STATISTICAL ANALYSIS
OF
FEDERAL RESERVE BANK OF BOSTON

目錄

1. Introduction and Data Description

(a) 分析目的

貸款，是公司需要資金、個人有買車買房需要大筆金錢或急需時，大部分會從這個合法的管道取得資源，而銀行在評估是否貸款給借用人也會進行多方考量，包含借用人過去的信貸資料、是否有不良的信用、工作收入…等，因此我們想了解有哪些項目是銀行評估的重要考量，是否只藉由幾個主要項目就能評斷借貸與否，以及觀察這些結果是否合理。

(b) 資料說明

為探討那些項目是銀行決定借貸的重要考量，本次分析使用 R datasets 內 Cross-section data on the Home Mortgage Disclosure Act (HMDA).

(https://www.rdocumentation.org/packages/AER/versions/1.2-6/topics/HMDA)，此資料集為 The federal reserve bank of Boston 於 1989 年收集 2380 個借貸申請人的資料，包含反應變數是否借貸，以及申請人是否為非裔美國人、過去借貸信用、單身與否……，共 14 個變數，變數名稱和敘述見下表，

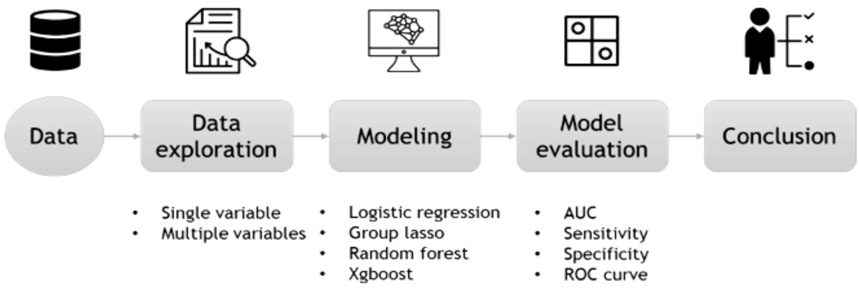
變數類別	變數名稱	變數解釋
反應變數	deny	是否拒絕抵押(1 = 是 ; 0 = 否)
變數	pirat	支出與收入比例
	hirat	住房費用與收入比例
	lvrat	貸款與抵押房價比例
	chist	過去消費支出信用 1 = if no “slow pay” account 2 = if one or two slow pay accounts 3 = if more than two slow pay accounts 4 = if insufficient credit history for determination 5 = delinqient credit history with 60 days past due 6 = serious delinquencies with 90 days past due
	mhist	過去償還貸款信用 1 = if no late payments 2 = if no payment history 3 = if one or two late payments 4 = if more than two late payments

變數	phist	是否有公共不良信用 (1 = 是 ; 0 = 否)
	unemp	申請人行業失業率
	selfemp	是否為自雇人士 (1 = 是 ; 0 = 否)
	insurance	是否拒絕申請房屋保險 (1 = 是 ; 0 = 否)
	condmin	抵押的房子是否為共有財產 (1 = 是 ; 0 = 否)
	afam	是否為非裔美國人 (1 = 是 ; 0 = 否)
	hschool	是否有高中文憑
	single	申請人是否單身 (1 = 是 ; 0 = 否)

(c) 分析流程

在本次分析當中，首先我們會進行資料探索，了解各個變數的分布和不同變數之間的關係；對資料有一定的了解後，由於分析目的是為了找出重要變數，故我們使用四種具選變數的方法建立模型，最後比較不同模型的結果。

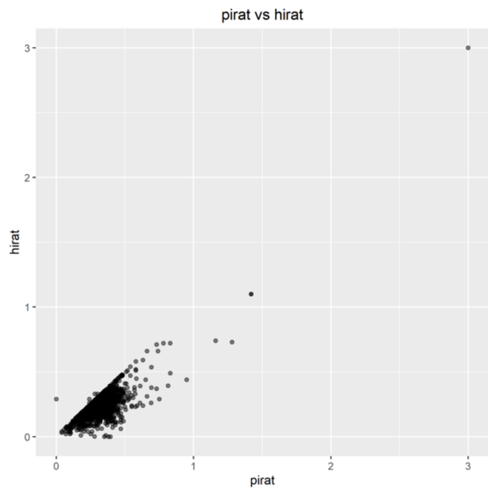
本分析報告分為五章節，第二章為對原始資料所進行的探索資料結果，第三章介紹使用的分析方法，接著第四章將說明各方法之分析結果和討論，最後，第五章為本報告之總結。



2. Exploratory Data Analysis

(a) Continuous Covariates

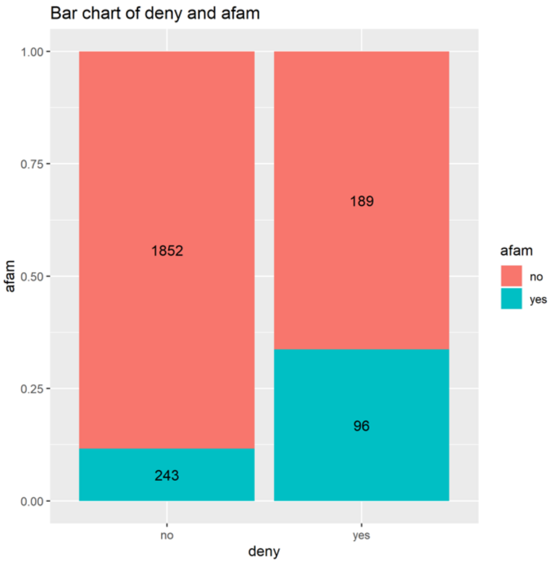
pirat v.s. hirat : 發現此兩個變數之間呈現高度正相關(相關係數:0.78)，可能是由於兩個變數皆是與收入的比例關係，且支出越高者越容易支付越高住房費用。

