STAT479 Project1 Wenyan Zhou

Abstract

The goal of the project is to predict INTRDVX in the BLS data, amount of interest and dividends, from other variables(excluding variables that are functions of INTRDVX). I used cross validation to compare different methods for prediction, including GUIDE, xgboost, rpart, ctree, randomForest, and ranger. Among these methods, I made predictions using GUIDE with some specific options and randomForest.

Data description & preparation

In "data_complete.rdata", which contains the training data, there are 25822 observations and 653 variables. What we are interested in are observations whose INTRDVX are flagged with "D" or "T". After filtration, there are only 2922 observations left, and the scatter plot of INTRDVX is shown in Figure1.1. We can see that observations flagged with "T" have extremely high INTRDVX, which is 98338.

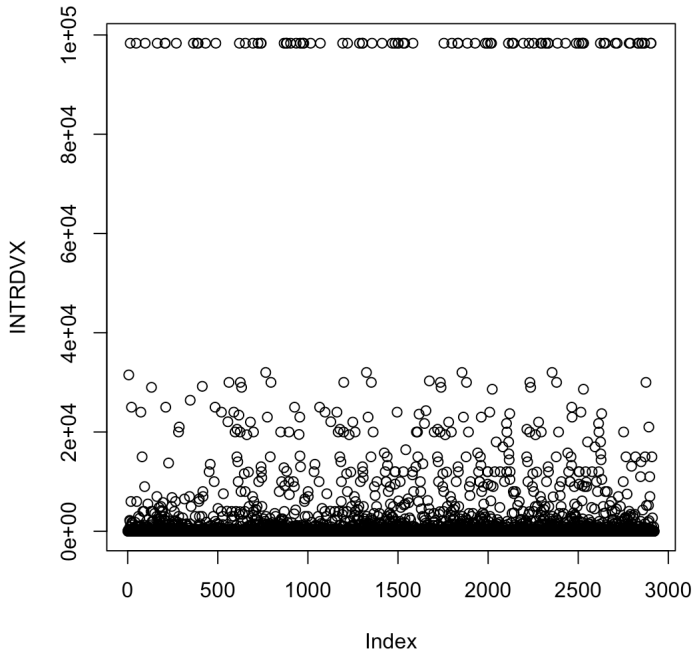


Figure1.1

It is easy for me to come up with an idea that we can first classify "D" and "T", and then predict them separately with different models. However, after some trials, the mean square does not decrease as expected, because once we make an error in classification, sum of squares will increase a lot, i.e. the error is cumulative. Considering it is a nonlinear model, I tried some tree-based methods. I first use GUIDE to do some preparation work:

1. Only keep important variables based on importance score. (part of them are listed below)

Predictor variables sorted by importance scores

Importance Scores

Scaled	Unscaled	Rank	Variable
100.0	2.53519E+01	1.00	STOCKX
94.5	2.39661E+01	2.00	STOCKYRX
92.6	2.34680E+01	3.00	FINCAT_X
92.4	2.34253E+01	4.00	FINCBT_X
87.5	2.21736E+01	5.00	CUTENURE
82.5	2.09138E+01	6.00	AGE_REF
72.0	1.82542E+01	7.00	RENTEQVX
71.5	1.81240E+01	8.00	STATE
60.1	1.52458E+01	9.00	PERSOT64
58.9	1.49342E+01	11.50	INCO_EY1
58.9	1.49342E+01	11.50	INC__RS1
58.9	1.49342E+01	11.50	INCN_NW1
58.9	1.49342E+01	11.50	OCCU_OD1
58.3	1.47711E+01	14.00	INCOMEY1
56.5	1.43175E+01	15.00	INCNONW1
54.6	1.38435E+01	16.00	RENT_QVX
54.5	1.38162E+01	17.00	AGE2
52.6	1.33389E+01	18.00	INC_HRS1
51.7	1.31011E+01	19.00	EARNCOMP
49.4	1.25284E+01	20.00	FSALARYX

2. Impute missing values in training data.

GUIDE

I tried several combinations of options both in data imputation and cross validation, and obtained mean square errors in Table2.1.

Table2.1

imputation	Cross validation	mse
constant	constant	137637659
constant	polynomial	169464487
constant	rforest	126677640
polynomial	constant	135199405
polynomial	polynomial	170474506
polynomial	stepwise linear	163787587
polynomial	rforest	121932561
polynomial	bagging	139664369
stepwise linear	rforest	123170268
rforest	rforest	137962744

From Table2.1, we can see that all mean square errors have the same order of magnitudes. Because cv groups are chosen randomly, it is hard to say which one is better. However, we can still see that those using rforest for cross validation have relatively smaller mean square error, which is reasonable for it is an ensemble method. Here, I used polynomial & rforest to get the prediction on the test data.

