La maintenance prédictive d'un compresseur à gaz

SOMMAIRE

REMERCIEMENTS

Je remercie tout d'abord la Business Unit SMART DATA & IOT avec Mr Maille Laurent, Mme Bonnafous Carole, Mr Bachelin Loïc et Mr Morisson Jérôme (SPIE Industrie) pour m'avoir fait confiance dans la mise en place du projet. Ensuite, je remercie l'équipe Innovation et en particulier les membres ayant participé à la mise en œuvre du projet. J'ai apprécié les échanges constructifs techniques et de gestion de projet.

Je remercie également l'ensemble des formateurs ayant permis de m'épanouir dans le développement en DATA/Intelligence Artificielle par leurs propositions de pistes de travail et les échanges relatifs à la construction d'une démarche intellectuelle, à la gestion de projet. Je tiens aussi à exprimer ma gratitude à l'ensemble des formateurs et responsables de gestion de projet pour m'avoir aidé à trouver une alternance, pour m'avoir permis de créer des contacts lors de différents évènements (Hackathon, meetings), pour m'avoir permis d'exprimer mon vécu lors de promotions du métier de développeur notamment dans les écoles et pour le contenu de la formation.

J'ai une pensée particulière pour Mr Dallard Benjamin (formateur) et Mme Boyer Elodie (responsable gestion de projet) pour leur bienveillance auprès de l'ensemble des alternants.

Enfin je remercie l'ensemble de la promotion avec lesquels j'ai vécu en communauté pendant presque deux ans.

1) INTRODUCTION

La maintenance prédictive aussi appelée maintenance prévisionnelle est basée sur l'anticipation du franchissement d'un seuil prédéfini qui permet de donner l'état de dégradation du bien avant sa détérioration complète (1). Elle fait appel à des algorithmes d'apprentissage machine (champ d'étude de l'intelligence artificielle) et promet une révolution dans l'industrie.

Mon projet, cadré par la maintenance prédictive, a été mis en œuvre lors d'une alternance de 6 mois au sein de SPIE ICS à Nîmes, dans le service Incubation comptant 7 personnes.
Cette alternance a été effectuée en pleine formation développeur DATA/Intelligence Artificielle de 2 ans.
Spie ICS, entreprise de services numériques, est une filiale de Spie France. Elle est spécialisée dans le conseil, l'ingénierie, l'intégration, l'infogérance, la maintenance, les services opérés et le Cloud. Elle compte 3000 employés.

Mes missions au sein de l'entreprise étaient d'utiliser l'apprentissage automatisé pour de la maintenance prédictive, tester différentes plateformes IA existantes, établir un système de classement de fichiers.
Plus largement cette alternance était pour moi l'occasion de m'immerger dans le milieu informatique et d'appréhender des méthodes de travail, agiles notamment. Elle avait pour objectif personnel de conforter mon envie de travailler dans le secteur DATA/Intelligence Artificielle.

Pour présenter la maintenance prédictive effectuée sur un compresseur à gaz, situé à Saint-Martin-de-Crau, pour le transport de gaz naturel j'ai divisé mon rapport en 5 parties.
Dans un premier temps, je vais aborder la problématique client et la traduction en objectifs.
Dans un deuxième temps, je vais faire un état de l'art sur les compresseurs à gaz, la maintenance prédictive et les algorithmes IA associés, exposer un use case.
Dans un troisième temps, j'aborderai la traduction des spécificités fonctionnelles en spécificités techniques
Dans un quatrième temps, je discuterai de ma gestion de projet et je ferai un retour d'expérience des outils et techniques utilisés.
Dans un cinquième temps, j'établirai un bilan et les axes d'améliorations à apporter au projet.
Enfin, j'effectuerai une conclusion valorisant l'apport de la maintenance prédictive dans la problématique client.

2) LA COMPREHENSION DU BESOIN CLIENT

L'appel à projet a été effectué par Spie Industrie mais le client final est un distributeur de gaz naturel.

La France ne fait pas partie des principaux producteurs de gaz naturel en Europe contrairement à la Grande-Bretagne et les Pays-Bas.

Sa consommation de gaz devrait diminuer de 1,2% par an dans les deux décennies à venir (atteignant 364 TWh en 2035 contre 413 TWh en 2016) (2).

Cependant la demande est croissante dans les secteurs des transports (GNV : gaz naturel pour véhicules), industries et centrales électriques (3).

La distribution de gaz naturel en France est contrôlée à hauteur de 96% par Gaz Réseau Distribution France (GRDF), filiale d'Engie.

Les pertes de pression dans le réseau du client augmentent la consommation et les coûts en énergie. La perte d'1 bar provoque une augmentation de 7 % de la consommation énergétique d'un compresseur, il est donc nécessaire d'effectuer une maintenance afin d'éviter une augmentation des coûts énergétiques (4).

Dans ce contexte, le client a décidé de choisir de stabiliser les performances de son réseau de distribution en adoptant la maintenance prédictive.

Le client se pose une question, comment la maintenance prédictive de compresseurs à gaz va lui permettre d'identifier les dysfonctionnements dès qu'ils apparaissent et ainsi augmenter la durée de fonctionnement et la longévité des équipements à moindre coût?

En effet la promesse tenue par la maintenance prédictive est de diminuer les temps d'arrêt, favoriser des coûts de réparation réduits, maitriser le budget de maintenance, garantir des performances élevées (5).

Les objectifs formulés par le client sont :

1) Prédire si dans X temps une dépressurisation en phase d'arrêt va avoir lieu ou pas ou prédire le temps restant avant la panne.
2) Définir les variables corrélées à la dépressurisation pour une maintenance utile.
3) Présenter les résultats obtenus sous forme de Dashboard accessibles via internet.
4) Identifier les oscillations de pression anormale (supérieures à 10 bar) en phase de fonctionnement du compresseur et la possible dégradation des variables corrélées (objectif secondaire).

Pour enrichir le projet dans un environnement IOT d'autres objectifs personnels ont été fixés :

5) Créer une base de donnée, la stocker dans un bucket, et présenter des données d'analyse.
6) Utiliser des techniques d'intégration de données en temps réel.
7) Créer, déployer un webservice et containeriser l'application.

3) L'ETAT DE L'ART

- **Les machines tournantes :**

Par abus de langage le terme compresseur à gaz a été utilisé tout au long de l'alternance. Le terme adéquat est turbine à gaz (ou encore turbine à gaz de combustion). Elle représente une machine tournante thermodynamique appartenant à la famille des moteurs à combustion interne.

Une turbine à gaz utilise le dioxygène de l'air ambiant comme comburant et le comprime. Par combustion du mélange comburant-carburant il se produit une augmentation brutale de la pression et du volume du mélange gazeux mettant en rotation rapide la turbine (6).

Les éléments constituants une turbine à gaz sont : l'entrée, le compresseur, la chambre de combustion, la turbine, la tuyère (figure 1).

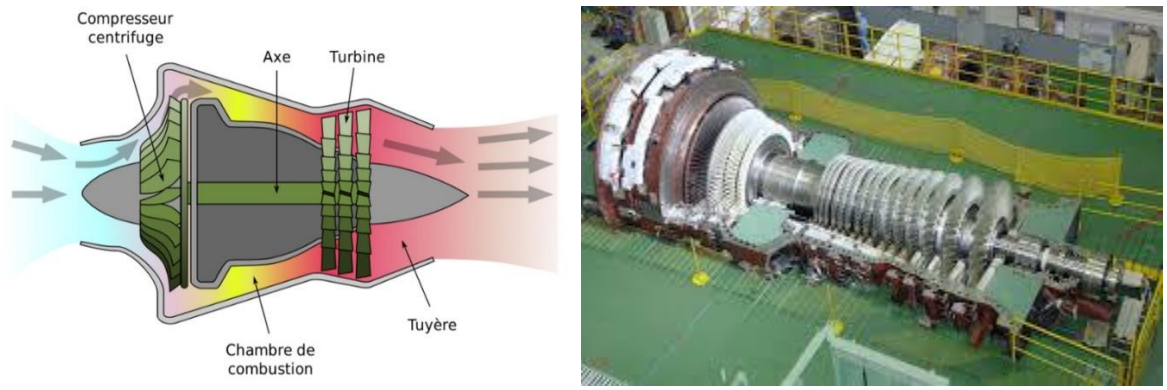


Figure 1 : Images de turbines à gaz

- **La maintenance prédictive :**
  - **Cadre et caractéristiques**

Elle repose sur 4 technologies clés (figure 2).

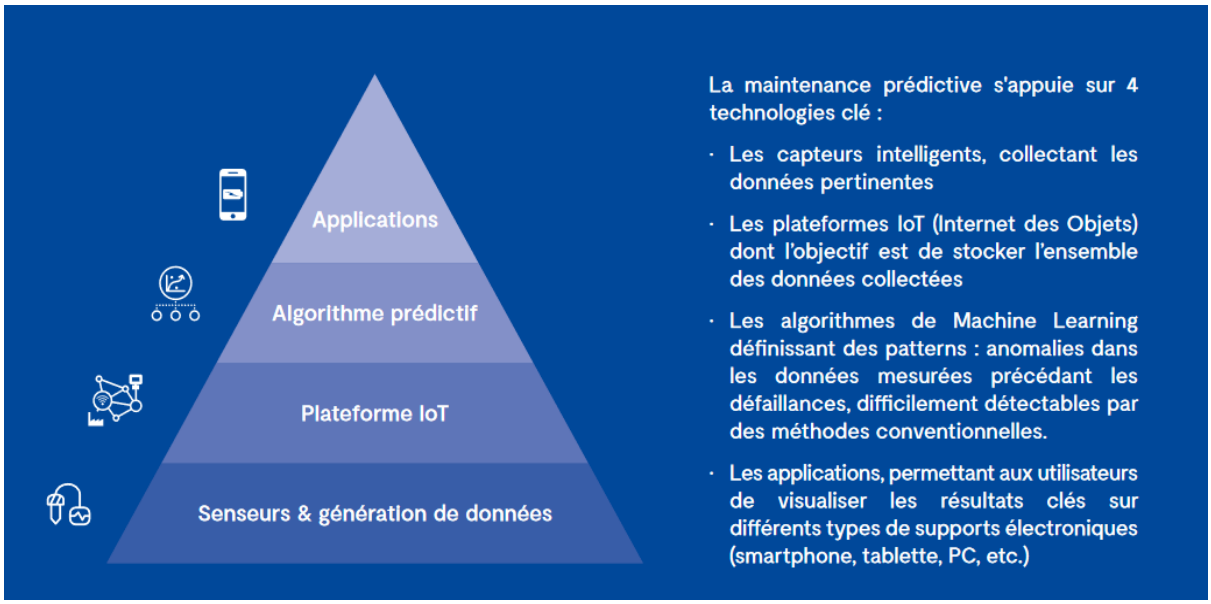


Figure 2 : Les 4 technologies clés de la maintenance prédictive

Le cadre de la maintenance prédictive est décrit dans la figure 3.
Elle utilise des données historiques et effectue une surveillance en temps réel du comportement d'un produit.

	Maintenance prédictive
Quel est son objectif ?	Surveiller les tendances prédisant les circonstances ; prévoir le moment auquel l'équipement pourrait tomber en panne et l'empêcher grâce à des opérations de maintenance
Par qui est-elle effectuée ?	Expert de contrôle, analystes, techniciens de maintenance
A quelle fréquence	Les taches sont effectuées seulement lorsqu'elles sont nécessaires, à des dates prévues
Ressources nécessaires	Equipements pour les méthodes prédictives telles que la thermographie, l'analyse de vibration et autres; outils analytiques (capteurs) et GMAO
Conséquences sur le cycle de production	Un équipement est mis à l'arrêt uniquement avant une panne prédite

Figure 3 : le cadre de la maintenance prédictive (8)

- ## Les enjeux

La maintenance prédictive s'inscrit dans une logique ROIste. En effet, il faut identifier le point limite d'utilisation d'une machine pour rentabiliser au maximum son investissement. Afin d'identifier cette capacité maximale d'utilisation, il faut accepter de laisser tomber en panne des machines. Cela permet aux algorithmes de prédiction d'intégrer le point de rupture de la machine pour mieux l'anticiper par la suite.

Les enjeux économiques sont variables selon le secteur (aéronautique, ferroviaire, automobile, énergétique) mais on peut obtenir selon les études réalisées (7) :

1) 70% de réduction de pannes.
2) 30% de réduction de coûts de maintenance.
3) 50% de réduction des temps d'arrêts non planifiés.

o Le choix des types d'apprentissage automatique et stratégies associées

La maintenance prédictive repose sur des algorithmes supervisés ou non supervisés (figure 4). La principale différence est le besoin d'informations sur les pannes précédentes pour effectuer des prédictions en supervisé contrairement au non supervisé.

	Supervised Learning	Unsupervised Learning
Nécessite des données d'incidents passés	Oui	Non
Capacité à prédire de nouveau types d'incidents	Non	Oui
Difficulté de développement	Modéré	Elevé
Optimisation et maintenance des algorithme	Elevé	Bas

Figure 4 : Comparaison des principales caractéristiques des algorithmes d'apprentissage supervisé et non supervisé

La technique de modélisation choisie dépend du type de problème que l'on essaye de résoudre et des données disponibles (9). La description des stratégies suivantes m'a permis de choisir les plus pertinentes pour la résolution des objectifs clients.

4 stratégies sont possibles :

1) Créer un modèle de régression supervisé pour prédire le temps restant avant la panne (RUL en anglais). Pour cela des données statiques (propriétés mécaniques, usage moyen, environnement machine …) et historiques sont nécessaires.

2) Créer un modèle de classification supervisée pour prédire la panne dans une plage de temps sélectionnée (répond par exemple à la question : Une panne va-t-elle survenir dans 12 heures ?). Pour cela des données statiques et historiques sont nécessaires. Plusieurs types de pannes peuvent être analysés ainsi avec le même algorithme.

3) Signaler un comportement anormal avec un apprentissage non supervisé. Cette stratégie permet d'analyser plusieurs types de pannes mais ne permet pas de savoir si une dégradation de fonctionnement va conduire à une panne, ni dans combien de temps.

Stratégie utile quand les données à analyser sont en faible quantité.

