Stroke PROGNOSIS

We want to help you build healthy families.

Table of Contents

01.
Our Health Clinic

Introduction & Mission Statement

Introduction

Our Health Clinic uses data-driven solutions to help you prevent, and improve outcomes for diabetes and/or related chronic lifestyle conditions such as high blood pressure, hypertension, stroke, heart diseases, obesity and stress control. Delivering personalized care to all groups and demographics: **Aim to help you build healthy families!**

One such area of use of data-driven solution is Stroke Prognosis using Machine Learning in python, identifying those-at-risk and reducing their risk of suffering stroke through our help in early intervention and lifestyle behavioural change.

Our Services: Chronic Disease Management, Weight Loss & Health Optimization & Health Screening.

Mission

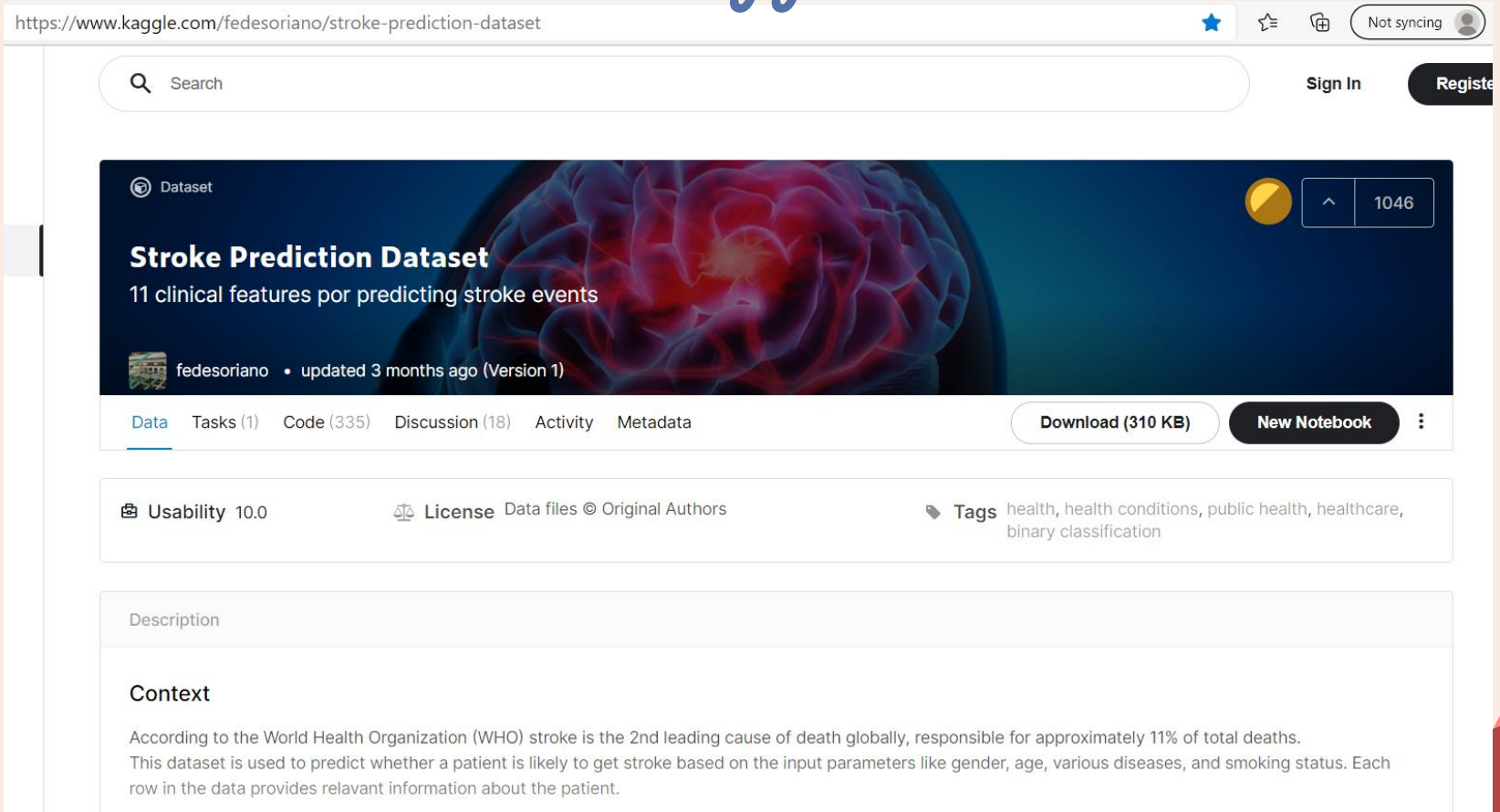
Build A Machine Learning Model To Predict Stroke As Accurate As Possible.

02.

Methodology

Dataset
Models, Metrics, Tools

Methodology : Dataset



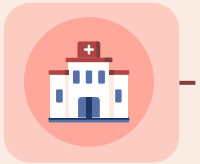
Methodology : Classification Model

1. Model	2. Metrics	3. Tools
Baseline Model: Logistic Regression Alternative Models: Random Forest, Decision Tree, SVM, K-nearest Neighbors, Naïve Bayes	- Precision, Recall, f1-score - Confusion Matrix - False Negative : predicted to be a –ve (non-stroke) when he is a potential +ve.	Scikit-learn, Pandas, NumPy, matplotlib, seaborn, Colab

03.
Process Workflow

Process Workflow

1. EDA & Data Preparation



2. Data Analysis & Feature Engineering

3. Data Transformation

4. Machine Learning Model Training & Evaluation

1. EDA & Data Preparation

Overview

Pandas Profiling Report

Overview | Warnings 2 | Reproduction

Dataset statistics

Number of variables	12
Number of observations	5110
Missing cells	201
Missing cells (%)	0.3%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	479.2 KiB
Average record size in memory	96.0 B

Variable types

Numeric	4
Categorical	7
Boolean	1

Stroke Pandas Profiling Report

1. EDA & Data Preparation

```
df_data.info()
# Observation: there are null values in bmi column
```

```
<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
```

