introduire

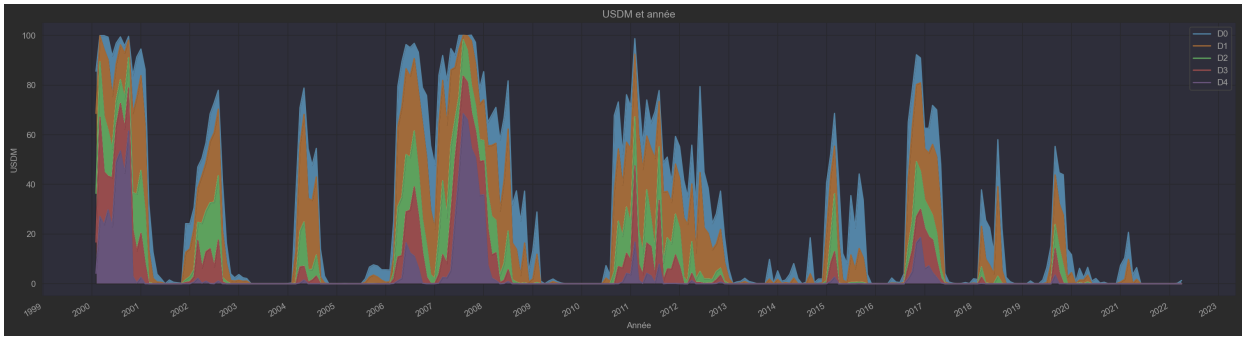
Dans ce projet, nous utilisons le modèle traditionnel de séries chronologiques ARIMA et deux modèles d'apprentissage automatique xgboost et random forest pour prédire l'ensemble de données chronologiques de la sécheresse aux États-Unis. Enfin, nous comparons les performances de différents modèles et analysons les raisons Dans le même temps, nous avons également regroupé les sécheresses dans différents états en utilisant une méthode de regroupement pour trouver des classifications potentielles.

Composition de l'ensemble de données

	DATE	D0	D1	D2	D3	D4	W0	W1	W2	W3	W4	state
0	1895-01-01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	alabama
1	1895-02-01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	alabama
2	1895-03-01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	alabama
3	1895-04-01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	alabama
4	1895-05-01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	alabama
...	...	...	...	...	...	...	...	...	...	...	...	...
73339	2021-12-01	45.2	33.7	19.1	8.9	1.8	6.9	2.7	0.0	0.0	0.0	wyoming
73340	2022-01-01	38.8	27.8	11.2	5.7	2.2	11.5	4.3	0.3	0.0	0.0	wyoming
73341	2022-02-01	40.5	27.6	11.9	4.7	1.4	13.6	8.5	1.9	0.1	0.0	wyoming
73342	2022-03-01	28.5	15.4	4.4	1.7	0.0	22.3	12.3	3.9	0.8	0.0	wyoming
73343	2022-04-01	18.0	10.1	4.4	2.6	0.7	26.8	13.7	1.6	0.6	0.0	wyoming

Une fois l'ensemble de données nettoyé, on peut voir qu'il s'agit d'un ensemble de données de séries chronologiques, dans lequel D0 à D4 représentent les pourcentages de superficie des zones à différents degrés de sécheresse. W0 à W4 représentent le pourcentage de surface des zones avec différents niveaux d'humidité, la variable d'état représente le nom de l'état et les données sont enregistrées une fois par mois, de 1895 à 2022

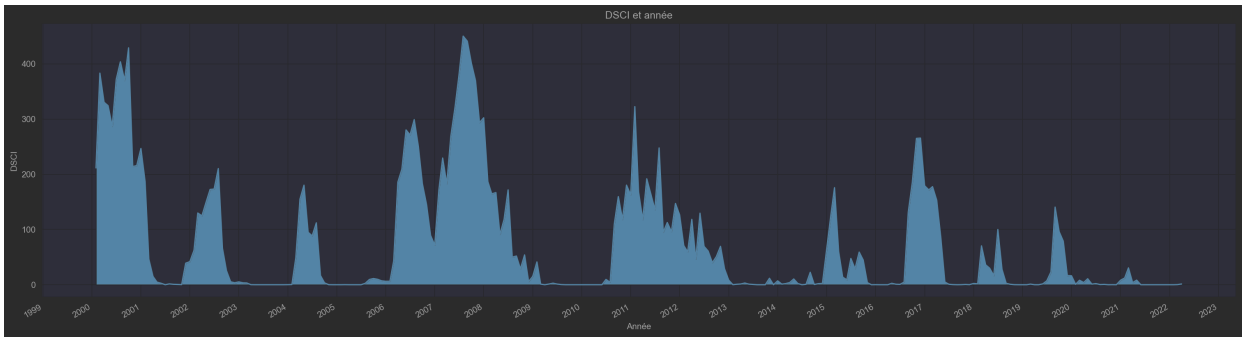
visualisation de données



Afin de voir clairement les données, nous n'observons que les données de l'État de Californie après 2000. Nous pouvons constater que certaines années spécifiques, l'État tombera dans une forte période de sécheresse, comme 2001 et 2008, avec plus de 50% de la zone tombe en sécheresse D4.