4) Crée un modèle d'analyse de survie pour la prédiction de probabilité de panne dans le temps. Contrairement aux modèles précédents celui-ci n'est pas focalisé sur la prédiction mais sur le processus de dégradation. Ce modèle nécessite des données statiques et la date de survenue de panne. Ce modèle fournit des estimations pour un groupe de machines aux caractéristiques similaires.

o Le choix des modèles d'apprentissage automatique

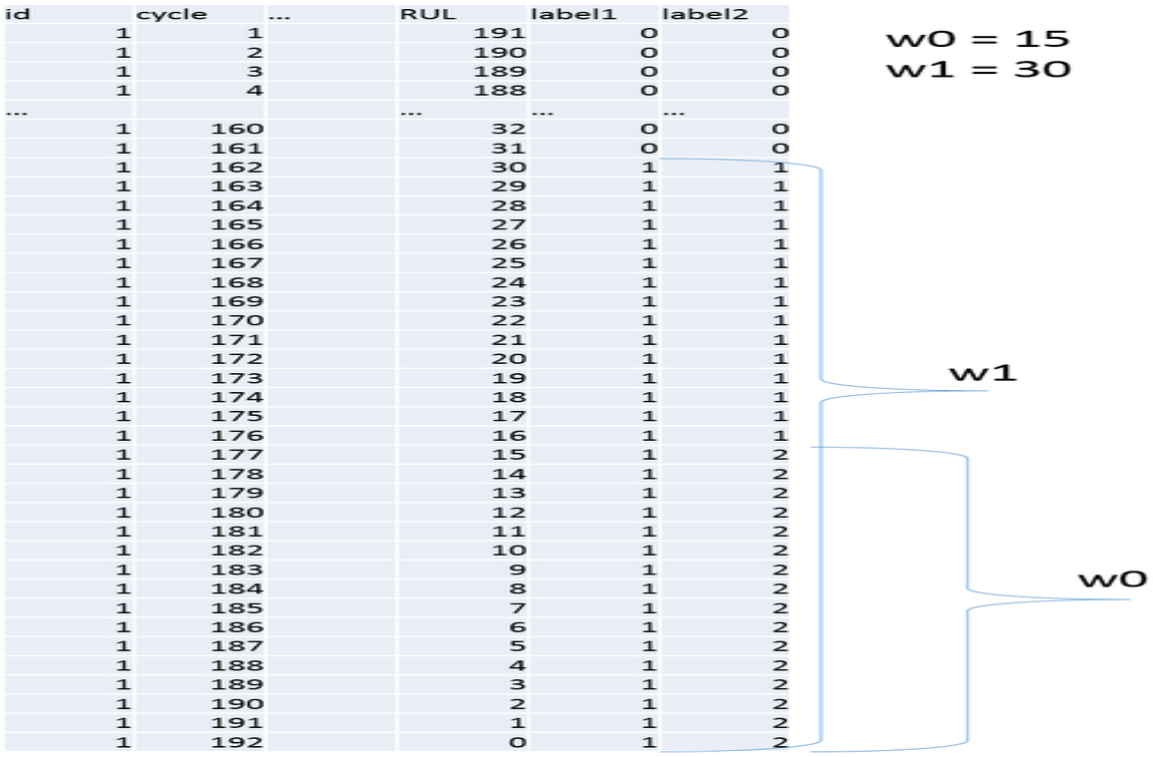
Les résultats de maintenance prédictive de moteurs d'avions à l'aide d'apprentissage automatisé ont été observés (10). En effet, dans un domaine où la précision et la performance sont nécessaires, il est intéressant de connaitre les pratiques. Je me suis donc basé sur les résultats de ce projet pour choisir mes algorithmes. Pour cet use case, plusieurs modèles de régression et classification (binaire ou multiclasses) ont été testés.

La labélisation des données d'entrainement est effectuée selon le protocole suivant (figure 5) :

1) Pour un modèle de régression la cible (target en anglais) est le RUL (Remaining Useful Life). Il reprend le nombre de cycles par ordre décroissant. Le cycle est un pas de temps associé aux enregistrements de données par capteurs. L'unité du cycle est arbitraire, non précisée dans l'use case. Elle peut être l'heure, la minute, la seconde ... Un RUL à 0 signifie « panne ».

2) Pour un modèle de classification binaire, une labélisation 0 ou 1 sur un critère défini à l'avance : une plage temporelle avant panne choisie selon les besoins du client afin de répondre à la question : la machine va-t-elle tomber en panne dans X temps ?

3) Pour un modèle de classification multiclasses, une labélisation 0 à n sur des critères définis à l'avance : des plages temporelles avant panne choisies selon les besoins du client.



id	cycle	...	RUL	label1	label2
1	1		191	0	0
1	2		190	0	0
1	3		189	0	0
1	4		188	0	0
...			...	...	...
1	160		32	0	0
1	161		31	0	0
1	162		30	1	1
1	163		29	1	1
1	164		28	1	1
1	165		27	1	1
1	166		26	1	1
1	167		25	1	1
1	168		24	1	1
1	169		23	1	1
1	170		22	1	1
1	171		21	1	1
1	172		20	1	1
1	173		19	1	1
1	174		18	1	1
1	175		17	1	1
1	176		16	1	1
1	177		15	1	2
1	178		14	1	2
1	179		13	1	2
1	180		12	1	2
1	181		11	1	2
1	182		10	1	2
1	183		9	1	2
1	184		8	1	2
1	185		7	1	2
1	186		6	1	2
1	187		5	1	2
1	188		4	1	2
1	189		3	1	2
1	190		2	1	2
1	191		1	1	2
1	192		0	1	2

1) Rul : Remaining useful life (temps restant avant la panne).
2) Cycle : pas de temps (heure, minute, seconde …).
3) Label 1 : labélisation en classification binaire.
4) Rul : Remaining useful life (temps restant avant la panne).
5) Cycle : pas de temps (heure, minute, seconde …).
6) Label 1 : labélisation en classification binaire.
7) Label 2 : Labélisation en classification multiclasses.
8) W0 et W1 : plages temporelles.

Figure 5 : labélisation des données en maintenance prédictive pour des algorithmes de classification et régression

Les résultats obtenus sont les suivants :

1) Concernant la régression, les arbres de décision sont les plus performants avec un RMSE (Root Mean Squared Error) d'environ 30 contre 86 pour les réseaux de neurones.

2) Concernant la classification binaire, les réseaux de neurones sont les plus performants avec un F1 score de 0.87 contre 0.81 pour les arbres de décision et la régression logistique.

3) Concernant la classification multiclasses, les réseaux de neurones sont plus performants avec une « accuracy » de 0.92 % contre 0.88% pour la régression logistique.

- *Le Random Forest (Forêt d'arbres décisionnels)*

D'après le paragraphe précédent, le premier algorithme choisi pour mon projet est le Random Forest. C'est une technique d'apprentissage assembliste.

Un arbre de décision est constitué de branches et de nœuds (figure 6). L'arbre est construit par partition récursive de chaque nœud. Cette partition est optimisée par homogénéité des descendants par rapport à la variable cible. La variable testée dans chaque nœud est donc celle qui maximise cette homogénéité. Le processus s'arrête quand les éléments d'un nœud ont la même valeur pour la variable cible (homogénéité) (11). C'est-à-dire quand toutes les valeurs sont « oui » ou « non » pour la variable cible dans le cas d'un problème de classification.

A chaque fois qu'un nœud terminal est atteint, une décision est prise.

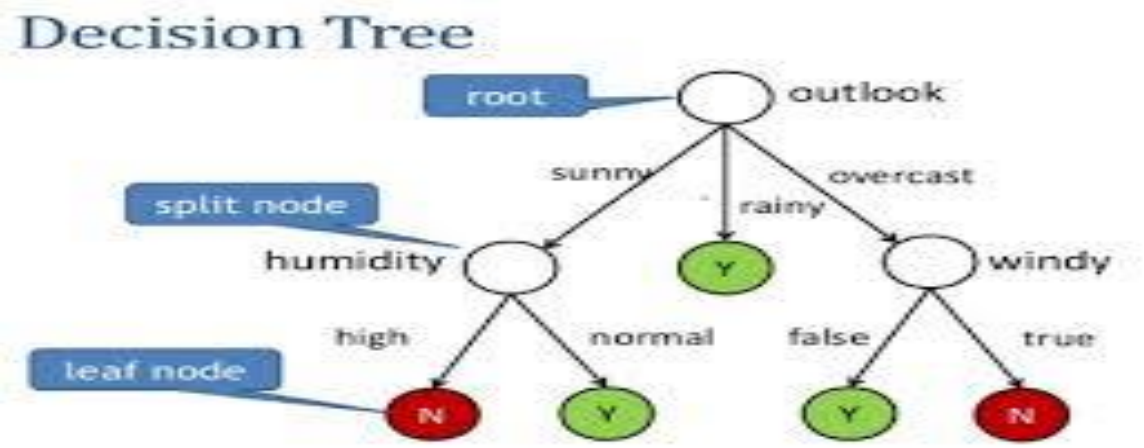


Figure 6 : structure d'un arbre de décision

Le random forest est un algorithme de choix puisqu'il fonctionne sur les deux types de problème : régression et classification.

De plus, l'algorithme a peu de chance d'overfitter (trop bien apprendre sur les données et mal généraliser). Pour éviter l'overfitting, il utilise le bagging et le pruning (taillage de certains niveaux des arbres). Le bagging quant à lui est inclus dans le fractionnement des données du Random Forest.

Il consiste à créer des sous-ensembles d'échantillons par tirage aléatoire avec remise dans l'échantillon d'origine. Ainsi, plusieurs arbres décisionnels sont créés et forment une forêt.
Une moyenne des estimations issues des différents arbres est réalisée afin de réduire la variance (12).
Le bagging est optionnel et utilisé par défaut.

Par ailleurs, le Random Forest a également un autre avantage, il a assez peu d'hyper-paramètres à régler. Ces derniers sont (13) :

1) le nombre d'arbres de décisions,
2) la profondeur de chaque arbre,
3) le nombre minimum d'échantillons pour diviser un nœud interne,
4) le nombre minimum d'échantillons présents à un nœud,
5) le nombre de variables à prendre en considération pour diviser un nœud.

- *L'XGBoost*

L'XGBoost (eXtreme Gradient Boosting) a également été choisi. En effet, c'est un algorithme d'apprentissage machiné supervisé basé sur les arbres de décision pour les problèmes de classification et régression. L'XGBoost fait donc partie de la famille des méthodes ensembliste.

Il s'entraine à plusieurs reprises après avoir segmenté le jeu de données.
XGBoost est un algorithme de choix. En effet, il utilise la méthode de boosting (ou gradient boosting) (14). Cela consiste à donner plus de poids aux prédictions difficiles à obtenir permettant au modèle de s'améliorer à chaque apprentissage. Des poids sont également donnés sur les nœuds des arbres, toujours dans un objectif de minimiser la fonction de perte de l'algorithme d'apprentissage machine.

- *LE LSTM*

Enfin, LE LSTM est le dernier algorithme choisi.
C'est un réseau de neurones particulier, un système informatique s'inspirant du fonctionnement du cerveau humain**.**
Le plus simple réseau de neurones est le perceptron. Il est caractérisé par des variables avec leur poids, une fonction de combinaison et une fonction d'activation (figure 7).

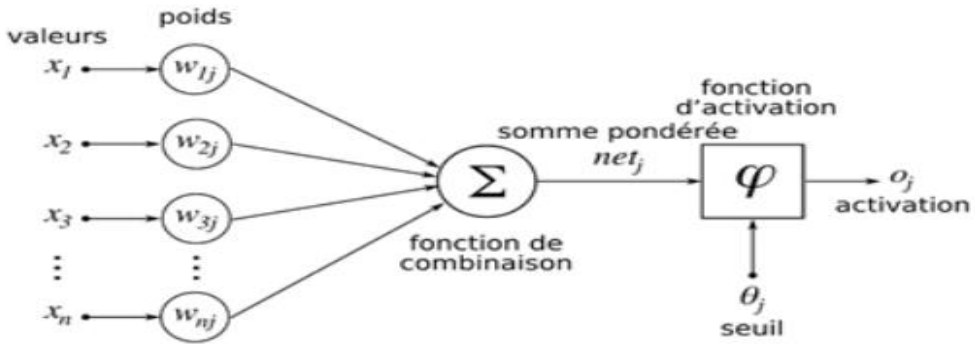


Figure 7 : structure d'un perceptron

Les réseaux de neurones contiennent trois couches (15) :

1) La couche d'entrée avec les données sous forme de tenseur.
2) Les couches cachées qui contiennent les valeurs intermédiaires calculées lors d'un entrainement du réseau.
3) La couche de sortie avec les neurones contenant les résultats d'objectifs fixés au préalable.

Ils utilisent des données d'entrainement pour « apprendre » et faire des prédictions.

Le LSTM est le réseau de neurones récurrent le plus connu. Il permet de gérer des séquences temporelles contrairement au réseau de neurones classique. En effet, sa structure introduit un mécanisme de mémoire des entrées précédentes qui persiste dans les états internes du réseau et peut ainsi impacter toute sortie future.

Un block LSTM contient (figure 8) (16) :

1) Une porte d'oubli.
2) Une porte d'entrée.
3) Une porte de sortie.

La porte d'oubli filtre les informations contenues dans la cellule mémoire précédente.

La porte d'entrée décide quelles nouvelles informations peuvent être mises en mémoire.

La porte de sortie fournit des résultats pour le bloc suivant et pour l'utilisateur en fonction des informations en mémoire et des nouvelles informations entrantes.

Les portes d'entrée utilisent des fonctions d'activation (sigmoïde, tangente hyperbolique) pour faire passer ou non les informations.

Pour effectuer une prédiction de séries temporelles, il faut introduire la chronologie en assemblant plusieurs blocks LSTM.

