Air crew problem, air05, binary programming problem

* 7195 decision variables
* 426 constraints

After change binary constraints to continuous constraints, time to solve LP problems with R package “Rglpk”, Rglpk is based on C++

* Solve 600 LPs: 1058s,

When training the prediction models, errors happen

Error: protect(): protection stack overflow

It seems there are too many predictors.

Open R with the following option, so that the stack space is increased

R --max-pp-size=500000

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Solver | Nearest Neighbor | Comment |
| 3-nearest neighbor | 86.18s | 22.79s | Solve 100 LPs |

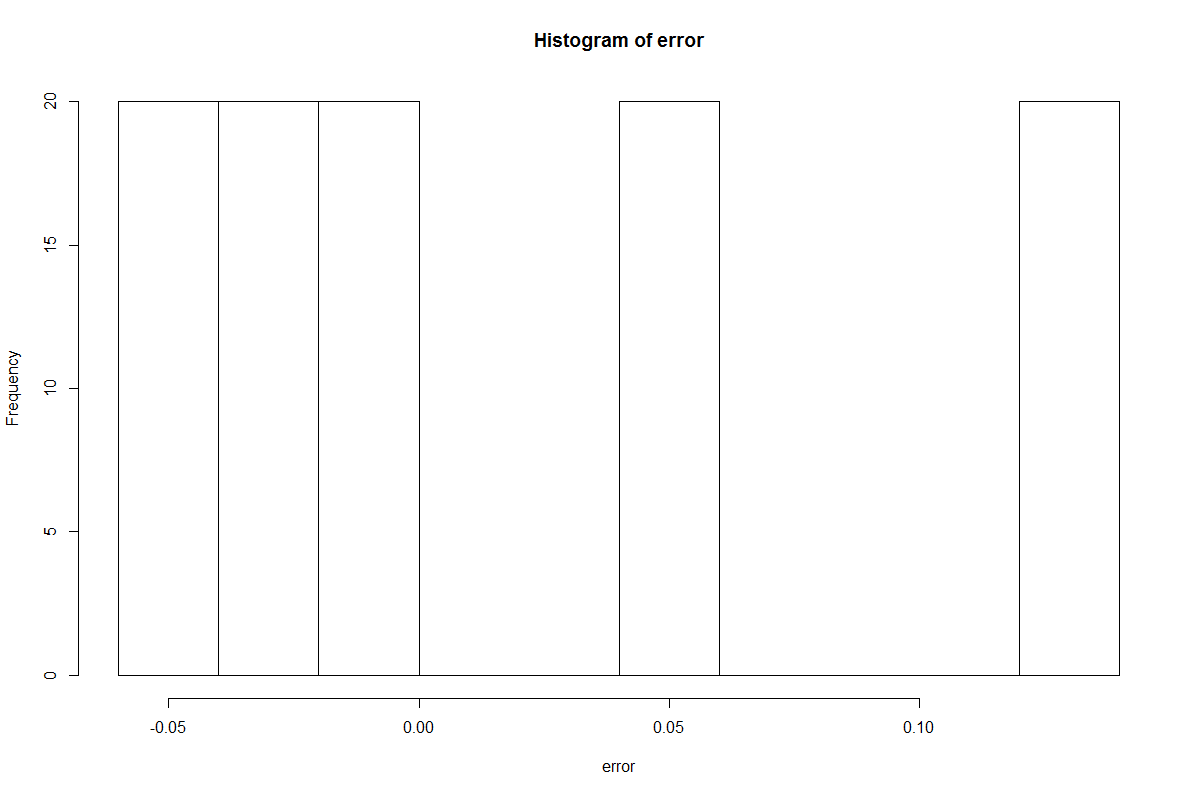


Figure 1: Objective Value Change

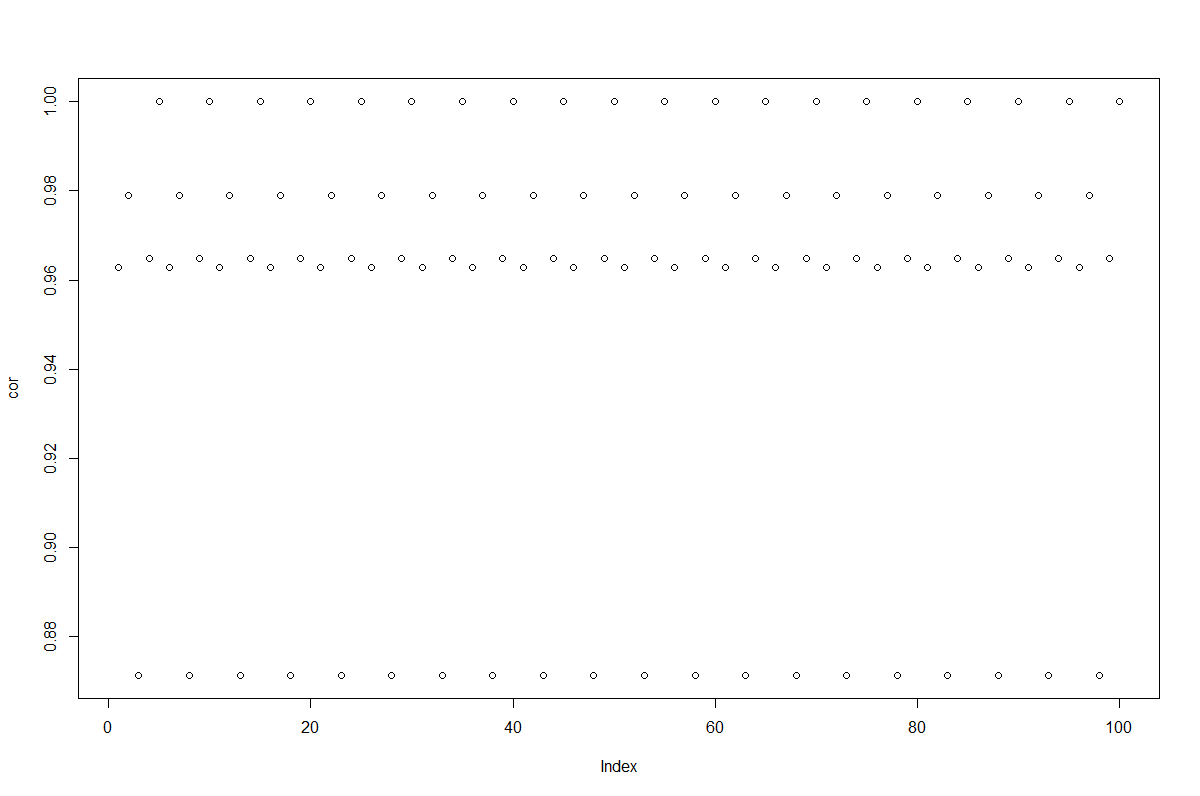
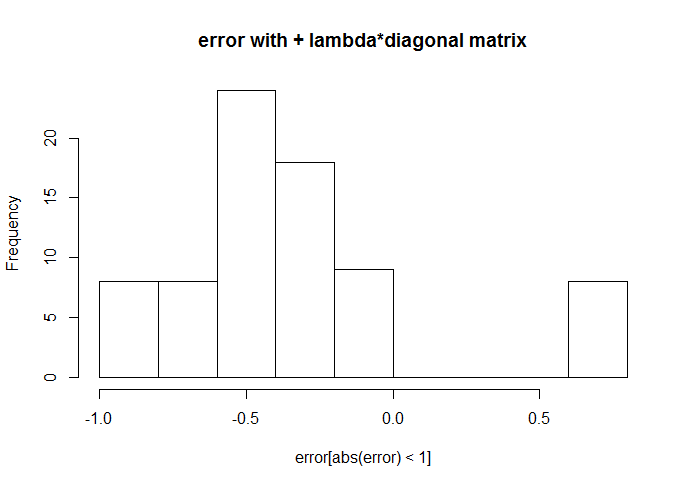


Figure 2: Correlation between solutions from solver and prediction

**Using**



P0033.mps problem

* 33 decision variables
* 16 constraints

Rglpk solve LP problems, 0.87 seconds for 600 LP problems.