Warnings

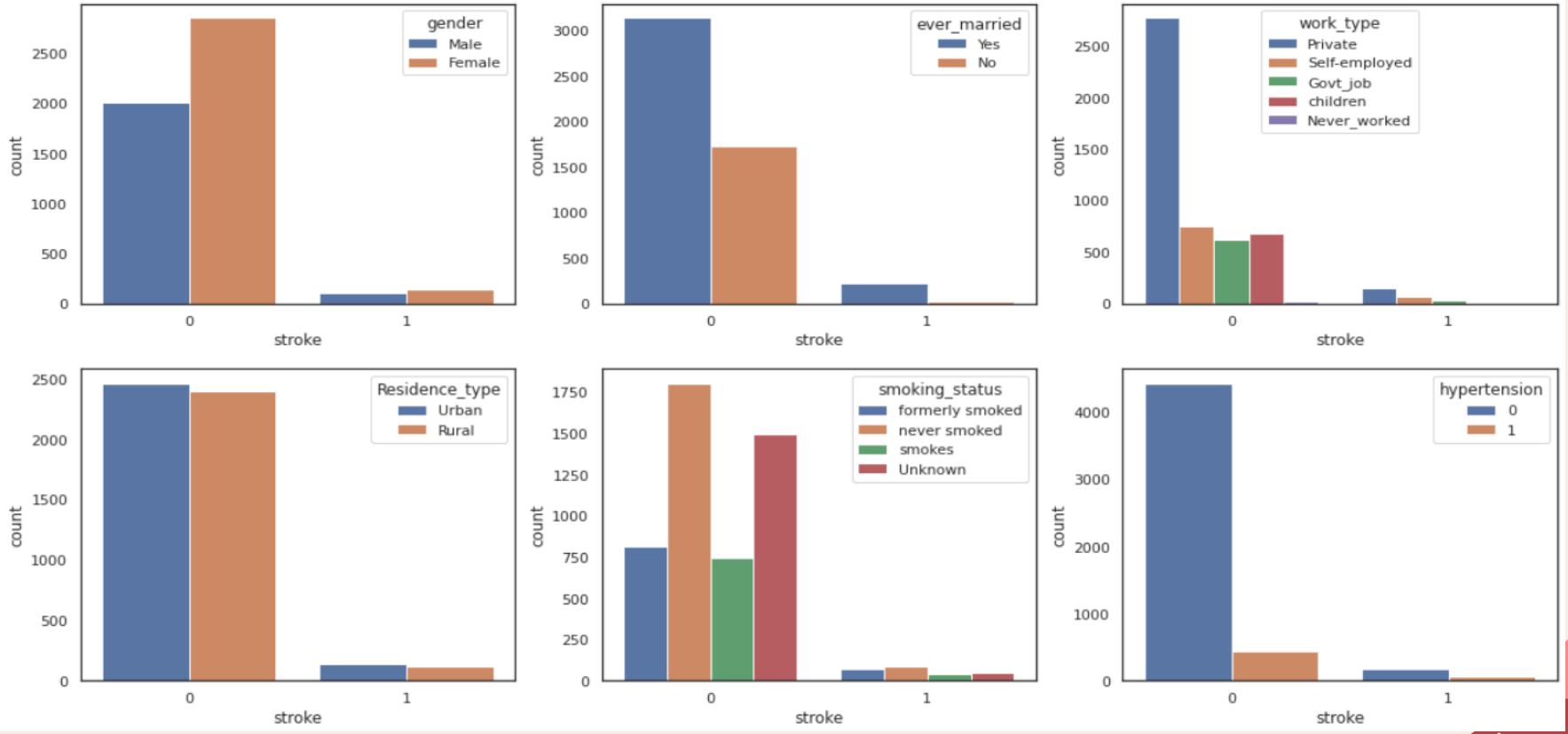
- `bmi` has 201 (3.9%) missing values
- `id` has unique values

1. EDA & Data Preparation

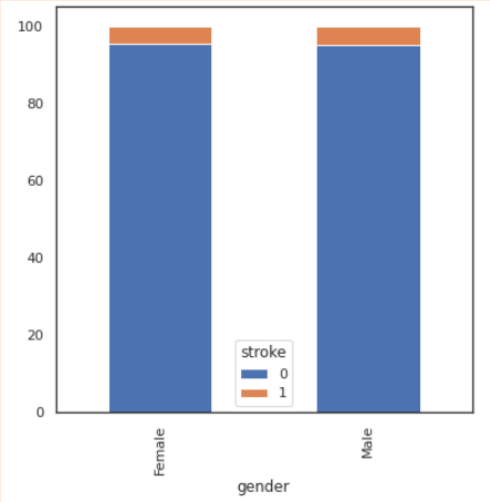
```
df_data.sample(5)
```

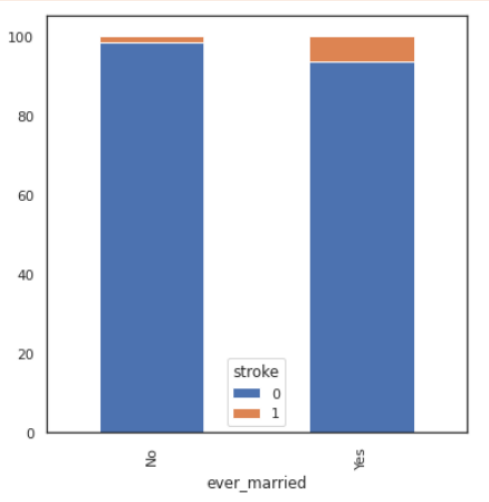
	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
2202	Male	73.0	0	1	Yes	Govt_job	Rural	70.23	28.1	never smoked	0
1620	Female	66.0	0	0	Yes	Govt_job	Rural	85.52	30.0	never smoked	0
3148	Male	78.0	0	0	Yes	Self-employed	Urban	201.58	30.6	Unknown	0
3101	Female	49.0	0	0	Yes	Private	Urban	105.99	29.8	never smoked	0
4262	Female	12.0	0	0	No	children	Urban	116.06	25.9	Unknown	0

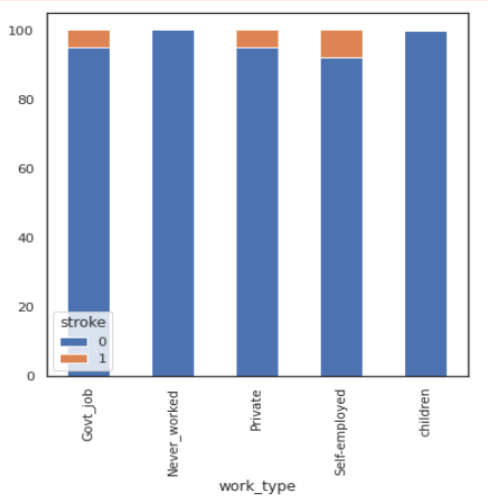
Let's Look At All The Categorical Variables And Their Distribution



