Supervised Learning - Classification

Haeji YUN

1. Dataset

Dans le cadre de ce projet d'apprentissage supervisé, notre objectif est de construire un modèle prédictif permettant d'attribuer une classe caractérisant le statut socio-économique non dévoilé à la population étudiée.

L'étude porte sur un ensemble de données comprenant 99.989 résidents de la France métropolitaine. Pour mener à bien la modélisation, nous utiliserons un sous-ensemble de 49.995 individus comme données d'entraînement, tandis que la prédiction sera effectuée sur les 49.994 individus restants.

2. Prétraitement des données

2.1. Données

La base de données comporte 9 variables explicatives dont une est quantitative.

Variable quantitative : *CURRENT_AGE*

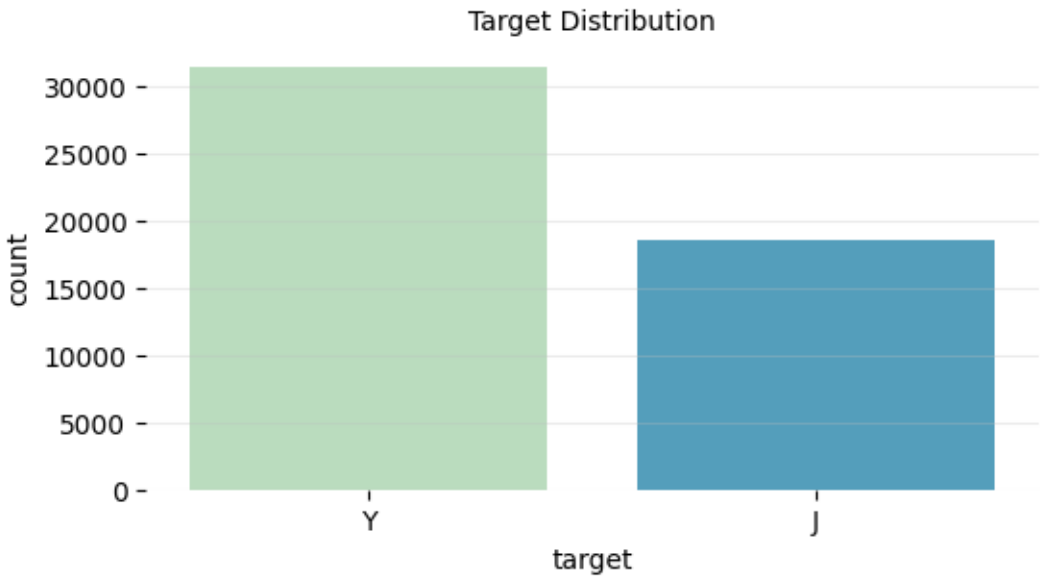
Variables qualitatives : *Primary_key*, *ACTIVITY_TYPE*, *SEX*, *INSEE_CODE*, *job_42*, *is_student*, *highest_diploma*, *household type*

Primary_key	1	4	8	15	16	17	23
ACTIVITY_TYPE	TACT2.1	TACT1.1	TACT1.1	TACT2.1	TACT2.1	TACT2.2	TACT2.1
SEX	Female	Male	Male	Male	Female	Male	Female
INSEE_CODE	01004	01004	01004	01004	01004	01004	01007
job_42	csp_7_5	csp_4_7	csp_6_8	csp_7_4	csp_7_7	csp_8_4	csp_7_7
CURRENT_AGE	76	49	53	87	81	19	74
is_student	False	False	False	False	False	True	False
highest_diploma	DIP1_8	DIP1_3	DIP3	DIP3	DIP1_3	DIP1_4	DIP1_4
household_type	FM4_4	FM4_1	FM4_2	FM4_3	FM1_2	FM4_1	FM4_4
target	Y	Y	J	J	Y	Y	J

Variable cible

target est la variable cible. Elle est composée de deux classes distinctes : Y et J. Cependant, une observation initiale révèle un déséquilibre significatif entre ces classes. En effet, la classe Y est majoritaire, représentant 63% des individus, tandis que la classe J ne représente que 37% de l'ensemble des individus.

Face à ce déséquilibre, nous prévoyons d'appliquer une technique de pondération des classes en attribuant des poids de pénalité aux observations de la classe minoritaire lors de l'entraînement du modèle. Ainsi, nous pourrons réduire l'impact du déséquilibre de classe sur le processus d'apprentissage, permettant ainsi au modèle de mieux généraliser et de produire des prédictions plus précises pour les deux classes.



