CS419M (Autumn 2018): Assignment 1

This assignment is due by **14/08/2018**. The submission portal on Moodle will close at **11:55 PM**. There will be a 10% reduction in marks for each day in case of late submission.
Please read the following important instructions before getting started on the assignment.

1. This assignment should be completed individually or in a group of 2.
2. The programming assignment is hosted as a Kaggle competition. Click here for further instructions on how to access Kaggle.
    1. If you don't have a Kaggle login already. Go to the Kaggle website.
    2. Create a new login using your roll number/GPO ID. Your assignment **will not be graded** if IITB roll number is not used as username.
    3. Details of Kaggle competitions are available here and here.
3. Your final submission should be a .tgz bundle of a directory organized exactly as described here.
   The directory should be submitted as a .tgz file using the command:

   ```
   tar -czf submission_roll_number.tgz submission_roll_number
   ```

   ```
   submission_roll_number/                (example:
   submission_150123456)
       +--report.pdf                      (contains plots/numbers)
       +--dataset_toy/
       |   |   +-train.py/train.cpp    (one of train.py or train.cpp
   depending on whether your code
       |   |                            is in Python or C++,
   respectively.)
       |   |   +-infer.py/infer.cpp
       |   |   +-output.csv
       |   |   +-model                 (use any format to save your
   model)
       +--dataset_kaggle1/
       |   |   +-train.py/train.cpp
       |   |   +-infer.py/infer.cpp
       |   |   +-output.csv
       |   |   +-model
       +--dataset_kaggle2/
       |   |   +-train.py/train.cpp
       |   |   +-infer.py/infer.cpp
       |   |   +-output.csv
       |   |   +-model
   ```

   Python 3.4 or above should be used. For C++, compile using g++ --std=c++11. The complied file in C++ should be with same name and .o extension.
4. The file train.py/train.cpp should create a decision tree model which should be saved in place of model in the above mentioned directory structure. This saved model should be loaded by infer.py/infer.cpp. The file infer.py/infer.cpp should write the output to output.csv as mentioned on Kaggle submission page.

Programming

Implementing a Regression Tree

In this assignment, you will be implementing a regression tree from scratch. Each decision node will correspond to one of the features in `train.csv`, which is selected by choosing the feature that minimises loss. For this assignment, you will be experimenting with two loss functions viz. mean squared loss and absolute loss. The files `train.py/train.cpp` and `infer.py/infer.cpp` should accept the command line argument `data_file`. Your code should also accept two command line arguments `mean_squared` and `absolute` as (example for python) `python infer.py --data_file data.csv --absolute`. The features in `train.csv` and `test.csv` are either continuous-valued or discrete-valued. Details about data attributes are given on Kaggle competition page.
The task for this problem is hosted on Kaggle. Please go to competition1 and competition2, after you have created an account on Kaggle using your roll number and GPO ID. Competition1 is public. Please use this link to participate in competition2. You can download this small toy dataset to save time on computation during intial experimentation.

A. Use either Python or C++ code to implement your Regression Tree. It should accept argument `data_file` which is used to pass the data in CSV format as given on Kaggle competition webpage. Inference should be implemented using `infer.py/infer.cpp` and output should be stored in `output.csv` as already mentioned above. **Report this best loss values to `report.pdf`**.
B. Build a complete regression tree using your training samples in `train.csv`. Prune the decision tree as discussed in the class. You can use 1-fold cross validation. Plot the graph between `loss` and `number of nodes` in the regression tree. Note that you will have two graphs, one for absolute loss and one for squared loss. **Report this graph to `submission/report.pdf`**
C. Predict the output on second dataset given here and modify the model to optimize for this bigger dataset. **Report the best loss value to `report.pdf`**
D. Training time and inference time should also be mentioned in report. Report should also contain a brief note on implementation and any extra experiments or modifications you wish to emphasize.