

Projet IHM : Interface homme-machine

Justine Coutelier

Julie Courgibet

I. Présentation du projet

1. Origine du projet

L'établissement Knowledge Media Institute a lancé un projet de learning analytics (domaine de la data science) sur le site : <https://analyse.kmi.open.ac.uk/>

C'est un challenge en ligne qui permet de s'exercer dans le domaine de l'analyse de données et de prédiction.

2. Continuité du projet de Machine Learning

C'est à partir de ces données librement accessibles que nous avons réalisé du machine learning pour un autre projet.

Il s'agit d'un dataset contenant 7 fichiers csv dont le modèle de données est décrit sur le site. On distingue 3 unités :

- Les activités des étudiants : 3 fichiers concernent cette unité avec les dates d'inscription, les notes pour les rendus (assignments), le nombre de consultations de l'ENF tous les jours (espace numérique de formation).
- Les informations concernant les étudiants regroupées en un seul fichier : leur identifiant (données anonymisées), âge, domicile (par région), et le résultat final.
- Les informations sur les modules contenues dans 3 fichiers csv, avec notamment la liste des devoirs attendues pour chaque module et les dates de rendu limite, les dates de démarrage et de fin de module.

A partir de ce dataset complet, pour l'UE Machine Learning, nous avons regroupé toutes ces données en les traitant et en les analysant dans le fichier data.csv (plus de 2Gb, avec la structure suivante : 39 colonnes et 10 466 040 lignes). Les données concernent 32593 clés uniques élève, module et année. L'élève est représenté par son identifiant, le module par le code module et l'année (ou période d'étude) par le code présentation.

Les 10466040 lignes obtenues correspondent à des indicateurs pour chaque clé unique pour un temps t. Dans le but de simplifier les prédictions, nous avons donc pris le parti de créer plusieurs fichiers de données correspondant au temps d'évolution du module : de 0 à 100%. On a donc divisé le fichier data.csv de plus de 10 millions de lignes en 11 fichiers de données de 32593 lignes. A partir de ces fichiers, nous avons entraîné un modèle pour prédire le score de l'élève (réussite ou échec).

Pour ce projet, nous utilisons donc les 11 fichiers contenant les indicateurs nécessaires à la prédiction et les 11 modèles entraînés pour les 11 temps d'évolution du module (0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% et 100%).

3. Création d'une interface

Pour mettre en valeur ce travail, nous avons souhaité assurer la continuité du projet en réalisant une interface ou API (Application Programming Interface), ou interface de programmation. Cette interface permettra d'afficher les prédictions réalisées tout en explicitant les facteurs qui ont conduit à cette prédiction.

II. Note technique

1. Travail préparatoire : UI/UX design

Avant de développer l'interface, l'étape préparatoire est de la définir et de la dessiner. Nous avons utilisé l'outil evrybo. D'autres outils plus complets existent mais ils sont payants.

Le projet evrybo est accessible sur cette page :

<https://app.evrybo.com/share/project/33448/443769/YWZtSbzFo8Vao98z4YK5>

Trois canvas ou dessins ont été réalisés :

- la page d'accueil :

Le design de l'interface utilisateur pour l'Outil de gestion Moodle est présenté sous forme de deux écrans distincts, chacun sur un fond vert vif, encadrés par un fond gris clair.

