Rapport SD201

Nous avons choisi de porter notre étude sur une base de données comprenant de nombreuses informations sur les élèves ainsi que leurs notes sur 3 trimestres. L'objectif de cette étude est d'évaluer la corrélation qui existe entre la vie d'un élève et ses notes. Il s'agit d'une question majeure de l'éducation. En effet, nous savons tous que nous ne sommes pas égaux face à notre scolarité, de nombreux facteurs externes favorisent certains élèves et en favorisent d'autres. Notre objectif va donc être d'attester par les chiffres de certaines corrélation ou au contraire de réfuter certaines.

1. Data Cleaning

Notre objectif est d'essayer de prédire les notes des élèves à partir de leurs informations. La base de données est très large, le nombre de colonnes y est très important. Toutes les colonnes ont une utilité pour déterminer la note d'un élève. Pourtant, leur importance n'est pas la même, il va de soi qu'il est bien plus pertinent de connaître le nombre de redoublements d'un élève que son temps de trajet. Nous décidons de nous séparer de certaines informations sur l'élève.

Premièrement, nous voyons que dans la majorité des cas la consommation d'alcool le week-end est souvent liée à la consommation quotidienne et que les différences sont présentes en trop petit nombre pour avoir un impact sur notre algorithme. Nous enlevons donc la colonne "Dalc". De même, nous enlevons d'autres données de la base. Nous supprimons aussi les notes G2 et G3 car nous ne travaillerons que sur la prédiction de G1 (1er trimestre), pour des raisons de simplicité.
Voici les features gardées:

1. *school*
2. *sex*
3. *age : de 15 à 22*
4. *address urban ou rural*
5. *Pstatus - parent's cohabitation status (binary: 'T' - living together or 'A' - apart)*
6. *Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education)*
7. *Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education)*
8. *reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')*
9. *guardian - student's guardian (nominal: 'mother', 'father' or 'other')*
10. *traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)*
11. *studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)*
12. *failures - number of past class failures (numeric: n if 1<=n<3, else 4)*
13. *schoolsup - extra educational support (binary: yes or no)*
14. *famsup - family educational support (binary: yes or no)*
15. *paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)*

16. *activities - extra-curricular activities (binary: yes or no)*
17. *internet - Internet access at home (binary: yes or no)*
18. *romantic - with a romantic relationship (binary: yes or no)*
19. *famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)*
20. *freetime - free time after school (numeric: from 1 - very low to 5 - very high)*
21. *goout - going out with friends (numeric: from 1 - very low to 5 - very high)*
22. *Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)*
23. *health - current health status (numeric: from 1 - very bad to 5 - very good)*
24. *absences - number of school absences (numeric: from 0 to 93)*

```
students = students.drop(["Dalc", "famsize", "nursery", "higher", "Mjob", "Fjob","G2","G3"],
```

Comme nous le verrons, ces features n'ont pas toutes la même influence sur la note finale. Désormais, afin de pouvoir appliquer des algorithmes de classification, nous donnons des valeurs numériques à chacune des informations.

2. Classification

Dans un premier temps nous allons tenter de prédire les notes des élèves. En effet, notre étude s'intéresse aux liens qu'il y a entre les informations des élèves et leurs notes. A partir des informations que nous avons sur les élèves, nous allons estimer leurs notes. Pour se faire , nous utilisons plusieurs algorithmes que nous allons développer par la suite.

2.1. Train and Test Dataset

Tout d'abord, il faut que nous séparions notre base de données en deux dataset, un pour train et un pour test. Le premier est celui sur lequel nous nous baserons pour faire nos prédictions tandis que le second nous permettra de tester les prédictions ainsi obtenues. Nous séparons par conséquent notre base de données en deux avec un coefficient de séparation de 85%. Nous avons choisi ce dernier assez élevé car notre base de données est assez petite. Pour avoir des algorithmes fonctionnels, il faut que nous puissions les train sur un suffisamment grand nombre d'élèves.

