

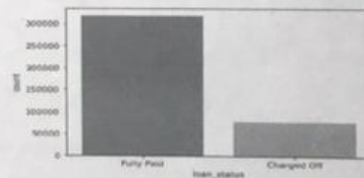
Examen d'Intelligence Artificielle

Deuxieme Session

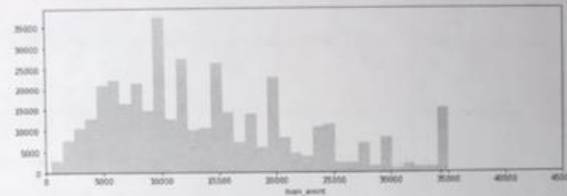
Soit le dataset suivant

LoanID	Description
0 loan_amnt	Le montant indiqué du prêt demandé par l'emprunteur. Si, à un moment donné, le service de crédit réduit le montant du prêt, cette valeur en tient compte
1 term	Le nombre de paiements sur le prêt. Les valeurs sont exprimées en mois et peuvent être de 36 ou 60
2 int_rate	Taux d'intérêt du prêt
3 installment	Le paiement mensuel dû par l'emprunteur si le prêt est d'origine
4 grade	Qualité du prêt attribuée par LC
5 sub_grade	Sous-niveau de qualité attribué par LC
6 emp_title	Le titre de poste fourni par l'emprunteur lors de la demande de prêt
7 emp_length	Durée de l'emploi en années. Les valeurs possibles sont comprises entre 0 et 10, où 0 signifie moins d'un an et 10 signifie dix ans ou plus
8 home_ownership	Le statut de propriétaire du logement fourni par l'emprunteur lors de l'enregistrement ou obtenu à partir du rapport de crédit. Nos valeurs sont : RENT (locataire), OWN (propriétaire), MORTGAGE (hypothèque), OTHER (autre)
9 annual_inc	Le revenu annuel déclaré par l'emprunteur lors de l'enregistrement
10 verification_status	Indique si le revenu a été vérifié par la LC, non vérifié, ou si la source de revenu a été vérifiée
11 issue_d	Le mois au cours duquel le prêt a été financé
12 loan_status	État actuel du prêt
13 purpose	Une catégorie fournie par l'emprunteur pour la demande de prêt
14 title	Le titre de prêt fourni par l'emprunteur
15 zip_code	Les 3 premiers chiffres du code postal fourni par l'emprunteur dans la demande de prêt
16 addr_state	L'état fourni par l'emprunteur dans la demande de prêt
17 dti	Un ratio calculé en utilisant le total des paiements mensuels de l'emprunteur sur le total des dettes, à l'exclusion de l'hypothèque et du prêt LC demandé, divisé par le revenu mensuel autodéclaré de l'emprunteur
18 delinq_12m	Le mois d'ouverture de la première ligne de crédit déclarée par l'emprunteur
19 open_acc	Le nombre de lignes de crédit ouvertes dans le dossier de crédit de l'emprunteur
20 pub_rec	Nombre d'enregistrements publics dérangeants
21 revol_bal	Solde total du crédit renouvelable
22 revol_util	Le taux d'utilisation de la ligne renouvelable, ou le montant du crédit utilisé par l'emprunteur par rapport à l'ensemble du crédit renouvelable disponible
23 total_acc	Le nombre total de lignes de crédit figurant actuellement dans le dossier de crédit de l'emprunteur
24 initial_list_status	Statut initial de l'inscription du prêt. Les valeurs possibles sont : RL, F
25 application_type	Indique si le prêt est une demande individuelle ou une demande conjointe avec deux co-emprunteurs
26 mort_acc	Nombre de comptes hypothécaires
27 pub_rec_banruptcies	Nombre de faillites enregistrées dans les registres publics

Tâche 1 : Puisque nous allons tenter de prédire l'état des prêts, créez un graphique de comptage (countplot) comme celui ci-dessous



Tâche2 : Créer un histogramme de la colonne `loan_amnt` comme celui ci-dessous avec 12,7 comme taille de figure



Tâche3 : Calculer la corrélation entre toutes les variables numériques continues en utilisant la méthode `.corr()`.

Tâche 4 : Visualisez ceci à l'aide d'une carte de chaleur (heatmap). En fonction de votre version de matplotlib, vous devrez peut-être ajuster manuellement la heatmap avec une taille de figure de 12,7

Tâche 5 : Créer un diagramme en boîtes (boxplot) montrant la relation entre le statut du prêt (`loan_status`) et le montant du prêt.

Tâche 5 : Calculer les statistiques sommaires pour le montant du prêt, regroupées par la feature `loan_status`.

Tâche 6 : Créer un graphique de comptage par notation (grade). Définissez le label `loan_status` pour le paramètre `hue`.

Tâche 7 : Créer une nouvelle colonne appelée `load_repaid` qui contiendra 1 si le statut du prêt était "Fully Paid" et un 0 s'il était "Charged Off".

Partie 2 : Pré-traitement des Données (/9)

Tâche 8 : Quelle est la taille du DataFrame ?

Tâche 9 : Créer une Série qui affiche le nombre total de valeurs manquantes par colonne

Tâche 10 : Listez toutes les colonnes qui sont actuellement non numériques.

Tâche 11 : Fixer les variables X et y aux valeurs des features et label.

Tâche 12 : Effectuer une répartition train/test avec `test_size=0,2` et un `random_state` de 101

Tâche 13 : Supprimer la colonne `loan_status` que nous avons créée précédemment, car c'est un doublon de la colonne `loan_repaid`. Nous utiliserons la colonne `loan_repaid`, puisqu'elle contient déjà des 0 et des 1.

Tâche 14 : Utiliser une mise à l'échelle MinMaxScaler pour normaliser les données de features X_train et X_test. Rappelez-vous que nous ne voulons pas de fuite de données du dataset, donc nous adaptons que sur les données X_train.

Après avoir exécuter la cellule ci-dessous pour importer les fonctions Keras nécessaires, continuez les tâches suivantes :

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
```

Tâche 15 : Construire un modèle séquentiel qui sera entraîné sur les données. Vous avez ici des options illimitées, mais voici ce que la solution utilise : un modèle schématisé ainsi : 78 --> 39 --> 19--> 1 neurone de sortie.

Tâche 16 : Adapter le modèle aux données d'entraînement pour au moins 25 Epochs. Ajoutez également les données de validation pour le traçage ultérieur et ajouter un batch_size de 256. et n'oubliez de sauvegarder votre modèle.

Tâche 17 : Déterminer la perte sur la validation par rapport à la perte sur l'entraînement.

Tâche 18 : Créer des prédictions à partir de l'ensemble X_test et afficher un rapport de classification et une matrice de confusion pour l'ensemble X_test.

Compte tenu du client ci-dessous, proposeriez-vous à cette personne un prêt ?

```
import random
random.seed(101)
random_ind = random.randint(0, len(df))
nouveau_client = df.drop('loan_repaid', axis=1).iloc[random_ind]
```

Tâche 19 : Vérifiez si cette personne a vraiment fini par rembourser son prêt ?

Bonne Chance !

Prof. David KUTANGILA

Il faut beaucoup prier !