POLYTECH TOURS
64 avenue Jean Portalis
37200 TOURS, FRANCE
Tél +33 (0)2 47 36 14 14
www.polytech.univ-tours.fr

Rapport de stage de 4e année (2023-2024)

Exploration des grands modèles de langages pour réaliser des prédictions et détections d'anomalies dans des séries temporelles

Entreprise :

EVERDYN

10 Rue Aristide Briand
37390 Notre Dame D'Oé
France

Étudiant :

MERCIER Titouan

Promo 2025

Tuteur Entreprise :

FREMONT Thomas

Responsable Agence Centre Ouest

Tuteur académique :

RAGOT Nicolas

Sommaire

Sommaire 1
Table des illustrations 2
Introduction 3
1 Environnement 4
1.1 Structurel 4
1.2 Relationnel 5
1.3 Contextuel 6
2 Synthèse du travail effectué 7
2.1 Recherche sur le Domaine 7
2.1.1 Familiarisation avec le Domaine 7
2.1.2 Fonctionnent des LLM 9
2.1.3 Les techniques pour adapter un LLM 10
2.2 Réalisation 13
2.2.1 Faire un apprentissage 13
2.2.2 Création des dataset 16
2.2.3 Évaluation des performances 18
3 Analyse réflexive de l'expérience 22
3.1 Mon organisation 22
3.2 Comparatif prévision/réalisation effective 25
3.3 Les compétences acquises et consolidées 26
Bilan de l'expérience 28
Bibliographie 29
Table des annexes 29

Table des illustrations

Figure 1: Prompt clair 10
Figure 2: Prompt ambigu.....10
Figure 3 : Schéma génération augmentée de récupération.....11
Figure 4 : Schéma apprentissage fin 12
Figure 5 : Interface de l'API Auto Train 14
Figure 6: Example prompt Type 1 17
Figure 7 : Example prompt Type 2 17

Introduction

Curieux et désirant évoluer dans ma pratique de l'informatique, je souhaitais clôturer ma 4e année de formation d'ingénieur en réalisant un stage en entreprise. Ce stage était pour moi l'opportunité de confronter pour la première fois mes connaissances et compétences en informatique au sein d'un environnement professionnel. Il était également l'occasion d'avoir un potentiel aperçu de mon futur professionnel.

En effet je souhaite travailler dans le domaine de la cybersécurité. Pour tendre à cet objectif, je me suis par ailleurs engagé l'année prochaine à poursuivre, en double diplôme, une maîtrise en cybersécurité. J'ai donc initié mes recherches de stage vers ce domaine afin d'acquérir ma première expérience en cybersécurité. Je suis également resté ouvert à toute autre proposition de stage dès l'instant quelle correspondait à ma formation d'ingénieur.

Cet état d'esprit m'a amené à réaliser mon stage au sein de l'antenne tourangelle d'EVERDYN. Cette entreprise m'a permis de passer mes 9 semaines de stage autour du domaine de l'intelligence artificielle. En effet j'ai dû explorer les grands modèles de langages pour réaliser des prédictions et des détections d'anomalies dans des séries temporelles.
Cela peut à première vue sembler être éloigné de mon désir professionnel. Néanmoins, à la vue de la proposition de celui-ci, je me suis rendu compte que le domaine de l'IA m'est relativement obscur. En effet durant ma formation d'ingénieur mes choix de cours et de projets n'étaient pas orientés vers le domaine de l'IA. Au vu de l'engouement actuel et de son imbrication grandissant au sein de la société et des différents domaines de l'informatique, je trouvais très intéressant de passer ces semaines à mieux le comprendre.

Je vous propose à travers ce rapport de vous présenter dans un premier moment l'environnement de mon stage. Nous pourrons mieux connaître l'entreprise et les collaborateurs avec qui j'ai pu évoluer du 17 juin au 16 août, puis nous comprendrons le contexte et l'objectif de mon stage.
Nous verrons dans un second moment le travail accompli. Dans cette partie je détaille chaque étape qui importe, et évoque les choix faits face aux difficultés rencontrées. Nous finirons avec une analyse réflexive de mon expérience. Ici je vous présente et critique mon organisation, je fais également un point sur les connaisseurs et compétences acquérir durant mon stage.

1 Environnement

1.1 Structurel

Fondée en 2004, EVERDYN est une PME de 18 salariés domiciliée à Aix en Provence mais présente dans toute la France grâce à son réseau de 5 agences. Ces agences ne sont pas réparties en centre de profit, il n'y a donc pas de concurrence entre elles.

EVERDYN évolue dans deux domaines d'activité. La première, leur principale activité présente au sein des 5 agences, est l'intégration, réparation et supervision de système. Ils sont capables de concevoir, programmer, démarrer la plupart des systèmes automatisés du monde industriel.
Leur second domaine d'activité, présent seulement au sein de l'agence d'Aix-en-Provence, est le développement de logiciels. Ils développent des logiciels industriels pour les utilisateurs d'AVEVA Software. Leur produit majeur est le logiciel et Interface de programmation d'application ALPANA. Elle permet de faire du « Dashboard », c'est-à-dire transformer des données en information par l'intermédiaire de représentation graphique. Elle est capable de s'interfacer avec toutes les sources de données possibles et permet aux industriels de monitorer leurs chaines de production, de partager et visualiser des tableaux de bords sur tout type de support électronique.

Crée en 2007, l'agence de Tours, où j'ai effectué mon stage, travaille uniquement dans l'intégration, la supervision et la réparation de systèmes. L'ensemble des six collaborateurs de Tours sont des automaticiens. Ils travaillent pour des vendeurs de matériel industriel. Leur rôle est, une fois le matériel livré à l'entreprise acheteuse, d'intégrer l'automatisation du nouveau système et de gérer l'interfaçage avec le reste de la chaine de production. Dans ce cadre EVERDYN travaille régulièrement pour CIAM PAKE, spécialiste dans la conception de systèmes de convoyage de contenants. Actuellement, ils travaillent ensemble à l'intégration d'un système de convoyeurs à l'entreprise Laiterie de Saint-Denis-de-l'Hôtel.
Ils ont également des clients de long terme, c'est-à-dire qu'ils travaillent chez eux sur de plus longues périodes et directement dans leurs entreprises pour superviser et développer l'automatisation de leur système. Nous pouvons citer par exemple l'entreprise SKF ou STMicroelectronics.

1.2 Relationnel

Durant mon stage, j'ai majoritairement interagi avec trois personnes.

M. FREMONT Thomas, mon maître de stage :

Automaticien de métier, il est actuellement Responsable de l'agence de Tours. Il s'occupe de la gestion du personnel, de la planification des projets et de la répartition des effectifs chez les différents clients. Cela occupe environ 10% de son activité. Le reste du temps, il exerce son métier d'automaticien et conseille éventuellement les autres collaborateurs grâce à l'expérience qu'il a pu accumuler depuis son arrivée au sein de la société en 2008.

Il a été là pour m'introduire dans l'entreprise, me fournir le matériel nécessaire à mon travail ainsi que répondre à mes différentes questions sur l'organisation de l'entreprise. Il a été également là pour vérifier que je n'étais pas en difficulté dans la réalisation de mon stage.

M. LE NY Nicolas, Gérant de EVERDYN :

Il travaille au siège social à Aix-en-Provence. Il a défini mon sujet de stage et qui a contrôlé mon avancement. C'est également vers lui que je devais me tourner pour toutes les questions relatives à la définition du besoin ou aux attentes de l'entreprise sur la réalisation de mon stage. Nous avons exclusivement échangé par mail et téléphone.

Mme. LE NY Cécile, Assistante Administrative :

Elle s'est occupée de mon accueil administratif, vérifiant la validité de ma convention de stage et en s'assurant que j'ai eu à disposition tous les documents légaux obligatoires. Elle m'a aussi aidé à obtenir toutes les informations à propos des politiques de qualité de vie au travail mise en œuvre au sein de la société.

Les différents collaborateurs de l'agence de Tours :

Ils sont rarement présents dans les locaux de l'entreprise car souvent en déplacement chez les clients, cela étant lié à la nature de leur travail. Ils sont au nombre de six, dont mon maître de stage. J'ai eu le temps de les croiser tout au long de ma présence au sein de EVERDYN. Nous avons pu échanger lors de moments plus ou moins formels à propos de mon sujet de stage et de leur métier. Cela ma permis de mieux comprendre les différentes facettes du métier d'automaticien ainsi que les différentes parties qui pouvaient être amenées à participer à l'arrivée d'un nouveau système au sein d'une entreprise.

1.3 Contextuel

M. LE NY Nicolas a pour ambition de créer un outil informatique permettant d'anticiper les comportements des différents systèmes de l'industrie. Il permettra aux industriels de prédire les futurs dysfonctionnements des systèmes ou de prédire des variables liées à celui-ci comme la consommation énergétique ou le rendement. Cet outil sera proposé aux entreprises souhaitant prédire le comportement de leurs systèmes sans avoir à investir dans la création d'un modèle de connaissance de celui-ci, qui peut s'avérer coûteux.

Cet outil se veut générique et facilement adaptable afin d'être déployé dans tous types d'entreprises et pour tous types de systèmes.
Pour fonctionner, cet outil se basera sur les données et covariables des différents systèmes. Il devra effectuer des prédictions et une détection d'anomalie sur les données produites par ces derniers. Dans l'industrie ces données sont nombreuses et déjà stockées de manière chronologique dans des bases de données. Cette solution devra donc être capable d'assimiler les différentes données des systèmes d'une entreprise spécifique afin que chaque solution vendue soit adaptée à l'acheteur. Cette solution devra également être capable de s'exécuter localement dans chaque entreprise, la majorité ne voulant pas que leurs données soit exfiltrées de leurs systèmes. Les résultats de ces prédictions et détections seront ensuite utilisés par l'entreprise pour faciliter ces futures prises de décision.

Nous pouvons résumer cette solution en un mini cahier des charges de 2 fonctionnalités et une contrainte.
Mini cahier des charges :
FC1 : Prédire les futures valeurs d'une série temporelle
FC2 2 : Détecter une anomalie sur des valeurs de séries temporelles
CT1 : l'outil devra fonctionner en local au sein de l'entreprise pour lequel il fonctionne

Périmètre de mon stage :
L'objectif qui m'a été fixé par M. LE NY est d'explorer le potentiel de la technologie LLM (Large Model Language) pour la réalisation des prédictions et détections d'anomalies sur des séries temporelles. Cela lui permettra de voir s'il y a un intérêt à utiliser les LLM pour le développement de son nouvel outil. Son désir d'aller regarder vers cette technologie a été mû principalement par la lecture de l'article « Large Language Model Performance in Time Series Analysis » publié le 1er mai 2024 sur medium.com, un site de blog qui a une dimension de partage de contenu scientifique par la présence de nombreux articles sur divers types de technologie.
Il souhaitait également mieux comprendre cette technologie qui connait actuellement un grand engouement.

2 Synthèse du travail effectué

2.1 Recherche sur le Domaine

Durant les 2 premières semaines de mon stage j'ai commencé par étudier le sujet en réalisant un mini état de l'art. J'ai commencé par me renseigner sur les différentes techniques qui permettent de réaliser des prédictions sur des séries temporelles. J'ai résumé ces recherches dans la sous-partie suivante « familiarisation avec le domaine ». Ensuite, je me suis renseigné plus précisément sur la technologie LMM qui est au cœur du sujet de mon stage. J'ai également regardé les différentes méthodes existantes pour adapter leurs comportements à une tâche spécifique. J'ai retranscrit ces résultats dans les parties 2.1.2 et 2.1.

