Name: Park, Hannah, Zhou, George

NYU ID: N18541255, N18212178

Date: 8/11/2023

Course Section Number: Summer 2023 Database Systems Section 001

Database Systems Project Part 4 - End-to-End Solution Integration and Data-Driven/Database Programming

Queries and Python codes can be found at:
git@github.com:gPwls1025/Database_Systems_Summer_2023.git

In the final part of the project, we focus on implementing a comprehensive end-to-end business case solution that builds upon the optimized OLTP/ODS relational database and leverages insights from a machine learning model applied to data collected. In order to enhance operational efficiency, we aim for facilitating continuous updates to data and minimizing human intervention in managing the data pipeline.

Business Use Case: Insurance Quote Generation

To meet the project's objectives, our initial step was to select a business use case: the development of an insurance quote generation tool. This tool is designed to serve potential customers seeking an estimated insurance price. By facilitating input of user information such as demographic data, a brief overview of their health condition, medical history and prior insurance coverage, the tool employs these inputs to compute a personalized insurance quote tailored to the individual's circumstances.

Building on the work we did previously, we decided to further develop the machine learning model we created in part 3 as a starting point. In part 3, we made a linear regression model that predicts insurance premiums and the model was stored and trained using raw data in BigQuery. In part 4, we took this trained model to calculate the insurance premium based on data entered by potential customers in real time. A frontend web application in which customers can input their personal and health information was designed using a third party frontend platform called "Retool" [1]. The interface not only prompted customers for necessary input but also enabled integration with BigQuery databases, allowing real-time data updates in our database.

To develop the frontend tool, our initial step involved integration of the web application with our BigQuery database. By establishing connection credentials within the Google Cloud project, the integration provided us with access to our BigQuery database through the Retool platform. Leveraging Retool's capabilities for web application development, we crafted an intuitive web application interface (Figure 1) to enable user data input.

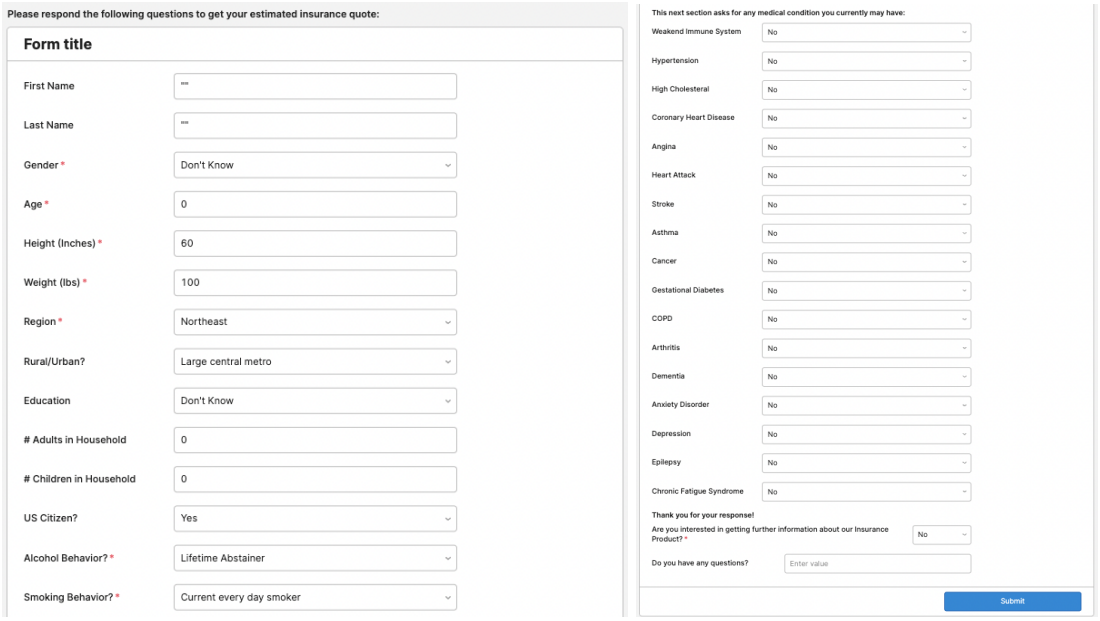

Figure 1. Web application form for data entry

Within this interface, we designed a table that returns personalized insurance quote details based on the processed input data. To enact such functionality, the *form1SubmitHandler.sql* script was used to execute the linear regression model upon user data submission. Furthermore, the *returnQuote.sql* script was executed to retrieve the calculated insurance quote tailored to each individual customer (Figure 2). *The storeNewData.sql* script was used to preserve user data for individuals who demonstrated an interest to acquire further information about insurance products. This query filtered users who had not indicated the interest and selectively archived the input data into the BigQuery database. For deployment of the program, we assumed that a deployment link (https://gz2214.retool.com/apps/586d1c1a-3ad9-11ee-8905-ef7fef67f62f/insurance-price) will be provided in the insurance company's website for potential customer's use.

