Vers des requêtes déclaratives en science des données : EASI-GDS pour Neo4J

Valentin Bouvresse*, Jacques Chabin*, Virgile Crvenka*, Mirian Halfeld-Ferrari*, Genoveva Vargas-Solar**, Lingchen Wang*

* LIFO – Université d'Orléans, INSA CVL, Orléans, France
{valentin.bouvresse,virgilecrvenka}@gmail.com,
{jchabin, mirian,lingchen.wang}@univ-orleans.fr
** CNRS, Univ Lyon, INSA Lyon, UCBL, LIRIS, UMR5205, 69622, Villeurbanne
genoveva.vargas-solar@cnrs.fr

Résumé. L'article décrit un outil convivial destiné à simplifier le paramétrage de différents modèles dans Neo4J GDS. Conçu spécialement pour les utilisateurs non experts, cet outil offre des interfaces graphiques et conversationnelles, éliminant ainsi la nécessité de remplir un modèle complexe dans Neo4J. Il facilite l'analyse des graphes de données et compare les résultats des modèles. Le meilleur résultat est automatiquement retourné sous forme d'un graphe ou d'une relation. Lorsque le résultat est présenté sous forme de graphe, les utilisateurs ont la possibilité d'effectuer des opérations supplémentaires.

À l'avenir, nous prévoyons d'enrichir davantage cet outil en y ajoutant des fonctionnalités telles que la réutilisation des résultats. Notre objectif est de permettre à un public plus large de s'engager dans des tâches d'analyse de données, favorisant l'accessibilité dans le domaine de la science des données.

1 Introduction

Avec la croissance rapide du volume et de l'accessibilité des données connectées, notamment sous forme de graphes, la science des données a conduit à l'émergence d'un nouveau type de requêtes 'complexes' pour répondre aux exigences de l'analyse des données. Dans ce contexte, notre objectif est de concevoir des outils mettant l'accent sur la spécification déclarative des requêtes en science des données, afin de permettre à des utilisateurs non spécialistes, aux compétences variées, de les exécuter efficacement.

Cet article introduit EASI-GDS, un outil d'interrogation déclarative pour l'analyse des graphes de propriétés stockés dans des SGBD graphes (par exemple, Neo4J). En effet, cette première version de notre framework est conçue comme interface amicale sur le module *data science* de Neo4J. EASI-GDS remplit les fonctions suivantes :

1. Il aide les utilisateurs à exprimer leurs requêtes de manière conversationnelle et conviviale, il les transforme en pipelines d'opérations pour définir la vue du graphe à analyser et sélectionner le type approprié de modèle d'analyse de données (tel que la prédiction, la détection de communautés ou la recherche de chemins).

2. Si plusieurs modèles sont utilisés pour une même tâche, par exemple la prédiction de liens, il met en évidence les résultats de l'approche la plus efficace, celle qui obtient les meilleurs scores.
3. Il permet de stocker les résultats de l'analyse dans la base de données pour les utiliser comme données d'entrée pour les sous-requêtes spécifiées dans la requête principale.

L'interface graphique d'EASI-GDS fournit un framework flexible permettant à plus d'utilisateurs de s'engager dans l'analyse des données sans avoir besoin de connaissances spécifiques en apprentissage automatique.

Le reste de cet article est structuré comme suit. Après une brève contextualisation de notre travail et un survol de certains outils et travaux connexes (section 2), la section 3 expose des principes liés à la conception d'outils déclaratifs pour l'analyse des données. Elle récapitule également les contributions d'EASI-GDS. La section 4 détaille l'architecture et les technologies employées pour la création de l'outil. Ensuite, la section 5 présente un scénario d'utilisation d'EASI-GDS. Enfin, la section 6 conclut l'article en abordant des perspectives futures.

2 Travaux liés, contexte

Notre travail s'inscrit dans le cadre des initiatives de l'action DOING [1] du GDR MADICS, axée sur l'étude des méthodes intelligentes de gestion des bases de données et l'exploration de nouvelles formes de requêtes. Plus précisément, DOING s'intéresse à la conception de requêtes déclaratives capables d'exprimer les besoins des utilisateurs en termes d'analyse de données en tant qu'aide à la prise de décision.

Notre engagement dans cette action reflète également notre intérêt marqué pour les bases de données graphes. En effet, les graphes mettent en lumière les relations entre les données, accordant ainsi une importance particulière aux requêtes portant sur la topologie, les clusters, et autres aspects similaires.

