Remerciements

Je tenais à remercier les tuteurs du projet métier Adil CHERGUI, Abdelali ZAKRANI, Ayoub BAHNASSE, Mustapha HAIN et Hicham MOUTACHAOUIK pour leurs conseils et leur aide, ainsi sans eux ce type de projets n’aurait probablement pas eu lieu à l’ENSAM Casablanca. En particulier, je m’adresse mes remerciements à M. Abdelali ZAKRANI qui a suivi de près mon avancement tout au long du projet.

Remerciements

J’adresse ensuite ma gratitude aux autres enseignants. Comment dépeindre l’apport considérable dont j’ai bénéficié, non seulement dans leur enseignement, mais aussi dans les conversations que nous avons eues. Je les remercie aussi pour la bienveillance et l’intérêt qu’ils manifestent à nous toutes et tous, étudiants.

Enfin, je tenais à exprimer ma très haute considération et mes vifs remerciements à tous les enseignants généreux qui préservent l’enseignement public et ses valeurs.

Table des matières

1. Présentation du sujet………………………………………………………
   1. Apprentissage automatique – Machine learning………………..
   2. Kaggle……………………………………………………………….
   3. Python……………………………………………………………….
   4. Choix du sujet………………………………………………………
   5. Cahier de charge…………………………………………………..
2. Travail réalisé……………………………………………………………..
   1. Étapes de résolution………………………………………………
   2. Environnement de travail…………………………………………
3. Rapport détaillé du travail……………………………………………….
   1. L’analyse exploratoire des données……………………………
      1. Analyse de la forme…………………………………………..
      2. Analyse du fond ……………………………………………...
   2. Le prétraitement des données………………………………….
   3. La modélisation ………………………………………………….

Conclusion……………………………………………………………….

1. Contexte du projet
   1. ***Apprentissage automatique – Machine learning***

Le machine learning est une sous-catégorie de l’intelligence artificielle. Il peut être vu comme l’implémentation d’algorithmes apprenant à partir de données. Celles-ci vont permettre à l’algorithme d’établir certaines propriétés sur ces données et ensuite de pouvoir résoudre différents problèmes, essentiellement de la classification, sur d’autres données. Les algorithmes de machine learning sont souvent divisés en deux catégories : les algorithmes d’apprentissage supervisé et ceux d’apprentissage non supervisé. On parle d’apprentissage supervisé lorsque les classes d’appartenance des données sont connues à l’avance, et d’apprentissage non supervisé lorsque ni leur nombre ni leur nature ne sont connus et vont être déterminés par l’algorithme lui-même. Ce projet concerne l’apprentissage supervisé.

* 1. ***Kaggle***

Kaggle est un site internet communautaire basé sur le data science. Il est possible de créer des compétitions avec récompenses à la clé (de l’argent ou des contrats d’embauche principalement). Le commanditaire de la compétition (souvent une entreprise) soumet un problème, composé d’une description et d’un ou plusieurs jeux de données, sur Kaggle, et le site s’occupe d’organiser une compétition autour de ce problème. Les participants choisissent donc les challenges auxquels ils veulent participer et tentent de résoudre le problème soumis de la meilleure façon possible, tout cela en compétition avec les autres participants. Pour cela, on a la possibilité de soumettre un fichier afin d’obtenir un score qui nous positionne dans un classement publique (qui regroupe tous les participants). Il est donc possible, à tout moment, de savoir qui sont les premiers, et de comparer son propre score à celui des autres. Le site tient à avoir un aspect communautaire et encourage vivement tous les participants à se rendre sur le forum (système de discussion asynchrone) associé à la compétition. On peut trouver sur celui-ci de nombreux utilisateurs détaillant leurs problèmes rencontrés, leurs solutions ou tout simplement des informations qui permettent d’aider à avancer dans la résolution du problème. A la fin de la compétition, le classement est bloqué et les récompenses sont distribuées (souvent au trio du podium). Il est tout de même possible de continuer à soumettre des résultats et d’être noté pour une compétition finie si on désire encore travailler et améliorer sa solution.

* 1. ***Python***

J’ai décidé d’utiliser le langage Python pour implémenter nos algorithmes. Celui-ci est doté de deux avantages importants pour ce projet. Le premier est qu’il est exécutable à la volée et très léger dans sa syntaxe, ce qui permet de coder et de debugger très rapidement. Le second est qu’en tant que langage de programmation fonctionnelle, on peut très facilement faire appel à des bibliothèques implémentant déjà les algorithmes utilisés. En l’occurrence, j’ai employé la bibliothèque scikit-learn qui fournit la plupart des algorithmes de machine learning. J’ai opté pour la version Python 3.9. En effet, la version Python 3 présente certaines améliorations pour des packages. J’ai utilisé l’ensemble de packages fournis par Anaconda, une distribution de Python : ceux-ci comprennent Python.3.8, toute la bibliothèque scikit-learn ainsi que l’éditeur de texte Jupyter adapté au langage Python. Scikit-learn est un site open-source qui propose sa propre bibliothèque et de nombreux codes illustrant l’utilisation de ses packages. On y retrouve aussi une liste exhaustive des algorithmes proposés pour le machine learning. Celle-ci m’a donné de nombreuses pistes pendant le projet. En 2015, l’IEEE (Institut des ingénieurs électriciens et électroniciens) publie une étude qui montre que Python est le troisième langage le plus demandé par les entreprises (étude réalisée aux États-Unis). Il m’a donc semblé judicieux d’apprendre ce langage afin d’étoffer mon CV. D’un point de vue fonctionnalité, certains packages Python sont codés en C et sont donc plus rapides en exécution, et nous permettent de gagner un temps non-négligeable pour certains calculs.

* 1. ***Choix du sujet***

L'Organisation mondiale de la santé (OMS) a qualifié le COVID-19, causé par le SRAS-CoV-2, de pandémie le 11 mars, alors que l'augmentation exponentielle du nombre de cas risquait de submerger les systèmes de santé du monde entier avec une demande pour les lits de soins intensifs bien au-dessus de la capacité existante, les régions d'Italie étant des exemples marquants.

Le Brésil a enregistré le premier cas de SRAS-CoV-2 le 26 février, et la transmission du virus est passée de cas importés uniquement à une transmission locale et enfin communautaire très rapidement, le gouvernement fédéral déclarant une transmission communautaire à l'échelle nationale le 20 mars.

Jusqu'au 27 mars, l'État de São Paulo avait enregistré 1 223 cas confirmés de COVID-19, avec 68 décès liés, tandis que le comté de São Paulo, avec une population d'environ 12 millions d'habitants et où se trouve l'hôpital Israelita Albert Einstein, en comptait 477 cas confirmés et 30 décès associés, au 23 mars. L'État et le comté de São Paulo ont décidé d'établir des mesures de quarantaine et de distanciation sociale, qui seront appliquées au moins jusqu'au début avril, dans le but de ralentir la propagation du virus.

L'une des motivations de ce défi est le fait que dans le contexte d'un système de santé débordé avec la limitation possible d'effectuer des tests pour la détection du SRAS-CoV-2, tester chaque cas serait impraticable et les résultats des tests pourraient être retardés même si seule une sous-population cible serait testée.

* 1. ***Cahier de charge***

1. **Détails de la tâche :** Prédire les cas confirmés de COVID-19 parmi les cas suspects.

Sur la base des résultats des tests de laboratoire couramment collectés pour un cas suspect de COVID-19 lors d'une visite aux urgences, serait-il possible de prédire le résultat du test pour le SRAS-Cov-2 (positif/négatif) ?

1. **Évaluation :**

Il ne s'agit pas d'un concours formel, nous ne mesurerons donc pas les résultats strictement par rapport à un ensemble de validation donné à l'aide d'une métrique stricte. Au contraire, ce que nous aimerions voir, c'est un processus bien défini pour construire un modèle qui peut fournir des résultats décents (évalués par vous-même).