Other models

For the following models, I used GUIDE to impute testing data for prediction.

Xgboost

Xgboost is also an ensemble method, which is very fast using sparse matrix. First, I used one-hot coding step to transform the data with categorical variables into a very sparse matrix of numeric features, and then perform 10-fold cross validation to obtain mean square error. After tuning parameters, the model is:

```
bst <- xgboost(data = sparse_xgb_train_data, label = xgb_train$INTRDVX, max.depth = 5,
               eta = 1, nthread = 2, nround = 20, objective = "reg:linear")
```

The mean square error is 59803411 in cross validation, which is much smaller. However, in prediction, there exists some negative predicted values. I am not sure whether it is a good idea to let these values equal 0. To be conservative, I will not choose to use the prediction.

Rpart

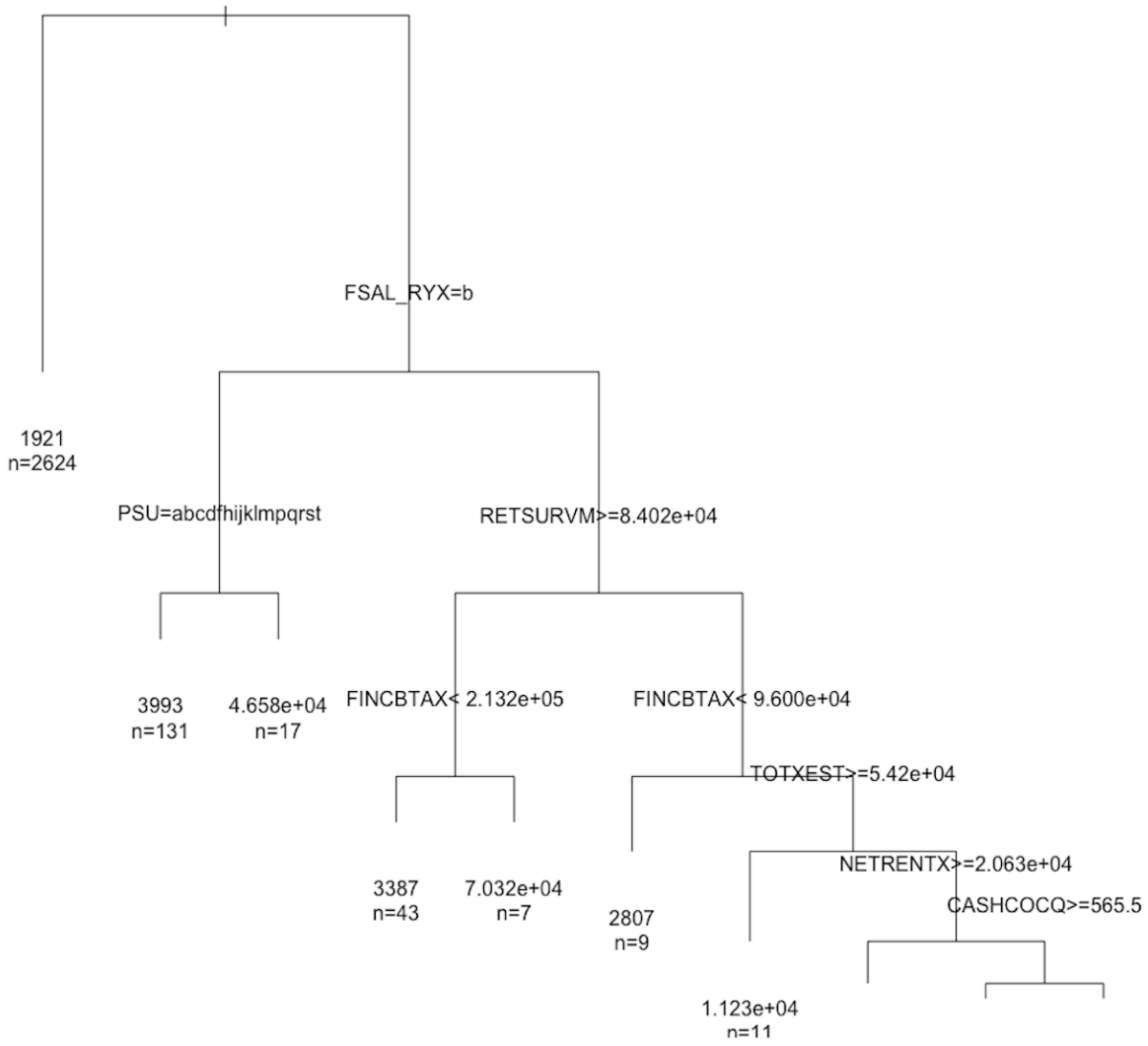
The model is:

```
fit_rpart <- rpart(mytrain$INTRDVX ~ ., method = "anova", data = mytrain[,-ncol(mytrain)])
```

