Health Analytics Project

Stroke Prediction:
Early prediction and intervention

Zijun Wu

Dec 2022

Agenda

Problem Statement

Assumptions & Hypotheses about Data

Data Engineering and Exploratory Data Analysis

Survey of Existing Solutions

Feature engineering

Analytical Models

Proposed Solution and Model Selection

Model Performance Expectation for New Population Cohort

Model Comparison with Existing Solution

Health Care Impact - Real World

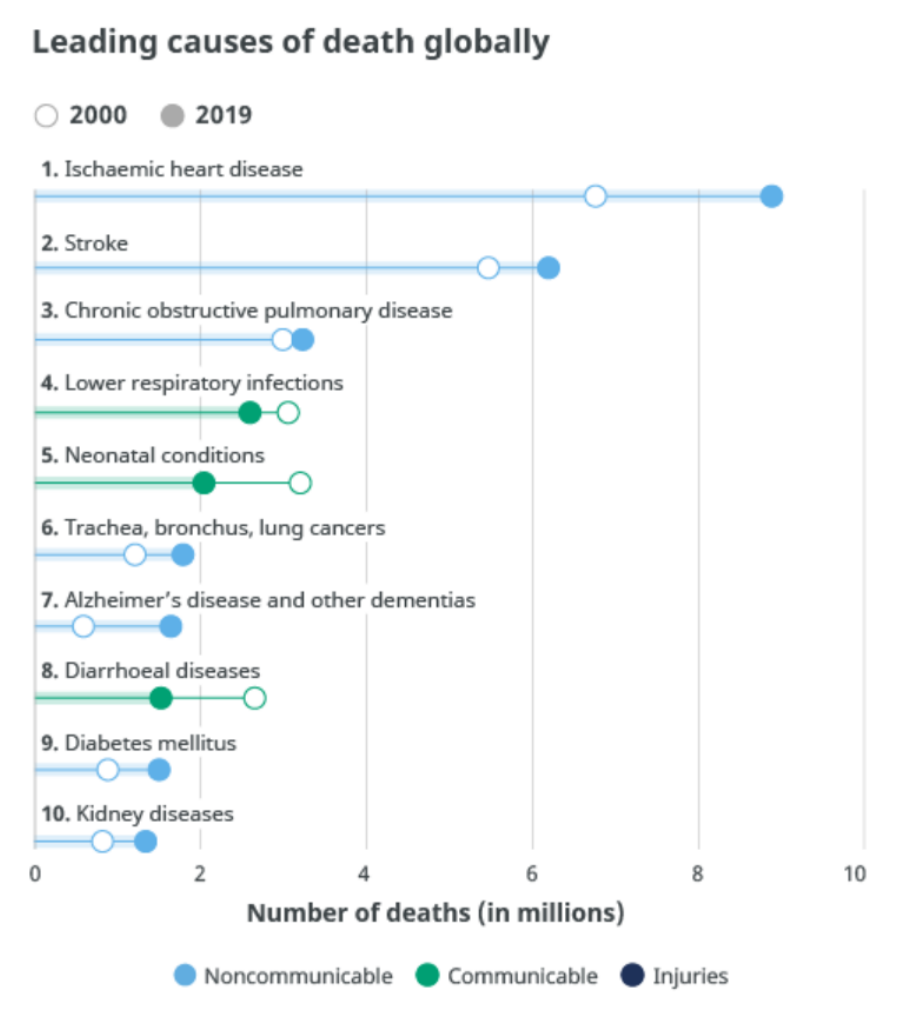
Solution Weaknesses and Future Improvement

Future Work (Other Models or Solutions)

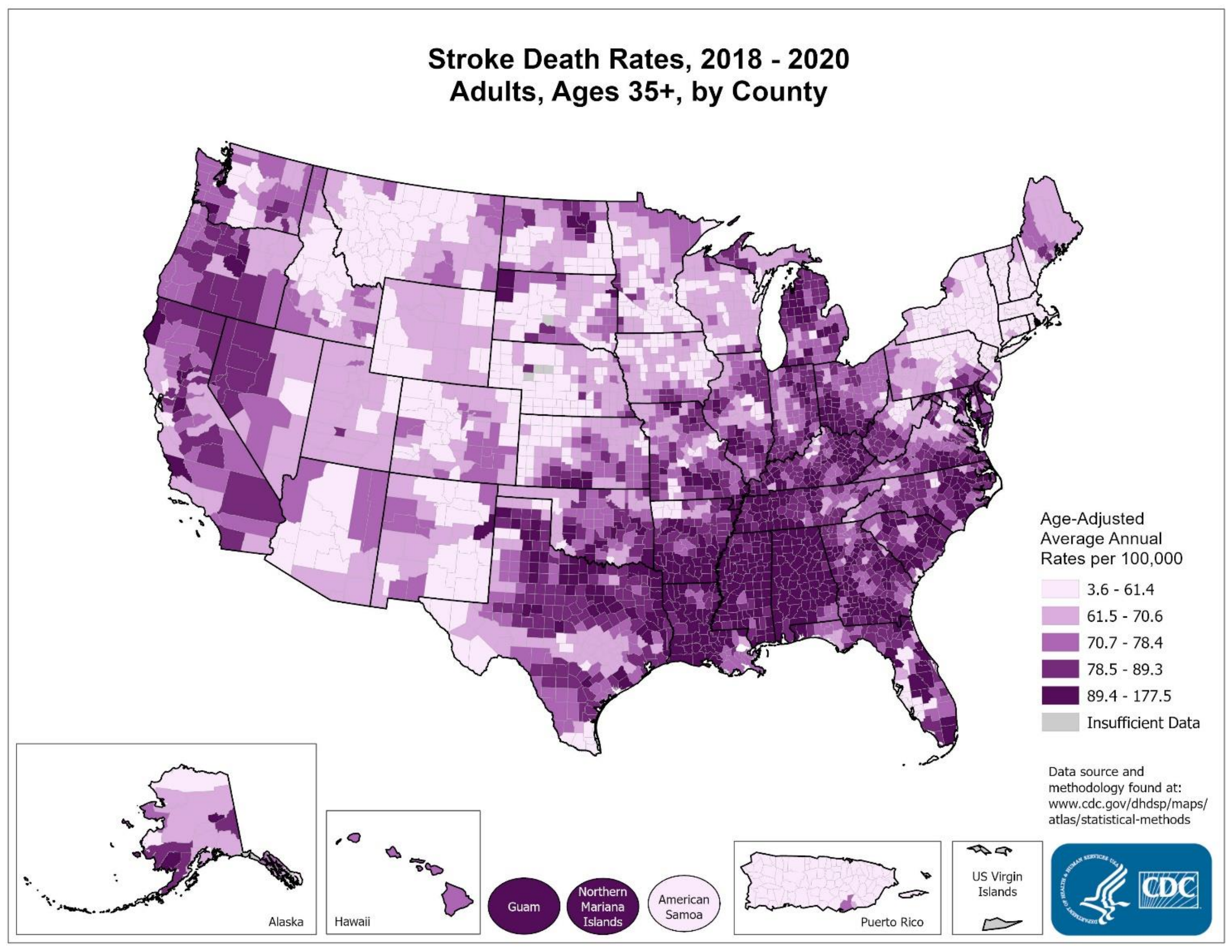
Problem Statement

- According to the World Health Organization (WHO), stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths.

