國立清華大學

工業工程與管理研究所

智慧化企業整合期末報告

----學生成績表現預測模型

授課教授：邱銘傳 博士

學生：朱文伃 108034536

一、 動機與目的

教育的重要性除了展現於總體層面，以促進一個國家的經濟發展、政治、文化、社會流動與社會和諧之外，就個體層面而言，教育的意義則在於開發智能並開啟機會與增加發展的可能性。雖然近年來國內教育程度以有大幅提升，但輟學人數或是科目被當人數還是占有一定的比例。因此本報告希望藉由機器學習與深度學習來預測學生的成績分布及表現，並藉此找出影響學生表現的關鍵因素，提前給予學生協助、輔導，甚至改善學校的教學與管理制度，以提升學生們的學習成果。

二、 資料集概述

本報告之資料集取自 UCI Machine Learning Repository 網站，其中包含性別、年齡、父母職業、家中是否有網路等 29 項特徵（見表一至表三）及一項預測結果——數學分數（0-20 分）。另外，資料集 1044 筆資料中，本報告以 80%、20%之比例分為訓練集和測試集(見圖一)。而由於在實驗過程中發現，若將分數分為五個等級進行預測，使用 SVM、MLP、DNN 即 Naïve Bayes 的結果都差強人意，因此本報告將預測項的分類分為以下兩種形式進行預測（見表四）。

1. 二分法：及格（等級 1~4/10-20 分）與不及格(等級 5/10 分以下)
2. 五個等級：等級 A 至等級 F（A 等級為 16-20 分、B 等級為 14-15 分、C 等級為 12-13 分、D 等級為 10-11 分、F 等級則為 0-9 分）

表一、特徵項描述(1)

Attribute	Description (Domain)
sex	student's sex (binary: female or male)
age	student's age (numeric: from 15 to 22)
school	student's school (binary: *Gabriel Pereira* or *Mousinho da Silveira*)
address	student's home address type (binary: urban or rural)
Pstatus	parent's cohabitation status (binary: living together or apart)
Medu	mother's education (numeric: from 0 to 4[a])
Mjob	mother's job (nominal[b])
Fedu	father's education (numeric: from 0 to 4[a])
Fjob	father's job (nominal[b])
guardian	student's guardian (nominal: mother, father or other)
famsize	family size (binary: ≤ 3 or > 3)
famrel	quality of family relationships (numeric: from 1 – very bad to 5 – excellent)
reason	reason to choose this school (nominal: close to home, school reputation, course preference or other)
traveltime	home to school travel time (numeric: 1 – $<$ 15 min., 2 – 15 to 30 min., 3 – 30 min. to 1 hour or 4 – $>$ 1 hour).

表二、特徵項描述(2)

studytime	weekly study time (numeric: 1 – < 2 hours, 2 – 2 to 5 hours, 3 – 5 to 10 hours or 4 – > 10 hours)
failures	number of past class failures (numeric: n if $1 \leq n < 3$, else 4)
schoolsup	extra educational school support (binary: yes or no)
famsup	family educational support (binary: yes or no)
activities	extra-curricular activities (binary: yes or no)
paidclass	extra paid classes (binary: yes or no)
internet	Internet access at home (binary: yes or no)
nursery	attended nursery school (binary: yes or no)
higher	wants to take higher education (binary: yes or no)
romantic	with a romantic relationship (binary: yes or no)
freetime	free time after school (numeric: from 1 – very low to 5 – very high)
goout	going out with friends (numeric: from 1 – very low to 5 – very high)
Walc	weekend alcohol consumption (numeric: from 1 – very low to 5 – very high)
Dalc	workday alcohol consumption (numeric: from 1 – very low to 5 – very high)
health	current health status (numeric: from 1 – very bad to 5 – very good)
absences	number of school absences (numeric: from 0 to 93)
G1	first period grade (numeric: from 0 to 20)

