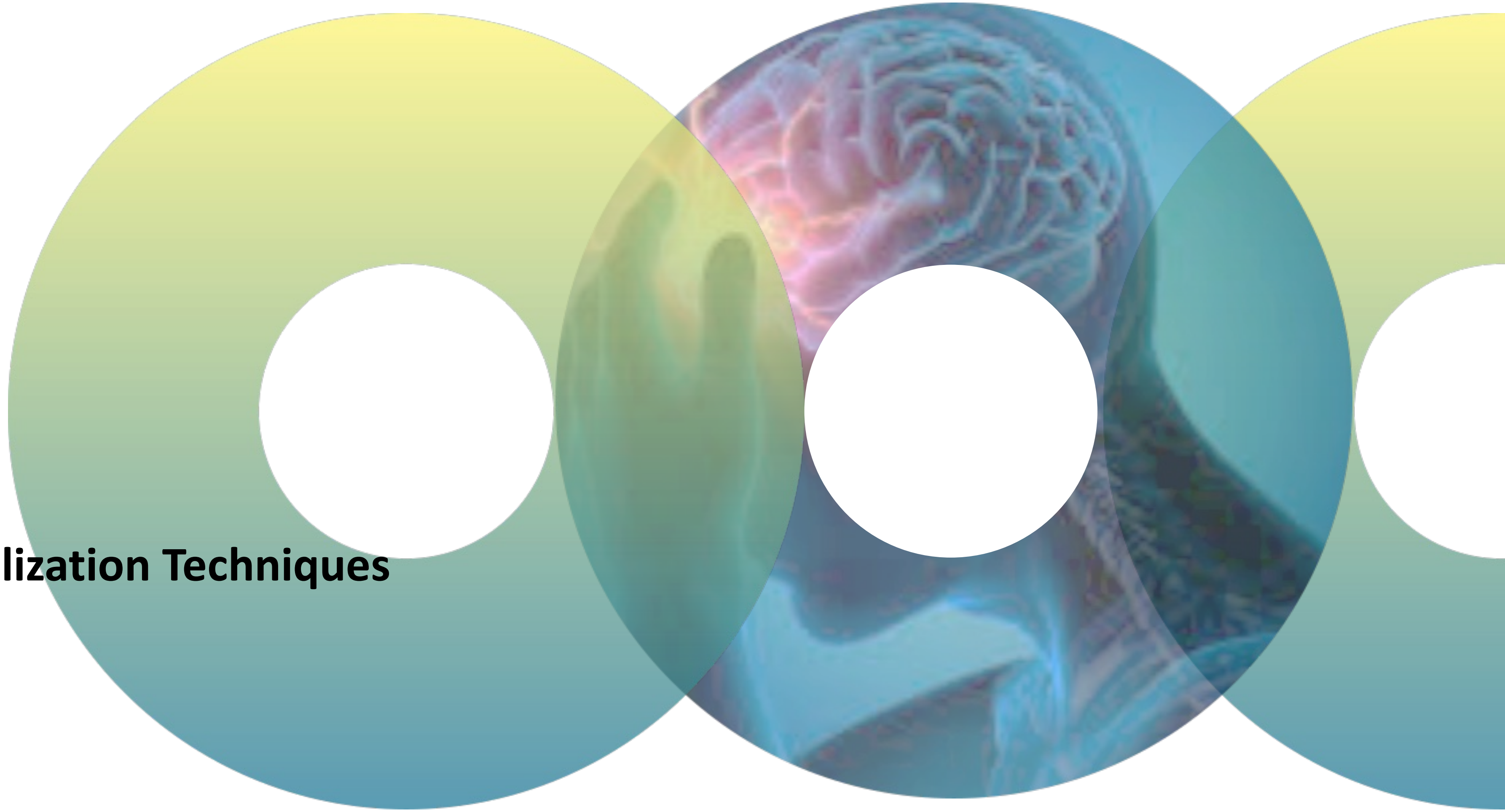
Predicting Stroke Risk

Analyzing Key Factors and Building a Robust Model



MSCA 32007 Data Visualization Techniques

March 9, 2023

TEAM 7

Team 7

Predicting Stroke Risk

Analyzing Key Factors and Building a Robust Model

Table of Contents

Predicting Stroke Risk

> Stroke is ***the 2nd leading cause of death*** globally, responsible for approximately 11% of total deaths.
>
> \- World Health Organization (WHO) -

Early Prediction + Key Factor Diagnosis = Business Facilitation

Our project aims to identify critical factors contributing to stroke and build a predictive model. Our final goal is to develop an application to monitor at-risk individuals to help them maintain good health and prevent stroke-related complications.

Data Analysis Methodology and Tools

Data Source

The data source used for this project is Kaggle.

EDA and Modeling

All of the data cleaning, exploratory data analysis, feature engineering and modeling process is done in Jupyter notebook using Python.

Prototyping

Microsoft Powerpoint is used to create a mockup for the Heart Stroke Prediction Application.

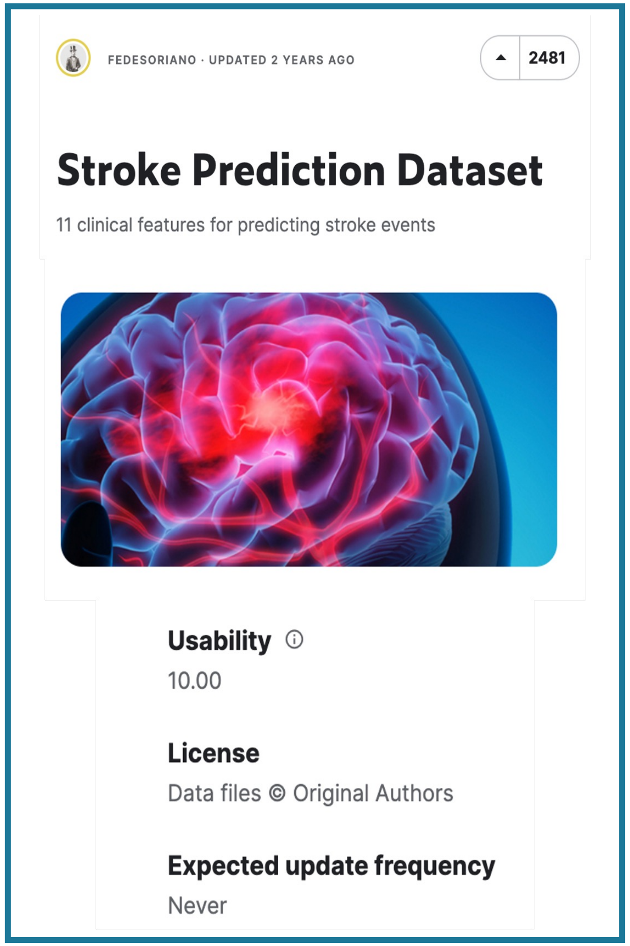
Dashboard

KPI, Metrics, and other visualizations to analyze the Heart Stroke case are built in Tableau.