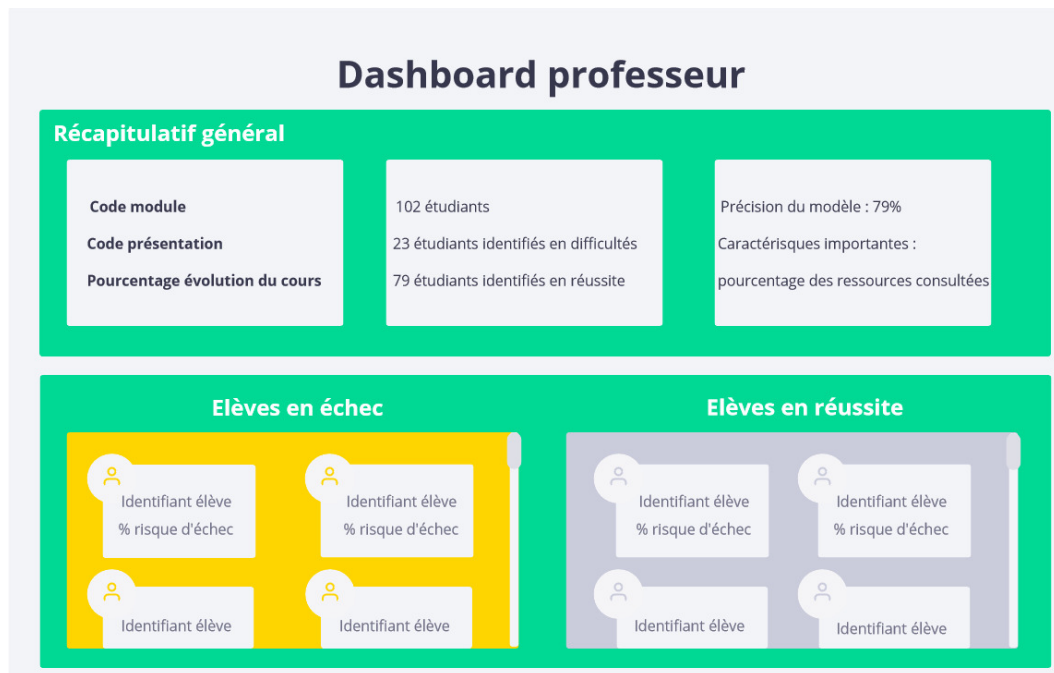
Écran Professeur :

- Titre : **Professeur**
- Formulaires : Deux champs de saisie avec des icônes de calendrier et d'horloge à gauche. Le premier est étiqueté "Code module" et le second "Code présentation".
- Progression : Un indicateur de progression à 80% avec une barre de progression et un pointeur.
- Bouton : Un bouton "Valider" en bas.

Écran Elève :

- Titre : **Elève**
- Formulaires : Un champ de saisie avec une icône de calendrier à gauche, étiqueté "Identifiant élève".
- Progression : Un indicateur de progression à 0% avec une barre de progression et un pointeur.
- Bouton : Un bouton "Valider" en bas.

- la page pour les professeurs (choix gauche de la page d'accueil) :



- la page pour les élèves (choix de droite de la page d'accueil) :



2. Choix des technologies utilisées

Python est le langage de programmation de la data science et celui avec lequel nous sommes le plus à l'aise. Pour coder l'interface, et respecter au mieux la partie préparatoire, nous avons privilégié html et css pour la partie front. Javascript aurait pu être utilisé pour rendre les pages

dynamiques. Pour le serveur et la gestion de l'application (le back), nous avons utilisé la librairie flask sous Python.

III. Guide utilisateur

1. Environnement technique

Le projet est entreposé sur github : <https://github.com/JuliCou/IHM>

Il a été développé avec Python 3.8. D'autres versions de Python peuvent très certainement fonctionner mais sans recherche de notre part. Pour faire tourner le projet, nous avons créé un environnement virtuel pour gérer les différentes versions de librairies.

Il suffit, en ligne de commande de réaliser les différentes commandes suivantes :

```
$ git clone https://github.com/JuliCou/IHM
$ cd IHM
$ virtualenv venv
$ source venv/Scripts/activate
$ pip install -r requirements.txt
```

La 4^{ème} commande sert à activer l'environnement virtuel, elle fonctionne sur Windows, sur Linux, il faudra très certainement l'adapter : source venv/bin/activate

Une fois les différentes installations prêtes, le lancement de l'interface est possible. Il faut taper la commande suivante :

```
$ python api.py
```