- Forget gate (porte d'oubli)
- Input gate (porte d'entrée)
- Output gate (porte de sortie)
- Hidden state (état caché)
- Cell state (état de la cellule)

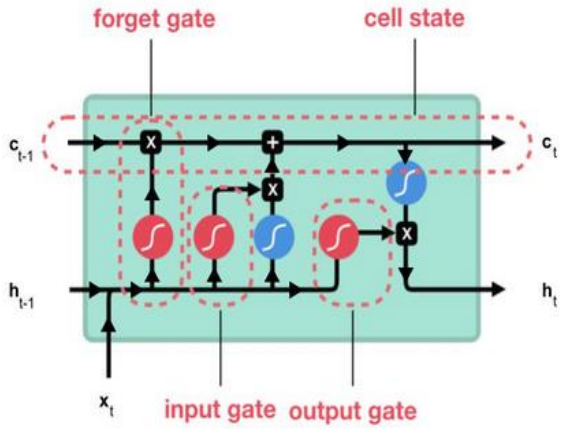


Figure 8 : Structure d'un block LSTM

Les principaux paramètres et hyper paramètres du LSTM sont les suivants :

1) le look back ou nombre de pas de temps (timestep),

2) le « batch size » ou nombre de données entrainées par étape d'entrainement,

3) le nombre d'épochs ou nombre d'entrainements de l'algorithme effectué sur les données,

4) le nombre de couches cachées et le nombre de nodes (nombre de neurones cachées) pouvant être sélectionnés par une équation (17),

5) les fonctions de perte et d'activation. La fonction de perte la plus connue est la descente de gradient.

6) Le learning rate contrôlant l'ajustement des poids des neurones du modèle par respect du gradient de perte (« loss gradient »). Le learning rate affecte la vitesse d'atteinte d'un minimum local de perte par back-propagation (18).

7) Le dropout afin d'ignorer certains neurones lors de la phase d'entrainement et prévenir l'overfitting (19).

4) LA TRADUCTION TECHNIQUE ET CHOIX TECHNIQUES DU PROJET

A partir des objectifs client et personnels j'ai établi un workflow en 5 étapes (figure 9).

Pour chaque étape j'ai effectué une traduction des spécifications fonctionnelles en spécifications techniques (démarches et procédés incluant les technologies).

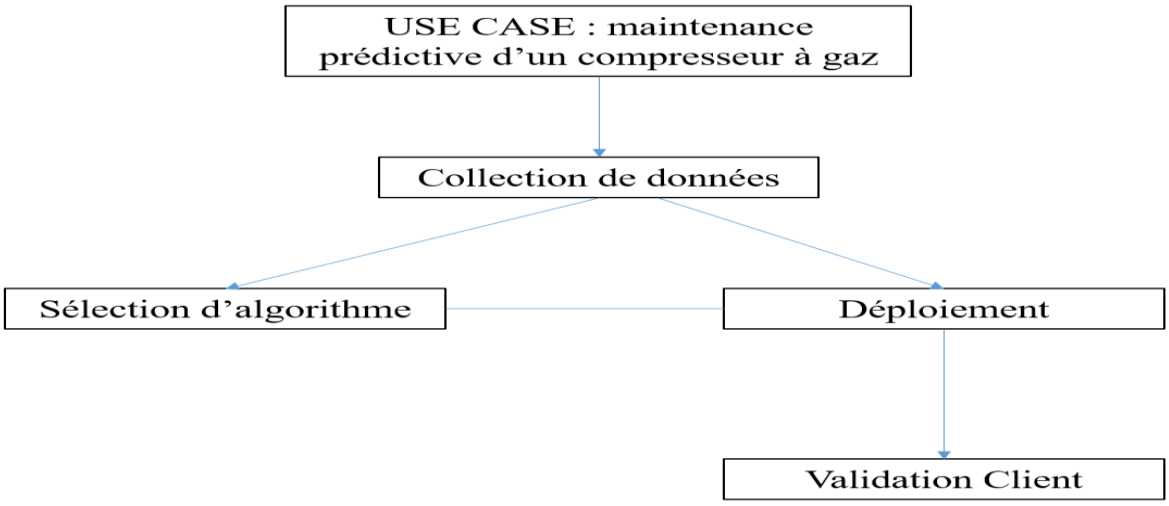


Figure 9 : Workflow du projet

Sur l'ensemble du projet, j'ai codé en Python grâce à Jupiter notebook et Visual Studio 2019. Ce dernier fournit plusieurs fonctionnalités python (interpréteur du langage, la gestion de projets en flask ou django, le switch rapide entre environnement). Python a été choisi parce qu'il est plutôt facile d'utilisation et d'apprentissage (pratiqué en formation) et qu'il possède de nombreuses librairies puissantes pour le machine learning (pandas, Scikit-learn, TensorFlow).

- **Les spécifications techniques du use case**

La première étape consiste à définir le cadre dans lequel s'inscrit la problématique client. Je l'ai l'obtenu en réalisant la veille technologique (chapitre précédent) sur le web et plusieurs réunions client. Après discussion avec le client j'ai fixé la démarche à mettre en place : une « preuve de concept » ou « Proof of conCept » (PoC) en anglais afin de valider la possibilité de pouvoir prédire des pannes de compresseur à gaz et d'identifier les variables corrélées. Puis une démarche de « preuve de valeur » ou « Proof of Value » (PoV) en anglais a été instaurée afin de répondre aux enjeux client et d'atteindre un niveau de performance de prédictions permettant de réduire les coûts de maintenance des compresseurs, d'augmenter les performances et la longévité des équipements.

- **Les spécifications techniques de la collecte de données**

La deuxième étape consiste à collecter les données afin de pouvoir les explorer et les manipuler pour fournir des résultats chiffrés permettant de répondre aux attentes du client.

o La création d'une base de données

Les données sont enregistrées par des capteurs machine et fournies par le client sous forme de fichiers csv (données historiques). J'ai donc décidé de créer une base de données (BDD) en local puis de la stocker sur un cloud dans l'objectif de présenter des données sous forme de dashboard dynamique.
Par ailleurs, les collections peuvent être utiles dans l'optique d'améliorer le projet. En effet, elles sont très utilisées dans le « Big Data » et l'IOT (Internet of Things), pour la collecte de données transportées. De plus, avec l'augmentation de la taille des données récoltées par capteurs, le cloud ou une plateforme personnelle peuvent rapidement devenir une solution de stockage. Enfin, la base de données permettrait de traiter le temps réel (20).

J'ai donc d'abord créé et rempli une base de données relationnelle SQLite à partir des fichiers csv fournis et à l'aide de la bibliothèque SQLite3 et des fonctions associées. J'ai choisi ce système de gestion afin d'obtenir une base de données relationnelle, simple d'utilisation, compatible avec python, ne nécessitant pas l'installation d'un logiciel, permettant la création de grandes BDD (maximum de 140 téraoctets), pour finaliser un projet web plutôt simple sans un accès prévu à plusieurs clients ou sans requêtes complexes.
La BDD a ensuite été convertie en MySQL à l'aide de l'outil « ESF Database Migration Toolkit » afin de pouvoir stocker un duplicat dans un « bucket » sur google cloud storage puis l'exporter dans une instance Google Cloud SQL et la connecter à Google Data Studio.
Par soucis de sécurisation des informations seul les 50000 premiers enregistrements de la base de données ont été dupliqués sur le Cloud.
Les clés étrangères de la BDD ont été définies avec MySQL Workbench. Une relation un à un a été définie entre les tables. Celle-ci est réalisée de la même manière qu'une relation un à plusieurs mais sans définition de hiérarchie parent-enfant. Le modèle Entité-Relation (ou Entité-Association) est visualisable dans la figure 10. Ce schéma traduit la logique suivante : chaque enregistrement est caractérisé par une date et une valeur (numérique ou catégorielle) relevée par les capteurs.

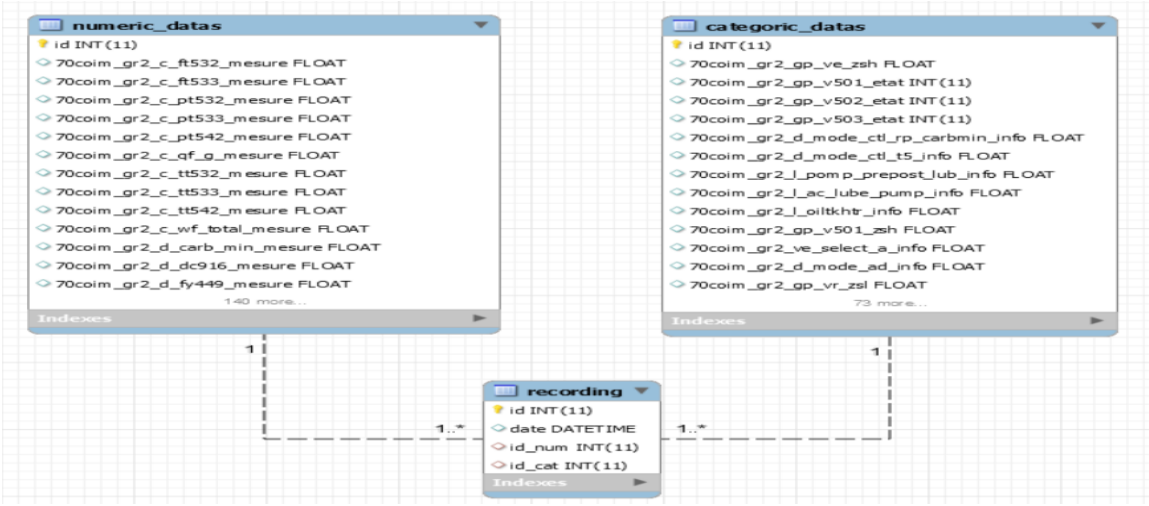


Figure 10 : Schéma Entité-Relation des données du compresseur à gaz obtenu avec MySQL Workbench

o L'exploration, le nettoyage et le prétraitement des données

Les données brutes issues de fichiers CSV stockés en local (239 variables : 153 numériques et 86 booléens) sont d'abord importées dans un dataframe sur python grâce à la bibliothèque Pandas. Un algorithme est créé puisque toutes les données Excel ne sont pas au même format et propices à l'importation. Celui-ci est basé sur des fonctions python utilisées sur des dataframes.

Les données pré nettoyées par le client sont ensuite explorées.
L'objectif est de visualiser les plages temporelles de marche-arrêt et les survenues historiques de pannes en utilisant la variable de pression du compresseur identifiée par le client. Pour cela la bibliothèque Matplotlib en python a été utilisée afin d'obtenir une courbe de la mesure d'intérêt (figure 11). A l'arrêt, la pression d'un compresseur devrait rester constante or ici elle diminue jusqu'à 0 bar (panne).

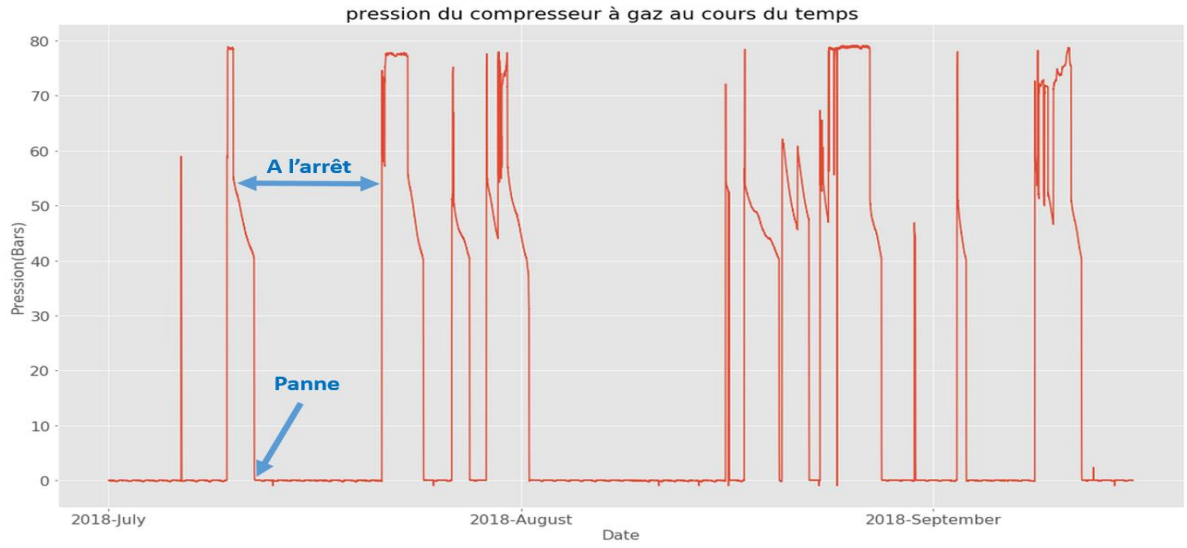


Figure 11 : Visualisation de la variable PT542 (pression du compresseur) au cours du temps

L'obtention de ces informations a permis de découper les données afin d'obtenir plusieurs jeux disponibles pour le machine learning (jeux de données avec ou sans plages temporelles de fonctionnement du compresseur à gaz).
L'obtention de ces informations a également permis de labéliser les données de façons spécifiques et différentes pour des problèmes de classification et régression en maintenance prédictive (voir le chapitre sur « Le choix des modèles d'apprentissage automatique »).
Ces éclaircissements ont également permis de mettre en place une remodélisation automatique des fichiers csv en dataframe contenant seulement les plages d'arrêt machine jusqu'à la dépressurisation du compresseur à gaz. Pour cela un algorithme avec différentes fonctions numpy et pandas a été créé. La création de cette automatisation permettrait la récolte de données en temps réel dans le futur.

Un descriptif statistique des données a également été réalisé.

Les données présentes dans un dataframe ont ensuite été nettoyées. En effet, celles-ci contenaient beaucoup de NaNs.

Dans un premier temps les variables contenant plus de 50 % de NaNs ont été supprimées ou les enregistrements avec des valeurs manquantes pour ces variables ont été effacés.

Les enregistrements liés aux valeurs manquantes des variables avec moins de 5% de NaNs ont également été supprimés. Ces seuils ont été déterminés après une étude des pratiques générales réalisées en machine learning.

La bibliothèque python Missingno a permis de visualiser les variables contenant de 5 à 50% de NaNs (figure 12).

