2023

RAPPORT PREDICTION DE PRÊT BANCAIRE

Encadrée par : Simplon , Sanofi

Lecarpentier Eden

SOMMAIRE

INTRODUCTION

Le projet :

Suite à ma formation en tant que développeur en intelligence artificiel est mon alternance chez Sanofi en tant que Assistant Power Platform Infra ingénieur je dois réaliser un projet en intelligence artificielle pour obtenir ma certification .

Mon projet consistera à déterminer si une personne peut être admissible à un prêt bancaire. Ce projet seras réalisée en utilisant un model de classification en machine learning .

Le besoin du client :

le client est un dirigeant d'une grande banque du nom Banceo . Il accorde souvent des prés mais pour certains clients il n'est pas sur s'il devrait .

Mon projet intervient maintenant en tant que data scientist nous allons construire un projet permettant de prédire si une personne auras besoin d'un pré ou non . Le modèle sera un support pour confirmer ce que la personne pense déjà est non un modèle 100 % utilisables sans aide humain et pour cela nous construirons également une application .
J'avais des points régulier avec mon client Guizar Arturo on organisée un meeting par mois pour voir l'évolution du projet et si j'avais d'autre questions durant se mois je contacté mon client par discord .

Contenue du rendue :

Un modèle d'intelligence artificielle bien entraîné .
Une application avec une ui et ux le plus facile à utiliser et efficace .
Un diaporama expliquant comment et le modèle etc.
Un rapport de 20 pages dans laquelle nous expliquons de manière concrète tout ce que l'on a réalisé pour obtenir notre résultat .

LES DONNÉES

Les données que je vais utiliser pour faire mon modèle de machine learning sont des données qui proviennent du site Kaggle.com .

Mon équipe à Sanofi était en No Code de ce fait je ne pouvais pas développer avec leur données .

Les données représentent différentes informations sur le client qui cherche à savoir s'ils peuvent prétendre à un prêt bancaire

les différentes colonnes sont les suivantes :

Loand_ID : L'identifiant du possible prêt .

Gender : Le genre homme ou femme .

Married : Marié ou non .

Dépendent : Dépendant ou non .

Éducation : Ont-ils un diplôme où sont-ils sans diplôme .

Self employed : On t'il un job fixe où sont il entrepreneur, freelance ect…

AppliantIncome : Combien gagne la personne qui demande le prêt .

CoapplicantIncome : Un Coapplicant est une personne de plus qui seras inclut dans le prêt . Il sera nécessaire de regarder les données du Coapplicant, là en l'occurrence on va regarder le montant qu'il gagne

loan Amount : Le montant du prêt .

LoanAmountTerm :C'est le temps nécessaire au payement du prêt .

Credit history : La capacité a remboursé et la responsabilité à payer les dettes .

Property area : Il habite dans un endroit rural ou urbain .

Property_Area : Il habite dans un endroit rural ou urbain .

Loan_Status : La target pour notre modèle de machine learning celle qui permet de savoir si oui ou non il on obtenue un près .
Target signifie la colonne dont on veut avoir la prédiction (la colonne ciblées)

BASSE DE DONNÉE SQL

Les données que je vais utiliser pour faire mon modèle de machine learning sont des données qui proviennent du site Kaggle.com .

Ces données seront ensuite incorporé dans une table nommée "loan_prediction" d'une base de donnée SQL .

Cela permet de pouvoir automatisée la basse de donnée avec le model de machine learning . Lorsque l'on utiliseras le modèles les données seront donc toujours a jours .

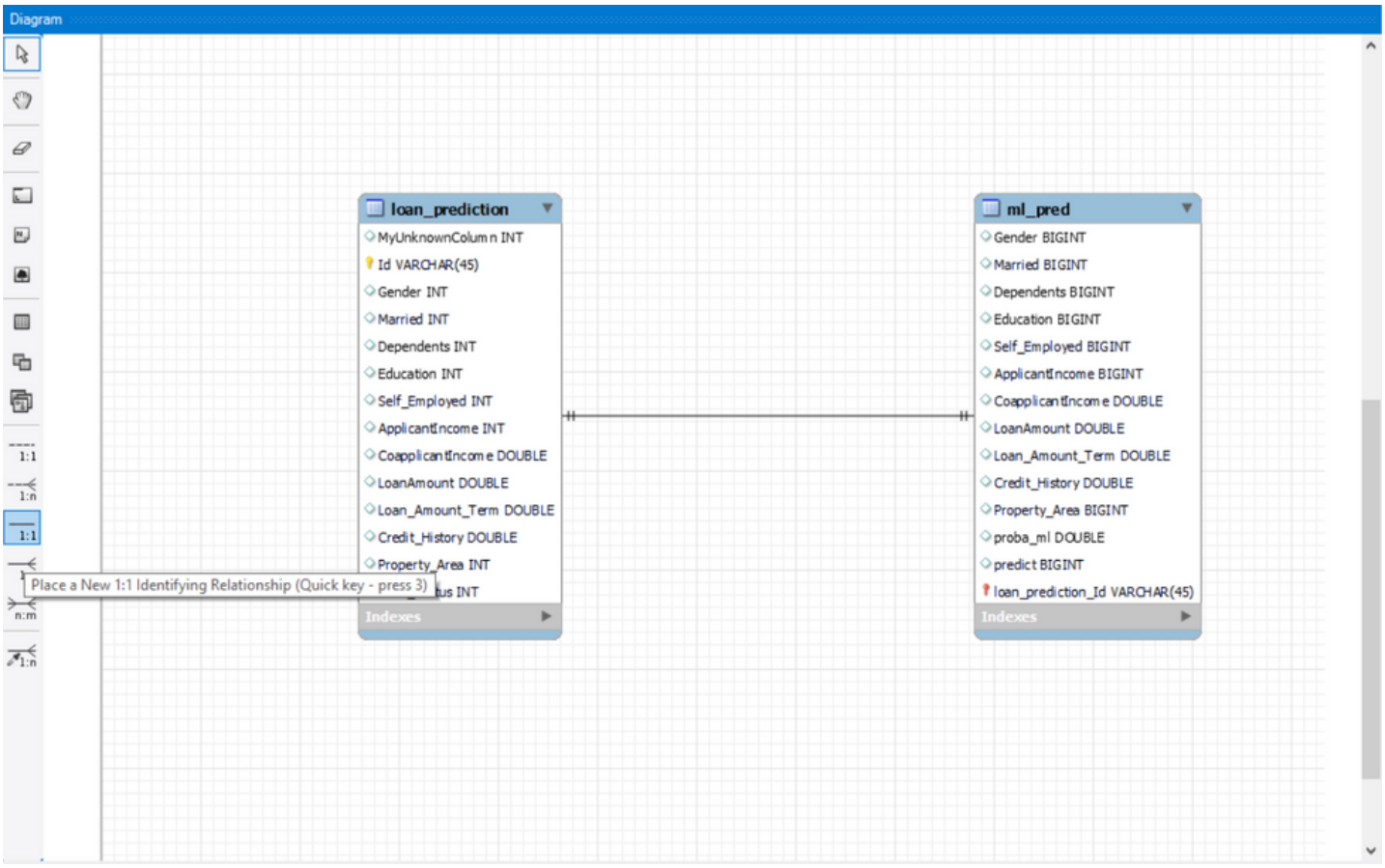
Les prédictions du modèles elle aussi sont stockée dans une table SQL nommée ml_pred dans la même basse de donnée que précédemment les données utilisée pour chaque prédiction seront aussi dans cette table .

