Paris-Saclay university
Master 2 Data science
Data camp

Single-cell RNA-seq classification A RAMP data-challenge on the prediction of cellular types based on genes expression level

Réalisé par :
Mamadou DIOUF

Encadré par :
Nicolas JOUVIN
François CAUD

Année universitaire : 2022 / 2023

Contents

List of Figures

1 Introduction

En biologie, les cellules portent (presque) la même information génomique, elles ont tendance à n'exprimer qu'une fraction de leurs gènes conduisant à une spécialisation en types spécifiques aux fonctions biologiques différentes. Ainsi, l'étude et la classification des types de cellules sont d'un intérêt primordial pour de nombreuses applications biologiques et médicales. Au cours de la dernière décennie, la mesure du niveau d'expression des gènes à l'échelle d'une cellule unique est devenue possible grâce à l'essor des technologies à haut débit appelées ARN-seq unicellulaire (scRNA-seq).

L'objectif de ce défi de données est la classification supervisée des types de cellules grâce à l'ensemble de données de référence scMARK de Mendonca et. Al. Les auteurs ont compilé l'expression de 100 000 cellules à partir de 10 études différentes pour servir de comparaison pour différentes approches d'apprentissage automatique, par analogie avec l'ensemble de données de référence MNIST pour la vision par ordinateur.

Ce data-challenge utilise une petite extraction avec seulement 4 types de cellules (les étiquettes à prédire) de scMARK: Cancer_cells, NK_cells, T_cells_CD4+ and T_cells_CD8+ .

2 Objectif du challenge

Dans ce challenge, nous avons l'intention d'utiliser des méthodes d'apprentissages statistiques sur le jeu de données l'extraction scMark définit ci-dessus pour prédire le type de cellules. En allant de l'exploration des données jusqu'à la sélection des modèles en passant la réduction de dimension.

L'ensemble de données public contient 1 500 points répartis en 1 000 points d'entraînement et 500 points de test. Il servira de référence locale pour l'élaboration de nos soumissions. Côté serveur, la soumission utilisera l'ensemble des 1500 points publics comme ensemble de formation, et un autre ensemble de données de test privé et indisponible, contenant 1500 points de test supplémentaires, sera utilisé pour le classement des participants. La distribution des étiquettes dans les jeux de données d'entraînement et de test publics (resp. privés) est la même.

3 Méthodes d'analyse de données et notions théoriques

3.1 Base de données scMark

La base de données scMark est un ensemble de données de référence accessible au public pour l'analyse de séquençage d'ARN unicellulaire (scRNA-seq). Il a été introduit dans un article de Zappia et al. en 2017 et est conçu pour évaluer les performances des méthodes d'analyse scRNA-seq pour différentes tâches, telles que le regroupement cellulaire, l'analyse de l'expression génique et l'analyse de l'expression différentielle.

Elle se compose de deux sous-ensembles de données : scMark-cons et scMark-rep. L'ensemble de données scMark-cons contient des données provenant de 48 cellules individuelles, qui ont été générées artificiellement pour avoir un profil d'expression cohérent pour un ensemble de gènes marqueurs. L'ensemble de données scMark-rep contient des données provenant de 48 cellules individuelles, qui ont été échantillonnées au hasard à partir d'une population de cellules souches embryonnaires de souris.

Les deux sous-ensembles de données ont été séquencés à l'aide du protocole Smart-seq2, qui permet le séquençage de l'ARNm sur toute la longueur. L'ensemble de données scMark fournit une annotation de référence pour les types de cellules et les gènes marqueurs, ce qui permet d'évaluer la précision des méthodes d'analyse scRNA-seq.

Dans l'ensemble, la base de données scMark est devenu un ensemble de données de référence largement utilisé pour évaluer les méthodes d'analyse de scRNA-seq et a contribué à faire progresser le domaine de la génomique unicellulaire.

Notre base est une petite extraction avec seulement 4 types de cellules (les étiquettes à prédire) de scMARK: Cancer_cells, NK_cells, T_cells_CD4+ and T_cells_CD8+. Cependant, nous avons un problème dimensionnel assez élevé avec 1000 points de données (cellules uniques) décrits par 14059 variables (gènes). Puisque nous mesurons le niveau d'expression, les données sont assez rares, avec de nombreux gènes non exprimés pour chaque cellule. Nous constatons que plus 75%

