UNIVERSITÉ ABDELMALEK ESSAÂDI
Faculté des Sciences et Techniques - Tanger
Département d'Informatique

MASTER SCIENCES ET TECHNIQUES
INTELLIGENCE ARTIFICIELLE ET SCIENCES DES DONNÉES

RAPPORT DE PROJET

Détection de Changement de Distribution (Data Drift)

Post-Déploiement d'un Modèle de Machine Learning

Cadre du projet

Module : Mathématiques pour les Sciences des Données

Réalisé par :
Ezzoubair ZARQI
Mahmoud EL GHARIB
Abderrahmane BELKASMI

Encadré par :
Pr. Tarik AMTOUT
Pr. Abdelaziz ASSADOUQ

Année Universitaire : 2025 – 2026

Résumé

Dans ce projet, nous étudions le problème de la détection de changement de distribution (*drift*) dans un contexte de modèles de machine learning déployés en production.

Nous simulons un changement progressif des données d'entrée, appliquons des tests statistiques non paramétriques (tels que le test de Kolmogorov-Smirnov pour détecter ce drift, puis analysons son impact sur la performance du modèle à l'aide d'intervalles de confiance.

Mots-clés : *Data Drift, Machine Learning, Kolmogorov-Smirnov, Surveillance de Modèle.*

Abstract

In this project, we address the problem of distribution shift detection (drift) in the context of machine learning models deployed in production.

We simulate a progressive shift in input data, apply non-parametric statistical tests (such as the Kolmogorov-Smirnov test to detect this drift, and analyze its impact on model performance using confidence intervals.

Keywords : *Data Drift, Machine Learning, Kolmogorov-Smirnov, Model Monitoring.*

Table des matières

Chapitre 1

Introduction

Les modèles de machine learning sont généralement entraînés sur des données historiques en supposant que les caractéristiques statistiques de ces données resteront stables au cours du temps. Cependant, dans des environnements réels et dynamiques, cette hypothèse est souvent mise en défaut. Les comportements des utilisateurs, les conditions économiques ou les politiques commerciales peuvent évoluer, entraînant ainsi des modifications dans la distribution des données d'entrée. Ce phénomène est communément désigné sous le terme de **changement de distribution**, ou **data drift**.

Le data drift constitue un enjeu majeur en data science, car il peut conduire à une dégradation progressive, voire brutale, des performances d'un modèle prédictif déployé en production. Un modèle initialement performant peut alors produire des prédictions erronées sans que cette détérioration ne soit immédiatement perceptible. Il devient donc essentiel de disposer de **méthodes mathématiques et statistiques permettant de détecter ces changements post-déploiement**, afin de garantir la fiabilité et la robustesse des systèmes décisionnels.

Dans ce contexte, ce projet s'inscrit dans une démarche de **surveillance des modèles de machine learning**, appliquée à un problème de maintenance prédictive industrielle. L'objectif principal est d'analyser l'impact de différentes formes de data drift sur les distributions des variables capteurs et sur les performances d'un modèle de détection de panne. Pour ce faire, un modèle de référence est d'abord entraîné sur des données de capteurs considérées comme stables, puis un scénario réaliste de dérive des données (usure, surchauffe) est simulé.

L'approche adoptée repose sur l'utilisation de **tests statistiques issus des mathématiques appliquées**, tels que le test de Kolmogorov-Smirnov et le test du Chi-deux, ainsi que sur des métriques de stabilité comme le Population Stability Index (PSI). L'impact du drift est ensuite évalué à travers l'analyse des performances du modèle, notamment en termes d'accuracy, complétée par des méthodes de validation par rééchantillonnage.

À travers ce travail, l'objectif est de mettre en évidence l'importance du **monitoring post-déploiement** des modèles de machine learning et de montrer comment les outils mathématiques permettent d'anticiper et de quantifier les risques liés à l'évolution des données dans un cadre opérationnel.

1.1 Contexte Théorique

Cette section présente les notions théoriques fondamentales nécessaires à la compréhension du phénomène de changement de distribution des données, communément appelé *data drift*. Elle introduit les différents types de drift rencontrés en data science ainsi que leur lien avec les hypothèses statistiques sous-jacentes aux modèles de machine learning.