Id	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
LP000001	1	1	0	1	0	2000	1900	700	360	0	0	1
LP001003	1	1	1	0	0	4583	1508	128	360	1	0	0
LP001005	1	1	0	0	1	3000	0	66	360	1	2	1
LP001006	1	1	0	1	0	2583	2358	120	360	1	2	1
LP001008	1	0	0	0	0	6000	0	141	360	1	2	1
LP001011	1	1	2	0	1	5417	4196	267	360	1	2	1
LP001013	1	1	0	1	0	2333	1516	95	360	1	2	1
LP001014	1	1	3	0	0	3036	2504	158	360	0	1	0
LP001018	1	1	2	0	0	4006	1526	168	360	1	2	1
LP001020	1	1	1	0	0	12841	10968	349	360	1	1	0
LP001024	1	1	2	0	0	3200	700	70	360	1	2	1
LP001028	1	1	2	0	0	3073	8106	200	360	1	2	1

Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	predict	proba_ml
1	1	2	0	0	6250	5654	188	180	1	1	1	0.8125
1	0	0	1	0	6783	0	130	360	1	1	1	0.8125
0	0	0	0	0	3159	0	100	360	1	1	1	0.8125
1	1	0	1	0	3000	1666	100	480	0	2	0	0.8125
1	1	2	0	0	3283	2035	148	360	1	2	1	0.8125
1	1	0	0	0	3727	1775	131	360	1	1	1	0.8125
1	1	2	0	0	3510	4416	243	360	1	0	1	0.8125
1	1	0	0	0	2083	3150	128	360	1	1	1	0.8125
0	1	0	0	0	2484	2302	137	360	1	1	1	0.8125
1	0	0	0	0	3229	2739	110	360	1	2	1	0.8125
1	1	1	0	0	12841	10968	349	360	1	1	1	0.8125
1	1	0	0	0	2499	2458	160	360	1	1	1	0.8125
0	0	0	0	0	2400	1863	104	360	0	2	0	0.8125

RELATION BASSE DE DONNÉE SQL

Les tables vont ensuite être connecté entre elle ces se qu'on appelle un eer .
Il y a 2 type de connection principale possible : Non Identifying Relationship et les Identifying Relationship .
Si une table et dépendente d'une autre table car ne posséde pas être identifier seul(il y a donc une relation parent enfant) il s'agit alors d'une identifying relationship et si l'inverse est vrai il s'agit d'une Non Identifying Relationship .



ANALYSE DES DONNÉES

Les objectifs de performance sont un bon moyen de suivre et de mesurer les progrès. Les rapports de performance peuvent inclure des détails tels que les indicateurs identifiés, les données recueillies et les activités réalisées liées aux ODD. Des objectifs de performance clairs et concrets facilitent la génération de données pertinentes, cohérentes et comparables au fil du temps, dans des formats que votre public peut comprendre et évaluer.

Nombre de ligne	Nombre de colonne	Nombre de valeurs nulle
• 614	• 13	• 149

Nombre de valeur dupliquer	Types de colonne
• 0	• Float64(4) , • Int64(1) , • Object(8)

Nombre de valeurs nulles

```
Loan_ID              0
Education            0
ApplicantIncome      0
CoapplicantIncome    0
Property_Area        0
Loan_Status          0
Married              3
Gender              13
Loan_Amount_Term    14
Dependents          15
LoanAmount          22
Self_Employed       32
Credit_History      50
dtype: int64
```

Le pourcentages des valeurs nulles

	Nan	%nan
Credit_History	50	8.14
Self_Employed	32	5.21
LoanAmount	22	3.58
Dependents	15	2.44
Loan_Amount_Term	14	2.28
Gender	13	2.12
Married	3	0.49
Loan_ID	0	0.00
Education	0	0.00
ApplicantIncome	0	0.00
CoapplicantIncome	0	0.00
Property_Area	0	0.00
Loan_Status	0	0.00

NETTOYAGE DES DONNÉES

Une fois l'analyse des données réalisé vue qu'elle sont plutôt propre je n'avais qu'une tâche unique a réalisé :
Suppression des valeurs nulles

Cette étape nous permet de passer les données de l'état brut à des données qui sont : reconnaissable, interprétable et efficace par les algorithmes d'intelligence artificielle .

Cette étape plusieurs encoding s'offre à moi, j'ai décidé d'utiliser le LabelEncoder .
Explication du processus de sélection :

Il y a 2 encodeurs basiques et efficaces pour ce modèle que l'on peut utiliser : LabelEncoder , OnehotEncoder ...

LabelEncoder	OnehotEncoder
• permet de transformer des données de textes en colonnes numériques • inconvénients : Imaginions une colonne avec plusieurs noms de villes, avec le LabelEncoder elles seront transformées en valeurs numériques mais comme il peut y avoir beaucoup de choix cela peut créé une hiérarchie des nombres dans ces colonnes 8 seraient supérieur à 1 . • Paris = 1 • Marseille = 2 • Lyon = 3 • Une hiérarchie pour se crée par exemple Paris mieux que Marseille et Marseille mieux que Lyon	• Le OneHotEncoder très similaires au LabelEncoder mais créera à la place des colonnes pour chaque une des différentes données • Par exemple : • Paris = 1 • Marseille = 2 • Lyon = 3 • cela créeras 3 colonnes

Pour mon dataset le meilleur utilisation sera le LabelEncoder car il y a très peu de colonnes ayant plus que 2 choix possibles il y a 2 colonnes dans cette situation utilisée le LabelEncoder nous éviteras donc de crée des colonnes inutilement.

VISUALISATION DES DONNÉES

Dans la photo ci-jointe vous pouvez voir le dataset nettoyer est prêt à l'utilisation pour mon modèle de machine learning.

