EEML HW2 Report

會計五 B06702064 林聖硯

1. (1%) 請比較說明generative model、logistic regression兩者的異同為何？再分別列出本次使用的資料中五個分得正確/不正確的sample，並說明為什麼如此？

Notice
本題的丟入兩個模型的資料，數值型(nuemrical)的資料只有經過**normalization**，類別型(categorical)的資料只有經過**one-hot encoding**進行處理，沒有透過其他方式減少資料維度。(因為後期再反覆嘗試的結果之後，發現**有別的模型一定會dominate這兩個模型**，所以就沒有在這兩個模型上有太多的調整)

Ans:
(1) Generative model背後的概念與貝氏統計理論相關，我們要先對每一個class猜測一個事前的機率(prior，或稱先驗機率)，而資料點的產生過程在統計上稱為likelihood，貝氏定理告訴我們將prior乘上likelihood，就能得到事後機率(posterior probability，或稱後驗機率)，並得到預測結果。但實作後發現generative model在此作業的成效不好，在(3)說明原因。

```
GM = GenerativeModel()
GM.train(X_train, y_train, idx_class1, idx_class2)
y_pred = GM.test(X_test)
✓ 0.1s
Training accuracy = 24.081%
```

(2) logistic regression是一個linear model，並且透過名為**logit**的link function來讓原本在$(-\infty, +\infty)$的輸出值，映射(mapping)到$(0, 1)$之間，並以**0.5**作為binary classification的分界點，來進行分類的任務。而在本次作業中，我實作了**adagrad**的optimizer，透過gradient descent的方式來找到最適合這個資料的parameters。實作後發現，與generative model比較，logistic regression的表現明顯比較好，在(3)說明原因。

```
print("=" * 10, " Best Model result ", "=" * 10)
print(f"Batch Size = {best_batch_size}, Epoch Size = {best_epoch_size} (Actually running {best_
print(f"Validation loss = {round(global_best_loss, 3)} (Accuracy: {round(global_best_acc*100, 3
                                                                                        Python
==========  Best Model result  ==========
Batch Size = 512, Epoch Size = 100 (Actually running 24 epoch), Learning rate = 0.1
Validation loss = 3.827 (Accuracy: 86.15%)
```

(3) 分別列出本次使用的資料中五個分得正確/不正確的sample，並說明為什麼如此？

generative model

我認為generative model分類不好的原因主要與one-hot encoding造成維度太大，以及matrix太sparse有關。在計算μ以及Σ時，我認為one-hot encoding後類別變數的平均值其實沒有什麼意義，但在計算covariance matrix時需要透過他的幫助，所以會影響到covariance matrix裡面許多共變異數都沒有什麼意義。但在預測時，我們卻會透過這樣子產生的covariance matrix計算$P(C_1|x)$以及$P(C_2|x)$。

generative model正確的前五個點

========== Classify correctly ==========

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_per_week	native_country	income
7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	45	United-States	1
8	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Female	14084	0	50	United-States	1
9	42	Private	159449	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	5178	0	40	United-States	1
10	37	Private	280464	Some-college	10	Married-civ-spouse	Exec-managerial	Husband	Black	Male	0	0	80	United-States	1
11	30	State-gov	141297	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	Asian-Pac-Islander	Male	0	0	40	India	1

generative model分錯的前五個點

========== Classify wrong ==========

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_per_week	native_country	income
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	0
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	0
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	0
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	0
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	0

logistic regression model

logistic regression分類正確的點，native_country幾乎都是United-States；而分類錯誤的點大多數的marital_staus幾乎都是Married-civ-spouse，且fnlwgt普遍較高，很有可能是模型沒有學到這兩個變數之間的交互作用，導致分類錯誤，或在切training與validation set時沒有做好stratified，導致training與validation的class非常的不平均。

logistic regression model分類正確的前五個點

========== Classify correctly ==========

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_per_week	native_country	income
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	0
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	0
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	0
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	0
6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Black	Female	0	0	16	Jamaica	0

logistic regression model分類錯誤的前五個點

========== Classify wrong ==========

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_per_week	native_country	income
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	0
5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40	United-States	0
7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	45	United-States	1
11	30	State-gov	141297	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	Asian-Pac-Islander	Male	0	0	40	India	1
14	40	Private	121772	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	Asian-Pac-Islander	Male	0	0	40	?	1

