UNIVERSITÉ DE VERSAILLES –
SAINT-QUENTIN-EN-YVELINES

ÉTUDE DE L'ALGORITHME GRADIENT STOCHASTIQUE

RÉALISÉ PAR:
- EYA KALBOUSSI
- ABDELHADI BOUALI

ENCADRÉ PAR:
- LUDOVIC JAVET
- LUC BOUGANIM

2023-2024

PLAN

- Introduction
- Le fonctionnement de l’algorithme
- Présentation des bases de données
- L’exploration et visualisation des données
- Entrainement du modèle
- Ajustement des hyperparamètres
- Les performances de SGD

Introduction

GRADIENT STOCHASTIQUE

VS

hyperparametre

Évaluation

L'IMMOBILIER EN CALIFORNIE

DÉTECTION DE FRAUDE PAR CARTE DE CRÉDIT

SGD

L’algorithme de descente de gradient stochastique

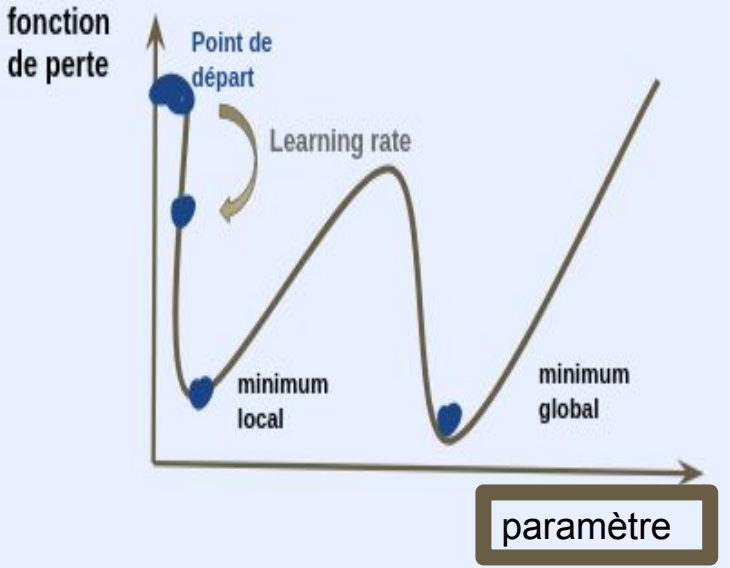
- L'objectif principal de SGD lors de l'entraînement d'un modèle est de minimiser une fonction de coût en ajustant itérativement les paramètres du modèle.
- Il utilise un échantillon aléatoire des données d'entraînement (mini-batch), Cela permet d'optimiser les paramètres du modèle afin de minimiser la fonction de coût associée à la tâche d'apprentissage.

SGD

- SGD est une méthode stochastique, ce qui signifie qu'il est basé sur le hasard.

- Par conséquent, permet de se rapprocher progressivement du minimum global de cette fonction de coût. Cela signifie que le modèle cherche à trouver la meilleure combinaison de paramètres qui minimise l'erreur de prédiction sur les données d'entraînement.

- Il peut ne pas converger vers le minimum global mais plutôt vers un minimum local. Un minimum local est un point où la fonction de coût est relativement bas, mais pas nécessairement le plus bas de tous les points possibles.



	Description	La valeur par défaut pour la première dataset \ la seconde dataset
Loss	**est une mesure utilisée pour évaluer à quel point les prédictions d'un modèle sont proches des valeurs réelles attendues.**	squared_error\hinge
Alpha	**Alpha est un paramètre de régularisation qui contrôle la pénalité des coefficients pour prévenir le surapprentissage.**	0.0001 \ 0.0001
Penalty	**Le paramètre contrôle la régularisation du modèle en utilisant des options telles que L2, L1 ou une combinaison des deux pour réduire le surapprentissage .**	L2 \ L2
Max iteration	**Il s'agit du nombre maximum d'itérations de la descente de gradient stochastique**	1000\ 1000
learning rate	**spécifie la méthode de mise à jour des poids lors de la descente de gradient**	invscaling \ optimal
(eta0)	**la vitesse d'apprentissage initial**	0.01 \0.0
tol **la tolérance de convergence**	**c'estle limite l'amélioration de la fonction de coût**	0.001 \ 1e-3

