多變量分析

期中考Mid-term Exam

指導教授：洪英超
統研碩一 董承 106354014
統研碩一 黃意珺 106354019

目錄

1. Download “European_Jobs.txt” and “European_Jobs_Description.txt” for the data set and its detailed variable description. Q1: Perform a complete Principal Components Analysis for this data and interpret the result. Note: The number of PCs must be determined by a formal statistical hypothesis test, while the relationships among objects and variables can be interpreted by using a 2D plot.

[問題解釋]

變數之間的相關性可能導致共線性，或者太多的解釋變數會使模型過於複雜，而PCA可以解決以上兩個問題。

[資料介紹]

1979年(資料蒐集於冷戰時期)，探討26個歐洲國家各產業狀況，主要分為六個產業：農業(Agr)、礦業(Min)、製造業(Man)、能源供應產業(PS)、營建業(Con)、服務業(SI)、金融業(Fin)、服務及銷售業(SPS)、交通業(TC)，如表1.1。

變數	解釋	資料型態
Country(共26個)	country in Europe	類別型
Agr	% of workforce employed in agriculture	連續型
Min	% in mining	連續型
Man	% in manufacturing	連續型
PS	% in power supply industries	連續型
Con	% in construction	連續型
SI	% in service industries	連續型
Fin	% in finance	連續型
SPS	% in social and personal services	連續型
TC	% in transport and communications	連續型

表1-1. 資料詳細說明

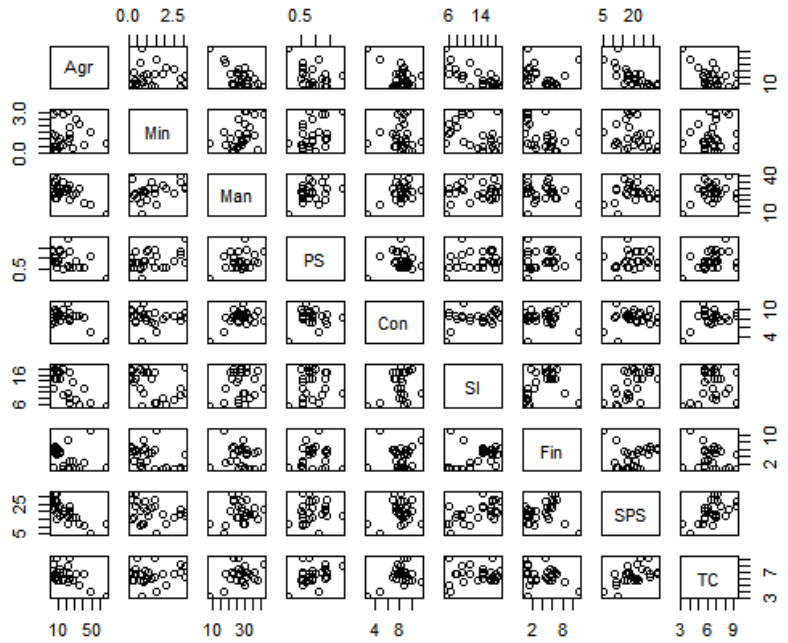


圖1-2. Scatter plot of European_Jobs variables

在開始PCA之前先對資料做分析了解有沒有特別的趨勢，從變數之間的散佈圖(圖1-2.)來看，變數之間是沒有高度的相關性，利用R的指令findLinearCombos尋找變數之間是否有共線性，結果顯示沒有一個變數可以被其他變數的線性組合取代。

[方法介紹]

主成分分析是一種降低為度的技術，透過線性組合組成多個主成分，而同時保留對變異數的解釋程度。主要概念是透過共變異數矩陣進行特徵分解，以得出數據的主成分與特徵值。

[結果解釋]

為了避免變數之間不同的scale導致某個變數變得相對重要，因而將資料標準化後再做PCA。

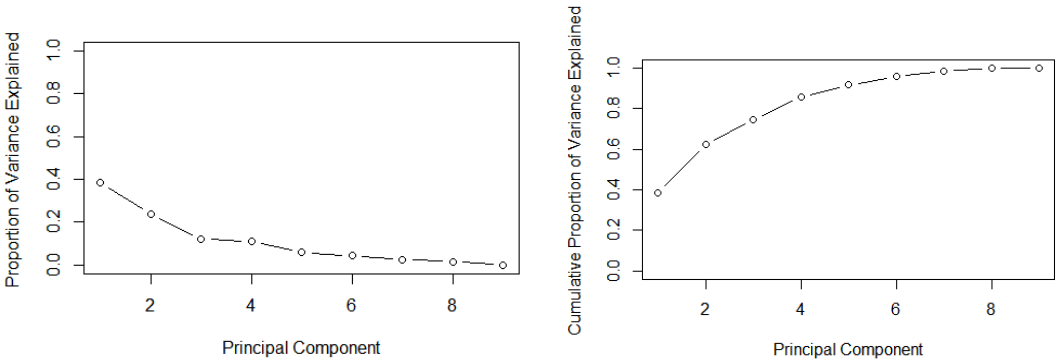


圖1-3. Variation explained by components

由圖1-3.可以看出每個Component對總變異量的解釋能力由大而小排列，而要取多少個component能有足夠的變異解釋力，則是由排列檢定(Permutation test)以及
eigenvalues-greater-than-one rule proposed by Kaiser (1960)來決定。
排列檢定(Permutation Test): λ代表每個Component解釋資料變異的比例，藉由重複抽取所有的λ來建立一個沒有關聯結構的排列分配(Permutation distribution)，再從第一個Component的λ開始檢定他的值在沒有關聯結構的分配(distribution)下是否為顯著的大。
eigenvalues-greater-than-one rule: 這個準則的概念是，每個λ值代表對應之成份變異數(Component Variance)，而原資料的個別變異數為1，因此保留那些Component的λ大於1。
**排列檢定(Permutation Test)

	Arg	Min	Man	PS	Con	SI	Fin	SPS	TC
P-value	0.000	0.000	0.995	0.920	1.000	1.000	1.000	1.000	1.000

圖1-5. 排列檢定結果

檢定所顯示的p-value代表僅有Component1及Component2的λ值是顯著的(即p-value<0.05)，因此由排列檢定的結果僅選擇前兩個成份(Component)。

**eigenvalues-greater-than-one rule

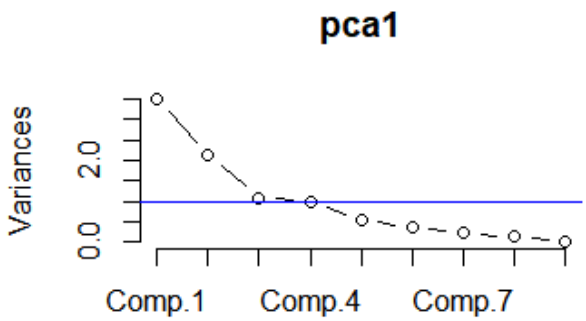


圖1-4. component λ value，藍線為Variance=1

由Kaiser eigenvalue-greater-than-one rule，我們選擇Component1~3，Component4的特徵值(eigenvalue)為0.99723<1，但是因為解釋變異的量太少因此不選。

	Component1	Component2	Component3	Component4
Standard deviation	1.8673915691	1.4595112679	1.0483117911	0.9972376737
Proportion of variance	0.3874612525	0.2366859046	0.1221064013	0.1104981086
Cumulative proportion	0.3874612525	0.6241471571	0.7462535583	0.8567516670

表1-5. summary of components

Component1~3的可解釋總變異量達到0.746254，Component1可解釋變異量為0.38746，Component2可解釋變異量為0.236686，Component3可解釋變異量為0.12211。從累積比例(Cumulative proportion)可以看到當Component4的時候，解釋變數達到約85%。

	Comp.1	Comp.2	Comp.3	Comp.4
Agr	-0.523791	-0.053594	0.048674	-0.028793
Min	-0.001323	-0.617807	-0.201100	-0.064085
Man	0.347495	-0.355054	-0.150463	0.346088
PS	0.255716	-0.261096	-0.561083	-0.393309
Con	0.325179	-0.051288	0.153321	0.668324
SI	0.378920	0.350172	-0.115096	0.050157
Fin	0.074374	0.453698	-0.587361	0.051567
SPS	0.387409	0.221521	0.311904	-0.412230
TC	0.366823	-0.202592	0.375106	-0.314372

表1-6. matrix of PCA loadings

從表1-6.可以知道，Component.1的組成為Agr*(-0.523791)+Min*(-0.00132)+...+TC*(0.366823)，其他的component也是以此類推。