Disease Ranking



Geographical Distribution – The Stroke Belt



Assumptions & Hypotheses about Data

Assumptions

No multicollinearity among independent variables

Large sample size to predict properly

Logistic Regression: Lack of strongly influential outliers

Random Forest: Data is distributed normally

Classification Models

1. KNN
2. Logistic Regression
3. Random Forest

Exploratory Data Analysis

Data Overview 1 – Feature Information

Stroke Prediction Dataset

Attribute Information
1) id: unique identifier
2) gender: "Male", "Female" or "Other"
3) age: age of the patient
4) hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
5) heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
6) ever_married: "No" or "Yes"
7) work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
8) Residence_type: "Rural" or "Urban"
9) avg_glucose_level: average glucose level in blood
10) bmi: body mass index
11) smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"*
12) stroke: 1 if the patient had a stroke or 0 if not
*Note: "Unknown" in smoking_status means that the information is unavailable for this patient

Data Profile

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
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)
memory usage: 479.2+ KB
```

- 1 key column
- 3 numeric column
- 8 categorical columns

Exploratory Data Analysis

Data Overview 2 - Data Engineering - Interpolate

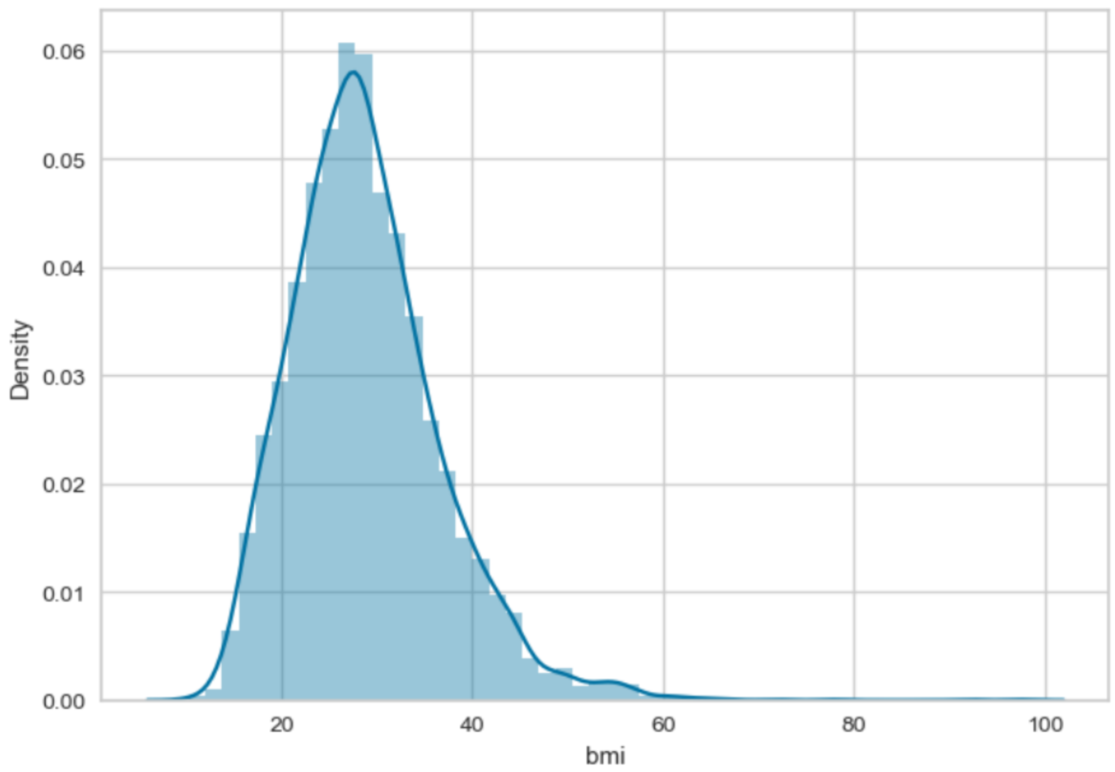
Interpolate Result

```
print("mean change: " + str(28.92728 - 28.893237))

print("std change: " + str(7.77531 - 7.854067))