c'est pour simplifier le problem, l'analyse de séries temporel multivariées est difficile, donc on va concentrer sur la série temporele avec une variable, c'est pourqoui qu'on utilise le DSCI pour represent le niveau de sécheresse.
Dans la question suivante, on va étudier la série temporele avec une variable. C'est-à-dire prévoir la

valeur de DSCI.
Dans le graphe dernier, on peut regarder toutjours D0<D1<D2<D3<D4, donc on a certitude de considrer ici le D0 est valeur accumulée，cad ‘D0’ = D0 +D1+D2+D3+D4 et donc DSCI = D0 + D1 + D2 + D3 + D4



De cette façon, nous pouvons juger du degré de sécheresse en fonction de la taille du DSCI

méthode de cluster

Intuitivement, nous pensons que les États adjacents devraient avoir des tendances similaires en matière de niveaux de sécheresse, comme l'Alabama et le Mississipi. Nous voulons savoir s'il existe un lien potentiel et une classification entre différents États en fonction de la distance temporelle des données. Nous utilisons donc le cluster méthode de classement

Nous utilisons d'abord la méthode Kmean, qui juge en calculant la distance euclidienne de deux ensembles de données, donc si les valeurs DSCI de deux états sont similaires, alors ils sont plus susceptibles d'être classés ensemble, et finalement nous pouvons obtenir le classement suivant :

class 0：['illinois', 'indiana', 'iowa', 'michigan', 'missouri', 'ohio', 'oklahoma', 'west-virginia', 'wisconsin']

class 1：['connecticut', 'delaware', 'maine', 'maryland', 'massachusetts', ..., 'new-york', 'pennsylvania', 'rhode-island', 'vermont', 'virginia']

class 2：['alabama', 'florida', 'georgia', 'south-carolina']

class 3：['california', 'colorado', 'nevada', 'utah']

class 4：['arizona', 'new-mexico', 'texas']

class 5：['idaho', 'oregon', 'washington']

class 6：['kansas', 'minnesota', 'montana', 'nebraska', 'north-dakota', 'south-dakota', 'wyoming']

class 7：['arkansas', 'kentucky', 'louisiana', 'mississippi', 'north-carolina', 'tennessee']

On peut voir sur la carte que différents états de la même catégorie sont généralement adjacents, ce qui montre que la tendance à la variation et la taille du degré de sécheresse dans les états adjacents sont généralement similaires, ce qui est conforme à la répartition géographique de l'emplacement, et nous pouvons tirer des conclusions : Les sécheresses sont généralement simultanées sur de vastes zones de différents états, couvrant une vaste zone et affectant plusieurs états

méthode de regression

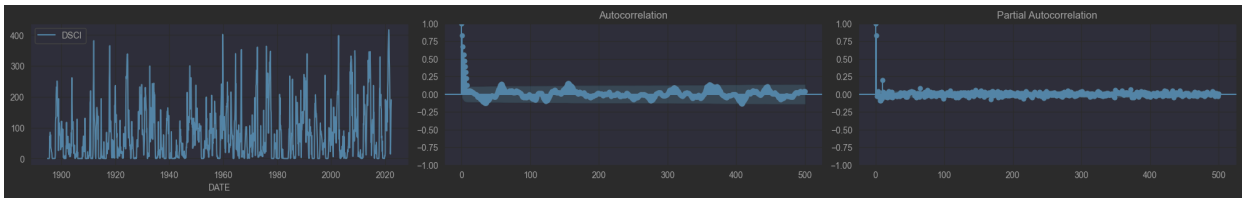
ARIMA

Nous utilisons d'abord la méthode traditionnelle des séries chronologiques ARIMA pour faire des prévisions.

