HW01 Steve Yang(sy3153)

Before going to 1.1, I want to share my dataset informations:

My dataset is Mushroom (https://archive.ics.uci.edu/ml/datasets/Mushroom). It is a categorical dataset with 8124 number of instances and 22 attributes (all nominally valued). There are missing values in the dataset and there are some features that are imbalanced.

Balance:

My dataset is focusing on predicting if the mushroom is poisonous, and the ratio of it in my dataset is edible:51.8%, poisonous 48.2%(edible:4208, poisonous:3916). So, the target feature of my dataset is balance.

I have attached my results of “table(mushroom$features)” at the last of the report. Due to having multiple imbalanced features, my strategy is training with original features first and doing undersampling if necessary. Because if I do undersampling/oversampling for a particular feature, the other 20 features are influenced. And if it is necessary, I will choose undersampling instead of oversampling, because In my opinion, oversampling will add additional observers in my dataset which is hard to handle.

Check for collinearity:

```
mushrooms<-read.csv("./mushroom.csv",sep=',',head=T,stringsAsFactors = F)

# m10percent<-sample(1:nrow(mushrooms),nrow(mushrooms)*0.10,replace=F)
#
# m10p<-mushrooms[m10percent,c(1:23)]
#

pairs(m10p, col = "dodgerblue")

round(cor(mushrooms), 2)
```

However, when I try to use pair() function to plot all possible scatterplots between pairs of variables in the dataset. It is hard to tell due to the large dataset, so I decided to use cor() function which when applied to a dataset, returns all pairwise correlations.

pair():

cor():

```
> round(cor(mushrooms),2)
```

	poisonous	capShape	capSurface	capColor	bruises	odor	gillAttachment	gillSpacing	gillSize	gillColor	stalkShape	stalkRoot	SSAR	SSBR	SCAR	SCBR	veilType	veilColor	ringNumber	ringType	sporePrintColor	population
poisonous	1.00	0.20	0.19	-0.06	0.50	-0.09	0.13	-0.35	0.54	-0.27	-0.10	0.17	-0.22	-0.14	-0.26	-0.25	NA	0.15	-0.21	-0.41	0.52	0.30
capShape	0.20	1.00	-0.01	-0.18	0.20	0.25	0.03	-0.06	0.26	-0.07	0.25	0.22	-0.07	-0.07	-0.06	-0.07	NA	0.04	-0.07	-0.30	0.25	0.13
capSurface	0.19	-0.01	1.00	-0.02	0.02	-0.11	-0.16	-0.10	0.27	-0.12	0.04	0.39	0.02	0.00	0.25	0.26	NA	-0.15	0.06	-0.17	0.31	-0.19
capColor	0.06	-0.18	-0.02	1.00	-0.03	-0.32	0.19	0.02	-0.09	-0.02	-0.24	-0.17	-0.02	-0.03	-0.04	-0.03	NA	0.19	0.01	0.09	-0.08	-0.02
bruises	0.50	0.20	0.02	-0.03	1.00	0.08	-0.14	0.30	0.37	-0.33	-0.10	0.46	-0.39	-0.32	-0.20	-0.21	NA	-0.12	-0.06	-0.69	0.32	-0.09
odor	-0.09	0.25	-0.11	-0.32	0.08	1.00	-0.09	0.11	0.04	0.07	0.34	0.15	-0.04	0.06	-0.01	-0.03	NA	-0.09	0.16	-0.16	0.09	0.00
gillAttachment	0.13	0.03	-0.16	0.19	-0.14	-0.09	1.00	0.07	0.11	-0.08	0.19	-0.19	-0.08	-0.08	0.12	0.12	NA	0.90	0.09	-0.15	-0.07	0.17
gillSpacing	-0.35	-0.06	-0.10	0.02	0.30	0.11	0.07	1.00	-0.11	-0.04	0.08	0.11	-0.30	-0.24	0.32	0.27	NA	0.07	0.24	-0.20	-0.09	-0.53
gillSize	0.54	0.26	-0.27	-0.09	0.37	0.04	0.11	-0.11	1.00	-0.33	0.21	0.59	0.09	0.09	0.27	0.23	NA	0.10	-0.17	-0.46	0.55	0.13
gillColor	-0.27	-0.07	-0.12	-0.02	-0.35	0.07	-0.08	-0.04	-0.33	1.00	-0.19	-0.28	0.11	0.06	-0.06	-0.10	NA	-0.05	0.24	0.39	-0.14	0.03
stalkShape	-0.10	0.25	0.04	-0.24	-0.10	0.34	0.19	0.08	0.21	-0.19	1.00	0.14	-0.04	0.06	0.23	0.25	NA	0.16	-0.29	-0.29	0.04	0.09
stalkRoot	0.17	0.22	0.39	-0.17	0.46	0.15	-0.19	0.11	0.59	-0.28	0.14	1.00	0.01	-0.07	0.35	0.35	NA	-0.20	0.25	-0.60	0.69	-0.28
SSAR	-0.22	-0.07	0.02	-0.02	-0.39	-0.04	-0.08	-0.30	0.09	0.11	-0.04	0.01	1.00	0.38	0.10	0.12	NA	-0.10	0.08	0.35	-0.09	0.16
SSBR	-0.14	-0.07	0.00	-0.03	-0.32	0.06	-0.08	-0.24	0.09	0.06	0.06	-0.07	0.38	1.00	0.04	0.07	NA	-0.10	0.11	0.30	-0.05	0.13
SCAR	-0.26	-0.06	0.25	-0.04	-0.20	-0.01	0.12	0.32	0.27	-0.06	0.23	0.35	0.10	0.04	1.00	0.63	NA	0.10	0.18	-0.04	0.01	-0.36
SCBR	-0.25	-0.07	0.26	-0.03	-0.21	-0.03	0.12	0.27	0.23	-0.10	0.25	0.35	0.12	0.07	0.63	1.00	NA	0.10	0.18	-0.01	-0.03	-0.36
veilType	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1	NA	NA	NA	NA	NA
veilColor	0.15	0.04	-0.15	0.19	-0.12	-0.09	0.90	0.07	0.10	-0.05	0.16	-0.20	-0.10	-0.10	0.10	0.10	NA	1.00	0.04	-0.14	-0.04	0.12
ringNumber	-0.21	-0.07	0.06	0.01	-0.06	0.16	0.09	0.24	-0.17	0.24	-0.29	0.25	0.08	0.11	0.18	0.18	NA	0.04	1.00	0.06	0.32	-0.24
ringType	-0.41	-0.30	-0.17	0.09	-0.69	-0.16	-0.15	-0.20	-0.46	0.39	-0.29	-0.60	0.35	0.30	-0.04	-0.01	NA	-0.14	0.06	1.00	-0.57	0.21
sporePrintColor	0.52	0.25	0.31	-0.08	0.52	0.09	-0.07	-0.09	0.55	-0.14	0.04	0.69	-0.09	-0.05	0.01	-0.03	NA	-0.04	0.32	-0.57	1.00	0.01
population	0.30	0.13	-0.19	-0.02	-0.09	0.00	0.17	-0.53	0.15	0.03	0.09	-0.28	0.16	0.13	-0.36	-0.36	NA	0.12	-0.24	0.21	0.01	1.00
habitat	-0.02	0.13	-0.19	-0.09	-0.31	0.13	0.12	-0.40	0.09	0.19	0.24	-0.27	0.29	0.28	-0.18	-0.16	NA	0.12	-0.15	0.22	-0.08	0.43

	habitat
poisonous	-0.02
capShape	0.13
capSurface	-0.19
capColor	-0.09
bruises	-0.31
odor	0.13
gillAttachment	0.12
gillSpacing	-0.40
gillSize	0.09
gillColor	0.19
stalkShape	0.24
stalkRoot	-0.27
SSAR	0.29
SSBR	0.28
SCAR	-0.18
SCBR	-0.16
veilType	NA
veilColor	0.12
ringNumber	-0.15
ringType	0.22
sporePrintColor	-0.08
population	0.43
habitat	1.00

According to the table, we can see that “veliType” is “NA” for all other predictors.

Constant Features:

I used “table(mushroom$features)” function to check if the feature is a constant feature, and “veliType” has only “p” in it (8124:0), which means it is meaningless, so I decided to remove it from my dataset.

Missing Data:

“stalkRoot” has 2480 missing data as “?” in it (2480:8124), because currently I have 21 features as predictors which I think is enough for a good performance model. So, for better performance, I decided to temporarily remove it from my dataset.

As a result, my dataset has 20 features as predictors to predict if the mushroom is “poisonous” and it is a binary classification.

```
> dim(mushroom_modify)
[1] 8124   21
```

```
> table(mushroom_modify$poisonous)

   e    p
4208 3916
```

1.0 RandomForest

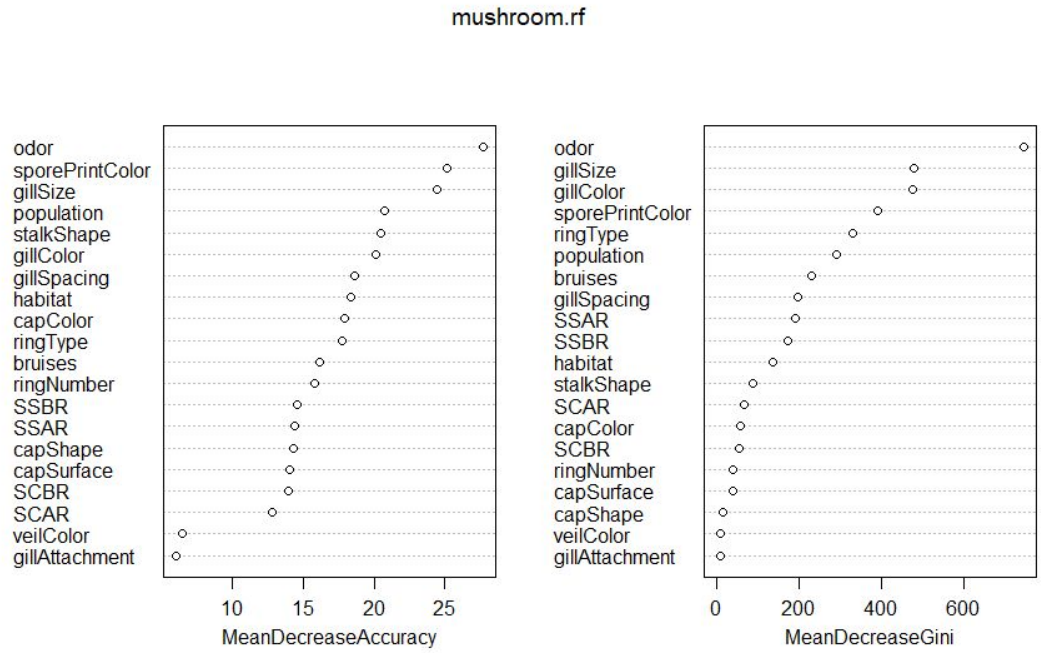
My first task is using randomForest and running varImp/varImpPlot to calculate the importance of different features.

```
mushroom_modify<-read.csv("./mushroom_modify.csv",sep=',',head=T,stringsAsFactors = F)

mushroom_modify$poisonous = factor(mushroom_modify$poisonous)

mushroom.rf<-randomForest(poisonous~.,data=mushroom_modify,importance=TRUE)

#importance(mushroom.rf)

varImp(mushroom.rf)

varImpPlot(mushroom.rf)
```

The result of varImp:

```
> varImp(mushroom.rf)
                       e         p
capShape        10.111844 10.111844
capSurface      11.616537 11.616537
capColor        13.454176 13.454176
bruises         14.410674 14.410674
odor            24.600941 24.600941
gillAttachment   5.225003  5.225003
gillSpacing     16.231569 16.231569
gillSize        22.391835 22.391835
gillColor       17.349199 17.349199
stalkShape      16.832171 16.832171
SSAR            12.462628 12.462628
SSBR            12.594783 12.594783
SCAR            11.280090 11.280090
SCBR            11.632858 11.632858
veilColor        6.013533  6.013533
ringNumber      13.803673 13.803673
ringType        15.568825 15.568825
sporePrintColor 21.698223 21.698223
population      17.271128 17.271128
habitat         17.371507 17.371507
```

The result of varImpPlot:



According to https://wiki.q-researchsoftware.com/wiki/Machine_Learning_-_Random_Forest ,*"As with `MeanDecreaseAccuracy`, high numbers indicate that a variable is more important as a predictor."*

As a result, my top 10 features in MeanDecreaseAccuracy are: "odor", "sporePrintColor", "gillSize", "population", "stalkShape", "gillColor", "gillSpacing", "habitat", "capColor", "ringType". I'm using forward selection, so I added those 10 features in my model.

Here is the link to my new dataset mushroomsRF.csv:

https://docs.google.com/spreadsheets/d/11in1NI2HJ5hFHcsHWD5cNolljj2bYbNS0whbXfpFtf4/edit?usp=sharing

1.2 Performance Matrix

I will put all my R script in the Appendix part.

