A9.1
First I created the linear regression model on first 4 principal components.
Model coefficients using this are:

```
##        M   So   Ed  Po1  Po2   LF   M.F  Pop  NW    U1   U2 Wealth Ineq Prob   Time
##[1,] -16.9 21.3 12.8 21.4 23.1 -347 -8.29 1.05 1.5 -1510 1.69   0.04 -6.9  145 -0.933
```

Intercept is 1666

So using this crime value for the given data points is **1113**

Below is the table of R2 and adjusted R2 using different number of principal components

Number of Principal Components	R squared	Adjusted R squared	R squared Cross Validated
3	0.263	0.230	0.0910
4	0.272	0.221	0.0666
5	0.309	0.243	0.1057
6	**0.645**	**0.602**	**0.4872**
7	0.659	0.607	0.4628
8	0.688	0.632	0.4562
9	0.690	0.625	0.3664
10	0.692	0.617	0.3337
11	0.696	0.612	0.2954
12	0.697	0.602	0.1863
13	0.769	0.688	0.3897
14	0.772	0.683	0.3902
15	0.791	0.700	0.4736

Using linear regression on Principal components, it looks like first 6 principal components should be used, based on Cross Validated R squared values.

Using this, Prediction is **1248**

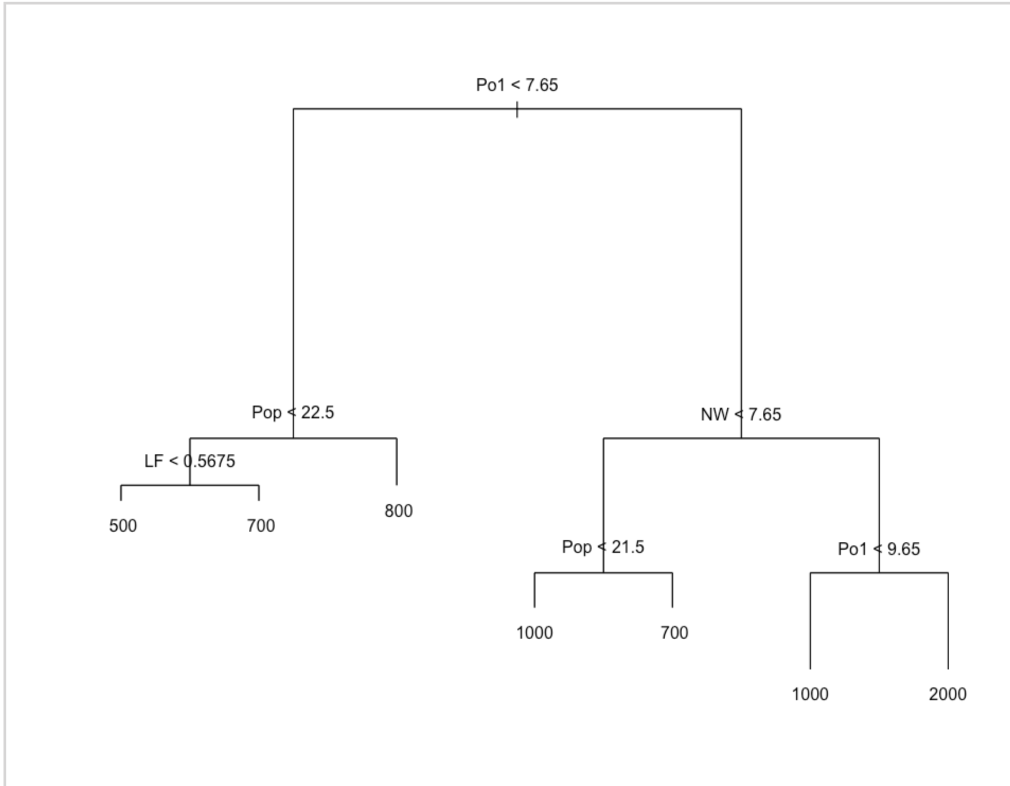
Comparing to my answers of 8.2, model I chose had the Cross Validated R squared value of 0.584 and prediction was 1304.

A 10.1 (a - regression tree model)

Tree model, default. Shows 7 leaf nodes.