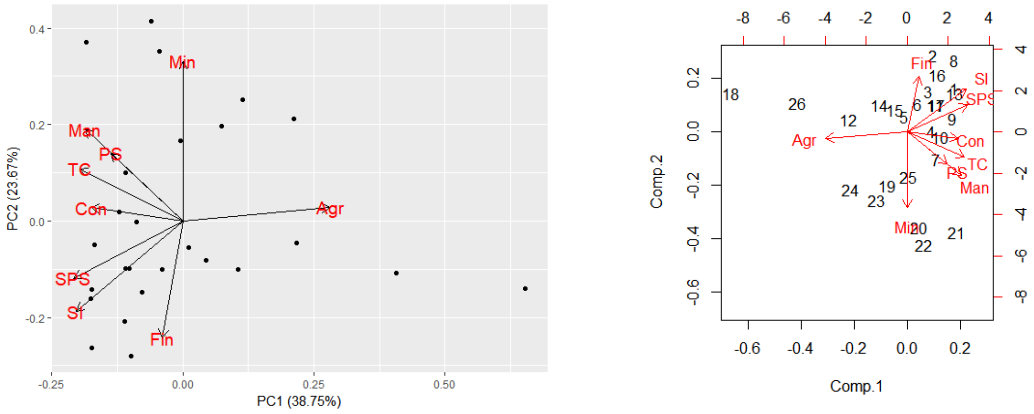


圖1-7. PCA biplot

由PCA的特性，我們先從可解釋變異量最大也是最重要的Comp.1的方向(水平)看起，其中Comp.1的組成為各個變數的線性組合，即Agr*(-0.523791)+Min*(-0.00132)+...+TC*(0.366823)，而其中比

例占前三重要的變數為Agr(農業),SI(服務業),SPS(服務及銷售業)，在這裡主觀的判斷為底層工作者指標。而第18(Turkey,土耳其)與第21(E. Germany,東德)筆資料看起來相差最遠。

```
> European_Jobs[c(18,21),]
# A tibble: 2 x 10
  Country      Agr   Min   Man    PS   Con    SI   Fin   SPS    TC
  <chr>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Turkey      66.8 0.700  7.90 0.100  2.80  5.20  1.10  11.9  3.20
2 E. Germany  4.20 2.90   41.2  1.30   7.60 11.2   1.20  22.1  8.40
```

表1-7. example for European_Jobs data

當我們實際對這兩筆資料做比較，可以觀察到18(Turkey,土耳其)這個國家的Agr(農業)值是比較高的，而21(E. Germany,東德)的SPS(服務及銷售業)值是比較高的。配合主觀對於Comp.1的底層工作者指標的想法，我們可以解釋這兩個國家的產業結構落差是相當大的，而實際情形也確實如此。

2. Suppose a human resource researcher would like to study the relationships between the employment proportions of various types of industry in Europe. He divides the variables in the data “European_Jobs.txt” into two groups: (Agr, Min, Man, PS, Con) and (SI, Fin, SPS, TC). The first group represents the industries that are more labor-intensive, while the second group represents the industries that are less labor-intensive. Q2: Perform a complete Canonical Correlation Analysis for these two groups of variables and interpret the result. Note: The number of canonical variates must be determined by a formal statistical hypothesis test, while the required model assumptions need to be validated.

[問題解釋]

人力資源研究員想了解歐洲產業類別的就業比例狀況。他將變數分割為兩群(groups)，第一類屬於勞力密集型產業，第二類屬於非勞力密集型產業。

[資料介紹]

如Q1

[方法介紹]

CCA : 兩個隨機變量向量 $X = (X_1, ..., X_n)$ 和 $Y = (Y_1, ..., Y_m)$ 並且它們是相關的，那麼典型相關分析會找出 X_i 和 Y_j 的相互相關最大的線性組合。

[結果解釋]

原始資料做出來的Correlation為0.9999381

產業類別	係數(α_i)
Agr	-0.265758454
Min	-0.016036718
Man	-0.120584580
PS	-0.006366807
Con	-0.027202724
SI	0.07855763
Fin	0.04785441
SPS	0.11565031
TC	0.02376452

表2-1. 原始資料係數

這次我們做了三種多變量常態檢定(Mardia's multivariate skewness and kurtosis coefficients, Henze-Zirkler's, Royston's)僅有Henze-Zirkler's多變量常態檢測通過，其他兩個都沒有通過。

檢定	統計量	p-value	是否通過檢定
Mardia Skewness	203.740914857322	0.0216172227136561	NO
Mardia Kurtosis	0.744455360290763	0.456601037760766	YES
MVN	<NA>	<NA>	NO

表2-2. Mardia's 多變量檢定

檢測	統計量	p-value	是否通過檢定
Henze-Zirkler	0.9273322	0.6995631	YES

表2-3. Henze-Zirkler's 多變量檢測

檢測	H	p-value	是否通過檢定
Royston	29.52238	8.482556E-05	NO

表2-4. Royston's多變量檢測

因此我們選擇檢測各個變數的離群值(outlier)，透過Shapiro-Wilk test檢測在95%信賴區間下是否服從常態分佈：

	檢定統計量	P-value	是否服從Normal
Agr	0.8616	0.0024	NO
Min	0.8873	0.0082	NO
Man	0.9713	0.6572	YES

PS	0.9481	0.2087	YES
Con	0.8842	0.0071	NO
SI	0.9046	0.0199	NO
Fin	0.9111	0.0279	NO
SPS	0.9806	0.8860	YES
TC	0.9634	0.4632	YES

表2-5. 在95%信賴區間之下，以Shapiro-Wilk 檢測各個變數是否服從常態

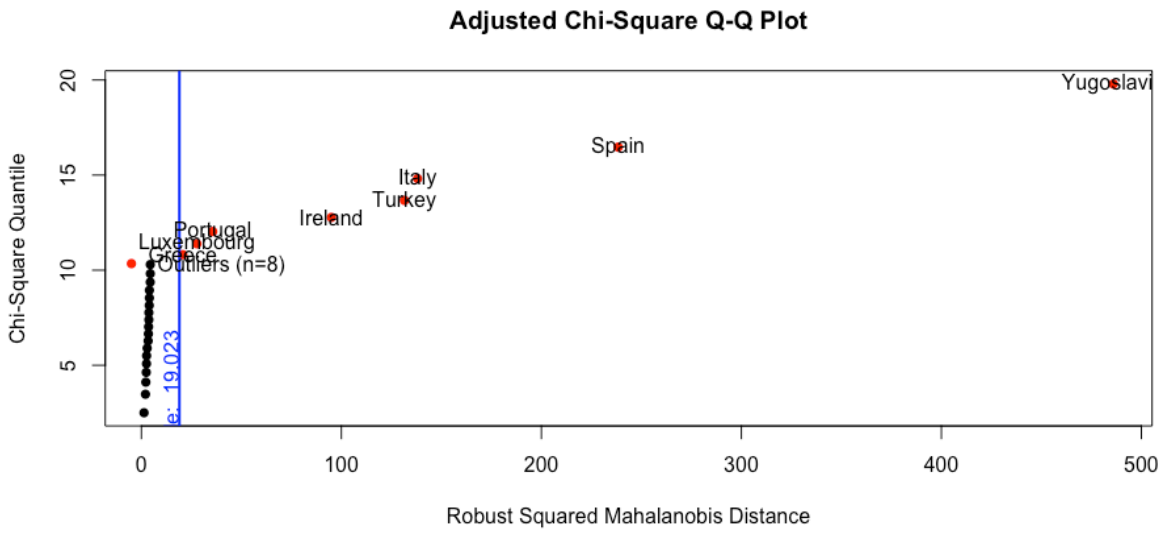


圖2-6.

將上述紅點資料刪除後，將剩下的18筆資料(即18個國家)，用Royston's多元常態檢測，新資料是否通過：

檢測	H	p-value	是否通過檢定
Royston	15.00213	0.01172966	NO

表2-7. 修正後資料進行Royston檢測

典型相關分析(Canonical Correlation Analysis)檢測

$$c_i = \sqrt{\lambda_i}$$

$$H_0 : c_r = c_{r+1} = \ldots = c_k = 0$$

$$H_a : \text{not } H_0$$

	統計量	近似值	df1	df2	p.value	拒絕H_0與否
r=1	6.178464E-06	103.787944	20	57.33250	0.000000000	拒絕
r=2	1.934289E-01	3.436589	12	47.91503	0.001106016	拒絕
r=3	5.353019E-01	2.322980	6	38.00000	0.052233815	不拒絕
r=4	8.867045E-01	1.277715	2	20.00000	0.300461585	不拒絕

表2-8. Wilk's λ, F-approximation test(Rao's F)

-	新資料		原始資料	
產業類別	係數	Correlation	係數	Correlation
Agr	-0.239754764		-0.265758454	
Min	-0.021180714		-0.016036718	
Man	-0.109725466		-0.120584580	
PS	-0.006219238		-0.006366807	
Con	-0.013170748	0.9999840	-0.027202724	0.9999381
SI	0.09391031		0.07855763	
Fin	0.04564809		0.04785441	
SPS	0.13356656		0.11565031	
TC	0.02517704		0.02376452	

將新資料(共18筆)做出來Correlation為0.9999840

表2-9. 新舊資料比較

由表2-9.可以看到農業(Agr)、製造業(Man)、服務及銷售業(SPS)所佔的係數絕對值最大，影響最終的結果。資料修正前及修正後的相關係數，表格中可以看到修正後的相關係數較接近1，也就是說資料比修正前更相關，亦符合我們的期待。