GLM :

After that, I want to implement GLM to plot the ROC curve and calculate the AUC of it with the same dataset. (Before that, I have converted my dataset into numerical)

Also, I made a 100 times loop for GLM predicting test data and get the informations as shown below:

Training data:
Use the model to predict training dataset:

```
> source("helper.R")
> calVar(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 1.437396e-05 2.464495e+03 3.018292e-05 2.091927e-05 2.977922e-05 3.018292e-05 5.969609e-06
>
> calMean(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1]    0.8627708 2265.8106538    0.8349955    0.9152252    0.8990308    0.8349955    0.9408567
>
```

AUC: (mean:0.941 varaiance: 5.970e-06)

```
> AUCListR
  [1] 0.9357142 0.9424603 0.9428539 0.9426154 0.9467960 0.9372515 0.9420712 0.9404115 0.9420094 0.9426858 0.9388582 0.9400194 0.9461058 0.9382673 0.9420322 0.9420877
 [17] 0.9343836 0.9407649 0.9422839 0.9397941 0.9360301 0.9395371 0.9441614 0.9405815 0.9401838 0.9402446 0.9415407 0.9404229 0.9395226 0.9445641 0.9445572 0.9404376
 [33] 0.9397757 0.9401006 0.9418980 0.9374862 0.9417241 0.9434623 0.9377047 0.9442205 0.9450922 0.9459641 0.9429675 0.9420818 0.9395004 0.9415727 0.9372988 0.9424481
 [49] 0.9422122 0.9364102 0.9418684 0.9406606 0.9422553 0.9411082 0.9369845 0.9399473 0.9447118 0.9386083 0.9404206 0.9419769 0.9409331 0.9384022 0.9400526 0.9372803
 [65] 0.9441433 0.9370660 0.9394406 0.9447008 0.9438683 0.9417379 0.9391858 0.9396529 0.9424402 0.9378557 0.9395334 0.9372525 0.9377297 0.9375337 0.9423086 0.9410750
 [81] 0.9432345 0.9427870 0.9422612 0.9415569 0.9404087 0.9423394 0.9390976 0.9418873 0.9402602 0.9403068 0.9412499 0.9393077 0.9389033 0.9408257 0.9382572 0.9427190
 [97] 0.9413394 0.9427074 0.9433768 0.9389440
```

Accuracy: (mean:0.8627708 variance:1.437e-05)

```
> accListR
  [1] 0.8640591 0.8624179 0.8681619 0.8629650 0.8624179 0.8646061 0.8654267 0.8594092 0.8572210 0.8717177 0.8599562 0.8676149 0.8637856 0.8673414 0.8667943 0.8580416
 [17] 0.8613239 0.8665208 0.8613239 0.8646061 0.8657002 0.8708972 0.8689825 0.8574945 0.8637856 0.8626915 0.8651532 0.8615974 0.8648796 0.8673414 0.8670678 0.8624179
 [33] 0.8651532 0.8695295 0.8635120 0.8564004 0.8602298 0.8613239 0.8624179 0.8607768 0.8681619 0.8714442 0.8670678 0.8687090 0.8596827 0.8591357 0.8626915 0.8632385
 [49] 0.8648796 0.8602298 0.8594092 0.8583151 0.8591357 0.8569475 0.8667943 0.8615974 0.8651532 0.8659737 0.8665208 0.8610503 0.8626915 0.8689825 0.8564004 0.8599562
 [65] 0.8667943 0.8618709 0.8624179 0.8632385 0.8624179 0.8591357 0.8607768 0.8629650 0.8596827 0.8637856 0.8618709 0.8676149 0.8659737 0.8583151 0.8564004 0.8657002
 [81] 0.8621444 0.8553063 0.8542123 0.8615974 0.8615974 0.8605033 0.8572210 0.8629650 0.8605033 0.8626915 0.8561269 0.8637856 0.8643326 0.8624179 0.8643326 0.8610503
 [97] 0.8594092 0.8646061 0.8596827 0.8564004
```

Residual Deviance:(mean:2265.81 variance:2.454e+03)

```
> devianceListR
  [1] 2288.218 2277.857 2325.476 2313.275 2247.781 2249.262 2306.892 2242.072 2143.228 2414.120 2214.853 2255.092 2299.539 2274.430 2252.157 2252.146 2246.660 2284.055
 [19] 2227.738 2333.841 2282.213 2375.611 2329.588 2204.115 2276.546 2252.866 2262.029 2261.815 2280.564 2294.245 2283.476 2267.981 2311.698 2258.921 2221.486 2171.942
 [37] 2237.948 2274.721 2350.939 2293.079 2341.119 2336.817 2338.820 2303.087 2274.151 2215.840 2294.925 2302.216 2244.857 2254.044 2219.639 2171.165 2213.226 2193.939
 [55] 2326.488 2323.543 2291.057 2345.485 2292.833 2290.601 2277.726 2346.150 2198.110 2250.717 2337.487 2291.511 2203.784 2279.323 2274.859 2248.972 2211.305 2328.656
 [73] 2168.395 2208.201 2248.238 2267.332 2285.899 2253.026 2201.463 2304.096 2294.781 2226.134 2178.827 2256.878 2240.467 2257.504 2180.124 2308.659 2266.780 2306.643
 [91] 2210.772 2269.322 2248.666 2187.570 2277.700 2244.142 2197.166 2249.973 2241.468 2237.913
```

Sensitivity:(mean: 0.835 variance:3.018e-05)

```
> sensitivityListR
  [1] 0.8280802 0.8351167 0.8389913 0.8342246 0.8337408 0.8423059 0.8460425 0.8323643 0.8297049 0.8432800 0.8322769 0.8386783 0.8404255 0.8459670 0.8428158 0.8292091
 [17] 0.8325359 0.8406145 0.8357211 0.8308718 0.8433735 0.8419261 0.8403643 0.8285303 0.8337373 0.8305583 0.8431953 0.8359338 0.8400978 0.8359073 0.8416667 0.8351861
 [33] 0.8363726 0.8451582 0.8409091 0.8292091 0.8350465 0.8297975 0.8341369 0.8327678 0.8389913 0.8427579 0.8358500 0.8442187 0.8302158 0.8258562 0.8293726 0.8361742
 [49] 0.8394231 0.8292919 0.8281623 0.8320537 0.8297767 0.8257722 0.8377990 0.8323699 0.8396135 0.8340528 0.8411037 0.8316252 0.8319731 0.8405728 0.8246036 0.8295066
 [65] 0.8385519 0.8363979 0.8346154 0.8366457 0.8342912 0.8319573 0.8301887 0.8322115 0.8351861 0.8354978 0.8319328 0.8417476 0.8391473 0.8245870 0.8310811 0.8460798
 [81] 0.8330088 0.8222863 0.8286680 0.8357349 0.8323587 0.8308216 0.8317984 0.8347660 0.8365612 0.8273839 0.8304432 0.8295509 0.8382710 0.8413926 0.8353511 0.8294909
 [97] 0.8373786 0.8456408 0.8283909 0.8321513
```

Precision:(mean:0.9152 variance:2.092e-05)

```
> pricisionListR
  [1] 0.9267771 0.9128123 0.9202128 0.9147122 0.9127409 0.9151139 0.9101765 0.9109226 0.9098143 0.9225053 0.9104000 0.9190628 0.9118573 0.9119958 0.9156627 0.9109808
 [17] 0.9172377 0.9181961 0.9147935 0.9256900 0.9133612 0.9217252 0.9231174 0.9131816 0.9173333 0.9157667 0.9071618 0.9174883 0.9114058 0.9227491 0.9132979 0.9142857
 [33] 0.9133477 0.9196661 0.9109481 0.9080765 0.9074468 0.9178667 0.9158285 0.9128587 0.9202128 0.9173866 0.9201072 0.9169732 0.9158730 0.9174705 0.9242585 0.9197917
 [49] 0.9160546 0.9149278 0.9184754 0.9116719 0.9067245 0.9135077 0.9220642 0.9162248 0.9152185 0.9235263 0.9198751 0.9126942 0.9193634 0.9239244 0.9147122 0.9124127
 [65] 0.9160877 0.9182195 0.9160950 0.9137110 0.9173249 0.9102019 0.9156884 0.9192777 0.9094737 0.9175911 0.9107143 0.9164905 0.9164021 0.9153182 0.9077491 0.9113990
 [81] 0.9139957 0.9150187 0.9060686 0.9138655 0.9133690 0.9134153 0.9074564 0.9185497 0.9152455 0.9190657 0.9083246 0.9221685 0.9137110 0.9086162 0.9170654 0.9186170
 [97] 0.9059874 0.9121274 0.9141631 0.9119171
```

Specificity:(mean:0.899 variance:2.978e-05)

```
> specificityListR
  [1] 0.9122919 0.8975000 0.9058971 0.8999375 0.8988206 0.8946692 0.8907828 0.8944724 0.8930145 0.9084639 0.8953271 0.9048811 0.8942065 0.8948686 0.8982301 0.8952978
 [17] 0.8997446 0.9008264 0.8956466 0.9100835 0.8950032 0.9081250 0.9070064 0.8958069 0.9026993 0.9033457 0.8925061 0.8968202 0.8963377 0.9084596 0.8991337 0.8979206
 [33] 0.9010449 0.9019108 0.8929471 0.8915361 0.8921265 0.9026549 0.8994943 0.8970496 0.9058971 0.9067073 0.9070493 0.9005664 0.8987906 0.9027164 0.9078608 0.9002591
 [49] 0.8984772 0.9002509 0.9013453 0.8931298 0.8951859 0.8977273 0.9054917 0.9000000 0.8984868 0.9083386 0.9009009 0.8985688 0.9033694 0.9071108 0.8984127 0.8986948
 [65] 0.9026055 0.8970684 0.8991117 0.8976773 0.8998724 0.8941766 0.9005664 0.9035533 0.8916194 0.9010780 0.8989590 0.9010025 0.9007538 0.9017522 0.8895202 0.8915663
 [81] 0.8995006 0.8989835 0.8876263 0.8958069 0.8990025 0.8986867 0.8901444 0.9007682 0.8934373 0.9075109 0.8898734 0.9085174 0.8979336 0.8897985 0.9019485 0.9027954
 [97] 0.8878446 0.8901734 0.8999375 0.8896820
```

Recall: (mean:0.835 variance:3.018e-05)

```
> recallListR
  [1] 0.8280802 0.8351167 0.8389913 0.8342246 0.8337408 0.8423059 0.8460425 0.8323643 0.8297049 0.8432800 0.8322769 0.8386783 0.8404255 0.8459670 0.8428158 0.8292091
 [17] 0.8325359 0.8406145 0.8357211 0.8308718 0.8433735 0.8419261 0.8403643 0.8285303 0.8337373 0.8305583 0.8431953 0.8359338 0.8400978 0.8359073 0.8416667 0.8351861
 [33] 0.8363726 0.8451582 0.8409091 0.8292091 0.8350465 0.8297975 0.8341369 0.8327678 0.8389913 0.8427579 0.8358500 0.8442187 0.8302158 0.8258562 0.8293726 0.8361742
 [49] 0.8394231 0.8292919 0.8281623 0.8320537 0.8297767 0.8257722 0.8377990 0.8323699 0.8396135 0.8340528 0.8411037 0.8316252 0.8319731 0.8405728 0.8246036 0.8295066
 [65] 0.8385519 0.8363979 0.8346154 0.8366457 0.8342912 0.8319573 0.8301887 0.8322115 0.8351861 0.8354978 0.8319328 0.8417476 0.8391473 0.8245870 0.8310811 0.8460798
 [81] 0.8330088 0.8222863 0.8286680 0.8357349 0.8323587 0.8308216 0.8317984 0.8347660 0.8365612 0.8273839 0.8304432 0.8295509 0.8382710 0.8413926 0.8353511 0.8294909
 [97] 0.8373786 0.8456408 0.8283909 0.8321513
```

Test Data:

```
> calVar(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 3.840812e-05 1.647513e+03 8.499122e-05 6.740429e-05 8.707357e-05 8.499122e-05 1.422074e-05
> calMean(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1]    0.8681358 3160.9620397    0.8385011    0.9223022    0.9071832    0.8385011    0.9402094
>
```

Accuracy: (mean:0.868 variance:3.84e-05)

```
> accListR
  [1] 0.8623519 0.8691887 0.8659982 0.8659982 0.8669098 0.8691887 0.8678213 0.8705561 0.8614403 0.8687329 0.8609845 0.8728350 0.8751139 0.8650866
 [15] 0.8550593 0.8755697 0.8769371 0.8609845 0.8783045 0.8696445 0.8746582 0.8678213 0.8650866 0.8669098 0.8696445 0.8719234 0.8760255 0.8687329
 [29] 0.8755697 0.8682771 0.8659982 0.8614403 0.8587056 0.8609845 0.8746582 0.8655424 0.8641750 0.8678213 0.8596171 0.8587056 0.8755697 0.8568824
 [43] 0.8682771 0.8710119 0.8605287 0.8623519 0.8582498 0.8737466 0.8646308 0.8646308 0.8751139 0.8737466 0.8837739 0.8673655 0.8751139 0.8609845
 [57] 0.8637192 0.8728350 0.8596171 0.8728350 0.8678213 0.8824066 0.8810392 0.8673655 0.8755697 0.8573382 0.8641750 0.8719234 0.8682771 0.8659982
 [71] 0.8632634 0.8769371 0.8701003 0.8577940 0.8678213 0.8614403 0.8701003 0.8755697 0.8696445 0.8737466 0.8764813 0.8646308 0.8691887 0.8678213
 [85] 0.8714676 0.8732908 0.8710119 0.8659982 0.8650866 0.8678213 0.8778487 0.8573382 0.8664540 0.8605287 0.8669098 0.8732908 0.8646308 0.8664540
 [99] 0.8764813 0.8605287
```