2.2. Nettoyage des données

Primary_key

Parmi les variables, nous allons éliminer *Primary_key* étant donné qu'elle est unique à chaque individu et ne fournit aucune information pertinente pour la classification.

INSEE_CODE

Quant à la variable *INSEE_CODE*, elle présente un total de 13.664 modalités. La table ci-dessous récapitule la distribution de ces modalités.

Nous observons que chaque catégorie est peu fréquente dans l'ensemble de données avec une moyenne de 3,66. Bien que la modalité la plus fréquente soit apparue 388 fois, la moitié des modalités n'est apparue qu'une seule fois.

Le faible nombre d'apparition des catégories et le déséquilibre entre les différentes catégories peuvent limiter la capacité à fournir des informations significative et avoir un impact négatif au modèle à généraliser correctement sur de nouvelles données.

```
count  13664.000000
mean       3.658885
std       10.337796
min        1.000000
25%        1.000000
50%        1.000000
75%        3.000000
max      388.000000
```

Pour remédier à ce problème, nous envisageons de regrouper les modalités similaires ensemble afin de réduire le nombre de catégories et de simplifier la représentation de la variable *INSEE_CODE,* tout en préservant les informations pertinentes pour notre modélisation.

La variable *INSEE_CODE* correspond aux communes. Nous allons la regrouper au niveau du département, *Dep* en combinant avec les données *city_admin.csv* qui donne le code de département associé à chaque commune et *city_loc.csv* qui donne le nom de chaque département.

Avec le regroupement au niveau du département, nous parvenons à obtenir une distribution plus équilibrée. La table ci-dessous fournit un résumé de cette distribution.

```
count    96.000000
mean    520.781250
std     398.193637
min      55.000000
25%     218.750000
50%     417.000000
75%     680.250000
max    1961.000000
```

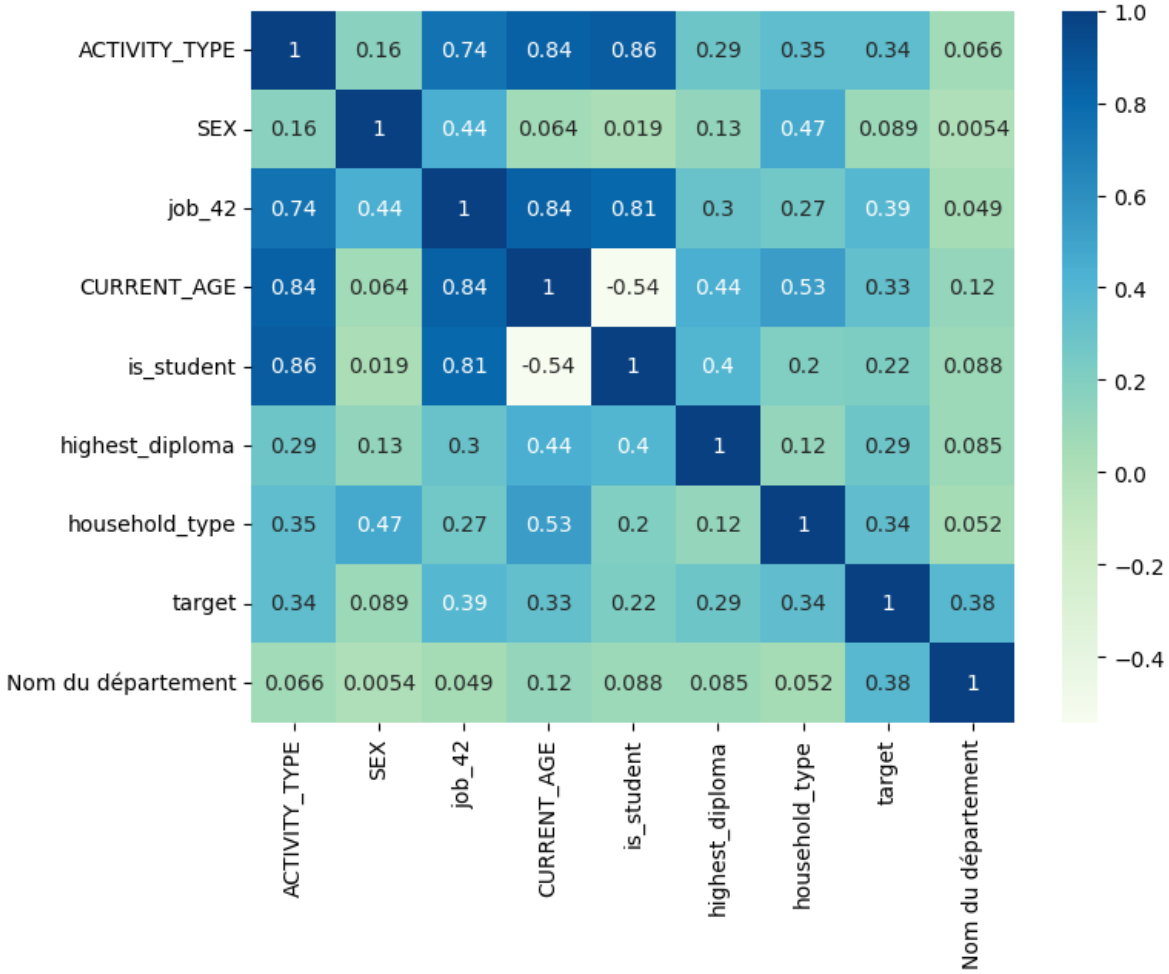
La variable compte maintenant 96 modalités, ce qui représente une réduction significative. De plus, la fréquence moyenne des modalités est d'environ 520 avec une dispersion modérée dans les fréquences des modalités.