R^2 value - 0.724

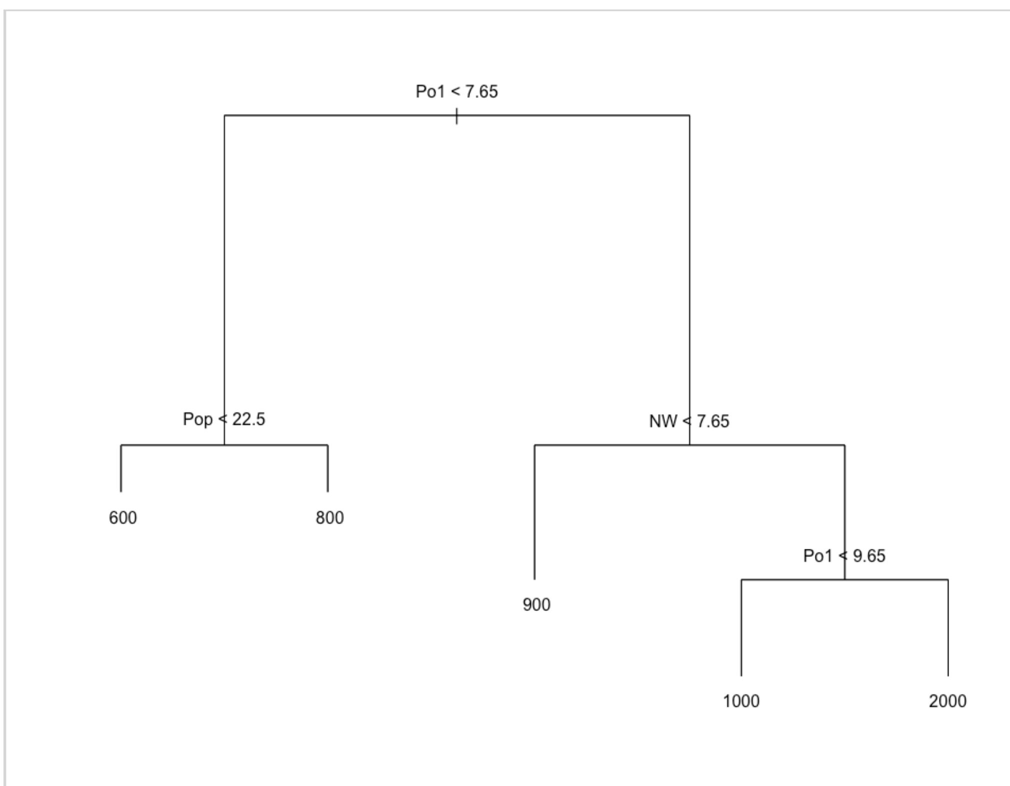
Prediction value for given data point - 725



Plot after pruning the tree to 5 leaf nodes.

