**Personalized Context-based Job Recommendation System based on Knowledge Graph**

Shivam NK Mishra - 1186614 | Rutvik Patel - 1191668 | Smit Mirani - 1188571

Nikhil Ailani - 1194456 | Ziyang Ren - 1145033

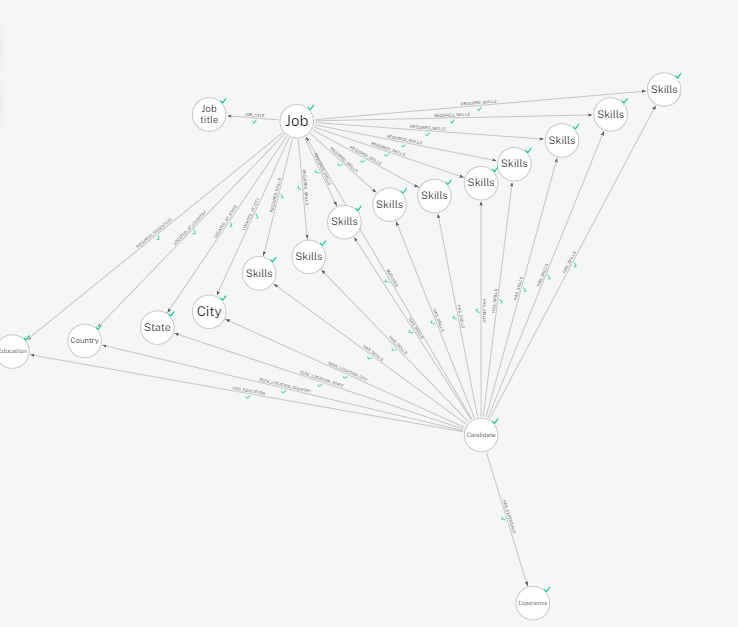
Department of Computer Science

Lakehead University

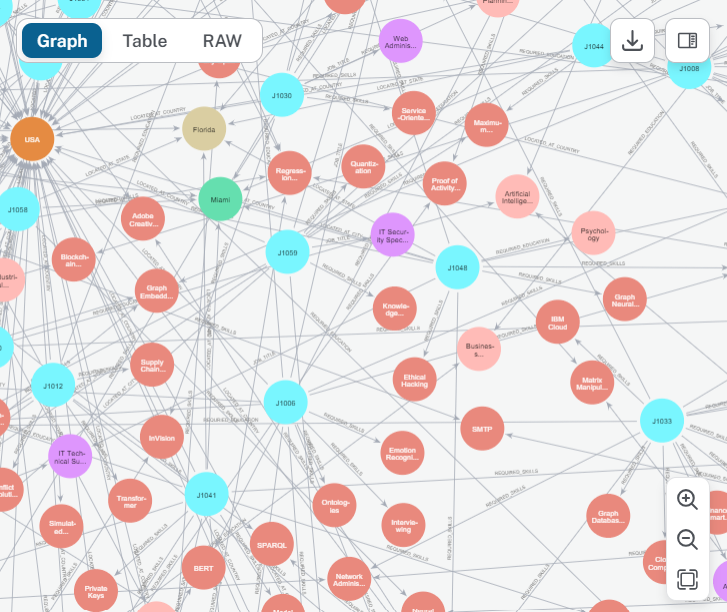
**Introduction:**

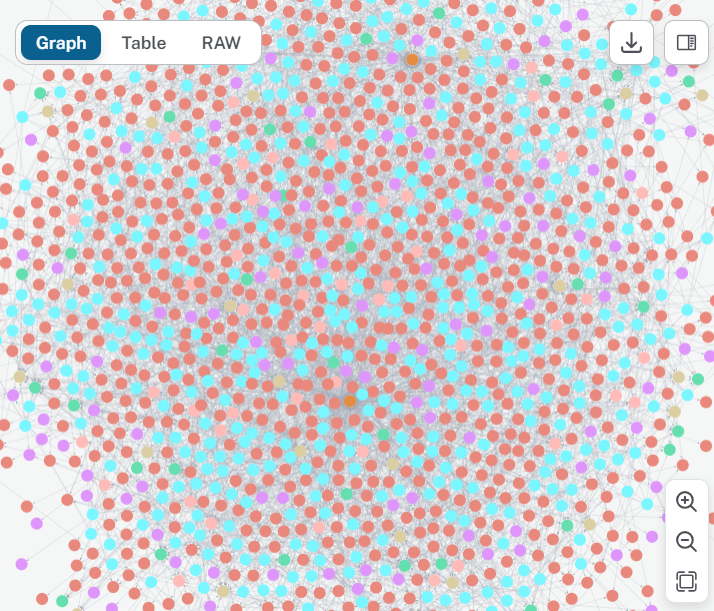
This project introduces a Context-Aware Personalized Job Recommendation System leveraging a Knowledge Graph powered by the robust Neo4j database. The knowledge graph captures intricate relationships between job and candidate entities, with primary nodes represented by "Job" and "Candidate." Additional contextual details such as skills, location, experience, and education are integrated as minor nodes. The system incorporates advanced machine learning techniques, employing the GraphSage algorithm for effective link prediction in the heterogeneous graph. Furthermore, HinSAGE layers are utilized for efficient sampling and training, enhancing the model's ability to capture diverse graph structures. The integration of TensorFlow and Keras facilitates the training process, while