République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option : *Génie Logiciel (G.L)*

Thème

Identification des paramètres d'amélioration du départ de la BMX Race

Réalisé par :

- **MEKHEZZEM Réda**

Présenté le 09 Septembre 2019 devant le jury composé de :

- *M. HADJILA FethAllah* *(Président)*
- *M. MATALLAH Hocine* *(Encadrant)*
- *Mme. AMRAOUI Asma* *(Examinatrice)*

Tuteur de stage :

- *M. MESMOUDI Amin* Maitre de conférences, LIAS - Université de Poitiers

Année universitaire : 2018-2019

Remerciements

En premier lieu, je tiens à exprimer les remerciements les plus sincères à mon encadrant M. amin MESMOUDI, maître de conférences à l'université de Poitiers pour ses précieux conseils, sa rigueur, son sérieux et ses encouragement. Je lui suis extrêmement reconnaissant pour le temps conséquent qu'il m'a accordé ainsi que pour l'attention qu'il a porté à mon travail.

Je tiens à remercier tout particulièrement M.Hocine MATALLAH, maître de conférences à l'UABT, pour ses conseils avisés, sa patience et sa réactivité. Ses qualités scientifiques et humaines, la disponibilité et la confiance dont il a fait preuve à mon égard m'ont donné une immense assurance et la capacité de mener ce travail à terme.

Je tiens à remercier M.Emmanuel GROLLEAU, directeur du LIAS, pour m'avoir accueilli au sein du laboratoire. Je tiens également à remercier l'ensemble du personnel du LIAS pour leurs accueil et je remercie l'ensemble des doctorants et stagiaires pour la bonne ambiance que nous entretenions au bureau.

J'adresse également mes remerciements les plus sincères à l'équipe de RoBioSS, pour m'avoir accompagné durant tout mon stage de fin d'études.

J'adresse mes remerciements les plus sincères aux membres du jury pour avoir accepté de m'accorder de leur temps précieux pour juger ce modeste travail.

Je remercie l'ensemble des enseignants de l'UABT pour la qualité de leur formation. Merci à l'ensemble de mes camarades de l'UABT pour ces cinq années passées en leurs compagnie.

Enfin, je tiens à remercier tout particulièrement M. Soulimane KAMNI, M. houssam eddine YOUSSEFI et M. AZZOUG Aghiles pour leur aide tout au long de cette période de stage.

Résumé

Ce projet de fin d'études s'inscrit dans le cadre des travaux multidisciplinaires menés conjointement par l'équipe IDD du laboratoire LIAS et l'équipe RoBioSS de l'Institut PPRIME et qui visent à améliorer les départs des pilotes de la BMX Race. L'équipe IDD est spécialisée dans le traitement de la donnée tant dis que l'équipe RoBioSS est spécialisée dans le domaine de la biomécanique. Elle propose une solution matérielle (capteurs Pédaliers, caméras, atelier d'essai pour les pilotes) afin de collecter les données liées au départ des pilotes élites de l'équipe de France. L'équipe RoBioSS propose des modèles biomécaniques pour représenter ces départs dans le but de les optimiser par la suite. Toutefois cette solution ne facilite pas l'identification des paramètres responsable de l'amélioration du départ et ce à cause de la complexité des modèles biomécaniques proposés. Afin de simplifier ses modèles et identifier que les paramètres pertinents impactant le départ de la BMX Race, l'institut PPRIME a fait appel au laboratoire LIAS et ce en travaillant sur de nouveaux modèles basés sur l'auto-apprentissage. L'apprentissage automatique est un procédé informatique qui vise à déduire un ensemble de règles à partir d'un jeu de données pour construire de nouvelles connaissances. Ce procédé a été appliqué avec succès dans de différents domaines, comme les systèmes d'analyse des anciennes ventes pour la prédiction du comportement du client et les prévisions météorologiques.

Ce projet vise à proposer un modèle simplifié mais prédictif de la performance en se basant sur des techniques d'apprentissage automatique. Le but de notre travail est d'étudier les données de départ fournies par l'équipe RoBioSS, concevoir une solution compatible avec ce type de données et tester les algorithmes sur ces mêmes données.

Mots-clés

apprentissage automatique, science de données, BMX, régression linéaire, apprentissage supervisé.

Abstract

This graduation project is a part of the multidisciplinary work carried out jointly by the IDD team of the LIAS laboratory and the RoBioSS team of the PPRIME Institute, which aim to improve the departure of the BMX Race drivers. The IDD team specializes in data processing whereas the RoBioSS team is specialized in the field of biomechanics. It offers a hardware solution (Cranks sensors, cameras, test workshop for pilots) to collect data related to the departure of the elite drivers of the French team. The RoBioSS team proposes biomechanical models to represent these departures in order to optimize them afterwards. However this solution does not facilitate the identification of the parameters responsible for the improvement of the departure and this is because of the complexity of the proposed biomechanical models. In order to simplify its models and to identify the relevant parameters impacting the departure of the BMX Race, the PPRIME institute called on the LIAS laboratory to work on new models based on a different field which is Machine learning. Machine learning is a computer process that aims to derive a set of rules from a dataset to build new knowledge. This process has been successfully applied in different areas, such as old sales analysis systems for predicting customer behavior and weather forecasts.

This project aims to propose a simplified model but predictive of performance based on machine learning techniques. The purpose of my work is to study the initial data provided by the RoBioSS team, to design a solution compatible with this type of data and to test the algorithms on these same data.

Keywords

Machine Learning, data science, BMX, Linear regression, supervised learning.

ملخص

مشروع التخرج الخاص بي هو جزء من العمل متعدد التخصصات الذي ينفذ بشكل مشترك من قبل فريق IDD في مختبر LIAS وفريق RoBioSS التابع لمعهد PPRIME ، والذي يهدف إلى تحسين رحيل سائقي BMX Race. فريق IDD متخصص في معالجة البيانات في حين أن فريق RoBioSS متخصص في مجال الميكانيك الحيوية. إنه يوفر حلاً ماديا عن طريق توفير أجهزة متخصصة (أجهزة استشعار السواعد والكاميرات وورشة اختبار السائقين) لجمع البيانات المتعلقة برحيل سائقي النخبة في الفريق الفرنسي. يقترح فريق RoBioSS نماذج ميكانيكية حيوية لتمثيل هذه الانطلاقة من أجل تحسينها بعد ذلك. ومع ذلك ، فإن هذا الحل لا يسهل تحديد المعلمات المسؤولة عن تحسين المغادرة وهذا بسبب تعقيد النماذج الميكانيكية الحيوية المقترحة. من أجل تبسيط النماذج وتحديد المعلمات التي تؤثر على رحيل سباق BMX ، دعا معهد PPRIME مختبر LIAS للعمل على نماذج جديدة تعتمد على مجال مختلف وهو تعلم الآلة. تعلم الآلة عبارة عن عملية كمبيوتر تهدف إلى اشتقاق مجموعة من القواعد من مجموعة البيانات لبناء معرفة جديدة. تم تطبيق هذه العملية بنجاح في مجالات مختلفة ، مثل أنظمة تحليل المبيعات القديمة للتنبؤ بسلوك العملاء وتوقعات الطقس.

يهدف هذا المشروع إلى اقتراح نموذج مبسط ولكن تنبؤي بالأداء على أساس تقنيات التعلم الآلي. الغرض من عملي هو دراسة البيانات الأولية المقدمة من فريق RoBioSS ، وتصميم حل متوافق مع هذا النوع من البيانات واختبار الخوارزميات على هذه البيانات نفسها.

الكلمات المفتاحية:

تعلم الآلة ، علم البيانات ، BMX ، الانحدار الخطي ، التعلم الخاضع للإشراف.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

BMX	Bicycle moto cross
RoBioSS	Robotique, Biomécanique, Sport, Santé
FFC	Fédération française du cyclisme
RMSE	Root mean square error
FPS	Frame par seconde
RNN	Recurrent neural network
LSTM	Long short-term memory
CSV	Comma-separated values
PDF	Portable Document Format

Chapitre 1

INTRODUCTION, CONTEXTE ET PROBLÉMATIQUE

Ce projet de fin d'études s'inscrit dans le cadre des travaux multidisciplinaires menés conjointement par l'équipe IDD du laboratoire LIAS et l'équipe RoBioSS de l'Institut PPRIME et qui visent à améliorer les départs des pilotes de la BMX Race. Nous proposons en effet de simplifier les modèles proposés par l'équipe RoBioSS afin d'identifier que les paramètres pertinents impactant le départ de la BMX Race. Nos modèles sont basés d'ailleurs sur l'auto-apprentissage.

Dans la suite de ce chapitre, nous donnerons quelques détails sur le travail demandé ainsi que sur le déroulement de mon stage.

1.1 Contexte et problématique

La course de BMX, appelée « BMX race » ou « BMX Supercross » est une discipline Olympique depuis les Jeux de Pékin en 2008. Un départ optimal en BMX est conditionné par de très nombreux paramètres parmi lesquels on peut citer le temps de réaction, les caractéristiques musculaires, les forces appliquées et la coordination gestuelle. Au regard des autres disciplines telles que le cyclisme sur piste ou sur route, le BMX Race souffre d'un manque flagrant de connaissances scientifiques qui permettraient de mieux comprendre la technique de départ et l'influence de chacun de ces paramètres, soit isolément, soit de manière combinée. Ainsi, seuls une vingtaine d'articles scientifiques sont référencés dans PubMed. Le manque de données scientifiques concernant le BMX provient de l'intérêt encore récent pour cette jeune discipline mais surtout des difficultés technologiques associées à la mesure in situ. Face à ce constat, la Fédération Française de Cyclisme a sollicité l'équipe RoBioSS [1] de PPRIME [2] afin d'établir des connaissances scientifiques et technologiques permettant de :

— améliorer leur gestuelle technique et adapter les entraînements des sportifs d'haut-niveau

1. `https://www.pprime.fr/?q=fr/robioss`
2. `https://www.pprime.fr/`

— concevoir des matériels optimisés pour répondre aux contraintes spécifiques du BMX Race.

Ce projet s'inscrit, à court terme dans l'objectif des JO de Tokyo 2020 et, à plus long terme, dans la perspective des JO de Paris 2024. C'est d'ailleurs dans ce contexte que l'équipe RoBioSS s'est rapproché l'équipe IDD.Un projet de recherche est rapidement monté afin d'analyser les données liées au départ de la BMX Race.

L'objectif du dit projet est d'améliorer les performances des pilotes de l'équipe de France de Cyclisme spécialistes de BMX Race. Cependant, la multiplicité des variables mesurées auxquelles s'ajoute les variables calculées à partir de ces variables mesurées et de modèles biomécaniques développés par Robioss, ne permet pas à l'heure actuelle de fournir des indications totalement satisfaisantes aux entraîneurs. En parallèle du développement de modèles mécaniques déterministes toujours plus complexes dans la perspective d'en augmenter la biofidélité, ce projet de collaboration PPRIME - LIAS consiste, au contraire, à simplifier les modèles afin de se focaliser uniquement sur les paramètres les plus impactant. À court terme nous allons poursuivre deux tâches :

— l'identification de ces paramètres via des techniques de Machine Learning.
— l'élaboration de modèles simplifiés mais néanmoins prédictifs de la performance permettant d'établir des consignes génériques.

À plus long terme, les deux points évoqués seront approfondis afin de proposer des recommandations personnalisées à chaque pilote ainsi qu'aider à prédire les performances futures des jeunes pilotes.

Le stage vise aussi à renforcer la collaboration entre le laboratoire LIAS et l'institut PPRIME à travers l'exploitation de données multi-disciplinaire.

1.2 Le stage

J'ai trouvé ce stage grâce à M. Kamni, doctorant au laboratoire LIAS et étudiant du Master GL de Tlemcen (2013-2018), qui a aussi effectué son stage au sein du laboratoire LIAS. J'ai pris l'initiative de transmettre mon CV à M. Kamni qui m'a fait savoir qu'il y une opportunité de stage. J'étais contacté par la suite par mon maître de stage pour passer un entretien. Dès l'accord officiel, nous avons déclenché la procédure de signature de la convention de stage et de visa par la suite. Le stage s'est déroulé au sein du laboratoire LIAS qui se trouve au niveau des locaux de L'école nationale supérieure de mécanique et d'aérotechnique (ISAE-ENSMA) située à Poitiers - France, et ce du 04 Mars au 19 Juillet 2019.

1.3 L'équipe IDD du LIAS

L'équipe IDD du LIAS s'est spécialisée depuis plusieurs année en science de données. En effet, avec l'explosion sans cesse des données du Web et celles issues des instruments d'observation et de simulation modernes d'un côté et la volonté des entreprises de publier leurs masses de données afin qu'elles soient partagées, intégrées et exploitées d'une manière efficace pour l'aide à la prise de décisions, l'équipe IDD participe activement à l'élaboration de nouvelles techniques/méthodes liées au traitement et à l'exploitation

des données. L'application du Machine Learning pour l'exploitation des Big Data est l'un des axes prioritaires du projet de l'équipe. J'étais sous la responsabilité scientifique de Monsieur Amin MESMOUDI, Maître de conférences à l'Université de Poitiers et membre de l'équipe IDD.

1.4 L'equipe RoBioSS

L'équipe RoBioSS [3] est issue du rapprochement de deux équipes historiques de la physique et de la mécanique sur Poitiers en 2010. La richesse de l'équipe RoBioSS réside dans sa double compétence Robotique-Biomécanique, qui permet aujourd'hui de répondre aux enjeux sociétaux et industriels (H2020 – Usine du futur, robotique collaborative, assistance à la personne). À travers ses activités de recherche, l'équipe conçoit ses propres dispositifs mécatroniques (ergomètres de sport, robots humanoïde, préhenseurs, robots de chirurgie, etc.)

L'équipe RoBioSS de PPRIME nous aide à comprendre les données et nous a fournis les connaissances métiers necessaire afin de comprendre le besoin et les données. Nous avons travaillé étroitement avec Mathieu Domalain, maitre de conférences à l'université de Poitiers et Marc Duquesnoy stagiaire Master 2 en statistiques,

Nous avons travaillé ensemble sur l'analyse des paramètres de départ. Marc s'est basé sur des approches statistiques avec des ACP. De notre côté, nous nous sommes intéressés plutôt à des approches à base Machine Learning.

On se voyait souvent pour se partager le travail et discuter les résultats obtenus. J'ai assuré toutes les taches de pré-traitement et nettoyage de données pour qu'ils puissent effectuer leurs approches statistiques.

Un mois après mon arrivée au LIAS, un nouvel étudiant en L2, Louis Bozier, a intégré l'équipe RoBioSS pour réaliser un site web pour les entraîneurs et les pilotes de la BMX Race afin qu'il puisse visualiser et analysé facilement leurs essais. Louis s'est occupé principalement du côté front-end, nous nous sommes occupé de tout ce qui est back-end afin de lui fournir les graphes et toutes autres informations. On s'est mis d'accord de travailler avec python alors il a utilisé Django [4] pour pouvoir intégrer mon code facilement a son site web.

1.5 Le laboratoire LIAS

Le LIAS [5] (**L**aboratoire d'**I**nformatique et d'**A**utomatique pour les **S**ystèmes) représente 35 enseignants chercheurs dans les disciplines de l'Automatique, du Génie électrique et de l'Informatique. Il a été créé le 1er janvier 2012, suite à la fusion des laboratoires du LAII (Laboratoire d'Automatique et d'Informatique Industrielle) et du LISI (Laboratoire d'Informatique Scientifique et Industrielle).

3. `https://www.pprime.fr/?q=fr/robioss`
4. `https://www.djangoproject.com/`
5. https ://www.lias-lab.fr/

Le laboratoire LIAS est composé de trois équipes de recherche : l'équipe Ingénierie des Données et des Modèles, l'équipe Systèmes embarqués Temps Réel et l'équipe Automatique et Système.

FIGURE 1.1 – Le logo du laboratoire **LIAS**

1.6 L'institut PPRIME

L'Institut Pprime (P') est une unité propre de recherche du CNRS [6] créée en 2010 en partenariat avec l'ISAE-ENSMA et l'Université de Poitiers. Elle est composée de plus de 600 personnes dont les thématiques de recherche concernent les Sciences pour l'Ingénieur et la Physique des matériaux. Elle est constituée de trois départements : Physique et Mécanique des Matériaux, Fluides-Thermique-Combustion, Génie Mécanique et Systèmes Complexes.

1.7 La fédération française du cyclisme

La Fédération française de cyclisme (ou FFC) organise les disciplines cyclistes en France. Ces disciplines sont : le cyclisme sur route, le vélo tout terrain, le cyclisme sur piste, le cyclo-cross, le BMX, le cyclisme en salle, le paracyclisme et le polo-vélo. La FFC est membre de l'Union cycliste internationale et de l'Union européenne de cyclisme.

1.8 Organisation du manuscrit

La suite de ce manuscrit sera organisée en six chapitres : Le chapitre 2 sera dédié à la présentation du bagage théorique liées aux techniques d'auto apprentissage utilisées. Je présenterai les données liées au départ de la BMX Race dans le chapitre 3. Le chapitre 4 est consacré à la conception de nos approches. Je donnerai des détails sur nos implementations dans le chapitre 5. Le chapitre 6 présentera une discussion un liée à ma méthode de travail. Enfin, une conclusion et quelques perspectives seront données dans le chapitre 7.

6. `http://www.cnrs.fr/fr/page-daccueil`

Chapitre 2

SYNTHÈSE BIBLIOGRAPHIQUE

2.1 Introduction

Cette première partie permet de synthétiser les résultats de la recherche bibliographique effectuée autour de l'apprentissage automatique. Nous commencerons notre synthèse par présenter les notions de base sur l'apprentissage automatique et les différents types existants.

Nous aborderons ensuite une petite définition de l'apprentissage profond (Deep Learning) et les métriques d'évaluation.

2.2 L'apprentissage automatique

2.2.1 Définition

Un programme informatique est dit apprendre de l'expérience E pour la tâche T et une mesure de performance P si sa performance sur T, comme mesurée par P, s'améliore avec l'expérience E[12].

Il existe plusieurs types d'apprentissage, les plus répandus étant **l'apprentissage supervisé** et l'apprentissage **non supervisé**, appelé aussi **clustering**. D'autres un peu moins connus comme l'apprentissage **semi supervisé** existent dans la littérature.

