Available online at www.sciencedirect.com
SciVerse ScienceDirect
Data Science and Machine Learning 00 (2020) 000–000

Procedia
Economics and Finance

www.elsevier.com/locate/procedia

Data Science and Machine Learning

Dealing with Imbalanced Datasets

陳明萱 [a], 陳瑄 [b]

[a] 國立中央大學資訊管理學系 108423016 碩士班 asc25860@gmail.com
[b] 國立中央大學資訊管理學系 108423019 碩士班 hsuanc@g.ncu.edu.tw

Abstract

資料前處理在機器學習領域中相當重要，好的資料前處理將有助於後續的模型訓練，而不平衡資料集的問題更是其中的一大研究方向。本文將深入研究不平衡資料集，首先介紹不平衡資料集的定義，並以一個實際資料集做說明，接著再闡述四種有效用來處理不平衡資料集的方法，最後再透過兩篇論文研究來探討不平衡資料集對於不同演算法的影響程度。

Keywords: imbalanced dataset, sampling, ensemble, cost-sensitive method, evaluation metric

1. What is an imbalanced dataset?

通常在分類問題上，訓練資料集中不同 Label 的資料數量相差太多或是分布不均而造成類別不平衡，這時就可以將其稱為不平衡資料集 (Imbalanced Dataset)。真實世界的資料多少都有類別不平衡的問題，一般資料類別比例在 4:6 上算是較輕微的類別不平衡 (Slight Imbalance)，比較嚴重的類別不平衡 (Severe Imbalance) 類別比例則可能會超過 1:100，例如醫療上的癌症診斷，通常只有少數 1%的人是被診斷出有得癌症的，如果將這份訓練資料丟入模型訓練，模型最終可能將原本診斷出有癌症的人都預測為不會得癌症，所預測的正確率雖然可以高達 99%，但是實際上關心的目標是「會得癌症」的人，因此導致此模型訓練效果不佳也較不可靠。

2. Find ONE imbalanced dataset and give a brief intro/description on it

2.1. Overview

資料來源是 Kaggle 網站公開的資料集 ([Real or Fake] Fake JobPosting Prediction n.d.)，目的是要判斷工作招募廣告資訊的真偽 (Real/Fake Job Posting)，更原始的資料是來自於希臘的愛琴大學，從 2012 到 2014 年收集的真實資料，總共 17,880 筆。

2.2. Attributes and Label

title	location	department	salary_rang	company_p	description	requirement	benefits	telecommu	has_compa	has_questi	employmer	required_ex	required_ed	industry	function	fraudulent
Marketing Intern	US, NY, N	Marketing		We're Food	Food52, a	Experience	with conte	0	1	0	Other	Internship			Marketing	0
Customer Service -	NZ, , Auck	Success		90 Second	Organised	What we ex	What you v	0	1	0	Full-time	Not Applicable		Marketing	Customer S	0
Commissioning Ma	US, IA, Wever			Valor Servi	Our client,	Implement	pre-commis	0	1	0						0
Account Executive	US, DC, W	Sales		Our passio	THE COM	EDUCATI	Our culture	0	1	0	Full-time	Mid-Senior	Bachelor's	Computer S	Sales	0
Bill Review Manag	US, FL, Fort Worth			SpotSource	JOB TITLE	QUALIFIC	Full Benefi	0	1	1	Full-time	Mid-Senior	Bachelor's	Hospital &	Health Car	0

Fig. 1. 資料範例

Table 1. Attributes and Label ([Real or Fake] Fake JobPosting Prediction n.d.)

Column	Type	Description
job_id (Unique Job ID)	ID	編號
title	Text	招募資訊的標題
location	Text	招募資訊的地點
department	Text	公司部門
salary_range	Text	薪資範圍
company_profile	Text	公司簡介
description	Text	招募的詳細說明
requirements	Text	職位要求
benefits	Text	員工福利
telecommuting	Binary	是否為遠端工作
has_company_logo	Binary	是否有公司的 logo
has_questions	Binary	是否有 screening questions
employment_type	Nominal	工作類型工作類型
required_experience	Nominal	經驗要求
industry	Nominal	行業
function	Nominal	工作職責
required_education	Nominal	學歷要求
fraudulent (1 為假、0 為真)	Binary	是否為假的招募資訊

