Name Chopra Aman, Ranjan Nidhi Date: 15th December 2022

NYU NetID: ac8511, nr2387

Section: CSCI-GA.2433-001

**Project Part 4**

We Aman Chopra and Nidhi Ranjan, hereby certify by submitting this project Part 1 that all the efforts put into this part are our own. We have referred the project support materials.

**Total in points** (100 points total): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Professor’s Comments:**

**A picture containing arrow

Description automatically generated**

**Affirmation of my Independent Effort:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**End-to-end Design:**

**Tool and libraries used:**

**Streamlit:**

We have used Streamlit to create the user interface

Streamlit is an open-source Python library for creating web-based data visualization and machine-learning tools. It allows you to build interactive, data-driven applications in just a few lines of Python code.

To use Streamlit, you simply write a Python script that calls the Streamlit API to build your app. Streamlit handles all of the underlying details of building a web app, including building the user interface, handling interactions, and serving the app over the web.

Streamlit has many more features and capabilities, including interactive widgets, maps, and markdown support.

**Psycopg2:**

psycopg2 is a Python library for interacting with a PostgreSQL database. It allows you to connect to a PostgreSQL database, execute SQL queries, and perform other database operations from within your Python code.

Here is an example of how to use psycopg2 to connect to a PostgreSQL database and execute a query:

psycopg2 is a powerful library that supports many advanced features, such as prepared statements, transactions, and asynchronous execution.

Text

Description automatically generated

**Pickle:**

The pickle module in Python allows you to serialize and deserialize Python objects. Serialization is the process of converting an object in memory to a byte stream that can be stored on disk or sent over a network. Deserialization is the reverse process of creating an object from a byte stream.

The pickle module provides functions for serializing and deserializing Python objects, including dump, which serializes an object to a file, and load, which deserializes an object from a file.

**Pandas:**

In Python, a pandas DataFrame is a 2-dimensional size-mutable, tabular data structure with rows and columns. It is similar to a spreadsheet or an Excel sheet. You can use it to store and manipulate tabular data, such as storing a collection of records with different data types (e.g., text, numerical, date).

You can create a DataFrame in a number of ways, such as from a NumPy array, from a list of dictionaries, or by loading data from a file (e.g., CSV). Here's an example of how to create a DataFrame from a list of dictionaries:

A screenshot of a computer

Description automatically generated with medium confidence

**Business Use Case:**

We are predicting the insurance prices of the customers and saving them in the database. This is an interaction between structured (the postgres database we created) and unstructured data entered by the user.

We first fetch the customer ID of the user and using that, we calculate the age (using DOB) and get the gender of the user. We then get the product the user is interested in to get the base price of the policy.

After that, we predict the probability of the patient having diabetes using the model we created in part 3. Using that we calculate the premium and save it to the database.

**Steps:**

First, we import streamlit for the user interface and Psycopg2 to communicate with the postgres database that is running. We also import the pandas dataframe to execute the query and pickle to load our model.

**Import statements:**

Text

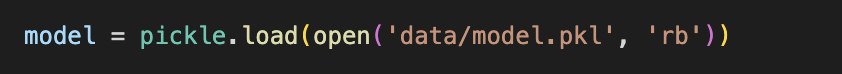
Description automatically generated

**Connect to the database:**

Text

Description automatically generated

**Load the model:**



**Display data:**

You can select the table and we show all the data from the table:

