Content

Import librarries for Analyse the Data Read dataset Get information about data Data cleaning Data visualization - getting insights Import librarries Of Machine Learning Data preprocessing Build Machine Learning Models Introduction to Ensemle Learning

Introduction **Introduction: Predicting Stroke Risk with Machine Learning and Data Science**

Stroke is a major health challenge worldwide, ranking as one of the leading causes of death and long-term disability. Early prediction can make a huge difference, helping to prevent strokes or provide timely treatment, ultimately saving lives and improving outcomes.

In this project, we developed a smart system that uses **Machine Learning (ML)** and **Data Science** to analyze medical data and predict the risk of stroke. To ensure accurate and reliable predictions, we worked with a range of machine learning models, including:

- **Support Vector Machine (SVM)**
- **Random Forest**
- **Logistic Regression**
- **K-Nearest Neighbors (KNN)**
- And more, to refine our results and optimize performance.

Our system processes medical data like age, blood pressure, blood sugar levels, medical history, and other key factors. We used data cleaning and analysis techniques to prepare the data and uncover patterns that influence stroke risk.

The goal of this project is to create a tool that helps doctors and healthcare providers identify people at higher risk of stroke and offer tailored recommendations to reduce those risks. We also compared the performance of different models to choose the most accurate and effective one.

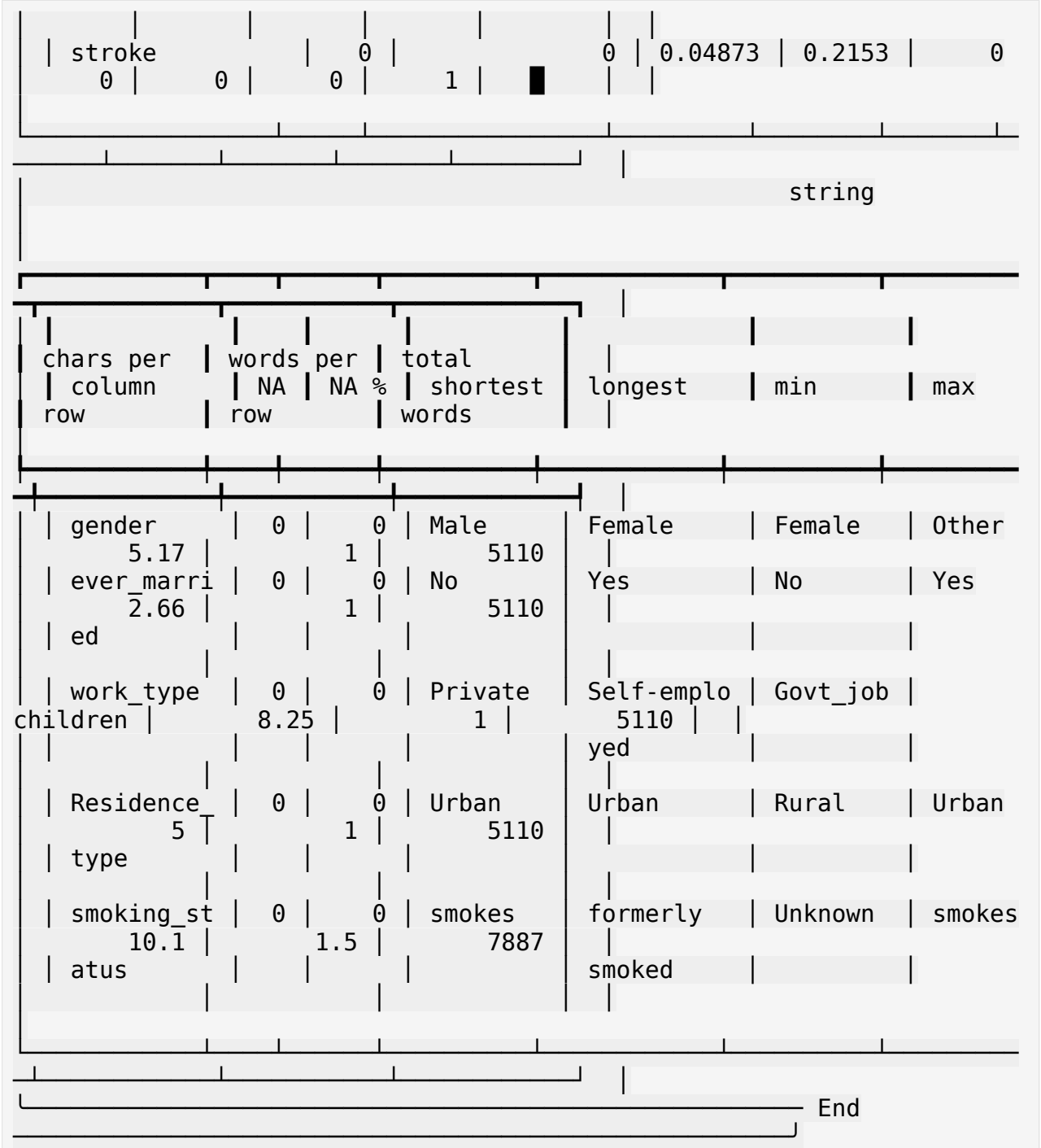
This project reflects how AI can be a powerful ally in healthcare, offering innovative solutions to tackle chronic illnesses, prevent life-threatening events, and improve the overall quality of life.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

from ydata_profiling import ProfileReport

from skimpy import skim
```

```python
df = pd.read_csv("healthcare-dataset-stroke-data.csv")
```

```python
skim(df)
```

```
╭──────────────────────────────────────────────── skimpy summary ─────────────────────────────────────────────────╮
│          Data Summary                Data Types                                                                  │
│ ┏━━━━━━━━━━━━━━━━━━━┳━━━━━━━━┓ ┏━━━━━━━━━━━━━┳━━━━━━━┓                                                           │
│ ┃ Dataframe         ┃ Values ┃ ┃ Column Type ┃ Count ┃                                                           │
│ ┡━━━━━━━━━━━━━━━━━━━╇━━━━━━━━┩ ┡━━━━━━━━━━━━━╇━━━━━━━┩                                                           │
│ │ Number of rows    │ 5110   │ │ string      │ 5     │                                                           │
│ │ Number of columns │ 12     │ │ int64       │ 4     │                                                           │
│ └───────────────────┴────────┘ │ float64     │ 3     │                                                           │
│                                └─────────────┴───────┘                                                           │
│                                                     number                                                       │
│ ┏━━━━━━━━━━━━━━━━━┳━━━━━┳━━━━━━━━━━━━━━━━┳━━━━━━━━━┳━━━━━━━━┳━━━━━━━┳━━━━━━━┳━━━━━━━┳━━━━━━━┳━━━━━━━┳━━━━━━━┓ │
│ ┃ column          ┃ NA  ┃ NA %           ┃ mean    ┃ sd     ┃ p0    ┃ p25   ┃ p50   ┃ p75   ┃ p100  ┃ hist   ┃ │
│ ┡━━━━━━━━━━━━━━━━━╇━━━━━╇━━━━━━━━━━━━━━━━╇━━━━━━━━━╇━━━━━━━━╇━━━━━━━╇━━━━━━━╇━━━━━━━╇━━━━━━━╇━━━━━━━╇━━━━━━━┩ │
│ │ id              │   0 │              0 │   36520 │  21160 │    67 │ 17740 │ 36930 │ 54680 │ 72940 │        │ │
│ │ age             │   0 │              0 │   43.23 │  22.61 │  0.08 │    25 │    45 │    61 │    82 │        │ │
│ │ hypertension    │   0 │              0 │ 0.09746 │ 0.2966 │     0 │     0 │     0 │     0 │     1 │        │ │
│ │ heart_disease   │   0 │              0 │ 0.05401 │ 0.2261 │     0 │     0 │     0 │     0 │     1 │        │ │
│ │ avg_glucose_lev │   0 │              0 │   106.1 │  45.28 │ 55.12 │ 77.25 │ 91.88 │ 114.1 │ 271.7 │        │ │
│ │ el              │     │                │         │        │       │       │       │       │       │        │ │
│ │ bmi             │ 201 │ 3.933463796477 │   28.89 │  7.854 │  10.3 │  23.5 │  28.1 │  33.1 │  97.6 │        │ │
│ │                 │     │           4952 │         │        │       │       │       │       │       │        │ │
```




```
df.profile_report()
```


```
{"model_id":"75aa94d0ca944701a0dacd0f946476de","version_major":2,"version_minor":0}
```

```
{"model_id":"e459e45008d54b538f283b7b0d70463d","version_major":2,"version_minor":0}
```

{"model_id":"beed59608d074a46bbddffee2c9733bc","version_major":2,"version_minor":0}