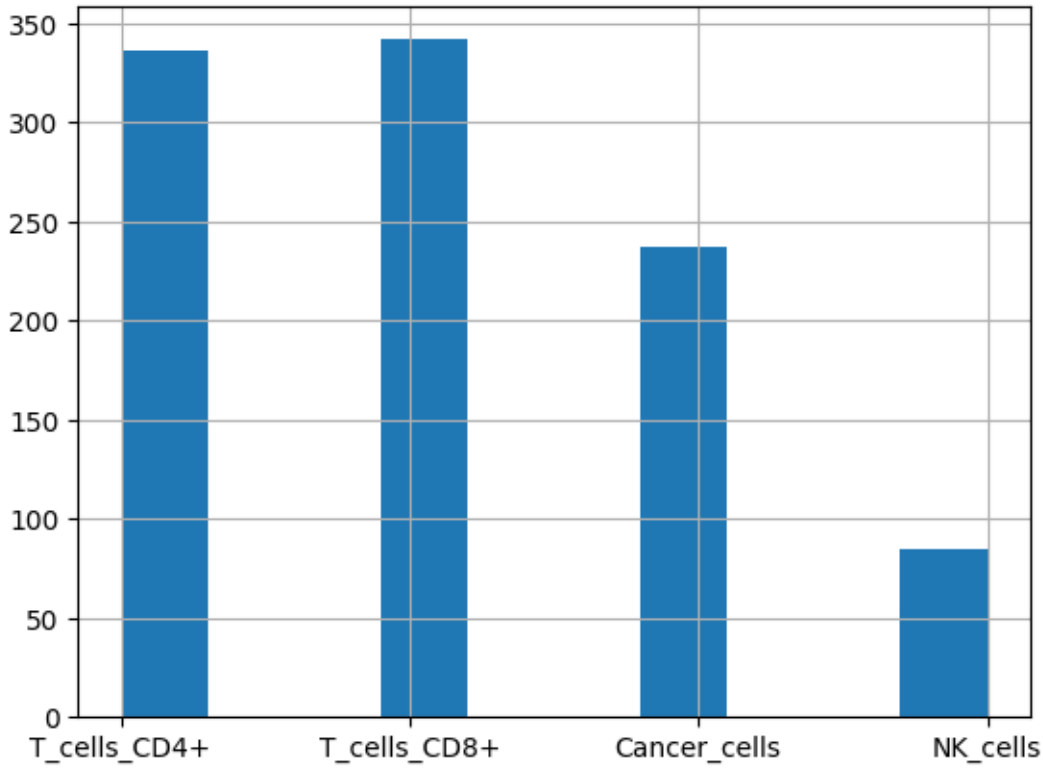

Figure 1: Jeu de données scMark(extraction)