Sensitivity:(mean:0.839 variance:8.499e-05)

```
> sensitivityListR
  [1] 0.8354331 0.8433253 0.8390438 0.8371522 0.8369653 0.8326725 0.8460905 0.8477399 0.8333333 0.8397699 0.8259141 0.8429150 0.8485342 0.8308419
 [15] 0.8233890 0.8381902 0.8521242 0.8149920 0.8492823 0.8417671 0.8399682 0.8333333 0.8240377 0.8345098 0.8493151 0.8478088 0.8417671 0.8412189
 [29] 0.8513189 0.8342765 0.8308943 0.8318440 0.8224666 0.8450363 0.8512658 0.8332003 0.8467213 0.8371336 0.8281124 0.8290196 0.8433829 0.8234350
 [43] 0.8245476 0.8382114 0.8318863 0.8341270 0.8283228 0.8448819 0.8377953 0.8300000 0.8517928 0.8455760 0.8570241 0.8410175 0.8481117 0.8226601
 [57] 0.8338710 0.8399035 0.8169355 0.8338583 0.8321624 0.8496063 0.8531524 0.8451613 0.8487530 0.8229167 0.8338658 0.8443018 0.8429618 0.8435266
 [71] 0.8336000 0.8418908 0.8358086 0.8280359 0.8491344 0.8355731 0.8318863 0.8541833 0.8383518 0.8521183 0.8529177 0.8526570 0.8418491 0.8360258
 [85] 0.8410915 0.8460305 0.8460292 0.8420641 0.8334655 0.8415435 0.8484127 0.8284790 0.8348106 0.8270799 0.8302344 0.8463435 0.8374700 0.8392283
 [99] 0.8465819 0.8207395
```

Precision:(mean:0.922 variance:6.74e-05)

```
> pricisionListR
  [1] 0.9194107 0.9205934 0.9196507 0.9150268 0.9201420 0.9325044 0.9089302 0.9207580 0.9119217 0.9165919 0.9235556 0.9245115 0.9221239 0.9287599
 [15] 0.9151194 0.9463203 0.9213781 0.9333333 0.9317585 0.9217238 0.9344553 0.9210526 0.9357716 0.9292576 0.9141370 0.9220104 0.9332146 0.9209833
 [29] 0.9244792 0.9247312 0.9223827 0.9134701 0.9257294 0.9025862 0.9251935 0.9238938 0.9029720 0.9194991 0.9164444 0.9199304 0.9365217 0.9160714
 [43] 0.9407540 0.9246637 0.9189189 0.9187063 0.9176161 0.9306158 0.9212121 0.9146006 0.9239412 0.9167421 0.9254937 0.9208007 0.9206774 0.9184235
 [57] 0.9174800 0.9288256 0.9259598 0.9396628 0.9342682 0.9415358 0.9328098 0.9136879 0.9254386 0.9177837 0.9206349 0.9236172 0.9151943 0.9163072
 [71] 0.9188713 0.9314698 0.9217470 0.9095792 0.9058927 0.9167389 0.9360568 0.9225473 0.9280702 0.9205527 0.9246101 0.9028133 0.9185841 0.9224599
 [85] 0.9257951 0.9246275 0.9182058 0.9213003 0.9243624 0.9135472 0.9328098 0.9102222 0.9217082 0.9151625 0.9260595 0.9188225 0.9175439 0.9182058
 [99] 0.9317585 0.9248188
```

Specificity: (mean:0.907 variance:8.707e-05)

```
> specificityListR
  [1] 0.8993506 0.9034995 0.9020234 0.9022634 0.9057592 0.9185423 0.8947906 0.9013934 0.8973029 0.9048106 0.9081197 0.9113660 0.9089027 0.9122427
 [15] 0.8975454 0.9303371 0.9082474 0.9223404 0.9170213 0.9062171 0.9211087 0.9108607 0.9218241 0.9118607 0.8961175 0.9041534 0.9209694 0.9049630
 [29] 0.9077413 0.9122257 0.9107884 0.8992731 0.9087948 0.8816754 0.9064516 0.9086079 0.8860370 0.9068323 0.9009484 0.8998912 0.9203926 0.9008439
 [43] 0.9284940 0.9128631 0.8996764 0.9004283 0.8989247 0.9134199 0.9015152 0.9064386 0.9062833 0.9076305 0.9162462 0.9027778 0.9088115 0.9088115
 [57] 0.9025157 0.9158780 0.9150943 0.9264069 0.9178532 0.9274892 0.9181722 0.8962264 0.9106204 0.9027484 0.9044586 0.9082278 0.9005181 0.8962567
 [71] 0.9025424 0.9214064 0.9124236 0.8955533 0.8909276 0.8966631 0.9223301 0.9041534 0.9120172 0.9024390 0.9077413 0.8802521 0.9042664 0.9089958
 [85] 0.9113924 0.9091869 0.9031250 0.8994536 0.9078242 0.9006148 0.9175589 0.8945720 0.9076600 0.9028926 0.9143156 0.9068577 0.9005291 0.9021053
 [99] 0.9166667 0.9126316
```

Recall: (mean:0.835 variance:8.499e-05)

```
> recallListR
  [1] 0.8354331 0.8433253 0.8390438 0.8371522 0.8369653 0.8326725 0.8460905 0.8477399 0.8333333 0.8397699 0.8259141 0.8429150 0.8485342 0.8308419
 [15] 0.8233890 0.8381902 0.8521242 0.8149920 0.8492823 0.8417671 0.8399682 0.8333333 0.8240377 0.8345098 0.8493151 0.8478088 0.8417671 0.8412189
 [29] 0.8513189 0.8342765 0.8308943 0.8318440 0.8224666 0.8450363 0.8512658 0.8332003 0.8467213 0.8371336 0.8281124 0.8290196 0.8433829 0.8234350
 [43] 0.8245476 0.8382114 0.8318863 0.8341270 0.8283228 0.8448819 0.8377953 0.8300000 0.8517928 0.8455760 0.8570241 0.8410175 0.8481117 0.8226601
 [57] 0.8338710 0.8399035 0.8169355 0.8338583 0.8321624 0.8496063 0.8531524 0.8451613 0.8487530 0.8229167 0.8338658 0.8443018 0.8429618 0.8435266
 [71] 0.8336000 0.8418908 0.8358086 0.8280359 0.8491344 0.8355731 0.8318863 0.8541833 0.8383518 0.8521183 0.8529177 0.8526570 0.8418491 0.8360258
 [85] 0.8410915 0.8460305 0.8460292 0.8420641 0.8334655 0.8415435 0.8484127 0.8284790 0.8348106 0.8270799 0.8302344 0.8463435 0.8374700 0.8392283
 [99] 0.8465819 0.8207395
```

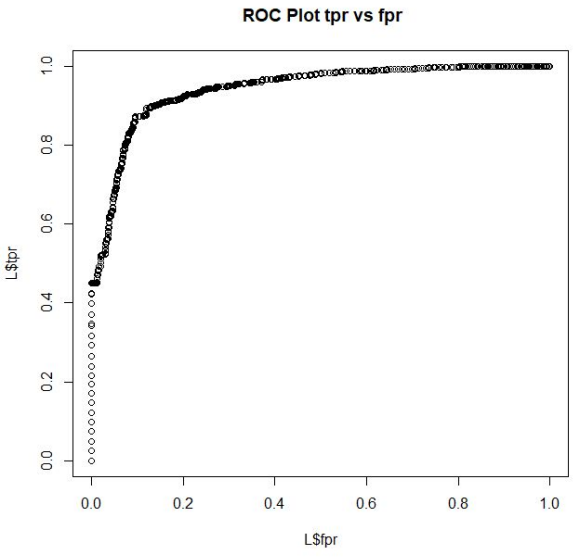
AUC:(mean:0.94 variance:1.422e-05)

```
> AUCListR
  [1] 0.9391056 0.9357698 0.9409427 0.9393716 0.9397393 0.9411518 0.9397473 0.9400682 0.9386437 0.9402212 0.9389469 0.9367420 0.9437371 0.9322164
 [15] 0.9395826 0.9492844 0.9393886 0.9406060 0.9442193 0.9381557 0.9399177 0.9421490 0.9422655 0.9358524 0.9365611 0.9438841 0.9410476 0.9376971
 [29] 0.9412605 0.9413689 0.9433236 0.9379748 0.9347467 0.9352515 0.9428657 0.9413085 0.9425666 0.9427449 0.9357098 0.9365198 0.9413385 0.9379057
 [43] 0.9414107 0.9388289 0.9430473 0.9401474 0.9275980 0.9461531 0.9386615 0.9422477 0.9408939 0.9461169 0.9459821 0.9373835 0.9419779 0.9418559
 [57] 0.9358541 0.9418066 0.9331702 0.9462890 0.9390348 0.9490428 0.9459229 0.9430681 0.9428077 0.9352404 0.9404454 0.9395379 0.9436542 0.9383043
 [71] 0.9437531 0.9429602 0.9413855 0.9362432 0.9381000 0.9390922 0.9421421 0.9446107 0.9430174 0.9443830 0.9421385 0.9394572 0.9394454 0.9422947
 [85] 0.9387157 0.9403532 0.9425008 0.9391350 0.9426939 0.9397165 0.9451860 0.9352250 0.9312461 0.9355375 0.9386370 0.9428208 0.9334523 0.9358292
 [99] 0.9475940 0.9348906
```

Confusion Matrix and ROC:

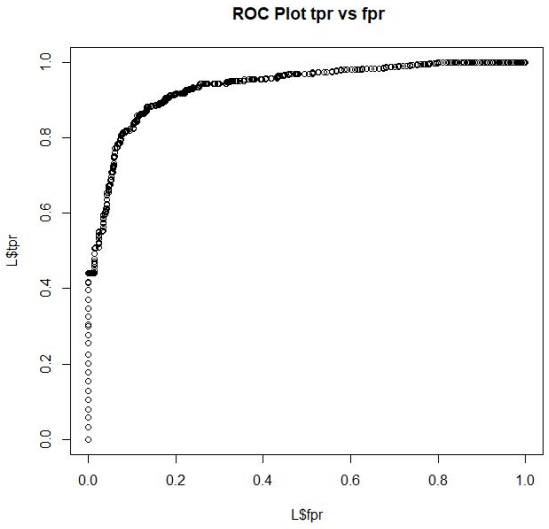
Training Data:

```
Confusion Matrix and Statistics

   predtrclass
      0    1
  0 2435  202
  1  478 2003

               Accuracy : 0.8671
                 95% CI : (0.8575, 0.8763)
    No Information Rate : 0.5692
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7331

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8359
            Specificity : 0.9084
         Pos Pred Value : 0.9234
         Neg Pred Value : 0.8073
             Prevalence : 0.5692
         Detection Rate : 0.4758
   Detection Prevalence : 0.5152
      Balanced Accuracy : 0.8721

       'Positive' Class : 0
```



Test Data:

```
Confusion Matrix and Statistics

   predtrclass
      0    1
 0 1021   83
 1  223  867

               Accuracy : 0.8605
                 95% CI : (0.8453, 0.8748)
    No Information Rate : 0.567
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7208

 Mcnemar's Test P-Value : 1.925e-15

            Sensitivity : 0.8207
            Specificity : 0.9126
         Pos Pred Value : 0.9248
         Neg Pred Value : 0.7954
             Prevalence : 0.5670
         Detection Rate : 0.4654
   Detection Prevalence : 0.5032
      Balanced Accuracy : 0.8667

       'Positive' Class : 0
```



Variance Estimation:

100% data:

```
> calVar(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 1.522955e-05 1.416869e+03 3.538958e-05 1.128552e-05 1.293813e-05 3.538958e-05 4.180062e-07
> calMean(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1]    0.8764409 3150.4238777    0.8506487    0.9280328    0.9112361    0.8506487    0.9443289
>
```

60% data:

```
>
> calVar(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 9.100402e-05 2.644565e+03 1.930658e-04 1.238548e-04 1.598830e-04 1.930658e-04 4.115619e-05
> calMean(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1]    0.8647844 3136.3456076    0.8325864    0.9224496    0.9074551    0.8325864    0.9396735
>
>
```

30% data:

```
> calVar(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 4.302812e-04 1.941874e+03 7.802329e-04 4.666852e-04 6.434098e-04 7.802329e-04 2.615264e-04
> calMean(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1]    0.8562963 3111.4370798    0.8326071    0.9028667    0.8864107    0.8326071    0.9253797
>
>
```

The variance of the accuracy of the varianceEstimation data is shown above. We can see that the variance is low and the accuracy is about 85% which is well-performance.

