Final Homework : Don't Overfit! II

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

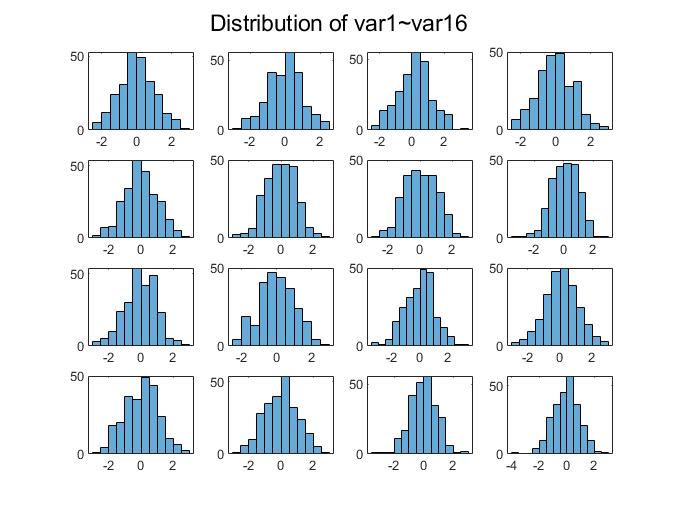
簡介

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)：

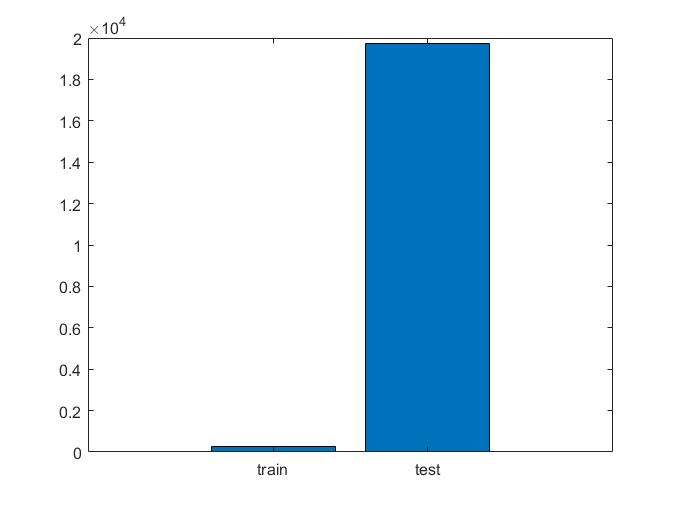
|  |  |
| --- | --- |
| 名次 | Public Leaderboard Score |
| 1st | 0.930 |
| Top 10% | 0.868 |
| 20% | 0.854 |
| 30% | 0.848 |
| 40% | 0.844 |
| 50% | 0.813 |
| 60% | 0.757 |
| 70% | 0.722 |
| 80% | 0.669 |

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，然後保留的features，其餘全數刪除，不參與接下來任何步驟。

1. 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  
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。

