Name: Park, Hannah, Zhou, George

Date: 8/1/2023

NYU ID: N18541255, N18212178

Course Section Number: Summer 2023 Database Systems Section 001

Database Systems Project Part 3 - Logical Schema Optimization and Machine Learning Model Creation

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

In today's business world, companies rely on traditional databases but are also seeking real-time insights from unstructured data for better decisions. Our project focuses on these needs, especially in an insurance company. From the previous parts of the project, we developed a logical database model and schema for an insurance company. As we move into Part 3, our project has now shifted its attention to building an actual database, optimization of the database and machine learning model implementation. We aim to see such implementations hold the promise of optimizing operation efficiency and refining the decision making process within the insurance company.

EDA Physical Database Design

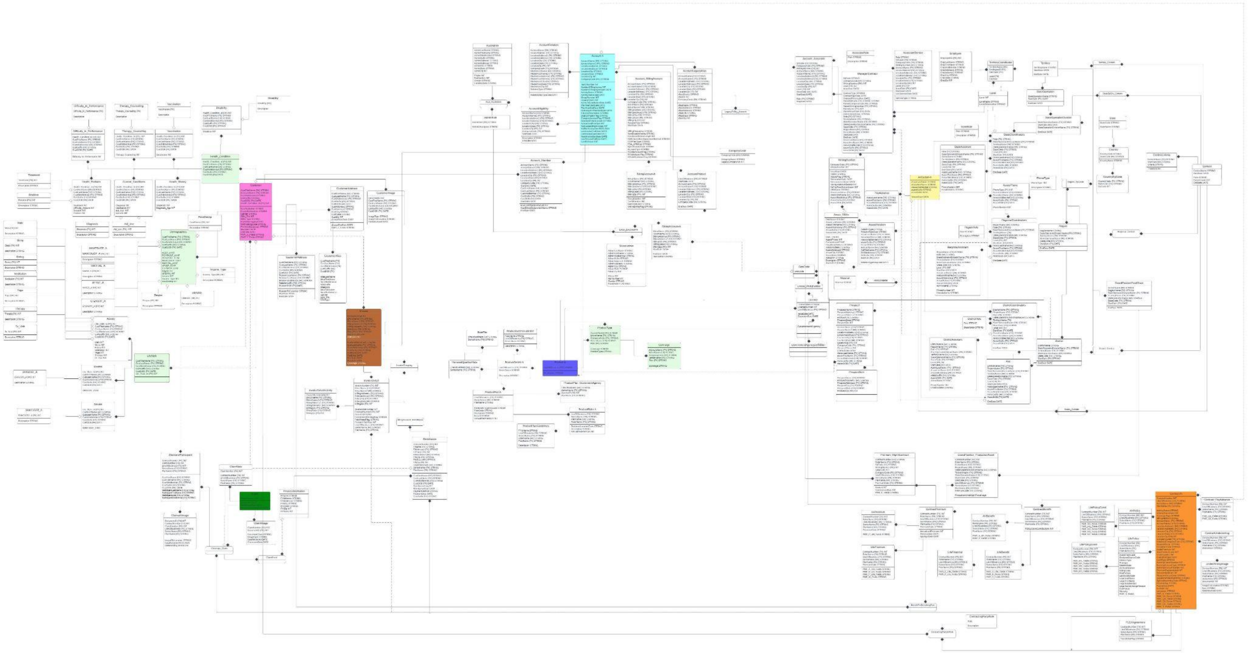

Figure 1. Logical Database Schema

For part 3 of the project, we first focused on optimizing the logical schema (Figure 1) by translating it into a robust physical database design. Initially, we embarked on the path of storing our raw data in AWS. However, due to our data usage surpassing their allocated free tier limits, we made a shift and chose BigQuery as our preferred database system. The exploratory data analysis (EDA) phase began with cleansing raw data and identifying initial patterns. After storing raw, unstructured dataset into the cloud platform, we developed SQL queries to create a set of tables mirroring our logical (RDBMS) schema. Techniques such as indexing and clustering were employed depending on their relevance to our data structure and anticipated query patterns. We developed supplementary queries that explained what each value option means to understand categorical variable responses, enhancing our data comprehension.

