International Journal of Innovation and Scientific Research
ISSN 2336-0046 Vol. 77 No. 1 Feb. 2025, pp. 129-138

http://www.ijisr.issr-journals.org/

Prédiction de la probabilité de réussite en sciences informatiques et orientation départementale

[Prediction of the probability of success in computer science and departmental orientation]

Mavuela Maniansa Richard

Institut Supérieur Pédagogique et Techniques de Kinshasa, RD Congo

ABSTRACT: Through an observation of student's performance in computer science, we want to reveal that the factors influencing success can be identified and modeled using advanced machine learning techniques.

KEYWORDS: orientation, predictive modeling, data analysis.

RESUME: A travers une observation de la performance des étudiants en sciences informatiques, nous voulons révéler que les facteurs y influençant la réussite peuvent être identifiés et modélisés grâce aux techniques avancées de l'apprentissage automatique.

MOTS-CLEFS: orientation, modélisation prédictive, analyse des données.

1 INTRODUCTION

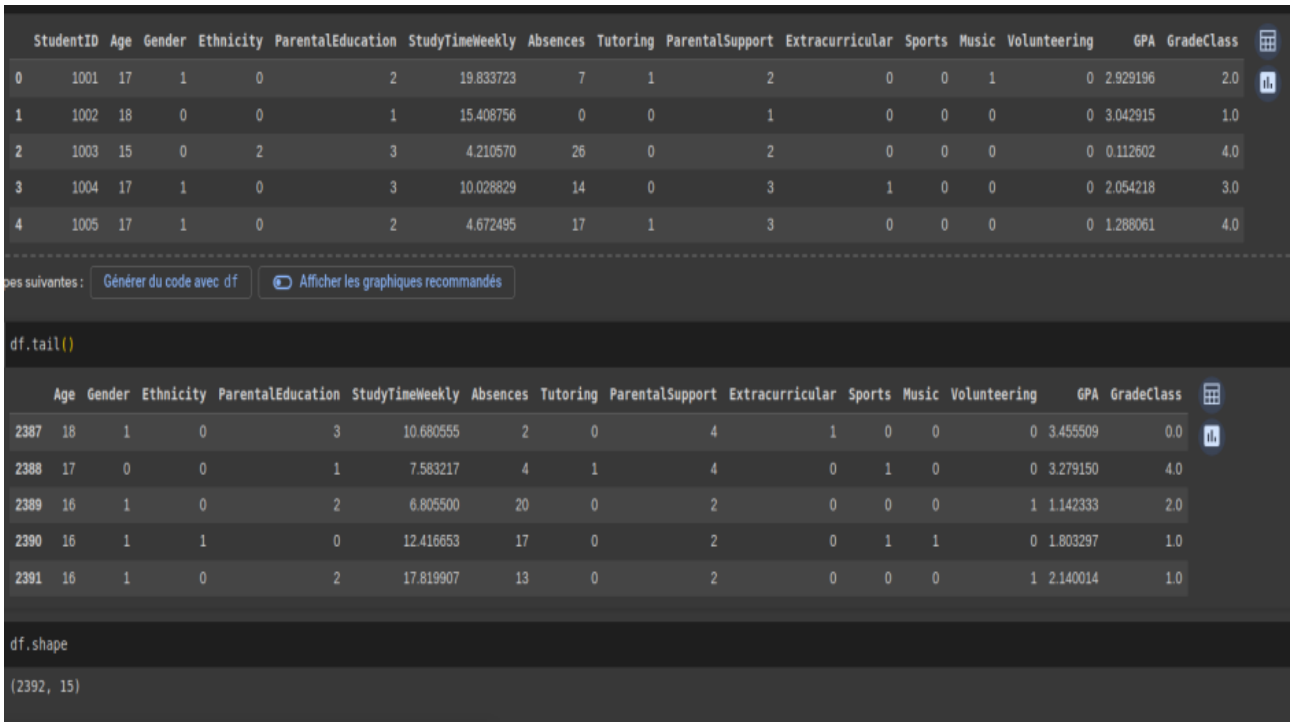
La réussite académique est essentielle pour le développement personnel et professionnel, en particulier dans les sciences informatiques. Elle dépend de nombreux facteurs, notamment des caractéristiques personnelles et de l'environnement socio-économique des étudiants. Ainsi, pour mieux orienter les étudiants, comprendre ces variables est crucial [1]. Cependant, les données de performance des post-bacheliers offrent des insights détaillés sur ces facteurs, incluant l'âge, le genre, l'origine ethnique, le niveau d'éducation des parents, le temps d'étude, l'absentéisme, le soutien parental et les activités parascolaires. Chaque variable permet d'analyser ce qui favorise ou entrave la réussite académique. En effet, l'analyse de ces données peut révéler des corrélations significatives, essentielles pour développer des modèles prédictifs capable d'estimer le succès d'un post-bachelier souhaitant s'inscrire dans la faculté des sciences informatiques. Ces outils prédictifs peuvent servir de support aux conseils académiques car anticiper les performances académiques réduira le taux d'échec et aidera à mettre en place des programmes d'intervention pour les étudiants à risque, favorisant une culture d'entraide [2]. Dans cette étude, nous explorerons différentes méthodes de prédiction des performances académiques en utilisant des techniques statistiques et d'apprentissage automatique.