2.3. Ratio of Imbalanced data

在 17,880 筆資料中，有 17,014 筆 Label 為真實的工作招募資訊，有 866 筆被標記成假的，其大類別與小類別的比例大約為 19:1，有相當懸殊的類別不平衡。

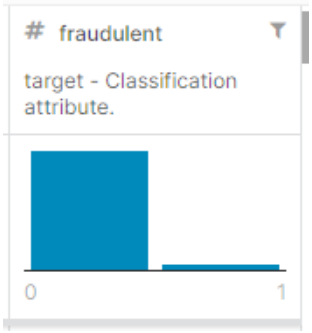

Fig. 2. Label 資料比例圖 ([Real or Fake] Fake JobPosting Prediction n.d.)

3. How to handle/deal with an imbalanced dataset?

在處理類別不平衡問題上，可以根據處理目標的策略區分為以下不同的做法，針對資料層級 (Data-Level) 去做處理的 Resampling Techniques、針對演算法層級 (Algorithm-Level) 去改善的做法、Cost-sensitive method 以及查看不同的衡量指標 (Evaluation Metric)，以下為各方法的詳細介紹。

3.1. Data-Level

資料層級上，分成 Under-sampling 以及 Over-sampling，目的是希望能夠在丟入模型訓練前，針對資料的類別比例做平衡，讓模型避免忽略小類別的資料，以取得更佳的模型訓練效果。

3.1.1. Under-sampling

Under-sampling 是去減少大類 (Majority) 的資料，或是可以想成從大類的資料中抽取部分的資料讓各類別資料的比例達到平衡。

- **Random Under-sampling**
  Random select 是最直接的方法，隨機地去減少大類的資料，但是這有可能導致失去許多原本對分類器是很有代表性或是重要的資料。因此也衍伸出其他常見的方法，包含 Cluster Centroids 以及 Tomek links。

- **Cluster Centroids**
  Cluster Centroids 是針對大類的資料使用 Clustering 的方式找到每一群的中心點作為新的資料並與小類 (Minority) 資料的 Label 比例達到平衡，例如大類有 200 筆資料而小類只有 10 筆資料，那就可以將大類的 200 筆找中心點後分成 10 群，根據最終 10 群裡最具代表的中心點作為原本大類的資料。跟 Random 方法相較之下比較能夠減少重要資料的遺失。

- **Tomek Links**
  Tomek Links 則是透過在分出兩類的邊界上建立一對一對不同類別的資料，將一對一對中大類的資料移除掉，直到所有距離最近的資料對都屬於同一個類別，目的是要讓不同的類別能夠好好區分開來，也移除掉了資料上的 Noise 及不必要的重疊 (Overlapping)。

3.1.2. Over-sampling

Over-sampling 的概念是將小類的資料增加，以讓各類別比例達到平衡。

- **Random Over-sampling**
  最基本的方法也是使用隨機的方式抽出小類的資料並複製多份添加到訓練集中。雖然不會像 Under-sampling 的方法一樣遺失掉許多重要的資料，但有可能會因為資料都是同一重複的小類資料而造成 Overfitting (過度擬合)，雖然可能可以把模型訓練得很好，但是當測試集有一個沒看過的資料出現的時候，分類器沒辦法有個很好的預測結果。因此也衍伸出其他方法如 SMOTE。

- **SMOTE (Synthetic Minority Oversampling Technique)**
  SMOTE 方法是隨機選擇某一資料與鄰近的 k 個點，在該資料點與其鄰近點之間生成新的資料，使兩類資料類別比例達到平衡，但是如果小類的資料含有在大類附近的 Noise data，在產生的新資料後可能會造成資料重疊的問題，導致分類更困難。

3.2. Algorithm-Level

除了在資料層級上的處理外，演算法層級上也有許多處理方式。演算法層級主要是透過 Ensemble 的方法，綜合多次分類器的預測來得到最終最佳的預測結果，本文以兩種不同的 Ensemble 方法來介紹，分別是 Bagging-based 以及 Boosting-based。

3.2.1. Ensemble

- **Bagging-based algorithm**
  Bagging 的概念是先對訓練資料隨機以取後放回 (Boostrap) 的方式生成幾份的資料集，再分別分類器中訓練，如 Decision Tree 或 SVM (Support Vector Machine)等等，最終他們會以 Voting (Majority vote[1])來當作最終預測結果，Random Forest 即是以此方式發展的一種模型。他的優點在於使用了隨機的方式生成資料集，能夠增加資料集的多樣性，讓模型在不同資料集訓練出來的結果都有點差異以防止 Overfitting 的問題。