Cette réduction drastique du nombre de catégories contribue à simplifier considérablement la structure de la variable et la fréquence plus élevée de chaque modalité indique une présence plus robuste des catégories individuelles. Ainsi, la variable *Dep* peut faciliter la modélisation avec des modalité potentiellement plus informatives.

2.3. Corrélation

Nous constatons des corrélations significatives entre certaines paires de variables. Nous pouvons supposer une relation étroite entre le type d'emploi et la catégorie socio-professionnelle des individus avec une forte corrélation entre *job_42* et *ACTIVITY_TYPE*. De plus, ces variables sont également corrélées avec *CURRENT_AGE* et *is_student*. Cette relation entre plusieurs variables pourrait fournir des informations utiles pour la modélisation et la prédiction du statut socio-économique.

En outre, certaines variables sont également corrélées avec la variable cible. *job_42*, *ACTIVITY_TYPE*, *CURRENT_AGE*, *Nom du département*, et *household_type* présentent des corrélations significatives avec la variable cible. Ces variables sont potentiellement importantes pour prédire le statut socio-économique des individus.



3. Modélisation

Comme modèles de classification, nous allons utiliser la régresion logistique, la forêt aléatoire et le SVM. Avant d'appliquer des modèles d'apprentissage, nous allons diviser l'ensemble de données en ensembles d'entraînement et de test. Puis nous allons dichotomiser les données pour pouvoir y appliquer la modélisation.

```
# Division des données
X_train, X_test, y_train, y_test = train_test_split(df,target,test_size = 0.2,
random_state = 1)

# Dichotomisation des données
X_train_encodded = pd.get_dummies(X_train, drop_first = True )
X_test_encodded = pd.get_dummies(X_test, drop_first = True )
```

Dans le but de maximiser l'efficacité de nos modèles, nous allons recourir à GridSearchCV pour déterminer les paramètres optimaux de nos algorithmes. Nous utiliserons une validation croisée stratifiée avec 3 splits.

Nous allons également appliquer l'argument *class_weight='balanced'* à chaque entraînement pour tenir en compte du déséquilibre de nos données.

Régression Logistique

Pour la régression logistique, nous avons exploré deux optimizers différents, *liblinear* et *saga*, ainsi que plusieurs valeurs pour le terme de régularisation *0,1*, *1* et *10*. Le modèle optimal obtenu a pour optimizer *liblinear* et le terme de régularisation *10*.

	precision	recall	f1-score	support
J	0.74	0.84	0.78	3688
Y	0.90	0.82	0.86	6311
accuracy			0.83	9999
macro avg	0.82	0.83	0.82	9999
weighted avg	0.84	0.83	0.83	9999

Avec une précision atteignant 83%, le modèle démontre une capacité globale à classer correctement les données. Il semble présenter une tendance à faire moins d'erreurs de classification pour la classe majoritaire *Y*, ce qui est corroboré par un F1-score légèrement supérieur pour cette classe par rapport à la classe *J*. Le modèle est plus efficace pour identifier les instances

de la classe majoritaire. Cependant, les résultats globaux restent solides avec une précision élevée et un bon équilibre entre rappel et précision pour les deux classes.