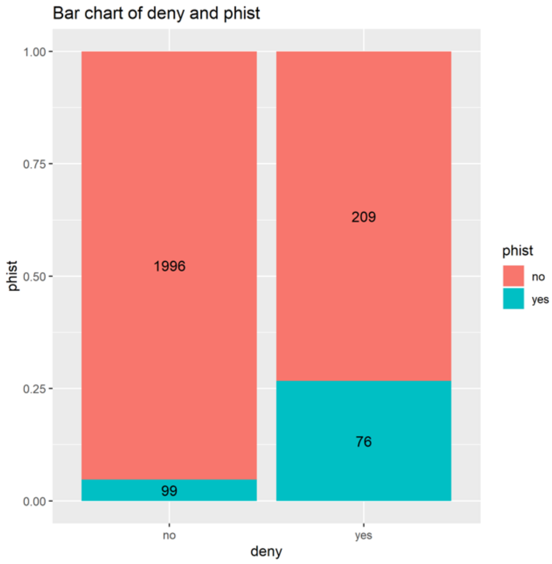


(b) Categorical Covariates

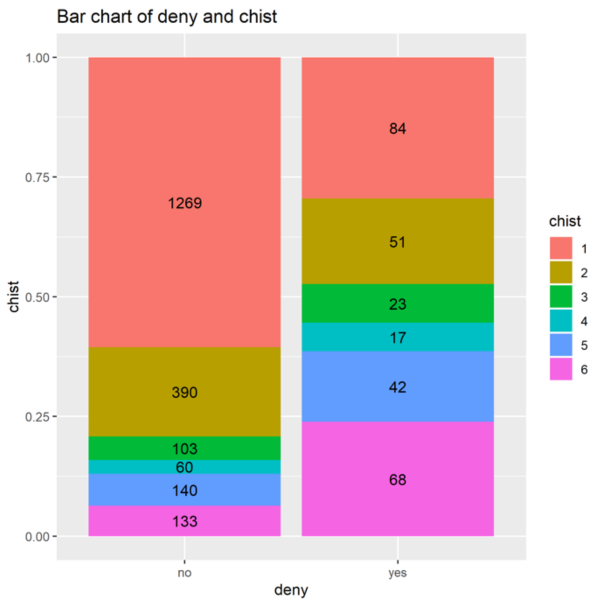
(1) Afam：從 bar chart of deny and afam 可以看到，被拒絕的申請人裡有大約三分之一的人是非裔美國人，比沒有被拒絕的比例高出很多，猜測銀行可能存有種族歧視。

(2) Phist：從 bar chart of deny and phist 可以看到 phist（公共不良信用）在拒絕與不拒絕裡的比例明顯不同，推論 phist 對於銀行是否拒絕可能是具有影響力的項目。

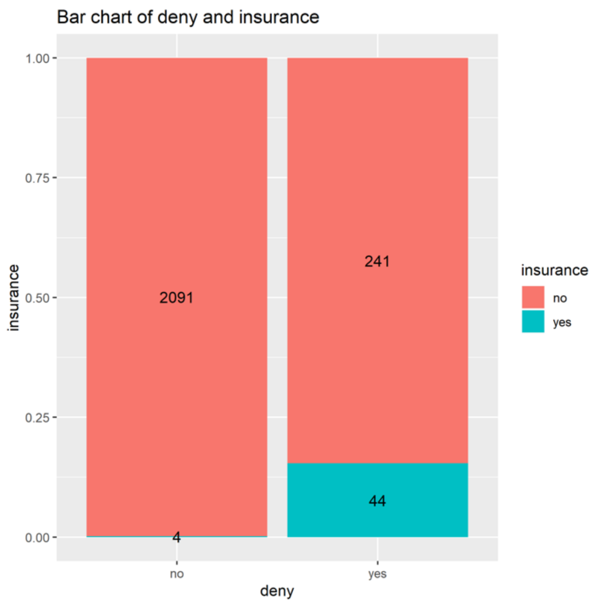




(3) Chist: 由 barchart of deny and chist 可以看出，chist（過去消費信用）除了在拒絕與不拒絕裡的各程度比例分配明顯不同外，也可看出 chist 排序越高被拒絕的比例也越高，推論 chist 可能是對銀行裁定具有影響力的項目。

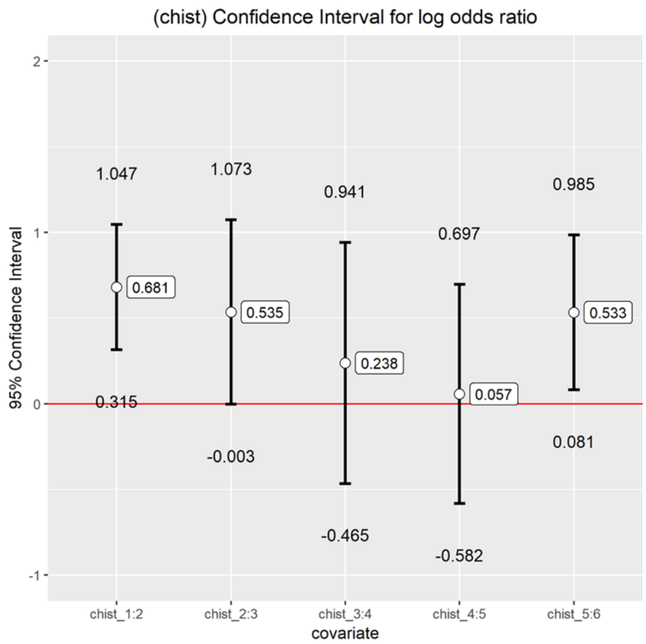


(4) Insurance: 由 barchart of deny and insurance 可以看出 insurance（是否拒絕抵押貸款保險）在拒絕與不拒絕裡的比例明顯不同，推論 insurance 對於銀行是否拒絕可能是一項具有影響力的項目。