- **Boosting-based algorithm**
  Boosting 的概念是將訓練資料丟入分類器，接著依據分類器的預測結果，針對分類錯誤的資料給予較高的權重，對於分類正確的資料則給予較低的權重，並重新丟回分類器做訓練。反覆以上步驟，使得下次訓練時分類器可以針對這些權重較高的資料去學習以獲得更好的預測效果。較常見的方法為 AdaBoost，在每一

[1] Majority vote: 是指多個分類器的預測的類別結果，出現次數最多的類別即代表此一資料的最終預測結果。

個 Iteration，都會更新權重，接著將更新完的資料用來訓練分類器，再根據訓練結果以上述方式不斷改善分類器的效能，以取得更佳的預測結果。

3.2.2. Hybrid

像是(Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou 2009)論文提到的 Easyensemble，Easyensemble 是將 Random Under-sampling 以及 Adaboost 做結合，每一次將大類的資料隨機 Sample 至和小類一樣的數量，再將這些資料以 Adaboost 的方法去做訓練，並取得最終的訓練結果，可以想像成是「ensemble of ensembles」。

3.3. Cost-sensitive method

Cost-sensitive method 主要是透過加權的方式，對不同類別根據其分類結果給定不同的權重，一般會透過成本矩陣 (Cost Matrix) 來計算各類別的權重，通常會與其他演算法合併使用，如：結合 SVM、Random Forest 或 Boosting。雖然可以利用權重來處理不平衡資料集的問題，但計算各類別的權重會增加計算成本。

3.4. Evaluation metrics for imbalanced datasets

要判斷一個模型的好壞，需要透過衡量指標來量化成效做評斷，若是選到不能有效評估模型預測結果的衡量指標，可能會導致無法達到預期目標，如 section 1 的癌症診斷案例是使用正確率來做判別，儘管正確率極高，但該模型依然無法有效預測出有罹患癌症的病患。尤其在不平衡資料的分類問題上，某些衡量指標可能會失去判別性。在二元分類或是多元分類問題中，通常會生成 Confusion Matrix (混淆矩陣)，再透過其相關的延伸指標來做衡量。

3.4.1. Confusion Matrix

Confusion Matrix 的行表示模型的預測類別 (Predicted Class)，列則代表資料的實際類別 (Actual Class)，兩者皆可再細分為 Positive Class 和 Negative Class。當實際類別與預測類別相同時就稱為 True，否則為 False，故 Confusion Matrix 內的元素可分成 True Positive (TP)、False Positive (FP)、False Negative (FN)與 True Negative (TN)，各矩陣元素的意涵如下：

- *TP：實際類別為 Positive 也被正確預測為 Positive，表示模型預測正確*
- *FP：實際類別為 Positive 但卻被預測成 Negative，表示模型預測錯誤*
- *FN：實際類別為 Negative 但卻被預測成 Positive，表示模型預測錯誤*
- *TN：實際類別為 Negative 也被正確預測為 Negative，表示模型預測正確*

依據 Confusion Matrix，又可延伸出三種衡量指標，即 Threshold Metrics、Ranking Metrics 與 Probabilistic Metrics。其中，分類問題中常使用前兩種類型，因此以下針對這兩種做介紹。

		Actual Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Fig. 3. Confusion Matrix

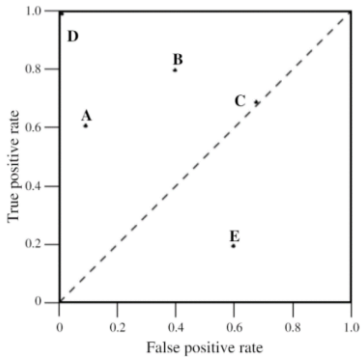

Fig. 4. A sample of Receiver Operating Characteristic Curve (ROC) (Fawcett 2006)

3.4.2. Threshold Metrics

目的是要用來測量分類預測的錯誤情形，了解模型的預測準確能力為何，最常用的標準指標為 Accuracy。

- **Accuracy**

  所有預測結果中，資料類別有被正確預測，也就是資料有被正確分類的比例。其公式為：

$$\text{Accuracy} = \frac{Correct\ Predictions}{Total\ Predictions} = \frac{TP+TN}{(TP+FP+FN+TN)} \quad (1)$$