2.2.2 Type d'apprentissage automatique

Apprentissage non supervisé

Dans ce type d'apprentissage, c'est uniquement les données d'entrée qui sont connues (c'est-à-dire les X_i), le but étant de construire un modèle pouvant les représenter de la manière la plus précise.
Il existe deux sous catégories :

1. **Le clustering**

   Le but de cet apprentissage est de regrouper les données hétérogènes en se basant sur des indices de similarité (des distances en général), c'est ensuite à l'opérateur

de déduire du sens pour chaque groupe.
Parmi les algorithmes de clustering les plus connus :
— k-means
— l'algorithme EM(Espérance-maximisation).
— DBSCAN (Density-Based Spatial Clustring of Applications with Noise)
— SOM (self Organizing Maps)

2. **Les associations**

Le but est de trouver des associations significatives, qui décrivent l'ensemble des individus, entres les items d'une base de données.
Formellement [9] :
Soient :
— $I = \{I_1, I_2, ..., I_n\}$ l'ensemble d'items dans la base de donnée, n étant leurs nombre.
— $T = \{T_1, T_2, ..., T_m\}$ l'ensemble des transactions de la base de donnée, m étant leurs nombre, une transaction étant définie comme un sou ensemble de I.
— $X \subseteq I, Y \subseteq I$ deux sous ensemble d'items
— Un sou ensemble X à un support s si le sous ensemble X apparaît au moins avec une fréquence s par rapport à toutes les transactions T.

Une règle d'association est définie comme suit :
$X \to Y \; SI \; X \cap Y = \emptyset$ et le support de $X \cup Y > s$
un exemple de règle d'association est : Si A et B alors C où A, B et c peuvent être des films, de la musique, etc..
Les algorithmes les plus connus sont
— FP-Growth
— L'algorithme Apriori
Ces algorithmes sont très utilisés dans les approches à filtrage collaboratif

Apprentissage supervisé

L'objectif dans ce type d'apprentissage est de trouver une fonction de correspondance de x à y, étant donné un ensemble d'entraînement composé de paires (x_i, y_i). Ici, les $y_i \in Y$ sont appelés les étiquettes des exemples x_i[13].
Plus formellement :
Etant donné un groupe des données d'observation $D = \{(X_1, Y_1), ..., (X_n, Y_n)\}$ où les X_i représentent les entrées appartenant à $\mathbb{R}^p$ (variable qualitative) et les Y_i les valeurs d'affectation.
où :
— $i = 1..n$: La taille de l'échantillon d'apprentissage.
— p : la dimension des données d'apprentissage (le nombre de caractéristiques).

Dans le cas de la classification les Y_i prennent des valeurs catégoriques (Dans $\mathbb{N}$) et dans le cas de la régression elles prennent des valeurs numérique (dans $\mathbb{R}$).
La classification consiste à trouver et optimiser une fonction f qui associe à chaque X_i sa valeur Y_i.
Parmi les algorithmes de classification supervisé, on peut trouver :
— Les K-plus proches voisins : KNN.

— Les machines à vecteur de support : SVM.
— Les arbres de décision et forêts aléatoires.
— La régression simple et multiple (linéaire et non linéaire).

Apprentissage semi-supervisé

L'apprentissage semi-supervisé (SSL) se situe à mi-chemin entre l'apprentissage supervisé et l'apprentissage non supervisé.
L'idée est d'utiliser les données non-labélisées pour compléter l'apprentissage supervisé. L'avantage principal est le gain de temps et d'argent, car la labélisation est coûteuse en termes de temps et nécessite l'intervention d'experts du domaine.

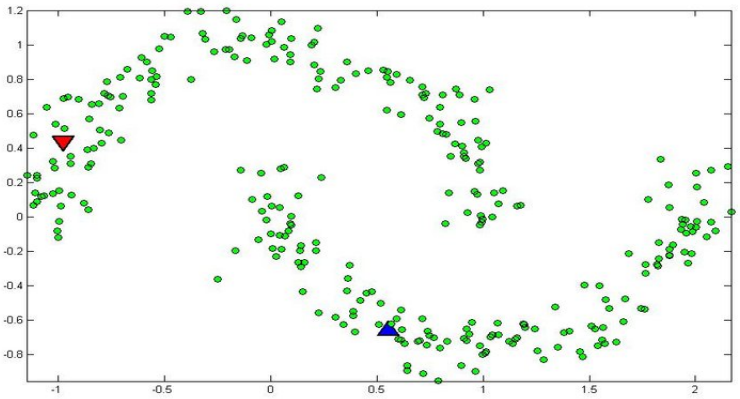


FIGURE 2.1 – Apprentissage semi-supervisé[**?**]

Dans cet exemple, on dispose d'un grand nombre de données non-labélisées (en vert) issus de deux classes. Seules deux données sont étiquetées (Le triangle Rouge de classe 1, et le triangle bleu de classe 2). Le but d'un algorithme d'apprentissage semi-supervisé sera de labéliser les points verts des deux classes. i.e. les points verts du demi-cercle du dessous devront être affectés à la classe 2 et les points du demi-cercle du dessus seront affecté à la classe 1.
Les techniques les plus utilisées sont :
— L'auto-apprentissage
— Le co-apprentissage
— S3VM : Séparateur Semi-Supervisé à Vaste Marge
— Les méthodes basées sur la réduction de l'incertitude ou la réduction de l'erreur

Apprentissage partiellement supervisé

C'est une extension de l'apprentissage semi-supervisé, dans le cas où on a un étiquetage partiel, c'est-à-dire des données dont on sait qu'elles n'appartiennent pas à une certaine classe, sans connaître leurs classe d'appartenance. Un exemple concret serait un médecin qui effectue un diagnostic, le médecin a le choix entre les maladies A, B ou C. Il est sûr que son patient n'est pas atteint de la maladie A, mais il hésite entre B et C.

Une des méthodes proposée dans la littérature est l'utilisation de modèles probabilistes basés sur des lois multinomiales pour la prédiction de la probabilité d'appartenance à chaque classe.

Apprentissage par renforcement

L'apprentissage par renforcement consiste à apprendre quoi faire, comment faire correspondre des situations sur des actions de manière à maximiser une récompense numérique. On ne dit pas à l'apprenant quelles actions entreprendre, mais il doit plutôt découvrir quelles actions donnent le plus de récompense en les essayant [1]. L'apprentissage par renforcement est considéré par beaucoup de chercheurs comme étant la forme la plus pure d'intelligence artificielle, celle qui imite le plus le comportement humain où le programme cherche, au travers d'expériences itérées, un comportement à apprendre. Les algorithmes les plus connus sont :

— Q-learning
— SARSA : State-Action-Reward-State-Action

2.3 Apprentissage profond

L'apprentissage profond est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires moyennant les réseaux de neurones profonds[11]. Ces techniques ont permis des progrès importants et rapides dans les domaines de l'analyse du signal sonore ou visuel et notamment de la reconnaissance faciale, de la reconnaissance vocale, de la vision par ordinateur et du traitement automatique du langage. Ce type d'apprentissage est très efficace pour les problèmes non linéaires. Leurs inconvénients principaux est le temps de calcul et le temps d'entraînement qui sont très importants.

2.4 Métrique d'évaluation

2.4.1 La matrice de confusion

La matrice de confusion est une table représentant les valeurs réelles en fonction des valeurs prédites, elle permet de calculer l'erreur de classification.

Dans ce qui suit on définit comme "positif" la classe 1 et comme "négatif" la classe 0.

		Calsse réelle	
		Négatifs	Positif
Classe Prédite	Négatifs	VN(Vrais négatifs)	FN(Faux négatifs)
	Positifs	FP(Faux positifs)	VP(Vrai positifs)

TABLE 2.1 – Matrice de confusion

A partir de cette matrice on peut définir plusieurs autres métriques, entre autres :

— **Le rappel** : appelé aussi la sensibilité, c'est le **taux** de vrais positifs, il représente la capacité du système à prédire correctement la classe positive. Sa formule est la suivante :

$$Rappel = \frac{VP}{VP + FN}$$

— **La précision** : la proportion de prédictions correctes parmi les points que l'on a prédits positifs. Sa formule :

$$Rappel = \frac{VP}{VP + FP}$$

2.4.2 RMSE

La RMSE (Root Mean Square Error) est une métrique d'évaluation d'erreur utilisé généralement dans la régression. L'avantage principal de cette métrique est qu'elle exagère les grandes erreurs. L'idéal, bien sûr, est d'avoir une RMSE égale à zéro. Sa formule est la suivante :

$$RMSE = \sqrt{\frac{1}{N} \times \sum_{i=1}^{N} (\bar{v}_i - v_i)^2}$$

Où N est le nombre de prédictions faites, v_i la valeur réelle et $\bar{v}_i$ la valeur prédite

2.4.3 Coefficient de détermination

Le coefficient de détermination, noté R^2 ou r^2 est un indicateur qui permet de juger la qualité d'une régression linéaire simple. Ce coefficient varie entre 0 et 1, soit entre un pouvoir de prédiction faible et un pouvoir de prédiction fort. Si le R^2 vaut 1, cela signifie que l'équation de la droite de régression est capable de déterminer 100 % de la distribution des points. Donc plus on se rapproche de la valeur 1 plus on a de bonne prédiction dans la régression et inversement. Sa formule est la suivante :

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (v_i - \hat{v}_i)^2}{\sum_{i=1}^{N} (v_i - \bar{v})^2}$$

Où N est le nombre de prédictions faites, v_i la valeur réelle et $\bar{v}_i$ la valeur prédite

2.5 Conclusion

Nous avons abordé dans ce chapitre les notions essentielles concernant l'apprentissage automatique. Ensuite, les type d'apprentissage automatique et leurs modèles les plus connus, chacun étant adapté à un problème précis. Nous avons également vu les trois principales métriques d'évaluation de ces modèles à savoir la matrice de confusion, le RMSE et le coefficient de détermination. Dans le chapitre suivant, nous aborderons le chapitre « Etude de l'existant »

Chapitre 3

Étude de l'existant

3.1 Étude de l'existant

Pour concevoir une solution qui répond au mieux aux besoins de l'équipe RoBioSS, nous avons d'abord étudié le fonctionnement de leur système et son architecture actuelle.

3.2 Fonctionnement de RoBioSS

RoBioSS travaille sur le projet de la fédération française de cyclisme qui vise à analyser et améliorer les performances des pilotes élites lors de leurs course de BMX Race. Pour ce faire, ils ont pu concevoir un pédale qui permet de mesurer la force appliquée lors des coups de pédales en 3D c'est-à-dire au trois axes (X, Y et Z). Ils ont aussi conçu un atelier pour faire les mesures lors du départ des pilotes élites. Dans cette atelier on trouve la butte de départ des pilotes où on a mis des caméras toute au long de la bute de départ pour pouvoir capter les mouvements gestuels des pilotes. Ensuite ils analysent les données fournies par ses caméras pour faire les calcules nécessaire pour obtenir les paramètres qui construisent la base données des essais des pilotes. Ce travail de calcul et d'analyse des données fournit par les caméras prend beaucoup de temps, et pour cela que ce n'ai pas facile d'avoir de nouvelle base de données rapidement. La base de données créée est sauvegardée sous format Matlab (.mat).

3.3 Architecture du Système actuelle

Nous présenterons maintenant l'architecture matérielle utilisée par l'équipe RoBioSS pour la création des bases de données Matlab.

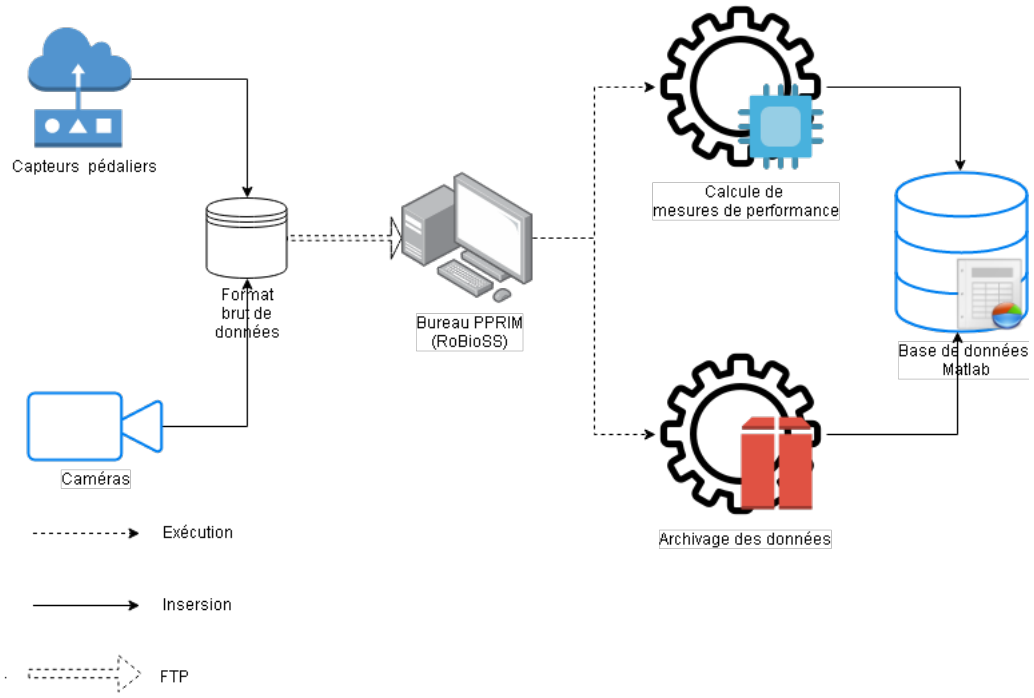


FIGURE 3.1 – Architecture actuelle de RoBioSS

La figure ci-dessus 3.1 illustre l'architecture actuelle utilisée. Elle est constituée des composants suivants :

Capteurs pédaliers : c'est des capteurs sous forme de pédales, il se place directement à la place des pédales. Ils permettent de fournir les données liées à la force appliquée au moment du coup de pédale. Les données sont fournies en trois dimensions (X, Y, Z).

caméras : c'est des caméras placées toutes au long de la partie du départ(la butte), elles permettent d'enregistrer les données gestuelles des pilotes, le déplacement, la vitesse, etc. Les caméras enregistrent en 300 FPS, comme le départ dure entre 1.5s à 2s donc pour chaque essai de pilotes on peut collecter de 450 à 600 observations par essai.

Format Brut de données : Comme expliqué dans l'architecture actuelle 3.1, principalement on a des caméras qui font l'enregistrement des données, donc les données sont dans le format vidéo. Pour faire ressortir le format numérique des données il faut faire des prétraitement et des calculs au niveau des vidéos enregistrés. Ce traitement prend énormément de temps ce qui rend la tâche de collection de données très complexe.

Base de données Matlab : c'est le type de base de données choisi par l'équipe RoBioSS pour faire le sauvegarde des données numérique. Cette base regroupe tous type de données enregistrées que ce soit des capteurs pédalier ou des caméras.

3.4 Description des données

Après avoir exposé les différentes composant de l'architecture matérielle utilisée par l'équipe RoBioSS, nous nous pencherons maintenant sur la structure logique de ses données.

3.4.1 Structure des données

Dans ce qui suit tous les données sont extraites à partir de la base de données Matlab fournie pour chaque essai. Nous présentons d'abord la structure d'une base de données d'un essai de pilote. La base c'est une matrice qui contient cinq structures. Une structure est elle-même une matrice qui peut contenir d'autre structures. Les cinq premières structures sont les suivantes A.6 :

InfoRider : Contient toutes les informations sur le pilote (nom, taille, poid, age ect.).

Info : contient d'autres structures imbriquées mais en général ce sont des informations sur l'essai (Braquet, Manivelle, temps d'arriver, etc.).

Markers : contient d'autre structures imbriquées, et elle aussi contient d'autres informations qui liées à l'essai.

CriterePerf : Contient des données liées à des moments clés du départ (PuissanceMAX, ForceMax, VitessEnBasDeButte, etc.).

Traitement : Cette dernière contient tous les autre données toutes au long de l'essai, ce sont des données a chaque instant du départ jusqu 'à l'arrivée en bas de butte. On les a appelé données temporelle puisque elles varient dans le temps. En Moyenne on a 500 observations par base.

Maintenant quand on connaît la structure générale de la base de données d'un essai, on peut voir directement sa complexité surtout en présence des structures, ce qui rend la phase d'analyse de données difficile. Pour palier a ce problème nous avons choisi d'aplatir les données A.2.2 c'est-à-dire on aura qu'un seul niveau d'imbrication et pas de notion de structure, la meilleure façon de le faire est de transformer notre base sous un format de fichiers CSV qui nous facilitera l'analyse et élimine les imbrications. Nous nous sommes mis d'accord avec l'équipe RoBioSS de PPRIME de fournir deux fichiers CSV par essai (au paravent un fichier .mat par essai), un fichier CSV pour tous ce qui est information sur le pilote, le matériel utilisé et d'autre information relative à cet essai. Nous avons appelé ce premier fichier les données discrètes. Le deuxième CSV contient toutes informations durant la phase de départ, tous ce qui est en relation avec la Vitesse, la Force, le moment, etc., nous avons appelé ce type de fichier les données temporelles.

3.4.2 Les données discrètes

C'est un fichier CSV contenant des paramètres(colonnes) et qui prennent une seule valeur réelle durant cet essai. On les a extrait principalement des 4 structures (InfoRider, Info, Markers, CriterePerf). Ces paramètres nous donnent des informations sur le pilote et sur l'essai, on peut citer le poids du pilote, sa taille, la longueur de la manivelle, le temps en bas de butte, etc.

3.4.3 Les données temporelles

C'est les données issues des caméras après calculs et prétraitements, c'est principalement les données qui sont extraites de la structure "Traitement" qu'on a pris et stockée dans un fichier CSV.

3.5 Conclusion

Nous avons vu, au cours de ce chapitre, l'architecture physique du système de RoBioSS ainsi que l'architecture logique des données mises à notre disposition et les transformations faites pour permettre l'analyse de données que nous allons détailler dans le chapitre suivant.

Chapitre 4

CONCEPTION

4.1 Introduction

Dans ce chapitre, nous abordons la partie conception et réalisation de notre proposition. On rappelle qu'il s'agit d'un projet de recherche, et qu'on cherche a faire une étude de faisabilité et d'application en utilisant des techniques d'apprentissage automatique sur les données fournies par le laboratoire **PPRIME**.

Dans la suite de ce chapitre nous allons d'abord présenter la démarche de conception utilisée. Ensuite, nous exposerons l'approche générale et son architecture. Nous présenterons par la suite processus de préparation et de transformation de données et l'ensemble de l'analyse qui a été fait sur ces dernières.

