**ST 6210 Data Visualization Research Project**

May 7, 2022

Team member: Chenfei Wang, Yu Xia, Rachel Ren, Haoze Wang

**Prediction of Patient Stroke Probability**

1. **Introduction**

The premise of the project revolves around predicting the probability of stroke occurrence for patients as it has become the second leading cause of death around the globe. About 11% of deaths in the world are related to stroke. Therefore, it is essential to explore which factors are more likely to cause stroke and predict whether a person is more likely to get a stroke.

1. **Data Processing & Data Analyzing**

Our primary goal is to extract the information from the dataset downloaded from Kaggle and make the best use of them to aid our analysis and prediction.

* 1. **Data cleaning**

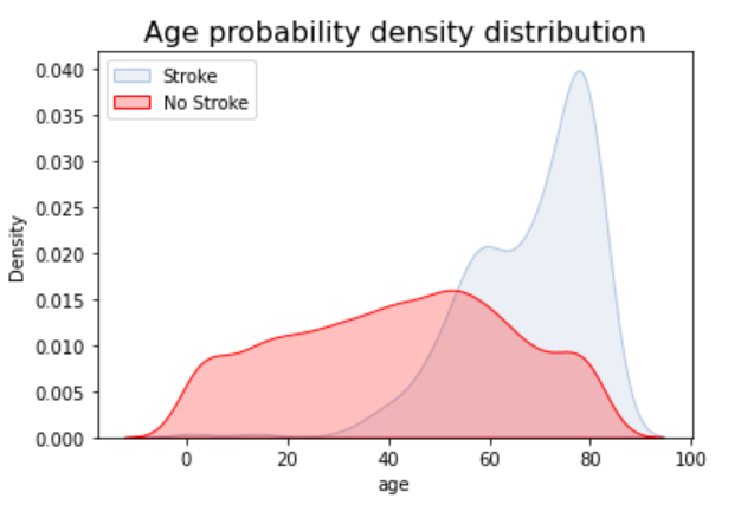
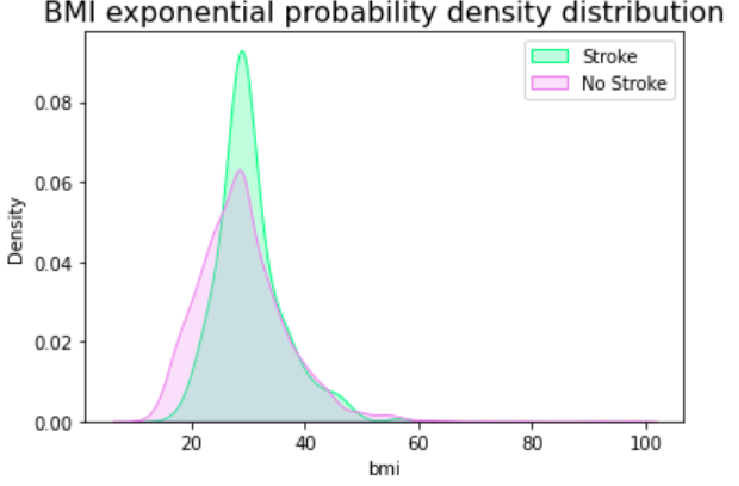
The dataset itself contains 12 features, as shown in the following figure. We first started with dropping unnecessary columns and filling NaN values in average.