2 METHODE

En vue de développer un modèle prédictif efficace, nous utiliserons une approche méthodologique impliquant plusieurs étapes clés:

2.1 COLLECTE DES DONNÉES

Nous avons collecté un ensemble de données qui contient des informations complètes sur 2 392 post-bacheliers, détaille leurs données démographiques, leurs habitudes d'étude, la participation des parents, leurs activités parascolaires et leurs résultats scolaires.



	StudentID	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular	Sports	Music	Volunteering	GPA	GradeClass
0	1001	17	1	0	2	19.833723	7	1	2	0	0	1	0	2.929196	2.0
1	1002	18	0	0	1	15.408756	0	0	1	0	0	0	0	3.042915	1.0
2	1003	15	0	2	3	4.210570	26	0	2	0	0	0	0	0.112602	4.0
3	1004	17	1	0	3	10.028829	14	0	3	1	0	0	0	2.054218	3.0
4	1005	17	1	0	2	4.672495	17	1	3	0	0	0	0	1.288061	4.0

	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular	Sports	Music	Volunteering	GPA	GradeClass
2387	18	1	0	3	10.680555	2	0	4	1	0	0	0	3.455509	0.0
2388	17	0	0	1	7.583217	4	1	4	0	1	0	0	3.279150	4.0
2389	16	1	0	2	6.805500	20	0	2	0	0	0	1	1.142333	2.0
2390	16	1	1	0	12.416653	17	0	2	0	1	1	0	1.803297	1.0
2391	16	1	0	2	17.819907	13	0	2	0	0	0	1	2.140014	1.0

Fig. 1. Jeu de données

Et nous avons les colonnes suivantes:

StudentID: Un identifiant unique attribué à chaque étudiant (1001 à 3392).

Age (Âge): L'âge des étudiants varie de 15 à 18 ans.

Gender (Sexe): Sexe des élèves, où 0 représente la femme et 1 représente l'homme.

Ethnicity (Origine ethnique): L'origine ethnique des élèves, codée comme suit:

0: Africain

1: Américain

2: Asiatique

3: Européen

ParentalEducation (Éducation parentale): Le niveau d'éducation des parents, codé comme suit:

0**:** Aucun;

1: Lycée;

2: Un peu d'université;

3: Baccalauréat;

4: Plus élevé

StudyTimeWeekly: Temps d'étude hebdomadaire en heures, allant de 0 à 20.

Absences: Nombre d'absences au cours de l'année scolaire, allant de 0 à 30.

Tutoring (Tutorat): statut du tutorat, où 0 indique Non et 1 indique Oui.

ParentalSupport (Soutien parental): Le niveau de soutien parental, codé tel que:

0: Aucun;

1: Faible;

2: Modéré;

3: Élevé;

4: Très élevé

Extracurricular (Activités parascolaires): Participation à des activités parascolaires, où 0 indique Non et 1 indique Oui.