Let's Look At All The Categorical Variables And Their Impact On Stroke





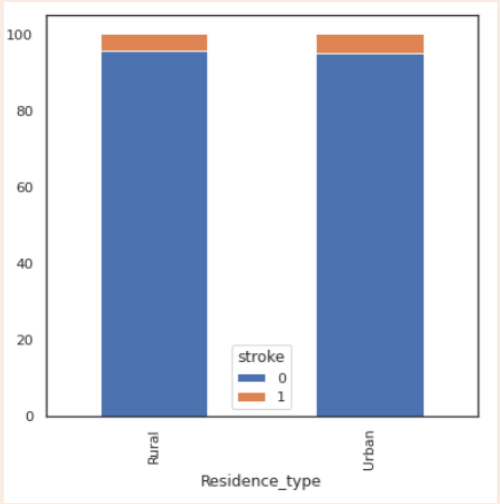


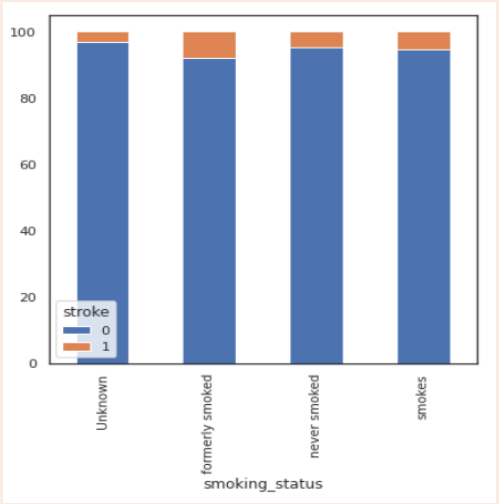
Gender :
Same likelihood to experience stroke.

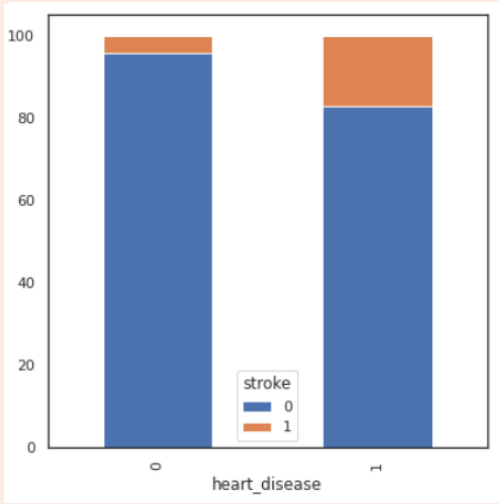
Marital Status :
Higher proportion of Married experienced stroke.

Work Type :
Work/Stress associated with stroke.

Let's Look At All The Categorical Variables And Their Impact On Stroke





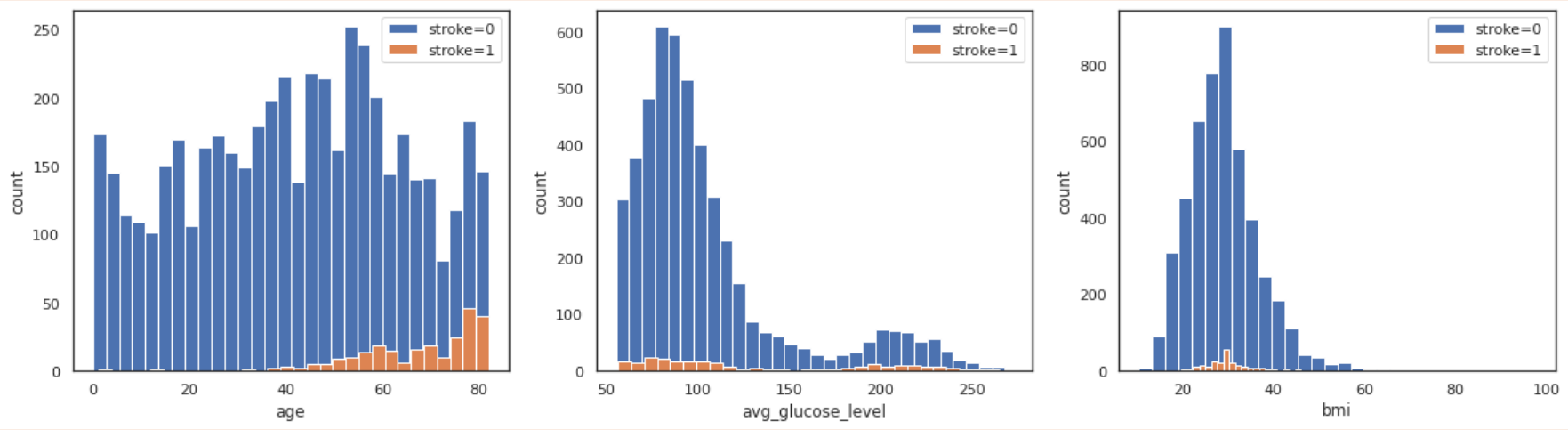


Residence Type : Same likelihood to experience stroke.

Smoking Status : There is association of smoking to stroke.

Hypertension/ Heart Disease : Higher proportion who suffered hypertension/heart disease experienced stroke.

Let's Look At The Continuous Variables And Their Impact On Stroke



Age :
Older patient more likely to suffer stroke than younger.

Diabetes :
Risk factor for stroke and prediabetic patients have increased risk of stroke.

BMI :
Between 25-35 highest to suffer stroke. Higher BMI does not increase stroke risk.

1. EDA & Data Preparation

Remove Irrelevant Features

User ID column dropped.

Remove Irrelevant Rows

1 Patient gender ‘Other’ dropped.

‘bmi’: Replace Missing Values

Replaced missing values with mean of bmi attribute.

1. EDA & Data Preparation

Remove Irrelevant Features

User ID column dropped.

Remove Irrelevant Rows

1 Patient gender 'Other' dropped.

'bmi': Replace Missing Values

Replaced missing values with mean of bmi attribute.

```
# Remaining rows in data
df_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5109 entries, 0 to 5109
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   gender             5109 non-null   object 
 1   age                5109 non-null   float64
 2   hypertension       5109 non-null   int64  
 3   heart_disease      5109 non-null   int64  
 4   ever_married       5109 non-null   object 
 5   work_type          5109 non-null   object 
 6   Residence_type     5109 non-null   object 
 7   avg_glucose_level  5109 non-null   float64
 8   bmi                5109 non-null   float64
 9   smoking_status     5109 non-null   object 
 10  stroke             5109 non-null   int64  
dtypes: float64(3), int64(3), object(5)
memory usage: 479.0+ KB
```

1. EDA & Data Preparation

- **Use Label Encoder for Features with 2 Classes**
  hypertension, heart_disease, ever_married, residence_type, stroke, gender
- **Encode Features with more than 2 Classes**
  work_type, smoking_status
- **Explore Target Variable 'stroke' Distribution**
  Moderate Imbalance Class.

```
[19] # limit to numerical data using df.select_dtypes()
     # all features converted to numerical
     df_num = df.select_dtypes(include=['number'])
     df_num.nunique()
```

```
gender                              2
age                               104
hypertension                        2
heart_disease                       2
ever_married                        2
Residence_type                      2
avg_glucose_level                3978
bmi                               418
stroke                              2
work_type_Govt_job                  2
work_type_Never_worked              2
work_type_Private                   2
work_type_Self-employed             2
work_type_children                  2
smoking_status_Unknown              2
smoking_status_formerly smoked      2
smoking_status_never smoked         2
smoking_status_smokes               2
dtype: int64
```