The mean square error is 102943042, and variable importance are:

```
    FINCBTAX     FSAL_RYX     FSALARYX     FINCBT_X      TOTXEST     FJSSDEDX     RETSURVM
327698392638 234592896216 224726387561 210015684236 199657666264 123247112570 107994657526
    RETSURVX     RETPENPQ          PSU     INC_RANK     LUMPSUMX      RETSURV     DEFBENRP
107994657526 102377841860  76590003783  75791996905  55855447268  36718183559  32398397258
    MISCTAXX     NETRENTX     OTHRINCX      UNISTRQ     OTHREGX_     NETR_NTX     PROPTXCQ
 28939098008  24283899611  21598931505  19655703463  14741777597  14570339767  12151342962
      IRAYRX     PROPTXPQ     RENTEQVX     INCNONW2     CASHCOCQ      OWNVACC     VOTHRLOC
 11559868240  11559868240   9827851731   9629773688   8825492458   7285169883   7285169883
    BEDROOMQ     PERINSPQ     INC_HRS2     EARNCOMP     ESHELTRC      HOUSCQ      MEDSUPCQ
  6421161360   6421161360   4856779922   4815871020   2521569274   2521569274   2521569274
    FDAWAYCQ
  1260784637
```

The tree looks like:

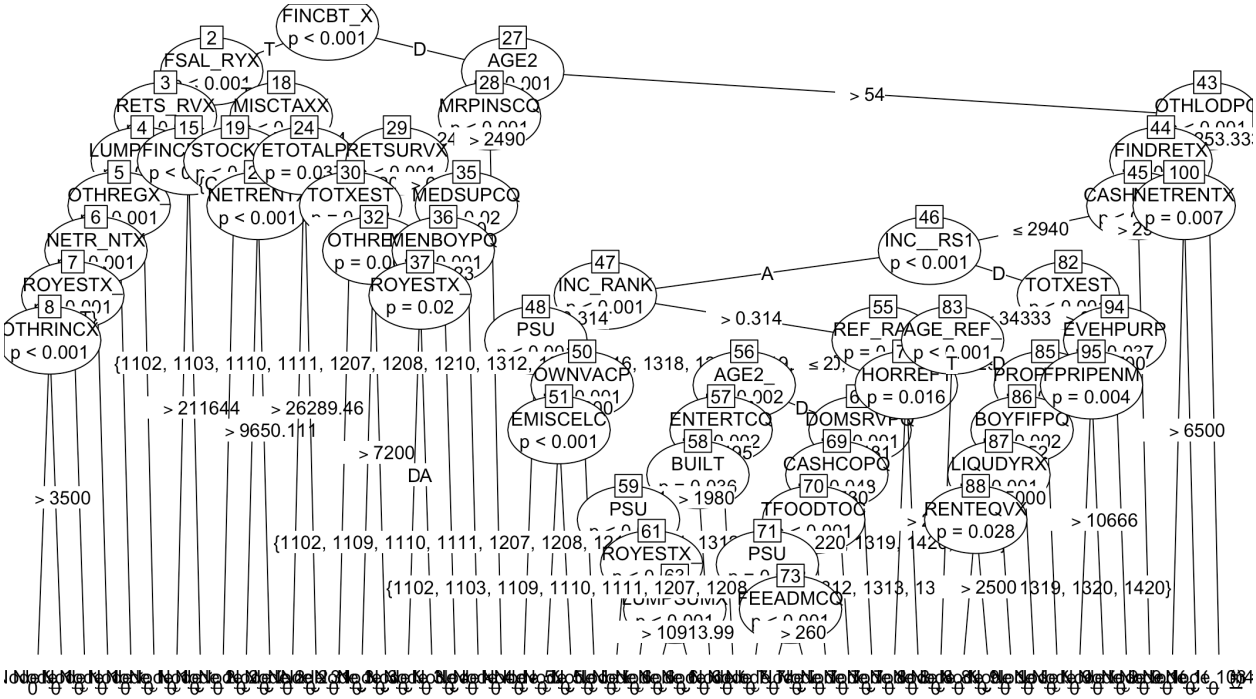


Ctree

The model is:

```
fit_ctree <- ctree(mytrain$INTRDVX ~ ., data = mytrain[,-ncol(mytrain)])
```