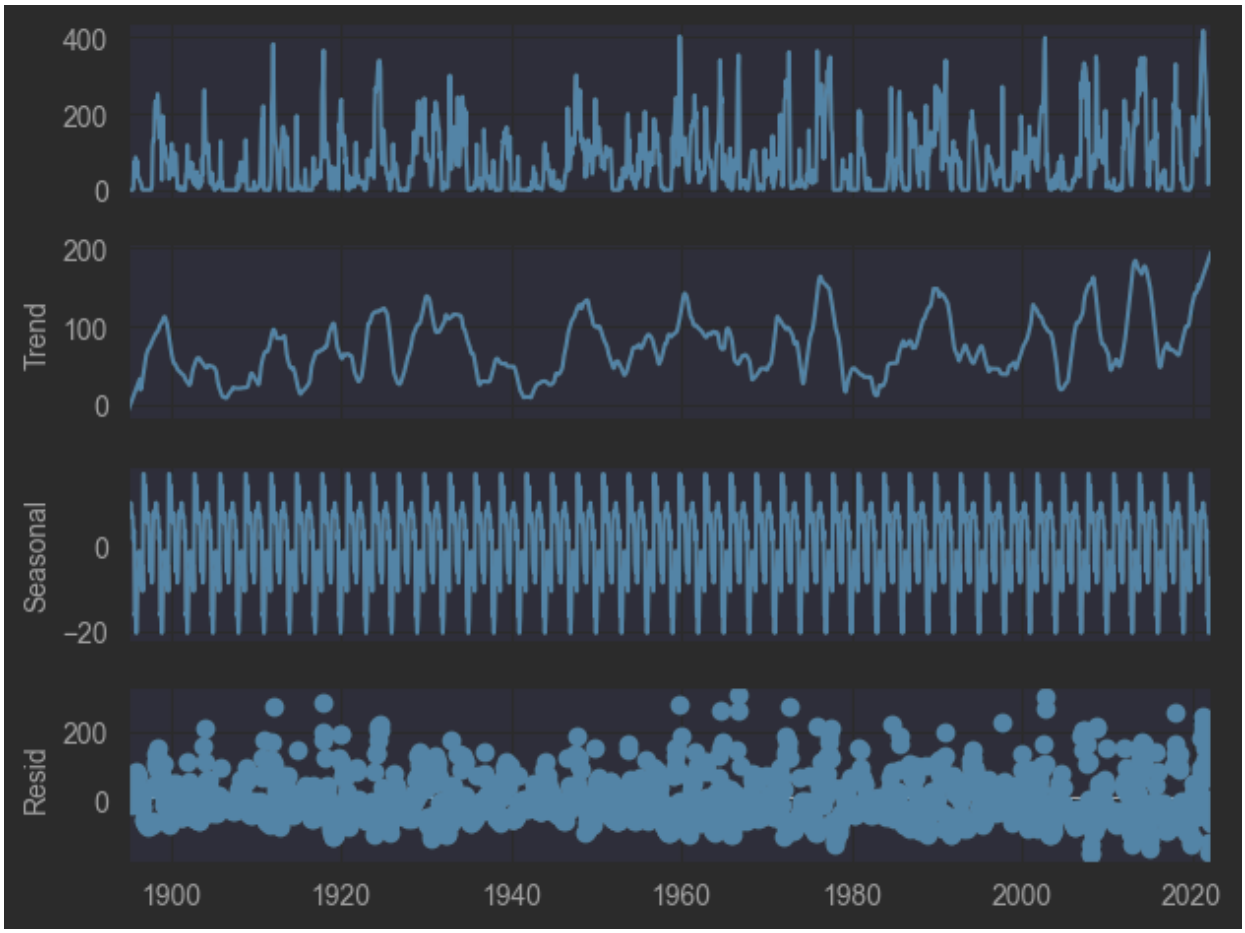
Introduction au principe ARIMA

preparation

Tout d'abord, nous dessinons les diagrammes acf et pacf pour juger si notre série temporelle est stable et périodique



On peut voir que le graphique du coefficient d’autocorrélation a des fluctuations sinusoïdales évidentes et ne se réduit pas à 0, ce qui montre que les données ne sont pas stationnaires et ont une périodicité Lorsque nous utilisons des modèles d’apprentissage automatique traditionnels pour faire des prévisions, le lissage des séries chronologiques peut améliorer la précision des prévisions Par conséquent, nous devons stabiliser cette série temporelle non stationnaire et la décomposer via Time-series Decomposition La méthode de décomposition des séries chronologiques décompose la série originale en 3 séries :Séries tendancielles, séries saisonnières, séries résiduelles
Ici, du fait de la présence de 0 dans DSCI, on ne peut utiliser que le modèle additif décomposé par STL au lieu du modèle multiplicatif



La décomposition STL nous permet d'obtenir la séquence de tendance Trend, qui peut refléter la tendance de changement de DSCI, et c'est aussi ce que nous devons prédire plus tard.

Notre résultat final : res = pred_trend + Seasonal + Resid

Après la décomposition STL, nous voulons juger si la séquence de tendance est stable, nous pouvons utiliser adfuller pour détecter, ici l'hypothèse nulle est que la séquence n'est pas stable