雖然 Accuracy 已經被廣泛運用在多種任務中，但卻不適合用在類別不平衡問題，例如：當資料集的不平衡率愈高，模型所獲得的 Accuracy 可能愈高，但實際上模型可能只能夠分出大類而無法正確辨識小類。因此，在類別不平衡問題上還會使用其他更為合適的衡量指標，其中，較常使用的衡量指標如下。

- **Sensitivity**

Sensitivity 又稱 True Positive Rate (TPR)，指的是實際類別為 Positive 的資料中有被正確判斷成 Positive 的比例，也就是模型有預測到多少 Positive 類別的資料。其公式為：

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (2)$$

- **Specificity**

Specificity 又稱 True Negative Rate (TNR)，指的是實際類別為 Negative 的資料中有被正確判斷成 Negative 的比例，也就是模型有預測到多少 Negative 類別的資料。其公式為：

$$\text{Specificity} = \frac{TN}{FP+TN} \quad (3)$$

這兩項指標的分數愈高，表示分類結果愈好，由於都只專注在一種類別上，因此適合用在類別不平衡問題，而其中又以 Sensitivity 更為合適。

- **G-mean**

G-mean 即 Geometric Mean，同時考慮到 Sensitivity 和 Specificity，故更為客觀。其公式為：

$$\text{G-Mean} = \sqrt{Sensitivity * Specificity} \quad (4)$$

- **Precision**

Precision 表示預測類別為 Positive 的情況下實際類別也為 Positive 的資料比例。其公式為：

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5)$$

- **Recall**

Recall 表示實際類別為 Positive 的情況下正確召回，也就是有被成功分類成 Positive 的比例，等同於 Sensitivity。其公式為：

$$\text{Recall} = \frac{TP}{TP+FN} \quad (6)$$

- **F_β-measure**

F_β-measure 因為同時考慮到 Precision 和 Recall，所以常被應用在類別不平衡問題。主要透過係數β來計算，當β愈大，表示 Recall 的重要性愈大，$\beta = 1$就表示認為 Precision 和 Recall 兩者的重要性相同，一般常用 F1-Score 來做衡量。其公式為：

$$F_\beta = \frac{1+\beta^2}{\frac{1}{precision}+\frac{\beta^2}{recall}} = \frac{(1+\beta^2)\cdot precision \cdot recall}{\beta^2 \cdot precision + recall} \quad (7)$$

- **F-measure**

F-measure 又稱為 F-Score 或 F1-Score，也就是$\beta = 1$的F_β-measure。其公式為：

$$F_1 = \frac{2}{\frac{1}{precision}+\frac{1}{recall}} = \frac{2\cdot precision \cdot recall}{precision+recall} \quad (8)$$

Threshold Metrics 雖然容易做計算，但是其前提假設是訓練集的資料類別分布會等同於測試集和現實世界狀況的資料分布，因此可能會造成模型學習狀況和現實不符之情形發生。

3.4.3. Rank Metrics

根據模型預測的分數，可以用不同的閾值 (Threshold) 來評估模型效能。在各種閾值範圍內分數較佳的那些模型表示模型效能較好。最常用的指標為 ROC Curve 和 AUC。

- **ROC Curve (Receiver Operating Characteristic Curve)**

將所有閾值下，模型的 FPR 和 TPR 繪製成圖形，該圖形即為 ROC Curve，用來分析分類器。其中，TPR 即 Sensitivity，而 FPR 即 Specificity。透過 ROC Curve 可以觀察模型在不同閾值下的效能為何。

- **AUC (Area under the Curve of ROC)**

  為 ROC Curve 的曲線下面積，可用來評估模型成效。AUC 的範圍為 0 到 1，但基本上分類器的 AUC 都至少為 0.5 以上，愈接近 1 表示模型成效愈好，關於不同 AUC 所代表的意義如表 3。

Table 3. AUC 數值範圍的表示意義 (整理自 Safari et al. 2016)

AUC = 0.5	幾乎沒有判別力(fail)
0.6 ≤ AUC < 0.7	很差的判別力(poor)
0.7 ≤ AUC < 0.8	一般的判別力(fair)
0.8 ≤ AUC < 0.9	好的判別力(good)
AUC ≥ 0.9	非常好的判別力(excellent)

