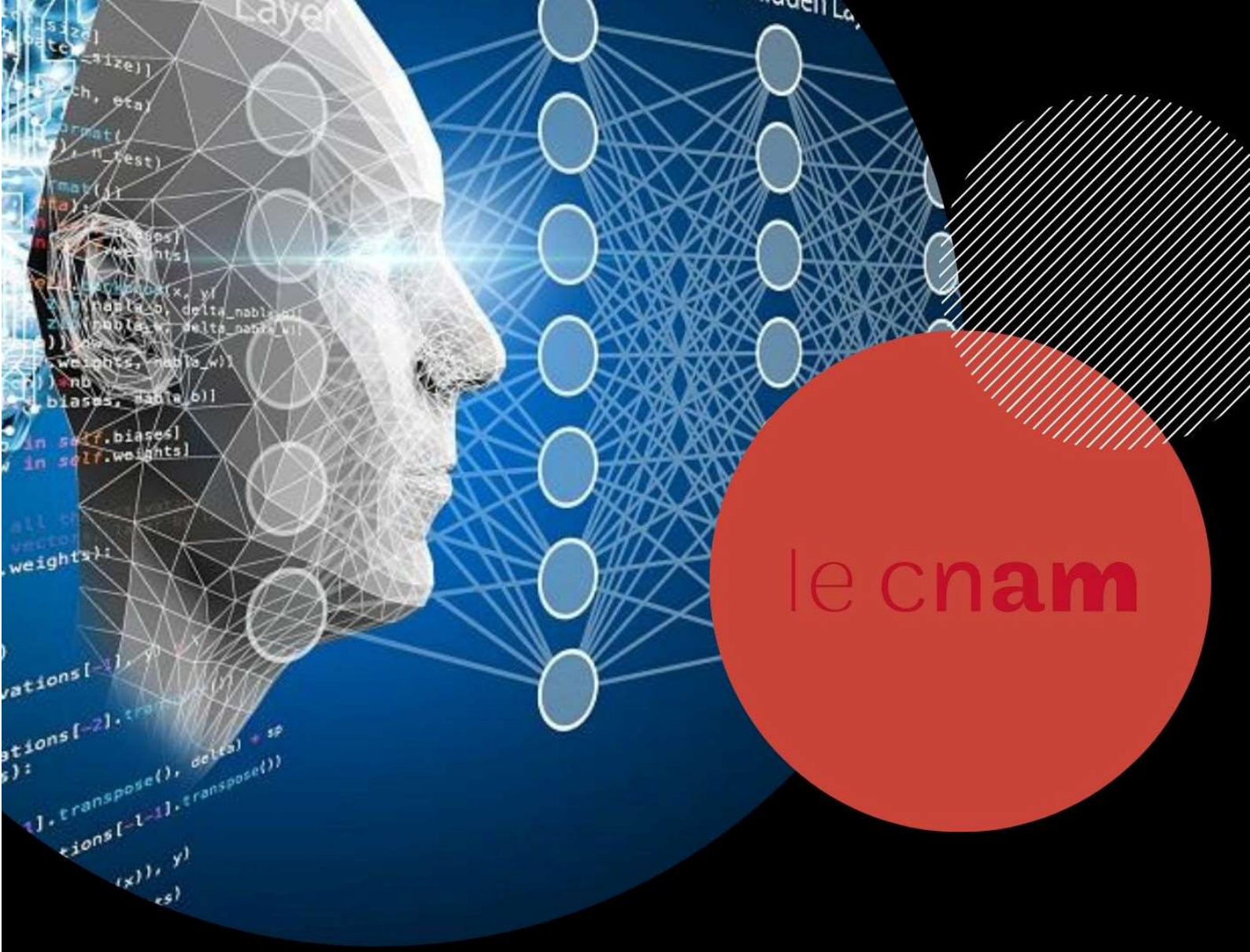


le cnam

LEARNING ANALYTICS

MACHINE LEARNING

Raphaëlle PAGOT
Nathan DESBROSSE
Léo JUBIEN

Table des matières

Objectif du projet

L'objectif de notre projet a été de devoir prédire les notes des étudiants en fonction des devoirs qu'ils ont réalisés.

Ensemble des données

Pour réaliser notre objectif, nous avons dû choisir parmi tous les fichiers qui nous ont été fournis au préalable, ceux qui sont nécessaire pour nous, à savoir :

- Le fichier mdl_course : Ce fichier nous a permis de récupérer le nom de l'ensemble des cours.
- Le fichier mdl_grade_items : Ce fichier nous a permis de récupérer le nom complet des devoirs donnés aux élèves.
- Le fichier mdl_assign_grades : Ce fichier nous a permis de récupérer l'id des correcteurs ainsi que le nombre de tentatives effectuées par devoir.
- Le fichier mdl_grades_grades : Ce fichier nous a permis de récupérer les notes obtenues par les élèves sur les devoirs.

