UNIVERSITÉ DE PARIS
UFR MATHÉMATIQUES ET INFORMATIQUE

Rapport Qualité des Données : Système ADQ

Master 2 Données Connaissance Intelligence

Jaber BENTAYEB - Marya BOUSSA – Mayas MOKHTARI

Enseignant Soror SAHRI

Année universitaire 2023 – 2024

Table Des Matières

1. Introduction

La validation de la qualité des données est une étape cruciale, particulièrement dans les applications modernes axées sur les données. Les erreurs dans les données peuvent entraîner des comportements inattendus, en particulier lors du déploiement de modèles de machine learning.

Face à cette problématique, nous allons vous présenter une approche centrée sur les données pour automatiser la validation de la qualité des données. Cette méthode a été développée par des chercheurs de l'Université de Berlin et d'Amsterdam : *Sergey Redyuk, Zoi Kaoudi, Volker Markl et Sebastian Schelter* ; et a été présentée dans l'article « *Automating Data Quality Validation for Dynamic Data Ingestion* »[1].

Cette méthode se distingue des solutions existantes en ce qu'elle ne requiert pas l'intervention d'experts du domaine pour définir des règles spécifiques ou fournir des exemples étiquetés. Au lieu de cela, elle s'adapte automatiquement aux variations temporelles dans les caractéristiques des données, offrant ainsi une flexibilité et une adaptabilité accrue.

1.1 Les problèmes rencontrés

Avant de nous intéresser à ADQ, nous avions initialement envisagé d'utiliser d'autres outils comme Data Quality DataFrame (DQDF) et MLinspect. Cependant, nous avons rencontré des obstacles significatifs qui nous ont orientés vers un nouveau système.

Tout d'abord, nous avions l'impossibilité d'utiliser DQDF, pour cause de contraintes liées aux droits d'utilisation et à la propriété intellectuelle. DQDF

[1] *https://sergred.github.io/files/edbt.reds.pdf*

utilise DQLearn, un outil interne d'IBM, pour ses premières validations. Puisque le code est sous la propriété d'IBM et fait partie de leur répertoire interne, il est sujet à des restrictions d'accès et de partage.

Ensuite, il y a eu MLinspect, un outil conçu pour inspecter et auditer les pipelines de machine learning. Mais malheureusement, nous avons rencontré un problème lors de son installation, et nous n’avons donc pas pu l’utiliser.

Donc pour toutes ces raisons nous nous sommes finalement tournés vers ADQ. Un système que nous allons vous décrire dans la partie suivante.

2. Présentation du système

Dans cette partie nous allons détailler le fonctionnement du système ADQ.
Afin d'automatiser la validation de la qualité des données dans l'ingestion dynamique, notre algo suit plusieurs étapes essentielles. Vous trouverez ci-dessous un graphique explicatif résumant l'approche des auteurs.

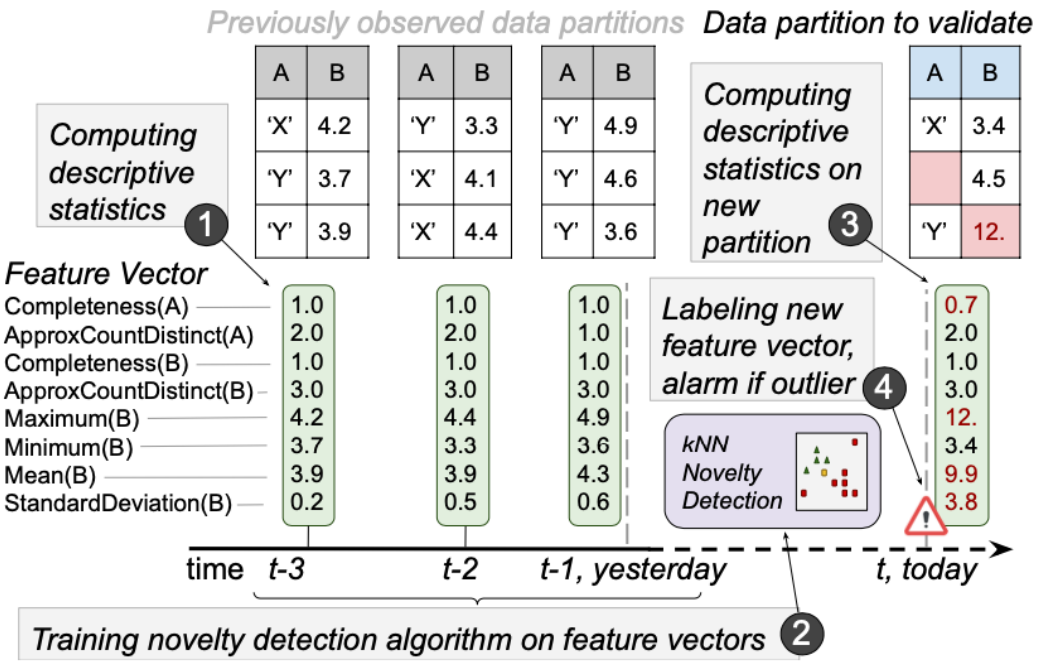


Figure 1 : Graphique montrant le Fonctionnement de l'algorithme

2.1 Calcul des Statistiques Descriptives

Dans cette première étape, un ensemble de statistiques descriptives est calculé pour chaque lot de données. Ces statistiques jouent un rôle crucial dans la compréhension des propriétés intrinsèques des données. Elles incluent :

- **Complétude** : Évalue dans quelle mesure les données sont complètes ou si elles comportent des lacunes.
- **Unicité** : Mesure l'unicité des enregistrements dans le jeu de données pour éviter les doublons.
- **Mesures pour Données Numériques** : Pour les données de type numérique, des statistiques telles que le maximum, la moyenne, le minimum, et l'écart type sont calculées. Ces mesures aident à comprendre la répartition et la variabilité des données.
- **Indice de Particularité pour Données Textuelles** : Pour les données textuelles, un indice spécifique est calculé pour évaluer leur caractère unique ou leur standardisation.