Speed comparison, solve LPs

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Solver | Prediction | Comment |
| 1 gbm | 0.12s | 10.89s |  |
| 1 gbm | 12.98s | 1.5s | Solve 10,000 LPs |
| 2 glm | 0.12 | 11.16s |  |
| 3 xgboost | 0.12 | 3.06s | Predict 1 by 1 |
| 3 xgboost | 0.12 | 0.46s | Predict 100 LPs simultaneously |
| 3 xgboost | 12.98s | 1.05s | Solve 10,000 LPs |
| 4 nearest neighbor | 0.12s | 0.19s | Solve 100 LPs one by one |

Discussion about the speed.

* Rglpk is based on C++, gbm is based on C++, glm is based on R, xgboost is based on Python
* when number of LPs increases, time of solver increases much while prediction’s time increases a little.
* Inverse of matrix is not considered here. Since Dr. Banerjee told me it is necessary to compute the inverse once.

Precision of LP, LP solution error (based on objective value)

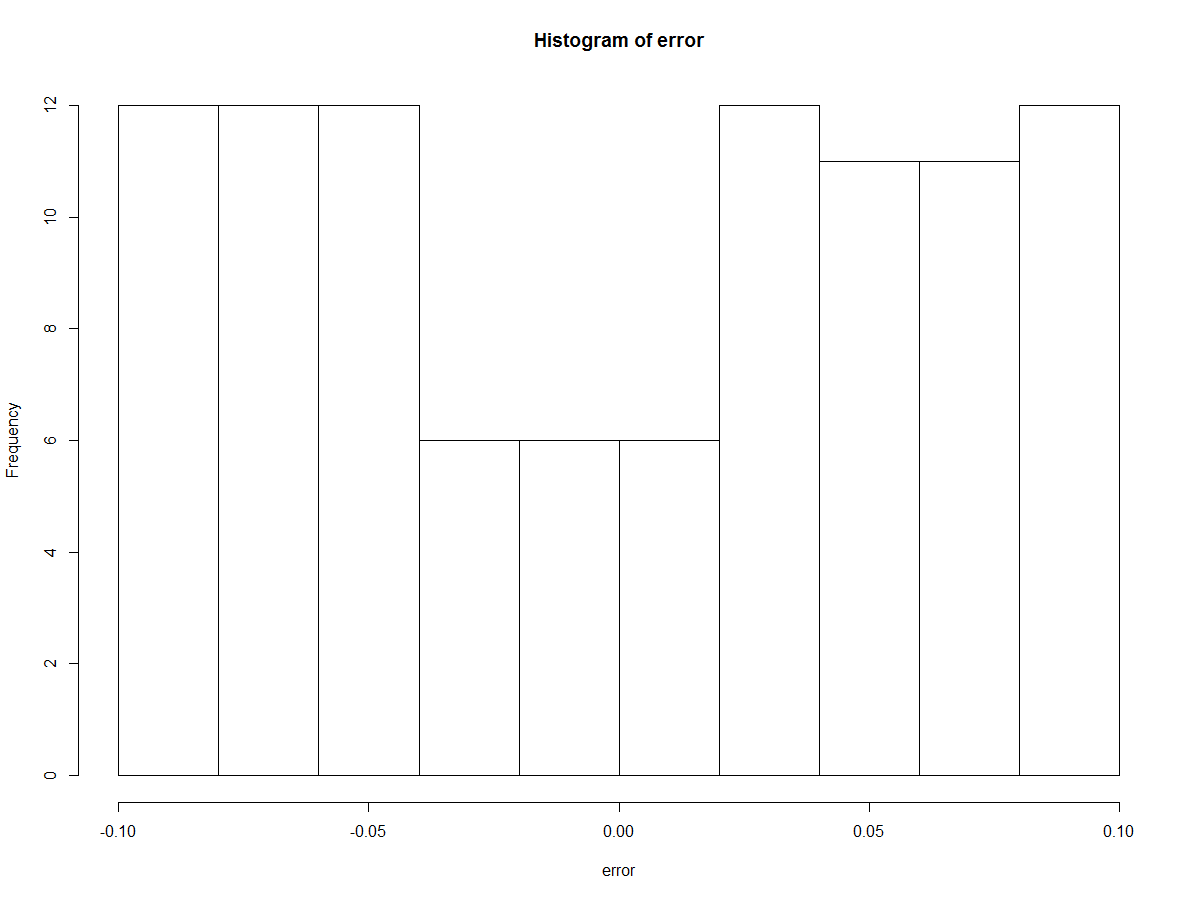


Figure 3: GBM error

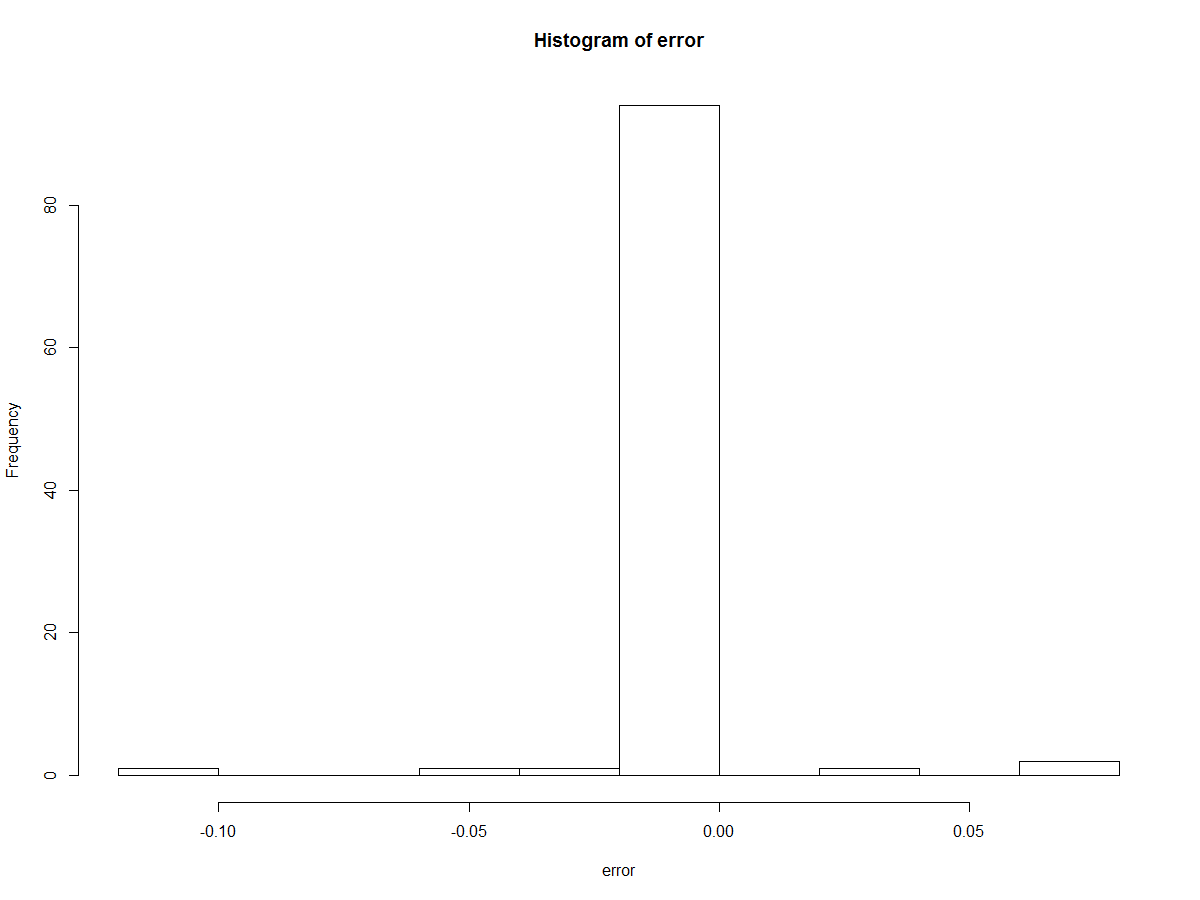


Figure 4: 1 Nearest Neighbour