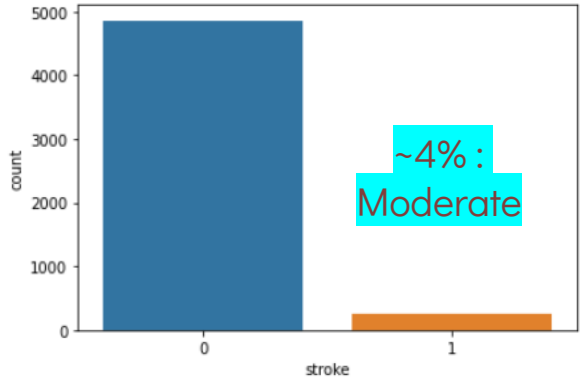
~4% : Moderate

1. EDA & Data Preparation

- **Use Label Encoder for Features with 2 Classes**
  hypertension, heart_disease, ever_married, residence_type, stroke, gender
- **Encode Features with more than 2 Classes**
  work_type, smoking_status
- **Explore Target Variable 'stroke' Distribution**
  Moderate Imbalance Class.

Imbalance Class	Proportion of Minority Class
Mild	20-40% of the data set
Moderate	1-20% of the data set
Extreme	<1% of the data set

```
[26] # To visualize how well balanced the target (dependent variable) is
     ## this is moderate imbalanced
     _ = sns.countplot(x=merged['stroke'])
```



2. Data Analysis & Feature Engineering

Feature Engineering : Adding Interaction Terms (Proposed)

1. Stress Factor : ever_married + work_type + smoking_status
2. Long Term Stress : (ever_married + work_type + smoking_status) * age
3. Hypertension coupled with Heart Disease : hypertension * heart_disease
4. Diabetes over BMI : (avg_glucose_level + Gender) / bmi
5. Gender roles in Marriage : ever_married + Gender

Add features to improve metrics performance

2. Data Analysis & Feature Engineering

Feature Engineering : Adding Interaction Terms (Proposed)

1. **Stress Factor**: ever_married + work_type + smoking_status
2. Long Term Stress : (ever_married + work_type + smoking_status) * age
3. **Hypertension coupled with Heart Disease :** hypertension * heart_disease
4. Diabetes over BMI : (avg_glucose_level + Gender) / bmi
5. Gender roles in Marriage : ever_married + Gender

Added features not highly correlated to core features

2. Data Analysis & Feature Engineering

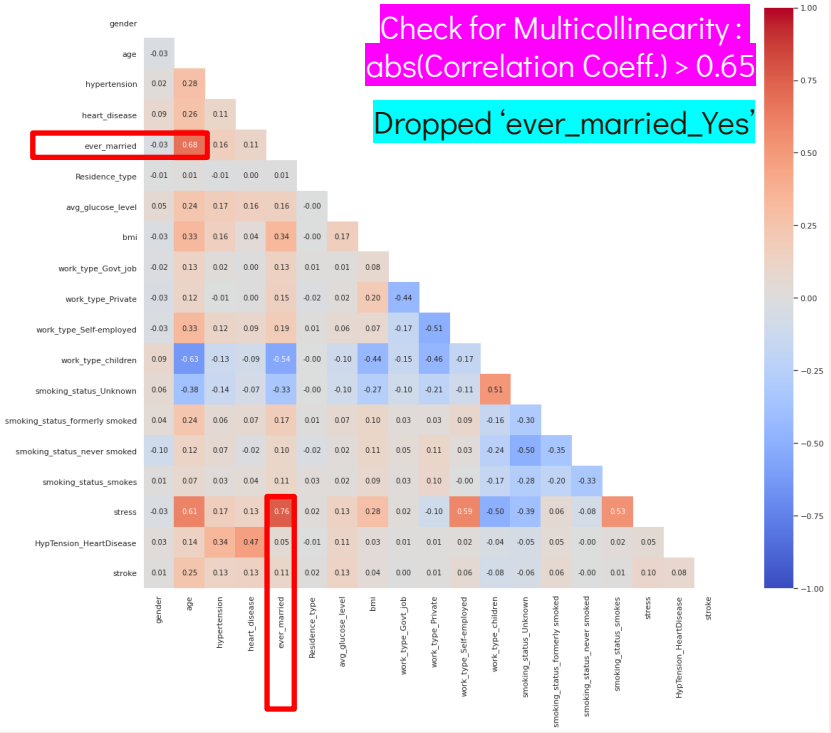
Filter Features by Variance

```
df.var()
```

Dropped low variance features

gender	0.242647
age	511.373788
hypertension	0.087991
heart_disease	0.051114
ever_married	0.225617
Residence_type	0.249983
avg_glucose_level	2050.731557
bmi	59.262825
work_type_Govt_job	0.112081
work_type_Never_worked	0.004288
work_type_Private	0.244817
work_type_Self-employed	0.134634
work_type_children	0.116410
smoking_status_Unknown	0.210921
smoking_status_formerly smoked	0.143117
smoking_status_never smoked	0.233231
smoking_status_smokes	0.130609
stress	0.593131
HypTension_HeartDisease	0.012372
stroke	0.046371
dtype: float64	

Filter Features by High Correlation



3. Data Transformation

SMOTE

Oversampling imbalance data:
- Create Class balance – 50:50

train_test_split

test_size=0.25, random_state=42

Feature Scaling

Min-Max Scaler

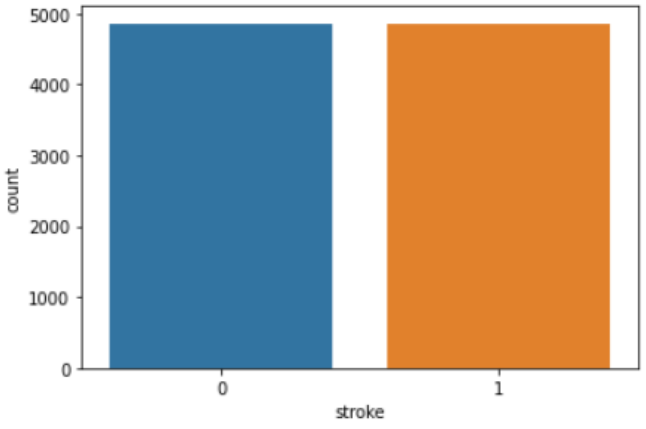
ML Model Training

Hyperparameter tuning using K-fold cross validation via Grid Search

3. Data Transformation

```
Shape of X before SMOTE: (5109, 17)
Shape of X after SMOTE: (9720, 17)
Shape of y after SMOTE: (9720,)

Balance of positive and negative classes (%):
1    50.0
0    50.0
Name: stroke, dtype: float64
```

```
_ = sns.countplot(x=y_sm)
```