表三、初始資料集

sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime	studytime	failures	schoolsup	famsu
F	18	U	GT3	A	4	4	at_home	teacher	course	mother	2	2	0	yes	no
F	17	U	GT3	T	1	1	at_home	other	course	father	1	2	0	no	yes
F	15	U	LE3	T	1	1	at_home	other	other	mother	1	2	3	yes	no
F	15	U	GT3	T	4	2	health	services	home	mother	1	3	0	no	yes
F	16	U	GT3	T	3	3	other	other	home	father	1	2	0	no	yes
M	16	U	LE3	T	4	3	services	other	reputation	mother	1	2	0	no	yes
M	16	U	LE3	T	2	2	other	other	home	mother	1	2	0	no	no

表四、分類標準

分類法\分數	16-20	14-15	12-13	10-11	0-9
二分法	及格(1)	及格(1)	及格(1)	及格(1)	不及格(0)
五個等級	A(1)	B(2)	C(3)	D(4)	E(5)

```
Number transactions X_train dataset:  (1170, 29)
Number transactions y_train dataset:  (1170,)
Number transactions X_test dataset:  (325, 29)
Number transactions y_test dataset:  (325,)
```

圖一、以 8:2 比例分割訓練集與測試集

三、 資料前處理

(一) 檢查遺漏值

若資料集有資料不完整的情況發生，不僅造成樣本數減少，甚至降低預測的有效性，因此本報告利用 isnull()來檢查資料集是否有遺漏值(見圖二)。輸出結果為 False，代表資料完整。

```
df = pd.read_csv('student.csv')
print(df.head)
print("Any missing sample in training set:",df.isnull().values.any())#看是否有遺漏值
```

```
[1044 rows x 30 columns]>
Any missing sample in training set: False
```

圖二、檢查遺漏值程式碼與結果

(二) 特徵值編碼

由於資料集內容並非全為數值形式，因此利用標籤編碼器(LabelEncoder)將資料集轉換為數字，並用 fit_transform 來對選定特徵進行轉換(見圖三)，使預測模型可以更好地理解這些數據，以便預測進行。

```
df = pd.read_csv("student_initial.csv")
df.columns = ['sex','age','address','famsize',  'Pstatus','Medu','Fedu','Mjob','Fjob','rea
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
df['sex']= labelencoder.fit_transform(df['sex'])
df['address']= labelencoder.fit_transform(df['address'])
df['famsize']= labelencoder.fit_transform(df['famsize'])
```

	sex	age	address	famsize	Pstatus	...	Dalc	Walc	health	absences
0	0	18	1	0	0	...	1	1	3	6
1	0	17	1	0	1	...	1	1	3	4
2	0	15	1	1	1	...	2	3	3	10
3	0	15	1	0	1	...	1	1	5	2
4	0	16	1	0	1	...	1	2	5	4

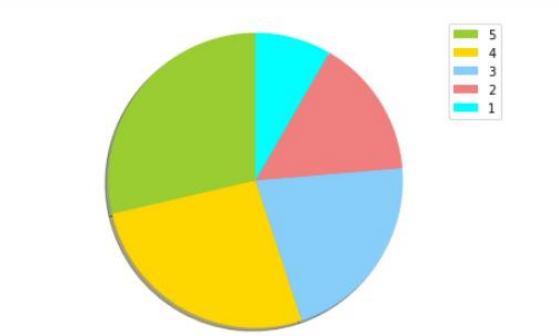
圖三、標籤編碼器程式碼與結果

(三) 過取樣

當解決分類問題時，若資料類別不平衡很可能會造成預測結果的偏差、使其不準確。在檢查訓練集中資料中兩種分類方法分別的類別數量是否有平衡的過程中發現，不論以五個類別或是兩個類別進行分類，皆呈現不平衡狀態(見圖四、圖六)，因此都應該要採取過取樣。

而本報告使用 SMOTE 作為過取樣方法(見圖八)，他改善了隨機過取樣容易過擬合的缺點，對少數類別樣本進行插值來人工合成新樣本至資料集中，使每個類別數量相同。在將訓練集進行過取樣後，五個類別各類別皆有 234 個樣本，兩個類別各類別則皆有 601 個樣本(見圖五、圖七)。

```
Before OverSampling, counts of label '1': [71]
Before OverSampling, counts of label '2': [134]
Before OverSampling, counts of label '3': [183]
Before OverSampling, counts of label '4': [213]
Before OverSampling, counts of label '5': [234]
```

