المادة	ADM
عنوان الوظيفة	مقارنة المصنفات الخمسة على مسألة القروض: Logistic Regression Decision Tree Bayes KNN SVM **تدمج مع وظيفة AWP**
إعداد	د. باسل الخطيب
تاريخ التوزيع	2022/08/23
تاريخ الإعادة	2022/10/1

المسألة: الموافقة على طلب قرض بنكي أم لا

تُعدّ مسألة الموافقة على طلب قرض بنكي من المسائل الهامة جداً لإدارات البنوك وذلك لأنها أحد مصادر الدخل لها. إلا أن القرار الخاطئ قد يُكلف البنك خسائر كبيرة. وهنا، نعني بالقرار الخاطئ إعطاء قرض لشخص لا يقوم بتسديد دفعات القرض المستحقة للبنك لاحقاً معرضاً البنك للخسائر. مما يعني أهمية قرار تصنيف الطلب إلى مقبول أو مرفوض وذلك وفق بيانات طالب القرض.

يستغرق موظفو البنك عادةً وقتاً في دراسة بيانات طالب القرض بهدف التحقق من ملاءته المالية مثل دخله ودخل زوجه ومستواه التعليمي وغير ذلك.

يُمكن لنا الاستفادة إذاً من أرشيف البنك أي قرارات البنك السابقة للموافقة أو رفض طلبات القروض لبناء نموذج متعلم يُمكن الاستفادة منه لاحقاً في المساعدة في اتخاذ القرار المناسب.

سنعمل في هذا المشروع على البيانات المرفقة.

وهي تأتي في ملف loan_data_set.csv يحوي أكثر من 600 صف row و 12 عمود column:

Loan_ID	رقم القرض
Gender	الجنس
Married	متزوج أم لا
Dependents	عدد المعالين من قبل طالب القرض
Education	درجة التعليم
Self_Employed	صاحب عمل خاص
ApplicantIncome	الدخل
CoapplicantIncome	دخل الزوج
LoanAmount	مبلغ القرض
Loan_Amount_Term	مدة القرض
Credit_History	تاريخ العميل بالقروض
Loan_Status	الموافقة أو الرفض

يُمكن أن نعاين البيانات في Excel لنأخذ فكرة أولية عنها:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Loan_Status
2	LP001002	Male	No	0	Graduate	No	5849	0		360	1	Y
3	LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	1	N
4	LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	1	Y
5	LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120	360	1	Y
6	LP001008	Male	No	0	Graduate	No	6000	0	141	360	1	Y
7	LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	1	Y
8	LP001013	Male	Yes	0	Not Graduate	No	2333	1516	95	360	1	Y
9	LP001014	Male	Yes	3+	Graduate	No	3036	2504	158	360	0	N
10	LP001018	Male	Yes	2	Graduate	No	4006	1526	168	360	1	Y
11	LP001020	Male	Yes	1	Graduate	No	12841	10968	349	360	1	N
12	LP001024	Male	Yes	2	Graduate	No	3200	700	70	360	1	Y
13	LP001027	Male	Yes	2	Graduate		2500	1840	109	360	1	Y
14	LP001028	Male	Yes	2	Graduate	No	3073	8106	200	360	1	Y

يُمكن أن نلاحظ أولاً ما يلي:

- يوجد أعمدة رقمية وأعمدة فئوية.
- يوجد بعض القيم الناقصة في البيانات (خلايا فارغة) مثل العمود "صاحب عمل خاص" Self_Employed.

- يوجد قيمة ليست رقمية (3+) في عمود "عدد المعالين" Dependents.