*Figure 1. Sample output of the CSV.*

* 1. **Briefly analyzing**

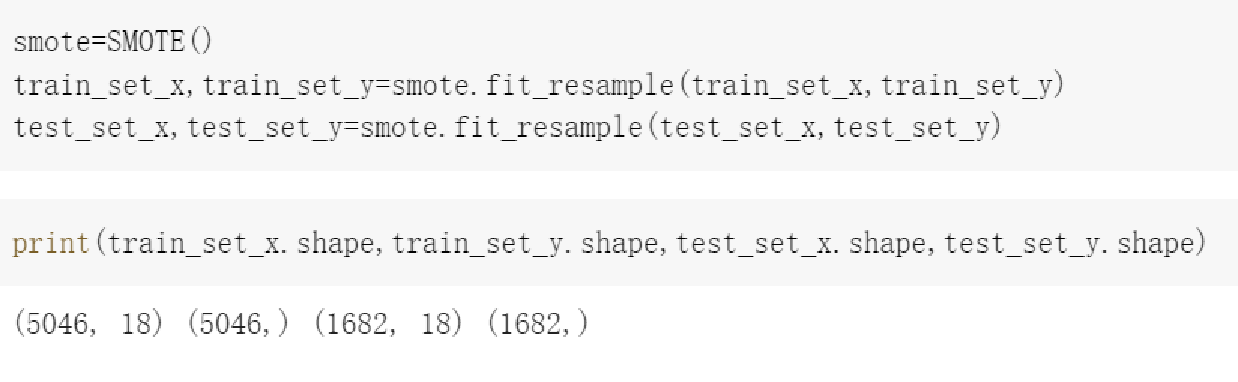
This data cleaning is the first part of the project. The next step is to extract information from the cleaned dataset. We did a simple data analysis on Google Colab and found that there are differences in the distribution of some features between the two groups, such as age. Stroke patient is often older than healthy people. In contrast, there was little difference in the distribution of other features between the two groups, such as body weight. So we should focus on how to find out those features with obvious differences between the two groups.

*Figure 2 & 3. Two cases with different features distribution.*

* 1. **Dataset balancing**

We used the get\_dummies function to convert string variables into numerical variables. Then we balanced the data set, because in the data set, the number of non-stroke patients is far bigger than stroke patients, and direct training will lead to mistakes. We can use the SMOTE method: to create new composite points from a few classes. Then as the picture shows, the data set is balanced. Then we began the next step of feature engineering.



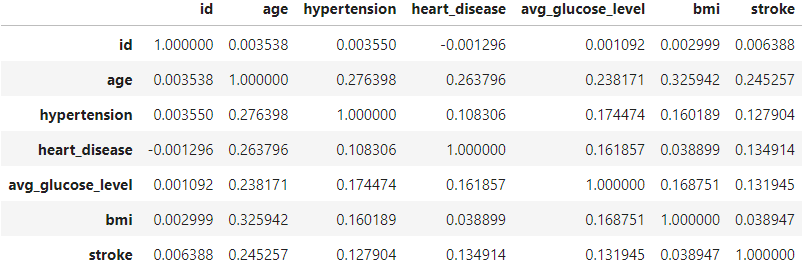
*Figure 4. Use the SMOTE function to balance the dataset.*

1. **Feature Engineering**

After consulting multiple online resources, we realized that most similar research done does not include feature engineering. We decided to incorporate this technique to make use of existing variables and create new features in the data to help the Machine Learning models we select perform better on this dataset.

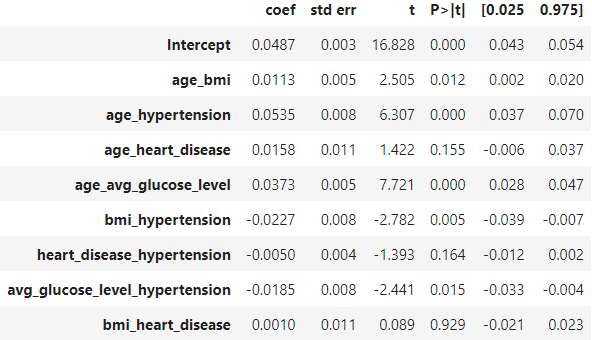
* 1. **New Features with Numerical data**

Our first step in making new features was to check the correlation between each variable. Based upon the correlation of the data frame, we created eight new features belonging to three categories. The first category contains features that are somewhat related to each other. In our case, 'age\_bmi'; 'age\_hypertension'; 'age\_heart\_disease'; 'age\_avg\_glucose\_level' are under this category. The name of different features are separated by an underscore and they are created by **multiplying the values inside the two features**. Similarly, the second and third categories are less related features and features that are least related, and they each contain 'bmi\_hypertension'; 'heart\_disease\_hypertension'; 'avg\_glucose\_level\_hypertension'; and 'bmi\_heart\_disease'.