4.2 Méthodologie de conception

4.2.1 Méthode CRISP

CRoss-Industry Srandard Process (intialement connue comme **CRISP-DM[5]** pour "Data Mining") est une méthode développée par **IBM** [1] pour réaliser des projets de Dataminig dans les années 60. Aujourd'hui généralisée à tous les types de projets de Data science. Elle est la méthode la plus populaire dans la mise en place de projets Data science compte tenue qu'elle est totalement indépendante des technologies utilisés, des outils et des applications qui sont en évolution constante dans le marché. Elle impose un schéma standard applicable à tout type de projet ou d'infrastructure.

1. `https://www.ibm.com/fr-fr`

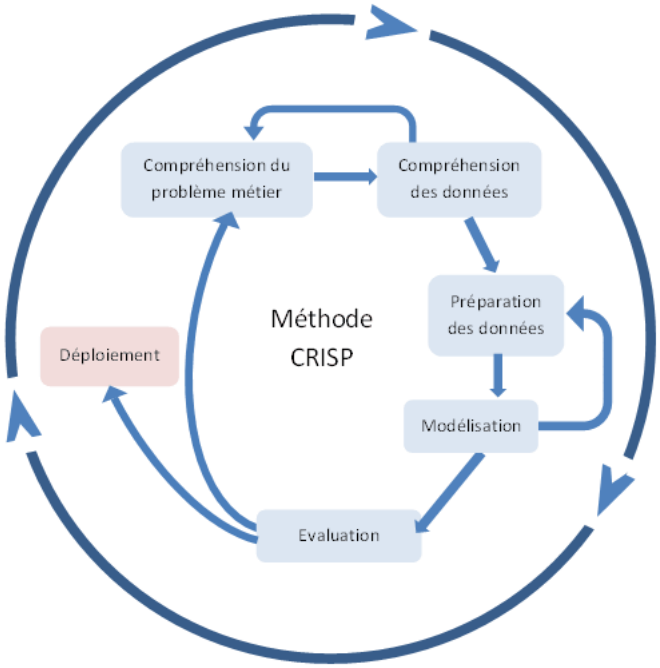


FIGURE 4.1 – Le cycle de vie de l'exploration des données[6]

Compréhension du problème métier

La première étape consiste à bien comprendre les éléments métiers et problématiques que la Data Science vise à résoudre ou à améliorer, puis convertir ces connaissances en une définition du problème.

Compréhension des données

Cette phase vise à se familiariser avec les données à utiliser, à trouver une description, une définition de leur valeur métier, à identifier la qualité des données, et de se faire une première idée des données ou détecter des sous-ensembles intéressants pour former des hypothèses sur ses informations cachées.

Préparation des données

La phase de préparation des données couvre toutes les activités visant à construire le jeu de données finales à partir des données brutes. Cette étape concerne aussi le nettoyage des données et leurs retransformations pour les rendre compatibles avec les algorithmes qui seront utilisés.

Modélisation

Cette phase consiste à la proposition d'un modèle donc le choix, le paramétrage et le test de différents algorithmes ainsi que leur enchaînement. Elle comprend aussi les choix techniques concernant les différentes solutions proposées.

Évaluation

La phase d'évaluation vise à vérifier les résultats obtenus en appliquant le modèle conçu durant la phase précédente. La décision de déployer le modèle en production ou de l'améliorer est prise à la fin de cette étape.

Déploiement

L'objectif de cette phase finale est de mettre les modèles conçus et validés à la disposition des utilisateurs finaux. Des perspectives d'amélioration sont proposées durant cette dernière étape.

L'objectif de chaque phase étant clair, aucune étape n'est superflue, seul le temps passé sur chacune d'elle peut varier d'un projet à un autre, voire d'une itération à une autre.

4.3 Approche générale

On rappelle que le but principal est l'identification des paramètres responsable de l'amélioration du départ des pilotes élites de la BMX Race via des techniques d'apprentissage automatique et l'élaboration d'un modèle simplifié mais néanmoins prédictifs de la performance qui permet par la suite d'établir des consignes génériques et ceci en utilisant les données fournies par le laboratoire PPRIME.

Les données issues de capteurs, fournies par le laboratoire PPRIME, sont un mélange entre des variables discrètes et d'autres temporelles.

Donc dans la proposition de notre solution, nous avons proposé deux sous-approche pour effectuer les prédictions voulues pour les deux différents types de données.

- Les données temporelles : ce sont les données pour lesquelles nous disposons de 300 lignes de données par seconde (les mesures sont prise en 300 fps), sachant qu'un départ dure environ 2 secondes donc on a un nombre suffisant de données pour utiliser de la régression linéaire.
- Les données discrètes : ces les informations sur les pilotes, la BMX et l'essai, sachant qu'on a 9 pilotes et de 6 à 8 essaies par pilote, on n'a vraiment pas suffisamment de données pour pouvoir utiliser la régression linéaire directement donc on choisit d'utiliser de la régression polynomiale qui nous permet de générer plusieurs autres variables calculées à partir des variables discrètes qu'on a déjà pour nous fournir suffisamment de données pour pouvoir utiliser la régression.

Les différents cas et les approches associées sont résumés dans la figure suivante (4.2)

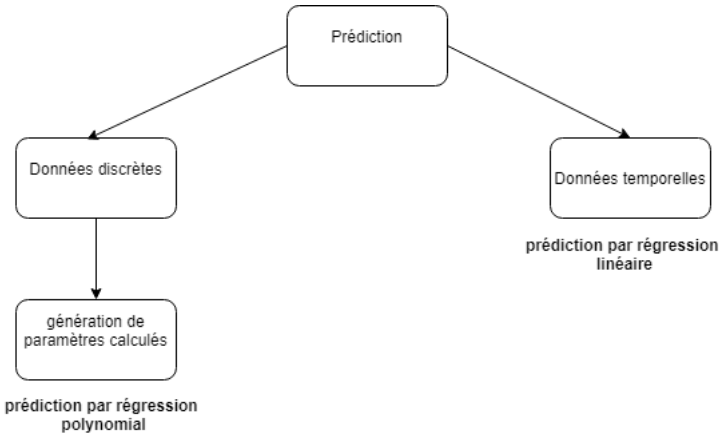


FIGURE 4.2 – Approche générale

4.4 Architecture générale

Nous avons projeté le système actuel de PPRIME par rapport à la solution proposée dans la section 4.3.

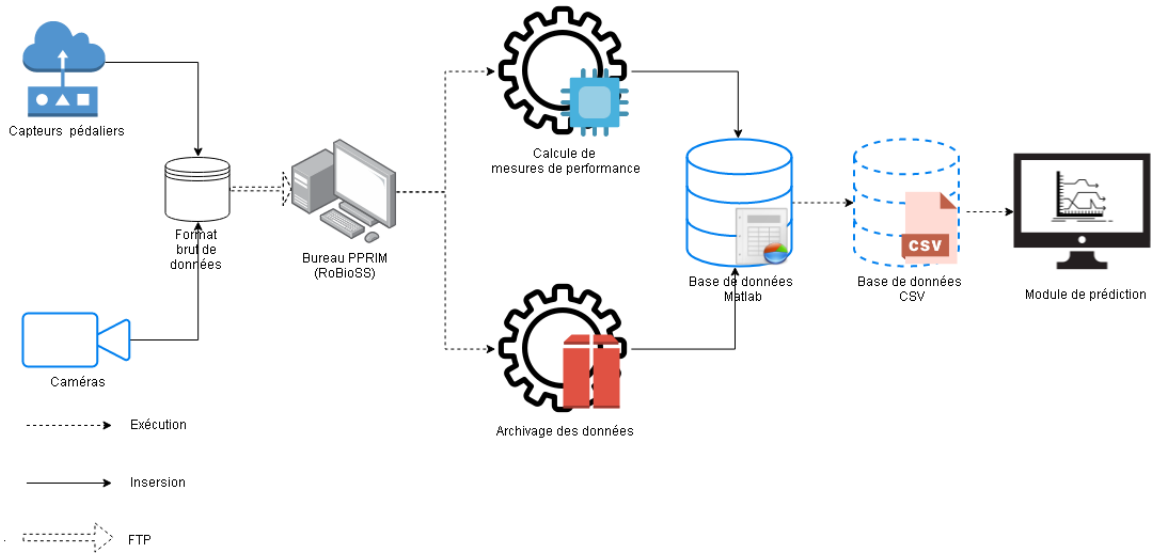


FIGURE 4.3 – Architecture générale

Les prédictions pourront être générées une fois qu'on a généré la base de données Matlab. Pour ce projet, nous n'allons pas déployer la solution. Notre travail est axé sur la prédiction des performances des pilotes, nous tâcherons de proposer une solution méthodologique précise, reproductible et surtout générale, que l'équipe PPRIME pourra par la suite exploiter et la déployer dans son système.

Les avantages de cette solution :

— Ne modifie pas l'architecture déjà en place.

— Permet une consultation des prédictions dans n'importe quelle instant et d'une façon instantanée pour n'importe quel jeu de données sans avoir à faire de nouvelle essaies avec les pilotes.
— Facile à comprendre et à modifier.

Les conditions de réussite :
— Bonne organisation de données.
— Présence de personnes ayant les connaissances métiers nécessaires.

4.5 Description fonctionnelle des besoins

Nous listerons ci-dessous les spécifications fonctionnelles de la solution proposée.

Identifiant	**Spécification**
1	Le système doit permettre à un expert de déclencher la transformation des données brutes(Matlab)
2	Le système doit permettre à un expert de diviser les données en deux catégories des données temporelles et autres discrètes.
3	Le système doit permettre la détection de l'intervalle du départ.
4	Le système doit permettre le découpage des données temporelles dans l'intervalle du départ.
7	Le système doit permettre le décalage du temps au moment du départ.
8	Le système doit permettre le stockage de données selon cette organisation `Data \ nom_pilote \ essaie \ *.csv`
9	Le système doit permettre la visualisation de données.
10	Le système doit permettre de ressortir toutes les corrélations entre les paramètres.
11	Le système doit permettre la prédiction de la force utile selon un modèle simplifié.
12	Le système doit permettre la prédiction du temps en fonction des paramètres choisi par un expert métier.

Table 4.1 – Spécifications fonctionnelles

4.6 Spécifications techniques

Les aspects techniques de la solution proposée sont énumérés ci-dessous.

Identifiant	**Spécification**
13	Les différentes données utilisées doivent être stockées en format CSV.
14	Le système doit prendre en charge les données sous leurs formats bruts (fichiers .mat)

Table 4.2 – Spécifications techniques

4.7 Compréhension des données

Après avoir présenté les données mises à notre disposition dans la section 3.4, nous présentons dans cette partie l'analyse faite sur les données fournies par PPRIME.

Cette analyse nous permettra par la suite de savoir quels algorithmes utilisés sur ses données et quels paramètres faudra garder et aussi quels traitements effectués sur ces dernières pour les rendre compatibles avec les algorithmes choisis.

4.7.1 Description des données

Avant de passer à l'analyse proprement dite, nous ferons un bref rappel les différentes données mise à notre disposition de la part de PPRIME.

Les données temporelles

Comme mentionné dans la section 3.4 ces informations son stocké sous format CSV et contiennent tous ce qui est force pieds, force utile, puissance, moment, vitesse, travail pied, etc. Dont certains paramètres sont calculés par exemple le moment ou la puissance.

Les données discrètes

Ces données représente des instants clés dans le départ d'une course BMX ou des renseignements sur le pilote élite et la BMX elle-même, on peut citer des exemples comme le temps en bas de butte, le temps de réaction, la distance d'alignement, le braquet, le poids du pilote, etc.

4.8 Analyse de données

La partie la plus importante de ce travail était de faire une analyse de données correcte, permettant de bien choisir les algorithmes à utiliser et les paramètre qu'il faut garder et analyser.

4.8.1 La corrélation entre les paramètres

Compte tenu du fait que notre travail dans un premier temps consistait à proposer un modèle simplifié qui permet de prédire la force utile, nous avons jugé raisonnable de voir les corrélations entre les paramètres.

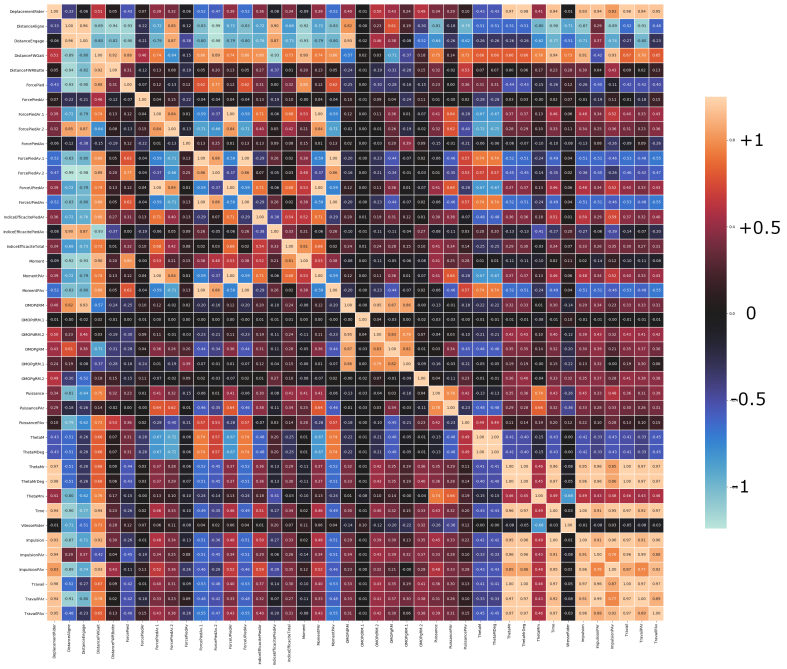


FIGURE 4.4 – La Matrice de corrélation

Pour simplifier un modèle il faut absolument qu'on supprime les paramètres qui sont fortement corrélés que ce soit d'une façon positive ou négative. Cependant, cela va augmenter la variance des coefficients de la régression et rend l'interprétation de ces deniers difficiles et pas fiable, en sachant que la suppression de l'un de ces deux paramètres corrélés n'a pas une incidence considérable sur la prédiction du modèle du coup on va ne garder qu'un seul paramètre des deux et le choix se fait selon une connaissance métier quand même.

Dans la figure qui suit on va expliquer la matrice de corrélation ci-dessus 4.4.

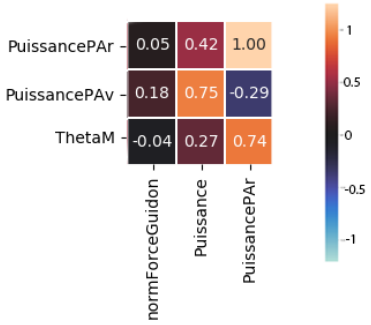


FIGURE 4.5 – Zoom sur La Matrice de corrélation

Un zoom sur la figure 4.4 permet de mieux voir les valeurs de corrélation entre deux paramètres. On voit clairement que quand on tend vers l'orange on dit que les deux variables sont corrélé positivement, sinon corrélé négativement.

4.8.2 Compréhension de la force Utile

Après avoir examiner les relations entre les paramètre, nous nous sommes intéressé à exprimer la force utile en fonction des paramètres pertinents et responsables sur l'évolution de cette force, car comme l'a été mentionné dans la section 3.2, des études faites par PPRIME, montre que la force utile est un paramètre responsable de l'amélioration des performances des pilotes comme le montre la figures qui suit :

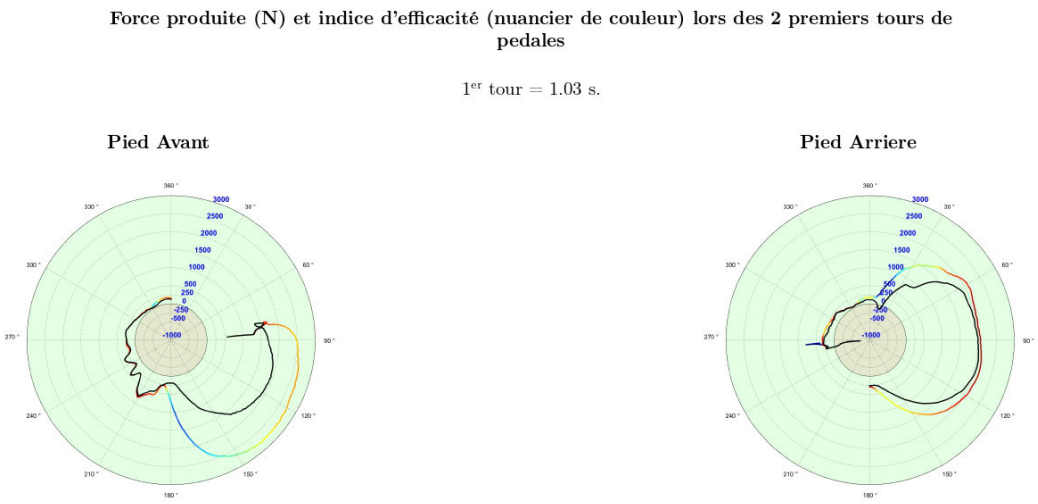


FIGURE 4.6 – graphe polaire Force - indice efficacité

Nous remarquons clairement que parfois en appliquant peu de force on obtient un indice d'efficacité élevé et parfois non, et c'est ce phénomène qu'on cherche à comprendre, car c'est l'indice d'efficacité ou bien la force utile qui fait avancer la BMX pour obtenir de meilleur résultats, donc on cherche à trouver les autres paramètres qui influencent l'évolution de cette indice en investissant un minimum de force.

Analyse de la force utile

Après avoir vu les résultats du graphe polaire, nous nous sommes intéressé à celles-ci, c'est-à-dire, nous avons cherché à faire des graphiques pour essayer de comparer la force utile des meilleurs essaies des pilotes et voir l'impact de la force utile sur ces derniers. Ci-dessous 4.7 les différents courbes de la force utile pour le meilleur essaie de chaque pilote. Dans cette analyse on a pris que les quatre meilleurs pilotes.