Neo4j est un système de gestion de base de données le plus populaire aujourd'hui, offrant différentes plateformes de travail et bibliothèques. Ainsi, Neo4J Desktop est une application permettant de gérer les instances locales de Neo4J, tandis que Neo4J Browser est une interface de navigation en ligne permettant d'interroger et de visualiser les données de la base. Neo4J Bloom est un outil de visualisation dirigés à de non-experts en base de données. L'idée de permettre aux utilisateurs d'analyser leur données plus facilement est donc d'actualité. Neo4J AuraDB est l'offre Neo4J comme un *database as a service* (DBaaS) qui permet aux utilisateurs d'accéder à un système de base de données en *cloud* et de l'utiliser sans avoir à acheter et à configurer leur propre matériel ou à installer leur propre logiciel de base de données.

Néanmoins, l'utilisation des bibliothèques et plateformes pour interagir avec une base de données requiert toujours la configuration de nombreux paramètres, ce qui n'est pas convivial pour les utilisateurs non-experts. Par conséquent, la création des interfaces interactives pour l'analyse des données dans Neo4J continue d'être une initiative importante.

Des études comme celle de Vargas-Solar et al. (2021) comparent les performances des bibliothèques de science des données de Networkx et de Neo4j. Vargas-Solar et al. (2023) et Bethaz et al. (2021) se concentrent sur l'utilisation du langage naturel(NL) pour exprimer les requêtes de science des données(DSQ). Alors que Vargas-Solar et al. (2023) propose un

1. Site :`https://www.univ-orleans.fr/lifo/evenements/doing/`

traducteur de NL en DSQ, Bethaz et al. (2021) traite de la démocratisation de l'analyse des données.

3 Construire des outils analytiques déclaratifs : la première étape

Dans le domaine de la science des données, les pipelines d'apprentissage automatique jouent un rôle crucial. Ils représentent une construction intégrale qui coordonne le flux de données à travers un modèle d'apprentissage automatique (ou un ensemble de modèles multiples) ainsi que les sorties générées par ce modèle. Un pipeline englobe l'ensemble du processus, allant de l'entrée des données brutes, des caractéristiques, du modèle d'apprentissage automatique et de ses paramètres, jusqu'aux sorties de prédiction.

La bibliothèque GDS Neo4J offre la possibilité de concevoir et mettre en œuvre ces pipelines. À cet effet, elle propose une collection de configurations pour les modèles candidats, qui est initialement vide. Cette collection est appelée l'espace des paramètres. La construction d'un pipeline implique alors la spécification des paramètres bien précis. Chaque configuration de modèle candidat contient des valeurs fixes ou des plages pour les paramètres d'apprentissage. Il implique une projection du graphe en amont, pour télécharger la vue pertinente dans la mémoire vive, une estimation des ressources nécessaires pour réaliser l'analyse et une application de modèle spécifiant les paramètres et le format des résultats à l'aide d'une syntaxe stricte. Les modèles d'une requête de science des données sont calibrés avec différents paramètres jusqu'à ce que le résultat converge. Les modèles résultants sont comparés entre eux ainsi avec les résultats de modèles équivalents afin de choisir celui qui offre les meilleures performances. Le processus peut être complexe pour les non-experts en raison de la nécessité de comprendre les paramètres et de prendre des décisions sur le stockage des modèles en fonction des ressources disponibles.

Inspiré de l'IAM (Intentional Analytics Methods) Vassiliadis et al. (2019); Sarker (2021), EASI-GDS permet aux utilisateurs d'exprimer des requêtes, de produire des pipelines de science des données sans s'occuper de détails nondispensables et de les exécuter sur une base de données Neo4J. En arrière plan, l'interprétation des demandes des utilisateurs implique la classification de différents types d'intentions et la conception de modèles, chacun représentant un pipeline qui fournit des réponses à des intentions spécifiques. Quatre catégories d'intentions sont proposées dans Vassiliadis et al. (2019) : (1) Describe : fournit une vue d'ensemble des données et de la topologie du graphe, en répondant à la question " Que s'est-il passé ? ". (2) Assess : évalue la qualité d'une situation par rapport à une référence, en se concentrant sur les mesures de similarité ou de centralité. (3) Explain : va au-delà de l'état actuel des données pour répondre à la question " Pourquoi cela s'est-il produit ? " en identifiant les facteurs sous-jacents. (4) Prédire : répond à la question "Que se passera-t-il à l'avenir ?" avec une forte probabilité, notamment en prévoyant les relations dans le graphe.