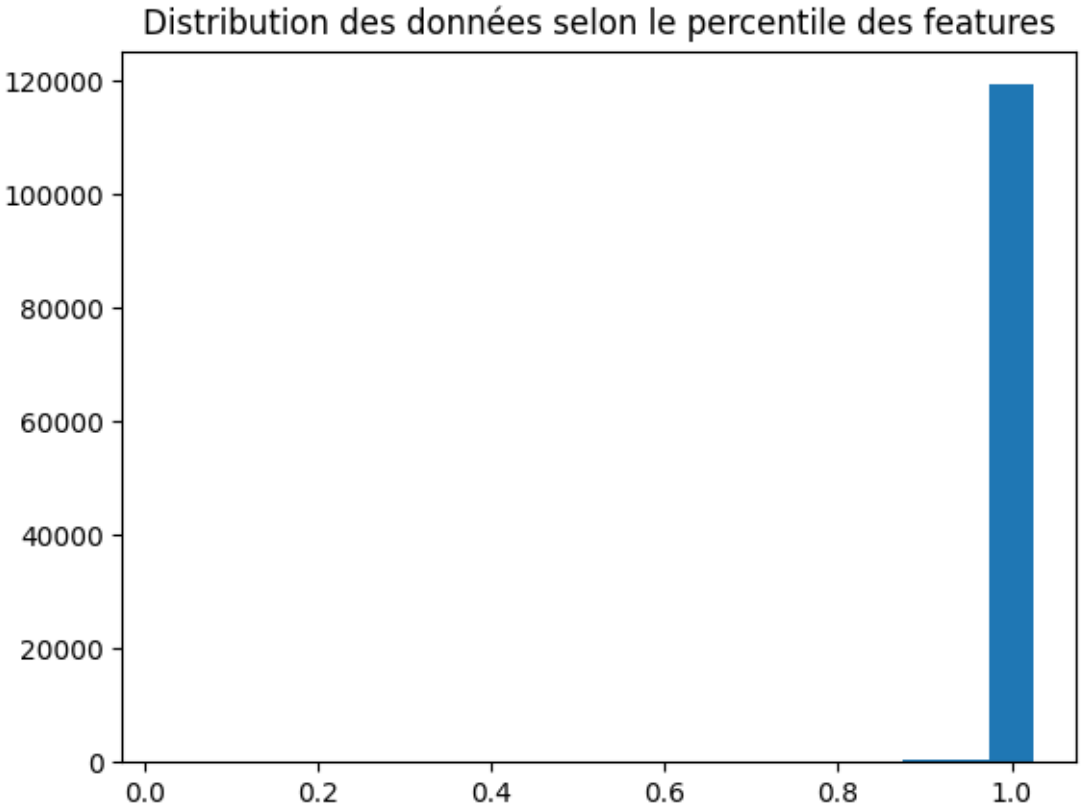

Figure 2: Distribution des données selon le percentile sur les features

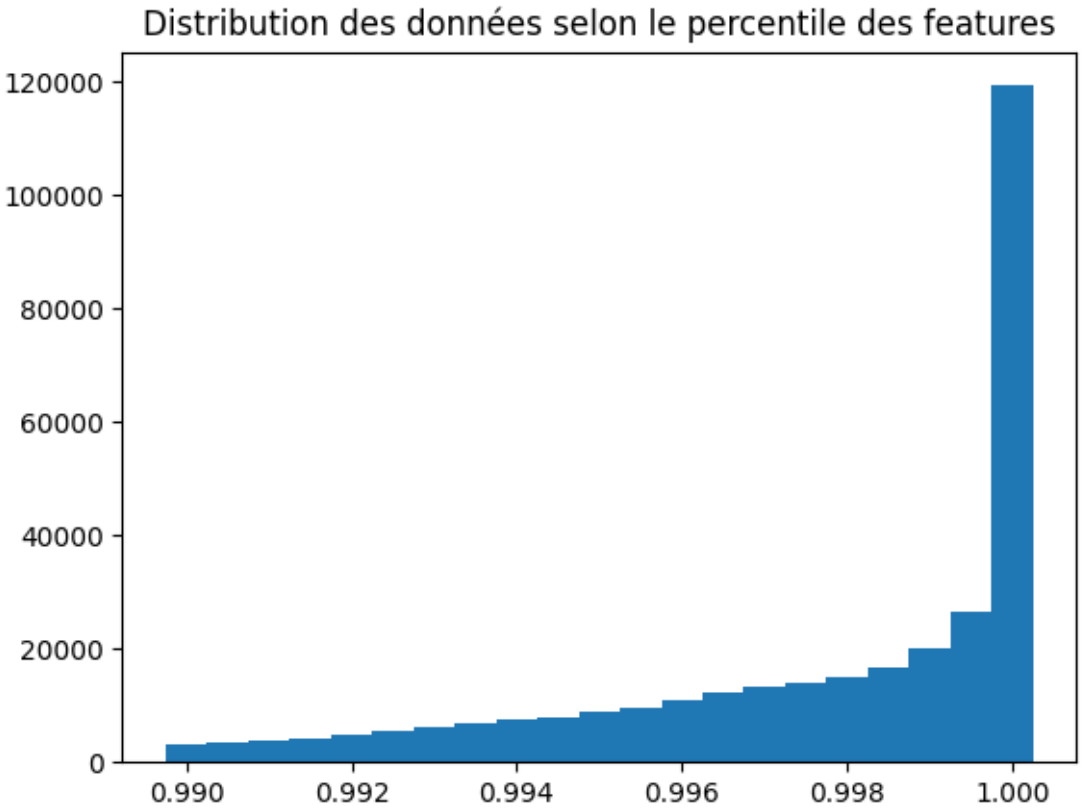

Figure 3: Distribution des données selon le percentile sur les features

3.2 Analyse en Composantes Principales (ACP)

L'analyse en composantes principales (ACP) est une technique de réduction de la dimensionnalité utilisée pour transformer un ensemble de données de grande dimension en un espace de dimension inférieure tout en conservant autant que possible la variance d'origine. L'ACP est largement utilisée dans l'analyse de données, la reconnaissance de formes et l'apprentissage automatique.

L'idée principale derrière l'ACP est de trouver les directions (c'est-à-dire les composantes principales) dans lesquelles les données varient le plus. Ces directions sont déterminées en trouvant les vecteurs propres de la matrice de covariance des données. Les vecteurs propres avec les valeurs propres les plus élevées correspondent aux directions de plus grande variance, et ces vecteurs propres sont utilisés comme composantes principales.

Pour effectuer une ACP sur un ensemble de données, les étapes suivantes sont généralement suivies :

1. Normaliser les données : la première étape consiste à normaliser les données afin que chaque caractéristique ait une moyenne nulle et une variance unitaire. Ceci est fait pour s'assurer que chaque caractéristique contribue de manière égale à l'analyse.
2. Calculez la matrice de covariance : L'étape suivante consiste à calculer la matrice de covariance des données normalisées. La matrice de covariance décrit les relations entre les différentes caractéristiques des données.
3. Calculer les vecteurs propres et les valeurs propres : Les vecteurs propres et les valeurs

propres de la matrice de covariance sont alors calculés. Les vecteurs propres sont les composantes principales et les valeurs propres indiquent la quantité de variance expliquée par chaque composante principale.

4. Choisir le nombre de composantes principales : L'étape suivante consiste à choisir le nombre de composantes principales à conserver. Cela peut être fait en regardant les valeurs propres et en sélectionnant les k vecteurs propres supérieurs avec les valeurs propres les plus élevées. Alternativement, un diagramme d'éboulis peut être utilisé pour déterminer le nombre optimal de composants principaux à conserver.