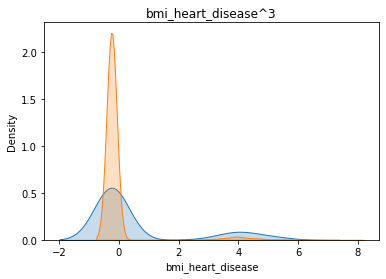
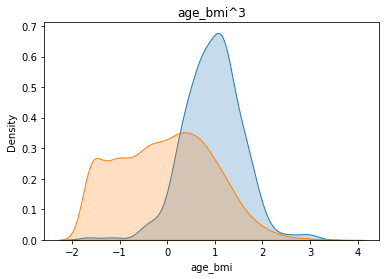
*Figure 5. Correlation of the data frame.*

We implemented the **Statsmodel Formula API** to check if the new features are valuable or statistically significant to the dataset. This API provides a summary of the model with R-squared value, f-value, p-value, t-value, coefficient, and so on.



*Figure 6. Some statistics of the new data.*

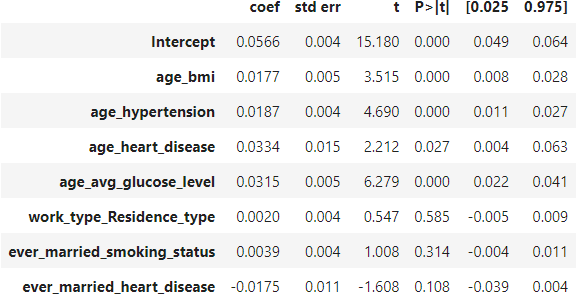
We wrote a function implementing both Statsmodel API and Seaborn KDE plot. The previous conclusion about the p-value states that the higher it is, the less significant the variable is to the model. It reflexes in the plots below as the feature 'bmi\_heart\_disease' has both high p-value and it consists of two features with very similar behavior.

****

*Figure 7 & 8. KDE plot of 'age\_bmi' on the left, 'bmi\_heart\_disease' on the right.*

* 1. **New Features with String data**

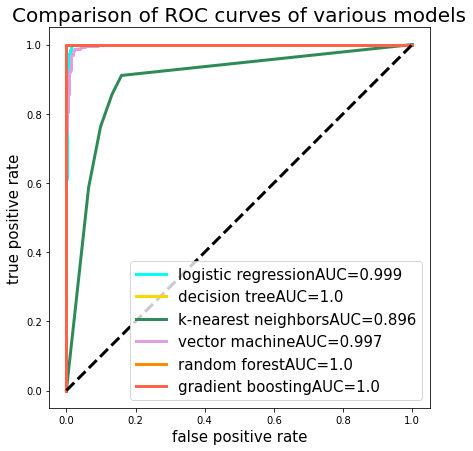
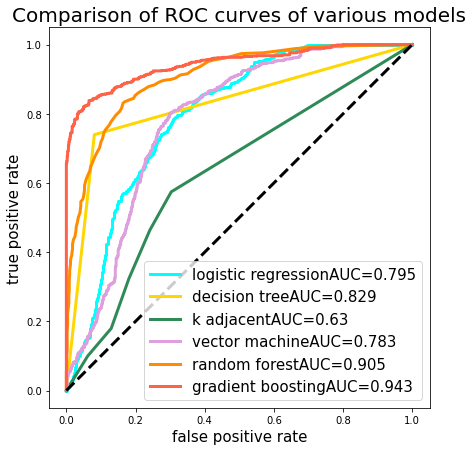
We repeated the steps above and made three new features with the encoded string data. They are 'work\_type\_Residence\_type', 'ever\_married\_smoking\_status', and 'ever\_married\_heart\_disease'. These features have higher p-values than the numerical data. This could mean that they make less of a difference than the numerical data.



*Figure 9. Some statistics of the new string data.*

* 1. **New Features’ Impacts on Results**

We added the new features to the data set and trained them to learn if they help increase the accuracy of our models.



