Auteurs: BAHEUX Mélina & DRAIA-NICOLAU Ondina - M2 Bio-informatique 2019-2020

Rapport de Machine Learning

1) Introduction (Data / Contexte)

Détecter les cancers chez les patients est devenu une mission importante de la médecine actuelle. Les nouvelles technologies permettent actuellement de récupérer un grand nombre de données et de nouvelles méthodes apparaissent pour analyser ces données et trouver une corrélation entre elles.
L'une des méthodes mathématiques utilisée est le machine learning. L'intérêt de cette méthode est de pouvoir entraîner un modèle sur des données pour qu'une fois qu'on lui donne d'autre données il soit capable de reconnaître ce pourquoi il a été entraîné.

Le jeu de données utilisé est composé de données RNA-seq de l'expression de certains gènes chez des patients atteints de tumeurs de type BRCA (Breast cancer), KIRC (Kidney Renal Clear Cell **Carcinoma**), COAD (Colon adenocarcinoma), LUAD (Lung Adenocarcinoma) ou PRAD (Prostate adenocarcinoma).

Le but de notre démarche consiste à intégrer ces données sur les tumeurs afin d'entraîner un modèle informatique à les reconnaître.
Pour cela nous avons utilisé diverses librairies Python, vues et utilisés en cours.

Versions des différents packages utilisés sous Python: Python = 3.7.4, Keras = 2.3.1, Numpy = 1.16.4, Tensorflow = 1.13.1

2) Choix de la programmation

Nous avons commencé l'exploration des données en créant deux modèles de base: un modèle employant la régression linéaire (vu en cours) et un autre utilisant SVM (support-vector machine).
Ensuite, nous avons utilisé la méthode des arbres de décision.

Sous *Python*, le code utilisé pour réaliser un modèle de type régression linéaire est le suivant (la fonction *linearModel*):

```
#Create a simple model
   logreg = linear_model.LogisticRegression(C=1e30, solver='liblinear')
   #Train the model
   logreg.fit(X_train,Y_train)
   Z = logreg.predict(X_test)
```

Le code utilisé pour réaliser le modèle SVM est le suivant (fonction *SVM*):

```
#Create a simple model
   # KNeighborsClassifier created
```

```
    model = svm.SVC(C=0.5, kernel='linear')
    #Train the model
    model.fit(X_train, Y_train)
```

Code utilisé pour réaliser un modèle de type arbre de décision utilisant l'algorithme RandomForest, l'outil *graphviz* nous a permis d'obtenir ensuite l'arbre de décisions associé (fonction *treeFunc*):

```
model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, Y_train)
label_names = np.array(['BRCA', 'COAD', 'KIRC', 'LUAD', 'PRAD'])
data_names = np.array(["Gene_"+str(i) for i in range(new_X.shape[1])])
estimator = model.estimators_[5]
dotfile = open("C:/Users/utilisateur/Documents/Cours/M2 DLAD
Bioinfo/Machine-Learning-Stats3-master/output/tree.dot", 'w')
tree.export_graphviz(estimator, out_file='output/tree.dot',
                feature_names = data_names,
                class_names = label_names,
                rounded = True, proportion = False,
                precision = 2, filled = True)
```

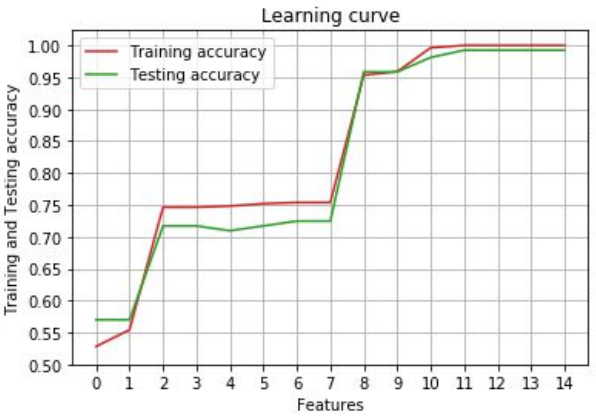
Suite aux résultats obtenus, nous avons décidé d'aller plus loin dans l'exploration des données en utilisant le modèle de la régression linéaire (fonction compTrainTest). Ainsi nous avons réalisé 5 graphes, 1 pour chaque type de cancer présent dans les jeux de données, pour les données test et pour les données de training.
Enfin, on a essayé de réalisé un modèle utilisant des réseaux de neurones artificiels afin de pouvoir tirer des conclusions satisfaisantes sur les données analysées, en utilisant le package keras (fonction *kerasNN*).
Ce mélange d'analyses va permettre à des caractéristiques particulières des données de ressortir.