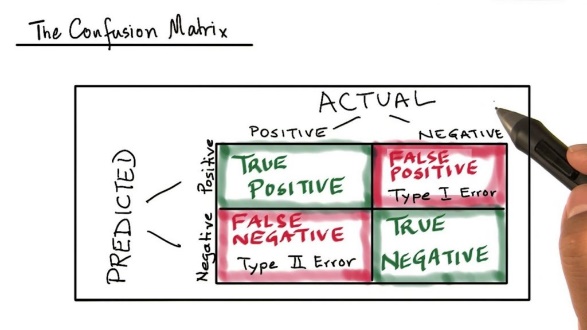
Notre équipe examinera:

1. ***Performances du modèle*** - Dans quelle mesure le modèle fonctionne-t-il sur les données réelles ? Peut-il être généralisé dans le temps ? Peut-il s'appliquer à d'autres scénarios? Était-ce surdimensionné?
2. ***Préparation des données***- Dans quelle mesure les données ont-elles été analysées avant de les intégrer au modèle ? Existe-t-il des visualisations utiles ? Le lecteur apprend-il de nouvelles techniques grâce à cette soumission ? Une excellente entrée sera informative, stimulante et fraîche à la fois.
3. ***Documentation***- Votre code, votre bloc-notes et vos sources de données supplémentaires sont-ils bien documentés afin qu'un lecteur puisse comprendre ce que vous avez fait ? Vos sources sont-elles clairement citées ? Une analyse de haute qualité doit être concise et claire à chaque étape afin que la justification soit facile à suivre et que le processus soit reproductible.
4. Travail réalisé
   1. ***Étapes de résolution :***
   2. **Définir un objectif mesurable :**

C’est une étape très importante qui vient avant de se pencher vers l’analyse et le traitement des données présentes dans le dataset, alors il pourra y avoir du hasard dans l’exploration. Bref, rien n’est scientifique. Or, ce projet se base sur la « Data Science ». Autrement dit, il faut être organisé et avoir une démarche scientifique. Donc pour commencer, il faut fixer un objectif spécifique.

Alors pour le cas de ce projet :

* **Objectif :** Prédire si une personne est infectée en fonction des données cliniques disponibles.
* **Métrique :** Pour cette mesure, l’ordinaire est de choisir l’«Accuracy» comme métrique et poser comme objectif d’obtenir un minimum de 90% comme valeur de l’exactitude, c’est-à-dire sur 100 prédictions, nous avons raison 90% du temps. Par contre, à cause du déséquilibre entre les catégories dans ce genre de dataset, nous avons le choix entre 3 autres métriques :
* ***Précision (Precision)*** : Elle permet de répondre à la question suivante, ‘’Quelle proportion d’identifications positives était effectivement correcte ?’’
* ***Rappel (Recall) :*** Il permet de répondre à la question suivante, ‘’Quelle proportion de résultats positifs a été identifiée correctement ?’’



Étant donnée une matrice de confusion, les deux métriques précédentes peuvent être calculées mathématiquement comme suivant :

* **Précision et rappel : un compromis permanent**

Pour évaluer les performances d'un modèle de façon complète, vous devez analyser **à la fois** la précision et le rappel. Malheureusement, précision et rappel sont fréquemment en tension. Ceci est dû au fait que l'amélioration de la précision se fait généralement au détriment du rappel et réciproquement.

* A cause du compromis cité au-dessus, la 3ème métrique est bien introduite nommée F1-score : Le F1-Score combine subtilement la précision et le rappel. Il est intéressant et plus intéressant que l’accuracy car le nombre de vrais négatifs (FP) n’est pas pris en compte.

🡺 Alors, je fixe l’objectif à F1-score 🡪 50% et Recall 🡪 70%

* 1. **EDA (Exploratory Data Analysis)**

En [statistiques](https://en.wikipedia.org/wiki/Statistics) , **l'analyse exploratoire des données** est une approche consistant à [analyser](https://en.wikipedia.org/wiki/Data_analysis)[des ensembles de données](https://en.wikipedia.org/wiki/Data_set) pour résumer leurs principales caractéristiques, souvent à l'aide de [graphiques statistiques](https://en.wikipedia.org/wiki/Statistical_graphics) et d'autres méthodes de [visualisation de données](https://en.wikipedia.org/wiki/Data_visualization) . Un [modèle statistique](https://en.wikipedia.org/wiki/Statistical_model) peut être utilisé ou non, mais l'EDA sert principalement à voir ce que les données peuvent nous dire au-delà de la tâche de modélisation formelle ou de test d'hypothèses. L'analyse exploratoire des données a été promue par [John Tukey](https://en.wikipedia.org/wiki/John_Tukey) pour encourager les statisticiens à explorer les données et éventuellement à formuler des hypothèses qui pourraient conduire à de nouvelles collectes de données et expériences. L'EDA est différente de [l'analyse initiale des données (IDA)](https://en.wikipedia.org/wiki/Data_analysis#Initial_data_analysis) , [[1]](https://en.wikipedia.org/wiki/Exploratory_data_analysis#cite_note-1)qui se concentre plus étroitement sur la vérification des hypothèses requises pour l'ajustement du modèle et les tests d'hypothèses, ainsi que sur la gestion des valeurs manquantes et la transformation des variables selon les besoins. L'EDA englobe l'IDA.

* 1. ***Pre-processing***

Le prétraitement des données est une étape importante dans le processus d' [exploration de données](https://en.wikipedia.org/wiki/Data_mining) . L'expression [« garbage in, garbage out »](https://en.wikipedia.org/wiki/GIGO) est particulièrement applicable aux projets d' [exploration de données](https://en.wikipedia.org/wiki/Data_mining) et d' [apprentissage automatique](https://en.wikipedia.org/wiki/Machine_learning) . [Les](https://en.wikipedia.org/wiki/Data_collection) méthodes de [collecte de données](https://en.wikipedia.org/wiki/Data_collection) sont souvent mal contrôlées, ce qui entraîne [des](https://en.wikipedia.org/w/index.php?title=Range_error&action=edit&redlink=1) valeurs [hors limites](https://en.wikipedia.org/w/index.php?title=Range_error&action=edit&redlink=1) (par exemple, Revenu : -100), des combinaisons de données impossibles (par exemple, Sexe : Homme, Enceinte : Oui) et [des valeurs manquantes](https://en.wikipedia.org/wiki/Missing_values) , etc. n'a pas été soigneusement examiné pour de tels problèmes peut produire des résultats trompeurs. Ainsi, la représentation et la [qualité des données](https://en.wikipedia.org/wiki/Data_quality) passent avant tout avant toute analyse. [[1]](https://en.wikipedia.org/wiki/Data_pre-processing#cite_note-1) Souvent, le prétraitement des données est la phase la plus importante d'un projet d' [apprentissage automatique](https://en.wikipedia.org/wiki/Machine_learning) , en particulier en [biologie computationnelle](https://en.wikipedia.org/wiki/Computational_biology) . [[2]](https://en.wikipedia.org/wiki/Data_pre-processing#cite_note-2)

S'il y a beaucoup d'informations non pertinentes et redondantes présentes ou de données bruitées et peu fiables, alors [la découverte des connaissances](https://en.wikipedia.org/wiki/Knowledge_discovery) pendant la phase de formation est plus difficile. Les étapes de préparation et de filtrage des données peuvent prendre un temps de traitement considérable. Le prétraitement des données comprend le [nettoyage](https://en.wikipedia.org/wiki/Data_cleaning) , la [sélection d' instances](https://en.wikipedia.org/wiki/Instance_selection) , la [normalisation](https://en.wikipedia.org/wiki/Data_normalization) , la [transformation](https://en.wikipedia.org/wiki/Data_transformation) , l' [extraction](https://en.wikipedia.org/wiki/Feature_extraction) et la [sélection de](https://en.wikipedia.org/wiki/Feature_selection)[caractéristiques](https://en.wikipedia.org/wiki/Feature_extraction), etc . Le produit du prétraitement des données est l' [ensemble d' apprentissage](https://en.wikipedia.org/wiki/Training_set) final .

Le prétraitement des données peut affecter la manière dont les résultats du traitement final des données peuvent être interprétés. [[3]](https://en.wikipedia.org/wiki/Data_pre-processing#cite_note-3) Cet aspect doit être soigneusement pris en compte lorsque l'interprétation des résultats est un point clé, comme dans le traitement multivarié de données chimiques ([chimiométrie](https://en.wikipedia.org/wiki/Chemometrics)).

* 1. ***Modelling***

##### **Une fois les données prêtes, il sera possible de se lancer dans la modélisation.**[Scikit-Learn](https://scikit-learn.org/stable/)**met à disposition un multitude de méthodes de régression, de classification et d’ensemble.**Le choix du modèle est évidemment à réaliser en fonction de la problématique donnée.