Data Description

	Stroke Prediction Dataset*
Topic	11 clinical features for predicting stroke events
Observations	5,110 observations
Variables	12 variables
	id, gender, age, hypertension, heart_disease, ever_married, work_type, Residence_type, avg_glucose_level, bmi, smoking_status, and stroke
Variable Types	1. Categorical variables (e.g., gender), 2. Binary variable (e.g., ever_married), and numeric variables (e.g.,age)
Null Values	Some missing values exist due to patients' unwillingness to provide their personal data and such missing values need to be handled before EDA

*https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset



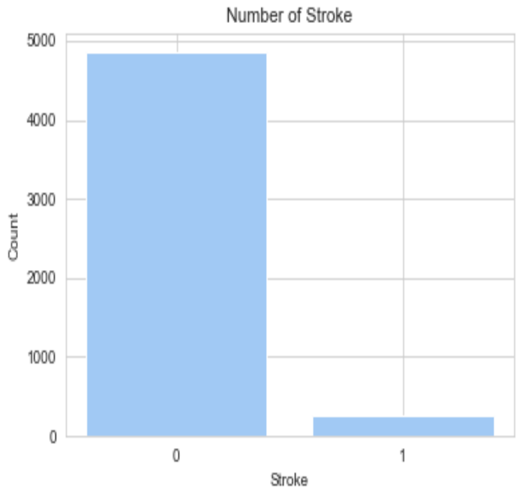
Data Preparation

Data Type

```
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 5110 non-null   int64
 1   gender             5110 non-null   object
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64
 4   heart_disease      5110 non-null   int64
 5   ever_married       5110 non-null   object
 6   work_type          5110 non-null   object
 7   Residence_type     5110 non-null   object
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object
 11  stroke             5110 non-null   int64
dtypes: float64(3), int64(4), object(5)
```

We categorize our features as either categorical or numerical variables and adjust their data types accordingly.

Data Structure



Our target variable has binary outcomes. The values are highly imbalanced and we may need to upsample the data in the future.

Null Values

```
df['bmi'].describe()

count    4908.00000
mean       28.89456
std         7.85432
min        10.30000
25%        23.50000
50%        28.10000
75%        33.10000
max        97.60000
```

```
df['bmi'].describe()

count    5109.000000
mean       28.928557
std         7.775535
min        10.300000
25%        23.600000
50%        28.100000
75%        33.100000
max        97.600000
```

In the BMI feature, approximately 4% of the data is missing(201/5,110). To impute these missing values, we will use the .interpolate() function.

Outliers and Null Values

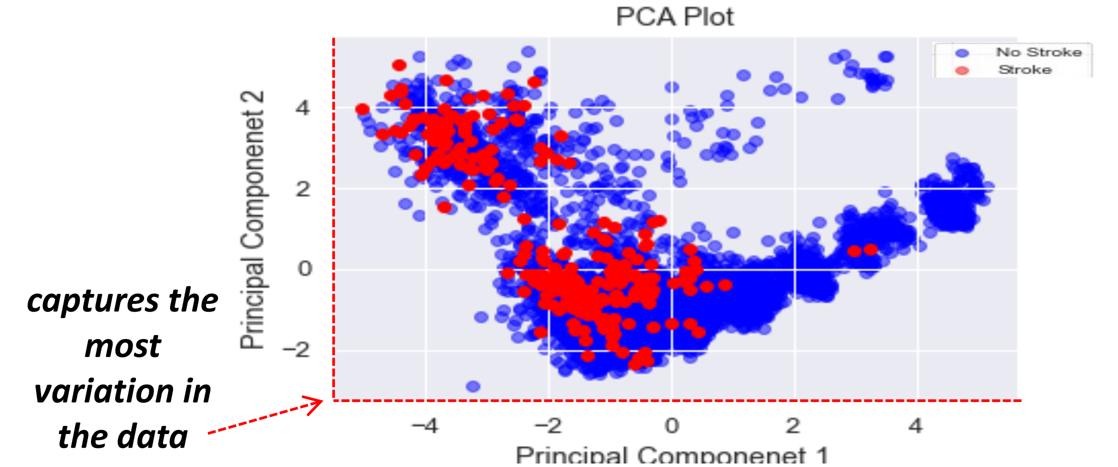
```
df['gender'].value_counts(ascending = False)

Female    2994
Male      2115
Other        1
Name: gender, dtype: int64

mask = df['gender'] == 'Other'
df = df.drop(df[mask].index)
```

Since there is only one observation in the gender variable categorized as 'other', we will simply drop this observation.

Data Visualization through Dimensionality Reduction

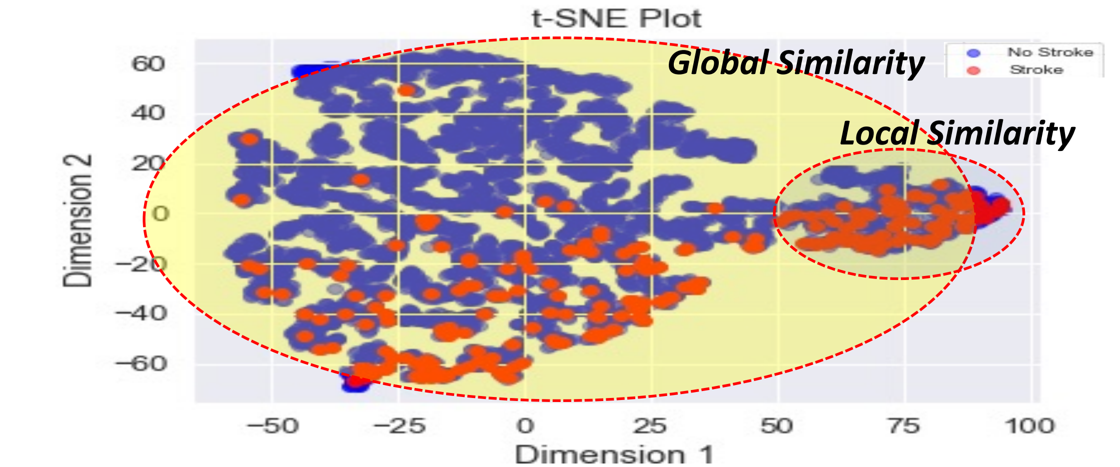
> How will our data be represented in ***2 dimensions***
> after we use dimensionality reduction techniques to compress our 11 features?

PCA (Principal Component Analysis)



From the PCA plot, we can understand the relationships within the data in the reduced-dimensional space while retaining as much of the original variability as possible.

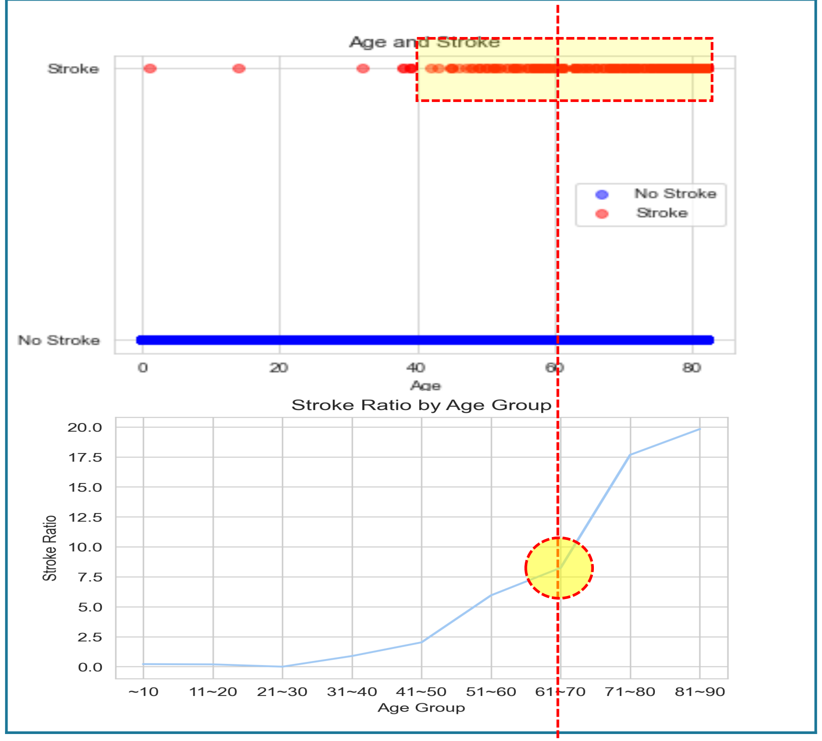
t-SNE (t-Distributed Stochastic Neighbor Embedding)



The clusters that are closely grouped together by t-SNE in the 2 dimension represent subgroups of the data that share similar characteristics or features in the higher dimensions.