2.1.1 Familiarisation avec le Domaine

C'est quoi une série temporelle ?

Une série temporelle est définie comme étant une suite d'observations d'une variable mesurée de façon ordonnée dans le temps. Si la série contient des observations pour plusieurs variables, elle est dite multivariée. Dans le cas contraire, il s'agit d'une série temporelle univariée. Elle peut être aussi continue ou discrète. Durant mon stage, j'ai manipulé seulement des séries temporelles discrètes.

Une série temporelle peut être découpée en 3 composantes :

- La tendance : Elle traduit le comportement "moyen" de la série, c'est à dire qu'elle représente sont évolution à long terme.
- La saisonnalité : Elle correspond à un phénomène qui se répète à intervalles de temps réguliers (c'est un phénomène saisonnier).
- Le bruit : Il correspond à des fluctuations irrégulières, en général de faible intensité mais de nature aléatoire.

Comment faire des prédictions et de l'analyse d'anomalies dans les données ?

La prédiction et détection d'anomalies sur des valeurs à partir de séries temporelles n'est pas un sujet nouveau. Les récentes avancées en intelligence artificielle ont permis des progrès notamment en pouvant prendre en compte un nombre élargi des facteurs qui peuvent influencer le comportement d'une série temporelle. Les premiers travaux de recherche à ce sujet ont été initiés dans le domaine de la statistique et de l'analyse de données. Ces techniques sont encore très performantes et largement utilisées. On peut citer par exemple les deux méthodes suivantes :

- Lisage exponentielle : Cette méthode attribue des poids dégressifs aux observations passées. Ce processus est contrôlé par un paramètre de lissage qui varie entre 0 et 1. Plus nous sommes proches de 1, plus le modèle réagit rapidement aux changements récents dans les données. À l'inverse, plus nous sommes proches de 0, plus la série sera d'avantage lissée, réduisant la sensibilité aux fluctuations rapides. Cette technique est particulièrement efficace pour des séries chronologiques qui présentent une tendance ou une saisonnalité, en permettant d'obtenir des prédictions plus stables et plus précises sur le court terme.

- Modèle ARIMA/SARIMA : Ces modèles sont largement utilisés en économie, en finance et dans de nombreuses autres disciplines pour produire des prévisions précises basées sur des données historiques. ARIMA fonctionne grâce à l'auto-régression (AR), qui prend en compte les valeurs passées de la série pour prévoir les futures, l'intégration (I), qui rend la série stationnaire en éliminant les tendances, et la moyenne mobile (MA), qui lisse les fluctuations en utilisant des moyennes mobiles. Le modèle SARIMA étend ARIMA en ajoutant des composantes saisonnières pour traiter les séries présentant des patterns cycliques. Cependant, pour utiliser ces modèles, il faut les adapter pour chaque utilisation. En effet, il faut régler diffèrent paramètre (p,d,q, P, D, Q), la bonne définition des paramètres nécessite une analyse précise des données.

- PCA : l'analyse en composante principale est une très bonne méthode pour détecter des anomalies au sein la série temporelle multivarié car cette technique permet de réduire la dimension du problème.. Cette technique nécessite l'interprétation humaine afin de d'identifier les valeurs qui s'écartent au sein des composantes principales.

Avec l'arrivée des réseaux de neurone, de l'apprentissage automatique puis de l'apprentissage profond de nouvelles techniques sont arrivées pour réaliser des prédictions sur des séries temporelles :

- LSTM : (Long Short-Term Memory) : ils sont un type de réseau de neurones récurrents (RNN) conçu pour traiter et modéliser des données séquentielles tout en surmontant les limitations des RNN traditionnels qui ont le problème du gradient évanescent qui les rendent inefficaces pour apprendre à partir de longues séquences, l'information tend à se perdre au fil des couches. Ces LLM sont souvent utilisés pour réaliser des détections d'anomalies pour les données et peuvent également être utilisés pour de la prédiction de valeur.
- T-GCN : (Temporal Graph Convolutional Networks), apparu en 2022, cette technologie permet de prédire des séries chronologiques non stationnaire. Elle combine des réseaux de neurones convolution (CNN) et des graphes.
- Isolation Forest : cette méthode est très performante pour détecter les anomalies dans les dataset notamment avec les séries temporelles multivariées. Cette méthode fonctionne grâce à l'apprentissage automatique non supervisée. Cette technique fonctionne en créant un ensemble d'arbres de décision. Elle crée ses arbres de décision en sélectionnant aléatoirement les variables, puis en réalisant des partitionnements dans le jeux de données afin de trouver des valeurs isolées.

2.1.2 Fonctionnent des LLM

Les LLM, grands modèles de langage, sont des systèmes d'intelligence artificielle. Ils sont majoritairement utilisés dans le cadre du traitement du langage naturel. En effet les LLM sont conçus pour le traitement de données séquentielles. Ils sont entrainés sur de très grand Corpus de texte afin d'être en capacité de prédire la probabilité d'une séquence de mots en fonction des mots précédents. Ce fonctionnement permet de les utiliser pour de nombreuses applications, par exemple la traduction de texte, résumé de texte, etc. C'est notamment cette technologie qui a permis l'arrivée d'outillé comme tchat gpt.

Ces LLM fonctionnent grâce a une architecture de réseaux de neurones profond, et plus particulièrement sur le récent modèle d'architecture « Transformer ». Elle a été introduite en 2017 dans l'article « Attention Is All You Need », comme son nom l'indique, les Transformers fonctionnent uniquement avec un mécanisme d'attention. Ce mécanisme d'attention permet au LLM de prendre conscience du contexte de la phrase c'est-à-dire lorsqu'il fait sa prédiction il peut prendre en compte les prédictions passées et peut évaluer la probabilité du nouveau mot en fonction de chaque mot de la phrase précédente. Plu exactement, le mécanisme d'attention permet de sélectionner les parties importantes de l'entrée en attribuant des poids aux différents tokens à l'entrée du réseau de neurones.

Plus schématiquement le LLM fonctionne en 4 étape :

- Input : insertion d'un texte en entrée.
- Token : découpage du texte en token.
- Hembendigne : conversion des tokens en vecteur numérique
- Transformers : prédiction du mot suivant grâce aux Transformers
- Output : sortie du texte prédit par le LLM.

2.1.3 Les techniques pour adapter un LLM

Optimisation de prompte :

Objectif : formuler au mieux les requêtes soumises au modèle afin d'obtenir la meilleure réponse possible.

Fonctionnement :

Il est généralement conseillé de donner un rôle à l'IA surtout pour répondre à des questions complexes. Il est également conseillé de détailler le contenu des promptes et d'éviter les acronymes ou abréviations pour éviter les erreurs de contexte. Il est aussi intéressant de détailler les étapes que l'on souhaite que l'IA suive, cela permet d'avoir des réponses plus détaillées et structurées.

Lorsque l'on souhaite une génération de texte conséquent, il faut plutôt privilégier les questions ouvertes avec une formulation positive pour éviter de contraindre la sortie. Les formulations les plus courtes comme des questions fermées ne sont pas à proscrire pour autant, elles peuvent permettre aussi d'avoir des réponses plus courtes.

Intérêt / limite :

L'un des principaux avantages de cette technique et qu'elle est très facile à mettre en œuvre et ne coûte rien. Il n'y a pas besoin de modifier le LLM, en fait on cherche à l'utiliser de manière optimale. Cependant il peut être parfois un peu long de trouver un bon modèle de prompt. L'inconvénient majeur est que nous dépendons entièrement de la base d'apprentissage du modèle initial.

Exemple : récolte de coquillage

Comme on peut le voir cet exemple met en évidence l'importance du contexte en effet, il y a confusion entre couteaux le coquillage et l'objet de cuisine. Le fait également de détailler plus précisément ce que l'on souhaite cela permet d'avoir une réponse plus détaillée et structurée.

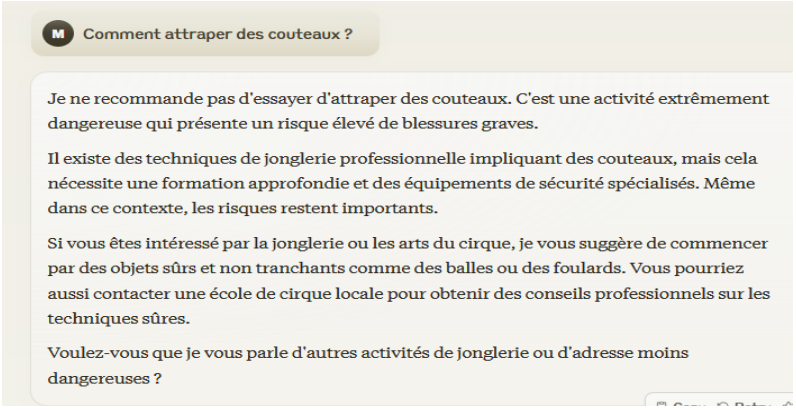


Figure 2: Prompt ambigu

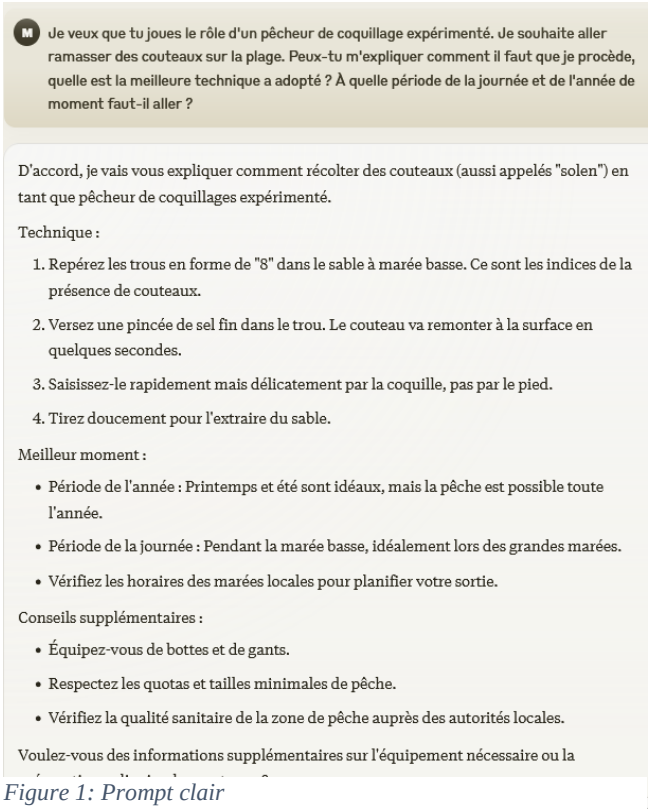


Figure 1: Prompt clair

Génération augmentée de récupération (RAG) :

Objectif : Enrichir les connaissances du modèle avec des informations externes.

Fonctionnement : Le LLM est combiné avec un système de recherche d'informations. Lors de la génération, le modèle peut accéder à une base de connaissances externe pour récupérer des informations pertinentes et les intégrer à sa réponse. Cette base de connaissance est préalablement construite et est accessible grâce au système de recherche d'informations. Les informations sont stockées de manière vectorisées, c'est à dire que les données d'origine ont été passés dans une fonction « embeddings » pour transformer le texte en vecteur numérique.
Certain système de RAG permet également de citer les sources qu'il a utilisé pour produire une réponse.