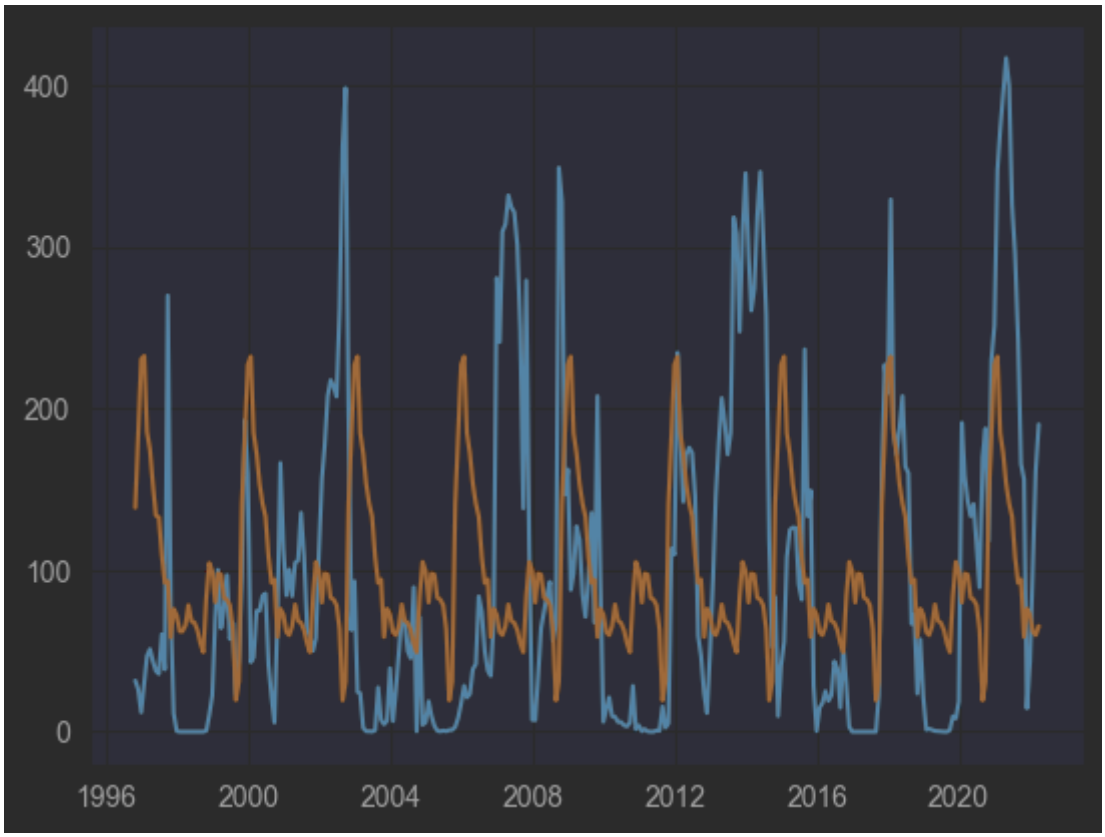
p=1.4651669972728275e-07,

La valeur de p est inférieure à 0,05, nous sommes donc en mesure de rejeter l'hypothèse nulle

Lorsque vous utilisez ARIMA, nous devons saisir manuellement les valeurs des paramètres, c'est-à-dire combien de fois nous devons faire des différences et combien de données sont utilisées pour prédire les données suivantes. Habituellement, ce nombre peut être obtenu en observant les diagrammes acf et pacf, mais pour une représentation plus précise, nous utilisons les informations AIC et BIC pour la recherche de grille, et obtenons enfin le train AIC (11, 0), le train BIC (10, 0) comme meilleurs paramètres

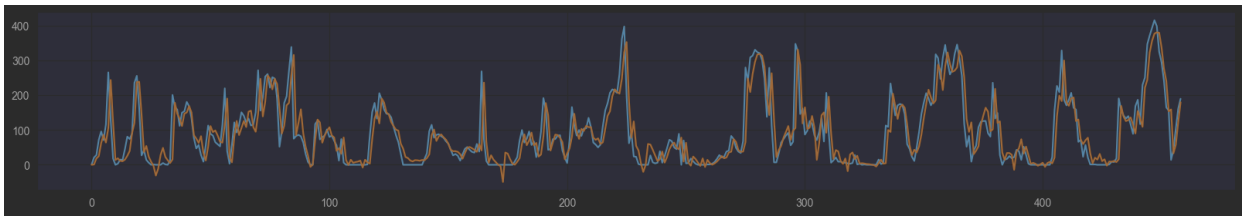
prédire

L'image ci-dessous est l'image du modèle ARIMA prédisant les séries temporelles décomposées par STL



On peut voir que le modèle ARIMA n'est pas performant en prévision à long terme, car le bruit de nos données d'origine est trop aléatoire, et des valeurs extrêmes apparaîtront dans certaines années, et le modèle ARIMA n'est pas très bon pour la prévision à long terme. Pour simuler ce processus, nous ne pouvons faire qu'une prévision approximative basée sur la périodicité

Cependant, le modèle ARIMA obtient généralement un meilleur résultat dans les prévisions à court terme. Si nous prédisons uniquement les prochaines données sur la base de données réelles, c'est-à-dire la valeur DSCI du mois prochain, le résultat sera généralement très bon.