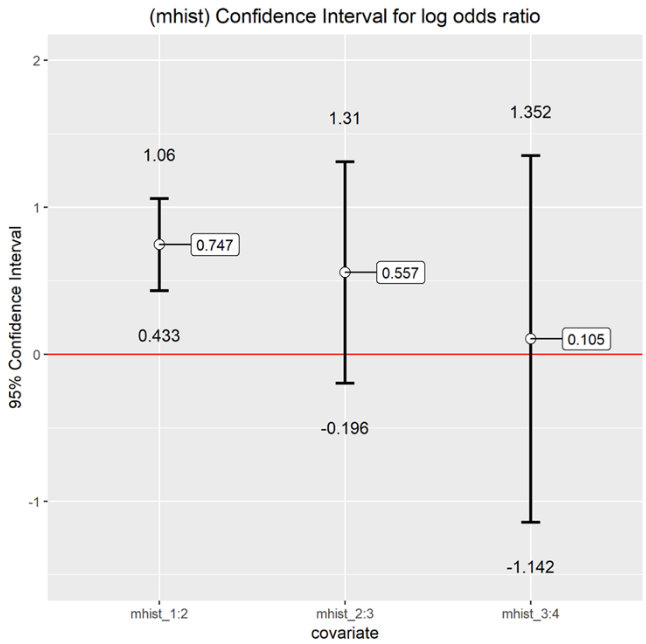
(c) Odds Ratio

(i) Chist



上圖為消費者支付信用記錄的勝算比。從排序 1 到 6，估計的勝算比（紅點）皆大於 1，但其中 3:4、4:5 組勝算比的信賴區間（紅色長條）皆跨過 0，可以說明排序上越高，被拒絕的勝算比越大，但其中三、四、五組較無明顯差別。經獨立性檢定，消費者支付信用記錄與被拒絕與否不無相關。

(ii) mhist

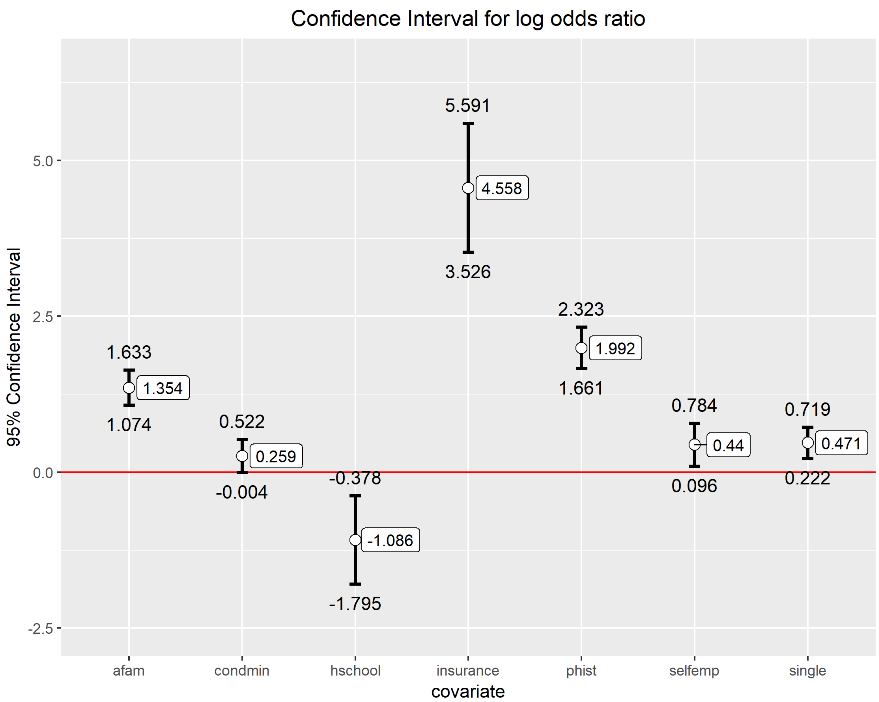


上圖為抵押付款的信用記錄的勝算比。其中估計的參數皆大於一，但後兩組的信賴區間跨過 0，差別較不明顯。

經獨立性檢定，抵押付款的信用記錄的排序與被拒絕貸款不無相關。

(iii) afam、condmin、hschool、insurance、phist、selfemp、single

下圖為上述七個變數的勝算比與其信賴區間：



綜上所述，接著將進行建模並評估變數重要性。

3. Methods

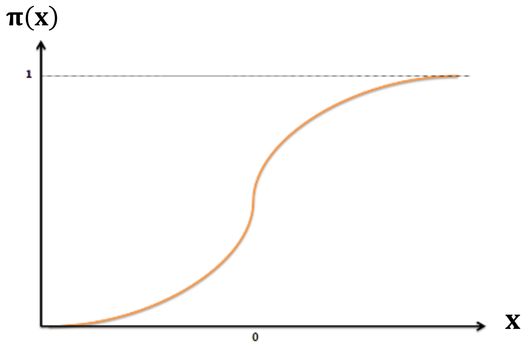
(a) Logistic regression

邏輯斯迴歸 (Logistic Regression) 為迴歸分析中的一種方法，主要用來建立「二元目標變數」 (binary response) 和解釋變數 (explanatory variable) 間的關係，其解釋變數可為連續型和類別型。

定義$\pi(\mathrm{x}) = P(Y = 1|X = x)$，模型為

$$\pi(\mathrm{x}) = \frac{\exp\,(\alpha + \beta x)}{1 + \exp\,(\alpha + \beta x)}$$

它的特性是其回傳的應變數值用遠介於 0~1 之間，圖形如下：



一個與邏輯斯迴歸密不可分的概念是「勝算（odds）」是指某件事情成功機率與失敗機率的比值，將 odds 取對數後，即可以一個線性結構逼近 π(x)。我們稱 log odds 為 logit：

$$\mathbf{logit}[\pi(\mathrm{x})] = \log\frac{\pi(\mathrm{x})}{1 - \pi(\mathrm{x})} = \alpha + \beta x$$

在配適模型時，若解釋變數中有類別型變數，則會用虛擬變數(dummy variable)來表示。假設現在解釋變數只放一個類別型變數(有 k 類)來配適簡單邏輯斯迴歸模型，則只需要 k-1 個虛擬變數，模型為

$$\mathbf{logit}[\pi(\mathrm{x})] = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{k-1} x_{k-1}$$

若要適配一個多元邏輯斯迴歸模型(multiple logistic regression)，

解釋變數 $\mathbf{x} = (x_1, \ldots, x_p)$有 p 個，假設$\pi(\mathbf{x}) = P(Y = 1|\boldsymbol{X} = \boldsymbol{x})$，模型為

$$\mathbf{logit}[\pi(\mathbf{x})] = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