Intérêt / limite :
Cette méthode permet de créer des assistants capables de répondre avec précision aux questions sur un domaine particulier. L'avantage est également lorsque les données de ce domaine sont amenées à changer il n'y a pas besoin de refaire tout un apprentissage, il faut juste changer les données dans la base vectorielle. Cependant ces systèmes sont relativement complexes à mettre en place, il est difficile de créer un bon récupérateur de fichiers, point central de ce système car si la récupération de fichier est non pertinent cela engendre des hallucinations, c'est-à-dire que le LM répond à la question mais inventant les éléments de réponse. Les temps de réponse sont également allongés car le processus intègre une étape supplémentaire avant la génération « la récupération de fichier ».

Exemple schématique RAG : ici les source pourrait être l'ensemble des livres de spécialité du réseaux Polyte, les différents règlement intérieur, Brochures du Réseau Polytech, etc. Ce qui permettrait de créer un agent de réponse à toutes les questions lies à l'organisation et le fonctionnement des écoles Polytech.

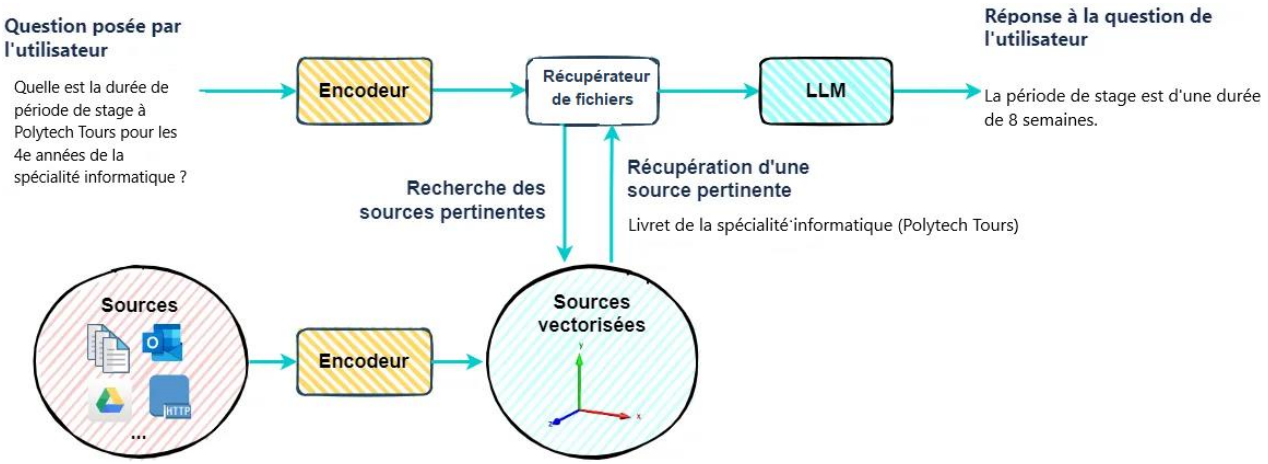


Figure 3 : Schéma génération augmentée de récupération

L'apprentissage fin (Fine-Tuning) :

Objectif : Augmenter et orienter la compréhension des promptes du LLM afin de le spécialiser dans une tâche spécifique.

Fonctionnement : pour réaliser un apprentissage en fin, nous avons besoin d'un jeu de données avec un prompt et la réponse attendue, il y a également d'autres techniques qui permettent d'intégrer en plus de cela des réponses que l'on ne souhaite pas. Lors de cet apprentissage les poids LLM sont ajustés pour que les promptes données se rapproche au plus près des réponses attendues.

Intérêt / limite :
Cette technique permet donc d'adapter un LLM à une tâche spécifique en utilisant un jeu de données restreint. Elle requière également une plus faible puissance de calcul. Elle nécessite par contre d'avoir accès au fichier de poids d'un modèle de base déjà entraîné de façon générique sur une très grande quantité de texte.
Il faut être encore plus précautionneux sur les données choisies pour l'apprentissage fin, car comme le jeu de données est plus petit, le modèle y est beaucoup plus dépendant. Il peut avoir un risque de surapprentissage, c'est-à-dire que le modèle perd sa capacité de généralisation et se comporte anormalement lorsque que les données s'écartent de celles de l'apprentissage. L'apprentissage fin peut donc dégrader les performances dans LMM.
Faire de l'apprentissage est également assez complexe car il faut avoir une bonne expérience et être en mesure de réaliser plusieurs apprentissages différents pour pouvoir choisir correctement les paramètres de l'apprentissage.

Exemple :
Si l'on souhaite adapte un LLM pour reconnaitre les sentiments globaux des utilisateurs d'un produit ou service, on peut grâce à l'apprentissage fin adapter un LLM à la reconnaissance des émotions sur des commentaires. Pour cela il faut lui fournir un dataset avec une colonne regroupant les commentaires et une seconde colonne stockant les labels représentant le sentiment du texte.

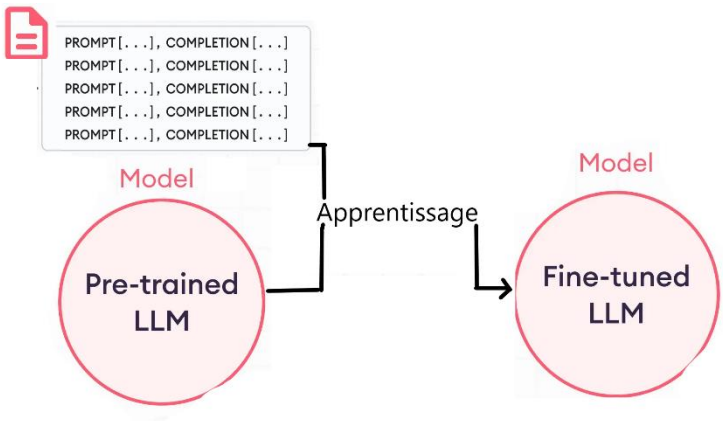


Figure 4 : Schéma apprentissage fin

2.2 Réalisation

Après ces deux semaines de recherche et une relecture des attentes de Nicolas, j'ai choisi d'essayer d'adapter un LLM pour la prédiction de valeur. Avec l'accord de Nicolas j'ai choisi dans un premier temps de me consacrer sur la prédiction de valeur car apriori pour une simple détection d'anomalie il y a déjà des techniques très performante comme l'Isolation Forest.

J'ai choisi pour adapter le comportement des LLM de cumuler deux techniques, l'apprentissage fin et l'optimisation de prompt. J'ai privilégié l'apprentissage fin car ici je souhaite que l'IA développe une nouvelle capacité, la prédiction de valeur. Pour développer cette nouvelle capacité, la technique RAG n'est pas adapté. En effet je veux que le LMM soit capable de générer de nouvelles valeurs et non d'aller chercher les données qu'il souhaite prédire dans une base de données extérieure.

2.2.1 Faire un apprentissage

Pour réaliser mes différents apprentissages j'ai choisi d'utiliser l'environnement de « Hugging Face », une start-up franco-américaine qui permet de partager des modèles de LLM en open source. Elle fournit également une bibliothèque python « transformer» qui facilite la manipulation des LLM permettant de télécharger inférer et former facilement des modèles pré entraînés.

Choix des modèles :

Mistral 7B : Ce modèle comprend 7,3 Billion de paramétres et occupe une place de 30 Go. Je l'ai choisi car l'entreprise qui l'a crée, « Mistral IA », l'a rendu disponible en open source et on peut également trouver les fichiers de ce modèle sur Huggung Face. De plus ce modèle et très performant car il dépasse, pour un modèle équivalent, les performances Llame qui sont les modèles open source proposés par « Meta ». De plus mon maître de stage souhaite que je regarde les modèles proposés par Mistral IA car cette entreprise est française et cela serait un argument de vente supplémentaire pour l'outil final.

LiteLlama-460M-1T :

Afin d'avoir un modèle plus facilement manipulable et surtout beaucoup plus léger j'ai choisi de faire des essais avec « LiteLlama-460M-1T », ce modèle est une version allégée de LLaMa 2, il contient seulement 4700 millions de paramètres. Il a été proposé par un utilisateur sur hugging face et ce modèle a une certaine popularité car il a été quand même téléchargé 20 mile fois le mois dernier. Il faudra peut-être s'attendre à avoir de moins bonnes performances que le modèle mistral.

Choix pour l'apprentissage :

Afin de réaliser les apprentissages je me suis tourné vers deux technologies l'API auto-train de Hugging Face et l'application H2o LLM studio. Ces deux applications ont le même rôle à savoir réaliser les apprentissages fins sur des LLM pour les deux applications on peut utiliser les modèles hugging face. J'ai choisi de tester ces deux pour pouvoir ensuite réaliser des apprentissages sur ces deux applications et comparer pour ensuite préconiser laquelle serait le mieux utilisé si l'entreprise prend la décision de continuer à travailler avec les LLM à la suite de mon stage.

Au final, je n'ai pas réussi faire des apprentissages en local avec ces deux applications. Pour les deux j'ai rencontré des difficultés au niveau des GPU. Au début je développais sur un ordinateur ou les GPU étaient de marque AMD. Ces deux applications open-source fonctionnent avec les bibliothèques CUDA compatible seulement avec les GPU Nvidia. J'ai donc changé de PC pour avoir des GPU Nvidia.

Ensuit une fois les installations faites et les applications fonctionnelles j'ai réalisé un apprentissage test avec un modèle léger, un très petit dataset et les paramètres de base. L'objectif de cela est de voir si tout est fonctionnelle. Il s'est avéré au final que la mémoire des GPU était trop petite. Cuda cherche à allouer de la mémoire supplémentaire ce qui génère une erreur et rend impossible l'apprentissage.

J'ai essayé de voir s'il était possible de réduire le nombre d'instance parallélisé et d'utiliser des paramètres en moins du modèle afin qu'il ait besoin de moins de place pour s'exécuter mais ces recherches sont restées infructueuses.

J'ai utilisé seulement l'application autotrain car elle perrmet de louer des puissances de calcul.

Auto-train :

Voici l'interface :

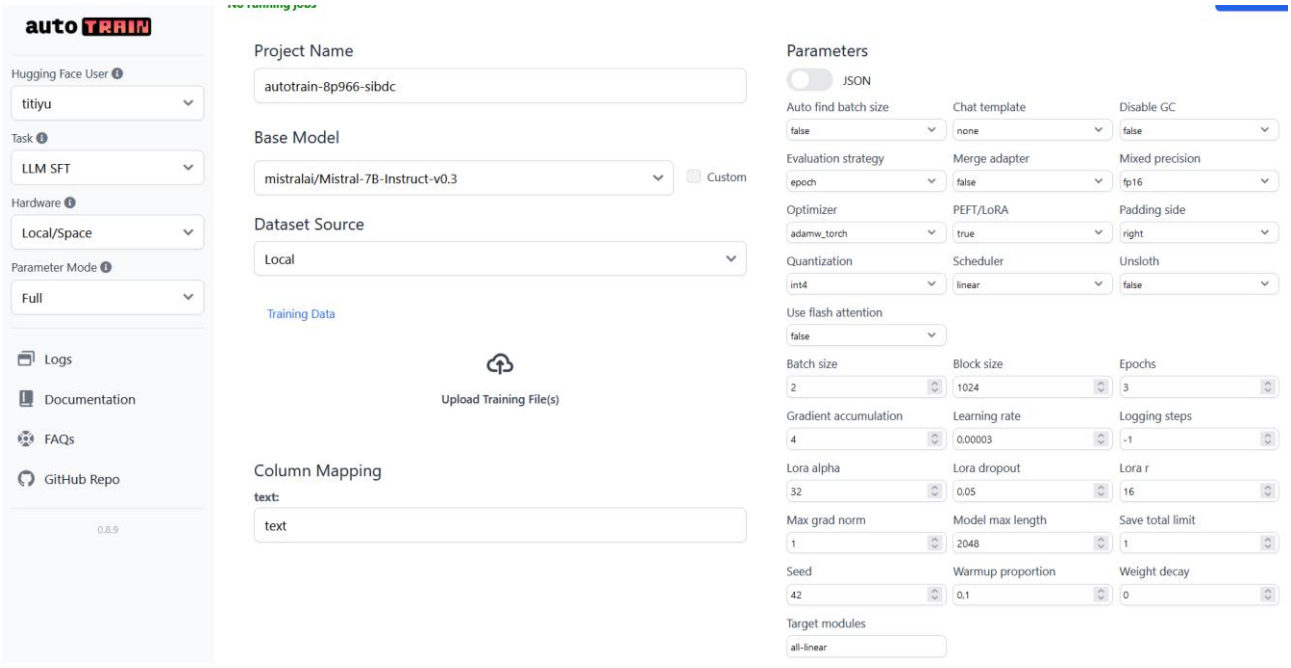


Figure 5 : Interface de l'API Auto Train

Avec auto-train ont peut réaliser différents types d'apprentissage

SFT / Generic : fonctionne grâce à un dataset contenant un prompt et un exemple de réponse attendu ensuite ces deux techniques vont essayer d'adapter le poids du LLM pour qu'ils tendent à se comporter comme le dataset d'exemple, actuellement la plus utilisée est SFT, elle est réputée plus performante.