evaluation metrics such as mean squared error and mean absolute error provide insights into the model's performance. This dynamic approach promises to revolutionize job matching, offering a personalized and context-aware system tailored to the complexities of the modern job market.

**Database:**



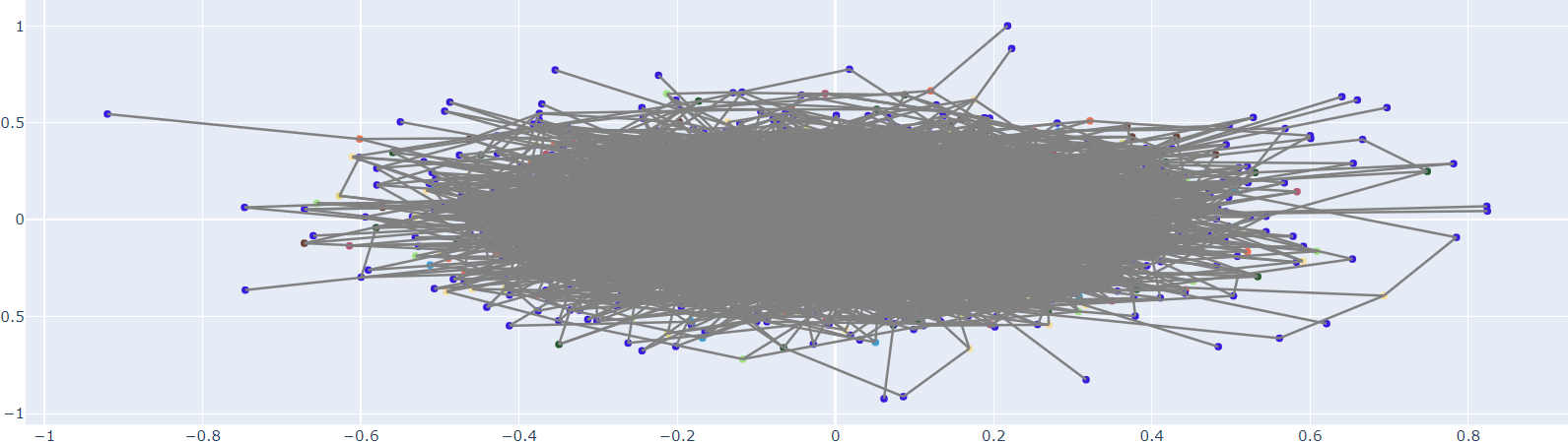
This diagram illustrates the process of loading a dataset into a Neo4j graph database, focusing on representing information related to job and candidate entities. The central nodes in this graph are the "Job" node and the "Candidate" node, serving as the primary entities. Other contextual details, such as skills, location, experience, and education, are considered secondary nodes, establishing connections to the two main entities. The graph comprehensively captures the interrelationships between jobs and candidates, encompassing both the attributes they are seeking and the qualifications they possess. This approach ensures a realistic representation of the job-candidate ecosystem, facilitating effective querying and analysis within the Neo4j graph database.





This is the representation of the whole knowledge graph.