MSE=2760

Xgboost

En plus d'utiliser la méthode traditionnelle de prévision des séries chronologiques ARIMA, nous pouvons également utiliser des méthodes d'apprentissage automatique pour la prévision
Ici, nous utilisons xgboost, une méthode d'apprentissage automatique basée sur des modèles d'arbres

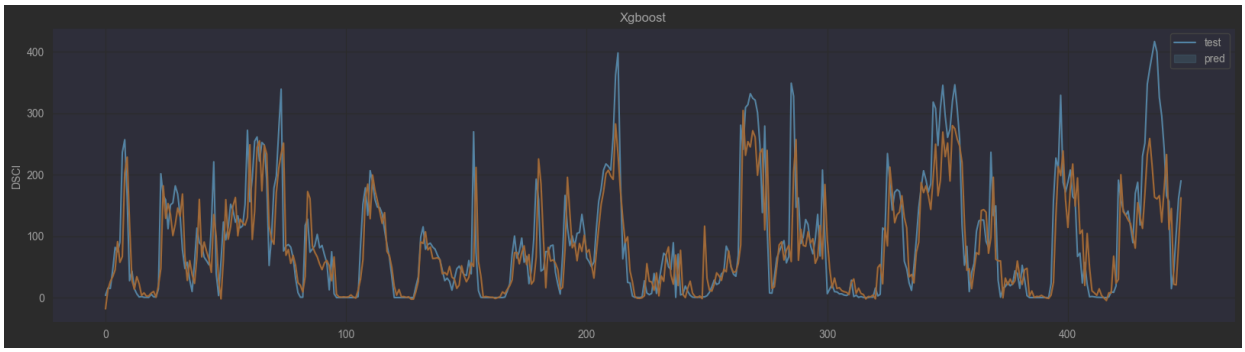
Lors de l'utilisation de méthodes d'apprentissage automatique, nous devons convertir les données de séries chronologiques en données d'apprentissage supervisé
Ici, nous avons également effectué une ingénierie des fonctionnalités, afin que chaque modèle de ligne puisse apprendre des informations d'il y a un an et d'il y a trois ans

En même temps, nous convertissons le temps en codage de variable de catégorie, c'est-à-dire one-hot. En fait, nous pouvons avoir un piège de dimension lorsqu'il y a plusieurs catégories, mais dans le modèle arborescent, encodage à one-hot fonctionne bien.
On prend aussi une fenêtre de longueur 3 et on ajoute la moyenne de cette fenêtre

La figure suivante est un diagramme schématique des données de formation lorsqu'un codage one hot n'est pas utilisé

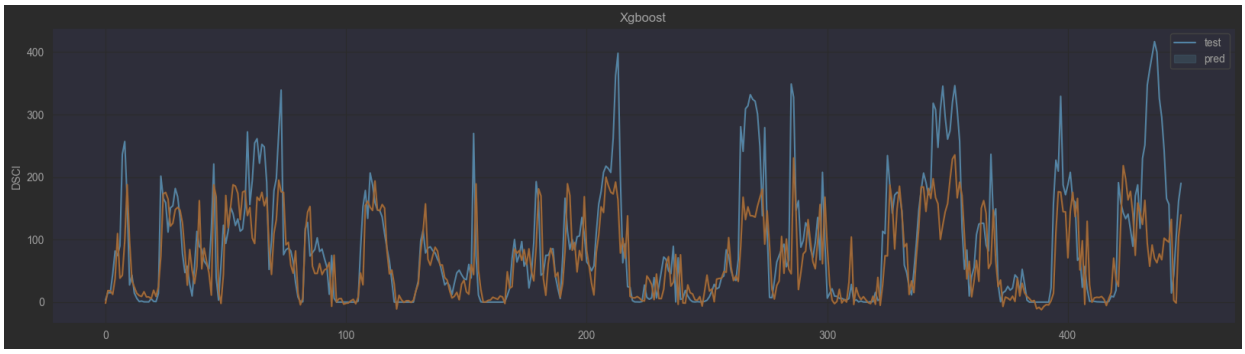
	year	month	day	1mago	2mago	3mago	1yago	2yago	3yago	moyenne
36	1898	1	1	111.5	11.1	0.2	0.0	35.2	0.0	40.933333
37	1898	2	1	132.3	111.5	11.1	0.0	76.7	0.0	84.966667
38	1898	3	1	131.2	132.3	111.5	0.0	39.3	0.0	125.000000
39	1898	4	1	205.7	131.2	132.3	0.0	32.9	0.0	156.400000
40	1898	5	1	231.7	205.7	131.2	0.0	29.6	0.0	189.533333
41	1898	6	1	198.4	231.7	205.7	0.0	22.1	0.0	211.933333
42	1898	7	1	191.4	198.4	231.7	0.0	20.5	0.0	207.166667
43	1898	8	1	250.8	191.4	198.4	0.0	15.3	0.0	213.533333
44	1898	9	1	229.4	250.8	191.4	0.0	2.6	2.6	223.866667
45	1898	10	1	172.5	229.4	250.8	0.2	0.0	71.6	217.566667

Après avoir utilisé 70 % des données comme données d'entraînement, nous pouvons obtenir la figure suivante



MSE = 3387

La figure suivante est un diagramme schématique des données de formation lorsqu'un codage one hot n'est pas utilisé