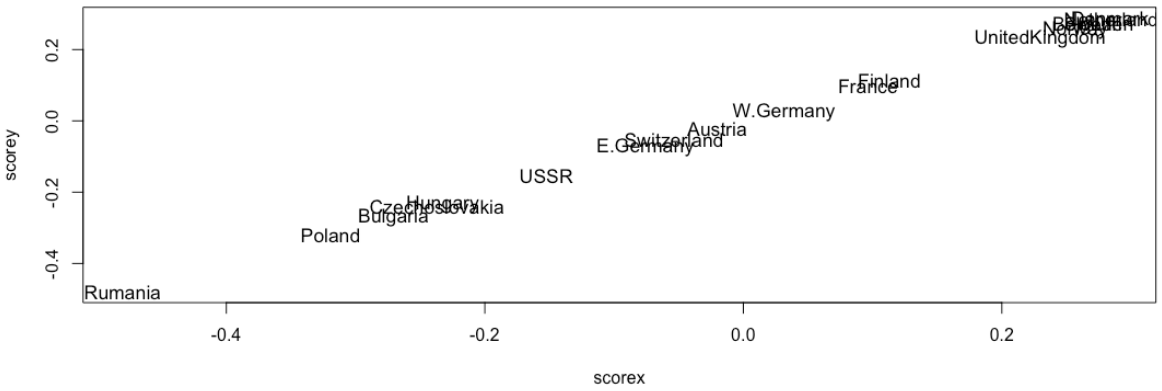


圖2-10.

$$scorex = \sum_i \alpha_i X_i$$

$$scorey = \sum_i \beta_i Y_i$$

由圖2-10. 可以看到羅馬尼亞(Rumania)、波蘭(Poland)、保加利亞(Bulgaria)、捷克斯洛維亞(Czechslovakia)、匈牙利(Hungary)皆位於負數的座標，也就是說，上述這些國家主要以勞力密集產業為主。

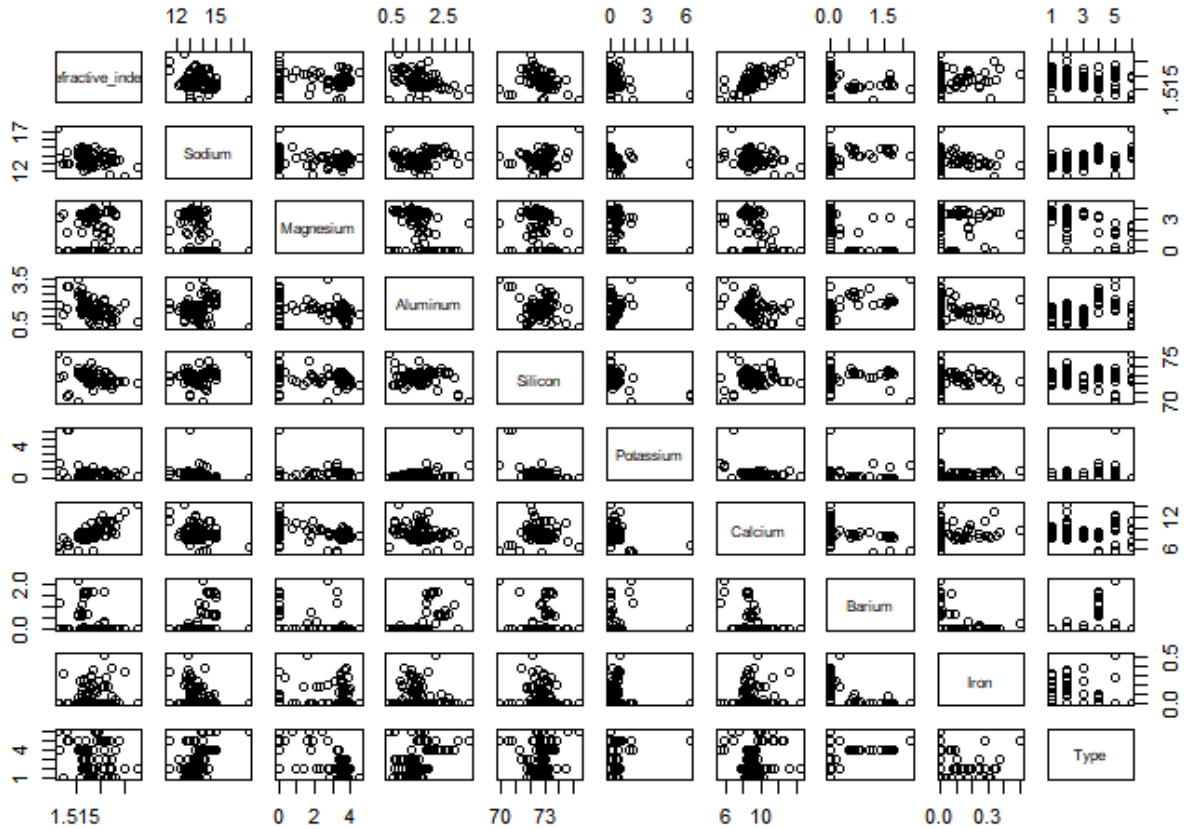
3.The data set “glass.dat” was collected by the department of criminological investigation, it consists of 114 observations with 6 types, where each observation is described by 9 attributes (please refer to “glass_description.txt” for variable description). At the scene of the crime, the glass left can be used as evidence, if its type is correctly identified. Q3: Construct the decision rules for classifying the types of glass using (i) Classification Tree; (ii) LDA; (iii) QDA; (v) Nearest Neighbor; and (vi) Logistic discrimination. Compare all your resulting decision rules and explain which one you will best recommend. Q4: Based on the control of “Classification Tree” obtained in Q3, develop two decision rules by using the strategies of (i) random forest and (ii) boosting. How do these two strategies compare with the decision rules in Q3 in terms of prediction accuracy?

用套件caret可以選擇用repeated 3 times 10-fold cross-validation，在理論上比起用trainging dataset與testing dataset計算準確度更好，因為每個觀測值都有機會被當作訓練集，也不會因為抽樣的誤差產生偏誤；另一種理論上最佳的做法是LOOCV(Leave-One-Out Cross-Validation)，這裡將這兩個當作比較演算法之間的準則，不會特別去探討訓練模型對訓練資料的apparent error rate，因為能夠藉由調整極端的參數去達到更高的準確度，但是可能會有過度配適的問題產生。

[資料介紹]

Vina conducted a comparison test of her rule-based system, BEAGLE, the nearest-neighbor algorithm, and discriminant analysis. BEAGLE is a product available through VRS Consulting, Inc.; 4676 Admiralty Way, Suite 206; Marina Del Ray, CA 90292 (213) 827-7890 and FAX: -3189. In determining whether the glass was a type of "float" glass or not, the following results were obtained (# incorrect answers):

圖3-1. Scatter plot



在開始之前先對資料做分析了解有沒有特別的趨勢，從變數之間的散佈圖來看，變數之間是沒有高度的相關性，利用R的指令findLinearCombos尋找變數之間是否有共線性，結果顯示沒有一個變數可以被其他變數的線性組合取代。

進行預測之前，將70%的資料作為訓練資料(training data)，另外30%的資料作為測試資料(testing data)。最後將會用訓練資料所訓練出來的模型做重複交叉驗證(10-fold Cross-Validation repeated 3 times)和LOOCV(Leave-One-Out Cross-Validation)，以獲取對真實分類正確率的估計，並且用訓練模型預測測試資料，以表格比較各方法與真實資料的情況。在這裡我們參考apparent error rate，僅比較其true error rate。

```
> glass$Type %>% table
.
 1  2  3  4  5  6
23 42 11 21 11  6
> training$Type %>% table
.
 1  2  3  4  5  6
17 30  8 15  8  5
```

圖3-2. counts of glass Type

檢測這筆資料欲分類的變數底下，而各個類別的次數落差很大，這類型的資料又稱不平衡資料(Unbalanced data)，可能對分類的準確度有影響，在這裡我們以過抽樣(Oversampling)的方式解決這個問題，即透過不斷抽取放回在訓練資料中類別個數較少的樣本，直到跟最多的類別個數一樣多，這樣的做法風險是類別較少的樣本如果是outlier的話，可能會使模型產生偏誤。

```
> table(up_train$Class)

 1  2  3  4  5  6
30 30 30 30 30 30
```

圖3-3. 過抽樣訓練資料

[方法介紹]

(i) Classification Tree

決策樹是用來處理分類問題的樹狀結構，使用方法為:選出分類能力最好的屬性做為樹的內部節點，將內部節點的所有不同資料產生出對應的分支，遞迴重複上面的過程直到滿足終止條件，ID3、C4.5 、C5.0、CHAID及CART是決策樹演算法的代表。

(ii) LDA、QDA

線性判別分析(Linear Discriminant Analysis)的想法與PCA相似，透過變數的線性組合組成LDs，而使這些LD對不同類別的分類有最好的分類效果。而LDA必須服從兩個假設，分別是各個變數需服從多元常態分配(multivariate Gaussian)，而且每個類別必須有共同的共變異矩陣。QDA則放寬了對後者假設的限制，使其在分類時可以用曲線或曲面的方式做分類。