```
<IPython.core.display.HTML object>
```


```
df.loc[df['age'] <1]
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type
363	7559	Female	0.64	0	0	No	children
376	22706	Female	0.88	0	0	No	children
564	61511	Female	0.32	0	0	No	children
982	54747	Male	0.88	0	0	No	children
996	53279	Male	0.24	0	0	No	children
1093	66772	Female	0.32	0	0	No	children
1206	68908	Female	0.72	0	0	No	children
1317	30084	Male	0.80	0	0	No	children

```
1600  40544    Male  0.40             0             0           No
children
1614  47350  Female  0.08             0             0           No
children
1808  53126  Female  0.64             0             0           No
children
1975   6596    Male  0.56             0             0           No
children
1999  42500    Male  0.24             0             0           No
children
2008  67099    Male  0.56             0             0           No
children
2012  34261    Male  0.64             0             0           No
children
2030  38920    Male  0.48             0             0           No
children
2358   1275    Male  0.88             0             0           No
children
2481  20257    Male  0.88             0             0           No
children
2490  48406    Male  0.88             0             0           No
children
2579  68382    Male  0.32             0             0           No
children
2630  61836  Female  0.80             0             0           No
children
2801    760    Male  0.80             0             0           No
children
2875  42938    Male  0.64             0             0           No
children
2898  64974    Male  0.24             0             0           No
children
3251  14877    Male  0.56             0             0           No
children
3295  29955    Male  0.08             0             0           No
children
3392  11371    Male  0.24             0             0           No
children
3440  18837    Male  0.56             0             0           No
children
3618  22877    Male  0.16             0             0           No
children
3626  23360    Male  0.80             0             0           No
children
3859  13857    Male  0.32             0             0           No
children
3894  69435  Female  0.56             0             0           No
children
3968  41500    Male  0.16             0             0           No
```

```
children
4007  43282    Male  0.72             0              0           No
children
4021   8247    Male  0.16             0              0           No
children
4053  40255  Female  0.48             0              0           No
children
4293  69222    Male  0.24             0              0           No
children
4409   1953  Female  0.72             0              0           No
children
4581  15728  Female  0.40             0              0           No
children
4645  25783  Female  0.48             0              0           No
children
4910  37622  Female  0.32             0              0           No
children
4929  29487    Male  0.72             0              0           No
children
5089  56714  Female  0.72             0              0           No
children

     Residence_type  avg_glucose_level   bmi smoking_status  stroke
363           Urban              83.82  24.9        Unknown       0
376           Rural              88.11  15.5        Unknown       0
564           Rural              73.71  16.2        Unknown       0
982           Rural             157.57  19.2        Unknown       0
996           Rural             118.87  16.3        Unknown       0
1093          Rural              55.86  16.0        Unknown       0
1206          Urban              66.36  23.0        Unknown       0
1317          Rural              98.67  17.5        Unknown       0
1600          Urban             109.56  14.3        Unknown       0
1614          Urban             139.67  14.1        Unknown       0
1808          Urban              62.27  17.3        Unknown       0
1975          Rural             111.77  21.1        Unknown       0
1999          Rural             146.97  18.5        Unknown       0
2008          Rural              57.02  20.7        Unknown       0
2012          Rural              86.74  16.2        Unknown       0
2030          Urban              73.02   NaN        Unknown       0
2358          Urban             112.19  18.9        Unknown       0
2481          Urban              90.62  22.4        Unknown       0
2490          Urban              85.38  23.4        Unknown       0
2579          Urban             127.78  20.8        Unknown       0
2630          Urban             106.59  15.5        Unknown       0
2801          Urban              75.22  33.1        Unknown       0
2875          Urban              60.40  17.3        Unknown       0
2898          Urban              58.35  18.6        Unknown       0
3251          Rural             127.23  20.1        Unknown       0
3295          Rural              70.33  16.9        Unknown       0
```

```
3392          Urban              89.28  14.2         Unknown       0
3440          Urban              98.23  14.1         Unknown       0
3618          Urban             114.71  17.4         Unknown       0
3626          Rural             114.54  15.1         Unknown       0
3859          Urban              89.04  17.8         Unknown       0
3894          Urban              80.92  18.3         Unknown       0
3968          Rural              69.79  13.0         Unknown       0
4007          Rural             159.79  19.9         Unknown       0
4021          Urban             109.52  13.9         Unknown       0
4053          Rural             118.75  17.4         Unknown       0
4293          Urban              57.09  19.4         Unknown       0
4409          Rural             112.19  20.1         Unknown       0
4581          Rural              85.65  17.4         Unknown       0
4645          Rural              94.06  14.8         Unknown       0
4910          Urban             108.63  19.6         Unknown       0
4929          Urban              80.08  16.4         Unknown       0
5089          Rural              62.13  16.8         Unknown       0
```

```
df.loc[df['bmi']>50]
```

```
         id  gender   age  hypertension  heart_disease ever_married  \
113   41069  Female  45.0             0              0          Yes
254   32257  Female  47.0             0              0          Yes
258   28674  Female  74.0             1              0          Yes
270   72911  Female  57.0             1              0          Yes
333    1703  Female  52.0             0              0          Yes
...     ...     ...   ...           ...            ...          ...
4650  68074    Male  54.0             0              0          Yes
4779  65892  Female  58.0             0              0          Yes
4838   5131  Female  51.0             0              0          Yes
4906  72696  Female  53.0             0              0          Yes
4952  16245    Male  51.0             1              0          Yes

          work_type Residence_type  avg_glucose_level   bmi
smoking_status  \
113         Private          Rural             224.10  56.6     never
smoked
254         Private          Urban             210.95  50.1
Unknown
258   Self-employed          Urban             205.84  54.6     never
smoked
270         Private          Rural             129.54  60.9
smokes
333         Private          Urban              82.24  54.7  formerly
smoked
...             ...            ...                ...   ...
...
4650        Private          Rural             100.47  50.2  formerly
smoked
4779  Self-employed          Urban              66.71  51.7     never
```

```
smoked
4838        Private          Urban             107.72  60.9
Unknown
4906        Private          Urban              70.51  54.1     never
smoked
4952  Self-employed          Rural             211.83  56.6     never
smoked

      stroke
113        1
254        0
258        0
270        0
333        0
...      ...
4650       0
4779       0
4838       0
4906       0
4952       0

[79 rows x 12 columns]
```

```
df.head()
```

```
      id  gender   age  hypertension  heart_disease ever_married  \
0   9046    Male  67.0             0              1          Yes
1  51676  Female  61.0             0              0          Yes
2  31112    Male  80.0             0              1          Yes
3  60182  Female  49.0             0              0          Yes
4   1665  Female  79.0             1              0          Yes

       work_type Residence_type  avg_glucose_level   bmi
smoking_status  \
0        Private          Urban             228.69  36.6  formerly
smoked
1  Self-employed          Rural             202.21   NaN     never
smoked
2        Private          Rural             105.92  32.5     never
smoked
3        Private          Urban             171.23  34.4
smokes
4  Self-employed          Rural             174.12  24.0     never
smoked

   stroke
0       1
1       1
2       1
```

```
3       1
4       1
```

```
df.sample(10)
```

```
         id  gender   age  hypertension  heart_disease ever_married  \
1753   5511    Male  66.0             0              0          Yes
370   28286    Male  44.0             0              0          Yes
4174   6574  Female  35.0             0              0          Yes
4891  18636  Female  26.0             0              0          Yes
4628  38658  Female  62.0             0              0          Yes
4272  50412  Female  17.0             0              0           No
3822  41153  Female  32.0             0              0          Yes
1234  37053    Male  53.0             0              0          Yes
4392  55681  Female   7.0             0              0           No
463   56735  Female  78.0             0              0          Yes

          work_type Residence_type  avg_glucose_level   bmi
smoking_status  \
1753  Self-employed          Urban              71.38   NaN  formerly
smoked
370         Private          Rural              74.91  37.5     never
smoked
4174  Self-employed          Urban             103.29  20.6     never
smoked
4891       Govt_job          Urban              72.56  35.4     never
smoked
4628  Self-employed          Rural             213.92  44.6     never
smoked
4272        Private          Urban              96.47  25.6
Unknown
3822        Private          Urban             100.01  37.2     never
smoked
1234       Govt_job          Rural              78.73  23.3     never
smoked
4392       children          Rural              63.98  23.0
Unknown
463   Self-employed          Rural             115.43  27.8     never
smoked

      stroke
1753       0
370        0
4174       0
4891       0
4628       0
4272       0
```

```
3822       0
1234       0
4392       0
463        0
```

```
df.info()

<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

df.describe()

                 id          age  hypertension  heart_disease  \
count   5110.000000  5110.000000   5110.000000    5110.000000   
mean   36517.829354    43.226614      0.097456       0.054012   
std    21161.721625    22.612647      0.296607       0.226063   
min       67.000000     0.080000      0.000000       0.000000   
25%    17741.250000    25.000000      0.000000       0.000000   
50%    36932.000000    45.000000      0.000000       0.000000   
75%    54682.000000    61.000000      0.000000       0.000000   
max    72940.000000    82.000000      1.000000       1.000000   

       avg_glucose_level          bmi       stroke  
count        5110.000000  4909.000000  5110.000000  
mean          106.147677    28.893237     0.048728  
std            45.283560     7.854067     0.215320  
min            55.120000    10.300000     0.000000  
25%            77.245000    23.500000     0.000000  
50%            91.885000    28.100000     0.000000  
75%           114.090000    33.100000     0.000000  
max           271.740000    97.600000     1.000000
```

Initial Observations and Insights

Dataset Overview:
The dataset contains information on 5,110 individuals across 12 columns, with the primary goal of predicting stroke occurrence (binary classification: 1 for stroke, 0 for no stroke).