**Community Detection Model:**



Model - The community detection code in our model encompasses a workflow involving graph data extraction, graph neural network (GNN) modeling, community detection, and community assignment update in a Neo4j graph database. Initially, it fetches graph data from a Neo4j database using a custom GraphDataExtractor class, which establishes a connection, executes Cypher queries to retrieve node and edge information, and then closes the connection. Subsequently, a GNN model (Graph Convolutional Network) is defined and initialized using the PyTorch Geometric library. Dummy graph data is generated, including node features, edge indices, and binary labels. The GNN model is then applied to the graph data to obtain embeddings, which are clustered using KMeans to detect communities. These community assignments are updated in the Neo4j database by iterating over nodes and executing Cypher queries to set their community property. Overall, this workflow showcases the end-to-end process of extracting, modelling, analyzing, and updating graph data, demonstrating the integration of machine learning techniques with graph databases for community detection and analysis.

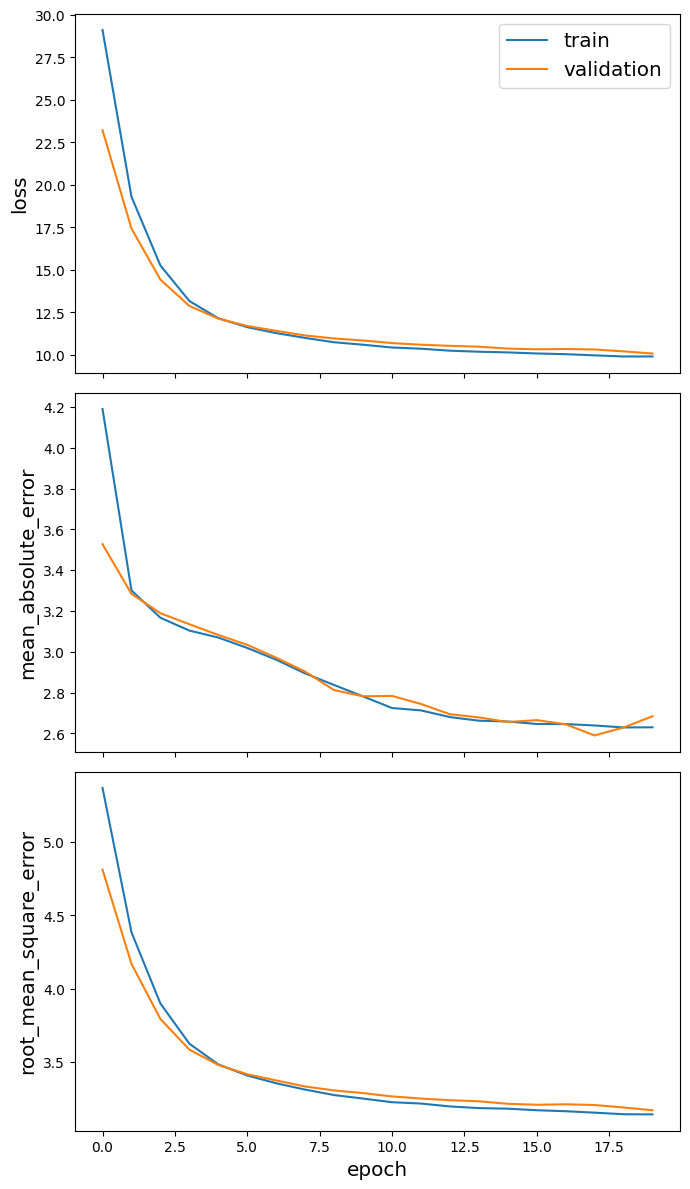
The result obtained - The result involves the community detection process, where node embeddings generated by the graph neural network (GNN) model are clustered using the KMeans algorithm. Each node is assigned to a community based on the cluster it belongs to, indicating its structural similarity to other nodes within the same community. Subsequently, the updated community assignments are written back to the Neo4j graph database, associating each node with its respective community. This process facilitates the organization and analysis of the graph data by identifying cohesive groups of nodes, enabling insights into the underlying structure and relationships within the graph.

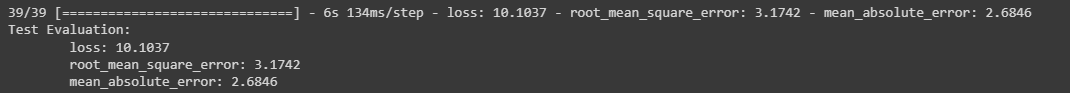
**GraphSage Recommendation Model:**

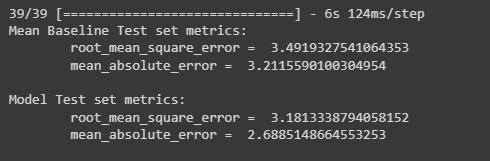
This modelling part focuses on link prediction in a heterogeneous graph using the GraphSage algorithm. GraphSage is chosen for its effectiveness in capturing structural information from diverse node and edge types. The project involves the preprocessing of graph data, splitting it into training and testing sets, and utilising a HinSAGE layer for efficient link prediction. The neural network model, implemented with TensorFlow and Keras, undergoes training and evaluation, with key metrics such as loss, mean squared error, and mean absolute error monitored during the process.