1.1.1 Notion de Data Drift

Dans le cadre de la modélisation statistique et du machine learning, un modèle est généralement entraîné sur un ensemble de données historiques, supposées représentatives des données futures. Cette hypothèse implique que la distribution jointe des variables explicatives X et de la variable cible Y reste inchangée au cours du temps. Formellement, on suppose que :

$$P_{\text{train}}(X, Y) = P_{\text{production}}(X, Y)$$

Cependant, dans un environnement réel, cette égalité est rarement vérifiée. Des facteurs externes tels que l'évolution du comportement des utilisateurs, des changements économiques ou des modifications des politiques commerciales peuvent entraîner une variation des distributions statistiques. Ce phénomène est désigné sous le terme de **data drift**.

Le data drift correspond donc à une situation dans laquelle la distribution des données observées en phase de production diffère de celle utilisée lors de l'entraînement du modèle. Cette divergence peut affecter la capacité du modèle à généraliser correctement, conduisant ainsi à une dégradation de ses performances prédictives.

1.1.2 Types de Changement de Distribution

En pratique, le data drift peut prendre plusieurs formes, selon la nature des distributions affectées. On distingue principalement trois types de changement de distribution.

Covariate Drift

Le *covariate drift* correspond à une modification de la distribution des variables explicatives X, tandis que la relation conditionnelle entre X et la variable cible Y reste inchangée. Formellement, ce cas peut être décrit par :

$$P_{\text{train}}(X) \neq P_{\text{production}}(X) \quad \text{et} \quad P(Y \mid X) \text{ constant}$$

Ce type de drift est fréquent dans les applications industrielles. Dans ce projet, les modifications simulées sur la variable `temperature` illustrent ce type de changement.

Prior Probability Drift

Le *prior probability drift*, également appelé *label drift*, se produit lorsque la distribution de la variable cible Y change, indépendamment des variables explicatives. On a alors :

$$P_{\text{train}}(Y) \neq P_{\text{production}}(Y)$$

Ce phénomène peut survenir, par exemple, lorsqu'un vieillissement entraîne une variation globale du taux de panne, sans modification significative des conditions opérationnelles instantanées.

Concept Drift

Le *concept drift* représente le cas le plus complexe, dans lequel la relation entre les variables explicatives et la variable cible évolue au cours du temps. Mathématiquement, cela se traduit par une variation de la distribution conditionnelle :

$$P_{\text{train}}(Y \mid X) \neq P_{\text{production}}(Y \mid X)$$

Dans ce cas, même si la distribution des variables explicatives reste stable, le modèle devient obsolète car le concept sous-jacent à prédire a changé. Ce type de drift nécessite généralement une mise à jour ou un réentraînement du modèle.

1.1.3 Lien avec les Hypothèses Statistiques

Les méthodes de détection du data drift reposent essentiellement sur des tests d'hypothèses statistiques. Ces tests visent à comparer les distributions observées en phase d'entraînement et en phase de production, en formulant l'hypothèse nulle suivante :

$$H_0 : P_{\text{train}}(X) = P_{\text{production}}(X)$$

Le rejet de cette hypothèse, à l'aide de tests tels que le test de Kolmogorov-Smirnov pour les variables numériques ou le test du Chi-deux pour les variables catégorielles, constitue un indicateur statistique de la présence d'un drift.

Ainsi, l'analyse du data drift s'inscrit pleinement dans le cadre des mathématiques appliquées et des statistiques inférentielles, en fournissant des outils rigoureux pour évaluer la stabilité des données et la fiabilité des modèles de machine learning déployés en production.

1.2 Génération des Données et Modèle Baseline

Dans cette section, nous décrivons la génération des données synthétiques utilisées dans le cadre du projet, ainsi que la construction du modèle de machine learning de référence. L'utilisation de données simulées permet de contrôler précisément les distributions statistiques et de reproduire des scénarios réalistes de changement de distribution.

1.2.1 Génération des Données d'Entraînement (Phase A)

Les données de la phase d'entraînement, considérées comme représentatives d'un environnement stable, sont générées artificiellement à l'aide de la bibliothèque `NumPy`. Cette approche permet de définir explicitement les lois de probabilité sous-jacentes aux variables explicatives.

