

# AI 4 INDUSTRY – CATIE

*Valentin FEVRE - Rémi CAZELLES - Yasemin KOCAK*

## *Introduction*

Dans le cadre de l'AI 4 Industry, nous avons eu l'opportunité de participer au Use Case de CATIE, centre technologique spécialisé dans la recherche, le développement et l'innovation dans les domaines de l'information et de l'électronique.

L'objectif de ce Use Case était de développer un dispositif permettant de reconnaître automatiquement les chiffres tracés à l'aide d'un capteur Z motion<sup>1</sup>.



*Figure 1 : Capteur Z Motion 6TRON*

Afin de pouvoir avoir un jeu de données à analyser, chacun des trois groupes a saisi, à l'aide des capteurs, plusieurs séries de nombres allant de 0 à 9, sur tables pour certains et dans l'espace (à l'horizontal mais aussi aléatoirement dans d'autres sens) pour d'autres. Ces données ont été collectées via la console Linux, en initiant un enregistrement via un script python fourni par CATIE.

## *Indications sur les données*

Les données collectées qui en sont ressorties sont les suivantes :

### **1. Pour la Configuration 1 :**

- « t » qui représente le timestamp (il n'est pas toujours régulier)

---

<sup>1</sup> Pour plus d'informations sur la configuration du Capteur Z Motion 6TRON : [https://6tron.io/z\\_motion/](https://6tron.io/z_motion/)

- « raw acceleration x », « raw acceleration y » et « raw acceleration z » qui représentent les accélérations linéaires du capteur mesurées selon les axes x, y et z.
- « magnetic field x », « magnetic field y » et « magnetic field z » qui correspondent aux champs magnétiques mesurés selon les axes x, y et z.

## 2. Pour la Configuration 3 :

- « t » qui représente le timestamp (il n'est pas toujours régulier)
- « raw acceleration x », « raw acceleration y » et « raw acceleration z » qui représentent les accélérations linéaires du capteur mesurées selon les axes x, y et z.
- « quaternion w », « quaternion x », « quaternion y » et « quaternion z » qui eux vont représenter les composantes d'un quaternion, une représentation mathématique d'une orientation dans l'espace. En d'autres termes, cela va décrire l'orientation 3D du capteur.

	t	raw_acceleration_x	raw_acceleration_y	raw_acceleration_z	quaternion_w	quaternion_x	quaternion_y	quaternion_z
0	20.206264	-0.37	0.22	-9.86	0.018188	0.931946	0.361816	-0.013367
1	20.229434	-0.37	0.23	-9.86	0.018188	0.931946	0.361816	-0.013367
2	20.251220	-0.36	0.18	-9.84	0.018127	0.931946	0.361816	-0.013428
3	20.271621	-0.36	0.17	-10.24	0.017944	0.931946	0.361816	-0.013489
4	20.294289	-0.41	0.17	-9.75	0.017883	0.931946	0.361816	-0.013489

*Figure 2 : Extrait du dataframe des données collectées en configuration 3 (chiffre 0)*

## *Visualisation des données et tests de différentes méthodes*

Afin de pouvoir proposer un modèle pertinent et en adéquation avec les attentes du client, à savoir, permettre au modèle de prédire à minima, 80% des échantillons donnés, nous nous sommes dans un premier temps penché sur diverses méthodes :

- Statistiques partielles des time series (méthode pandas « describe() ») : **La méthode qui a été conservée**
- Statistiques complètes des time series
- Segmentation des time series en quartiers et statistiques complètes
- Statistiques complètes et nombre de changement de signes
- Statistiques complètes et changement

## *Modèles*

Chacune de ces méthodes a été testée en utilisant 4 algorithmes de classification différents :

- **Une régression logistique**, utilisé pour prédire la probabilité qu'une observation appartienne à une classe particulière en se basant sur variables indépendantes.
- **Un arbre de décision**, modèle qui divise l'espace des caractéristiques en segments homogènes en fonction des valeurs de ces caractéristiques, formant ainsi une structure arborescente de décisions.
- **Un Support Vector Classifier**, algorithme de classification qui cherche à trouver l'hyperplan optimal qui sépare les classes dans un espace de dimensions supérieures.
- **Un Random Forest Classifier**, modèle basé sur l'ensemble d'arbres de décision, où plusieurs arbres de décision sont entraînés sur différents sous-ensembles des données d'entraînement.

