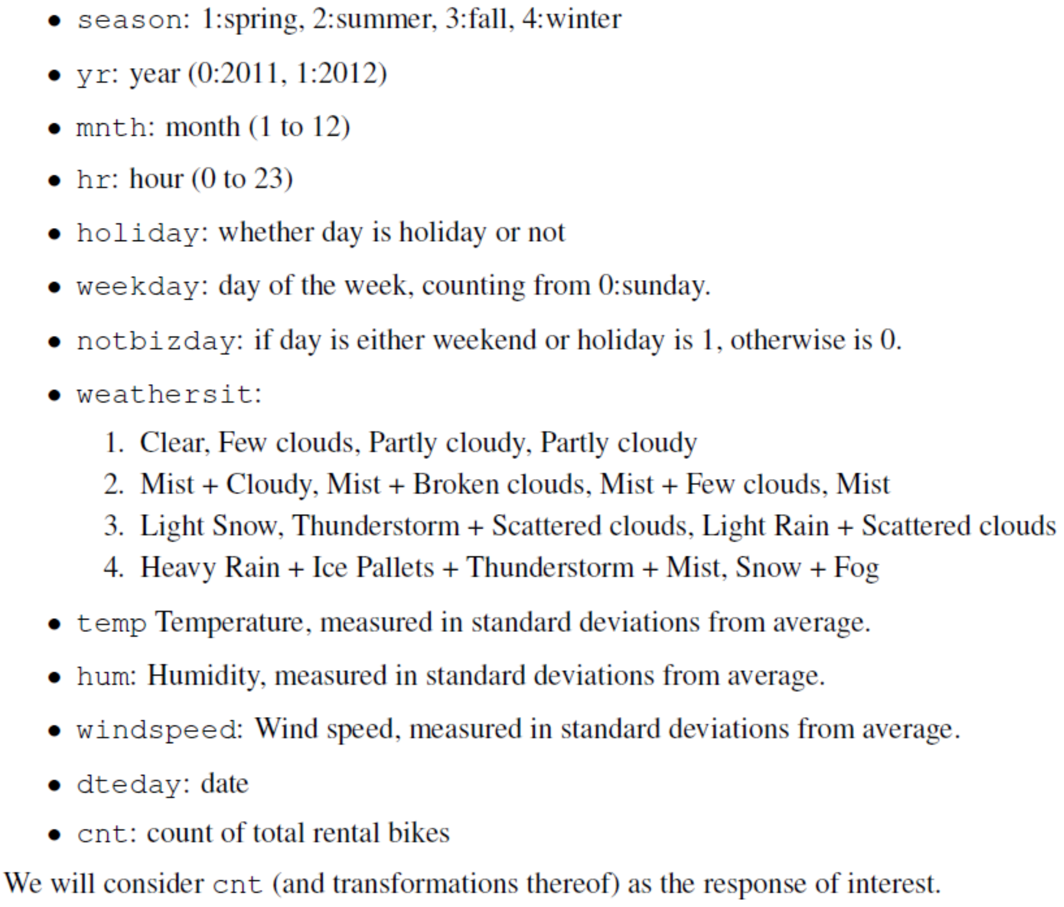
Washington DC Bikeshare Data (155 points)

Instruction:

- This is a group-work assignment!
- You are expected to submit the **.ipynb** file and the exported **.html**.
- Only one member in each group needs to submit the assignment. It will be automatically submitted for the rest of group members.
- This is a long assignment, start early!

Question 1 KNN Regression (50 points)

In this exercise I want you to apply KNN regression model to the bikeshare data set which is available on the GitHub folder for HW7. The data are 17379 observations of hourly counts from 2011 to 2012 for bike rides (rentals) from the Capital Bikeshare system in Washington DC. It was originally compiled by Fanaee and Gama in ‘Event labeling combining ensemble detectors and background knowledge’ (2013). Import the bikeshare.csv as a data frame and name it as df. bikeshare.csv contains:



- `season`: 1:spring, 2:summer, 3:fall, 4:winter
- `yr`: year (0:2011, 1:2012)
- `mnth`: month (1 to 12)
- `hr`: hour (0 to 23)
- `holiday`: whether day is holiday or not
- `weekday`: day of the week, counting from 0:sunday.
- `notbizday`: if day is either weekend or holiday is 1, otherwise is 0.
- `weathersit`:
  1. Clear, Few clouds, Partly cloudy, Partly cloudy
  2. Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
  3. Light Snow, Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
  4. Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
- `temp` Temperature, measured in standard deviations from average.
- `hum`: Humidity, measured in standard deviations from average.
- `windspeed`: Wind speed, measured in standard deviations from average.
- `dteday`: date
- `cnt`: count of total rental bikes

We will consider `cnt` (and transformations thereof) as the response of interest.