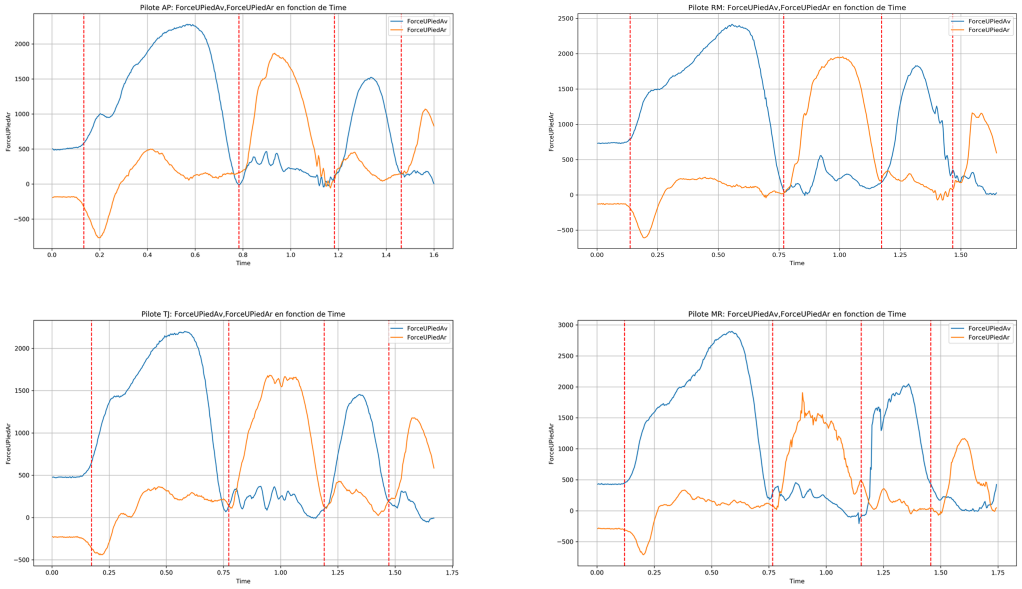


FIGURE 4.7 – Force utile du pieds avant et arrière pour les Pilotes : Arthur Pilard(AP), Romain Racine(RM), Toumas jouve(TJ) et Mathis Ragot(MR).

Cette figure ne montre pas directement que la force utile avait un impact important. En voyant que le premier (AP) arrivé en bat de but avait une force utile bien plus inférieur que le quatrième (MR) pour les 3 premiers coups de pédales. Cela est due aux autres paramètres qui sont externe cette fois-ci comme le montre la figure 4.8.

AP time: 1.597s

Poids: 84.2
taille: 178.0
braquet: 2.764
Longueur Manivelle: 177.5

TJ time: 1.667s

Poids: 85.0
taille: 176.0
braquet: 2.764
Longueur Manivelle: 175.0

RM time: 1.643s

Poids: 85.9
taille: 177.0
braquet: 2.793
Longueur Manivelle: 175.0

MR time: 1.743s

Poids: 95.6
taille: 187.0
braquet: 2.6875
Longueur Manivelle: 175.0

FIGURE 4.8 – Détails des pilotes et matériel utilisé pour l'essai

En comparant les poids des deux pilotes nous constatons qu'il y a $\approx 10kg$ de différence, le poids est un facteur responsable de la quantité de force appliquée plus on pèse plus on applique de force, sans oublier le matériel utilisé (longueur manivelle/Braquet)

et les autres paramètres temporels. Or, si on remarque bien les poids du premier (AP) et le troisième (TJ) sont presque égale et de même pour le matériel utilisé, en revenant à la figure 4.7 nous pouvons constater que cette fois-ci c'était (AP) qui a appliqué plus de force utile pour les 3 premiers coups de pédales d'où son meilleur classement par rapport à (TJ).

4.8.3 analyse des paramètres discrets

4.9 Préparation des données

Étape appelée aussi prétraitement. C'est l'étape la plus couteuse en temps, durant celle-ci les données sont nettoyées et agrégées.

Chaque type de fichier est préparé différemment.

4.9.1 Les données temporelles

Le but de ce traitement est d'extraire que les paramètres qui sont en relation avec le temps. Comme mentionné dans la section 3.4 les données fournies par PPRIME sont sous le format Matlab, on est tombé dans le besoin de reconvertir ces derniers sous un autre format (CSV), et cela pour pouvoir utiliser les technologies de l'analyse de données et de Machine Learning fournies par Python facilement.

La façon dont les données nous ont été fournit était un fichier Matlab (.mat) par essai de pilote, donc la première étape de prétraitement était de faire sortir pour chaque paramètre de ces derniers un fichier CSV. Ce travail a été fait avec des scriptes Matlab parce que c'était le seul choix car les données étais complexe : les données fournies étaient des structures, une structure en Matlab c'est une matrice. Ces valeurs sont également une matrice et cette matrice c'est le paramètre qu'on a mesuré. Ensuite, sélectionner parmi les fichiers CSV obtenus que ceux qui représentent les paramètres temporels et les concaténer tous en un seul fichier CSV.

L'extraction des fichiers CSV et la concaténation sont représentées dans la figures suivantes 4.9 :

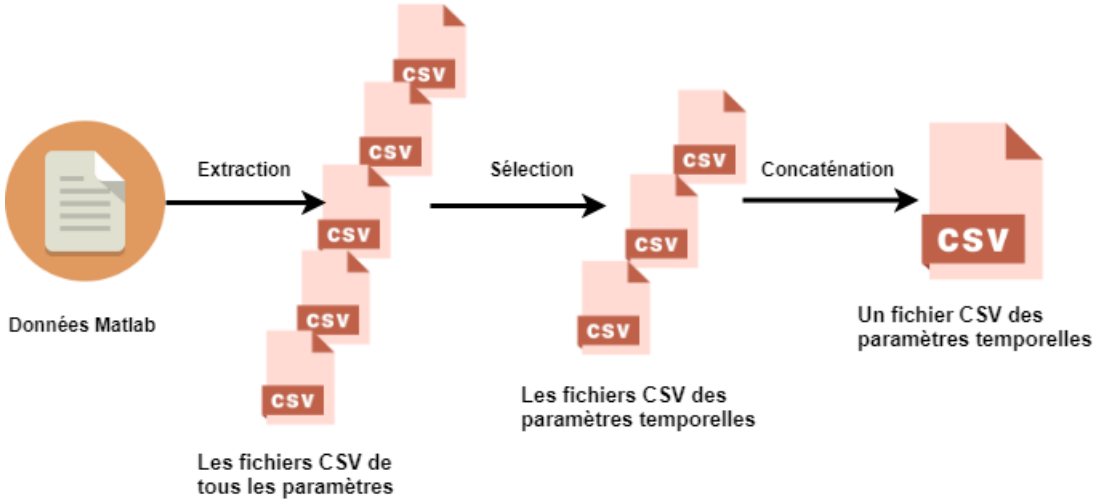


FIGURE 4.9 – Processus d'extraction et de sélection pour les données temporelles

```
Index(['alphaGait', 'DeplacementRider', 'DistanceAligne', 'DistanceEngage',
       'DistanceFWGait', 'DistanceFWRButte', 'ForcePied', 'ForcePiedAr',
       'ForcePiedAr.1', 'ForcePiedAr.2', 'ForcePiedAv', 'ForcePiedAv.1',
       'ForcePiedAv.2', 'ForceUPiedAr', 'ForceUPiedAv',
       'IndiceEfficacitePiedAr', 'IndiceEfficacitePiedAv',
       'IndiceEfficaciteTotal', 'Moment', 'MomentPAr', 'MomentPAv',
       'Puissance', 'PuissancePAr', 'PuissancePAv', 'ThetaM', 'ThetaMDeg',
       'ThetaMr', 'ThetaMrDeg', 'ThetaMrv', 'Time', 'VitesseRider',
       'Impulsion', 'ImpulsionPAr', 'ImpulsionPAv', 'Travail', 'TravailPAr',
       'TravailPAv'],
      dtype='object')
```

FIGURE 4.10 – Liste des paramètres temporelles obtenu.

Nous obtenons un vecteurs de 53 paramètres par observation, un exemple est donné par le tableau 4.9.1.

Distance Aligne	Force Pied	Moment	Puissance	...	Time	Travail PAv
0.10	884.25	138.18	2024.10	...	1.57	248096.81

TABLE 4.3 – Exemple d'une donnée temporelle (une observation).

4.9.2 Les données discrètes

Pour les données discrets nous avons suivi le même processus déjà expliqué dans la section précédente 4.9.1.

La liste des paramètres obtenues dans le fichier des données discrètes est représentée la figures suivantes 4.11

```
Index(['Alignement', 'AlphaGaitDmin', 'BasDeButte', 'BasDeButte2', 'Bip',
       'BraquetRider', 'c3d', 'CGR18', 'CGR28', 'ChuteGrille', 'CoupsDePedal',
       'CoupsDePedal.1', 'CoupsDePedal.2', 'CoupsDePedal.3', 'CoupsDePedal.4',
       'CoupsDePedal.5', 'CoupsDePedal.6', 'CoupsDePedal.7', 'CoupsDePedal.8',
       'CoupsDePedal.9', 'DateNaissanceRider', 'DatePassageRider',
       'DebutOrdre', 'DistanceDmin', 'DistanceRecul', 'Dmin', 'Engagement',
       'EngagementDmin', 'FinMarkerVisible', 'FinOrdre', 'FinSignalSensix',
       'FinSignalSonore', 'ForceGuidonMax', 'ForceUDepart', 'ForceUPiedArMax',
       'ForceUPiedAvMax', 'Franchissement', 'GaitDown', 'HauteurFWDmin',
       'HauteurFWRecul', 'ImpulsionParCoups', 'ImpulsionParCoupsPAr',
       'ImpulsionParCoupsPAv', 'longueurManivelle', 'MasseRider',
       'ModeleVeloRider', 'MoyennePuissanceBasDeButte',
       'MoyennePuissanceButteTotale', 'MoyennePuissancePremCassure',
       'PDenHaut', 'PGenHaut', 'pied2depart', 'PourcFuPremCassure',
       'PremCassure', 'PuissanceMaxBasDeButte', 'PuissanceMaxPremCassure',
       'Rate', 'RateForceDeveloppement', 'Recul', 'TailleRider',
       'ThetaManivelleDepart', 'ThetaManivelleRecul', 'TimeToPeak',
       'TpsBasDeButte', 'TpsDmin', 'TpsPassageGrille', 'TpsPremCassure',
       'TpsReaction', 'TpsRecul', 'TpsTroism', 'TravailParCoups',
       'TravailParCoupsPAr', 'TravailParCoupsPAv', 'Troism',
       'VitesseBasDeButte', 'VitessePassageGrille', 'VitessePremCassure',
       'VitesseTroism', 'VMaxBasDeButte', 'VMaxPremCassure'],
      dtype='object')
```

FIGURE 4.11 – Liste des paramètres discrets obtenu.

4.10 Modélisation

Cette phase correspond au choix, au paramétrage et au test de différents algorithmes. Nous utiliserons, en entrée de ces algorithmes, les fichiers résultats de la phase de prétraitement.

4.10.1 Les données temporelles

Pour la prédiction de valeurs continue qui est la force utile appliquée, il est clair d'utiliser une prédiction basée sur la régression. Il en existe deux grandes familles : les approches statistiques et les approches basées sur les réseaux de neurones.

- **Les approches basées sur les réseaux de neurones** : on peut citer à titre d'exemple les RNN ou les LSTM. Ces approches ont été exclues d'emblée à cause du faible nombre d'observations. En effet, on ne dispose que de 400 a 500 observations au maximum par essais, ce qui est clairement insuffisant pour des approches basées sur les réseaux de neurones.
- **Les approches statistiques** : ce sont les approches les plus anciennes et les mieux étudiées, on peut citer les modèles sans régularisation ou ceux avec une régularisation.

Nous avons opté pour un modèle avec une régularisation appelée "Ridge". Étant le modèle le plus général, disposant d'une régularisation de norme L2 ou régularisation de Thikonov[8]. Ce terme pénalise l'erreur lorsque on ajoute le carré de la somme des

valeurs absolues des coefficients. La régularisation peut être calibrée selon un paramètre lambda. La force de régularisation augmente avec lambda. Le but de la régularisation est de généraliser le modèle pour de nouvelles données et d'éviter le sur apprentissage.

Les paramètres sélectionnés dans les données temporelles ont été normalisées en traitant chaque paramètre (colonne) comme une variable aléatoire, en soustrayant sa moyenne et en le divisant par son écart-type. L'ensemble des variables suivront dans ce cas des lois normales centrées réduites. La formule de normalisation, pour chaque variable, est la suivante [3] :

$$x = \frac{X - \mu}{\sigma}$$

Où x est la nouvelle variable (ayant une moyenne nulle et un écart-type de 1), μ sa moyenne initiale et σ son écart-type.

Cette normalisation est nécessaire, car les paramètres contiennent plusieurs caractéristiques, certaines dans l'ordres des 10000 et d'autres comprises entre 0 et 1, la distance euclidienne utilisée directement est inadaptée, à cause des différences d'échelles, une différence de 2 dans ceux de l'échelle de 10000 aurait plus de poids qu'une différence de 90% dans un paramètre compris entre 1 et 0, or avec la normalisation, toutes les variables ont le même poids (ou la même influence sur la régression).
Un exemple de la normalisation du profil, représenté par le tableau 4.9.1, est donné par le tableau 4.10.1.

Distance Aligne	**Force Pied**	**Moment**	**Puissance**	**...**	**Time**	**Travail PAv**
-0.44	-1.01	-0.034585	1.30	...	1.72	1.85

TABLE 4.4 – Exemple d'une donnée temporelle normalisé.

Nous pouvons voir que les valeurs sont toutes à la même échelle, dépourvues d'unités et ayant la même influence sur la régression.
À noter que les valeurs négatives signifient que la valeur initiale était inférieure à la moyenne.

La formule de la régularisation linéaire de norme L2 proposé ci-dessus 4.10.1 est donnée par l'équation suivante. Celle-ci est inspirée par la formule proposée par [7] :

$$j(\omega) = \sum_{i=1}^{m} (y - \sum_{j=0}^{n} \omega_j \times x_i j)^2 + \lambda \sum_{j=0}^{n} \omega_j^2 \tag{4.1}$$

Où ω est un vecteur de poids lié à chaque variable x_j, m est le nombre d'observations et n c'est le nombre de variables. En revenant a l'équation 4.1 on peut voir que si $\lambda \to 0$, la fonction de coût devient similaire a une régression linéaire simple c'est-à-dire une régression sans régularisation.

4.10.2 Les données discrètes

Pour la prédiction du temps en bas de butte en fonction des caractéristiques du pilote et le matériel utilisé, nous avons opté pour une approche appelée le "Model Selection"

[2]. Étant la méthode qui nous permet de calculer plus de paramètres (colonne) donc plus de données vu le faible nombre d'observations (60 observations en total). La génération des paramètres calculer est toutes les combinaisons possible entre les paramètres selon un ordre de polynôme choisi. Le "model Selection" nous permet aussi d'itérer sur de différents modèles déjà existants tels que Ridge, Lasso, ElasticNet, LinearRegression etc, pour trouver le modèle le mieux adapté a notre problème. Dans cette méthode il est impératif de diviser l'ensemble d'obseravtion en trois sous ensemble : l'ensemble d'entraînement, de test et de la validation pour éviter le surapprentissage (overfitting).

Le choix du meilleur modèle et polynôme pour notre problème se fait selon le meilleur R2 score dans l'ensemble de test et de validation à la fois, et cela qui nous permet d'éviter le surapprentissage.

4.11 Evaluation

Dans cette partie nous détaillerons les métriques d'évaluation utilisées. Comme le but principal de l'approche est de la force utile pour les données temporelles et le temps (la performance) pour les données discrètes, nous avons choisi la RMSE (Root Mean Square Error) comme métrique d'erreur 2.4.2 et Le R2 score pour évaluer la régression. Nous avons décidé d'évaluer les deux parties de notre approche de la même méthodologie, étant valide pour les deux types de données. Nous allons détailler, dans ce qui suit, le procédé suivi pour évaluer l'approche proposée. La méthode proposée ne peut être évalué par rapport à d'autre méthode car il n'existe pas encore des méthodes de références, notre méthode est la première.

4.11.1 Evaluation des prédictions pour la force utile, Temps

Ci-dessous la méthodologie utilisée, sous forme d'algorithme.

Algorithm 1 Méthode d'évaluation

Diviser l'ensemble d'observation en deux sous ensemble : entrainement 85% et test %15 ;
Entrainer le modèle sur l'ensemble d'entrainement ;
Prédire les valeur n de l'ensemble de test à travers le modèle entrainé ;
Calculer la RMSE et le R2 score ;

4.12 Conclusion

Nous avons vu au cours de ce chapitre les différentes phases suivies lors de la conception de notre solution. Nous avons détaillé la méthodologie utilisée, nous avons également projeté notre système sur l'architecture actuelle de l'entreprise. Nous avons ensuite exposé l'analyse faite sur l'ensemble des données à notre disposition et également le prétraitement effectué sur ces dernières. Enfin, Nous avons justifié l'utilisation des différents algorithmes proposé et nous avons détaillé notre méthodologie d'évaluation. La conception des différentes approches était la partie la plus importante, la plus longue

et la plus difficile à réaliser durant ce stage. Pour aboutir à une bonne solution, nous avons suivi un procédé itératif. Le résultat final, qui a été exposé dans ce chapitre, a donc été obtenu après plusieurs aller-retour avec la partie réalisation (chapitre) car l'évaluation n'est faite qu'à cette étape-là.

Chapitre 5

RÉALISATION

5.1 Introduction

Après avoir présenter notre méthodologie d'analyse et de conception logicielle, nous passons maintenant à sa réalisation. Nous présenterons, en premier lieu, l'environnement dans lequel notre système a été construit, entre autres, les différentes technologies et outils utilisés. Nous exposerons ensuite les différents tests et résultats qui nous ont aidés à déterminer les différents paramètres optimaux pour les algorithmes choisis.

5.2 technologies utilisées

Pour réaliser les différents tests et effectuer le développement des modules mentionnés lors du chapitre précédent, nous avons dû utiliser plusieurs technologies. Notre choix s'est porté sur des technologies très répandues dans le domaine de la science des données et facilitant le prototypage.