Différentes instructions apparaissent. Cela peut prendre plusieurs secondes (environ 1 minute maximum), le temps de charger les différents fichiers appelés pour l'interface :

```

$ python api.py
C:\Users\juliette.courgibet\Documents\cours CNAM\3A\DV_IHM\projet\IHM\venv\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.preprocessing.label module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.preprocessing. Anything that cannot be imported from sklearn.preprocessing is now part of the private API.
  warnings.warn(message, FutureWarning)
C:\Users\juliette.courgibet\Documents\cours CNAM\3A\DV_IHM\projet\IHM\venv\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator LabelEncoder from version 0.20.3 when using version 0.23.2. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
* Serving Flask app "api" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
C:\Users\juliette.courgibet\Documents\cours CNAM\3A\DV_IHM\projet\IHM\venv\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.preprocessing.label module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.preprocessing. Anything that cannot be imported from sklearn.preprocessing is now part of the private API.
  warnings.warn(message, FutureWarning)
C:\Users\juliette.courgibet\Documents\cours CNAM\3A\DV_IHM\projet\IHM\venv\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator LabelEncoder from version 0.20.3 when using version 0.23.2. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
* Debugger is active!
* Debugger PIN: 164-311-750
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```

Une fois que l’instruction Running on <http://127.0.0.1:5000/> apparait cela signifie que l’API est prête.