將這個等式反推回去可以得到

$$\pi(\mathbf{x}) = \frac{\exp\,(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)}{1 + \exp\,(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)}$$

在這裡估計參數 α 和 β_j 的方法並不是之前常用的最小平方法，而是使用最大概似估計法(Maximum likelihood estimation)，原因是因為在迴歸分析中的 Y 是已經觀察到的資料，而邏輯斯迴歸中$P(Y=1|\boldsymbol{X}=\boldsymbol{x})$是資料裡面無法觀察到的，因此無法使用最小平方法估計。而在使用最大概似法估計之前，要先建立$\mathbf{x}$的概似函數，假設現在母體有$Y_1,\dots,Y_N$，隨機抽 n 個$Y_1,\dots,Y_n$為樣本，設$p_i=\pi(x_i)$，則x_i的質量密度函數(probability mass function)為$\pi(x_i)^{y_i}[1-\pi(x_i)]^{1-y_i}$，$\mathbf{x}$的概似函數為

$$\prod_{i=1}^{n}\pi(x_i)^{y_i}[1-\pi(x_i)]^{1-y_i}=\prod_{i=1}^{n}\left\{\exp\left(\log\left(\frac{\pi(x_i)}{1-\pi(x_i)}\right)^{y_i}[1-\pi(x_i)]\right)\right\}$$

$$=\prod_{i=1}^{n}\left\{\exp\left(y_i\log\left(\frac{\pi(x_i)}{1-\pi(x_i)}\right)+\log[1-\pi(x_i)]\right)\right\}$$

$$=\exp\left\{\sum_i y_i\log\left(\frac{\pi(x_i)}{1-\pi(x_i)}\right)\right\}\prod_{i=1}^{n}[1-\pi(x_i)]$$

因為 $\log\left(\frac{\pi(x_i)}{1-\pi(x_i)}\right)=\sum_j\beta_j x_{ij}\rightarrow\exp\left\{\sum_i y_i\log\left(\frac{\pi(x_i)}{1-\pi(x_i)}\right)\right\}=\exp\left\{\sum_j(\sum_i y_i x_{ij})\beta_j\right\}$

和 $1-\pi(\boldsymbol{x_i})=(1+\exp(\sum_j\beta_j x_{ij}))^{-1}$，因此 log likelihood function 為

$$\mathrm{L}(\boldsymbol{\beta})=\log\{\exp\left\{\sum_j(\sum_i y_i x_{ij})\beta_j\right\}\prod_{i=1}^{n}\left(1+\exp\left(\sum_j\beta_j x_{ij}\right)\right)^{-1}\}$$

$$=\sum_j(\sum_i y_i x_{ij})\beta_j-\sum_i\log[1+\exp(\sum_j\beta_j x_{ij})]$$

我們想找到一個$\boldsymbol{\beta}$(α包含在裡面)使得 log likelihood 值達到最大，如下:

$$\underset{\boldsymbol{\beta}}{\operatorname{argmax}}\,\mathrm{L}(\boldsymbol{\beta})=\underset{\boldsymbol{\beta}}{\operatorname{argmax}}\sum_j(\sum_i y_i x_{ij})\beta_j-\sum_i\log[1+\exp(\sum_j\beta_j x_{ij})]$$

因此我們可以對$\mathrm{L}(\boldsymbol{\beta})$作一階導數並令它為 0 求極值:

$$\frac{\partial\mathrm{L}(\boldsymbol{\beta})}{\partial\beta_j}=\sum_i y_i x_{ij}-\sum_i x_{ij}\frac{\exp(\sum_k\beta_k x_{ik})}{1+\exp(\sum_k\beta_k x_{ik})}=0$$

則概似方程(likelihood equation)為

$$\sum_i y_i x_{ij}-\sum_i\hat{\pi}_i x_{ij}=0\,,\mathrm{j}=1,\dots,p$$

其中 $\hat{\pi}_i$ 為 $\pi(\boldsymbol{x_i})$的 MLE 且

$$\hat{\pi}_i = \frac{\exp\left(\sum_k \hat{\beta}_k x_{ik}\right)}{1+\exp\left(\sum_k \hat{\beta}_k x_{ik}\right)} = \frac{1}{1+\exp\left(-\sum_k \hat{\beta}_k x_{ik}\right)}$$

在$\frac{\partial \mathrm{L}(\boldsymbol{\beta})}{\partial \beta_j}$的部分因為沒有 closed form，可以用牛頓法迭代出$\hat{\beta}_k$，就可以根據上面的式子得到我們想要知道的目標函數$\hat{\pi}_i$。

(b) Group LASSO

為了改善線性迴歸中 overfitting 造成的預測誤差進而提高預測準確率，以及在建立模型中同時選取重要的變數，Robert Tibshirani 在 1996 年提出 LASSO(least absolute shrinkage and selection operator)方法，此方法是基於最小平方法，同時對估計參數給定一限制式，目標函數為：

$$min \sum_{i=1}^{n} (y_i - x_i'\beta)^2 \ \ s.t. \sum_{j=1}^{p} |\beta_j| \le t$$

其中，x_i為的解釋變數、y_i為反應變數，t 則為限制估計參數的範圍。上式可以拉格朗日形式改寫為：

$$min \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

其中，λ為調控參數，$\lambda \sum_{j=1}^{p} |\beta_j|$為懲罰項(penalty term)，當$\lambda \to 0$時等同於$t \to \infty$，則未對參數有任何限制，即為最小平方法的目標函數；當λ遞增 t 遞減時，參數會被限制在一範圍內，以此避免 overfitting，甚至有些參數會被壓縮至 0，則對應參數不為 0 的變數則為被挑選出的重要變數。

在上述的方法中，若遇到解釋變數為類別型的變數，則需要轉為 dummy variable，若該類別中有 m 個 class，則會以 m-1 個新變數替代原本的類別型變數。當要透過參數壓縮的方式挑選重要變數，則會遇到某個類別型變數中的部分 class 被挑出，而非原本的類別型變數被挑出，因此上述的 lasso 方法並不適用於有類別型解釋變數的資料集。然而，只要類別變數中 m 個 class 所形成的 m-1 個新變數對應的估計參數同時為 0 會不為 0，就能解決上述問題，因此可以利用 group LASSO 方法處理包含類別型變數的資料集。