المعالجة الأولية للبيانات

نقوم في الشيفرة البرمجية التالية بقراءة الملف loan_data_set.csv ووضعه في إطار بيانات ومن ثم إجراء العمليات التالية عليه:

- استبدال القيم النصية بقيم رقمية باستخدام دالة الاستبدال replace مع قيمة المعامل inplace=True مما يعني تنفيذ التعديلات على إطار البيانات. لاحظ أننا نقوم في هذا المشروع بترميز القيم الفئوية بأرقام دون استخدام المرمزات LabelEncoder وذلك بهدف استعرض كل الطرق الممكنة في بايثون.
- استبدال القيم الناقصة باستخدام الدالة fillna على عمود مبلغ القرض بالمتوسط الحسابي (mean) لعمود مبلغ القرض.
- استبدال القيم الناقصة في جميع الأعمدة الأخرى بمنوال العمود (mode) أي القيمة الأكثر تكراراً في العمود.
- حذف عمود رقم القرض باستخدام دالة الحذف drop لأنه لا يُستخدم في مسألة التصنيف.

؟

الدخل والخرج

نقوم في الشيفرة البرمجية التالية باختيار كل الأعمدة (عدا حالة القرض) لتكون الدخل X وعمود حالة القرض Loan_Status ليكون الخرج y:

؟

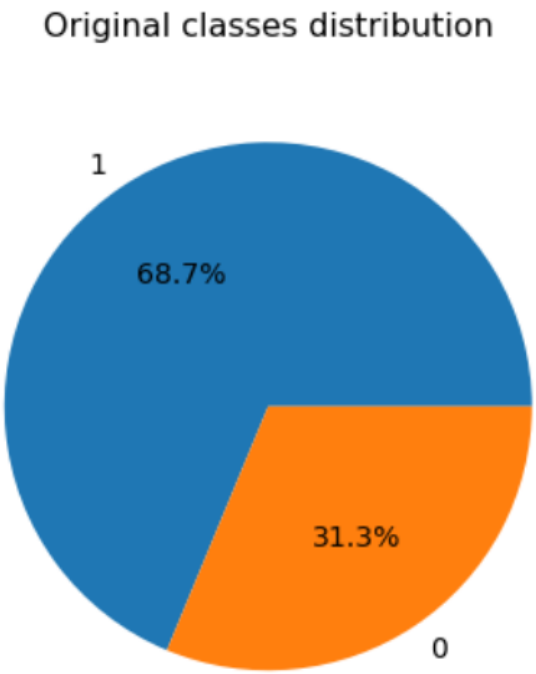
موازنة الصفوف

يُمكن أن تتأثر عملية بناء نموذج التصنيف بمسألة عدم توازن الصفوف. بمعنى أنه في حال كان عدد أمثلة صف أكبر بكثير من عدد أمثلة صف آخر فيُمكن أن ينحاز المُصنف للصف ذو عدد الأمثلة الأكبر.

نقوم في الشيفرة التالية أولاً بمعاينة نسب توزع الصفوف في بياناتنا وذلك عن طريق رسم فطيرة تُظهر النسب المئوية لعدد الأمثلة من كل صف:

```
؟
```

مما يُظهر:

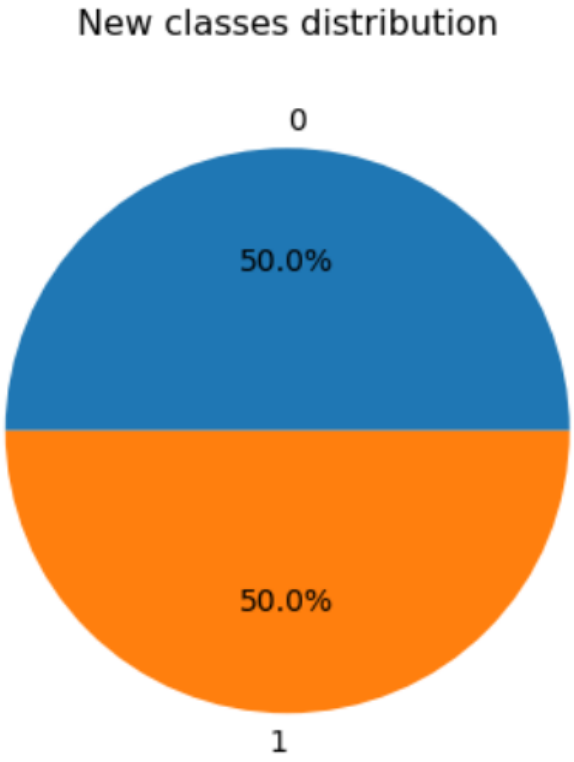


يُبين الرسم السابق ضرورة موازنة الصفوف نظراً لطغيان الصف 1 (No).