Préparation des données

Avant de pouvoir exploiter nos fichiers, un important travail de nettoyage a été nécessaire.

- **Traitement des chaînes vides**

Dans un premier temps, nous avons dû corriger une problématique liée aux chaînes vides notifiées sous la forme « /N ». Cette notation, interprétée comme une chaîne de caractères par notre outil de visualisation, empêchait une gestion correcte des valeurs manquantes. Nous les avons donc remplacées par une valeur nulle standard, « NaN », afin d'assurer une meilleure compatibilité avec notre système.

- **Suppression des balises HTML**

Nos fichiers contenaient également des balises et des chaînes HTML insérées dans certains champs de texte, probablement pour améliorer l'affichage. Toutefois, ces éléments constituaient une gêne dans notre traitement des données. Nous avons procédé à leur suppression en les remplaçant par des espaces vides, grâce à une méthode de remplacement systématique.

- **Mise à jour des types de données**

Certaines colonnes nécessitaient une mise à jour de leur type de données, notamment pour les valeurs de type float et integer. Cette étape a permis d'assurer une cohérence dans la manipulation des données et leur intégration dans notre outil de visualisation.

- **Renommage des colonnes**

Enfin, nous avons procédé au renommage de toutes les colonnes afin d'uniformiser leur présentation. Cette harmonisation a permis de rendre les données plus compréhensibles et d'en faciliter l'analyse et l'exploitation dans les étapes suivantes.

Prédictions

Préparation des données d'entraînement

Après avoir préparé nos données, nous avons pu commencer la phase de prédiction. Comme énoncé dans l'objectif du projet, nous nous sommes intéressés aux notes et plus particulièrement à prédire quelle note allait obtenir un élève à un devoir donné en fonction de certains critères.

Pour ce faire, nous avons utilisé deux des fichiers qui nous étaient fourni qui sont le fichier **mdl_grade_grades** contenant les notes et le fichier **mdl_grade_items** contenant des informations sur les devoirs.

A partir de ces deux fichiers, nous avons pu ensuite appliquer certaines transformations sur les données pour arriver à notre jeu de données d'entraînement final.

- **Jointure des fichiers**
  Dans un premier temps, nous avons fait une jointure entre nos deux fichiers pour obtenir toutes les données dans un seul et même dataframe. Dans le fichier **mdl_grade_grades**, nous avions une colonne **itemid** qui nous a permis de faire la jointure avec la table **mdl_grade_items** qui possédait des identifiants pour chaque devoir. Avec cette jointure, nous obtenons donc un dataframe avec les données des deux fichiers fusionnés.

- **Filtrage des données**
  Pour nos prédictions, nous avons décidé de nous concentrer uniquement sur les devoirs de type **« assign »**. Nous avons donc filtré notre dataframe de l'étape précédente en gardant seulement les lignes pour lesquelles la colonne **itemmodule** est égale à **« assign »**. Afin de pouvoir faire nos prédictions, nous avons également supprimer les lignes où la note du devoir n'était pas spécifiée.

- **Normalisation des notes**
  Parmi les données qui nous étaient fournies, les barèmes de notation n'étaient pas les mêmes pour tous les devoirs. Certains devoirs étaient notés sur 2, d'autres sur 20, d'autres sur 100 etc… Afin d'uniformiser nos prédictions, nous avons normaliser nos notes et les avons toutes ramenées sur un barème sur 20.

- **Sélection des colonnes**
  Après avoir fait la jointure entre nos deux fichiers, nous avons donc obtenu un dataframe conséquent. Cependant, toutes les données ne sont pas pertinentes pour la prédiction des notes donc nous avons décidé de garder seulement 6 colonnes qui sont :
    - Itemid : Identifiant du devoir
    - Userid : Identifiant de l'étudiant
    - Courseid : Identifiant du cours
    - Rawgrademax : Barème du devoir
    - Aggregationweight : Coefficient du devoir
    - Grade : Note qui est notre variable cible

- **Ajout de nouveaux champs calculés**
  Pour augmenter nos données d'entraînement, nous avons décidé de rajouter des champs calculés qui nous semblaient pertinents pour les prédictions. Pour chaque ligne de notre dataframe, nous avons donc ajoutés la **moyenne de l'étudiant** sur toutes les notes que nous disposions, la **moyenne des notes dans le cours** concerné et la **moyenne de l'étudiant dans ce cours**.