Ensuite, nous déterminons aussi deux types de données dans la bdd. Il y a des informations sur les élèves et leurs notes. La première nous sert à prédire tandis que l'autre est la prédiction (la note).

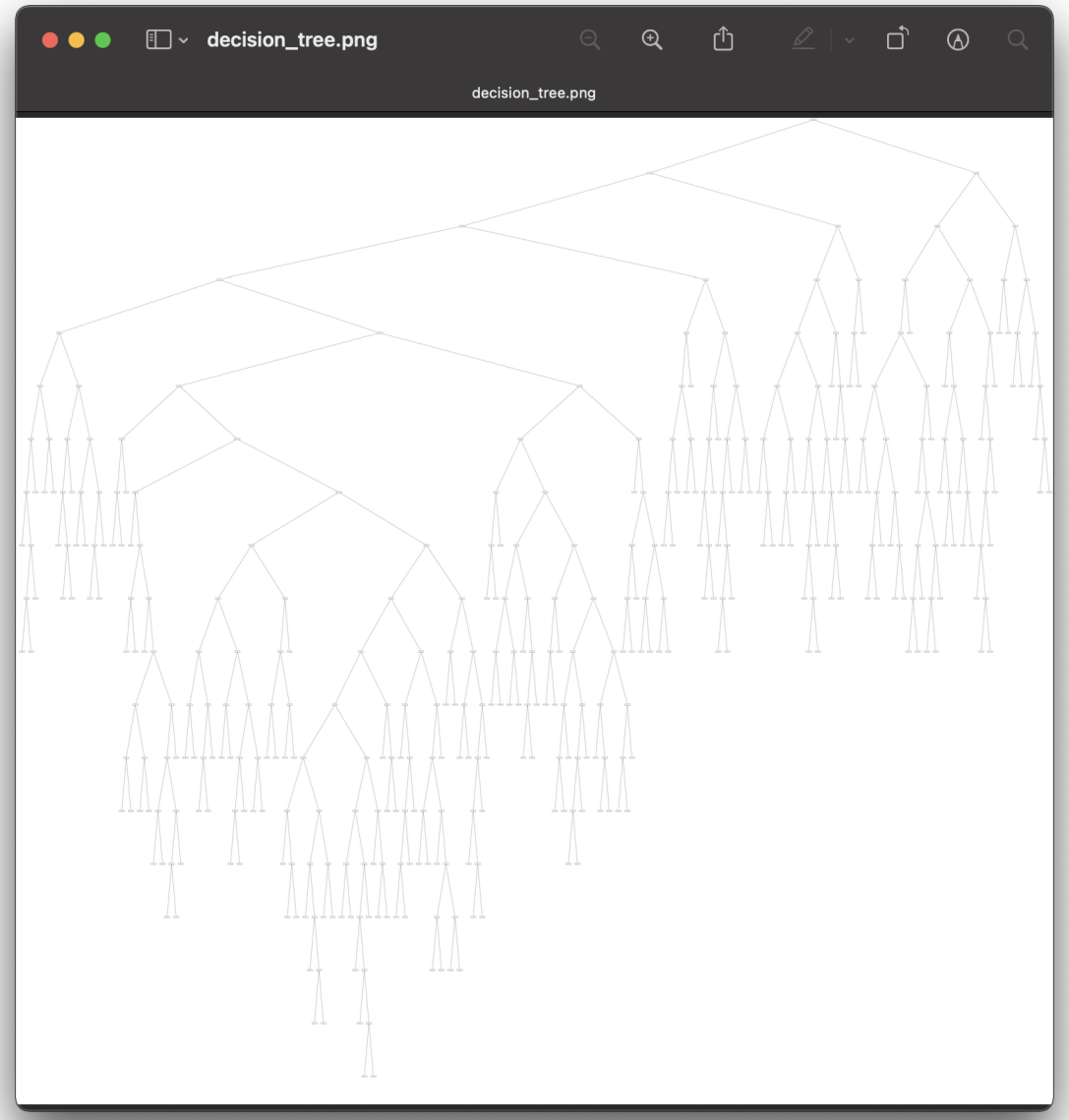
```
from sklearn.model_selection import train_test_split
X=students2.drop("G1",axis=1)
Y=students2["G1"]

X_train, X_test, y_train,y_test = train_test_split(X, Y, test_size=0.20)
```