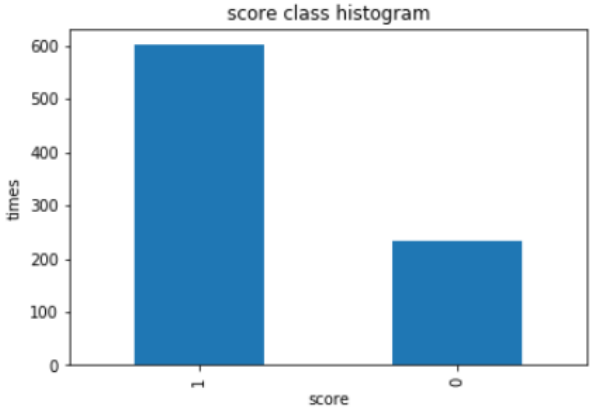


圖四、過取樣前訓練集中各類別樣本數(五個類別)

```
After OverSampling, counts of label '1': 234
After OverSampling, counts of label '2': 234
After OverSampling, counts of label '3': 234
After OverSampling, counts of label '4': 234
After OverSampling, counts of label '5': 234
```

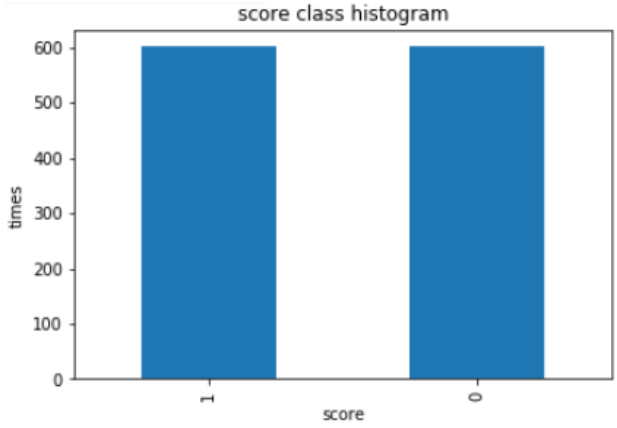
圖五、過取樣後訓練集中各類別樣本數(五個類別)

```
Before OverSampling, counts of label '1': [601]
Before OverSampling, counts of label '0': [234]
```



圖六、過取樣前訓練集中各類別樣本數(兩個類別)

```
After OverSampling, counts of label '1': 601
After OverSampling, counts of label '0': 601
```



圖七、過取樣後訓練集中各類別樣本數(兩個類別)

```
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state=2)
X_train_res, y_train_res = sm.fit_sample(X_train, y_train.ravel())
```

圖八、SMOTE 程式碼

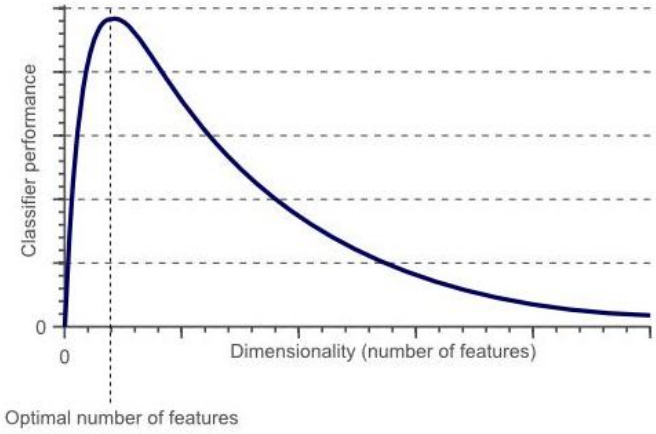
(四) 特徵標準化

因為有些分類器需要計算樣本間的距離，例如 KNN，若特徵值的範圍非常大，距離計算便會取決於此特徵，若範圍小的特徵項較重要的話，得到的結果便會相反，此時就需要進行特徵標準化。特徵標準化是將特徵資料按照比例縮放，使資料落在一特定區間內，除了可以優化梯度下降法外，還可以提高精準度。