Sports: Participation à des sports, où 0 indique Non et 1 indique Oui.

Music (Musique): Participation à des activités musicales, où 0 indique Non et 1 indique Oui.

Volunteering (Volontariat): Participation au bénévolat, où 0 indique Non et 1 indique Oui.

GPA (Grade Point Average): Moyenne générale de l'étudiant sur une échelle de 0 à 4,0, influencée par les habitudes d'étude, l'implication des parents et les activités parascolaires.

GradeClass: Classification des notes des élèves en fonction de la moyenne générale:

0: « A » (moyenne >= 3,5);

1: « B » (3,0 <= GPA < 3,5);

2: « C » (2,5 <= GPA < 3,0);

3: « D » (2,0 <= GPA < 2,5);

4: « F » (GPA < 2.0)

- Entrepôt de données

Le modèle type d'entrepôt des données de notre système est celui-ci dessous:

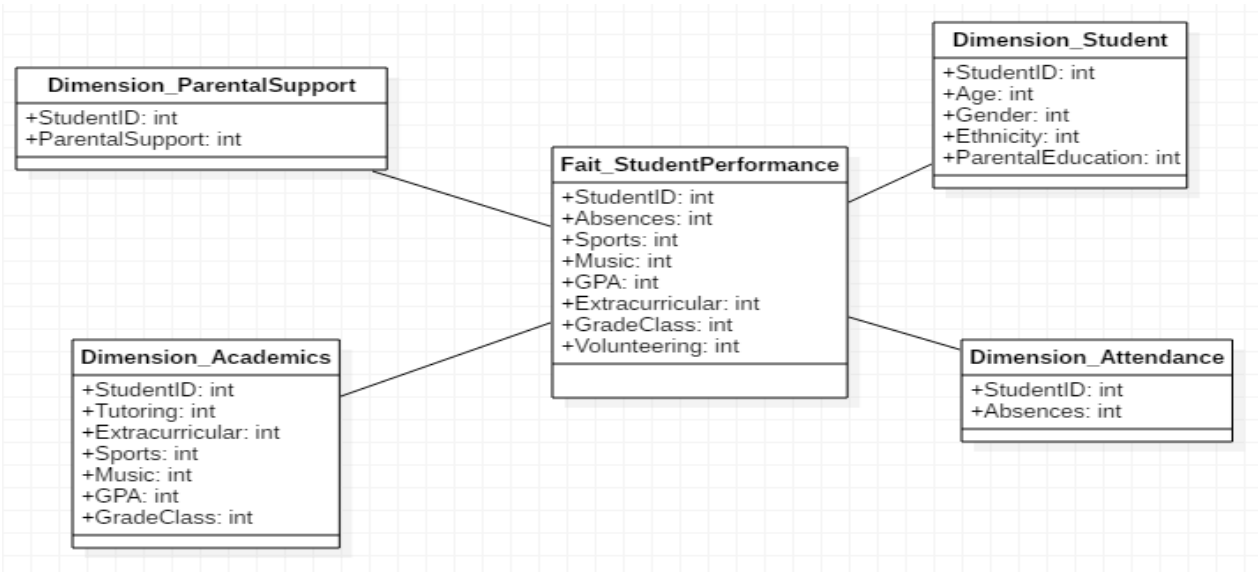


Fig. 2. Entrepôt de données

Nous avons séparé les données en plusieurs dimensions pour faciliter l'analyse.

- La dimension 'Student' contient les informations de base sur les élèves.
- La dimension 'Attendance' contient les informations sur leur absence.

- La dimension ‘Academics’ contient les informations liées à la performance académique des élèves.
- La dimension ‘ParentalSupport’ contient les informations sur le soutien parental.
- La table de faits ‘StudentPerformance’ relie toutes ces dimensions et contient les mesure contient les mesures clés comme GPA, GradeClass, etc.

Ce modèle est un **modèle logique en étoile** qui met en relation les dimensions ainsi que les faits et facilite l’analyse sur les données [3].