2.2. Decision Tree Classifier

2.2.1. Decision Tree

Après application de l'algorithme, nous obtenons l'arbre suivant :

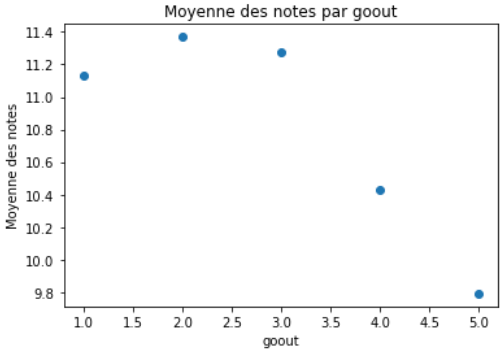


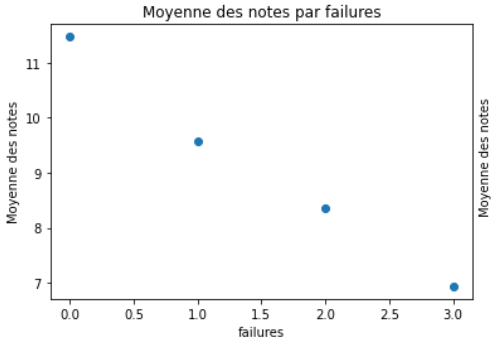
https://drive.google.com/file/d/1NER6E9HgP858_2razqC3xsSyHsJFpuy2/view?usp=sharing
(Image lourde à télécharger)

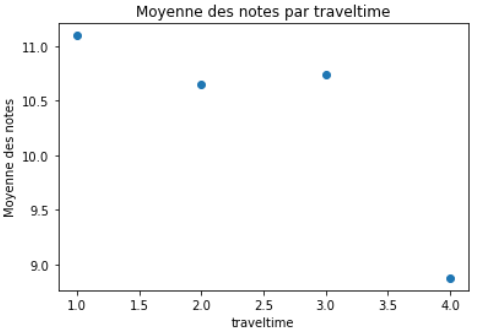
2.2.2. Interprétation

La première discrimination, à la racine de l'arbre, est sur *absences*. Ce premier résultat est logique, il est communément admis que l'assiduité en cours a un impact immédiat sur les notes. La seconde discrimination s'effectue sur *failures*, de même ce résultat n'est pas surprenant même si il est intéressant. Les élèves qui ont redoublé dans leur passé sont

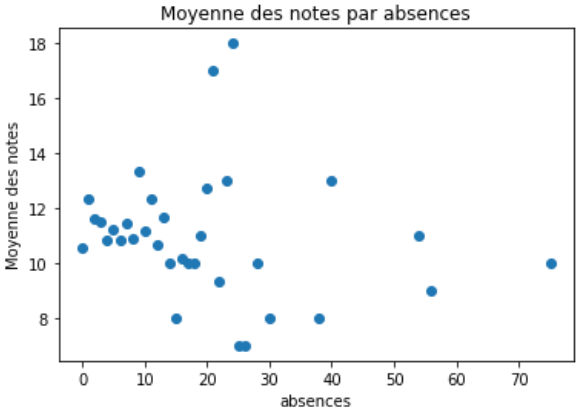
souvent ceux qui éprouvent le plus de difficultés et qui donc n'ont pas les meilleures notes. Ensuite, le nombre de sorties dans la semaine se révèle être un bon moyen de discriminer. En effet, *goout* a un impact assez conséquent. En outre, nous voyons aussi que le traveltime a aussi un impact sur les notes.
Ainsi, nous avons décidé de tracer, la moyenne des notes obtenues en fonction de ces paramètres.
Premièrement, nous voyons sur le graphe des *goout* que la fonction est décroissante et qu'elle permet en effet de fortement discriminer ceux qui sortent très souvent des autres. De la même manière, pour le traveltime et les failures, on observe une décroissance logique.