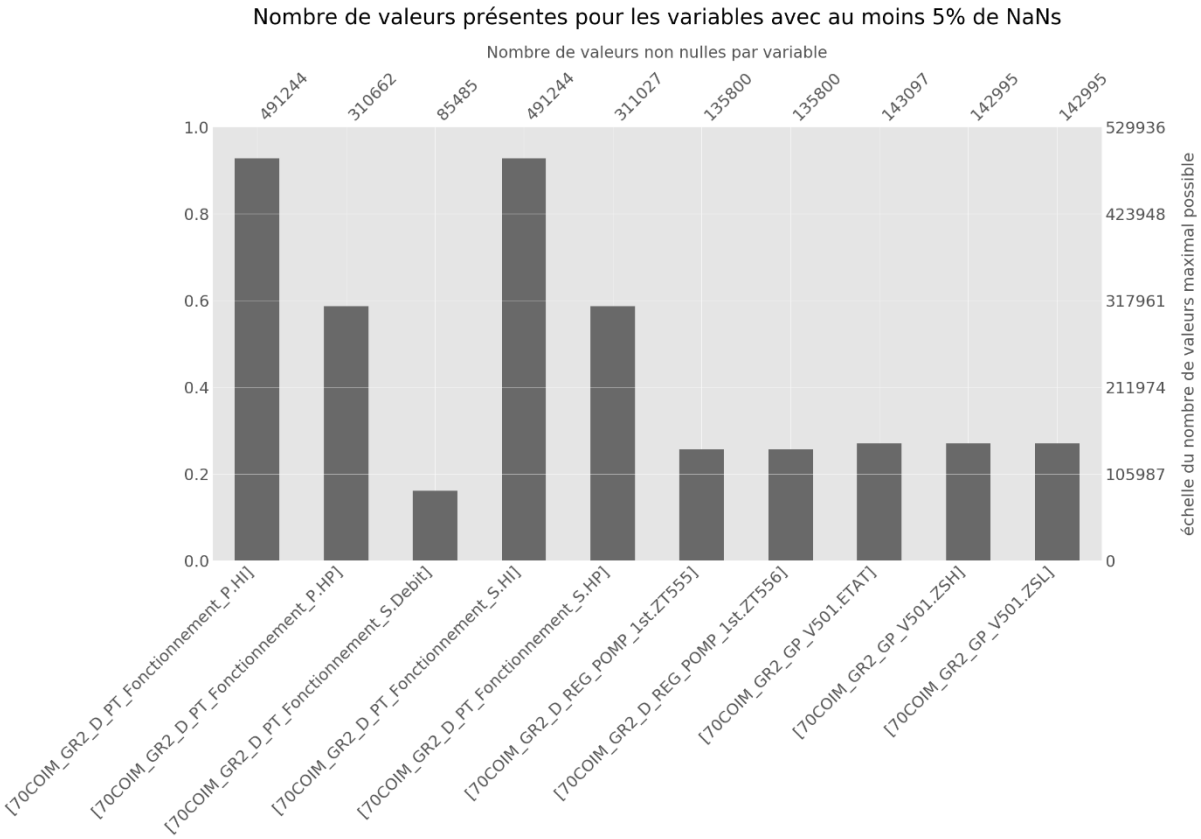


Figure 12 : Nombre de valeurs présentes pour les variables avec au moins 5% de NaNs

Certaines de ces variables ont été supprimées, des enregistrements du Dataframe ont été effacés ou bien les valeurs manquantes ont été remplacées (remplacement par la dernière ou la prochaine valeur non nulle ou bien avec la moyenne des valeurs de la variable ou encore par interpolation linéaire avec les bibliothèques python fillna et interpolate). Le choix final des méthodes et techniques s'est basé sur les résultats de la qualité de prédictions obtenus par machine learning.

La matrice de corrélation de nullité Missigno a permis de visualiser des corrélations entre variables (figure 13). Cette corrélation mesure l'effet de l'impact de la présence ou non d'une valeur d'une variable sur l'apparition de valeurs d'une autre variable. Les variables avec une corrélation proche de 1 ont donc subi un même traitement sur les NaNs. A l'inverse, les variables avec une corrélation proche de 0 ont donc subi un traitement individualisé avec les méthodes et techniques citées précédemment.

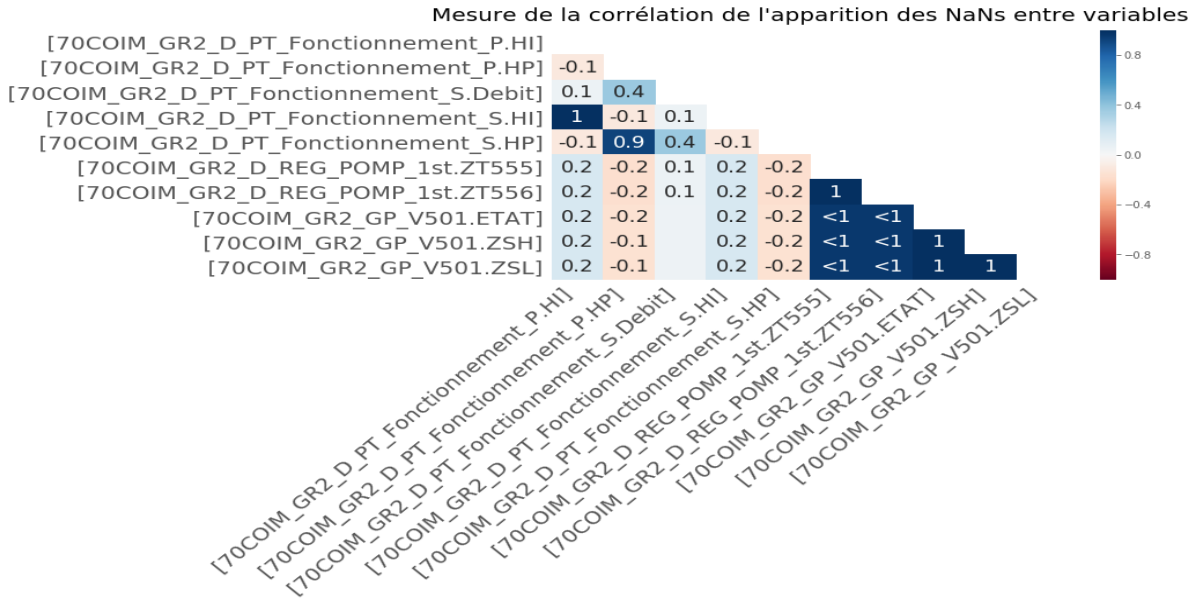


Figure 13 : Matrice de corrélation Missingno mesurant la corrélation d'apparition des NaNs entre variables

Des variables ont ensuite été créées. Travaillant sur des séries temporelles (considération possible, voir « axes d'améliorations »), une colonne « Id » a été créée. Elle s'incrémente à chaque nouvelle plage temporelle arrêt machine-panne. Cette colonne est nécessaire pour intégrer la cyclicité du fonctionnement d'un compresseur et ainsi compléter la préparation de séries temporelles avant utilisation par le LSTM.

D'autres variables combinées ont été fabriquées : la variance et la moyenne mobiles des données brutes. Celles-ci pourraient permettre d'améliorer les prédictions du LSTM en facilitant l'identification de cyclicité ou une tendance d'évolution possible de séries temporelles.

Enfin, tous les traitements sur les données citées au-dessus ont permis d'obtenir des jeux de données différents et de comparer leur apport dans les performances de modèles de machine et deep learning.

- **Les spécifications techniques de la sélection et utilisation d'algorithmes de machine et deep learning**

La troisième étape consiste à sélectionner les algorithmes de prédiction de pannes. Pour cela je me suis basé sur les résultats des métriques d'évaluation obtenus par plusieurs algorithmes utilisés en maintenance prédictive (voir le chapitre « Le choix des modèles d'apprentissage automatique »). Ainsi j'ai sélectionné la forêt d'arbres décisionnels (random forest en anglais) et le Long Short-Term Memory

(LSTM) en classification et régression supervisées. En effet deux types de problèmes sont identifiés par le deuxième objectif client et doivent donc faire intervenir différents procédés. Par ailleurs le LSTM est utile pour faire des prédictions de séries temporelles.
Le Random forest quant à lui est pratique avec un grand nombre de variables explicatives et utilise le bagging ou *(*Bootstrap Aggregation) (12). Cette caractéristique est associée au dilemme biais-variance[1].
La performance des prédictions avec le random forest a été comparée aux résultats obtenus avec l'algorithme XGBoost (eXtreme Gradient Boosting, algorithme d'arbres de décision) fourni par la bibliothèque xgboost de python. Ces deux derniers algorithmes ne sont pas configurés initialement pour faire des prédictions sur des séries temporelles. Cependant, leur choix est justifié puisque les séries temporelles supportent très mal les observations manquantes pour établir des prédictions « accurates ».
Trois algorithmes ont donc été utilisés pour comparés les résultats obtenus.

A titre indicatif, afin d'identifier les temps de survenues de surpression pendant le fonctionnement du compresseur, un algorithme a été créé. Celui-ci est basé sur la moyenne et la variance de la variable de pression du compresseur. Pour identifier, les oscillations anormales, un seuil a été réglé à 3.

○ La préparation des données

Tout d'abord, les variables nombreuses ont été sélectionnées par « feature selection » afin de possiblement améliorer les performances de l'algorithme de prédiction. Cette méthode se base entre autre sur les retours métiers. Ici, le client a fourni les 7 variables principales associées à la dépressurisation et un visuel a été établi sur chacune d'elle par l'élaboration de courbes avec Matplotlib.
Des méthodes de filtration ont été utilisées avec des bibliothèques python comme chi2, mutual_info_classif, mutual_info_regression et SelectKBest de sklearn.
La réduction de la dimensionnalité des données a également été réalisée par le module PCA (Principal component analysis) de sklearn.

Puis les données issues de « feature selection » ou pas ont été normalisées avec la fonction MinMaxScaler de sklearn. En effet, cette dernière permet de mettre toutes les données à la même échelle, sur un intervalle [0,1] et donc de réduire l'effet des valeurs aberrantes, « outliers » en anglais (21).

Ensuite, le train/test split en classification a été réalisé. Deux méthodes basées sur la sélection de données avec ou sans les phases de marche du compresseur ont été mises en place. Les plages

[1] Dilemme biais-variance, aussi appelé compromis biais-variance, son objectif est de minimiser les erreurs des algorithmes d'apprentissage.
Le biais mesure la distance entre les valeurs prédites et les valeurs exactes, c'est l'erreur insérée dans l'algorithme par manque de relation pertinentes entre données notamment.
La variance mesure la dispersion des valeurs prédites autour de la valeur exacte, elle représente la sensibilité aux échantillons d'apprentissages.

temporaires consécutives avec ou sans arrêt machine contenant 80 % du nombre total de pannes du dataframe d'intérêt ont été sélectionnées pour le train et le reste pour le test. Ce choix s'est basé sur le principe de Pareto[2].
Le train/test split en régression a été fabriqué différemment. Les plages temporaires avec ou sans arrêt machine contenant toutes les observations précédant un temps d'intérêt pour le test se retrouvent dans le train. Le test ne contient donc que quelques observations relatives au temps choisi pour faire la prédiction.
Dans la bibliographie, le train/test est unique et correspond à celui réalisé en régression (22). Pour éviter d'obtenir un test avec peu de prédictions en classification et donc avoir un doute sur la fiabilité des résultats des métriques d'évaluation j'ai changé le protocole.

Enfin, la mise en forme des données pour le LSTM a été effectuée. Un algorithme permettant un look back de 30 sur les données de train et de test a été créé et utilisé.
La matrice 3D des données [échantillon, pas de temps, nombre de variables] a ainsi été obtenue (23). La valeur du look back a été choisi afin de ne surtout pas dépasser la plage temporelle (30min) de données nécessaire pour entrainer un LSTM en régression permettant de répondre à l'objectif client d'une prédiction 30 minutes après arrêt.
L'échantillon correspond aux nombres de mesures par pas de temps.
Pour éviter un problème de mémoire l'algorithme a inclus un générateur « yield ». La fonction génératrice permet d'effectuer en boucle la mise en forme des données [échantillon, pas de temps, nombre de variables] pour chaque plage temporelle début_enregistrement-panne_machine.
La préparation des données d'entrée et le paramétrage du modèle LSTM a suivi en très grande partie un github de référence (22).

Les données de la variable cible en classification ont toujours été balancées. Un Upsampling (entrainement) a toutefois été réalisé en utilisant le module Smotenc de la bibliothèque imblearn.

o Les paramétrages du Random Forest et du LSTM

Les hyperparamètres du Random forest et du LSTM ont été optimisés. Concernant le random forest, la validation croisée[3] et le module GridSearchCV de la bibliothèque python sklearn ont été utilisés pour régler automatiquement les hyper paramètres. Lors de cette manipulation, l'overfitting a été évité par réglage du paramètre cross-validation(cv) à 3 (24). Le meilleur modèle a été sauvegardé en local avec la fonction dump du module joblib de python.

[2] Principe de Pareto, aussi appelé loi de Pareto ou encore loi des 80-20, phénomène empirique constaté dans certains domaines : environ 80% des effets sont le produit de 20% des causes.
[3] Validation croisée, utilisée en apprentissage automatique, méthode d'estimation de fiabilité d'un modèle fondé sur une technique d'échantillonnage. Méthode permettant d'évaluer la capacité de généralisation d'un modèle. Méthode basée sur la division du jeu de données en plusieurs parties avec itérations successives d'apprentissage automatique. Le jeu de données test est différent à chaque itération.