5.2.1 Python

FIGURE 5.1 – Logo de Python

Python [1] est un langage de programmation interprété, multi-paradigme et multiplateformes [10]. Il est le langage de programmation le plus utilisé dans le domaine du Machine Learning, du Big Data et de la Data Science. Il est doté d'un typage dynamique fort, d'une gestion de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions. Python a été créé au début des années 1990 par Guido van Rossum à Stichting Mathematisch Centrum [4] aux Pays-Bas pour succéder à un langage appelé ABC. Nous l'avons utilisé, dans sa version 3.6.5, comme langage principal de développement

1. `https://www.python.org/`

5.2.2 Gnu Octave

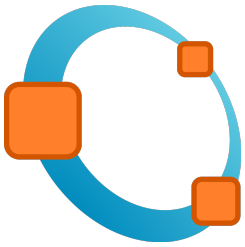


FIGURE 5.2 – Logo d'Octave

Gnu Octave est un logiciel libre qui dispose de nombreux outils pour résoudre les problèmes courants d'algèbre linéaire. Il est utilisé pour le calcul numérique. Il fournit une interface de ligne de commande pratique pour la résolution numérique de problèmes linéaires et non linéaires, ainsi que pour la réalisation d'autres expériences numériques à l'aide d'un langage principalement compatible avec Matlab. Nous l'avons utilisé principalement pour l'extraction des fichiers CSV à partir des données matricielles fournies.

5.2.3 Matlab

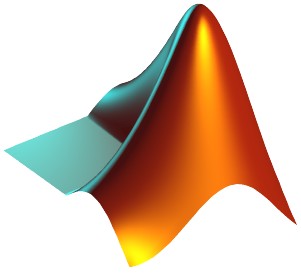


FIGURE 5.3 – Logo de Matlab

MATLAB [2] "matrix laboratory" est un langage de programmation de quatrième génération, il est utilisé à des fins de calcul numérique.

Il permet de manipuler des matrices et d'exprimer directement les mathématiques sous forme de tableaux et de matrices. Il est séveloppé par la société The MathWorks [3]. Nous l'avons utilisé pour la compréhension du format de données fournies, comme il propose une interface simple pour afficher les imbrications et les structures dans une matrice.

5.3 Outils utilisés

Les outils utilisés, en adéquation avec les technologies citées précédemment, seront énumérés ci-dessous.

5.3.1 Pycharm

FIGURE 5.4 – Logo de Pycharm

PyCharm [4] est un environnement de développement intégré (IDE) utilisé pour la programmation en Python. Il est développé par la société tchèque JetBrains [5]. Il fournit une analyse de code, un débogueur graphique, un testeur d'unité intégré, une intégration de systèmes de contrôle de version (VCS) et prend en charge le développement

2. https://fr.mathworks.com/products/matlab.html
3. https://www.mathworks.com/company/aboutus.html
4. https://www.jetbrains.com/pycharm/
5. https://www.jetbrains.com/company/

Web avec Django.PyCharm Community Edition est publié sous la licence Apache [6].

Il existe également une édition professionnelle avec des fonctionnalités supplémentaires - publiée sous une licence propriétaire.

5.3.2 Anaconda

Anaconda [7] est une distribution libre et open source des langages de programmation Python et R appliquée au développement d'applications dédiées à la science des données et à l'apprentissage automatique, qui vise à simplifier la gestion des packages et le déploiement. Cet outil permet également de créer des environnements d'exécution indépendants et gérer les conflits entre packages, dont leur nombre dépasse les 1500.

FIGURE 5.5 – Logo de Anaconda

5.4 Tests et résultats

Nous présenterons dans cette partie les différents tests et les résultats obtenus en exécutant ceux-ci. Ces tests nous ont permis de comparer entre les différentes approches choisies et de choisir les meilleurs paramètres pour ces dernières.

5.4.1 Prédiction de la force utile

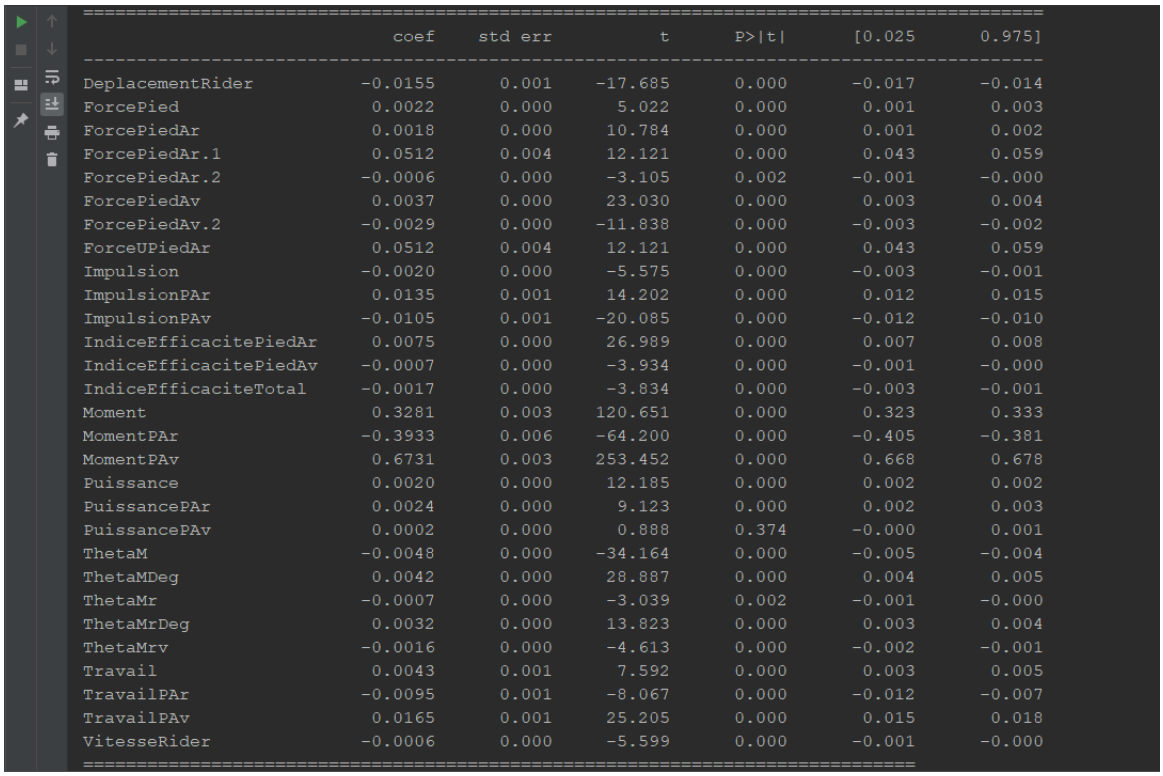
Pour la prédiction de la force utile nous avons utilisé une approche basée sur la régression linéaire et plus précisément nous avons utilisé le modèle "Ridge" déjà mentionné dans la section 4.10.1. Le but est à la fois prédire la force utile et réduire le nombre de paramètres pour ne garder que les paramètres responsables de l'amélioration de cette dernière. Pour ce faire nous nous somme basé sur le test statistique appelé le test de Student et aussi sur des connaissances à travers les réunions avec nos collègues de PPRIME.

Test de Student

En statistique, le test de Student, ou test t, est un ensemble de tests statistiques paramétriques où la statistique de test calculée suit une loi de Student lorsque l'hypothèse nulle est vraie. Nous l'avons utilisé pour tester si les coefficients de la régression sont significatif ou non, pour supprimer les variables associer à ces coefficients non significatifs.

Dans la figure ci-dessous 5.6 nous montrons le résultat de l'exécution de cette approche.

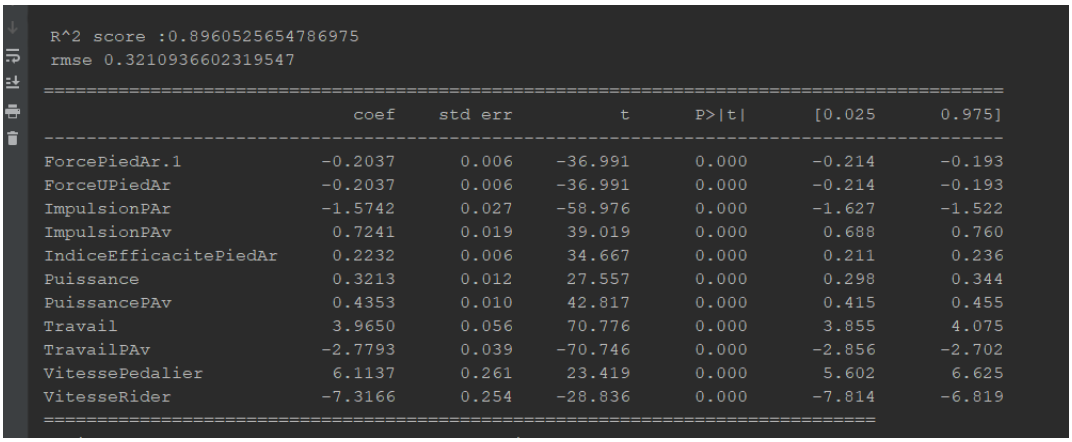
6. https://www.apache.org/licenses/LICENSE-2.0
7. https://www.anaconda.com/



	coef	std err	t	P>|t|	[0.025	0.975]
DeplacementRider	-0.0155	0.001	-17.685	0.000	-0.017	-0.014
ForcePied	0.0022	0.000	5.022	0.000	0.001	0.003
ForcePiedAr	0.0018	0.000	10.784	0.000	0.001	0.002
ForcePiedAr.1	0.0512	0.004	12.121	0.000	0.043	0.059
ForcePiedAr.2	-0.0006	0.000	-3.105	0.002	-0.001	-0.000
ForcePiedAv	0.0037	0.000	23.030	0.000	0.003	0.004
ForcePiedAv.2	-0.0029	0.000	-11.838	0.000	-0.003	-0.002
ForceUPiedAr	0.0512	0.004	12.121	0.000	0.043	0.059
Impulsion	-0.0020	0.000	-5.575	0.000	-0.003	-0.001
ImpulsionPAr	0.0135	0.001	14.202	0.000	0.012	0.015
ImpulsionPAv	-0.0105	0.001	-20.085	0.000	-0.012	-0.010
IndiceEfficacitePiedAr	0.0075	0.000	26.989	0.000	0.007	0.008
IndiceEfficacitePiedAv	-0.0007	0.000	-3.934	0.000	-0.001	-0.000
IndiceEfficaciteTotal	-0.0017	0.000	-3.834	0.000	-0.003	-0.001
Moment	0.3281	0.003	120.651	0.000	0.323	0.333
MomentPAr	-0.3933	0.006	-64.200	0.000	-0.405	-0.381
MomentPAv	0.6731	0.003	253.452	0.000	0.668	0.678
Puissance	0.0020	0.000	12.185	0.000	0.002	0.002
PuissancePAr	0.0024	0.000	9.123	0.000	0.002	0.003
PuissancePAv	0.0002	0.000	0.888	0.374	-0.000	0.001
ThetaM	-0.0048	0.000	-34.164	0.000	-0.005	-0.004
ThetaMDeg	0.0042	0.000	28.887	0.000	0.004	0.005
ThetaMr	-0.0007	0.000	-3.039	0.002	-0.001	-0.000
ThetaMrDeg	0.0032	0.000	13.823	0.000	0.003	0.004
ThetaMrv	-0.0016	0.000	-4.613	0.000	-0.002	-0.001
Travail	0.0043	0.001	7.592	0.000	0.003	0.005
TravailPAr	-0.0095	0.001	-8.067	0.000	-0.012	-0.007
TravailPAv	0.0165	0.001	25.205	0.000	0.015	0.018
VitesseRider	-0.0006	0.000	-5.599	0.000	-0.001	-0.000

FIGURE 5.6 – Affichage des coefficients et des valeurs obtenus par le test de Student.

La figure ci-dessus 5.6 nous présente un tableau qui nous permet d'afficher les résultats du test de Student pour chaque coefficient et de savoir si ce dernier est significatif. Pour savoir si un coefficient est significatif il faut définir un niveau de signification qui généralement égale à 0.05. En d'autres termes on dit qu'on a la probabilité de 5% que le coefficient ne soit pas significatif. Par conséquent on ne prend que les coefficients qui ont une valeur de test de Student strictement inférieur a cette valeur de 0.05. En revenant à notre exemple on va s'intéresser à la 4e colonne pour supprimer les coefficients non significatifs. Pour le paramètre PuissancePAv on peut dire qu'il n'est pas significatif vu qu'il est supérieur au niveau de signification choisi ($0.374 > 0.05$), de manière itératif nous répétons ce processus pour éliminer les paramètres non significatifs. Le résultat de cette opération est présenté dans la figure ci-dessous 5.7.



```
R^2 score :0.8960525654786975
rmse 0.3210936602319547
==============================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------
ForcePiedAr.1             -0.2037      0.006    -36.991      0.000      -0.214      -0.193
ForceUPiedAr              -0.2037      0.006    -36.991      0.000      -0.214      -0.193
ImpulsionPAr              -1.5742      0.027    -58.976      0.000      -1.627      -1.522
ImpulsionPAv               0.7241      0.019     39.019      0.000       0.688       0.760
IndiceEfficacitePiedAr     0.2232      0.006     34.667      0.000       0.211       0.236
Puissance                  0.3213      0.012     27.557      0.000       0.298       0.344
PuissancePAv               0.4353      0.010     42.817      0.000       0.415       0.455
Travail                    3.9650      0.056     70.776      0.000       3.855       4.075
TravailPAv                -2.7793      0.039    -70.746      0.000      -2.856      -2.702
VitessePedalier            6.1137      0.261     23.419      0.000       5.602       6.625
VitesseRider              -7.3166      0.254    -28.836      0.000      -7.814      -6.819
==============================================================================
```

FIGURE 5.7 – les paramètres responsables de l'amélioration de la force utile

On voit bien dans cette figure qu'on a réussie à réduire le nombre de paramètres toute en gardant de bon résultat de prédiction pour la force utile avec un score d'évaluation 89.6% et on voit aussi que tous les paramètres restant ont des coefficients qui sont significatif statistiquement. Donc maintenant on peut expliquer la force utile en utilisant que ces 11 paramètres à travers ce modèle. Ces 11 paramètres sont le minimum possible sinon en réduisant encore plus, la performance du modèle diminuera et nous aurons des résultats erronés.

5.4.2 Prédiction du temps pour les données discrètes

On rappelle que la technique du "Model selection" est appliquée pour la prédiction du temps en bas de butte dans les données discrètes.

Model Selection

On rappelle qu'on a opté pour cette solution en premier lieu à cause du manque de données et aussi pour améliorer les prédictions en utilisant des formules polynomiales au lieu d'une formule linéaire simple.

Dans un premier temps, nous avons sélectionné que quelques paramètres parmi la liste des paramètres discrets montrés dans la figure 4.11, les premiers paramètres choisis sont : la longueur manivelle, le braquet, le poids et la taille du pilote tout cela en fonction du temps. Après avoir pris ces paramètres on a généré une modélisation polynomiale en calculant tous les combinaisons possibles entres ces paramètres. Ensuit on divise l'ensemble d'observation en trois sous ensembles train-test-cross validation. Nous avons pris 70% pour l'ensemble d'entraînement, 15% pour le test et 15% pour l'ensemble de la validation du test. L'exécution se fait de manière itératif pour chaque modèle choisi et selon un plafond de polynôme comme montré dans la figures ci-dessus 5.8

```
frames = frames[['Braquet', 'MasseRider', 'TailleRider',
                 'longueurManivelle', 'TimeEnd']]
models = [Ridge(alpha=1), LinearRegression(), Lasso(), ElasticNet()]
for model in models:
    print("-" * 50)
    print(model)
    print("=" * 50)
    for degree in range(1, 10):
        """..."""

        X = frames.drop(['TimeEnd'], axis=1)
        y = frames['TimeEnd']
        y = y.fillna(y.mean())

        polynomial_features = PolynomialFeatures(degree=degree)
        X = polynomial_features.fit_transform(X)

        # test set
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15, random_state=11)
        # cross validation set
        X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.15, random_state=11)

        lm = model
        lm.fit(X_train, y_train)
        y_pred = lm.predict(X_test)

        y_pred_val = lm.predict(X_val)
        r2_score_test = r2_score(y_test, y_pred)
        r2_score_val = r2_score(y_val, y_pred_val)
```

FIGURE 5.8 – Technique du "Model Selction".

Cette figure ci-dessus montre l'application de cette technique. A noter que le nombre de colonnes à prendre en compte, le degré maximum et les modèles à tester sont un choix personnel.

Après avoir vu la technique du Model Selction maintenant nous allons voir les résultats obtenus et comment pourrait-on les interprétés pour faire le bon choix du modèle et du degré associé. La figure ci-dessous 5.9 montre les résultats obtenus après l'entraînement des différents modèles.

```
--------------------------------------------------
Ridge(alpha=1, copy_X=True, fit_intercept=True, max_iter=None, normalize=False,
      random_state=None, solver='auto', tol=0.001)
==================================================
degree 2, test_R2 0.707, val_R2 0.618
degree 3, test_R2 0.667, val_R2 0.857
degree 4, test_R2 0.666, val_R2 0.857
degree 5, test_R2 0.666, val_R2 0.857
degree 6, test_R2 0.665, val_R2 0.857
degree 7, test_R2 0.664, val_R2 0.857
degree 8, test_R2 0.663, val_R2 0.857
degree 9, test_R2 0.662, val_R2 0.857
--------------------------------------------------
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
==================================================
--------------------------------------------------
Lasso(alpha=1.0, copy_X=True, fit_intercept=True, max_iter=1000,
      normalize=False, positive=False, precompute=False, random_state=None,
      selection='cyclic', tol=0.0001, warm_start=False)
==================================================
degree 5, test_R2 0.571, val_R2 0.250
degree 6, test_R2 0.587, val_R2 0.284
degree 7, test_R2 0.624, val_R2 0.366
degree 8, test_R2 0.673, val_R2 0.479
degree 9, test_R2 0.715, val_R2 0.599
--------------------------------------------------
ElasticNet(alpha=1.0, copy_X=True, fit_intercept=True, l1_ratio=0.5,
           max_iter=1000, normalize=False, positive=False, precompute=False,
           random_state=None, selection='cyclic', tol=0.0001, warm_start=False)
==================================================
degree 5, test_R2 0.571, val_R2 0.250
degree 6, test_R2 0.589, val_R2 0.286
degree 7, test_R2 0.624, val_R2 0.369
degree 8, test_R2 0.675, val_R2 0.484
degree 9, test_R2 0.716, val_R2 0.604
```

FIGURE 5.9 – Technique du "Model Selction".

Dans la figure ci-dessus nous avons pris en compte que les résultats qui ont un R2 score supérieur a 50% (0.5) pour faciliter le choix et de ne pas afficher des résultats qui sont moins intéressant, par exemple pour le modèle "LinearRegression" on remarque qu'aucun résultat n'est affiché donc on l'élimine directement.