(iii) Nearest Neighbor

最近鄰居法(K-Nearest Neighbor)，概念是選擇離本身距離最近的K個點，依照所選的點中最多的類別即為該類別的預測。

(iv) Logistic discrimination

將一個類別作為基準，考量這個基準與其他類別後驗機率的比，即構成一個線性模型，若為兩個類別的模型即為羅吉斯迴歸的概念。而在線性模型的概念之下，各個變數必須服從常態假設，而且各變數之間的共變異矩陣必須相同。

(v) random forest

隨機森林是一個包含多棵決策樹的分類器，先以bootstraping抽取樣本，隨後透過隨機選取m(小於總變數個數)個變數建立出決策樹，重複做到指定的次數為止，最後利用這些樹的投票結果來決定分類的類別。

(vi) boosting, bagging

boosting與bagging都是Ensemble的概念，就是透過重複採樣行成多個弱分類器，最後結合為一個強分類器，進而達到提升分類正確率的目標。兩者差異在於boosting對於分類錯誤的樣本會提升權重，提高該樣本前一次錯誤而下一次分類正確的機率。

[結果解釋]

(i) Classification Tree

利用R中rpart的套件做決策樹，其中的參數minbucket(最終節點可以接受樣本各數)選擇5，因為訓練資料中類別6的各數最少只有5個；而minsplit選擇則是依照常用的方式，即前一個參數的個數乘以三倍為15。

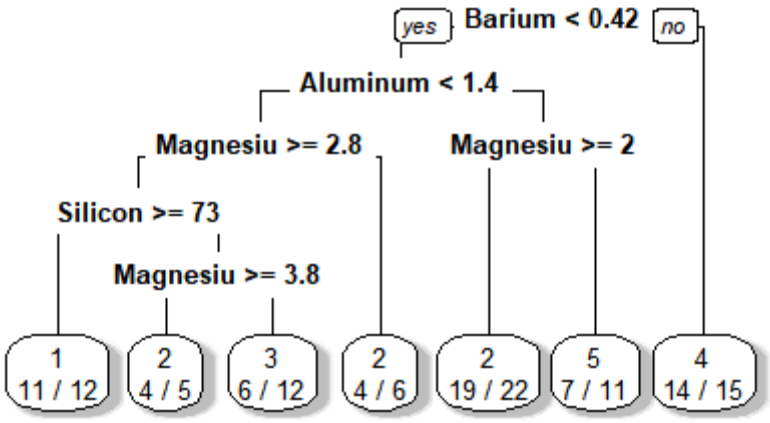


圖3-4. 決策樹

決策樹使用了四個變數做分類，最左邊的節點代表，訓練資料按照上方的分類規則來到這個節點，分類的結果為”1”共有12次，而其中有11次被正確份類為"1"。這是一顆未被修剪過的樹，因此分類方式比較複雜，我們透過選擇適當的參數讓這個樹被修剪。

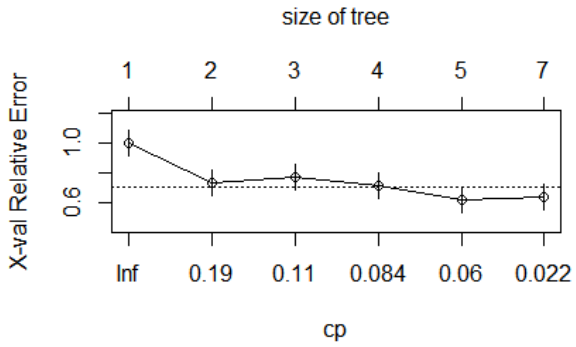
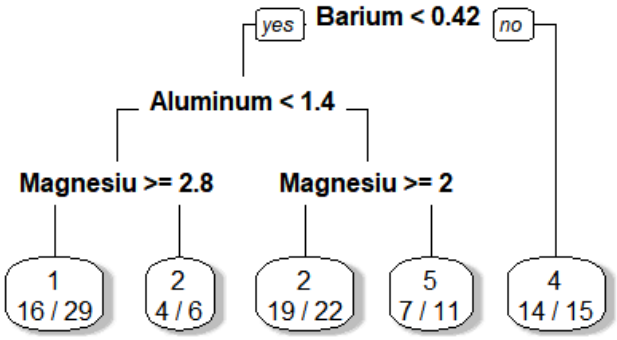


圖3-5. relationship between CP and true error rate

透過relative error與cp和size of tree的圖可以發現，當cp越小，我們得到的錯誤率也會越低，因此透過尋找最小的"xerror"(true error rate)來幫助我們尋找最佳的參數，並用此參數作為修剪這棵樹的原則。

圖3-6. 修剪後的樹

修剪過後的樹一樣使用了四個變數作為分類規則，而分類的方式更為簡單。



```
> printcp(glass.treecv)  # true accuracy rate for nsp
lit = 6: 1-53*0.64151/83 = 0.5903611

Classification tree:
rpart(formula = Type ~ ., data = training, method = "
class",
    control = glass.control)

Variables actually used in tree construction:
[1] Aluminum  Barium    Magnesium Silicon

Root node error: 53/83 = 0.63855

n= 83

        CP nsplit rel error  xerror     xstd
1 0.264151      0   1.00000 1.00000 0.082582
2 0.132075      1   0.73585 0.73585 0.085791
3 0.094340      2   0.60377 0.67925 0.085189
4 0.075472      3   0.50943 0.67925 0.085189
5 0.047170      4   0.43396 0.52830 0.081273
6 0.010000      6   0.33962 0.64151 0.084532
```

在使用LOOCV(leave-one-out cross-validation)下，所得出的true accuracy rate為

$$1 - 53 * \frac{0.64151}{83} = 0.5903611$$

Oversampling

考量到資料本身是不平衡資料(Unbalanced data)，可能使模型的準確度不高或是類別個數較少的資料不容易被預測到，這裡用過抽樣(Oversampling)搭配10-fold Cross-Validation repeated 3 times，去看true accuracy rate，而參數的選擇由caret套件自動選擇使準確度最佳的參數。

```
> print(glass.oversamp)
CART

180 samples
  9 predictor
  6 classes: '1', '2', '3', '4', '5', '6'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 time
s)
Summary of sample sizes: 162, 162, 162, 162, 162, 162
, ...
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.04666667  0.7925926  0.7511111
  0.11666667  0.6833333  0.6200000
  0.19000000  0.4574074  0.3488889

Accuracy was used to select the optimal
 model using the largest value.
The final value used for the model was cp
 = 0.04666667.
```

圖3-7. oversampling decision tree

從圖3-7.的結果可以看出自動選出最佳參數CP為0.466667的true accuracy rate為0.7925926，比起用原本的訓練資料所建立出的模型之true accuracy rate增加了0.2022315。

(ii) LDA

如同方法介紹所述，在使用LDA之前資料必須服從兩個假設，分別是變數服從多元常態，以及組內的共變異矩陣相等。我們先檢定多元常態的假設。

```
> mvn(data = training[,-10],mvnTest = "hz")  # do not pass MVN test
$multivariateNormality
           Test       HZ p value MVN
1 Henze-Zirkler 2.692478       0  NO

$univariateNormality
          Test         Variable Statistic   p value Normality
1 Shapiro-Wilk refractive_index    0.9272   2e-04      NO
2 Shapiro-Wilk      Sodium         0.9449  0.0014      NO
3 Shapiro-Wilk    Magnesium        0.7506  <0.001      NO
4 Shapiro-Wilk     Aluminum        0.9301   2e-04      NO
5 Shapiro-Wilk     Silicon         0.9104  <0.001      NO
6 Shapiro-Wilk    Potassium        0.4301  <0.001      NO
7 Shapiro-Wilk     Calcium         0.8647  <0.001      NO
8 Shapiro-Wilk      Barium         0.5119  <0.001      NO
9 Shapiro-Wilk       Iron          0.5715  <0.001      NO
```

圖3-8. MVN test for training data

LDA和QDA都有共線性的假設，但是變數都非服從常態，整體也不服從多元常態。雖然可以強制執行建立出分類模型，即使違背演算法的假設，但是在這裡還是將這個方法拿進來做比較，看看線性分類與非線性分類模型是否會有更好的分類效果。

```
Proportion of trace:
   LD1    LD2    LD3    LD4    LD5
0.6986 0.1988 0.0669 0.0284 0.0072
```