In order to identify appropriate business use cases, we undertook a systematic analysis to develop a workflow-based application catering to database users seeking insurance quotes and policies. Here are the two business uses cases we created:

Business use case 1: Obtaining an Insurance Cost by Database User

In the scenario where a database user (presumably an insurance company employee) is seeking to obtain an insurance quote for existing customers, the workflow is tailored to the interaction between the user and the insurance applications system. The database user initiates the process by logging into the application's backend and accessing the quote generation module. Then they input the customer's ID which will retrieve the customer's demographic information and health condition data which encompasses relevant medical history and current health status. The application system, utilizing the retrieved data, calculates an insurance cost that will be charged to the customer based on the linear regression algorithm model. The calculated quote will be presented to the insurance company employee who can review the breakdown of factors contributing to the premium. If needed, the employee can modify certain parameters to see how they can affect the quote in order to provide a dynamic insight into pricing variations.

Business use case 2: Dynamic Pricing based on Disease Data

The concept of dynamic pricing based on disease data introduces a responsive approach to insurance product costs. The workflow begins with the collection of comprehensive disease related data as well as customer-specific data, including lifestyle factors such as exercise habits, smoking and alcohol use, and other relevant attributes. Using correlation analysis and decision tree algorithms, the insurance application's backend system analyzes the data and identifies patterns and correlation between diseases, customer characteristics, and actual insurance costs paid. As the system continuously updates with new data, it will change the prices of insurance plans to show how different factors like individual profiles, disease risks, and past cost information work together. For instance, if data indicates that customers with certain lifestyle tend to present higher claims due to specific diseases, the system can strategically adjust the pricing of insurance products for those individuals.

Machine Learning Model Creation

In order to create an effective machine learning model that harnesses insights from the unstructured data, a structured series of steps were performed. These steps include the development of both linear regression and decision tree models to facilitate informed business decisions. For the machine learning model development, we decided to use Vertex Ai workbench which allows integration of BigQuery queries. We began with a refinement of the previously identified business use cases.

We leveraged the power of BigQuery to retrieve tables generated from our raw dataset. For the preprocessing, we handled missing values and encoded categorical variables. Also, due to the nature of our original dataset (sourced from the Center for Disease Control and Prevention (CDC) website, specifically the National Health Interview Surveys (NHIS), Sample Adult Interview[1] which incorporated people's data of both who have public insurance and private insurance, we excluded data entries related to individuals with public insurance as they are not subject to insurance premium charges. Moreover, due to the presence of a long-tailed distribution in the insurance premium data, we applied a logarithmic transformation to the "premium", which serves as our target variable. The data preprocessing resulted in approximately 12,000 data entries.

The linear regression model was created for business use case 1: computing insurance cost for customers. In this scenario, database users - this could potentially be insurance company employees - interact with the application's backend to get insurance costs that were calculated based on customer-specific data. The linear regression algorithm computes the insurance cost based on a range of attributes encompassing demographic, health, and lifestyle factors. Using the preprocessed data, we splitted the dataset into training and test sets with 8:2 ratio. Categorical variables selected as feature inputs were converted into categorical data types to optimize model processing efficiency. However, upon model evaluation, the achieved R-squared value of 0.017 proved to be relatively low. The Spearman rank correlation coefficient of 0.15 highlighted a relatively weak positive monotonic relationship among the variables under consideration. In an effort to improve model performance, hyperparameter tuning was executed through Grid search, and this time Lasso and Ridge regression was used. Lasso regression exhibited an R squared value of 0.016 while Ridge regression model exhibited slightly improved R squared value of 0.017. The Spearman rank correlation coefficients for both models indicated modestly positive relationships with the target variable. These findings prompted further explorations.

By reviewing the outcomes of the linear regression model, an investigation was initiated to discern potential factors contributing to the low performance. Initial considerations posted to the selected independent variables that were anticipated to impact insurance cost predictions. In light of this, two different approaches were adopted. Firstly, we conducted a correlation analysis to identify which columns could strongly affect our target variable "premium". This analysis helped us find variables that are strongly linked to insurance premium variations. Furthermore, considering that variables may interact non-linearly, we decided to create a non-linear algorithm that can understand more complex patterns in the data. For this model, we focused on business use case 2: insurance price change based on data collected about disease. A decision tree model was employed to capture the intricate interactions between disease data and insurance pricing.