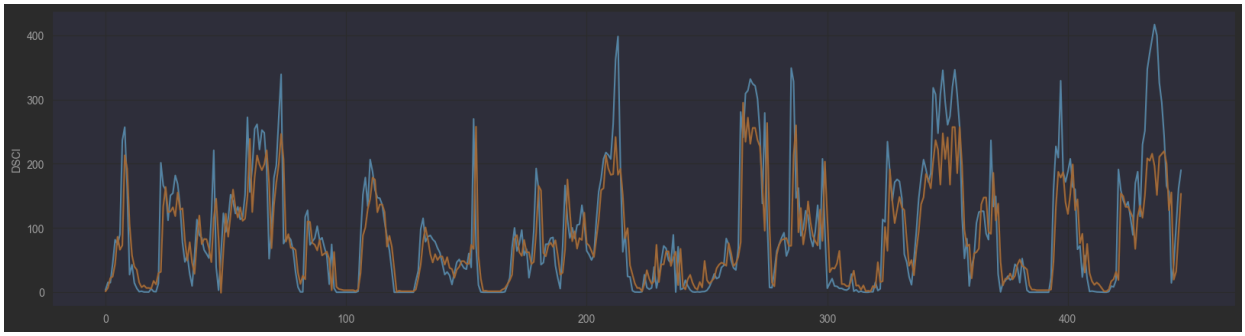
MSE= 5297

En effet, xgboost calcule automatiquement le mois et l'année en tant que variable continue, mais en fait, le mois et l'année doivent être une variable de catégorie. Nous devons donc en utiliser une à chaud pour l'encodage.

Ensuite, nous voulons ajuster les paramètres pour obtenir le modèle de paramètre optimal.Nous utilisons la grille de recherche dans sklearn pour rechercher les paramètres, obtenir n_estimators = 14 est le meilleur, puis fixer n_estimators pour rechercher gamma, et rechercher sous-échantillon et colsample_bytree après avoir obtenu gamma Les paramètres finaux sont les suivants :

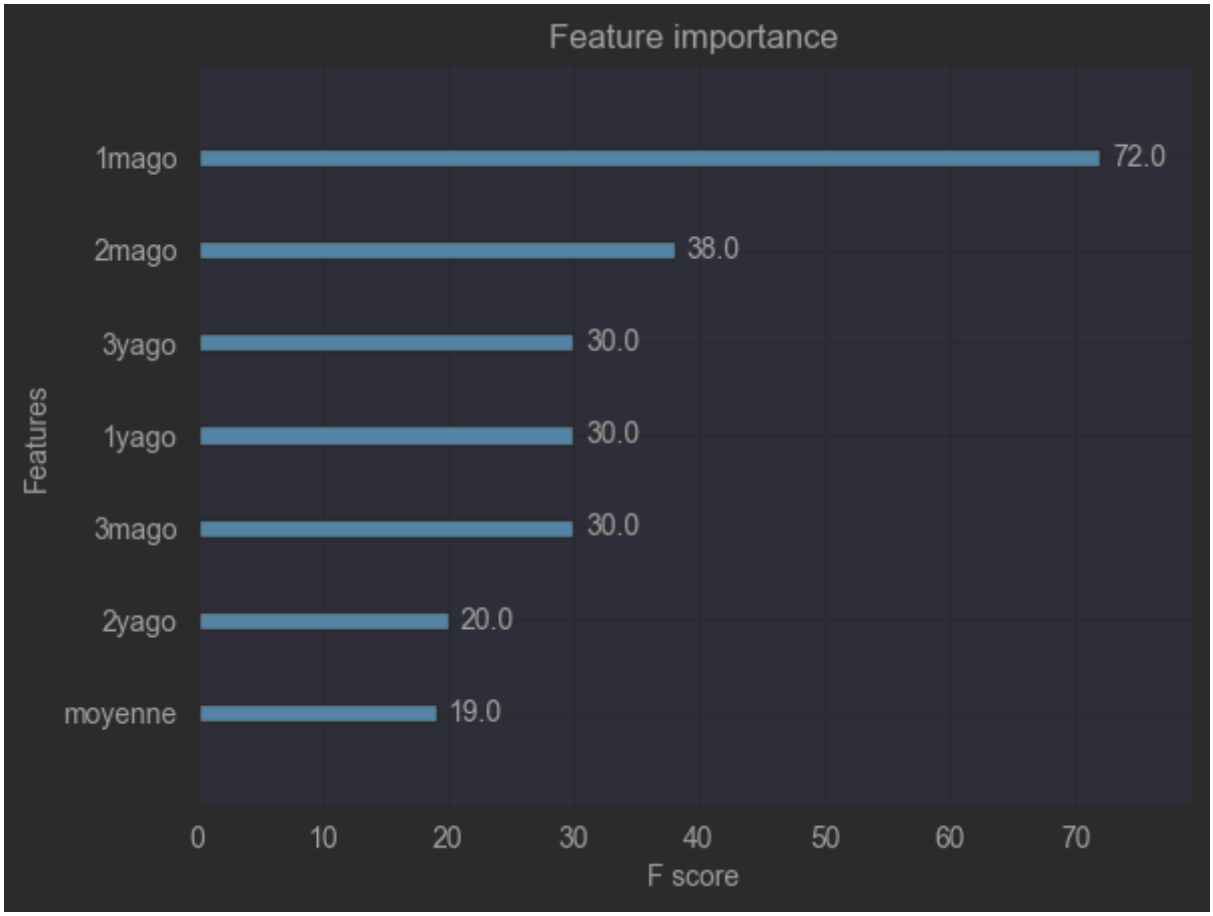
n_estimators=14,gamma=0.4,colsample_bytree=0.6,subsample=0.8