2. (1%) 請實作兩種feature scaling的方法 (feature normalization, feature standardization)，並說明哪種方法適合用在本次作業？

以下以分類結果較佳的**logistic regression**作為模型參考，分別測試feature normalization/features standardization在validation以及public/private test上的結果。

資料處理	validation acc	public acc	private acc
normalization (標準化，z-score transformation)	86.334%	0.85356	0.85075
standardization (min-max scaling)	86.078%	0.85222	0.85257

我認為feature standardization比較適合。第一，雖然以上結果看起來差不多，但其實可以看的出來normalization相比standardization稍微**overfit**了一點。第二，因為其實此次dataset中很多feature的分布都非常的skewed，且有許多outliers。因此，**這些outliers會拉大feature normalization的μ以及σ**；做normalizaion也會使得資料點分布轉變為常態分佈，破壞feature原本的分布，反而沒辦法讓模型學出好的結果；而feature standardization能夠使features在轉換後的分布維持，並且將其範圍縮小至[0, 1]之間，模型學習的結果會比較好。

3. (1%) 請說明你實作的best model及其背後「原理」為何？你覺得這次作業的dataset比較適合哪個model？為什麼？

我最後使用的是**Random forest**，但在data preprocessing我有特別調整讓model分類的結果提升。

Data Preprocessing

- Numerical features: robust scaling
- Ordered categorical features: NaN
- Categorical features: Target Encoding

Model: Random forest

- Tuning: n_estimators, max_depth, min_samples_split, min_samples_leaf, oob_score = True
- 5-Fold Cross Validation

在Data preprocessing的部分，我發現其實 `education_num` 和 `education` 呈現的是**一對一**的關係，而 `education` 其實可以視為是有序型的類別變數(ordered categorical variables)，所以他不應該被scaling。另外，在數值型的變數上，由於dataset中很多feature的分布都非常的skewed，我選擇的scaling方法為**robust scaling**，公式如下：

$$\bar{x} = \frac{X - median(X)}{\text{75th quantile(x)} - \text{25th quantile(x)}}$$

robust scaling能夠很好的去除min-max scaling(standardization)以及normalization因為outliers而造成對全距、平均以及變異數的影響。

而在類別變數的部分，我則選擇使用target encoding的方式來對類別變數做encoding。其原理如下為，**計算某一個features之下的某一個類別，對應到y出現的機率**。

	Animal	Target
0	cat	1
1	hamster	0
2	cat	0
3	cat	1
4	dog	1
5	hamster	1
6	cat	0
7	dog	1
8	cat	0
9	dog	0

	Animal	Target	Encoded Animal
0	cat	1	0.40
1	hamster	0	0.50
2	cat	0	0.40
3	cat	1	0.40
4	dog	1	0.67
5	hamster	1	0.50
6	cat	0	0.40
7	dog	1	0.67
8	cat	0	0.40
9	dog	0	0.67

我認為類別變數在此問題中是影響模型表現的最大關鍵，但one-hot encoding會使data變得**過於sparse**，容易使得模型表現不好，故我假設每個類別對應到的label出現的機率與結果預測有關，所以才選擇target encoding。

模型的部分我選擇的是**Random forest**，他是一種ensembling learning中的**bagging模型**。演算法每次會從樣本中抽樣(bootstrap)，並從features裡面選出$p=\sqrt{m}$個features出來訓練decision tree，重複上述的結果多次(次數為可調整的參數)，在最後透過majority vote的方式決定分類別的結果。Bagging的優點在於原始訓練樣本中有噪聲資料(不好的資料)，透過Bagging抽樣就有機會不讓有噪聲資料被訓練到，所以可以降低模型的不穩定性。

我其實沒有特別認為這次作業的dataset比較適合任何一個model，model的選擇很大一部分也與data preprocessing有關。

我其實認為本次任務比較大的問題是features的dimension過大，即使我在嘗試透過EDA將一些不重要的類別變數去掉後，logistic regression的Accuracy仍然無法突破85.5%。後來我是在改用KNN之後，模型的準確度才又更提升的一個層級，來到了86%左右(但對categorical仍然做的是one-hot)。後來，將one-hot encoding後換成了target encoding後，模型也改成random forest後，accuracy才又提升了0.5%。