Result:

According to the result from prediction on training data and test data we can see that the mean accuracy of the model is around (86%~87%) and the AUC is around (94%) which means the model is good performing and not over-fitting.

NB:

For naiveBayes, I'm using the same dataset as I used in multinom and GLM. I also made a 100 for loop, the following results are predicting training dataset and test dataset (10% for variance estimation. In the rest of 90%, 30% are test data, 70% are training data).

Training dataset:

```
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 2.446076e-06 7.313339e-06 5.761276e-07 7.071497e-07 7.313339e-06 6.688450e-07
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9781324 0.9614365 0.9977403 0.9974801 0.9614365 0.9953433
```

Accuracy: (mean:0.978 variance:2.446e-06)

```
> accListR
  [1] 0.9779254 0.9785114 0.9802696 0.9789021 0.9779254 0.9749951 0.9777300 0.9798789 0.9806603 0.9785114 0.9783161 0.9800742 0.9806603 0.9781207 0.9777300 0.9771440
 [17] 0.9800742 0.9771440 0.9744091 0.9747998 0.9798789 0.9796835 0.9763626 0.9777300 0.9792928 0.9802696 0.9785114 0.9757765 0.9779254 0.9759719 0.9771440 0.9781207
 [33] 0.9777300 0.9785114 0.9798789 0.9779254 0.9779254 0.9763626 0.9781207 0.9798789 0.9767533 0.9746044 0.9790975 0.9781207 0.9802696 0.9796835 0.9779254 0.9789021
 [49] 0.9779254 0.9775347 0.9787068 0.9765579 0.9777300 0.9759719 0.9816370 0.9785114 0.9781207 0.9804649 0.9794882 0.9759719 0.9775347 0.9796835 0.9755812 0.9777300
 [65] 0.9773393 0.9814417 0.9783161 0.9755812 0.9763626 0.9767533 0.9775347 0.9794882 0.9773393 0.9769486 0.9769486 0.9798789 0.9781207 0.9783161 0.9773393 0.9773393
 [81] 0.9785114 0.9808556 0.9804649 0.9802696 0.9779254 0.9767533 0.9781207 0.9773393 0.9806603 0.9767533 0.9792928 0.9802696 0.9781207 0.9777300 0.9771440 0.9773393
 [97] 0.9761672 0.9789021 0.9769486 0.9773393
```

Sensitivity: (mean:0.961 variance:7.313e-06)

```
> sensitivityListR
  [1] 0.9600726 0.9628540 0.9653790 0.9633297 0.9615242 0.9547757 0.9607483 0.9643382 0.9660266 0.9614553 0.9609489 0.9645313 0.9654412 0.9621543 0.9607985 0.9606328
 [17] 0.9638290 0.9598422 0.9544139 0.9576362 0.9639971 0.9637681 0.9592579 0.9620299 0.9639308 0.9639476 0.9613984 0.9570242 0.9608918 0.9576909 0.9601305 0.9617084
 [33] 0.9604356 0.9615942 0.9665211 0.9595038 0.9612487 0.9578641 0.9625360 0.9644283 0.9590370 0.9558555 0.9632623 0.9609121 0.9650864 0.9634369 0.9612689 0.9628551
 [49] 0.9621543 0.9596922 0.9614124 0.9600726 0.9601171 0.9580946 0.9670088 0.9619048 0.9612880 0.9653775 0.9642336 0.9583333 0.9605310 0.9656684 0.9561435 0.9610484
 [65] 0.9603461 0.9671173 0.9615244 0.9569931 0.9576797 0.9596657 0.9603815 0.9642466 0.9605216 0.9588991 0.9599278 0.9630307 0.9616088 0.9616498 0.9602601 0.9595186
 [81] 0.9620438 0.9667529 0.9646373 0.9640499 0.9602649 0.9594497 0.9613042 0.9601160 0.9654030 0.9596182 0.9633028 0.9661449 0.9610058 0.9602026 0.9600719 0.9609916
 [97] 0.9576642 0.9623598 0.9593761 0.9600871
```

Precision: (mean:0.998 variance:5.761e-07)

```
> pricisionListR
  [1] 0.9988671 0.9965740 0.9977401 0.9969639 0.9965491 0.9988645 0.9973343 0.9977178 0.9973313 0.9988772 0.9984831 0.9985013 0.9980996 0.9969834 0.9977384 0.9961847
 [17] 0.9988641 0.9981357 0.9981082 0.9962963 0.9980981 0.9984985 0.9965986 0.9962193 0.9969547 0.9992450 0.9984871 0.9981168 0.9977230 0.9981315 0.9973645 0.9969466
 [33] 0.9981139 0.9984951 0.9958755 0.9992401 0.9981371 0.9981075 0.9970149 0.9981217 0.9981301 0.9965766 0.9973374 0.9984957 0.9977204 0.9984843 0.9969396 0.9977358
 [49] 0.9966076 0.9980945 0.9988654 0.9962335 0.9980981 0.9977620 0.9984860 0.9977204 0.9981217 0.9977160 0.9973575 0.9973832 0.9981357 0.9962321 0.9984860 0.9973555
 [65] 0.9977528 0.9981146 0.9977186 0.9977393 0.9980989 0.9969800 0.9973333 0.9973585 0.9973674 0.9977090 0.9973743 0.9992404 0.9973455 0.9981224 0.9977477 0.9981032
 [81] 0.9977290 0.9969524 0.9988675 0.9988597 0.9980880 0.9973654 0.9985114 0.9977393 0.9980974 0.9965688 0.9977195 0.9969947 0.9981075 0.9984951 0.9977570 0.9965974
 [97] 0.9977186 0.9984979 0.9977367 0.9977376
```

Specificity: (mean:0.997 variance:7.071e-07)

```
> specificityListR
  [1] 0.9987310 0.9962500 0.9974737 0.9966555 0.9962748 0.9987261 0.9970748 0.9974990 0.9970966 0.9987196 0.9983186 0.9983022 0.9979158 0.9966259 0.9974619 0.9958351
 [17] 0.9987406 0.9978541 0.9978769 0.9956710 0.9979141 0.9983044 0.9962025 0.9957983 0.9966694 0.9991572 0.9983144 0.9978723 0.9974822 0.9978541 0.9970339 0.9966708
 [33] 0.9978849 0.9983044 0.9953606 0.9991590 0.9978559 0.9978867 0.9965856 0.9978849 0.9978596 0.9962153 0.9970797 0.9983022 0.9974979 0.9983221 0.9966777 0.9974716
 [49] 0.9962041 0.9979079 0.9987352 0.9957699 0.9979044 0.9974216 0.9983271 0.9974885 0.9978769 0.9975042 0.9970576 0.9970021 0.9978559 0.9958001 0.9983051 0.9970489
 [65] 0.9974414 0.9979009 0.9974895 0.9974490 0.9978974 0.9966202 0.9970748 0.9970563 0.9970314 0.9974937 0.9970200 0.9991621 0.9970638 0.9978769 0.9974479 0.9978965
 [81] 0.9974779 0.9966833 0.9987374 0.9987463 0.9979175 0.9970301 0.9982818 0.9974587 0.9979184 0.9962422 0.9974937 0.9966273 0.9978947 0.9983015 0.9974348 0.9962121
 [97] 0.9974779 0.9983022 0.9974598 0.9974609
```

Recall: (mean:0.961 variance:7.313e-06)

```
> recallListR
  [1] 0.9600726 0.9628540 0.9653790 0.9633297 0.9615242 0.9547757 0.9607483 0.9643382 0.9660266 0.9614553 0.9609489 0.9645313 0.9654412 0.9621543 0.9607985 0.9606328
 [17] 0.9638290 0.9598422 0.9544139 0.9576362 0.9639971 0.9637681 0.9592579 0.9620299 0.9639308 0.9639476 0.9613984 0.9570242 0.9608918 0.9576909 0.9601305 0.9617084
 [33] 0.9604356 0.9615942 0.9665211 0.9595038 0.9612487 0.9578641 0.9625360 0.9644283 0.9590370 0.9558555 0.9632623 0.9609121 0.9650864 0.9634369 0.9612689 0.9628551
 [49] 0.9621543 0.9596922 0.9614124 0.9600726 0.9601171 0.9580946 0.9670088 0.9619048 0.9612880 0.9653775 0.9642336 0.9583333 0.9605310 0.9656684 0.9561435 0.9610484
 [65] 0.9603461 0.9671173 0.9615244 0.9569931 0.9576797 0.9596657 0.9603815 0.9642466 0.9605216 0.9588991 0.9599278 0.9630307 0.9616088 0.9616498 0.9602601 0.9595186
 [81] 0.9620438 0.9667529 0.9646373 0.9640499 0.9602649 0.9594497 0.9613042 0.9601160 0.9654030 0.9596182 0.9633028 0.9661449 0.9610058 0.9602026 0.9600719 0.9609916
 [97] 0.9576642 0.9623598 0.9593761 0.9600871
```

AUC: (mean:0.995 variance:6.688e-07)

```
> AUCListR
  [1] 0.9936024 0.9958282 0.9946502 0.9955824 0.9960382 0.9953424 0.9957577 0.9959001 0.9962090 0.9950137 0.9948491 0.9963306 0.9966330 0.9947173 0.9951178 0.9960799
 [17] 0.9951286 0.9948313 0.9949767 0.9943863 0.9973400 0.9950677 0.9952366 0.9961106 0.9961138 0.9956291 0.9966026 0.9949193 0.9955784 0.9954875 0.9954935 0.9969167
 [33] 0.9949205 0.9955623 0.9968317 0.9956239 0.9942560 0.9941158 0.9948376 0.9957610 0.9956408 0.9948990 0.9961845 0.9947339 0.9969310 0.9961716 0.9959450 0.9943826
 [49] 0.9942820 0.9949432 0.9947526 0.9959882 0.9955412 0.9939005 0.9957916 0.9959349 0.9936404 0.9955109 0.9951795 0.9953431 0.9946031 0.9966113 0.9967327 0.9954712
 [65] 0.9936300 0.9950714 0.9952142 0.9945156 0.9953461 0.9957368 0.9955689 0.9947091 0.9941602 0.9961976 0.9956271 0.9949077 0.9953863 0.9942757 0.9948936 0.9950373
 [81] 0.9959682 0.9961868 0.9951965 0.9964373 0.9955430 0.9953395 0.9951449 0.9951445 0.9960084 0.9946112 0.9943783 0.9965361 0.9960006 0.9945124 0.9932172 0.9955253
 [97] 0.9939689 0.9959341 0.9943901 0.9953178
```

Test dataset:

```
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 1.673719e-05 5.366342e-05 3.441162e-06 4.444353e-06 5.366342e-05 4.450070e-06
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9779024 0.9610436 0.9977765 0.9975026 0.9610436 0.9943002
>
```

Accuracy: (mean:0.978 variance:1.673e-05)

```
> accListR
  [1] 0.9781122 0.9835841 0.9758322 0.9822161 0.9808482 0.9785682 0.9740082 0.9685363 0.9785682 0.9803922 0.9762882 0.9762882 0.9808482 0.9781122 0.9790242 0.9849521
 [17] 0.9726402 0.9803922 0.9726402 0.9858641 0.9803922 0.9726402 0.9844961 0.9826721 0.9744642 0.9762882 0.9776562 0.9799362 0.9813041 0.9817601 0.9822161 0.9776562
 [33] 0.9872321 0.9781122 0.9735522 0.9776562 0.9803922 0.9785682 0.9703602 0.9781122 0.9735522 0.9694482 0.9721842 0.9730962 0.9822161 0.9753762 0.9781122 0.9799362
 [49] 0.9794802 0.9730962 0.9831281 0.9730962 0.9712722 0.9803922 0.9808482 0.9721842 0.9803922 0.9749202 0.9813041 0.9835841 0.9799362 0.9808482 0.9803922 0.9758322
 [65] 0.9799362 0.9799362 0.9730962 0.9803922 0.9744642 0.9744642 0.9753762 0.9831281 0.9753762 0.9776562 0.9854081 0.9744642 0.9799362 0.9776562 0.9781122 0.9781122
 [81] 0.9740082 0.9822161 0.9822161 0.9772002 0.9813041 0.9776562 0.9858641 0.9776562 0.9753762 0.9758322 0.9785682 0.9749202 0.9708162 0.9749202 0.9790242 0.9744642
 [97] 0.9740082 0.9676243 0.9822161 0.9790242
```