Utilisez ce paramètre pour obtenir l'image prédite ci-dessous :



MSE = 3246

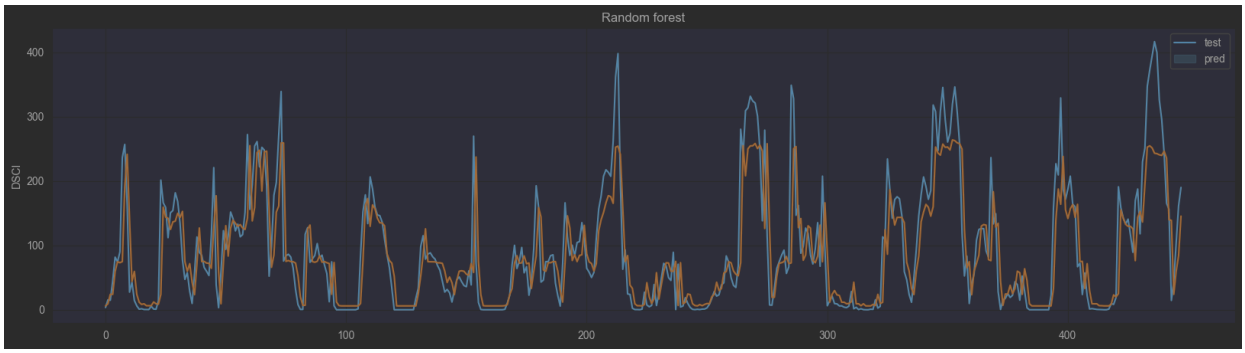
Nous voulons également voir quelles variables contribuent le plus au modèle, comme le montre la figure suivante :



On peut voir que la principale variable de prévision est la valeur DSCI il y a un mois, et Moyenne contribue le moins

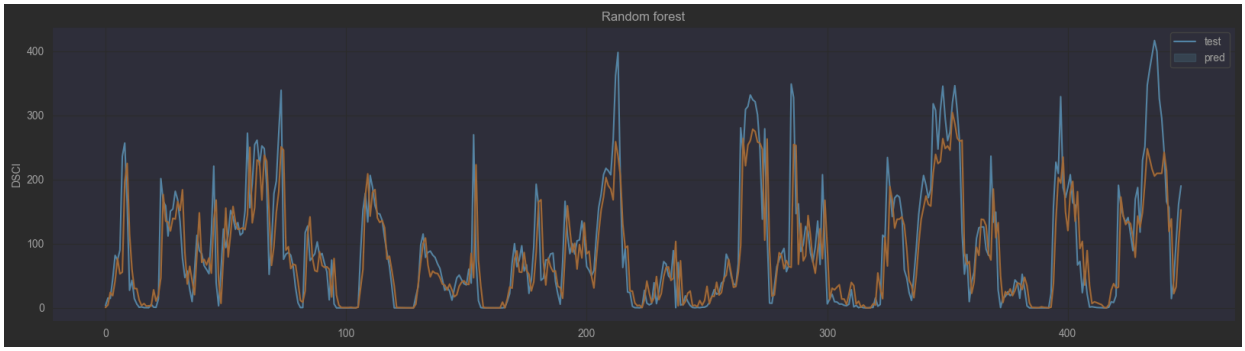
random forest

Pour le traitement des feature, nous adoptons la même méthode de traitement que xgboost, et nous pouvons alors obtenir la figure suivante



MSE = 2991

Prédiction après l'obtention des paramètres optimaux après la recherche de grille