## Résultats

Les résultats obtenus sont la moyenne des accuracy obtenue après cross validation en utilisant 10 splits. Nous utilisons la méthode *StratifiedShuffleSplit*, qui permet de préserver le pourcentage d'échantillons pour chaque classe. Ci-après les résultats obtenus :

```

Dataset Vertical
*****

"mean" et "sdt" du describe
Generalization score without hyperparameters tuning: 0.655 ± 0.045
Generalization score without hyperparameters tuning: 0.786 ± 0.031
Generalization score without hyperparameters tuning: 0.267 ± 0.051
Generalization score without hyperparameters tuning: 0.876 ± 0.029
*****

"mean" et "sdt" du describe + feature engineering "nb sign"
Generalization score without hyperparameters tuning: 0.656 ± 0.046
Generalization score without hyperparameters tuning: 0.766 ± 0.034
Generalization score without hyperparameters tuning: 0.253 ± 0.043
Generalization score without hyperparameters tuning: 0.885 ± 0.034
*****

"mean" et "sdt" du describe + segments temps
Generalization score without hyperparameters tuning: 0.656 ± 0.040
Generalization score without hyperparameters tuning: 0.766 ± 0.034
Generalization score without hyperparameters tuning: 0.253 ± 0.043
Generalization score without hyperparameters tuning: 0.895 ± 0.025
*****

```

Toutes les statistiques du describe

Generalization score without hyperparameters tuning:  $0.687 \pm 0.039$

Generalization score without hyperparameters tuning:  $0.863 \pm 0.021$

Generalization score without hyperparameters tuning:  $0.327 \pm 0.058$

Generalization score without hyperparameters tuning:  $0.932 \pm 0.028$

La matrice de confusion, qui compare les prédictions du modèle avec les valeurs réelles données, nous donne le résultat suivant :

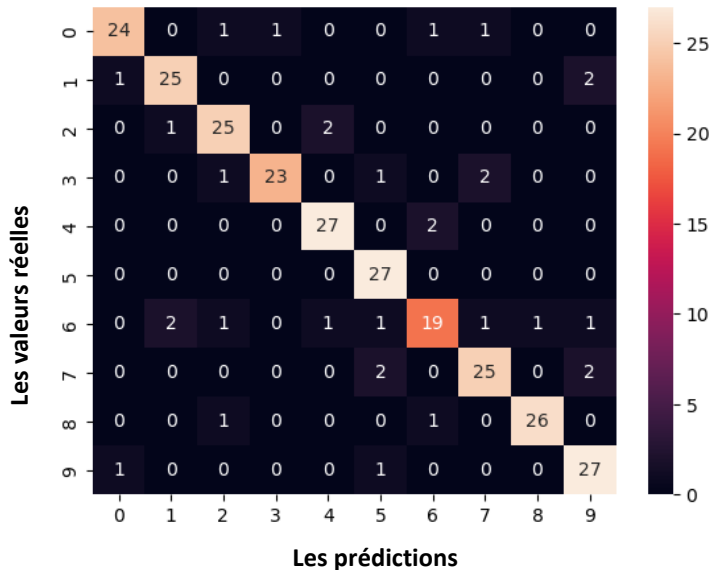


Figure 3 : Matrice de confusion

Ici par exemple, les nombres les plus faciles à prédire sont les 4, 5, 8 et 9, tandis que le 6 est le chiffre sur lequel le modèle effectue le plus d'erreurs. Le modèle qui prédit le mieux est le random forest avec une prédiction à **93%**.

## Conclusion

Ce rapport détaille l'étude visant à développer un système de reconnaissance automatique des chiffres tracés à l'aide d'un capteur Z motion, réalisée dans le cadre du Use Case de CATIE. Plusieurs méthodes d'analyse des séries temporelles ont été testées avec différents algorithmes de classification. **Les résultats montrent que le modèle Random Forest obtient la meilleure précision de prédiction, atteignant 93%, bien que certaines améliorations puissent être apportées pour augmenter la précision pour certains chiffres. La prochaine étape pourrait consister à affiner le modèle pour améliorer la prédiction des chiffres plus difficiles, comme le 6 mais également de prédire ces derniers dessinés dans l'espace et non plus en 2D.**