SMOTE

Oversampling imbalance data:
- Create Class balance – 50:50

train_test_split

test_size=0.25, random_state=42

Feature Scaling

Min-Max Scaler

ML Model Training

Hyperparameter tuning using K-fold cross validation via Grid Search

4. Machine Learning Model Training & Evaluation

(No Feature Engineering)

4. ML Model Training & Evaluation

Baseline Model: Logistic Regression (without SMOTE)

Hyperparameter: (max_iter=5000)

Classification report:

	precision	recall	f1-score	support
0	0.96	1.00	0.98	902
1	0.00	0.00	0.00	42
accuracy			0.95	944
macro avg	0.48	0.50	0.49	944
weighted avg	0.91	0.95	0.93	944

Confusion Matrix:

```
array([[900,   2],
       [ 42,   0]])
```

Classification Report

Confusion Matrix

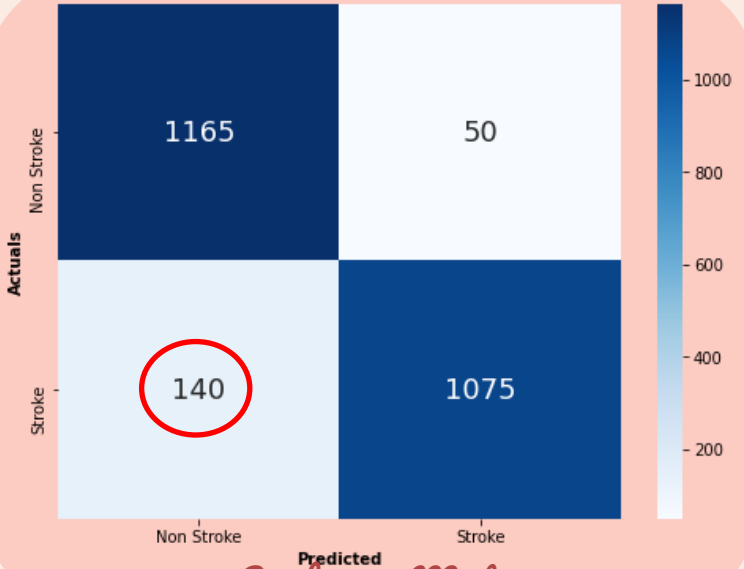
4. ML Model Training & Evaluation

Baseline Model: Logistic Regression (SMOTE)

Hyperparameter: (n_jobs=-1)

```
Classification report:

              precision    recall  f1-score   support

           0       0.89      0.96      0.92      1215
           1       0.96      0.88      0.92      1215

    accuracy                           0.92      2430
   macro avg       0.92      0.92      0.92      2430
weighted avg       0.92      0.92      0.92      2430


Confusion Matrix:
array([[1165,   50],
       [ 140, 1075]])
```

Classification Report



Confusion Matrix

4. ML Model Training & Evaluation

Alternative Model: Random Forest

Hyperparameter: (criterion='entropy', max_depth=15, min_samples_split=3, n_estimators=150, n_jobs=-1)

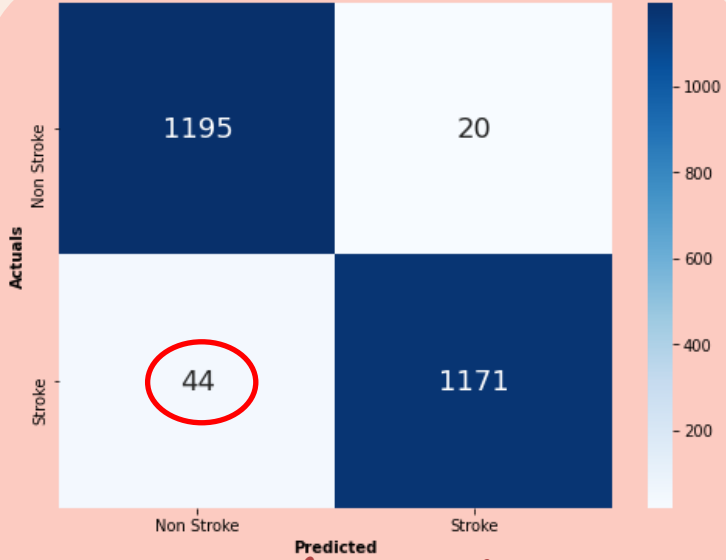
Classification report:

	precision	recall	f1-score	support
0	0.96	0.98	0.97	1215
1	0.98	0.96	0.97	1215
accuracy			0.97	2430
macro avg	0.97	0.97	0.97	2430
weighted avg	0.97	0.97	0.97	2430

Confusion Matrix:

```
array([[1195,   20],
       [  44, 1171]])
```

Classification Report



Confusion Matrix

4. ML Model Training & Evaluation

Alternative Model: Decision Tree

Hyperparameter: (criterion='entropy', max_depth=15, min_samples_split=5, random_state=42)

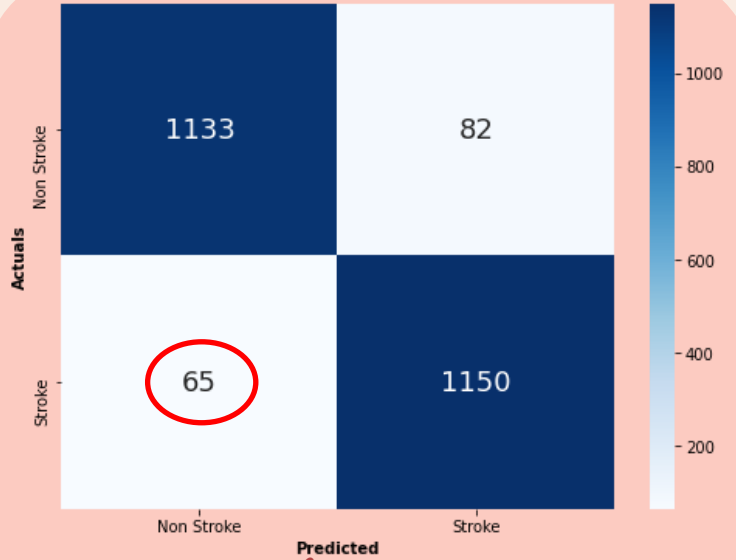
Classification report:

	precision	recall	f1-score	support
0	0.95	0.93	0.94	1215
1	0.93	0.95	0.94	1215
accuracy			0.94	2430
macro avg	0.94	0.94	0.94	2430
weighted avg	0.94	0.94	0.94	2430