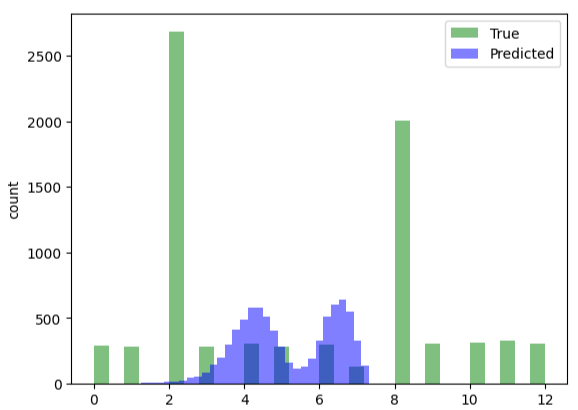
* Establishes a connection to a Neo4j graph database to extract graph data.
* Utilises Stellargraph for graph representation and employs a GraphSage variant, HinSAGE, for link prediction.
* Performs data preprocessing by splitting edges into training and testing sets.
* Defines a neural network model with a regression layer for link prediction.
* Compiles the model with Adam optimizer and mean squared error loss.
* Evaluates the untrained model on the test set, providing initial performance insights.
* Trains the model, tracking the loss curve and key metrics over epochs.

Upon training the model, the obtained results are visualised in three graphs. The first graph illustrates the loss curve, showcasing a decreasing trend, indicative of the model's learning process. The following two images depict metrics such as mean absolute error and root mean square error. These visual representations provide insights into the model's performance during training, offering valuable information about its convergence and predictive accuracy.