Si on clique sur le lien (ctrl+C) ou on le copie/colle dans un navigateur, on aura accès à l’interface et des commentaires vont apparaître dans la console (terminal).

2. Structure des sources

Les sources contiennent plusieurs dossiers et un fichier exécutable correspondant à l’API. C’est le fichier principal qu’il faudra lancer. Le reste contient les fichiers nécessaires au fonctionnement de l’API.

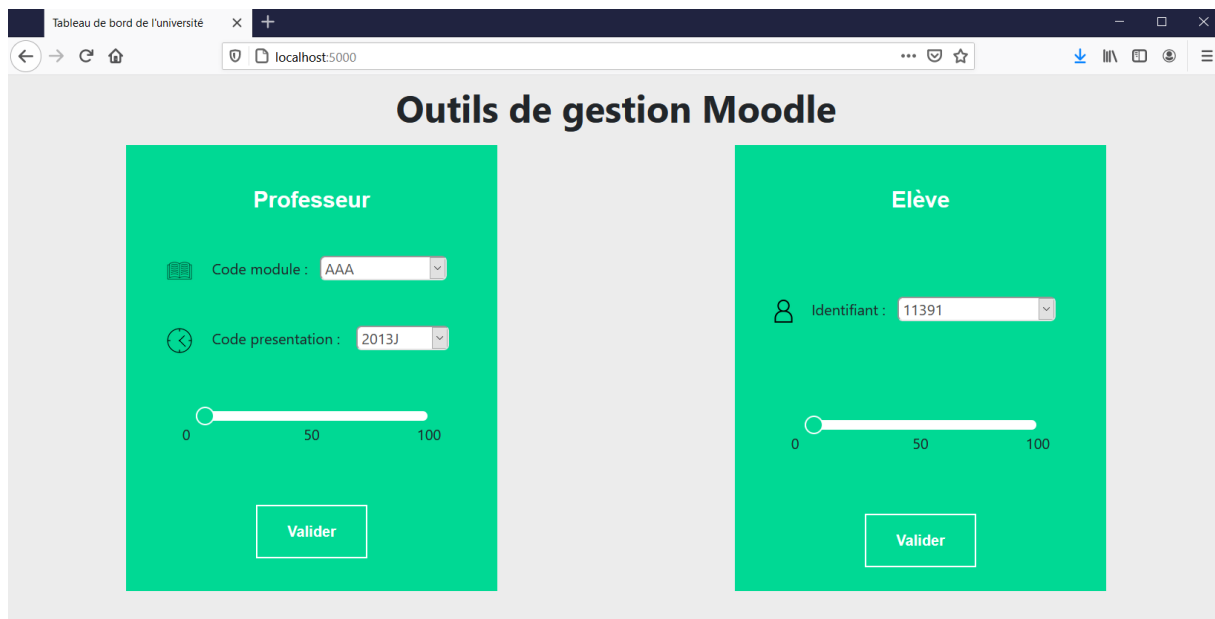
Le dossier `complete_dataframes` contient les 11 fichiers de données pour les 11 temps différents, avec chacun 32593 lignes et 39 colonnes. Le dossier `moodle_files` contient le fichier csv `studentInfo`, issu du site initial. Il contient l’ensemble des clés primaires (`id_student`, `code_presentation` et `code_module`). Le dossier `models` contient les 11 modèles entraînés et permettant de faire des prédictions de réussite à partir des données contenues dans les fichiers de `complete_dataframes`. Enfin, les deux derniers dossiers, `templates` et `static`, contiennent tous les fichiers permettant l’affichage de l’application.

Dans `templates`, on retrouve les 3 fichiers html pour les trois pages présentées (voir partie sur la préparation et définition de l’interface UI/UX). Le dossier `static` contient les images affichées sur le front (comme des logos) et le code css pour la gestion du style.

3. Navigation sur l'interface

Le navigateur Firefox a été utilisé pour interagir avec l'interface. Le rendu est différent d'un navigateur à l'autre dû aux différences d'interprétation du css.

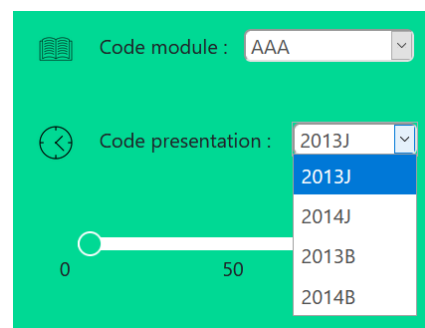
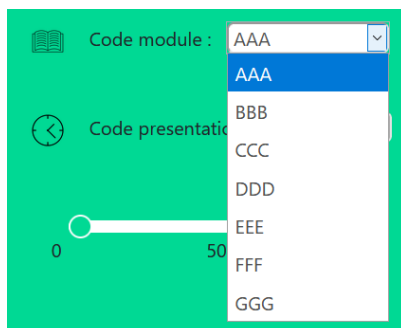