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
1	1	1	1	0	0	4583	1508.0	128.0	360.0	1.0	0	0
2	1	1	0	0	1	3000	0.0	66.0	360.0	1.0	2	1
3	1	1	0	1	0	2583	2358.0	120.0	360.0	1.0	2	1
4	1	0	0	0	0	6000	0.0	141.0	360.0	1.0	2	1
5	1	1	2	0	1	5417	4196.0	267.0	360.0	1.0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...
609	0	0	0	0	0	2900	0.0	71.0	360.0	1.0	0	1
610	1	1	3	0	0	4106	0.0	40.0	180.0	1.0	0	1
611	1	1	1	0	0	8072	240.0	253.0	360.0	1.0	2	1
612	1	1	2	0	0	7583	0.0	187.0	360.0	1.0	2	1
613	0	0	0	0	1	4583	0.0	133.0	360.0	0.0	1	0

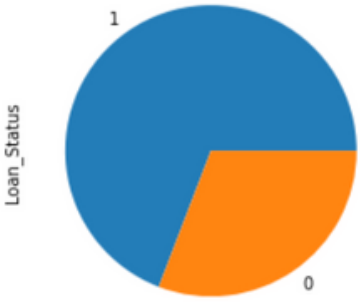
480 rows × 12 columns

La visualisation des données consiste à faires des graphiques en Python ou avec un outil de dashboard(Power Bi, Tableau)
Je vais utiliser Python .

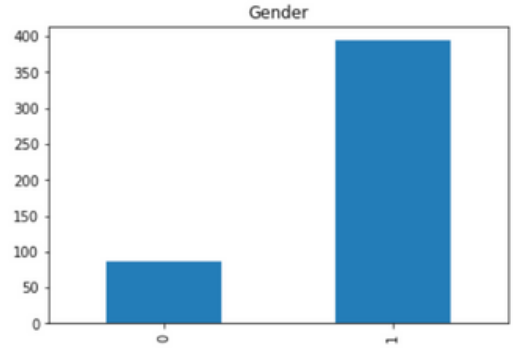
Je vais analyser 10 des colonnes
Les 3 colonnes que je ne vais pas analyser sont tout simplement des colonnes avec énormément de donnée différente se sont donc des graphiques flous à comprendre et sans grande nécessitée car toutes les valeurs généré ou presque seront unique.

Les 3 colonnes en questions : AppliantIncome , CoApplicantIncome , LoanAmountTerm

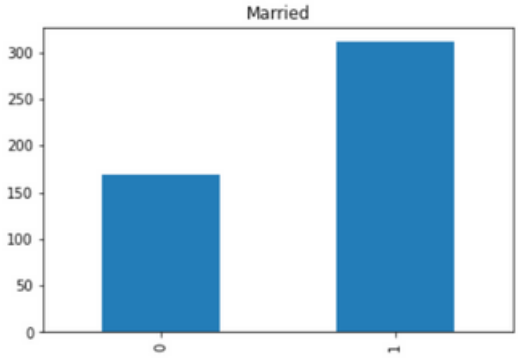
VISUALISATION DES DONNÉES



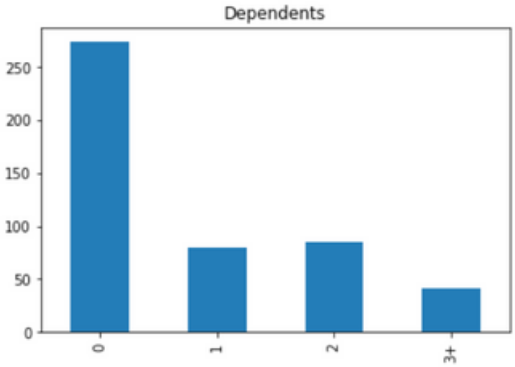
Ce premier graphique nous permet de vérifier si notre target a le même nombre de données entre le 0 et le 1 target = Colonne ciblées qui répondras à si oui ou non on pourras prétendre à un prêt bancaire .
Ce graphique sans surprise n'est pas égal car il y a jamais le même nombre de personnes qui ont été accepter que refuser un prêt bancaire1 = Accepter
0 = Refuser



Ce graphique nous montre le nombre de femmes comparé au nombre d'hommes qui font une demande de près .
0 = Femme
1 = Homme
Il y a environ 90 femmes qui vont faire une demande comparer aux hommes qui sont environ 380.

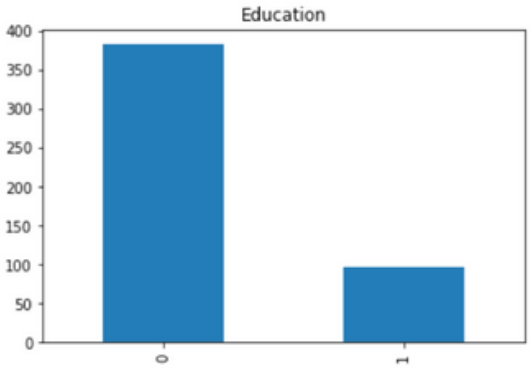


Nous pouvons comparer le nombre de personnes mariées ou non.0 = Non marié
1 =Marié
Il y a 170 personnes non mariées qui vont demander un prêt et plus de 300 personnes mariées vont faire la demande

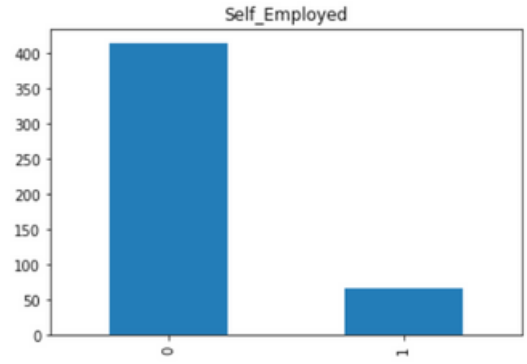


Sur ce graphique nous pouvons voir si les personnes sont indépendantes et dans quelles catégories de dépendent ces personnes sont on peut voir qu'il y a 270 personnes indépendantes à un niveau 0, 80 indépendants au niveau 1, 90 indépendants au niveau 2, 50 indépendants au niveau 3+ .

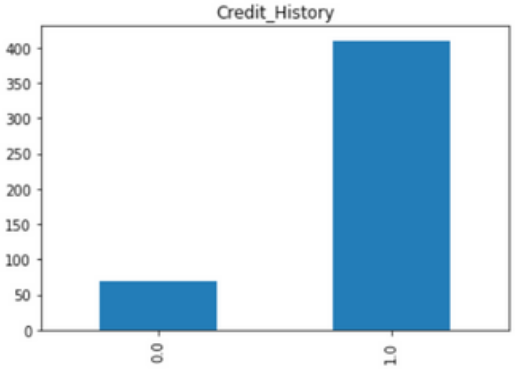
VISUALISATION DES DONNÉES



CLe graphique suivant nous montre l'éducation d'une personne demandeuse de près .
0 = diplomé
1 = Non diplomé
Il y a 380 de diplomés et 100 personnes de non diplomés

