Maintenance prédictive : Prédire la panne de la pompe à eau

Zahra ELHAMRAOUI

Un rapport soumis en tant que micro-projet pour le rôle d'ingénieur en apprentissage automatique chez le groupe MSDA

juillet 2020

Table des matières

1 Problème / Cas d'utilisation

Définir le problème est très important dans la résolution de notre challenge, en principe une équipe qui prend en charge les pompes à eau d'une petite zone, a eu **7 pannes de système** l'année dernière.
Ces échecs causent un énorme problème pour de nombreuses personnes et entraînent également de graves problèmes de vie pour certaines familles. L'équipe ne voit aucun motif dans le données lorsque le système tombe en panne, de sorte qu'ils ne savent pas où mettre plus d'attention.

L'**objectif** est l'utilisation des techniques d'apprentissage automatique pour prédire la panne de la pompe à eau avant qu'elle ne se produise.

On peut dire clairement que c'est un problème du **Maintenance prédictive (PdM)**, PdM est le type de maintenance le plus avancé actuellement disponible. Avec une maintenance basée sur le temps, les organisations courent le risque d'effectuer trop ou pas assez de maintenance. Et avec la maintenance réactive, la maintenance est effectuée en cas de besoin, mais au prix de temps d'arrêt imprévus. La maintenance prédictive résout ces problèmes. La maintenance n'est planifiée que lorsque des conditions spécifiques sont remplies et avant que l'actif ne tombe en panne.

2 L'analyse exploratoire des données (AED)

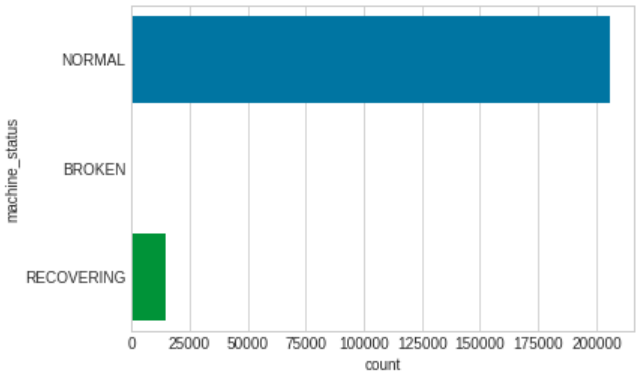
2.1 Description du jeu de données

L'ensemble de données est extrait de Kaggle et contient des informations provenant de 52 capteurs différents.

Le résultat contient trois catégories : NORMAL, RECOVERY, BROKEN.

Nous visons à prédire la panne des machines avant qu'elle ne se produise, car cela créera du temps pour une arrêt afin d'éviter des dommages plus graves ou l'arrivée d'un technicien et enquêtant sur le problème. Les données sont récupérées une fois par minute, ce qui nous fournit un ensemble de données valide pour les prédictions. L'ensemble de données se compose de 220320 observations, qui sont réparties dans les groupes suivants :

Normal : 205,836 Recovering : 14,477 Broken :7.

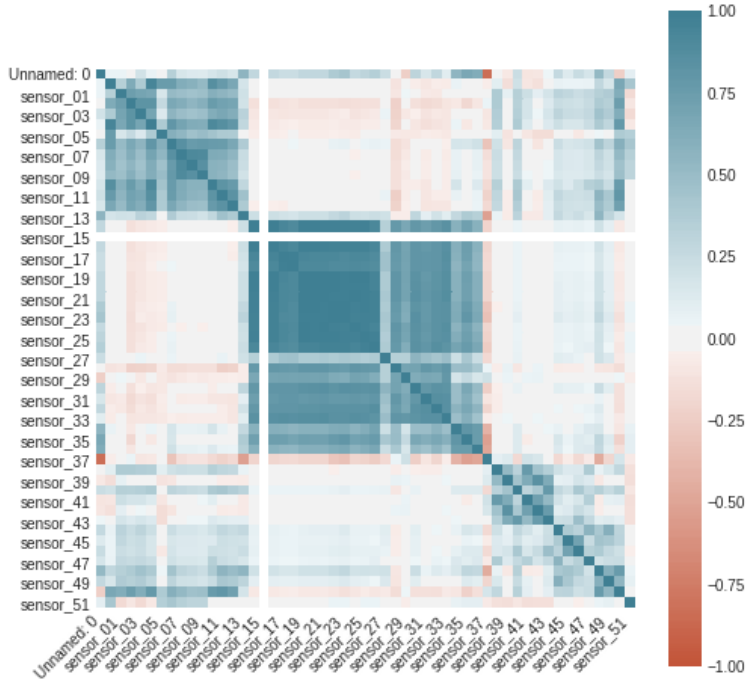


Nous avons un ensemble de données déséquilibré dont les observations normales contribuent avec 93,4%, La récupération contribue avec 6,5%, et enfin cassé contribue avec 0,003%.