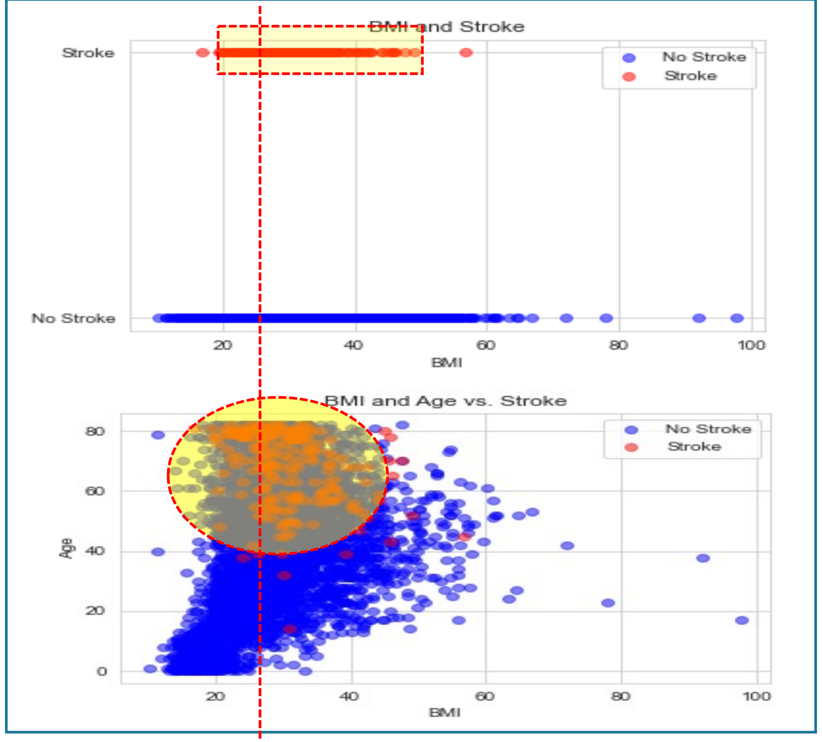
Exploratory Data Analysis: Numerical Variables

Age



The risk of stroke increases rapidly after the age of 40, and significantly rises after the age of 60.

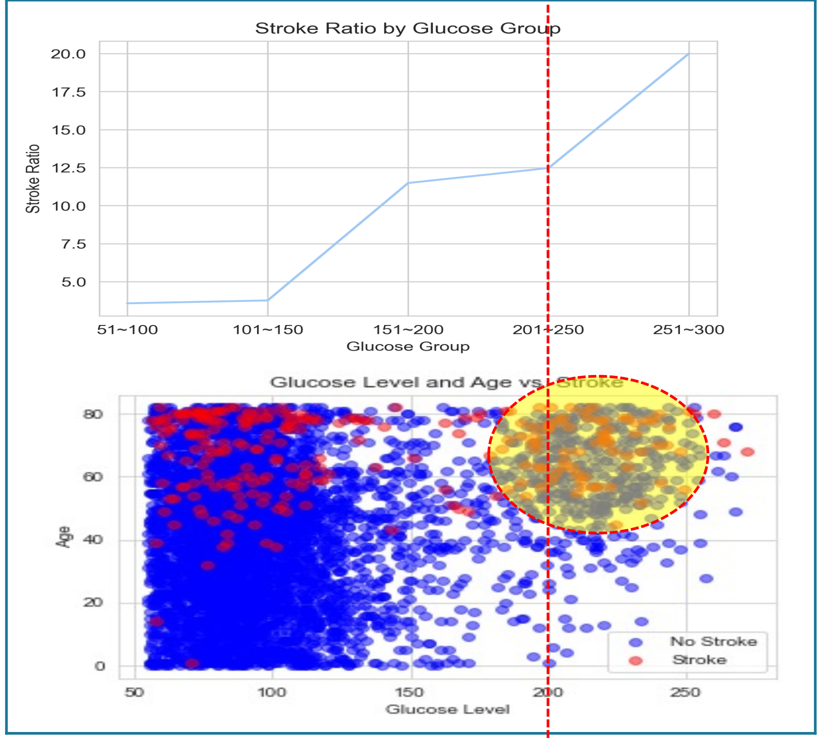
BMI* (Body Mass Index)



*BMI = kg/m^2

Having a BMI over 25(overweight or obese) raises the likelihood of stroke, with obesity after 60 years old being a notable contributing factor.