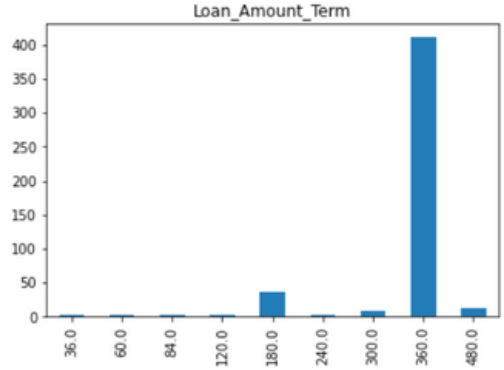


Se graphique nous montre si les personnes sont travailleur indépendant ou non indépendant
0 = Non indépendant
1 = Indépendant
Il y a plus de 400 personnes qui sont non indépendantes est uniquement 80 pour les personnes indépendantes



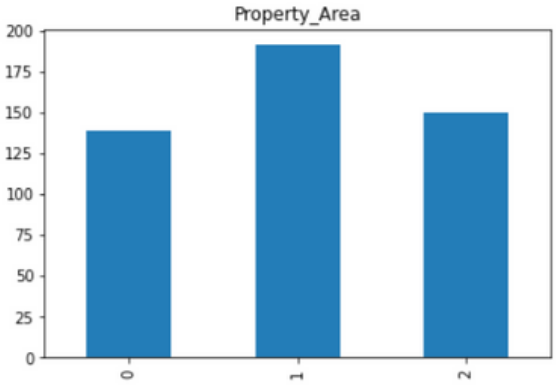
Ce graphique nous montre si des personnes ont précédemment eu un crédit ou non et le montant de celui-ci

0 = sans crédit passé
1=Avec crédit passé
On peut voir 80 personnes sans crédit et 400 avec crédit .



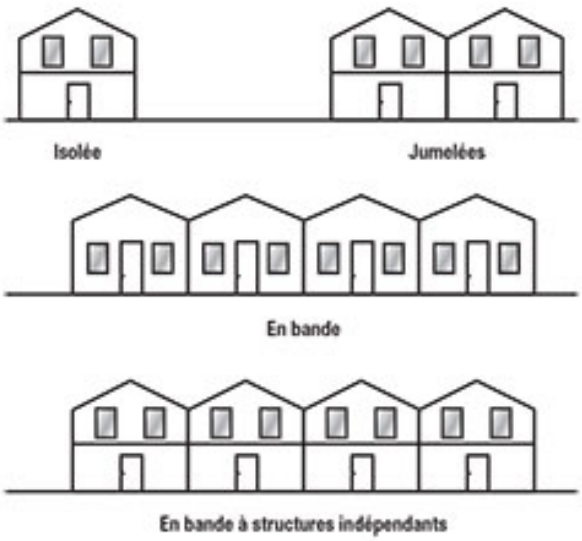
Ce graphique montre le temps de remboursement nécessaire

VISUALISATION DES DONNÉES



Le graphique suivant nous permet de voir dans quel type d'habitations sont les demandeurs de prêts il y a 3 types de catégories d'habitations : catégories 0, catégories 1 et catégories 2 .
On peut voir que la catégorie dominante ainsi que la catégorie 1 ont environ 200 habitants suivis de la seconde catégories les habitants sont au nombre de 150 pour finir la catégorie 0 ils sont entre 125 et 145

1ère famille		Individuelle	Habitations isolées en bande à structures non indépendantes en bande à structures indépendantes	Niveaux maximum ≤ R + 1 R + 0 R + 1	
2ème famille			Habitations isolées jumelées en bande à structures non indépendantes en bande à structures indépendantes	> R + 1 R + 1 > R + 1	
		Collective	Habitations collectives	≤ R + 3 (II)	Bluepack Treuil mécanique & pneumatique
3ème famille	A		3 conditions : R + 7 maxi D < 7 M (I) accès escalier atteint par voie échelle	≤ R + 3 (II) R + 7	Bluepack Treuil électrique
	B		Hauteur < 28 m, une seule des conditions ci-dessus non satisfaite Accès aux escaliers à moins de 50 mètres d'une voie ouverte à la circulation	H ≤ 28 m (II)	Bluepack Treuil pneumatique



PRÉPARATIONS DE LA CRÉATION DU MODÈLE :

Suite à mon nettoyage des données et à mes différentes analyses nous allons désormais pouvoir passer à la création du modèle pour ceci nous allons utiliser des algorithmes de machine learning ..

Machine Learning Définittions

La Machine Learning fait partie d'une des sous-disciplines de l'intelligence artificielle cela consiste à détecter des tendances au sein d'un historique de connaissances.

C'est à dire :
comme exemple une personne gagne 3000 euros mensuellement elle peut prétendre à un prêt de 21000 euros par contre si une personne gagne moins elle ne pourra pas y prétendre...
Concrètement :
dans le cas de la prédiction de prêts bancaires cela veut dire que l'on utilise des données historiques (que l'on possède déjà)qui viennent d'événement produit ultérieurement pour prédire si oui ou non le prêt bancaire qui seras demandé dans le futur sera réalisable .

Axe X et Y

- Le modèles de machine learning a besoin de l'axe X et Y .
-
- Définition des axes X et Y :
-
- Les axes X et Y sont utilisés pour savoir sûr quelles colonnes le modèle va se baser pour les prédictions dans notre cas présent nous allons nous baser sur la colonne « Loan Status » qui est notre target(voir définition target page 6) .
-
- X = base de donnée sans la target
- Y = La target
-

PRÉPARATION DES DONNÉES

dataset = base de donnée

Une fois les axes X et Y définis on peut réaliser la séparation des données dans un dataset test et dans un dataset train .

Définition divisé les données :

Les données sont divisées à l'aide de la fonction train_test_split cette fonction nous permet de définir X_train , X_test , y_train , y_test .

Grâce à cette méthode on divise le dataset en 2 parties le train(entraînements) et le test cela permet d'avoir environ 80 % des données que le modèle connaîtra(dataset train) est 20 % qui sont les données du dataset teste des valeurs totalement inconnues au modèle.

Le modèle de machine learning s'entrainera sur le dataset train et testeras sa fiabilité sur le test .

Lorsque l'on analyse un modèle le test de la fiabilité du modèle est souvent inférieur à l'analyse des données train car le modèle les découvre pour la première fois ..

La préparation des données :

Le préprocessing c'est le fait de préparer Les données d'une façon à ce que les algorithmes d'intelligence artificielle comprennent et interprète au mieux les données.
Lorsque l'on arrive à cette étape il y 3 possibilités qui ont toutes une utilisation différentes :

Scaling : le scaling permet de changer l'intervalle des valeurs . La distribution de ces données ne sera pas affectée .
Le scale ou l'intervalle des valeurs sera généralement entre 0 et 1
La standardisation est également un dérivée du scaling avec ses propres spécificités

L'algorithme de preprocessing utilisé :

Pour notre base de données j'ai décidé d'utiliser l'algorithme StandardScaler il fonctionne comme la définition du Scaling donnée précédemment.