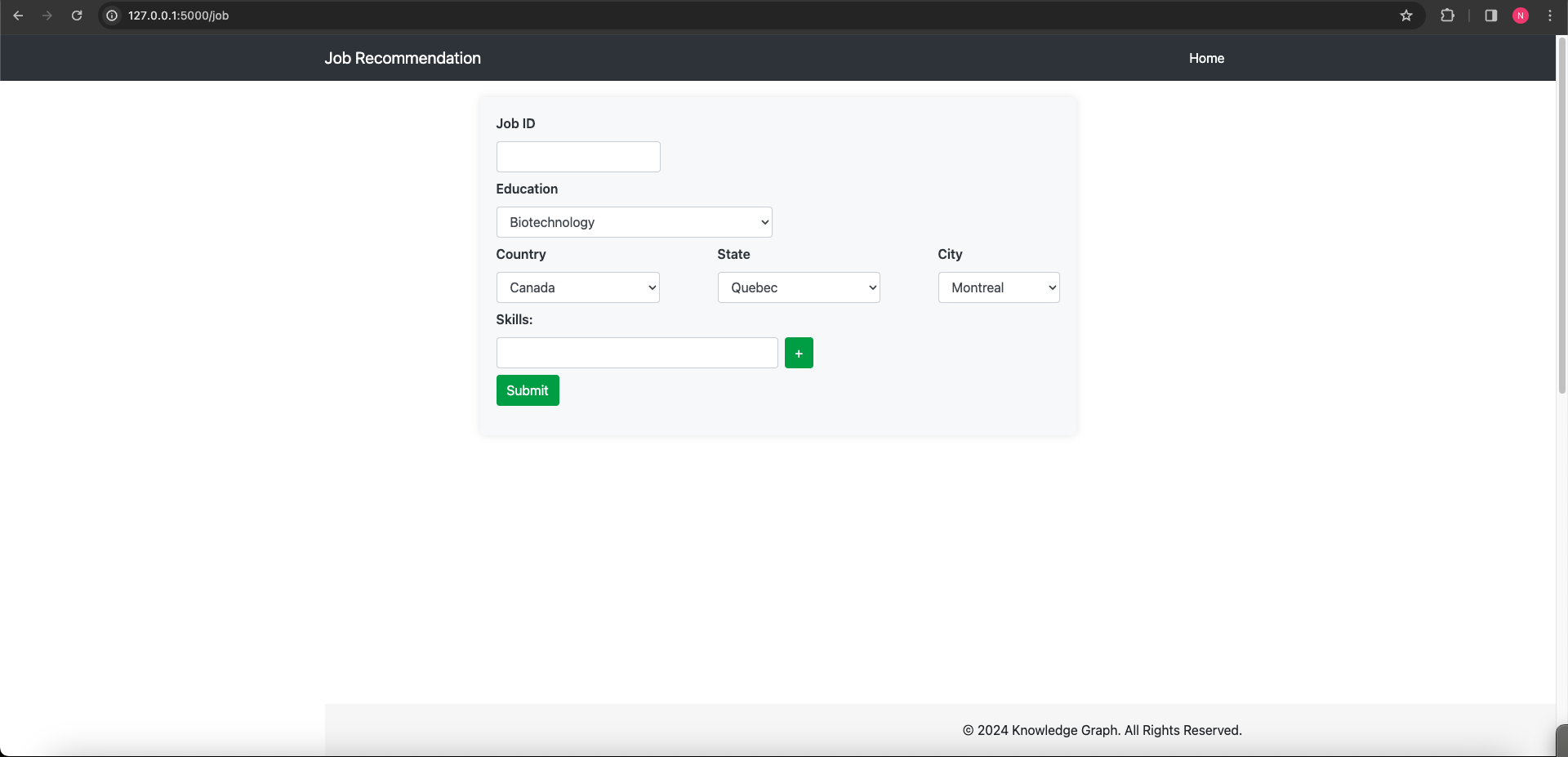
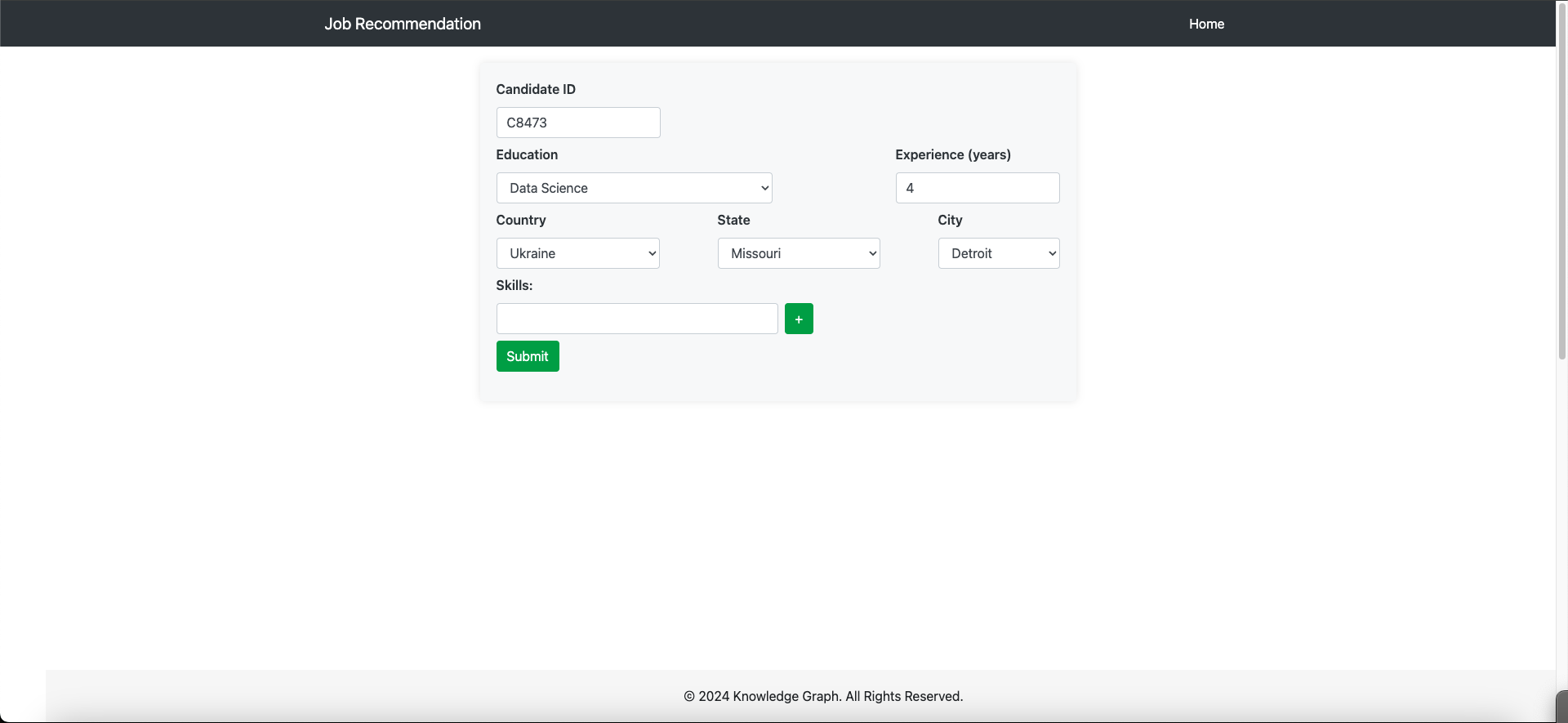
Here is the overall pattern of True nodes value and predicted nodes values. For testing data.