2.2 Prétraitement Et Exploration Des Données

2.2.1 Prétraitement Des Données

Notre jeu de données a 2392 lignes et 15 colonnes, les toutes les données sont numériques et sans ligne en double.

```
[ ] #afficher toutes les colonnes
    print(data.columns)

Index(['StudentID', 'Age', 'Gender', 'Ethnicity', 'ParentalEducation',
       'StudyTimeWeekly', 'Absences', 'Tutoring', 'ParentalSupport',
       'Extracurricular', 'Sports', 'Music', 'Volunteering', 'GPA',
       'GradeClass'],
      dtype='object')
```

```
duplicate_rows_df = df[df.duplicated()]
print("nombre de lignes en double :", duplicate_rows_df.shape)

nombre de lignes en double : (0, 14)
```

Fig. 3. Prétraitement 1

Ni présence de valeurs double et valeur aberrante

```
df.isnull().sum()

Age                0
Gender             0
Ethnicity          0
ParentalEducation  0
StudyTimeWeekly    0
Absences           0
Tutoring           0
ParentalSupport    0
Extracurricular    0
Sports             0
Music              0
Volunteering       0
GPA                0
GradeClass         0
dtype: int64
```

Fig. 4. Prétraitement 3

```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

```
Age                 2.000000
Gender              1.000000
Ethnicity           2.000000
ParentalEducation   1.000000
StudyTimeWeekly     9.365330
Absences           15.000000
Tutoring            1.000000
ParentalSupport     2.000000
Extracurricular     1.000000
Sports              1.000000
Music               0.000000
Volunteering        0.000000
GPA                 1.447413
GradeClass          2.000000
dtype: float64
```

Fig. 5. Prétraitement 4

2.2.2 EXPLORATION DES DONNÉES