LES DIFFÉRENTS MODELS

Le type de modèle de Machine Learning don je vais avoir besoin sera un modèle de classification . Un modèle de classification c'est un modèle qui va catégoriser ce que l'on cherche à obtenir par exemple : Le modèle créé ici classifiera si oui ou non ce client peut avoir un prêt bancaire le client sera alors placé (classifié) dans la catégorie Oui ou Non qui suite à la transformation des données ont changé 1(Oui), 0(non) (voir pages 6 et 7) .

Les modèles les plus couramment utilisée :

Logistic Regression	Knn	Decision Tree
La logisticRegression va répondre a une sortie binaire comme oui ou non elle va analyser toutes les features (différentes colonnes) pour trouver les corrélations entre les données	K nearest neighbors est une methode de machine learning de classification qui permet de déterminer un point de data qui est le plus proches d'un groupe de donnée pour l'ajouter dans celui-ci . Cette méthode par du principe que des données similaire sont proche	Le modèles suivant fonctionner comme un arbre on peut donc imaginer un arbre a l'envert et on partiras des features qui seront toutes présente une fois le bas haut de l'arbre atteint cela nouspermet de trouvée les relations possible entre les différentes features

J'ai utilisé ces 3 modèles le plus performants de ces algorithmes pour mon projet étant la LogisticRegression .

suite à l'entraînement des différents models nous allons désormais pouvoir examiner les performances de la LogisticRegression

PERFORMANCE DU MODEL

Terminez votre rapport en faisant un retour sur les faits importants, et en renouvelant votre engagement à continuer à travailler sur les ODD accessibles avant 2030.

```
accuracy train : 0.810
accuracy test : 0.799
              precision    recall  f1-score   support

           0       1.00      0.34      0.51        44
           1       0.78      1.00      0.87       100

    accuracy                           0.80       144
   macro avg       0.89      0.67      0.69       144
weighted avg       0.84      0.80      0.76       144
```

Les scores :

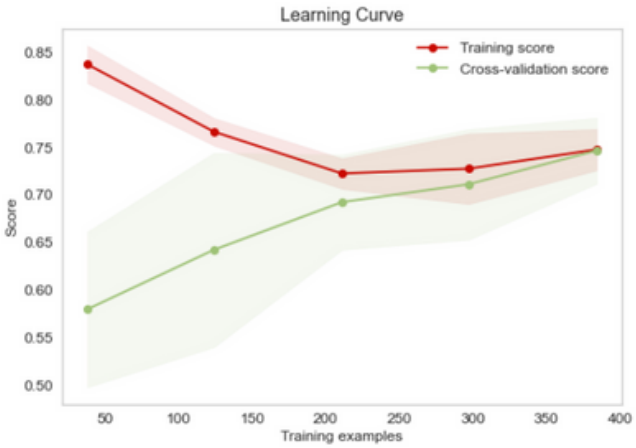
Sur le tableau plein de différentes mesures sont notées mais uniquement 4 de ces valeurs nous intéressent dans notre cas : accuracy train, accuracy test, recall

Le recall permet de savoir à quel point le modèle est performant la valeur 0 doit être le plus proche de 0 que possible est pour le 1 il doit arriver le plus proche du 1 mai en essayant de ne pas afficher 1 si possible car cela pourra amener différentes problématiques.
Mon 1 est probablement du à de l'overifitting ou underfitting .

On reviendra sur le sujet de l'overfitting ou underfitting(voir page 23)pour déterminer ou non si ces vrais cela se passera après l'explication des différents graphiques on pourra déterminer uniquement à ce moment précis.

Pourquoi 2 accuracy ?
L'accuracy train et la précision avec là quel le modèle c'est entrainé
L'accuracy teste et la précision du modèle une fois entraîné on donne des données inconnues au modèle qui ne sont donc pas dans la base de données d'entrainements pour voir les compétences du modèle.

PERFORMANCE DU MODEL



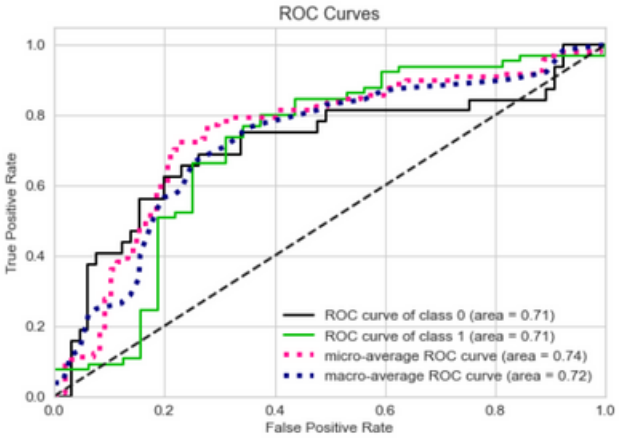
La Learning Curve :

La learning curve permet de voir comment le modèle c'est entraîner et comment les données on put être utilisé par la LogisticRegression .

Pour qu'une learning curve sois parfaite il faut que les 2 courbes(curves) train et test se rejoignent au centre mais qu'elle ne se touche pas ou peu

l'espace qu'il y a entre les 2 lignes montre le taux d'erreur possible

On peut voir que ma learning curve n'est pas parfaite mais également que l'entrainement et le test se sont plutôt bien passé malgré que les 2 lignes se touchent au quatrième point le modèle reste plutôt performant et nous informent d'un taux d'erreur très bas.



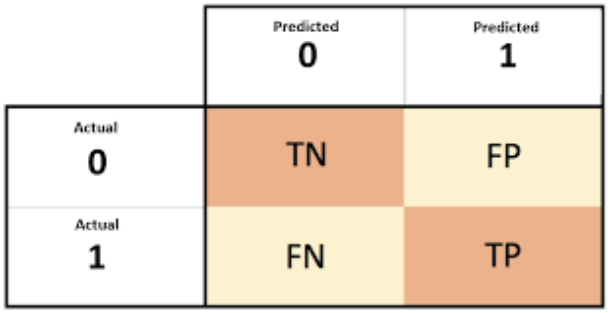
La ROC AUC :

La ROC AUC permet de montrer à quel point l'entraînement c'est bien réaliser plus le ROC AUC monte rapidement mieux le modèle ces entraîné .

Les résultats de notre modèle nous montrant de très bons scores on pourrait donc s'attendre à une meilleur ROC AUC .

Probablement dû à un manque de donnée elle n'a pas pus s'entraîner asser.