# good interpolation result
```

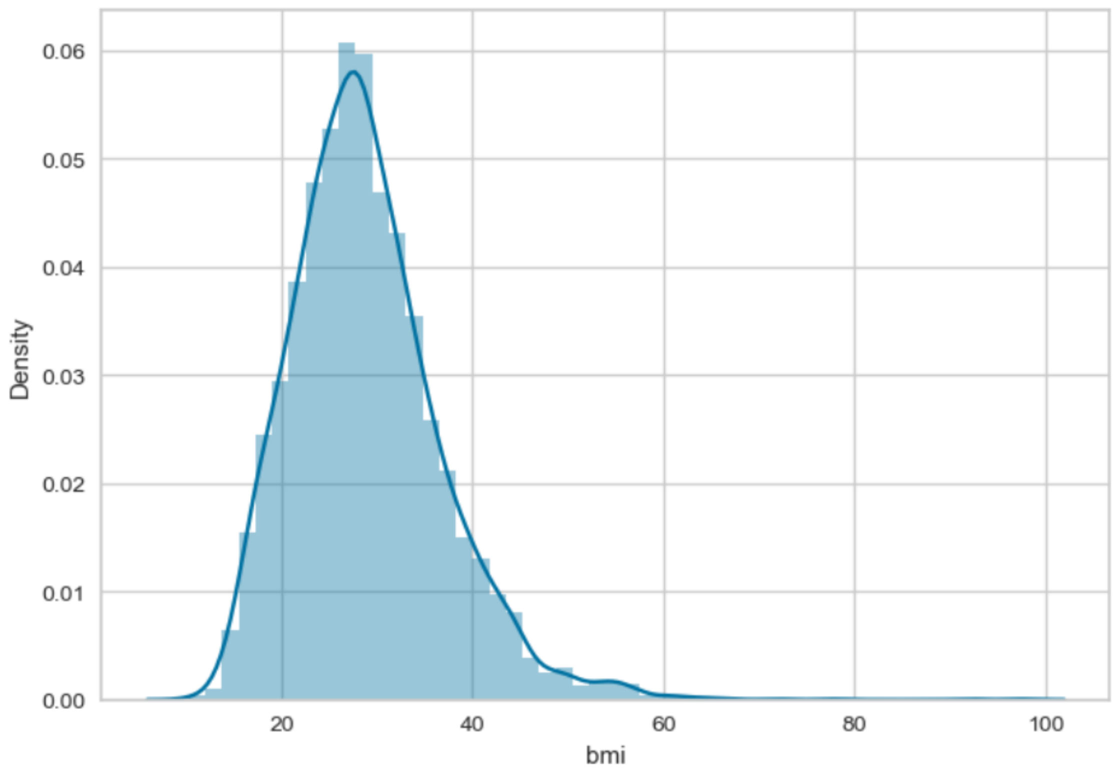
```
mean change: 0.03404300000000049
std change: -0.07875699999999952
```

- Negligible change in mean and standard deviation
- Maintain distribution shape

BMI before interpolate



BMI after interpolate



Exploratory Data Analysis

Data Overview 3 – Statistics, Correlation & Heatmap

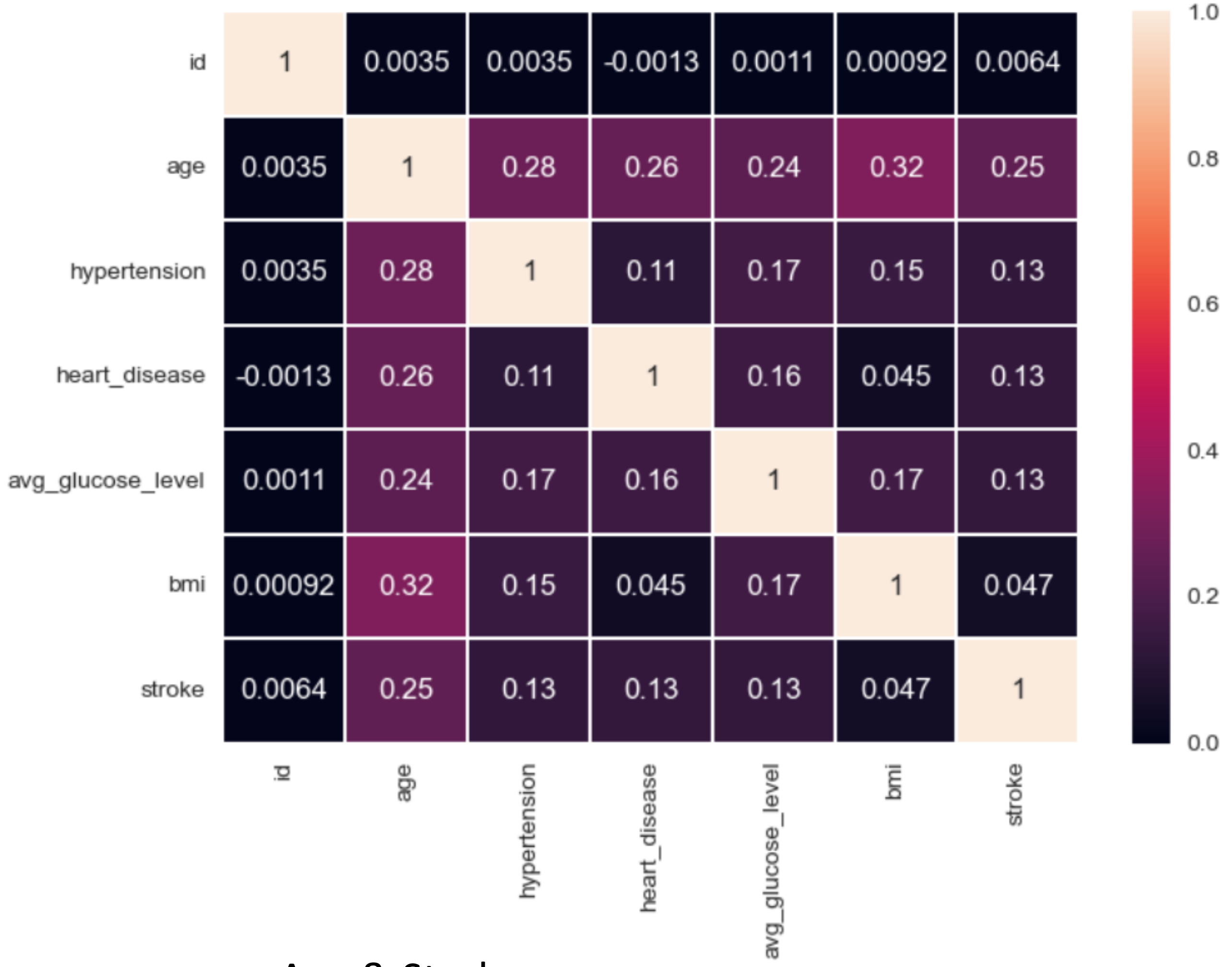
Data Summary Statistics

	count	mean	std	min	25%	50%	75%	max
id	5110.0	36517.829354	21161.721625	67.00	17741.250	36932.000	54682.00	72940.00
age	5110.0	43.226614	22.612647	0.08	25.000	45.000	61.00	82.00
hypertension	5110.0	0.097456	0.296607	0.00	0.000	0.000	0.00	1.00
heart_disease	5110.0	0.054012	0.226063	0.00	0.000	0.000	0.00	1.00
avg_glucose_level	5110.0	106.147677	45.283560	55.12	77.245	91.885	114.09	271.74
bmi	5110.0	28.927280	7.775310	10.30	23.600	28.100	33.10	97.60
stroke	5110.0	0.048728	0.215320	0.00	0.000	0.000	0.00	1.00