圖3-9. LDA proportion of trace

我們可以看到由前兩個LDs可以獲取0.8974的組間變異。

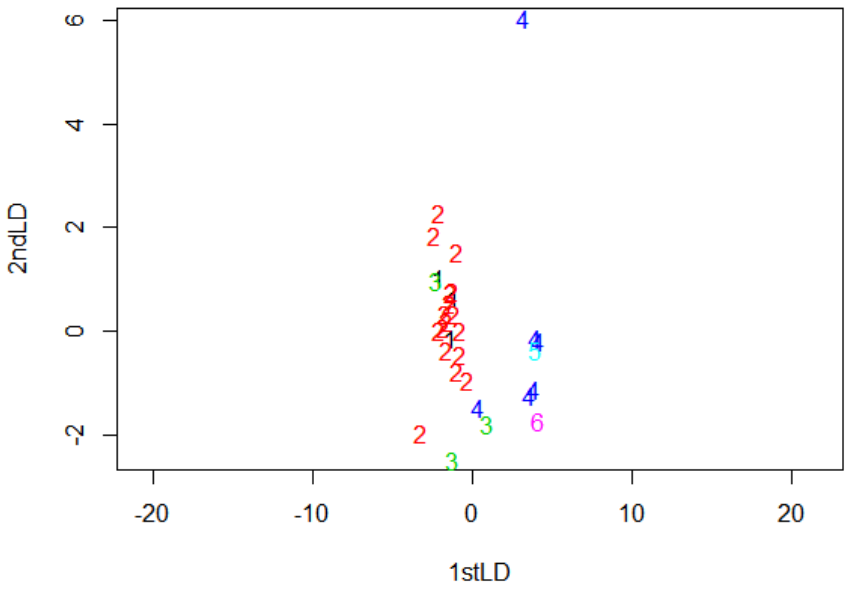


圖3-10. LDA plot

利用訓練資料建立的來預測測試資料，並且在兩個線性判別分析(Linear Discriminant)的維度上將測試資料點放上，從圖中可以看出兩個線性判別分析(Linear Discriminant)對於不同類別的區分效果普通，有幾個類別的部分資料會遠離自己類別大部分的資料。

```
> print(glass.lda2)  # accuracy = 0.5896164021
Linear Discriminant Analysis

83 samples
 9 predictor
 6 classes: '1', '2', '3', '4', '5', '6'

Pre-processing: scaled (9), centered (9)
Resampling: Cross-Validated (10 fold, repeated 3
times)
Summary of sample sizes: 75, 76, 75, 74, 74, 73,
...
Resampling results:

  Accuracy   Kappa
  0.5811905  0.4276297
```

圖3-11. true accuracy rate by LDA

用訓練資料建立LDA模型並且用10-fold cross-validation repeated 3 times的準確度為0.5811905，分類的準確度並不高，因此考慮使用oversampling的資料集建立LDA模型看準確度是否可以提升。

```
> print(oversampl.lda)  # accuracy = 0.7592592593
Linear Discriminant Analysis

180 samples
  9 predictor
  6 classes: '1', '2', '3', '4', '5', '6'

Pre-processing: scaled (9), centered (9)
Resampling: Cross-Validated (10 fold, repeated 3
times)
Summary of sample sizes: 162, 162, 162, 162, 162,
 162, ...
Resampling results:

  Accuracy   Kappa
  0.7592593  0.7111111
```

圖3-12. true accuracy rate by oversampling LDA

一樣使用10-fold cross-validation repeated 3 times的oversampling LDA模型的true accuracy rate為0.7592593，比起用原本的訓練資料所建立出的模型之true accuracy rate增加了0.1780688。

(iii) QDA

如同LDA的假設，QDA也需要變數服從多元常態的假設，但是從先前的結果來看是無法達到這個條件。在R執行QDA時會遇到幾個問題，分別是

(a) 預測變數的某個類別太少而無法執行QDA

(b) 出現rank deficiency的錯誤 ，對應不同的問題嘗試不同的解決方式。

```
> glass.qda <- qda(Type ~. ,data=training.qda)  # some group
 is too small for qda, use ovesampling
Error in qda.default(x, grouping, ...) :
  some group is too small for 'qda'
> oversampl.qda2 <- qda(up_train$Class ~. ,data = up_train)
 # rank deficiency
Error in qda.default(x, grouping, ...) : rank deficiency in
group 1
```

圖3-13. error in QDA

** 嘗試解決方法

1. 變數標準化
2. oversampling
3. 標準化後oversampling
4. 拿掉相關性較高(中度相關)的其中一個變數(Calcium)

(a) 預測變數的其中一個類別個數太少

對於類別太少而無法執行QDA我們一樣用oversampling的方式處理，但是在使用oversampling後的訓練資料執行QDA會回到(b.)的問題

(b) rank deficiency in group

可能發生rank deficiency的原因歸類為以下兩種：

(i)變數之間有高度共線性。

在資料探索與預處理時，已經檢測過變數之間是沒有高度共線性的，而我們也嘗試過拿掉其中相關性最高的其中一個變數(Calcium)，結果也是無法執行QDA。

(ii)樣本數遠小於變數個數(n < p)。

訓練資料的個數並沒有少於變數個數。

因為在R上無法執行QDA，我們改用Python來執行，而執行時想當然的會出現warning，在統計的角度上是不應該使用這個方法，但是作為方法的比較我們還是把它拿來做為參考。

```
qda = QDA()
std_scalar = StandardScaler()
steps = [('std_scalar',std_scalar),('QDA',qda)]
qda_model = Pipeline(steps)

qda_model = qda_model.fit(X_train,y_train)
qda_model.score(X_train,y_train)
```

```
C:\Users\js\Anaconda3\lib\site-packages\sklearn\discriminant_analysis.py:68
2: UserWarning: Variables are collinear
  warnings.warn("Variables are collinear")
0.8734177215189873
```

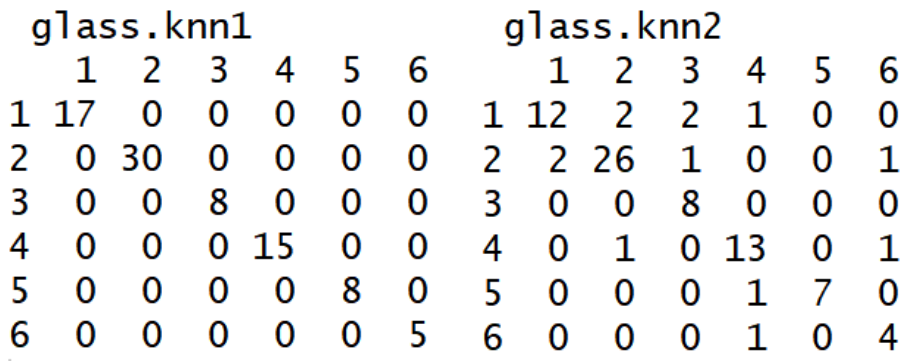
```
pred = qda_model.predict(X_test)
accuracy_score(y_test,pred)
```

```
0.5142857142857142
```

圖3-14. QDA in Python

在Python之下執行QDA的利用testing data去估計true accuracy rate為0.51428571。

(iv) Nearest Neighbor



glass.knn1

	1	2	3	4	5	6
1	17	0	0	0	0	0
2	0	30	0	0	0	0
3	0	0	8	0	0	0
4	0	0	0	15	0	0
5	0	0	0	0	8	0
6	0	0	0	0	0	5

glass.knn2

	1	2	3	4	5	6
1	12	2	2	1	0	0
2	2	26	1	0	0	1
3	0	0	8	0	0	0
4	0	1	0	13	0	1
5	0	0	0	1	7	0
6	0	0	0	0	1	4

圖3-15. confusion matrix of NN-1 & NN-2

選擇所有變數之下，利用KNN選擇當K = 5開始時建立模型，發現當K越小\apparent accuracy rate會越高，當K = 2時apparent accuracy rate為0.843373；K = 1，apparent accuracy rate為1.00。因此

我們利用NN-1的模型做LOOCV來預測true accuracy rate，而為了與前面的方法做比較，這裡選用與decision tree所選的四個變數，Barium, Calcium, Magnesium, Sodium。

```
> (confusion.matrix <- table(training$Type,
glass.kncv))
   glass.kncv
     1  2  3  4  5  6
  1 17  0  0  0  0  0
  2  0 30  0  0  0  0
  3  0  1  7  0  0  0
  4  0  0  1 13  1  0
  5  0  0  0  1  7  0
  6  0  0  0  0  0  5
```

圖3-16. NN-1 LOOCV

而估計出的true accuracy rate高達0.9518072，說明這個方法非常適合這筆資料，而在之後也會用測試資料來看是否真能分類的這麼好。

(v) Logistic discrimination

在做Logistic discrimination之間必須滿足兩個假設，分別是變數必須服從多元常態假設，以及變數之間有著相同covariance matrix。而第一個條件從資料探索與預處理已經得知不符合。因此在統計意義上對這筆資料做Logistic discrimination是有疑慮的，不過這裡只將其結果拿來做比較。

```
> glass.logd  # treated "1" as reference
Call:
multinom(formula = Type ~ ., data = training, maxit = 250)

Coefficients:
  (Intercept) refractive_index      Sodium   Magnesium  Aluminum
2   35.865940       123.781205  0.55240361  -5.0495688  2.927621
3   99.204639        57.564216  0.01180663  -3.1848741 -2.116048
4   -2.180136        55.160482 13.21159394  -0.5828086 -8.026108
5    5.883057         7.560285  5.11393918   0.4142671 14.312455
6   -3.900511        -7.624786 42.25312443 -11.3698153 13.356497
    Silicon   Potassium    Calcium     Barium        Iron
2 -2.569967   -2.274665  -3.365239   17.44105    3.163228
3 -1.981444   -4.244331  -3.172150   48.03544   -8.976774
4 -4.293897   48.097198   4.112281  289.99570 -124.807650
5 -3.256499   70.309539   9.445730  271.97091 -138.514321
6 -4.992569 -176.936562 -18.946559 -135.37334 -109.421018

Residual Deviance: 72.76153
AIC: 172.7615
```