Forêt Aléatoire

Nous avons entraîné le modèle de forêt aléatoire avec le nombre d'estimateurs *50* et *100*. Le modèle optimal obtenu est celui avec un nombre d'estimateurs de *100*.

Le modèle donne une précision de 81%. On remarque une tendance où le modèle semble mieux performant pour la classe majoritaire *Y*, comme le confirme le F1-score légèrement plus élevé pour cette classe par rapport à la classe *J*. Malgré cette différence, les résultats sont satisfaisants avec un bon équilibre entre le rappel et la précision pour les deux classes.

Real Class	J	Y
Predicted Class		
J	2608	838
Y	1080	5473

	precision	recall	f1-score	support
J	0.76	0.71	0.73	3688
Y	0.84	0.87	0.85	6311
accuracy			0.81	9999
macro avg	0.80	0.79	0.79	9999
weighted avg	0.81	0.81	0.81	9999

SVM

Pour le modèle SVM, nous avons exploré le paramètre de régularisation *1* et *10* avec les noyaux *RBF* et *poly*. Le modèle optimal a un paramètre de régularisation égal à *1* et un noyau *poly*.

Avec une précision de 75%, le modèle a une capacité de classification correcte mais reste le moins performant parmi tous les modèles, surtout pour la classe minoritaire *J* avec un F1-socre de 0,64 par rapport à 0,81 pour la classe majoritaire *Y*.

	precision	recall	f1-score	support
J	0.70	0.59	0.64	3688
Y	0.78	0.85	0.81	6311
accuracy			0.75	9999
macro avg	0.74	0.72	0.73	9999
weighted avg	0.75	0.75	0.75	9999