PERFORMANCE DU MODEL



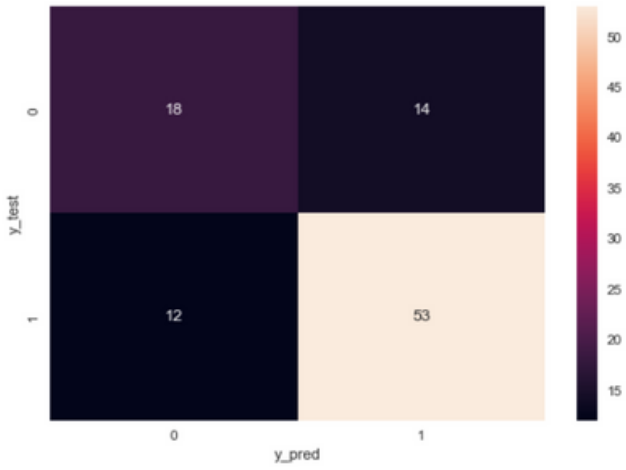
La Confusion Matrix :

Ce graphique permet de nous dire combien il y a de True Positive, True Negative, False positive et False négative .

True Positive : Ce sont les prédictions qui se sont avérées vraies
True Négative : Ce sont les prédictions qui sont fausses et réel .

Positive: ce sont les prédictions qui se sont avéré fausses
False Négative : Ce sont les prédictions qui se sont avéré être non alors qu'elle était oui .

Maintenant que le graphique est compris on peut analyser le graphique de mon modèle .



Mon graphique en haut à gauche les True négatif sont de 15 ce qui peut potentiellement dire que ce sont des personnes qui ne pouvaient pas prétendre à un pré mais on leur a tout de même accordé pour la banque cela pourrais avoir des conséquences t'elle que non-payement en temps et en heures etc.

En haut a droit ce sont les False Positive donc les personnes qui auraient pu prétendre à un prêt mais mon modèle ne l'est à pas compter comme t'elle cela n'aurait pas de réelles conséquences sur l'entreprise, probablement un client mécontent....
En bas à gauche il y a les False negative ce sont les personnes qui n'auraient pas pu prétendre à un prêt mais en ont tout de même eux un .

En bas à droite il y a les True Positive c'est-à-dire les personnes pouvant prétendre à un prêt et qui ont obtenu leurs prêts...

Désormais nous avons une aperçue des compétences comme on à pus le comprendre il n'est pas le plus performant cela est probablement dû en partie a de l'overfitting ou underfitting voyons sa dans la prochaine page .

OVERFITTING / UNDERFITTING

Suite à l'analyse de tous les graphiques pour voir la qualité du modèle on va pouvoir parler de l'overfitting et underfitting .

Overfitting

Lorsque l'on examine la Learning Curve on voit que la ligne de train et rester très haut ce qui veut dire qu'elle n'a pas spécialement appris comme nécessaire cela entraînera des scores beaucoup moins bon ou beaucoup trop bon comme le 1 que l'on a obtenue dans le recall
Comment détecter l'overfitting :

Il y a plusieurs façons de détecter l'overfitting par exemple lors des résultats des modèles si l'on constate un 1 parfait on pourrait déjà avoir des doutes et en examinant les graphiques en général nos doutes seront confirmés .

Underfitting

L'underfitting est l'exact opposé de l'overfitting le modèle n'a pas du tout réussi à s'entrainer et sera alors inutilisable
cela nous affichera donc pour La Learning Curve une barre pour train très basse .
Le résultat sera toujours affiché sur le 1 .

pourquoi ça arrive ?

L'underfitting et l'overfitting arrivent pour des raisons similaires : trop peu de donnés , pas de cohérence entre les données, données mal préparées pour un modèle..

DÉPLOIEMENT

Le déploiement c'est l'action de construire une application que l'on mettra ensuite sur un service en ligne .

Je vais donc construire une application qui permettra d'utiliser mon modèle de machine learning sur la prédiction de prêt bancaire ...

Heroku est le service en ligne que l'on va utilisée nous allons également utilisées la librairie python nommée Streamlit .

Streamlit me permettra d'avoir un rendue UI et UX le mieux possible l'utilisation sera très simple et le design sobres sans complexité pour aider à la compréhension l'application seras donc facile d'utilisation et iras directement au but .

Suite à la création de l'application la première possibilité pour voir si elle fonctionne et de l'exécuter en local .

L'exécution locale c'est le fait de faire tourner l'application sur notre machine sans passer par des services tiers .

Une fois que l'application en local est fonctionnelle on peut désormais commencer le déploiement sur Heroku (mettre l'application en ligne)..

Le déploiement en ligne va demander plusieurs fichiers que l'on devra créer : Procfile, requirements.txt, runtime.txt , setupsh , app py

Procfile	requirements.txt	Runtime.txt
• Permet à heroku de reconnaître de quels types de langage de programmation il s'agit.	• Ce document texte nous permet d'indiquer toutes les versions des librairies python que l'on veut utiliser . Il est obligatoire de faire ce document car heroku ne possèdent pas les librairies utilisées .	• Permet de sélectionner la bonne version de python pour ne pas avoir de conflit •

setup.sh	app.py
• Ce sont des variables utilisées par le système d'exploitation pour communiquer entre les différentes applications / service utilisées	• Le code de l'application sera mise dans ce fichier .py étant le diminutif de python

DÉPLOIEMENT

Le déploiement c'est l'action de construire une application que l'on mettra ensuite sur un service en ligne .

Je vais donc construire une application qui permettra d'utiliser mon modèle de machine learning sur la prédiction de prêt bancaire ...

Heroku est le service en ligne que l'on va utilisée nous allons également utilisées la librairie python nommée Streamlit .

Streamlit me permettra d'avoir un rendue UI et UX le mieux possible l'utilisation sera très simple et le design sobres sans complexité pour aider à la compréhension l'application seras donc facile d'utilisation et iras directement au but .

Suite à la création de l'application la première possibilité pour voir si elle fonctionne et de l'exécuter en local .

L'exécution locale c'est le fait de faire tourner l'application sur notre machine sans passer par des services tiers .