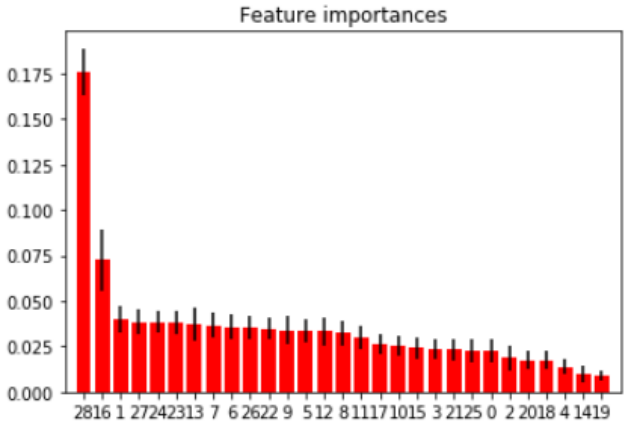
常見的方法包含最小值最大值正規化和 Z 分數標準化。在此報告中，我們採用最小值最大值正規化，將資料等比例縮放置[-1,1]區間中()。

(五) 特徵選取

在機器學習中往往會遇到過擬合及維度災難的問題，當特徵數量增加分類效果會隨之上升，但超過一定值時，其效果反而會下降(見圖九)。因此本報告利用極限隨機樹(ExtRa Trees)進行特徵選取，選擇 29 個特徵中重要分數大於 0.35 者，包含缺席率、課後休閒時間等十項特徵(見圖十)，來降低上述風險，以使預測模型更為精準。



圖九、維度災難



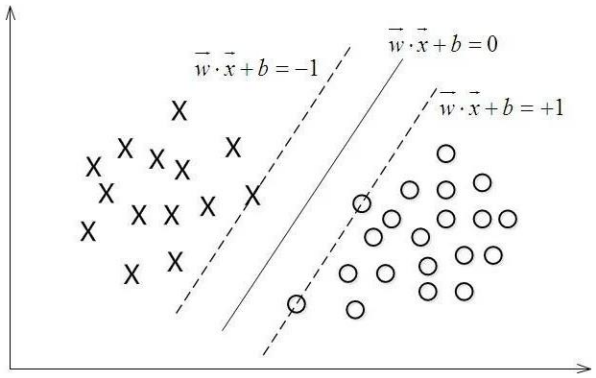
圖十、極限隨機樹特徵選取結果

四、 預測模型

(一)支援向量機 (SVM)

1. 概念介紹

SVM 是一種監督式學習的方法，用統計風險最小化的原則來估計一個分類的超平面，也就是找到一個決策的邊界使類別之間的邊界最大化，使其盡可能遠離此超平面(見圖十一)。如下圖所示，實現部分即為目標之超平面，而 SVM 的目的是為了使兩邊分類能盡可能區隔開來，也就是保證虛線部分的點盡可能遠離即可，其中虛線上的點稱為支援向量。



圖十一、超平面示意圖

2. 參數設定

由於 kernel 有線性(linear)、徑向基函數核(RBF)、多項式(polynomial)等不同類型，無法提前知道選擇何者能使結果最佳化。因此，本報告選擇 RBF 和線性兩種 kernel，並設定多組 gamma 值與正規化參數 C 來進行比較。接著在將資料分為五組的情況下進行交叉驗證，最後選定效能較好者進行訓練即預測，詳細程式碼見圖十二。

```
# 設定多種參數
params_grid = [{'kernel': ['rbf'], 'gamma': [1e-3, 1e-4],
                    'C': [1, 10, 100, 1000]},
                   {'kernel': ['linear'], 'C': [1, 10, 100, 1000]}]
svm_model = GridSearchCV(SVC(), params_grid, cv=5)
svm_model.fit(X_train,y_train)
print('Best score for training data:', svm_model.best_score_,"\n")

# 顯示最佳的參數為何
print('Best C:',svm_model.best_estimator_.C,"\n")
print('Best Kernel:',svm_model.best_estimator_.kernel,"\n")
print('Best Gamma:',svm_model.best_estimator_.gamma,"\n")

final_model = svm_model.best_estimator_
y_pred = final_model.predict(X_test)
print(confusion_matrix(y_test,y_pred))
print("\n")
print(classification_report(y_test,y_pred))

print("Training set score for SVM: %f" % final_model.score(X_train , y_train))
print("Testing  set score for SVM: %f" % final_model.score(X_test, y_test ))

svm_model.score
```