5. Projeter les données sur le nouvel espace : L'étape finale consiste à projeter les données d'origine sur le nouvel espace de dimension inférieure défini par les composantes principales sélectionnées. Cela se fait en multipliant les données d'origine par les vecteurs propres sélectionnés.

PCA peut être utilisé pour une variété de tâches, telles que la visualisation de données, la compression de données et l'extraction de caractéristiques. Il est particulièrement utile lorsqu'il s'agit de données de grande dimension, car il nous permet de réduire le nombre de caractéristiques tout en conservant la plupart des informations contenues dans les données.

3.3 TruncatedSVD (Singular Value Decomposition)

TruncatedSVD (Singular Value Decomposition) est une technique utilisée pour la réduction de la dimensionnalité des ensembles de données de grande dimension. Elle est similaire à l'analyse en composantes principales (ACP), mais fonctionne spécifiquement sur des matrices creuses, qui peuvent être beaucoup plus grandes et plus complexes en termes de calcul que les matrices denses.

TruncatedSVD décompose une matrice creuse de grande dimension en deux matrices plus petites. La première matrice contient les vecteurs singuliers gauches et la seconde matrice contient les vecteurs singuliers droits. Les vecteurs singuliers représentent les directions dans lesquelles les données d'origine varient le plus et peuvent être utilisés comme nouvelles caractéristiques pour la représentation à dimension réduite des données.

Le nombre de vecteurs singuliers à conserver, et donc le nombre de nouvelles caractéristiques dans la représentation à dimension réduite, est spécifié par l'utilisateur. Il s'agit de la partie "tronquée" de TruncatedSVD - nous ne gardons que les k vecteurs singuliers supérieurs qui expliquent le plus de variance dans les données.

L'avantage de TruncatedSVD par rapport à PCA est qu'il peut gérer de grandes matrices clairsemées. De plus, comme il ne prend en compte que les k vecteurs singuliers supérieurs, il peut être plus rapide et plus économe en mémoire que l'ACP pour les grands ensembles de données. Cependant, il est important de noter que TruncatedSVD ne fournit pas la même interprétation des caractéristiques réduites que PCA, car les vecteurs singuliers ne sont pas garantis orthogonaux.

TruncatedSVD est couramment utilisé dans le traitement de texte et le traitement du langage naturel, où les ensembles de données sont souvent volumineux et clairsemés. Il peut être utilisé pour des tâches telles que la classification de texte, le regroupement de documents et la

modélisation de sujets.

3.4 SelectPercentile

SelectPercentile est une méthode de sélection de fonctionnalités dans l'apprentissage automatique qui sélectionne les k principales fonctionnalités d'un ensemble de données en fonction de leur signification statistique. Plus précisément, SelectPercentile sélectionne les fonctionnalités qui ont les scores les plus élevés selon une fonction de notation spécifiée. La fonction de notation mesure l'association entre chaque caractéristique et la variable cible et peut être basée sur des tests statistiques, tels que le chi carré, le F test ou l'information mutuelle.

SelectPercentile fonctionne en classant les fonctionnalités en fonction de leur score, puis en sélectionnant les k principales fonctionnalités qui correspondent à un centile spécifié des scores. Par exemple, si le centile est défini sur 25%, SelectPercentile sélectionnera les 25% d'entités ayant les scores les plus élevés.

L'avantage de SelectPercentile est qu'il peut être utilisé pour sélectionner automatiquement les fonctionnalités les plus pertinentes à partir d'un grand ensemble de données, ce qui peut améliorer la précision et la généralisation des modèles d'apprentissage automatique. En sélectionnant uniquement les fonctionnalités les plus informatives, SelectPercentile peut réduire le surajustement, accélérer le temps de formation et simplifier le modèle.

SelectPercentile est implémenté dans scikit-learn, une bibliothèque d'apprentissage automatique populaire en Python. Il peut être utilisé en combinaison avec n'importe quel estimateur qui accepte la sélection de caractéristiques comme entrée, comme un modèle de classification ou de régression. SelectPercentile peut également être utilisé conjointement avec d'autres méthodes de sélection de fonctionnalités, telles que l'ACP ou l'élimination récursive de fonctionnalités, pour affiner davantage l'ensemble de fonctionnalités.

3.4.1 f_classif ou F-test

f_classif est un test statistique utilisé pour la sélection de fonctionnalités dans l'apprentissage automatique. Il s'agit d'une fonction de notation qui mesure l'association entre chaque caractéristique et la variable cible dans un problème de classification. Plus précisément, f_classif calcule la valeur F et la valeur p correspondante pour chaque caractéristique, qui indiquent l'importance de l'association entre la caractéristique et la variable cible (calcule la valeur F-value ANOVA pour l'échantillon fourni).