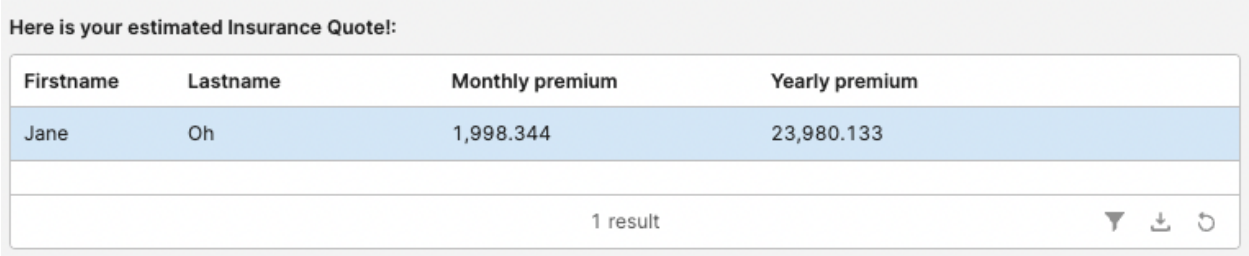

Figure 2. Table returning an individualized insurance quote.

Seamless process of collecting input data from potential customers, processing these submissions in real time and leading the data to organized storage within the database was orchestrated. Such strategic steps have given rise to a data-driven program module that efficiently manages the power of dynamic data streams with the least amount of human intervention with data. A crucial part of this module is adding newly submitted data to the database due to two main purposes. Firstly, the incoming data augment the pool of information used to train our machine learning model, enhancing its predictive capabilities. This re-training process involves carefully matching the variable names and types between the user input form and the database. Once newly acquired data finds its repository in the designated "new_data" table within the database, it gets joined with previously collected data that is stored in another table. The merged table sets the stage for the subsequent execution of the regression model. The regression model performs calculations to make predictions in this combined dataset. This coordinated process highlights how the evolving user inputs and the model's predictive abilities work together effectively. Moreover, by successfully implementing a workflow-based application that integrates with a database using a data-driven program module, the module demonstrates its ability to handle end-to-end data flows.

As our primary focus was on developing an insurance quote generation tool - which required largely semi-structured data - as our business use case solution, the necessity to gather unstructured data did not arise in this context. However, when the need to collect or manage unstructured data arises, our system is well-equipped to handle it seamlessly. Retool not only offers convenient options for uploading files and images and formatting stored data into JSON files using either SQL or Javascript, but it also benefits from the established connection with BigQuery. This integration ensures that a solid connection is in place, enabling us to effectively handle unstructured data

The implementation of our solution aligns seamlessly with the stipulated project requirements outlined in part 3. Our solution allows front-end users to complete a user friendly form, and after submission, users are provided with an estimated premium quote immediately. The new data is stored into our database (if users indicated they are interested in new insurance), and the premium is calculated by executing a model in the database. This allows for an end-to-end experience for the users, from input to personalized output.. Only data pertinent to

user experience is asked to be completed on the form, making our solution as optimized as possible.

We have built our solution with careful attention to how to handle data. Each piece of information is handled carefully, making sure it's accurate and important. As mentioned previously, data is only stored one row at a time if the user indicates if they are interested in insurance. This allows for only relevant users beneficial in this business case to be in our database. Another significant consideration for data governance is upholding data integrity. To ensure the accuracy and consistency of the provided information, validations were added to questions asked in web application form. For example, only texts were allowed in some questions (e.g. first name, last name, etc.) and only numbers were allowed in others (e.g. age, weight, height, etc.), safeguarding against potential inaccuracies (Figure 3). Furthermore, unique ids were assigned to each form stored in the database to prevent duplicate responses. For instance, "health_condition" table that stores health condition related data has *health_condition_Id* and the "lifestyle" table with lifestyle related data has *lifestyle_Id*. These IDs are separated from general "ID" that sets individuals apart in the broader context. On the front-end, *IDgeneration.js* script was employed to generate unique IDs for new rows getting added to the database. Such identification layers effectively eliminate the possibility of duplicate entries in the database.

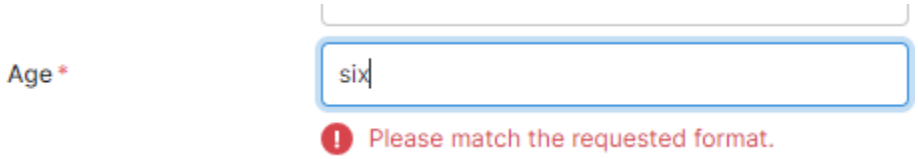

Figure 3. Age must be entered as digits

In conclusion, our project has successfully integrated an optimized OLTP/ODS relational database with a machine learning model to create an end-to-end solution for a business case of an insurance company. The data-driven approach ensures continuous updates and minimal human intervention in the data pipeline, while data governance principles maintain consistency. The workflow-based application processes user inputs in real time, calculates personalized quotes and stores data efficiently. It is important to recognize that the development of the insurance quote tool is just one aspect of business use cases as in real world scenarios there are many more possibilities to explore and implement. However, understanding how the data-driven approach operates within a database will significantly help us in tackling future projects.

References

[1] *Build Any Business Software, Remarkably Fast.* https://retool.com/. Accessed 16 Aug. 2023.

[2] ChatGPT, identifying and correcting errors. Used from August 11, 2023 to August 17, 2023. Retrieved from https://openai.com/chatgpt/