correlation table

	id	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
id	1.000000	0.003538	0.003550	-0.001296	0.001092	0.000925	0.006388
age	0.003538	1.000000	0.276398	0.263796	0.238171	0.321631	0.245257
hypertension	0.003550	0.276398	1.000000	0.108306	0.174474	0.149985	0.127904
heart_disease	-0.001296	0.263796	0.108306	1.000000	0.161857	0.044599	0.134914
avg_glucose_level	0.001092	0.238171	0.174474	0.161857	1.000000	0.168539	0.131945
bmi	0.000925	0.321631	0.149985	0.044599	0.168539	1.000000	0.047351
stroke	0.006388	0.245257	0.127904	0.134914	0.131945	0.047351	1.000000

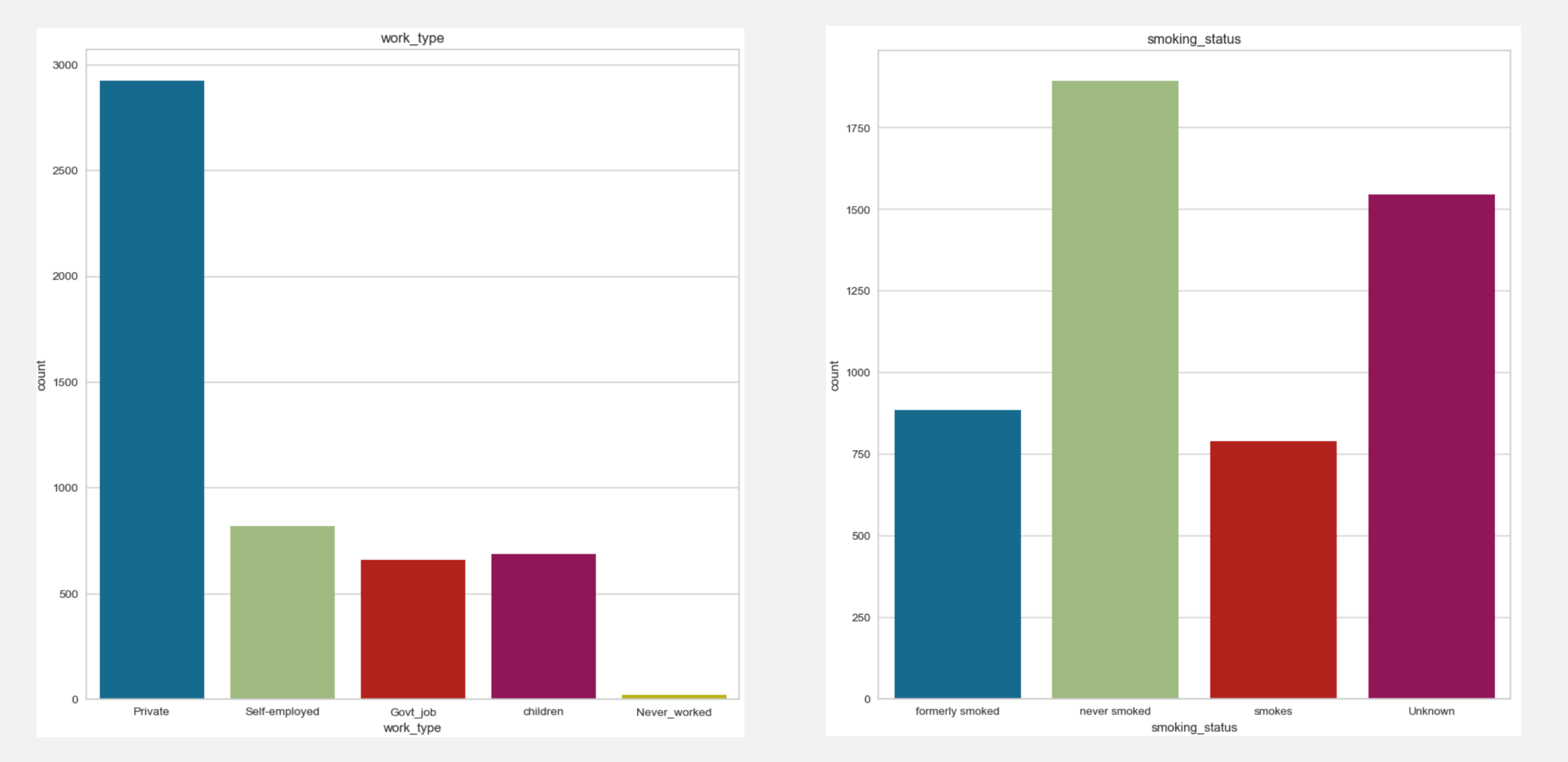
Heatmap



- Age & Stroke

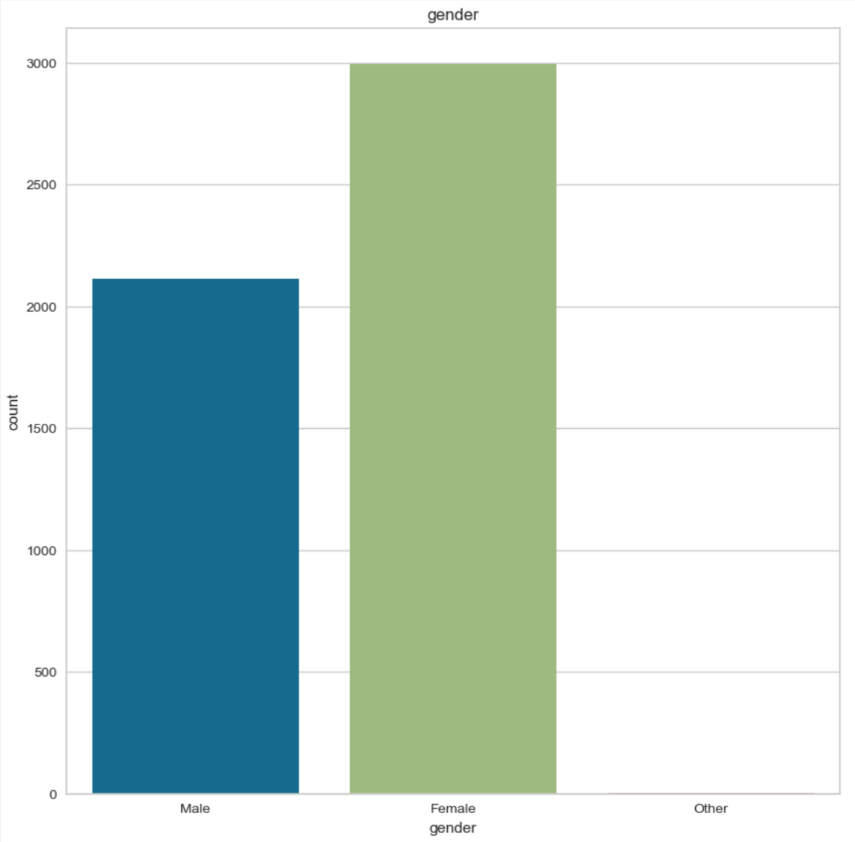
Exploratory Data Analysis

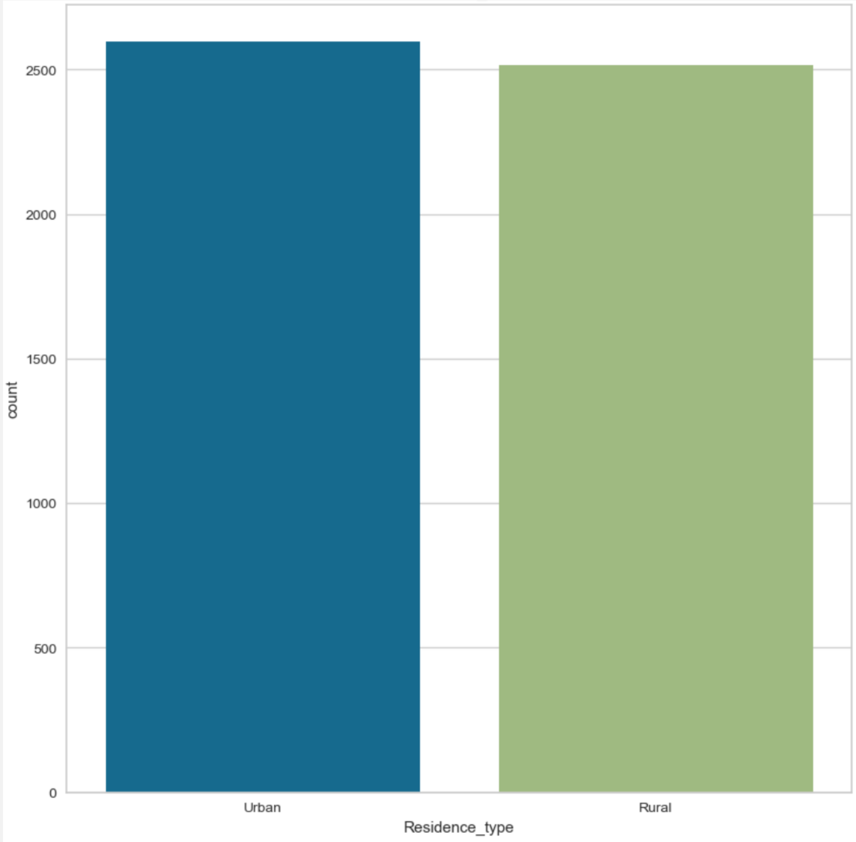
Data Visualization 1 – Multiple Categories

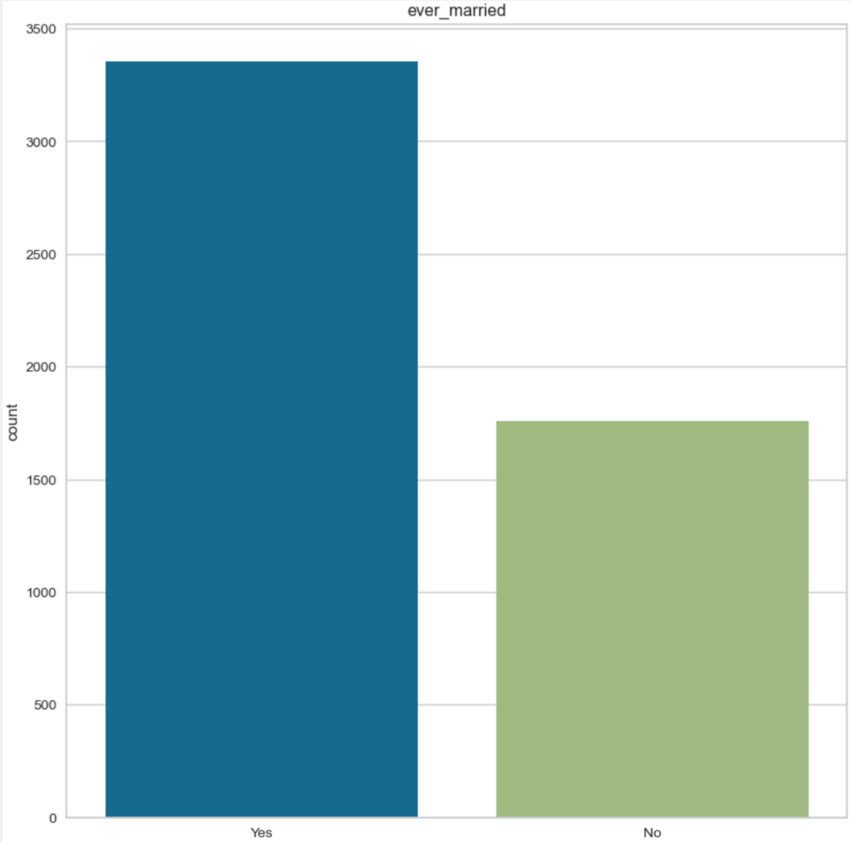


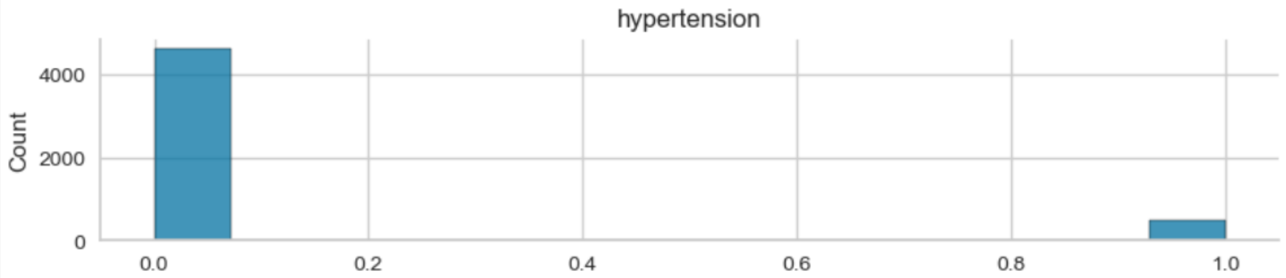