a. Page d'accueil



La page d'accueil contient le nom de l'onglet : « Tableau de bord de l'université ».

En haut de la page, en gras, le titre « Outils de gestion Moodle ».

Deux boîtes vertes prennent l'entièreté de la page. La boîte de gauche correspond au choix professeur. Un professeur remplit le cours (code module) et l'année (code présentation). Il s'agit de listes déroulantes :

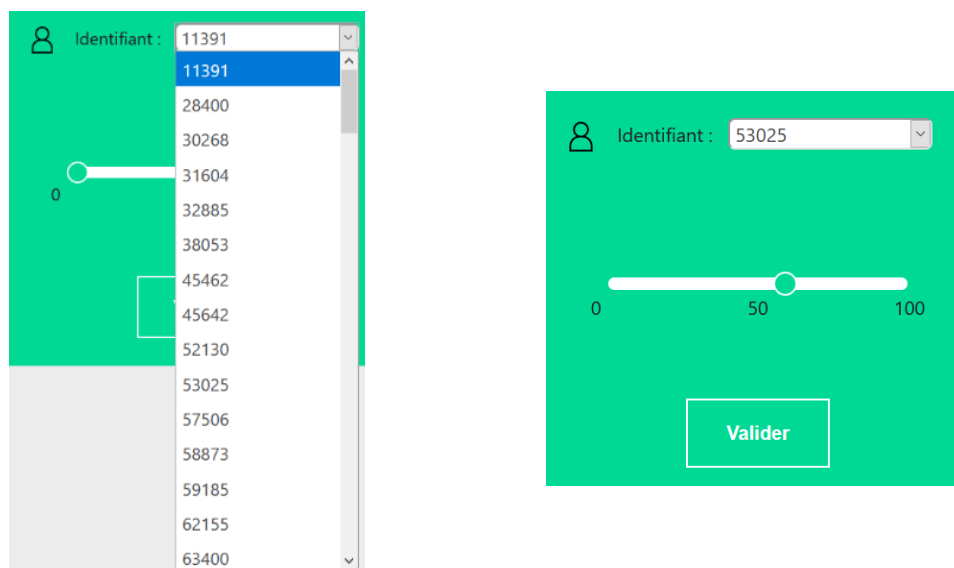


On peut cliquer dessus avec une souris et faire notre choix soit avec la souris, soit avec les flèches du clavier.

Le troisième paramètre est mis sous forme de barre de progression (slidebar). Il s'agit du temps de progression du module. Le choix le plus à gauche correspond à $t=0\%$ et le plus à droite à $t=100\%$.

Après avoir sélectionné les 3 paramètres, on appuie sur le bouton « valider ». Cela va charger la page profil d'utilisateur professeur (voir point b).

Dans l'autre cas (profil élève), il s'agit de la boîte à droite. Seulement deux paramètres sont nécessaires dans ce cas : l'identifiant de l'élève et la barre de progression. L'identifiant de l'élève est une liste déroulante aussi :



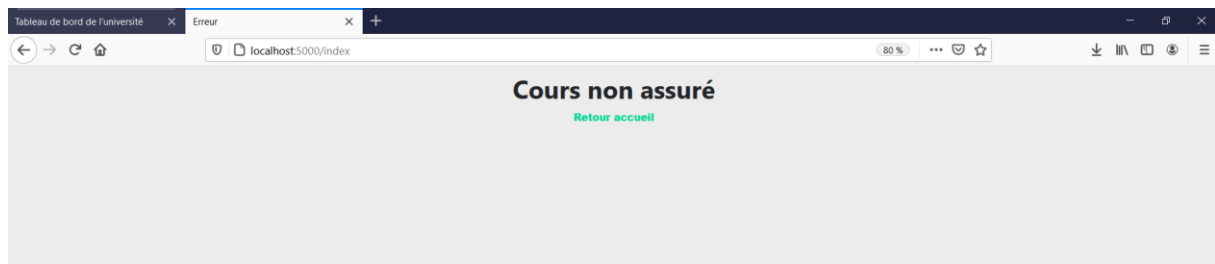
Après avoir sélectionné l'identifiant de l'étudiant, ici 53025, on peut changer la valeur de la slidebar, (ici 60%), et valider notre choix. Cela affichera l'interface présentée au point c.

b. Profil d'utilisateur professeur

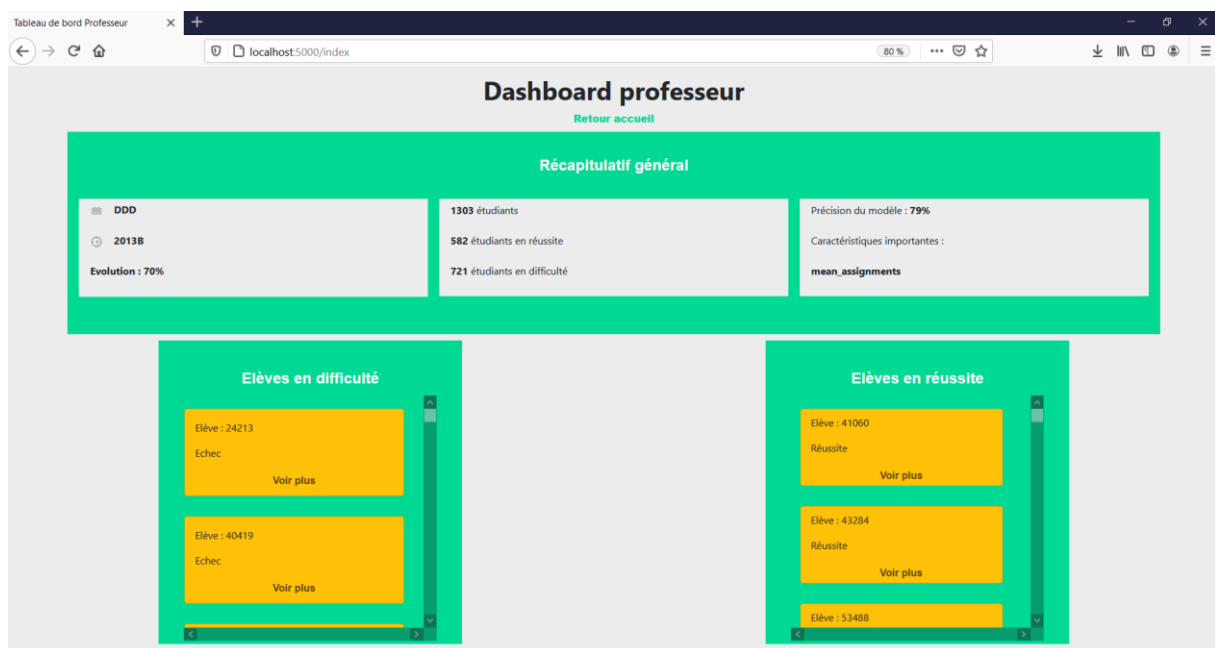
Une fois le cours choisi (code présentation et code module), ainsi que le temps d'évolution du cours, le tableau de bord du professeur apparaît.

Il y a deux possibilités : soit le cours existe, soit il est inexistant.