Notre application propose six modèles mis en œuvre par les pipelines : similarité, centralité, détection de communautés, classification, régression et prédiction de liens topologiques. Ces modèles sont nécessaires pour mettre en œuvre les intentions des utilisateurs citées plus haut. EASI-GDS fournit deux interfaces conviviales qui aident les utilisateurs à remplir les modèles pour chaque pipeline.

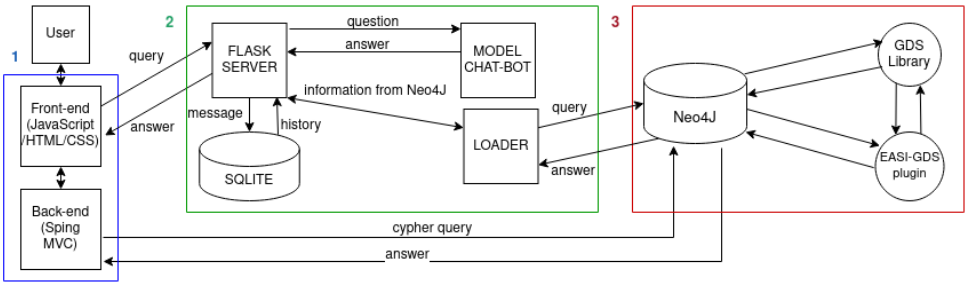

FIG. 1 – *Architecture générale de l'EASI-GDS .*

Contributions. EASI-GDS complète certaines fonctions de la librairie GDS de Neo4J. Tout d'abord, il applique la classification des propriétés des nœuds sur les données textuelles dans un appel de procédure, ce qui surmonte la limitation de Neo4j à ne prédire que les nombres. Deuxièmement, il prédit les relations basées sur des chemins de taille supérieure ou égale à deux, ce qui va au-delà de la prédiction d'un seul lien dans le GDS de Neo4J. Enfin, il automatise la gestion des modèles en sélectionnant les modèles à conserver lorsque l'espace de stockage est limité. Cette stratégie favorise une utilisation judicieuse des ressources de stockage. En effet, les modèles entraînés peuvent être stockés dans le GDS de Neo4J en fonction des limites d'accès de l'utilisateur (les utilisateurs de la version d'essai gratuite ne peuvent conserver que trois modèles). La version actuelle d'EASI-GDS met en œuvre une politique de suppression qui élimine automatiquement un modèle pour faire de la place à un nouveau. EASI-GDS classe les modèles par ordre de priorité en fonction de leur importance et de leur temps d'exécution. La priorité est calculée en multipliant le temps de géneration de chaque modèle par son nombre de fois où il a été utilisé. Le modèle ayant le score le plus bas (rapide à former ou rarement utilisé) est supprimé, ce qui garantit une gestion judicieuse du stockage des modèles.

4 Faire en sorte que les utilisateurs se sentent à l'aise : des interfaces conviviales

L'architecture à trois niveaux d'EASI-GDS (figure 1) se compose du module Query-Reply Generator (boîte 1 dans la figure 1) mis en œuvre avec Spring-MVC et Thymeleaf pour les modèles HTML, pour préparer et exécuter des requêtes en interagissant avec Neo4J. Le *front-end* de EASI-GDS est implémenté en JavaScript et fournit deux interfaces pour aider les utilisateurs à exprimer et à exécuter des requêtes analytiques :

1) Interface graphique. Les requêtes déclaratives peuvent être exprimées par des *templates*. Par exemple, la figure 2 présente un exemple de *template*, conçu pour la prédiction. Notre travail a été donc, au début, de concevoir les modèles (*templates*) qu'on souhaiterait utiliser. Au lieu de demander aux utilisateurs d'écrire leurs requêtes en suivant ces modèles, nous leur proposons une interface graphique où ils peuvent spécifier certains choix, remplissant ainsi, indirectement, les modèles qui sont à l'origine des requêtes à effectuer. De plus, si un utilisateur ne sait pas choisir, ces modèles peuvent être remplis par des choix pré-déterminés par des spécialistes.