圖3-17. Logistic discrimination output

從圖3-17.可以看出這個方法用類別”1”作為基準，分別對其他五個類別配適了模型，我們直接看LOOCV估出來的true accuracy rate為0.7831。

(vi) 隨機森林(random forest)

在random forest中有兩個參數必須設定

mtry：是隨機森林在變數投入形成決策樹，在內部節點以隨機方式進行分裂的數量，利用隨機分裂方式降低變數相依性。

ntree：為決策樹生長的多寡。ntree 的大小與袋外錯誤率有負相關的關係；當 ntree 越 大時，袋外錯誤率逐漸降低，最後錯誤率呈現平穩狀態。

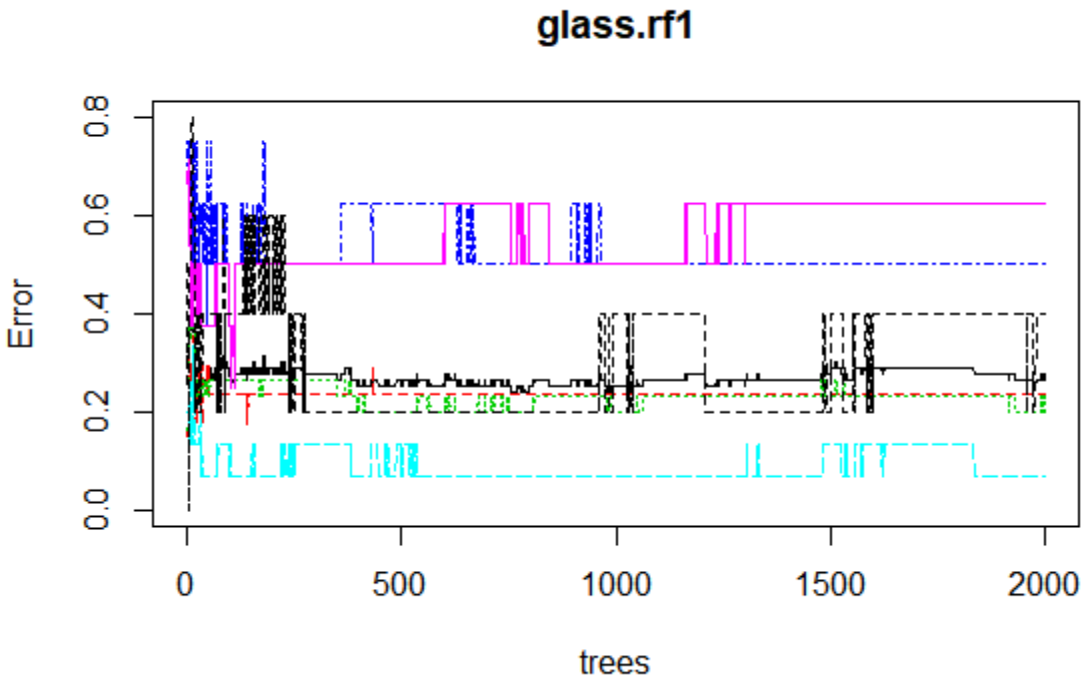


圖3-18. relation between number of trees(ntree)

由於Q3中classification tree僅使用了4個變數，因此設定random forest的參數mtry為4，從圖3-18.可以看到當ntree在數字大時準確度會趨於穩定，因此取夠大的值2000來建立隨機森林(random forest)。用70%資料作為訓練資料，估計出OOB(out of bag) error rate為0.3614，也就是說準確度為0.6386。

由於參數的設定對準確度會有影響，在這裡我們嘗試了兩種做法尋找最佳的參數設定，分別是棋盤搜尋(Grid Search)與隨機搜尋(Random Search)，前者的概念是將每個合理的參數一個一個帶入嘗試，如同棋盤方格，將上面的每一個點都做測試；而後者則是隨機將值拿來做測試。

```
83 samples
 9 predictor
 6 classes: '1', '2', '3', '4', '5', '6'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated
 3 times)
Summary of sample sizes: 73, 74, 74, 73, 75, 7
6, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  1     0.6652249  0.5414265
  2     0.7013228  0.5996081
  3     0.7091005  0.6102184
  4     0.6870767  0.5786276
  5     0.6642989  0.5478841
  7     0.6633333  0.5438133
  8     0.6616534  0.5457734
  9     0.6769312  0.5623982

Accuracy was used to select the
 optimal model using the largest value.
The final value used for the model was
 mtry = 3.
```

圖3-19. random forest(grid search)

Grid Search與Random Search所找出的最佳參數相同，準確率也大致相同。從圖3-19. 中可以看到最佳的參數為mtry = 3，而透過10-fold Cross-Validation repeated 3 times可以得知估計的true accuracy 為0.7091005，比mtry = 4的準確度上升了約0.0705。

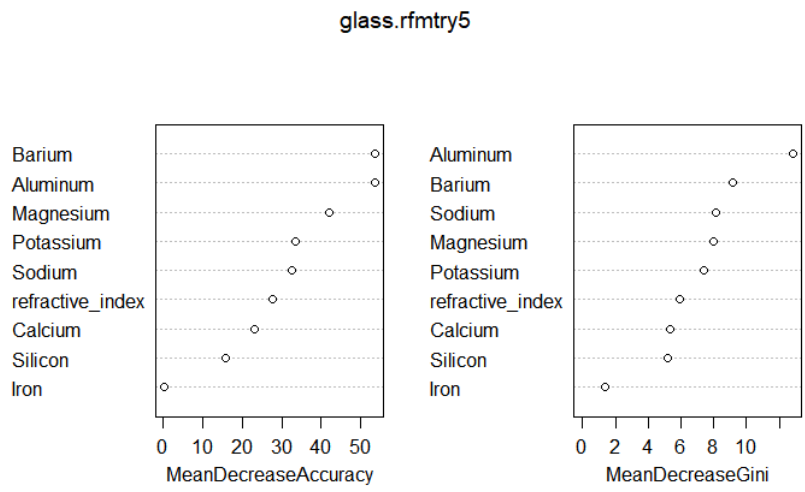


圖3-20. Variables Importance Plot

隨機森林除了是一個很好的分類器以外，他同時也是一個特徵選取(feature selection)常用的方法之一，而這種特徵選取的方式稱為Embedded，也就是考量了變數之間與分類器之間相互的影響。在

這裡我們看左圖的下標MeanDecreaseAccuracy，左邊圖示的大小是依據這個變數捨棄後會損失多少accuracy，作為變數重要性的指標。

(vii) boosting

```
> glass.adaboostcv[-1]
$confusion
                Observed Class
Predicted Class  1  2  3  4  5  6
              1 12  5  3  1  0  0
              2  4 19  2  0  4  3
              3  1  1  3  0  0  0
              4  0  0  0 14  1  0
              5  0  3  0  0  3  0
              6  0  2  0  0  0  2

$error
[1] 0.3614458
```

圖3-21. confusion matrix of boosting CV

這裡我們直接使用boosting的方法並以LOOCV估計true accuracy rate，iteration=20並使用bootstrap重抽讓他計算準確度，最後得到的true error rate為0.3614458，即true accuracy rate為0.6385542。

(viii) bagging

```
> glassbag.cv[-1]
$confusion
                Observed Class
Predicted Class  1  2  3  4  5  6
              1 10  6  5  1  0  0
              2  5 21  3  0  4  4
              3  2  0  0  0  0  0
              4  0  0  0 14  1  0
              5  0  1  0  0  3  1
              6  0  2  0  0  0  0

$error
[1] 0.4216867
```

圖3-22. confusion matrix of bagging CV

這裡我們直接執行bagging的方法並以LOOCV估計true accuracy rate，參數mfinal選擇20次並使用bootstrap重抽讓他計算準確度，最後得到的true error rate為0.4216867，即true accuracy rate為0.5783133。

oversampling rf	rf	logistic	NN-1	oversampling LDA	LDA	oversampling tree	LOOCV tree	10-fold CV tree	Method	TRUE
3	3	3	1	3	2	1	1	1	5	1
2	2	2	1	2	2	2	2	2	6	1
1	1	2	1	1	1	1	1	1	13	1
2	2	2	1	1	2	2	2	2	14	1
1	1	1	1	1	1	1	1	1	19	1
1	1	6	1	3	3	1	1	1	20	1
2	2	3	2	3	3	3	1	1	25	2
2	2	2	2	2	2	2	2	2	26	2
2	2	4	2	3	2	3	1	1	27	2
2	2	2	2	2	2	2	2	2	29	2
3	2	2	2	2	1	2	2	2	42	2
2	2	2	2	2	1	1	1	1	47	2
1	1	2	2	2	2	1	1	1	50	2
5	2	4	2	2	2	5	2	1	54	2
2	2	2	2	2	2	2	2	2	55	2
2	2	2	2	2	2	2	2	2	57	2
1	1	2	2	1	1	3	1	1	58	2
1	1	2	1	2	1	3	1	1	64	2
1	1	2	3	2	2	3	1	1	68	3
1	1	6	3	3	3	1	1	1	72	3
1	1	2	3	3	2	1	1	1	76	3
2	5	5	2	5	5	2	2	5	85	5
5	5	5	5	5	5	5	5	2	86	5
5	2	2	5	5	2	5	5	2	87	5
6	6	6	6	6	6	6	2	1	90	6
4	4	5	4	4	4	4	4	4	94	4
4	4	5	4	4	4	4	4	4	102	4
4	4	4	4	4	4	4	4	4	103	4
4	4	5	4	6	4	4	4	4	108	4