Soit $X = (X_1, X_2, \ldots, X_p)$ un vecteur de variables d'entrée. Dans ce projet, les variables numériques sont générées à partir de distributions continues, par exemple des lois normales :

$$X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2), \quad X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$$

La variable cible Y est ensuite générée en fonction des variables explicatives à l'aide d'une fonction de décision, simulant un comportement réaliste de classification binaire. Cette approche permet de disposer d'un jeu de données contrôlé, tout en conservant une structure proche d'un problème réel de machine learning.

Les données générées constituent la **Phase A**, correspondant aux données historiques utilisées pour l'entraînement du modèle.

1.2.2 Entraînement du Modèle de Référence

Un modèle de classification est entraîné sur les données de la Phase A afin de simuler un modèle déjà déployé en production. Dans ce projet, un algorithme de machine learning issu de la bibliothèque `scikit-learn` est utilisé.

Les données sont divisées en ensembles d'entraînement et de validation afin d'évaluer les performances initiales du modèle. Le score obtenu sur ces données constitue la performance de référence, notée S_A, servant de point de comparaison pour l'analyse post-déploiement.

1.2.3 Génération des Données de Production (Phase B)

Afin de simuler un changement de distribution post-déploiement, un nouveau jeu de données est généré à l'aide de `NumPy`. Contrairement à la Phase A, la distribution d'une variable clé (la température) est volontairement modifiée. Par exemple, la variable `temperature` initialement distribuée selon :

$$X_{\text{temp}}^{(A)} \sim \mathcal{N}(75, \sigma^2)$$

est remplacée en Phase B par :

$$X_{\text{temp}}^{(B)} \sim \mathcal{N}(82, \sigma^2)$$

où le déplacement de la moyenne ($+7$) simule une surchauffe progressive ou un dérèglement du capteur. Cette modification correspond à un *covariate drift* sévère. Les données générées forment la **Phase B**, représentant les données observées en production après le déploiement du modèle.

1.2.4 Objectif de la Comparaison

L'objectif principal est de comparer les distributions des variables entre les Phases A et B à l'aide de tests statistiques non paramétriques, puis d'évaluer l'impact de ce changement de distribution sur la performance du modèle de machine learning. Cette analyse permet de déterminer si le modèle initialement entraîné demeure valide face à l'évolution des données.

Chapitre 2

Détection Statistique du Changement de Distribution

Ce chapitre est consacré à la détection statistique du changement de distribution entre les données d'entraînement (Phase A) et les données de production (Phase B). L'objectif est de déterminer, à l'aide de tests statistiques rigoureux, si les différences observées entre les deux phases sont statistiquement significatives et peuvent être interprétées comme un phénomène de data drift.

2.1 Motivation des Tests Non-Paramétriques

Dans un contexte de surveillance post-déploiement, les distributions des données observées en production sont généralement inconnues et peuvent ne pas suivre des lois paramétriques classiques. Il est donc nécessaire d'utiliser des tests statistiques ne reposant pas sur des hypothèses fortes concernant la forme des distributions.

Les tests non-paramétriques sont particulièrement adaptés à ce contexte, car ils permettent de comparer des distributions empiriques sans supposer de modèle probabiliste spécifique. Dans ce projet, le test de Kolmogorov-Smirnov est utilisé pour comparer les distributions des variables numériques entre la Phase A et la Phase B, indépendamment de leur loi sous-jacente.

2.2 Test de Kolmogorov-Smirnov

Le test de Kolmogorov-Smirnov (KS) est un test non-paramétrique permettant de comparer deux distributions continues à partir de leurs fonctions de répartition empiriques.

Soient $F_A(x)$ et $F_B(x)$ les fonctions de répartition empiriques associées respectivement aux données de la Phase A et de la Phase B. La statistique du test KS est définie par :

$$D = \sup_x |F_A(x) - F_B(x)|$$

L'hypothèse nulle du test est formulée comme suit :

$$H_0 : F_A(x) = F_B(x)$$

contre l'hypothèse alternative :

$$H_1 : F_A(x) \neq F_B(x)$$

Une p-value inférieure au seuil de significativité $\alpha = 0.05$ conduit au rejet de l'hypothèse nulle, indiquant un changement statistiquement significatif de distribution.