Confusion Matrix:

```
array([[1133,   82],
       [  65, 1150]])
```

Classification Report



Confusion Matrix

4. ML Model Training & Evaluation

Alternative Model: K-nearest Neighbour

Hyperparameter: (n_neighbors=2)

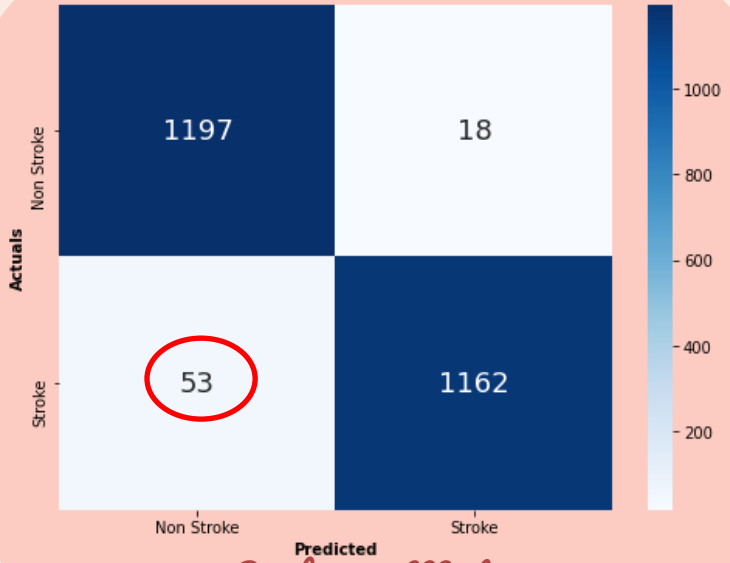
Classification report:

	precision	recall	f1-score	support
0	0.96	0.99	0.97	1215
1	0.98	0.96	0.97	1215
accuracy			0.97	2430
macro avg	0.97	0.97	0.97	2430
weighted avg	0.97	0.97	0.97	2430

Confusion Matrix:

```
array([[1197,   18],
       [  53, 1162]])
```

Classification Report



Confusion Matrix

4. ML Model Training & Evaluation

Alternative Model: LinearSVC

Hyperparameter: (C=0.1, max_iter=500)

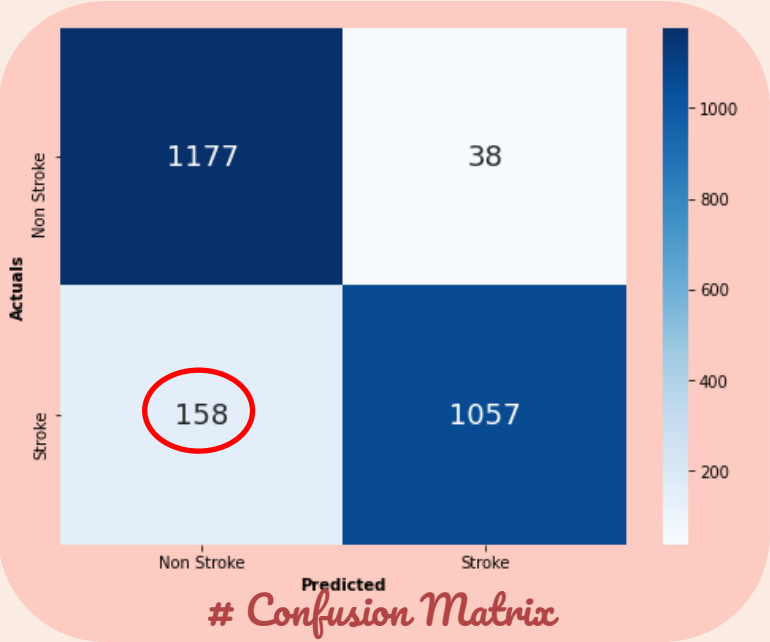
Classification report:

	precision	recall	f1-score	support
0	0.88	0.97	0.92	1215
1	0.97	0.87	0.92	1215
accuracy			0.92	2430
macro avg	0.92	0.92	0.92	2430
weighted avg	0.92	0.92	0.92	2430

Confusion Matrix:

```
array([[1177,   38],
       [ 158, 1057]])
```

Classification Report



Confusion Matrix

4. ML Model Training & Evaluation

Alternative Model: Mixed Naïve Bayes

No hyperparameter tuning

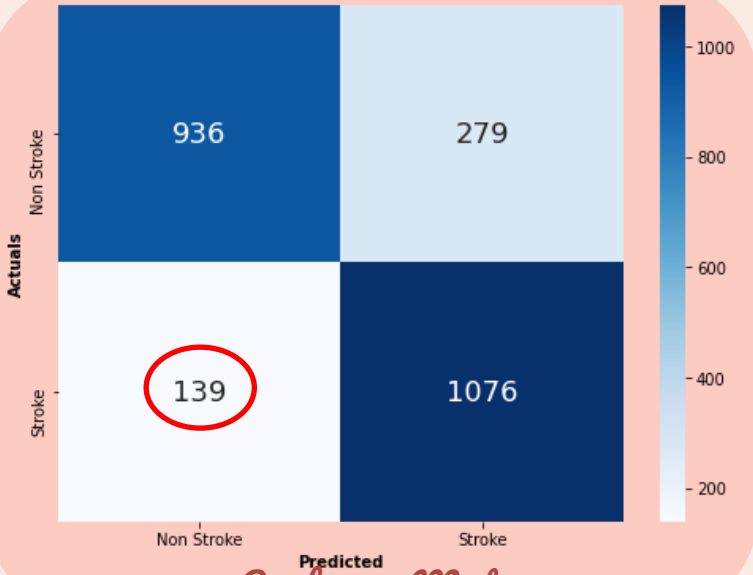
Classification report:

	precision	recall	f1-score	support
0	0.87	0.77	0.82	1215
1	0.79	0.89	0.84	1215
accuracy			0.83	2430
macro avg	0.83	0.83	0.83	2430
weighted avg	0.83	0.83	0.83	2430

Confusion Matrix:

```
array([[ 936,  279],
       [ 139, 1076]])
```

Classification Report



Confusion Matrix

Results (No Feature Engineering)