PRIX DE L'IMMOBILIER EN CALIFORNIE

Régression
prévision des prix des logements (valeur continue)

Notre jeu de données contient des informations sur les logements dans différentes régions de la Californie. Il est composé de **20 640 observations sur 8 variables.** Les données d'entrée qui utilisées par le modèle pour faire des prédictions comme suit :

***MedInc** : revenu médian du bloc. Cette caractéristique représente le revenu médian du bloc dans une zone géographique donnée.
***HouseAge** : âge médian des maisons dans le bloc.
***AveRooms** : nombre moyen de pièces par logement.
***AveBedrms** : nombre moyen de chambres par logement.
***Population** : population du bloc.
***AveOccup** : occupation moyenne du ménage.
***Latitude** : latitude du bloc.
***Longitude :** longitude du bloc.

Notre output est **Med house val** la variable que nous cherchons à prédire, à savoir la valeur médiane des maisons dans une région donnée. La variable cible est la valeur médiane de la maison en unités de 100 000 dollars américains.

L'EXPLORATION DES DONNÉES

Vérification de valeurs manquantes

```
MedInc          0
HouseAge        0
AveRooms        0
AveBedrms       0
Population      0
AveOccup        0
Latitude        0
Longitude       0
MedHouseVal     0
dtype: int64
```

