VizML x KG4Viz

Final Report of the Capstone Project

João Miguel Peixoto Lamas
Pedro Afonso Nunes Fernandes
Tiago de Pinho Bastos de Oliveira Pinheiro
Tiago Grilo Ribeiro Rocha

Bachelor in Informatics and Computer Engineering

Supervised by: Alexandre Miguel Barbosa Valle de Carvalho

March 12, 2025

Contents

1 Introduction

This report presents the development and analysis of a capstone project focused on VizML and KG4Viz, two different approaches for visualization recomendation. The project was developed in an academic setting, with the goal of exploring and comparing these two methodologies for recommending data visualizations.

The main motivation was to analyze and compare these methods and their efficiency in recommending visual encodings.

1.1 Objectives and expected results

The primary objectives of this project were to analyze and compare the VizML and KG4Vis approaches for visualization recommendation, to understand the strengths and limitations of each method, particularly in terms of accuracy, explainability, scalability, and computational costs and to successfully run both models using the provided dataset and evaluate their performance.

The expected results include a detailed comparison of the two approaches, insights into their computational requirements, and a working implementation of both models.

1.2 Report structure

There are four sections in this report. An overview of the project, including its goals and anticipated outcomes, is given in Section 1. The approach, stakeholders, and activities established are all covered in Section 2, which also details the primary actions completed during the project. In Section 3, the solution's development—including its architecture, technology, and requirements—is presented, along with the solution's validation. Section 4 wraps up the study by summarizing the outcomes attained, the lessons discovered, and recommendations for further research.

2 Methodology and Development Process

2.1 Methodology used

The project followed an iterative development approach, with weekly meetings to discuss progress, analyze findings, evaluate the results and plan next steps. Thanks to this process, as the project developed, we were able to get feedback and make the required changes.

Key resources used during the project included GitHub for collaboration on the report, PyTorch for neural network training, and TransE embeddings for the KG4Vis model. The team also relied on the original papers for VizML and KG4Vis to guide the implementation and analysis.

2.2 Stakeholders, roles and responsibilities

There were several participants in this project, and each had different roles and responsibilities contributing to the project's success::

- **Project Coordinator and Tutor** - Professor at Faculdade de Engenharia da Universidade do Porto Alexandre Miguel Barbosa Valle de Carvalho, responsible for guiding the project, providing feedback, and evaluating the final results.
- **Team Members**

  - João Miguel Peixoto Lamas (up202208948) -

  - Pedro Afonso Nunes Fernandes (up202207987) -

  - Tiago de Pinho Bastos de Oliveira Pinheiro (up202207890) -

  - Tiago Grilo Ribeiro Rocha (202206232) -

2.3 Activities developed

The project activities were carried out over several weeks, with the following key milestones:

- **First Meeting (13/02/25)** - Initial discussion of the VizML paper and verification of the dataset.
- **Second Meeting (20/02/25)**: Analysis of the VizML and KG4Vis papers making explanatory diagrams for each one, dataset characterization, and proceed to do the software installation.
- **Third Meeting (27/02/25)**: Exploration of neural network training and embedding learning, start running the dataset.
- **Fourth Meeting (06/03/25)**: Deep dive into scalability challenges, computational costs, and preparation for model training.