La valeur F mesure le rapport de la variance inter-classe à la variance intra-classe pour une caractéristique donnée. En d'autres termes, il quantifie à quel point les moyennes des valeurs de caractéristiques diffèrent entre les différentes classes par rapport à la variabilité des valeurs de caractéristiques au sein de chaque classe. Une valeur F élevée indique que la caractéristique est hautement discriminante pour la variable cible et doit donc être sélectionnée pour le modèle.

La valeur p mesure la probabilité que la valeur F observée ait pu se produire par hasard, en supposant qu'il n'y a pas d'association réelle entre la caractéristique et la variable cible. Une valeur de p faible (par exemple, $p < 0{,}05$) indique que l'association observée entre la caractéristique et la variable cible est peu susceptible d'être due au hasard, et soutient donc l'hypothèse

selon laquelle la caractéristique est importante pour la classification.

f_classif est couramment utilisé en conjonction avec des méthodes de sélection d'entités telles que SelectKBest ou SelectPercentile pour sélectionner les k principales entités en fonction de leurs valeurs F. Il est implémenté dans scikit-learn, une bibliothèque d'apprentissage automatique populaire en Python. f_classif peut être utilisé pour les problèmes de classification binaire et multi-classes.

3.4.2 chi2 ou chi squared

Chi2, ou chi squared, est un test statistique utilisé pour la sélection de fonctionnalités dans l'apprentissage automatique. Il s'agit d'une fonction de notation qui mesure l'association entre chaque caractéristique et la variable cible dans un problème de classification. Plus précisément, chi2 calcule la statistique chi2 et la valeur p correspondante pour chaque caractéristique, qui indiquent l'importance de l'association entre la caractéristique et la variable cible.

La statistique du chi2 mesure la différence entre la fréquence observée et la fréquence attendue d'une caractéristique donnée pour chaque classe. En d'autres termes, il quantifie à quel point la distribution des valeurs de caractéristique diffère entre les différentes classes par rapport à la distribution attendue sous l'hypothèse d'indépendance entre la caractéristique et la variable cible. Une valeur chi2 élevée indique que la caractéristique est hautement discriminante pour la variable cible et doit donc être sélectionnée pour le modèle.

La valeur p mesure la probabilité que la valeur chi2 observée ait pu se produire par hasard, en supposant qu'il n'y a pas d'association réelle entre la caractéristique et la variable cible. Une valeur de p faible (par exemple, $p < 0{,}05$) indique que l'association observée entre la caractéristique et la variable cible est peu susceptible d'être due au hasard, et soutient donc l'hypothèse selon laquelle la caractéristique est importante pour la classification.

Chi2 est couramment utilisé en conjonction avec des méthodes de sélection de fonctionnalités telles que SelectKBest ou SelectPercentile pour sélectionner les k principales fonctionnalités en fonction de leurs valeurs de chi2. Il est implémenté dans scikit-learn, une bibliothèque d'apprentissage automatique populaire en Python. chi2 peut être utilisé pour les problèmes de classification binaire et multiclasse, mais suppose que les valeurs des caractéristiques ne sont pas négatives et représentent des décomptes ou des fréquences.

4 Approche et comparaison des résultats obtenus

Il existe plusieurs modèles de classifications en apprentissage statistique . Parmi ces modèles on trouve le gradient boosting classifier, le support vector machine classifier, random forest classifier.

4.1 Modèles utilisés

4.1.1 Gradient boosting classifier

Gradient Boosting Classifier est un algorithme d'apprentissage automatique populaire qui est largement utilisé pour les problèmes de classification. Il s'agit d'une technique d'apprentissage d'ensemble qui combine plusieurs apprenants faibles pour construire un classificateur fort.

Dans le gradient boosting, une séquence d'arbres de décision est construite de manière itérative, chaque nouvel arbre apprenant à corriger les erreurs commises par les arbres précédents. L'algorithme commence par construire un arbre de décision simple qui prédit la variable cible en fonction d'une seule caractéristique. L'arbre est entraîné à l'aide des données d'entraînement et ses prédictions sont comparées aux vraies valeurs de la variable cible. Les erreurs entre les valeurs prédites et vraies sont ensuite utilisées pour former un nouvel arbre de décision qui corrige les erreurs commises par le premier arbre. Ce processus est répété plusieurs fois, chaque nouvel arbre apprenant à corriger les erreurs commises par les arbres précédents. La prédiction finale est faite en combinant les prédictions de tous les arbres de la séquence.

L'idée clé derrière le gradient boosting est d'apprendre des erreurs des arbres précédents en ajoutant de nouveaux arbres qui se concentrent sur les erreurs restantes. L'algorithme utilise une méthode d'optimisation de descente de gradient pour trouver les poids optimaux pour chaque arbre de la séquence, de sorte que l'erreur globale soit minimisée. La méthode de descente de gradient met à jour les poids des arbres de la séquence en utilisant le gradient de la fonction de perte par rapport aux prédictions des arbres précédents. Cela garantit que l'algorithme converge vers la solution optimale, tout en évitant le sur-ajustement en pénalisant les poids des arbres complexes.