如果要說哪個datset比較適合哪個模型比較適合的話，在try-and-error後，我認為one-hot encoding+KNN與target encoding+random forest都是不錯的組合。前者的原因可能是，KNN的原理就是抓出與training data分布相近的點，而one-hot encoding的做法使得類別變數不是0就是1(兩點的距離很大)，所以對KNN來說能夠有效的預測這樣子的資料集。而後者的原因為，對decision tree這樣子一刀一刀切的演算法，對類別變數做這樣的mapping比較有空間能夠讓演算法去找到一個boundary來切出兩塊區域。(相較於做one-hot encoding，非0即1會很容易使得decision tree不知道怎麼切)

KNN+one-hot

submission_1025_3.csv 4 days ago by b06702064_Martin robust scaling, ont-hot encoding / 5 num + 16 cat + 1 order cat / KNN + tuning / training with all data finally / hard labeling in special case	0.86107	0.86154	☐

Random forest + target encoding

submission_1028_2.csv 2 days ago by b06702064_Martin robust scaling, target encoding with sum / random forest / training with all data finally / hard labeling	0.86402	0.86547	☑

Math Problem

(這次作業來不及用latex打，抱歉QQ)

1. prior: $p(C_k) = \pi_k$.

likelihood function: $p(X_i \mid C_{X_i})$, $i = 1, 2, \cdots, N$.

C_{X_i}: the class that X_i belongs to.

In order to find MLE of "prior probability", we need to use MAP (Maximum A Posterior) here.

$\mathcal{L} = P(t \mid X)$, where X is data matrix and t is the corresponding label vector, $t = [t_{X_1}^T, t_{X_2}^T, \cdots, t_{X_N}^T]$

$= P(C \mid X)$, where $C = [C_{X_1}, C_{X_2}, \cdots C_{X_N}]$ (for simplicity and reading).

$= \dfrac{P(X \mid C) P(C)}{P(X)}$ (Bayes' rule)

$\propto P(X \mid C) P(C)$ (since $P(X)$ is const.)

$= \prod_{i=1}^{N} P(C_{X_i}) \cdot P(X_i \mid C_{X_i})$ (independence assumption.)

$$\log \mathcal{L} = \sum_{i=1}^{N} \log P(C_{X_i}) + \sum_{i=1}^{N} \log P(X_i \mid C_{X_i})$$

$$= \sum_{k=1}^{K} N_k \log P(C_k) + \sum_{i=1}^{N} \log P(X_i \mid C_{X_i}),$$ where N_k is the number of data points in C_k.

$$\underset{\pi_1, \cdots, \pi_K}{\operatorname{argmax}} \log \mathcal{L} = \underset{\pi_1, \cdots, \pi_K}{\operatorname{argmax}} \sum_{k=1}^{K} N_k \underbrace{\log P(C_k)}_{\pi_k} + \sum_{i=1}^{N} \log P(X_i \mid C_{X_i})$$

since $\sum_{k=1}^{K} \pi_k = 1$, let $g = \sum_{k=1}^{K} \pi_k - 1$

$$\mathcal{L}ar(\pi_1, \cdots, \pi_K, \lambda) = \log \mathcal{L} + \lambda g.$$

$$\frac{\partial \mathcal{L}ar}{\partial \pi_k} = \frac{N_k}{\pi_k} + \lambda = 0 \Rightarrow \pi_k = -\frac{N_k}{\lambda}$$

$$\frac{\partial \mathcal{L}ar}{\partial \lambda} = \sum_{k=1}^{K} \pi_k - 1 = 0$$

$$\Rightarrow \sum_{k=1}^{K} -\frac{N_k}{\lambda} - 1 = 0$$

$$\Rightarrow -\frac{N}{\lambda} - 1 = 0$$

$$\Rightarrow \lambda = -N, \quad \pi_k = \frac{N_k}{N} \#.$$

2.

$$\frac{\partial \log(\det \Sigma)}{\partial \sigma_{ij}} = \frac{1}{\det \Sigma}\,\frac{\partial \det\Sigma}{\partial \sigma_{ij}}$$

(cofactor formula) $= \dfrac{1}{\det\Sigma}\,\dfrac{\partial}{\partial \sigma_{ij}}\left[\sum_{k=1}^{n} \sigma_{ik} C_{ik}\right]$, where C_{ij} is the cofactor at ith row and jth column.