3 Solution development

3.1 Overview of VizML and KG4Vis

3.1.1 VizML Overview

3.1.2 KG4Vis Overview

3.2 Dataset Analysis

3.2.1 Dataset Collection

3.2.2 Dataset Structure

3.2.3 Dataset Execution

3.3 Methodology

3.3.1 VizML Methodology

3.3.2 KG4Vis Methodology

3.4 Technical Implementation

3.4.1 Neural Network Training (VizML)

3.4.2 Embedding Learning (KG4Vis)

3.4.3 Computational Costs

3.5 Visualization Recommendation Process

3.5.1 VizML's Direct Prediction

3.5.2 KG4Vis' Explanation-Based Recommendations

3.6 Comparative Analysis

3.6.1 Core Approach

3.6.2 Explainability

3.6.3 Performance

3.6.4 Extensibility

3.6.5 Scalability

3.7 Technology Requirements

3.7.1 VizML Technology Stack

3.7.2 KG4Vis Technology Stack

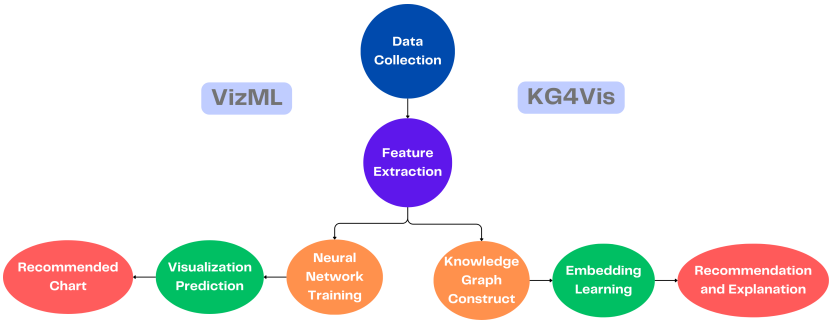
3.8 Execution of Datasets

3.8.1 VizML Dataset Execution

3.8.2 KG4Vis Dataset Execution

The architecture used in this project involved two main components: the VizML and KG4Vis models. VizML is a machine learning-based approach using a fully-

connected feedforward neural network with three hidden layers, while KG4Vis is a knowledge graph-based approach using TransE embeddings to model relationships between dataset features and visualization choices.



Both models rely on a large dataset of dataset-visualization pairs, and both typically get their data from Plotly (KG4Vis getting it from VizML which gets it from Plotly).

When extracting, the goal is to quantify dataset characteristics to help in selecting the right representation, with VizML extracting numerical features, like statistical properties (mean, variance, etc) and structural attributes (number of columns, missing values, etc), and KG4Vis encoding the features as nodes in a knowledge graph.

VizML

VizML uses a supervised deep learning model, which means it learns from labeled dataset-visualization pairs. It processes 841 extracted features per dataset and gives them to a fully connected neural network. The goal is for the network to map dataset features to the best-fitting visualization type.

As mentioned before, VizML's makes use of pytorch to turn dataset features into sensors, which then allows the network to learn through minimizing prediction error, by adjusting weights through backpropagation. This process is done in small batches as to improve efficiency, but even so it's computationally expensive, often requiring larger RAM and VRAM capacity and GPU acceleration, for example, to handle large datasets.

Once the model has finished its training and learning, it can predict the most suitable visualization type in real time using a dataset's feature vector for a single forward pass through the network, thus having a fast inference time. This, however, reduces its explainability, as it quickly gives an answer, but does not justify or elaborate it.

KG4ViS

KG4Vis builds knowledge graphs with the extracted features, with the nodes representing the dataset attributes, visualization types and encoding choices and the edges defining meaningful relations, for example, categorical data being best represented by a bar chart ("Categorical data ——— Bar Char"). This makes it dynamically expandable, meaning it can receive and integrate new dataset features and/or visualization types without retraining and starting again from

the beginning. It also helps with interpretability, since all relations are explicitly defined.

KG4Vis learns vectors representations (embeddings) for each node and relation in the knowledge graph. It makes use of TransE embeddings, which dictate that if A and B are related by R, then in vector space A + R B. Pytorch is used in this case to help train these embeddings, as mentioned before, by helping computing and updating the knowledge graphs, so the model can predict possible missing links from it. The embedding space preserves relations between data and visualization types, which helps with explainability.

KG4Vis, instead of directly predicting, retrieves instead the most relevant visualization based on its learned knowledge graph structure post training. Due to the nature of the graph, it can then trace back the recommendation, allowing it to explain why that particular visualization is suitable, offering better explainability.

Present the developed solution from the user's point of view, with the help of screenshots.

Description of the validation of the developed solution (e.g. experimental evaluation results, tests carried out, feedback from users or experts, etc.)

4 Conclusions

4.1 Achieved results

Summarize the results achieved and contributions (in relation to the objectives).

4.2 Distribution of work

In the case of group work, clarify the individual contributions, in qualitative and quantitative terms (percentage).

4.3 Lessons learned

Reflect on the lessons learned (taking into account the learning objectives).

4.4 Future work

Ideas for improvement and future work.

5 Glossary

- **AAA** - BBBBBBBBBBBBBBBBBBBB

References