圖十二、SVM 程式碼

3. 預測結果

對於五個類別而言，使用 RBF 作為 kernel、C 為 1000、gamma 為 0.001 的情況下能使準確率達到最高；對於兩個類別而言，使用 RBF 作為 kernel、C 為 100、gamma 為 0.001 的情況下能使準確率達到最高。其預測結果如表五所示。

表五、SVM 預測結果

	五個類別	兩個類別
未正規化	0.374	0.765
正規化	0.387	0.770

從結果可以發現若將預測項分為五個類別進行預測，不論是否將資料正規化，結果都不甚理想；若分為兩個類別進行預測，則能提升近 40%的準確率。

(二)多層感知機 (MLP)

1. 概念介紹

MLP 是一種前向傳遞類神經網路，至少包含三層基本結構——輸入層、隱藏層和輸出層，並且利用倒傳遞的方法達到監督式學習的目標。而除了輸入層之外，每個節點都是一個神經元且必須使用非線性的激活函數來傳遞至下一層。常見的激活函數包括 sigmoid、hyperbolic tangent、ReLU 等，其中 sigmoid 和 hyperbolic tangent 很容易發生梯度消失的問題，是類神經網路加深時主要的訓練障礙。而 ReLU 的分段線性性值能有效克服此問題且其計算量小，只需判斷輸入是否大於零即可，因此本報告選擇 ReLU 作為 MLP 模型的激活函數。

在模型訓練過程中，尋找一個適當的最佳化演算法使損失函數最小化並提升整理表現也是非常重要的。常見的優化器包含 AdaGrad、RMSProp 和 Adam 等，而 Adam 因為能對參數的偏離進行校正，使參數的更新更為平穩，且其所需記憶體容量小，因此在本報告中我們選用 Adam 當作 MLP 的優化器。

2. 參數設定

在同時使損失函數下降與準確率上升的情況下，本報告對於兩種分類方法、正規化及非正規化的資料集設定了不同的神經元個數及隱藏層層數。以圖十三為例，hidden_layer_sizes=(21,50)代表第一層隱藏層及第二層隱藏層神經元個數分別為 21 和 50，在此參數設定下，損失函數會降低為 0.29。

```
clf = MLPClassifier(hidden_layer_sizes=(21,50), max_iter=1000,activation = 'relu',solver='adam',random_state=1)
clf.fit(X_train, y_train)
print (clf.n_layers_)
print("Accuracy of MLPClassifier : ", clf.score(X_test,y_test))
```

圖十三、MLP 程式碼

3. 預測結果

以 MLP 模型訓練並預測的結果如表六所示。我們可以發現其結果和以 SVM 模型預測結果相似，以兩個類別進行預測的準確率能比以五個類別進行預測的結果高出近 40%。

表六、MLP 預測結果

	五個類別	兩個類別
未正規化	0.369	0.770
正規化	0.392	0.728

（三）深度神經網路 (DNN)

1. 基本介紹

DNN 可以理解為有很多隱藏層的神經網路，前述的 MLP 即為其一特例。而 DNN 很容易產生過擬合的情況，此時便可以使用丟棄法(dropout)作為正規化方法，也就是在訓練過程中隨機丟棄移部分隱藏層的神經元來降低過擬合狀況。

本報告中建構的 DNN 共有三層隱藏層，基於前一段所述多個激活函數和優化器之比較，我們選擇之激活函數和優化器與 MLP 相同。和 MLP 不同的是，DNN 必須設定每次訓練樣本數(batch size)、訓練次數(epoch)，並且設定 10%的樣本作為驗證集。

```
model = Sequential()
model.add(Dense(70, input_dim=29, activation='relu'))
model.add(Dense(30, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(10, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
print(model.summary())
```

```
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
train_history=model.fit(X_train_res, y_train_res,batch_size=100,
                        epochs=30,verbose=2,
                        validation_split=0.1)
scores = model.evaluate(X_test, y_test, verbose=1)
print("Accuracy:",scores[1])
```