*Figure 10 & 11. ROC curve plot of numerical data on the left, string data on the right.*

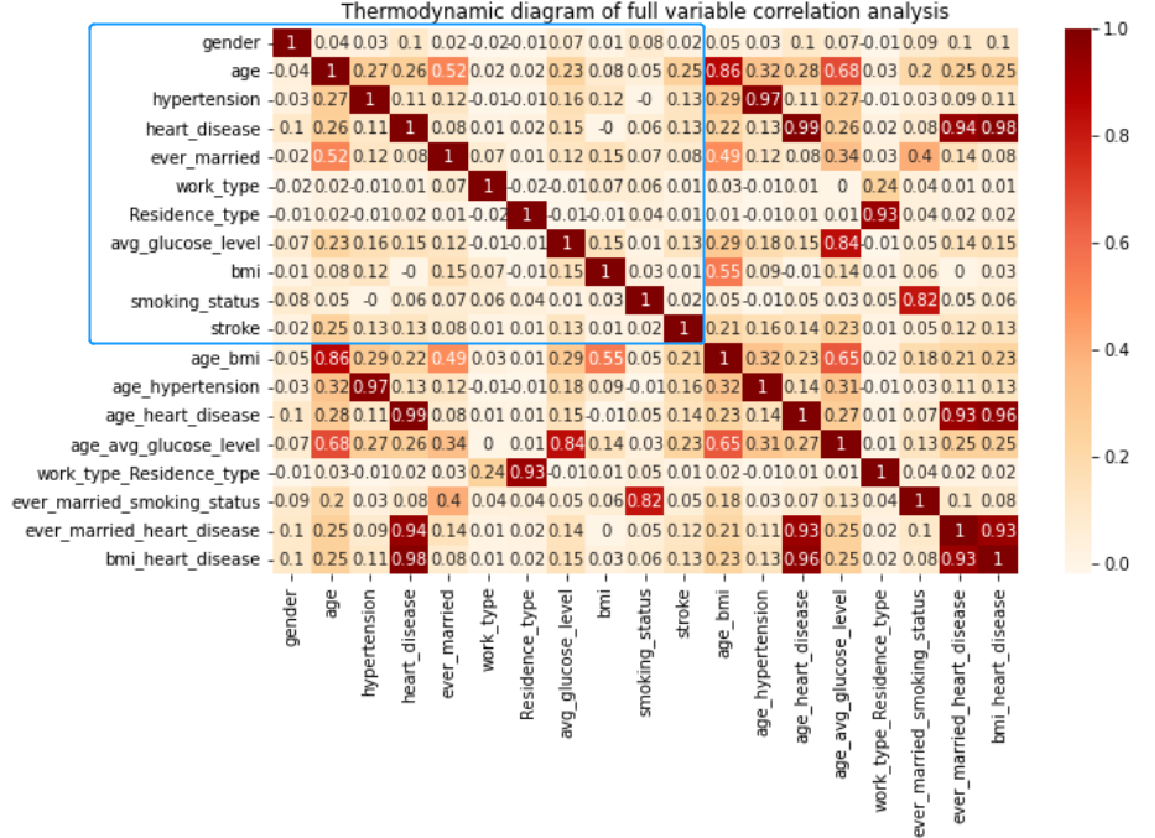
Both types of new features made positive effects on the models. From the plots, it is clear that the string data makes the most impact on the results. Improvements of around 1% in accuracy were seen with the addition of numerical data, while the accuracy of most models is close to 100% with the addition of string data. We suspect that overfitting is occurring in the results and we look to revise the model with cross-validation to avoid overfitting in future works.

1. **Model training & predicting**

Now that the data has been extracted, our next goal was to provide methods that would allow us to more easily observe if there were any patterns, trends, or other insights that can be gleaned from high-level analysis of the data.

* 1. **correlation analysis**

We analyzed the correlation between variables and found that only age is strongly related to marriage because older people are more likely to be married. The correlation between other single features is not strong, so it will not have a great impact on the subsequent prediction.