Group LASSO 是 2006 年由 Yuan and Lin 提出的方法，其概念是將 p 個變數分成不重疊的 K 組，也就是$(1,2,\ldots,p) = \cup_{k=1}^{K} I_k$且$I_k \cap I_{k'} = \emptyset$，其中$I_k$集合中的個數是$p_k$，則目標函數為：

$$min \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \sum_{k=1}^{K} \sqrt{p_k} ||\beta^{(k)}||_2$$

其中$||\beta^{(k)}||_2 = \sqrt{\sum_{j \in I_k} \beta_j^2}$ ，在此方法中，除了將原本的變數分組，同組的變數會

同時被壓縮至 0，也利用$\sqrt{p_k}$做為懲罰項的權重，當$\sqrt{p_k}$全為 1 時，即為原本的 LASSO。

回到我們同一類別變數的 m-1 個新變數是否可以同時被選取的問題，立用 group LASSO 分組的方式，指定同一個類別變數所產生的 dummy variable 新變數為同一組，就可以解決只選到特定 class 的問題。

另外，由於我們所使用的資料反應變數為類別型，因此無法使用 squared error loss function，取而代之的是利用 logistic regression 中的 negative binomial log-likelihood 函數做為 loss function，因此目標函數為：

$$\min -\left[\sum_{i=1}^{n} y_i(\beta_0 + x_i'\beta) - \log\left(1 + e^{(\beta_0 + x_i'\beta)}\right)\right] + \lambda \sum_{k=1}^{K} \sqrt{p_k}||\beta^{(k)}||_2$$

(c) Random forest

Random Forest 是一種 Ensemble Learning 的演算法，而 Ensemble Learning 概念上是結合多個弱學習器來建構一個強穩的模型。每次會用類似 Bootstrap Aggregation 的方法取得一個新的 Decision Tree，再將所有的 Decision Tree 結合起來。

其中 Decision tree（決策樹）是一種監督式機器學習模型，利用像樹一樣的圖形去建構預測模型，適用於類別及連續資料類型的預測，相較於其他 Machine Learning 的模型，Decision Tree 的過程直覺、單純且執行效率也很高。決策樹的特點是每個決策階段都很清楚，每個節點代表一個屬性（變數)、每個分支代表對應該屬性的某些可能值（變數範圍)、每個葉節點代表滿足對應路徑的條件下之最終的預測值。

因此將所有的 Decision Tree 結合起來不僅能夠保持 Decision Tree 的差異性，還能減少 fully-grown 造成的 overfitting。Random Forest 最大的精神就是隨機，除了樣本是利用 Bootstrap 採取抽後放回的隨機抽取的概念外，變數也是採隨機抽取的方式，也因此 Random Forest 具有高度多樣性。

$$\text{Model: } \hat{f}^{rf}(x) = \underset{k=1,2,,,K}{argmax} \sum_{b=1}^{B} I\left(\hat{f}^{tree,b}(x) = k\right)$$

Random Forest 的結果可以計算每個特徵的重要程度，在 R 語言中，最後估計出的模型會提供「Mean Decrease Accurary」及「Mean Decrease Gini」，兩者皆可用來進行特徵選取。Mean Decrease Accurary 大致上是將 data 中第 i 個變數抽取出後隨機打亂再放入 data 中，將新 data 代入模型計算出新的 Accuracy 後，比較 Accuracy 與原先的差異。Mean Decrease Gini 則是計算該變數讓整個模型之不純度下降的比例。

其中不純度（impurity）可以為(1) Misclassification rate ,(2) Entropy 或 (3) Gini Index

$$(1)\ \emptyset(p_1, p_2, \ldots, p_K|t) = 1 - max(p_1, p_2, \ldots, p_K|t),$$
$$(2)\ \emptyset(p_1, p_2, \ldots, p_K|t) = -\sum_{j=1}^{K} p(j|t)\log p(j|t),$$
$$(3)\ \emptyset(p_1, p_2, \ldots, p_K|t) = 1 - \sum_{j=1}^{K} p(j|t)^2,$$

where $p(k|t)$ is estimated by $\hat{p}(k|t) = \frac{1}{n_t}\sum_{x_i \in node_t} I(y_i = k)$

(d) XGBOOST

所謂的 GBM 算是一種概念，是將梯度下降法（Gradient Descending）跟 Boosting 套件節合在一起的演算法，而後面的 Machine 指不特定的模型，只要能用梯度下降法找尋方向的模型都可以。

如果使用 gbm 的套件，基本上都是 Tree-based 為主，也就是將數百個弱決策樹（CART），跟梯度下降法和 Boosting 結合在一起。

其中使用 xgb.cv()的函式，搭配 cross validation 的技巧，找出最佳的決策樹數量 nrounds。過程中，設定 early_stopping_rounds = 30（如果當 nrounds < 30 時，就已經有 overfitting 情況發生，那表示不用繼續 tune 下去了，可以提早停止），程式會根據 Train 和 Validation 的平均表現，自動判斷模型是否有 overfitting，最後找出較好的 nrounds，會是一個最不 overfitting 的模型。

要注意的是，這個最不 overfitting 的模型，是建立在一開始的基本參數設定之下，所以不一定是最好的。（上述 Validation 這個字在 cv.xgb() 的 output 會是 Test 這個字）

XGBoost 是 Gradient Boosting Decision Tree（GBDT）的改良版本，使用多個弱分類器來建構一個強分類器，使用前 m-1 次迭代結果的負梯度（negative gradient）當作新的反應變數進行下一次迭代，產生新的弱分類器（tree）加到前一次的估計函數上當作新的估計函數。

MODEL :

$$\hat{y}_i = \hat{f}^M(x_i) = \sum_{m=1}^{M} h_m(x_i) = \hat{f}^{M-1}(x_i) + h_M(x_i)$$