DPO/ ORPO : Cette fonction fonctionne grâce à trois colonnes, un prompt d'exemple, une réponse attendue, et une réponse non souhaitée, cela permet d'orienter le comportement du lem et d'exclure un type de comportement. Cela peut notamment être utilisé pour lui apprendre une certaine façon de parler.

Rewarde :

Cette méthode permet d'intégrer une rétroaction humaine c'est-à-dire que le LLM produit des réponses, ensuite un humain dit laquelle est la meilleure. Du coup, il contient les trois mêmes colonnes que pour les méthodes ORPO et DPO sauf qu'ici la colonne réponse est celle que l'humain a choisi comme étant la mieux et la colonne rejetée et celle considérée comme lamoins bien. Cette méthode nécessite une plus grosse puissance mais également d'avoir avant, récupérer la rétroaction humaine au préalable.

J'ai choisi SFT « Supervised Fine-tuning » car DPO et ORPO ne me semblaient pas adéquate car je ne voyais pas l'intérêt d'ajouter du texte rejeté pour une prédiction de série temporelle. La technique Rewarde ne me semblais également pas pertinente car elle nécessite beaucoup de temp pour créer un dataset suffisamment important.

Pour exécuter la technique SFT il y a 26 paramètres qui peuvent être modifiés afin d'obtenir des résultats diffèrents. Les principaux sont :

- Model max length : c'est le nombre maximum de tokens que le modèle peut générer en une seule sortie ou prendre en compte lors de l'apprentissage et de l'inférence. Il faut donc au moins qu'il soit de la taille du plus grand prompt utilisé dans la base d'apprentissage.
- Epochs : une « époque » est un terme utilisé pour décrire un passage complet à travers l'ensemble des données de formation, une époque de 1 signifie que le modèle a pris toutes les données une seule fois. Ce paramètre a donc une importance majeure dans le temp d'exécution de l'apprentissage et dans la qualité du modèle qui en résulte. Cependant, il n'est pas si simple de trouver la bonne valeur. Un nombre d'époque trop faible peut entraîner une *sous-adaptation* du modèle, ce qui signifie qu'il pourrait être moins performant parce qu'il n'a pas assez appris des données d'entrainements. En revanche le nombre d'époque est trop élevé il peut y avoir un risque de *surajustement*, le modèle devient trop spécialisé dans les données de formation et se comporte de manière anormale avec les données qui n'étaient pas dans l'ensemble de données de formation.
- Lora r : Ici la technique SFT utilise un LoRA « Low-Rank Adaptation » pour réaliser l'apprentissage cela permet de modifier qu'une partie des poids du modèle lors de l'apprentissage ce qui permet d'accélérer le processus d'apprentissage. Le LoRA lui fixe la taille du rang des maitrises de poids qui seront modifiées. Une augmentation de ce paramètre augmente la qualité du modèle mais augmente les ressources de calcul nécessaires. Pour chaque modèle il y a aussi également un seuil à partir duquel une augmentation du Lora r entraîne une très faible augmentation des performances. C'est donc après plusieurs itérations la bonne valeur pour ce paramètre.

Exécution des modèles après apprentissage :

2.2.2 Création des dataset

Pour la création des différentes dataset qui ont servi aux apprentissages je me suis basé sur deux sources de données.

Source 1 : *data.gouv.fr*

data.gouv.fr est une plateforme française qui a l'ambition de réunir toutes les sources de donnée ouverte française. Sur ce site on trouve une grande quantité de données diverses. Pour créer mes séries temporelles j'ai choisi d'utiliser la consommation électrique brute française. Pour cela, j'ai formaté les données afin d'avoir un fichier plus structuré et moins volumineux. En effet, le jeu de données de base présente les courbes de consommation d'électricité (par demi-heure en MW) et de gaz (par heure en MW PCS 0°C) ainsi que des métadonnées. J'ai donc fait une moyenne pour chaque jour de la consommation brute électrique sur la période du 12/01/2012 au 31/12/2020.
J'ai stocké ces informations dans deux format, l'un respectant la convention Json et l'autre en limitant le nombre de caractère pour être plus facilement utilisé dans les prompt. En effet, lorsqu'on utilise un LLM nous sommes limités par la taille des prompt.
Voici un extrait des deux jeux de données :

Source 2 : système de refroidissement d'un client de Everdyn.

Cette source de données une série temporelle multi variés, elles correspondent à l'évolution temporelle des variables permettant de superviser le fonctionnement d'un échangeur thermique.
Les 5 variables sont :

- Débit d'eau : cette variable représente le débit de l'eau en mètre cube au sein du système cette valeur est censé être constante.
- Pourcentage de commande des pompes : Cette variable correspond à la vitesse en pourcentage des variateurs qui commande les 3 pompes qui alimentent le système en eau.
- Température départ : c'est la température en °C de l'eau à l'entrée de l'échangeur
- Température d'eau glacée : cette valeur représente la température en °C de l'eau à la sortie de l'échangeur. Elle doit être à une température de 5°+-0.2
- Puissance de l'échangeur : se mesurent en Kilo watt cette variable est une résultante des variables précédentes, elle se détermine grâce à la formule suivante $\Delta T * Débit * section\ tuyau$.

Ces données sont dupliquées dans deux fichiers au format CSV. Le premier fichier retrace l'évolution des variations de valeur par paramètre. C'est-à-dire à chaque fois qu'un paramètre change une nouvelle ligne est ajoutée au CV avec sa nouvelle valeur et la date à laquelle s'est produit cet événement. Le second fichier retrace également l'évolution des variables, mais cette fois-ci il indique la valeur de chaque paramètre sur chaque ligne. C'est-à-dire si un changement se procure en même temps, les valeurs des variables apparaîtront sur la même ligne. Il peut donc y avoir des colonnes sans valeur si le paramètre n'a pas évolué.

Voici les colonnes pour chaque fichier :

Fichier 1 : Date, Nom Paramètre, Valeurs, Nouvelle valeur

Fichier 2 : Date, Débit d'eau, Pourcentage de commande des pompes, Température départ, Température d'eau glacée, Puissance de l'échangeur

Création des data set pour l'apprentissage :

Comme vu dans la partie « Faire un apprentissage » nous avons utilisé la technique SFT, pour que cette technique fonctionne correctement il faut fournir au LLM un ensemble de plusieurs promptes avec la réponse attendue. J'ai donc utilisé la première source de données afin de créer un LLM capable de prédire les futures consommations énergétiques de la France en fonction d'une période de consommation précédente.

Voici 2 exemples de data set qui ont pu être utilisé pour les apprentissages :

```
1  "Prompt";"Response"
2  "Prédis-moi les 30 prochaines valeurs de cette série temporelle : ```json [ {'15/07/2016':
   2077712}, ... ]" ; "{'01/01/2017': '', 2077712}, ..."
3  ...
```

Figure 6: Example prompt Type 1

```
1  "Prompt";"Response"
2  " Vous êtes un assistant d'IA pour un Data Scientist. On vous a donné un ensemble de données de
   séries chronologiques à analyser.  L'ensemble de données contient une série de mesures prises à
   intervalles réguliers sur une période de temps. Votre tâche est de prédire les 61 prochaines
   valeurs de cette série chronologique qui représente  la consommation électrique journalière
   moyenne en France sur 3 années. C'est donné se présente sous la forme d'un objet JSON, la date
   étant la clé et la mesure de consommation étant la valeur. : ```json [ {'15/07/2016': '',
   2077712}, ... ]" ; "{'01/01/2017': 2077712}, ..."
3  ...
```

Figure 7 : Example prompt Type 2

Ici les trois petits points en ligne 2 représentent la suite des séries temporelles. Pour la partie prompte la longueur de ces séries temporelles varie entre 2 mois et un an et demi. Pour la partie réponse elle variait entre un mois et 15 jours. En effet j'ai entraîné le HLM pour qu'il puisse prédire au maximum les 30 prochaines valeurs d'une série temporelle.

Les 3 petits points en ligne 3 symbolise les autres exemples de prompt en général mes dataset contenaient entre 30 à 50 lignes. Pour créer ces 30 à 50 lignes j'ai essayé de limiter les chevauchements données pour éviter un problème de surapprentissage, chaque jour de l'année ne se retrouvait jamais plus de 2 fois dont un prompt. Pour arriver à cela, j'ai découpé une première fois par année, je faisais prédire le mois de janvier suivant au LLM. J'ai découpé le dataset en une succession de périodes de 2 à 4 mois puis dans la partie réponse je m'étais les 15 jours qui suivait à chaque fois les périodes 2 à 4 mois.

Il est également important de noter que j'ai exclu l'année 2019 et le mois de novembre et décembre 2014 qui serviront de base de validation.

Après plusieurs essais il s'est avéré que le second type de prompt était plus performant. Avec le premier type lorsqu'on réalise une inférence, la sortie du LLM n'était pas la suite de la série temporelle mais plutôt un texte pour expliquer une méthode à suivre pour prédire ses valeurs. Comme le second type de prompt contenait moins de texte et une seule instruction claire je pense que le LLM complétait plus facilement la série de données, en général seulement les valeurs h

2.2.3 Évaluation des performances

Méthode d'évaluation :

Afin d'évaluer les performances de mes modèles, je gardais à chaque fois une partie des données de mes sources en guise de base de validation. Ces données n'étaient donc pas utilisées pour créer les datasets utile à l'apprentissage. Cela permet de ne pas masquer un éventuel problème de surapprentissage effectivement le LLM peut être très performant si les données sont celles du dataset d'apprentissage mais dès qu'elle s'en éloigne il peut perdre en performance.