```
data_gender= data['Gender']. value_counts()
print(data_gender)
```

```
Gender
1    1222
0    1170
Name: count, dtype: int64
```

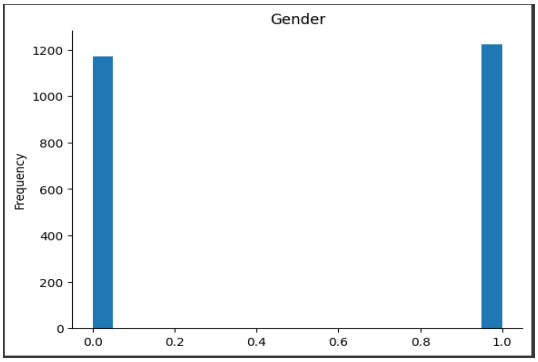


Fig. 6. Count_Gender

Sur cette ligne de code, nous avons compté le nombre de fois que chaque genre apparait dans la colonne ‘Gender’ et avons affiché les résultats du décompte des valeurs uniques. En sortie nous avons 1222 Hommes et 1170 Femmes.

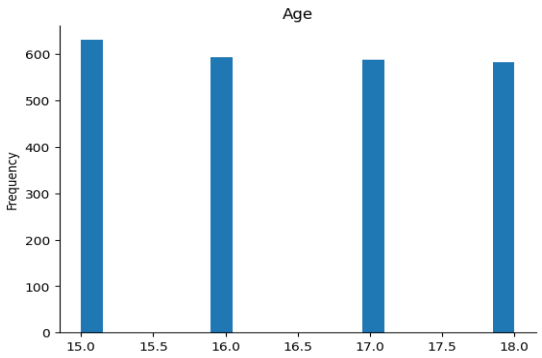


Fig. 7. Age

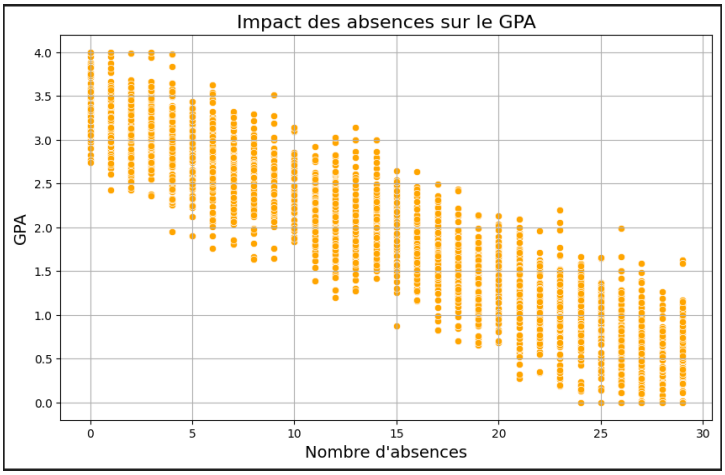


Fig. 8. Absence par GPA

[Figure 8] L'axe des abscisses représente la Fréquence (Frequency) allant de 0 à 600, et l'axe des ordonnées montre l'âge des étudiants variant de 15 à 18 ans. Les barres sont colorées selon la variable "Age". On constate que les individus ayant 15ans d'âge sont plus nombreux que ceux ayant 16ans, 17ans et 18ans.

[Figure 9] Sur ce graphique de dispersion, l'axe des abscisses représente le nombre d'absences, allant de 0 à 30, tandis que l'axe des ordonnées représente le GPA (Grade Point Average), allant de 0 à 4. Chaque point représente un étudiant avec son nombre d'absence, coloré selon le genre (0 ou 1). On observe une tendance négative, où une augmentation du nombre d'absences est associée à une diminution du GPA. Les points sont densément regroupés, montrant une corrélation forte entre les absences et la performance académique.

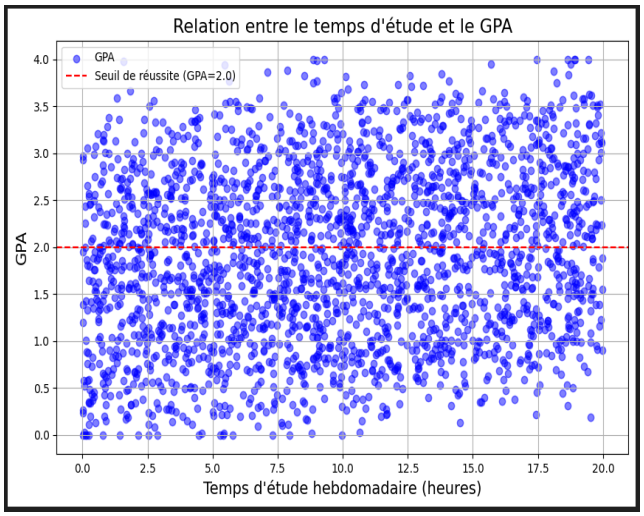


Fig. 9. Relation Temps d'étude et GPA

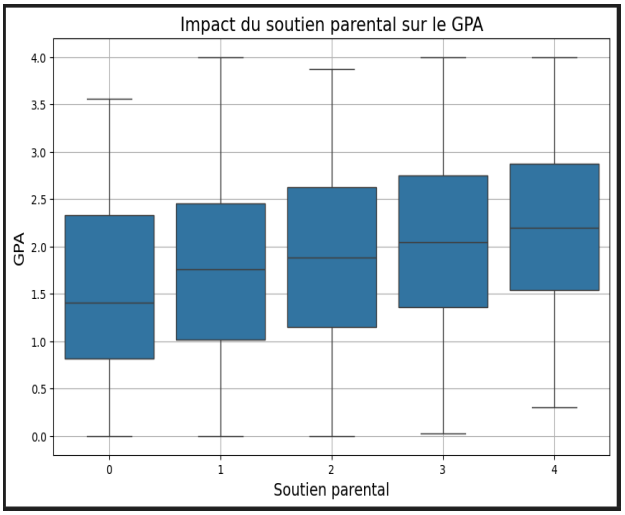


Fig. 10. Impact soutien parental

[Figure 10] Un nuage de points qui visualise la relation entre le temps d'étude hebdomadaire et la performance académique (GPA) des étudiants, avec la ligne indiquant le seuil de réussite. Les éléments du graphique sont soigneusement étiquetés et nous observons une hausse de GPA où il y a plus de temps d'étude et basse où il y a moins de temps d'étude ce qui revient à dire que le temps d'étude influence la performance soit en bien ou en mal.

[Figure 11] Un boxplot qui montre l'impact du soutien parental sur le GPA des étudiants. Ce boxplot permet de visualiser la médiane, les quartiles et les valeurs aberrantes du GPA pour chaque catégorie de soutien parental. Les éléments graphiques sont soigneusement étiquetés pour faciliter la compréhension et analyse des données.

2.3 MODÉLISATION ET INTERPRÉTATION DES RÉSULTATS

Tout d'abord, nous avons créé notre sujet de prédiction, la variable 'Réussite' qui est notre variable cible étant donné qu'elle ne figurait pas dans notre jeu de données collectées.