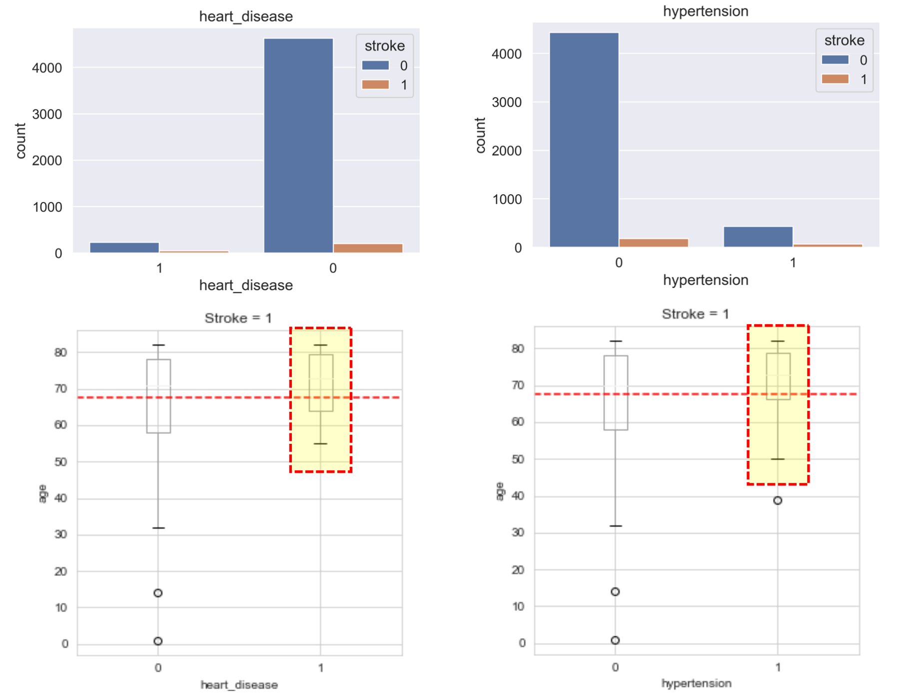
Glucose Level



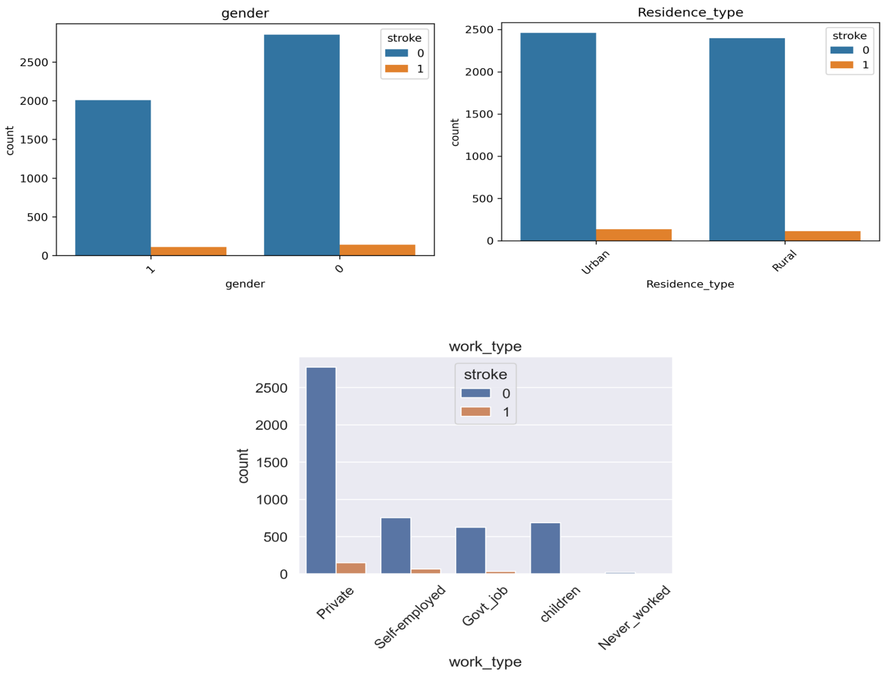
Elevated glucose levels exceeding 200 indicate diabetes, and when coupled with advanced age, may constitute a crucial risk factor for stroke.

Exploratory Data Analysis: Categorical Variables

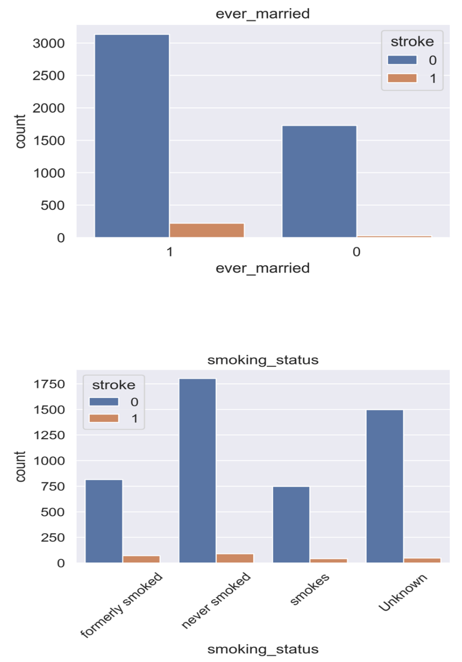
Medical History



Demographic and Social Status



Life Style



Individuals with a medical history who experienced a stroke tended to have a higher average age.

The factors that account for the highest proportion of stroke occurrence include being married, never smoking, female gender, living in an urban area, and working in the private sector. However, it is important to analyze the stroke occurrence ratios within these groups.

Feature Engineering

> Feature engineering improves machine learning by creating useful features from raw data. Techniques like ***binning, one-hot encoding, and feature selection*** help models capture patterns and relationships in the data.

One-Hot Encoding

- If a variable has only two unique values, we can assign 0 or 1 to each value.
- For variables with multiple unique values we can use one-hot encoding to create binary features for each unique value.

Binning

- We created three new feature columns in our dataset: age_bin, glucose_bin, and bmi_bin. Age_bin has 10-year intervals, glucose_bin indicates whether a person has diabetes or not, and bmi_bin indicates whether a person is obese or not.

Feature Selection

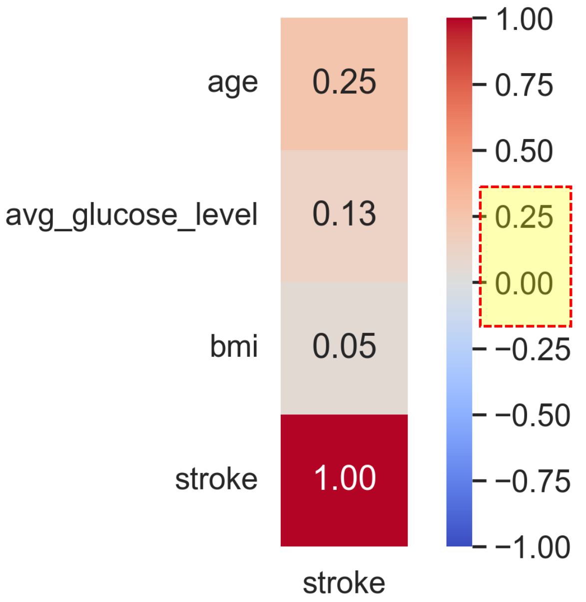
- To analyze the relationship between the target variable and numerical variables, we calculate their correlation coefficients. For categorical variables, we use chi-square statistics to determine the significance of p-values.

Feature Engineering

Feature Selection

- To analyze the relationship between the target variable and numerical variables, we calculate their correlation coefficients. For categorical variables, we use chi-square statistics to determine the significance of p-values.

Correlation Plot (Numerical Variables)



The correlation coefficients for the numerical variables fall in the range of 0 to 0.25. This is a weak correlation, even though it does not necessarily imply no relationship.

Chi-Square Statistics Table (Categorical Variables)

Features	Chi-Square	P-value
gender	0.34	0.5598
hypertension	81.57	0.0000
…	…	…
Residence_type	1.07	0.2998
smoking_status	29.23	0.0000
…	…	…

We may consider removing the 'gender' and 'Residence_type' variables due to the lack of significance on the target variable.

Feature Engineering

Target Variable

- stroke

Original variables

Medical History

- hypertension
- heart_disease

Social Status

- ever_married
- work_type

Life Style

- smoking_status

Dropped Variables

- age
- bmi
- avg_glucose_level

Weak Correlation

- id
- gender
- Residence_type

Insignificant P-values

Created Variables

10 Year Interval

- age_groups

Obesity

- bmi_groups

Diabetes

- glucose_groups

Modeling

Data Preparation

- Removing unnecessary variables
- Handling missing values
- Encoding categorical variables

Data Splitting

- Splitting data into training and testing sets

Model Selection

- Testing multiple classifier models
- Evaluating accuracy scores to narrow down options

Model Comparison

- Comparing top models using confusion matrices.
- Analyzing feature importances to understand variable importance

Model Evaluation

- Creating ROC-AUC Curve.
- Performing cross-validation to detect overfitting.
- Selecting final model based on evaluation results

Candidate Models

- Linear models: Logistic Regression, Linear SVC, Perceptron, Stochastic Gradient Descent
- Non-linear models: Support Vector Machines, K-Nearest Neighbors, Decision Tree, Naive Bayes
- Non-linear / Ensemble model: Random Forest