Après toutes ces étapes, nous avons donc pu obtenir notre jeu de données (constitué de 9 colonnes et 52222 lignes) nettoyé et avec des données pertinentes pour la prédiction.

	itemid	userid	courseid	rawgrademax	aggregationweight	grade	user_grade_avg	course_grade_avg	course_user_grade_avg
0	13553	113873	4140	2.0	1.0000	20.0	15.50	20.00	20.0
1	13553	114080	4140	2.0	1.0000	20.0	20.00	20.00	20.0
2	13553	113924	4140	2.0	1.0000	20.0	20.00	20.00	20.0
3	7094	109055	2966	6.0	0.0122	16.0	16.67	14.04	16.0
4	7223	109055	2966	6.0	0.0122	20.0	16.67	14.04	16.0
...	...	...	...	...	...	...	...	...	...
52217	228813	914821	45757	20.0	1.0000	17.0	17.00	9.01	17.0
52218	228813	923692	45757	20.0	1.0000	17.0	17.00	9.01	17.0
52219	228813	926890	45757	20.0	1.0000	17.0	17.00	9.01	17.0
52220	220872	877195	48130	100.0	1.0000	0.0	0.00	7.44	0.0
52221	220872	159760	48130	100.0	1.0000	0.0	8.19	7.44	0.0

52222 rows × 9 columns

C’est ce jeu de données qui nous a permis de prédire les notes, ce que nous allons vous expliquer dans la suite du rapport.

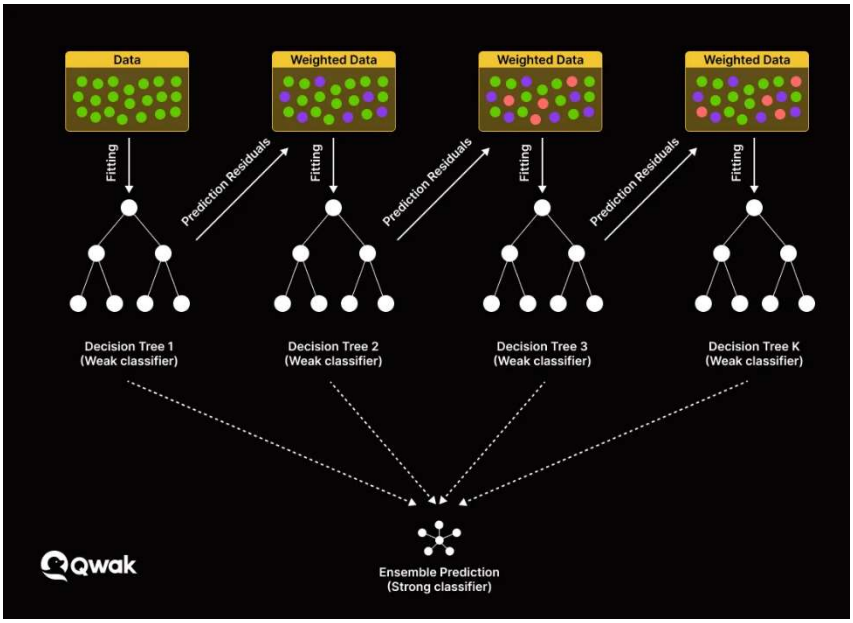
Modèles de prédiction

Prédiction d’une note sur 20 par régression

Afin de répondre à notre objectif, nous avons étudié la problématique selon deux aspects. Dans un premier temps, nous avons mis en place un problème de **régression** pour prédire la note exacte de l’élève entre 0 et 20.

Pour cela, nous avons d’abord séparer nos données issues de notre dataframe initial avec d’un côté notre variable cible à savoir la note et de l’autre côté, le reste des colonnes qui vont nous permettre de faire notre prédiction. Nous avons ensuite reparti notre variable cible et nos variables explicatives en données d’entrainement et données de test avec un répartition de 80% pour l’entrainement et 20% pour le test du modèle.

Pour notre problème de regression, nous avons décidé d’utiliser le modèle **XGBoost Regressor**. Ce modèle est un modèle d’apprentissage supervisé basé sur la technique de gradient boosting, optimisé pour les tâches de régression. XGBoost (eXtreme Gradient Boosting) améliore la précision et l’efficacité en combinant plusieurs arbres de décision faibles de manière séquentielle. À chaque itération, il corrige les erreurs des prédictions précédentes en ajustant les poids des observations mal prédites et en minimisant une fonction de perte.