##### **Il sera sans doute nécessaire de se replonger dans la première étape en élucidant la question sur la nature du problème. Ensuite évidemment il n’existe pas un unique algorithme de régression ou de classification. Il existe deux possibilités:**

##### **Tous les tester et prendre le plus performant (Sans doute trop coûteux)**

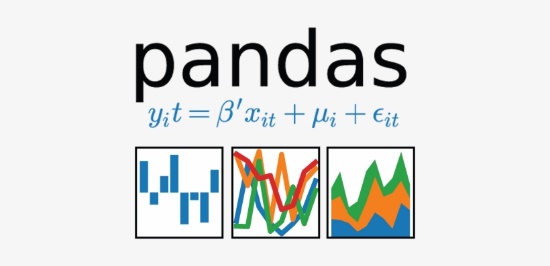
##### **Arbitrer sur celui à tester en fonction de vos données et des ressources à disposition**

##### **Une fois que le modèle est choisi, se posera la question du paramétrage : comment optimiser les paramètres de l’algorithme pour limiter l’overfitting ?  Envisager une recherche par quadrillage peut-être une solution mais elle peut se révéler également coûteuse en temps suivant vos ressources.**

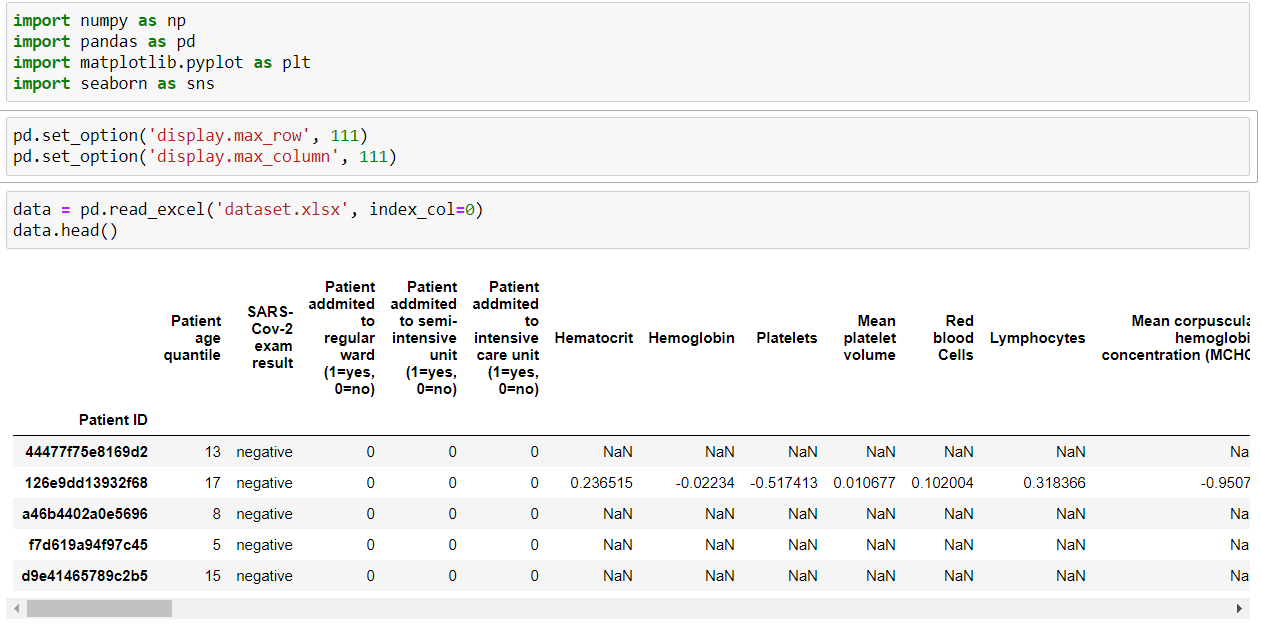
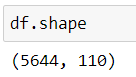
* 1. ***Environement de travail***

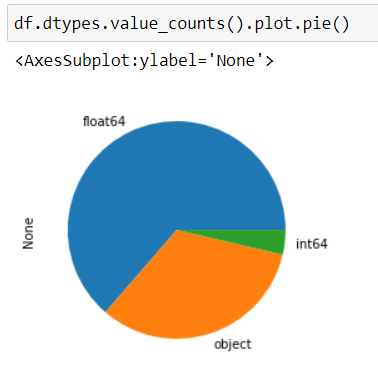






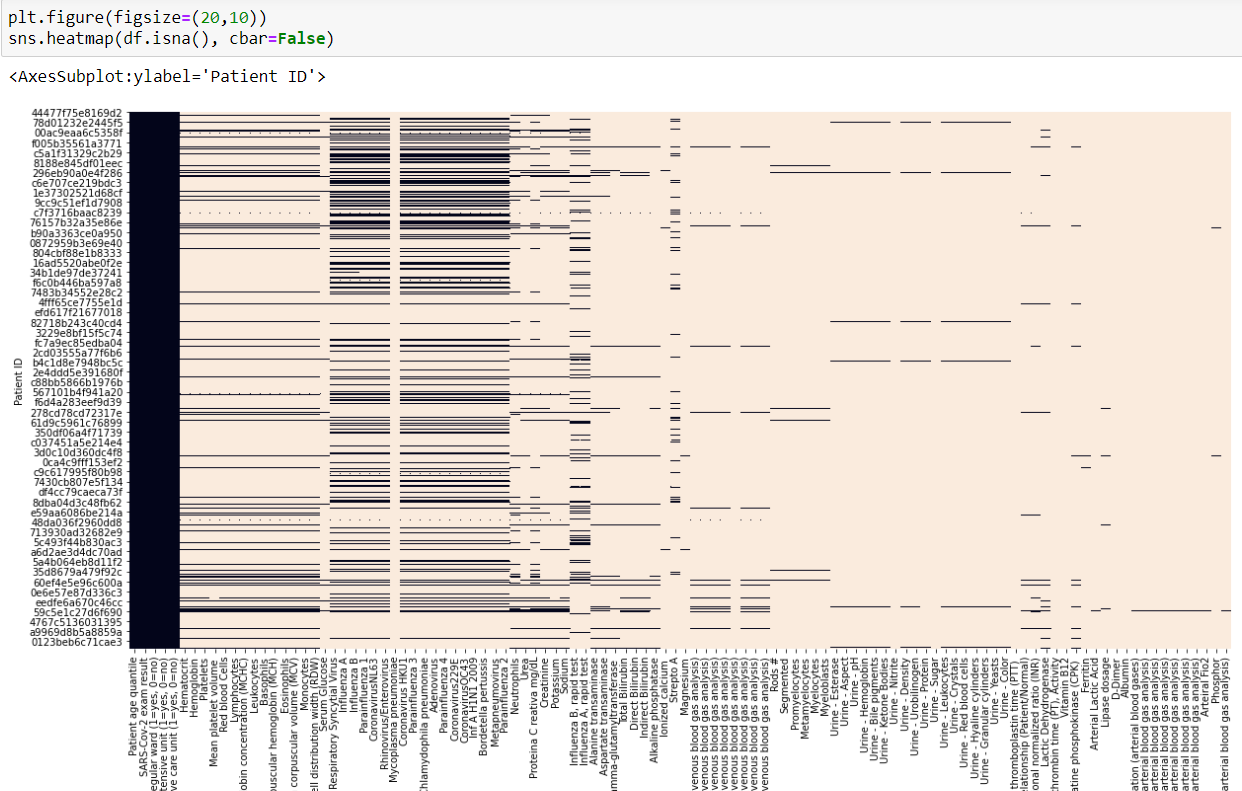
1. Rapport détaillé du travail
   1. ***L’analyse exploratoire des données***