En revanche, pour le nombre d'absences, on voit qu'il n'y a pas nécessairement de corrélation très marquée avec les notes. Cependant, cela permet tout de même de bien discriminer, d'où un coefficient de gini élevé et sa place à la racine. En effet, les élèves ayant plus de 15 absences ont tous (à l'exception de 1) des notes inférieures à 11.



2.2.3. Fonction score et interprétation des prédictions

La métrique la plus utilisée pour évaluer la qualité d'un decision Tree est la fonction f1_score. Cependant, nous obtenons de très mauvaises valeurs.

```
print("f1_score =",f1_score(y_test, y_pred,average='micro'))
```

```
f1_score = 0.16666666666666666
```

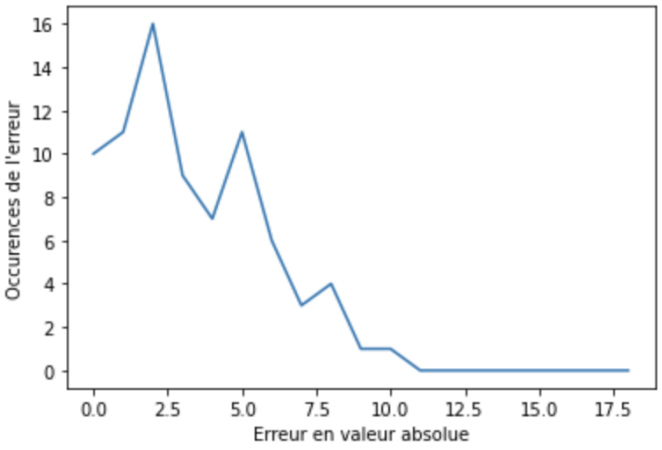
Cela peut s’expliquer par plusieurs facteurs:

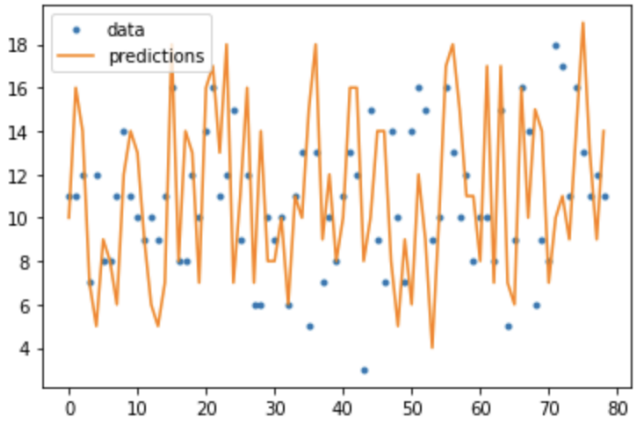
- Premièrement, notre base de donnée est petite, avec un coefficient de split de 85%, on train sur seulement sur 340 étudiants. Nos résultats ne peuvent donc pas être parfaits.

- De plus, essayons de prendre du recul sur ces chiffres. Le f1_score est une métrique qui s’appuie sur l’exactitude des valeurs obtenues. Ce processus est très binaire, soit un 8 est prédit en tant que 8, soit il ne l’est pas. Or, il est facile de comprendre que si la valeur prédite est 9, le résultat de la prédiction est beaucoup plus satisfaisant que s' il s’agit d’un 12. Le f1_score n’est donc pas une métrique très adaptée. Nous choisissons donc de travailler avec la MAE, qui nous permet d’évaluer de manière plus correcte la différence obtenue. Ici nous avons calculé l’erreur absolue maximum pour chacun des éléments de notre X_test. En moyenne, nous obtenons une différence entre la valeur prédite et la note que l’élève a réellement obtenue de 2,6. Ce résultat est plutôt satisfaisant d’un point de vue algorithmique. Cela signifie que en moyenne, nous pouvons donner un intervalle de 5 notes dans lequel la note de l’élève se trouvera. On pourrait ainsi penser à une classification par intervalle (mauvaise, moyenne, bien, excellente). Cependant cette dernière aurait aussi des inconvénients. En effet, la différence entre deux notes aboutirait probablement à des effets de bords, la différence entre 12 et 13 sera inexistante, mais grande entre 13 et 14 si notre catégorie "moyen" s'arrête à 13.

```
MAE= 3.3037974683544302


Erreur absolue maximum = 10
```



