

# Final Homework : Don't Overfit! II

108B (5403) Machine Learning 王傳鈞 0416047、謝仁杰 0412246

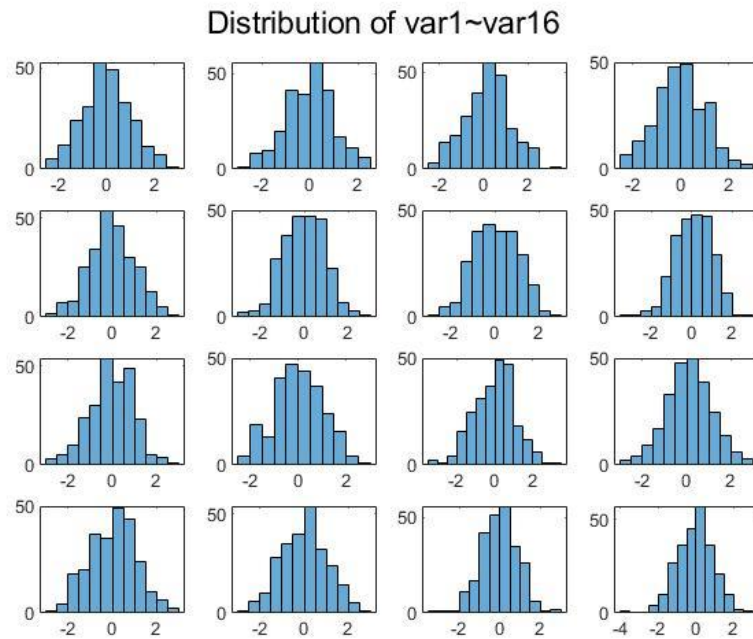
## 簡介

此題目為 binary classification 問題，參賽者要用題目給予的 250 個 training data 訓練模型，分類 19750 個 testing data，每個 data 有 300 個 feature，但沒有說明 data 的出處和 feature 的意義。此題目用 AUCROC 計算分數，以下為比賽結束時，各名次的分數 (Public leaderboard)：

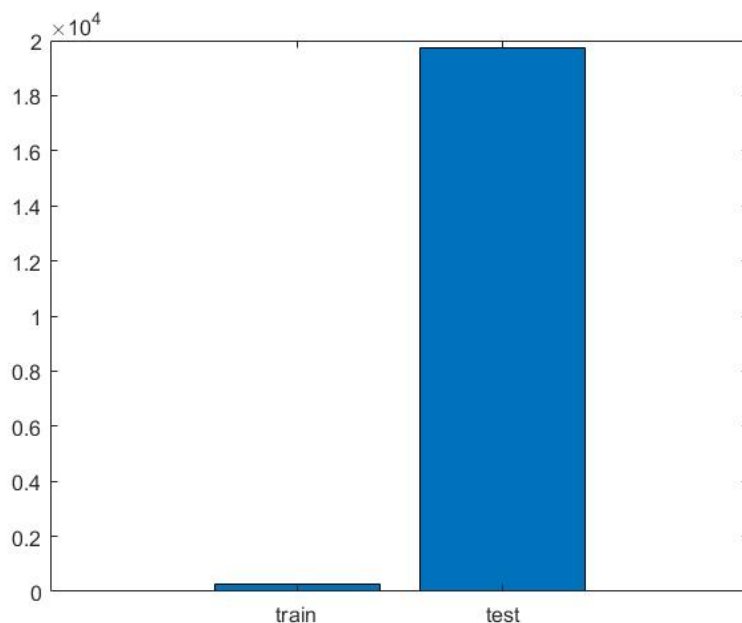
名次	Public Leaderboard Score
1 <sup>st</sup>	0.930
Top 10%	0.868
20%	0.854
30%	0.848
40%	0.844
50%	0.813
60%	0.757
70%	0.722
80%	0.669

## Data Exploration & Visualization

因為比賽提供的 training data 太少，而且 feature 太多，在不做任何預處理的情況下容易發生 overfitting。首先，我們可以從以下直方圖可看出每個 feature 大致符合中心位置約為零的常態分布。



接著，從 target 的分佈可以看出 training data 非常不平衡，所以需要做 oversampling，以避免模型出現傾向分類為特定一種 label 的行為。



## Data Cleaning & Feature Selection

為了避免各 feature 的 scale 不同，所以我們先對訓練集資料(training data)和測試集資料(testing data)做標準化。針對 label 分佈不平衡的現象，我們採用

SMOTE 演算法來處理 training data，控制 positive label 和 negative label 的數目到達一致。

SMOTE：此演算法利用插值來增加少數類的樣本數，使 training data 中 target 為 1 的 instances 和 target 為 0 的 instances 一樣多。

我們使用以下三種 feature selection 的方法：

(1) Pearson's correlation coefficient

先算出 training data 中各 feature 與 target 之間的 correlation，然後保留  $|\text{correlation}| > 0.05$  的 features，其餘全數刪除，不參與接下來任何步驟。