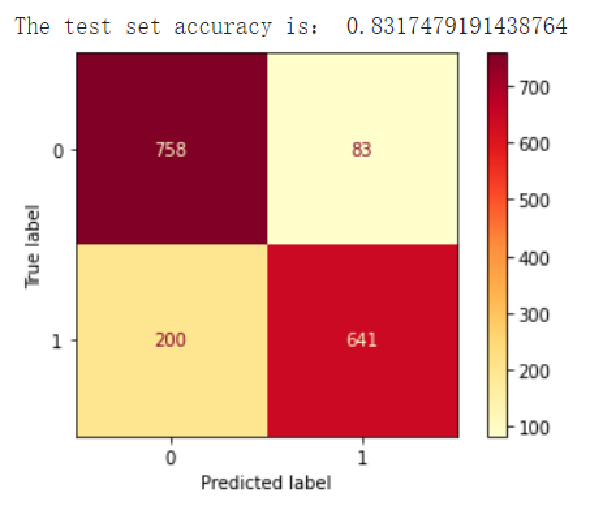
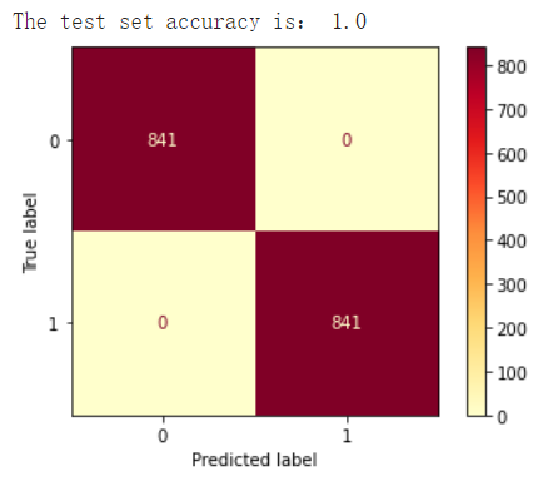
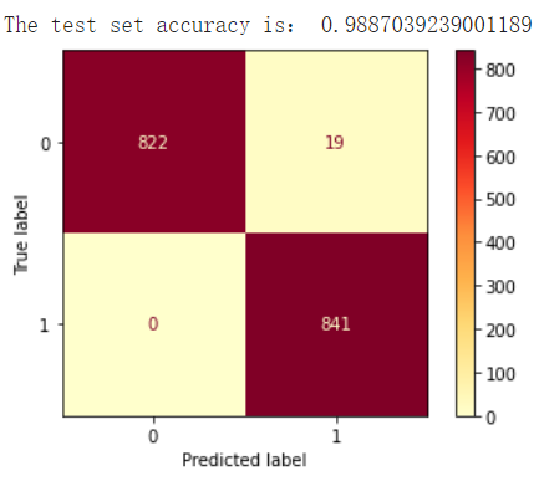


*Figure 12. Correlation between single features.*

* 1. **Model prediction**

We imported the Sklearn function library into Python and used the built-in machine learning model algorithm to train and predict the results of our processed datasets.

We used logistic regression, decision tree, k-nearest neighbors, support vector machine, random forest, gradient boosting, and gradient descent algorithms to process the dataset. We found that after constructing the new features, only the K-nearest Neighbors(KNN) algorithm performs slightly worse, and the other six algorithms all performed well.



*Figure 13-15. Prediction results of some models.*

1. **The GUI**
   1. **The choice of development tools**

In this project, we chose PyQt5 and QtDesigner to design and implement a Graphical User Interface(GUI). Because we compared the Tkinter library and the QT library, we found that the QT library has more functions, and its development speed and maintainability are better, while QT is more suitable for beginners to use.