```
data ['Réussite'] = (data ['GPA'] > 2.0).astype (int)
```

Cette ligne de code crée une nouvelle colonne Réussite dans notre jeu de données, où chaque étudiant est marqué avec 1 s'il a réussi (GPA > 2.0) et un 0 s'il n'a pas réussi.

Ensuite, avons testé plusieurs algorithmes de machine learning, tels que:

→ Régression logistique

```
X = sm.add_constant (data ['StudyTimeWeekly']) # Ajout d'une constante à la
variable indépendante 'StudyTimeWeekly'

model = sm.Logit (data ['Réussite'], X) # Création d'un modèle de régression
logistique

result = model.fit () # Ajustement du modèle de régression logistique
```

Nous avons:

- Ajouter une constante à la variable indépendante 'StudyTimeWeekly'.

Dans notre cas, l'inclusion de la constante nous aide à mieux comprendre à quel niveau la variable indépendante 'Temps d'étude hebdomadaire' influence la variable cible/dépendante 'Réussite' en nous fournissant des estimations plus fiable; Ce qui améliorent la validité des test d'hypothèses.

La constante représente la valeur prédit lorsque toutes les variables sont à zéro (cela permet de capturer le niveau basse de la variable indépendant).

- Créer un modèle de régression logistique pour prédire la variable cible 'Réussite à partir de la variable indépendante définie dans X. Ici le modèle est préparé à l'entrainement.
- Enfin ajuster le modèle créé aux données pour la prédiction sur des nouvelles données en optimisant les paramètres du modèle en vue de minimiser l'erreur, ce qui améliore la capacité du modèle à fournir des résultats fiables.

Résultat

```
Résumé des résultats de la régression logistique :
                           Logit Regression Results
==============================================================================
Dep. Variable:               Réussite   No. Observations:                 2392
Model:                          Logit   Df Residuals:                     2390
Method:                           MLE   Df Model:                            1
Date:                Thu, 26 Sep 2024   Pseudo R-squ.:                0.01040
Time:                        15:56:34   Log-Likelihood:                -1635.7
converged:                       True   LL-Null:                       -1652.9
Covariance Type:            nonrobust   LLR p-value:                 4.556e-09
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const             -0.5506      0.083     -6.605      0.000      -0.714      -0.387
StudyTimeWeekly    0.0428      0.007      5.824      0.000       0.028       0.057
==============================================================================
```

Fig. 11. Résultat Logistique

Ces résultats indiquent que le modèles est ajusté avec succès et que le variable Temps d'étude z un effet significatif sur la variable Réussite.

→ Régression linéaire multiple.