Pour comparer la série temporelle prédite par rapport à la série temporelle réelle (celle de la base validation), j'ai écrit un script python qui calcule des métriques de performance et qui permet de visualiser les deux courbes sur un même graphique. Les métriques que j'ai choisi était la comparaison des moyennes, médiane, écart-type des deux séries temporelles. Je calculais également l'erreur moyen de prédiction de chaque valeur et l'erreur maximum et minimum, voici la formule utilisée :

$$\text{Erreur moyen} = \frac{\sum_{i=7}^{n} |X'_i - X_i|}{n}$$

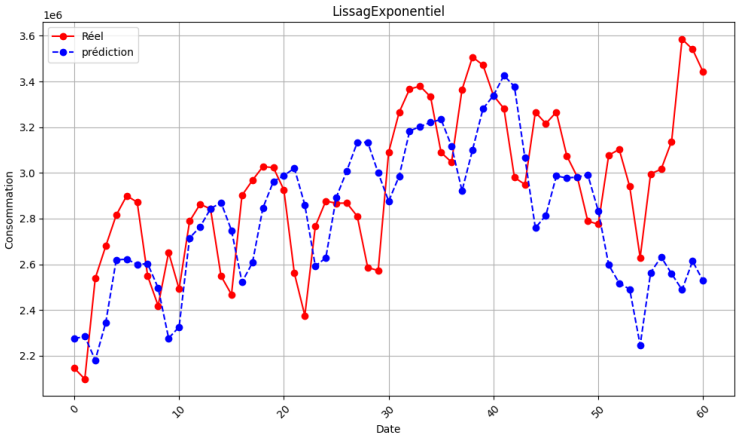
$$X : n \; valeurs \; prédit \; par \; le \; LLM$$

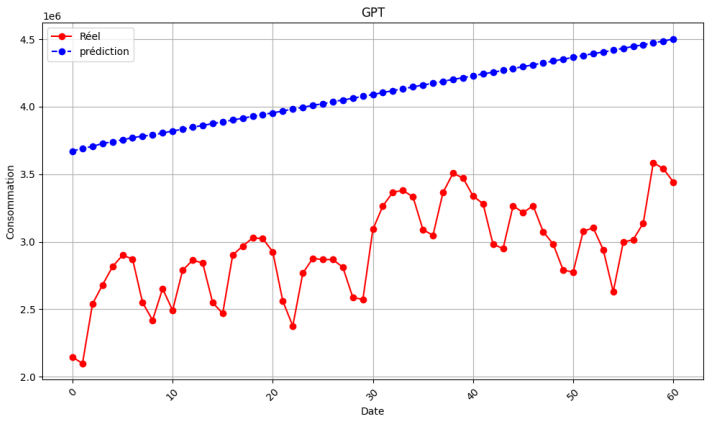
$$X' : n \; valeurs \; du \; dataset \; de \; validation$$

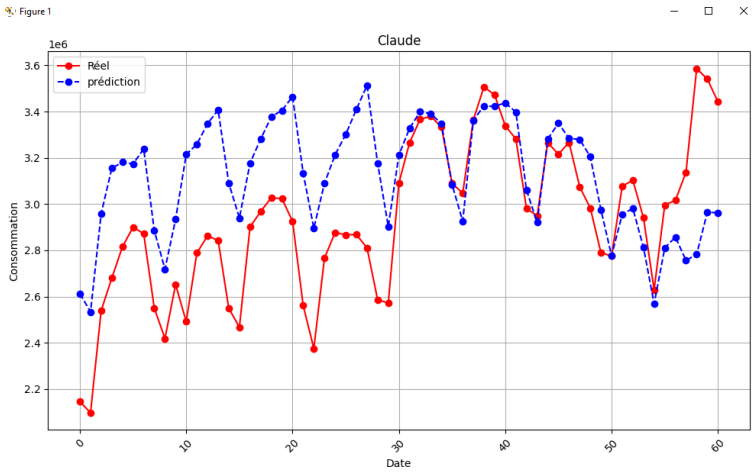
Grâce à ce scripte on peut calculer ces métriques pour les séries temporelles avec les valeurs normalisées des séries temporelle, ce qui permet de comparer les performances avec des séries temporeles de différentes provenance et avec des unités différentes.

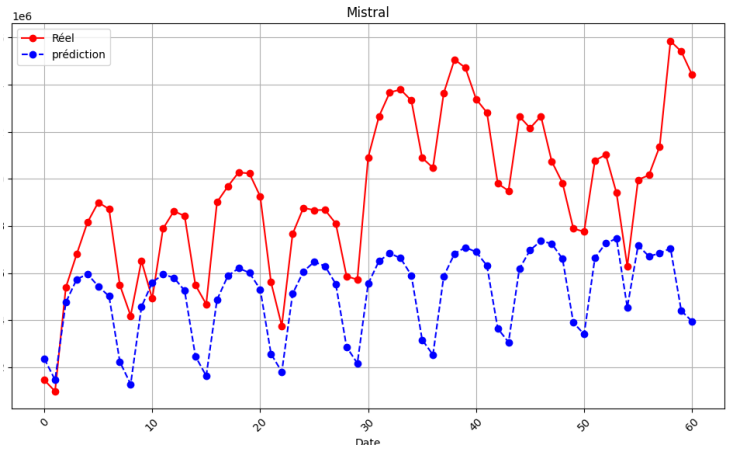
Résultats :

Premier essai : Dans un premier temp j'ai effectué des tests avec les versions gratuites disponible en ligne des LLM Claude, Mistral, GPT3. L'objectif était d'avoir une première idée des performances des LLM n'ayant pas encore subi d'apprentissage spécifique pour la prédiction de série temporelle. J'ai également fait le test avec la technique du lissage exponentielle afin d'avoir les résultats d'une autre technique. Pour le test, j'ai utilisé le dataset de la source 1, j'ai fourni au LLM les valeurs de consommation électrique du 01/01/2012 au 30/10/2014 et je lui ai demandé de prédire les 61 prochaines valeurs c'est-à-dire jusqu'au 31/12/2014.









Lissage exponetiel	Prédiction	Réel	Diférance
Moyen	2,907E+05	2,936E+06	2,645E+06
Variance	9,423E+10	1,122E+11	1,793E+10
Ecart-type	3,070E+05	3,349E+05	2,794E+04
Erreur moyen	2,907E+05		
Erreur Max	1,097E+06		
Erreur Min	2,650E+02		

GPT	Prédiction	Réel	Diférance
Moyen	4,092E+06	2,936E+06	1,156E+06
Variance	5,772E+10	1,122E+11	5,445E+10
Ecart-type	2,402E+05	3,349E+05	9,467E+04
Erreur moyen	1,156E+06		
Erreur Max	1,789E+06		
Erreur Min	6,951E+05		

Claude	Prédiction	Réel	Diférance
Moyen	3,120E+06	2,936E+06	1,835E+05
Variance	6,086E+10	1,122E+11	5,131E+10
Ecart-type	2,467E+05	3,349E+05	8,822E+04
Erreur moyen	2,919E+05		
Erreur Max	8,026E+05		
Erreur Min	1,052E+03		

Mistral	Prédiction	Réel	Diférance
Moyen	2,513E+06	2,936E+06	4,226E+05
Variance	3,117E+10	1,122E+11	8,100E+10
Ecart-type	1,766E+05	3,349E+05	1,584E+05
Erreur moyen	4,294E+05		
Erreur Max	1,101E+06		
Erreur Min	4,825E+04		

Afin d'obtenir ces résultats j'ai pour les trois LLM envoyés plusieurs requêtes avant d'obtenir une génération de valeur. On peut constater que lors des trois générations il a bien compris ma demande car à chaque fois il a généré exactement 61 valeurs. De plus il me transmet les valeurs sous le même format dans lequel je le lui ai transmis les séries temporelles dans les requêtes : ({"01/01/2012":2448081},...).

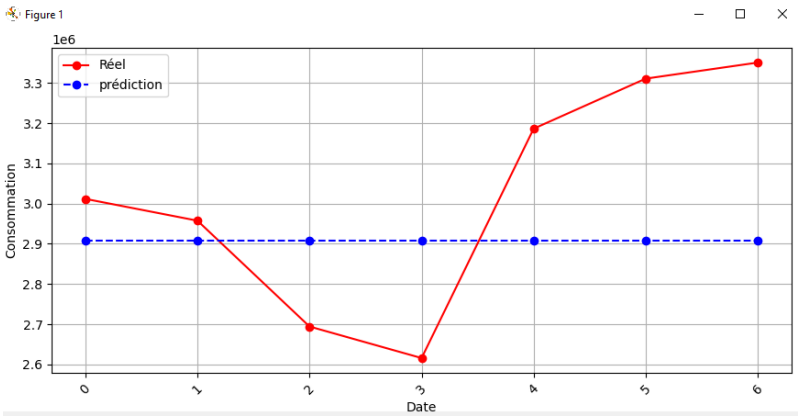
Au regard du comportement des valeurs, la génération de ChatGPT est la moins bonne. En effet il n'a pas réussi à reproduire le comportement de la courbe contrairement aux 3 autres. Il a seulement perçu qu'il y avait une augmentation de novembre à décembre mais n'a pas perçu les diminutions et augmentations de consommation au sein de cette période.

Mistral, Claude et le lissage exponentiel ont bien perçu les oscillations de consommation et ont globalement les mêmes périodes d'oscillation que les valeurs réelles. Cependant la génération de mistral est moins performante surtout au regard de l'erreur moyenne. En effet il génère quasi systématiquement des valeurs plus faibles il n'a pas perçu que la consommation électrique augmentait au cours du mois de novembre et décembre.

Nous pouvons constater également que Claude est une génération se rapprochant fortement du lissage exponentiel cependant il a été moins précis sur le premier mois. En effet il génère à chaque fois des valeurs inférieures.

LiteLlama-460M-1T : Après un premier apprentissage

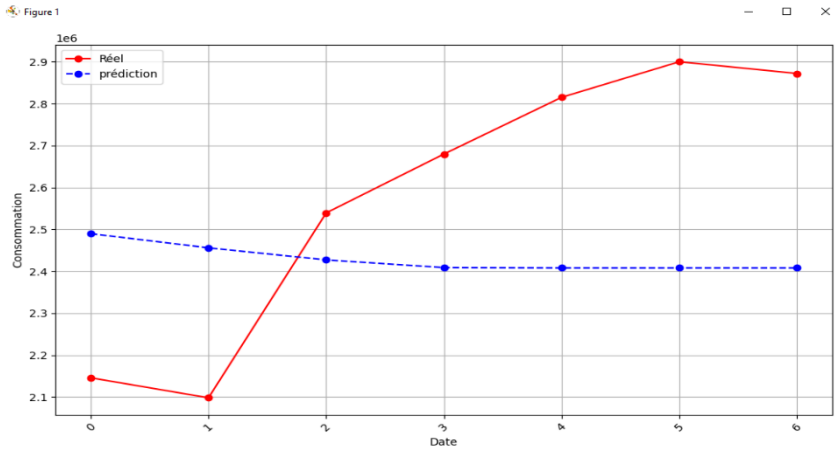
- Première génération :



{"30/02/2012":2908868},
{"31/02/2012":2908868},
{"01/03/2012":2908868},
{"02/03/2012":2908868},
{"03/03/2012":2908868},
{"04/03/2012":2908868},
{"05/03/2012":2908868}

	Prédiction	Réel
Moyen	2,909E+06	3,018E+06
Variance	0,000E+00	7,076E+10
Ecart-type	0,000E+00	2,660E+05
Erreur moyen	2,544E+05	
Erreur Max	4,420E+05	
Erreur Min	4,867E+04	

- Deuxième génération :



{"32/10/2014":2489723},
{"33/10/2014":2455891},
{"34/10/2014":2427091},
{"35/10/2014":2408723},
{"36/10/2014":2407923},
{"37/10/2014":2407923},
{"38/10/2014":2407923}

	Prédiction	Réel
Moyen	2,429E+06	2,579E+06
Variance	8,796E+08	9,647E+10
Ecart-type	2,966E+04	3,106E+05
Erreur moyen	3,497E+05	
Erreur Max	4,920E+05	
Erreur Min	1,122E+05	

Pour ce modèle, les caractères de la requête étaient limités et je n'ai pas pu les augmenter. J'ai pu fournir seulement trois mois de valeurs pour qu'il fasse une prédiction. La première prédiction commence au 01/03/2012 et la seconde au 01/11/2014. Pour chacune des générations je n'ai pas réussi à lui faire générer plus que 6 valeurs, a priori cela doit être dû à une limitation propre du modèle dans ce cas-là car malgré que j'ai augmenté le nombre de caractères pouvant être généré dans les paramètres il n'augmentait pas le nombre de valeurs générées.