4	4	5	4	6	4	4	4	4	110	4
4	4	4	4	4	4	4	4	4	113	4

表3-23. Comparison table

method	accuracy by CV
10-fold CV tree	0.5793519
LOOCV tree	0.5903611
oversampling tree	0.7925926
LDA	0.59%
oversamplin LDA	0.7592593
QDA(by Python)	0.514285
NN-1	0.975903614
logistic	0.7831325
bagging	0.5783133
boosting	0.6385542
randomforest(mtry= 4)	0.6386
randomforest(grid search)	0.736005291
oversampling rf	0.9333333

表3-24. true accuracy rate

發現logistic的估計true accuracy rate雖然高，但是在testing裡的表現很糟糕。

4.The data “new_wages.txt” contains information regarding types of wages and some categorical characteristics from a random sample of 534 persons. The description of all variables is given in “new_description.txt”. Q5) Construct a suitable decision rule to classify the types of wages based on the methods introduced in our class. Also, comment on the performance of your decision rule in terms of prediction accuracy.

[資料介紹]

New Wages Data包含了各種不同薪資的種類，以及其他類別變數，資料由隨機的方式抽取了534位民眾。

The New Wages Data Set contains information regarding types of wages and other categorical characteristics from a random sample of 534 persons. The variables are described next:

1) Wage type: A=(wage<=$5/hr), B=($5/hr<wage<=$10/hr), C=(wage>$10/hr)
2) Education level: 1=very low, 2=low, 3=medium, 4=high
3) South: 1=person lives in the south, 0=otherwise
4) Gender: 1=female, 0=male
5) Experience: 1=low, 2=medium, 3=high
6) Union: 1=union member, 0=otherwise
7) Age: 1=less than 33 years old, 2=between 33 and 48 years old, 3=older than 48

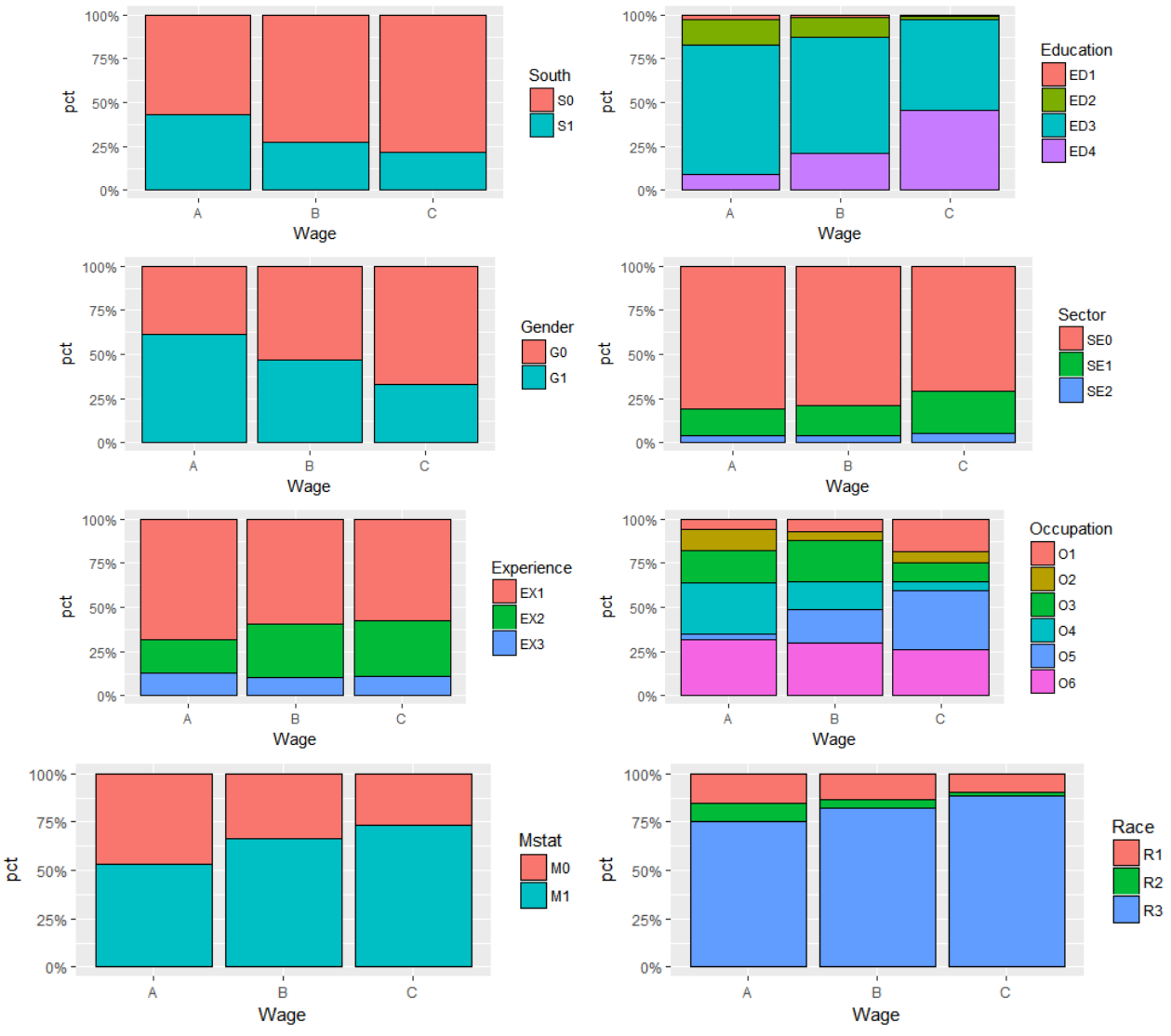
8) Race: 1=other, 2=hispanic, 3=caucasian
9) Occupation: 1=management, 2=sales, 3=clerical, 4=service, 5=professional, 6=other
10) Sector: 0=other, 1=manufacturing, 2=construction
11) Mstat: 1=married, 0=otherwise

[資料探索與預處理]

```
> new_wages$Wage %>% table()
.
  A   B   C
125 243 166
> training.1$Wage %>% table()
.
  A   B   C
 88 171 117
```

圖4-1.counts of target variable in data & training data

這裡我們一樣用70%的資料作為訓練資料，剩下的30%作為測試資料。



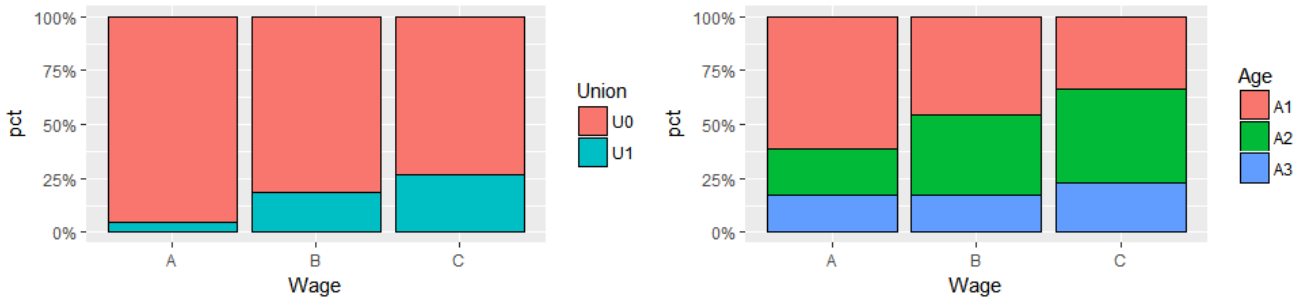


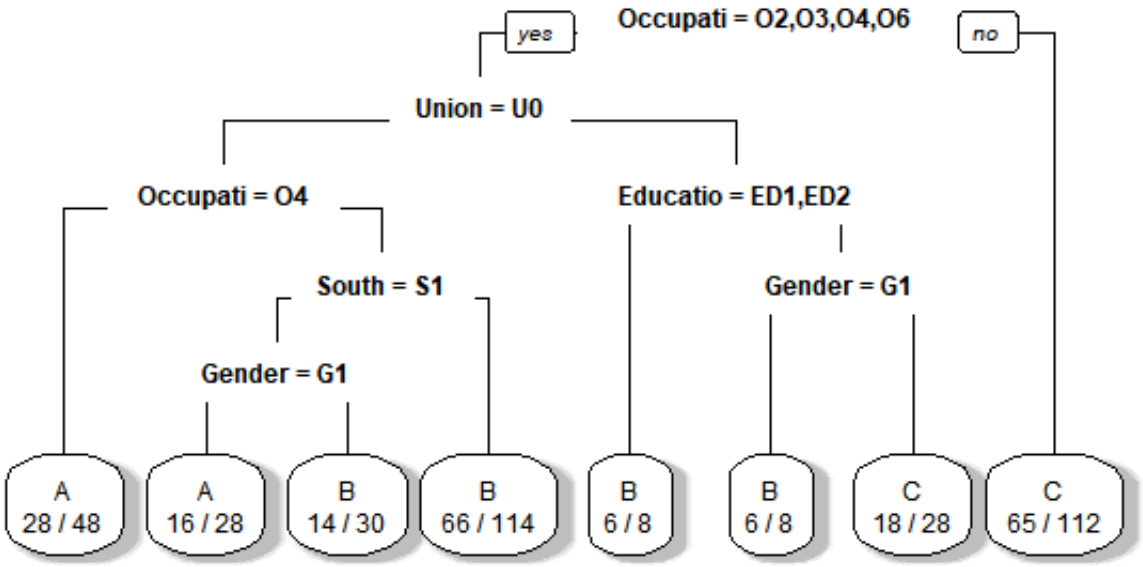
圖4-2. target variable and variables relation plot