1. Do nothing

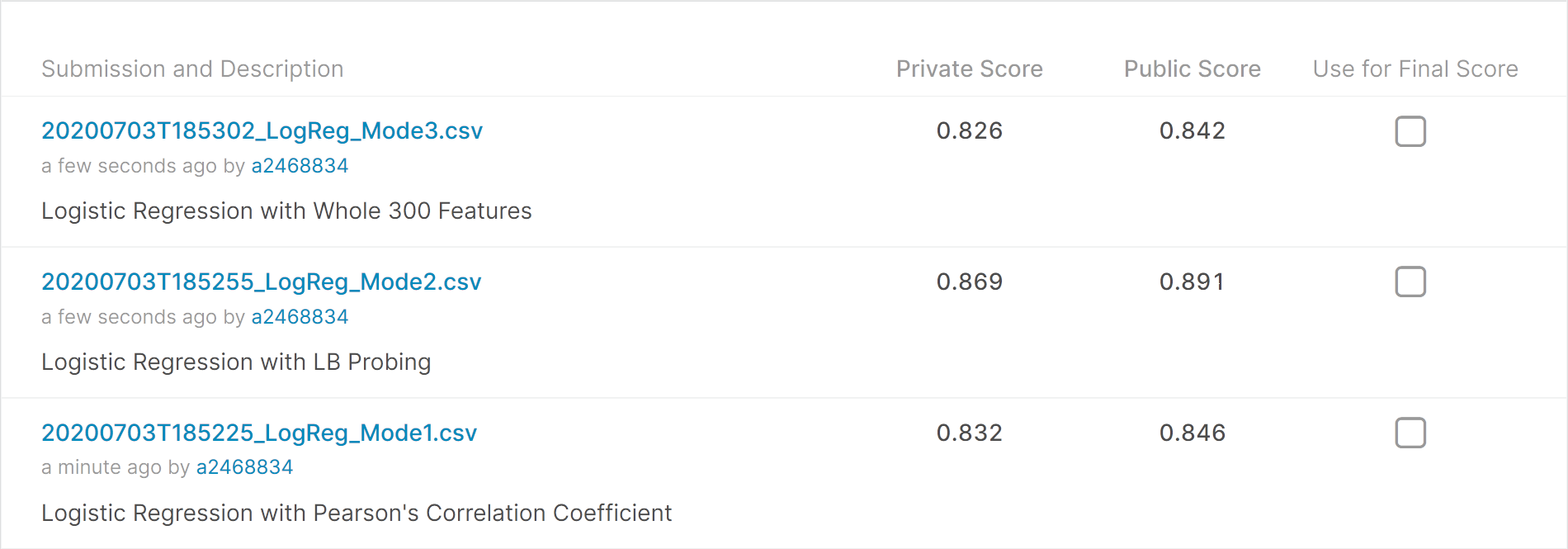
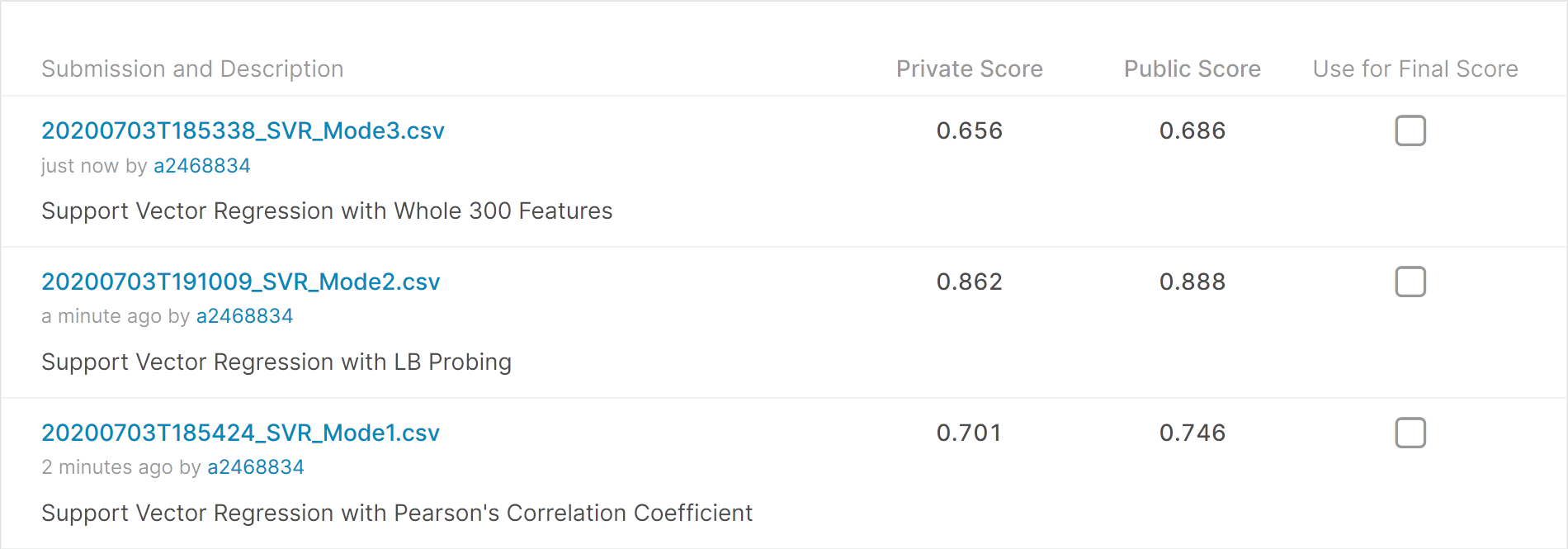
完全不刪除任何features，直接進行模型訓練。

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：若模型本身就能建立複雜的分類策略，則AUCROC分數會很差；若我們使用了太多參數，建立出過複雜的模型 (例如採用高維度polynomial kernel的SVM)，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。

1. 「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。