M 為迭代次數，h_m為每次迭代所新增的弱分類器。

藉由最小化目標函數

$$Obj = \sum_i L\left(y_i, \hat{f}^M(x_i)\right) + \sum_m \Omega(h_m)$$

來找到估計函數，而其中$\Omega(h_m)$ 為 constraint function。

這裡使用 Logistic loss function:

$$L\big(y_i, f(x)\big) = yln\big(1 + e^{-f(x)}\big) + (1 - y)\ln\,(1 + e^{f(x)})$$

第 m 次的迭代結果，目標函數可以寫成:

$$Obj^{(m)} = \sum_i L\left(y_i, \hat{f}^{m-1}(x_i) + h_m(x_i)\right) + \Omega(h_m) + constant$$

我們可以把 loss function 當做要展開的函數，使用泰勒展開式的二次逼近

$$Obj^{(m)} = \sum_i \left[L\left(y_i, \hat{f}^{m-1}(x_i)\right) + \mathrm{g}_i h_m(x_i) + \frac{1}{2}\mathrm{s}_i h_m^2(x_i)\right] + \Omega(h_m) + constant$$

其中$\mathrm{g}_i = \partial_{\hat{f}(m-1)} L\left(y_i, \hat{f}^{m-1}(x_i)\right)$ ， $\mathrm{s}_i = \partial_{\hat{f}(m-1)}^2 L\left(y_i, \hat{f}^{m-1}(x_i)\right)$

因為前 m-1 次的結果已確定，$L\left(y_i, \hat{f}^{m-1}(x_i)\right)$ 為已知常數併入常數項，目標函數可以改寫成

$$Obj^{(m)} = \sum_i \left[\mathrm{g}_i h_m(x_i) + \frac{1}{2}\mathrm{s}_i h_m^2(x_i)\right] + \Omega(h_m) + constant$$

假設樣本 x 的輸出落在第 j 個葉子上，那麼樣本 x 的輸出值為 w_j 為每個葉節點相對應權重。

給定一顆樹，起到分類作用的其實是 nodes，輸入一個樣本，葉子的輸出值 h_m 就是預測的一顆決策樹，預測的結果是由決策樹的 nodes 所決定的。

對於分類問題，決策樹的葉子就是指類別，對於回歸問題，葉子的值就是數值。

$$h_m(x_i) = \sum_{j=1}^{T_m} w_j I(x_i \in R_{jm})$$

其中 T_m為 nodes 個數。

正則項：決策樹的正則一般考慮的是葉子節點數和葉子權值，常見的是使用葉子節點總數和葉子權值平方和的加權作為正則項：

$$\Omega(h_m) = \gamma T_m + \frac{1}{2}\lambda \sum_{j=1}^{T_m} w_j^2$$

$$\begin{aligned}
Obj^{(m)} &\approx \sum_i \left[\mathrm{g}_i h_m(x_i) + \frac{1}{2}\mathrm{s}_i h_m^2(x_i)\right] + \Omega(h_m) \\
&\approx \sum_i \left[\mathrm{g}_i \sum_{j=1}^{T_m} w_j I(x_i \in R_{jm}) + \frac{1}{2}\mathrm{s}_i \sum_{j=1}^{T_m} w_j^2 I(x_i \in R_{jm})\right] + \gamma T_m + \frac{1}{2}\lambda \sum_{j=1}^{T_m} w_j^2 \\
&\approx \sum_{j=1}^{T_m} \left[\left(\sum_{x_i \in R_{jm}} \mathrm{g}_i\right) w_j + \frac{1}{2}\left(\sum_{x_i \in R_{jm}} \mathrm{s}_i + \lambda\right) w_j^2\right] + \gamma T_m
\end{aligned}$$

Define $G_j = \sum_{x_i \in R_{jm}} g_i$ ， $S_j = \sum_{x_i \in R_{jm}} s_i$

$$\approx \sum_{j=1}^{T_m}\left[G_j w_j + \frac{1}{2}(S_j + \lambda) w_j^2\right] + \gamma T_m$$

把$Obj^{(m)}$對w_j微分求最大值→ $\frac{\partial Obj^{(m)}}{\partial w_j} = 0 \rightarrow w_j^* = -\frac{G_j}{S_j + \lambda}$ 帶回$Obj^{(m)}$

$$Obj^{(m)} = -\frac{1}{2}\sum_{j=1}^{T_m}\frac{G_j^2}{S_j + \lambda} + \gamma T_m$$

從一開始深度為 0 開始切割，找到有最大分數 Gain 的切割點

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{S_L + \lambda} + \frac{G_R^2}{S_R + \lambda} - \frac{G^2}{S + \lambda}\right] - \gamma$$

不斷地切割下去，直到 Gain < 0 ∀spilt。

把此次迭代結果$h_m(x_i)$乘上 learning rate ν 加到前一次迭代結果上

$$\hat{f}^{(m)}(x_i) = \hat{f}^{(m-1)}(x_i) + \nu h_m(x_i)$$

我們不會在每個步驟中進行 full optimization，有助於防止 overfitting

以下為參數調控：

初始參數給定 nround=200，主要調控 eta 以及 max_depth 對於樹的模型結構影響比較大的參數，使用 5-fold CV 以 AUC 當作選模標準進行調控。

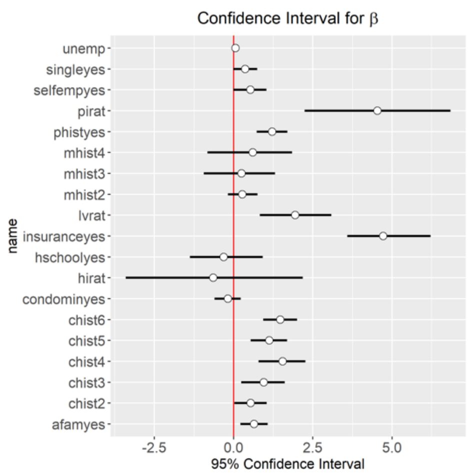
Booster Parameters（模型參數）	
eta	shrinkage 參數，用於更新葉子節點權重時，乘以該係數，避免步長過大。
min_child_weight	控制葉子節點中二階導的和的最小值，該參數值越小，越容易 overfitting。
max_depth	每顆樹的最大深度，樹高越深，越容易 overfitting。
subsample	樣本隨機採樣，較低的值使得算法更加保守，防止 overfitting，但是太小的值也會造成 underfitting。

colsample_bytree	列採樣，對每棵樹的生成用的特徵進行列採樣。
lambda	控制模型複雜度的權重值的 L2 正則化項參數，參數越大，模型越不容易 overfitting。
alpha	控制模型複雜程度的權重值的 L1 正則項參數，參數值越大，模型越不容易 overfitting。
Learning Task Parameters（學習任務參數）	
Objective = ‘binary=logistic’	定義最小化損失函數類型，binary:logistic: logistic regression for binary classification
eval_metric = ‘auc’	The metric to be used for validation data. auc: Area under the curve