2.2 Analyse exploratoire des données

Heat Map

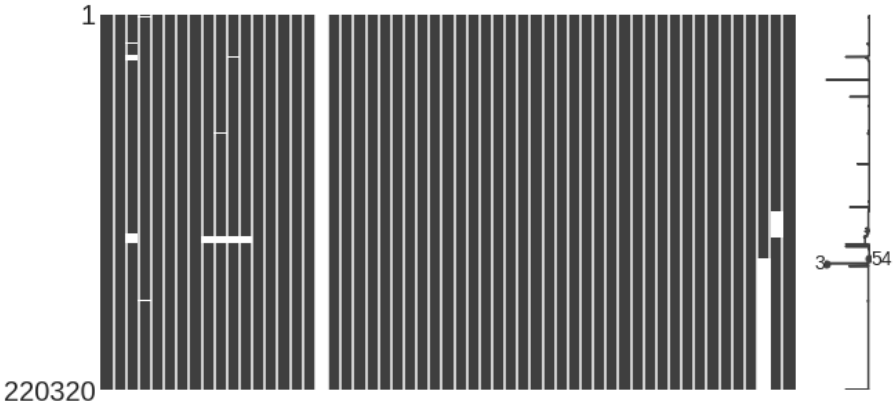
Pour étudier les corrélations entre les valeurs, nous créons une carte thermique (Heat Map) :



Sur la base de la corrélation, il y a trois ensembles de grappes Fondamentalement, cela signifie que certains capteurs de 0 à 14 ont des corrélations élevées les uns avec les autres Les capteurs 14 à 36 ont une forte corrélation les uns avec les autres, puis les capteurs 38 à 41 ont des corrélations entre eux mais il n'y a pas de corrélation significative entre ces clusters de capteurs.

Traiter les valeurs manquantes :

Normalement, lorsque vous traitez pour la première fois vos données, le principal problème dont vous souhaitez vous débarrasser concerne les valeurs manquantes et comme vous pouvez le remarquer sur la figure, le blanc se réfère à NaN / valeur manquante, nous pouvons remarquer un capteur sans données d'autres avec beaucoup de données manquantes.



Après analyse des valeurs manquantes du notre ensemble de données on a eu un capteur vide sensor_15 d'où nous avons pris la décision de l'enlever, pour les autres valeurs manquantes nous avons travaillé avec **df = df.fillna(method='ffill')** qui fait le remplissage avec la dernière observation.

2.3 prétraitement des données

Codage des données

l'état de la machine est une fonction catégorique. Il doit donc être one Hot encoded, et il faut normaliser les données dans chaque état de la machine.

Analyse en composantes principales (ACP)

la dimension d'entrée est trop élevée, entrer autant de fonctionnalités dans notre modèle peut être délicat, ACP est une méthode que nous pouvons exploiter.
L'ACP est une technique statistique non supervisée et non paramétrique principalement utilisée pour la réduction de la dimensionnalité dans l'apprentissage automatique. Une dimensionnalité élevée signifie que le jeu de données a un grand nombre d'entités. Le problème principal associé à la haute dimensionnalité dans le domaine de l'apprentissage automatique est le surajustement du modèle, qui réduit la capacité de généraliser au-delà des exemples de l'ensemble d'apprentissage.

3 Approche classique de l'apprentissage automatique

Nous utilisons d'abord l'apprentissage automatique pour générer un modèle de prédiction ainsi que pour générer des fonctionnalités importance. De plus, nous utilisons des modèles d'apprentissage automatique spécifiques comme Random Forest car ils fonctionnent très bien sur l'ensemble de données déséquilibrés. Dans ce modèle, nous conservons les trois classifications et les réaffectons comme 0 =Normal, 1 = récupération et 2 = cassé.

---Cross-validation Accuracy Scores---

	Model	Score
6	Random Forest	99.99
0	KNN	99.96
5	Decision Tree	99.96
4	Linear SVC	99.58
1	Logistic Regression	99.51
3	Stochastic Gradient Decent	99.45
2	Naive Bayes	97.74

---Reuglar Accuracy Scores---

	Model	Score
5	Decision Tree	100.00
6	Random Forest	100.00
0	KNN	99.98
4	Linear SVC	99.69
1	Logistic Regression	99.51
3	Stochastic Gradient Decent	99.48
2	Naive Bayes	97.74

En utilisant Random Forest, nous pouvons créer un modèle avec une précision de 99,99%, ce qui nécessiterait normalement des inquiétudes quant à savoir si le modèle est sur-ajusté, et c'est pour cela que nous irons voir les autres métriques. Test Accuracy : 0.9998789639312515. Nous obtenons un résultat parfait dans les données de test, donc on a pas ce problème d'overfitting. De plus, sur la base des résultats de l'analyse descriptive, il se peut que nous puissions atteindre cette précision élevée, car chaque classe diffère considérablement les uns des autres.