À partir d'un modèle rempli, les requêtes sont ensuite générées par le module *Query-Reply Generator*. La requête de science des données qui en résulte met en œuvre un pipeline de sous-requêtes à exécuter par Neo4J.

2) Interface conversationnelle. Un *chatbot fait-maison* (boîte 2 de la figure 1) permet aux utilisateurs d'engager des conversations simples avec EASI-GDS . Il est construit à partir d'un réseau de neurones entraîné et d'un modèle de prédiction. Le fichier d'entrainement JSON contient un objet *intents* qui énumère les différents types de questions que les utilisateurs sont censés poser. À chaque type de question correspondent des "modèles" indiquant les phrases que les utilisateurs peuvent exprimer. En outre, une liste de réponses est disponible pour fournir des réponses variées, améliorant ainsi l'expérience conversationnelle des utilisateurs. Un serveur Flask relie l'interface Web EASI-GDS au *chatbot*. Enfin, le troisième niveau d'EASI-GDS (boîte 3 de la figure 2) comprend le SGBD Neo4J avec sa bibliothèque de science des données (GDS) et un *plugin* de procédure Java que nous avons mis en œuvre pour simplifier l'utilisation des modèles GDS.

Visualisation. Les résultats des requêtes peuvent être visualisés sous forme de graphes ou de tables. Par défaut, à l'exception des ensembles de résultats vides ou volumineux, le format graphe est utilisé, en utilisant la bibliothèque neo4jD3. EASI-GDS enrichit neo4jD3 en y ajoutant des fonctionnalités qui permettent aux utilisateurs : de visualiser un graphe centré, de centraliser la souris, de représenter des relations réflexives ou multiples entre les noeuds. De plus, la taille des étiquettes de nœuds et la taille des nœuds est dynamiquement adaptée selon les valeurs de certains attributs. Il s'agit d'un moyen puissant et flexible de visualiser et d'analyser les résultats des requêtes.

5 Démonstration

Lors de la démonstration de EASI-GDS, notre objectif est de mettre en évidence la capacité de l'outil à permettre aux utilisateurs d'exprimer des requêtes de prédiction sur des graphes de propriétés stockés dans Neo4J. Plus spécifiquement, cette démonstration se focalise sur l'exemple des requêtes de prédiction de liens.

La figure 2 est un exemple de *template* ou modèle de requête. Comme déjà expliqué, le but des interfaces de EASI-GDS sera de guider les utilisateurs de manière à qu'un *template* comme celui là soit rempli, constituant ainsi une requête déclarative dont exécution sera mise en place par un pipeline. Il convient donc d'expliquer rapidement ce *template* qui utilise le mot-clé "PREDICT". Les utilisateurs peuvent spécifier la classe d'analyse demandée, le type de prédiction (lien, classification, régression) et les paramètres obligatoires. Les paramètres facultatifs sont indiqués à l'aide de "WITH".

5.1 *Template* de prédiction EASI-GDS

Un *template* EASI-GDS présente une collection de configurations pour un pipeline, les configurations pouvant varier en fonction des cas d'utilisation réels. Il prend en considération l'intégralité du pipeline de prédiction, débutant par la projection et l'*embedding* des graphes (une méthode prédéfinie est établie en fonction des requêtes possibles et du domaine d'application).

Le pipeline peut également faire appel à différents modèles tels que la forêt aléatoire, le perceptron multilinéaire et la régression logistique pour prédire les arêtes d'un graphe.

EASI-GDS permet de tester plusieurs modèles pour une intention d'utilisateur donnée et renvoie les résultats triés en fonction de leur performance. Il fournit également des paramètres afin de simplifier le processus pour les utilisateurs qui ne sont pas familiers avec ces modèles. Certains modèles, tels que le perceptron, nécessitent un entraînement. EASI-GDS gère cette tâche en lançant le processus d'entraînement sur un sous-graphe et renvoie le modèle entraîné pour la prédiction souhaitée.

```
MATCH (s:Label1)-(r:Label2)->(t:Label3)
  PREDICT Link (s, t)
MATCH (s:Label1) PREDICT classification s.iris
MATCH (s:Label1) PREDICT regression s.ph
!OPTIONAL!
WITH property[list of prop]
AND algo [{RF},{MLP},..]
AND write AND (embedding=FastRP)
AND results:{TOPK(K),(TOPN(N)) AND name=relName
```