2.3 Application du Test KS aux Données Simulées

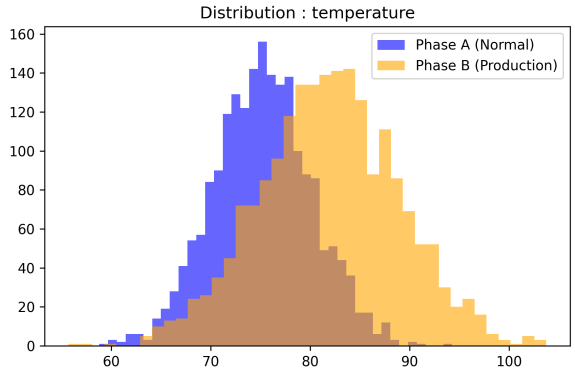
Le test de Kolmogorov-Smirnov a été appliqué aux variables numériques générées synthétiquement afin de comparer les distributions observées en Phase A et en Phase B. Chaque variable a été analysée indépendamment afin d'identifier la présence éventuelle d'un *covariate drift*.

Les résultats obtenus montrent que certaines variables présentent une valeur élevée de la statistique KS associée à une p-value très inférieure au seuil de significativité de 5%. Ces résultats conduisent au rejet de l'hypothèse nulle d'égalité des distributions pour ces variables, indiquant un changement de distribution statistiquement significatif entre les données historiques et les données de production.

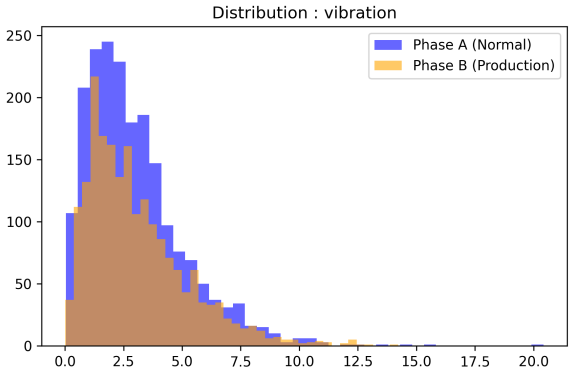
À l'inverse, d'autres variables présentent une p-value élevée, suggérant l'absence de différence statistiquement significative entre les deux phases. Cela indique que toutes les variables ne sont pas affectées par le drift, ce qui correspond à un scénario réaliste en environnement de production.

2.4 Visualisation des Distributions Avant et Après Drift

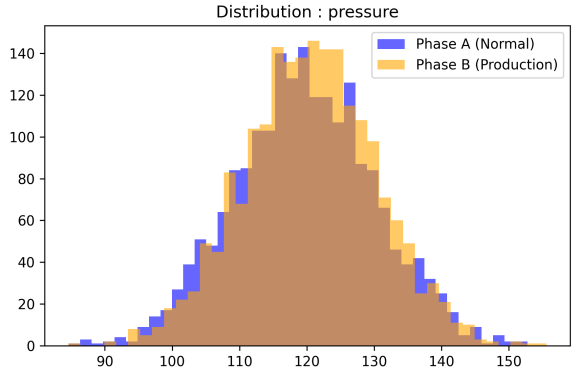
Afin de compléter l'analyse statistique, une visualisation des distributions des variables numériques a été réalisée pour les données de la Phase A et de la Phase B. Ces graphiques permettent d'observer visuellement les différences de distribution mises en évidence par le test de Kolmogorov-Smirnov.



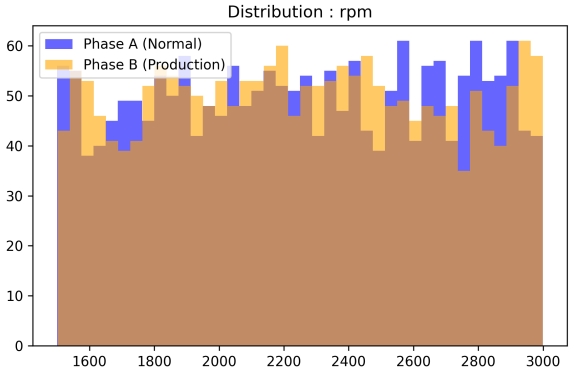
(a) Distribution de `Temperature`



(b) Distribution de `Vibration`



(c) Distribution de `Pressure`



(d) Distribution de `RPM`

FIGURE 2.1 – Comparaison des distributions des variables capteurs entre la Phase A (saine) et la Phase B (défaut/drift).