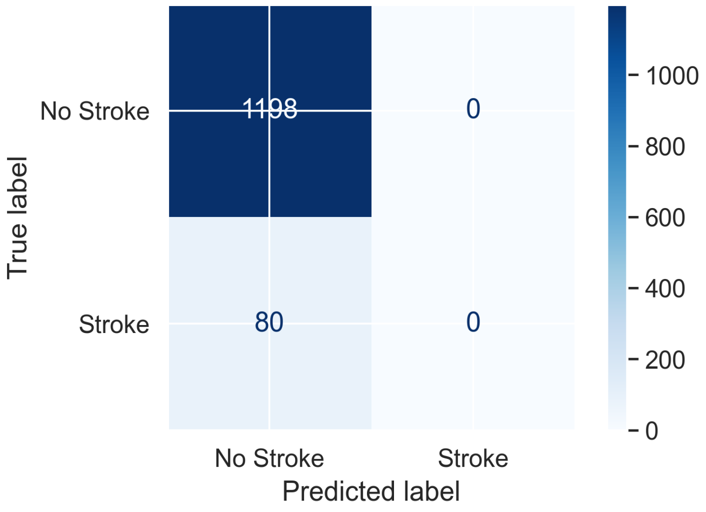
Prediction

Accuracy by Classifier Models

Model	Accuracy Score
Logistic Regression	93.74
Support Vector Machines	93.74
Linear SVC	93.74
Perceptron	93.74
Stochastic Gradient Decent	93.74
KNN	93.66
Random Forest	92.57
Decision Tree	92.02
Naïve Bayes	36.07

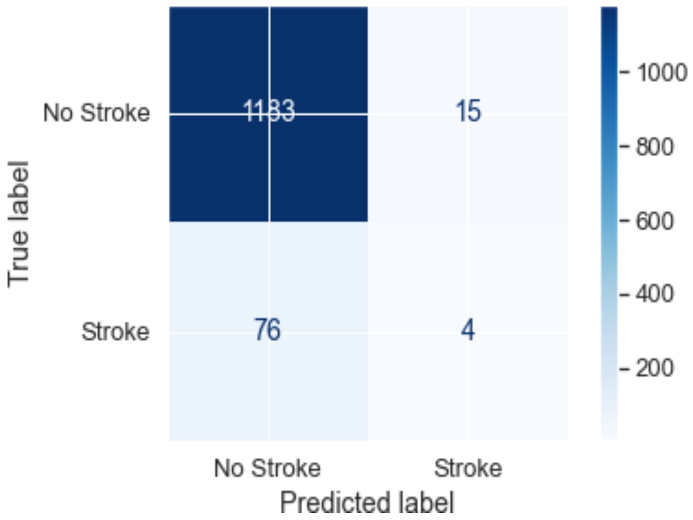
** Logistic Regression: Linear Classification*
** Random Forest: Non-linear Classification*

Logistic Regression



	Precision	Recall	F1-Score
No Stroke	0.94	1.00	0.97
Stroke	0.00	0.00	0.00

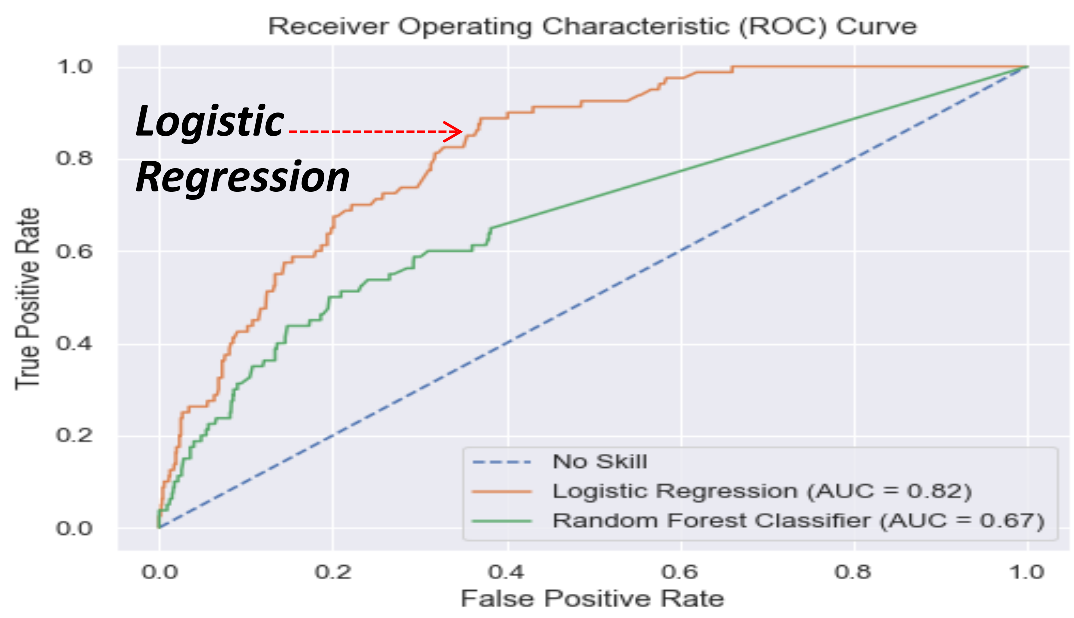
Random Forest



	Precision	Recall	F1-Score
No Stroke	0.94	0.99	0.96
Stroke	0.21	0.05	0.08

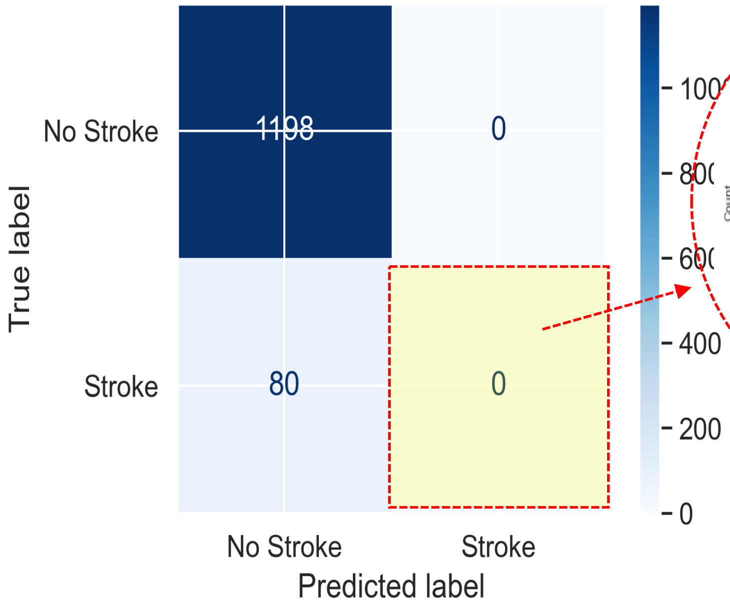
Prediction

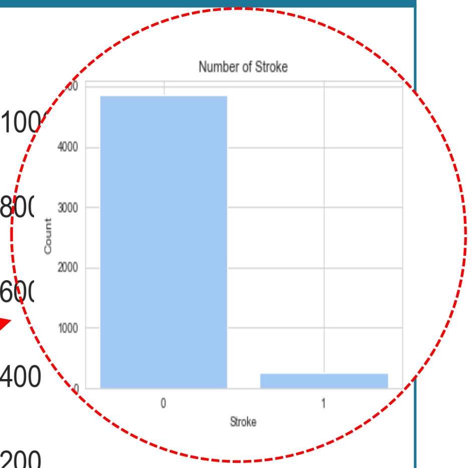
ROC – AUC Curve



ROC-AUC curve allows us to visually inspect the model's trade-off between sensitivity and specificity and to compare the different models' performance. Logistic Regression model's higher AUC-ROC score indicates a better ability to distinguish between positive and negative classes.