FIG. 2 – *Modèle de requête pour la prédiction sur les bases de données Neo4J*

5.2 Interfaces EASI-GDS

La figure 3 présente la page d'accueil d'EASI-GDS, où l'interface graphique occupe une place centrale, mettant en avant les six modèles (classification et régression pour la prédiction des propriétés des nœuds, prédiction des liens, centralité, détection des communautés et similarité), ainsi que le schéma de la base de données graphes. En cliquant sur l'un des modèles, un nouvel écran (figure 4) offre plusieurs choix aux utilisateurs. Ces derniers ont la possibilité de modifier la valeur de certains paramètres. Cependant, pour un utilisateur non-expert, EASI-GDS calcule le résultat en se basant sur des paramètres prédéfinis.

L'interface conversationnelle est située dans le coin inférieur droit, permettant aux utilisateurs d'envoyer des questions contenant des mots-clés. Tous les messages sont analysés et génèrent les requêtes correspondantes en retour.

Notre **scénario de démonstration** repose sur le graphe *Movies* et met en œuvre un scénario de recommandation de films visant à prédire si un individu apprécierait un film donné. Ainsi, les résultats se présentent sous la forme d'arêtes connectant les individus aux films. Dans la vidéo de démonstration [2], nous illustrons comment EASI-GDS guide les utilisateurs, même non spécialistes de l'apprentissage automatique, à travers la spécification de leurs intentions en utilisant les interfaces graphiques et conversationnelles, et comment celles-ci sont ensuite traduites en pipelines exécutés par Neo4J.

Notre démonstration se concentre sur l'intention de *prédiction*, car il s'agit du cas le plus sophistiqué que nous ayons mis en œuvre, faisant appel à différents modèles. Par conséquent, il s'agit d'un exemple permettant d'illustrer une comparaison entre les modèles. Toutefois, il est important de noter que EASI-GDS traite également d'autres types de pipelines pour la science des données, tels que la centralité, la similarité et la détection de communautés, qui sont plus simples à paramétrer [3].

2. La vidéo de démonstration est accessible à l'adresse suivante : `https://youtu.be/pd1s7hOVMx8`
3. Les instructions pour l'installation d'EASI-GDS sont disponibles à l'adresse suivante : `https://gitlab.com/mirian/easi-gds_install`

