智慧化企業整合

Final Project

建構最小化借貸風險之模型

109034511 邱綉雅

目錄

摘要

近年來貸款逐漸盛行，無論是個人貸款或是企業貸款的數量都有逐漸上漲的趨勢。然而，無論是借貸公司，抑或是銀行，資金皆有限，面對眾多的借款申請人，該如何評估是否應貸款給該借款申請人，避免貸款有去無回，成為一大難題。因此，本研究希望能透過建構一神經網路模型，來決定是否應貸款給該借款申請人，以最小化貸款風險。

1. 緒論

1.1 研究背景

近年來貸款逐漸盛行，無論是個人貸款或是企業貸款的數量都有逐漸上漲的趨勢。然而，在許多時候，借款人並未妥善評估自身是否有能力能夠償還貸款，以企業而言，可能會因為經營不善而倒閉，無法償還債款；而對借款人而言，也可能因突發狀況，如：疫情原因導致無法償還債款，甚至有部分借款人會利用不停的貸款，來償還先前的貸款，形成惡性循環。

1.2 研究動機與目的

然而，無論是借貸公司，抑或是銀行，資金皆有限，面對眾多的借款申請人，該如何評估是否應貸款給該借款申請人，要如何妥善運用資金，避免貸款有去無回，成為一大難題。因此，本研究希望能透過建構一神經網路模型，來評估該借款申請人之資料，決定是否應貸款給該借款申請人，以最小化貸款風險。

2. 文獻回顧

2.1 類神經網路(Neural Network，NN)

類神經網路在機器學習和認知科學領域，是一種模仿生物神經網路（動物的中樞神經系統，特別是大腦）的結構和功能的數學模型或計算模型，用於對函式進行估計或近似。神經網路由大量的人工神經元聯結進行計算。大多數情況下人工神經網路能在外界資訊的基礎上改變內部結構，是一種自適應系統。現代神經網路是一種非線性統計性資料建模工具，神經網路通常是通過一個基於數學統計學類型的學習方法（Learning Method）得以最佳化

2.2 CatBoost

CatBoost 是俄羅斯的搜尋巨頭 Yandex 在 2017 年開源的機器學習庫，是 Boosting 族演算法的一種。其基於梯度增強，性能優於許多現有的增強算法，如 XGBoost、Light GBM 等。它實現了對稱決策樹（oblivious trees），有助於減少預測時間，並且在分類變量索引方面具有相當的靈活性，能夠用於各種統計上的分類特徵和數值特徵的組合，擅長處理類別資料，能夠將分類值編碼成數字。

3.研究方法

3.1 研究架構

本研究之研究架構共分三個步驟，分別為資料分析、資料前處理和模型建構。

3.2 資料分析

本研究之資料來源為 Kaggle 線上開放式資料庫所提供之全球最大網路借貸平台----Lending Club 之借貸資料，共 396030 筆資料，分為 27 個資料欄位。