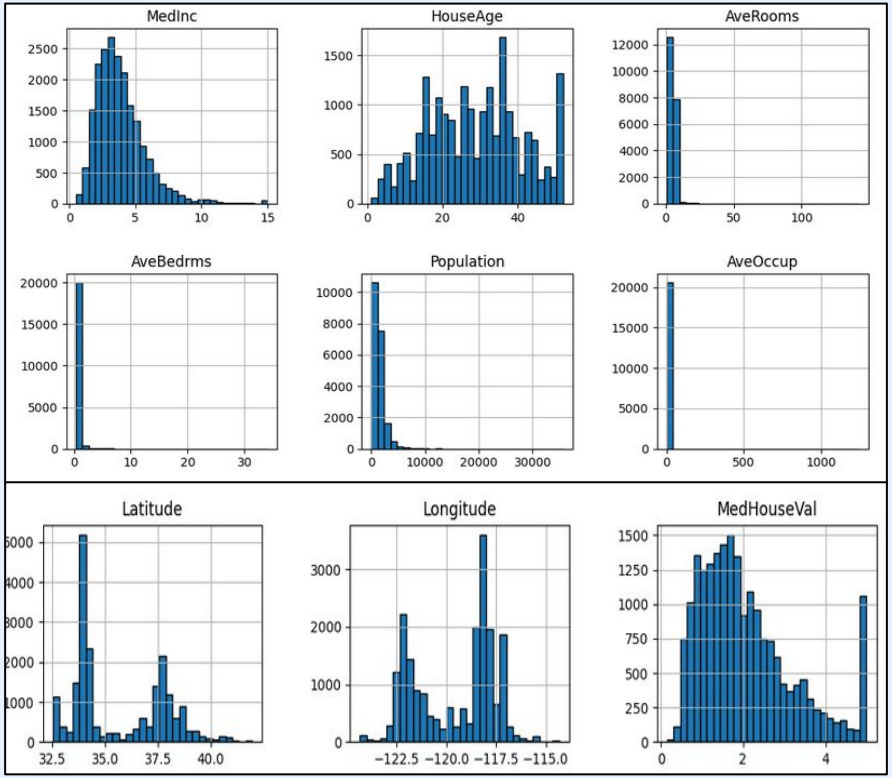
Affichage des information

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   MedInc       20640 non-null  float64
 1   HouseAge     20640 non-null  float64
 2   AveRooms     20640 non-null  float64
 3   AveBedrms    20640 non-null  float64
 4   Population   20640 non-null  float64
 5   AveOccup     20640 non-null  float64
 6   Latitude     20640 non-null  float64
 7   Longitude    20640 non-null  float64
 8   MedHouseVal  20640 non-null  float64
```

Statistiques descriptives

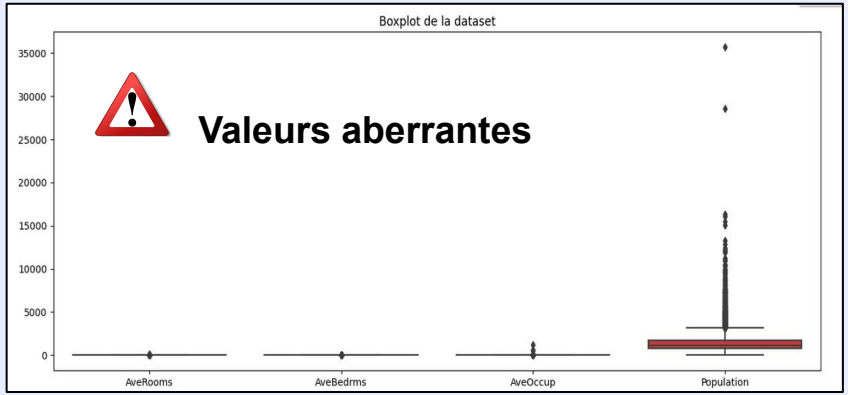
	AveRooms	AveBedrms	AveOccup	Population
count	20640.000000	20640.000000	20640.000000	20640.000000
mean	5.429000	1.096675	3.070655	1425.476744
std	2.474173	0.473911	10.386050	1132.462122
min	0.846154	0.333333	0.692308	3.000000
25%	4.440716	1.006079	2.429741	787.000000
50%	5.229129	1.048780	2.818116	1166.000000
75%	6.052381	1.099526	3.282261	1725.000000
max	141.909091	34.066667	1243.333333	35682.000000

La distribution des caractéristiques en traçant leurs histogrammes



Valeurs aberrantes

NETTOYAGE ET NORMALISATION DES DONNÉES



Valeurs aberrantes

Suppression des Valeurs aberrantes

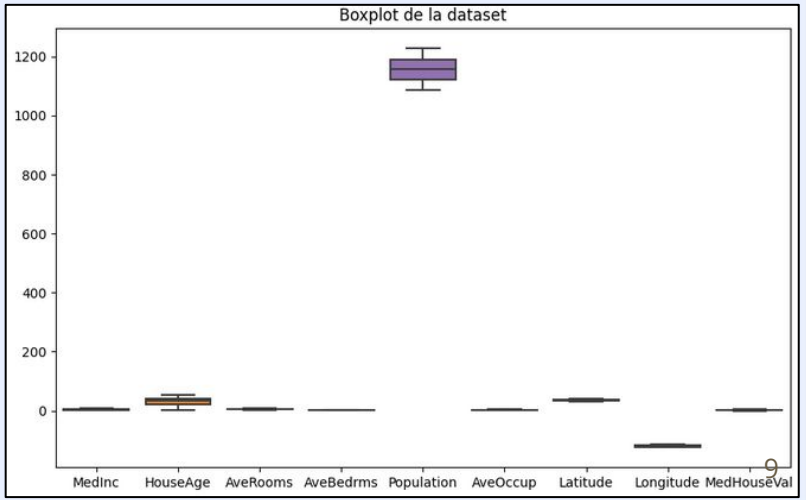
Elle consiste à définir les limites supérieure et inférieure en utilisant les quartiles Q1 et Q3, puis à identifier les observations situées en dehors de ces limites.

Normalisation
Elle permet de mettre les données à la même échelle pour faciliter l'entraînement des modèles

Nombre de valeurs aberrantes identifiées : 0

	MedInc	HouseAge	AveRooms	AveBedrms	Population
count	1499.000000	1499.000000	1499.000000	1499.000000	1499.000000
mean	3.572823	31.546364	5.133290	1.046716	1156.398933
std	1.325878	12.081265	0.921912	0.063841	40.307475
min	0.768300	2.000000	2.684022	0.882812	1086.000000
25%	2.559750	22.000000	4.505546	1.004151	1123.000000
50%	3.500000	33.000000	5.082126	1.045767	1156.000000
75%	4.438300	40.000000	5.765422	1.087517	1190.000000
max	7.211800	52.000000	7.641161	1.211699	1227.000000

	AveOccup	Latitude	Longitude	MedHouseVal
count	1499.000000	1499.000000	1499.000000	1499.000000
mean	2.851514	35.693763	-119.639153	1.890371
std	0.559937	2.157937	2.026066	0.924825
min	1.461171	32.540000	-124.270000	0.427000
25%	2.457249	33.940000	-121.855000	1.132500
50%	2.806202	34.280000	-118.490000	1.740000
75%	3.210676	37.740000	-118.060000	2.474500
max	4.324528	41.950000	-114.600000	4.477000



L'ENTRAÎNEMENT DU MODÉLE

1. **Diviser les données en ensemble d'entraînement et ensemble de test**

Nous fixons 30 % pour la partie test et 70 % pour la partie entraînement. En fixant le paramètre "random state" à une valeur fixe, par random state=42

2. **Le choix des métriques d'évaluation**