Concernant le LSTM, le choix des paramètres du réseau de neurones comme le nombre d'épochs, le nombre de couches, le nombre de neurones et les fonctions d'activations a été effectué à l'aide du git hub de référence avec cependant quelques différences (figure 14). La fonction sequential de keras a été utilisée afin d'ajouter toutes les couches au modèle. Le batch_size a été chois afin de ne pas trop ralentir les temps de calcul. Le callback a été utilisé afin de pouvoir obtenir une vue interne du modèle durant l'entrainement.

```
# design network
model = Sequential()
model.add(LSTM(100, return_sequences = True, input_shape = (X_train.shape[1], X_train.shape[2]))) #input_shape=(t
model.add(Dropout(0.2))
model.add(LSTM(50, return_sequences = False))
model.add(Dropout(0.2))
#model.add(Dense(y.shape[1]))

model.add(Dense(1)) # number of output = 1.
model.add(Activation('linear'))
model.compile(loss="mean_squared_error", optimizer="adam")
model.summary()
# fit network
history = model.fit(X_train, y_train, epochs=150, batch_size=200, validation_data=(X_test, y_test), verbose=1, sh
                    callbacks = [keras.callbacks.EarlyStopping(monitor='val_loss', min_delta=0, patience=10, verbo
```

Figure 14 : Paramétrage du LSTM en régression

Le nombre d'épochs a été défini afin d'éviter l'overfitting. Pour cela les courbes historiques de la fonction de perte lors de l'entrainement et de la validation ont été visualisées avec matplotlib et grâce au callback (figure 15). En ajoutant un epoch supplémentaire, les courbes de training et validation se croisent permettant de remarquer un overfitting. De manière générale des tests empiriques sur les hyper paramètres ont été effectués sur le LSTM, le Random Forest et l'XGBoost.

Figure 15 : Evolution de la parte de gradient en fonction du nombre d'epochs pour l'entrainement et la validation d'un LSTM

o L’évaluation des algorithmes de prédiction

Pour comparer les différents modèles mis en place différentes métriques de scoring ont été utilisées.
Pour les algorithmes de classification, la matrice de confusion, la précision et le recall ont été obtenus (figure 16). Ces métriques sont intéressantes pour évaluer un modèle qui retourne des variables binaires. Le rappel (recall) ou sensibilité est le taux de vrais positifs et doit être observé avec la précision (proportion de prédictions correctes) pour évaluer efficacement le modèle. En effet, des prédictions binaires déséquilibrées conduisent à surestimer la valeur d’une des deux métriques et en complément à sous-estimer la valeur de la métrique associée (25). Pour évaluer un compromis entre rappel et précision, j’ai calculé le F1 score aussi appelé F-measure (figure 16).
Pour les algorithmes de régression, la métrique utilisée est la racine carrée de l’écart quadratique moyen (RMSE en anglais). Le RMSE comprend à la fois la variance et le biais de l'estimateur. Sa formule est reportée sur la figure 16. Ce dernier a été préféré au MSE afin de ramener la métrique à l’échelle des prédictions effectuées.

Precision	$\frac{TP}{TP+FP}$
Rappel	$\frac{TP}{TP+FN}$
F-measure	$\frac{2.precision.rappel}{precision+rappel}$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

- TP = True Positive, FP = False Positive, FN = False negative.
- Predicted = valeur prédicte, Acutal = vraie valeur, N = nombre de prédictions.

Figure 16 : métriques d’évaluation utilisées avec les algorithmes de classification et régression

o La sélection de features

Comme dans l’un des chapitres précédents la sélection de features a été mise en place, cette fois, par les méthodes « wrapper » et « embedded » (26) et l’utilisation des modules python feature_importances et SelectFromModel. Cette sélection est basée sur les résultats obtenus par les algorithmes d’apprentissage machine à partir de combinaisons avec toutes les variables.

L'intérêt de ces techniques ici est double : améliorer potentiellement les performances de l'algorithme de prédiction et connaitre les variables les plus corrélées à la panne (dépressurisation), aux oscillations anormales de pression afin de répondre aux objectifs clients.
Celles-ci sont plus « accurate » que la méthode par filtration mais prennent plus de temps de calcul et sont donc plus adaptées aux prédictions à partir d'un nombre limité de variables explicatives.
La matrice de corrélation (heatmap) voulue par le client et mesurant la corrélation entre les variables sélectionnées a été créée grâce à la bibliothèque Seaborn. Celle-ci est présentée dans le chapitre suivant : « Le bilan de projet, l'atteinte des objectifs ».

o Les spécifications techniques du déploiement

La 4ème étape consiste au déploiement de bases de données et/ou d'algorithmes de machine learning. Je ne travaille pas sur un projet de BDD, la seule chose que j'ai déployée c'est l'algorithme de machine learning afin de passer du développement à la production et fournir un webservice au client. Pour cela j'ai utilisé streamlit, un framework de machine learning permettant de créer une application web. J'ai ainsi pu présenter les différents résultats permettant de répondre aux objectifs client. Ensuite, j'ai utilisé docker sous linux pour créer un container pour une application scalable et portable (libérée des contraintes de systèmes d'exploitation).
Puis, j'ai utilisé Heroku pour déployer mon application et permettre au client de l'utiliser en toute autonomie.
Flask a aussi été utilisé afin de pouvoir présenter les résultats au client par un lien web.

o Les spécifications techniques de la validation client

La dernière étape consiste à faire valider l'application par le client (Proof of Validation).
Cela a été fait par mesure des métriques des performances des modèles de machine learning utilisées. Ces métriques établissent la fiabilité des prédictions dans un contexte donné.
L'évaluation clé a été de mesurer la fiabilité du modèle Random Forest Regression avec de nouvelles données client. Le client a fourni 3 heures de données d'arrêts machines pour prédire les temps de dépressurisation.

5) LA GESTION DE PROJET

Deux étapes distinctes successives ont eu lieu dans la gestion de projet. La première s'est déroulée en entreprise afin d'établir le POV et la deuxième s'est déroulée en formation et sur mon temps libre pour enrichir le projet.

• La gestion de projet en entreprise

La première étape a été la prise de connaissance du projet par communication email et par partage de fichiers des données sur SharePoint. Dans la foulée, lors d'un déplacement sur le site de Spie Industrie, de nouvelles informations nous ont été communiquées :

1) L'environnement dans lequel s'intègre le projet : plateforme PAAS (Plate-forme en tant que service) avec un logiciel d'analyse de données et prédictions.
2) La présentation d'un compresseur à gaz.
3) La définition des objectifs.
4) La définition d'étapes et les deadlines à respecter.
5) La présentation des données (légèrement technique).

Sur ces informations j'ai réalisé un compte rendu sous forme de document word.

L'objectif prioritaire du client étant de pouvoir effectuer des prédictions sur les dates d'arrivées de dépressurisation en phase d'arrêt de la machine et d'obtenir des pistes sur la cause de celle-ci.
J'ai donc dans un premier temps défini précisément un plan d'action en 3 phases pour répondre à cet objectif (figure 17). Ce dernier commence par la récupération de données, puis le nettoyage et l'analyse de données et enfin l'utilisation d'algorithmes de prédiction.

Tâches	Charge de travail (en j)
Récupération des données : - Création d'un algorithme permettant de récupérer automatiquement les CSV stockés en locale	4
Préparation et analyse des données : - Mise au même format de différents dataframes pour concaténation en un dataframe unique - Exploration du jeu de données (courbes des principales variables) - Identification et obtention de la target : remaining useful life (RUL) à partir des dates de jours de panne. Création d'un dataframe de la forme de la figure 5 - Sélection de variables d'intérêt (feature selection, suppression des variables avec trop de NaNs, remplacement des NaNs) - Création des colonnes variance et moyenne mobiles - Obtention d'un jeu de données équilibré	6 5 6 7 4 5
Préparation et utilisation du machine learning : - Découpage des données : train/validation/test - Création et évaluation de modèles de machine et deep learning (LSTM, Random Forest) - Elaborer des représentations graphiques sur les résultats importants d'analyse et les prédictions	6 20 7

Figure 17 : plan d'action initial pour répondre aux objectifs prioritaires du client.

Ce plan a été validé par mon tuteur et j'ai travaillé dessus pendant deux mois.

Le travail a été réalisé en agilité. En effet, le Kanban a été utilisé pour suivre les objectifs fixés, le daily meeting pour communiquer sur l'avancée du projet et discuter avec d'autres développeurs sur la veille technologique, les obstacles et les choix techniques envisagés.

Un compte rendu fréquent de mon travail était fourni à mon tuteur. Le projet et les VT ont été partagés à l'équipe sur GitLab et Teams.
Des sprints avec un collègue de travail ont également été faits sur les éléments du plan d'action défini précédemment.

Une présentation pdf du travail réalisé et des résultats obtenus a ensuite été effectuée auprès du client(annexes).

Dans la foulée, une collaboration avec un développeur, un graphiste et moi-même a eu lieu pour la réalisation d'une maquette d'interface graphique et sa mise à disposition sur internet. Cette première version a été réalisée par un collègue de travail et non par moi-même à cause d'un impératif de temps.

Un mois plus tard le client a fourni de nouvelles données remontées par les capteurs afin de tester et valider l'algorithme mis en place.

Pour répondre aux objectifs du client relatifs aux oscillations anormales de pression de compresseur en marche j'ai fait un plan d'action. Celui-ci a été validé par mon tuteur puis j'ai bien avancé le travail. Le rendu a été partagé lors d'une discussion via teams avec le client. Le travail non prioritaire n'a pas fait l'objet d'une évaluation ou de partage de connaissances supplémentaires.

- # La gestion de projet en formation et sur le temps personnel

Afin d'enrichir le projet avec les nouvelles compétences obtenues en formation j'ai réalisé des sprints sur une semaine.
Les priorités ont été définies pour être en adéquation avec le référentiel de compétences de la certification, le workflow préalablement défini (schéma 2) et aussi pour s'intégrer avec de futures demandes du client dans le projet actuel concernant l'IOT et la maintenance prédictive.
Les sprints réalisés avaient pour contenu les éléments suivants :

1) Pour le premier, la mise en place d'une BDD et la création d'un web service.
2) Pour le deuxième, la fondation d'un bucket de stockage de la BDD et l'élaboration d'un dashboard.
3) Pour le troisième la génération d'un container et le déploiement de l'application.

Les prévalences ont été déterminées après discussions avec le formateur. Les résultats de sprints et les éléments du projet en général ont été partagés avec le formateur sur GitHub.

- **Retours sur les outils et techniques de gestion de projet**

La méthodologie utilisée dans ma gestion de projet est la méthode agile, elle a été associée à plusieurs outils et techniques :

1) Le visuel Kanban (Trello, Klaxoon, GitLab boards)
2) Le sprint
3) Le daily meeting
4) GitHub/GitLab
5) Le pack office 365

La méthodologie utilisée a eu plusieurs avantages :

1) L'adaptation aux changements d'objectifs décidés par le client. En effet, celui-ci a décidé après retour des premiers résultats de se focaliser sur une prédiction de panne à un temps précis avec un algorithme de régression plutôt que sur la prédiction d'une panne dans une fenêtre temporelle.
2) La concentration sur un seul projet sur une plage de temps donnée avec l'utilisation de sprints.
3) Le partage sur les avancées, les difficultés rencontrées avec les daily meeting et les sprints pour permettre un travail collaboratif et obtenir de nouvelles pistes de travail.
4) La gestion de l'avancement des tâches, du nombre de tâches en cours avec les outils visuels Trello, Klaxoon, les boards de GitLab (méthode Kanban) et la remontée plus facile d'alertes si nécessaire.
5) La détection d'évolutions nécessaires du travail déjà réalisé par la revue de sprints en interne pendant l'alternance. Celles-ci par exemple ont permis la mise en place de l'automatisation d'utilisation de données par algorithmes IA après discussion entre les membres de l'équipe sur le projet partagé sur GitLab et GitHub.
6) La détermination de nouveaux objectifs et leur concrétisation par des sprints en formation.
7) Une augmentation de mes compétences en science des données et machine learning par une alternance entre VT et travaux pratiques.

La méthodologie utilisée a eu un inconvénient principal :

1) Le manque de flexibilité dans les deadlines.

Le pack office 365 a été largement utilisé avec Outlook, OneDrive, Word, Excel, Powerpoint, Sharepoint, Teams et Yammer.

Son utilisation a eu pour avantage :

1) D'être performant dans le domaine des Technologies de l'Information et de la Communication (TIC) et donc de faciliter le travail collaboratif.

En effet, il a facilité l'accès aux sources d'informations, au stockage et la manipulation de données, à la production et à la transmission d'informations sous différentes formes.

6) LE BILAN DE PROJET

- **L'atteinte des objectifs**

Le client avait deux objectifs principaux.

Le premier objectif principal a été atteint. En effet, les prédictions de panne 30 minutes après l'arrêt du compresseur ont été obtenues avec le LSTM et le Random Forest. Elles sont satisfaisantes pour le client. Avec le LSTM, des valeurs quasiment uniques de prédictions sont toujours obtenues et le RMSE est très élevé. Avec le Random Forest, le RMSE est de 203. Ce dernier modèle obtient donc de bien meilleurs résultats, ces derniers sont présentés ici (figure 18).

La procédure pour atteindre ce 1er objectif est la suivante :

1) Le client a fourni un jeu de données de test contenant les données du même compresseur étudié mais postérieures à celles ayant servi à entrainer mon modèle.

2) Les données ont été relevées sur les 3 premières heures d'arrêt machine.

3) L'algorithme de Random Forest est le plus performant seulement avec les données numériques reliées au temps d'arrêt du compresseur pour son entrainement.

4) L'algorithme de Random Forest a été utilisé pour prédire la date de panne à plus ou moins 3h20 près (RMSE = 200) (figure 19).

5) Un webservice fournit les informations relatives à la prédiction de pannes.

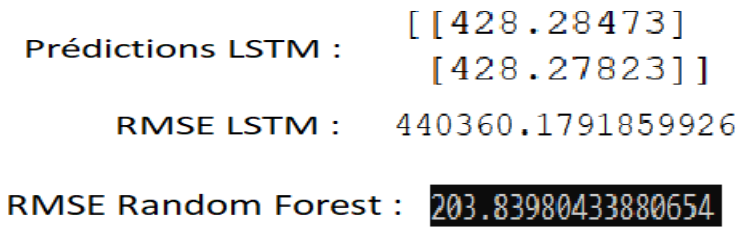


Figure 18 : RMSE obtenus sur différentes prédictions avec le LSTM et le Random Forest en régression 30 minutes après l'arrêt du compresseur

Identifiant évènement	date prévisionnelle : défaut de dépressurisation	fiabilité	variables significatives associées par ordre décroissant d'importance	dates réelles	delta
1	08/01/2019 12:24	RMSE : 200min.	PT_532 PT_542 PT_533 PT_544	08/01/2019 13:03	40 min
2	11/01/2019 00:42			10/01/2019 21:46	3h
3	12/01/2019 21:54			12/01/2019 19:27	2h30
4	20/01/2019 23:47			20/01/2019 13:38	10h
5	26/01/2019 03:09			25/01/2019 23:55	3h10
6	07/02/2019 05:59			07/02/2019 00:31	4h30
7	28/02/2019 07:38			28/02/2019 04:20	3h10

Figure 19 : Prédiction des dates de pannes du compresseur avec le Random Forest en régression

Le deuxième objectif principal a été atteint. En effet, les 6 variables les plus corrélées à la dépressurisation (non corrélées entre elles) ont été relevées grâce à une méthode de sélection de données associée au Random Forest (figure 20) et une visualisation par heatmap (figure 21).