而最後利用 grid search 的方法去調控參數，概念就是針對每一個參數組合，都會適配出一個模型，然後從中挑選出最佳的模型。

4. Data analysis

(a) Logistic Regression



圖為 Logistic regression model 其 β 的信賴區間，由此圖可以清楚分辨哪個變數為顯著變數，如果信賴區間沒有包含紅色線($\beta = 0$)，即代表此變數為顯著，在這裡顯著的變數有：pirat、lvrat、chist、phist、selfemp、insurance、afam、single。

以下是針對顯著變數的係數解釋：

(1) 支出與收入比例每增加一單位的支出其被拒絕的 logit 是原來的 4.54 倍

(2) 貸款與房屋價比例每增加一單位的貸款其被拒絕的 logit 是原來 1.94 倍

(3) 過去消費者支出信用第二類被拒絕的 odds 為第一類被拒絕 odds 的 1.71 倍

(4) 過去消費者支出信用第三類被拒絕的 odds 為第一類被拒絕 odds 的 2.6 倍

(5) 過去消費者支出信用第四類被拒絕的 odds 為第一類被拒絕 odds 的 4.7 倍

(6) 過去消費者支出信用第五類被拒絕的 odds 為第一類被拒絕 odds 的 3.08 倍

(7) 過去消費者支出信用第六類被拒絕的 odds 為第一類被拒絕 odds 的 4.36 倍

(8) 有公共信用不良者其 odds 為沒有公共信用不良者其 odds 的 3.38 倍

(9) 自雇人士其 odds 為不是自雇人士的 1.7 倍

(10) 拒絕貸款保險者其 odds 為沒拒絕貸款保險者 odds 的 111.8 倍

(11) 非裔美國人其 odds 為不是非裔美國人 odds 的 1.91 倍

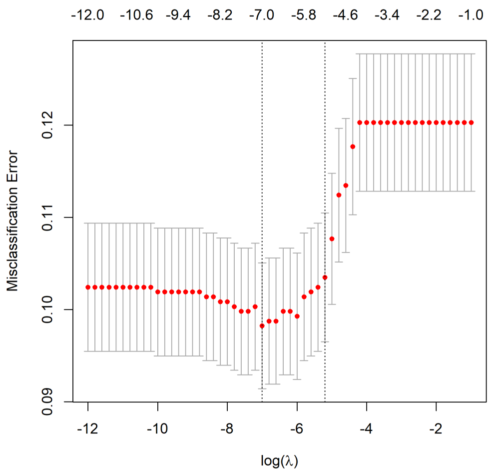
(12) 申請人為單身其 odds 是申請人不為單身 odds 的 1.44 倍

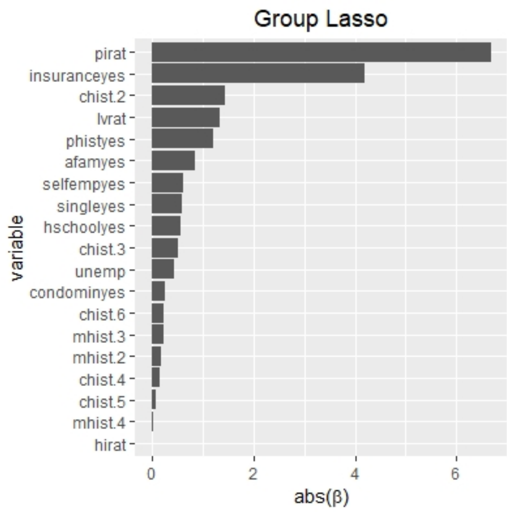
使用 glm 計算 logistic regression 時，family 要設為 binomial，結果如下：

	Sensitivity	Specificity	AUC	ACC
Logistic	0.286	0.99	0.858	0.908

(b) Group LASSO

跟傳統 Lasso 不同的地方在於，Group Lasso 會針對有多類別變數給予同一個 group，才不會發生同一個類別變數中，有些顯著有些不顯著的問題，因此我們給予的 group 個數，會與原本的變數個數同。首先透過 corss validation 的方式(如下左圖)，選出適當的λ值，最後選取到的λ =0.00091。以此建立模型，所選出來的變數除了 hirat 與 hschool 這兩個變數外，其餘皆被選為重要變數(如下右圖)，其中，pirat、insurance、lvrat、phistype、chist2 為前五個影響較大變數，以模型的表現來看，其 Specificity 高達 99.3%，Accuracy 也有 90.5%。





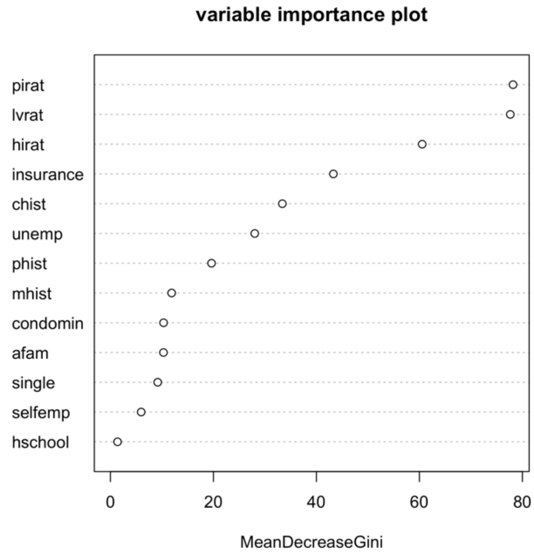
	Sensitivity	Specificity	AUC	ACC
Group Lasso	0.250	0.993	0.851	0.905