Key Features:
The dataset includes demographic, health, and lifestyle-related attributes such as age, gender, hypertension, heart disease, smoking status, and more.

Data Cleaning and Preprocessing Insights:

1. **Missing Values:**
   - The BMI column contains 201 missing values.
   - These will be imputed using the median to ensure consistent data quality.
2. **Categorical Features:**
   - Categorical variables like gender, work type, and smoking status will need to be converted into numerical representations (e.g., one-hot encoding).
   - The gender category "Other" appears only once and will be removed due to insufficient representation.
3. **Special Cases:**
   - The "Unknown" category in smoking status is significant and may be treated as a separate group to evaluate its potential relationship with stroke risk.
   - Age ranges from 0.08 to 82 years. Further evaluation may be required for extremely low values.

Key Insights from Descriptive Statistics:

1. **Imbalanced Target Variable:**
   - Only 249 individuals experienced a stroke, making this an imbalanced dataset. Resampling techniques may be required to address this.
2. **Feature Relationships with Stroke:**
   - **Age:** Stroke is more common in older individuals.
   - **Hypertension and Heart Disease:** Higher rates of both conditions are observed in individuals who have had a stroke.
   - **Glucose Levels and BMI:** Elevated glucose levels and BMI are linked to an increased risk of stroke.
   - **Smoking Status:** Smokers and former smokers have a higher prevalence of stroke.

Next Steps

1. **Data Imputation:**
   - Replace missing BMI values with the median.
2. **Feature Encoding:**
   - Convert categorical variables into numerical representations using techniques like one-hot encoding.
3. **Exploratory Data Analysis (EDA):**
   - Visualize feature distributions for stroke and non-stroke groups using histograms and box plots.
   - Analyze correlations between features and the target variable through heatmaps and scatter plots.
   - Explore relationships between pairs of features.
4. **Feature Engineering:**
   - Create new features, such as age groups, to capture meaningful patterns.
5. **Model Development:**
   - Train and evaluate machine learning models (e.g., Random Forest, SVM, Logistic Regression, etc.).
   - Address data imbalance through oversampling, undersampling, or cost-sensitive methods.

Summary of Key Insights:

- The dataset contains valuable features that influence stroke risk.
- Missing values and categorical variables need preprocessing.
- The dataset is imbalanced, requiring specific handling for accurate model predictions.
- Strong associations exist between stroke and factors like age, hypertension, heart disease, glucose levels, and smoking.

```
df.drop(['id'],axis=1,inplace = True)

#نحن لا نحترم من ليسو على الفطره
df.loc[df["gender"]=="Other"]
```

```
     gender   age  hypertension  heart_disease ever_married work_type
\
3116  Other  26.0             0              0           No   Private


     Residence_type  avg_glucose_level   bmi   smoking_status  stroke

3116          Rural              143.33  22.4  formerly smoked       0


df.drop(3116,inplace=True)
```

```
df.drop_duplicates(inplace=True)
```

```
df.isnull().sum()
```

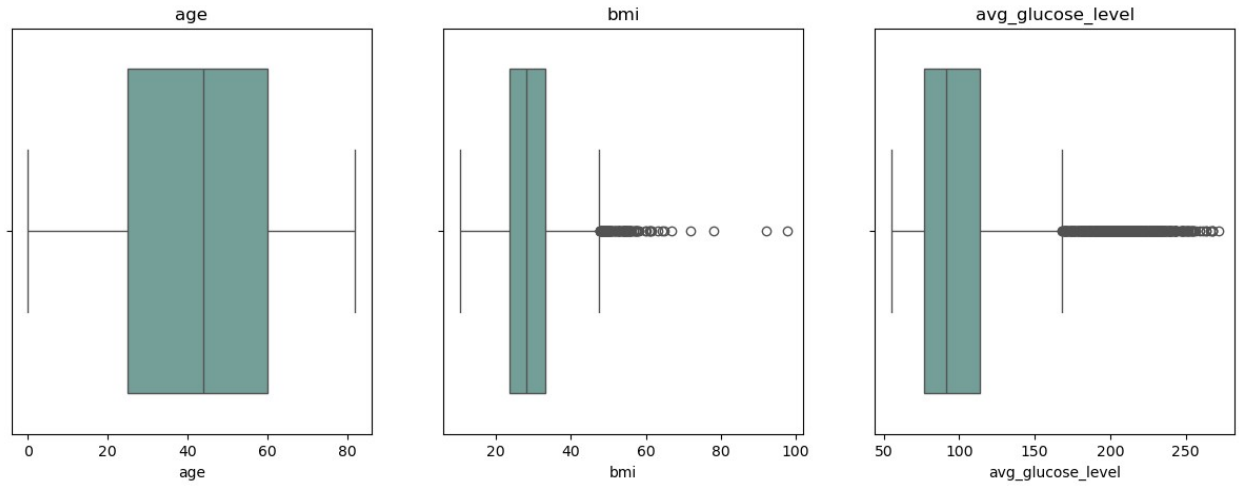
```
gender                 0
age                    0
hypertension           0
heart_disease          0
ever_married           0
work_type              0
Residence_type         0
avg_glucose_level      0
bmi                  201
smoking_status         0
stroke                 0
dtype: int64
```

```
# because we can't calcuate bmi from the data , we will drop the null
data

df.dropna(inplace=True)

df.isnull().sum().sum()
```

```
0
```

```
import tkinter
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
```

```
num_cols = ['age','bmi','avg_glucose_level']

plt.figure(figsize=(15, 5))
for i in range(3) :
    plt.subplot(1,3,i+1)

    sns.boxplot(x=df[num_cols[i]],color='#6DA59D')
    plt.title(num_cols[i])
plt.show()
```



Observations from Box Plots:

1. **Age:**
   - **Central Tendency (Median):** The median age (M_age) is approximately 45 years based on the box plot.
   - **Interquartile Range (IQR):** The first quartile (Q1_age) is around 20, and the third quartile (Q3_age) is around 60, giving an IQR (Q3_age - Q1_age) of approximately 40.
   - **Range:** The age ranges from a minimum value of about 0 to a maximum of approximately 85.
   - **Skewness:** The distribution of age is slightly skewed toward higher values, indicating a positive skew.
2. **BMI (Body Mass Index):**
   - **Central Tendency (Median):** The median BMI (M_bmi) is approximately 28.
   - **Interquartile Range (IQR):** The first quartile (Q1_bmi) is about 20, and the third quartile (Q3_bmi) is around 40, giving an IQR (Q3_bmi - Q1_bmi) of approximately 20.

- **Outliers:** There are some outliers represented by data points above the upper whisker. The exact number of outliers can be determined from the plot.

- **Skewness:** The BMI distribution is slightly skewed toward higher values, indicating a positive skew.

3. **Average Glucose Level:**
   - **Central Tendency (Median):** The median glucose level (M_glucose) is roughly 90.

   - **Interquartile Range (IQR):** The first quartile (Q1_glucose) is about 70, and the third quartile (Q3_glucose) is approximately 120, giving an IQR (Q3_glucose - Q1_glucose) of around 50.

   - **Outliers:** There are several outliers represented by data points beyond the upper whisker. The number of outliers is significantly higher compared to BMI.

   - **Skewness:** The glucose level distribution is highly skewed toward higher values, indicating a strong positive skew.

Mathematical Summary:

Metric	Age	BMI	Avg Glucose Level
Median	M_age ≈ 45	M_bmi ≈ 28	M_glucose ≈ 90
IQR	IQR_age ≈ 40	IQR_bmi ≈ 20	IQR_glucose ≈ 50
Range	0 to 85	10 to 45	40 to 170
Skewness	Positive	Positive	Positive
Outliers	None	Some	Many

Key Insights:

1. **Central Tendency:** The median values (M_age, M_bmi, M_glucose) reflect the central point for each feature's distribution.

2. **Dispersion:** The IQR highlights the spread of the middle 50% of data, showing the concentration of values.

3. **Range:** The minimum and maximum values provide an overview of the data's boundaries.

4. **Skewness:** All three features show positive skewness, with glucose levels having the most extreme skew.