2.5 Interprétation des Résultats

Les résultats du test de Kolmogorov-Smirnov, corroborés par l'analyse visuelle, conduisent aux conclusions suivantes pour les capteurs surveillés :

— **Température (`temperature`)** : Un drift significatif est détecté (p-value $\ll 0.05$). La statistique KS élevée et la visualisation (Figure 2.1a) confirment un décalage net de la moyenne (de 75°C à 82°C), simulant une surchauffe ou un décalibrage du capteur.
— **Vibration, Pression, RPM** : Aucun changement de distribution significatif n'est observé pour ces variables (p-value > 0.05). Les courbes de densité (Figures 2.1b, 2.1c, 2.1d) se superposent parfaitement entre la Phase A et la Phase B.

Cette stabilité des autres paramètres renforce la crédibilité du diagnostic : le problème n'est pas systémique mais localisé sur la température. Cependant, comme la température est une variable prédictive critique, ce drift unique suffit à compromettre la fiabilité du modèle de détection de panne.

La détection statistique du data drift constitue une étape essentielle du monitoring post-déploiement. Toutefois, la mise en évidence d'un changement de distribution ne permet pas, à

elle seule, de conclure sur la validité du modèle en production. Il est donc nécessaire d'analyser l'impact de ce drift sur les performances du modèle de machine learning, ce qui fait l'objet du chapitre suivant.

Chapitre 3

Impact du Changement de Distribution sur la Performance du Modèle

Après avoir mis en évidence la présence d'un changement de distribution entre les données d'entraînement (Phase A) et les données de production (Phase B), ce chapitre vise à analyser l'impact de ce data drift sur les performances du modèle de machine learning. L'objectif est de déterminer si le changement détecté affecte de manière significative la capacité prédictive du modèle déployé.

3.1 Entraînement du Modèle de Référence

Un modèle de classification binaire a été entraîné sur les données de la Phase A afin de simuler un modèle déjà déployé en production. L'algorithme choisi est une régression logistique, implémentée à l'aide de la bibliothèque `scikit-learn`. Ce choix est motivé par sa simplicité, son interprétabilité et son adéquation avec un cadre mathématique rigoureux.

Avant l'entraînement, les variables numériques ont été standardisées afin d'assurer une mise à l'échelle homogène des données. Le jeu de données de la Phase A a ensuite été divisé en ensembles d'entraînement et de test, permettant d'évaluer les performances initiales du modèle. Les scores obtenus sur ces données constituent la performance de référence du modèle, notée S_A.

3.2 Évaluation du Modèle sur les Données de Production

Le modèle entraîné sur la Phase A a été appliqué tel quel aux données de la Phase B, sans réentraînement, afin de reproduire un scénario réaliste de déploiement. Les performances obtenues sur ces données de production sont notées S_B.

La comparaison entre S_A et S_B permet d'observer une variation des performances du modèle après le changement de distribution. Toutefois, une simple comparaison ponctuelle des scores ne permet pas de conclure sur la significativité statistique de cette variation. Il est donc nécessaire d'adopter une approche d'inférence statistique.

3.3 Bootstrapping des Performances

Afin de quantifier l'incertitude associée aux scores de performance, une méthode de rééchantillonnage par *bootstrapping* a été utilisée. Cette approche consiste à générer un grand nombre d'échantillons bootstrap à partir des données originales, puis à calculer le score de performance pour chacun de ces échantillons.

Pour chaque phase, une distribution empirique du score d'accuracy a ainsi été obtenue. Cette distribution permet d'estimer des intervalles de confiance robustes, sans hypothèse paramétrique sur la distribution du score.

3.4 Intervalles de Confiance et Analyse Statistique

À partir des distributions bootstrap, des intervalles de confiance à 95% ont été calculés pour les performances du modèle sur les données de la Phase A et de la Phase B. Ces intervalles fournissent une estimation de la variabilité du score de performance et permettent une comparaison statistique entre les deux phases.