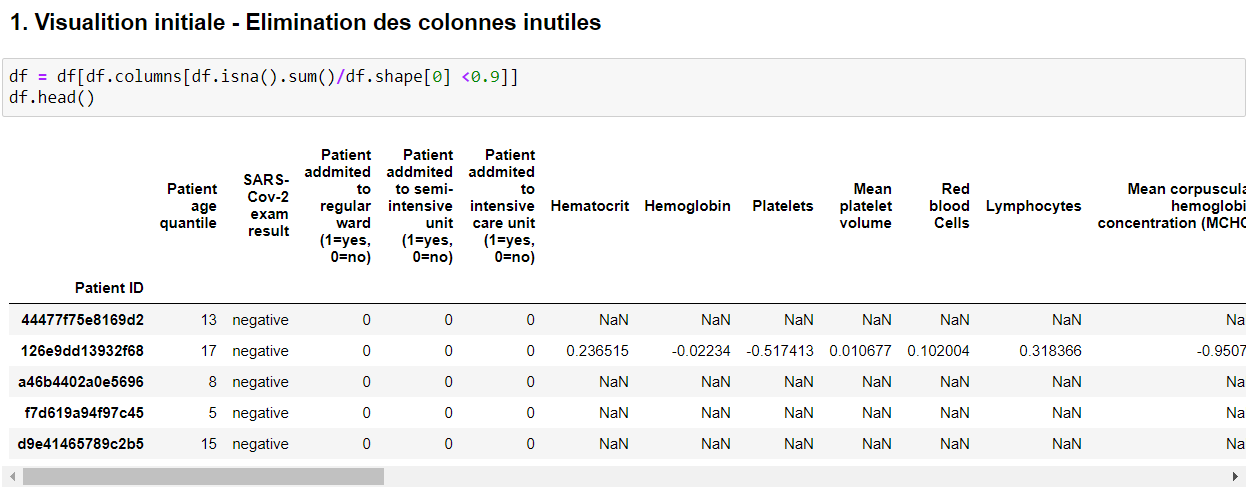
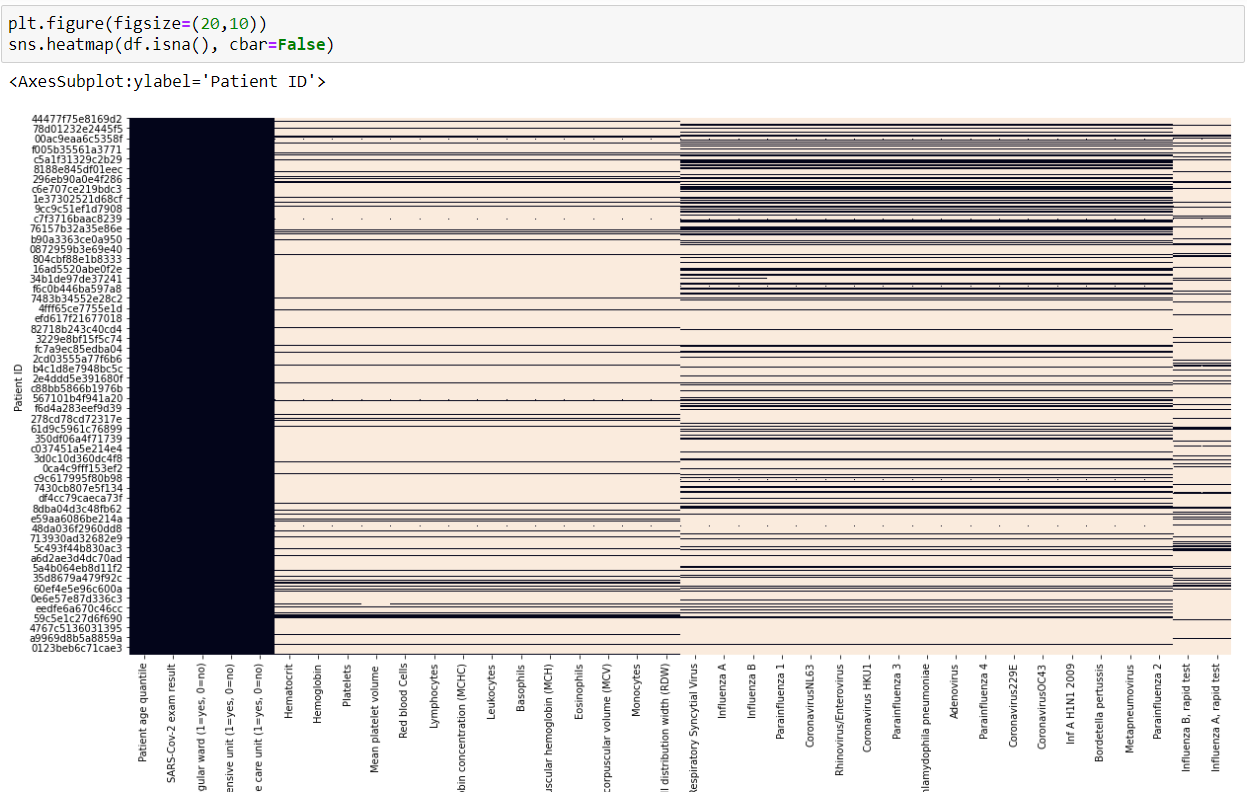
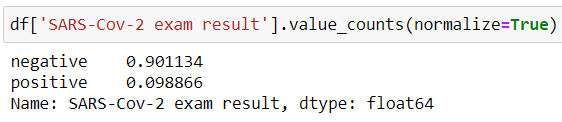
* Objectif : Comprendre au maximum les données dont on dispose pour définir une stratégie de modélisation.
* Évidemment, la première étape à faire est l’importation des librairies essentielles pour la lecture (pandas, numpy) et la visualisation des données (matplotlib, seaborn). Puis nous stockons la base de données dans une variable à l’aide de la fonction « read\_excel » de pandas.
  + 1. Analyse de la forme :
* ***Identification de la target:*** A partir de l’étape précédente, la variable target est bel et bien « SARS-Cov-2 exam result ».
* ***Nombre de lignes et de colonnes :*** Tel que df = data.copy(). Alors, on déduit que la base de données contient 5466 lignes et 110 colonnes.
* ***Types de variables :***

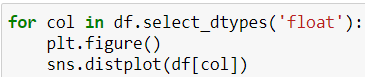
****** + Qualitatives : 70

+ Quantitatives : 41

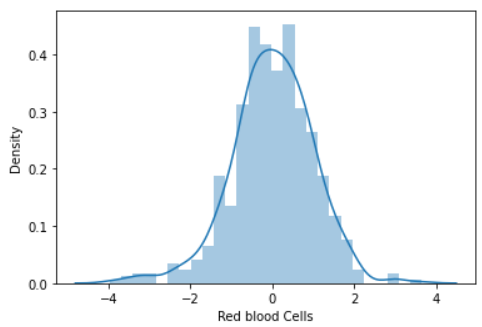
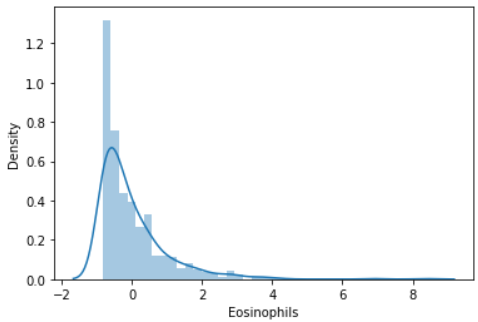
* ***Identification des valeurs manquantes :***

******

* Lors de la première observation, il parait que la base de données contient trop de valeurs manquants pour certaines variables (taux dépasse 90%). Ainsi vu la linéarité de la représentation de quelques attributs, on peut poser l’hypothèse qu’il existe une relation et un rapport entre ces attributs.
* Après l’exécution de la commande :
* Beaucoup de NaN (moitié des variables > 90% de NaN)
* 2 groupes de données : 76% -> Test viral, 89% -> taux sanguins
  + 1. Analyse du fond
* ***Élimination des attributs à beaucoup de valeurs manquantes :***
* ***Visualisation de la target :***
* Problème « Unbalanced Dataset » : valeurs non équilibrées donc je ne vais pas utiliser la mesure de performance "accuracy", plutôt "F1-score ".
* 10% de cas positifs (558 / 5000)
* ***Compréhension des différentes variables :***
  1. Visualisation des variables continues

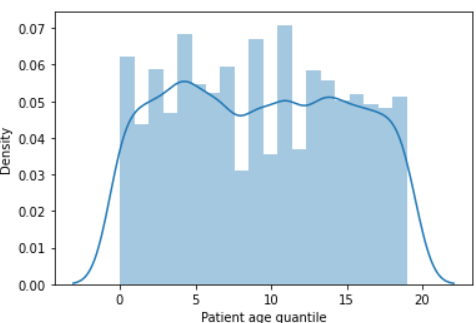


La fonction « sns.distplot(col) » permet de tracer la distribution des colonnes.



La plupart des variables suivant des distributions normales || D’autres suivent des distributions asymétriques

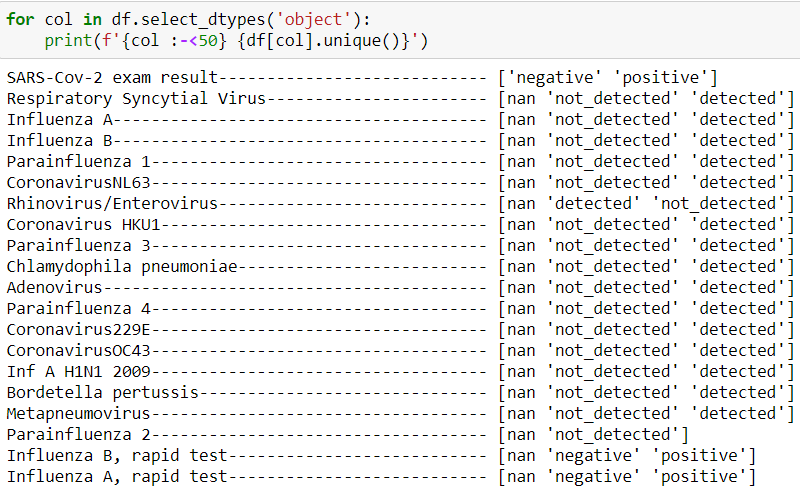
* Le point commun entre la totalité des courbes est qu’elles sont centrées à 0 et ont un écart-type=1, donc les données sont déjà standardisées.