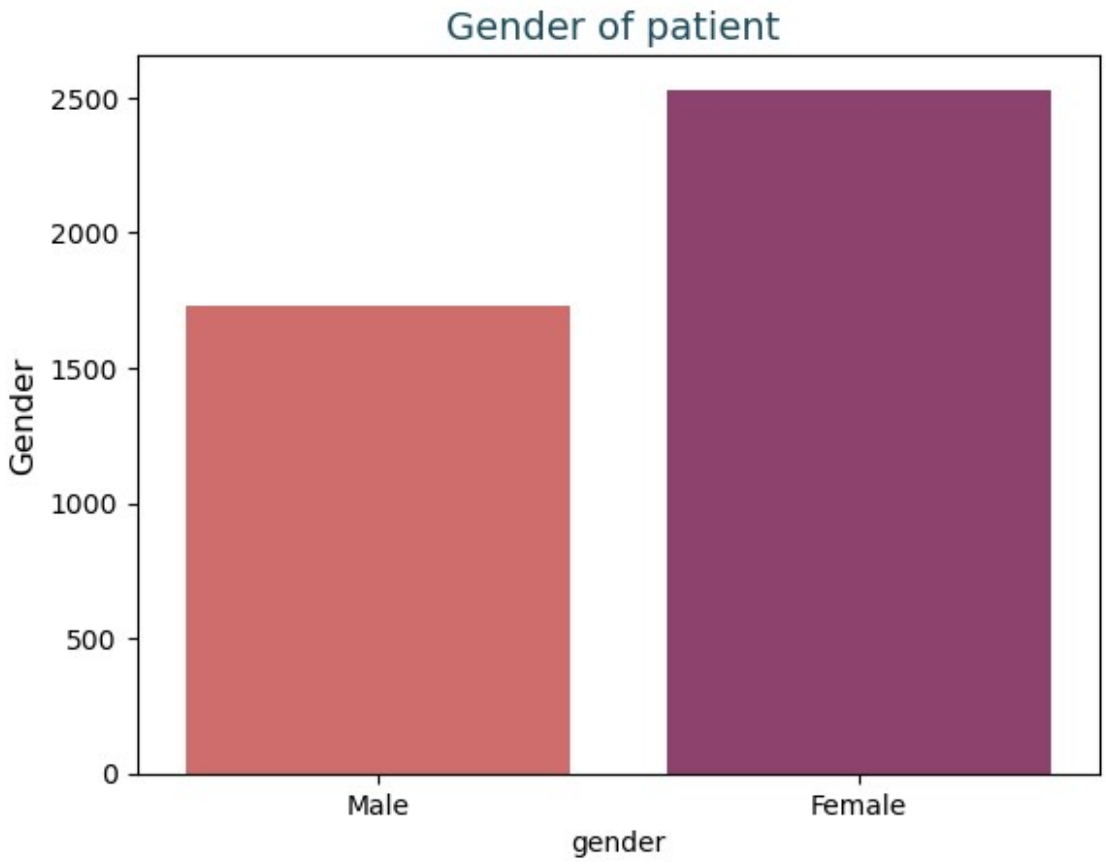
5. **Outliers:** BMI and glucose levels have significant outliers, which might need attention during preprocessing.

```
def detect_outliers(data,column):
    q1 = df[column].quantile(.25)
    q3= df[column].quantile(.75)
    IQR = q3-q1

    lower_bound = q1 - (1.5*IQR)
    upper_bound = q3 + (1.5*IQR)

    ls = df.index[(df[column] <lower_bound) | (df[column] >
upper_bound)]

    return ls

index_list = []

for column in num_cols:
    index_list.extend(detect_outliers(df,column))

# remove duplicated indices in the index_list and sort it
index_list = sorted(set(index_list))

before_remove = df.shape

df =df.drop(index_list)
after_remove = df.shape

print(f'''Shape of data before removing outliers : {before_remove}
Shape of data after remove : {after_remove}''')
```

```
Shape of data before removing outliers : (4908, 11)
Shape of data after remove : (4258, 11)
```

```
sns.countplot(data = df, x = "gender",palette="flare")
plt.title('Gender of patient ' , size = 14,color = '#1D4B5B')
plt.ylabel('Gender',size = 12)
```

```
/var/folders/kq/hjk_kv4j39zc870bgy4c4r5c0000gn/T/
ipykernel_30402/2880881142.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.countplot(data = df, x = "gender",palette="flare")

Text(0, 0.5, 'Gender')
```



Overall Chart Type: This is a bar chart, used to compare the counts or frequencies of different categories in this case 'Male' vs 'Female' within the dataset.

Observations from the Bar Chart:

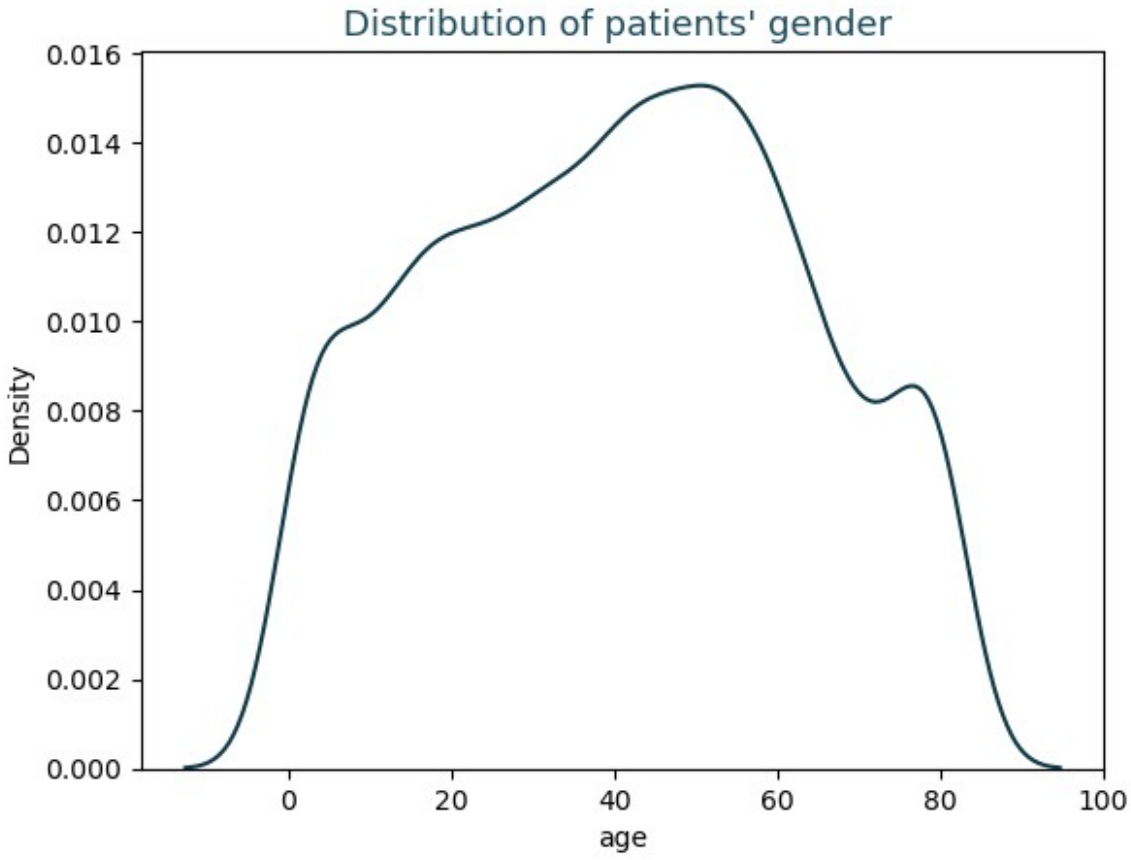
1. **Categories:** The x-axis displays two distinct categories: "Male" and "Female". This indicates we are examining the distribution of gender within the dataset.
2. **Counts:** The y-axis represents the "Gender" count. Let's denote the number of 'Male' individuals as `count_male`, and the number of 'Female' individuals as `count_female`.
   - Based on the visual, the height of the 'Male' bar is approximately around 1750. So, `count_male` ≈ 1750
   - The height of the 'Female' bar is greater, roughly around 2500. So, `count_female` ≈ 2500
3. **Comparison:** Visually, the 'Female' bar is significantly taller than the 'Male' bar, indicating a notable imbalance between the two genders in this dataset.

Insights and Interpretations:

- **Gender Imbalance:** The number of female patients is higher than the number of male patients by almost 750.
- **Data Bias**: This gender imbalance should be considered. This will be important when evaluating a model, or for any kind of demographic analysis

---

```
sns.kdeplot(df['age'] , color = '#103846')
plt.title('Distribution of patients\' gender ',color = '#1D4B5B',size
= 13 )

Text(0.5, 1.0, "Distribution of patients' gender ")
```



Overall Chart Type: This is a density plot, also known as a kernel density estimate (KDE) plot. Unlike a histogram, it provides a smooth estimate of the underlying probability density function of the data. It's useful for visualizing the shape of distributions without being influenced by bin choices.

Observations from the Density Plot:

1. **X-axis:** The x-axis represents the 'age' of the patients in the dataset. It ranges from approximately -10 to 100.