Sensitivity:(mean:0.961 variance:5.366e-05)

```
> sensitivityListR
  [1] 0.9609508 0.9682819 0.9602369 0.9689076 0.9648370 0.9603376 0.9549703 0.9464286 0.9640719 0.9645868 0.9564846 0.9572864 0.9649573 0.9611158 0.9610500 0.9738903
 [17] 0.9487179 0.9652248 0.9516949 0.9757525 0.9697733 0.9492691 0.9735974 0.9686971 0.9510917 0.9594937 0.9590793 0.9639564 0.9674936 0.9696203 0.9699115 0.9627175
 [33] 0.9761296 0.9597947 0.9524221 0.9593777 0.9649708 0.9615713 0.9461538 0.9597653 0.9526971 0.9441667 0.9529412 0.9602649 0.9681034 0.9564124 0.9619611 0.9653179
 [49] 0.9667802 0.9541735 0.9717707 0.9501718 0.9497161 0.9650767 0.9650350 0.9501689 0.9635733 0.9538729 0.9661922 0.9711375 0.9631902 0.9656652 0.9665831 0.9626736
 [65] 0.9640468 0.9683069 0.9576060 0.9647463 0.9529110 0.9551122 0.9574845 0.9692570 0.9562290 0.9618644 0.9730669 0.9531381 0.9634354 0.9593565 0.9606235 0.9616056
 [81] 0.9563025 0.9681475 0.9704890 0.9634855 0.9663300 0.9573171 0.9754653 0.9629941 0.9573743 0.9569257 0.9595070 0.9577703 0.9482902 0.9547782 0.9621343 0.9539749
 [97] 0.9534081 0.9461475 0.9698276 0.9625850
```

Precision:(mean:0.998 variance:3.441e-06)

```
> pricisionListR
  [1] 0.9982363 1.0000000 0.9947415 0.9982684 0.9991119 1.0000000 0.9964539 0.9974337 0.9955830 0.9991266 0.9991087 0.9991259 0.9991150 0.9982441 1.0000000 0.9973262
 [17] 1.0000000 0.9982456 0.9973357 0.9982891 0.9939759 0.9990950 0.9983080 0.9991274 1.0000000 0.9964943 0.9991119 0.9991312 0.9973545 0.9965308 0.9954587 0.9965695
 [33] 1.0000000 0.9991095 0.9972826 0.9982014 0.9991364 0.9982270 0.9981966 1.0000000 0.9991297 1.0000000 0.9956102 0.9906063 0.9982222 0.9982502 0.9973707 0.9982921
 [49] 0.9947415 0.9974337 0.9964912 0.9990967 0.9991468 0.9982379 0.9981917 0.9982254 0.9991007 0.9981785 0.9972452 0.9982548 0.9981835 0.9982254 0.9974138 0.9910634
 [65] 0.9991334 0.9948586 0.9931034 0.9982206 0.9991023 0.9982624 0.9944802 0.9991197 0.9982425 0.9964881 0.9991079 1.0000000 0.9991182 0.9991182 1.0000000 0.9963834
 [81] 0.9956255 0.9991349 0.9965368 0.9948586 0.9991297 1.0000000 0.9982684 0.9956522 0.9964508 0.9982379 0.9990834 0.9956102 0.9982441 0.9982159 0.9982143 0.9991236
 [97] 0.9972924 0.9947735 0.9964570 0.9982363
```

Specificity:(mean:0.998 variance:4.44e-06)

```
> specificityListR
  [1] 0.9980296 1.0000000 0.9940653 0.9980060 0.9990263 1.0000000 0.9960630 0.9968783 0.9951172 0.9990070 0.9990206 0.9989990 0.9990225 0.9980198 1.0000000 0.9971264
 [17] 1.0000000 0.9980276 0.9970385 0.9979940 0.9930140 0.9990291 0.9979613 0.9990109 1.0000000 0.9960317 0.9990196 0.9990000 0.9970703 0.9960317 0.9952963 0.9959432
 [33] 1.0000000 0.9990234 0.9971070 0.9980695 0.9989940 0.9980431 0.9980450 1.0000000 0.9989879 1.0000000 0.9950150 0.9888325 0.9980639 0.9980000 0.9970297 0.9979633
 [49] 0.9941119 0.9969104 0.9960938 0.9990282 0.9989583 0.9980373 0.9980934 0.9980178 0.9990385 0.9980843 0.9971936 0.9980296 0.9980989 0.9980545 0.9969880 0.9903939
 [65] 0.9989970 0.9939638 0.9919192 0.9980583 0.9990244 0.9979798 0.9943609 0.9990215 0.9980100 0.9960513 0.9990403 1.0000000 0.9990167 0.9990119 1.0000000 0.9961796
 [81] 0.9950150 0.9990000 0.9960278 0.9939271 0.9990050 1.0000000 0.9980218 0.9950199 0.9960784 0.9980178 0.9990539 0.9950446 0.9979879 0.9980411 0.9980601 0.9989980
 [97] 0.9970986 0.9939148 0.9961278 0.9980334
```

Recall: (mean:0.961 variance:5.366e-05)

```
> recallListR
  [1] 0.9609508 0.9682819 0.9602369 0.9689076 0.9648370 0.9603376 0.9549703 0.9464286 0.9640719 0.9645868 0.9564846 0.9572864 0.9649573 0.9611158 0.9610500 0.9738903
 [17] 0.9487179 0.9652248 0.9516949 0.9757525 0.9697733 0.9492691 0.9735974 0.9686971 0.9510917 0.9594937 0.9590793 0.9639564 0.9674936 0.9696203 0.9699115 0.9627175
 [33] 0.9761296 0.9597947 0.9524221 0.9593777 0.9649708 0.9615713 0.9461538 0.9597653 0.9526971 0.9441667 0.9529412 0.9602649 0.9681034 0.9564124 0.9619611 0.9653179
 [49] 0.9667802 0.9541735 0.9717707 0.9501718 0.9497161 0.9650767 0.9650350 0.9501689 0.9635733 0.9538729 0.9661922 0.9711375 0.9631902 0.9656652 0.9665831 0.9626736
 [65] 0.9640468 0.9683069 0.9576060 0.9647463 0.9529110 0.9551122 0.9574845 0.9692570 0.9562290 0.9618644 0.9730669 0.9531381 0.9634354 0.9593565 0.9606235 0.9616056
 [81] 0.9563025 0.9681475 0.9704890 0.9634855 0.9663300 0.9573171 0.9754653 0.9629941 0.9573743 0.9569257 0.9595070 0.9577703 0.9482902 0.9547782 0.9621343 0.9539749
 [97] 0.9534081 0.9461475 0.9698276 0.9625850
```

AUC:(mean:0.994 variance:4.450e-06)

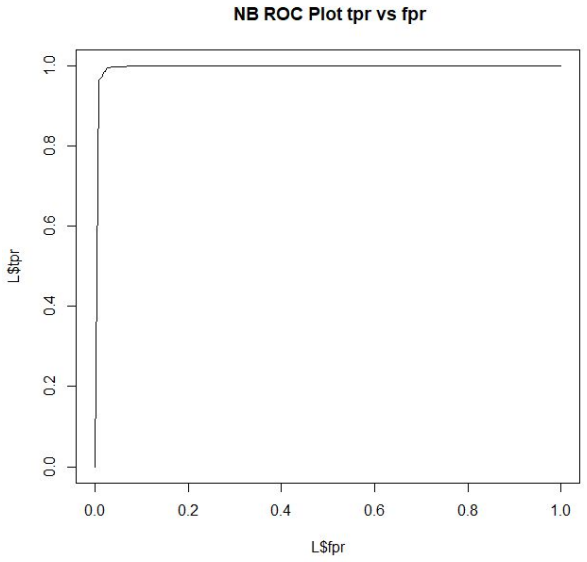
```
> AUCListR
  [1] 0.9946407 0.9971634 0.9956804 0.9975652 0.9977806 0.9948155 0.9905371 0.9904904 0.9965114 0.9923243 0.9927775 0.9932541 0.9971499 0.9942074 0.9938709 0.9977132
 [17] 0.9948616 0.9932840 0.9907082 0.9956025 0.9953131 0.9936776 0.9964812 0.9961587 0.9925552 0.9932656 0.9934462 0.9946271 0.9952811 0.9950043 0.9947866 0.9954375
 [33] 0.9982416 0.9943089 0.9932410 0.9935008 0.9945295 0.9963082 0.9916870 0.9926189 0.9940553 0.9930561 0.9904969 0.9901225 0.9951552 0.9954327 0.9974882 0.9945833
 [49] 0.9951526 0.9933015 0.9964783 0.9915850 0.9889872 0.9964891 0.9967444 0.9923680 0.9949783 0.9928766 0.9963581 0.9975360 0.9935444 0.9956471 0.9959029 0.9931853
 [65] 0.9936985 0.9968259 0.9930968 0.9928742 0.9941722 0.9911735 0.9937516 0.9970090 0.9917470 0.9911237 0.9958122 0.9909730 0.9933238 0.9931335 0.9906072 0.9951124
 [81] 0.9937333 0.9952681 0.9959496 0.9966167 0.9964720 0.9941417 0.9981012 0.9942182 0.9941368 0.9928391 0.9972565 0.9925719 0.9925427 0.9955817 0.9960895 0.9936680
 [97] 0.9945836 0.9908499 0.9971022 0.9909323
```

As we can see, the accuracy is about 99% in predicting both training data set and test data set, which means the model is not over-fitting.

Confusion Matrix and ROC:

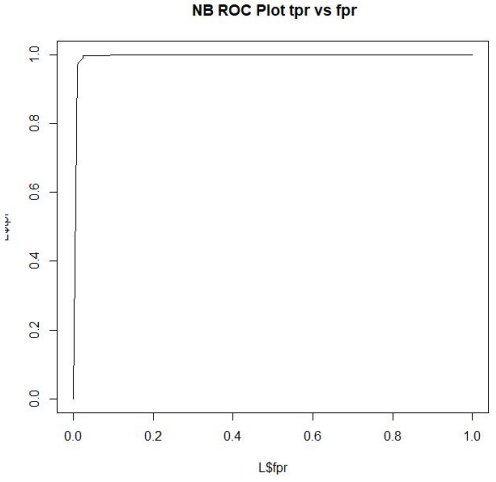
Training data:

```
> result
Confusion Matrix and Statistics

   nb.pred
       e    p
  e 2646    6
  p  110 2357

               Accuracy : 0.9773
                 95% CI : (0.9729, 0.9812)
    No Information Rate : 0.5384
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9546

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9601
            Specificity : 0.9975
         Pos Pred Value : 0.9977
         Neg Pred Value : 0.9554
             Prevalence : 0.5384
         Detection Rate : 0.5169
   Detection Prevalence : 0.5181
      Balanced Accuracy : 0.9788

       'Positive' Class : e
```



Test data:

```
Confusion Matrix and Statistics

   nb.pred
       e    p
  e 1127    2
  p   46 1018

               Accuracy : 0.9781
                 95% CI : (0.9711, 0.9838)
    No Information Rate : 0.5349
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9561

 Mcnemar's Test P-Value : 5.417e-10

            Sensitivity : 0.9608
            Specificity : 0.9980
         Pos Pred Value : 0.9982
         Neg Pred Value : 0.9568
             Prevalence : 0.5349
         Detection Rate : 0.5139
   Detection Prevalence : 0.5148
      Balanced Accuracy : 0.9794

       'Positive' Class : e
```



Variance Estimation:

100% data:

```
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 6.227647e-06 1.020342e-05 1.117584e-05 1.628125e-05 1.020342e-05 1.397400e-06
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9723768 0.9547938 0.9955862 0.9946604 0.9547938 0.9929750
>
>
>
```

we can see the variance of accuracy is 6.227e-06 which is pretty low

60% data:

```
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 1.600487e-05 4.716544e-05 9.941517e-06 1.504072e-05 4.716544e-05 4.889801e-06
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9802669 0.9696911 0.9948275 0.9936425 0.9696911 0.9943961
>
```

we can see the variance of accuracy is 1.6e-05

30% data:

```
>
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 6.483915e-05 1.623035e-04 3.077231e-05 4.300906e-05 1.623035e-04 1.298029e-05
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9809053 0.9712666 0.9937670 0.9926187 0.9712666 0.9959341
>
```

we can see the variance of accuracy is 6.48e-05

Result:
The ten features in kNN model is well-performance, not over-fitting and has nearly 97% of accuracy and 98% of AUC.

Knn:

For Knn, I'm using the same dataset as I used in multinom and GLM. I also made a 100 times for loop. The following results are predicting training dataset and test dataset (10% for variance estimation. In the rest of 90%, 30% are test data, 70% are training data).