- Le MSE est une mesure de l'erreur quadratique moyenne des prédictions du modèle, et une valeur plus faible indique une meilleure performance du modèle .
Dans notre cas, le MSE de **0.316** est faible.
- Le R^2 mesure la proportion de variance de la variable de réponse qui est expliquée par le modèle, et une valeur plus élevée indique une meilleure performance du modèle
Dans ce cas, le R^2 de **0.58**.

=====> Le modèle explique environ 58 % de la variance des données de test, ce qui peut être considéré comme acceptable.

AJUSTEMENET DES HYPERPARAMÈTRES

Hyperparamètre a tester	etas	max_iters	Penalty	**alpha**	**learning rate**	**Tolérance**
Intervalle de valeur	[0.001, 0.01, 0.1, 1]	[100, 1000, 10000]	['l1', 'l2', 'elasticnet']	[0.0001, 0.001, 0.01, 0.1, 1]	invscaling', 'adaptive', 'constant', 'optimal	[0.0001, 0.001, 0.01, 0.1, 1]
La meilleur valeur	1	10000	l2	0.0001	adaptive	0.01
MSE	0.2919	0.2919	0.2919	0.2919	0,29	0.2848
R2	0.6127	0.6127	0.6127	0.612	0,58	0.6221

AJUSTEMENET DES HYPERPARAMÈTRES

En ajustant ces hyperparamètres, vous avez réussi à obtenir une meilleure performance avec un MSE réduit et un R2 score amélioré qui sont **MSE = 0.2848** le plus faible et le **R2 = 0.6221** avec cette combinaison des hyperparamètres:

- **Alpha = 0.0001**
  - La valeur de alpha est très faible (0.0001), ce qui signifie que le modèle est peu régularisé. Cela peut expliquer pourquoi nous avons obtenu des résultats relativement bons.
- **penalty= L2**
  - L'utilisation de la pénalité L2 peut aider à améliorer la qualité du modèle en réduisant l'overfitting et en généralisant mieux sur les données de test.
- **Tolerance : 0.01**
  - la valeur optimale pour la tolérance de convergence 0.01

  Cela signifie que pour cette valeur de tolérance, le modèle a la meilleure capacité de prédiction pour les données de test.
- **Learning rate = adaptive**
- **La vitesse d'apprentissage initiale (eta0) =1**

DÉTECTION DE FRAUDE PAR CARTE DE CRÉDIT

Classification Binaire
deux catégories distinctes

- Le jeu de données contient des transactions de carte de crédit anonymisées effectuées par des titulaires de carte européens en septembre 2013.
- Le jeu de données contient un total de 284 807 transactions, dont 492 sont des fraudes.
- Le jeu de données contient les variables suivantes : *
    - Temps : Nombre de secondes écoulées entre cette transaction et la première transaction de l'ensemble de données.
    - V1, V2, ..., V28 : Variables d'entrée anonymisées pour des raisons de confidentialité.
    - Montant : Montant de la transaction.

*La variable cible **Classe** prend la valeur 1 en cas de fraude et 0 sinon.

L'EXPLORATION DES DONNÉES

Vérification de valeurs manquantes