2. **Y-axis:** The y-axis represents the 'density', indicating the relative frequency of observations in that region.
3. **Shape of the Distribution:**
   - **Peaks:** The plot has several noticeable peaks. The main peak is centered around the age of 50. There are also smaller peaks at about 20 and 80
   - **Spread:** The distribution is more spread out, with some frequency even around 0.
   - **Skewness:** The plot appears to have positive skewness, indicating a longer tail towards the higher ages.
4. **Range:** From the graph, the values appears to range from less than zero to around 100.

Insights and Interpretations:

- **Age Clusters:** The multiple peaks in the plot suggest that there may be clusters or groups of patients with differing age distributions. The prominent peak around 50 suggests this is the most frequent age group within the dataset.
- **Younger Individuals:** The plot has a non-zero density around 0. This indicates that there are younger individuals in the dataset. However, there are fewer individuals in this age range relative to other groups.
- **Older Population:** We see a small peak around age 80, indicating that a significant number of patients are in this age range as well.

Summary in Mathematical Terms:

Let's represent the key features:

- **Mode(s):** `Mode_1_age` ≈ 50 (Primary mode) , `Mode_2_age` ≈ 20 , `Mode_3_age` ≈ 80. The modes are the peaks of the density plot where data is most concentrated. These represent ages where more individuals are present in the data set.
- **Range:** The range of the data is from approximately `min_age ≈ 0` to `max_age ≈ 100`.
- **Skewness:** Distribution is skewed positively, `Skewness_age > 0`

---

married = dict(df['ever_married'].value_counts()) fig = px.pie(names = married.keys(),values = married.values(),title = 'Ever Married',color_discrete_sequence=px.colors.sequential.Aggrnyl) fig.update_traces(textposition='inside', textinfo='percent+label')

```
h_disease = dict(df['heart_disease'].value_counts())
fig = px.pie(names = ['False' , 'True'],values =
h_disease.values(),title = 'Had a Heart Disease
',color_discrete_sequence=px.colors.sequential.Aggrnyl)
fig.update_traces(textposition='inside', textinfo='percent+label')
```

Had a Heart Disease

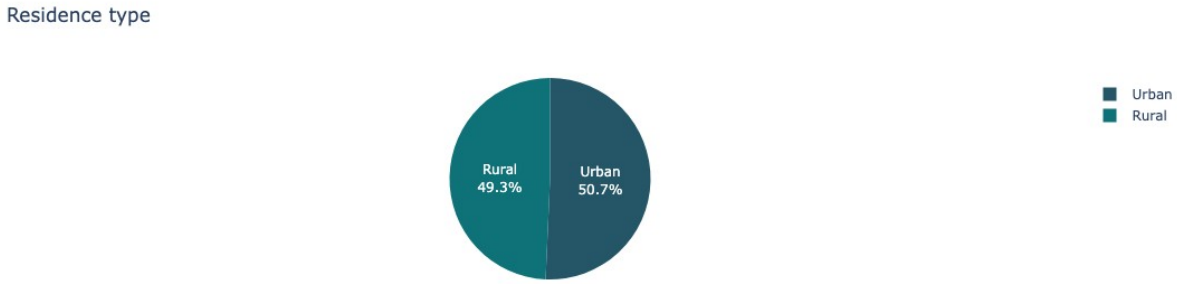

```
hypertension = dict(df['hypertension'].value_counts())
fig = px.pie(names =['False','True'],values =
hypertension.values(),title = 'Had a
Hypertension',color_discrete_sequence=px.colors.sequential.Aggrnyl)
fig.update_traces(textposition='inside', textinfo='percent+label')
```

Had a Hypertension

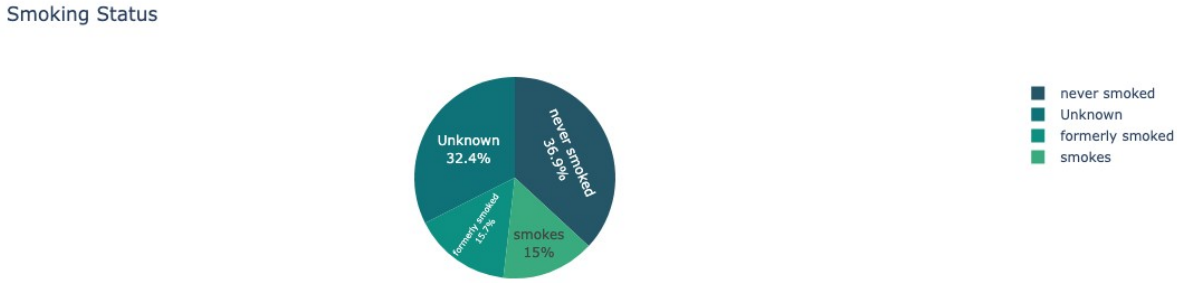

```
work_types = dict(df['work_type'].value_counts())
fig = px.pie(names = work_types.keys(),values =
work_types.values(),title = 'Work
Type',color_discrete_sequence=px.colors.sequential.Aggrnyl)
fig.update_traces(textposition='inside', textinfo='percent+label')
```

Work Type

```
Residence_types = dict(df['Residence_type'].value_counts())
fig = px.pie(names = Residence_types.keys(),values =
Residence_types.values(),title = 'Residence
type',color_discrete_sequence=px.colors.sequential.Aggrnyl)
fig.update_traces(textposition='inside', textinfo='percent+label')
```




```
smoking_status = dict(df['smoking_status'].value_counts())

fig = px.pie(names = smoking_status.keys(),values =
smoking_status.values(),title = 'Smoking
Status',color_discrete_sequence=px.colors.sequential.Aggrnyl)
fig.update_traces(textposition='inside', textinfo='percent+label')
```



Concise Summary of Pie Chart Insights:

1. Health Conditions:

- **Low Prevalence:** The dataset primarily represents individuals *without* heart disease (96.4%) or hypertension (93.1%). This indicates a relatively healthy population in these two categories.

2. Work and Residence:

- **Dominant Work Type:** The majority of the dataset is composed of individuals working in the *Private* sector (56.7%).
- **Balanced Residency:** The dataset is almost equally distributed between *Rural* (49.3%) and *Urban* (50.7%) residents.

3. Smoking Habits:

- **Significant Unknowns:** A large segment of the population have an *Unknown* smoking status (32.4%).
- **Smoking Distribution**: The remaining three categories are distributed somewhat equally.

Mathematical Summary (Simplified):

- **Heart Disease:** `p(Heart Disease = True)` ≈ 0.036 (3.6%)
- **Hypertension:** `p(Hypertension = True)` ≈ 0.069 (6.9%)
- **Work Type:** `p(Private)` ≈ 0.567 (56.7%), `p(children)` ≈ 0.155 (15.5%), `p(Self-employed)` ≈ 0.248 (24.8%), `p(Govt_job)` ≈ 0.125 (12.5%)
- **Residence:** `p(Rural)` ≈ 0.493 (49.3%), `p(Urban)` ≈ 0.507 (50.7%)
- **Smoking:** `p(never smoked)` ≈ 0.37 (37%), `p(Unknown)` ≈ 0.324 (32.4%), `p(formerly smoked)` ≈ 0.15 (15%) and `p(smokes)` ≈ 0.155 (15.5%).

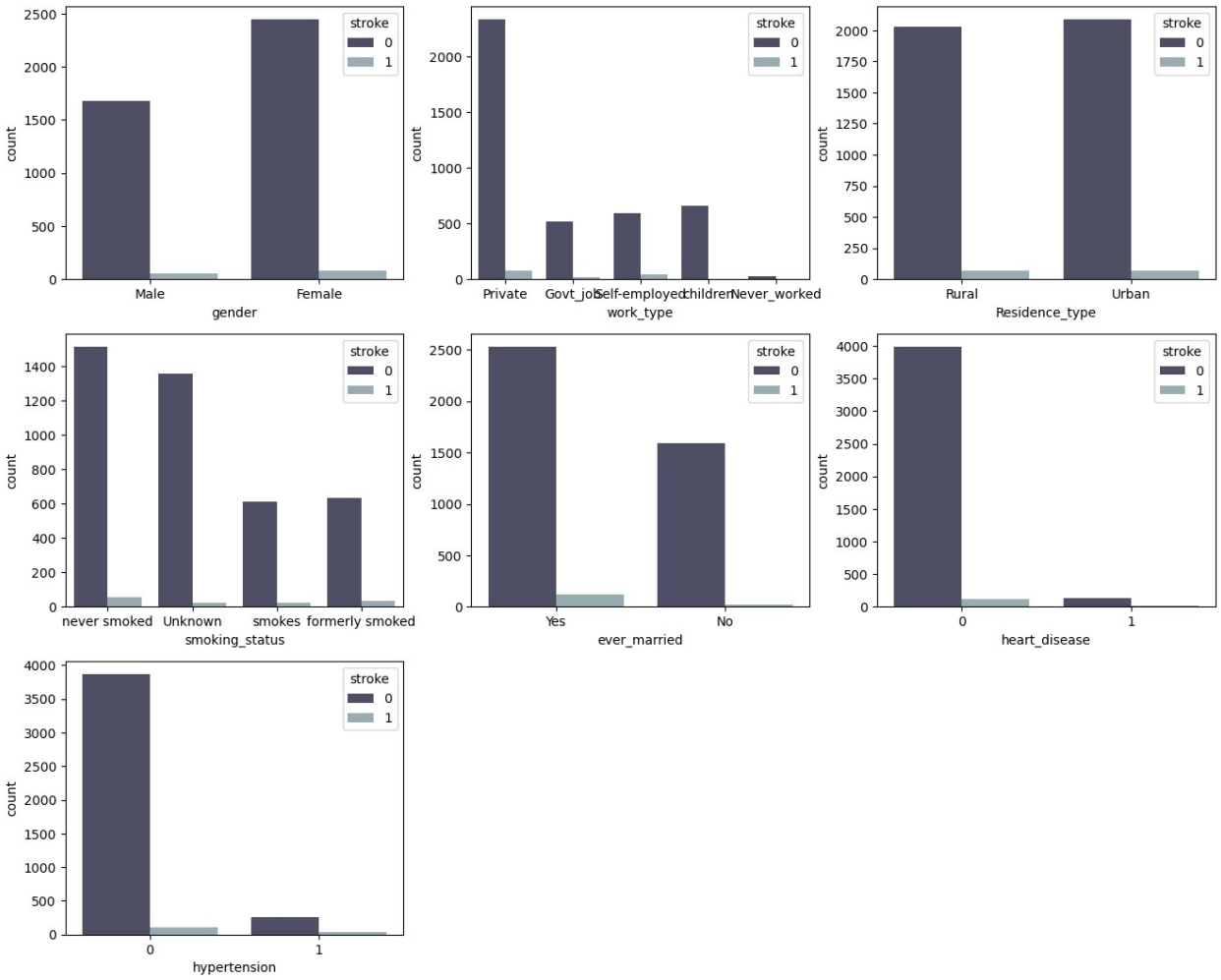
Most Impactful Takeaways:

- **Imbalanced Health Features:** Heart disease and hypertension are relatively infrequent in this sample.
- **Private Sector Majority:** Most individuals are employed in the private sector.
- **Smoking Data Issue:** The large proportion of unknown smoking status presents a significant limitation
- **Balanced Location:** Residency is nearly evenly split between urban and rural areas.

Implications for Modeling:

- The class imbalance in heart disease and hypertension needs to be considered while model training.
- The large proportion of "Unknown" responses in smoking status might necessitate imputation strategies.

```
cols =
['gender','work_type','Residence_type','smoking_status','ever_married'
,'heart_disease','hypertension']
plt.figure(figsize=(16,13))
for i in range(len(cols)):
    plt.subplot(3,3,i+1)

    sns.countplot(x=df[cols[i]],hue = df['stroke'],palette = 'bone')
```



Each chart shows the distribution of a categorical feature, split by the target variable (`stroke` = 0 or 1).

Overall Chart Type: These are grouped bar charts, also known as clustered bar charts. They allow us to visually compare how the distribution of a categorical feature differs across the two groups of the target variable.

Observations from the Grouped Bar Charts:

1. **Gender vs. Stroke:**
   - Both genders are primarily associated with a "stroke = 0".
   - Visually, the number of strokes in females is slightly more than the males
   - This seems to confirm the previous observation that females are more present in the dataset overall.
2. **Work Type vs. Stroke:**
   - "Private" sector has the highest number of individuals without stroke.

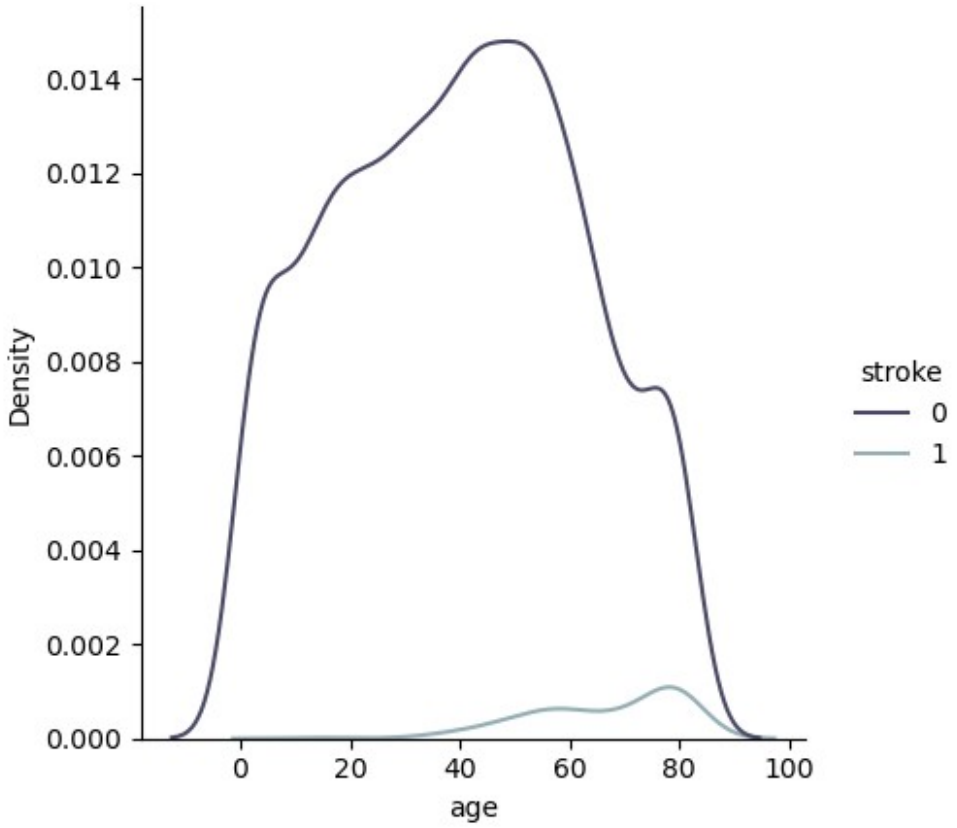
- The 'Self-employed' and 'Private' categories have a noticeable number of individuals with stroke compare to others. The stroke rates look similar for all groups.

3. **Residence Type vs. Stroke:**
   - Both 'Rural' and 'Urban' residence show a higher number of people without stroke,
   - There are slightly more people in Urban area with a stroke compared to the Rural areas.
4. **Smoking Status vs. Stroke:**
   - The 'Never smoked' category has the largest number of people without a stroke.
   - The number of people with stroke was higher for people who formerly smoked than those who never smoked.
   - "Unknown" category shows higher frequency for non-stroke patients
5. **Ever Married vs. Stroke:**
   - Both married (Yes) and not married (No) people have more number of people without stroke than with a stroke.
   - Married people have a greater number of people with a stroke.
6. **Heart Disease vs. Stroke:**
   - The number of people without stroke is much larger for both categories of heart disease (0 and 1).
   - The number of people with stroke is higher for people with a heart disease than for those without it.
7. **Hypertension vs. Stroke:**
   - The number of people with hypertension is significantly lower than those without it in the dataset.
   - The number of people with stroke is higher for people with hypertension than for those without it.

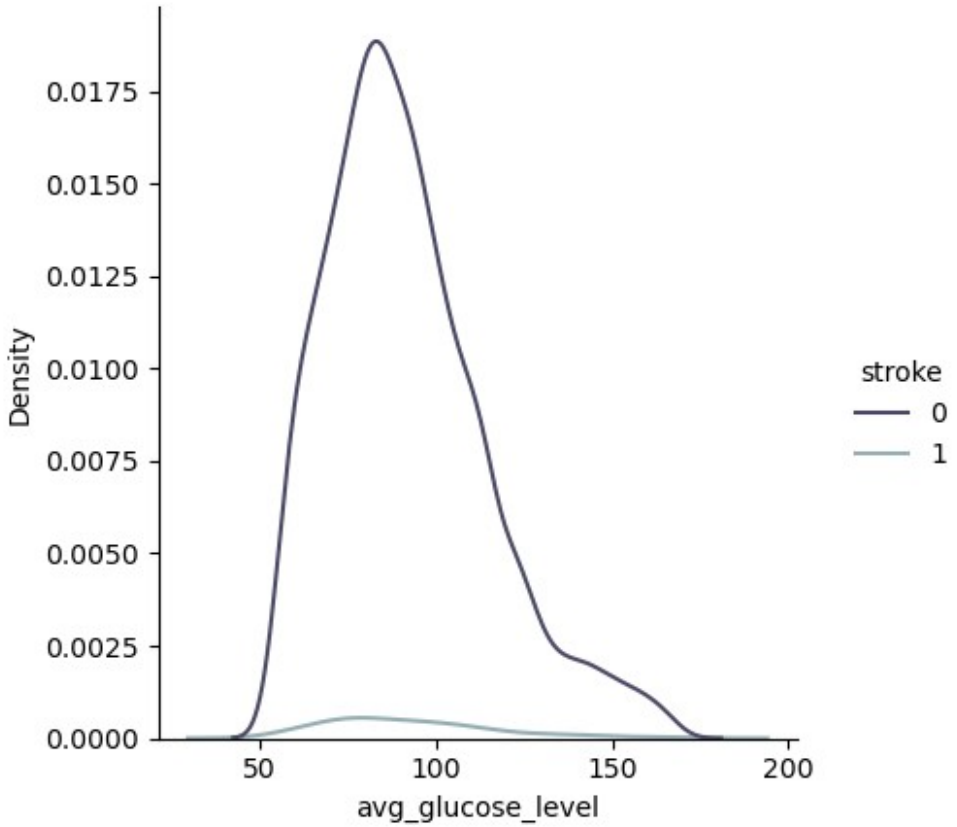
Insights and Interpretations:

- **Stroke and Gender:** The number of individuals who had a stroke was higher for women. The difference is not significantly high.
- **Stroke and Work Type:** Self-employed and Private sector people had higher rates of stroke.
- **Stroke and Residency:** Urban and Rural residency showed nearly similar stroke frequency.
- **Stroke and Smoking:** There are more strokes in the "formerly smoked" group and those who are active smokers than non-smokers.
- **Stroke and Marriage:** Married individuals are at a higher rate of stroke.
- **Stroke and Heart Disease/Hypertension:** The number of people with stroke is higher for people with heart diseases or hypertension compared to those who did not have these conditions.