我們可以觀察到目標變數與大部份預測變數之間的關係都不明顯，這裡明顯指的是在不同Wage之下，預測變數的比例沒有太大的差異，也就是說預測變數對於目標變數的辨別度並不高，可想而知預測的正確率可能也會因此表現不理想。

[方法介紹]
** XGboost

eXtreme Gradient Boosting(極限梯度提升)，主要是基於梯度提升決策樹 (Gradient Boosted Decision Tree，GBT)。 此模型為 Tree Ensemble，所謂 Tree Ensemble 主要是由分類和迴歸樹 (Classification And Regression Trees，CART) 所組成，同時 Boosted Tree 最基本組成部份就是 CART，所謂 CART 會將輸入根據不同的屬性分配至各個葉子節點，每個葉子節點皆會對應一個分數，此時我們可以針對葉子節點的分數進行更多模型結果的解釋。然而一個 CART 通常過於簡單而無法有效的進行預測，透過增量訓練 (Additive Training) 的方式，每一次保留原來的模型不變，並且加入一個新的函數至我們的模型中，也就是說每一步我們皆會在前一步的基礎上增加一顆樹，以利修復上一顆樹的不足，有助於提升目標函數。

[結果解釋]
(i) Decision Tree



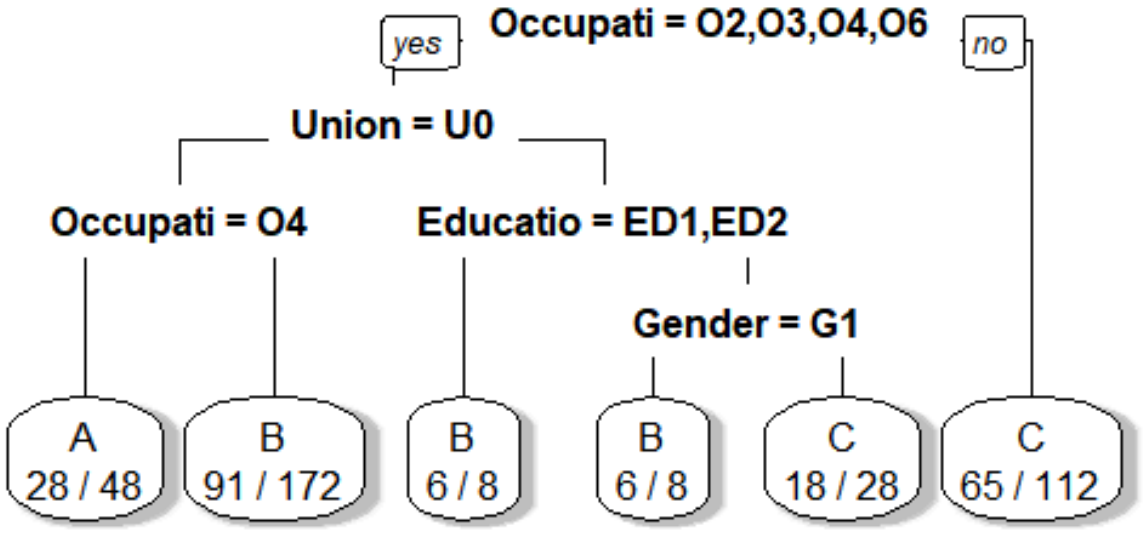


圖4-3. unpruned/pruned Decision Tree

與Q3的概念相同，我們選擇了能夠使accuracy最高的CP作為修剪樹的指標，未經修剪得樹使用了五個變數，Education Gender Occupation South Union，而修剪過後的樹僅使用了四個變數，少了South，作為分類規則，而分類的方式更為簡單。

```
> printcp(nw.treecv)  # true error rate for nsplit = 7:
205*0.88292683/376 = 48.14%

Classification tree:
rpart(formula = Wage ~ ., data = training.1, method = "c
lass",
    control = nw.control)

Variables actually used in tree construction:
[1] Education  Gender     Occupation South
[5] Union

Root node error: 205/376 = 0.54521

n= 376

        CP nsplit rel error  xerror     xstd
1 0.112195      0   1.00000 1.00000 0.047101
2 0.026829      1   0.88780 0.88780 0.047270
3 0.021951      3   0.83415 0.90732 0.047292
4 0.012195      5   0.79024 0.79024 0.046840
5 0.010000      7   0.76585 0.88293 0.047262
```

圖4-4. Decision Tree LOOCV

最後我們使用LOOCV來對true accuracy做估計，而true error rate為0.4814 ，即true accuracy rate為0.5286。

(ii) Random Forest

```
> print(nw.rf4)  # best mtry = 2 , acc. = 0.5155985945
Random Forest

376 samples
 10 predictor
  3 classes: 'A', 'B', 'C'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 339, 338, 338, 337, 338, 339, .
..
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   1    0.4807161  0.07482386
   2    0.5155986  0.18655616
   3    0.5025944  0.18475276
   4    0.5005841  0.19471316
   5    0.4964103  0.19562036
   6    0.4945585  0.19653534
   7    0.4919507  0.19762999
   8    0.4918621  0.20109251
   9    0.4918846  0.20115089
  10    0.4839624  0.18906742

Accuracy was used to select the optimal model
 using the largest value.
The final value used for the model was mtry = 2.
```

圖4-4. accuracy by grid search

這裡我們直接用grid search找出最佳參數，並且以10-fold Cross-Validtion來估計他的true accuracy rate，所選出的最佳參數mtry = 2，true accuracy為0.5155986。

圖4-5. Variable Importance

利用平均損失準確度作為變數重要性指標的選取，前三重要的變數為Occupation, Union, Education。

(iii) XGboost

```
> result.cv <- xgb.cv(nfold = 10,param=param, data=xdata, label=Y, nround
s=20)
[1]     train-merror:0.309116+0.015189  test-merror:0.508108+0.077564
[2]     train-merror:0.299064+0.012543  test-merror:0.510882+0.082985
[3]     train-merror:0.286945+0.013430  test-merror:0.513656+0.086573
[4]     train-merror:0.277783+0.016641  test-merror:0.518919+0.082276
[5]     train-merror:0.267738+0.016211  test-merror:0.518848+0.083409
[6]     train-merror:0.261239+0.013604  test-merror:0.513656+0.089842
[7]     train-merror:0.252965+0.013918  test-merror:0.518990+0.086985
[8]     train-merror:0.244101+0.016201  test-merror:0.502845+0.093007
[9]     train-merror:0.238485+0.019235  test-merror:0.508321+0.092097
[10]    train-merror:0.233455+0.017432  test-merror:0.502916+0.094034
[11]    train-merror:0.226066+0.015239  test-merror:0.516145+0.082123
[12]    train-merror:0.221931+0.015891  test-merror:0.521551+0.073705
[13]    train-merror:0.216020+0.014004  test-merror:0.513727+0.088539
[14]    train-merror:0.208043+0.013769  test-merror:0.524324+0.089730
[15]    train-merror:0.205679+0.014202  test-merror:0.521551+0.087672
[16]    train-merror:0.203310+0.012038  test-merror:0.518990+0.090669
[17]    train-merror:0.197107+0.010029  test-merror:0.524324+0.083510
[18]    train-merror:0.195924+0.010780  test-merror:0.524324+0.082676
[19]    train-merror:0.191785+0.010099  test-merror:0.535064+0.081221
[20]    train-merror:0.187943+0.009162  test-merror:0.537624+0.072664
```

圖4-6. Cross-Validation for XGboost

這裡直接用Cross-Validation來看XGboost所建立的模型估計true error rate為0.537624，即true accuracy rate為0.462376。

表4-7. 三種方法比較True Accuracy Rate

Method	True Accuracy rate
Decision Tree	0.5286
Random Forest	0.5155986
XGboost	0.462376

由上述得知，oversampling rf跟NN-1有overfitting的疑慮。

參考資料：

https://zh.wikipedia.org/wiki/典型相关
https://www.zhihu.com/question/269698662