```
Non-missing values: 284807
Missing values: 0
```

Statistiques descriptives

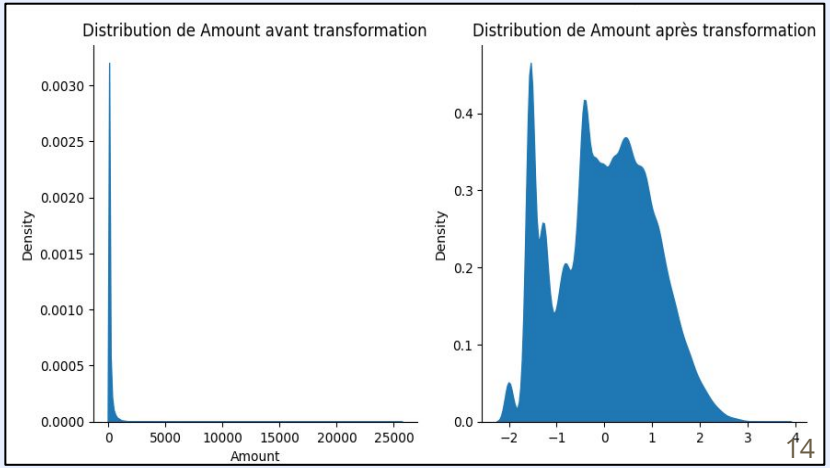
	Time	V1	V2	V3	V4	V5	V6	V7	V8
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05
mean	94813.859575	1.168375e-15	3.416908e-16	-1.379537e-15	2.074095e-15	9.604066e-16	1.487313e-15	-5.556467e-16	1.213481e-16
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00	1.194353e+00
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01	-7.321672e+01
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01

V21	V22	V23	V24	V25	V26	V27	V28	Amount
2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	284807.000000
1.654067e-16	-3.568593e-16	2.578648e-16	4.473266e-15	5.340915e-16	1.683437e-15	-3.660091e-16	-1.227390e-16	88.349619
7.345240e-01	7.257016e-01	6.244603e-01	6.056471e-01	5.212781e-01	4.822270e-01	4.036325e-01	3.300833e-01	250.120109
-3.483038e+01	-1.093314e+01	-4.480774e+01	-2.836627e+00	-1.029540e+01	-2.604551e+00	-2.256568e+01	-1.543008e+01	0.000000
-2.283949e-01	-5.423504e-01	-1.618463e-01	-3.545861e-01	-3.171451e-01	-3.269839e-01	-7.083953e-02	-5.295979e-02	5.600000
-2.945017e-02	6.781943e-03	-1.119293e-02	4.097606e-02	1.659350e-02	-5.213911e-02	1.342146e-03	1.124383e-02	22.000000
1.863772e-01	5.285536e-01	1.476421e-01	4.395266e-01	3.507156e-01	2.409522e-01	9.104512e-02	7.827995e-02	77.165000
2.720284e+01	1.050309e+01	2.252841e+01	4.584549e+00	7.519589e+00	3.517346e+00	3.161220e+01	3.384781e+01	25691.160000

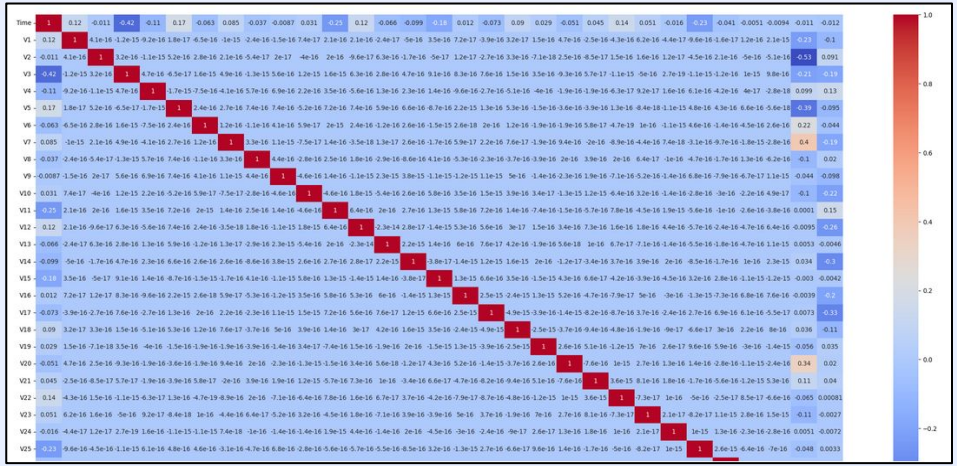
- Montant moyen d'une transaction frauduleuse : 122,21