Dans le cas où le cours serait inexistant (par exemple GGG en 2013B), la page suivante s'affiche :



L'utilisateur a l'information selon laquelle le cours n'est pas assuré et peut revenir à l'accueil.
Dans le cas où le cours est bien assuré, on peut voir le dashboard du professeur :

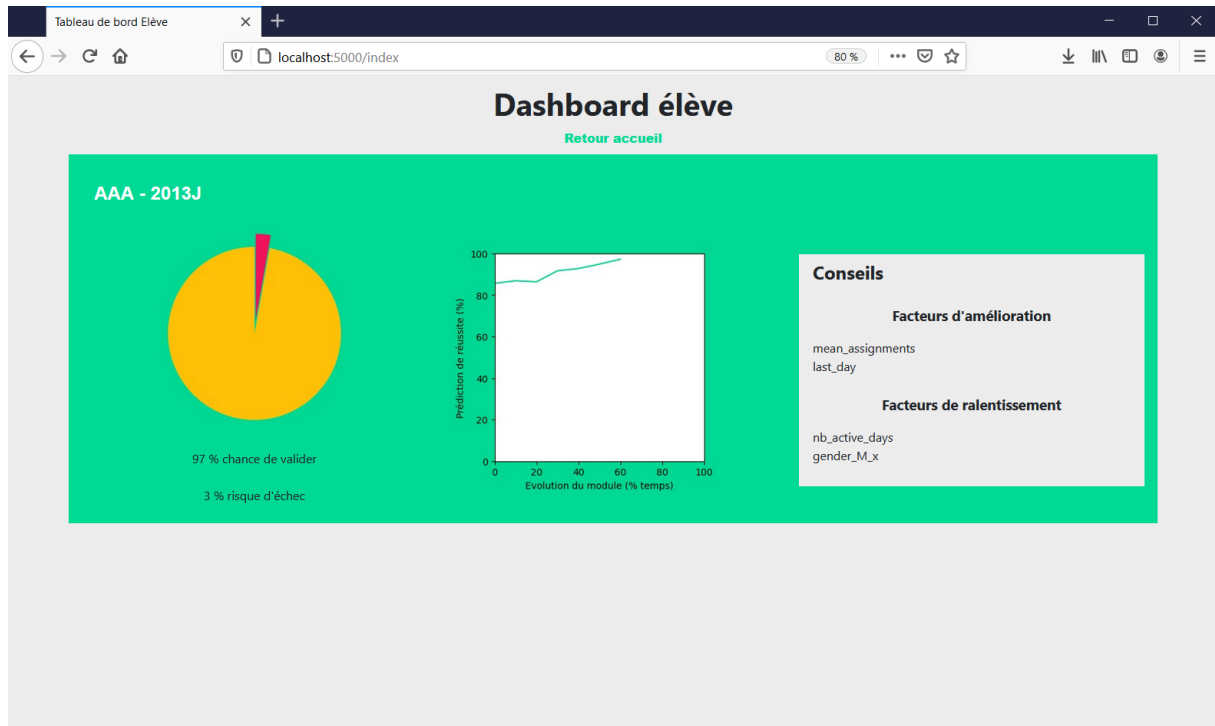


La page est composée de 3 zones principales :

- La première boîte correspond au récapitulatif général. On peut lire dans l'encadré à gauche les informations de la requête initiale (code module, code présentation et temps d'évolution). Dans l'encadré du milieu, on a une lecture sur le nombre d'étudiants total du cours en question avec la répartition des élèves classés en réussite et ceux classés en difficulté (prédiction d'échec). Enfin la dernière boîte, à droite, indique quelle est la précision du modèle (ici 70% de prédictions correctes) et donne quel est l'indicateur ayant le plus d'impact dans les prédictions (ici la moyenne des rendus des élèves).
- La deuxième boîte à gauche correspond à l'ensemble des élèves en difficulté. Le prof a une visibilité sur chaque étudiant (une carte par étudiant) avec l'identifiant et un lien vers le profil de l'élève (en cliquant sur « voir plus »).
- La troisième boîte à droite correspond à l'ensemble des élèves en réussite. Le professeur peut de même voir le profil de chaque étudiant avec le lien Voir plus.

Le lien vers le profil de l'étudiant amène vers le tableau de bord de l'élève. Cependant, en affichant le profil de l'élève, on ne verra que le module recherché initialement, et non l'ensemble des modules suivis par l'élève.

c. Profil d'utilisateur élève