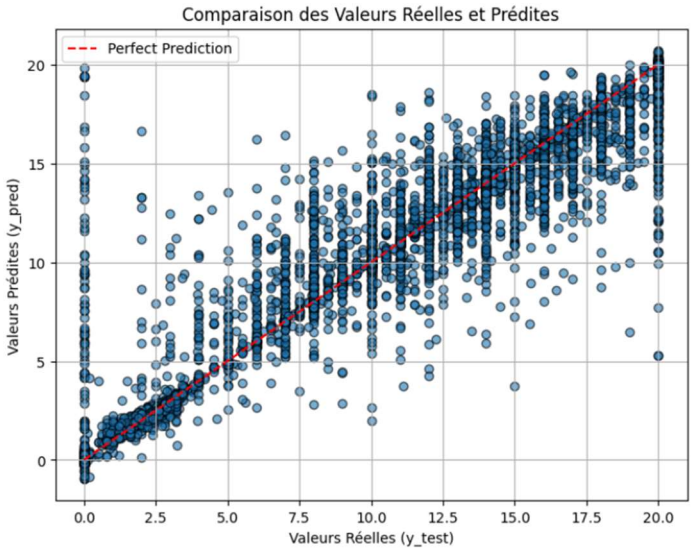
Afin de savoir quels paramètres attribuer à notre modèle, nous avons utilisé la méthode **Grid Search**. Cette méthode est une technique d’optimisation utilisée pour trouver les meilleurs hyperparamètres d’un modèle d’apprentissage automatique. Elle consiste à effectuer une recherche exhaustive sur un espace prédéfini de combinaisons d’hyperparamètres. Chaque combinaison est testée en entraînant le modèle et en évaluant ses performances, généralement à l’aide de la validation croisée. À la fin, Grid

Search sélectionne la combinaison d'hyperparamètres qui maximise la performance selon une métrique définie.

Après avoir obtenu nos hyperparamètres optimaux, nous avons pu entraîner notre modèle sur nos données d'entraînement puis par la suite, utiliser notre modèle entrainé sur nos données de test pour faire nos prédictions et mesurer la performance du modèle.

Lorsqu'on observe nos résultats sur un petit échantillon de données, on voit que nos prédictions sont relativement proches de la réalité. On voit cependant que pour la dernière valeur de l'exemple on à tout de même 2,67 points de différence entre la note réelle et la note prédite. Pour obtenir des résultats plus précis sur la performance de notre modèle, nous avons créer un graphique comparant les notes réelles et les notes prédites.

	Actual	Predicted
0	15.50	15.490000
1	15.71	15.750000
2	16.00	16.139999
3	14.00	14.480000
4	11.00	13.670000



Sur ce graphique, on a en abscisse les notes réelles issues de notre jeu de données de test et en ordonnée les notes prédites par notre modèle. La diagonale rouge représente les valeurs idéales où la valeur prédite est égale à la valeur réelle. On remarque que la tendance globale suit la diagonale mais pour cette valeur, on a des écarts importants notamment sur les valeurs extrêmes à savoir 0 et 20.

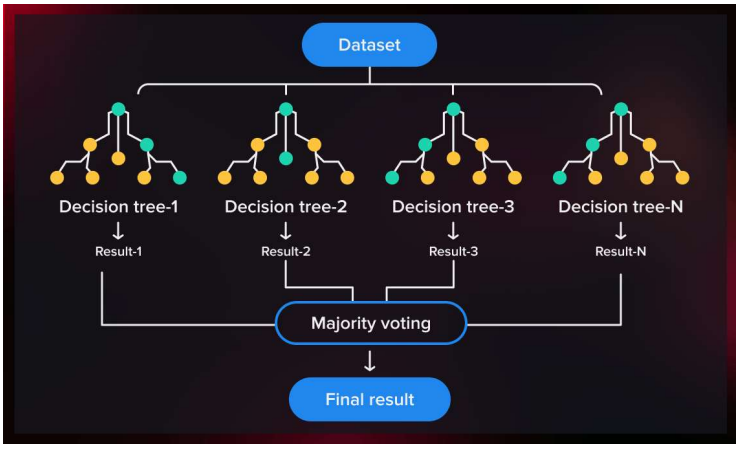
On observe que dans certains cas lorsque la note réelle est 0, le modèle prédit des notes supérieures à la moyenne allant même jusqu'à 20. Pour montrer cela, nous avons utilisé la **MSE** (Mean Squarred Error) pour évaluer les écarts moyens entre les valeurs réelles et les valeurs prédites et avons obtenu une **MSE** de **4,76** ce qui signifie que l'écart moyen entre les notes réelles et prédites est de 2,18 points (racine carrée de 4,76). La performance du modèle peut aussi être évaluée par le R^2 qui montre la corrélation entre les valeurs prédites et réelles. Sur notre modèle, nous avons obtenu un **R^2** de **0,87**, sachant qu'un R^2 égal à 1 symbolise une corrélation parfaite entre les deux variables.

Nous avons donc enregistré ce modèle déjà entrainé et nous avons pu l'utiliser pour prédire les notes de tous les devoirs que nous avions pour les introduire dans un fichier CSV qui nous a permis d'afficher nos données dans le tableau de bord.