```
model = sm.OLS (y, X).fit ()
```

OLS: Ordinary Least Squares

Nous avons effectué une régression linéaire pour prédire une variable continue (y) à partir de plusieurs variables indépendantes.

Ce modèle suggère que le temps d'étude, le soutien parental, le tutorat, la participation extrascolaire et les absences sont des facteurs importants influençant la réussite des étudiants.

Résultat

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    GPA   R-squared:                       0.941
Model:                            OLS   Adj. R-squared:                  0.941
Method:                 Least Squares   F-statistic:                     7580.
Date:                Thu, 26 Sep 2024   Prob (F-statistic):               0.00
Time:                        15:56:50   Log-Likelihood:                198.89
No. Observations:                2392   AIC:                            -385.8
Df Residuals:                    2386   BIC:                            -351.1
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const              2.5959      0.015    169.188      0.000       2.566       2.626
StudyTimeWeekly    0.0291      0.001     36.039      0.000       0.028       0.031
ParentalSupport    0.1526      0.004     37.552      0.000       0.145       0.161
Tutoring           0.2498      0.010     25.137      0.000       0.230       0.269
Extracurricular    0.1872      0.009     19.961      0.000       0.169       0.206
Absences          -0.0994      0.001   -184.524      0.000      -0.100      -0.098
==============================================================================
Omnibus:                        1.880   Durbin-Watson:                   2.001
Prob(Omnibus):                  0.391   Jarque-Bera (JB):                1.916
Skew:                           0.050   Prob(JB):                        0.384
...
==============================================================================
```

Fig. 12. Résultat Linéaire

En fin, Nous avons évalué la probabilité d'admission d'un nouvel étudiant en se basant sur des étudiants similaires et leurs performances passées.

```
# Fonction de prédiction basée sur la similarité
def predict_student_admission (new_student_data, selected_department):
new_student_scaled = scaler.transform ([new_student_data])
distances = pairwise_distances (X_scaled, new_student_scaled)
nearest_indices = np.argsort (distances.flatten ()) [: 5]
nearest_students = data.iloc [nearest_indices]
admission_probabilities = nearest_students ['GPA'].mean ()
department_info = departments ["Informatique"], [selected_department]
gpa_range = department_info ["gpa_range"]
```

En utilisant ces données en entrée, la fonction va calculer les distances entre les données du nouvel étudiant et les étudiants existants ensuite prédire les chances d'admission en se fondant la moyenne des GPA des étudiants les plus proches ainsi que critères d'admission du département choisi.

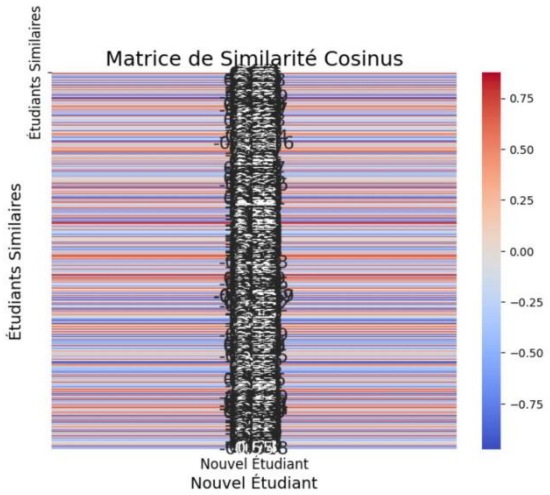


Fig. 13. Matrice de similarité

2.4 DÉPLOIEMENT

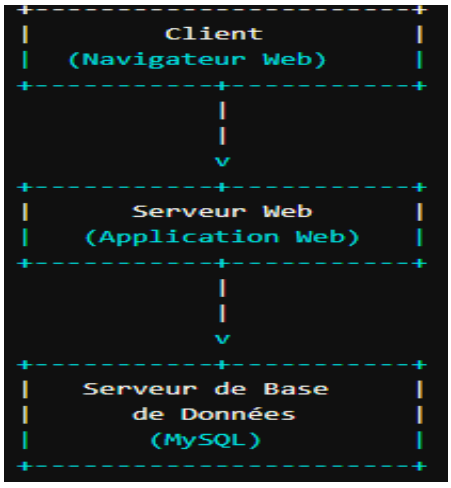


Fig. 14. Déploiement

3 SIMULATION

L'interface utilisateur est programmée sous python grâce à la bibliothèque tkinter, une librairie standard qui permet de concevoir des applications avec des éléments interactifs, des champs de texte, etc. [4] pour notre cas, il nous a fourni une interface utilisateur conviviale pour l'interaction qui permet aux utilisateurs de saisir ses informations afin de prédire son admission dans un département donné en évaluant la similarité des données.