根據資料分析結果，可得出：有部分資料屬於類別資料，部分為數值資料，且其中有部分資料欄位有空值。

3.3 資料前處理

本研究所做之資料前處理分為以下幾步驟：

(1) 將是否償還貸款的欄位轉為數值資料，並令償還債款為 1，無法償還債款為 0。

(2) 找出其餘數值資料欄位和是否償還貸款欄位的相關性。

(3) 找出有較多遺漏值之欄位，根據與是否償還貸款欄位的相關性做處理，將相關性高的欄位補值，相關性較低之欄位去除。

(4) 檢視處理後的資料，將仍有遺漏值之資料整筆刪除。

(5) 將類別資料轉換為數值資料。

(6) 分割資料集，將 64%資料集作為訓練資料集，16%資料集作為驗證資料集，20%資料集作為測試資料集。

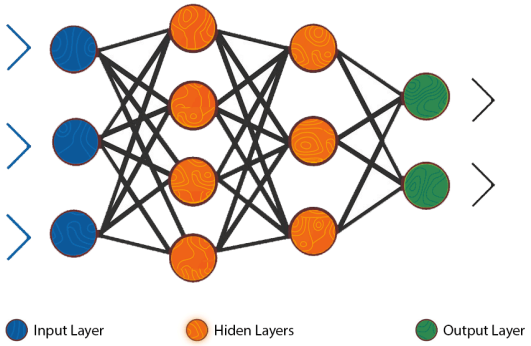
(7) 將資料做正規化處理。

3.4 模型建構

本研究使用了兩種模型，分別為類神經網路和 CatBoost 模型。

3.4.1 類神經網路

本研究所建構之類神經網路模型由一層輸入層、兩層隱藏層和一層輸出層組成，並在輸入層和隱藏層加入隨機關閉神經元，避免過擬合情況發生。



下圖為本研究所建構之類神經網路模型：

```
#build  a  model
model  =  Sequential()

#  input  layer
model.add(Dense(78,activation='relu'))
model.add(Dropout(0.2))

#  hidden  layer
model.add(Dense(39,activation='relu'))
model.add(Dropout(0.2))

#  hidden  layer
model.add(Dense(19,activation='relu'))
model.add(Dropout(0.2))

#  output  layer
model.add(Dense(1,  activation='sigmoid'))

#  compile  model
model.compile(optimizer="adam",  loss='binary_crossentropy',metrics=['accuracy'])
```

3.4.2 CatBoost

下圖為本研究所建構之 CatBoost 模型：

```
from catboost import CatBoostClassifier, Pool
pool_train = Pool(X_train, y_train)
pool_test = Pool(X_test, y_test)
pool_val = Pool(X_val, y_val)
model = CatBoostClassifier(learning_rate=0.03,
                           iterations=1000,
                           early_stopping_rounds=100,
                           verbose=False,
                           random_state=0)
model.fit(pool_train, eval_set=pool_val, plot=True);
```

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix, precision_recall_curve
y_pred_test = model.predict(pool_test)

acc_test = accuracy_score(y_test, y_pred_test)
prec_test = precision_score(y_test, y_pred_test)
rec_test = recall_score(y_test, y_pred_test)
print(f'''Accuracy (test): {acc_test:.3f}
Precision (test): {prec_test:.3f}
Recall (test): {rec_test:.3f}''')

cm = confusion_matrix(y_test, y_pred_test)
ax = sns.heatmap(cm, cmap='viridis_r', annot=True, fmt='d', square=True)
ax.set_xlabel('Predicted')
ax.set_ylabel('True');
```

4. 個案研究

本研究之研究對象為全球最大網路借貸平台----Lending Club。針對該平台之借貸資料，經過資料前處理後，將資料代入模型中，得出以下之結果。

4.1 類神經網路

將個案之借貸資料代入本研究所建構的類神經網路模型中，並進行參數調整，使模型能夠更準確地預測結果。

4.1.1 參數調整

以下為參數調整之過程：

optimizer	結果	選擇
adam	0.888	✓
sgd	0.887	
RMSprop	0.887	
adagrad	0.887	

loss function	結果	選擇
mean_squared_error	0.888	
binary_crossentropy	0.888	✓
squared_hinge	0.845	
logcosh	0.888	

Input+hidden layer activation function	結果	選擇
relu	0.888	✓
tanh	0.888	
exponential	0.198	
linear	0.887	

output layer activation function	結果	選擇
sigmoid	0.888	✓

tanh	0.887
softmax	0.802
relu	0.888

4.1.2 輸出結果

經由參數調整後，最終之參數組合為 optimizer 選擇 adam，loss function 選擇 binary_crossentropy，Input layer 和 hidden layer 之 activation function 選擇 relu，output layer 之 activation function 選擇 sigmoid。

最終可得訓練準確率為 0.891，測試準確率為 0.888。

4.2 CatBoost

將個案之借貸資料代入本研究所建構的 CatBoost 模型後，再進一步做模型改善，使模型能夠更準確地預測結果。

4.2.1 模型改善

因為在原先建構之模型中，致力於最小化偽陰性誤差(被預測為無法償還貸款，但實際能夠償還貸款)和偽陽性(被預測為能夠償還貸款，但實際無法償還貸款)誤差，因此導致預測準確率僅有 88.9%，為提高準確率，因此針對模型做進一步的改善。

由於本研究主要旨在找出「未來會償還貸款之借款人」，若偽陽性誤差過大，將導致借貸風險提高，然而，偽陰性誤差大小僅是損失因借貸所獲得之利益，因此無論偽陰性誤差大小，皆應使模型之偽陽性誤差最小化。

下圖為改善後之 CatBoost 模型：