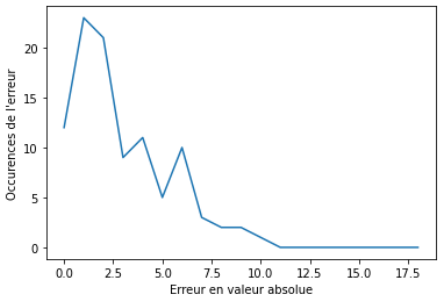


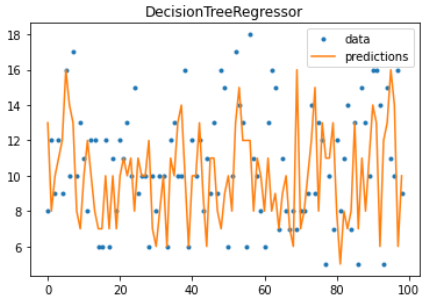
- Enfin, intéressons-nous à l'objectif que nous nous étions fixés. Nous souhaitons, à l'aide de cet arbre de décision, prédire les notes des élèves à partir de leurs informations. La conclusion que nous pouvons en tirer est la suivante: nous ne pouvons pas déterminer avec exactitude les notes des élèves à partir d'une poignée d'informations à leur sujet. Ce résultat est d'une part logique mais il est aussi rassurant. Nous avons montré qu'il y a avait des corrélations entre le passé scolaire d'un élève, sa fréquence de sortie ou encore sa consommation d'alcool mais pourtant cela ne détermine pas totalement sa note. De plus, le problème que nous tentons de résoudre s'apparente plus à un problème de régression qu'à un problème de classification. En effet, nous cherchons à attribuer une valeur entre 0 et 20 à des élèves, nous allons donc essayer de travailler avec un Decision Tree Regressor.

2.3. Regression Tree Classifier

```
MAE= 2.888888888888889

Erreur absolue maximum = 10
```





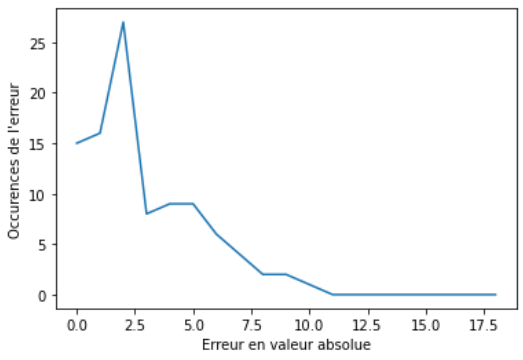
Voici les résultats que nous obtenons avec un Regression Tree Classifier, algorithme qui semble plus adapté. Tout d'abord, on voit que la MAE est plus faible cela semble intéressant. Mais en comparant les prédictions avec les data, les prédictions semblent bien moins bonnes.

