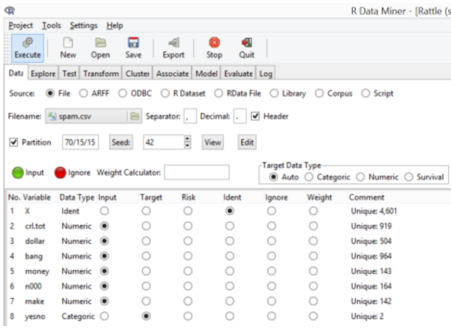
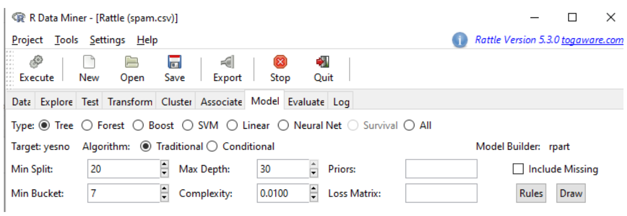
Solution a) Read the data



b) Fitting the model



c. Training Error

Default

```
Error matrix for the Decision Tree model on spam.csv [validate]
(counts):


      Predicted

Actual    n    y Error

     n 382   46  10.7

     y  52 210  19.8


Error matrix for the Decision Tree model on spam.csv [**train**]
(counts):


      Predicted

Actual     n     y Error

     n 1733  212  10.9
```

```
    y  234 1041  18.4

Error matrix for the Decision Tree model on spam.csv [**train**]
(proportions):

      Predicted
Actual    n    y Error
     n 53.8  6.6  10.9
     y  7.3 32.3  18.4

Overall error: 13.9%, Averaged class error: 14.65%
```

d. Validation Error
e. ==
f.
g. Error matrix for the Decision Tree model on spam.csv [validate] (proportions):
h.
i.   Predicted
j. Actual   n   y Error
k.   n 55.4 6.7 10.7
l.   y 7.5 30.4 19.8
m.
n. Overall error: 14.2%, Averaged class error: 15.25%
o.
p. Rattle timestamp: 2020-04-11 12:15:13 sridhar

We see error rate in the validation set is slightly higher than that in the training set. This indicates there might be slight overfitting. We can try various alternatives here. To reduce variance (overfitting) we can either INCREASE “min split”, DECRAESE “max depth”, INCREASE “min bucket”, or INCRAESE Complexity Parameter(CP). There are many possible models one could fit to improve performance. In fact, one might that fitting even more complex model can be helpful, thus we leave it to the participant to choose any number of models they want to choose. All we care for is a model that improves on the default model and also shows not much divergence between train set error and test set error.

2) These parameters are used to control overfitting (high variance) or underfitting (high bias). We can use either (more than one at a time) of them to improve the model whichever way we deem useful.

- **Min Split** – It specifies that a node should be split only if there are a minimum number of observations. Thus, if we increase this number, the node is less likely to be split. Therefore a higher number may help in curbing overfitting. However, a very high number may bias the model.
- **Max Depth** - This determines how deep can our tree grow. The deeper a tree, more complex it is. Thus, if we keep this number low, our tree will not overfit. Choose a lower number to control variance.
- **Min Bucket** – This is the minimum number of observations in any leaf node. A higher number controls the tree from overgrowing.
- **Complexity Parameter (CP)** – This determines the minimum "benefit" that must be gained at each split. Thus, a tree with CP=0 will grow as much as possible (depending on the values of other parameters). A higher value controls the growth of the tree.

For example following is sample analysis

Changing max depth to 10 does not make much diff

Changing min bucket to 3 makes no difference

Changed CP to zero

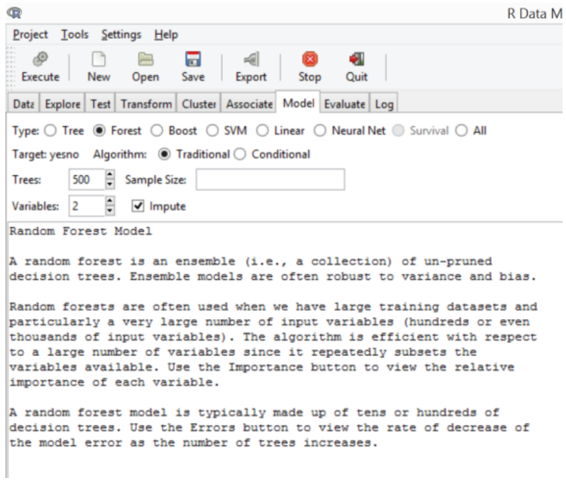
```
Error matrix for the Decision Tree model on spam.csv [**train**]
(counts):

      Predicted

Actual    n    y Error

     n 1845  100   5.1

     y  251 1024  19.7


Error matrix for the Decision Tree model on spam.csv [**train**]
(proportions):

      Predicted

Actual    n    y Error

     n 57.3  3.1   5.1
```

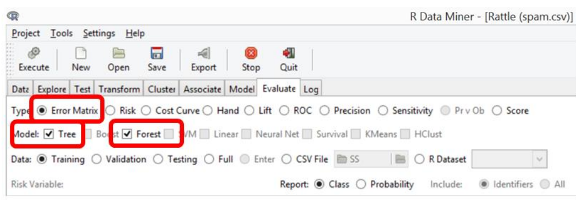
```
     y  7.8 31.8  19.7


Overall error: 10.9%, Averaged class error: 12.4%

Rattle timestamp: 2020-04-11 12:23:16 sridhar
```

```
Error matrix for the Decision Tree model on spam.csv [validate]
(counts):

      Predicted

Actual   n   y Error

     n 402  26   6.1

     y  56 206  21.4


Error matrix for the Decision Tree model on spam.csv [validate]
(proportions):

      Predicted

Actual    n    y Error

     n 58.3  3.8   6.1

     y  8.1 29.9  21.4


Overall error: 11.8%, Averaged class error: 13.75%


Rattle timestamp: 2020-04-11 12:24:12 sridhar
```

Note that training error comes down. Validation error reduces but spam error (y) goes up. The trade-off (not as clear here) is fitting well on training data versus predicting well on validation data. Note that the model can be doing poorly on training data because more features or more data needed.

3) Random Forest – Using Default option





Error Matrix for Random Forest – Default parameters

```
Rattle timestamp: 2020-05-09 17:12:01 ruchi
======================================================================
Error matrix for the Random Forest model on spam.csv [**train**] (counts):

     Predicted
Actual    n    y Error
     n 1921   24   1.2
     y  219 1056  17.2

Error matrix for the Random Forest model on spam.csv [**train**] (proportions):

     Predicted
Actual    n    y Error
     n 59.7  0.7   1.2
     y  6.8 32.8  17.2

Overall error: 7.5%, Averaged class error: 9.2%

Rattle timestamp: 2020-05-09 17:12:02 ruchi
======================================================================
```

```
Rattle timestamp: 2020-05-09 17:14:27 ruchi
======================================================================
Error matrix for the Random Forest model on spam.csv [validate] (counts):

      Predicted
Actual   n   y Error
     n 406  22   5.1
     y  57 205  21.8

Error matrix for the Random Forest model on spam.csv [validate] (proportions):

      Predicted
Actual    n    y Error
     n 58.8  3.2   5.1
     y  8.3 29.7  21.8

Overall error: 11.5%, Averaged class error: 13.45%

Rattle timestamp: 2020-05-09 17:14:27 ruchi
======================================================================
```

We see that the error rate (validation set) in Random Forest is lower than that in Decision Tree. However, the disadvantage of Random Forest is we lose the nice interpretability of the rules that Decision Tree Provides.