Confusion Matrix: Logistic Regression



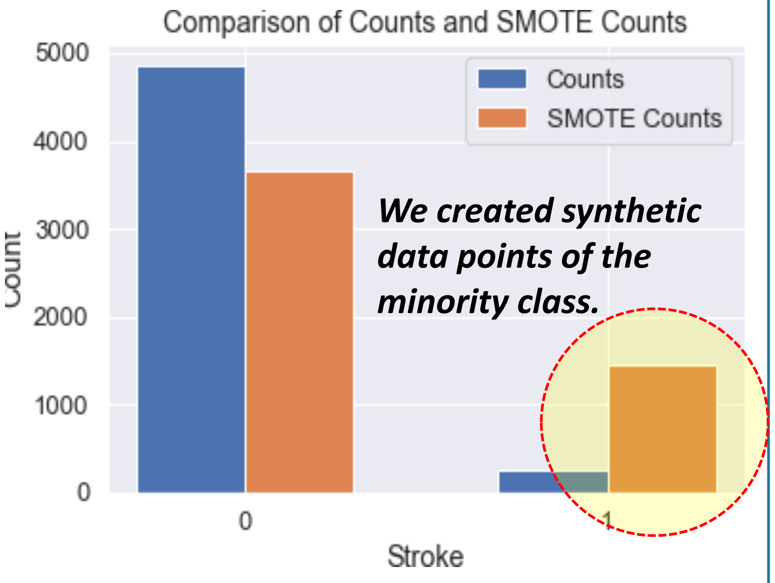


The model fails to correctly predict the minority class due to the imbalance in the target variable.

	Precision	Recall	F1-Score
No Stroke	0.94	1.00	0.97
Stroke	0.00	0.00	0.00

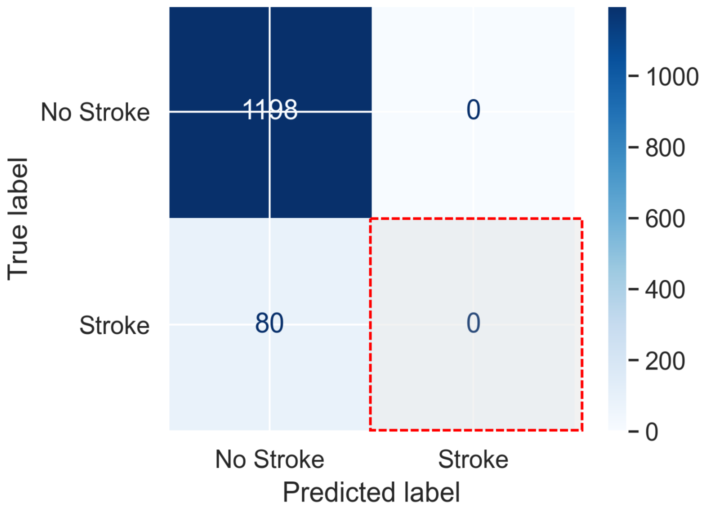
Final Model

SMOTE: Upsampling



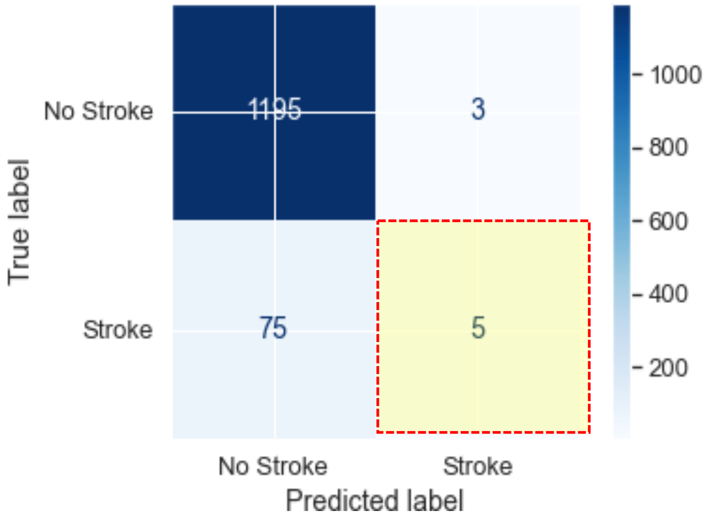
	Stroke (1)	No Stroke (0)
Original Data	249 (5%)	4,860 (85%)
SMOTE Data	1,464 (29%)	3,662 (71%)

Before SMOTE



	Precision	Recall	F1-Score
No Stroke	0.94	1.00	0.97
Stroke	0.00	0.00	0.00

After SMOTE



	Precision	Recall	F1-Score
No Stroke	0.94	1.00	0.97
Stroke	0.62	0.06	0.11

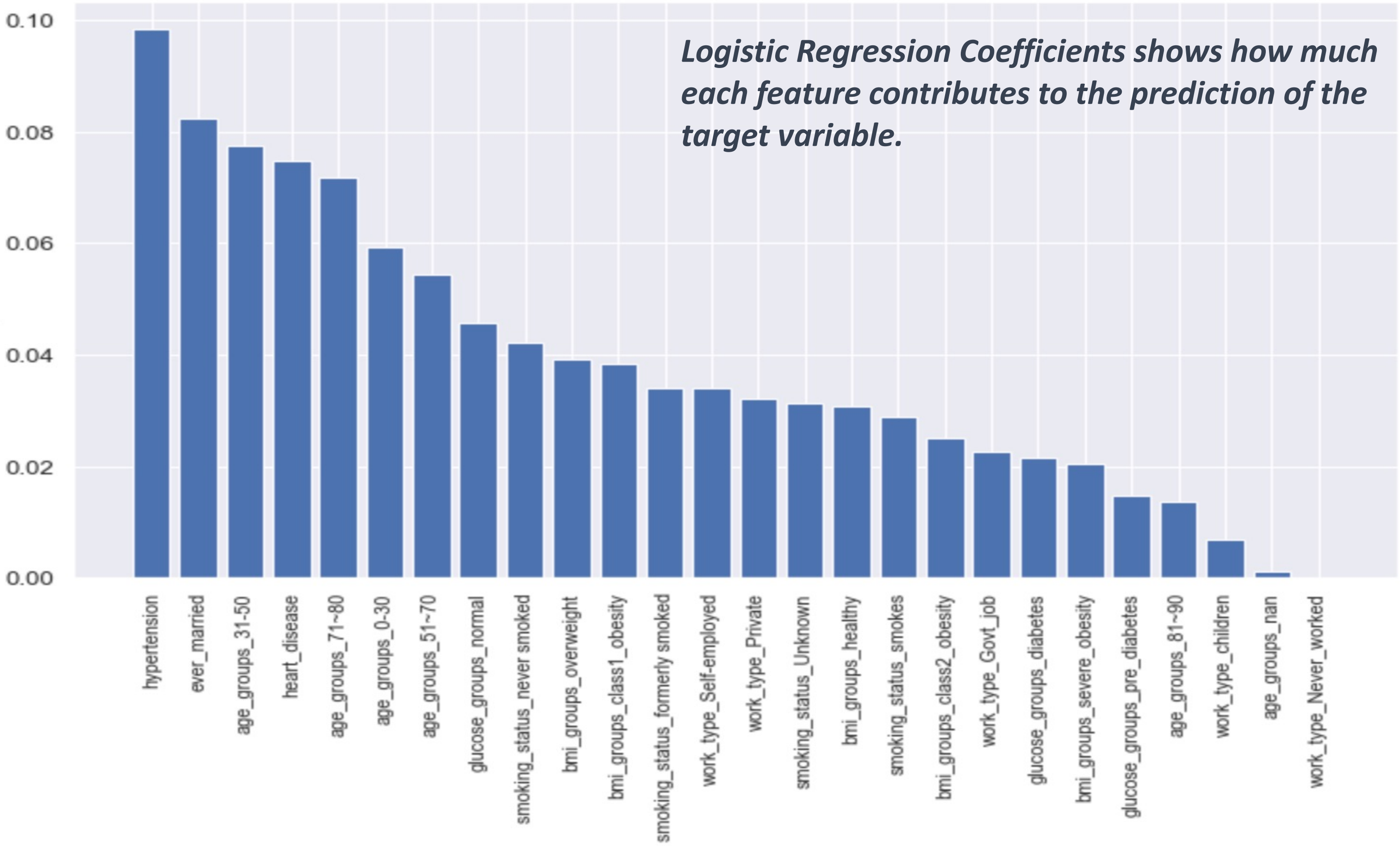
****After SMOTE, logistic regression still scored the highest accuracy 93.90, +0.26 compared to the model before SMOTE.***