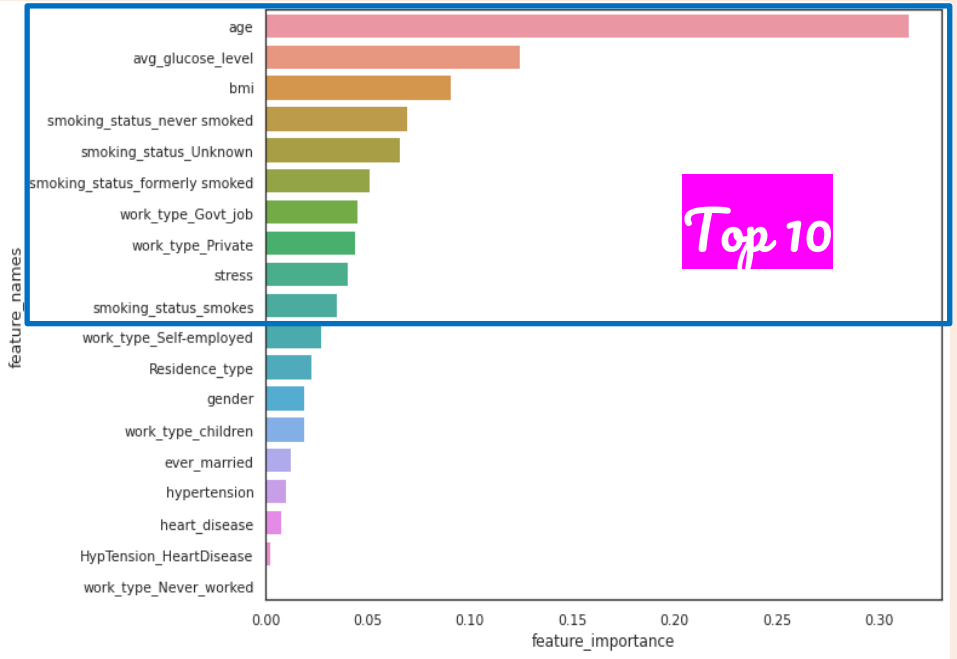
```
model_metrics.sort_values(by='F1 Score', ascending=False)
```

	Model	Training Acc	Testing Acc	Precision	Recall	F1 Score	False Negative
1	Random Forest	0.999588	0.973663	0.973847	0.973663	0.973660	44
3	K-nearest Neighbors	0.979561	0.970782	0.971173	0.970782	0.970776	53
2	Decision Tree	0.988889	0.939506	0.939592	0.939506	0.939503	65
0	Logistic Regression	0.917695	0.921811	0.924138	0.921811	0.921703	140
4	Support Vector Machine	0.919753	0.919342	0.923472	0.919342	0.919144	158
5	Mixed Naive Bayes	0.834431	0.827984	0.832397	0.827984	0.827411	139

4. Machine Learning Model Training & Evaluation

(Feature Engineering)

Feature Importance vs SelectKBest



Feature Importance

```
# Feature Selection by SelectKBest
new_features
```

```
['gender',
 'age',
 'Residence_type',
 'avg_glucose_level',
 'bmi',
 'work_type_Govt_job',
 'work_type_Private',
 'work_type_children',
 'smoking_status_Unknown',
 'smoking_status_never smoked',
 'smoking_status_smokes']
```

8 Selected

SelectKBest(k=11)

Results : Selection #1

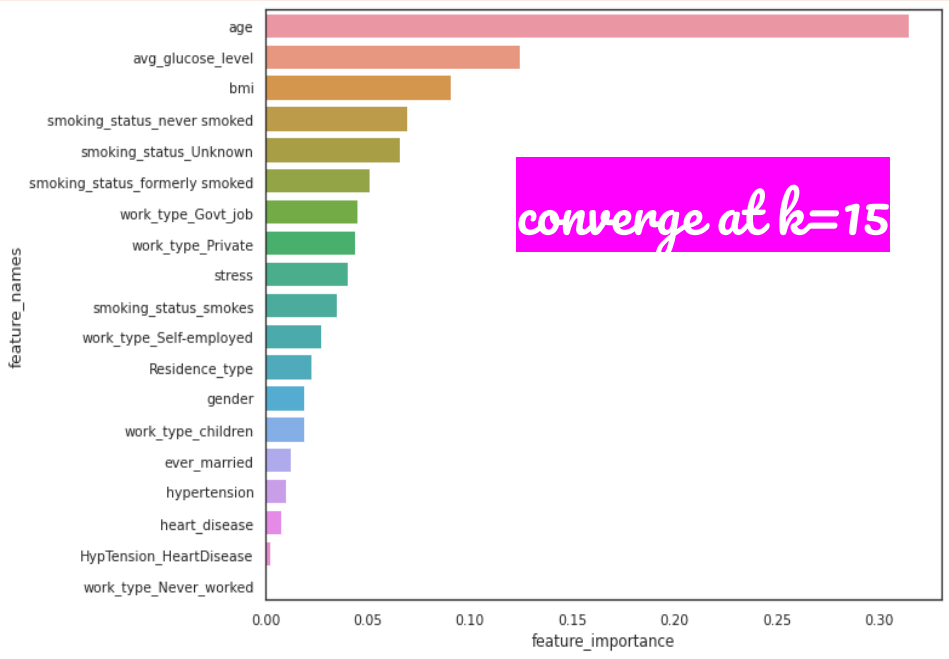
All Features

	Model	F1 Score	False Negative
1	Random Forest	0.972014	45
3	K-nearest Neighbors	0.970364	54
2	Decision Tree	0.941555	56
0	Logistic Regression	0.920879	141
4	Support Vector Machine	0.918722	160
5	Mixed Naive Bayes	0.822456	145

After Selection

	Model	F1 Score	False Negative
1	Random Forest	0.938625	37
3	K-nearest Neighbors	0.930837	60
2	Decision Tree	0.909376	72
4	Support Vector Machine	0.847699	166
0	Logistic Regression	0.844844	178
5	Mixed Naive Bayes	0.814406	168

Feature Importance with SelectKBest



Feature Importance

```
# Feature Selection by SelectKBest(15)
new_features
```

```
['gender',
 'age',
 'ever_married',
 'Residence_type',
 'avg_glucose_level',
 'work_type_Govt_job',
 'work_type_Never_worked',
 'work_type_Private',
 'work_type_Self-employed',
 'work_type_children',
 'smoking_status_Unknown',
 'smoking_status_formerly smoked',
 'smoking_status_never smoked',
 'smoking_status_smokes',
 'HypTension_HeartDisease']
```

SelectKBest (k=15)

