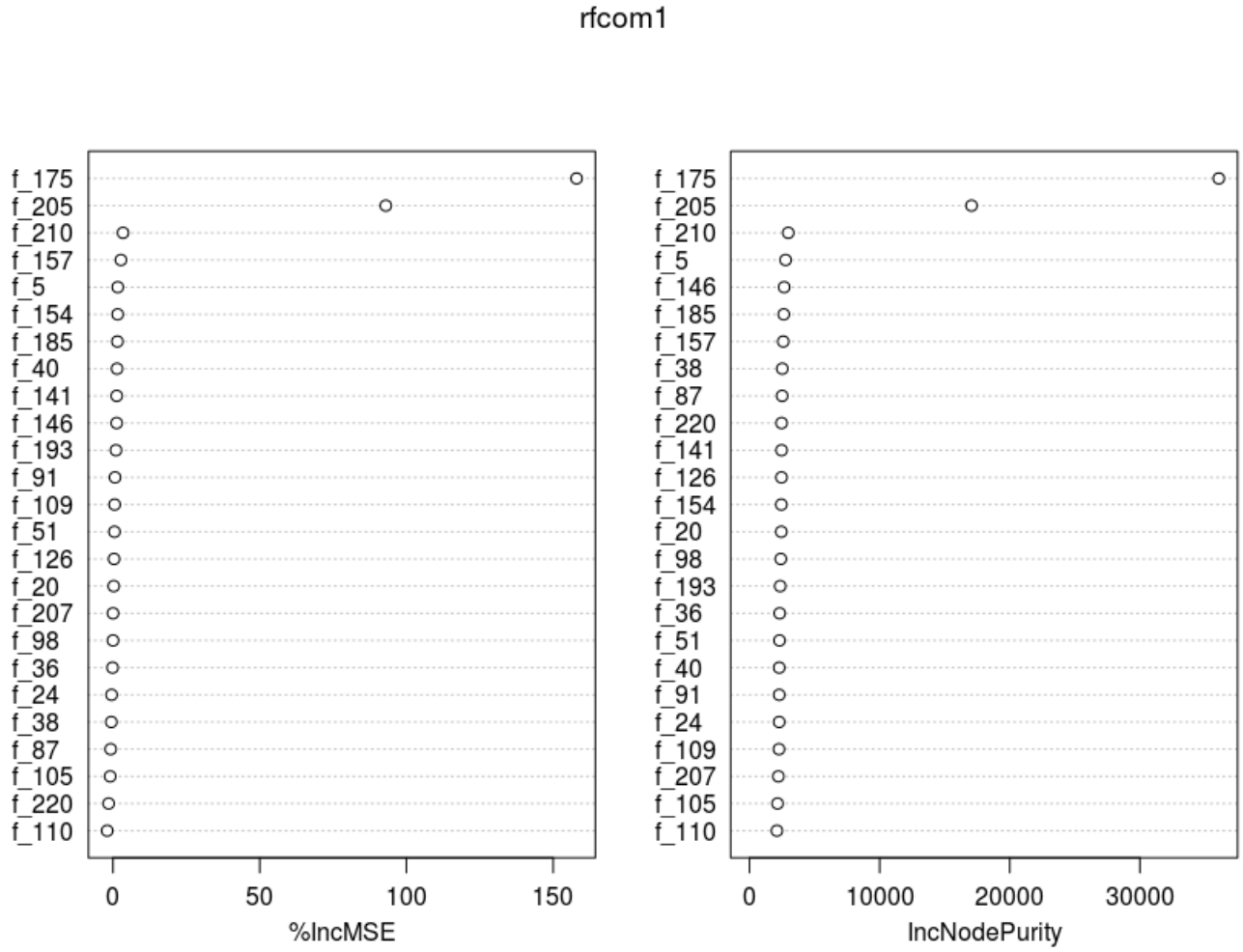
TL;DR
I primarily used two kinds of prediction methods: Random Forest and Gradient Boosting. After quite a lot of tweaking, I finally arrived on one model. Initially, Random Forest was used for feature selection, and Mice was used to impute the missing values. The models were trained with all sorts of combination of features and data. The final model is a GBM with "miced" data and a depth of 6, using all features as predictors. It had an MSE(mean square error) of 3.9% on the validation dataset, needed an optimal 317 iterations.

Timeline:

- Divided the training dataset into Training and Validation
- Trained the first RF model using the same sort of command:

rf80 <- randomForest(target ~ ., data=ctrain, mtry=80, ntree=1000, na.action=na.omit, importance=T)

- used varPlot and Importance factors to determine the significant features.



- Trained another RF model using these 25 features:

rf102 <- randomForest(target ~ f_5 + f_20 + f_24 + f_36 + f_38 + f_40 + f_51 + f_87 + f_91 + f_98 + f_105 + f_109 + f_110 + f_126 + f_141 + f_146 + f_154 + f_157 + f_175 + f_185 + f_193 + f_205 + f_207 + f_210 + f_220 , data=ctrain, mtry=80, ntree=1000, na.action=na.omit, importance=T)

- Also trained a GBM using the following sort of command:

gbm3 <- gbm(target ~ ., data=ctrain, n.trees = 3000, shrinkage = .05, cv.folds = 10)

- So far the model performance for both methods was okay but not satisfactory.
- Imputed missing values of training dataset using the Mice package.
- Fitted several Random Forests and GBMs using this dataset, called it 'comm', and after several adaptations of the inner parameters (like shrinkage, interaction.depth, ntrees ets) , and checking error rates and comparing model performances, arrives at the final *gbmcom06* model.
- Errors were calculated using MSE package, and performance measure was the OOB error for GBM and error rate for RF.
- Despite a good feature selection, ultimately minimum error was given by *gbmcom06*, which used all the features.
- The final model trained using the following command:

```
gbmcom6 <- gbm(target ~ ., data=comm, n.trees = 3000, shrinkage = .05,
interaction.depth = 6)
```

required a minimum of 274 iterations only, to drop the OOB to minimum.

- The following command generates the following graph:

```
> gbm.perf(gbmcom6, oobag.curve = TRUE, overlay = TRUE, method="OOB")
      [1] 274
```