Final Model

Final Model Summary

Model	Logistic Regression
Target Variable	Stroke (Yes, No)
Input Variable	hypertension, heart_disease, ever_married, work_type, smoking_status, age_groups, bmi_groups, glucose_groups
Accuracy	**93.90**
Major Coefficients	1) Hypertension 2) ever_married 3) age_group_31~50 4) heart_disease 5) age_group_71~80

Logistic Regression Coefficients (sorted by absolute value)



Logistic Regression Coefficients shows how much each feature contributes to the prediction of the target variable.

Final Model

Stroke Data

- 5,110 Observations
- 11 Input Variables, 1 Output Variable

Preprocessing

Nulls, outliers, anomalies

Upsampling

SMOTE

- Imbalanced data (Stroke Case: 5%)
- Synthetic observation of the minority class

Feature Engineering

- From 11 to 8 Input Variables

Binning / Categorizing

- Age, Obesity, Diabetes Group

One-hot Encoding

Feature Selection

Feature Importance

P-value

Correlation

- Major Features: Individuals who are married, aged 31-50 or 71-80, and have heart disease

Modeling

Linear Classifier

Non-Linear Classifier

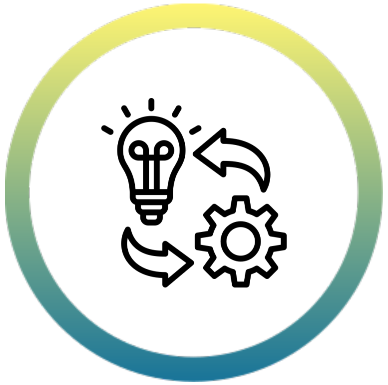
Logistic Regression

Random Forest

- Computational efficiency
- Model performance (ROC-AUC)

Final Model: Logistic Regression | Stroke Predictive Accuracy: 93.90

Recommendations



Current Model

- Model: Logistic Regression
- Predictive Accuracy: 93.74
- Features: 8
- Feature Importance
  1. Hypertension
  2. Marital status
  3. Age Group
  4. Heart Disease History

Model Improvement

- Hyperparameter tuning (Grid search, Random search, Bayesian optimization…)
- Ensembling (bagging, boosting…)
- Collect more data (stroke case) to resolve imbalance

Model Fine-Tuning

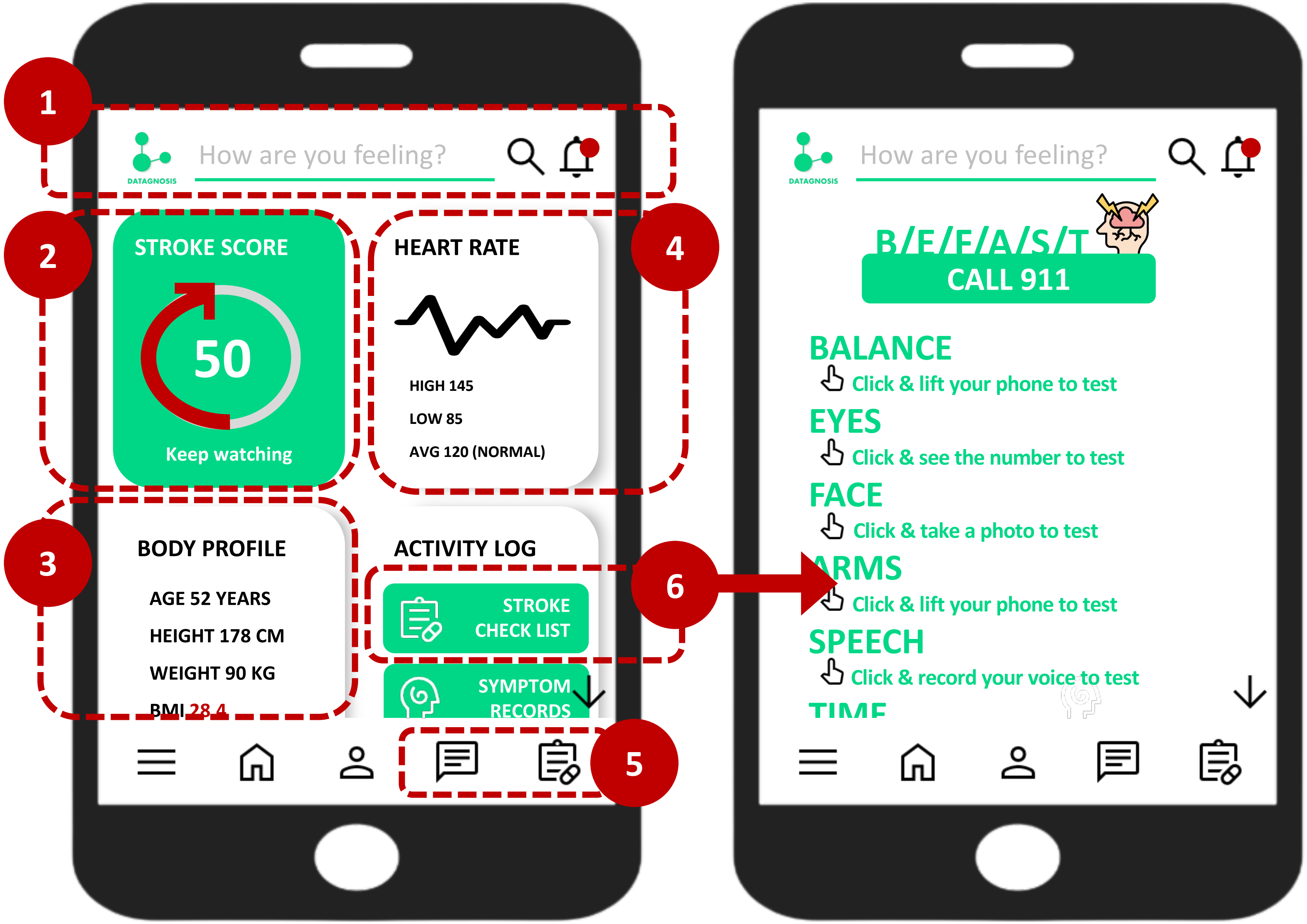
- Monitoring performance
- Regular retraining
- Updating the algorithm
- Regular code reviews
- Mitigating overfitting