La comparaison des intervalles de confiance permet de déterminer si la différence observée entre les performances est statistiquement significative. L'absence de chevauchement entre les intervalles indique une dégradation significative des performances du modèle, tandis qu'un chevauchement suggère que la différence observée peut être due à la variabilité statistique.

3.5 Visualisation des Résultats

La Figure 3.1 présente les distributions des scores d'accuracy obtenues par bootstrapping pour les données de la Phase A et de la Phase B. Les lignes verticales indiquent les bornes des intervalles de confiance à 95% pour chaque distribution. De plus, la Figure 3.2 offre une vue comparative globale via des boxplots.

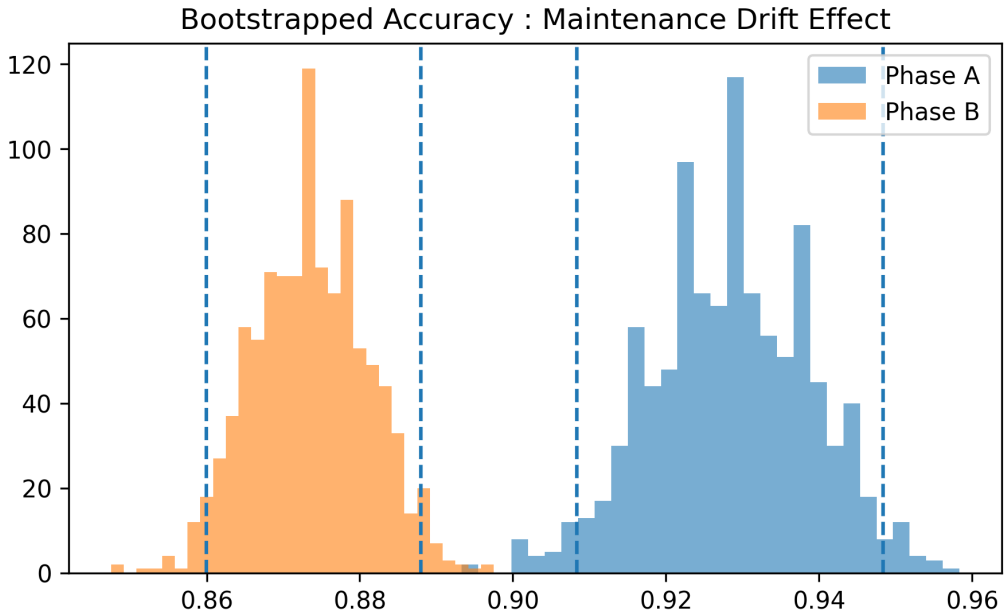


FIGURE 3.1 – Distributions bootstrap des scores d'accuracy pour les Phases A et B, avec intervalles de confiance à 95%.