2.2 Modèle de Détection de Nouveauté

L'étape suivante implique l'utilisation d'un modèle de machine learning, le modèle utilisé est KNN :

- **Entraînement sur les Vecteurs de Caractéristiques** : Le modèle est entraîné sur les vecteurs de caractéristiques dérivés des statistiques descriptives. Cela permet au modèle d'apprendre quelles sont les caractéristiques des données acceptables.
- **Évaluation de Nouveaux Lots de Données** : Le modèle entraîné est ensuite utilisé pour évaluer de nouveaux lots de données. Il détecte les écarts significatifs par rapport aux normes établies, ce qui est essentiel pour identifier les anomalies.

2.3 Réentraînement et Adaptation

Cette phase garantit que le modèle reste actuel et efficace :

- **Mise à Jour Régulière** : Le modèle est régulièrement mis à jour avec de nouveaux lots de données. Cette mise à jour continue permet au modèle de s'adapter aux changements dans les caractéristiques des données.
- **Adaptabilité** : Cette adaptation constante garantit que le modèle reste sensible aux nouvelles tendances ou anomalies dans les données.

2.4 Avantages de cette Méthode

- **Détection Automatique d'Anomalies** : Cette approche permet de détecter automatiquement les anomalies sans nécessiter de règles spécifiques ou d'exemples propres à un domaine.
- **Adaptabilité et Efficacité** : La méthode est adaptable aux changements des caractéristiques des données et efficace en termes de calcul, ce qui la rend

particulièrement utile dans des environnements dynamiques et en évolution rapide.

En résumé, cette méthode offre une solution complète pour la surveillance et l'analyse des données, permettant une détection rapide et précise des anomalies, tout en s'adaptant aux changements continus des caractéristiques des données.

3. Expérimentations et Résultats

Dans cette partie, nous allons détailler les expérimentations que nous avons effectuées.
Tous les graphiques que nous allons présenter sont issus de notre code disponible sur GitHub[2].

3.1 Présentation des datasets

Pour nos experimentations, nous avons utilisé en tout trois datasets différents. Ce sont des données publiques que nous avons trouvées sur Kaggle. Dans cette sous-partie, nous allons décrire chacun des datasets.

1. Chicago Taxi Trips [3]: Ce jeu de données comprend des trajets de taxi pour l'année 2016, rapportés à la Ville de Chicago dans son rôle d'organisme de réglementation. C'est le dataset le plus lourd avec lequel nous avons travaillé (481.5 Mo).

   Les principales variables de ce dataset sont répertoriées dans la table suivante

Features	Type	Description
taxi_id	INTEGER	A unique identifier for the taxi.
trip_start_timestamp	TIMESTAMP	When the trip started, rounded to the nearest 15 minutes.
trip_end_timestamp	TIMESTAMP	When the trip ended, rounded to the nearest 15 minutes.
trip_seconds	INTEGER	Time of the trip in seconds.
trip_miles	FLOAT	Distance of the trip in miles.
pickup_census_tract	INTEGER	The Census Tract where the trip began. For privacy, this Census Tract is not shown for some trips.
dropoff_census_tract	INTEGER	The Census Tract where the trip ended. For privacy, this Census Tract is not shown for some trips.
pickup_community_area	INTEGER	The Community Area where the trip began.
dropoff_community_area	INTEGER	The Community Area where the trip ended.
fare	FLOAT	The fare for the trip.
tips	FLOAT	The tip for the trip. Cash tips generally will not be recorded.

[2] *https://github.com/ben-tayeb-jab/qualite_project_ADQ*
[3] *https://www.kaggle.com/datasets/chicago/chicago-taxi-rides-2016*

tolls	FLOAT	The tolls for the trip.
extras	FLOAT	Extra charges for the trip.
trip_total	FLOAT	Total cost of the trip, the total of the fare, tips, tolls, and extras.
payment_type	STRING	Type of payment for the trip.
company	INTEGER	The id code for the taxi company.
pickup_latitude	INTEGER	The id code for the latitude of the center of the pickup census tract or the community area if the census tract has been hidden for privacy.
pickup_longitude	INTEGER	The id code for the center of the pickup census tract or the community area if the census tract has been hidden for privacy.
pickup_location	STRING	The location of the center of the pickup census tract or the community area if the census tract has been hidden for privacy.
dropoff_latitude	INTEGER	The id code for the center of the dropoff census tract or the community area if the census tract has been hidden for privacy.
dropoff_longitude	INTEGER	The id code for the center of the dropoff census tract or the community area if the census tract has been hidden for privacy.
dropoff_location	STRING	The location of the center of the dropoff census tract or the community area if the census tract has been hidden for privacy.

2. King County House Sales [4]: Ce jeu de données contient les prix de vente de maisons pour le comté de King, qui inclut Seattle. Il comprend les maisons vendues entre mai 2014 et mai 2015.

   Les principales variables de ce dataset sont répertoriées dans la table suivante :

Feature Name	Type	Description
id	Integer	A unique identifier for each house.
date	Date	The date when the house was sold.
price	Numeric	The sale price of the house (target variable).
bedrooms	Integer	The number of bedrooms in the house.
bathrooms	Numeric	The number of bathrooms in the house, often in half-bath increments.
sqft_living	Numeric	The square footage of the home's interior living space.