Test Data:

```
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 9.439514e-07 2.184413e-06 1.604462e-06 1.869530e-06 2.184413e-06 9.634455e-07
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9991067 0.9986918 0.9995982 0.9995628 0.9986918 0.9990830
>
```

Accuracy (mean:0.999 variance:1.251e-06)

```
> accListR
  [1] 0.9990884 0.9990884 0.9981768 1.0000000 0.9949863 1.0000000 0.9940747 0.9990884 0.9972653 0.9968095 0.9963537 0.9995442 0.9981768 0.9995442
 [15] 0.9990884 0.9986326 1.0000000 1.0000000 1.0000000 0.9949863 0.9995442 0.9995442 0.9977211 0.9981768 0.9995442 0.9990884 1.0000000 0.9990884
 [29] 0.9990884 0.9986326 0.9990884 0.9990884 0.9977211 1.0000000 0.9990884 1.0000000 1.0000000 0.9990884 0.9990884 0.9995442 1.0000000 1.0000000
 [43] 1.0000000 0.9981768 0.9995442 1.0000000 1.0000000 0.9972653 0.9986326 0.9995442 1.0000000 0.9981768 0.9990884 0.9977211 0.9995442 0.9990884
 [57] 0.9986326 0.9972653 0.9986326 1.0000000 0.9990884 0.9981768 0.9990884 0.9995442 0.9981768 0.9977211 1.0000000 0.9995442 0.9995442 0.9990884
 [71] 0.9995442 1.0000000 0.9981768 0.9977211 0.9986326 0.9990884 0.9986326 1.0000000 0.9986326 0.9986326 1.0000000 0.9977211 0.9981768 0.9977211
 [85] 0.9995442 1.0000000 1.0000000 0.9990884 0.9990884 0.9986326 0.9995442 0.9990884 0.9990884 0.9990884 0.9990884 1.0000000 0.9986326 0.9995442
 [99] 0.9986326 0.9995442
```

Sensitivity:

```
> sensitivityListR
  [1] 1.0000000 0.9991424 0.9991327 0.9973776 1.0000000 1.0000000 1.0000000 0.9957301 1.0000000 1.0000000 1.0000000 0.9974003 1.0000000 1.0000000
 [15] 1.0000000 0.9991197 0.9965368 0.9991055 1.0000000 0.9947507 0.9965547 1.0000000 0.9982047 0.9982639 1.0000000 1.0000000 0.9991205 1.0000000
 [29] 1.0000000 0.9965035 0.9973545 0.9973451 1.0000000 0.9991364 0.9991394 1.0000000 0.9991166 1.0000000 0.9974249 1.0000000 1.0000000 1.0000000
 [43] 0.9991213 1.0000000 0.9982441 1.0000000 0.9982803 0.9991158 0.9982127 0.9973776 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9965066
 [57] 0.9982111 0.9991087 1.0000000 1.0000000 0.9974446 0.9973958 1.0000000 0.9982394 0.9991127 1.0000000 1.0000000 0.9982472 0.9991236 1.0000000
 [71] 1.0000000 0.9965368 0.9956822 1.0000000 0.9974315 0.9964881 0.9965547 0.9974705 0.9982410 1.0000000 0.9991220 0.9982578 1.0000000 1.0000000
 [85] 1.0000000 0.9982654 0.9974403 0.9991158 0.9974337 0.9964061 0.9973822 0.9929886 0.9982548 0.9973753 0.9991143 0.9982317 0.9955986 0.9973381
 [99] 1.0000000 0.9982517
```

Precision:

```
> pricisionListR
  [1] 1.0000000 0.9982862 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
 [15] 1.0000000 0.9991197 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9956484 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9974003
 [29] 1.0000000 1.0000000 1.0000000 0.9991135 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9920283 1.0000000 1.0000000 1.0000000 1.0000000
 [43] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9955117 0.9964974 0.9991023 1.0000000 1.0000000
 [57] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9973822 1.0000000 1.0000000 1.0000000
 [71] 1.0000000 1.0000000 1.0000000 1.0000000 0.9965783 1.0000000 0.9948409 0.9983122 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
 [85] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
 [99] 1.0000000 1.0000000
```

Specificity:

```
> specificityListR
  [1] 1.0000000 0.9980545 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
 [15] 1.0000000 0.9990548 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9952381 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9971237
 [29] 1.0000000 1.0000000 1.0000000 0.9990602 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9916201 1.0000000 1.0000000 1.0000000 1.0000000
 [43] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9953917 0.9962121 0.9990749 1.0000000 1.0000000
 [57] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9971456 1.0000000 1.0000000 1.0000000 1.0000000
 [71] 1.0000000 1.0000000 1.0000000 1.0000000 0.9961014 1.0000000 0.9941917 0.9980159 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
 [85] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
 [99] 1.0000000 1.0000000
```

Recall:

```
> recallListR
  [1] 1.0000000 0.9991424 0.9991327 0.9973776 1.0000000 1.0000000 1.0000000 0.9957301 1.0000000 1.0000000 1.0000000 0.9974003 1.0000000 1.0000000
 [15] 1.0000000 0.9991197 0.9965368 0.9991055 1.0000000 0.9947507 0.9965547 1.0000000 0.9982047 0.9982639 1.0000000 1.0000000 0.9991205 1.0000000
 [29] 1.0000000 0.9965035 0.9973545 0.9973451 1.0000000 0.9991364 0.9991394 1.0000000 0.9991166 1.0000000 0.9974249 1.0000000 1.0000000 1.0000000
 [43] 0.9991213 1.0000000 0.9982441 1.0000000 0.9982803 0.9991158 0.9982127 0.9973776 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9965066
 [57] 0.9982111 0.9991087 1.0000000 1.0000000 0.9974446 0.9973958 1.0000000 0.9982394 0.9991127 1.0000000 1.0000000 0.9982472 0.9991236 1.0000000
 [71] 1.0000000 0.9965368 0.9956822 1.0000000 0.9974315 0.9964881 0.9965547 0.9974705 0.9982410 1.0000000 0.9991220 0.9982578 1.0000000 1.0000000
 [85] 1.0000000 0.9982654 0.9974403 0.9991158 0.9974337 0.9964061 0.9973822 0.9929886 0.9982548 0.9973753 0.9991143 0.9982317 0.9955986 0.9973381
 [99] 1.0000000 0.9982517
```

AUC: (mean:0.999 variance:1.261e-06)

```
> AUCListR
  [1] 0.9990 0.9990 0.9980 1.0000 0.9953 1.0000 0.9940 0.9990 0.9970 0.9968 0.9965 0.9995 0.9981 0.9995 0.9990 0.9986 1.0000 1.0000 1.0000 0.9947
 [21] 0.9995 0.9995 0.9977 0.9981 0.9995 0.9990 1.0000 0.9990 0.9990 0.9986 0.9991 0.9991 0.9977 1.0000 0.9991 1.0000 1.0000 0.9991 0.9991 0.9995
 [41] 1.0000 1.0000 1.0000 0.9981 0.9995 1.0000 1.0000 0.9973 0.9985 0.9995 1.0000 0.9981 0.9991 0.9977 0.9995 0.9991 0.9986 0.9974 0.9986 1.0000
 [61] 0.9990 0.9982 0.9991 0.9995 0.9981 0.9976 1.0000 0.9995 0.9995 0.9990 0.9995 1.0000 0.9980 0.9978 0.9986 0.9991 0.9986 1.0000 0.9987 0.9987
 [81] 1.0000 0.9977 0.9981 0.9977 0.9995 1.0000 1.0000 0.9990 0.9990 0.9986 0.9995 0.9991 0.9991 0.9991 0.9990 1.0000 0.9985 0.9995 0.9986 0.9995
```

Training data:

```
> calvarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 9.891645e-08 3.055287e-07 9.797450e-08 1.160556e-07 3.055287e-07 1.036404e-07
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9996229 0.9994107 0.9998650 0.9998532 0.9994107 0.9996140
```

Accuracy (mean:0.999 variance:6.698e-08)

```
> accListR
  [1] 0.9994138 1.0000000 1.0000000 0.9998046 1.0000000 0.9996092 0.9998046 0.9996092 0.9996092 1.0000000 0.9996092 0.9994138 0.9992184 0.9998046
 [15] 0.9998046 1.0000000 0.9996092 0.9996092 1.0000000 1.0000000 1.0000000 1.0000000 0.9996092 1.0000000 0.9994138 1.0000000 1.0000000 0.9996092
 [29] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9996092 0.9994138 1.0000000 1.0000000 1.0000000 0.9996092 0.9992184 0.9998046
 [43] 0.9998046 1.0000000 0.9996092 0.9996092 0.9998046 1.0000000 0.9992184 0.9998046 0.9994138 1.0000000 1.0000000 1.0000000 0.9996092 1.0000000
 [57] 1.0000000 0.9992184 1.0000000 0.9996092 0.9996092 0.9998046 1.0000000 0.9998046 0.9992184 0.9998046 0.9992184 0.9996092 0.9998046 0.9994138
 [71] 0.9994138 0.9998046 0.9996092 1.0000000 0.9998046 1.0000000 0.9992184 0.9996092 0.9994138 1.0000000 0.9992184 0.9998046 1.0000000 0.9996092
 [85] 1.0000000 0.9996092 0.9998046 1.0000000 1.0000000 1.0000000 0.9998046 1.0000000 1.0000000 1.0000000 0.9998046 0.9992184 0.9998046 1.0000000
 [99] 0.9998046 0.9994138
```

AUC: (mean:0.999 variance:7.0181e-08)

```
> AUCListR
  [1] 0.9994 1.0000 1.0000 0.9998 1.0000 0.9996 0.9998 0.9996 0.9996 1.0000 0.9996 0.9994 0.9992 0.9998 0.9998 1.0000 0.9996 0.9996 1.0000 1.0000
 [21] 1.0000 1.0000 0.9996 1.0000 0.9994 1.0000 1.0000 0.9996 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 0.9996 0.9994 1.0000 1.0000 1.0000 0.9996
 [41] 0.9992 0.9998 0.9998 1.0000 0.9996 0.9996 0.9998 1.0000 0.9992 0.9998 0.9994 1.0000 1.0000 1.0000 0.9996 1.0000 1.0000 0.9992 1.0000 0.9996
 [61] 0.9996 0.9998 1.0000 0.9998 0.9992 0.9998 0.9992 0.9996 0.9998 0.9994 0.9994 0.9998 0.9996 1.0000 0.9998 1.0000 0.9992 0.9996 0.9994 1.0000
 [81] 0.9992 0.9998 1.0000 0.9996 1.0000 0.9996 0.9998 1.0000 1.0000 1.0000 0.9998 1.0000 1.0000 1.0000 0.9998 0.9992 0.9998 1.0000 0.9998 0.9994
```

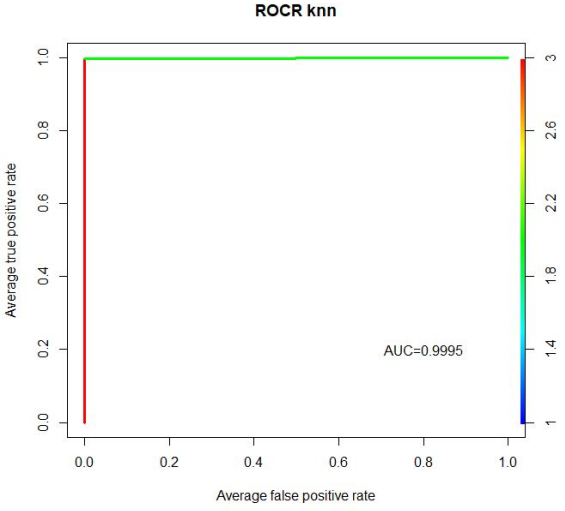
others:

```
> sensitivityListR
  [1] 0.9996267 1.0000000 1.0000000 1.0000000 0.9977316 0.9988793 0.9985052 0.9992478 0.9984968 0.9984911 0.9992562 0.9992604 0.9981182 0.9996253
 [15] 0.9992393 0.9996277 1.0000000 0.9996242 0.9984979 0.9988768 0.9992444 0.9996222 0.9996232 1.0000000 0.9992584 0.9977620 0.9996277 0.9992543
 [29] 1.0000000 1.0000000 0.9992537 1.0000000 0.9988806 0.9992487 0.9996241 0.9988688 0.9996271 1.0000000 0.9996262 0.9996273 0.9988696 0.9992461
 [43] 1.0000000 1.0000000 1.0000000 0.9992487 0.9996229 0.9992478 0.9996246 0.9996260 1.0000000 0.9988701 0.9996236 0.9996245 0.9996274 0.9996218
 [57] 0.9996221 0.9996255 0.9996236 1.0000000 0.9996248 0.9988701 1.0000000 0.9996292 1.0000000 1.0000000 0.9996169 0.9988688 0.9988798 0.9992470
 [71] 0.9996270 0.9988760 1.0000000 1.0000000 0.9977562 0.9988726 0.9984848 1.0000000 0.9996229 0.9996241 0.9992498 1.0000000 0.9984917 1.0000000
 [85] 0.9996238 0.9992504 1.0000000 0.9996259 0.9996225 0.9996285 1.0000000 0.9996246 1.0000000 0.9985024 0.9996201 0.9996252 0.9992492 0.9988810
 [99] 0.9988692 0.9996232
> pricisionListR
  [1] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9992604 1.0000000 1.0000000
 [15] 1.0000000 1.0000000 0.9992512 0.9992487 1.0000000 0.9996253 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9996270
 [29] 1.0000000 0.9996299 1.0000000 1.0000000 1.0000000 0.9996242 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9992461
 [43] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9992551 1.0000000
 [57] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9996169 1.0000000 1.0000000 1.0000000
 [71] 0.9985097 0.9992504 1.0000000 0.9992461 1.0000000 0.9992481 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9984957
 [85] 1.0000000 0.9992504 0.9992337 1.0000000 1.0000000 0.9992573 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
 [99] 1.0000000 0.9996232
> specificityListR
  [1] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9991715 1.0000000 1.0000000
 [15] 1.0000000 1.0000000 0.9991833 0.9991860 1.0000000 0.9995913 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9995895
 [29] 1.0000000 0.9995863 1.0000000 1.0000000 1.0000000 0.9995928 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9991886
 [43] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9991783 1.0000000
 [57] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9996013 1.0000000 1.0000000 1.0000000
 [71] 0.9983586 0.9991833 1.0000000 0.9991893 1.0000000 0.9991860 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9983760
 [85] 1.0000000 0.9991837 0.9992032 1.0000000 1.0000000 0.9991756 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
 [99] 1.0000000 0.9995942
> recallListR
  [1] 0.9996267 1.0000000 1.0000000 1.0000000 0.9977316 0.9988793 0.9985052 0.9992478 0.9984968 0.9984911 0.9992562 0.9992604 0.9981182 0.9996253
 [15] 0.9992393 0.9996277 1.0000000 0.9996242 0.9984979 0.9988768 0.9992444 0.9996222 0.9996232 1.0000000 0.9992584 0.9977620 0.9996277 0.9992543
 [29] 1.0000000 1.0000000 0.9992537 1.0000000 0.9988806 0.9992487 0.9996241 0.9988688 0.9996271 1.0000000 0.9996262 0.9996273 0.9988696 0.9992461
 [43] 1.0000000 1.0000000 1.0000000 0.9992487 0.9996229 0.9992478 0.9996246 0.9996260 1.0000000 0.9988701 0.9996236 0.9996245 0.9996274 0.9996218
 [57] 0.9996221 0.9996255 0.9996236 1.0000000 0.9996248 0.9988701 1.0000000 0.9996292 1.0000000 1.0000000 0.9996169 0.9988688 0.9988798 0.9992470
 [71] 0.9996270 0.9988760 1.0000000 1.0000000 0.9977562 0.9988726 0.9984848 1.0000000 0.9996229 0.9996241 0.9992498 1.0000000 0.9984917 1.0000000
 [85] 0.9996238 0.9992504 1.0000000 0.9996259 0.9996225 0.9996285 1.0000000 0.9996246 1.0000000 0.9985024 0.9996201 0.9996252 0.9992492 0.9988810
 [99] 0.9988692 0.9996232
```