Conducting the correlation analysis involved the application of one-hot encoding to the categorical variables. Subsequently, correlation coefficients were computed to identify the top 10 features with the highest correlation values to our target variable "LogPremium". This selection of 10 columns served as our feature variables while "LogPremium" remained as the target variable. The dataset was then partitioned into training and test sets with 8:2 ratio. The decision tree model was evaluated using mean squared error (MSE) and R-squared to gauge its performance. Our model exhibited a mean squared error value of 2.567 and a R-squared value of 0.019. Additionally, we examined the Spearman rank correlation coefficient which resulted in a value of 0.193.

Among the two machine learning models that we constructed, our preference learned towards incorporating a linear regression model into the database environment. This decision stems from our recognition that business model users are more interested in comprehending and extracting meaningful insights from the data. By choosing a linear regression model, our focus is on providing an approach that enables business users to grasp meaningful insights that can be directly communicated to customers. To facilitate the implementation of our developed machine learning models, we came up with an approach within the framework of our database ecosystem. Leveraging the capabilities of BigQuery, we have stored the coefficients associated with each column of our model. This enables us to run queries that yield the outcome of our machine learning models directly within the database environment. Such integration allows database users to retrieve the insurance price data they require. By embedding this capability within the database system, we have streamlined the process of acquiring insurance cost predictions and enhanced the efficiency of the decision making process.

While our current database's scalability is constrained by the free-tier limitations of BigQuery, our Big Data platform is strategically designed to cater to existing data demands and anticipate future expansion. The database's architecture makes it capable of accommodating data from diverse sources such as APIs, manual inputs, or imported datasets. This adaptability ensures that as data volumes surge, the database can expand while maintaining integration using different primary and foreign keys generated to prevent disruptions.

As we transition into the next phase of our project, our focus should now converge on leveraging hybrid data to tailor the reference architecture that is most suitable for the insurance company. While our preliminary database system takes shape, our paramount object lies in utilizing both structured and unstructured data. Although our project's dataset primarily comprises structured data, the operational reality for an insurance company includes the accumulation of both structured and unstructured data streams. Even within our database schema, we recognize the inclusion of unstructured data elements such as images of insurance claim documents or insurance billing records.

The raw data we used for the project mostly consisted of structured data but in reality, the insurance company will collect huge volumes of both structured and unstructured data. Even our database schema includes unstructured data such as images of insurance claim documents or insurance billing documents. To further develop the reference architecture, we can do comprehensive data analysis of the structured and unstructured data to identify patterns, trends and correlations within the data to gain insights into the types of information that hold strategic value. Also, we can develop a strategy for integrating hybrid data within the architecture. Defining data pipelines, transformation processes and reinforcing data enrichment techniques to ensure data consistency will help with the integration. Additionally, we can implement a metadata management framework that tracks the origin, lineage, and quality of both structured and unstructured data. Such a framework can address data security, compliance and other considerations.

Further tasks can be done using BigQuery, a cloud data warehouse we used for the project. BigQuery offers several powerful Big Data Analytics capabilities that enable users to efficiently extract, filter, store and analyze datasets.

- Data extraction: BigQuery provides tools for extracting data from various sources. It supports table or query results exports through its "EXPORT DATA" feature. Exports can be stored in file formats such as JSON, Avro, CSV, etc.
- Data filtering: BigQuery's SQL querying language enables sophisticated filtering and transformation of data during extraction. SQL queries using "WHERE" clauses can be generated to extract a subset of data. Also, it supports filtering resources using labels. Users can utilize the search bar in the Google Cloud console to find the data they would like to filter, and create filter specification for use in the API, bq command-line too, or client libraries.
- Data storage: BigQuery automatically stores data in a columnar format that is optimized for analytical queries. We can create partitioning or clustering queries to enhance data storage.
- Data analysis: SQL querying language supported by BigQuery allows for complex analytics operations including aggregation, JOINs, window functions and more. Furthermore, BigQuery's capability to integrate with python codes allows users to build and deploy even more complex models to perform analytics.

In the Database Systems Project Part 3, our focus on addressing real-time insights from both structured and unstructured data within the insurance sector has led to the development of a logical database model, schema optimization, and machine learning model implementation. The project led us to crucial insights, underscoring the significance of meeting the growing demand for immediate insights from data within today’s business landscape.

References

[1] Data Source Link: https://www.cdc.gov/nchs/nhis/2022nhis.htm

[2] ChatGPT, identifying and correcting errors. Used from August 2, 2023 to August 9, 2023. Retrieved from https://openai.com/chatgpt/