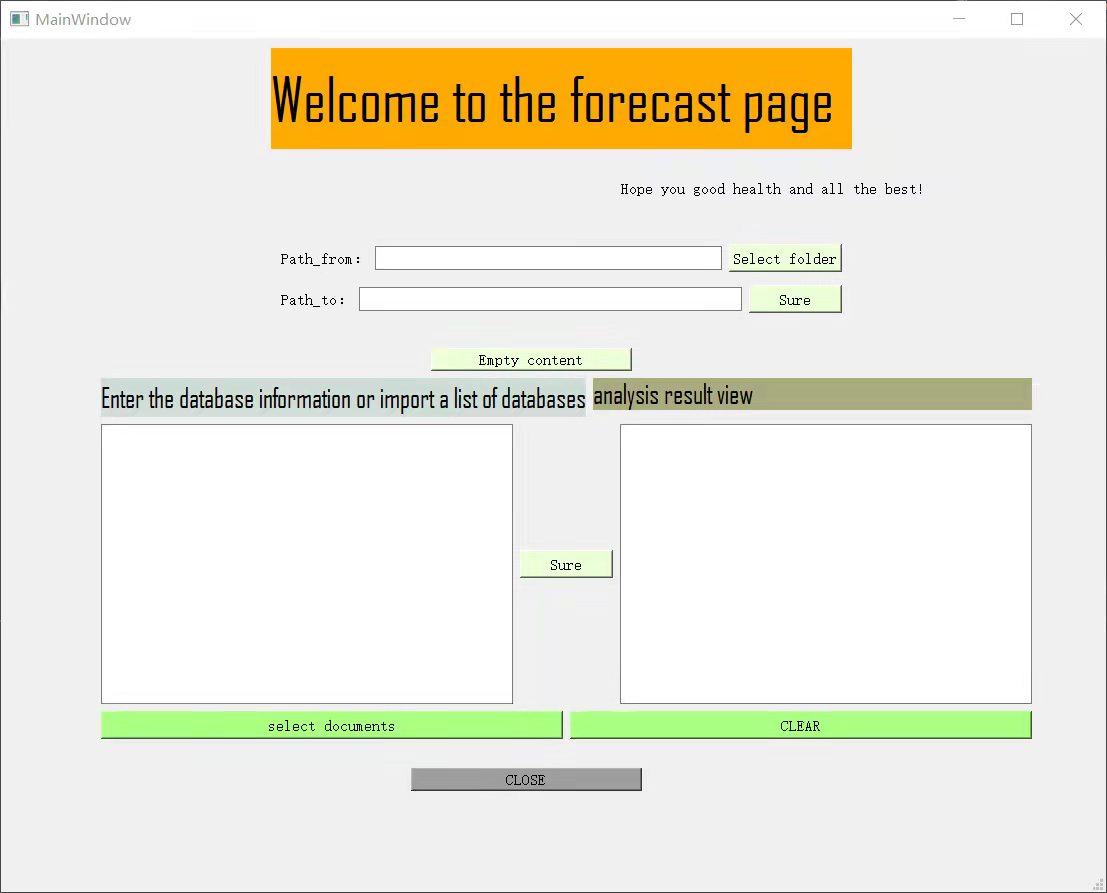
* 1. **Function demands**

1. Build a prediction model that can be connected to the graphical user interface to accept user input data and give model inference results.