* La variable « Patient age quantile » :

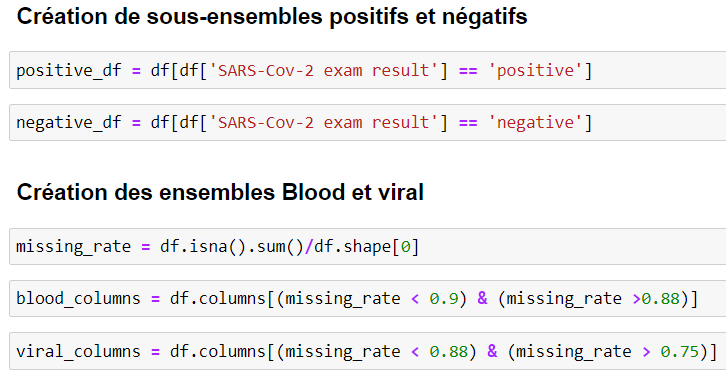


* Je remarque que la valeur maximale de cette variable est 20. Alors c’est difficile d'interpréter ce graphique. Clairement ces données ont été traitées, on pourrait penser qu’elles représentent les tranches d’âge, mais cela pourrait aussi être une transformation mathématique. On peut pas savoir car la personne qui a mis ce dataset n’a pas précisé la signification nulle part. Mais cela n'est pas très important
  1. Examen des variables qualitatives

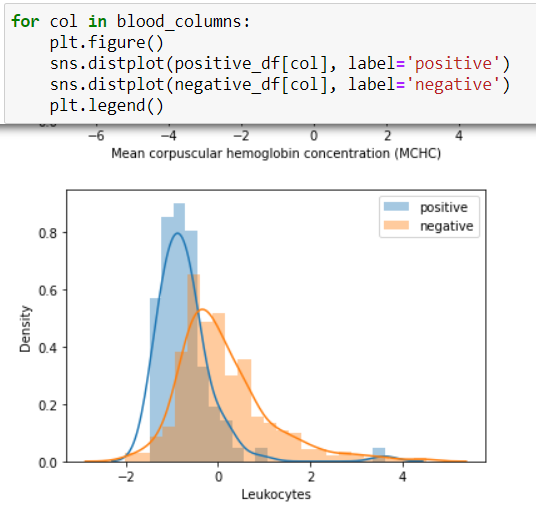


* + La totalité des variables qualitatives ont des valeurs binaires (0,1).
  + Elles sont divisées en 2 groupes : la variable Target (SARS-Cov-2 exam result) et les variables des tests viraux.
  + L’attribut « Parainfluenza 2 » possède une seule valeur, donc à éliminer après.
* ***Visualisation des relations features-target***

Afin de faciliter l’analyse pour après, il sera mieux de diviser les features et target en des sous-ensembles, pour bien visualiser la relation entre les catégories par paires des variables.



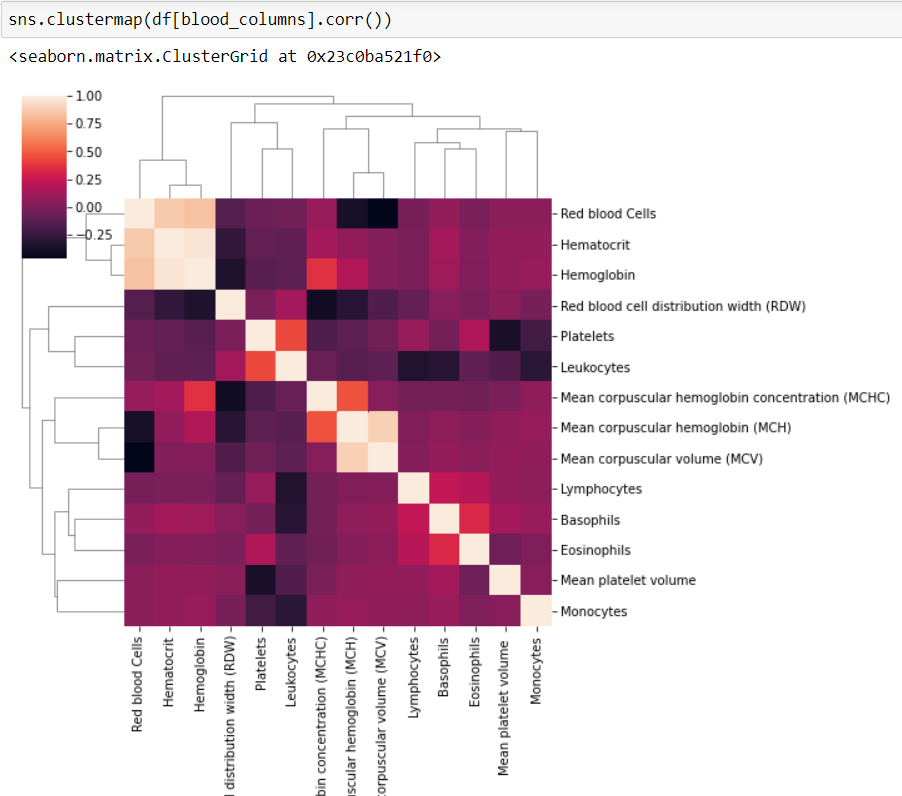
* + 1. **Target / Blood**



* + J’essaie de diviser les valeurs positives et négatives du test Covid afin de visualiser la distribution des taux sanguins chez chaque catégorie.
  + Ce qui a mené à la remarque suivante : « Les taux de Monocytes, Platelets et Leukocytes semblent liés au covid-19, à cause de la différence entre les deux courbes de ces taux sanguins » -> hypothèse à tester.
  1. **Target / Viral**



* Les doubles maladies sont très rares.
* Rhinovirus/Enterovirus positif, alors covid-19 négatif ? -> hypothèse à tester ? mais il est possible que la région est subie une épidémie de ce virus. De plus on peut très bien avoir 2 virus en même temps. Tout ça n'a aucun lien avec le covid-19.
* ***Conclusions initiales :***
* Beaucoup de données manquantes (au mieux on garde 20% du dataset)
* Groupes de données intéressantes (viral,sanguin)
* Presque pas de variable "discriminante" pour distinguer les cas positifs/négatifs, ce qui nous permet de dire qu'il n'est pas vraiment approprié de vouloir prédire si un individu est atteint du Covid-19 en se basant sur ces simples tests sanguins. Mais ce n'est pas grave, il faut quand même poursuivre l'analyse pour essayer de voir ce qu'on peut apprendre. Rome ne s'est pas faite en un jour, et le ML ne peut pas transformer l'eau en vin. Mais ça n'est pas une raison pour abandonner le projet. Un bon data scientist il va jusqu'au bout de sa mission. Si en cours de route, on se rend compte que l'histoire n'est pas aussi simple qu'on pouvait l'espérer, alors il faut pouvoir le démontrer de façon robuste en allant jusqu'au bout des choses.
* Donc maintenant quand même quelque chose de positif: On a pu identifier des variables intéressantes qui sont susceptibles de jour un rôle non négligeable (monocytes, etc)
  + 1. Analyse plus détaillée :
* ***Relations Variable / Variable***
  + 1. **Relation entre taux sanguins**

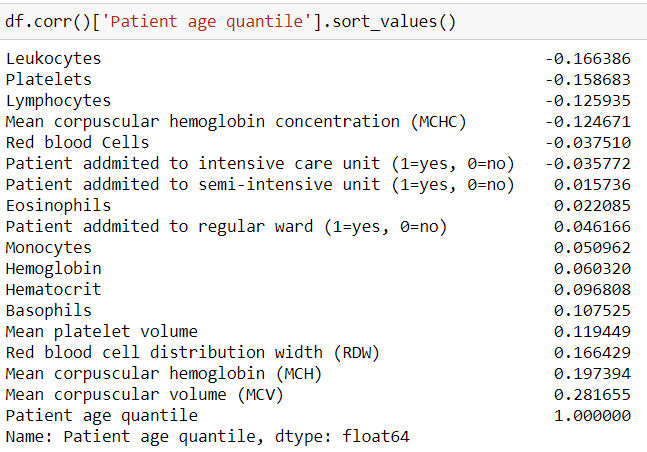
****

+ Pour vérifier la nature de la relation entre deux variables quantitatives, la plus célèbre méthode est de calculer la corrélation signifiant l’ajustement d’une variable par rapport à l’autre par une relation [affine](https://fr.wikipedia.org/wiki/Fonction_affine) obtenue par [régression linéaire](https://fr.wikipedia.org/wiki/R%C3%A9gression_lin%C3%A9aire).

+ Plus la corrélation est proche de 1, plus les deux variables évoluent positivement les unes avec les autres, quand l’une augmente, l’autre augmente aussi, comme le schéma à gauche nous montre (exemple de Hematocrit et Hemoglobin).

+ La fonction clustermap rassemble directement les variables et les organise d’une manière à former des clusters.