Prédiction d'une note inférieure ou supérieure à la moyenne par classification

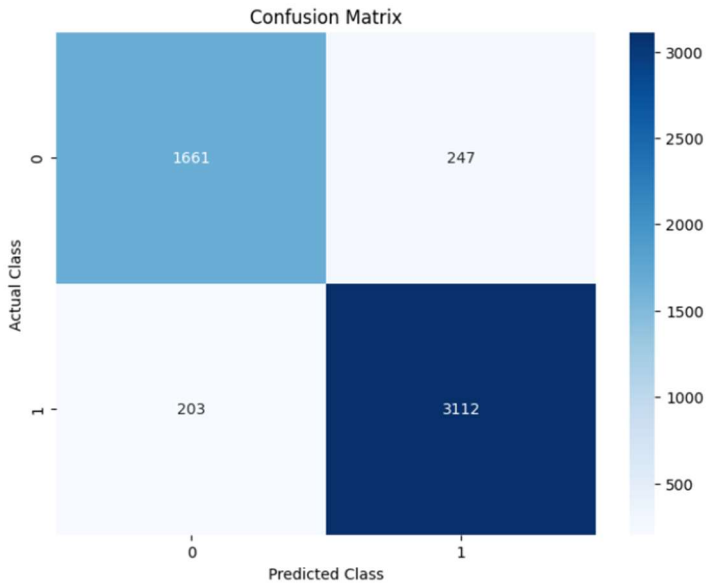
Dans un second temps, nous avons transformer le problème de régression en problème de classification afin d'observer si nous pouvions obtenir de meilleurs résultats et une meilleure précision du modèle. Pour ce faire, nous avons dû appliquer une modification sur notre variable cible. Nous avons transformé la note obtenue en classe avec une valeur égale à 0 pour les notes en dessous de 10 et une valeur égale à 1 pour les notes au-dessus de 10. Avec cette approche, nous avons un problème de classification à deux classes où l'objectif est de prédire si un élève va obtenir une note supérieure ou inférieure à la moyenne.

Après avoir modifié notre variable cible, nous avons séparé, de la même manière que dans l'approche précédente, les données en données d'entraînement et données de test avec un répartition de 80/20. Pour le problème de classification, nous avons utilisé ici le modèle **Random Forest Classifier**. Ce modèle est un modèle d'apprentissage supervisé utilisé pour les tâches de classification. Il fonctionne en combinant une multitude d'arbres de décision, chacun formé sur des sous-échantillons aléatoires des données d'entraînement et un sous-ensemble aléatoire des caractéristiques. Lorsqu'il effectue une prédiction, chaque arbre vote pour une classe, et la classe majoritaire parmi les votes est choisie comme résultat final. Cette approche réduit les risques de surapprentissage en diversifiant les arbres et en limitant leur corrélation.



Comme pour le problème de régression, nous avons utiliser la méthode **Grid Search** pour trouver des paramètres optimaux pour ce modèle afin de maximiser nos performances et nos prédictions. Nous avons par la suite pu entraîner notre modèle avec les données d'entraînement avant de prédire nos classes avec les données de test.

Pour évaluer les résultats de nos prédictions, nous avons tout d'abord la **précision** du modèle qui est dans notre cas à **91%** et mesure la proportion des prédictions correctes par rapport au nombre total de prédictions effectuées. Plus la précision d'un modèle se rapproche des 100% et plus il classifie les individus dans les bonnes classes. Notre modèle obtient donc un score plutôt bon. Pour voir plus précisément la répartition des prédictions, nous avons utilisés une matrice de confusion.



Sur cette **matrice de confusion**, on peut voir que le modèle a bien classifié 1661 notes dans la classe 0 (note inférieure à 10/20) et 3112 dans la classe 1 (note supérieure à 10/20). A l'inverse, il classé 203 notes dans la classe 0 alors qu'elles sont en réalité dans la classe 1 et 247 dans la classe 1 alors qu'elles sont en réalité dans la classe 0.