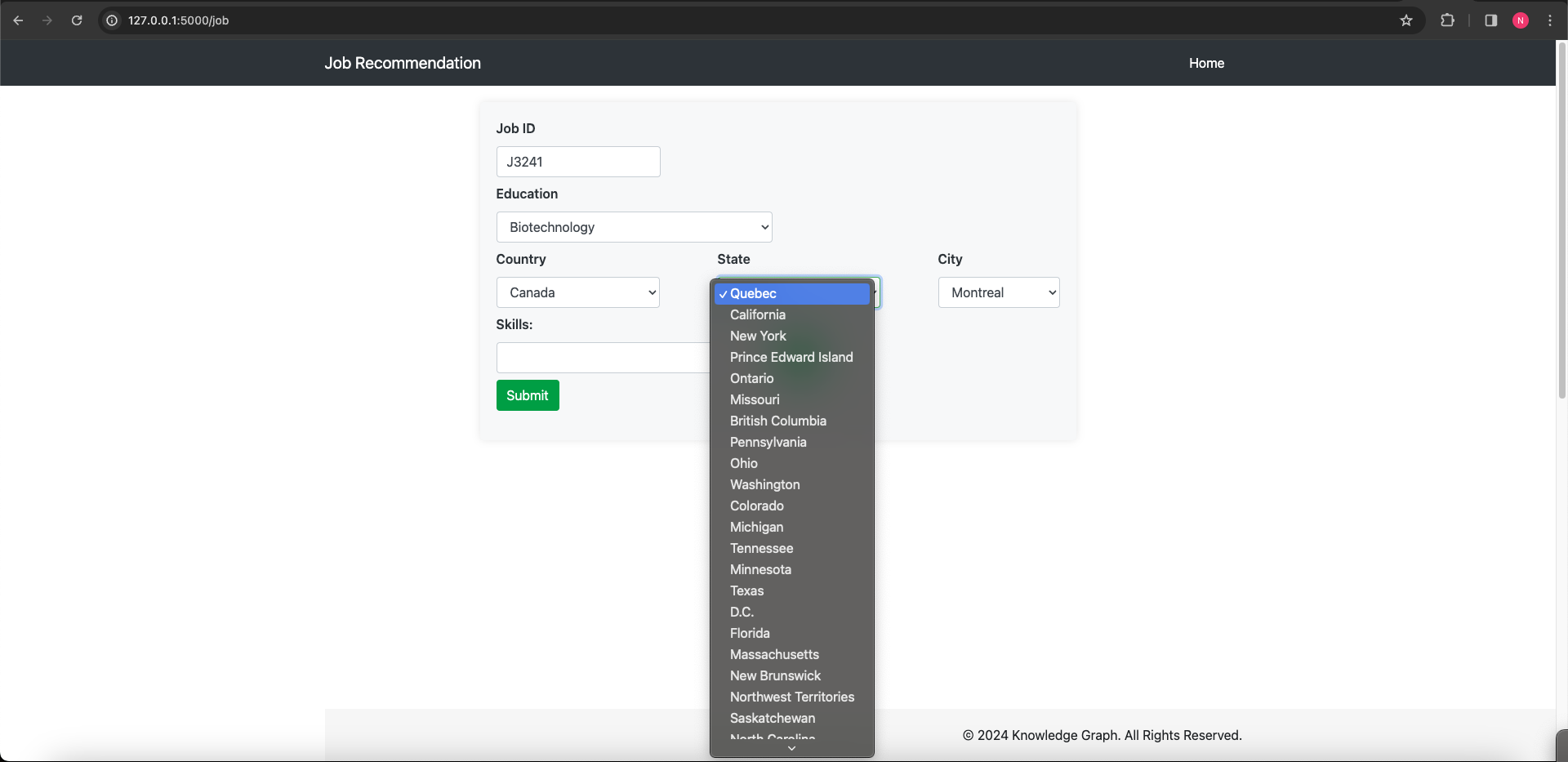
**Scenarios of Use -**

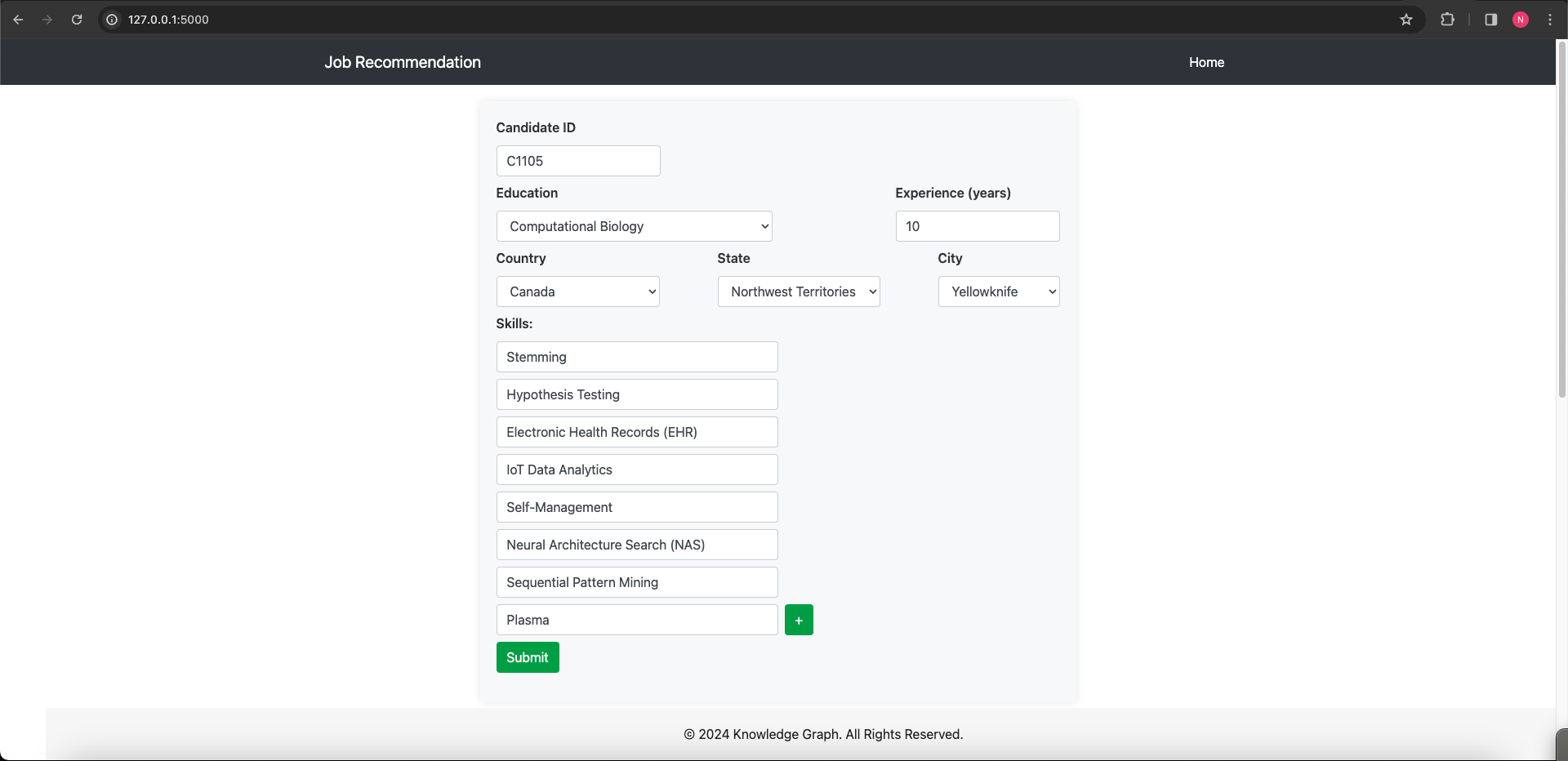
The user will input their information, including skills and experiences, followed by their qualifications and education. Upon submitting the details, they will be transmitted to the knowledge graph, creating a specific node representing the test customer. This node will serve as input for the recommendation model. The model will then utilize this node as testing data to generate five job recommendations tailored to the user's profile. Finally, the recommendations will be promptly sent to the user as output, facilitating informed decisions regarding potential job opportunities.