4 Réseau de neuronal récurrent

Le réseau neuronal récurrent est également connu sous le nom de RNN, est une classe de réseaux neuronaux qui permettent aux sorties précédentes d'être utilisées comme entrées tout en ayant des états cachés.

4.1 Mémoire à long-court terme(LSTM)

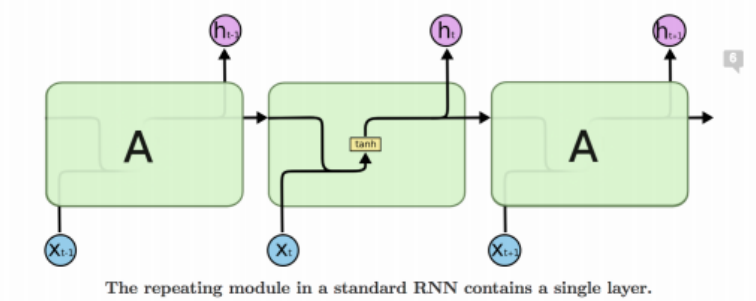
LSTM est un réseau neuronal complexe avec une cellule de mémoire, une entrée, une sortie et oublier (portes) (voir la figure ci-dessous). La cellule de mémoire peut stocker les données précédentes dans une quantité spécifiée, ce qui peut être utilisé pour aider à améliorer la prédiction. À l'aide d'un portail, il est possible d'ajouter ou de supprimer informations dans une cellule.

L'algorithme LSTM nécessite une forme d'entrée en 3 dimensions. Les trois dimensions sont :

*** Batch size :** le nombre d'observations.

*** Time steps :** combien d'observations précédentes à retenir.

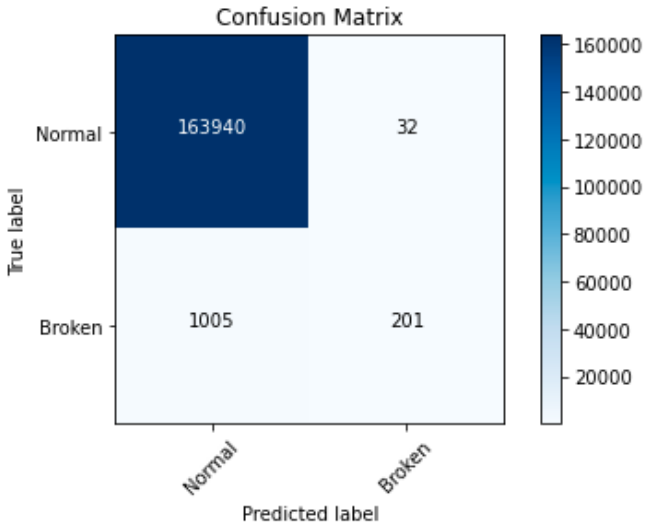
*** Features (input_dim) :** le nombre de fonctionnalités différentes que nous utilisons pour prédire le résultat.



The repeating module in a standard RNN contains a single layer.

4.2 Résultats

Le LSTM multicouche (2-LSTM) est formé. La première couche est une couche LSTM avec 100 unités suivie d'une autre couche LSTM avec 50 unités. Le Dropout est également appliquée après chaque couche LSTM pour contrôler le sur-ajustement. La couche finale est une couche de sortie dense avec une unité et une activation sigmoïd car il s'agit d'un problème de classification binaire. le réseau est optimisé avec Adam. Nous avons eu un score de précision du train de 99.37%.



En règle générale, les ensembles de données de maintenance prédictive ne sont pas équilibrés, vous n'utilisez donc pas de mesures de précision standard.I mesures à prendre en compte sont comme l'attention que vous voulez accorder au taux vraiment positif ou au taux négatif.
Dans notre cas, nous nous intéressons à la qualité de notre modèle. prédire la probabilité du futur panne, d'où on peut prendre nos précautions afin d'éviter l'échec.
Cela signifie que nous acceptons certains faux positifs, car ceux-ci fonctionnent comme une marque pour un échec a venir.
Training Precision : 0.86, Training Recall : 0.16, Training F1 Score : 0.27, F-beta score : 0.47.

5 Conclusion et travail futur

J'ai opté pour la résolution de ce problème deux approches une l'apprentissage automatique classique qui pourra être bénéficiaire dans une découverte puissante de perspicacité avec Random Forest et une précision presque parfaite après j'ai utilisé les réseaux de neurones récurrentes plus précisément LSTM qui m'a permit de prédire l'échec dans les prochains jours avec une précision de 99%.

J'ai pas eu le temps de tester plusieurs autre approches comme :

* **Grid-LSTM** qui s'attaque au gradient qui disparaît problème et le gradient qui explose problème mieux que le LSTM traditionnel.

* **Utilisation de différentes architectures** avec différents nombres de couches et nœuds.