Ainsi on remarque que parmi les variables les plus corrélées à la dépressurisation, la matrice de corrélation ne reprend que certaines d'entre elles après réduction. Le maximum de coefficient de corrélation de Pearson[4] fixé entre deux variables est de 0.81.

D'après la documentation client, des correspondances entre les variables corrélées et leurs mesures associées ont pu être établis :

1) 70COIM_GR2_E_PIT514.MESURE surveille les joints d'étanchéité à gaz,
2) 70COIM_GR2_D_NGP_SP. MESURE surveille la consigne préétablie de débit de carburant à amener à la chambre de combustion,
3) 70COIM_GR2_E_PDIT511.MESURE surveille l'étanchéité des filtres à air,
4) 70COIM_GR2_E_FIT514.MESURE surveille les joints d'étanchéité à gaz,
5) 70COIM_GR2_C_Wf_Total.MESURE surveille le débit de combustible,
6) 70COIM_GR2_C_PT532.MESURE surveille la pression d'aspiration d'air.

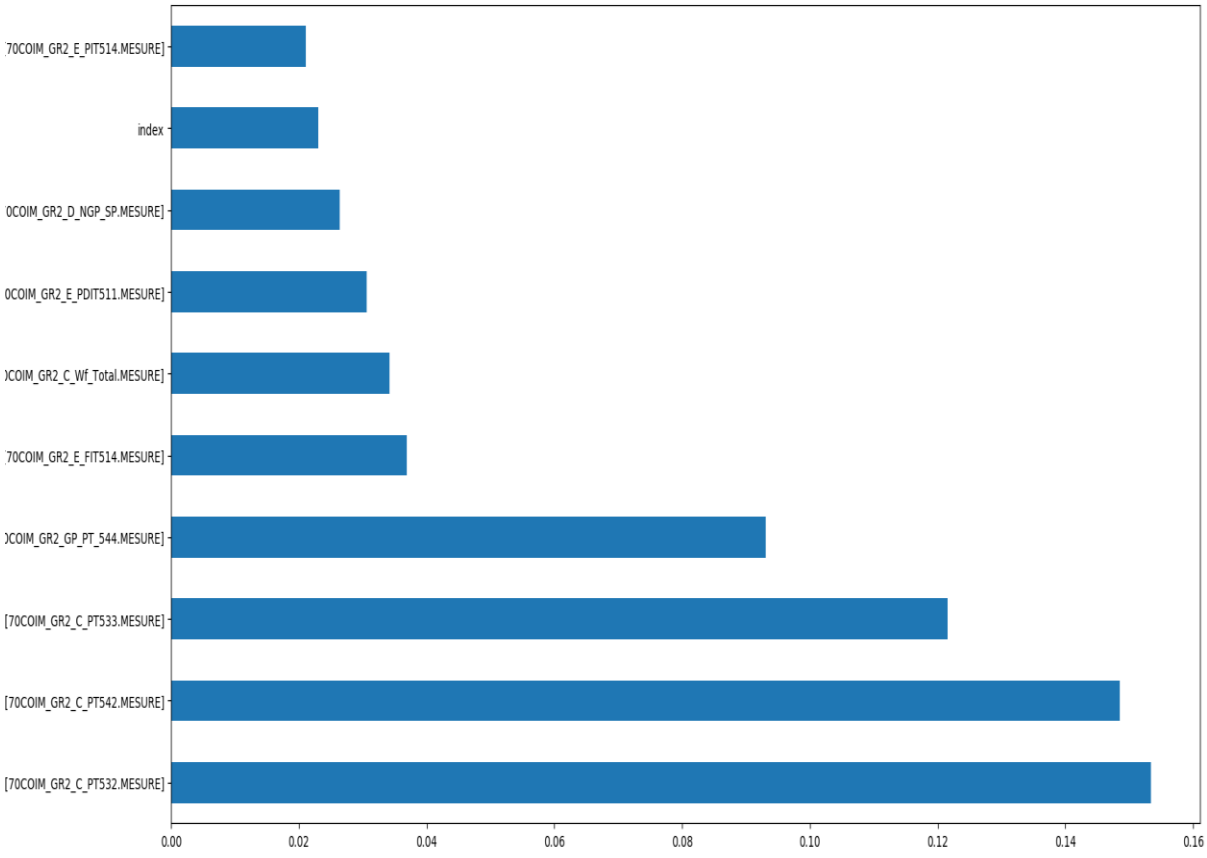


Figure 20 : Sélection des variables (features selection) les plus impactantes (%) dans les prédictions par le Random Forest

[4] Le coefficient de corrélation de Pearson, également appelé r de Pearson, est une statistique qui mesure la corrélation linéaire entre deux variables X et Y.

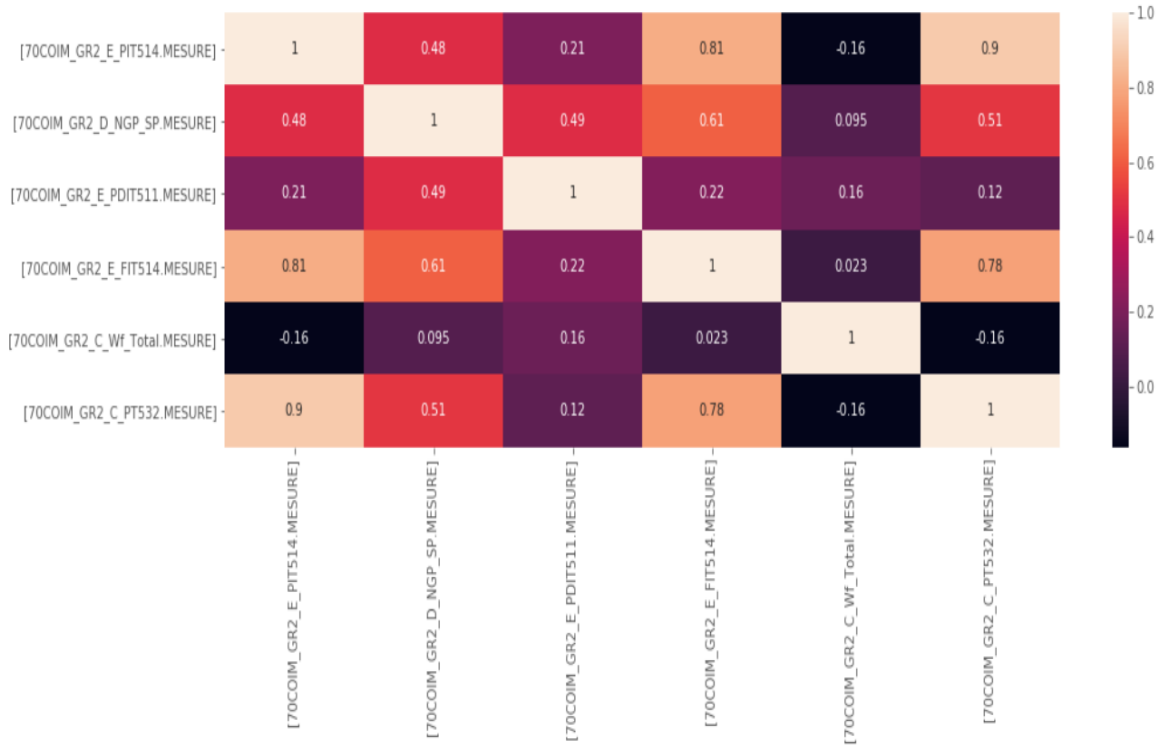


Figure 21 : matrice de corrélation des variables les plus corrélées à la variable de dépressurisation mais peu corrélées entre elles

Les autres objectifs client et personnels ont été atteints :

1) les prédictions de panne 12 heures à l'avance ont été obtenues avec le LSTM et le Random Forest (figure 22). Le LSTM propose une précision de 78% et un recall de 100 % alors que le Random Forest propose une précision de 95% et un recall de 88%. Ce dernier modèle obtient de meilleurs résultats.

LSTM :

```
[[ 352  400]
 [   0 1442]]
precision =  0.7828447339847991
 recall =  1.0
```

Random Forest :

```
[[ 687   65]
 [ 169 1273]]
precision =  0.9514200298953662
 recall =  0.8828016643550625
```

Figure 22 : Matrice de confusion, précision et recall obtenus sur différentes prédictions avec le LSTM et le Random Forest en classification 12 heures à l'avance

2) les résultats obtenus ont été présentés sous forme de dashboard via internet, (fait par collègue de travail),

3) Aucune pression anormale (supérieures à 10 bar) en phase de fonctionnement du compresseur n'a été détectée après visualisation des données et création d'un algorithme,

4) une base de données a été créée,

5) des algorithmes d'intégration de données en temps réel ont été créés,

6) un webservice a été créé et déployé après stockage d'un algorithme IA dans un container,

7) une base de donnée a été stockée sur le cloud et un rapport a été fabriqué (figure 23).

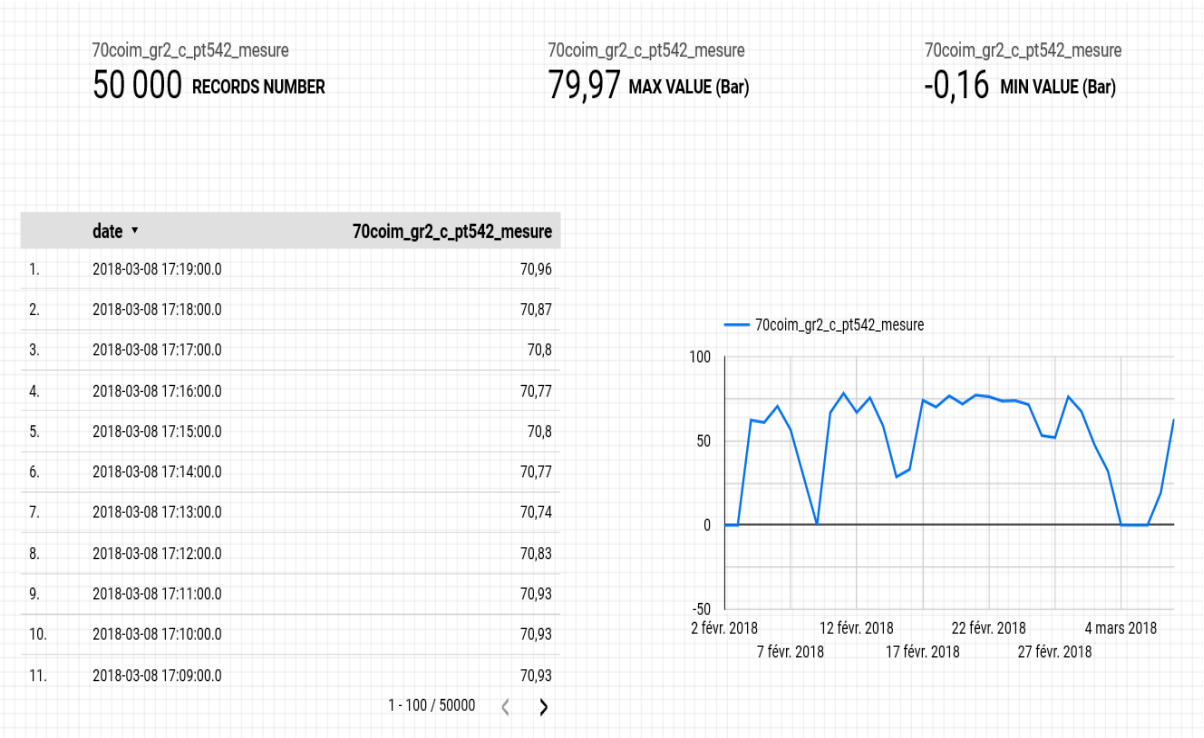


Figure 23 : Rapport Google Data Studio sur la variable principale du compresseur à gaz

- **Les dates limites**

Elles ont été respectées. Voir le cahier des charges en annexe ayant suivi la réunion de cadrage.

- **Le bilan méthodologique**

Le client a décidé d'une gestion de projet en cascade avec d'emblée un projet à concrétiser avec deadlines. Les demandes ont cependant légèrement évolué. L'organisation en interne a quant à elle était agile.

Cette organisation agile a permis une bonne avancée en équipe, pas à pas avec une bonne communication sur les avancées et interrogations.

- # La répartition des tâches

Les 4 personnes ayant travaillé sur le projet (3 développeurs et un graphiste) ont toutes eu leur importance.
L'un deux a travaillé sur une autre piste que moi lors d'un sprint : le deep learning classique pour la prévision de séries temporelles. Cela a fourni une émulation positive avec de nombreux échanges techniques.
Un autre collègue a créé une maquette pour fournir un visuel des résultats au client (annexe1).
Enfin le dernier à mis en place un dashboard sur le web à partir de la maquette.
Par ailleurs, Spie et deux autres entreprises ont été mises en concurrence sur ce projet. Cela m'a également donné une motivation supplémentaire.

- # Les choix techniques et matériel

La VT a été efficace pour définir le use case (la maintenance prédictive) mais aussi utile pour le choix d'outils et techniques pertinents pour tout le workflow. L'abonnement mensuel à « Medium » s'est trouvé être particulièrement intéressant.
La documentation interne technique s'est avérée efficace pour déterminer le rôle des capteurs impliqués dans la dépressurisation.
Python avec toutes ses bibliothèques a permis de faire du machine learning en maintenance prédictive et sa popularité a permis de trouver des exemples de réalisation de projets dans le même domaine.
Jupyter a été efficient comme débuggeur sur des bouts de code alors que Visual Studio 2019 a été intéressant pour déboguer lors de l'exécution d'un code entier.
Tout mon travail a été fait sur l'ordinateur fourni par la formation.
Un serveur fourni transitoirement par Spie à eu la puissance de calcul nécessaire pour faire de la filtration de variables « embedded ».