$$= \frac{1}{\det\Sigma}\cdot C_{ij}$$

$= \dfrac{1}{\det\Sigma}\left[adj(\Sigma)\right]_{ji}$ ($\because C^T = adj(A)$, where $adj(A)$ is the classical adjoint matrix of A)

$$= \left[\Sigma^{-1}\right]_{ji}$$

$$= e_j \Sigma^{-1} e_i^T \quad \#$$

3.

from problem 1.

$$\log \mathcal{L} = \sum_{k=1}^{N} N_k \log \pi_k + \sum_{i=1}^{N} \log P(X_i \mid C_{x_i})$$

$$= \sum_{k=1}^{N} N_k \log \pi_k + \sum_{k=1}^{K}\sum_{i=1}^{N} t_{ik} \log P(X_i \mid C_k)$$

$$P(X \mid C_k) = N(X \mid u_k, \Sigma)$$

$$= \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} e^{-\frac{1}{2}(x-u_k)^T \Sigma^{-1}(X-u_k)}$$

$$\log P(X \mid C_k) = -\frac{1}{2}(X-u_k)^T \Sigma^{-1}(X-u_k) - \frac{1}{2}\log(|\Sigma|) - \frac{N}{2}\log 2\pi$$

① 求 u_k (from MLE)

$$\frac{\partial \log \mathcal{L}}{\partial u_k} = \frac{\partial}{\partial u_k}\left[\sum_{k=1}^{K}\sum_{i=1}^{N} t_{ik} \log P(X_i \mid C_k)\right]$$

$$= \frac{\partial}{\partial u_k}\left[\sum_{k=1}^{K}\sum_{i=1}^{N} t_{ik}\left(-\frac{1}{2}\underbrace{(X_i-u_k)^T \Sigma^{-1}(X_i-u_k)}_{X_i^T X_i - u_k^T X_i - X_i^T u_k + u_k^T u_k.} - \cancel{\frac{1}{2}\log(|\Sigma|)} - \cancel{\frac{N}{2}\log 2\pi}\right]\right.$$

$$= \Sigma^{-1}\left(-\frac{1}{2}\sum_{i=1}^{N} t_{ik}(2u_k - 2X_i)\right)$$

$$= \Sigma^{-1}\sum_{i=1}^{N} t_{ik}(X_i - u_k)$$

$$= \Sigma^{-1}\left(\sum_{i=1}^{N} t_{ik}X_i - N_k u_k\right) \stackrel{\wedge}{=} 0$$

$$\Rightarrow u_k = \frac{1}{N_k}\sum_{i=1}^{N} t_{ik} X_i.$$

② 求 Σ.

$$\frac{\partial \log \mathcal{L}}{\partial \Sigma^{-1}} = \frac{\partial \log \mathcal{L}}{\partial \Sigma^{-1}}\left[\sum_{k=1}^{K}\sum_{i=1}^{N} t_{ik}\left(-\frac{1}{2}(X_i-u_k)^T\Sigma^{-1}(X_i-u_k) \overbrace{-\frac{1}{2}\log(|\Sigma|)}^{+\frac{1}{2}\log(|\Sigma^{-1}|) \,=\, -\frac{1}{2}\log\left(\frac{1}{|\Sigma^{-1}|}\right) \,=}\right)\right].$$

$$\left(\frac{\partial}{\partial A}X^T A X = XX^T\right) = \sum_{k=1}^{K}\sum_{i=1}^{N} t_{ik}\left(-\frac{1}{2}(X_i-u_k)(X_i-u_k)^T + \frac{1}{2}\Sigma\right) = 0$$

$$\Rightarrow \underbrace{\sum_{k=1}^{K}\sum_{i=1}^{N} t_{ik}}_{N}\cdot\Sigma = \sum_{k=1}^{K}\sum_{i=1}^{N} t_{ik}(X_i-u_k)(X_i-u_k)^T$$

$$\Rightarrow \Sigma = \sum_{k=1}^{K}\frac{1}{N}\sum_{i=1}^{N} t_{ik}(X_i-u_k)(X_i-u_k)^T$$

$$= \sum_{k=1}^{K}\frac{N_k}{N}\underbrace{\left[\frac{1}{N_k}\sum_{i=1}^{N} t_{ik}(X_i-u_k)(X_i-u_k)^T\right]}_{S_k} \quad \#.$$