Exploratory Data Analysis

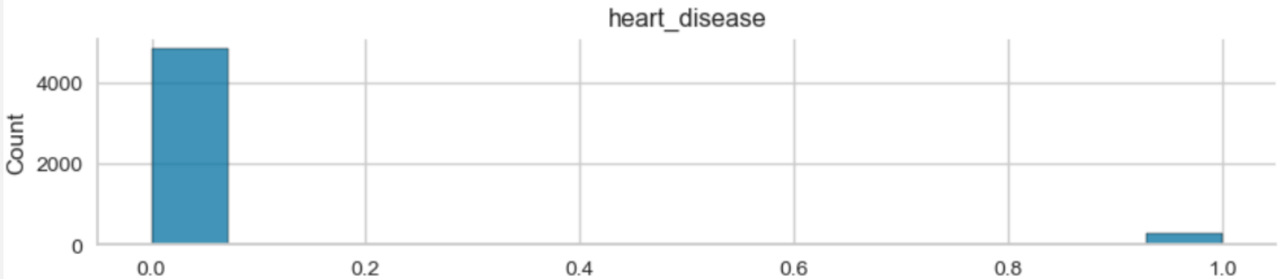
Data Visualization 2 – Binary Categorical Column





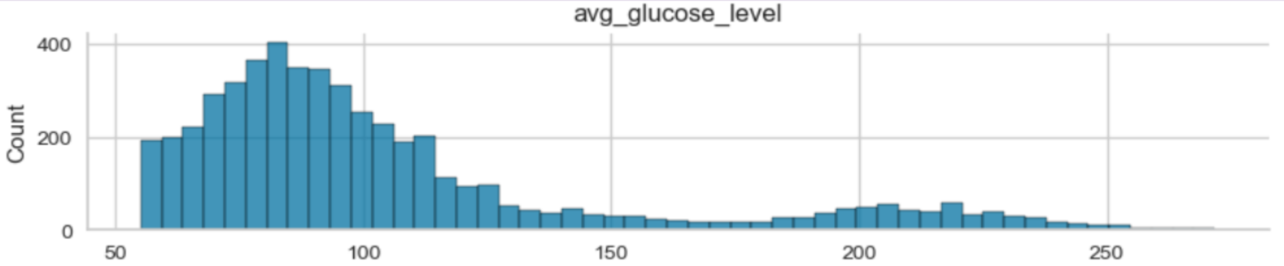


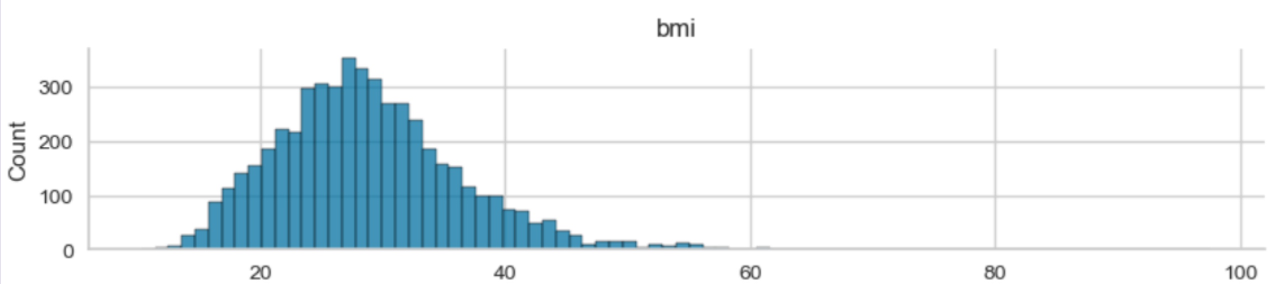


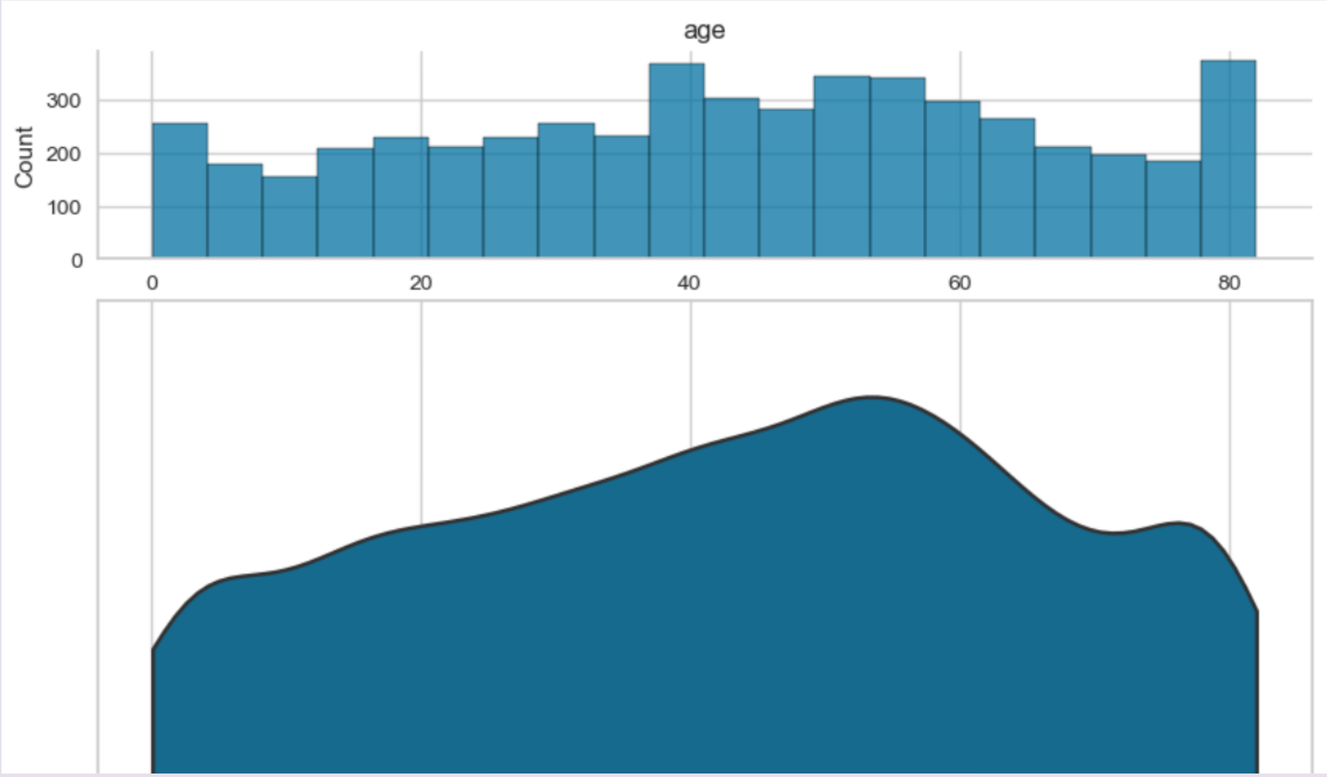


Exploratory Data Analysis

Data Visualization 3 – Numerical columns



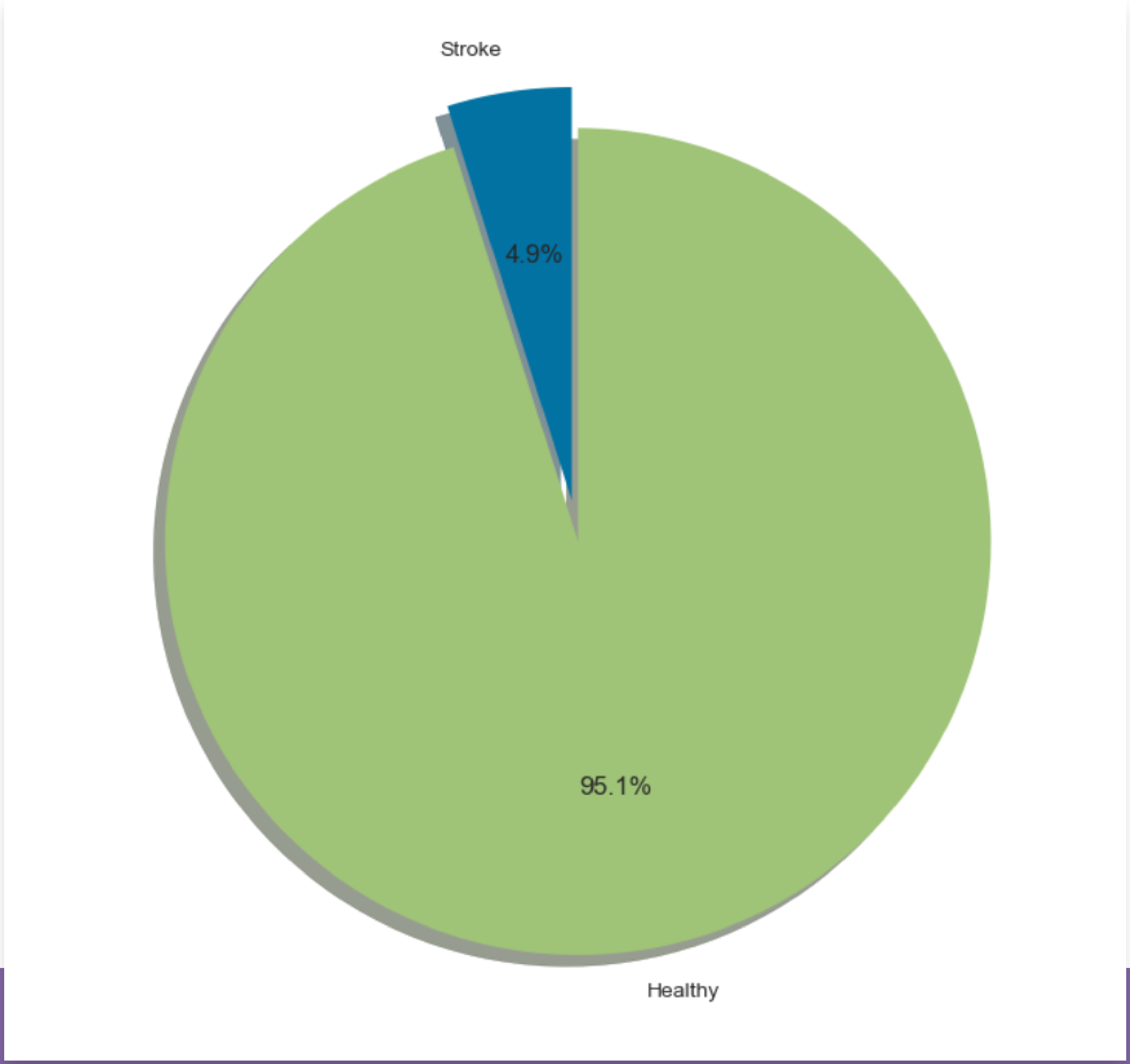




Exploratory Data Analysis

Data Visualization 4 – target variable

- Only 4.9% patient have stroke disease
- Stroke proportion highly imbalance



Survey of Existing Solution

Stroke Prediction Kaggle Project Using Same Dataset

Reason:

- 4.9% stroke cases are not captured evenly in both training set and test set.
- The model is not learning the pattern effectively for stroke cases.

Model Results:

- High Accuracy (94%)
- High precision & recall for non-stroke cases
- No precision & recall for stroke cases

Logistic Regression