Ici on affiche le profil de l'élève 53025. On voit qu'il ne suit qu'un module (1 seule boîte verte). Il suit le module AAA avec le code présentation 2013J.

A gauche, nous affichons un pie chart ou camembert, avec en rouge la proportion de prédiction d'échec et en vert la proportion de prédiction de réussite. Ici, nous prédisons 97% chance de valider à 60% d'évolution du module.

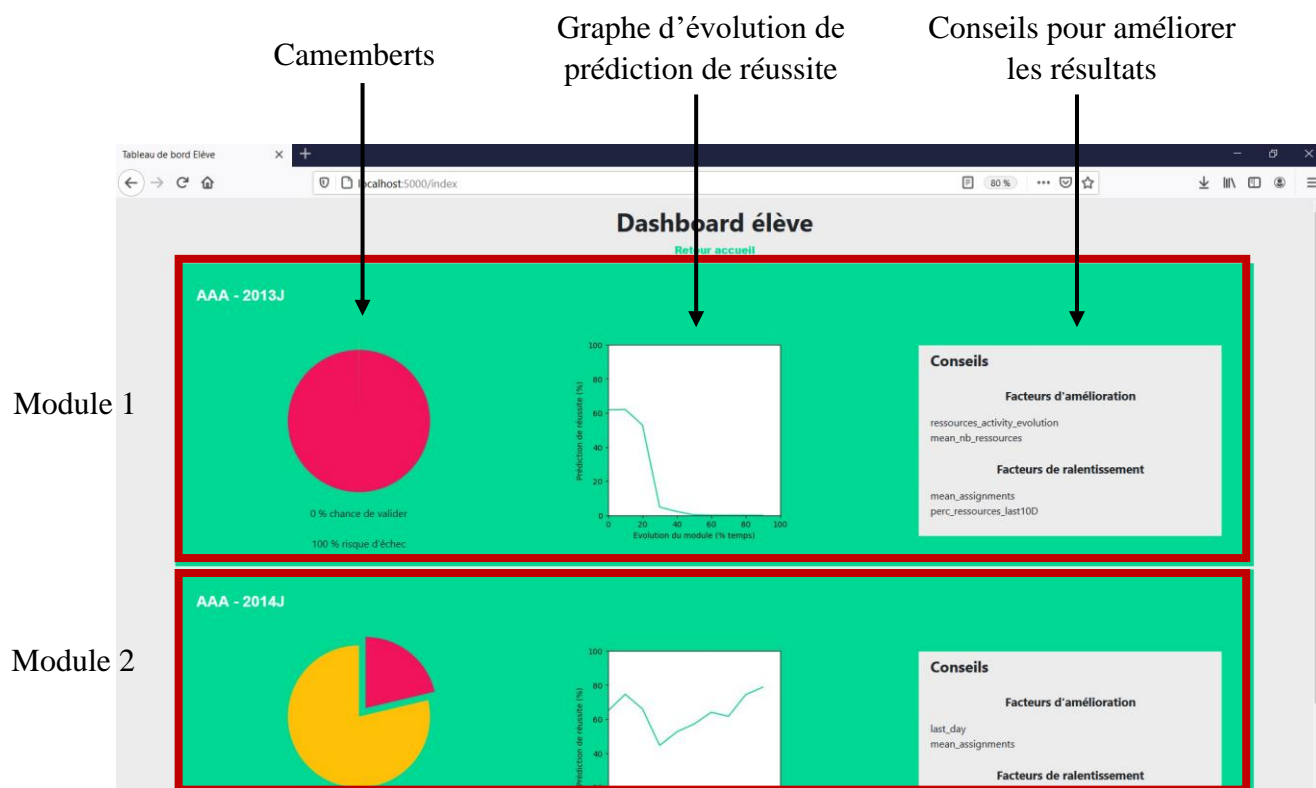
Au milieu, nous affichons un graphe avec en abscisse l'évolution du module (de 0 à 100%) et en ordonnée le pourcentage prédit de réussite. On voit avec cet exemple, que cette prédiction n'a fait qu'augmenter.

Enfin, dans le dernier encart, à droite, nous affichons les conseils pour améliorer les scores. Cela correspond à l'explication de la prédiction. Ainsi, la moyenne des rendus (assignments) est suffisamment haute pour prédire une réussite future. De plus, l'étudiant s'est connecté très récemment (assiduité) avec la variable last_day. On affiche aussi facteurs de ralentissement. Le nombre total de jours actifs semble bas (donc assiduité hétérogène dans le temps) et le genre de la personne.

D'autres identifiants d'étudiants permettent d'afficher deux modules :

65002, 94961, 129955, 135335, 135400, 141377, 147756, 148993, 155984, 159954, 188278, 235507

En affichant la page pour l'élève 235507, on voit bien deux modules apparaître.



Celui en haut correspond au module AAA pour le temps 2013J. Et le second module correspond aussi au cours AAA pour le temps 2014J.

Enfin, juste en dessous du titre de la page, « Dashboard élève », il y a le lien « Retour accueil » qui permet de revenir sur la page d'accueil (voir partie a).