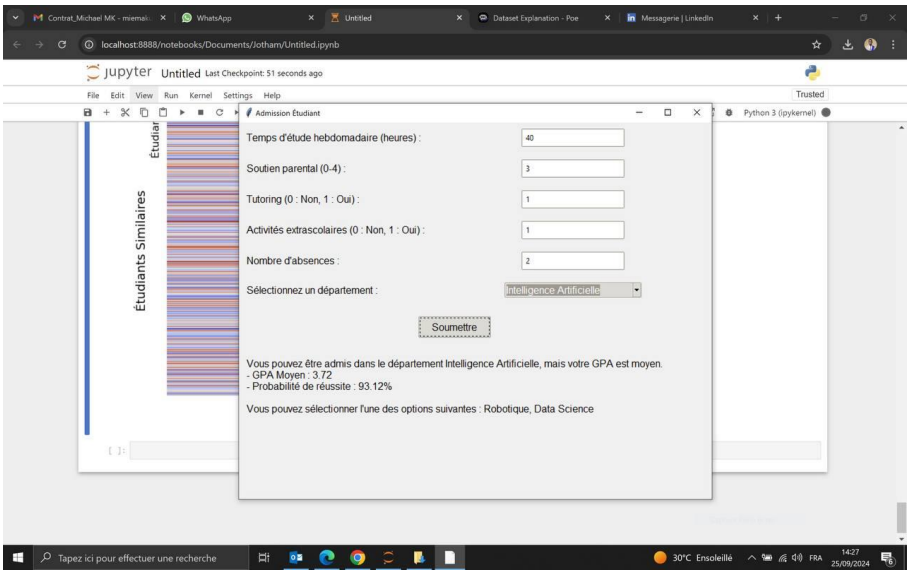


Fig. 15. Réalisation

4 CONCLUSION

Dans notre étude nous avons voulu faciliter le processus d'orientation à l'aide d'algorithme de machine Learning et du profil de l'apprenant, à l'ère de l'émergence technologique. Bien que les conseillers d'orientation effectuent déjà ce travail, nous avons proposé un Framework (modèle) compte tenu du nombre élevé des post bacheliers souhaitant adhérer l'écosystème des sciences informatiques ainsi que des facteurs influençant la réussite. Afin de déterminer les valeurs qui influencent la performance des étudiants, notre variable continu baptisé (y), nous avons testé différents modèles de prédiction [5] tels que la régression logistique et linéaire multiple dont l'un dit que la variable Temps d'étude a un effet significatif sur la variable Réussite et l'autre suggère que le temps d'étude, le soutien parental, le tutorat, la participation extrascolaire et les absences sont des facteurs importants influençant la réussite des étudiants. Après le test de ces deux modèles, nous pouvons remarquer que la régression linéaire multiple a prédit une variable continue (y) à partir de plusieurs variables indépendantes. Par conséquent, nous avons opté pour ce modèle au détriment de la logistique.

Puisqu'il s'agit d'une prédiction de la probabilité, nous ajoutons alors une fonction de similarité qui non seulement évalue la probabilité d'admission en se basant sur des étudiants similaires ainsi que leurs performances, elle fournit également des informations sur le département choisi et leurs exigences en moyenne générale (GPA) qui varient d'un intervalle de 2 à 4 (chaque département est assigné un intervalle de GPA influé par le temps d'étude, le soutien parental, le tutorat, la participation extrascolaire et les absences).

REFERENCES

[1] Chassagne, C. (2018). L'éducation et l'orientation: Chemins de formation. Paris: Magnard.
[2] Nguyen, T. N., Lucas, D., Artus, K. G., & Lares, S. (2010). Recommender system for predicting student performance. procedia Computer Science.
[3] Azencott, C. A. (2014). Introduction au Machine Learning.
[4] Lutz, M. (2013). Programming Python (éd. 3ème). Californie: O'Reilly Media.
[5] Massaron, L. (2016). Machine Learning (éd. 1ère). Dummies.