- Montant moyen dans une transaction valide : 88,29

la transaction monétaire moyenne pour les fraudeurs est supérieure Cela rend un problème.

Transformation (power transformation) pour le montant pour modifier la distribution afin d'améliorer la qualité des résultats

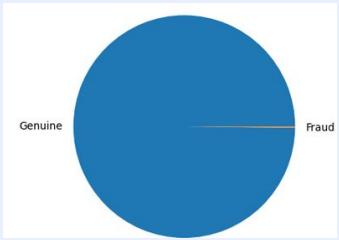


VISUALISATION DES DONNÉES



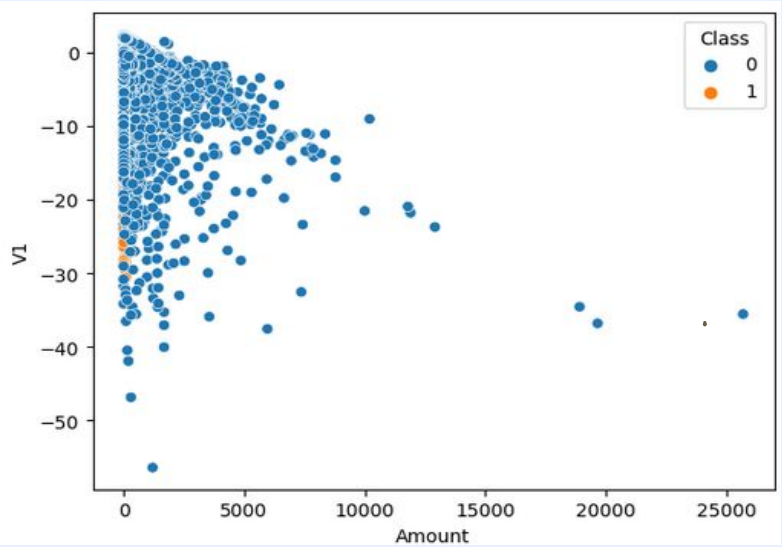
Matrice de corrélation

Pas une relation de causalité entre les variables car les valeurs sont faibles et les couleurs claires indiquent une faible corrélation.



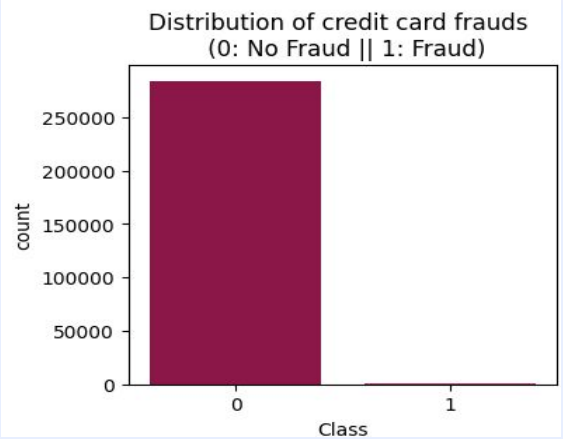
cas de fraude: 492
cas valide: 284315
pourcentage de fraude : 0.0017
=====> Il y a un déséquilibre dans les données

Diagramme circulaire



Nuage de points

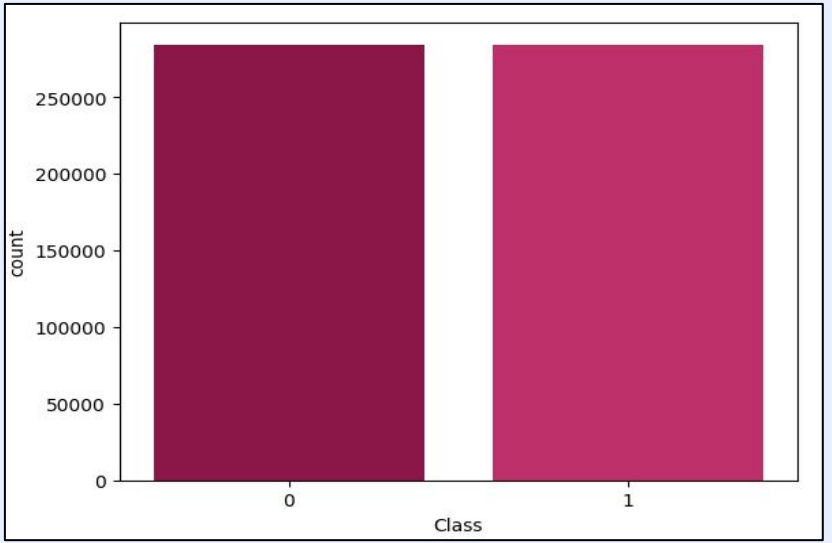
VISUALISATION DES DONNÉES



Graphique de comptage

classe majoritaire "0" et l'étiquette de classe minoritaire "1
Un déséquilibre dans les données qui peut causer un **Surajustement**

SMOTE
qui génère de manière synthétique de nouveaux échantillons pour la classe minoritaire en utilisant des techniques d'interpolation afin d'équilibrer les classes d'un ensemble de données.



NETTOYAGE ET NORMALISATION DES DONNÉES

Valeurs aberrantes

Boxplot

suppression des valeurs aberrantes

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V22	V23	V24	V25
count	315058.000000	315058.000000	315058.000000	315058.000000	315058.000000	315058.000000	315058.000000	315058.000000	315058.000000	315058.000000	...	315058.000000	315058.000000	315058.000000	315058.000000