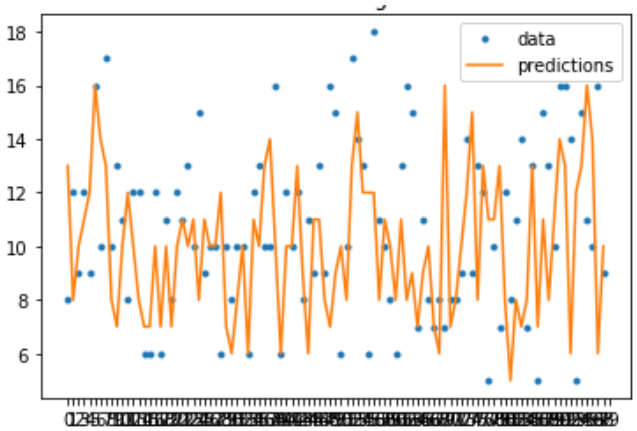
2.4. Random Forest

Ainsi, nous revenons donc à un algorithme Classifier. Mais nous utilisons une random forest plutôt qu'un unique arbre. Les résultats sont plus satisfaisants mais on voit aussi que l'on n'améliore pas vraiment l'intervalle des notes. Nous ne sommes passés que de 3.2 à 2.8 (soit un gain de 1 point).

```
MAE= 2.8585858585858586

Erreur absolue maximum = 10
```





2.5. Conclusion

Nous pouvons ainsi en conclure que les méthodes de Decision Tree aboutissent à un intervalle d'en moyenne 6 points. A partir des informations de chaque élève, nous pouvons donc prédire leur note avec une erreur relative de trois points. Prenons désormais du recul. Cela signifie que, en moyenne, la note d'un élève peut être déterminée à 3 points près en moyenne, en s'appuyant sur des informations de l'élève. Nous pouvons en tirer quelques conclusions. Premièrement, comme nous l'avons montré avec les coefficients de gini, certaines informations sont très discriminantes comme le passé scolaire (redoublements), le taux d'absence ou encore le temps de trajet pour se rendre à l'école. De plus, la catégorie de note obtenue par un élève est très souvent liée à ses informations. Pour autant, il ne faut pas être fataliste, certains élèves parviennent à sortir de ces intervalles.

* * *

3. Clustering

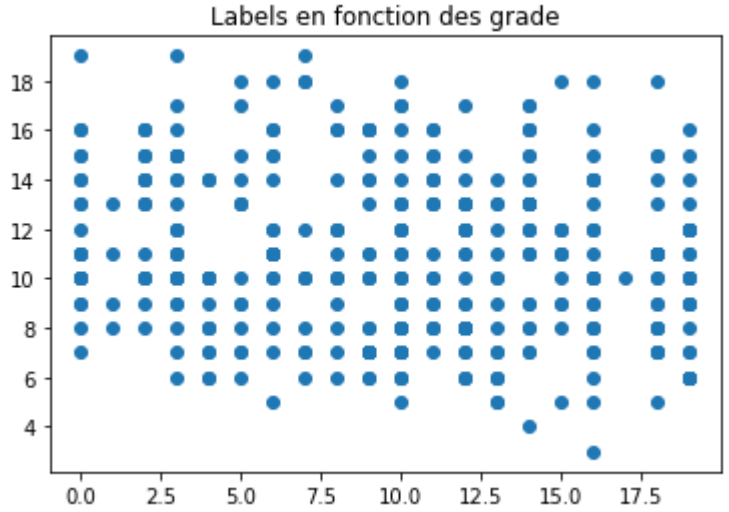
3.1. Objectif

Rappelons ce que nous cherchons à faire. L'objectif global est d'essayer de déterminer s' il existe un lien entre les informations d'un élève et les notes qu'il obtient. Notre démarche précédente s'appuyait davantage sur la prédiction des notes pour essayer de déterminer à partir des informations d'un élève la catégorie de notes qu'il est susceptible d'obtenir. En nous appuyant sur les algorithmes de prédiction, nous avons aussi pu déterminer les informations les plus discriminantes pour un élève. Cependant, pour répondre à notre question initiale, il peut être intéressant de clusteriser notre base de données. En effet, si nous prenons les informations dont nous disposons et que nous formons 20 clusters à partir de ces dernières, nous pouvons imaginer que si une telle corrélation existe alors chaque cluster correspondra à une note. On obtiendrait alors des clusters de notes, les élèves seraient regroupés par leurs informations et cela correspondrait à la note qu'ils obtiennent.