* + 1. **Relation Age / Sang :**

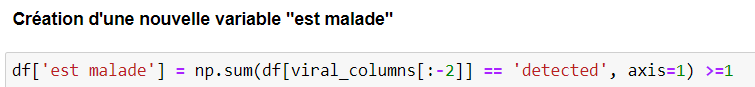


* Très faible corrélation entre l’âge et les attributs du sang.
  + 1. **Relation viral / viral**

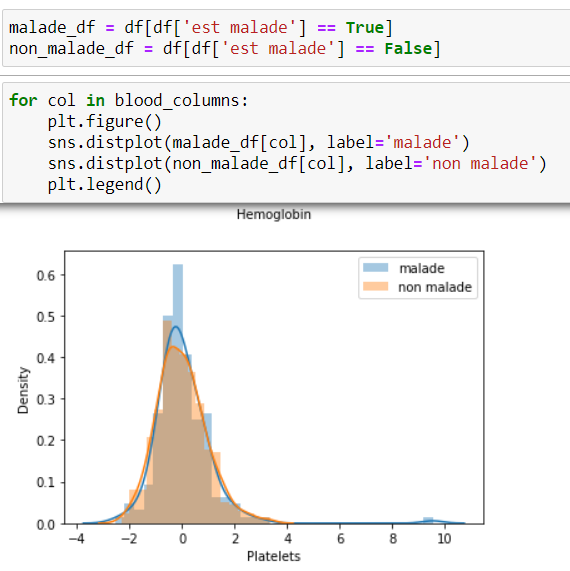
Pour cette partie, j’ai fait attention aux deux variables dont je n’ai pas parlé depuis le début. Et cela concerne la variable « Rapid Test des deux types d’Influenza ». Grâce à un crosstab entre la variable d’influenza et son rapid test correspondant, ainsi à l’aide d’une petite recherche sur google, il s’apparaît que ce test n’est pas vraiment fiable, vu que sa sensibilité est juste proche de 50% ou 55%. En addition, les résultats obtenus par mon analyse ont montré qu’un rapid test peut fournir un résultat positif pour des personnes non atteintes par l’Influenza.

* + 1. **Relation Viral / Sang**

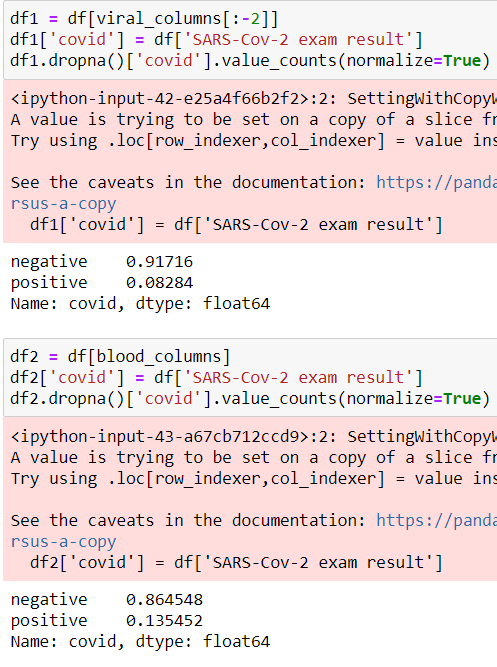
Pour simplifier la tâche, il est essentiel de constater qu’une personne atteinte par un virus est appelée une personne malade, par N’IMPORTE QUEL VIRUS. Alors il sera intelligent de combiner tous les attributs viraux dans une seule variable nommée ‘est malade’ pour réduire un peu la taille des données à étudier la relation avec d’autres comme suivant :



De la même façon à traiter la relation target / sang, dont j’ai bien divisé la target en sous-ensembles positifs et négatifs, je ferai de même avec la variable « est malade », ainsi que je vais étudier la relation par un « distplot » pour voir s’il y a une différence des taux sanguins entre les gens malades (autre que Covid) et les gens atteints par le coronavirus.



* + C’était vraiment le cas déjà mentionné. J’ai constaté que les taux sanguins qui étaient différents chez les gens positifs ou négatifs à Coronavirus, ne sont pas identiques à ceux qui sont différent chez les malades et non malades, comme les Platelets (voir figure à gauche), et j’ai trouvé le taux des Lymphocytes est bien différent dans ce cas. Alors tout cela reste une idée, une hypothèse à vérifier par un (Test Student).
* ***L’analyse des valeurs manquantes :***

Cette partie a pour but de savoir, à titre informatif, qu’est-ce qui se passerait dans notre dataset si j’élimine certaines valeurs manquantes. Le problème demeure en le fait que nos valeurs ne sont pas alignées (selon le 1er heatmap), alors si on essaye de supprimer les NaN values, il nous restera que très peu de valeurs (99 valeurs), ce qui n’est pas du tout pratique et ne servira à rien.

Une autre alternative qui puisse apparaître est de supprimer les attributs ayant moins de données, qui sont – dans ce cas – les attributs de la variable ‘Sang’ (603 vs 1354 pour la variable ‘Viral’). D’où, je peux déduire qu’il sera mieux d’éliminer les taux sanguins. Néanmoins, l’EDA a bien montré que la variable ‘Sang’ est un Feature très important.

Il nous reste à savoir l’état de la Target si on décide d’éliminer les valeurs manquantes d’un groupe. Pour le groupe ‘Viral’, le rapport des valeurs négatives et positives est un peu près identique à la proportion dans le dataset original. Faisons la même chose à la variable ‘Sang’, il est remarquable qu’on obtient des meilleures proportions, les classes sont plus balancées.

🡺 Viral : 1350(92/8), blood : 600(87/13).

* ***T-Test***

- Définition :

En [statistique](https://fr.wikipedia.org/wiki/Statistique), le **test de Student**, ou **test *t***, est un **test statistique** permettant de **comparer les moyennes** de deux groupes d’échantillons. Il s’agit donc de savoir si les moyennes des deux groupes sont significativement différentes au point de vue **statistique**. C’est un ensemble de [tests statistiques](https://fr.wikipedia.org/wiki/Test_d%27hypoth%C3%A8se) paramétriques où la [statistique de test](https://fr.wikipedia.org/wiki/Statistique_de_test) calculée suit une [loi de Student](https://fr.wikipedia.org/wiki/Loi_de_Student) lorsque l’[hypothèse nulle](https://fr.wikipedia.org/wiki/Hypoth%C3%A8se_nulle) est vraie.

* + Exemples d’utilisation :
* **Comparaison de moyenne d'une**[**loi normale**](https://fr.wikipedia.org/wiki/Loi_normale)**à une valeur si la variance est inconnue.**
* **Comparaison de deux moyennes issues de deux**[**lois normales**](https://fr.wikipedia.org/wiki/Loi_normale)**si leurs variances sont égales et inconnues, ou si leurs variances sont différentes et inconnues (**[**Test t de Welch**](https://fr.wikipedia.org/wiki/Test_t_de_Welch)**).**
* **Test sur les coefficients dans le cadre d'une**[**régression linéaire**](https://fr.wikipedia.org/wiki/R%C3%A9gression_lin%C3%A9aire)**.**
* **Test sur des échantillons appariés.**

Pour exécuter un test t, vous avez besoin de 2 choses :

1. Le nombre de valeurs de données dans chaque groupe
2. Une valeur (alpha). Il s'agit d'un paramètre indiquant le risque de vous tromper que vous êtes prêt à accepter. Un de 0,05 signifie un risque de 5 %.

Les sorties d'un test t sont :

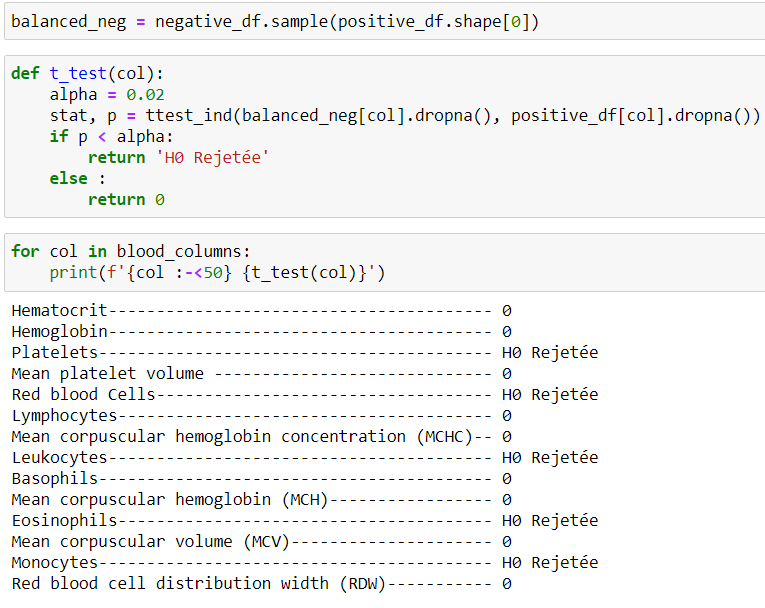
### ***a) valeur t***

Celui-ci est composé de deux éléments : la différence entre les moyennes de vos deux groupes et la variance entre eux. Ces deux éléments sont exprimés sous forme de rapport. S'il est petit, il n'y a pas beaucoup de différence entre les groupes. S'il est plus grand, il y a plus de différence.