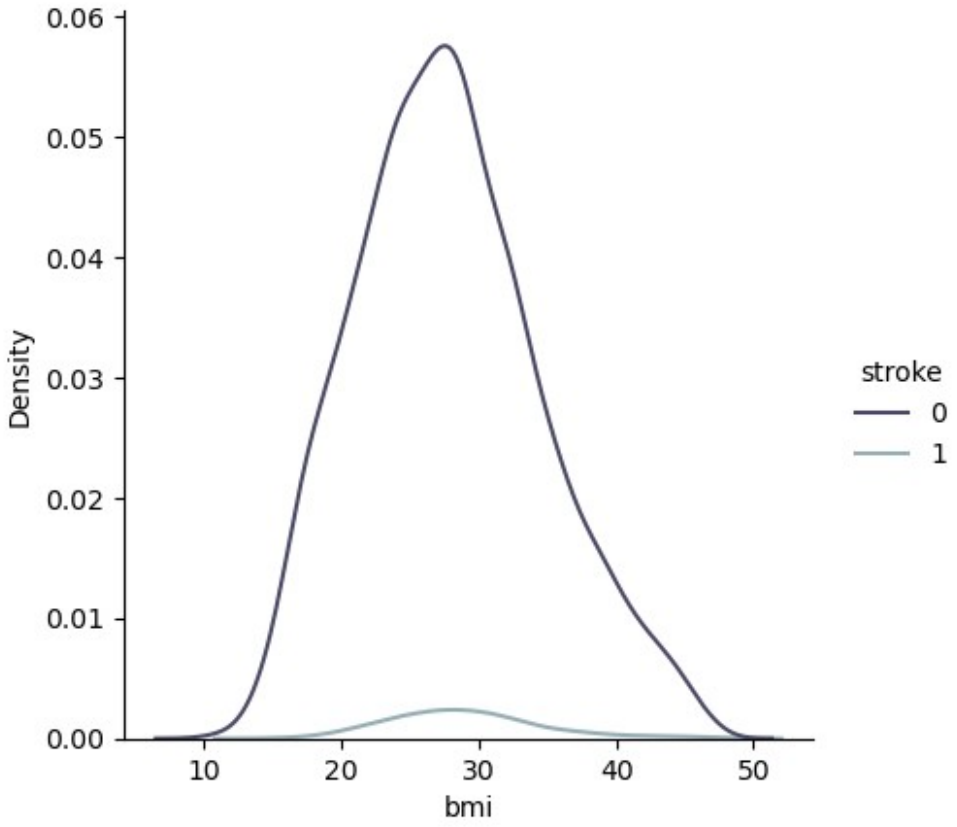
```
sns.displot(data = df , x='age',hue = 'stroke',kind = 'kde',palette =
'bone',height=4.5 )
plt.show()
```



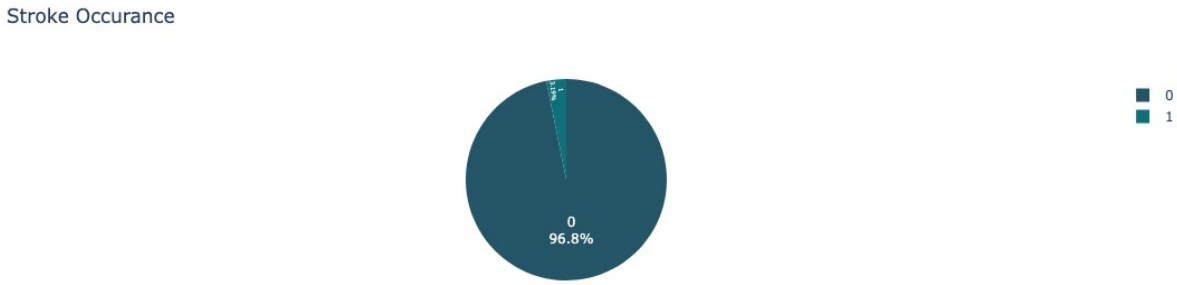

```
sns.displot(data = df , x='avg_glucose_level',hue = 'stroke',kind =
'kde',palette = 'bone',height=4.5 )
plt.show()
```




```
sns.displot(data = df , x='bmi',hue = 'stroke',kind = 'kde',palette =
'bone',height=4.5 )
plt.show()
```




```
stroke = dict(df['stroke'].value_counts())

fig = px.pie(names = stroke.keys(),values = stroke.values(),title =
'Stroke
Occurance',color_discrete_sequence=px.colors.sequential.Aggrnyl)
fig.update_traces(textposition='inside', textinfo='percent+label')
```



---

insights from the density plots, focusing on the most important distinctions between the stroke and non-stroke groups:

Concise Summary of Density Plot Insights:

1. Age and Stroke:

- **Higher Age Risk:** Stroke incidence skews towards older age groups compared to individuals without a stroke, who tend to be concentrated in the middle-age range.

2. Average Glucose Level and Stroke:

- **Clear Separation:** Individuals with a stroke tend to have notably *higher* and more *widely spread* average glucose levels, whereas those without a stroke have distributions concentrated in lower glucose level values. This is the most distinct separation seen across the three plots.

3. BMI and Stroke:

- **Higher BMI**: Stroke patients tend to have higher values of BMI.

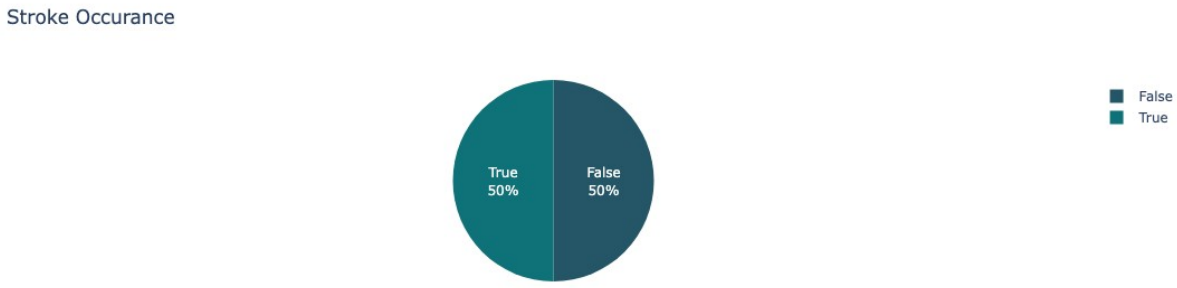
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix , accuracy_score ,
classification_report

from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn .ensemble import RandomForestClassifier
from sklearn .ensemble import VotingClassifier , BaggingClassifier ,
StackingClassifier
```

```
df_0 = df[df.iloc[:,-1]==0]
df_1 = df[df.iloc[:,-1]==1]

df['stroke'].value_counts()
```

```
stroke
0    4122
1     136
Name: count, dtype: int64
```

```
from sklearn.utils import resample
```

```
df_1 = resample(df_1,replace=True , n_samples=df_0.shape[0] ,
random_state=123 )


#concatenate upsampled data
df = np.concatenate((df_0,df_1))

#create the balanced dataframe
df = pd.DataFrame(df)
df.columns = ['gender', 'age', 'hypertension', 'heart_disease',
'ever_married','work_type', 'Residence_type', 'avg_glucose_level',
'bmi','smoking_status', 'stroke']

# visualize balanced data
stroke = dict(df['stroke'].value_counts())
fig = px.pie(names = ['False','True'],values = stroke.values(),title =
'Stroke
Occurance',color_discrete_sequence=px.colors.sequential.Aggrnyl)
fig.update_traces(textposition='inside', textinfo='percent+label')
```