[4] *https://www.kaggle.com/datasets/harlfoxem/housesalesprediction*

sqft_lot	Numeric	The square footage of the land space.
floors	Numeric	The number of floors (levels) in the house.
waterfront	Categorical/Binary	Indicates whether the house is on a waterfront (1) or not (0).
view	Integer	An index from 0 to 4 of how good the view of the property is.
condition	Integer	An index from 1 to 5 on the condition of the house.
grade	Integer	An index from 1 to 13, where 1-3 falls short, 7 is average, and 11-13 is high quality.
sqft_above	Numeric	The square footage of the house apart from the basement.
sqft_basement	Numeric	The square footage of the basement.
yr_built	Integer	The year when the house was initially built.
yr_renovated	Integer	The year of the house's last renovation.
zipcode	Integer	The ZIP code area of the house.
lat	Numeric	The latitude coordinate of the house.
long	Numeric	The longitude coordinate of the house.
sqft_living15	Numeric	The square footage of interior housing living space for the nearest 15 neighbors.
sqft_lot15	Numeric	The square footage of the land lots of the nearest 15 neighbors.

3. Steel Industry Energy Consumption [5]: Ce jeu de données provient du dépôt d'apprentissage automatique de l'UCI (Université de Californie à Irvine). Cette entreprise produit plusieurs types de bobines, de plaques d'acier et de plaques de fer. Les informations sur la consommation d'électricité sont conservées dans un système basé sur le cloud. Les informations sur la consommation d'énergie de l'industrie sont stockées sur le site web de la Korea Electric Power Corporation (pccs.kepco.go.kr), et les perspectives sur les données quotidiennes, mensuelles et annuelles sont calculées et affichées.

[5] *https://www.kaggle.com/datasets/csafrit2/steel-industry-energy-consumption*

Feature Name	Type	Description
Date	Continuous	Time data recorded on the first of the month.
Usage_kWh	Continuous	Industry energy consumption measured in kilowatt-hours (kWh).
Lagging Current	Continuous	Reactive power (lagging) measured in kilovolt-ampere reactive hours (kVarh).
Leading Current	Continuous	Reactive power (leading) measured in kilovolt-ampere reactive hours (kVarh).
CO2	Continuous	Carbon dioxide emissions measured in parts per million (ppm).
NSM	Continuous	Number of Seconds from Midnight - a continuous measure of time in seconds since midnight.
Week status	Categorical	Indicates whether the day is a weekend (0) or a weekday (1).
Day of week	Categorical	The specific day of the week, ranging from Sunday to Saturday.
Load Type	Categorical	The category of load demand, classified as Light Load, Medium Load, or Maximum Load.

3.2 Métriques de cohérence

Dans le cadre de nos expérimentations, nous avons défini et calculé des métriques de cohérence spécifiques pour chaque dataset. Ces métriques nous permettent d'évaluer la qualité et la fiabilité des données. Elles servent à identifier les incohérences ou les anomalies au sein des ensembles de données, ce qui est crucial pour garantir la précision des analyses ultérieures. Voici un aperçu détaillé des métriques utilisées pour chaque dataset.

3.2.1 **Dataset Chicago Taxi Trips**

1. **Coherence Temporelle** : Cette métrique calcule la proportion de trajets où la date et l'heure de fin sont postérieures à la date et l'heure de début. Un score de **0.56** indique que plus de la moitié des enregistrements sont chronologiquement cohérents.

2. **Coherence Durée** : Évalue la cohérence entre la durée réelle du trajet (calculée à partir des timestamps) et la durée déclarée. Une tolérance de 60 secondes est permise.

3.2.2 Dataset House Sales

1. **Positive Price** : Vérifie que les prix de vente des maisons sont des nombres positifs.
2. **Correct Year Built** : Assure que l'année de construction est logiquement antérieure à l'année de vente. Le score élevé de **0.99** montre une grande cohérence dans ce domaine.
3. **Non-negative Bedrooms/Bathrooms/Floors** : Confirme que le nombre de chambres, salles de bain et étages est positif.
4. **Correct Sqrt Living** : Vérifie que la surface habitable est égale à la somme des surfaces du sous-sol et de la maison. Un score de **1** pour toutes ces mesures indique une cohérence parfaite.

3.2.3 Dataset Steel Industry

1. **Usage kWh Score** : Évalue si les valeurs de consommation d'énergie (Usage_kWh) sont non négatives. Un score parfait de **1** indique que toutes les valeurs sont non négatives.

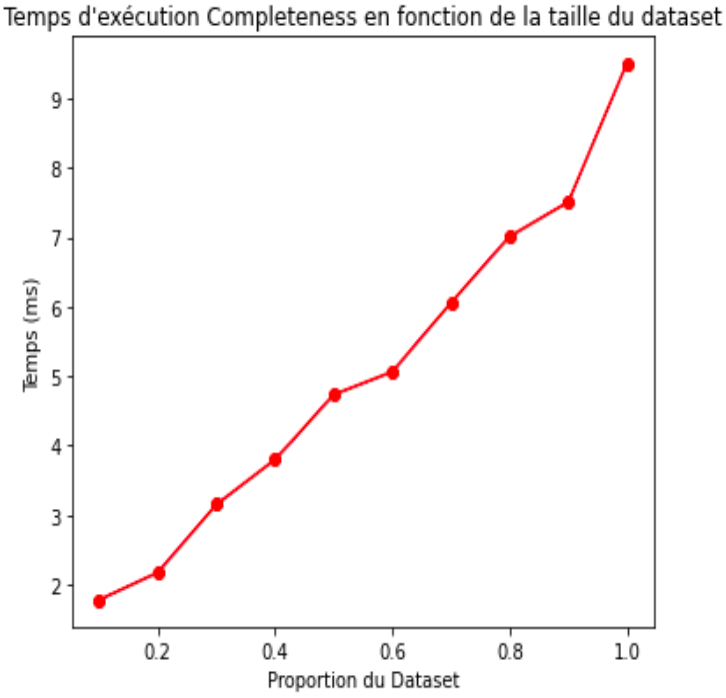
3.2.4 Tableau Récapitulatif des Métriques et Valeurs