نقوم في الشيفرة التالية باستخدام المكتبة imblearn.over_sampling وذلك بإنشاء غرض من الصف SMOTE ومن ثم استدعاء الدالة fit_resample التي تُضيف أمثلة جديدة كي تُصبح الصفوف متوازنة:

```
؟
```

ويكون لدينا أخيراً:



تقسيم البيانات إلى تدريب واختبار

نعمد عادةً إلى تقسيم البيانات إلى قسمين: القسم الأول (80% من البيانات عادةً) ويتم تدريب نموذج التعلم على هذا القسم. أما القسم الثاني فيُستخدم لحساب معايير تقييم نموذج التعلم.

نستخدم المكتبة sklearn.model_selection للقيام بذلك كما تُبين الشيفرة التالية:

```
؟
```

لاحظ أن الدالة train_test_split والتي نُمرر لها كل من:

- X: إطار بيانات يحوي كل الأعمدة عدا عمود الصف.
- y: مصفوفة أحادية (عمود الصف).
- test_size: النسبة المئوية لبيانات الاختبار من البيانات الكليه (20% بشكل افتراضي)

تُعيد:

- X_train: بيانات التدريب المختارة عشوائيًا من X.
- X_test: بيانات الاختبار المختارة عشوائيًا من X.
- y_train: صفوف بيانات التدريب.
- y_test: صفوف بيانات الاختبار.

بناء نموذج التصنيف

نقوم في الشيفرة التالية بملائمة نموذج التصنيف مع بيانات التدريب fit(X_train, y_train) :

```
؟
```

حساب معايير تقييم النموذج

يُمكن الآن إظهار معايير التقييم للنموذج المتعلم وذلك بحساب نتيجة النموذج y_pred مع بيانات الاختبار X_test ومن ثم حساب معايير التقييم التي تُقارن بين بيانات الاختبار y_test ونتيجة النموذج y_pred:

```
؟
```

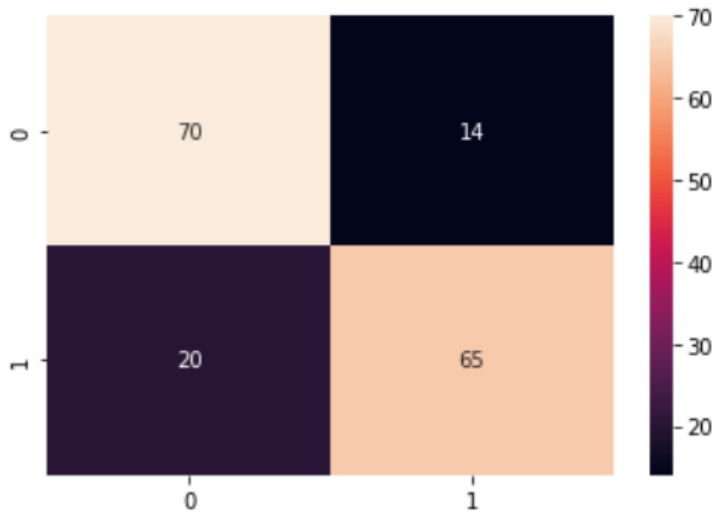
تُبين معايير التقييم الناتجة كفاءة النموذج المولد حيث معظم المقاييس أكبر من 75%:

```
Accuracy: 79.88


Precision: 82.28


Recall: 76.47


Recall: 79.27
```

يُمكن أيضاً حساب مصفوفة الارتباك باستخدام الدالة confusion_matrix والتي نمرر لها قيم الاختبار و قيم التنبؤ لتقارن بينهم، ومن ثم نرسم التمثيل البياني للمصفوفة باستخدام الدالة heatmap من المكتبة seaborn:

```
؟
```