Confusion Matrix and Roc:

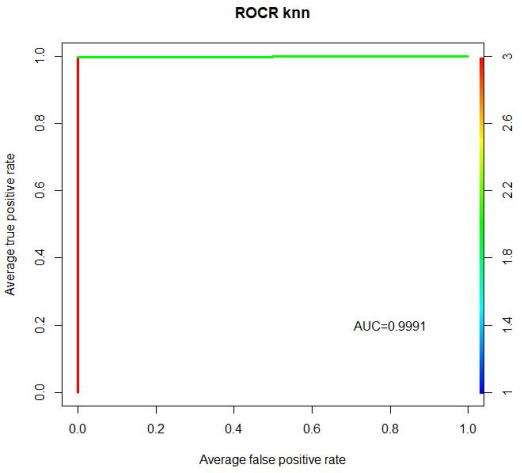
Training data:

```
Confusion Matrix and Statistics

   knnPred
      0    1
 0 2660    0
 1    3 2455

               Accuracy : 0.9994
                 95% CI : (0.9983, 0.9999)
    No Information Rate : 0.5203
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9988

 Mcnemar's Test P-Value : 0.2482

            Sensitivity : 0.9989
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9988
             Prevalence : 0.5203
         Detection Rate : 0.5197
   Detection Prevalence : 0.5197
      Balanced Accuracy : 0.9994

       'Positive' Class : 0
```



Test data:

```
Confusion Matrix and Statistics

   knnPred
      0    1
 0 1102    0
 1    2 1090

               Accuracy : 0.9991
                 95% CI : (0.9967, 0.9999)
    No Information Rate : 0.5032
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9982

 Mcnemar's Test P-Value : 0.4795

            Sensitivity : 0.9982
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9982
             Prevalence : 0.5032
         Detection Rate : 0.5023
   Detection Prevalence : 0.5023
      Balanced Accuracy : 0.9991

       'Positive' Class : 0
```



Variace Estimation:

100%data:

I'm using the model to predict the data in varianceEstimation dataset:

```
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 1.303714e-07 5.642035e-08 4.374814e-07 4.983718e-07 5.642035e-08 1.261364e-07
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9999138 0.9999762 0.9998571 0.9998475 0.9999762 0.9999150
>
```

we can see the variance of accuracy is 1.304e-07 which is pretty low

60%data:

```
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 1.112023e-06 3.366120e-06 2.823590e-07 4.009410e-07 3.366120e-06 1.291122e-06
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9993224 0.9988320 0.9999245 0.9999100 0.9988320 0.9992670
>
```

we can see the variance of accuracy is 1.112e-06

30%data:

```
>
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 5.917017e-06 2.740535e-06 1.868174e-05 2.218541e-05 2.740535e-06 5.430707e-06
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9991358 0.9996638 0.9986412 0.9985699 0.9996638 0.9991600
>
>
```

we can see the variance of accuracy is 5.917e-06

In my kNN model, as the variance estimation uses more data, the variance get smaller.

Result:
The ten features in kNN model is well-performance, not over-fitting and has nearly 100% accuracy and AUC.

Motinom:

My basic thought is firstly, divide the dataset into 10% and 90%. 10% is for variance estimation, and as we can see, I set them as unreplaceable. Because I will use them to calculate the variance for 100 different datasets.

```
mush10percent<-sample(1:nrow(mushroom_rf),nrow(mushroom_rf)*0.10,replace=F)
```

I'm using the 10 features that I decided through randomForest, and get "mush10percent" and "mush90TR" for 10% of data for variance estimation and 90% of data for training.

I made a 100 times for loop, and for each loop I record accuracy, AIC, Residual Deviance, sensitivity, precision, specificity and recall and made three lists for them.

```r
formulaStr<-"poisonous~."

accList<-list()
AICList<-list()
devianceList<-list()
sensitivityList<-list()
pricisionList<-list()
specificityList<-list()
recallList<-list()


for (i in 1:100) {


  tr50<-sample(1:nrow(mush90TR),0.5*nrow(mush90TR),replace = F)
  #trainingSet<-mush90TR[sample(1:nrow(mush90TR),0.5*nrow(mush90TR),replace=F),]
  trainingSet<-mush90TR[tr50,c(1:11)]
  tst<-mush90TR[-tr50,c(2:11)]
  tstR<-mush90TR[-tr50,c(1:11)]
  model<-multinom(formulaStr,data=trainingSet)

  #predictions<-predict(model,predictors,probability=TRUE)
  predictions<-predict(model,tst,probability=TRUE)
  #result = caret::confusionMatrix(table(m10p[,1],predictions))
  result = caret::confusionMatrix(table(tstR[,1],predictions))
  acc = result$overall[1]
  accList <- c(accList,acc)
  AICList <- c(AICList,model$AIC)
  devianceList <- c(devianceList,model$deviance)
  sensitivityList<-c(sensitivityList,result$byClass[1])
  pricisionList<-c(pricisionList,result$byClass[5])
  specificityList<-c(specificityList,result$byClass[2])
  recallList<-c(recallList,result$byClass[6])

}
```

Note: the comment lines predict the variance estimation data, and current loop is predict the test data, I will list these two results.

Test data:

```
> calVar(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.000000e+00 4.351884e-06 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
> calMean(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 1.00000000 0.01078527 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000
>
```

Details:

```
> accListR
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [72] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> sensitivityListR
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [72] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> devianceListR
  [1] 0.0113207966 0.0129895467 0.0107599192 0.0094444391 0.0107340403 0.0082166938 0.0108170480 0.0001400185 0.0112026096 0.0111191959
 [11] 0.0120407460 0.0093182413 0.0123925308 0.0096367639 0.0096711432 0.0125276254 0.0118002223 0.0097565924 0.0130311941 0.0100172312
 [21] 0.0126223777 0.0104941853 0.0114810073 0.0105661884 0.0100864671 0.0147504248 0.0108959214 0.0112006760 0.0091918041 0.0111662851
 [31] 0.0086429483 0.0111881899 0.0135633208 0.0108795113 0.0098684407 0.0085873522 0.0102258444 0.0095810988 0.0120186182 0.0130397555
 [41] 0.0130422371 0.0001480485 0.0121191818 0.0079923094 0.0105138580 0.0103262544 0.0105753993 0.0112314297 0.0098340678 0.0106789559
 [51] 0.0100342088 0.0080411599 0.0090281523 0.0106948605 0.0117226411 0.0129018235 0.0118929797 0.0102306107 0.0128694992 0.0127871649
 [61] 0.0140230233 0.0104099866 0.0127818238 0.0127330647 0.0095297303 0.0105024683 0.0120165382 0.0115876990 0.0104345805 0.0105787043
 [71] 0.0123299493 0.0117400920 0.0120502202 0.0109310090 0.0117341210 0.0102424565 0.0112686269 0.0090241862 0.0081060059 0.0104898550
 [81] 0.0109561968 0.0105144927 0.0133078156 0.0124082823 0.0103941500 0.0106126431 0.0135507503 0.0133130471 0.0108500467 0.0121115005
 [91] 0.0090031250 0.0107781895 0.0102596680 0.0102827715 0.0108877186 0.0114425301 0.0108387483 0.0119104599 0.0077811980 0.0111574070
> pricisionListR
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [72] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> specificityListR
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [72] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> recallListR
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [72] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> AUCListR
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [72] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

The accuracy is 100%

Training Data:

```
>
> calVar(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.000000e+00 4.990304e-06 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
> calMean(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 1.00000000 0.01095728 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000
>
```

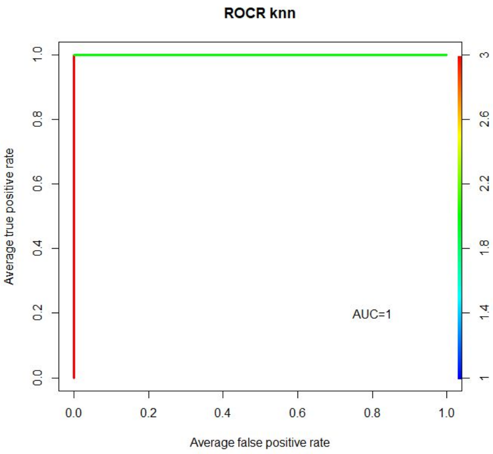
```
> accListR
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [72] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> devianceListR
  [1] 0.011263803 0.012589038 0.012664011 0.009739697 0.011931729 0.012315800 0.010960991 0.008468134 0.010883691 0.009128723 0.011817096
 [12] 0.011270104 0.010198013 0.010205842 0.010203374 0.012021822 0.012287545 0.012791337 0.010797911 0.012685610 0.006964820 0.019515955
 [23] 0.010004500 0.012155896 0.009749465 0.011567049 0.009679024 0.016371699 0.008558285 0.011531450 0.008543996 0.011507818 0.010053220
 [34] 0.011323238 0.009206538 0.013617180 0.008741210 0.015074896 0.009273962 0.009851348 0.008742918 0.013785414 0.011044818 0.009991429
 [45] 0.009186979 0.010280379 0.013183721 0.008287843 0.010309428 0.011463386 0.015365163 0.010114396 0.012031684 0.013410678 0.008284880
 [56] 0.013688151 0.009618054 0.008784036 0.010781410 0.010070255 0.008908291 0.017959312 0.009715482 0.007890779 0.011485017 0.010031929
 [67] 0.011575350 0.008911941 0.010926531 0.012760555 0.010672202 0.007350293 0.012214115 0.007727521 0.009692247 0.008975837 0.008482954
 [78] 0.009626226 0.010518608 0.009393198 0.012172789 0.011578185 0.015337736 0.012774397 0.013823833 0.008853742 0.014995081 0.011223796
 [89] 0.010444810 0.011185851 0.010005167 0.011846048 0.008703923 0.009960087 0.009621028 0.010413672 0.014931830 0.010587036 0.007424780
[100] 0.009084727
> sensitivityListR
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [72] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> pricisionListR
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [72] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> specificityListR
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [72] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> recallListR
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [72] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

The accuracy is also 100%

Confusion Matrix and ROC:

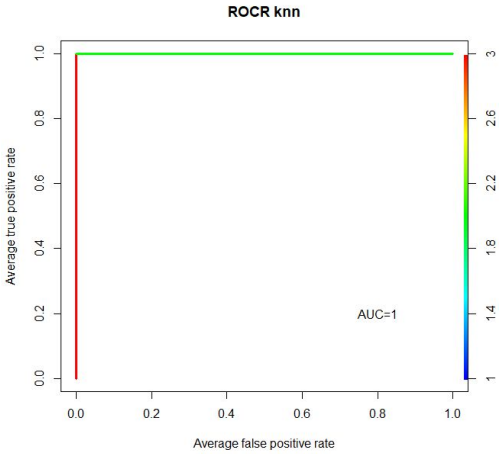
Training Data:

```
Confusion Matrix and Statistics

   predictions
       e    p
  e 2655    0
  p    0 2463

               Accuracy : 1
                 95% CI : (0.9993, 1)
    No Information Rate : 0.5188
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.5188
         Detection Rate : 0.5188
   Detection Prevalence : 0.5188
      Balanced Accuracy : 1.0000

       'Positive' Class : e
```



Test Data:

```
> result
Confusion Matrix and Statistics

          predictions
             0    1
        0 1687    0
        1    0 1563

                    Accuracy : 1
                      95% CI : (0.9989, 1)
       No Information Rate : 0.5191
       P-Value [Acc > NIR] : < 2.2e-16

                       Kappa : 1

     Mcnemar's Test P-Value : NA

                  Sensitivity : 1.0000
                  Specificity : 1.0000
               Pos Pred Value : 1.0000
               Neg Pred Value : 1.0000
                   Prevalence : 0.5191
               Detection Rate : 0.5191
         Detection Prevalence : 0.5191
            Balanced Accuracy : 1.0000

             'Positive' Class : 0
```



VarianceEstimationPartition Results:

30% Data:

```
>
> calVar(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.000000e+00 3.329885e-06 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
> calMean(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 1.00000000 0.01072639 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000
>
```

60% Data:

```
>
> calVar(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.000000e+00 6.824064e-06 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
> calMean(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 1.00000000 0.01022557 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000
>
```

100% Data:

```
>
> calVar(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.000000e+00 1.172054e-05 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
> calMean(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 1.000000000 0.009931759 1.000000000 1.000000000 1.000000000 1.000000000 1.000000000
>
```

According to those results, when we use the top 10 features from randomForest with 8124 observations (3250 of them are used to training/3250 of them are used to test/1624 of them are used to do variance estimation) the accuracy is almost 1, and the residual deviance is pretty low (around 0.0001~0.009).