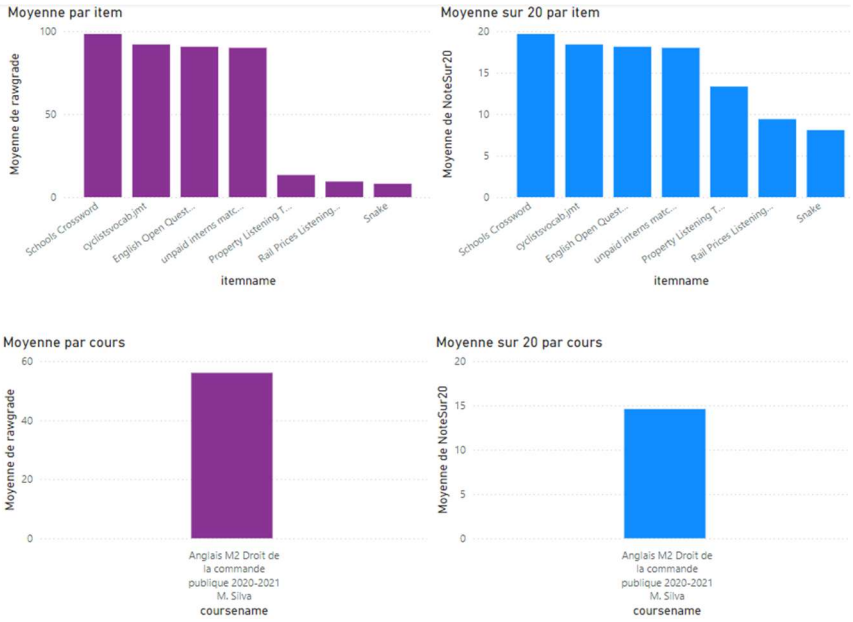
Comme pour le modèle précédent, nous avons enregistré ce modèle entrainé et nous avons pu l'utiliser pour prédire les classes des notes de tous les devoirs que nous avions pour les introduire dans un fichier CSV qui nous a permis d'afficher nos données dans le tableau de bord.

Visualisation

Pour la visualisation de nos données, notre choix s'est porté sur **Power BI**. Cette décision repose sur deux principales raisons : tout d'abord, Power BI est l'un des outils les plus populaires et performants du marché. Ensuite, nous disposions déjà d'une expérience préalable avec cet outil, ce qui a facilité sa prise en main pour l'ensemble de notre équipe.

Notre tableau de bord comporte **sept onglets**, chacun dédié à un aspect spécifique de l'analyse :

- **Étude des notes par item et par cours :**
  Le premier onglet présente une analyse détaillée des notes par item et par cours. Nous avons constaté que toutes les notes ne suivent pas le même barème. Afin de garantir une cohérence visuelle et analytique, nous avons normalisé l'ensemble des notes sur 20. Cela nous permet de produire des graphiques homogènes pour les moyennes.