I specifically want you to do the followings:

1. Drop the dteday variable and then define your feature space and target variables. Split the data into test (20%) and train set (80%) (5 points) **(5 points)**
2. From sklearn.neighbors import the relevant function for KNN regression. Do the followings: **(25 points)**
   1. Train all the model with the default features. (5 points)
   2. Make predictions on the test set and save them as y_hat (5 points)

   3. Construct a data frame named df_predictions with 2 columns. y_test, and y_hat from previous part (5 points)
   4. Visualize actual vs predicted counts in the test set using an scatterplot. Are you visually satisfied with the regression model? (5 points)
   5. Report the RMSE_test for the KNN regression model. (5 points)

3. Cross validation: **15 points**
   1. Estimate the RMSE_test by doing a 5 fold cross validation on the train set and name it as RMSE_CV. (5 points)
   2. Plot the RMSE_CV vs K and find the optimal value for K in the KNN regression model. (10 points)
4. Report the R-squared in the test set with the optimal k for KNN. Compare this number with R-squared in the test set using multiple regression model. Which model is performing better in the test set? **(5 points)**

Question 2 KNN Classification (105 points)

The managers of Capital Bikeshare have found that the system works smoothly until more than 500 bikes are rented in any one hour. At that point, it becomes necessary to insert extra bikes into the system and move them across stations to balance loads. Do the followings:

1. Define a binary target variable overload. $Overload = 1$ if $cnt > 500$ and 0 otherwise. What are the proportions of overload vs non-overload in your data set? Is the target variable balanced or imbalanced? **(5 points)**
2. Along with the target variable, define your feature space (X) and split the data into test (30%) and train set (70%) **(5 points)**
3. From sklearn.neighbors import the relevant function for KNN classification. Do the followings: **(25 points)**
   1. Train the KNN classification model using its default parameters. (5 points)
   2. Generate the predicted probabilities and predicted classifications and save them as y_hat_probs, y_hat respectively. (5 points)
   3. Plot the histogram of y_hat_probs? Explain what you see? Is there a probability threshold at which the model always predict negative or positive? (5 points)
   4. Generate predicted classifications for two different thresholds (30% and 70% threshold). Save these new predictions as y_hat_30 and y_hat_70. Which threshold should you use if your goal is to avoid too many false negatives? Explain your answer. (10 points)
   5. Construct a data frame named df_predictions with 5 columns. y_test, and the 4 y_hats from previous parts (5 points)

4. Borrow my_KNN_report() function from the python notebook of class 13 and do the followings **(25 points)**
   1. Report the Accuracy, precision, recall and f1 score along with the confusion matrix for threshold =0.5. Interpret all these statistics. Do you trust the accuracy of the model? why? (15 points)
   2. Now use threshold = 0.3 in the my_logistic_report() function. what happens to accuracy, precision, recall and f1 score? what happens to false negatives? is this consistent with you answer to question 5.4? (10 points)
5. Plot the ROC curve and report the AUC score. Is your model doing a better job than random prediction (no skill)? **(10 points)**
6. Cross validation: **(15 points)**
   1. Estimate the error_rate_test by doing a 5 fold cross validation on the train set and name it as error_rate_CV. (5 points)
   2. Plot the error_rate_CV vs K and find the optimal value for K in the KNN classification model. (10 points)
7. As the manager of Capital Bikeshare, you are dealing with a trade off between unexpected overload cost and cost of idle bikes. If the cost of a single idle bike is smaller than the cost of a single unexpected overload, then which of the following probability thresholds would satisfy your objective? 0.3, 0.5 or 0.7? **(15 points)**
   Hint: idle bike = False overload and unexpected overload = False non-overload

Good luck and enjoy machine learning!