VarianceEstimation of the model is also 1.

2. Null Deviance/Residual Deviance

I write an R script to test the best dataset in Null Deviance or Residual Deviance.
The script can randomly select x columns in the datasets and combine it with the target feature to generate a training set, and it will collect the residual deviance of the set and the features the model was using. I know this is not the most rigorous method but I run it a 1000 times for loop which make it more reasonable.

```
mushroomRD<-read.csv("./mushroom.csv",sep=',',head=T,stringsAsFactors = F)
target<-mushroomRD[c(1)]
mushroomRD<-mushroomRD[-c(1)]

devianceLowest<-10000000

for (i in 1:1000) {
   rd<-sample(mushroomRD, size=15)
   #training<-mushroomRD[rd,c(1:10)]
   trset<-cbind(rd,target)
   model<-glm("poisonous~.",data=trset, family = "binomial")

   if(model$deviance < devianceLowest){
      devianceLowest <- model$deviance
      namesRD<-c(names(rd))
   }

}

devianceLowest
namesRD
```

First, I run it by selecting 10 features. (size=10)

```
> devianceLowest
[1] 2346.941
> namesRD
 [1] "gillSize"       "odor"           "stalkRoot"      "stalkShape"     "sporePrintColor" "gillSpacing"    "bruises"
 [8] "ringType"       "SCBR"           "capSurface"
```

Second, I run it by selecting 15 features. (size = 15)

```
> devianceLowest
[1] 2076.945
> namesRD
 [1] "capSurface"     "capColor"       "sporePrintColor" "gillSize"       "ringType"       "stalkRoot"      "capShape"
 [8] "SCAR"           "SCBR"           "gillAttachment"  "bruises"        "gillSpacing"    "odor"           "stalkShape"
[15] "population"
```

Finally, I run it by selecting 20 features. (size = 20)

```
>
> devianceLowest
[1] 2012.721
> namesRD
 [1] "SSBR"           "stalkShape"     "SSAR"           "capShape"       "SCAR"           "bruises"        "SCBR"
 [8] "capColor"       "capSurface"     "odor"           "ringType"       "sporePrintColor" "gillColor"      "habitat"
[15] "stalkRoot"      "gillSpacing"    "gillSize"       "population"     "ringNumber"     "gillAttachment"
```

We can see that, as the number of features becomes larger, the residual deviance gets lower. Under that situation, I'm going to take both 10 size dataset and 20 size dataset as comparison datasets. I will run these two datasets 100 times in GLM, NB and kNN to compare the accuracy.

Table of Residual Deviance:

	Size=10	Size=13	Size=15	Size=17	Size=20
Residual Deviance	**2346.941**	**2185.469**	**2076.945**	**2034.12**	**2012.721**

size=17:

```
> devianceLowest
[1] 2034.12
> namesRD
 [1] "stalkShape"      "SSAR"            "bruises"         "gillSpacing"     "SCAR"            "ringType"
 [7] "habitat"         "gillColor"       "stalkRoot"       "gillSize"        "sporePrintColor" "gillAttachment"
[13] "population"      "capShape"        "capSurface"      "SCBR"            "capColor"
>
```

size=13:

```
> devianceLowest
[1] 2185.469
> namesRD
 [1] "ringType"      "odor"          "bruises"       "gillSpacing"   "SCBR"          "gillColor"     "capSurface"
 [8] "population"    "habitat"       "stalkRoot"     "gillAttachment" "SSAR"          "gillSize"
> .
```

3. Comparation

size 10 dataset:

GLM: (acc:0.95 var:2.35e-05)

```
>
> calVar(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 2.349007e-05 2.141205e+03 6.651028e-05 5.615082e-05 5.867950e-05 6.651028e-05 5.843065e-06
> calMean(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1]    0.9533227 1479.7536034    0.9586009    0.9509173    0.9478438    0.9586009    0.9828792
>
```

NB: (acc:0.97 var:6.07e-08)

```
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 6.066636e-08 1.765516e-10 2.224991e-07 2.521144e-07 1.765516e-10 3.455475e-08
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9728818 0.9718297 0.9763679 0.9740431 0.9718297 0.9814574
>
>
>
```

kNN:(acc:0.99 var:2.71e-07)

```
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 2.711993e-07 9.797802e-07 0.000000e+00 0.000000e+00 9.797802e-07 2.963879e-07
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9998815 0.9997748 1.0000000 1.0000000 0.9997748 0.9998760
>
```

size 20 dataset:

GLM:(acc:0.91 var:2.94e-05)

```
> calVar(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 2.938309e-05 3.108348e+03 8.778615e-05 2.455466e-05 3.041476e-05 8.778615e-05 1.038790e-05
> calMean(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1]    0.9120009 2651.0292258    0.8797142    0.9617988    0.9543522    0.8797142    0.9629067
>
```

NB:(acc:0.99 var:5.90e-07)

```
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 5.899650e-07 8.388627e-07 1.071229e-06 1.465734e-06 8.388627e-07 9.395505e-07
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9902340 0.9859028 0.9960550 0.9953716 0.9859028 0.9960625
>
```

kNN:(acc:0.99 var:5.6e-06)

```
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 5.600672e-06 1.800569e-05 4.013020e-06 4.743675e-06 1.800569e-05 6.053813e-06
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9924795 0.9877281 0.9979122 0.9977207 0.9877281 0.9922750
>
>
```

randomForest dataset:

GLM:(acc:0.87 var:3.84e-05)

```
> calVar(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 3.840812e-05 1.647513e+03 8.499122e-05 6.740429e-05 8.707357e-05 8.499122e-05 1.422074e-05
> calMean(accListR, devianceListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1]    0.8681358 3160.9620397    0.8385011    0.9223022    0.9071832    0.8385011    0.9402094
>
```

NB:(acc:0.98 var:1.67e-05)

```
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 1.673719e-05 5.366342e-05 3.441162e-06 4.444353e-06 5.366342e-05 4.450070e-06
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9779024 0.9610436 0.9977765 0.9975026 0.9610436 0.9943002
>
```

kNN:(acc:0.99 var:9.89e-08)

```
> calVarNB(accListR,  sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 9.891645e-08 3.055287e-07 9.797450e-08 1.160556e-07 3.055287e-07 1.036404e-07
> calMeanNB(accListR, sensitivityListR, pricisionListR, specificityListR, recallListR, AUCListR)
[1] 0.9996229 0.9994107 0.9998650 0.9998532 0.9994107 0.9996140
```

According to my results, I tabulate tables of the results of those dataset.

Table of accuracy

	randomForest	size=10	size=20
GLM	0.87	0.95	0.91
NB	0.98	0.97	0.99
kNN	0.99	0.99	0.99

Conclusion: For my dataset, the data selected from residual deviance has higher accuracy when running the same algorithm.

Appendix :

Results of table(mushroom$features):

	poisonous	**capShape**	**capSurface**	**capColor**	**bruises**	**odor**	**gillAttachment**
Balance?	Yes	No	No	No	Yes	No	No
Missing data?	No	No	No	No	No	No	No

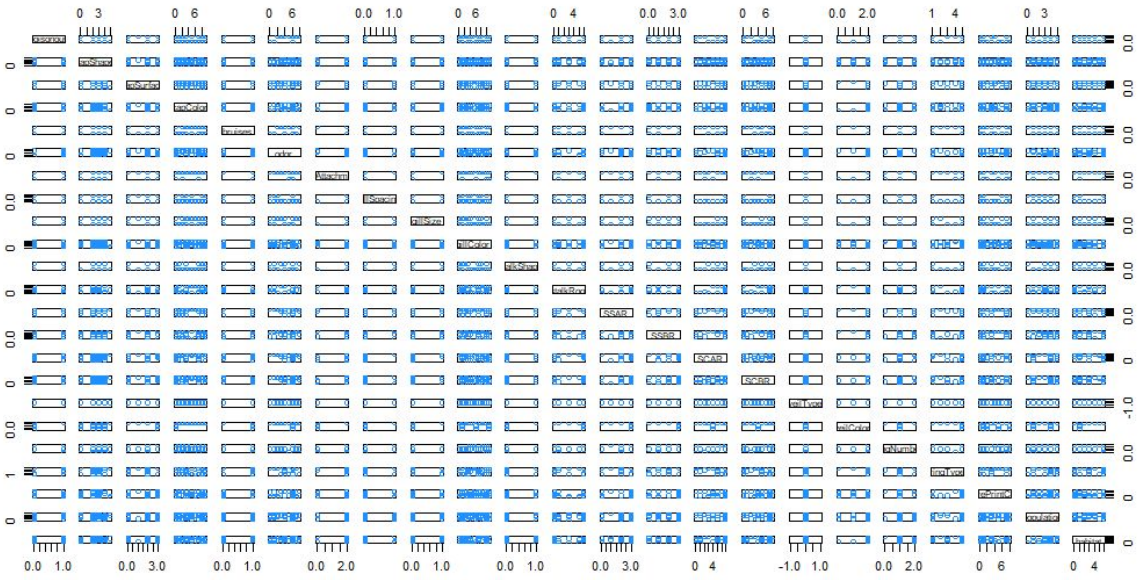
gillSpacing	**gillSize**	**gillColor**	**stalkShape**	**stalkRoot**	**SSAR**	**SSBR**	**SCAR**
No	No	No	Yes	No	No	No	No
No	No	No	No	Yes	No	No	No

SCBR	**veilType**	**veilColor**	**ringNumber**	**ringType**	**sporePrintColor**	**population**	**habitat**

No	No	No	No	No	No	No	No
No	No	No	No	No	No	No	No

```
> table(mushrooms$poisonous)

   e    p
4208 3916
> table(mushrooms$capShape)

   b    c    f    k    s    x
 452    4 3152  828   32 3656
> table(mushrooms$capSurface)

   f    g    s    y
2320    4 2556 3244
> table(mushrooms$capColor)

   b    c    e    g    n    p    r    u    w    y
 168   44 1500 1840 2284  144   16   16 1040 1072
> table(mushrooms$bruises)

   f    t
4748 3376
> table(mushrooms$odor)

   a    c    f    l    m    n    p    s    y
 400  192 2160  400   36 3528  256  576  576
> table(mushrooms$gillAttachment)

   a    f
 210 7914
> table(mushrooms$gillSpacing)

   c    w
6812 1312
> table(mushrooms$gillSize)

   b    n
5612 2512
> table(mushrooms$gillColor)

   b    e    g    h    k    n    o    p    r    u    w    y
1728   96  752  732  408 1048   64 1492   24  492 1202   86
> table(mushrooms$stalkShape)

   e    t
3516 4608
> table(mushrooms$stalkRoot)

   ?    b    c    e    r
2480 3776  556 1120  192
> table(mushrooms$SSAR)

   f    k    s    y
 552 2372 5176   24
```

```
> table(mushrooms$SSBR)

   f    k    s    y
 600 2304 4936  284
> table(mushrooms$SCAR)

   b    c    e    g    n    o    p    w    y
 432   36   96  576  448  192 1872 4464    8
> table(mushrooms$SCBR)

   b    c    e    g    n    o    p    w    y
 432   36   96  576  512  192 1872 4384   24
> table(mushrooms$veilType)

   p
8124
> table(mushrooms$veilColor)

   n    o    w    y
  96   96 7924    8
> table(mushrooms$ringNumber)

   n    o    t
  36 7488  600
> table(mushrooms$ringType)

   e    f    l    n    p
2776   48 1296   36 3968
> table(mushrooms$sporePrintColor)

   b    h    k    n    o    r    u    w    y
  48 1632 1872 1968   48   72   48 2388   48
> table(mushrooms$population)

   a    c    n    s    v    y
 384  340  400 1248 4040 1712
> table(mushrooms$habitat)

   d    g    l    m    p    u    w
3148 2148  832  292 1144  368  192
> |
```



R scripts:

https://github.com/duzixsiansheng/CS-GY-6923HW01/tree/main