- # La satisfaction du client

Il est satisfait des résultats obtenus et s'en sert pour promouvoir les compétences de Spie Industries auprès de prospects.

7) LES AXES D'AMELIORATION

- ## Relatifs à la gestion de projet

J'ai trouvé la gestion de projet satisfaisante.
Le seul bémol était de ne pas avoir de référent technique en Data/Intelligence Artificielle.
Cela rendait la qualité de mon travail plus compliquée à évaluer. En effet, les pratiques techniques mises en place étaient basées sur mes compétences acquises en formation et ne pouvaient pas être comparées ou évaluées par des experts métier. Cependant quelques réflexions techniques ont été abordées avec mes formateurs.

- ## Relatifs aux aspects techniques

Les prédictions de séries temporelles effectuées par le LSTM en régression n'ont pas donné des résultats satisfaisants.
Selon moi, la constance dans les prédictions avec le LSTM et le RMSE élevé sont dus à la suppression de plusieurs variables numériques ou d'enregistrements. Ces six variables contenaient un nombre très important de valeurs manquantes (>60%). Cette hypothèse est plausible. En effet, deux variables dans le jeu de données sont nécessaires pour améliorer le model de 15% alors qu'elles contiennent environ 40% de NaNs. Une technique avancée de remplissage des valeurs manquantes n'a pas été utilisée et semble être la plus performante. Il s'agit d'utiliser des modèles de classification ou régression, de les entrainer sur un jeu de données sans trous et de les tester sur un jeu de données avec valeurs manquantes afin de les trouver (27).
Le Random Forest aurait, lui, fait de meilleures prédictions puisqu'il n'utilise pas de séries temporelles.

Par ailleurs, il aurait peut-être été pertinent d'effectuer un nettoyage des données différent. Une focalisation sur les variables importantes pour la prédiction avec un nombre important de NaNs aurait pu être obtenu en croisant les résultats obtenus par la feature selection avec la méthode de filtration et le comptage du nombre de valeurs manquantes par variable. Ces variables auraient ainsi pu bénéficier d'un remplissage de leurs valeurs manquantes par la technique citée ci-dessus.

Enfin, le dernier axe d'amélioration concerne la préparation à l'intégration continue de données. Par exemple, des updates de BDD, des prédictions automatiques à temps précis auraient pu être envisagés afin d'envisager l'évolution du projet. Le stockage des données est prévu sur une plateforme cloud Spie et le stockage de l'algorithme dans un container sur cette même plateforme.
Des alternants DevOps sont en train de créer cette plateforme, celle-ci héberge notamment un logiciel d'intelligence artificielle. Les futurs enjeux sont donc relatifs à la collecte (connecteurs), le stockage et l'analyse continue de données du Big Data.

8) CONCLUSION

Le travail permettant la réalisation de ce rapport a été effectué au sein de Spie ICS, en alternance, durant mes études de développeur DATA/Intelligence Artificielle. Il a impliqué 4 personnes (2 développeurs, un graphiste et le manageur) et donc un travail en équipe.

Le rapport de ma démarche choisie afin de répondre aux enjeux internes fixés par Spie industrie sur le thème de la maintenance prédictive est écrit.
Ce rapport détaille le contexte, les choix techniques et méthodologiques, les apports et axes d'amélioration du projet.

Au cours de ce stage, j'ai apprécié manipuler une variable : le temps restant avant panne (Remaining Useful Life ou RUL, en anglais) conduisant à l'utilisation de techniques singulières de prédictions.
Me concernant, la collection de données et la sélection d'algorithmes ont été les plus estimées.

Pendant toute l'alternance, le travail en méthode agile a été un plus. En effet, en l'utilisant, l'organisation, la communication et l'apprentissage autour du projet ont été renforcés.
En formation, le travail agile a continué au travers des sprints. Celui-ci fut intéressant pour identifier les différentes fonctionnalités d'une application par la revue de sprints. Ces derniers ont permis d'enrichir mon projet par la création d'un webservice, le stockage d'une base de données et la présentation d'un dashboard associé. Le webservice permet de fournir certaines fonctionnalités au client, les dashboards permettent de présenter les résultats au client, le stockage de la BDD est nécessaire puisque cette dernière est amenée à grandir.
Au niveau personnel, ce rapport m'a permis d'être plus pragmatique. Pour cela j'ai dû conceptualiser mes futures actions et en pratique comprendre l'intérêt d'une tâche et résoudre les erreurs dans l'exécution de celle-ci. Ces erreurs étaient de natures diverses : manque de connaissance, d'attention, de chance, problème de raisonnement.

La réalisation du POV a eu un retour positif du client sur l'atteinte des objectifs (les principaux : prédictions de pannes et causes relatives). Ces objectifs jouent un rôle important dans l'identification de dysfonctionnements et dans l'augmentation de la longévité des compresseurs à moindre coût.
Le client pense à une mise en production de l'algorithme lui permettant d'obtenir des prédictions 30 minutes après l'arrêt du compresseur et avec une fiabilité à plus ou moins deux heures.

Enfin, l'évolution naturelle du projet est l'intégration du temps réel avec intégration continue des données.

En annexe 2, différentes tâches réalisées en entreprise sont répertoriées.

BIBLIOGRAPHIE

Sources internet :

(1) https://fr.wikipedia.org/wiki/Maintenance_pr%C3%A9visionnelle

(2) https://www.connaissancedesenergies.org/vers-30-de-gaz-renouvelable-en-france-171121

(4) https://magazinemci.com/2015/04/06/comment-revoir-les-processus-de-maintenance-predictive-pour-assurer-la-longevite-des-equipements-a-moindres-cout/

(5) https://www.lesechos.fr/idees-debats/cercle/pourquoi-la-maintenance-predictive-va-t-elle-revolutionner-lindustrie-131697

(6) https://fr.wikipedia.org/wiki/Turbine_%C3%A0_gaz

(8) https://www.mobility-work.com/fr/blog/maintenance-predictive-vs-maintenance-preventive-strategie-entreprise

(9) https://medium.com/bigdatarepublic/machine-learning-for-predictive-maintenance-where-to-start-5f3b7586acfb

(10) https://gallery.azure.ai/Experiment/a677f8eececf40eaa158699a2b27e3c8

(11) http://cedric.cnam.fr/vertigo/Cours/ml2/coursArbresDecision.html

(12) https://lovelyanalytics.com/2016/08/20/random-forest-comment-ca-marche/

(13) https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e920d

(14) https://medium.com/@gabrieltseng/gradient-boosting-and-xgboost-c306c1bcfaf5

(15) https://mc.ai/introduction-au-deep-learning-et-aux-reseaux-de-neurones-pour-les-nuls/

(16) https://medium.com/smileinnovation/lstm-intelligence-artificielle-9d302c723eda

(17) https://towardsdatascience.com/choosing-the-right-hyperparameters-for-a-simple-lstm-using-keras-f8e9ed76f046

(18) https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10

(19) https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5

(20) https://www.decideo.fr/Big-Data-IoT-IA-quelle-gestion-de-la-donnee_a11227.html
https://www.silicon.fr/hub/hpe-intel-hub/collecter-transporter-stocker-et-traiter-les-donnees-iot

(21) https://mrmint.fr/data-preprocessing-feature-scaling-python

(22) https://github.com/Samimust/predictive-maintenance

(23) https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

(24) https://elitedatascience.com/overfitting-in-machine-learning

(25) https://openclassrooms.com/fr/courses/4297211-evaluez-les-performances-dun-modele-de-machine-learning/4308256-evaluez-un-algorithme-de-classification-qui-retourne-des-valeurs-binaires

(26) https://medium.com/@cxu24/common-methods-for-feature-selection-you-should-know-2346847fdf31

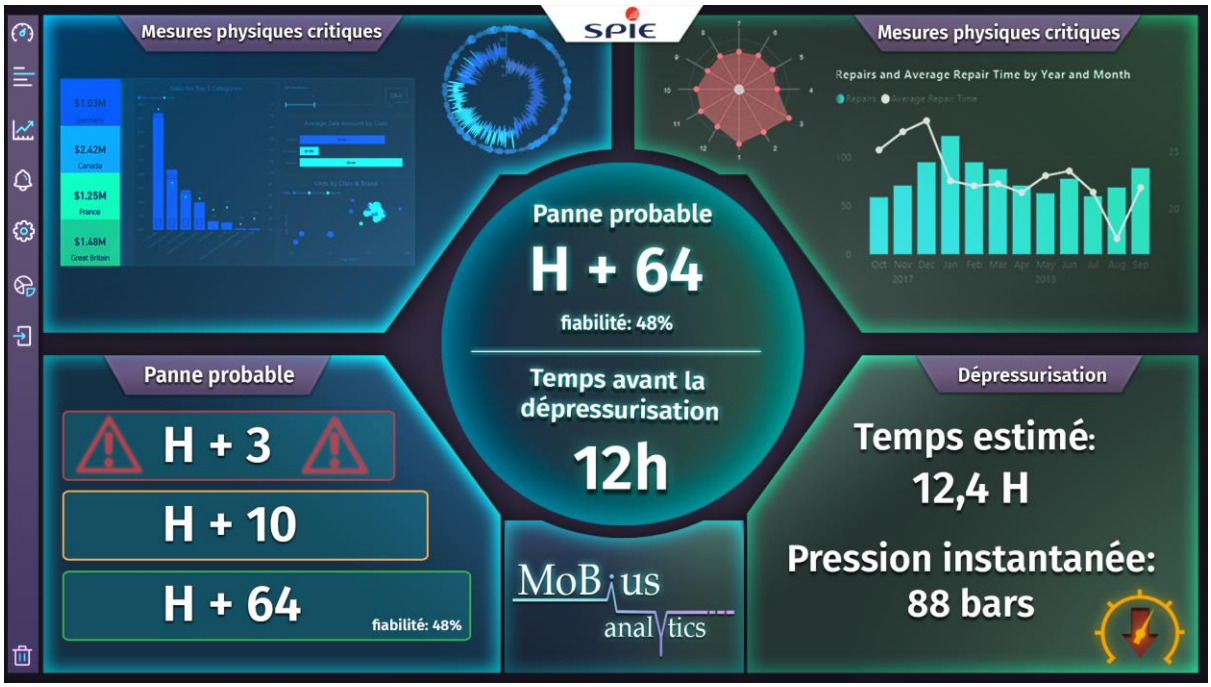
(27) https://towardsdatascience.com/the-tale-of-missing-values-in-python-c96beb0e8a9d

Sources documents :

(3) http://www.grtgaz.com/fileadmin/medias/communiques/2018/FR/Presentation-bilan-gaz-2017.pdf

(7) https://www.iacpartners.com/uploads/Publications/Pred%20Maintenance/iac-partners-maintenance-predictive.pdf

ANNEXES



Annexe 1 : Maquette du projet Pov compressor

Charte graphique validée.
Prévision en haut à gauche et droite : mettre des graphiques (courbes et histogrammes) des variables les plus corrélées à la dépressurisation et leur covariance avec la variable cible.
Prévision en bas à gauche : prédiction de panne dans une plage horaire à venir (algorithme de classification).
Prévision en bas à droite : prédiction de panne à un temps précis (algorithme de régression).
Prévision au centre : récapitulatif des deux prédictions réalisées.

ANALYSE			
Objectifs pédagogiques	**Activités , tâches**	**détail des tâches réalisées**	**Technologies / outils utilisés**
A1. Développement d'une base de données		**Création d'un modèle entité-association et passage au modèle relationnel avec clés primaires et étrangères**	**Sqlite, MySQL**
C1. Concevoir et structurer physiquement une base de données relationnelle ou non, à partir des besoins, contraintes et données du commanditaire.	**Identification du type de base de données approprié à la demande.** **- Conception du modèle de données en respectant les standards.** **- Création d'une base de données relationnelles et/ou NoSQL.** **- Mise en place d'une planification des sauvegardes de la base de données.**	**Création, remplissage et visualisation d'une BDD**	**Sqlite et DB Browser.**
C2. Acquérir des données, les combiner et les structurer en données propres en vue de leur intégration dans la structure de la base de données.	**Recensement des données à utiliser, leurs formats, leurs sources, leurs structures ainsi que leurs détenteurs.** **- Collecte des données.** **- Nettoyage des données à importer, à l'aide de scripts ou de logiciels spécifiques appropriés.** **- Manipulation des données sous divers formats de fichier plats (XML, JSON, CSV)** **- Création des fichiers de sauvegarde des données propres.** **- Gestion des fichiers de métadonnées associés aux fichiers : création, mise à jour ou suppression.**	**Manipulation des données sous divers formats de fichier plats (XML, JSON, CSV)** **Nettoyage des données.** **Combiner : création de colonnes variance et moyenne mobiles sur un intervalle de temps (projet gaz compressor et Airbus).** **Structurer en données propres : remplissage ou suppression des valeurs manquantes. Création d'une colonne target représentant le temps restant avant la panne.** **Enregistrement automatique : algorithme de détection des temps entre la mise en route de gaz compresseur et la dépressurisation. Recueil de données et traitement automatisé sur ces plages de temps.** **Création des fichiers de sauvegarde des données propres.**	**Algorithmes d'automatisation en python (gaz compressor), sauvegarde poids algorithme pickle (.pkl) / joblib.dump (.z), sauvegarde fichier avec : to_excel.**
C3. Intégrer des données propres et préparées dans la base de données finale, en utilisant des langages informatiques, logiciels ou outils.	**Choix de la méthode d'import.** **- Intégration, à partir de fichiers plats, de tables ou d'une interface de programmation, automatiquement ou manuellement, les données dans la base.** **- Import des données en continu ou en temps réel à partir de tâche planifiée, de stream ou tout autre moyen justifié.**	**Importation en continu sur les données (pas de données en continu. Intégration, à partir de tables).**	