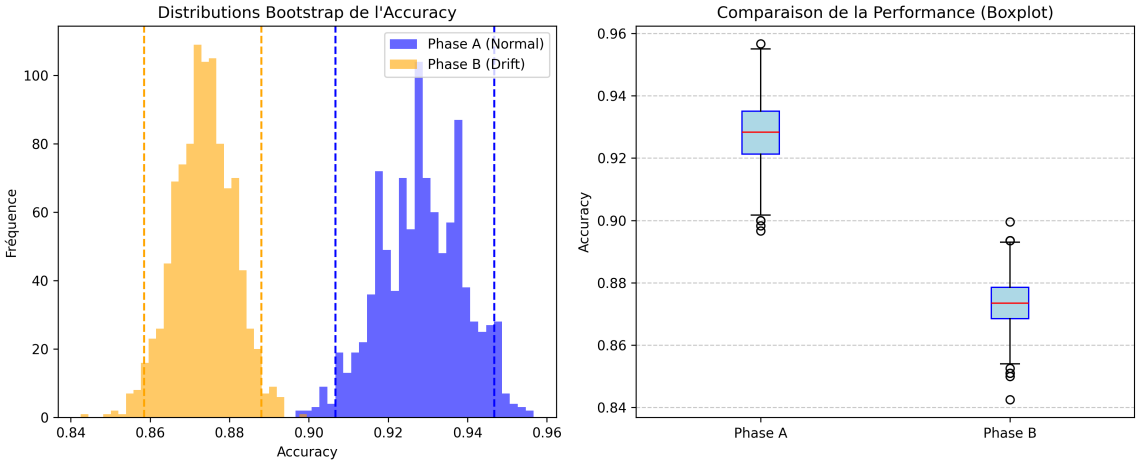


FIGURE 3.2 – Comparaison détaillée de la performance (Accuracy) entre la Phase A et la Phase B (Histogrammes et Boxplots).

3.6 Interprétation des Résultats

L'analyse des distributions bootstrap et des intervalles de confiance met en évidence une baisse significative de la performance du modèle sur les données de production (Phase B) par rapport aux données historiques (Phase A). Comme illustré dans la Figure 3.1, les deux distributions sont clairement séparées. L'intervalle de confiance de la Phase A est centré autour de valeurs élevées (indiquant un modèle performant sur les données saines), tandis que celui de la Phase B est décalé vers des valeurs inférieures.

L'absence de chevauchement entre les intervalles (ou leur très faible recouvrement) confirme

que cette dégradation n'est pas due au hasard, mais bien à l'impact du *covariate drift* (surchauffe détectée sur la température). Le modèle, entraîné sur des températures plus basses, peine à généraliser correctement sur le régime de fonctionnement "surchauffe", ce qui entraîne une augmentation des erreurs de classification (faux positifs ou faux négatifs de panne).

Ce résultat justifie la mise en place d'une alerte de monitoring et suggère la nécessité d'un réentraînement du modèle avec ces nouvelles données opérationnelles pour l'adapter aux conditions actuelles (Phase B).

3.7 Conclusion du Chapitre

Ce chapitre démontre que la détection d'un changement de distribution doit être complétée par une analyse de son impact sur les performances du modèle. L'utilisation du bootstrapping et des intervalles de confiance permet d'apporter une conclusion statistiquement fondée sur la validité du modèle en production. Cette approche constitue une étape essentielle dans une stratégie de monitoring post-déploiement des systèmes de machine learning.

Chapitre 4

Conclusion Générale et Perspectives

Ce projet a permis d'explorer la problématique cruciale du *data drift* (changement de distribution) dans le cycle de vie d'un modèle de machine learning, en l'illustrant par un cas d'usage concret de maintenance prédictive industrielle.

4.1 Synthèse des Travaux

Notre démarche s'est articulée autour de trois axes principaux : la simulation de données réalistes, la détection statistique du drift et l'évaluation de son impact sur la performance prédictive.

Premièrement, nous avons généré un jeu de données synthétique simulant des relevés de capteurs industriels (température, vibration, pression, RPM). Nous avons reproduit un scénario de dérive spécifique : une augmentation progressive de la température moyenne, simulant une usure ou un décalibrage de capteur, tandis que les autres paramètres restaient stables.

Deuxièmement, nous avons appliqué le test de Kolmogorov-Smirnov (KS) pour surveiller ces variables. Ce test non-paramétrique s'est révélé particulièrement efficace pour isoler la variable défaillante (`temperature`), avec une p-value tendant vers zéro, tout en validant la stabilité des autres capteurs. Cette étape confirme l'importance des tests statistiques pour un diagnostic précis et automatisable.

Troisièmement, nous avons quantifié l'impact de ce drift sur le modèle de prédiction de pannes. En utilisant la méthode du bootstrapping, nous avons construit des intervalles de confiance pour l'accuracy du modèle. La comparaison entre la phase saine (Phase A) et la phase détériorée (Phase B) a mis en évidence une chute significative des performances. Cette dégradation souligne un risque majeur : un modèle "aveugle" au changement de données continue de faire des prédictions, mais avec une fiabilité considérablement réduite.

4.2 Apports du Projet

Ce travail met en lumière plusieurs enseignements clés pour l'industrialisation de l'IA :

— **La nécessité du monitoring constant :** Le déploiement d'un modèle n'est pas une fin en soi. Sans surveillance active de la distribution des données d'entrée, la qualité des prédictions peut s'effondrer silencieusement.