The mean square error is 101552174, and the tree looks like:



Randomforest

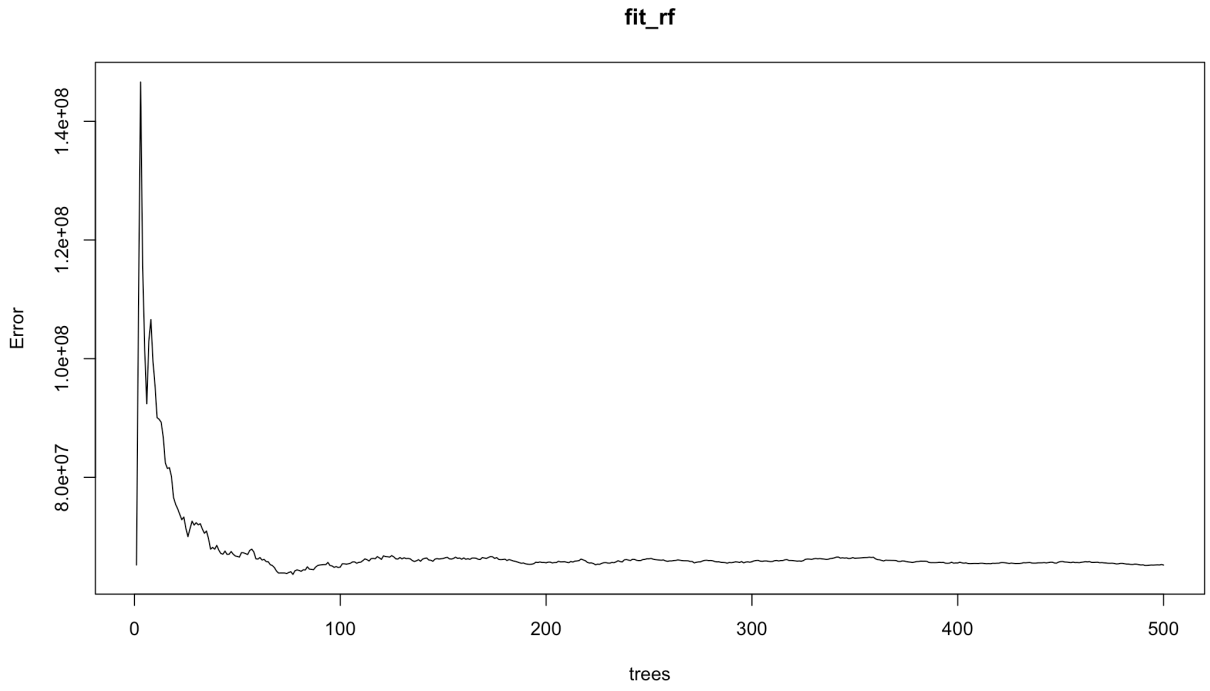
The model is:

```
fit_rf <- randomForest(mytrain$INTRDVX ~ ., data = mytrain[,-ncol(mytrain)])
```

The mean square error is 73069480, some important variables are:

	variable	imp.score
1	FINCBT_X	1.503279e+11
2	FINCBTAX	4.958366e+10
3	PSU	3.598470e+10
4	FSALARYX	3.372601e+10
5	RETSURVM	2.921292e+10
6	RETSURVX	2.802310e+10
7	FSAL_RYX	2.210272e+10
8	STOCKX_	2.075556e+10
9	TOTXEST	2.025832e+10
10	STOCKYRX	1.907386e+10
11	FJSSDEDX	1.832040e+10
12	RETS_RVX	1.487886e+10
13	INC_RANK	1.386806e+10
14	MISCTAXX	1.290500e+10
15	STOCKX	1.219646e+10
16	ETOTALP	1.125086e+10
17	INC_HRS1	9.498915e+09
18	ETOTAPX4	9.364961e+09
19	NETRENTX	8.100744e+09
20	NETR_NTX	8.069273e+09

The error plot is:



Ranger

The model is:

```
fit_ranger <- ranger(mytrain$INTRDVX ~ ., data = mytrain[,-ncol(mytrain)])
```

The mean square error is 122875860.

Conclusion

MSE using cross validation of different models:

model	mse
Xgboost	59803411
Rpart	102943042
Ctree	101552174
Randomforest	73069480
Ranger	122875860

The lowest mean square error is given by xgboost, but there is still something ambiguous about it. Randomforest also performs well in cross validation, but it is really time consuming.

Finally, my two columns of prediction are given by GUIDE and Randomforest.