```
Confusion Matrix :

[[929    0]
 [ 53    0]]

Classification Report :

              precision    recall  f1-score   support

           0       0.95      1.00      0.97       929
           1       0.00      0.00      0.00        53

    accuracy                           0.95       982
   macro avg       0.47      0.50      0.49       982
weighted avg       0.89      0.95      0.92       982

The Accuracy of Logistic Regression is 94.6 %
```

Random Forest

```
Confusion Matrix :

[[929    0]
 [ 53    0]]

Classification Report :

              precision    recall  f1-score   support

           0       0.95      1.00      0.97       929
           1       0.00      0.00      0.00        53

    accuracy                           0.95       982
   macro avg       0.47      0.50      0.49       982
weighted avg       0.89      0.95      0.92       982

The Accuracy of Random Forest Classifier is 94.6 %
```

Feature Engineering

Normalization

- Original DataFrame

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.60	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	34.55	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.50	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.40	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.00	never smoked	1

- Standard scaler: Normalized **age**, **avg_glucose_level** and **bmi**.

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	1.051434	0	1	Yes	Private	Urban	2.706375	0.986902	formerly smoked	1
1	51676	Female	0.786070	0	0	Yes	Self-employed	Rural	2.121559	0.723221	never smoked	1
2	31112	Male	1.626390	0	1	Yes	Private	Rural	-0.005028	0.459540	never smoked	1
3	60182	Female	0.255342	0	0	Yes	Private	Urban	1.437358	0.703928	smokes	1
4	1665	Female	1.582163	1	0	Yes	Self-employed	Rural	1.501184	-0.633770	never smoked	1

Feature Engineering

Feature Creation & Transformations

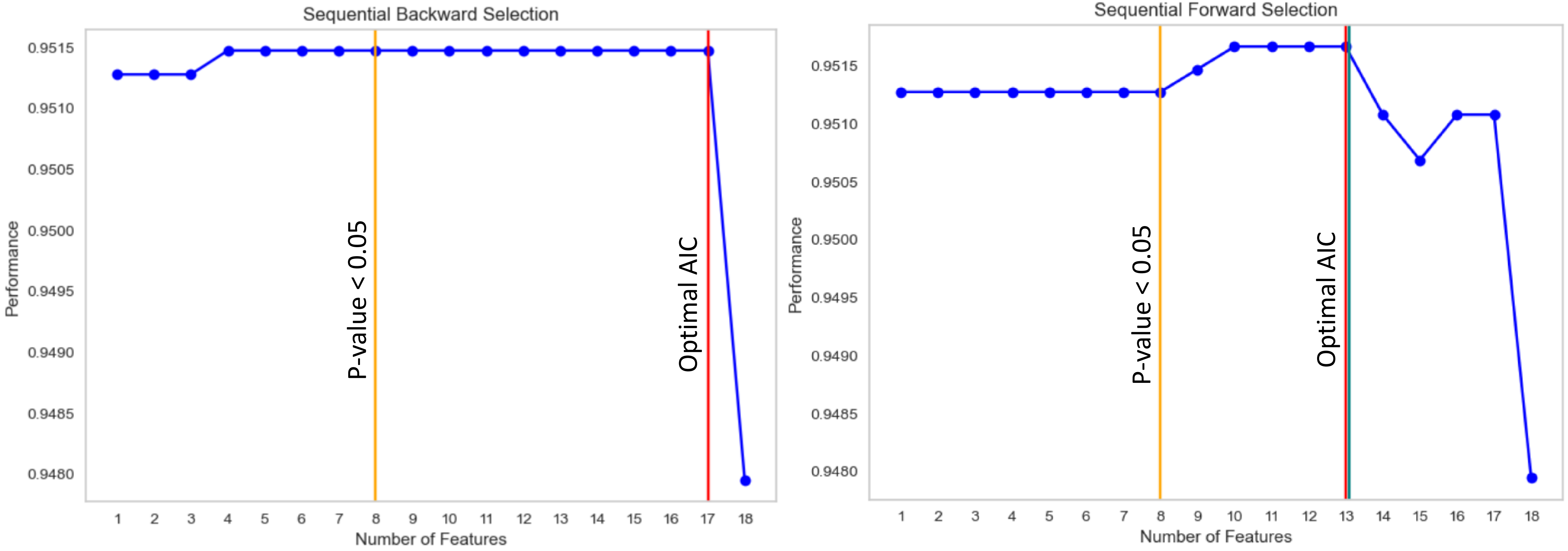
- Convert categorical features to binary numeric columns

	Data Type	Nulls	Zeros	Min	Median	Max	Mean	Standard Deviation	Unique	Top Frequency
id	int64	0	0	67	36932	72940	36517.83	21159.65	5110	1
age	float64	0	0	-2	0.078	1.71	0.000000000000000050	1.00	104	102
hypertension	int64	0	0	0	0	1	0.097	0.30	2	4612
heart_disease	int64	0	0	0	0	1	0.054	0.23	2	4834
avg_glucose_level	float64	0	0	-1	-0	3.66	0.00000000000000010	1	3979	6
bmi	float64	0	0	-2	-0	8.83	0.00000000000000025	1	520	41
stroke	int64	0	0	0	0	1	0.049	0.22	2	4861
male	int64	0	0	0	0	1	0.41	0.49	2	2995
married	int64	0	0	0	1	1	0.66	0.47	2	3353
private	int64	0	0	0	1	1	0.57	0.49	2	2925
self_employed	int64	0	0	0	0	1	0.16	0.37	2	4291
children	int64	0	0	0	0	1	0.13	0.34	2	4423
govt_job	int64	0	0	0	0	1	0.13	0.33	2	4453
never_worked	int64	0	0	0	0	1	0.0043	0.065	2	5088
urban	int64	0	0	0	1	1	0.51	0.50	2	2596
never_smoked	int64	0	0	0	0	1	0.37	0.48	2	3218
formerly_smoked	int64	0	0	0	0	1	0.17	0.38	2	4225
smokes	int64	0	0	0	0	1	0.15	0.36	2	4321
unknown_smoke	int64	0	0	0	0	1	0.30	0.46	2	3566

Feature Engineering

Feature Selection: Backward vs. Forward (Joint predictive ability)

Choose 15 features for trade-off between p-values and maximum AIC



Explanatory Variables

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5110 entries, 9046 to 44679
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   age                5110 non-null   float64
 1   heart_disease      5110 non-null   int64
 2   avg_glucose_level  5110 non-null   float64
 3   hypertension       5110 non-null   int64
 4   married            5110 non-null   int64
 5   formerly_smoked    5110 non-null   int64
 6   self_employed      5110 non-null   int64
 7   bmi                5110 non-null   float64
 8   urban              5110 non-null   int64
 9   private            5110 non-null   int64
 10  male               5110 non-null   int64
 11  smokes             5110 non-null   int64
 12  govt_job           5110 non-null   int64
 13  never_smoked       5110 non-null   int64
 14  unknown_smoke      5110 non-null   int64
dtypes: float64(3), int64(12)
memory usage: 638.8 KB
```