Gradient Boosting Classifier est un algorithme puissant qui peut gérer des ensembles de données complexes et de grande dimension. Il est robuste aux valeurs aberrantes et peut gérer les valeurs manquantes. Il a également la capacité d'apprendre des relations non linéaires entre les caractéristiques et la variable cible, et peut capturer des interactions complexes entre les caractéristiques. Cependant, il peut être sensible au choix des hyper-paramètres, tels que le taux d'apprentissage, la profondeur maximale des arbres et le nombre d'arbres dans la séquence. Un réglage correct des hyper-paramètres est essentiel pour obtenir de bonnes performances avec Gradient Boosting Classifier.

4.1.2 LightGBM (Light Gradient Boosting Machine)

LightGBM est un framework de renforcement de gradient qui utilise des algorithmes d'apprentissage basés sur des arbres. Il est conçu pour être efficace et évolutif, et est capable de gérer des ensembles de données à grande échelle avec des millions d'instances et de fonctionnalités. LightGBM est développé par Microsoft et est open source, disponible sous licence MIT.

LightGBM est basé sur le framework de renforcement de gradient Cependant, il introduit plusieurs optimisations et améliorations qui le rendent plus rapide et plus précis que les algorithmes traditionnels d'amplification de gradient. Ces optimisations incluent :

- Échantillonnage unilatéral basé sur le gradient (GOSS) : cette technique utilise une approche d'échantillonnage pondérée pour sélectionner les instances et les fonctionnalités les plus informatives pour la formation. Il réduit le coût de calcul et l'utilisation de la mémoire tout en améliorant la précision du modèle.

- DART (Dropout Additive Regression Trees) est un algorithme d'amplification de gradient qui étend le cadre standard d'amplification de gradient en ajoutant une régularisation d'abandon aux arbres individuels de l'ensemble. L'abandon est une technique de régularisation couramment utilisée dans les réseaux de neurones qui abandonne de manière aléatoire un sous-ensemble de neurones pendant l'entraînement. Ce faisant, il évite le sur-ajustement et améliore les performances de généralisation du réseau.

Dans DART, la régularisation de l'abandon est appliquée aux arbres individuels dans l'ensemble d'amplification du gradient. Au cours de la formation, chaque arbre est formé sur un ensemble de fonctionnalités et d'instances sous-échantillonnées de manière aléatoire, et un masque de suppression est appliqué aux nœuds feuilles de l'arbre pour supprimer de manière aléatoire un sous-ensemble de nœuds. Cela encourage l'arbre à apprendre des représentations multiples et diverses des données et empêche le sur-ajustement. La régularisation de l'abandon est appliquée uniquement pendant la formation et l'ensemble complet des arbres est utilisé pour la prédiction.

- Regroupement exclusif de fonctionnalités (EFB) : cette technique combine des fonctionnalités corrélées en faisceaux et les traite comme une seule fonctionnalité pendant la formation. Il réduit la dimensionnalité du jeu de données et améliore les performances de généralisation du modèle.

- Croissance de l'arbre dans le sens des feuilles : cette technique fait croître l'arbre en divisant les nœuds feuilles avec le gain le plus élevé, plutôt que les nœuds avec le gain le plus élevé d'une manière par niveau. Il réduit le nombre de divisions nécessaires pour atteindre la solution optimale et améliore la précision du modèle.

LightGBM prend également en charge plusieurs fonctions objectives, notamment la classification binaire, la classification multi-classe et la régression. Il peut gérer à la fois des données d'entrée clairsemées et denses et prend en charge l'accélération CPU et GPU.

Dans l'ensemble, LightGBM est un algorithme d'apprentissage automatique puissant et efficace qui peut gérer des ensembles de données à grande échelle avec une grande précision. Il est largement utilisé dans divers domaines, notamment le classement des recherches sur le Web, les

systèmes de recommandation et la classification des images.

4.1.3 SVC (Support Vector Machine for Classification)

SVC est un type d'algorithme d'apprentissage automatique utilisé pour les problèmes de classification binaire et multi-classes. L'algorithme fonctionne en trouvant l'hyperplan qui sépare le mieux les différentes classes dans les données d'entrée.

Dans un problème de classification binaire, l'hyperplan est une ligne qui sépare les deux classes. Dans un problème de classification multi-classes, l'hyperplan est un ensemble de lignes qui séparent les différentes classes. L'hyperplan est choisi de sorte que la distance entre l'hyperplan et les points de données les plus proches de chaque classe soit maximisée. Ces points les plus proches sont appelés vecteurs de support, et la distance entre l'hyperplan et les vecteurs de support est appelée la marge.

SVC peut être appliqué à la fois aux données séparables linéairement et non linéairement séparables. Dans le cas de données non linéairement séparables, l'algorithme utilise une fonction noyau pour transformer les données d'entrée dans un espace de dimension supérieure où les données peuvent être séparées par un hyperplan. C'est ce qu'on appelle l'astuce du noyau.