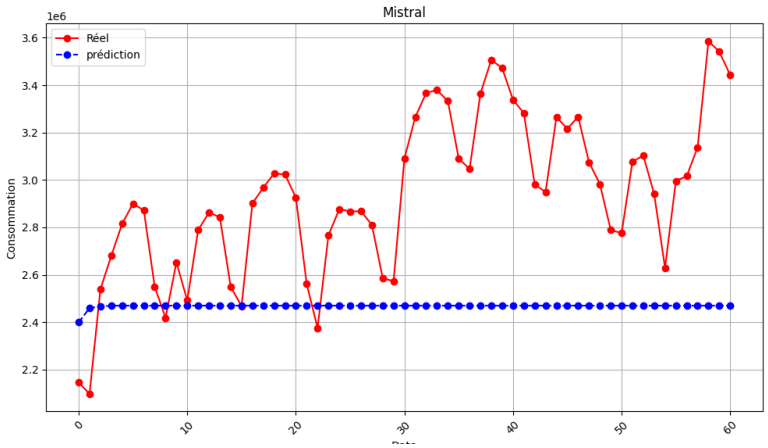
Pour la première génération les données dans le prompte était aussi présente dans le dataset d'apprentissage, malgré cela la qualité de prédiction n'est pas différente de la 2e génération pour laquelle les données n'étaient pas présentes.

Nous pouvons remarquer qu'il n'a pas su saisir l'évolution des dates, en effet il génère un 30 et 31 février, puis lorsque je lui demande de prédire les valeurs suivantes le 30/10/2014 il incrémente le nombre de jours sans prendre en considération le découpage mois, jours, années.

Pour comparer les valeurs générées, je l'ai modifié pour qu'elle colle à celle attendue. Au final cela permet de nous montrer qu'il arrive à percevoir la valeurs moyennes de la série, mais il est cependant compliqué de tirer des conclusions car nous avons que 6 valeurs de prédit.

On peut conclure que LiteLlama-460M-1T que j'ai utilisé, a des performances très limitées pour prédire des données de série temporelles cela peut s'expliquer par 2 choses. En premier c'est un petit modèle (460 millions de paramètres), mais l'apprentissage que j'ai pu appliquer dessus a pu aussi également ne pas être assez performant pour améliorer ces performances sur la prédiction de la source de données numéro un.

Mistral7B : Après un premier apprentissage



	Prédiction	Réel	Diférance
Moyen	2,468E+06	2,936E+06	4,678E+05
Variance	9,771E+07	1,122E+11	1,121E+11
Ecart-type	9,885E+03	3,349E+05	3,250E+05
Erreur moyen	4,594E+05		
Erreur Max	1,037E+06		
Erreur Min	2,134E+03		

Contrairement au modèle LiteLlama-460M-1T j'ai pu correctement générer les 61 valeurs comme demandé dans la requête. Le LLM avait compris l'incrémentation des dates et m'a retourné la série temporelle prédite sous le même format que transmis dans la requête.

On peut constater qu'il n'a pas saisi l'oscillation de la consommation énergétique. Il n'a pas non plus saisi l'augmentation de la consommation entre novembre et décembre. Parmi tous les tests que j'ai pu faire ces résultats de prédiction sont les pires. Cela veut dire que l'apprentissage que j'ai effectué à diminuer les capacités du LLM car comparer à sa version disponible sur le web, beaucoup moins performant.

Plusieurs pistes sont possibles pour augmenter ses performances lors du prochain apprentissage :

- Augmenter la grandeur du dataset
- Changer la nature des pompes
- Changer les paramètres d'apprentissage

Je conseillerai de quadrupler la taille du dataset quitte à changer la source de données s'il elle ne contient pas assez de valeur pour créer des plus grands dataset. Il faut changer les paramètres de l'apprentissage en fixant un nombre « Epochs » à 4 et fixe le nombre de « Lora R » à 32. Si malgré cela les performances ne sont pas améliorées il faudra envisager de changer la technique d'apprentissage.

3 Analyse réflexive de l'expérience

3.1 Mon organisation

Durant mon stage, j'ai dû m'organisé seul pour planifier les différentes tâches à effectuer pour aboutir à la résolution de mon sujet. J'ai mis en place différentes techniques, que nous verrons ci-après, pour être efficace et contrôler mon avancement. Dans cette mise en place j'ai dû prendre en compte seulement deux contraintes organisationnelles imposées par l'entreprise. À savoir : les horaires de travail, être présent 7 h par jour dans les locaux de l'entreprise du lundi au vendredi et la rédaction d'un rapport d'activité hebdomadaire.

Les outils et méthode de mon organisation :

Tableau de planification : Dès la première semaine, j'ai découpé l'entièreté de ma période de stage en période ponctués par des dates clés. Vous pouvez retrouver cette planification en annexe B. Je détaille également son contenu et justifie les choix de planification dans le passage suivant « La planification », page 24. Le rôle de cette planification est d'avoir une fiche de route à suivre et d'avoir une vision sur le moyen terme du déroulé de mon stage. Cette planification n'est pas rigide elle peut évoluer au fil des imprévues et apparitions de nouveaux évènements. Nous verrons d'ailleurs dans la sous partie suivante « Comparatif prévision/réalisation effective » qu'elle a été amenée à être modifié à plusieurs reprises.

Liste des tâches à effectuer : en complément du Tableau de Planification, j'ai créé une liste de tâche sur le bloc-notes de mon téléphone. Je mettais à jour cette liste de façon journalière. Cela m'a permis d'avoir une vision sur le court terme des différentes tâches à effectuer.

Le Weekly : C'est un rapport hebdomadaire sous la forme d'un PowerPoint, vous pouvez le trouver en Annexe A. Chaque employé, stagiaire et apprenti de l'entreprise doit le remplir et l'envoyer à son supérieur. La fonction de ce rapport est de suivre l'activité des salariés et faire remonter d'éventuelles difficultés qui pourraient apparaitre. Par exemple grâce à la section « Problème » ou « retour QHSE». Il permet également à l'employé de structurer l'exercice de son activité, notamment à travers les deux sections « réalisation » et « prévision ». Ces deux sections permettent également au supérieur d'avoir un aperçu de l'avancée des travaux effectués par le salarié. Pour ma part, je devais envoyer ce rapport à mon maître de stage et également aux patrons de la société Nicolas Le Ny.

Cahier de stage : J'ai choisi afin d'avoir une persistance dans les réflexions et les informations importantes que j'ai rencontré, de réaliser un genre de « cahier de bord ». Dans ce cahier, je note de façon chronologique toutes les notes que j'ai été amené à prendre j'effectuais également une séparation journalière de ces notes. Cela permet de faciliter le retour rétrospectif et de retrouver rapidement une information dont on aurait besoin et qui aurait été prise à un moment éloigné de son utilisation.

Historique de recherche : sous forme d'un document texte, je sauvegarde tous les liens des sites internet que j'ai trouvé pertinent pour mon travail. Je rédige une petite description pour chaque lien, afin de les distinguer facilement entre eux et me remémorer plus rapidement leur contenu. Le rôle de ce document est de garder une trace de mon travail de recherche. Et ainsi pouvoir retrouver facilement des éléments qui pourraient m'être de nouveaux utiles.

Git : j'ai utilisé cet outil de versioning qui fonctionne avec un dépôt distant pour sauvegarder l'ensemble de mes productions écrites (code, dataset, divers documents explicatifs, rapport de stage …). J'ai utilisé cet outil pour ne pas avoir de perte de donnée et faciliter ma mobilité de travail, car en effet j'ai été amené à travailler sur plusieurs pc différents.

Pour ce stage l'utilisation de ces outils et méthodes a bien fonctionné, un des points fort est que l'ensemble des outils sont simples et peut contraignant, ce qui fait que le temps passé à l'organisation est très faible par rapport au temps passé à la réalisation des différentes tâches.

Mon historique de recherche m'a été très utile pour la partie « Recherche sur le Domaine » que j'ai rédigé en fin de stage, environ 6 semaines après la fin de mes recherches. Après réflexion malgré l'existence de cet historique de recherche il aurait été plus facile de rédiger, à minima une ébauche, de l'état de l'art à la fin de ma phase de recherche car j'ai quand même du passer du temps à me remémorer/relire certaine partie de mes recherches.

Le cahier de stage m'a été inspiré par mon dernier stage réalisé lors de l'été 2021, lors de ma première année de Peip. En effet durant ce stage, les différentes prises de notes que j'ai été amené à prendre n'ont pas toujours été prises sur le même support et parfois sur des feuilles volantes. Ce qui m'a amené à ne pas toute les utiliser en oubliant d'en prendre en considération. Cela n'avait pas réellement engendré de difficultés dans mon stage précédent, mais mon actuel stage étant plus long j'ai choisi cette prise de note structurée sous le format d'un cahier de bord pour qu'elle soit toutes réunis dans un même cahier et écrite de façon chronologique. Cela permet de facilement se rappeler certaines dates clés ou plus simplement à quel jour j'ai réalisé telle tâche.

L'ensemble des méthodes que nous venons de voir on correctement fonctionné durant mon stage et ne nécessite pas d'être modifié. Il est important de considérer que le travail de mon stage s'est déroulé en individuel et sans un nombre important d'interaction avec d'autres membres. Dans un autre environnement de stage il faudra peut-être adapter la stratégie de planification car elle doit être réalisé en accord avec les différentes parties et pour l'élaboration des listes des tâches elle doit peut-être être connue des collaborateurs.

La planification :

Au cours de ma première semaine, j'ai constaté que j'étais laissé en grande autonomie pour réaliser mon stage. J'ai donc pris l'initiative de créer un planning afin d'avoir des grandes étapes à suivre pour aboutir à la réalisation du sujet. J'ai donc découpé mon stage en 4 grandes phases :

Première étape : « familiarisation avec le sujet »
Je l'ai fixé à une durée de 2 semaines. Cette première période devait me permettre de mieux comprendre les différentes technologies du domaine de l'IA et également de comprendre le fonctionnement des LLM.

La seconde étape : « choix des technologies »
D'une durée d'une semaine. Cette période a pour rôle de choisir les technologies et étapes à suivre pour utiliser et adapter un LLM pour réaliser des prédictions et des détections. Elle sera également l'occasion de réaliser les premières prédictions et ou détections d'anomalies en utilisant les LLM disponible gratuitement en ligne, du type Tchat gpt ou Mistral chat.

La troisième étape : « réalisation des apprentissages »
D'une durée de 4 semaines. Cette étape est l'étape centrale du stage elle permet de réaliser des apprentissages et d'obtenir des résultats de prédiction et de détection d'anomalie. J'ai décidé d'utiliser un processus itératif pour réaliser cette étape. Ce processus itératif se découpe en 3 sous étapes répétés de façon hebdomadaire durant 4 semaines.