2. Generate a display interface. Users only need to input values as prompted to obtain corresponding reasoning results.

3. Export and import forecast data of the training model and display the data in the original data set.

* 1. **Data analysis and system design**

****

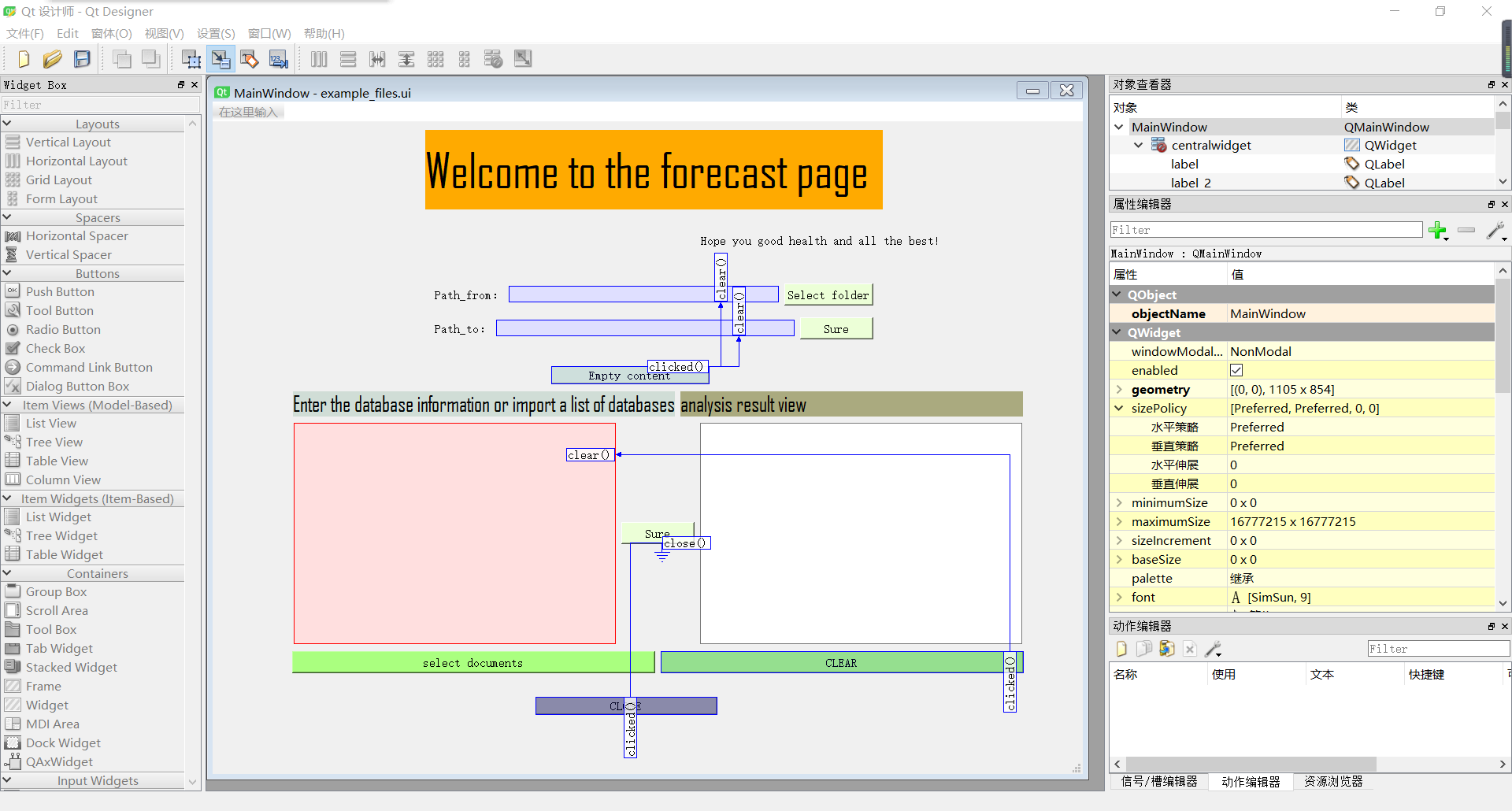
*Figure 16. Start Screen.*

There are three modules in this main window:

The first module is the data folder dump. It uses two custom functions: get\_dir and copy\_file. It reads the destination folder through the QtFileDialog class and copies the folder to the destination path.

The second module is the call to the prediction model. We added the trained prediction model to the project, and at the same time, we completed the comparison between the values of user personal data and the prediction data by reading the values in the text box. It is worth mentioning that we have designed two data entry methods. Users can manually enter ten items of personal data, or directly import the data text prepared in the folder into the text box.

The third module is to translate through a web crawler. Considering that users from different regions use different languages, we found a solution by learning Python crawler translation. We define a function called translate\_file, which takes the contents of the text box and then translates the text by calling the crawler function.

****

*Figure 17. Design interface of QT Designer.*

Using Qt Designer, we set up signals and slots directly, customizing how buttons and text boxes are connected as needed.

* 1. **The difficulty and solving measures in the system exploitation**

Because most of the references about the interface we found on the Internet are completed by pycharm, we save the code file on Google Colab as .py format, and then open it in pycharm. In this way, we can use the previous training model in the process of interface design.

Disadvantages: Considering the different languages used by users in the result presentation, members of our team learned Python crawler translation, but found that the translation was not successful in the final presentation. It can only translate the first line of text. We will continue to learn and improve.

1. **Conclusion**

We constructed a reasonable and innovative prediction method to predict whether patients are easy to get strokes by using some patient features and designed a GUI to output the results. We believe the outcome of our project is very optimistic. In the future, we can continue to deepen our work in extracting more significant features and optimizing interface design.

1. **Some feedback from our group**

Chenfei: I think this project is very helpful to me. I learned a lot about using Python for data analysis and processing, and I also improved my oral English in the process of interacting with the professor. In addition, as the team leader, my leadership in this project has also been improved. All these will benefit me all my life.

Haoze: Since my majors are mathematics and economics, I was not familiar with python and data visualization before the project. Under the guidance of Professor Mark and the explanation of the teacher during the mentor session, I began to be skilled in using various libraries through classroom tasks and homework to achieve the purpose of data visualization. And finally, in the end, completed the final project with the team members.