R^2 value - 0.669

Prediction value for given data point - 887

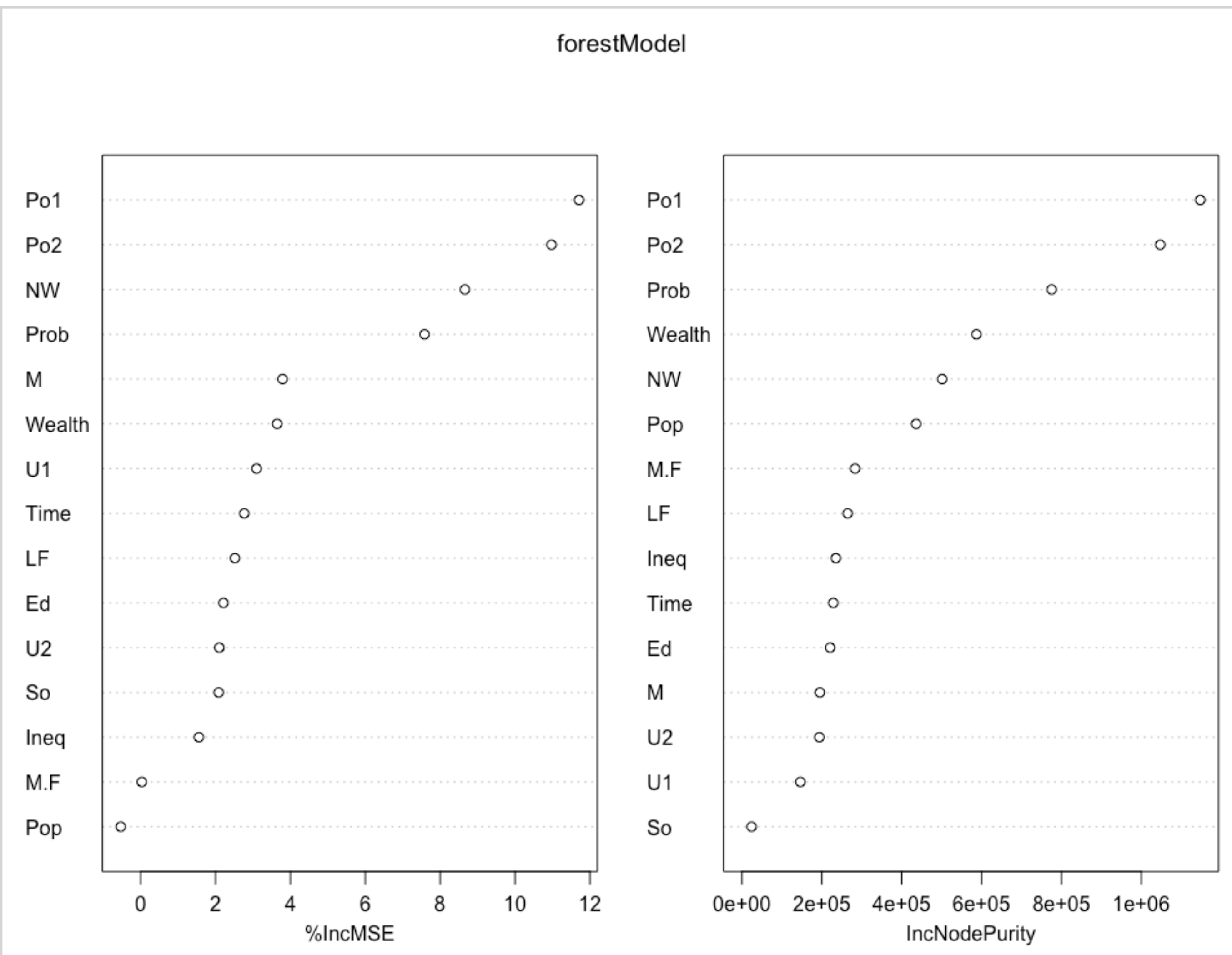


A 10.1 (b - random forest model)

% IncMSE - based on mean decrease in accuracy of predictions, when the given variable is excluded from the model