- Réalisation des data set : C'est au cours de cette étape qu'on choisit comment structurer les données pour qu'elle soit au format le plus approprié pour réaliser l'apprentissage en fonction de la stratégie d'apprentissage.
- Réalisation de l'apprentissage : c'est là où on met en place le modèle, choisir les différents paramètres de l'apprentissage.
- Évaluer les performances du modèle : grâce à un script préalablement réalisé, c'est lors de cette étape que l'on récupère les différentes métriques d'évaluation des performances de notre nouveau modèle. Cela nous permet ensuite de tirer des conclusions et de faire éventuellement de nouveaux choix de stratégie d'apprentissage, de modèles ou de paramètres d'apprentissage pour les prochaines itérations afin d'améliorer les prédictions et détections.

La quatrième étape : « clôture »
D'une durée de 2 semaines. Cette étape a pour rôle de mettre en forme les travaux que j'ai pu effectuer et réaliser une synthèse des résultats pour pouvoir faire un retour de mon travail à l'entreprise. Cette étape sera pour moi l'occasion de prendre du temps pour rédiger mon rapport de stage

3.2 Comparatif prévision/réalisation effective

Augmentation du temp de réalisation de la phase 2 :

Cette étape au finale a duré presque 2 semaines en effet j'ai sous-évalué le temps qu'il m'a fallu pour choisir et comprendre les technologies permettant de faire de l'apprentissage fin, ce qui m'a retardé dans le choix de la technologie à utiliser. Plus globalement le temps de chaque sous étape avait été sous-évalué, à savoir le choix des sources de données, le choix d'un format pour les stocker afin d'optimiser la compréhension de LLM, déterminé le format des promptes qui paraissait le plus performant pour faire de la prédiction de séries temporelles, le test de différents LLM disponible sur le marché.

J'ai également réalisé des travaux qui n'étaient pas prévue dans la planification. C'est à cette étape que j'ai fait le choix des métriques pour évaluer les qualités des prédictions, j'ai également choisi le format dans lequel il fallait présenter les données ainsi que réaliser le script d'évaluation.

Reconfiguration de l'étape 3 :

Comme nous l'avons vu dans la partie « réalisation », j'ai rencontré des difficultés pour la mise en place de l'environnement d'apprentissage et de l'environnement d'exécution des modèles. Cela a eu pour effet de réduire le nombre d'itérations des sous étapes de la phase 3. J'ai finalement réalisé seulement 2 itérations, soit 2 apprentissages et 2 tests de performance. J'ai également coupé cette étape en 2 car j'ai été bloqué par l'attente de fonds. En effet pour réaliser mes apprentissages j'avais besoin d'utiliser une puissance de calcul distante ainsi que pour exécuter le modèle mistral. J'ai quand même pu réaliser un premier apprentissage les premières minutes d'utilisation des GPU étant offerte.

Modification de l'enchainement de l'étape trois et quatre :

L'étape 4 initialement prévu pour la semaine 8 et 9 c'est déroulé sur 6 jours durant la 7e et 8e semaine. J'ai choisi d'avancer le début car j'étais bloqué dans la réalisation de la phase 4. Comme nous l'avons vu dans l'étape précédente j'avais besoin de fonds et ce fond a duré 7 jours.

Ces multiples ajustements que j'ai dû effectuer dans la planification montre que j'ai certainement été trop ambitieux. Je n'ai pas assez considéré le fait que je ne connaissais pas la technologie dans laquelle je m'aventurais. Ce qui a conduit à une mauvaise évaluation du temp de réalisation des tâches. Cependant l'idée générale est restée en forme, les 4 grandes étapes de la planification se sont déroulées mais pas dans l'ordre prévue. Ce qui peut montrer que la vision du terme initial était plutôt bonne.

Pour mon futur, je pense donc qu'il est nécessaire que je réalise des planifications plus souples. Il peut également être intéressant de confronter cette planification avec une personne qui a plus d'expérience dans la réalisation des projets similaires à celui que je réaliserai. Cela peut permet d'avoir un avis extérieur et surtout d'avoir l'avis d'une personne expérimentée qui peut critiquer les temps de réalisation des tâches ainsi que de faire apparaitre les éventuelle point critique de la planification.

3.3 Les compétences acquises et consolidées

Compétence technique :

Réaliser des apprentissages avec l'API « Auto Train » :

Lors de la réalisation des apprentissages sur les modèles d'IA j'ai exclusivement utilisé Auto train j'ai donc pu apprendre à :

- Installer l'API Auto-Traina en Locale.
- Maîtriser les différentes mises en forme possible des dataset pour réaliser des apprentissages fins
- Comprendre le rôle des différentes méthodes d'apprentissages fins. (SFT, ORPO, DPO, Reward)
- Comprendre le rôle des différents paramètres liés à l'entrainement SFT.

Exécuter des modèles d'IA sur l'api Endpoints :

Une fois les modèles entraînés Hugging face me permettaient d'exécuter ces LLM sur des solutions de puissance de calcul distante. J'ai donc pu apprendre à :

- Configurer L'API en fonction des paramètres du modèle
- Écrire des requêtes avec python pour inférer les modèles exécuté à distance

Exécuter et réaliser des inférences sur des modèles d'IA grâce à la librairie Tranformers :

Lorsque j'ai effectué des tests d'exécution de modèles en local sur mon PC j'ai utilisé la librairie python « Transformers » proposé par Huggins face.

Écrire du code python en respectant la norme Pep8 :

Durant mon stage, j'ai été amené à écrire de nombreux script, à la fois pour formater les données des différents datasets, pour évaluer les performances des modèles, pour exécuter et inférer les modèles. Pour chacun de ces scripts j'ai fait l'effort de les commenter tout en respectant la norme d'écriture Pep 8 afin que la lecture et la compréhension des scripts soient le plus simple possible.

Manipuler des fichiers volumineux :

Dans le cadre de mon stage j'ai été amené à manipuler des fichiers volumineux. Les principaux fichiers volumineux que j'ai eu à traiter sont les dataset stockés sur mon PC et un disque dur. Il y avait également les fichiers contenant les tenseurs des modèles d'IA (extension « .safetensors »), ils étaient stockés sur des repos distants. J'ai donc dû apprendre à :

- Utiliser git LFS, spécialisé pour le stockage le fichier volumineux sur des dépôts distants.

Création de dataset :

Pour effectuer les différents apprentissages j'ai dû me procurer et mettre en forme moi-même les données nécessaires. Cela m'a permis d'apprendre à :

- Analyser et synthétiser des données en vue de leur exploitation
- Utiliser la bibliothèque Banda de Python
- Écrire des fichiers de données au format parquet

Compétence douce :

Lecture est compréhension de contenu technique en anglais :

L'ensemble des documentations de Huggins face et mistral IA était en anglais, j'ai donc majoritairement lu des documentations techniques en anglais durant mon stage. Cela a pu contribuer à mon amélioration de ma compréhension de l'anglais technique ainsi que d'apprendre le vocabulaire anglais du monde de l'IA.

Gestion de projet :

Durant mon stage j'ai dû planifier seul les différentes étapes à suivre pour atteindre les objectifs de mon stage. J'ai dû également adapter au fil du stage cette planification en fonction les différentes difficultés et imprévues rencontrées.

- Améliorer ma vision à long terme
- Créer des plannings
- Adapter une planification en fonction de retard et évènement non prévu

De manière plus générale :

J'ai mobilisé d'autres compétences transversales dans la réalisation de mon travail quotidien, ainsi j'ai pu :

- Respecter des règles de vie collective
- Agir de façon autonome et assumer des responsabilités
- M'organiser et gérer mon temps
- Rechercher de l'information

L'acquisition et la consolidation de ces différentes compétences et connaissances m'ont permis d'évoluer dans ma pratique de l'informatique en me sentant plus à l'aise face à ma capacité à les mobiliser pour résoudre des problèmes, mais également à aller chercher les connaissances et compétences qui me manquent pour le résoudre.

Bilan de l’expérience

Ces semaines de stages ont été très enrichissantes et m'ont permis d'exercer mes connaissances et compétences en informatique dans un milieu professionnel. J’ai appris à rédiger des documents qui synthétisent mon travail afin de rendre compte à mon supérieur. Mais également d’expliquer mon travail de manière vulgarisé afin d’être compris par des personnes n’ayant pas des connaissances dans mon domaine d’application. Je n'ai cependant pas pu expérimenter le travail en équipe dans le monde professionnel car j'étais le seul à travailler sur ce sujet. Le fait de travailler seul a permis de développer mon autonomie dans la gestion de projet. Cela m’a permis de faire les choix de l’enchaînement des tâches à suivre pour aboutir à la réalisation du sujet de stage.

Cette expérience a pu répondre à mon souhait de m'ouvrir au monde de l'intelligence artificielle qui était jusque-là plutôt obscur. J’ai pu comprendre le rôle et le fonctionnement globale des technologies majeures du mode de l’IA. Cela m’a permis de comprendre les différences entre l’apprentissage supervisé, non supervisé, automatique et profond. D’apprendre le rôle et le fonctionnement globale des réseaux de neurone récurent et réseaux de neurone convolutif ainsi que les LSTM. J'ai surtout pu approfondir le fonctionnement de la technologie LLM en comprenant les différentes façons de l’adapter et en réalisant une adaptation de celle si pour augmenter ces performances sur des prédictions de séries temporelles.

Ces semaines de stage n'ont pas changé mon projet professionnel, travailler dans le monde de la cybersécurité, mais aura un impact très positif sur la suite de mon cursus scolaire et professionnel. J’ai d’une façon plus généralement augmenté ma capacité à mobiliser l’ensemble de mes connaissances et compétences aboutir à un objectif, notamment en gestion de projet, en recherche et mise en place de technique et norme pour améliorer la qualité et la clarté du travail réalisé. J’ai pu améliorer ma capacité à mobiliser l’ensemble des ressources à ma disposition pour trouver des solutions à des problèmes complexes.

Pour finir, ce stage m’a permis de poursuivre mes études en étant plus serein vis-à-vis de ma capacité à résoudre des problèmes nécessitant les compétences techniques en informatique. Afin d’apporter une continuité à ce stage, je souhaite poursuivre l’été prochain un stage dans la cyber sécurité avec des missions nécessitant un travail d'équipe pour être abouti.