Results : Selection #2

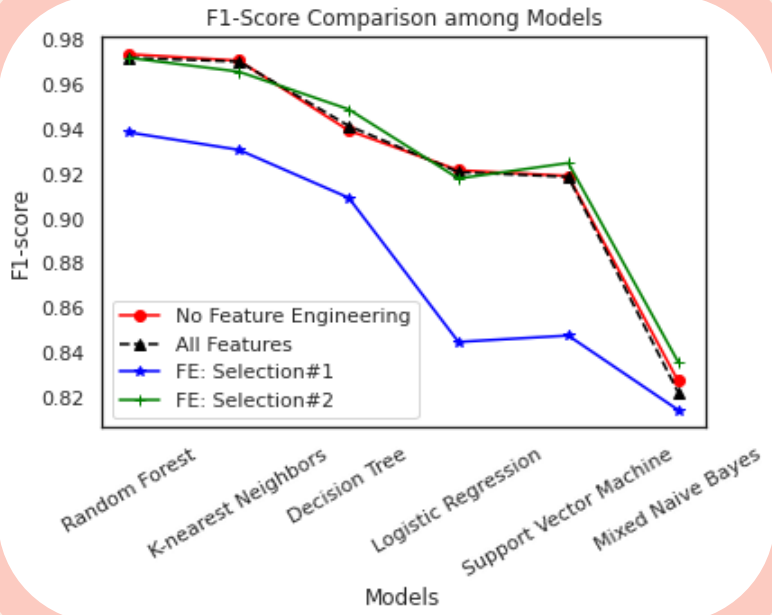
All Features

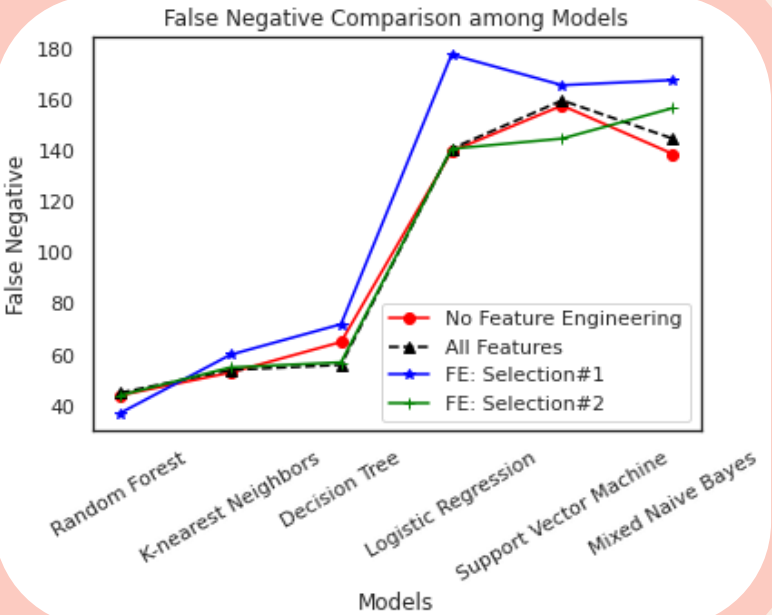
	Model	F1 Score	False Negative
1	Random Forest	0.972014	45
3	K-nearest Neighbors	0.970364	54
2	Decision Tree	0.941555	56
0	Logistic Regression	0.920879	141
4	Support Vector Machine	0.918722	160
5	Mixed Naive Bayes	0.822456	145

After Selection

	Model	F1 Score	False Negative
1	Random Forest	0.972015 ↑	↓ 44
3	K-nearest Neighbors	0.965839	55
2	Decision Tree	0.948970	57
4	Support Vector Machine	0.924955	145
0	Logistic Regression	0.918011	141
5	Mixed Naive Bayes	0.836018	157

Comparison of Results



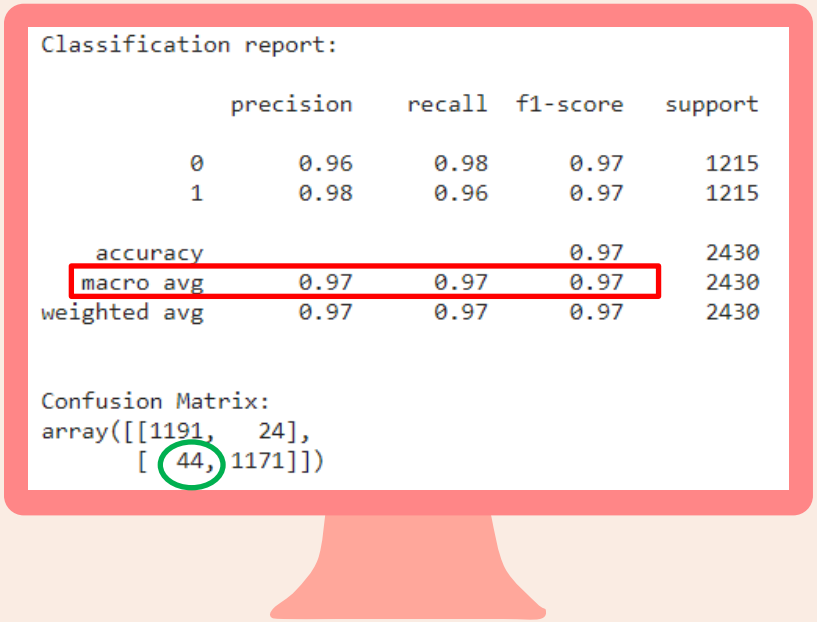


Conclusion

Best Model : Random Forest

Feature Selection #2 :

['gender',
 'age',
 'Residence_type',
 'avg_glucose_level',
 'work_type_Govt_job',
 'work_type_Private',
 'work_type_Self-employed',
 'work_type_children',
 'smoking_status_Unknown',
 'smoking_status_formerly smoked',
 'smoking_status_never smoked',
 'smoking_status_smokes',
 'HypTension_HeartDisease']



Classification report:

	precision	recall	f1-score	support
0	0.96	0.98	0.97	1215
1	0.98	0.96	0.97	1215
accuracy			0.97	2430
macro avg	0.97	0.97	0.97	2430
weighted avg	0.97	0.97	0.97	2430

Confusion Matrix:

```
array([[1191,   24],
       [  44, 1171]])
```

Conclusion

Business Benefits:

With this Model, we can predict if you and your family members would be at risk of suffering stroke.

If identified as a potential positive, we can assist to help you reduce the risk of suffering stroke through our Chronic Disease Management, Weight Loss & Health Optimization programs personalized to suit your schedule and meet your expectation.

Conclusion

Limitations/ Future Opportunities:

Features: Not comprehensive
Proposed Dataset to include:
Medical history, exercise regimen, lifestyle (alcohol), diet, family medical history, socioeconomic factors, sleep habits, biological samples analysis.

Comprehensive Dataset:
Use to assess progression to other diseases (e.g. diabetes, heart diseases, artery blockage, etc).

The End!

References

1. https://www.kaggle.com/fedesoriano/stroke-prediction-dataset
2. How to Use StandardScaler and MinMaxScaler Transforms in Python (machinelearningmastery.com)
3. https://python-bloggers.com/2020/12/how-to-effortlessly-handle-class-imbalance-with-python-and-smote/
4. https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/
5. https://www.stroke.org/en/about-stroke/stroke-risk-factors/additional-factors-that-may-be-linked-to-higher-stroke-risks