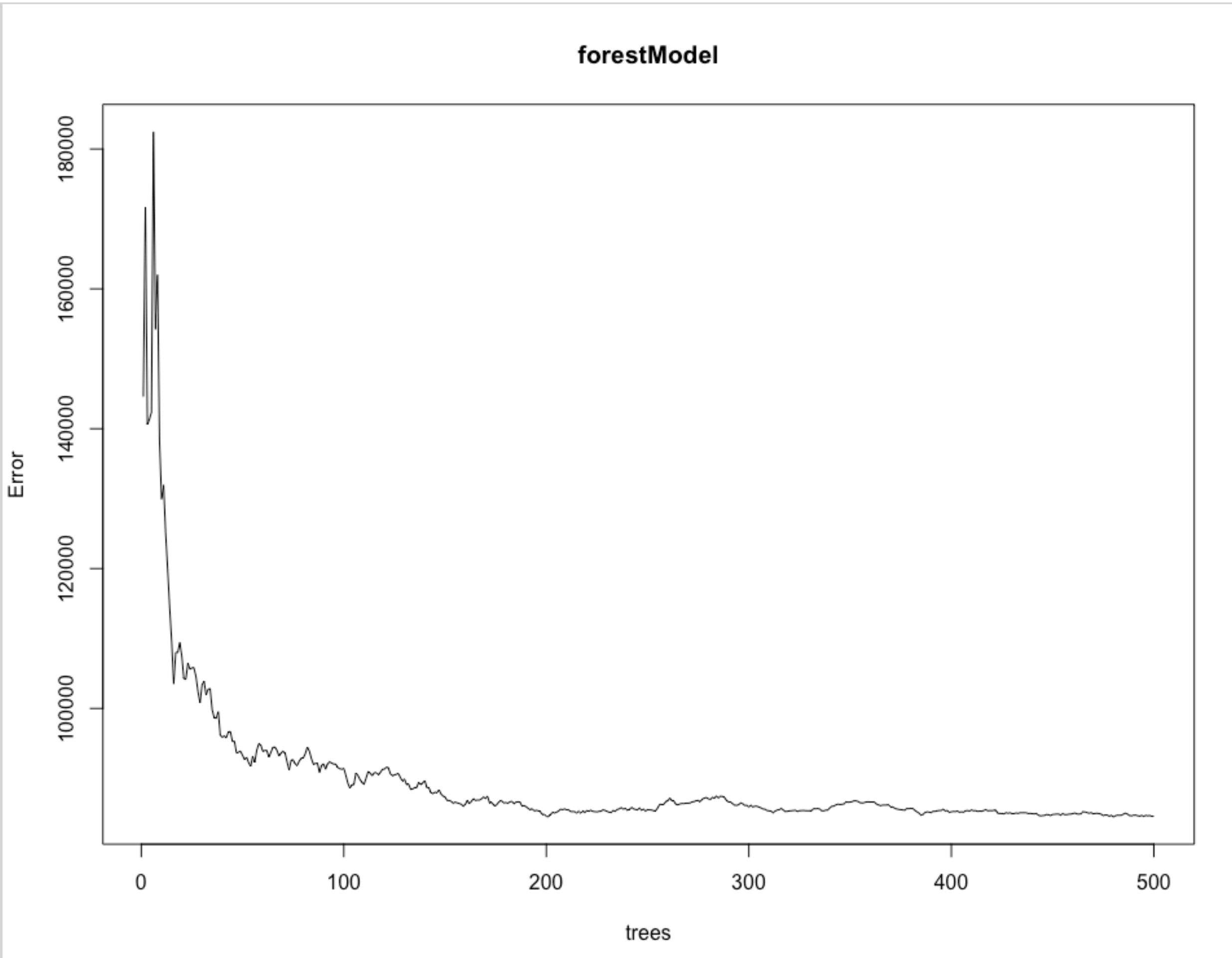
IncNodePurity - measure of the total decrease in node impurity that results from splits over that variable.

In both metrics, Higher the value means its more important.



Random forest with 500 trees
No. of variables tried at each split: 4

R^2 value: 0.422
Prediction value for given data point - 1204



Looking at the results, random forest model seems to be providing better predictions. It produces lots of different trees using randomly chosen factors. In the end, average of the regression trees is used to provide predicted response.

A10.2
In my organization, a financial institution, its important to distinguish a fraudulent login (fraudster logged in by getting the credentials) vs a genuine a customer login. Logistic regression can be used to categorize the logins. Some of the predictors used could be -

- New IP Address for customer (Y/N)
- last successful login date
- Recent password change
- Request from “blacklisted” IPs / Countries
- Nearby Previous failed attempts

A10.3 (part 1)
First I created logistic regression model. I used 70% of the data for training and 30% of it for validation/tetsing.

Call:
glm(formula = data.train$V21 ~ V1 + V2 + V3 + V4 + V5 + V6 +
    V8 + V9 + V10 + V14 + V15 + V20, family = binomial(link = "logit"),
    data = data.train)