Le modèle "Ridge" avec le polynôme de degré 3 est le bon choix ici, on a une modélisation pas très complexe vu qu'on est seulement en ordre 3, ce qui nous a permis d'obtenir 30 autres paramètres calculés. Le même modèle avec le degré 2 peut être pris en compte, ici on aura que 10 autres paramètres calculés.

5.4.3 Prédiction du temps avec l'intégration de paramètres Temporelles

On a voulu intégrer plus de paramètre a ce modèle qui a donné de bons résultats ; en premier lieu on a voulu intégrer la puissance avec ces paramètres disettes ce qui est pas possible, donc on a choisi de prendre la moyenne de la puissance par coup de pédale pour chaque essai, comme dans un départ de BMX Race on a 4 coups de pédale donc on aura 4 nouveaux paramètres ou colonnes qui sont la moyenne de la puissance de chaque coup de pédale. On a réussi à intégrer ces derniers avec les paramètres discrets. Et on a lancé l'algorithme. Les résultats était mauvais par rapport à chaque modèle

et degré, et cela est dû a deux raisons principales. Le faible nombre d'observations et la complexité de la nouvelle distribution, ce qui rend la modélisation plus complexe et très difficile à trouver même impossible avec ce faible nombre d'observations.

```
--------------------------------------------------
Ridge(alpha=1, copy_X=True, fit_intercept=True, max_iter=None, normalize=False,
      random_state=None, solver='auto', tol=0.001)
==================================================
degree 1, test_R2 0.345, val_R2 0.482
degree 2, test_R2 -0.586, val_R2 -1.839
--------------------------------------------------
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
==================================================
degree 1, test_R2 0.364, val_R2 0.337
degree 2, test_R2 -32.732, val_R2 -179.179
--------------------------------------------------
Lasso(alpha=1.0, copy_X=True, fit_intercept=True, max_iter=1000,
      normalize=False, positive=False, precompute=False, random_state=None,
      selection='cyclic', tol=0.0001, warm_start=False)
==================================================
degree 1, test_R2 0.297, val_R2 0.297
degree 2, test_R2 -0.851, val_R2 0.153
--------------------------------------------------
ElasticNet(alpha=1.0, copy_X=True, fit_intercept=True, l1_ratio=0.5,
           max_iter=1000, normalize=False, positive=False, precompute=False,
           random_state=None, selection='cyclic', tol=0.0001, warm_start=False)
==================================================
degree 1, test_R2 0.307, val_R2 0.293
degree 2, test_R2 -0.970, val_R2 0.128
```

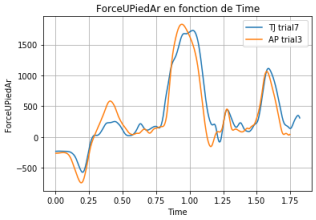
FIGURE 5.10 – Technique du "Model Selction" avec intégration des paramètres.

Dans la figure ci-dessus 5.10 on remarque bien qu'aucun des modèles n'a donné de bon résultats, vu la complexité de la nouvelle distribution.

5.4.4 Visualisation de données

Vu le nombre important des paramètres issus des capteurs a lesquelles on rajoute les paramètres calculés ça devient une tâche difficile de visualiser les données d'une façon flexible surtout quand on veut afficher un paramètre en fonction de 2 autres en même temps ou bien de comparer deux, trois pilotes, etc. Cela d'une part, et d'autre part le projet conçu pour la FFC pour réaliser une application WEB qui permet d'afficher les performances des pilotes dans leurs cours d'une manière simple. Donc nous avons proposé une solution qui répond a ce problème qui prend la base de données déjà fournie à travers l'extraction des fichiers CSV. Les figures ci-dessous nous montrent les résultats de cette proposition.

```
obj = PlotBestTestPerPilote()
columns = ['ForceUPiedAr', 'ForceUPiedAv']
xaxis = ['Time']
obj.plot_by_pilotes_names_trials_nums_dates(data_indir='C:\\Users\\mekhezzr\\PycharmProjects\\bmx_race\\data_v2',
                                            pilotes_names=['ThomasJouve', 'ArthurPilard'],
                                            trials_nums=[7, 3],
                                            dates_trial=['2018-06-20', '2018-06-21'],
                                            x_axis=xaxis,
                                            y_axis=columns
                                            )
```



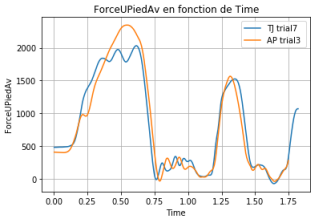


FIGURE 5.11 – Comparaison entre deux pilotes pour un nombre de paramètres(colonnes) supérieur à 2.

La figure ci-dessus 5.11 nous montre une fonction qui prend en paramètres la base de données des fichiers csv, les noms des pilotes, le numéro d'essai du pilote, la date de cette essai, les paramètres (colonnes) qu'on veut tracer et en fonction de quelles colonnes, et elle retourne un graphe qui trace la performance de ses pilotes d'une façon superposée pour pouvoir comparer entre ses pilotes. Dans cette exemple on voit le retour de 2 graphes car dans notre implémentions nous avons choisi de séparer les colonnes qu'on veut tracer pour faciliter l'interprétation. Ici nous avons un graphe pour la force utile du pied arrière pour les deux pilotes Arthure Pilard et Thomas Jouve, et un deuxième pour la force utile du pied avant, toujours, pour les mêmes pilotes.

```
obj = PlotBestTestPerPilote()
columns = ['Puissance']
xaxis = ['Time']
obj.plot_by_pilotes_names_trials_nums_dates(data_indir='C:\\Users\\mekhezzr\\PycharmProjects\\bmx_race\\data_v2',
                                            pilotes_names=['ThomasJouve', 'RomainRacine', 'SimbaDarnand'],
                                            trials_nums=[7, 3, 4],
                                            dates_trial=['2018-06-20', '2018-06-22', '2018-06-21'],
                                            x_axis=xaxis,
                                            y_axis=columns
                                            )
```

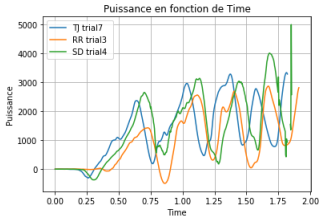


FIGURE 5.12 – Comparaison entre plusieurs pilotes.

Dans la figure ci-dessus 5.12 nous présentons cette même fonction pour plus de deux pilotes, on n'a pas mis une limite pour les pilotes qu'on veut comparer, tant que

c'est intéressant et interprétable pour nous on peut ajouter d'autre pilote à comparer dans un même graphe. Ici nous avons choisi par exemple de comparer la puissance entre les pilotes Thomas Jouve, Romain Racine et Simba Darnand.

```
obj = PlotBestTestPerPilote()
columns = ['Moment']
xaxis = ['Time']
obj.plot_by_pilotes_names_trials_nums_dates(data_indir='C:\\Users\\mekhezzr\\PycharmProjects\\bmx_race\\data_v2',
                                            pilotes_names=['RomainMahieu', 'RomainMahieu'],
                                            trials_nums=[1, 8],
                                            dates_trial=['2018-12-13', '2018-12-13'],
                                            x_axis=xaxis,
                                            y_axis=columns
                                            )
```

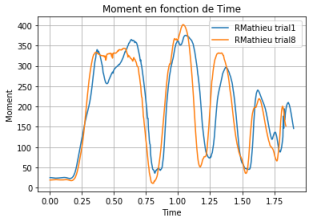


FIGURE 5.13 – Comparaison entre plusieurs essais du même pilote.

La figure 5.13 montre qu'on peut aussi à travers cette fonction comparer les diffèrent essais du même pilote, et cela avec un appel de la fonction avec le même nom pilote plus d'une fois avec un numéro d'essai diffèrent comme montré dans cette figure.

Nous avons choisi de faire qu'une seule fonction pour répondre aux besoins de nos collaborateurs de PPRIME pour facilité l'intégration de notre code avec le leurs. Le code est partagé à travers github avec nos collaborateurs de PPRIME.

5.5 Conclusion

Dans ce chapitre, nous avons passé en revue les différents outils et technologies utilisés pour élaborer notre solution. Nous avons détaillé, les différents tests et résultats. Ensuite nous avons présenté les méthodes de visualisation de données. L'équipe RoBioSS a manifesté sa satisfaction envers la solution proposée, et nous a fait part, lors de la dernière réunion, de sa volonté d'intégrer ce module dans l'application web en cours de développement.

Chapitre 6

Gestion de projet

6.1 Introduction

Dans ce chapitre, nous relaterons différents aspects liés à la gestion du projet. Nous détaillerons le suivi du projet, notamment les outils utilisés, les interactions avec les différents encadrants et le personnel de PPRIME. Nous détaillerons également les livrables qui ont permis à l'ensemble des collaborateurs d'être tenus informés de l'état d'avancement du projet. Nous clôturerons ce chapitre par un bilan général du stage.

6.2 Les outils collaboratifs

Nous présenterons dans ce qui suit les différents outils collaboratifs utilisés pour permettre le partage de certaines ressources et le suivi détaillé du projet.

6.2.1 ShareLaTeX

ShareLaTeX [1] est un éditeur LaTeX en ligne, collaboratif, en temps réel et disposant d'un compilateur PDF. Le laboratoire dispose d'une version en local accessible après authentification, cette version inclut les fonctionnalités suivantes :

- la collaboration en temps réel ;
- la vérification orthographique dans différentes langues ;
- la consultation de l'historique de modification ;
- la compilation et la visualisation en PDF sur la navigateur.

Nous l'avons utilisé pour la rédaction du rapport.

FIGURE 6.1 – logo de ShareLaTeX

1. `https://www.sharelatex.com/`

6.2.2 Github

Github [2] est un logiciel libre de gestion de versions qui permet de suivre l'évolution d'un code source (principalement) et le travail en équipe. Nous l'avons utilisé pour faciliter l'application de la méthodologie de prototypage et garder une trace des modifications effectuées. Nous l'avons utilisé dans sa version 2.18.0 pendant toute la durée du stage.

FIGURE 6.2 – logo de Github

6.2.3 Trello

Trello est un outil en ligne, ergonomique et gratuit. Nous utilisons cet outil pour organiser nos tâches, consigner les informations essentielles et tenir un planning avec tous les membres de notre équipe. Grâce à cet outil nous simplifions le suivi de notre projet. Ce service en ligne nous aide à mieux organiser nos activités.

FIGURE 6.3 – logo de Trello

6.2.4 Google Drive

Google Drive [3] est un service de stockage et de partage de fichiers dans le cloud lancé par la société Google. Il regroupe Google Docs, Sheets et Slides et Drawings, est une suite bureautique permettant de modifier des documents, des feuilles de calcul, des présentations etc. Nous l'avons utilisé pour partager différents documents, et les différentes présentations avec l'équipe d'encadrement.

FIGURE 6.4 – logo de Google Drive

6.3 Suivi du projet

Ce projet a été suivi par trois entités : L'UABT représentée par l'encadrant pédagogique (M. Houcine Matallah), le Laboratoire d'Informatique et d'Automatique pour les Systèmes (LIAS) à travers l'encadrants (M. Amin Messmoudi) et enfin, l'institut PPRIME , représenté par l'équipe RoBioSS (représentée par M. Mathieu Domalain).

2. https://github.com/
3. https://www.google.com/drive/

6.4 Planning

Le projet s'est déroulé dans les locaux du laboratoire sur une période s'étalant du 04 Mars 2019 au 19 juillet 2019. La figure 6.5 montre le planning des phases du projet selon un diagramme de Gantt.

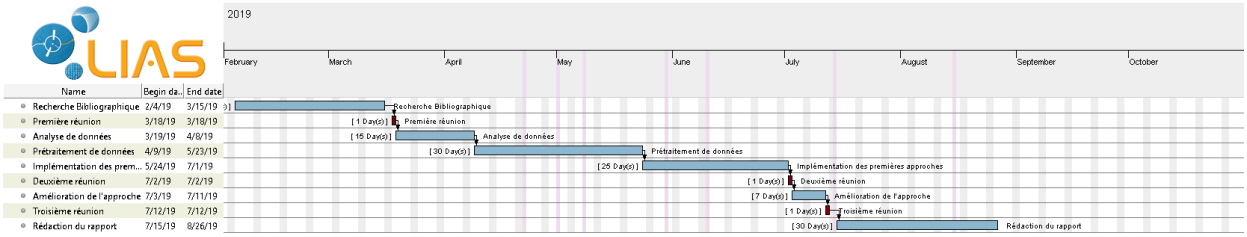


FIGURE 6.5 – Planning du projet

6.5 Livrables

L'ensemble des livrables ci-dessous ont été créé pendant la durée du stage et ont été partagés avec l'ensemble des encadrants et l'équipe RoBioSS

- **code source** : ensemble de fichiers en langage python concernant l'ensemble des fonctionnalité développées, à savoir, le prétraitement, la prédiction et la visualisation des données.
- **Notice d'utilisation et de de l'analyse de données** : ensemble de documents au format PDF décrivant le détail du travail effectué et la manière dont on peut l'exploiter pour assurer la continuité.
- **présentations** : les trois supports visuels, aux formats PPTX et PDF utilisés lors des présentations avec l'équipe RoBioSS.

.

6.6 Bilan

Nous citerons maintenant les principaux éléments qui ont permis d'atteindre les objectifs du projet et la satisfaction de l'équipe RoBioSS, nous citerons également les difficultés rencontrées. Nous commencerons par les points forts :

- la présence quotidienne de l'encadrant dans l'établissement d'accueil et son assistance régulière ;
- les différentes réunions de travail organisées avec l'encadrant où l'ensemble des solutions étaient discutées et critiquées ;
- les différentes réunions de travail organisées avec l'équipe RoBioSS qui nous ont permis d'avoir le point de vue du donneur d'ordre et de nous concentrer sur ses exigences.

Concernant les difficultés rencontrées, nous pouvons citer :

— le manque de réactivité de l'équipe de RoBioSS par rapport à l'envoi des données demandées ;
— le faible volume de données rendant l'amélioration des prédictions des algorithmes extrêmement difficile ;
— L'absence d'une méthode de référence pour pouvoir comparer les résultats.

Chapitre 7

Conclusion et perspectives

7.1 Conclusion

Le projet rapporté dans ce mémoire est né du besoin qu'avait l'équipe RoBioSS en termes d'exploration de nouvelles approches pour la modélisation des départs des pilotes élite de BMX race. N'ayant pas assez d'expertise dans le domaine de l'apprentissage automatique, l'équipe a fait appel au LIAS pour proposer une approche d'apprentissage automatique afin d'avoir des modèles simplifiés et prédictifs.

Pour ce faire, nous avons partagé le déroulement du projet en deux grandes étapes. La première étant une phase d'exploration et de documentation où nous avons étudié les principales approches d'apprentissage automatique.

D'un autre coté, la seconde phase consistait à proposer une solution pour l'équipe RoBioSS. Nous avons commencé par étudier les données dont nous disposions. Nous nous sommes ensuite occupés du prétraitement des données, cette étape a consommé le plus de temps, notamment à cause du format fournit très complexe. Nous avons ensuite développé les différents algorithmes pour la proposition de modèle simplifié toute en gardant de bonne prédiction. Nous avons évalué leurs capacités à prédire à travers une métrique adaptée (R2 score). Nous avons ensuite fournit un manuel d'utilisation pour assurer la continuité du travail.

Finalement, nous avons obtenus des résultats qui répondent aux différentes attentes de l'équipe. Concernant les données temporelles nous avons proposé un modèle simplifié qui utilise que 11 paramètres et permet une prédiction de 89%. Concernant les données que nous avons appelés discrètes nous avons proposé un modèle prédicatif de la performance (temps) en se basant sur les caractéristiques du pilote ainsi que le matériel utilisé. Nous avons proposé une amélioration de cette approche en incluant des paramètres calculés à partir des données temporelles, cependant les résultats obtenus n'étais pas fiable, donc nous avons gardé la méthode actuelle.

Les résultats obtenus étant satisfaisants pour l'équipe RoBioSS, ils prévoient de tester notre solution sur de nouvelle données prochainement disponible afin de l'évaluer

dans un cadre plus diversifié.

Dans ce qui suit, nous tenons à proposer quelques pistes que l'équipe pourra approfondir dans le futur.

7.2 Perspectives

Ces perspectives, ont été les résultats de l'étude réalisée dans la synthèse bibliographique ainsi que les différentes discussions que nous avons eu durant les différentes réunions organisées avec RoBioSS.

— Passer à une solution basée sur l'apprentissage profond : d'après l'équipe RoBioSS de nouvelles essais sont mis en place ce qui implique la collecte de nouvelle données qui permettra le passage au réseau de neurones.
— Conception et réalisation d'une interface exploitant le travail d'analyse de prétraitement et de visualisation de données : l'équipe RoBioss ne dispose toujours pas d'un module qui permet d'exploiter ce travail sans passer par la ligne de commande, ce travail peut facilement s'intégrer dans une application Web par exemple.

7.3 Appréciation personnelle

Ce stage de fin d'études au sein du LIAS a été une expérience très enrichissante, tant sur le plan personnel que professionnel. Il m'a permis, à la fois, de m'immiscer dans le monde de la recherche mais aussi dans le monde du travail. Il m'a permis d'appliquer les connaissances acquises durant ma formation à l'UABT et de faire face à des problèmes concrets issus du monde réel.

Ayant acquis de nouvelles connaissances et compétences, j'estime que ce stage a convenablement valorisé ma formation de master en Génie Logiciel au sein de mon université, Abou Bekr Belakaid Tlemcen.

Bibliographie

[1] Richard S. A. Sutton and Andrew G. Barto. Reinforcement learning : An introduction. pages 1–417, 2010.

[2] Sylvain Arlot. A survey of cross-validation procedures for model selection. *Statistics Surveys*, page 40–79, 2010.

[3] Willett P. Bath P.A., Morris C.A. Effect of standardization on fragment-based measures of structural similarity. *Chemometrics*, pages 543–550, 1993.

[4] CWI. centrum wiskunde informatica. `https://www.cwi.nl/about`, 2019. [En ligne ; dernier accée 12-06-2019].

[5] IBM. CRoss Industry Standard Process. `https://www.ibm.com/support/knowledgecenter/fr/SS3RA7_sub/modeler_crispdm_ddita/clementine/crisp_help/crisp_overview.html`, 2008. [En ligne ; dernier accée 12-06-2019].

[6] IBM. Guide CRISP-DM de IBM SPSS Modeler. `https://inseaddataanalytics.github.io/INSEADAnalytics/CRISP_DM.pdf`, 2012. [En ligne ; dernier accée 14-06-2019].