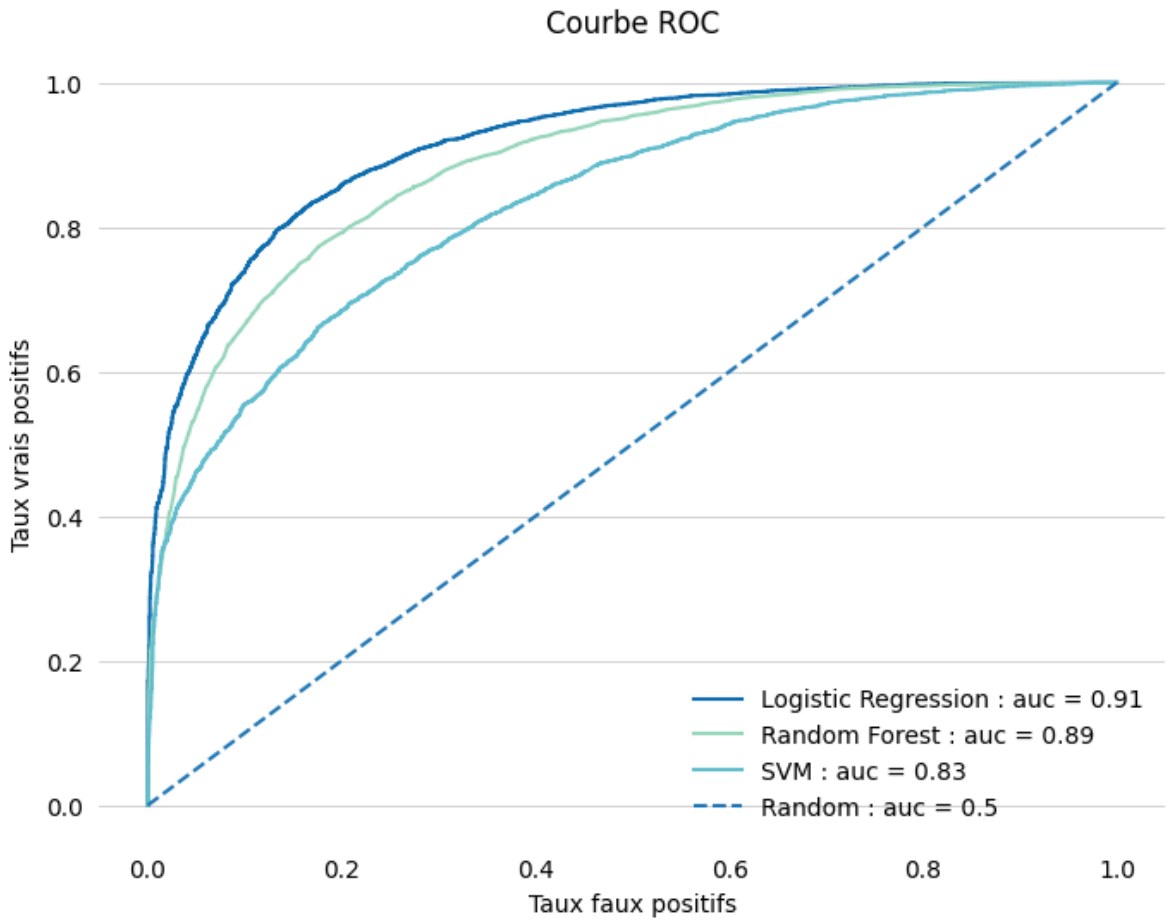
Meilleur Modèle

La table ci-dessous récapitule les meilleurs paramètres obtenus pour chaque algorithme et nous allons choisir le meilleur modèle parmi les trois.

	Model	params	mean_test_score	std_test_score
4	Logistic Regression	{'C': 10, 'solver': 'liblinear'}	0.832583	0.001490
1	Random Forest	{'n_estimators': 100}	0.810981	0.001123
3	SVM	{'C': 1, 'kernel': 'poly'}	0.747925	0.003077

Pour cela, nous allons comparer la courbe ROC et la AUC des trois algorithmes.

Nous observons que la régression logistique a sa courbe ROC toujours au-dessus des courbes ROC des deux autres modèles. Elle a une meilleure performance globale en termes de sensibilité et de spécificité pour différentes valeurs de seuil de classification.



La régression logistique se distingue avec une AUC de 0,91, la plus élevée parmi les modèles examinés, suivie de près par la forêt aléatoire avec une AUC de 0,89, puis par le SVM avec une AUC de 0,83. Cette performance remarquable souligne la capacité de la régression logistique à bien séparer les classes, avec une précision relativement élevée, et à mieux discriminer entre les classes positives et négatives dans notre ensemble de données.

Ainsi, la régression logistique se démarque comme le choix optimal pour notre modèle final. Pour perfectionner davantage la classification, nous allons optimiser le seuil de décision en maximisant la différence entre les vrais positifs *tp* et les faux positifs *fp*. En ajustant ce seuil, nous visons à obtenir un équilibre optimal entre la sensibilité et la spécificité du modèle.

Le seuil optimal pour maximiser l'écart entre les vrais positifs *tp* et les faux positifs *fp* est égal à 0,497212, ce qui génère un écart de 0,664346 entre *tp* et *fp*.

	threshold	tp-fp
1120	0.497212	0.664346

4. Prédiction

Nous allons utiliser la régression logistique avec le seuil optimal pour effectuer la prédiction sur les données de test. Notre modèle présente une capacité relativement solide à prédire la classe Y avec 5.216 vrais positifs. Cependant, il rencontre des difficultés lors de la prédiction de la classe J avec 1.095 faux négatifs et 598 faux positifs.

Real Class	J	Y
Predicted Class		
J	3090	1095
Y	598	5216

Malgré ces défis, le modèle affiche une performance globalement satisfaisante, avec une précision de 83%.

	precision	recall	f1-score	support
J	0.74	0.84	0.78	3688
Y	0.90	0.83	0.86	6311
accuracy			0.83	9999
macro avg	0.82	0.83	0.82	9999
weighted avg	0.84	0.83	0.83	9999

Il se distingue notamment dans la prédiction de la classe Y avec une précision de 90% et un rappel de 83%. Bien que la précision pour la classe J soit légèrement inférieure à 74%, le rappel est plus élevé à 84%. Dans l'ensemble, le modèle semble bien équilibré dans sa capacité à identifier les deux classes, J et Y.

5. Conclusion

Nous avons entrepris de modéliser la prédiction du statut socio-économique de la population française en nous appuyant sur diverses caractéristiques telles que le type d'activité, l'âge, le genre, le niveau de diplôme, le type de famille et le lieu d'habitation.

Parmi les différentes approches explorées, la régression logistique s'est démarquée comme le choix optimal avec une précision plus élevée et une capacité à mieux discriminer entre les classes positives et négatives. Malgré les défis persistants rencontrés dans la prédiction de la classe minoritaire, la régression logistique a néanmoins démontré une capacité satisfaisante à identifier les deux classes avec une précision globale de 83%. Ce résultat témoigne de la robustesse de notre approche et souligne son potentiel à capturer les nuances subtiles du statut socio-économique dans la population française.

***Pour exécuter le code fourni, veuillez renseinger le path du fichier où sont enregistrées les bases de données, à la première ligne dans le code.**