(c) Random Forest

Random Forest 中主要需要調控的參數為 ntree 及 mtry，其中 ntree 為每次生成隨機森林中樹的個數、mtry 為每個節點中要進行分支時所考慮的變數個數。會發現當樹的個數到 100 棵時，誤差便會趨於穩定，且藉由 OOB 可以得到在每次考慮 4 個變數時會有最小的誤差，這也恰好大約等於$\sqrt{p}$，因此利用 ntree=100、mtry=5 再 fit 一次 model 後預測 testing data 會得到以下表格。

	Sensitivity	Specificity	AUC	ACC
Random Forest	0.304	0.993	0.861	0.912

接著利用 mean decrease Gini 去選擇重要變數，可得 pirat、lvrat、hirat、insurance、chist 為前五個重要變數。



(d) XGBOOST

初始參數給定 nround=200，主要調控 eta 以及 max_depth 對於樹的模型結構影響比較大的參數，使用 5-fold CV 以 AUC 當作選模標準進行調控。

Booster Parameters（模型參數）	
eta	shrinkage 參數，用於更新 nodes 權重時，乘以該係數，避免步長過大。
min_child_weight	控制葉子節點中二階導的和的最小值，該參數值越小，越容易 overfitting。
max_depth	每顆樹的最大深度，樹高越深，越容易 overfitting。
subsample	樣本隨機採樣，較低的值使得算法更加保守，防止 overfitting，但是太小的值也會造成 underfitting。
colsample_bytree	列採樣，對每棵樹的生成用的特徵進行列採樣。
lambda	控制模型複雜度的權重值的 L2 正則化項參數，參數越大，模型越不容易 overfitting。

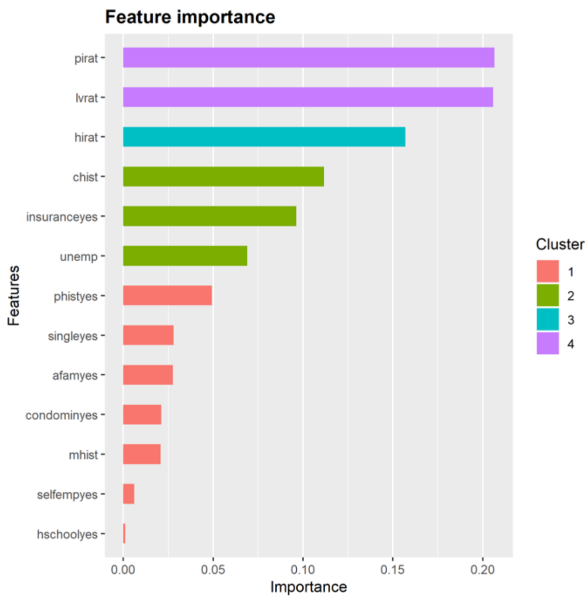
Learning Task Parameters（學習任務參數）	
Objective = ‘binary:logistic’	定義最小化損失函數類型，binary:logistic: logistic regression for binary classification
eval_metric = "auc"	The metric to be used for validation data. auc: Area under the curve

而最後利用 grid search 的方法去調控參數，概念就是針對每一個參數組合，都會適配出一個模型，然後從中挑選出最佳的模型。

Booster Parameters（模型參數）					
eta	0.1	subsample	0.4	lambda	0.8
max_depth	8	colsample_bytree	0.8	alpha	0.2

	Sensitivity	Specificity	AUC	ACC
Xgboost	0.304	0.99	0.91	0.875

接著利用 Gain 值去選擇重要變數，可得 pirat、lvrat、hirat、chist、insurance 為前五個重要變數。



5. Conclusion

(a) 變數討論

我們將各個模型選出來比較重要的前五個變數拿出來看，如下

GroupLasso	RandomForest	Xgboost
pirat	pirat	pirat
insurance	lvrat	lvrat
chist.2	hirat	hirat
lvrat	insurance	chist
phist	chist	insurance

根據上表發現 Random Forest 跟 XGboost 選到的前五個重要變數是一樣的，而「pirat」、「lvrat」、「insurance」、「chist」三種模型都有選到此四個變數，推測這四個變數是影響銀行借貸決策重要依據。其中比較奇怪的是「hirat」在 Random Forest 與 XGboost 兩種方法都是前幾名重要的變數，但在 GroupLasso 卻沒有被選到，發現是因為「hirat」與「pirat」相關性很高，造成有共線性的情況，我們嘗試把「pirat」拿掉後，發現「hirat」變得非常顯著，這說明遇到變數間有共線性的狀況時，必須對變數加以處理，否則可能會造成原本顯著的變數，因為另一個變數在模型裡面，導致其變得不顯著，但在 Random Forest 與 XGboost 兩種方法，不會受到共線性的影響，兩者變數都被選為重要變數，因此在資料分析時，必須觀察資料變數的型態，這是需要注意的地方。

(b) 結論

我們根據以上的分析，找到影響銀行借貸決策的重要變數有「支出與收入比例」、「房屋相關費用與收入比例」、「貸款與抵押房價比例」、「是否拒絕申請房屋保險」、「過去消費者支出信用」前兩項主要是針對其生活開銷與收入的平衡狀態，如果開銷太高，銀行可能會認為借款風險太高，故傾向拒絕借貸;「貸款與抵押房價比例」、「是否拒絕申請房屋保險」 這兩項主要是針對銀行考量房屋價值的變數，前者偏向房屋價值與貸款金額是否能夠平衡，後者偏向保障房屋的價值，銀行認為房屋價值不足或失去價值，則可能傾向拒絕貸款;「過去消費者支出信用」與個人信用相關，若是申請人信用不佳，銀行可能也會認為借款風險太高，故傾向拒絕借貸，綜合以上結果，我們推測「收支比例」、「抵押物價值」、「個人信用」為決定是否借貸的主要依據。