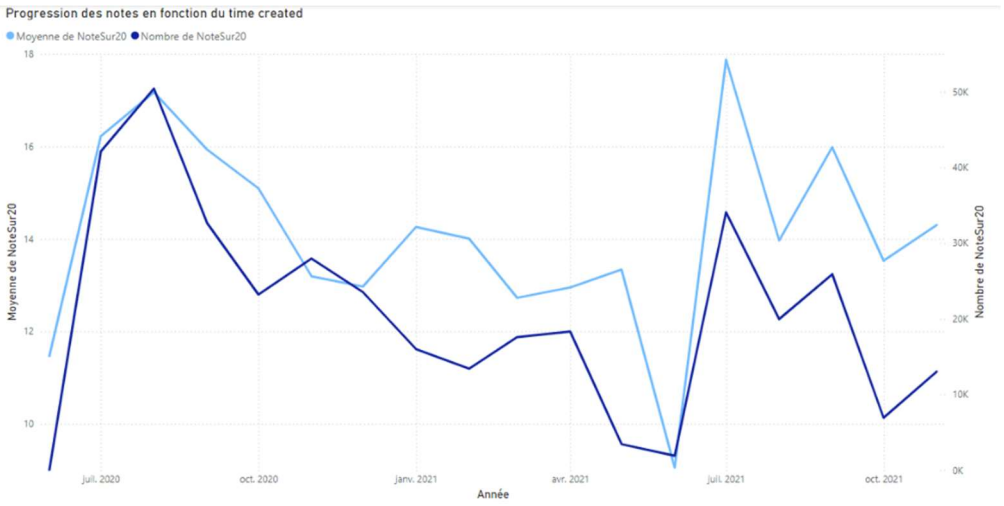


- **Étude des notes par élève :**

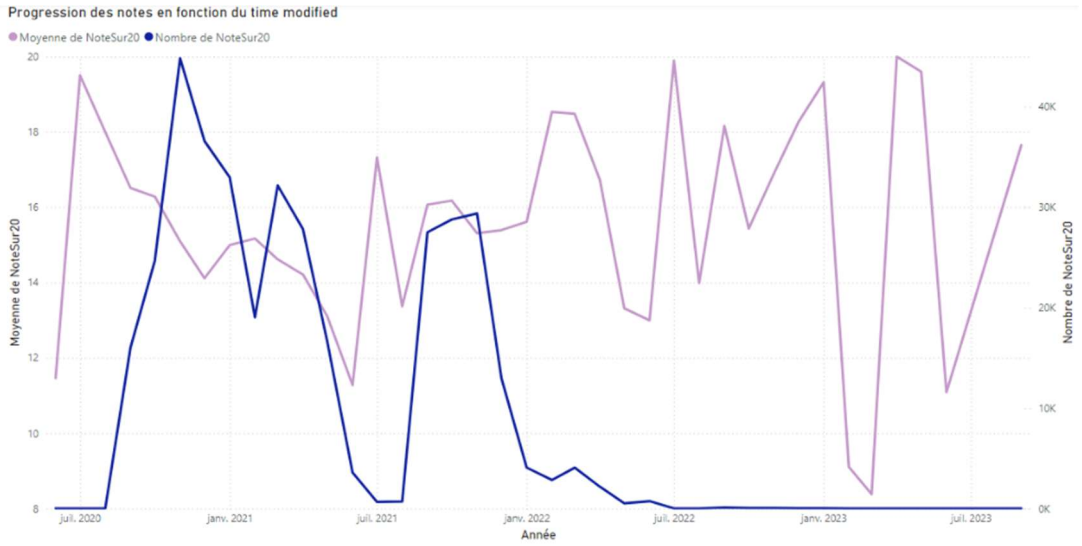
  Le second onglet propose une analyse des résultats par élève. Ce tableau interactif permet d'examiner plusieurs indicateurs : la proportion d'élèves ayant validé ou non leur année, le pourcentage de réussite, la répartition des notes et la moyenne par élève. Des filtres permettent d'affiner ces analyses par cours ou par devoir.

- **Nombre de devoirs et moyennes des notes générées :**
  Dans le troisième onglet, nous analysons le nombre de devoirs créés et la moyenne des notes obtenues pour ces devoirs. Ce graphique offre une vue d'ensemble de l'activité pédagogique et des performances associées. On remarque une corrélation apparente entre l'intensité de l'activité pédagogique et les résultats :
    - **Périodes d'intense activité pédagogique** : Lorsqu'un grand nombre de devoirs sont créés, les moyennes des notes tendent à être élevées. Cela pourrait refléter une dynamique positive, où des évaluations fréquentes contribuent à de meilleures performances des élèves.
    - **Périodes de faible activité pédagogique** : Inversement, pendant les périodes où peu de devoirs sont créés, les moyennes des notes chutent. Cela pourrait indiquer un impact négatif dû à un manque de suivi ou d'encadrement pédagogique régulier.

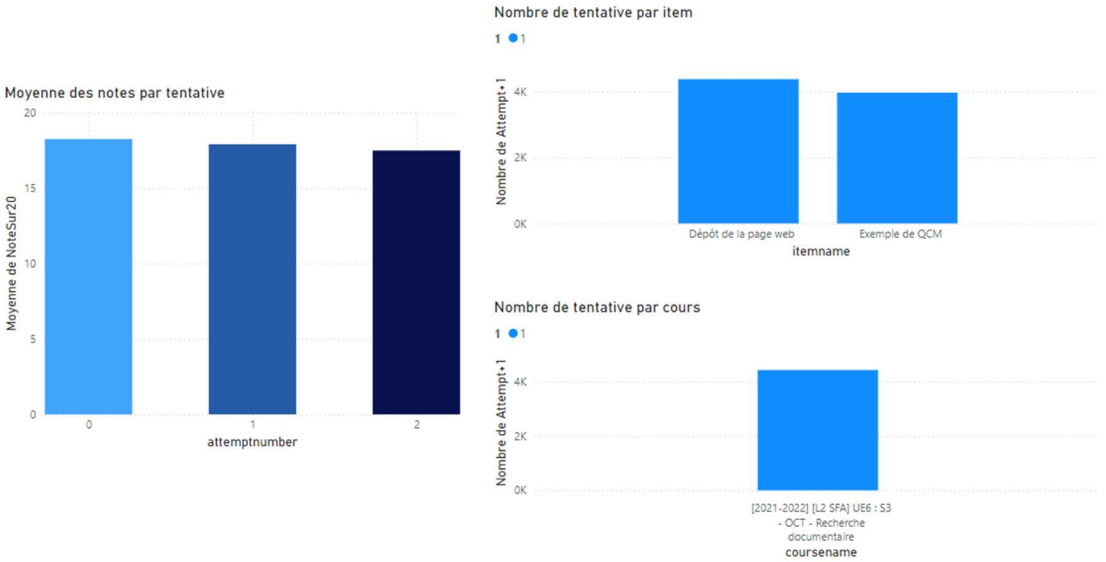
  Ce graphique met donc en évidence l'importance de maintenir une activité pédagogique continue pour favoriser les performances des élèves.



- **Analyse des modifications de notes :**
  Le quatrième onglet est consacré aux notes modifiées. Il affiche le nombre de notes ayant été révisées ainsi que leur moyenne après modification, offrant une perspective sur l'évolution des résultats.