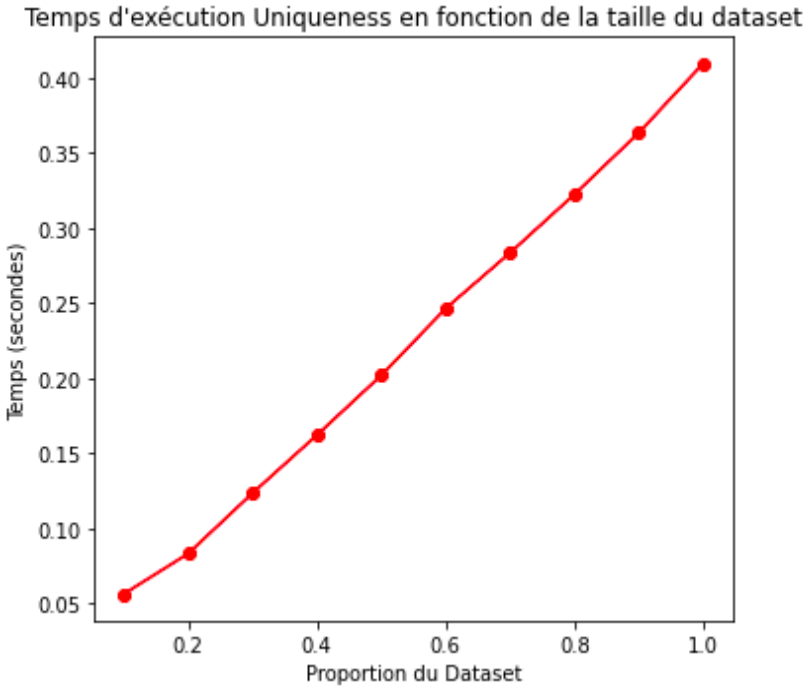
Dataset	Métrique	Valeur
Chicago Taxi Trips	Coherence Temporelle	0.56
	Coherence Durée	0.21
House Sales	Positive Price	1
	Correct Year Built	0.99
	Non-negative Bedrooms/Bathrooms/Floors	1
	Correct Sqrt Living	1
Steel Industry	Usage kWh Score	1

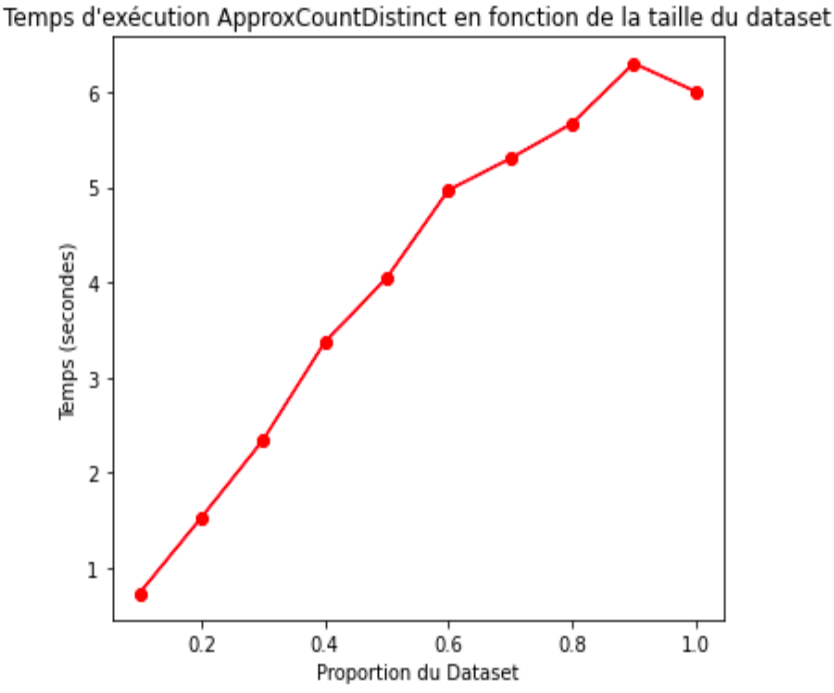
Ces métriques nous offrent une vision approfondie de la qualité des données, nous permettant d'identifier et de rectifier d'éventuelles incohérences, assurant ainsi la fiabilité de nos analyses.