SVC a plusieurs hyperparamètres qui doivent être réglés pour des performances optimales, y compris le paramètre de régularisation C, le type de noyau et les paramètres de fonction du noyau. L'algorithme est efficace en termes de calcul, car il n'utilise qu'un sous-ensemble des données d'apprentissage pour trouver les vecteurs de support et l'hyperplan.

Il a été démontré que SVC atteint une grande précision sur de nombreux ensembles de données de référence et est largement utilisé dans diverses applications, notamment la classification d'images, la classification de textes et la bio-informatique.

4.2 Résultats

Après avoir essayer plusieurs modèles, ils sont très performants sur les données d'entraînements les accuracies sur les données test sont alentours de 80% sans une validation croisée.

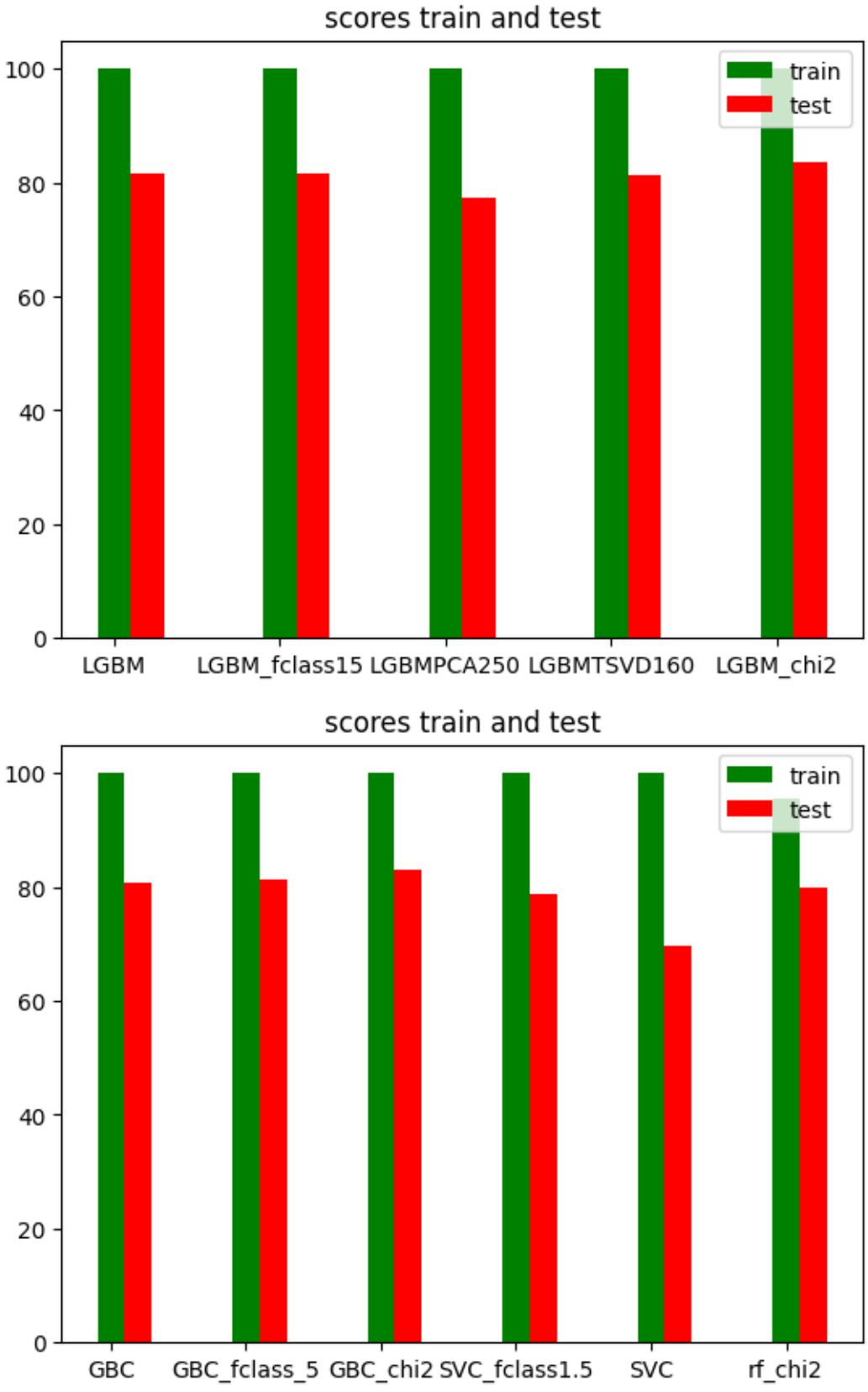

Figure 4: scores sur train et test

Après une validation croisée sur nos modèles constatons des améliorations les scores sur les données test et validation avec des valeurs d'environ 82% sur test et 84% sur validation pour la majorité des modèles.