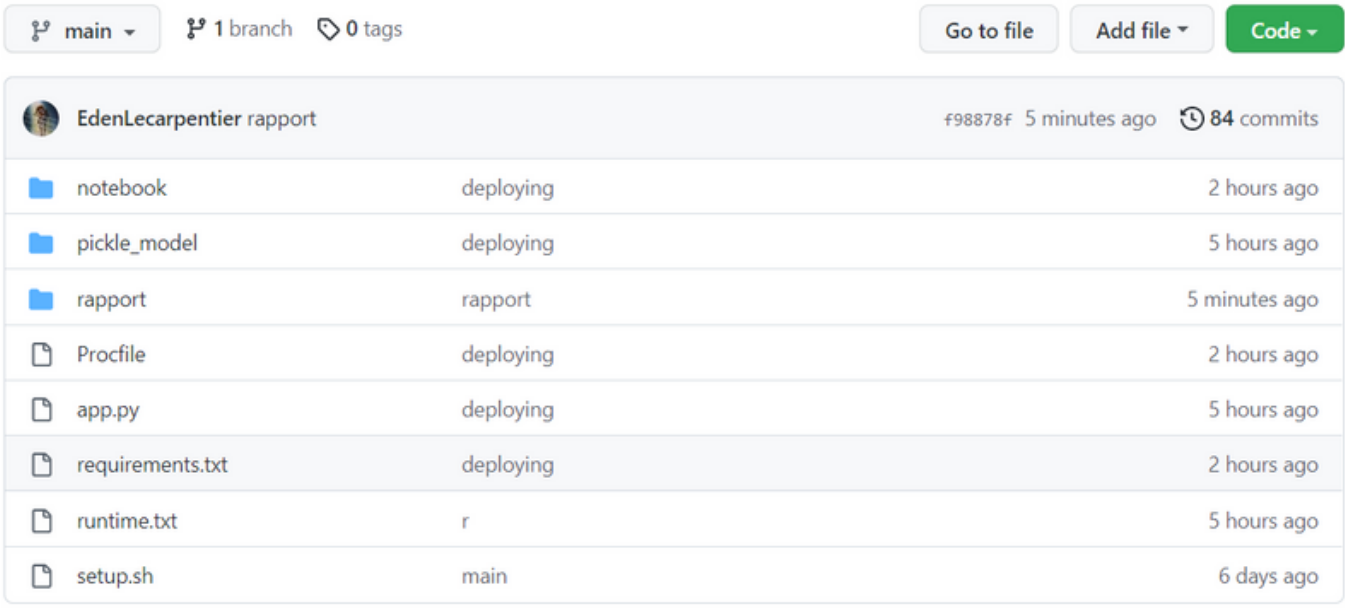
Une fois que l'application en local est fonctionnelle on peut désormais commencer le déploiement sur Heroku (mettre l'application en ligne)..

Le déploiement en ligne va demander plusieurs fichiers que l'on devra créer : Procfile, requirements.txt, runtime.txt , setupsh , app py

Procfile	requirements.txt	Runtime.txt
• Permet à heroku de reconnaître de quels types de langage de programmation il s'agit.	• Ce document texte nous permet d'indiquer toutes les versions des librairies python que l'on veut utiliser . Il est obligatoire de faire ce document car heroku ne possèdent pas les librairies utilisées .	• Permet de sélectionner la bonne version de python pour ne pas avoir de conflit •

setup.sh	app.py
• Ce sont des variables utilisées par le système d'exploitation pour communiquer entre les différentes applications / service utilisées	• Le code de l'application sera mise dans ce fichier .py étant le diminutif de python

DÉPLOIEMENT



Une fois tous ces fichiers créent on peut alors envoyer tous ces dossiers sur notre repository github.

Le github que j'ai créé dès le début du projet me sert de back up en cas de perte d'un fichier et également pour déployer l'application .

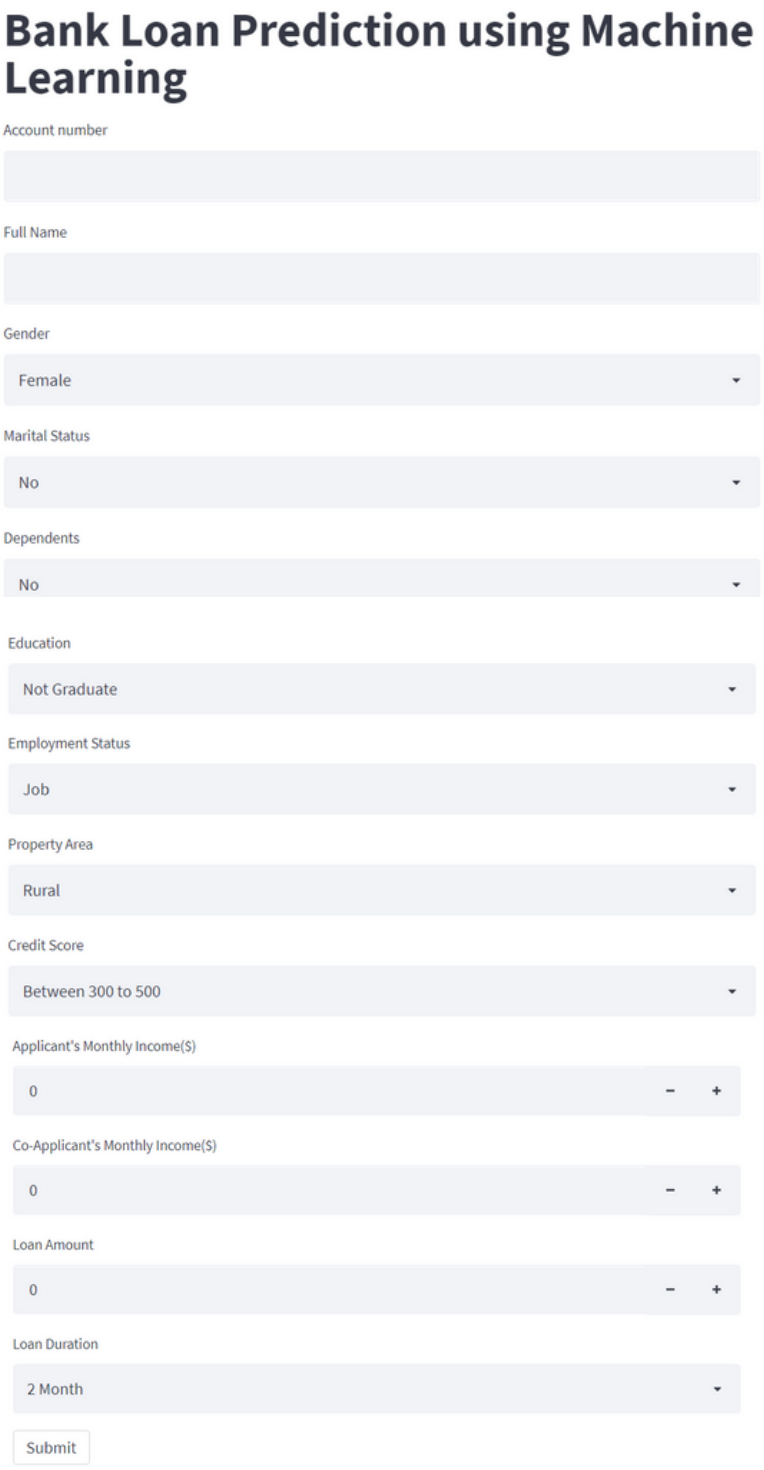
Une fois le Github créé on va pouvoir alors push(mettre) nos fichiers crée préalablement sur Github

Je suis ensuite allée sur Heroku ou j'ai créé une nouvelle application nommée loanpredictionfilerouge j'ai ensuite connecté cette application au Github .

Les 2 services sont maintenant connectés on peut déployer .
Si le déploiement ne ce passe pas comme prévu il sera stoppé net et les logs(ligne montrant l'évolution du déploiement) me montrerons l'erreur dans ce cas : correction, retour à l'étape de push sur Github .