4. Quick survey on which algorithm(s) perform well/poorly on imbalanced datasets?

4.1. Sampling techniques on imbalanced datasets

我們找到一篇論文(Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou 2009) ，其實驗中比較了許多處理類別不平衡問題的演算法，包括了上述介紹的 Data-Level 與 Algorithm-Level，分別以 AUC、F-measure 和 G-mean 來對 UCI datasets 中不平衡資料集做衡量。除了 Bagging 和 Random Forest 相關之演算法，其它方法在預測結果上採用 Adaboost 做最終預測並以單一的 CART 決策樹作為分類器。

Table 4. Sampling techniques / Algorithms (整理自 Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou 2009)

Sampling techniques / Algorithms	
Bagg	Bagging
Ada	Adaboost
Under	Under-sampling + AdaBoost
SMOTE	SMOTE + AdaBoost
Chan	Chan & Stolfo's method + AdaBoost
Cascade	BalanceCascade
Easy	Easyensemble
RF	Random Forests
BRF	Balanced Random Forests
Under-RF	Under-sampling + Random Forests
Over-RF	Over-sampling + Random Forests
Asym	AsymBoost, cost-sensitive variant of AdaBoost
CART	Classification and regression trees, single classifier

4.1.1. Results

AUC	*mf-morph*	*mf-zernike*	*pima*	*vehicle*	*wpbc*	***avg.***
Bagg	.887 ± .004	.855 ± .002	.821 ± .003	.859 ± .003	.688 ± .009	.757 ± .129
Ada	.888 ± .002	.795 ± .003	.788 ± .006	.854 ± .003	.716 ± .009	.760 ± .088
Asym	.888 ± .001	.801 ± .005	.788 ± .005	.853 ± .002	.721 ± .012	.761 ± .088
SMB	.897 ± .002	.788 ± .007	.790 ± .003	.864 ± .003	.720 ± .013	.763 ± .092
Under	.916 ± .001	.881 ± .003	.789 ± .002	.846 ± .003	.694 ± .010	.769 ± .100
Over	.889 ± .002	.779 ± .007	.791 ± .004	.855 ± .003	.711 ± .010	.751 ± .103
SMOTE	.912 ± .001	.862 ± .004	.792 ± .003	.858 ± .004	.709 ± .004	.772 ± .097
Chan	.912 ± .002	.903 ± .002	.786 ± .007	.856 ± .002	.706 ± .009	.781 ± .097
Cascade	.905 ± .001	.891 ± .001	.799 ± .005	.856 ± .002	.712 ± .011	.778 ± .093
Easy	.918 ± .002	.904 ± .001	.809 ± .004	.859 ± .004	.707 ± .009	.787 ± .096
RF	.880 ± .007	.840 ± .008	.821 ± .004	.869 ± .008	.677 ± .030	.749 ± .133
BRF	.901 ± .002	.866 ± .009	.809 ± .003	.850 ± .002	.646 ± .014	.764 ± .109
Under-RF	.919 ± .003	.889 ± .002	.818 ± .004	.855 ± .002	.661 ± .008	.772 ± .110
Over-RF	.881 ± .004	.854 ± .003	.819 ± .004	.866 ± .003	.670 ± .010	.750 ± .130

Fig. 5. AUC results (Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou 2009)

F Measure	*mf-morph*	*mf-zernike*	*pima*	*vehicle*	*wpbc*	**avg.**
CART	.251 ± .022	.216 ± .015	.584 ± .029	.523 ± .019	.373 ± .023	.329 ± .158
Bagg	.263 ± .016	.183 ± .014	.644 ± .007	.526 ± .011	.410 ± .019	.331 ± .177
Ada	.321 ± .014	.188 ± .017	.611 ± .007	.545 ± .010	.432 ± .014	.352 ± .173
Asym	.344 ± .015	.191 ± .010	.613 ± .011	.561 ± .008	.444 ± .015	.362 ± .175
SMB	.351 ± .013	.295 ± .018	.641 ± .006	.606 ± .012	.452 ± .011	.393 ± .175
Under	.579 ± .004	.538 ± .004	.644 ± .002	.623 ± .005	.449 ± .008	.477 ± .132
Over	.319 ± .012	.166 ± .011	.609 ± .009	.539 ± .017	.427 ± .010	.345 ± .175
SMOTE	.560 ± .005	.538 ± .007	.627 ± .004	.615 ± .006	.459 ± .009	.468 ± .134
Chan	.635 ± .001	.577 ± .002	.618 ± .006	.608 ± .003	.448 ± .018	.478 ± .140
Cascade	.596 ± .006	.543 ± .004	.643 ± .007	.629 ± .012	.454 ± .007	.485 ± .128
Easy	.624 ± .002	.564 ± .002	.660 ± .005	.638 ± .007	.452 ± .014	.497 ± .136
RF	.261 ± .025	.144 ± .034	.641 ± .013	.544 ± .024	.393 ± .027	.328 ± .181
BRF	.627 ± .003	.500 ± .013	.663 ± .005	.633 ± .007	.401 ± .006	.480 ± .140
Under-RF	.602 ± .004	.530 ± .004	.668 ± .006	.633 ± .007	.419 ± .008	.481 ± .140
Over-RF	.349 ± .014	.292 ± .012	.656 ± .005	.564 ± .015	.397 ± .019	.376 ± .171