```
y_proba_val = model.predict_proba(pool_val)[:, 1]
p_val, r_val, t_val = precision_recall_curve(y_val, y_proba_val)
plt.plot(r_val, p_val)
plt.xlabel('Recall')
plt.ylabel('Precision');
```

```
p_max = p_val[p_val != 1].max()
t_all = np.insert(t_val, 0, 0)
t_adj_val = t_all[p_val == p_max]
y_adj_val = (y_proba_val > t_adj_val).astype(int)
p_adj_val = precision_score(y_val, y_adj_val)
print(f'Adjusted precision (validation): {p_adj_val:.3f}')
```

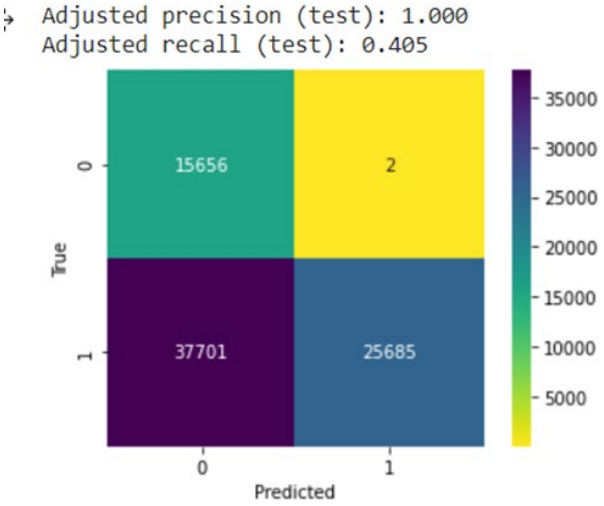
```
y_proba_test = model.predict_proba(pool_test)[:, 1]
y_adj_test = (y_proba_test > t_adj_val).astype(int)
p_adj_test = precision_score(y_test, y_adj_test)
r_adj_test = recall_score(y_test, y_adj_test)
print(f'''Adjusted precision (test): {p_adj_test:.3f}
Adjusted recall (test): {r_adj_test:.3f}''')

cm_test = confusion_matrix(y_test, y_adj_test)
ax = sns.heatmap(cm_test, cmap='viridis_r', annot=True, fmt='d', square=True)
ax.set_xlabel('Predicted')
ax.set_ylabel('True');
```

4.2.2 輸出結果

經由模型改善後，可得到 100%之精度，能夠有效預測結果。

下圖為改善後之 CatBoost 模型預測結果：



5. 結論

針對此資料集而言，在經過兩模型比較後，可以看出 Catboost 在改善前和 NN 模型的準確率差異不大，但在經過模型改善後，因為 Catboost 可選擇僅最小化偽陽性誤差，因此能夠有較好的預測準確率。

未來則可以考慮納入其餘考量因素，如：疫情期間，一些風險高的工作從業者可能較難還出貸款，或是針對企業貸款和個人貸款做分類研究，使預測結果更加精確。

6.參考文獻

1. https://zh.wikipedia.org/wiki/%E4%BA%BA%E5%B7%A5%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C
2. https://kknews.cc/code/ejrk4pr.html
3. https://www.twblogs.net/a/5bd379212b717778ac204cdc
4. https://www.kaggle.com/tomasmantero/minimizing-risks-for-loan-investments-keras-ann#1.-Introduction
5. https://www.kaggle.com/pavlofesenko/minimizing-risks-for-loan-investments/notebook#3.-Modelling-approach
6. https://www.mdeditor.tw/pl/p4Eg/zh-tw