Response Variable

```
<class 'pandas.core.series.Series'>
Int64Index: 5110 entries, 9046 to 44679
Series name: stroke
Non-Null Count  Dtype
--------------  -----
5110 non-null   int64
dtypes: int64(1)
memory usage: 79.8 KB
```

Analytical Models

Model Selection	Reasons
Logistic Regression	• Easy to implement, interpret, and very efficient to train
Random Forest	• Aggregate many decision trees to limit overfitting as well as error due to bias • Robust to outliers and less affected by noise
KNN	• No assumptions about data

Proposed Solution and Model Selection

Classification Report, Confusion Matrix, roc_auc_score, recall_score, brier score

brier1_KNN	brier1_LogisticRegression	brier_RandomForest
0.046036	0.040782	0.043324

$$Recall = \frac{TP}{TP + FN}$$

Logistic Regression

```
Classification Report for LR Model:

              precision    recall  f1-score   support

           0       0.98      0.86      0.92       972
           1       0.21      0.72      0.32        50

    accuracy                           0.85      1022
   macro avg       0.60      0.79      0.62      1022
weighted avg       0.95      0.85      0.89      1022


Confusion Matrix for LR Model:

 [[835 137]
 [ 14  36]]

roc_auc_score for LR Model:

 0.7895267489711935

recall_score for LR Model:

 0.72
```

Random Forest

```
Classification Report for RF Model:

              precision    recall  f1-score   support

           0       0.97      0.94      0.95       972
           1       0.26      0.40      0.31        50

    accuracy                           0.91      1022
   macro avg       0.61      0.67      0.63      1022
weighted avg       0.93      0.91      0.92      1022


Confusion Matrix for RF Model:

 [[915  57]
 [ 30  20]]

roc_auc_score for RF Model:

 0.670679012345679

recall_score for RF Model:

 0.4
```