(2) LB Probing

上傳 testing data 當中的某個 feature 當作答案，並用 AUC 結果來推估該 feature 和 ground truth label 之間的關聯程度，然後只保留  $|\text{score} - 0.5| \geq 0.04$  的 features。這是被許多 Kaggle 參賽者應用的技巧，可以當作一種啟發 model selection 靈感或是挖掘更多 data pattern 的技巧。

(3) Do nothing

完全不刪除任何 features，直接進行模型訓練。

## Training a Model

我們用兩種不同的演算法來訓練預測模型，每種演算法都使用 grid search 找出最佳參數。

最初，我們打算本課程裡著墨最多的 support vector machine (SVM) 來解題；但是，經過眾多嘗試之後，我們發現 SVM 受限於輸出結果只能為「0」或「1」，反而失去了某種程度的靈活性。

考量到本題目以 AUCROC 當作 leaderboard score，因此我們的重點應該放

在針對每個 testing data，盡可能正確地給出其屬於 label 0 或 1 的「機率值」，到底以什麼數值範圍的資料上傳並不會影響 AUCROC。為此，我們改以回歸分析的角度來思考，藉由預測其屬於 label 0 或 1 的「機率值」，來提升 AUCROC。

(1) Logistic regression (LogReg)

(2) Support vector regression (SVR)

由以上的結果，我們可以推得以下幾個特點：

(1) LogReg 不管應用在哪一種 feature selection 方法，都比 SVR 來的優秀

在本題目的討論版上，有許多參賽者皆提及 logistic regression 相當適合本題目。根據我們是過眾多的模型裡 (包含 decision tree、KNN、discriminant analysis、naïve Bayes、SVM、SVR、XGBoost 等)，基本上呈現兩種現象：①若模型本身就能建立複雜的分類策略，則 AUCROC 分數會很差；②若我們使用了

Submission and Description	Private Score	Public Score	Use for Final Score
<a href="#">20200703T185302_LogReg_Mode3.csv</a> a few seconds ago by <a href="#">a2468834</a> Logistic Regression with Whole 300 Features	0.826	0.842	<input type="checkbox"/>
<a href="#">20200703T185255_LogReg_Mode2.csv</a> a few seconds ago by <a href="#">a2468834</a> Logistic Regression with LB Probing	0.869	0.891	<input type="checkbox"/>
<a href="#">20200703T185225_LogReg_Mode1.csv</a> a minute ago by <a href="#">a2468834</a> Logistic Regression with Pearson's Correlation Coefficient	0.832	0.846	<input type="checkbox"/>

太多參數，建立出過複雜的模型 (例如採用高維度 polynomial kernel 的 SVM)，

Submission and Description	Private Score	Public Score	Use for Final Score
<a href="#">20200703T185338_SVR_Mode3.csv</a> just now by <a href="#">a2468834</a> Support Vector Regression with Whole 300 Features	0.656	0.686	<input type="checkbox"/>
<a href="#">20200703T191009_SVR_Mode2.csv</a> a minute ago by <a href="#">a2468834</a> Support Vector Regression with LB Probing	0.862	0.888	<input type="checkbox"/>
<a href="#">20200703T185424_SVR_Mode1.csv</a> 2 minutes ago by <a href="#">a2468834</a> Support Vector Regression with Pearson's Correlation Coefficient	0.701	0.746	<input type="checkbox"/>

則 AUCROC 分數比起相同模型但是簡單參數來的差。

因此，我們推測本題目因為提供的 training instance 數目本來就不多 (只有 250 筆，甚至少於 feature 數目)，所以本質上就不適合複雜的模型，容易落入 overfitting 之處境。

(2) 「完全不做 feature selection」顯然不是好方法，對於 LogReg 和 SVR 皆是這個現象完全可以在事前預見，因為本題目提供的 training instance 數目相當少(只有 250 筆)，甚至小於 feature 數目，所以若不採取 feature selection 或是 oversampling 方式擴增樣本數，則可以肯定必面臨 overfitting 之處境。

## Conclusion

一開始，我們依據題目所給的 dataset，來嘗試許多 feature selection 方法，並建立很多不同的分類模型，但顯然距離 baseline(約 0.64)相當遙遠。然而，我們在無意間看到有參賽者討論 training data 不一致的情形。此時，我們才意識到也許題目因為某種緣故，所以目前放置的資料集並不為本題所屬。後續又發現有些參賽者使用相當簡單的模型，就能輕鬆達到還不錯的分數，這又印證了我們的猜想。因此，我們從這個解題的過程，學習到確認資料來源的重要性。

透過這次解題的經驗，我們充分聊解到 overfitting 是如何發生與如何避免。在本課程中，我們理解到機器學習之目的是建立可以預測未知資料的模型，而非建立只能分類手上資料的模型，要如何區別這兩種狀況的不一致，就是察覺是否落入 overfitting，這也是我們覺得這個題目值得一試的原因。