[7] R. Tibshirani. J. Friedman, T. Hastie. The element of statistical learning. *Springer*, pages 79–91, 2008.

[8] Michel Kern. Problèmes inverses : aspects numériques. *Nature Methods*, page 138, 2011.

[9] G. Manku. Approximate frequency counts over data streams. *VLDB '02 - Proceedings of the 28th VLDB Conference*, pages 346 – 357, 2002.

[10] Python. About the python Technology. `https://docs.python.org/3/`, 2019. [En ligne ; dernier accée 12-06-2019].

[11] Nicole Rusk. Deep learning. *Nature Methods*, 13 :35, 2015.

[12] J.Friedman T.Hastie, R.Tibshirani. *Machine Learning Book*. McGraw-Hill Science/Engineering/Math ; (March 1, 1997), 2009.

[13] X.Chen Z.Hailat, A.Komarichev. *Deep Semi-Supervised Learning*, volume 2018-August. 2018.

Annexe A

Notice d'extraction et de prétraitement des données

A.1 Organisation de données

A.1.1 Format des données brutes

Les données sont fournies compressées sous le format (.rar). La première étape consiste donc à les décompresser.

La figure suivante montre comment les données sont organisées.

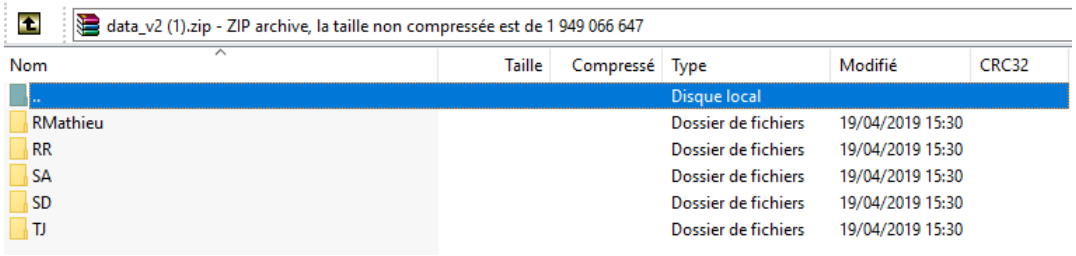


FIGURE A.1 – Organisation de données.

On a en effet un fichier compressé "data", dans lequel on trouve des dossiers qui ont le même nom que les pilotes qui on fait les essais. En entrant à chacun de ces dossiers on trouve des fichiers Matlab, le nombre de ces fichiers Matlab varie selon le nombre d'essais effectué par chacun de ces pilotes, car un fichier Matlab représente un essai de piloter. Les deux figures ci-dessous le montre :

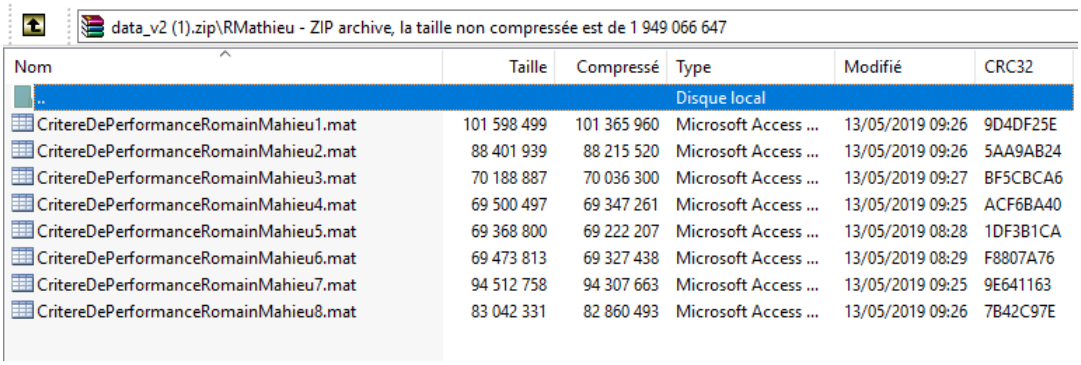


FIGURE A.2 – Contenue du fichier RMahieu.

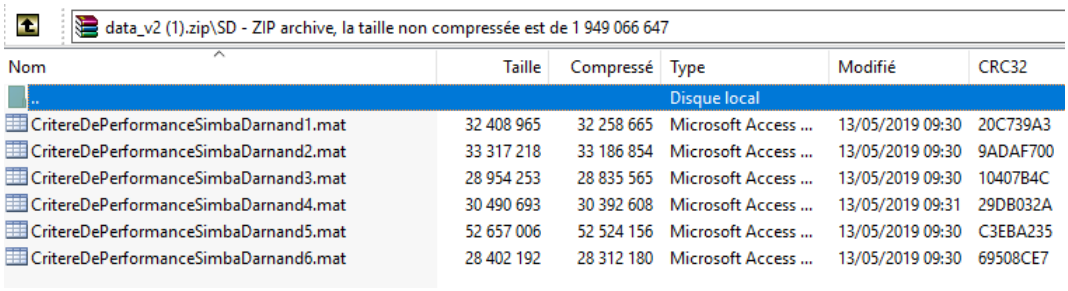


FIGURE A.3 – Contenu du Fichier SD.

Comme on peut le voir pour le contenu du fichier RMahieu A.2, on a 8 fichiers Matlab, donc on a bien 8 essais de Romain Mahieu, pareille pour SD (Simba Darnand A.3) on a les 6 essais qu'il a effectués.

A.1.2 Décompression de données

Cette étape se fait manuellement parce que ça nous ne nous prend pas de temps, car on va juste décompresser le fichier "data" pour obtenir un dossier data là où on a tous les sous fichiers sous les noms des pilotes comme le montre la figure qui suit :

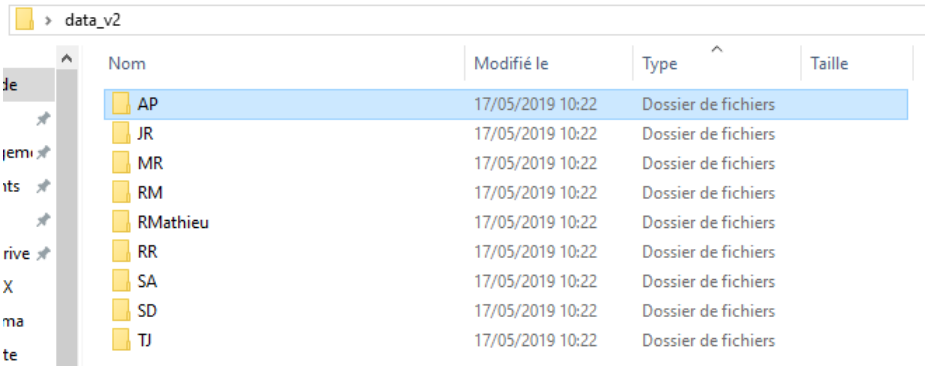


FIGURE A.4 – Contenu du fichier data décompressé

Comme vous l'avez déjà deviné dans chacun de ces fichiers on a les données Matlab A.5, déjà décrit dans la section précédente A.3.

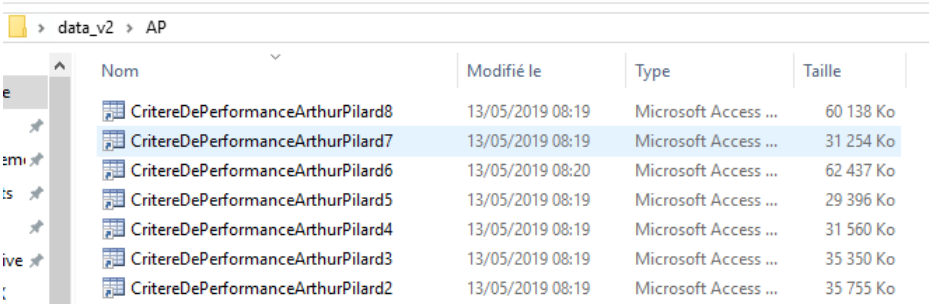


FIGURE A.5 – Contenu du dossier AP

Avec cette étape on vient de terminer la partie de la décompression de données reçu. Dans l'étape qui suit nous allons manipuler que les dossiers et les fichiers Matlab pour faire l'extraction et de concaténation de données.

A.2 Extraction des fichiers CSV

A.2.1 format des données Matlab

Dans cette section nous allons expliquer comment sont enregistré les données dans le fichier Matlab. Tous d'abord nous allons charger un fichier Matlab, et après afficher les données qui sont stockées de dedans. cela n'est malheureusement pas possible directement via un "disp (Data)", car en est fait le contenu ce n'est pas une matrice mais plutôt une structure, donc on va afficher avant les champs de cette structure, la figure A.6 montre ce processus.

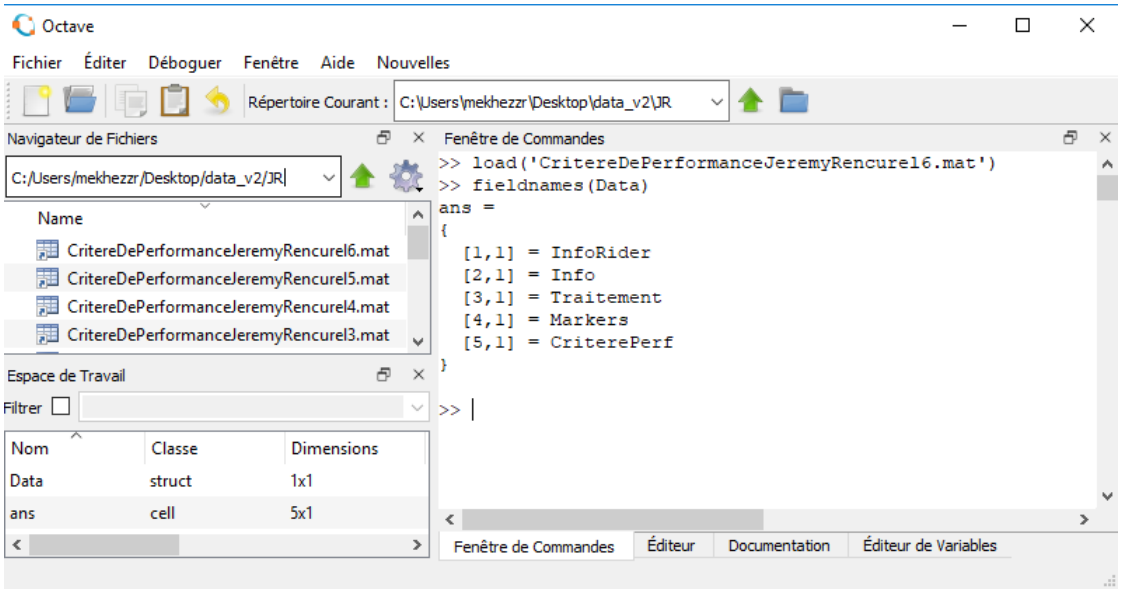


FIGURE A.6 – Les champs principale de la structure de donnée fournie.

On obtient un vecteur avec les noms des champs existant dans cette structure. Avec octave on ne peut pas voir si les champs obtenus sont des matrices ou bien eux aussi sont des structures, alors pour simplifier je vais le montrer à l'aide de Matlab parce que c'est visible et on peut le voir directement sans passer par la ligne de commande. Du coup on va accéder à Traitement pour voir ce qu'il y' a à l'intérieur A.7.

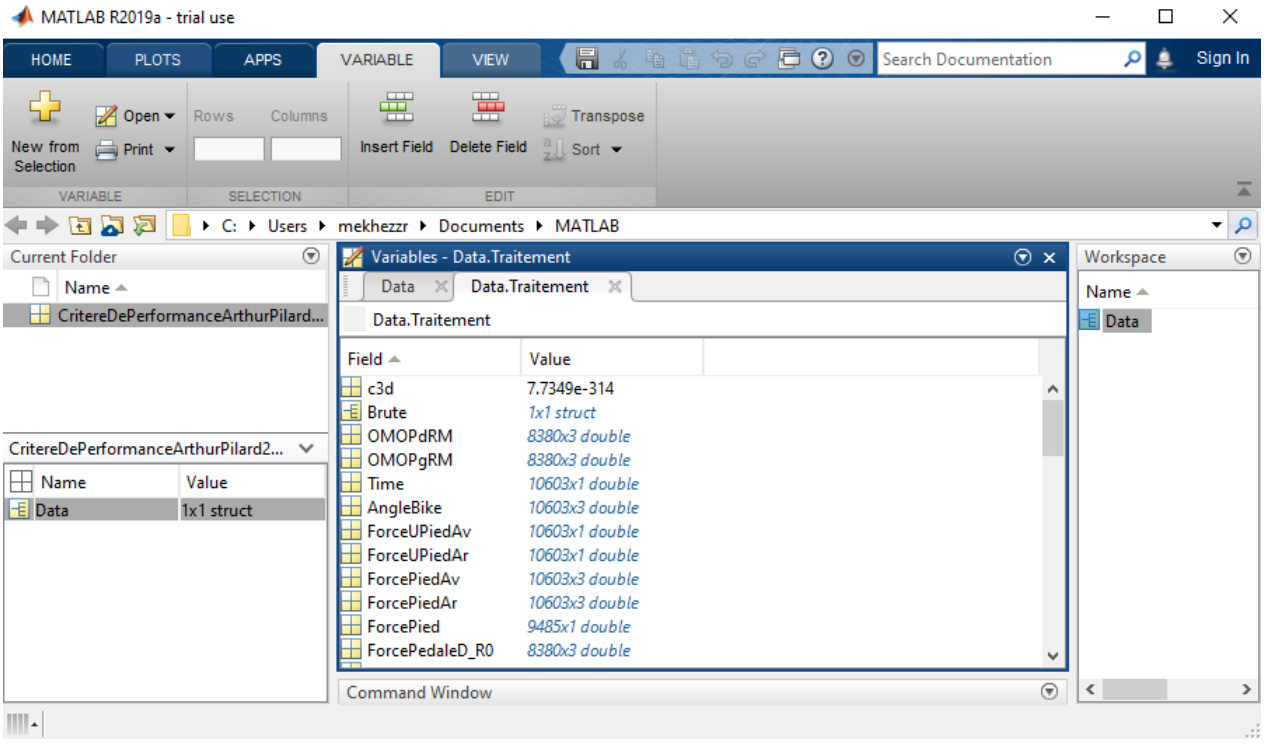


FIGURE A.7 – Contenu de la structure traitement

La figure ci-dessus nous montre en premier lieu que déjà le champ traitement est une structure parce qu'il contient des champs qui sont des matrices et d'autres qui sont eux même des structures, comme ici le champ "Brute" c'est une structure on peut le voir aussi a l'aide de Matlab via la colonne value, Matlab nous fournit aussi la possibilité de le voir directement le format des matrices qui sont imbriqué dans cette structure comme ici le champ "OMOPdRM" est une matrice de 8380 lignes et 3 colonnes.

A.2.2 Aplatir les données

Le but de cette étape est de faire ressortir tous les matrices existant dans ce fichier Matlab en un seul niveau et supprimé les imbrications pour pouvoir créer un fichier CSV sans aucun niveau d'imbrication, pour avoir en colonne le nom du paramètre et en ligne les données de ces paramètres.

Extraire les matrices

Les données fournies sont complexe, surtout avec la présence des structures et les imbrications de ces derniers, donc pour palier a ce problème il fallait développer des scripts en Matlab pour faire cette extraction.

On a utilisé 4 scripts pour traiter tous les cas possibles qu'on a pu trouver, mais si jamais il y 'a un cas qui n'est pas traité on peut modifier dans le script convcsv.m, car toute la logique d'extraction est dedans. La figure suivante A.8 montre les 4 scriptes utilisés.

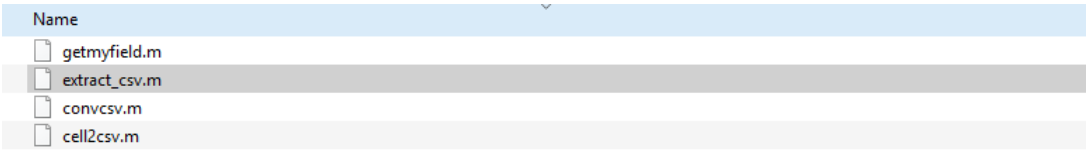


FIGURE A.8 – Les quatre scriptes responsable de l'extraction des matrices

La fonction getmyfield.m :

Cette fonction prend 2 paramètres, le premier est la structure de donnée que ça soit une matrice ou une structure ou même une cellule, le deuxième paramètre c'est le nom du champ quand cherche à trouver.

Cette fonction permet de parcourir les champs d'une structure d'une façon récursive afin trouver le champ désirer et affiche son contenu. Un exemple d'appel pourrait être comme suit :

$getmyfield(Data.Traitement,'Time')$: Le retour sera la matrice Time.
Si le champ rechercher et une structure, donc il faut redescendre dans l'imbrication et cela est implémenté dans le scripte convcsv.m.

La fonction convcsv.m :

Cette fonction prend 2 paramètres, le chemin vers un fichier Matlab pour le charger, et le nom du dossier de sortie où on va mettre les fichiers CSV extraits. La logique de

création des dossiers et fichiers CSV est implémenté ici, donc si on veut changer un comportement il faut changer uniquement dans cette fonction. Un exemple d'appel pourrait être comme suit :

```
convcsv('c:\\..\\CritereDePerformanceRomainRacine1.mat',
'c:\\..\\data_v2\\RomainRacine1')
```

Le retour sera un dossier de nom RomainRacine1 et dedans les fichiers CSV générer à partir des matrices trouver dans le fichier qu'on vient de passer en paramètres. La figure suivante montre le résultat de cet appel A.9 :

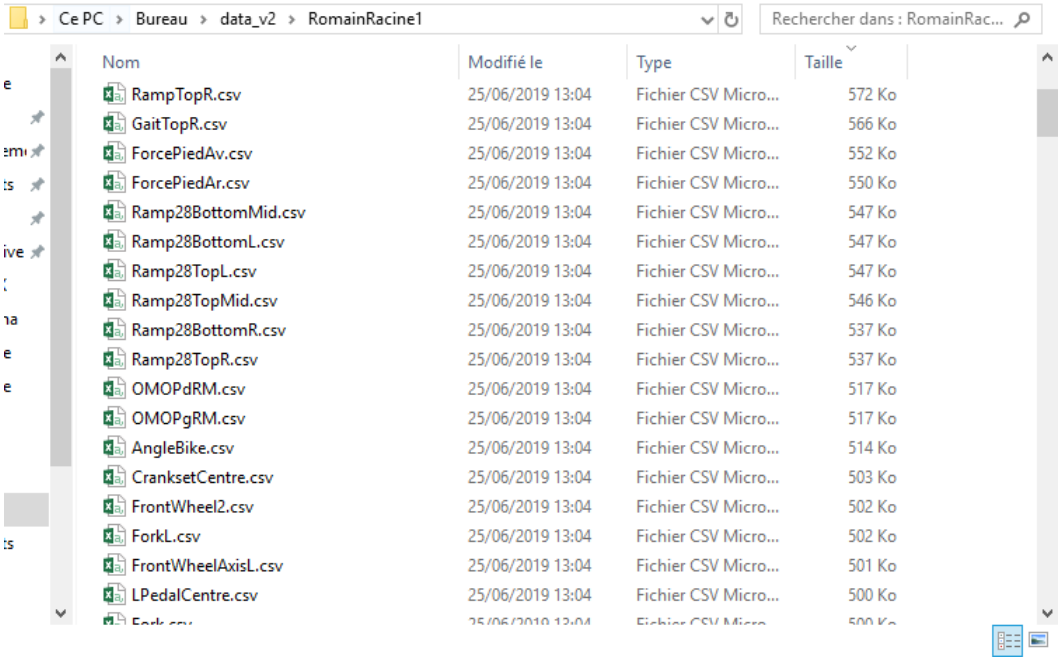


FIGURE A.9 – résultats d'extraction des fichiers CSV.