```
# from sklearn.preprocessing import LabelEncoder

# label_encoder = LabelEncoder()
# columns_to_encode
=['gender','ever_married','work_type','Residence_type','smoking_status
']

# for column in columns_to_encode:
#     df[column] = label_encoder.fit_transform(df[column])

df = pd.get_dummies(data =df , columns =
['gender','ever_married','work_type','Residence_type','smoking_status'
] ,drop_first=True )

df.head()
```

```
    age hypertension heart_disease avg_glucose_level   bmi stroke  \
0   3.0            0             0             95.12  18.0      0
1  58.0            1             0             87.96  39.2      0
2   8.0            0             0            110.89  17.6      0
3  70.0            0             0             69.04  35.9      0
4  14.0            0             0            161.28  19.1      0

   gender_Male  ever_married_Yes  work_type_Never_worked
work_type_Private  \
0         True             False                   False
False
1         True              True                   False
True
2        False             False                   False
True
3        False              True                   False
True
4         True             False                    True
False

   work_type_Self-employed  work_type_children
Residence_type_Urban  \
0                    False                True                   False

1                    False               False                    True

2                    False               False                    True

3                    False               False                   False

4                    False               False                   False


   smoking_status_formerly smoked  smoking_status_never smoked  \
0                           False                         False
1                           False                          True
2                           False                         False
3                            True                         False
4                           False                         False

   smoking_status_smokes
0                  False
1                  False
2                  False
3                  False
4                  False
```

```
x = df.drop('stroke', axis = 1)
y = pd.to_numeric( df['stroke'])

x.head()
```

```
    age hypertension heart_disease avg_glucose_level   bmi
gender_Male  \
0   3.0            0             0             95.12  18.0
True
1  58.0            1             0             87.96  39.2
True
2   8.0            0             0            110.89  17.6
False
3  70.0            0             0             69.04  35.9
False
4  14.0            0             0            161.28  19.1
True

   ever_married_Yes  work_type_Never_worked  work_type_Private  \
0             False                   False              False
1              True                   False               True
2             False                   False               True
3              True                   False               True
4             False                    True              False

   work_type_Self-employed  work_type_children
Residence_type_Urban  \
0                    False                True                 False

1                    False               False                  True

2                    False               False                  True

3                    False               False                 False

4                    False               False                 False


   smoking_status_formerly smoked  smoking_status_never smoked  \
0                           False                         False
1                           False                          True
2                           False                         False
3                            True                         False
4                           False                         False

   smoking_status_smokes
0                  False
1                  False
2                  False
3                  False
4                  False
```

```
y.head()
```

```
0    0
1    0
2    0
3    0
4    0
Name: stroke, dtype: int64
```

```
scaler = StandardScaler()

x = scaler.fit_transform(x)
```

```
x_train , x_test , y_train , y_test = train_test_split(x,y,test_size = .20)
```

```
tree_model = DecisionTreeClassifier(criterion='entropy')
tree_model.fit(x_train,y_train)
```

```
DecisionTreeClassifier(criterion='entropy')
```

```
y_pred = tree_model.predict(x_test)

accuracy_score(y_test, y_pred)
```

```
0.9787750151607034
```

```
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(x_train,y_train)
```

```
KNeighborsClassifier(n_neighbors=3)
```

```
y_pred = knn.predict(x_test)
accuracy_score(y_test,y_pred)
```

```
0.9617950272892662
```

```
NB_model = GaussianNB()
NB_model.fit(x_train , y_train)
```

```
GaussianNB()
```

```
y_pred = NB_model.predict(x_test)
accuracy_score(y_test,y_pred)
```

```
0.5712553062462098
```

```
svm = SVC()
svm.fit(x_train , y_train)
```

```
SVC()
```

```
y_pred = svm.predict(x_test)
accuracy_score(y_test,y_pred)
```

```
0.8829593693147362
```

```
lr_model = LogisticRegression()
lr_model.fit(x_train,y_train)
```

```
LogisticRegression()
```

```
y_pred = lr_model.predict(x_test)
accuracy_score(y_test,y_pred)
```

```
0.7568223165554882
```

Ensemble Methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone

---

```
rf_model =
RandomForestClassifier(n_estimators=150,criterion='entropy',random_sta
te = 123)
rf_model.fit(x_train,y_train)
```

```
RandomForestClassifier(criterion='entropy', n_estimators=150,
random_state=123)
```

```
y_pred = rf_model.predict(x_test)
accuracy_score(y_test,y_pred)
```

```
0.9939357186173439
```

Voting Classifier is a Ensamble Learning Technique that trains various base models or estimators and predicts the output on the basis of aggregating the findings of each base estimator

```
svm = SVC()
LR = LogisticRegression()
tree = DecisionTreeClassifier()
knn = KNeighborsClassifier(n_neighbors=3)

models = [('SVM',svm),('Decision Tree',tree),('Logistic
Regerssion',LR) , ('KNN',knn)]

voting_model = VotingClassifier(
    estimators= models
)


voting_model.fit(x_train, y_train)
```

```
VotingClassifier(estimators=[('SVM', SVC()),
                             ('Decision Tree',
DecisionTreeClassifier()),
                             ('Logistic Regerssion',
LogisticRegression()),
                             ('KNN',
KNeighborsClassifier(n_neighbors=3))])
```

```
y_pred = voting_model.predict(x_test)
accuracy_score(y_test,y_pred)
```

```
0.9393571861734384
```

```
bagging = BaggingClassifier(

    n_estimators = 10
)


bagging.fit(x_train , y_train)
```

```
BaggingClassifier()
```

```
y_pred = bagging.predict(x_test)
accuracy_score(y_test,y_pred)
```

```
0.9878714372346877
```

```
base_models = [('SVM',SVC()),('Decision
Tree',DecisionTreeClassifier()),('Logistic
Regerssion',LogisticRegression()) ,
('KNN',KNeighborsClassifier(n_neighbors=3))]
stacking = StackingClassifier(
    estimators = base_models ,
    final_estimator = LogisticRegression(),
    cv = 5
)

stacking.fit(x_train , y_train)
```

```
StackingClassifier(cv=5,
                   estimators=[('SVM', SVC()),
                               ('Decision Tree',
DecisionTreeClassifier()),
                               ('Logistic Regerssion',
LogisticRegression()),
                               ('KNN',
KNeighborsClassifier(n_neighbors=3))],
                   final_estimator=LogisticRegression())
```

```
y_pred = stacking.predict(x_test)
accuracy_score(y_test,y_pred)
```

```
0.9939357186173439
```

Table: Summary of Stroke Prediction Project

Category	Item	Summary
Project Overview	**Goal**	Develop a predictive model for stroke risk using machine learning.
	Data	5110 records with demographic, health, and lifestyle information; imbalanced stroke cases.
	Methods	Data cleaning, EDA, machine learning model training and evaluation, including ensemble methods.
Key Findings	**Age**	Risk of stroke increases with age, particularly after 60.
	Glucose Level	Elevated average glucose levels strongly indicate increased stroke risk.
	BMI	Higher BMI may slightly increase the likelihood of stroke.
	Smoking Status	Active smokers and former smokers show higher stroke rates than non-smokers.
	Heart Disease/Hypertension	History of heart disease or hypertension significantly increases the risk of stroke.
	Work Type	Private sector or self-employed individuals show higher stroke incidence than others.
	Residence Type	No significant difference in stroke rates was observed between urban and rural residents.
	Model Performance	Ensemble models (Random Forest and Stacking Classifier) achieved highest accuracy (near 99.4%). Decision Tree and KNN also performed well. Naive Bayes performed poorly
Model Results	**Model**	**Accuracy**
	Decision Tree	0.9788
	KNN	0.9618
	Gaussian Naive Bayes	0.5713
	Support Vector Classifier	0.8830
	Logistic Regression	0.7568
	Random Forest	0.9939
	Bagging	0.9879
	Voting Classifier	0.9839

Category	Item	Summary
	Stacking Classifier	0.9939
Implications	**Model Selection**	Model selection is critical, with tree based models showing higher accuracy
	Data Bias	Imbalance in stroke cases and gender proportions may affect model training.
	Feature Importance	Average glucose level, age, smoking and heart disease are important predictors of stroke.
Recommendations	**Evaluation Metrics**	Use metrics beyond accuracy: precision, recall, F1-score, and AUC.
	Hyperparameter Tuning	Optimize the performance of classification models through hyperparameter tuning.
	Cross-validation	Implement cross-validation for robust model evaluation.
	Further Exploration	Evaluate other models and ensemble methods to find better models than what was trained here.
	Public Health	Promote early detection, and healthier lifestyles to reduce stroke risk.

Explanation of the Table:

- **Category:** Organizes the information into distinct sections (Project Overview, Key Findings, Implications, Recommendations).
- **Item:** Specific item that provides a brief idea about the row
- **Summary:** Provides concise and key information related to the item.

Benefits of Table Format:

- **Organization:** The table presents information clearly in a structured manner.
- **Readability:** It enhances readability and makes it easy to find key pieces of information.
- **Conciseness**: Helps in a summary of the key information.