3.2. Clusterisation naïve

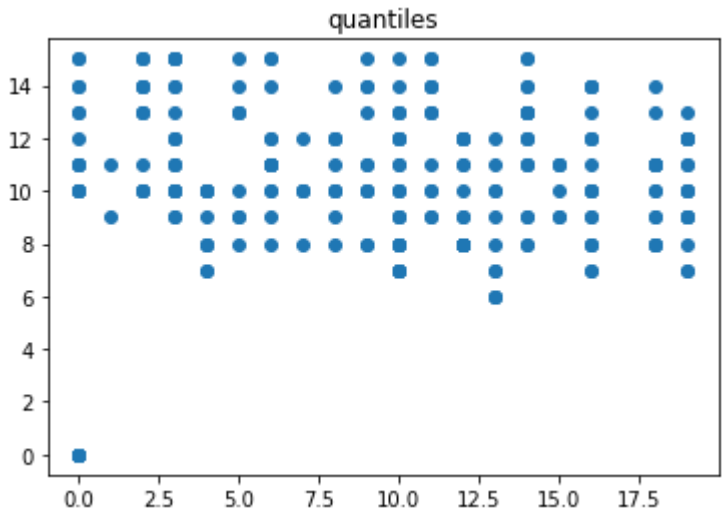
Nous avons donc implémenté cela. Nous exécutons et nous obtenons ce graphique avec en abscisse les labels et en ordonnée la note. Chaque point (label, grade) correspond donc à un élève qui a été rangé dans un label et qui a obtenu la note grade. Supposons qu'une certaine corrélation au sens du cluster existe. Alors, les élèves d'un même cluster devraient obtenir approximativement la même note. Or, on voit que le contraire se produit. Pour chaque cluster, la répartition des notes se fait sur une dizaine de notes environ. Ce résultat est alors encore moins corrélant que celui de la partie précédente car nous obtenions un intervalle de 6. Ainsi, les clusters que nous obtenons ne sont donc pas liés à la note des élèves et ne nous permettent pas de conclure sur une possible corrélation.

Cependant, essayons de réfléchir au graphe que nous venons d'obtenir. Nous voyons qu'il n'y a pas 395 points sur ce dernier. La raison est simple certains sont probablement superposés. Ainsi, il peut être intéressant d'avoir d'autres approches.