mean	93968.775810	0.022823	0.800985	-1.028507	1.157852	-0.141279	-0.633243	-0.587595	0.194100	-0.601699	...	-0.003693	-0.028448	-0.070589	0.047650
std	48573.868383	1.512342	1.349987	2.382941	2.161466	1.031599	0.874590	1.415443	0.560913	1.313588	...	0.621184	0.225762	0.486026	0.440138
min	0.000000	-7.646441	-4.414816	-9.664266	-5.560118	-5.201640	-3.890670	-8.407173	-1.773441	-4.944054	...	-2.035130	-0.839102	-1.470535	-1.307953
25%	53413.764827	-0.938588	-0.132335	-2.097710	-0.404671	-0.758150	-1.121615	-1.175640	-0.143379	-1.373117	...	-0.488937	-0.166708	-0.380859	-0.256911
50%	82806.512851	-0.035015	0.665956	-0.428396	0.663091	-0.113893	-0.613725	-0.243300	0.093308	-0.428369	...	-0.027366	-0.034459	-0.025608	0.071640
75%	141692.341661	1.216429	1.497820	0.706739	2.479740	0.502471	-0.107480	0.349841	0.466232	0.246740	...	0.443458	0.109633	0.282772	0.362712
max	172792.000000	2.454930	6.662377	4.187811	8.057871	3.806810	2.382027	2.859481	2.427890	3.680447	...	2.064052	0.803925	1.299212	1.372085

Statistiques descriptives

L'ENTRAÎNEMENT DU MODÉLE

1. Diviser les données en ensemble d'entraînement et ensemble de test

Nous fixons 30 % pour la partie test et 70 % pour la partie entraînement.

2. L'entraînement du modèle

Le modèle est entraîné par algorithme d'apprentissage automatique appelé SGDClassifier, en utilisant une fonction de perte de type logistique.

3. Le choix des métriques d'évaluation

- La précision
- Le score F1
- Rappel
- La matrice de confusion

L'ENTRAÎNEMENT DU MODÉLE