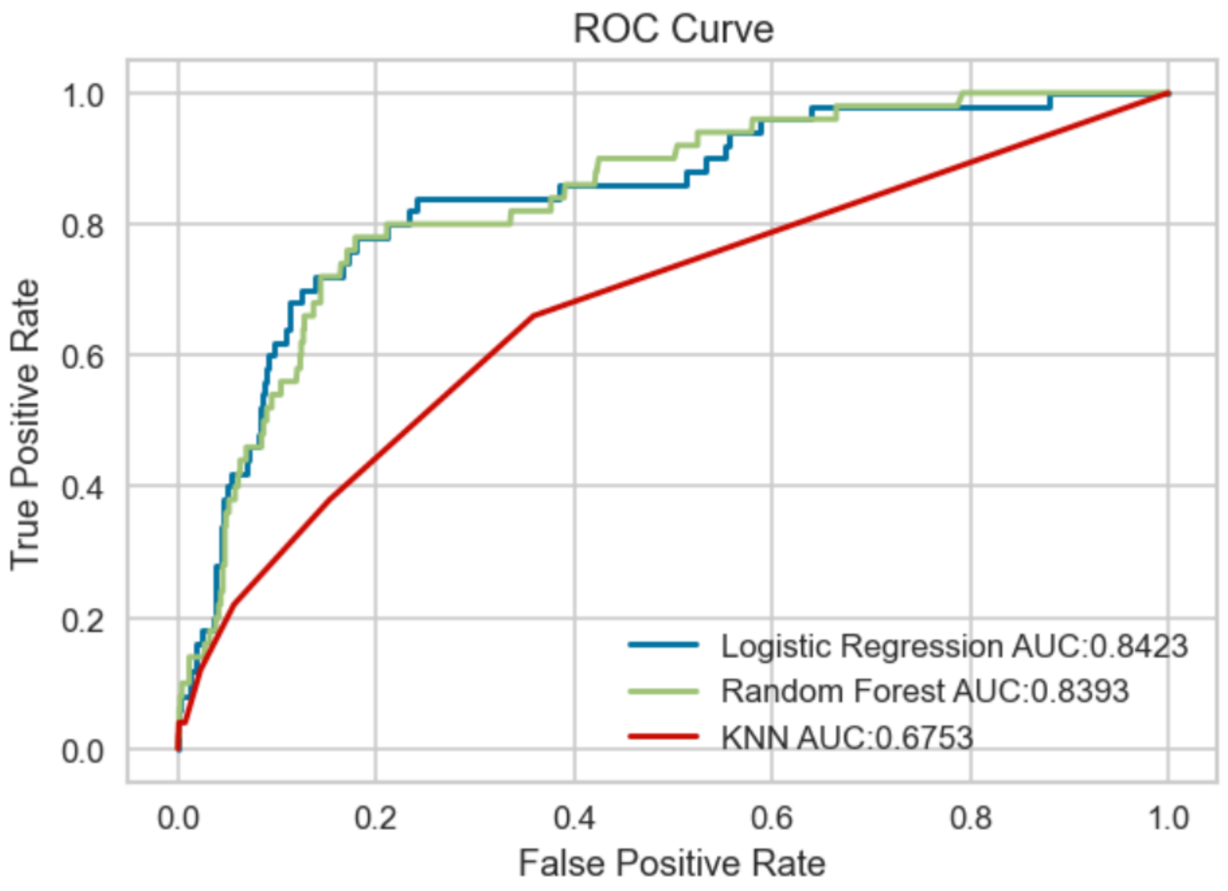
KNN

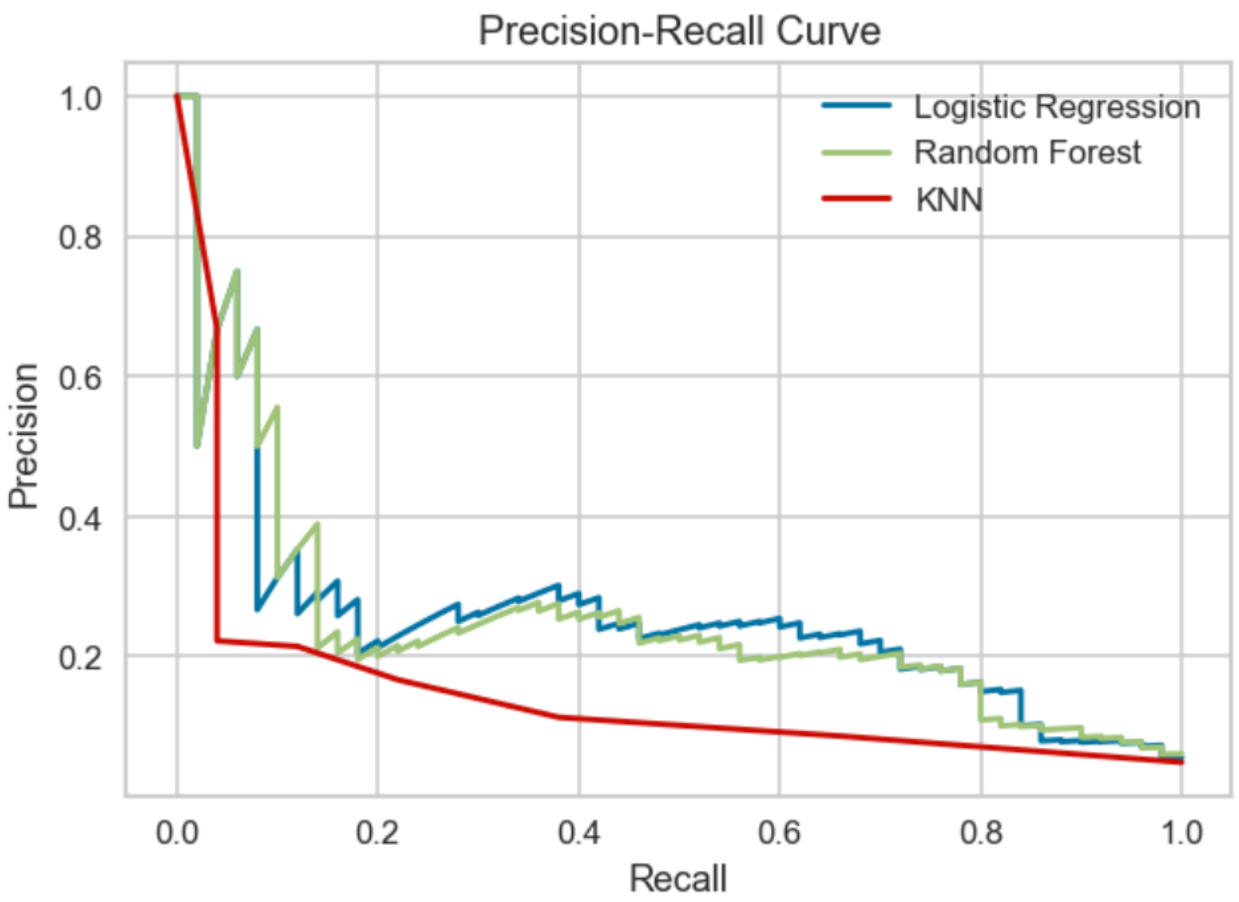
```
Classification Report for KNN Model:

              precision    recall  f1-score   support

           0       0.96      0.94      0.95       972
           1       0.17      0.22      0.19        50

    accuracy                           0.91      1022
   macro avg       0.56      0.58      0.57      1022
weighted avg       0.92      0.91      0.91      1022


Confusion Matrix for KNN Model:

 [[917  55]
 [ 39  11]]

roc_auc_score for KNN Model:

 0.5817078189300412

recall_score for KNN Model:

 0.22
```

Proposed Solution and Model Selection

Logistic regression model is the best-performing model

- Roc-Auc Curve
- Precision-Recall Curve





Proposed Solution and Model Selection

Logistic Regression model

- Highest AOC-ROC performance: 0.84
- Highest F1 score
- Relatively high precision and recall trade-off
- Lowest brier score: 0.04

brier1_KNN	brier1_LogisticRegression	brier_RandomForest
0.046036	0.040782	0.043324

```
Classification Report for LR Model:

              precision    recall  f1-score   support

           0       0.98      0.86      0.92       972
           1       0.21      0.72      0.32        50

    accuracy                           0.85      1022
   macro avg       0.60      0.79      0.62      1022
weighted avg       0.95      0.85      0.89      1022


Confusion Matrix for LR Model:

 [[835 137]
 [ 14  36]]

roc_auc_score for LR Model:

 0.7895267489711935

recall_score for LR Model:

 0.72
```

$$Recall = \frac{TP}{TP + FN}$$

Model Performance Expectation for New Population Cohort

Can this model be used out-of-the-box for a new population cohort and why?

Reason 1

Train-test-split stratified will adjust proportion of stroke cases in train and test accordingly and automatically.

Reason 2

High recall on stroke cases give model great generalization ability on detecting potential stroke cases.

Reason 3

y_pred probability threshold test out the most appropriate cutoff for new model stroke probability.

Strategy[1]

Stratified Ratio

Train-test-split stratified will keep the same proportion of data for train dataset and test dataset

High Recall

Higher recall minimize the false negative cases and avoid the risk of not detecting probably cases or delay treatment.

Probability Threshold

Different probability cutoff from 0 to 1 with 0.5 as each step to test out optimal stroke probability

Model Comparison with Existing Solution

New Logistic Regression model:

- Higher precision-recall for stroke cases
- Higher F1 score
- Lower false negative cases

Survey Solution

Logistic Regression

```
Confusion Matrix :

[[929    0]
 [ 53    0]]

Classification Report :

              precision    recall  f1-score   support

           0       0.95      1.00      0.97       929
           1       0.00      0.00      0.00        53

    accuracy                           0.95       982
   macro avg       0.47      0.50      0.49       982
weighted avg       0.89      0.95      0.92       982


The Accuracy of Logistic Regression is 94.6 %
```

New Solution

```
Classification Report for LR Model:

              precision    recall  f1-score   support

           0       0.98      0.86      0.92       972
           1       0.21      0.72      0.32        50

    accuracy                           0.85      1022
   macro avg       0.60      0.79      0.62      1022
weighted avg       0.95      0.85      0.89      1022


Confusion Matrix for LR Model:

 [[835 137]
 [ 14  36]]

roc_auc_score for LR Model:

 0.7895267489711935

recall_score for LR Model:

 0.72
```

Health Care Impact

Stroke Prediction: Early prediction and intervention

Early prediction

Efficiently predict the disease of a human, based on the symptoms and health history.

Medical Resources

Save medical resources and government budget by detecting disease at the early stage.

Early Interventions

Act as an early risk warning for high-risk individuals and a signal to monitor the patients' health conditions.

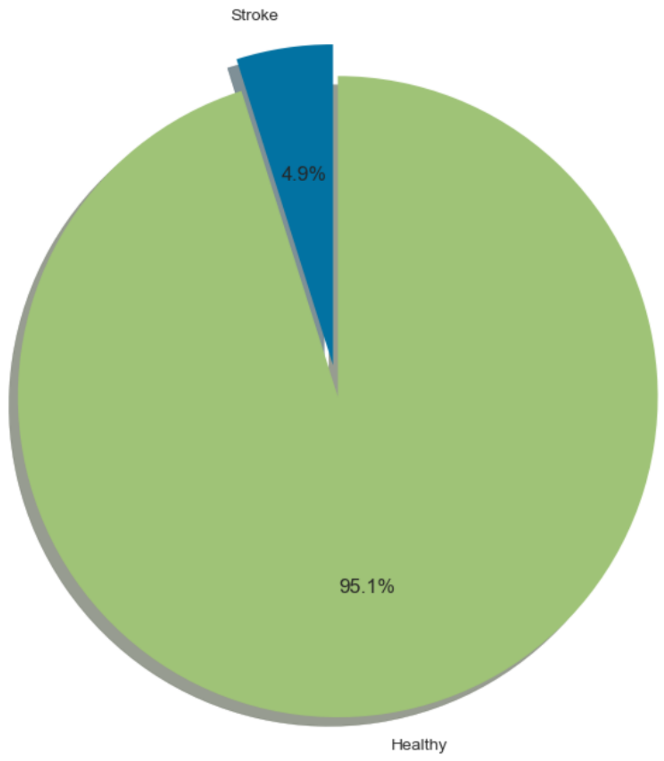
Mortality Rate

Decrease the mortality rate of potential individuals by increasing the prevention awareness of patients and their families.

Solution Weaknesses and Future Improvement

Weakness	• Precision-recall for stroke cases not performing very ideal • Stroke cases are too few for the model to learn
Improvement	1. More data needed (only 4k-5k rows) 2. Very small proportion of people have stroke (5%) 3. Geographical data sampling

Stroke Proportion



Future Work (Other Models or Solutions)

Data Collection

Increase data quantity and generate more data to improve model learning result.

Improve sampling of the data and include more patient sample with stroke disease.

Data & Feature Engineering

Try out other normalization methods on numerical columns.

Transform categorical features into different categories than before by combining similar categories.

Other models

Naive Bayes is easy to implement, highly scalable, and make real-time predictions

XGBoost works well with data that is nonlinear, non-monotonic, or with segregated clusters.

Appendix A: Sources

References

- https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death
- https://www.cdc.gov/stroke/facts.htm
- https://www.verywellhealth.com/united-states-stroke-belt-4068563
- https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset?datasetId=1120859&sortBy=voteCount
- https://www.kaggle.com/code/hasibalmuzdadid/brain-stroke-analysis-accuracy-96-03