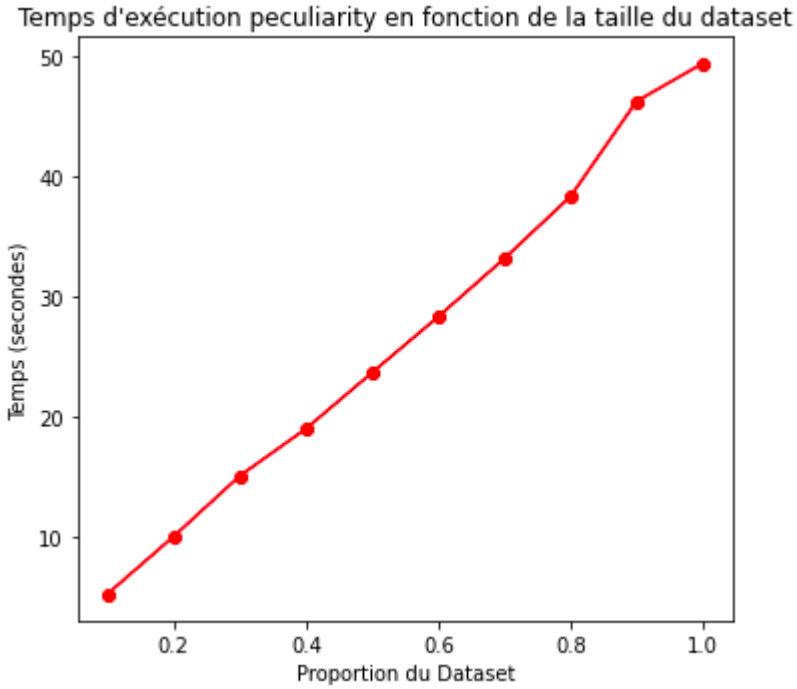
3.3 Scalabilité

Nous avons mené une évaluation approfondie de la scalabilité de nos métriques en utilisant des indicateurs tels que la complétude, l'unicité, l'approxcountdistinct et la peculiarity. Pour cette analyse, nous avons exclusivement utilisé le jeu de données des trajets en taxi de Chicago de 2016, en nous concentrant particulièrement sur la variable 'taxi_id'. Cette décision a été prise après avoir constaté que la comparaison de la scalabilité entre différents jeux de données n'était pas pertinente, car cela aurait résulté en des graphiques similaires et donc non informatifs.



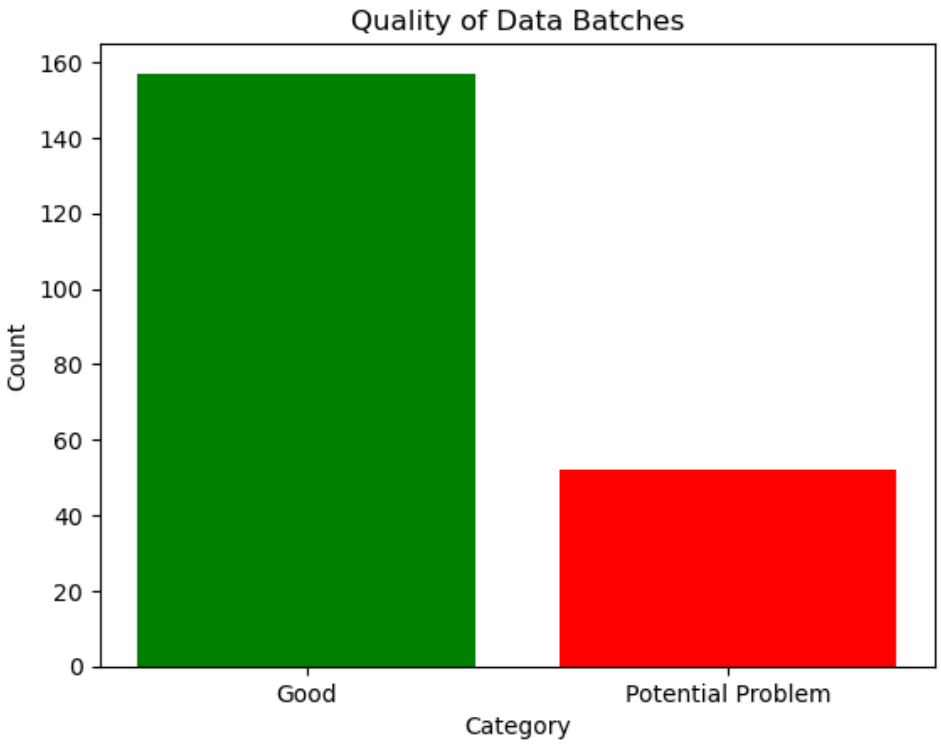

Temps d'exécution Uniqueness en fonction de la taille du dataset
Temps (secondes)
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.2
0.4
0.6
0.8
1.0
Proportion du Dataset


Temps d'exécution ApproxCountDistinct en fonction de la taille du dataset
Temps (secondes)
6
5
4
3
2
1
0.2
0.4
0.6
0.8
1.0
Proportion du Dataset