Une fois le déploiement fini on peut ouvrir notre application web et l'utilisée, on peut également avoir des erreurs à l'ouverture de l'application dans ce cas refaire les étapes précédentes..

DÉPLOYEMENT

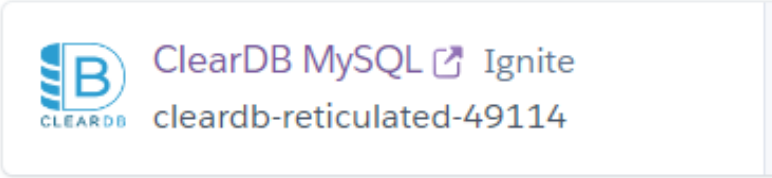


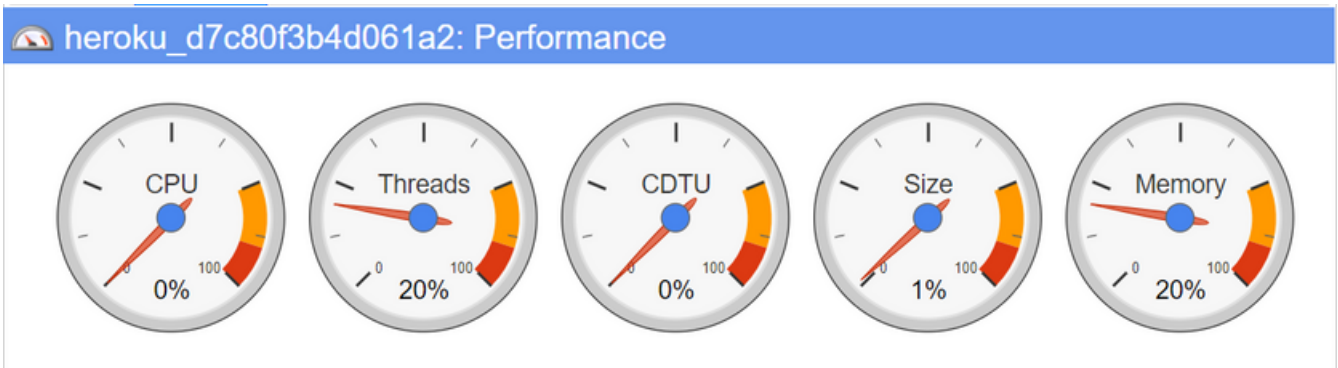
L'application s'utilise de façon très simple on indique :
Numéro de compte , nom, prénom .

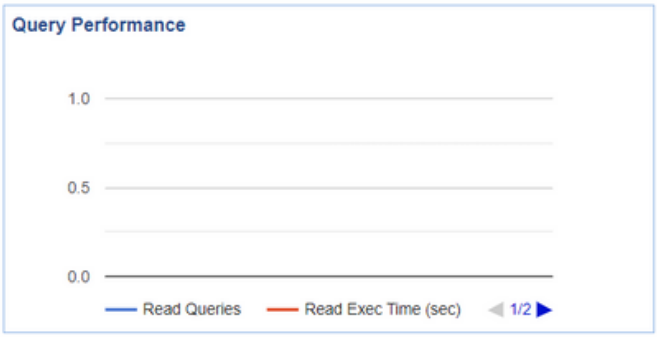
Ensuite on sélectionne : le genre, marié ou non, dépendent ou non, dernier diplôme obtenu, type de travail , endroit d'habitation, le score de notre fidélité a la banque, le salaire, salaire du cooapplicant, la somme du prêt, la durée du prêt

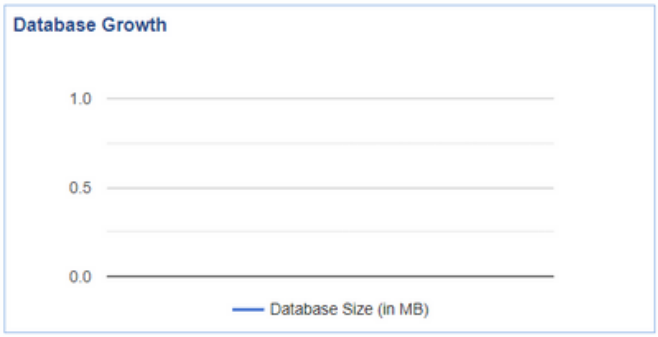
CONNECTION DE LA BASE DE DONNÉE

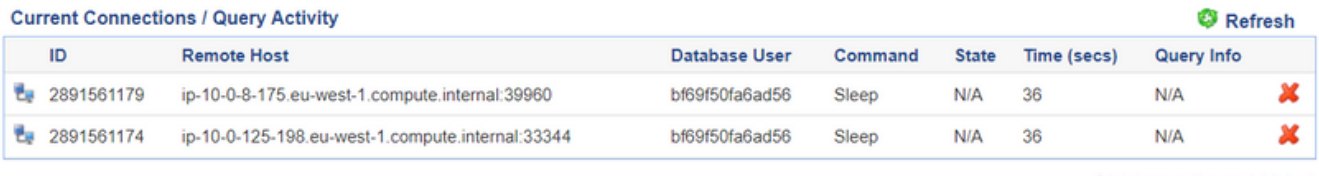
Une fois le déploiement réalisé la base de donnée a était connectée a l'application cela permet lorsqu'il y auras de nouvelle données de les ajoutées , également permet au modèle d'utilisée de nouvelle donnée se qui le renforceras .
ClearDB MySQL est une extension que l'on peut ajouté a notre application qui permet de connectée une base de donnée dans mon cas MySQLWorkbench,
la connection se réalise en utilisant Config Var dans les parametre de l'application Heroku . Une fois le Config Var crée on obtient : hostname , username ,password et schema
tout c'est information seront entrée dans une nouvelle connexion MySQLWorkbench ainsi lorsque l'on accéde a heroku et que l'on sélectionne l'application on peut voir la base de donnée et des graphique représentant cette base de donnée











MLFLOW TRACKING

MLflow est un outil que l'on utilise en data science pour tracké les modèles d'intelligence artificielle que l'on fait .
MflLow nous permet de voir si notre reste constamment dans de bonne performance par exemple on feras 2 expérience une obtiendrass 80% tandis que l'autre obtiendras 60% dans ce cas le modèles n'est pas stable et doit être réentrainé dans le cas contraire le modèles évolues se qui est positif .

J'ai donc installée la librairie python MLflow et modifié le fichier python pour pouvoir utilisée cette librairie .

Une fois ces modifications réalisée on peut ouvrir MLflow en localhost cela nous permettras de voir l'évolutions des résultats de notre modèles voir si il : continue d'évoluer , il stagne , il régresse .

Le modèle que j'ai crée stagne il affiche tout le temps une accuracy de 0.81 qui est la métrique que j'utilise , en connaissant les compétences de notre modèle il ne serais .

Il faudrait avoir plus de donnée varié pour être sur que le modèle et aussi performant que ceux que l'on aperçoit