### ***b) la valeur p***

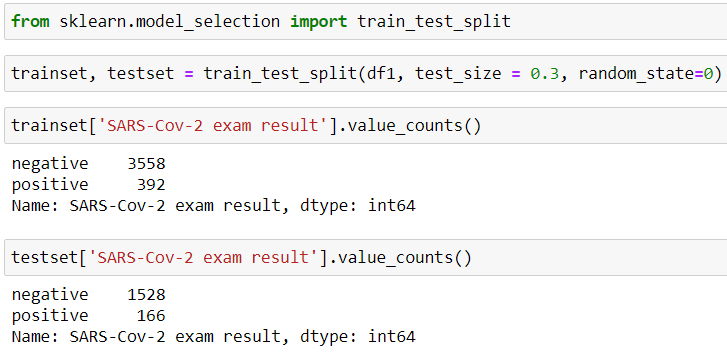
C'est le cœur du problème - il vous indique la probabilité que votre valeur t se produise par hasard. Plus la valeur p est petite, plus vous pouvez être sûr de la signification statistique de vos résultats.

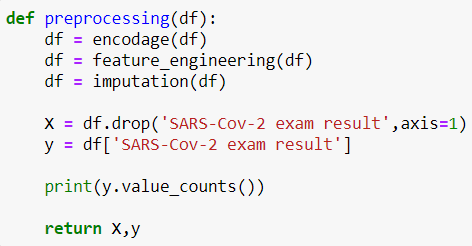
* Pour ce projet, je voudrais bien chercher ***si les individus atteints du covid-19 ont des taux de Leukocytes, Monocytes, Platelets significativement différents. D’où je pose l’hypothèse nulle suivante : « H0 = Les taux moyens sont ÉGAUX chez les individus positifs et négatifs ».***



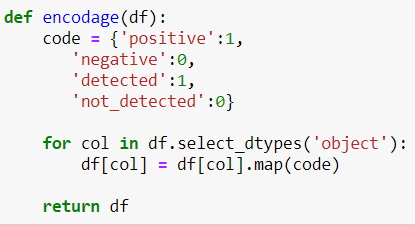
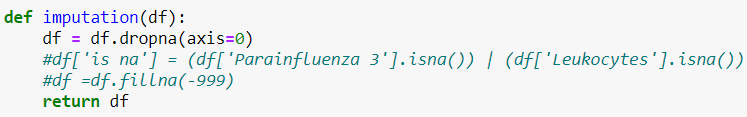
La procédure suivie pour mon T-test est :

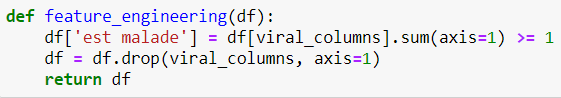
* + Afin de respecter l’égalité de la taille des deux échantillons, j’ai essayé de réduire la taille de l’échantillon des résultats négatifs du test Covid en prenant aléatoirement des données de l’échantillon intégral.
  + J’ai opté pour une petite valeur comme étant un seuil de vérification de l’hypothèse.
  + L’hypothèse H0 était rejetée précisément pour les taux sanguins que j’ai mentionnés avant.
  1. ***Le prétraitement des données***
* Objectif : Transformer le data pour le mettre dans un format propice au Machine Learning.
* ***Création du Train set (70% du dataset) / Test set (30% du dataset)***
* Il ne faut jamais évaluer la performance d’un modèle sur les données d’entraînement.

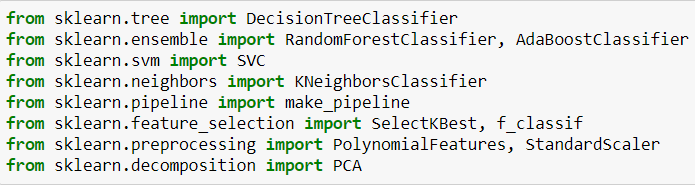


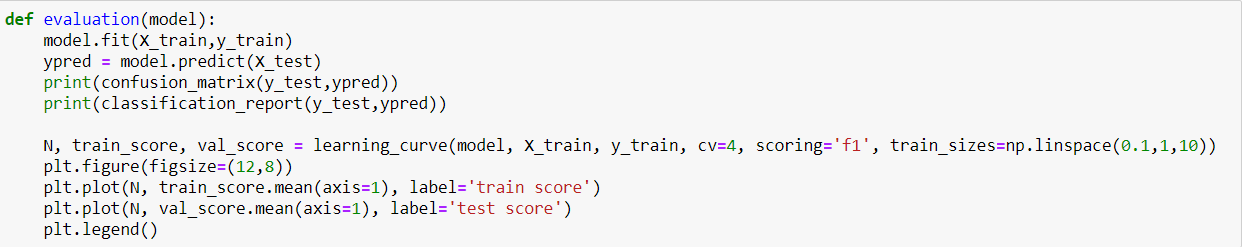
* + - * L’intérêt de calculer le nombre des cas positifs et négatifs dans chaque set est de vérifier que ce split a conservé le rapport entre les deux cas (Positif : 90% et négatif : 10%).
* ***Preprocessing :***

🡺 C’est la fonction principale de cette partie, car elle contient les 3 opérations essentielles lors du prétraitement des données, qui sont:

* 1. **Encodage :**
* Toute variable catégorielle doit être transformée en variable numérique.
  1. **Imputation :**
* Une opération permettant de nettoyer le dataset en éliminant ou en remplaçant les données manquantes présentes dans celui-ci.
  1. **Feature engineering**
* Fondamentalement, tous les algorithmes d'apprentissage automatique utilisent des données d'entrée pour créer des sorties. Ces données d'entrée comprennent des caractéristiques, qui se présentent généralement sous la forme de colonnes structurées. Les algorithmes nécessitent des fonctionnalités avec des caractéristiques spécifiques pour fonctionner correctement. Ici, le besoin d’**ingénierie des fonctionnalités** se pose. Je pense que les efforts d'ingénierie des fonctionnalités ont principalement deux objectifs :
* Préparer l'ensemble de données d'entrée approprié, compatible avec les exigences de l'algorithme d'apprentissage automatique.
* Améliorer les performances des modèles d'apprentissage automatique.

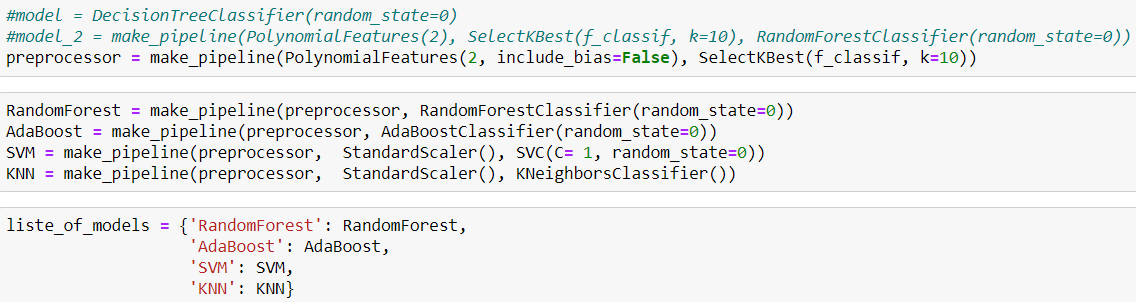
🡺En se basant sur les analyses de l’étape « EDA », la meilleure modification à réaliser est de remplacer les colonnes des tests viraux par la colonne ‘est malade’ qui donne une signification de la maladie d’une personne.