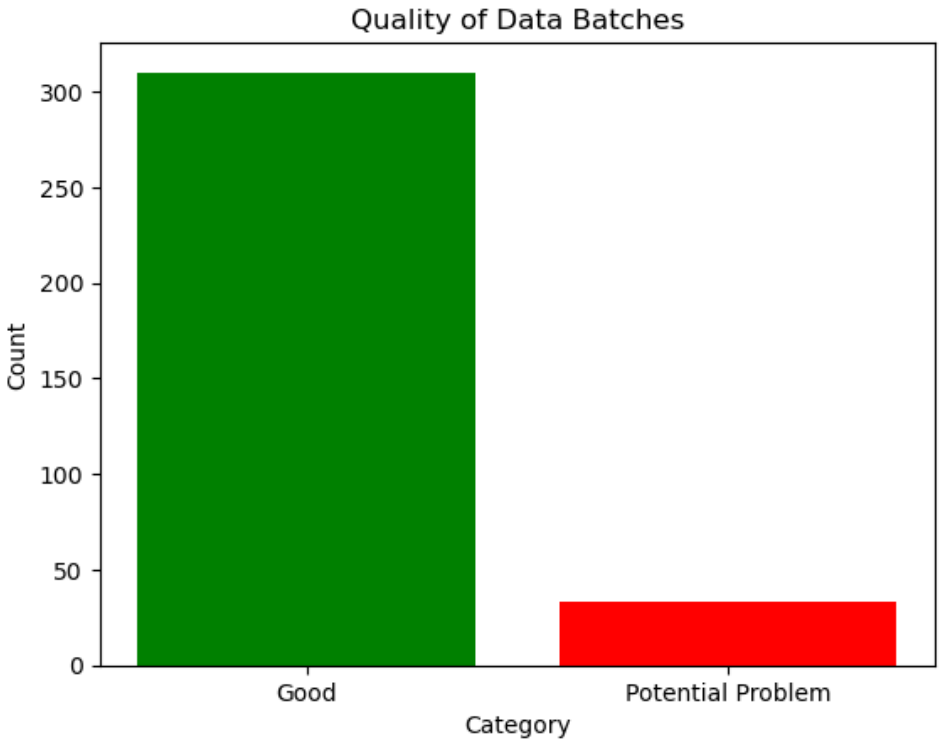


En analysant ces graphiques, il apparaît clairement que le temps d'exécution augmente de manière linéaire avec la taille du jeu de données. Les mesures de complétude et d'unicité se situent dans l'ordre de quelques millisecondes. En revanche, la métrique approxcountdistinct prend environ 6 secondes pour s'exécuter, tandis que la métrique de peculiarity est la plus longue à calculer, prenant presque une minute pour l'ensemble du jeu de données. Cette différence de performance est logique étant donné la complexité des fonctions : les fonctions de complétude et d'unicité sont définies en une ou deux lignes, tandis que approxcountdistinct intègre une boucle for et que la fonction de particularité est plus élaborée, s'étendant sur plusieurs lignes.

3.4 Application de notre algorithme



King County House Sales



Steel Industry Energy

Lors de l'application de l'algorithme de notre système ADQ sur les jeux de données King County House Sales et Steel Industry Energy Consumption, nous avons observé une prédominance de résultats classifiés comme 'Good' par rapport à ceux identifiés comme 'Potential Problem'. Cette tendance suggère que la qualité des données dans ces deux ensembles est généralement élevée.

3.5 Synthèse

Notre système automatisé pour la validation de la qualité des données se révèle être un outil puissant pour les environnements de données dynamiques et en constante évolution. Il élimine le besoin de règles spécifiques au domaine et de supervision manuelle, tout en s'adaptant aux variations temporelles des données. Cette flexibilité et cette efficacité rendent notre approche particulièrement adaptée aux applications modernes de science des données, où la rapidité et la précision sont primordiales.

4. Comparatif avec les systèmes Deequ et TFDV

4.1 Comparatif temps d'exécution Completeness

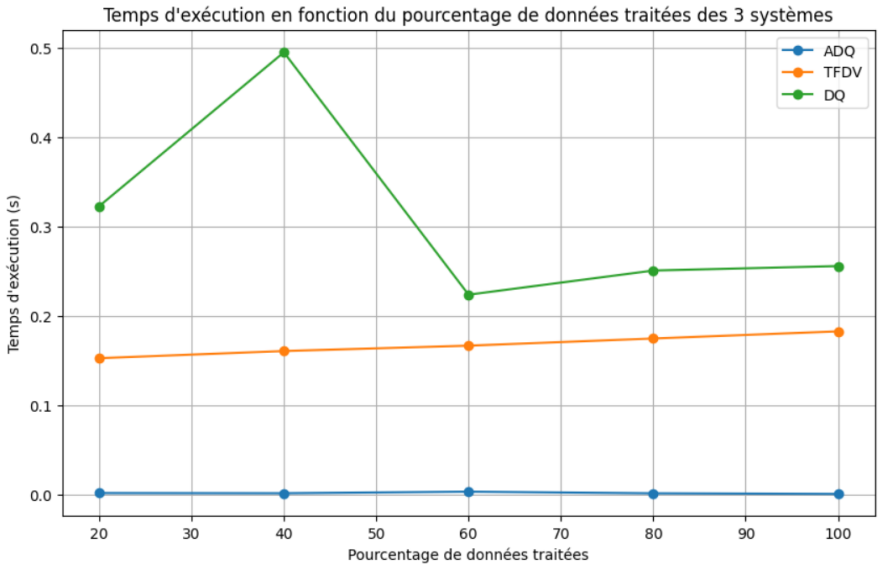


Figure 2 : Graphique comparatif de la Completeness des trois systèmes sur le dataset House Sales

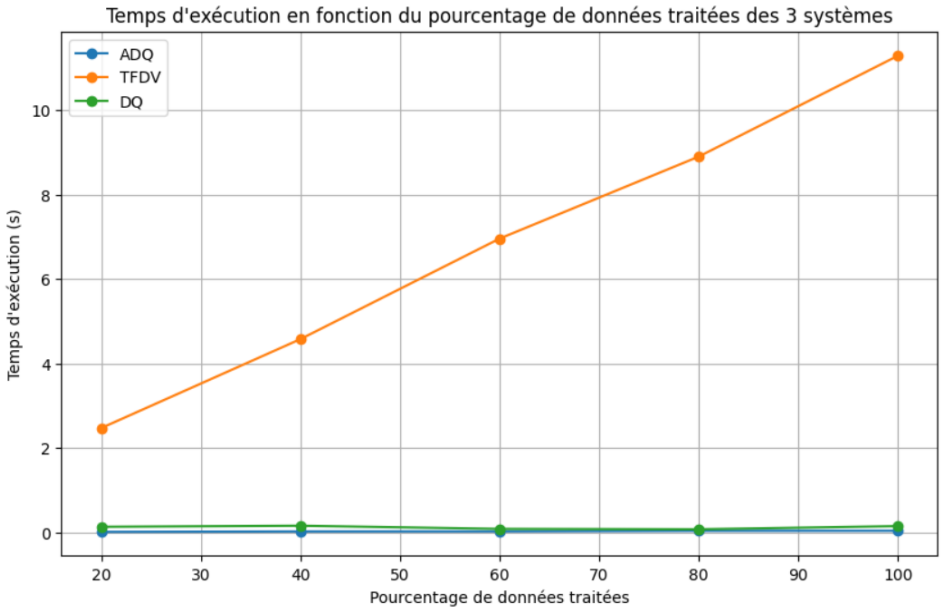


Figure 3 : Graphique comparatif de la Completeness des trois systèmes sur le dataset Chicago Taxi Trips