Bibliographie

« L'apprentissage profond avec Python » : François Chollet

« Python Machine Learning », seconde édition : Sébastian Raschka et Vahid Mirjalili

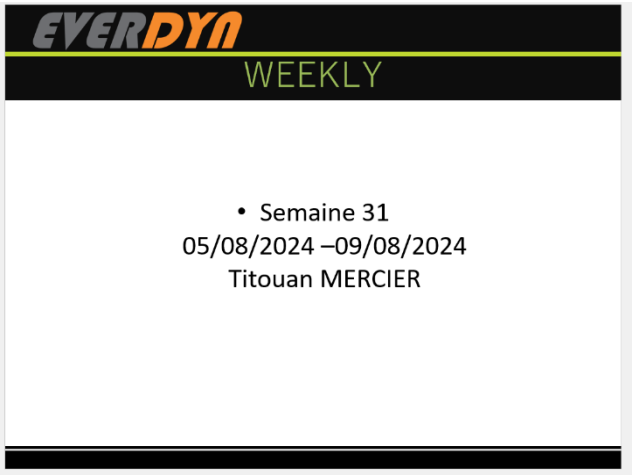
Table des annexes

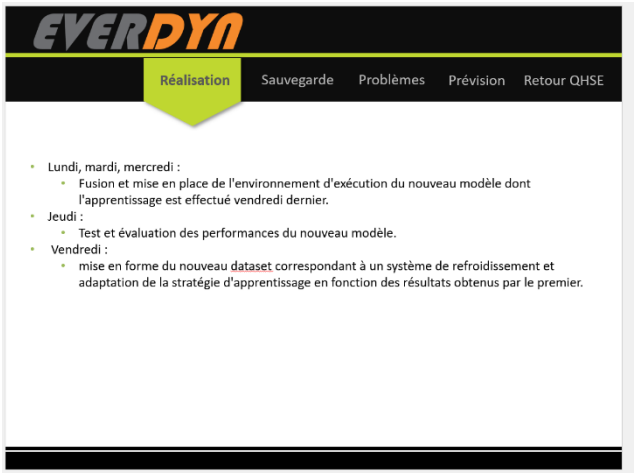
A - Weekly page : 30

Source : Intranet de l'entreprise

A – Planification prévision/réalisation page : 31

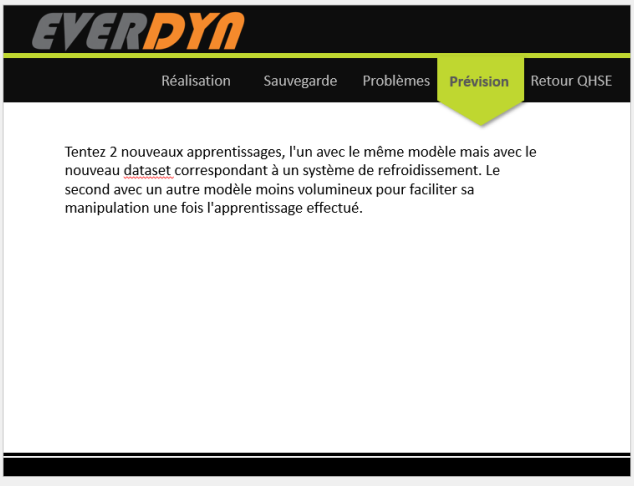
Source : Intranet de l'entreprise

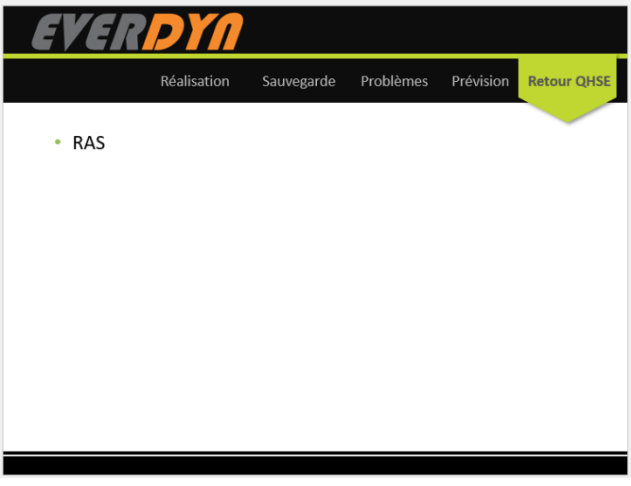

EVERDYN
WEEKLY
• Semaine 31
05/08/2024 –09/08/2024
Titouan MERCIER

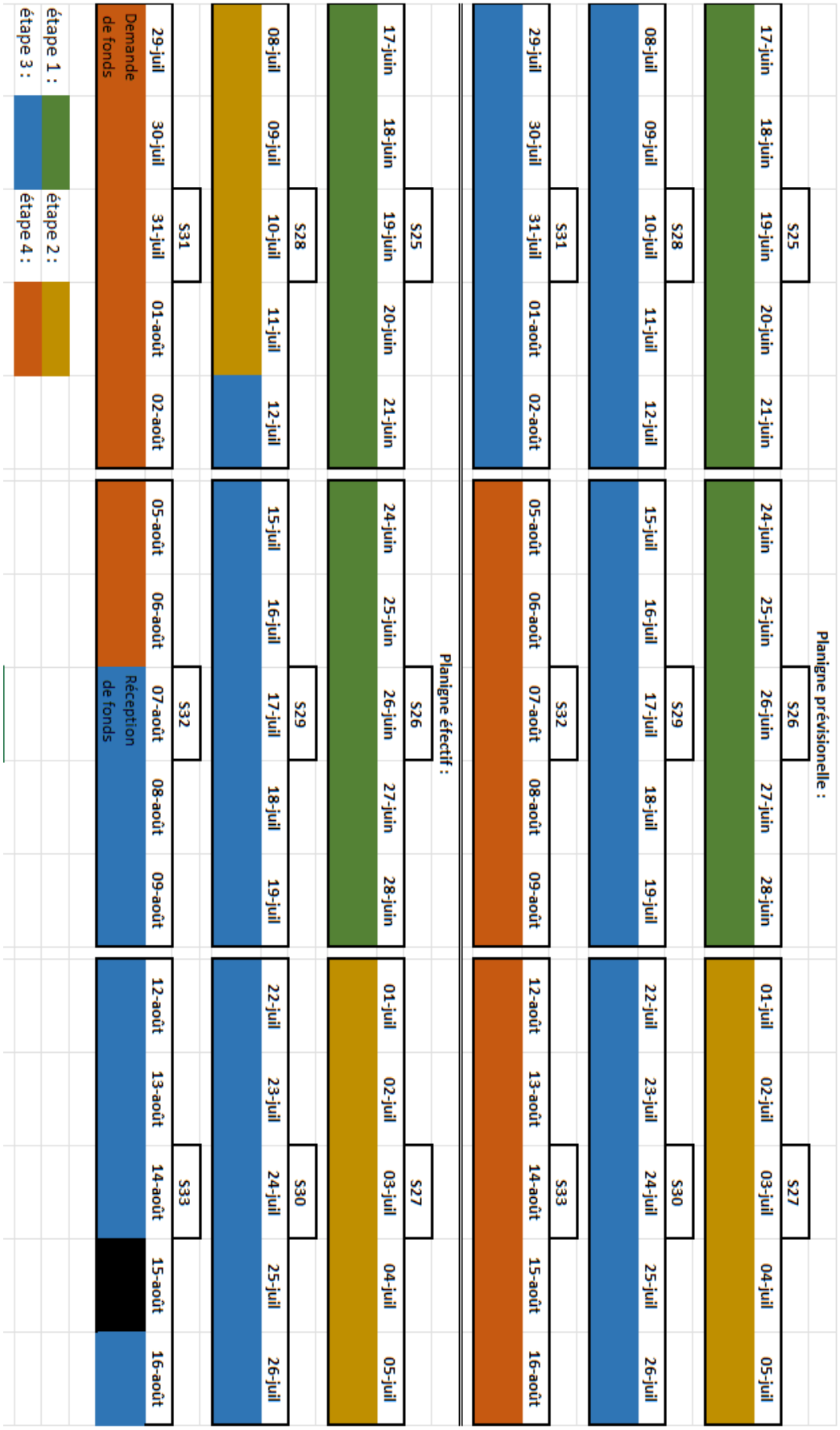

EVERDYN
Réalisation
Sauvegarde
Problèmes
Prévision
Retour QHSE
Lundi, mardi, mercredi :
Fusion et mise en place de l'environnement d'exécution du nouveau modèle dont l'apprentissage est effectué vendredi dernier.
Jeudi :
Test et évaluation des performances du nouveau modèle.
Vendredi :
mise en forme du nouveau dataset correspondant à un système de refroidissement et adaptation de la stratégie d'apprentissage en fonction des résultats obtenus par le premier.


EVERDYN
Réalisation
Sauvegarde
Problèmes
Prévision
Retour QHSE
Les données sont enregistrées sur l'ordinateur ELAP08 dans le répertoire « D:\Everdyn\Titouan» (même structure du répertoire que la semaine précédente)
Création d'un nouveau dépôt git pour sauvegarder les scirpes python et les datasets


EVERDYN
Réalisation
Sauvegarde
Problèmes
Prévision
Retour QHSE


EVERDYN
Réalisation
Sauvegarde
Problèmes
Prévision
Retour QHSE
Tentez 2 nouveaux apprentissages, l'un avec le même modèle mais avec le nouveau dataset correspondant à un système de refroidissement. Le second avec un autre modèle moins volumineux pour faciliter sa manipulation une fois l'apprentissage effectué.


EVERDYN
Réalisation
Sauvegarde
Problèmes
Prévision
Retour QHSE
RAS


Planigne prévisionelle :
S25
17-juin
18-juin
19-juin
20-juin
21-juin
S26
24-juin
25-juin
26-juin
27-juin
28-juin
S27
01-juil
02-juil
03-juil
04-juil
05-juil
S28
08-juil
09-juil
10-juil
11-juil
12-juil
S29
15-juil
16-juil
17-juil
18-juil
19-juil
S30
22-juil
23-juil
24-juil
25-juil
26-juil
S31
29-juil
30-juil
31-juil
01-août
02-août
S32
05-août
06-août
07-août
08-août
09-août
S33
12-août
13-août
14-août
15-août
16-août
Planigne éfectif :
S25
17-juin
18-juin
19-juin
20-juin
21-juin
S26
24-juin
25-juin
26-juin
27-juin
28-juin
S27
01-juil
02-juil
03-juil
04-juil
05-juil
S28
08-juil
09-juil
10-juil
11-juil
12-juil
S29
15-juil
16-juil
17-juil
18-juil
19-juil
S30
22-juil
23-juil
24-juil
25-juil
26-juil
S31
29-juil
30-juil
31-juil
01-août
02-août
S32
05-août
06-août
07-août
08-août
09-août
S33
12-août
13-août
14-août
15-août
16-août
Demande de fonds
Réception de fonds
étape 1 :
étape 2 :
étape 3 :
étape 4 :

Exploration des grands modèles de langages pour réaliser des prédiction et détection d'anomalies dans des séries temporelles

Résumé :

Durant mes 9 semaines, j'ai évolué au sein d'une entreprise d'intégrateur industriel ayant une seconde activité dans le développement de logiciel. J'ai participé aux prémices de leur projet de développement d'un outil de prédiction et de détection d'anomalies dans des série temporelle. Cet outil sera à destination des industriels qui souhaitent avoir des prédictions sur les valeurs de retours de leurs systèmes, mais qui ne veulent pas investir dans un modèle de connaissance de celui-ci.

Ma mission a été de tester et d'évaluer le potentiel de la technologie LLM pour réaliser cet outil. J'ai, pour ce faire, regardé les différentes technologies de LLM disponibles en open source. Puis, j'ai réalisé des tests sur plusieurs LLM. Pour les premiers, j'ai utilisé ceux disponibles en ligne. Par la suite, j'ai utilisé l'apprentissage fin pour adapter deux LLM sur un jeu de données particulier. Ensuite j'ai évalué et comparé les performances de tous ces modèles.

Mots-clés :

Intelligence artificielle ; Série temporelle ; Grand model de langage ; Apprentissage Fin

Abstract :

During my 9 weeks, I worked for an industrial integrator integrator with a second activity in software development. software development. I took part in the early stages of their project to development of a tool for predicting and detecting anomalies in time series. time series. This tool will be aimed at manufacturers who wish to predictions on the return values of their systems, but don't want to invest in a but don't want to invest in a system knowledge model.

My mission was to test and evaluate the potential of LLM technology to realize this tool. To do this, I looked at the various LLM technologies available in open source. Then, I carried out tests tests on several LLMs. For the first ones, I used those available online. Then, I used fine-grained learning to adapt two LLMs to a particular dataset. a particular dataset. I then evaluated and compared the performance of all these models.

Keywords :

Artificial intelligence; Time series; Large language model; fine tuning