圖十四、DNN 程式碼

2. 參數設定

經過多次的測試，在使準確率達到最高的條件下，發現將單次訓練樣本數設為 100，訓練次數設為 30，且取 10%樣本作為驗證集可以使表現最佳。

3. 預測結果

以 DNN 模型訓練之結果如表七所示。我們可以發現預測項分為兩個類別的預測結果會比五個類別的結果高出 30%，且是否將資料正規化對準確率並沒有顯著影響。

表七、DNN 預測結果

	五個類別	兩個類別
未正規化	0.421	0.751
正規化	0.416	0.732

(四)單純貝氏分類器 (Naïve Bayes)

1. 概念介紹

在機器學習中，貝氏分類法(Bayesian Classifier)是藉由資料中分析屬性與反應變數間的機率模型，根據貝氏定理（Bayes' theorem）來更新資訊以判斷樣本資料歸屬類別，作為分類和推論的依據。

單純貝氏是一種構建分類器的簡單方法。該分類器模型會給問題實體分配用特徵值表示的類標籤，類標籤取自有限集合。對於某些類型的機率模型，在監督式學習的資料集中能取得得非常好的分類效果。

```
#建立模型並且訓練
model = GaussianNB()
model.fit(X_train, y_train)
#預測結果
y_pred = model.predict(X_test)
```

圖十五、單純貝氏分類器程式碼

2. 預測結果

透過 Naïve Bayes 模型進行訓練與預測的結果如表八所示。我們可以對於兩個類別的預測效果和前三個模型差不多，都在 75%以上，而對於五個類別的預測效果極差，準確率低於 30%。

表八、預測結果

	五個類別	兩個類別
未正規化	0.258	0.766
正規化	0.254	0.765

(五)結果比較

前述四個模型之準確率比較如表九所示。我們可以發現不論資料是否經過正規化，對準確率都沒有顯著的影響。而對於五個類別的分類方法，四個模型的表現都不甚理想，皆低於 50%，其中以 Naïve Bayes 模型效果最差；對於兩個類別的分類方法而言，四個模型沒有太大差異，準確率都在 75%左右。

表九、四個模型之準確率

		SVM	MLP	DNN	Naïve Bayes
五個類別	沒正規化	0.374	0.369	0.421	0.258
	正規化	0.387	0.392	0.416	0.254
兩個類別	沒正規化	0.765	0.770	0.751	0.766
	正規化	0.770	0.728	0.732	0.765

五、 結論與未來展望

本報告在一開始對於資料集進行特徵選取、特徵值編碼、過取樣及特徵標準化四種資料前處理，以助模型準確率的提升。而經過 SVM、MLP、DNN、Naïve Bayes 四種模型的訓練及預測，可以發現將預測項分為兩個類別進行模型訓練與預測會比分為五個類別表現好許多，從準確率皆低於 50%提升至 75%左右，其

中又以 Naïve Bayes 提升最多，提升 50%左右的準確率。而造成五個類別的準確率極低的原因可能包含樣本數不夠，因此資訊不夠充足、各樣本點太相近以致無法成功進行分類等。

而在建構模型過程中，雖然都是引入套件進行訓練，但因為 MLP 及 DNN 兩模型需要不斷地進行參數調整，才能達到可接受的結果，且沒有標準方法可以決定最佳隱藏層的層數和神經元個數，因此本報告最佳準確率依舊不到 80%可能是參數設定非最佳化的原因。而從特徵選取中的特徵重要度也可以看出，即使排名第一的特徵項其重要程度也只有 0.175 左右，其餘都低於 0.075，因此特徵項與預測結果關聯度不高也是導致準確率不高的原因之一。有鑑於此，以後在做類似預測模型時，應更謹慎的蒐集資料，使資料集夠完整且彼此相關性夠高，才能得到完美的結果。