La fonction `extract_csv.m`

Cette fonction prend en paramètre le chemin vers le dossiers "data". c'est la fonction qu'il faut appeler pour la création automatique de toutes la base de données, c'est la fonction où on fait appel a convcsv et aussi à getmyfield pour extraire les matrices en fichiers CSV.

Un exemple d'appel de cette fonction :

```
extract_csv('c:\\..\\data_v2')
```

Cette appel va générer dans chaque sous dossier de `data_v2` un dossier nommé comme suit : NomPrénomNum ou Num c'est numéro de l'essai, et met dans ces dossiers les fichiers CSV extraits de cette essai. comme le montre la figure A.10

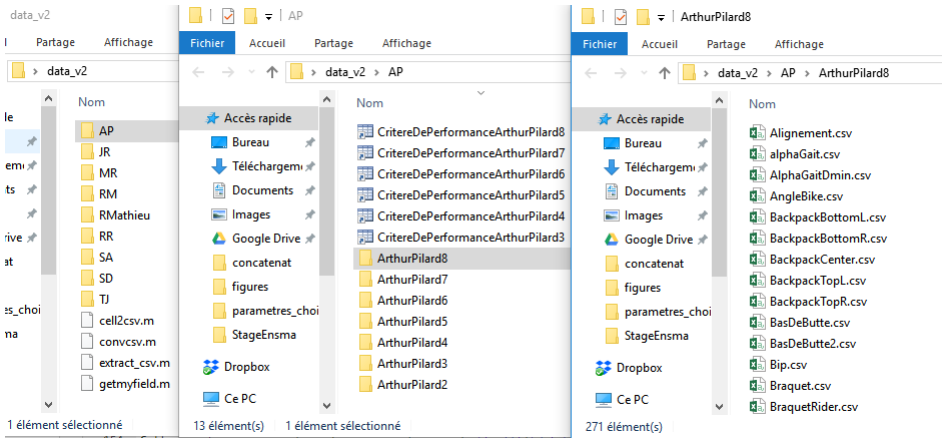


FIGURE A.10 – résultats finale d'extraction des fichiers CSV

Si en remarque bien les noms de ces fichiers CSV porte les mêmes noms des matrices trouver dans les données fournies, donc maintenant on a plus cette imbrication de matrice et la notion de structure, et cela va nous faciliter énormément l'étape de l'analyse de données.

A.2.3 Concaténation de données

Maintenant puisqu'on a les données déjà aplatie et sous le format CSV, le but et de concaténer ces fichiers CSV pour obtenir à la fin deux gros fichiers CSV, un qui regroupe toutes les données temporelles et l'autre toutes les données discrètes.

Pour cette partie on a choisi de travailler avec Python car il n'y a plus de matrices et de structures, et c'est plus facile de manipuler les fichiers CSV avec Python car il dispose de plein de bibliothèques très utile et optimiser comme la bibliothèque numpy, Pandas, etc.

Concaténation des fichiers CSV

Dans cette section nous allons concaténer les fichiers CSV Pour obtenir 3 gros fichiers CSV, le premier et pour les données "Traitement" (données temporelles), le deuxième pour les données "frames" (données discrètes) et le troisième pour les données "travail" on a séparé les données "travail" des données temporelles parce que ces derniers sont calculé que dans la partie du départ, or les données temporelles c'est les données toutes au long de la course et même avant le départ.

La fonction `concat_to_frames_traitement_travail` :

Pour effectuer la concaténation pour ces trois fichiers, on a implémenté une fonction en Python qui va prendre en paramètre 2 chemins, le premier chemin c'est notre fichier "data", et le deuxième c'est le chemin où on veut stocker les données. Un appel de cette fonction pourrait être comme suit :

```
concat_to_frames_traitement_travail('c:\\..\\data_v2', 'c:\\..\\data_Lts')
```

Le résultat de cette appel est montré dans la figure suivante A.11

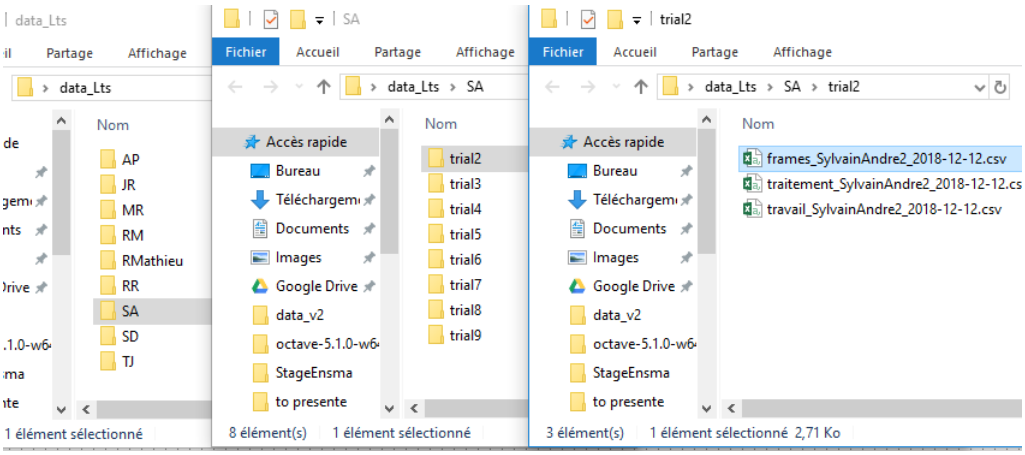


FIGURE A.11 – résultats de concaténation des fichiers CSV

Comme on peut voir dans la figure A.11 la fonction va créer un dossier de nom `data_Lts` et dedans un dossier pour chaque pilote et pour chaque pilote des dossiers nommé trial c'est pour designer les numéros des essais et dans chaque essai, on va créer les trois fichiers CSV travail, frames et traitement.
Le nom de ces fichiers est particulier car on le stocke de la façon suivante : `typeFichier_NomPrenom_numeroEssai_DateEssai.csv`, et cela c'est pour facilité l'utilisation et l'exploitation de ces derniers.

La fonction `data_cut` :

Pour cette deuxième étape nous allons couper le fichier CSV de traitement que dans la partie de départ parce que comme déjà expliquer dans la section A.2.3 les données du fichier traitement commence bien avant le départ et se termine en fin de course et pour notre analyse on va s'occuper que de la partie du départ. Donc cette fonction prend un seul paramètre qui est le chemin du dossier qu'on vient de créer `"data_Lts"`. Un exemple d'appel :

```
data_cut('c:\\..\\data_Lts')
```

Avec ce simple appel on vient de créer un autre fichier CSV qui est la copie de notre gros fichier CSV traitement mais couper que dans la partie de départ. Pour la partie de départ c'est la partie qui commence de BIP et se termine à finDep. Dans la figure ci-dessous A.12 on montre le résultat de cette appel.

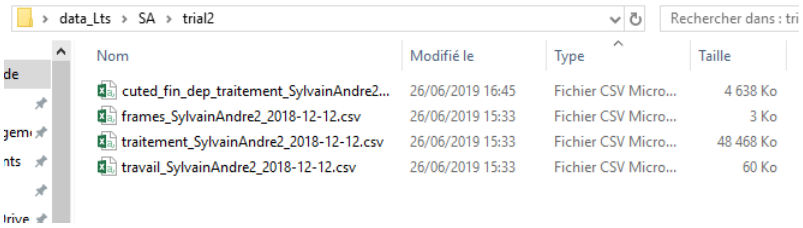


FIGURE A.12 – Coupure du fichier traitement.

Donc on vient de créer un quatrième fichier couper dans la partie de départ, on remarque aussi que la taille du nouveau fichiers est bien plus petite.

La fonction `concat_traitement_travail` :

La troisième étape est de concaténé les deux fichiers travail et le fichier traitement qu'on vient de couper parce que maintenant ils ont le même nombre de données comme le montre la figure ci-dessous A.13

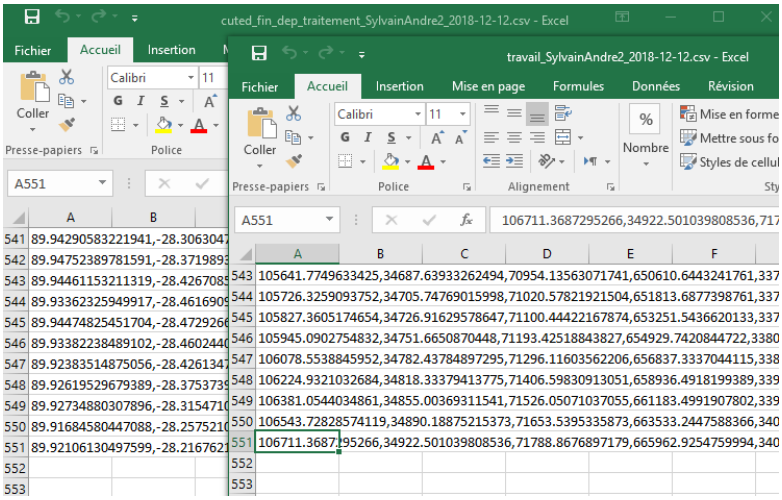


FIGURE A.13 – le nombre d'observation de "Traitement couper" et "Travail".

La fonction `concat_traitement_travail` prend un seul paramètre qui est le chemin du dossier `"data_Lts"` déjà créer dans l'étape A.2.3 Un exemple d'appel :

```
concat_traitement_travail('c:\\..\\data_Lts')
```

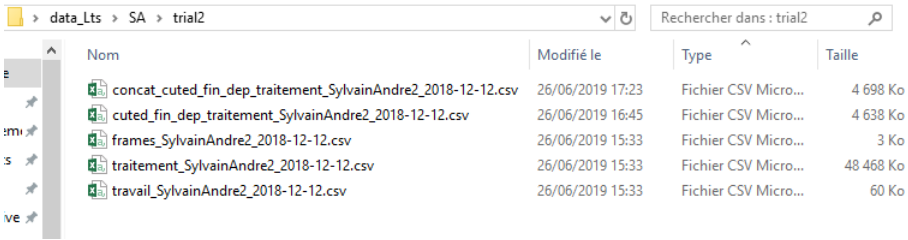


FIGURE A.14 – Résultat de concaténation Traitement - Travail.

Donc on vient de créer un nouveau fichier CSV qui commence par "concat", et du coup maintenant on a tous ce qu'il faut pour commencer l'analyse de données.

Puisque il y a quelques fichiers qui sont en plus et dont on a forcément pas besoin d'utiliser on a fait une fonction pour la copie des fichiers qu'on a besoin dans une nouvelle base.

La fonction `copy_files` :

Cette fonction prend en entrer deux paramètres, le premier c'est le chemin de notre base, et le deuxième c'est le chemin où on veut déplacer notre nouvelle base nettoyer.

Un exemple d'appel de cette fonction pourrait être comme suit :

```
copy_files('c:\\..\\data_Lts', 'c:\\..\\data_cleaned')
```

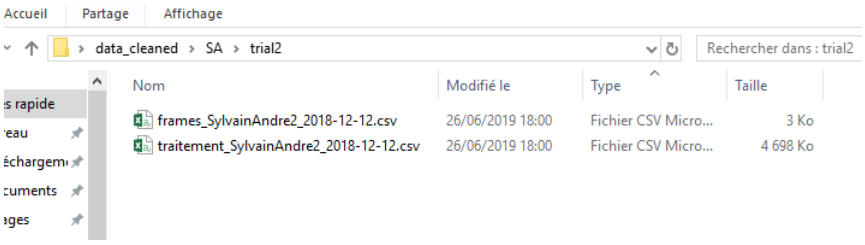


FIGURE A.15 – Résultat du nettoyage des données.

Avec le résultat montré dans la figure A.15, la première phase de prétraitement de données est terminé. Maintenant le data analyst peut directement effectuer ces analyses facilement et même appliquer d'autre prétraitement et préparation de données rapidement.

Résumé

Ce projet de fin d'études s'inscrit dans le cadre des travaux multidisciplinaires menés conjointement par l'équipe IDD du laboratoire LIAS et l'équipe RoBioSS de l'Institut PPRIME et qui visent à améliorer les départs des pilotes de la BMX Race. L'équipe IDD est spécialisée dans le traitement de la donnée tant dis que l'équipe RoBioSS est spécialisée dans le domaine de la biomécanique. Elle propose une solution matérielle (capteurs Pédaliers, caméras, atelier d'essai pour les pilotes) afin de collecter les données liées au départ des pilotes élites de l'équipe de France. L'équipe RoBioSS propose des modèles biomécaniques pour représenter ces départs dans le but de les optimiser par la suite. Toutefois cette solution ne facilite pas l'identification des paramètres responsable de l'amélioration du départ et ce à cause de la complexité des modèles biomécaniques proposés. Afin de simplifier ses modèles et identifier que les paramètres pertinents impactant le départ de la BMX Race, l'institut PPRIME a fait appel au laboratoire LIAS et ce en travaillant sur de nouveaux modèles basés sur l'auto-apprentissage. L'apprentissage automatique est un procédé informatique qui vise à déduire un ensemble de règles à partir d'un jeu de données pour construire de nouvelles connaissances. Ce procédé a été appliqué avec succès dans de différents domaines, comme les systèmes d'analyse des anciennes ventes pour la prédiction du comportement du client et les prévisions météorologiques

Ce projet vise à proposer un modèle simplifié mais prédictif de la performance en se basant sur des techniques d'apprentissage automatique. Le but de notre travail est d'étudier les données de départ fournies par l'équipe RoBioSS, concevoir une solution compatible avec ce type de données et tester les algorithmes sur ces mêmes données.

Mots-clés: apprentissage automatique, science de données, BMX, régression linéaire, apprentissage supervisé.

Abstract

My graduation project is a part of the multidisciplinary work carried out jointly by the IDD team of the LIAS laboratory and the RoBioSS team of the PPRIME Institute, which aim to improve the departure of the BMX Race drivers. The IDD team specializes in data processing whereas the RoBioSS team is specialized in the field of biomechanics. It offers a hardware solution (Cranks sensors, cameras, test workshop for pilots) to collect data related to the departure of the elite drivers of the French team. The RoBioSS team proposes biomechanical models to represent these departures in order to optimize them afterwards. However this solution does not facilitate the identification of the parameters responsible for the improvement of the departure and this is because of the complexity of the proposed biomechanical models. In order to simplify its models and to identify the relevant parameters impacting the departure of the BMX Race, the PPRIME institute called on the LIAS laboratory to work on new models based on a different field which is Machine learning. Machine learning is a computer process that aims to derive a set of rules from a dataset to build new knowledge. This process has been successfully applied in different areas, such as old sales analysis systems for predicting customer behavior and weather forecasts.

This project aims to propose a simplified model but predictive of performance based on machine learning techniques. The purpose of my work is to study the initial data provided by the RoBioSS team, to design a solution compatible with this type of data and to test the algorithms on these same data.

Keywords: Machine Learning, data science, BMX, Linear regression, supervised learning.

ملخص

مشروع التخرج الخاص بي هو جزء من العمل متعدد التخصصات الذي ينفذ بشكل مشترك من قبل فريق IDD في مختبر LIAS وفريق RoBioSS التابع لمعهد PPRIME ، والذي يهدف إلى تحسين رحيل سائقي BMX Race. فريق IDD متخصص في معالجة البيانات في حين أن فريق RoBioSS متخصص في مجال الميكانيك الحيوية. إنه يوفر حلاً ماديا عن طريق توفير أجهزة متخصصة (أجهزة استشعار السواعد والكاميرات وورشة اختبارالسائقين) لجمع البيانات المتعلقة برحيل سائقي النخبة في الفريق الفرنسي. يقترح فريق RoBioSS نماذج ميكانيكية حيوية لتمثيل هذه الانطلاقة من أجل تحسينها بعد ذلك. ومع ذلك ، فإن هذا الحل لا يسهل تحديد المعلمات المسؤولة عن تحسين المغادرة وهذا بسبب تعقيد النماذج الميكانيكية الحيوية المقترحة. من أجل تبسيط النماذج وتحديد المعلمات التي تؤثر على رحيل سباق BMX ، دعا معهد PPRIME مختبر LIAS للعمل على نماذج جديدة تعتمد على مجال مختلف وهو تعلم الآلة. تعلم الآلة عبارة عن عملية كمبيوتر تهدف إلى اشتقاق مجموعة من القواعد من مجموعة البيانات لبناء معرفة جديدة. تم تطبيق هذه العملية بنجاح في مجالات مختلفة ، مثل أنظمة تحليل المبيعات القديمة للتنبؤ بسلوك العملاء وتوقعات الطقس.

يهدف هذا المشروع إلى اقتراح نموذج مبسط ولكن تنبؤي بالأداء على أساس تقنيات التعلم الآلي. الغرض من عملي هو دراسة البيانات الأولية المقدمة من فريق RoBioSS ، وتصميم حل متوافق مع هذا النوع من البيانات واختبار الخوارزميات على هذه البيانات نفسها.

الكلمات المفتاحية:تعلم الآلة ، علم البيانات ، BMX ، الانحدار الخطي ، التعلم الخاضع للإشراف.