يكون الإظهار:



لاحظ أن النموذج أصاب في:

135 حالة = 70 TP + 65 TN

وأخطأ في:

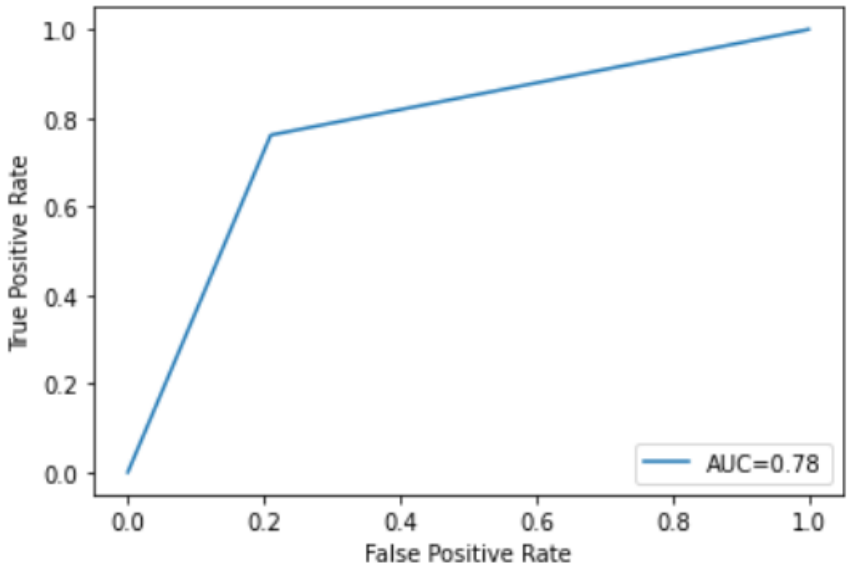
34 حالة = 14 FP + 20 FN

يُمكن استخدام الشيفرة التالية لرسم المنحني المستقبل ROC باستخدام الدالة roc_auc_score وحساب المساحة تحته AUC باستخدام الدالة roc_auc_score من الصف metrics في المكتبة sklearn:

```
؟
```

يكون الإظهار:

(لاحظ أن المساحة تحت المنحني تساوي 0.78 مما يعني جودة المصنف)



حفظ النموذج المتعلم في ملف

يُمكن حفظ النموذج المتعلم في ملف لاستخدام الملف في التنبؤ لاحقاً وعدم بناء النموذج في كل مرة. نستخدم في الشيفرة التالية الدالة dump من المكتبة pickle والتي نُمرر لها النموذج والملف الذي سيُحفظ فيه:

```
؟
```

يُمكن لنا التصريح عن دالة مخصصة لاستخدامها للتنبؤ في كل مرة. نُمرر لهذه الدالة المثال (قائمة فيها عنصر واحد هو قائمة من القيم) واسم ملف النموذج. تُعيد هذه الدالة إما No إذا كان التنبؤ 0 (هو ترميز No) و Yes إذا كان التنبؤ 1:

```
؟
```

تُبين الشيفرة التالية استدعاء الدالة السابقة مع مثال:

```
z=[[0,1,2,1,0,4006,1526,168,360,1]]
p=predictResult(z,'DTmodel')
print(p)
```

يكون الإظهار:

```
Yes
```

	Logistic Regression	Decision Tree	Bayes	KNN	SVM
Precision					
Accuracy					
Recall					
F1					
AUC					

ماذا ترسل للأستاذ: لا شيء

1- تحميل الوظيفة على Moodle (**طالب واحد فقط يرفع مو كل واحد**).

سلم التصحيح	
الطلب	**العلامة العظمى**
برمجة الخوارزميات	50
المقارنة	10
صفحة الويب عليها كل نتائج المقارنة للخمسة خوارزميات بشكل أنيق	20
الاستضافة على الويب	**10**
جودة التقرير	**10**
تسليم الوظيفة قبل الموعد	**5 علامات إضافية**
الإجمالي	**100**