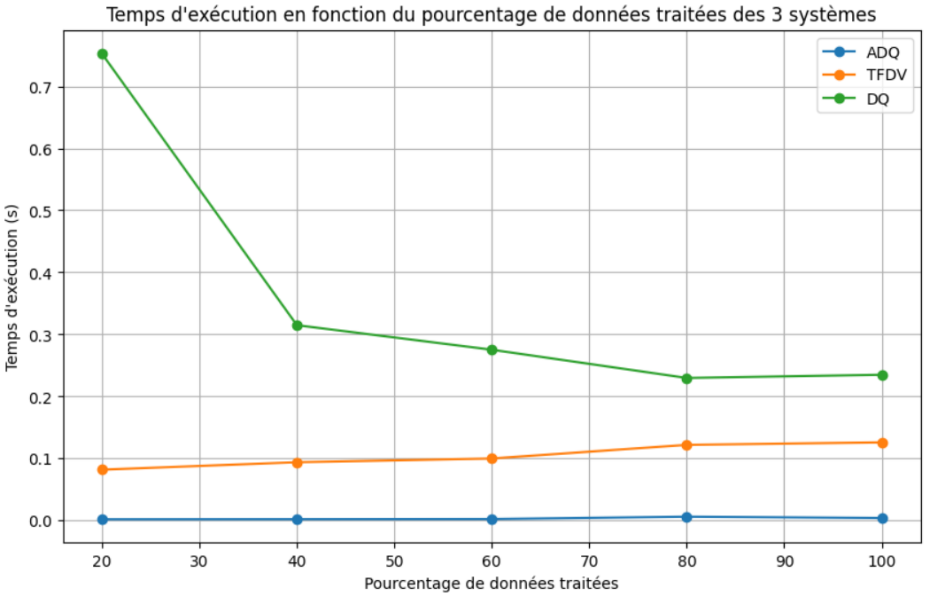


Figure 4 : Graphique comparatif de la Completeness des trois systèmes sur le dataset Steel Industry

4.2 Comparatif temps d'exécution Uniqueness

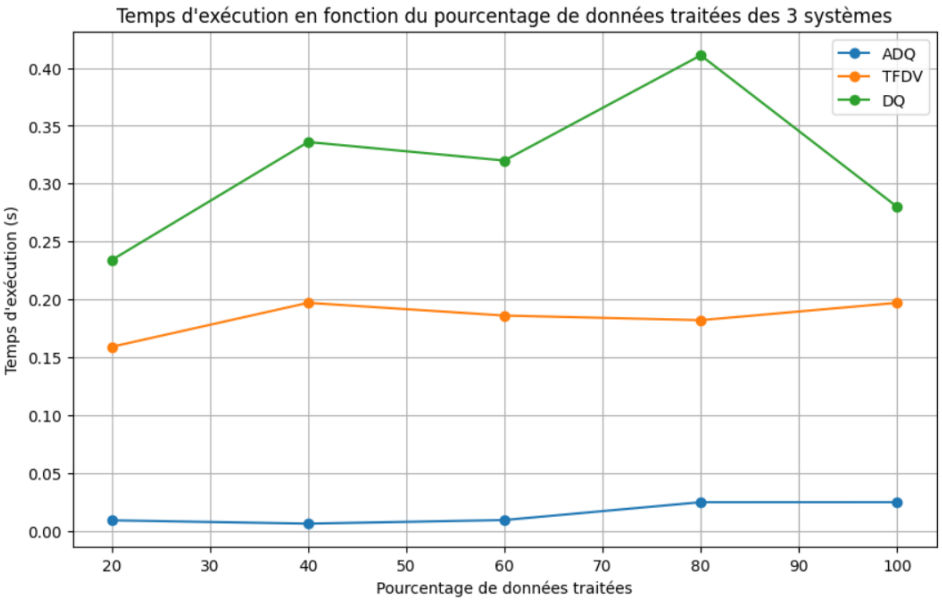


Figure 5 : Graphique comparatif de la Uniqueness des trois systèmes sur le dataset House Sales

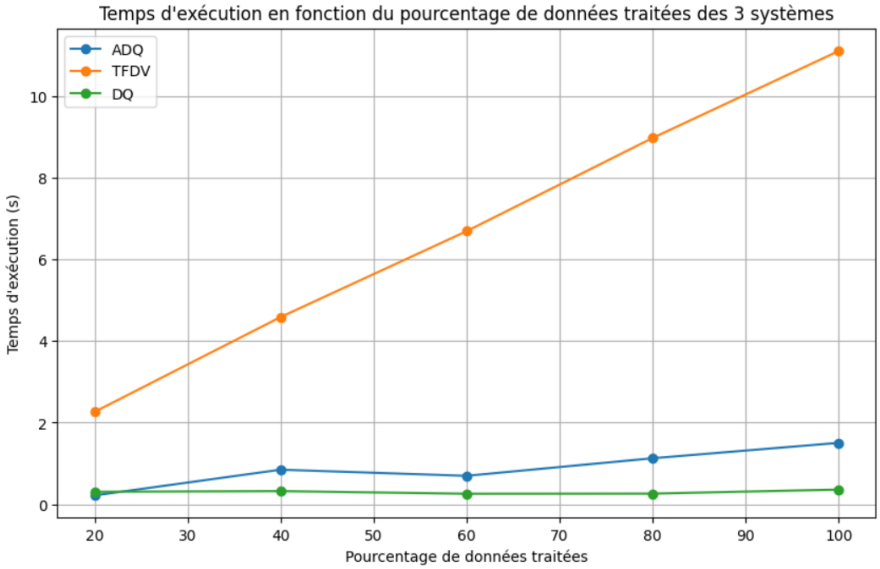


Figure 6 : Graphique comparatif de la Uniqueness des trois systèmes sur le dataset Taxi Trips