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

*Graphical user interface, text, application

Description automatically generated*

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text

Description automatically generated

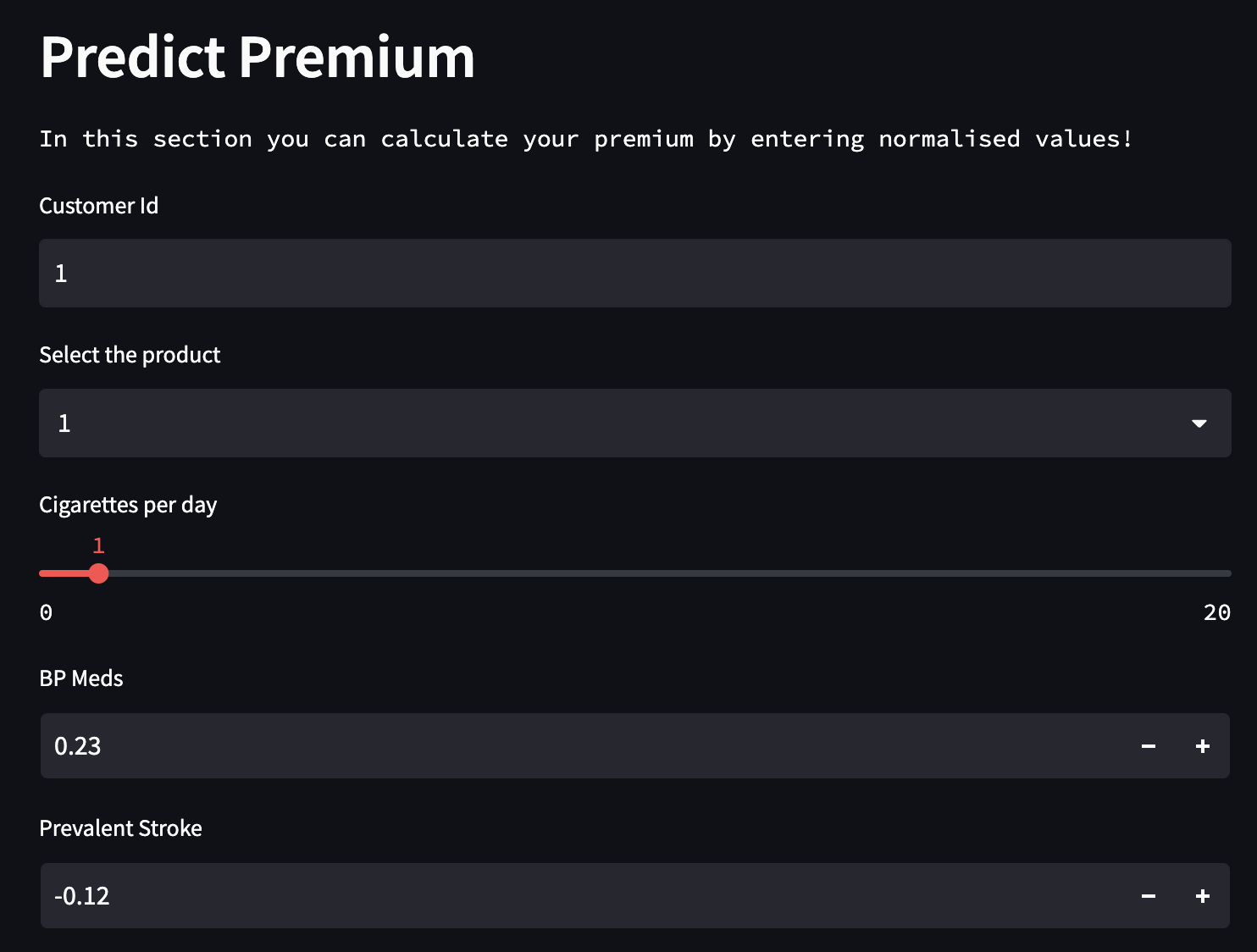
Graphical user interface, application

Description automatically generated

**Predict Premium:**

Text

Description automatically generated



Background pattern

Description automatically generated

**Get Premium:**

Text

Description automatically generated

We first get the age and gender from Customer ID and get the base price from the Product\_Id. After that, we predict the premium after running our model.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

**Conclusion:**

This is an end–to–end solution that communicates with the Machine Learning model as well as our postgres database to do a business use case. The User Interface is an easy way to interact with our postgres database as well as our Machine Learning model.

**GitHub Link:**

<https://github.com/Aman-Chopra/DBMS-Car-Insurance-Prediction>