Fig. 6. F-measure results (Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou 2009)

G-mean	*mf-morph*	*mf-zernike*	*pima*	*vehicle*	*wpbc*	**avg.**
CART	.473 ± .022	.428 ± .020	.673 ± .024	.658 ± .013	.513 ± .032	.479 ± .178
Bagg	.483 ± .016	.378 ± .021	.720 ± .006	.642 ± .008	.510 ± .032	.461 ± .187
Ada	.560 ± .012	.386 ± .020	.694 ± .006	.664 ± .008	.537 ± .025	.492 ± .189
Asym	.594 ± .014	.392 ± .013	.696 ± .009	.679 ± .007	.549 ± .028	.504 ± .193
SMB	.605 ± .013	.524 ± .019	.719 ± .006	.728 ± .009	.584 ± .021	.545 ± .196
Under	.873 ± .003	.848 ± .004	.719 ± .001	.768 ± .004	.617 ± .008	.709 ± .102
Over	.559 ± .012	.358 ± .015	.692 ± .007	.657 ± .013	.527 ± .013	.482 ± .191
SMOTE	.841 ± .006	.813 ± .007	.708 ± .003	.743 ± .005	.610 ± .009	.680 ± .111
Chan	.920 ± .001	.854 ± .002	.700 ± .005	.738 ± .004	.585 ± .021	.690 ± .134
Cascade	.874 ± .006	.820 ± .003	.735 ± .005	.760 ± .011	.623 ± .007	.709 ± .092
Easy	.914 ± .001	.869 ± .003	.734 ± .004	.781 ± .005	.623 ± .014	.728 ± .107
RF	.479 ± .022	.326 ± .049	.717 ± .010	.659 ± .018	.477 ± .019	.459 ± .190
BRF	.918 ± .002	.831 ± .007	.735 ± .004	.780 ± .007	.567 ± .007	.714 ± .114
Under-RF	.888 ± .005	.844 ± .002	.740 ± .005	.779 ± .006	.588 ± .011	.712 ± .111
Over-RF	.597 ± .013	.519 ± .016	.731 ± .004	.689 ± .013	.494 ± .022	.522 ± .193

Fig. 7. G-mean results (Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou 2009)

從 AUC、F-measure 和 G-mean 的平均上可以看出 Easyensemble 大多都比其他的方法好，因為透過多樣的 subsets，比起一般的 Ensemble method 來說可以獲得更多有用的資訊。

4.2. Classification algorithms on imbalanced datasets

在另一篇論文(Seiffert et al. 2014) 的實驗中，分別評估不同分類器經過 Sampling techniques (包含 Random Under-sampling、Random Over-sampling、One-Sided Selection、Wilson's editing、Cluster-Based Over-sampling、SMOTE 與 Borderline-SMOTE 7 種方法) 後，在不同類別不平衡之比例下的模型效能，並將每個分類器的不同結果做平均後以圖呈現。以下是實驗結果。

Table 5. Sampling techniques / Algorithms (整理自 Seiffert et al. 2014)

7 Sampling techniques / Algorithms	
2NN	k-nearest neighbors (k=2)
5NN	k-nearest neighbors (k=5)
C4.5(D)	Decision Tree (Default Parameter)
C4.5(N)	Decision Tree (disables pruning and enables Laplace smoothing)
LR	Logistic regression
MLP	Multilayer Perceptron
NB	Naive Bayes
RBF	Radial Basis Function Network
RF	Random Forest
RIPPER	Repeated Incremental Pruning to Produce Error Reduction
SVM	Support Vector Machine