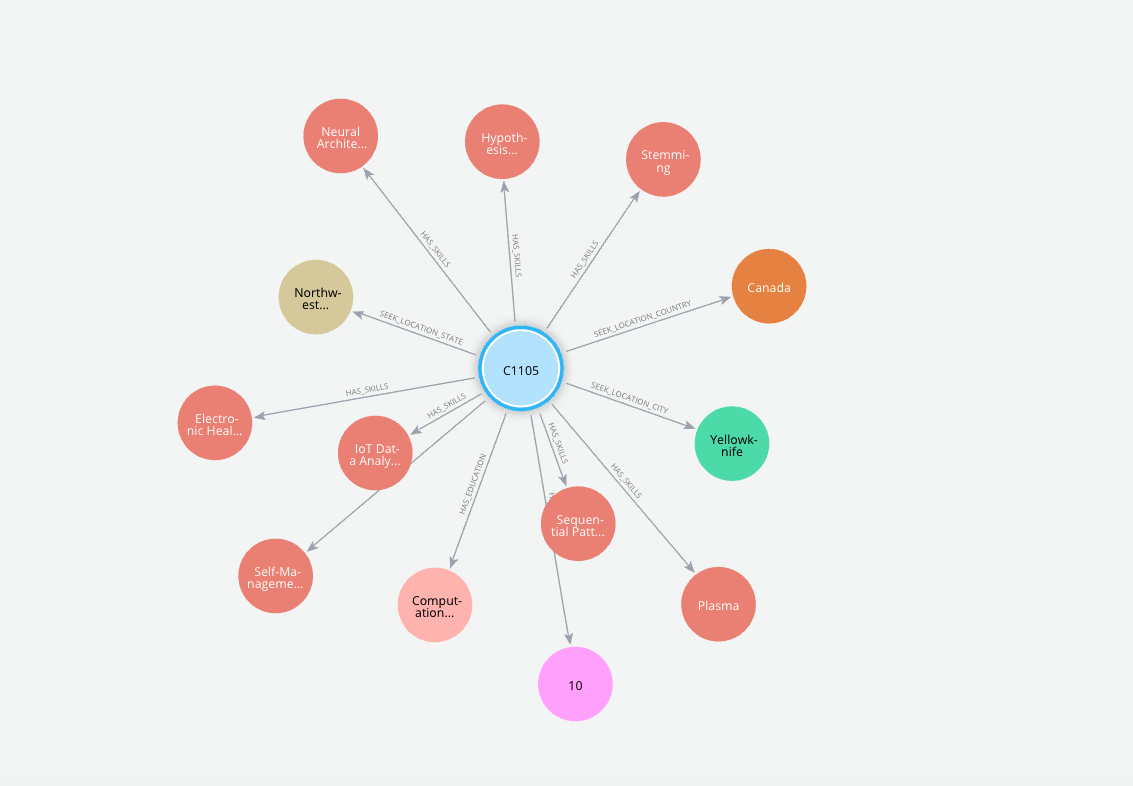
1. Candidate Profile and Job Posting

Use Case: A candidate inputs personal information, including skills, experiences, qualifications, and education. For a job, the education required, location of the job and skills required are added.

Output: The system processes the candidate's data, creates a specific node in the knowledge graph, and utilizes this node as input for the recommendation model. The job posting page takes information related to the job posting and creates nodes and relationships in the knowledge graph for the same.The country, states and cities are fetched dynamically from the knowledge graph database as a choice for jobs and users. This adds new entries in the database for the user to have their information injected to find suitable jobs according to their interests, qualifications and skills.



Let’s take an example of creating a user with Candidate ID C1105, with some education and experience and some skillset. 

You can see in the below figure the entry added in the application reflects in the neo4j database visually from the neo4j console.

2. Job Recommendations Generation

Use Case: The recommendation model utilizes the candidate's node in the knowledge graph to predict new output.

Output: The model generates five job recommendations tailored to the candidate's profile, leveraging GraphSage and the knowledge graph structure.

3. Output Delivery to candidate

Use Case: The personalized job recommendations are sent promptly to the candidate.

Output: Users receive informed decisions regarding potential job opportunities, enhancing their job search experience.