C4. Optimiser une base de données afin d'en maintenir la fiabilité et la qualité des données. Nettoyer et améliorer les performances.	**Recherche automatique ou manuelle des erreurs en base de données (doublon, format)** **- Mise à jour et suppression des données en erreur.** **- Mesure des performances des requêtes en utilisant des outils ou des fonctions spécifiques.** **- Optimisation des performances des requêtes en utilisant des outils ou des fonctions spécifiques.**	**Mesure des performances des requêtes par le temps d'exécution et fonctions spécifiques de ranking de tickets d'incident (Projet ranking EMMA).**	
A2. Exploitation d'une base de données			
C5. Interroger et traiter, simultanément et au niveau approprié, des données afin de les stocker en sécurité, brutes ou traitées, provisoirement ou durablement, en fonction du résultat recherché	**Identification du mode de récupération et de traitement des données stockées.** **- Interrogation de la base de données** **- Traitement des données récupérées au besoin et niveau de complexité nécessaire.** **- Mise en forme des données extraites en respectant les spécifications attendues.** **- Mise à jour des données en base de données afin de conserver les résultats obtenus dans l'optique d'une utilisation immédiate ou future.**	**Interrogation et traitement de données en BDD pour établir un algorithme de ranking en SQL server. (projet ranking EMMA).**	**SQL server**
C6. Concevoir et réaliser un rendu visuel des données issues du processus d'extraction, à l'aide d'un (des) support(s) adapté(s) répondant aux attentes du commanditaire.	**Choix des représentations visuelles des données en adéquation avec les contraintes techniques, réglementaires, la demande du commanditaire et leur utilisation** **- Réalisation des représentations visuelles esthétiques et fonctionnelles des données en utilisant des outils spécifiques de visualisation .** **- Génération, si nécessaire, de données complémentaires indispensables à la réalisation des supports.**	**Création de plusieurs rendus visuels partagés lors de réunions sur Teams.** **Rendus visuels (projet Airbus) sur la position de modules d'assemblage de moteurs d'avion. (des gaps observés) + requêtes sur Dataframe pour focus sur données d'intérêt.** **Rendus visuels pour le projet gaz compressor : presentation PPT (monitoring, heatmap, selection de features, spécificité des dataframes en maintenance predictive, courbes d'accuracy avec et sans overfitting),** **Heatmap (covariance) sur le projet gas compressor (supression de variables trop proches), visualisation des NaNs.**	**Seaborn, heatmap, Streamlit(gaz compressor, garbage detection project), matplotlib, selectKbest, missingno …**

C7. Mettre à disposition les rendus visuels simples des données en accès libre ou contrôlé.	**Choix du support de à diffusion des rendus visuels.** **- Mise en conformité avec la réglementation relative aux données.** **- Réalisation du (des) support(s) statiques et dynamiques.** **- Mise en place des droits d'accès et d'utilisation en fonction du support (serveur – Internet).**	**Mise en conformité avec la réglementation relative aux données. - Réalisation du (des) support(s) statiques et dynamique** **Mise en place des droits d'accès et d'utilisation en fonction du support (website)** **Réalisation du (des) support(s) statique et dynamique**	**Création d'un webservice (Garbage detection) utilisé dans un container Docker (garbage detection project). PDF en présentation interne. WordPress. Supports dynamiques avec Metabase, Superset,**
A3. Gestion de projet et qualité			
C8.Analyser et formaliser la demande ou le besoin en développement et en exploitation de base de données.	**Analyse de la demande client.** **- Identification, à partir du cahier des charges, les utilisateurs et leurs profils, les différents besoins, les contraintes techniques et réglementaires ainsi que les données du commanditaire.** **- Le cas échéant, formalisation d'un cahier des charges du projet à partir de la demande client.**	**Formalisation.** **Commmunication du besoin client.**	
C9. Autocontrôler, tout au long du processus de développement, la cohérence des données et la conformité à la demande.		**Autoévaluation des offres avec le besoin de client,**	**Travail en agilité.** **Planification de taches version 1 puis 2...**
C10. Suivre, adapter et rendre compte de la réalisation du projet à partir du planning projet validé.	**Suivi, adaptation et communication de la réalisation du projet à partir du planning projet validé.** **- Suivi du projet, dans un objectif d'optimisation, en utilisant une méthodologie adaptée.** **- Adaptation du projet aux contraintes et problématiques rencontrées** **- Animation des réunions de travail ou d'ajustement du projet.**	**feed bac,** **rendre compte des problématiques.**	**Travail en agilité.**

C11. Rechercher des solutions pour la résolution de problèmes techniques rencontrés au moyen des ressources disponibles (documentation, sites Internet, communautés, etc..).	**Documentation et analyse des informations sur les technologies informatiques récentes pour répondre à un besoin de compréhension ou de recherche d'information** **- Recherche de solutions pertinentes pour la résolution de problèmes techniques à partir de :** **- sites spécialisés** **- communautés de spécialistes des données accessibles par internet.** **- autres**	**Utilisation des ressources pour répondre aux problématiques**	**Internet.**
A4.Exploiter l'intelligence artificielle dans le développement d'applications			
Traitement et analyse des données permettant la mise en place de modèles d'apprentissage suivant une méthodologie définie			
C12. Constituer un jeu de données exploitable de manière à entraîner un modèle d'apprentissage en utilisant la méthodologie et/ou l'outil approprié en fonction des standards de l'écosystème	**Sélection de l'outil d'analyse de données en fonction des standards de l'écosystème technique du projet** **- Détection des valeurs anormales dans le jeu de données / Validation des données par la détection de valeurs anormales** **- Nettoyage et traitement des données exploitables à l'aide d'une bibliothèque logicielle*** **- Constitution d'un jeu de donnée au format de donnée préalablement identifié/sélectionné**	**Préparation données projet chef d'œuvre.**	**Jeu de données (poc compressor) exploitable obtenu en python. Methodo type : Algo classification et régression en maintenance prédictive.**
C13. Interpréter les données grâce à des outils de visualisation de données en vue d'expliquer les caractéristiques du jeu de données	**Encodage* des données au format adapté à l'aide de l'outil préalablement sélectionné** **- Génération de données pour augmenter la quantité de données exploitables** **- Réduction de la dimensionnalité des données** **- Visualisation des données à l'aide d'outils**	**décodage des données binaires(16 bits) en décimales (Poc Airbus)** **- Génération de données pour augmenter la quantité de données exploitables : moyenne et variance mobile (Poc airbus et gas compressor)** **- Réduction de la dimensionnalité des données (PCA, heatmap)** **- Visualisation des données à l'aide d'outils (matplotlib pour accuracy lors de l'entrainement de modèles (poc compressor).**	**matplotlib, heatmap PCA, moyenne et variance mobiles**

*Exploitation d'un modèle d'apprentissage en utilisant les méthodes du machine learning**			
C14. Exploiter un modèle d'apprentissage supervisé ou non supervisé permettant la classification ou la prédiction d'une variable en fonction des données disponibles et des outils sélectionnés	**Identification du modèle d'apprentissage optimal en fonction du problème à résoudre, des données disponibles et de leurs natures** **- Sélection de l'outil (langage*, bibliothèque*, framework*, plateformes)** **- Entraînement et exploitation d'un modèle d'apprentissage supervisé* à l'aide d'outils préalablement sélectionnés** **- Classification ou prédiction d'une variable à partir d'un modèle d'apprentissage supervisé** **- Réalisation de divers traitements à l'aide d'un modèle d'apprentissage :** **● langage naturel** **● séries temporelles** **● vision par ordinateur** **- Utilisation de l'apprentissage non supervisé pour créer des catégories**	**Choix du modèle d'apprentissage. Ce qu'il se fait en maintenance prédictive(VT).** **Choix de l'outil choisi par VT.** **Choix d'un modèle pour série temporelle.**	**LSTM, Random forest**
C15. Améliorer les performances d'un modèle d'apprentissage à l'aide d'une évaluation de la qualité des données et de la technique de modélisation afin de réduire les biais et les anomalies de résultats	**Evaluation de la performance d'un modèle d'apprentissage avec les métriques standards* et spécifiques** **- Identification des hyper-paramètres du modèle** **- Amélioration de données d'apprentissage d'après une analyse des métriques de performance** **- Combinaison de plusieurs modèles en un modèle plus performant**	**Poc compressor :** **Evaluation de la performance d'un modèle d'apprentissage avec les métriques standards et spécifiques.** **- Identification des hyper-paramètres du modèle.** **- Amélioration de données d'apprentissage d'après une analyse des métriques de performance (automatisation du choix des meilleurs hyperparamètres, selection des données d'apprentissage : filter method, wrapper based, embedded (sur VM)).** **- Test XGBoost (poc gaz compressor).**	**MSE, RMSE, F1 score, accuracy, hyperparamètres Random Forest, filter method(selectKbest, selectionKhi2), wrapped (SelectFromModel Random forest) (poc compressor), XGBoost**

Assemblage d'un modèle d'apprentissage profond			
C16. Concevoir un modèle d'apprentissage efficient en exploitant les méthodes standards d'apprentissage profond pour répondre à une problématique identifiée	**Sélection d'une architecture d'apprentissage profond standard en fonction des données disponibles** **- Implémentation d'un modèle d'apprentissage profond préalablement sélectionné à l'aide d'une bibliothèque** **- Utilisation d'un modèle d'apprentissage profond pré-entraîné (apprentissage par transfert)** **- Entraînement d'un modèle d'apprentissage profond**	**Garbage detection : Utilisation d'un modèle d'apprentissage profond pré-entraîné** **Poc Compressor : utilisation du LSTM en classification et regression.**	**Yolov3, LSTM**
Intégration de solutions IA pré-existantes pour optimiser la réponse aux besoins du client final			
C17. Sélectionner l'outil le plus adapté aux objectifs préalablement définis grâce aux services IA disponibles sur une plateforme cloud afin de répondre aux enjeux rencontrés par le client	**Traduction des enjeux du client interne/externe en objectifs** **- Identification des différents services et solutions IA disponible sur une plateforme cloud** **- Sélection de l'outil le plus adapté aux enjeux préalablement définis**	**Utilisation de Thingworx après VT et discussion avec le client de l'intérêt de la plateforme (IA, temps réel).**	**Thingworx**
C18. Améliorer une application en étendant ses fonctionnalités grâce à l'utilisation d'API web des services IA de manière à répondre aux objectifs préalablement définis avec le client	**Identification des méthodes cloud permettant de faciliter la gestion et l'exploitation d'un modèle d'apprentissage** **- Utilisation de divers traitements à l'aide d'une plateforme cloud :** **● langage naturel** **● séries temporelles** **● image et vision artificielle** **- Entraînement d'un modèle d'apprentissage à l'aide d'une plateforme cloud** **- Utilisation d'une plateforme cloud pour exposer un modèle d'apprentissage** **- Exploitation d'une API web* exposant des services d'IA**	**Eu accès aux Algorithmes et à la réalisation de prédictions par IA grâce aux services de Microsoft Azure + test rapide.** **Test de Poc compressor sur la plateforme IOT Thingworx : Compréhension et utilisation de tous les modules de thingworx analytics, entrainement de modèles IA.** **+ VT sur thingworx (module thing, template...) pour analyse de données en temps réel.** **+ VT et test de salesforce (plateforme de service avec IA) : compte d'entrainement.**	**Microsoft Azure, Thingworx Analytics, Salesforce (Einstein).**

Développer des applications exploitables par le client final en intégrant les solutions IA réalisées et/ou pré-existantes			
C19. Développer une application et/ou des fonctionnalités utilisant le traitement de données généré par l'IA de manière à être exploitable par le client/utilisateur final	**Développement des composants d'une interface utilisateur intégrant les fonctionnalités d'IA** **- Exposition d'un modèle d'apprentissage dans un web service simple pour faciliter son utilisation à une personne tierce** **- Utilisation d'un gestionnaire de conteneur** **- Automatisation du déploiement d'applications dans des conteneurs logiciels** **- Versionnage du code source** **- Partage des différentes sources à l'aide du système de versionnage**	**Exposition d'un modèle d'apprentissage dans un web service.** **- Utilisation d'un gestionnaire de conteneur : webservice dans un container.**	**Streamlit, Docker, Linux**
C20. Réaliser des visualisations adaptées au public visé afin de communiquer les résultats d'un projet mené	**Identification de la cible auprès de laquelle communiquer (interne/externe, équipe projet ou direction opérationnelle, tout public ou restreint..)** **- Sélection des moyens de diffusion des résultats auprès de la cible** **- Réalisation des visualisations afin de communiquer ses résultats**	**- Identification de la cible auprès de laquelle communiquer (interne : Spie Industrie (Pov compressor), demande tuteur(Garbage detection)** **- Sélection des moyens de diffusion des résultats auprès de la cible : webservice (Garbage detection).** **- Réalisation de visualisations.**	**Webservice, Streamlit**
Conception et mise en oeuvre d'un système de veille technologique pour aider à la prise de décision			
C21. Concevoir un système de veille technologique permettant de collecter, classifier, analyser et diffuser l'information aux différents acteurs de l'entreprise/l'organisation afin d'améliorer la prise de décisions techniques	**Sélection des sources d'information pertinentes et état de l'art en français et anglais** **- Collecte des données/informations liées aux problématiques rencontrées par l'organisation (évolutions ou émergences de nouvelles techniques…)** **- Analyse des informations collectées** **- Mise en oeuvre d'un outil d'aide à la décision afin de résoudre un problème concret**	**Nombreuses VT diffusées sur Teams et envoyées par email :** **Salesforces et Salesforce Einstein, AiOps, Thingworx, Bert, Metabase, Superset, maintenance prédictive…**	**Teams, Outlook**

Annexe 2 : Compétences à valider et tâches réalisées en entreprise

LEXIQUE

BDD : base de données.

GMAO : Logiciel de Gestion de Maintenance assistée par Ordinateur. L'une de ces utilités est la gestion des maintenances correctives et préventives (systématique, prévisionnelle).

IA : intelligence artificielle.

IOT : Internet of Things. En français, Internet des Objets (IdO).

ROI : Return On Investment. En français, Retour Sur Investissement (RSI).

RUL : Remaining Useful Lifetime.

Précision =

F1 score = 2*((precision*recall)/(precision + recall))

Accuracy = nombre de prédictions correctes / nombre total de prédictions.