— **La complémentarité des approches :** Les tests statistiques (comme KS) agissent comme des "capteurs de capteurs", alertant sur la nature du changement, tandis que l'inférence sur la performance (bootstrapping) mesure la gravité de l'impact métier.
— **L'interprétabilité :** Savoir qu'un modèle performe moins bien est utile, mais comprendre *pourquoi* (ici, à cause de la température) permet aux équipes opérationnelles d'intervenir plus vite (réparation capteur ou maintenance machine).

4.3 Limites et Perspectives

Bien que les résultats obtenus soient probants, cette étude repose sur des données simulées avec un drift relativement simple (changement de moyenne). Dans un environnement réel, les phénomènes peuvent être plus complexes :

— **Drifts multiples et corrélés :** Plusieurs capteurs peuvent dériver simultanément de manière non linéaire.
— **Concept Drift :** La relation entre les variables et la panne peut changer (par exemple, une machine peut devenir plus robuste à la chaleur après une mise à jour matérielle), rendant la définition même de la "panne" mouvante.

Pour aller plus loin, plusieurs pistes d'amélioration peuvent être envisagées :

— **Mise en place d'un réentraînement automatique :** Déclencher un pipeline de *retraining* dès qu'un drift statistique dépasse un certain seuil.
— **Utilisation d'algorithmes de détection en ligne :** Adapter les méthodes pour des flux de données en temps réel (streaming) plutôt que par lots (batch).
— **Exploration de méthodes multivariées :** Utiliser des auto-encodeurs ou d'autres techniques de détection d'anomalies pour capter des changements dans les interactions entre variables, invisibles aux tests univariés comme KS.

En conclusion, la détection et la gestion du data drift sont des composantes indispensables d'une stratégie MLOps robuste, garantissant que les modèles d'intelligence artificielle restent des outils d'aide à la décision fiables et pérennes.

Annexe A

Annexes

A.1 Extraits de Code Python

Cette section présente les fonctions clés implémentées dans les notebooks pour la simulation et la détection du drift.

A.1.1 Simulation du Drift de Température

Le code suivant montre comment les données de production (Phase B) ont été générées avec un décalage de la moyenne de température (+7°C) pour simuler une surchauffe.

```
# Paramètres de simulation
n_samples = 1000

# Phase A : Données Normales
temp_A = np.random.normal(loc=75, scale=5, size=n_samples)

# Phase B : Données avec Drift (Surchauffe)
# Moyenne augmentée de 75 à 82
temp_B = np.random.normal(loc=82, scale=5, size=n_samples)

# Les autres variables (Vibration, Pression) restent stables
vib_A = np.random.normal(loc=0.5, scale=0.1, size=n_samples)
vib_B = np.random.normal(loc=0.5, scale=0.1, size=n_samples)
```

Listing A.1 – Génération des données avec Drift

A.1.2 Fonction de Bootstrapping

Implémentation de la fonction utilisée pour calculer les intervalles de confiance de l'accuracy.

```
def bootstrap_metric(X, y, model, metric_fn, n_bootstrap=1000):
    """
    Génère une distribution de scores par rééchantillonnage avec remise.
    """
    scores = []
    n = len(X)

    for _ in range(n_bootstrap):
        # Tirage avec remise (resampling)
        idx = np.random.choice(n, size=n, replace=True)
```

```
        X_sample = X.iloc[idx]
        y_sample = y.iloc[idx]

        # Prédiction et calcul du score
        y_pred = model.predict(X_sample)
        scores.append(metric_fn(y_sample, y_pred))

    return np.array(scores)
```

Listing A.2 – Fonction de Bootstrap pour l'inférence

A.2 Environnement Technique

Le projet a été réalisé sous Python 3.10. Les principales bibliothèques utilisées sont listées ci-dessous avec leurs versions.

Bibliothèque	**Usage Principal**
`numpy`	Calcul matriciel et génération aléatoire
`pandas`	Manipulation des DataFrames
`scikit-learn`	Modélisation (Pipeline, LogisticRegression)
`scipy`	Tests statistiques (Kolmogorov-Smirnov)
`matplotlib`	Visualisation des données

TABLE A.1 – Environnement technique du projet