Business Implementation

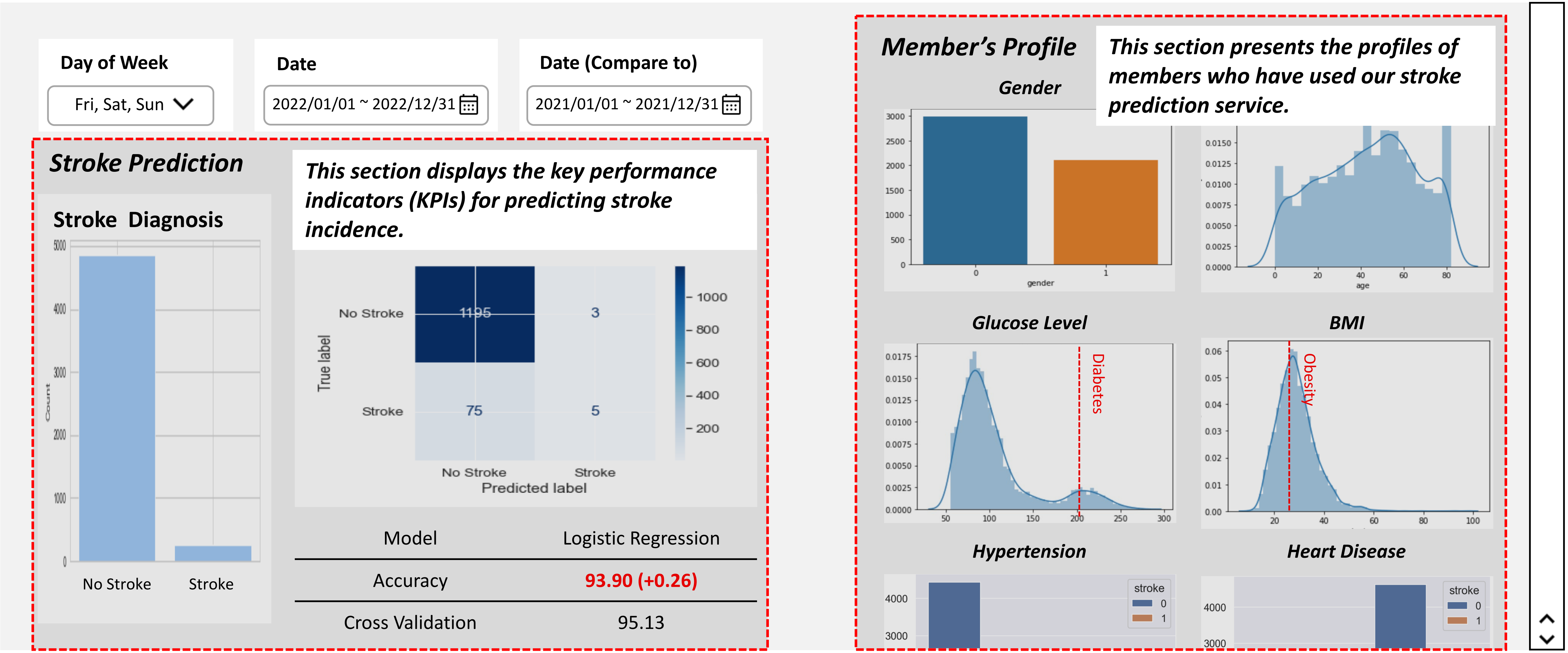
- Investment in app development based on our product prototype
- Extra health care services for cohorts that are more likely to get heart strokes

UI/UX MOCKUPs

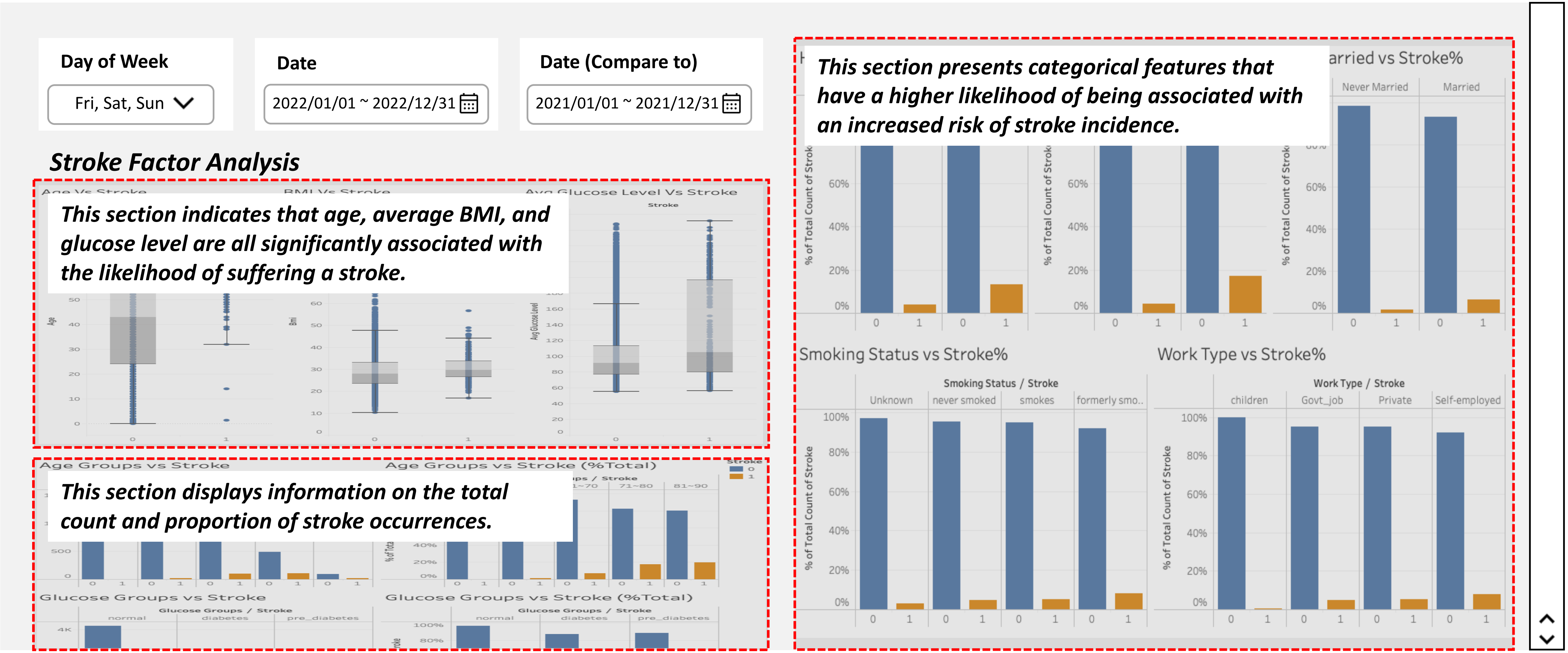
1. Enter Symptoms directly
2. Calculate the stroke score and initiate alert
3. Capture body profile and document family history
4. Document body measurements and physical activities
5. Consult with a doctor and prescribe medication
6. Utilize motion recognition on a mobile phone to diagnose a stroke



Stroke Analysis Dashboard



Stroke Analysis Dashboard



Thank You

References

1. Dataset
   - https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset
2. Glucose-Level Interpretation
   - https://www.cdc.gov/diabetes/basics/getting-tested.html
3. BMI Interpretation
   - https://www.nhs.uk/common-health-questions/lifestyle/what-is-the-body-mass-index-bmi/
4. UI/UX Mockup Design
   - https://shakuro.com/blog/how-to-design-a-healthcare-app-that-makes-its-users-happier
5. Dashboard
   - https://samples.boldbi.com/solutions/healthcare/patient-experience-analysis-dashboard