- **Étude des tentatives :**
  Le cinquième onglet explore les tentatives des élèves. Il comprend des indicateurs tels que la moyenne des notes par tentative, le nombre de tentatives par cours et par devoir. Tous ces visuels peuvent être filtrés pour une analyse plus fine.

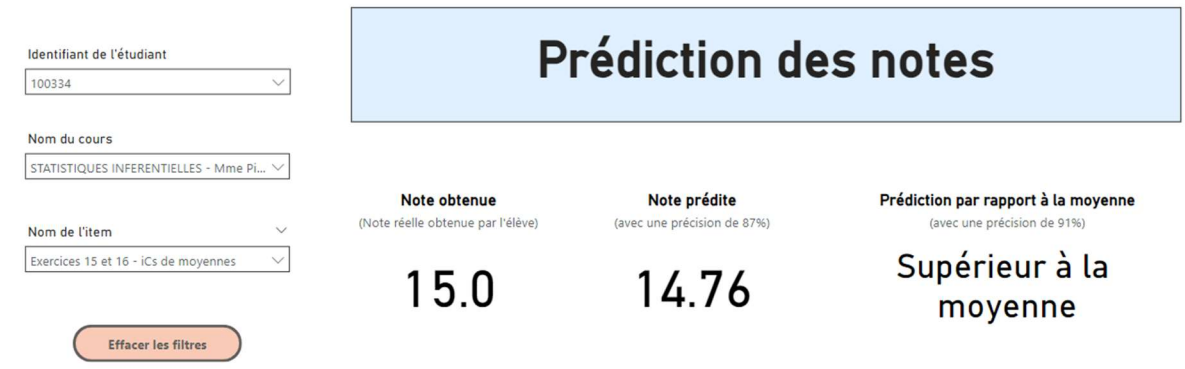


- **Moyenne des notes par devoir et correcteur :**
  Le sixième onglet présente un tableau croisé qui met en lumière les moyennes des notes obtenues pour chaque devoir, en fonction des correcteurs. Cet onglet permet d'approfondir l'analyse des résultats d'un élève dans une matière donnée et d'identifier les éventuelles variations liées aux correcteurs.

Moyenne des notes par item par évaluateur

coursename	0	702	777	783	1317	1998	2061	2136	2148	2151	2166	2169	2172	2175	2283	2355	2589	2823
[2021-2022] [L2 SFA] UE6 : S3 - OCT - Recherche documentaire	18,13	18,25	19,60	18,06	19,17	18,75	12,00	18,20	19,03	18,53	20,00	18,90	19,40	17,06	17,71	17,22	17,55	17,36
Dépôt de la page web	20,00	20,00	19,20	19,90	20,00	19,48	4,00	20,00	20,00	19,67	20,00	19,40	19,95	19,75	20,00	20,00	19,90	19,25
Exemple de QCM	16,25	16,50	20,00	16,32	18,33	17,99	20,00	16,58	18,06	17,17	20,00	18,33	18,85	14,38	15,42	14,44	15,21	14,33
Total	**18,13**	**18,25**	**19,60**	**18,06**	**19,17**	**18,75**	**12,00**	**18,20**	**19,03**	**18,53**	**20,00**	**18,90**	**19,40**	**17,06**	**17,71**	**17,22**	**17,55**	**17,36**

- **Prédictions des résultats :**
  Le dernier onglet est dédié aux prédictions. L'utilisateur sélectionne un élève, un cours auquel il est inscrit, ainsi qu'un devoir qu'il a réalisé. La visualisation affiche alors :
  - La note réelle obtenue par l'élève,
  - La note prédite par notre modèle,
  - Une seconde prédiction indiquant si l'élève a obtenu une note au-dessus ou en dessous de la moyenne.



userid	nom_cours	nom_item	user_grade_avg	course_grade_avg	course_user_grade_avg	grade	predictions_note	predictions_classe_note
100334	STATISTIQUES INFERENTIELLES - Mme Pintoux	Exercices 15 et 16 - iCs de moyennes	12.25	15.79	15.0	15.0	14.76	Supérieur à la moyenne

Conclusion

Pour conclure ce rapport, ce projet aura été pour nous l'occasion d'appliquer des modèles de Machine Learning à des problématiques réels. Les données mises à notre disposition étant des données brutes, nous avons dû passer par une phase assez importante de nettoyage des données pour les exploiter, ce qui reflète réellement ce à quoi nous pouvons être confronté en entreprise à l'avenir. De plus, le thème des Learning Analytics nous semble être une pratique qui devrait être de plus en plus démocratisée pour permettre aux élèves de les aider dans leur apprentissage.