3) Analyse des résultats

<u>Courbe obtenue pour l'apprentissage en utilisant la régression linéaire:</u>

- La précision de l'apprentissage des données de test a l'air toujours légèrement supérieure à celle des données de training. Nos données étant nombreuses, la régression linéaire est un modèle qui reste en underfitting constant mais reste intéressant à utiliser si les données ne sont pas trop complexes.

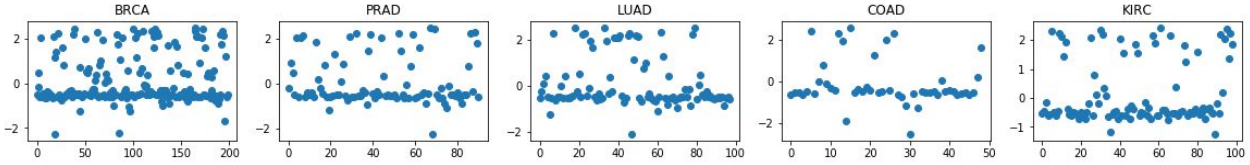
Courbe obtenue pour l'apprentissage en utilisant le SVM:



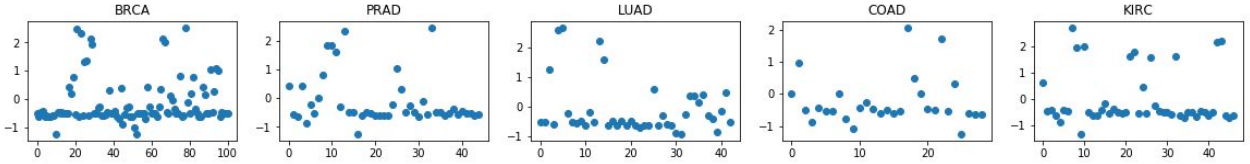

```
model = svm.SVC(C=100, kernel='linear')
```

```
model = svm.SVC(C=0.5, kernel='linear')
```

Comparaison entre la précision du modèle SVM et du modèle de régression linéaire:
Avec le modèle SVM il est possible d'avoir un exemple d'underfitting et de fitting normal. A gauche, en ayant utilisé 100 pour la valeur de C, on obtient des courbes traduisant un apprentissage correct des données. Dans le graphique de droite, on obtient du underfitting étant donné que les courbes de training et de test sont très rapprochées.
La valeur C est l'inverse de la puissance de régularisation. des petits valeurs représentent une forte régularisation.
➢ Nous sommes allées plus loin dans l'exploration des données en utilisant le modèle de régression linéaire. L'obtention de graphiques de type "nuage de points", pour chaque type de cancer présent dans le jeu de données, d'abord pour les données de training puis les données de test, nous a permis d'obtenir quelques caractéristiques particulières des données:
Graphique des données de training:



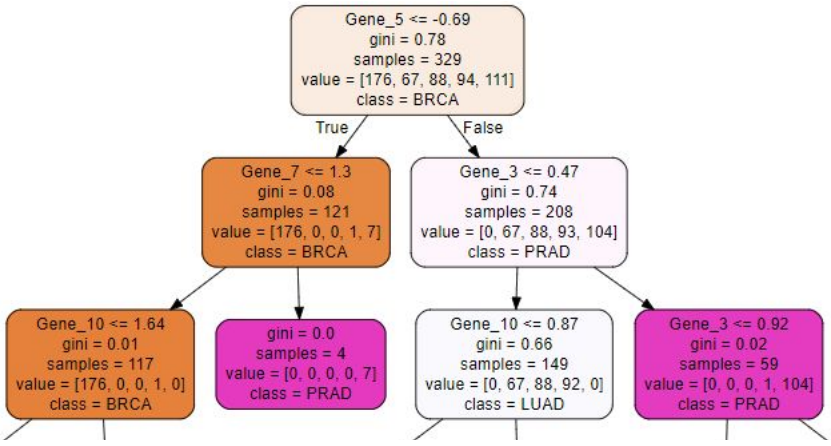
Graphique des données de test:



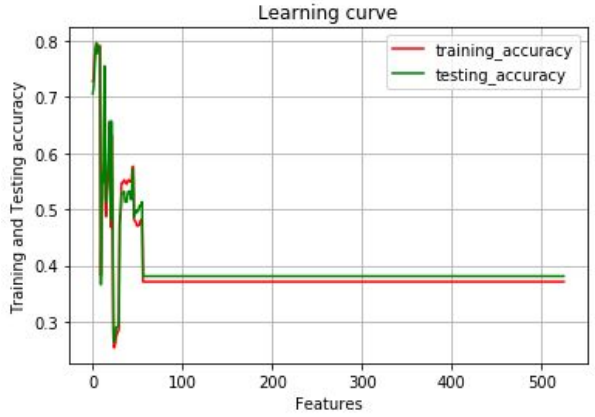
Le cancer BRCA, cancer du sein regroupe beaucoup plus d'individus, ce qui est normal étant donné que c'est le plus récurrent et qui se soigne le mieux.

On observe également une ligne constante aux environs de -0.5. Cela est un bruit obtenu à cause de données qui sont constantes dans le jeu de données. L'interpréteur marquait cette particularité de ces données en tant que warning.
Sur l'axe des ordonnées on a les valeurs d'expression des gènes, et sur l'axe des abscisses le nombre des gènes.
On observe une particularité pour chaque type de cancer, les gènes numérotées en 0 et 20 sont toujours exprimées fortement.

Arbres de décisions (méthode random forest) créé à partir des données:



Interprétation: Étant donné que le jeu de données possède une quantité très grande de gènes, il est montré ici, grâce à cet arbre de décisions quelle incidence de cancer il peut y avoir en partant du gène 5 lié au cancer du sein qui peut être actif ou pas. On peut voir en haut de la figure que si ce gène est actif, il y a une possibilité de développer un PRAD ou, avec moins de chances, un LUAD. Sinon, si le gène n'est pas fortement exprimé, la possibilité de développer un PRAD est plus haute que dans le cas précédent, et sinon, un LUAD.
Utilisation de méthodes plus avancées de machine learning, le module keras:



On observe qu'avec le modèle crée, la training accuracy et la testing accuracy suivent la même tendance: suivant les features la précision de l'apprentissage est forte puis à partir de 80 features, elle devient constante. Les données sont soient redondantes ou le modèle représente un underfitting clair de l'apprentissage des données par le modèle.

4) Conclusion

Le but de cet exercice était d'explorer quelques outils afin de découvrir les possibilités d'étude que nous offre le machine learning. Notre démarche s'est orientée vers la recherche d'informations dans les données plus que la compréhension des données elles même.

Vu la grande quantitée de données, nous nous sommes souvent retrouvé en underfitting. Pour régler le soucis, nous avons dû diminuer la régularisation et ajouter des features. Mis à part pour la comparaison avec le dot-plot, tous les types de cancer sont traité ensemble.

Il a aussi été intéressant d'utiliser les fonctions de Keras nous ayant permis de voir si un modèle neuronal pouvait apprendre ces données. Sachant qu'il a été possible pour l'algorithme utilisant des arbres de décision de trouver une certaine logique au niveau du lien entre les données, le réseau neuronal a pu donc apprendre d'une certaine mesure ces données. Mais, étant donné que les informations utilisées ne sont pas d'une grande complexité et probablement que les pourcentages d'incidence de cancers dus au même gène sont issus de coïncidences, le modèle n'est pas très précis pour apprendre les caractéristiques de ces données.

Il serait intéressant afin de bien optimiser l'utilisation des packages de faire des études sur des données plus complexes. Cela nous permettrait de bien comprendre l'étendue des possibilités de l'utilisation de Machine Learning.