```
Matrice de confusion:
 [[62019   339]
 [ 1529 30631]]
```

- Résultat de prédiction

- Le score de **précision** est de 0,98, ce qui signifie que 98% des exemples prédits comme positifs sont réellement positifs
- Le score de **rappel** est également de 0,98, ce qui signifie que 98% des exemples de la classe positive ont été correctement identifiés par le modèle.
- **Le score F1** est de 0,98 est élevé indique un équilibre entre la précision et le rappel
  ====> Ces résultats suggèrent que le modèle est performant et peut être utilisé pour prédire avec précision la classe de nouveaux exemples.

AJUSTEMENET DES HYPERPARAMÈTRES

La recherche par grille (Grid Search)

Les hyperparamètres qui sont testé:

- loss : [’hinge’, ’log’]
- alpha : [0.0001, 0.001 0.01]
- penalty : [’l1’, ’l2’, ’elasticnet’]
- max iter : [1000, 5000, 10000]

La meilleure combinaison des hyperparamètres :

- alpha : 0.0001
- loss:hinge
- max iter: 10000
- sgd penalty : l1

Conclusion

Après avoir effectué l’ajustement des hyperparamètres, le modèle a été amélioré avec une augmentation significative du score de précision à 0,991 et une légère diminution du score de rappel à 0,965 par rapport aux scores initiaux de 0,980 pour les deux métriques.

LES PERFORMANCES DE SGD

- **Le volume de dataset**

  Le deuxième jeu de données est environ 14 fois plus grand que le premier jeu de données en termes d'observation

  - le modèle peut améliorer sa capacité à généraliser les schémas et les tendances présents dans les données.

  - le modèle a une plus grande probabilité de rencontrer des exemples pertinents pour la tâche d'apprentissage et de capturer les variations et les relations entre les caractéristiques de manière plus précise

  - **Réduction du surapprentissage (overfitting)**

    Le surapprentissage se produit lorsque le modèle s'adapte trop précisément aux exemples d'entraînement spécifiques, mais échoue à généraliser correctement sur de nouvelles données.

    L'utilisation d'un ensemble de donnéesplus volumineux peut aider à réduire le surapprentissage en fournissant plus d'exemples pour apprendre les caractéristiques essentielles et éviter une mémorisation excessive des exemples spécifiques

BIBLIOGRAPHIE

Le lien pour les valeurs par défaut des hyperparamètres de l'algorithme SGD pour regression:
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDRegressor.html

Le lien pour les valeurs par défaut des hyperparamètres de l'algorithme SGD pour classification:

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html#sklearn.linear_model.SGDClassifier

Le lien vers 1er data:
https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html#sklearn-datasets-fetch-california-housing

Le lien vers 2eme data: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?resource=download

Chawla, N.V., Bowyer, K.W., Hall, L.O., et Kegelmeyer, W.P. (2002). SMOTE :Synthetic Minority Over sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357. https ://scikit learn.org/stable/modules/generate

Le lien = lsklearn.metrics.meansquarederror.html

MERCI POUR VOTRE ATTENTION