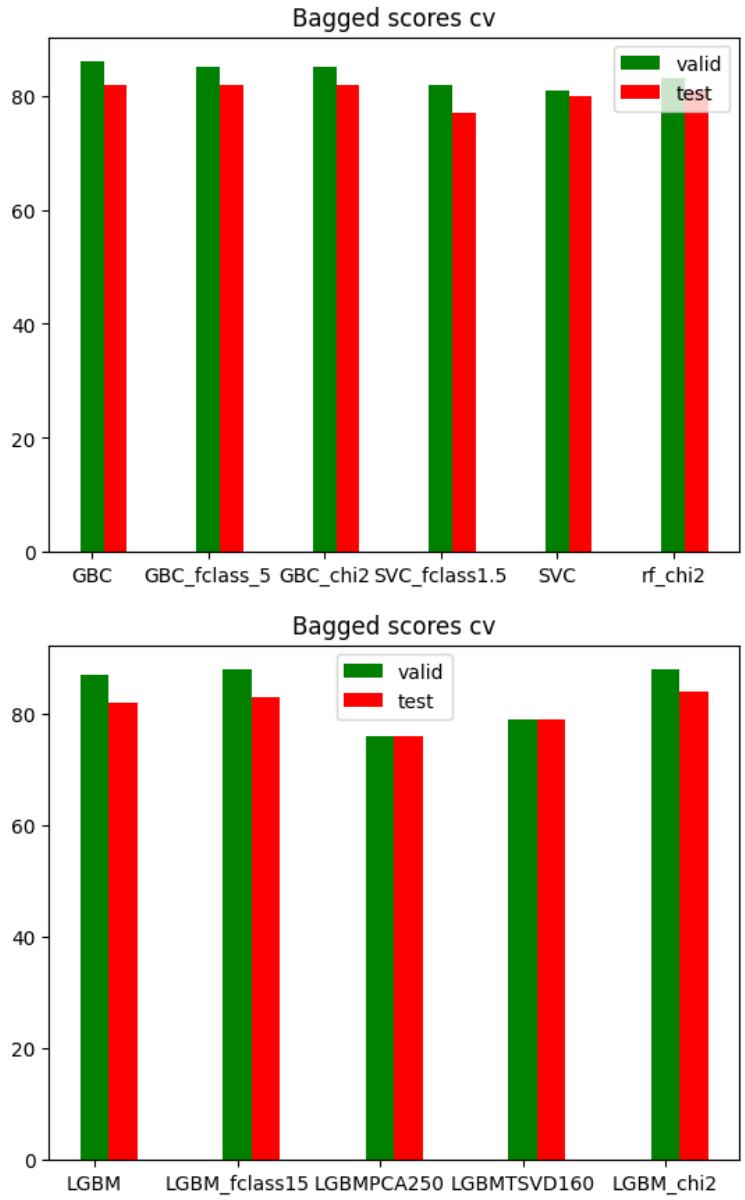

Figure 5: bagged scores sur valid et test

En terme de scores, nos meilleurs modèles après la validation croisée sont le gradient boosting classifier, gradient boosting avec f_classif et $chi2$, le Lightgbm sans et avec f_classif et $chi2$ qui ont respectivement des scores validation $86, 85, 85, 87, 88$ et 88. Et sur les données test nous avons $82, 82, 82, 82, 82, 83$ et 84. Donc le meilleur le modèle est celui de Lightgbm $chi2$ suivant le bagged score de la validation croisée qui enrégistre 88% sur validation et 84% sur test .

En terme de temps aussi, il est le meilleur comparer aux autres modèles cités ci dessus. Cependant, les SVC muni du f_classif et random forest muni chi2 ont des temps inférieurs respectivement 1.2 et 1.9 secondes mais en termes de bagged scores ils ont 81 et 83 sur validation et 80 et 81 sur test respectivement.

Modèles	GBC	GBC_fclass_5	GBC_chi2	$SVC\_fclass1.5$	SVC	rf_chi2
Times	493	58	55	1.2	40	1.8

Modèles	$LGBM$	$LGBM\_fclass15$	$LGBMPCA250$	$LGBMTSVD160$	$LGBM\_chi2$
Times	5.7	2.9	4.1	3.5	1.9

Figure 6: Mean time in cross cv

5 Conclusion

Dans cette étude, nous avons effectué une analyse de classification sur les données génétiques pour prédire quatre types de cellules : Cancer_cells, NK_cells, T_cells_CD4+ and T_cells_CD8+ . Notre analyse a montré que les classificateurs avec la sélection des variables SelectPercentile ont donné les meilleurs résultats, atteignant 88% mais aussi avec de très temps .

Dans l'ensemble, nos résultats suggèrent que les données génétiques peuvent être utilisées pour prédire le type de cellule, et qu'un classificateur gradient boosting avec sélection de caractéristiques est une méthode efficace pour cette tâche.

6 Annexe

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html
https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifie
html
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClass
html
https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html
https://github.com/ramp-kits/scMARK_classification