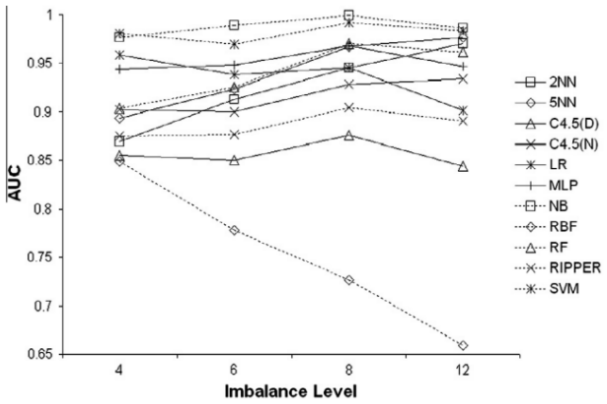

Fig. 8. Classification algorithm performance for different levels of imbalance (Seiffert et al. 2014)

從結果來看，在類別不平衡問題上，當資料集愈不平衡時，RBF 的效能愈差 (平均 AUC 最低)，表示較容易受到類別不平衡的影響；而 NB 和 SVM 的表現則一直維持在一定的分數範圍內，代表兩者受類別不平衡問題的影響較小。

5. Conclusion

在文獻整理的過程中了解到很多不同層級上處理類別不平衡問題的方法，以及從論文 Survey 的過程中學習到在使用不同的分類器上必須選擇適當的資料前處理方法，如此一來不僅能夠避免類別不平衡問題影響模型的效能，甚至能大幅提升模型的結果，最後必須透過選擇適當的評估指標來衡量模型預測結果。處理類別不平衡不是只要處理一個環節或是動作就能有效解決的問題，在每一個步驟與環節都有需要注意的地方，如果其中有一個過程沒有注意到可能就會導致實驗無效或是看不到正確的結果。

6. References

Brownlee, Jason. 2020. “Tour of Evaluation Metrics for Imbalanced Classification.” *Machine Learning Mastery*. https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/ (March 16, 2020).

Elrahman, Shaza M Abd, and Ajith Abraham. “A Review of Class Imbalance Problem.” : 9.

Fawcett, Tom. 2006. “Introduction to ROC Analysis.” *Pattern Recognition Letters* 27: 861–74.

Hsu, Chih-Ling. “Imbalanced Data Classification.” *An Explorer of Things*. https://chih-ling-hsu.github.io/2017/07/25/Imbalanced-Data-Classification (March 18, 2020).

“Imbalanced Data : How to Handle Imbalanced Classification Problems.” 2017. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2017/03/imbalanced-data-classification/ (March 16, 2020).

Lin, Wei-Chao, Chih-Fong Tsai, Ya-Han Hu, and Jing-Shang Jhang. 2017. “Clustering-Based Undersampling in Class-Imbalanced Data.” *Information Sciences* 409–410: 17–26. http://www.sciencedirect.com/science/article/pii/S0020025517307235 (March 16, 2020).

“[Real or Fake] Fake JobPosting Prediction.” https://kaggle.com/shivamb/real-or-fake-fake-jobposting-prediction (March 22, 2020).

Safari, Saeed, Alireza Baratloo, Mohamed Elfil, and Ahmed Negida. 2016. “Evidence Based Emergency Medicine; Part 5 Receiver Operating Curve and Area under the Curve.” *Emergency* 4(2): 111–13. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4893763/ (March 22, 2020).

Seiffert, Chris, Taghi M. Khoshgoftaar, Jason Van Hulse, and Andres Folleco. 2014. “An Empirical Study of the Classification Performance of Learners on Imbalanced and Noisy Software Quality Data.” *Information Sciences* 259: 571–95. http://www.sciencedirect.com/science/article/pii/S0020025511000065 (March 19, 2020).

Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. 2009. “Exploratory Undersampling for Class-Imbalance Learning.” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39(2): 539–50. http://ieeexplore.ieee.org/document/4717268/ (March 22, 2020).

Zou, Quan et al. 2013. “An Approach for Identifying Cytokines Based On a Novel Ensemble Classifier.” *BioMed research international* 2013: 686090.