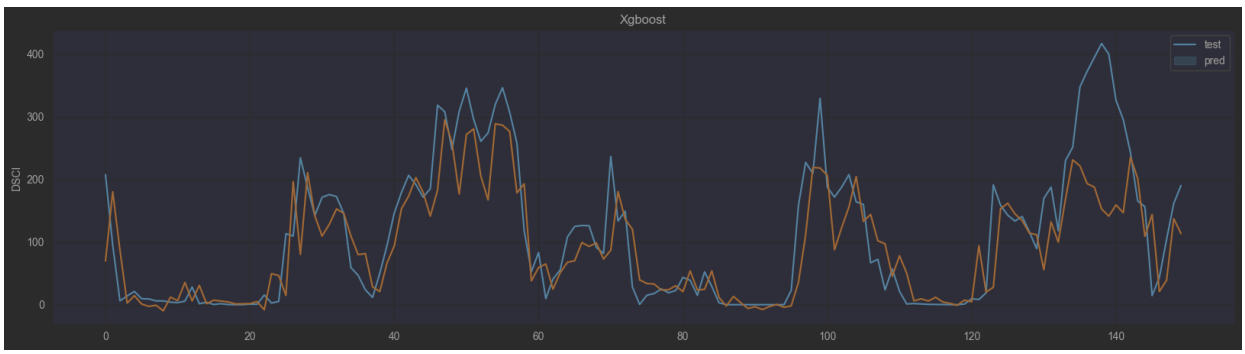


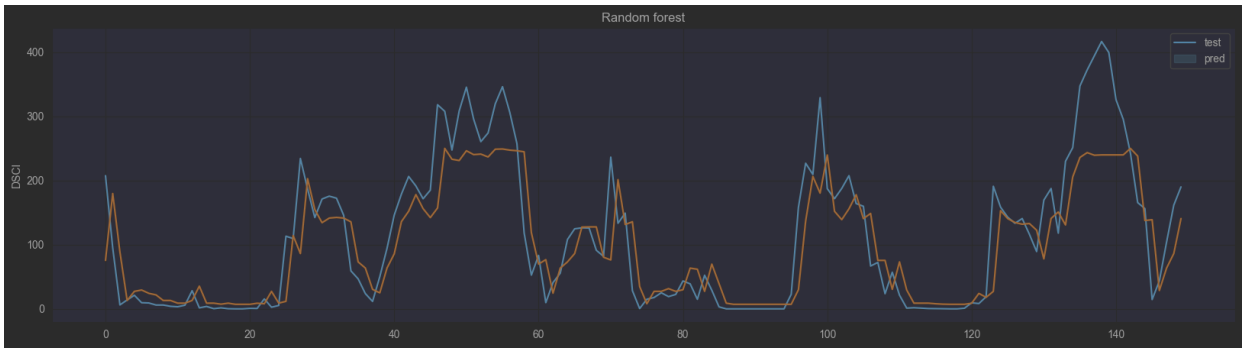
MSE = 2940

comparaison de modèles

Dans ARIMA, xgboost et random forest, le modèle ARIMA est le meilleur selon MSE. Cependant, le modèle ARIMA ne peut prédire les valeurs que sur une très courte période de temps, comme la prochaine fois.Si vous souhaitez faire des prédictions à long terme, car ARIMA utilise les données précédentes pour prédire les données suivantes, des erreurs se produiront. s'accumulent, et finalement le modèle ARIMA convergera vers une valeur, qui est aussi le défaut d'ARMA

Les deux modèles d'apprentissage automatique Xgboost et la forêt aléatoire sont basés sur le modèle arborescent. Bien que le modèle de forêt aléatoire soit meilleur que Xgboost du point de vue de MSE, il y a évidemment un grand décalage dans la valeur de prédiction DSCI du modèle rf. Le changement de données n'est pas aussi sensible que xgboost, comme le montre la figure ci-dessous





Comparé à xgboost, le modèle rf n'est pas très sensible aux données. En même temps, les deux modèles ont un inconvénient. Leurs prédictions pour les valeurs extrêmes sont conservatrices, c'est-à-dire qu'elles ne peuvent pas atteindre le point le plus élevé du test. .

problèmes existants

Lors de l'utilisation de méthodes d'apprentissage automatique, il y a toujours un délai entre les données prédites et les données réelles, ce qui fait que le modèle ne reflète pas bien les changements de la valeur réelle. Lorsque ce problème se produit, le modèle doit être mis à jour après plusieurs pas de temps. afin de percevoir la tendance à la hausse ou à la baisse, par exemple, lorsque l'on prend trois pas de temps X0, x1, x2 pour la prédiction, uniquement lorsque x0<x1<x2, le modèle sait qu'il y a une tendance à la baisse à ce moment, ce qui est également car dans notre Il n'y a pas assez de variables dans l'ensemble de données pour fournir des informations, nous ne pouvons donc faire que des prévisions basées sur des valeurs DSCI historiques. Si nous pouvons fournir plus d'informations telles que les précipitations locales, la température et le climat, nous devrions être en mesure de réduire davantage ce délai, rendant ainsi les prédictions variables.

Dans le même temps, étant donné que la fonction de perte vise à minimiser la MSE, si la valeur du pas de temps précédent est directement prise comme valeur actuelle à chaque fois, la MSE peut également paraître petite, mais un tel modèle apprend à peine efficace Par conséquent, nous devons utiliser une fenêtre glissante pour augmenter la variable

L'objectif futur est d'essayer d'utiliser des méthodes d'apprentissage en profondeur, telles que les méthodes RNN et les réseaux LSTM améliorés, pour prédire les ensembles de données de séries chronologiques