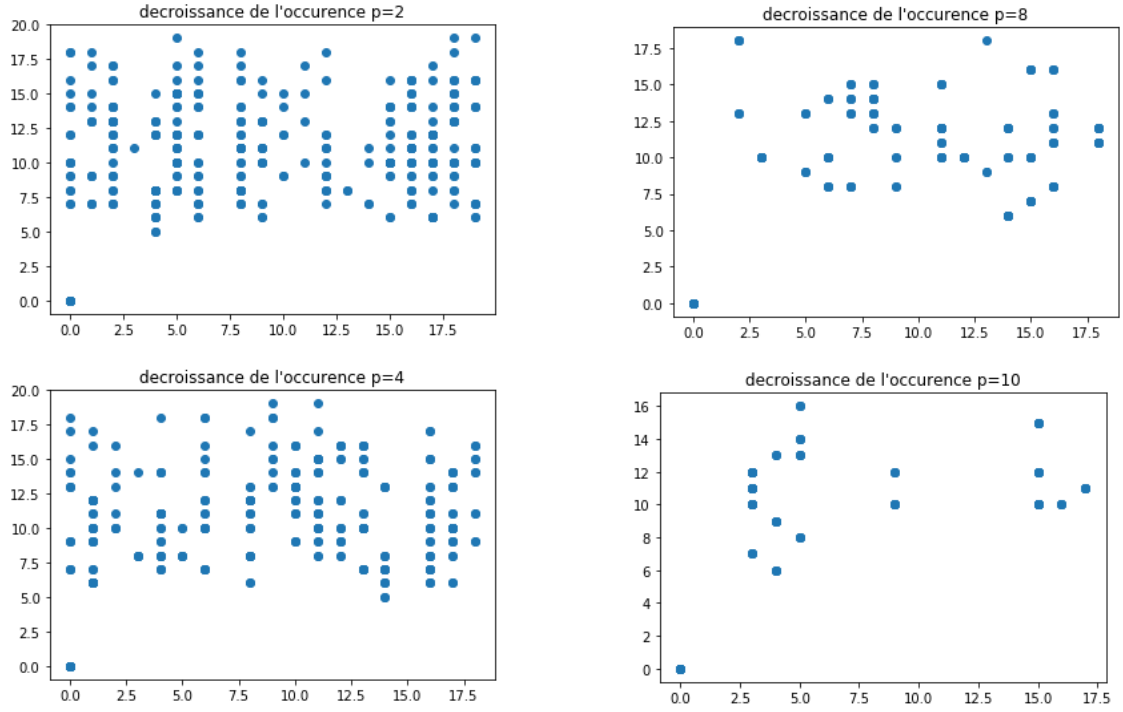
3.3. Approche des quantiles

Dans un premier temps, nous pouvons estimer que les valeurs extrêmes posent problèmes et sont les moins représentés dans un cluster. Ainsi, elles ne font pas l'objet des superpositions dont nous parlions. Pour chaque label, nous retirons alors 20% des valeurs. On voit assez rapidement, qu'il n'y a toujours pas de corrélation. Cette méthode tend uniquement à conserver les valeurs médianes mais fait donc tendre tous les clusters vers des notes comprises entre 10 et 14.



3.4. Approche par décroissance de l'occurrence

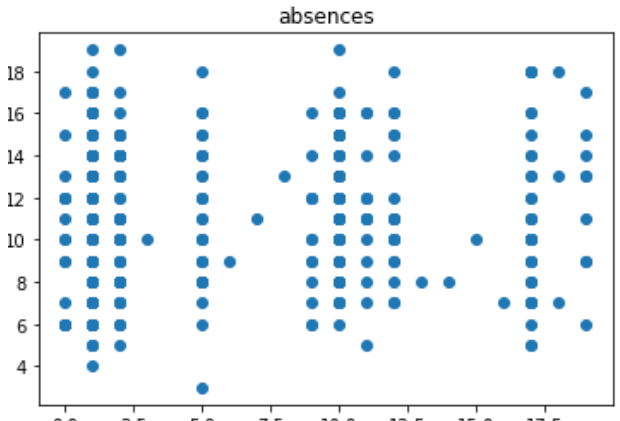
Une autre approche peut être, pour chaque cluster obtenu, de ne conserver que les notes avec la plus forte occurrence. Ainsi, nous pensons pouvoir nous affranchir du problème de superposition auquel nous sommes confrontés.



Analysons ces résultats. On voit que pour p = 2 et p = 4, nous obtenons des résultats similaires à ceux que nous avions précédemment. Pour p=10, on voit que certains clusters sont vides, le résultat n'est plus pertinent. Pour p=8, nous semblons avoir mit fin au problème de superposition. Cependant, nous voyons surtout qu'il y a une corrélation qui se dessine entre la note et le cluster.

3.5. Clusterisation à partir de quelques facteurs

Nous avons ensuite essayé de déterminer si certains facteurs permettaient d'aboutir à une clusterisation correcte. Nous avons donc repris ceux qui impliquaent la pluys forte discrimination lors de la construction du Classification Tree. Les résultats ne sont pas plus concluants. Nous voyons aussi que les clusters ne correspondent pas à des notes.



3.6. Conclusion

La clusterisation ne permet pas de répondre à notre objectif. Les clusters qui sont créés ne correspondent pas à la note obtenue par les élèves même si nous parvenons à le faire tendre vers cela. Cela ne signifie pas qu'il n'y a aucun lien entre les notes et les informations.