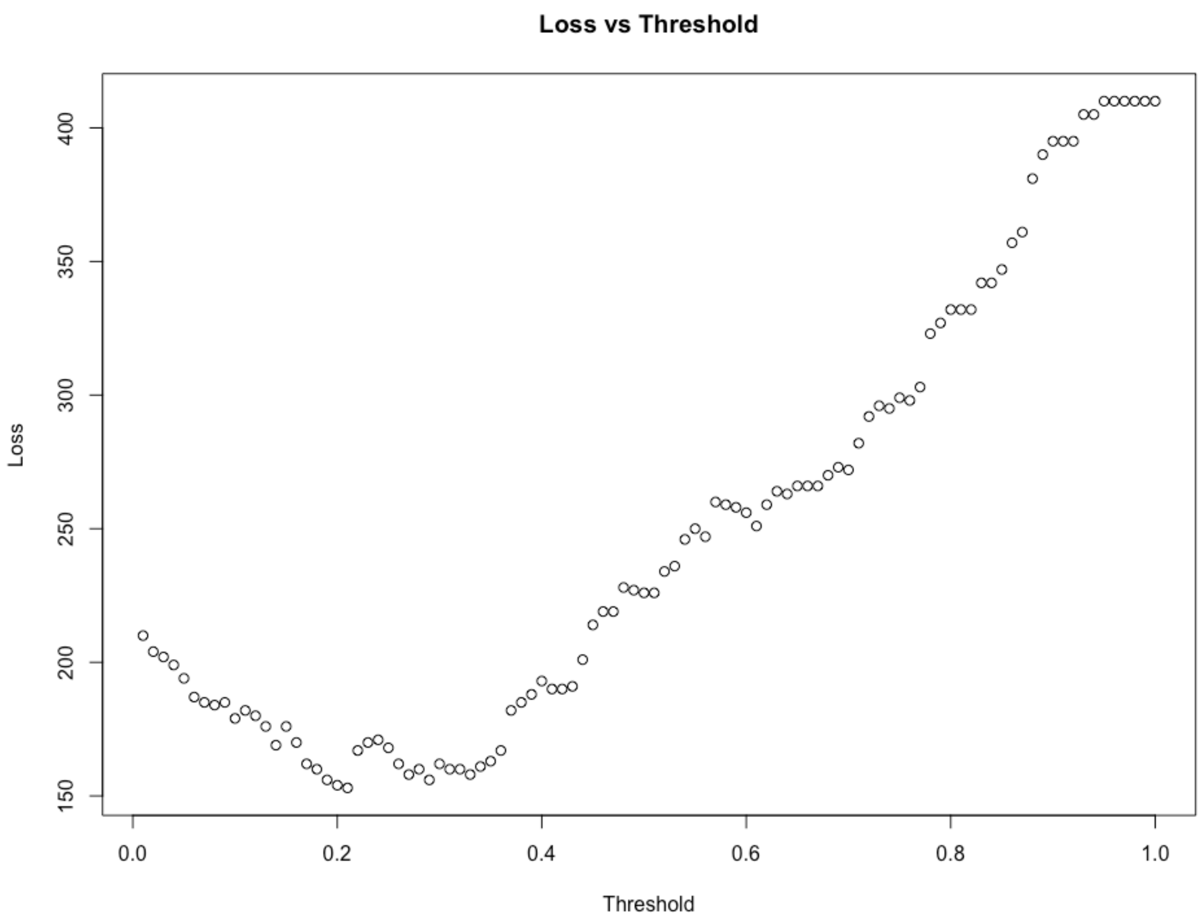
Table 1-1

Factors	**Coefficients**	**Factors**	**Coefficients**
(Intercept)	1.43E+00		
V1A12	-3.05E-01	V9A92	-4.33E-01
V1A13	-1.23E+00	V9A93	-8.11E-01
V1A14	-1.49E+00	V9A94	-6.06E-01
V2	2.31E-02	V10A102	7.43E-01
V3A31	2.83E-01	V10A103	-1.72E+00
V3A32	-6.48E-01	V14A142	1.21E-01
V3A33	-4.56E-01	V14A143	-6.5E-01
V3A34	-1.40E+00	V15A152	-8.25E-01
V4A41	-1.60E+00	V15A153	-5.91E-01
V4A410	-1.22E+00	V20A202	-2.22E+00
V4A42	-5.06E-01		
V4A43	-8.23E-01		
V4A44	-2.56E-01		
V4A45	-1.13E-01		
V4A46	3.14E-01		
V4A48	-2.03E+00		
V4A49	-5.38E-01		
V5	8.99E-05		
V6A62	-1.01E-01		
V6A63	-6.04E-01		
V6A64	-1.17E+00		
V6A65	-1.27E+00		
V8	3.38E-01		

Using the threshold as 0.5, model accuracy is 75.3%.

A 10.3 (Part 2)
Since the cost of False positive and False negatives is not usually same, we have to choose a threshold so that total cost is lowest, given that cost of identifying a bad customer as good, is 5 times worse than incorrectly classifying a good customer as bad.

Here is the plot of threshold value from 0.01 to 1.00



Using this, the cost is lowest with threshold of 0.21.
Cost is about 153 units.