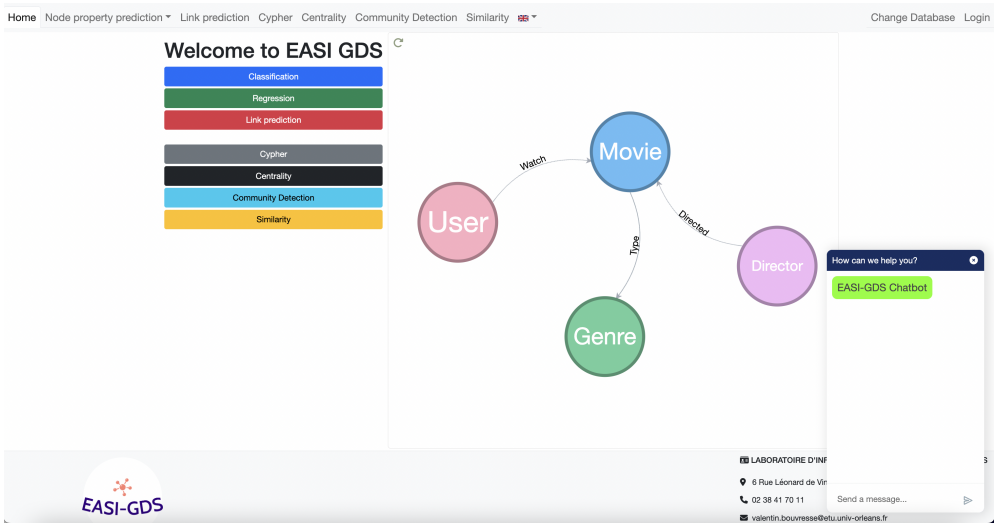

FIG. 3 – *Page d'accueil EASI-GDS*

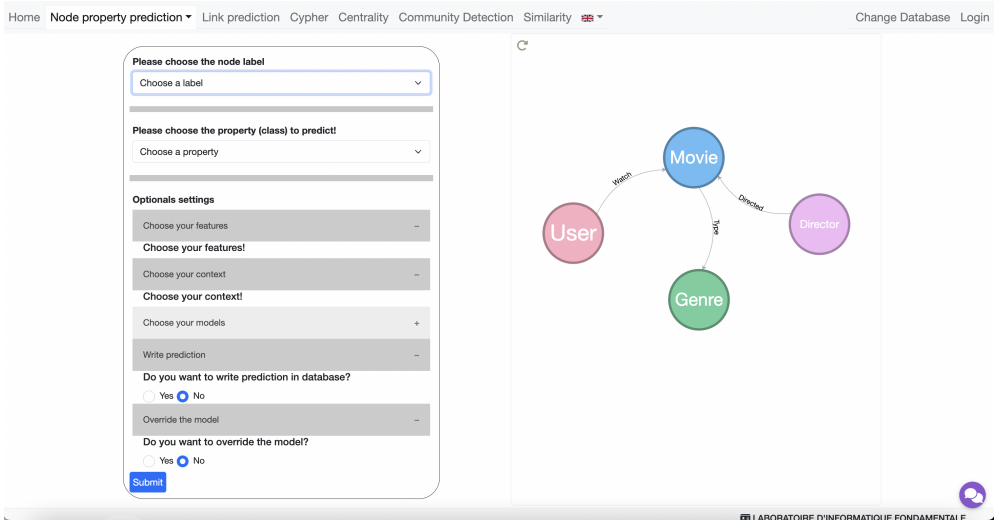

FIG. 4 – *Paramétrage EASI-GDS*.

6 Conclusions et travaux futurs

EASI-GDS est une étape vers la mise en place d'un système d'interrogation déclaratif dédié aux requêtes analytiques sur les bases de données graphiques. En simplifiant l'interaction avec la bibliothèque de science des données de Neo4J, il promet d'enrichir les possibilités d'exploitation pour les utilisateurs.

EASI-GDS est un projet prometteur qui évolue vers un *framework* doté de multiples caractéristiques, telles que la visualisation de divers algorithmes d'*embedding*, l'amélioration de l'expérience conversationnelle grâce à l'exploitation de ChatGPT, et l'évolution de la version actuelle pour passer d'une machine locale mono-utilisateur à un environnement multi-utilisateur plus robuste. Nous envisageons également l'intégration de toutes les fonctionnalités de la bibliothèque GDS. Bien que son implémentation actuelle soit spécifique à Neo4J, EASI-GDS est conçu pour être adaptable à d'autres bases de données graphes à l'avenir. Cette vision pour EASI-GDS s'avère à la fois passionnante et prometteuse !

Références

Bethaz, P., K. Belhajjame, G. Vargas-Solar, et T. Cerquitelli (2021). DS4ALL : all you need for democratizing data exploration and analysis. In *IEEE BigData*, pp. 4235–4242. IEEE.

Sarker, I. H. (2021). Data science and analytics : An overview from data-driven smart computing, decision-making and applications perspective. *SN Comput. Sci. 2*(5), 377.

Vargas-Solar, G., K. Dao, et J. A. Espinosa-Oviedo (2023). Translating data science queries from natural language into graph analytics queries using NLDS-QL. In *EDBT/ICDT Workshops*, Volume 3379 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Vargas-Solar, G., P. Marrec, et M. Halfeld Ferrari Alves (2021). Comparing graph data science libraries for querying and analysing datasets : Towards data science queries on graphs. In *ICSOC Workshops*, Volume 13236 of *Lecture Notes in Computer Science*, pp. 205–216. Springer.

Vassiliadis, P., P. Marcel, et S. Rizzi (2019). Beyond roll-up's and drill-down's : An intentional analytics model to reinvent OLAP. *Inf. Syst. 85*, 68–91.

Summary

This article presents a user-friendly tool which helps to simplify the parameterization of various models in Neo4J GDS. Designed specifically for non-expert users, this tool offers graphical and conversational interfaces, eliminating the need to fill in a complex model in Neo4J. It facilitates the analysis of graph data and compares model results. The best result is automatically returned as a graph or a table. When the result is presented as a graph, users can perform additional operations to explore more information.

In the future, we plan to further enhance this tool with features such as result reuse. Our aim is to enable more users to engage in data analysis tasks, promoting accessibility in the field of data science.