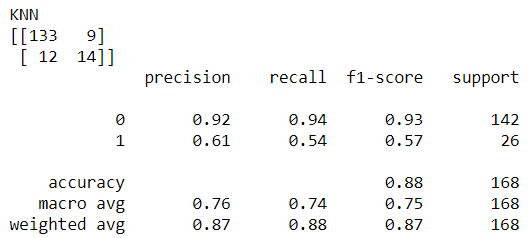
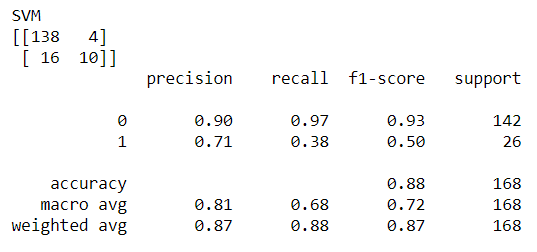
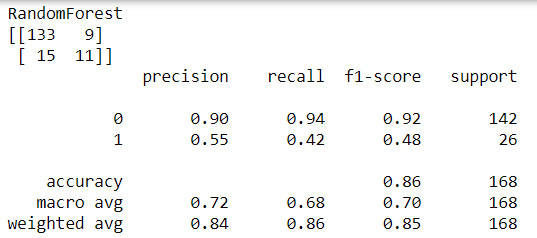
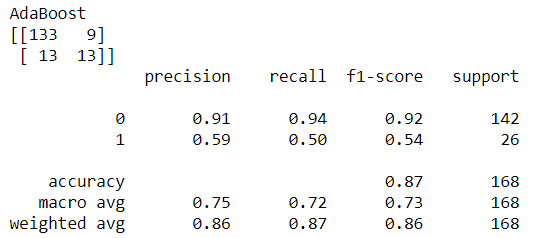
* 1. ***La modélisation***
* Objectif : Développer un modèle de machine learning qui répond à l’objectif final.
* En premier, il y aura importation des différents libraires des modèles connus du machine learning et différentes méthodes de Feature Selection et Feature Engineering.
  1. **Procédure d’évaluation :**

Dès le début, c’est indispensable de créer un système d’évaluation fiable et reproductible, afin de comparer la performance de maints modèles de Machine Learning qui peuvent être entrainés sur différentes configurations du dataset à cause du preprocessing.

* 1. **Entrainement de différents modèles**

En premier lieu, je cite la liste des modèles à entrainer. Pour ce projet, j’ai bien opté pour 4 différents types de modèles.

* **Bagging model : RandomForestClassifier**
* **Boosting model : AdaBoostClassifier**
* **SVM**
* **KNN : KNeighborsClassifier**

 Après entrainement des modèles, nous avons obtenu les résultats suivants :

On peut déduire à partir des résultats ci-dessus que les deux modèles de Boosting et KNN donnent des meilleures performances en terme de F1-score et la sensibilité (recall).

***N.B : Pour l’étape suivante, je vais choisir le modèle AdaBoost pour l’optimiser vu que le KNN model se base sur le nombre des instances, ce qui ne donne pas des performances qui sont très générales.***

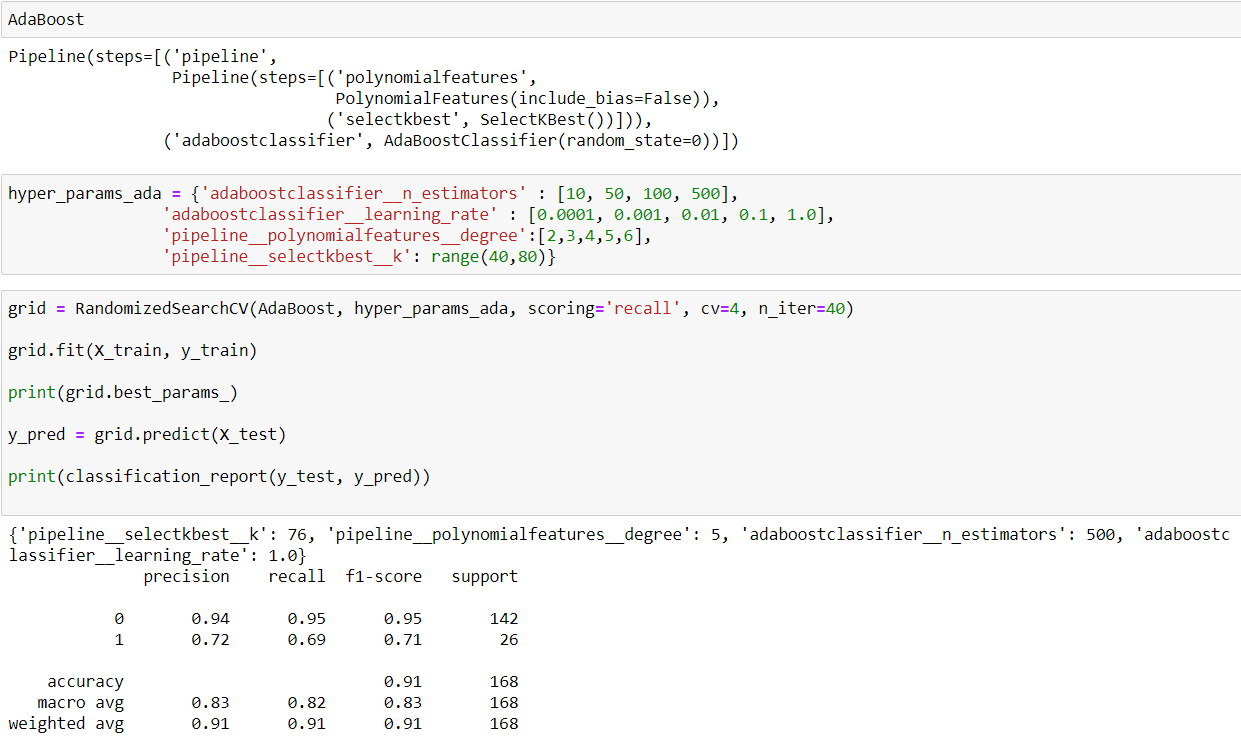
* 1. ***Optimisation***

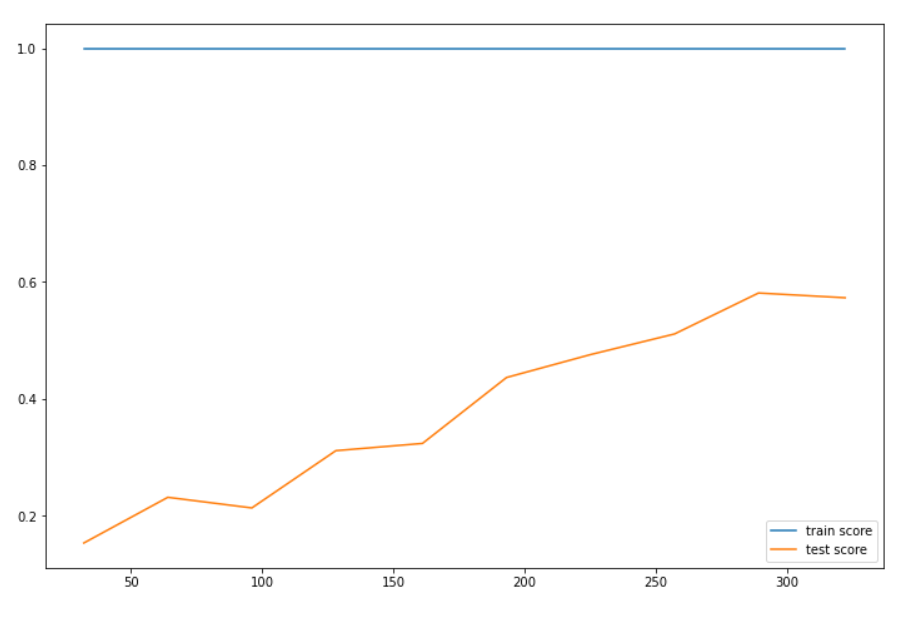
Contrairement aux paramètres du modèle qui sont appris pendant la formation, les hyperparamètres du modèle sont définis par le Data Scientist avant la formation et contrôlent les aspects de mise en œuvre du modèle. Les poids appris lors de l'apprentissage d'un modèle de régression linéaire sont des paramètres tandis que le nombre d'arbres dans une RandomForest est un hyperparamètre de modèle car il est défini par le data scientist. Les hyperparamètres peuvent être considérés comme des paramètres de modèle. Ces paramètres doivent être ajustés pour chaque problème, car les meilleurs hyperparamètres de modèle pour un jeu de données particulier ne seront pas les meilleurs pour tous les jeux de données. Le processus de réglage des hyperparamètres (également appelé optimisation des hyperparamètres) consiste à trouver la combinaison de valeurs d'hyperparamètres pour un modèle d'apprentissage automatique qui fonctionne le mieux - tel que mesuré sur un ensemble de données de validation - pour un problème.

Il existe beaucoup d’approches pour réaliser un tunning des hyperparamètres, parmi eux :

* ***GridSearch :*** mettre en place une grille de valeurs d'hyperparamètres et pour chaque combinaison, entraîner un modèle et un score sur les données de validation. Dans cette approche, chaque combinaison de valeurs d'hyperparamètres est essayée, ce qui peut être très inefficace !
* ***RandomSearch :*** mettre en place une grille de valeurs d'hyperparamètres et sélectionner des combinaisons aléatoires pour entraîner le modèle et le score. Le nombre d'itérations de recherche est défini en fonction du temps/des ressources.

**\*\*\* Passons à la pratique !!!**

****



Conclusion :