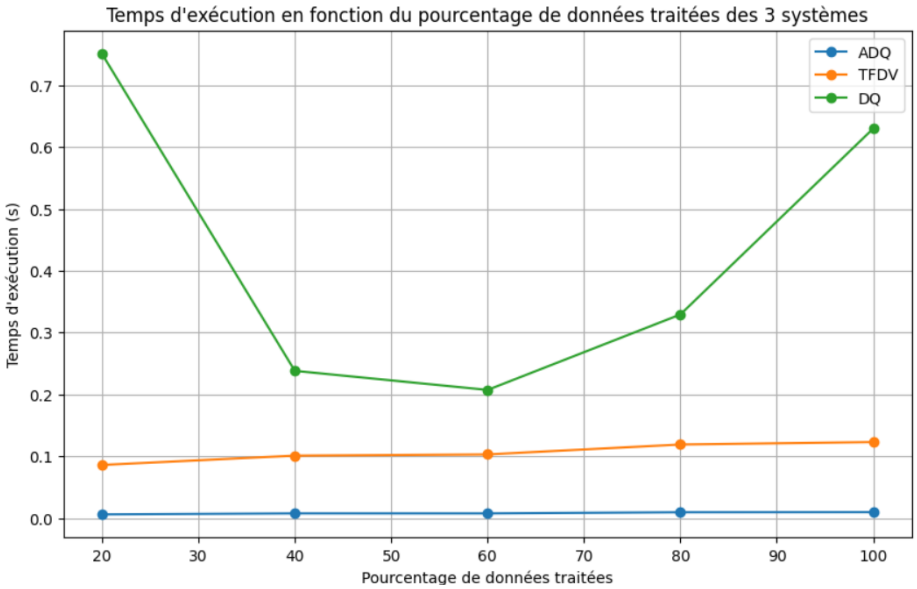


Figure 7 : Graphique comparatif de la Uniqueness des trois systèmes sur le dataset Steel Industry

Dans notre comparatif de scalabilité impliquant les systèmes Deequ, ADQ et TFDV, en nous basant sur nos graphiques, il ressort clairement que le système ADQ surpasse les autres en termes de vitesse. Il est suivi de près par TFDV, tandis que Deequ se positionne en dernier. Cette hiérarchie en termes de rapidité, particulièrement notable sur les métriques évaluées, est en adéquation avec la conception intrinsèque de notre système ADQ. En effet, dans ADQ, les fonctions sont établies de façon simple et épurée, en contraste avec les structures plus complexes et les contraintes additionnelles présentes dans Deequ et TFDV.

4.3 Comparatif des scores Completeness et Uniqueness

System	KC House Sales		Chicago Taxi Trips		Steel Industry	
	Cmpl	Uniq	Cmpl	Uniq	Cmplt	Uniq
ADQ	1.0	0.9836672	0.999981	0.000105	1.0	1.0
Deequ	1.0	0.983667237	0.999982	0.000713	1.0	1.0
TFDV	1.0	0.046275	0.9999808	2.5813273e-06	1.0	0.0292483

Lors de notre comparaison des calculs des scores des différentes métriques (Completeness et Uniqueness), nous observons une similarité frappante entre les scores obtenus avec Deequ et ADQ. En revanche, bien que le système TFDV affiche des

scores de complétude comparables à ceux de Deequ et ADQ, il se distingue nettement en ce qui concerne la métrique d'unicité. Cette divergence notable dans les résultats de l'unicité entre TFDV et les deux autres systèmes pourrait indiquer que la méthode de calcul de cette métrique chez TFDV n'est pas aussi bien définie ou efficace que chez Deequ et ADQ.

5. Conclusion

Ce qu'on pourrait conclure, c'est que le système sur lequel nous avons travaillé, basé sur le modèle KNN, classifie directement les ensembles de données comme "Good" ou "Potential Problem" en fonction de leur profilage. Cette approche fournit une réponse binaire directe basée sur l'apprentissage machine.

TFDV et Deequ, en revanche, fournissent une analyse plus granulaire des données, mettant en évidence des aspects spécifiques qui peuvent être problématiques. Ils nécessitent une interprétation plus approfondie des résultats pour déterminer si un ensemble de données est "bon" ou "mauvais".

Bien que TFDV et Deequ ne fournissent pas directement une étiquette "good" or " Potential Problem", ils offrent des outils puissants pour une compréhension détaillée de la qualité des données. Ces outils sont particulièrement utiles dans des environnements où la compréhension fine des caractéristiques de qualité des données est cruciale, même si cela exige une interprétation et une analyse plus poussées par l'utilisateur. Le système ADQ, avec sa classification binaire, est plus direct mais pourrait manquer de cette granularité et de cette profondeur d'analyse.

En fin de compte, la comparaison de la vitesse d'exécution est utile, mais elle doit être contextualisée dans le cadre plus large des objectifs, de la précision, et de l'utilisation de chaque outil. La vitesse est un avantage si elle répond à un besoin spécifique sans compromettre de manière significative d'autres aspects essentiels tels que la précision et la fiabilité.