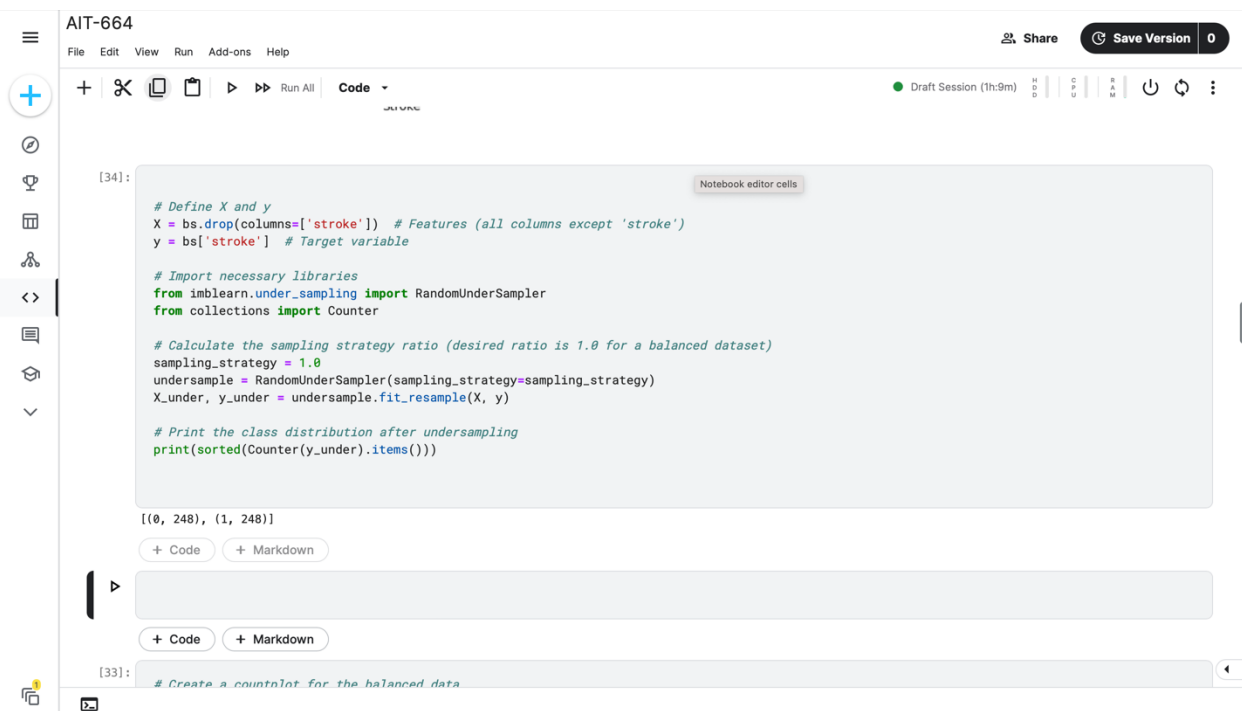


BRAIN STROKE PREDICTION

Data visualization plays a vital role in understanding complex datasets. In this report, we will visualize the Brain Stroke Prediction dataset obtained from Kaggle, curated by Izzet Turkalp Akbasli. Our goal is to gain insights into factors contributing to strokes and predict the likelihood of stroke occurrence from the visualizations that we generated. The target audience includes healthcare professionals, researchers, and policymakers interested in stroke prevention strategies.

In the Part 2 of the Data analysis, I generated the visualizations before balancing the data. For accurate results, in this part I generated the visualizations after the under sampling. Balancing the data is not always required for visualizations, but it can provide a clearer understanding of the relationships between features and the target variable, especially when dealing with imbalanced datasets. By balancing the data, we ensure that both classes have an equal representation, allowing for a fair comparison between different categories and making the insights drawn from the visualizations more reliable.

Under sampling:



```
AIT-664
File Edit View Run Add-ons Help
+ ✂ 📄 📄 ▶ ▶▶ Run All Code
Draft Session (1h:9m) C O D H E A R ⏻ ↺ ⋮

[34]:
# Define X and y
X = bs.drop(columns=['stroke']) # Features (all columns except 'stroke')
y = bs['stroke'] # Target variable

# Import necessary libraries
from imblearn.under_sampling import RandomUnderSampler
from collections import Counter

# Calculate the sampling strategy ratio (desired ratio is 1.0 for a balanced dataset)
sampling_strategy = 1.0
undersample = RandomUnderSampler(sampling_strategy=sampling_strategy)
X_under, y_under = undersample.fit_resample(X, y)

# Print the class distribution after undersampling
print(sorted(Counter(y_under).items()))

[(0, 248), (1, 248)]
+ Code + Markdown

▶

+ Code + Markdown

[33]: # Create a countplot for the balanced data
```

Fig (1)



Fig (2)

Visualization Techniques Used:

I used pie charts, count plot, histogram, box plot and heat map for visualizations, I will discuss where and why I used these plots.

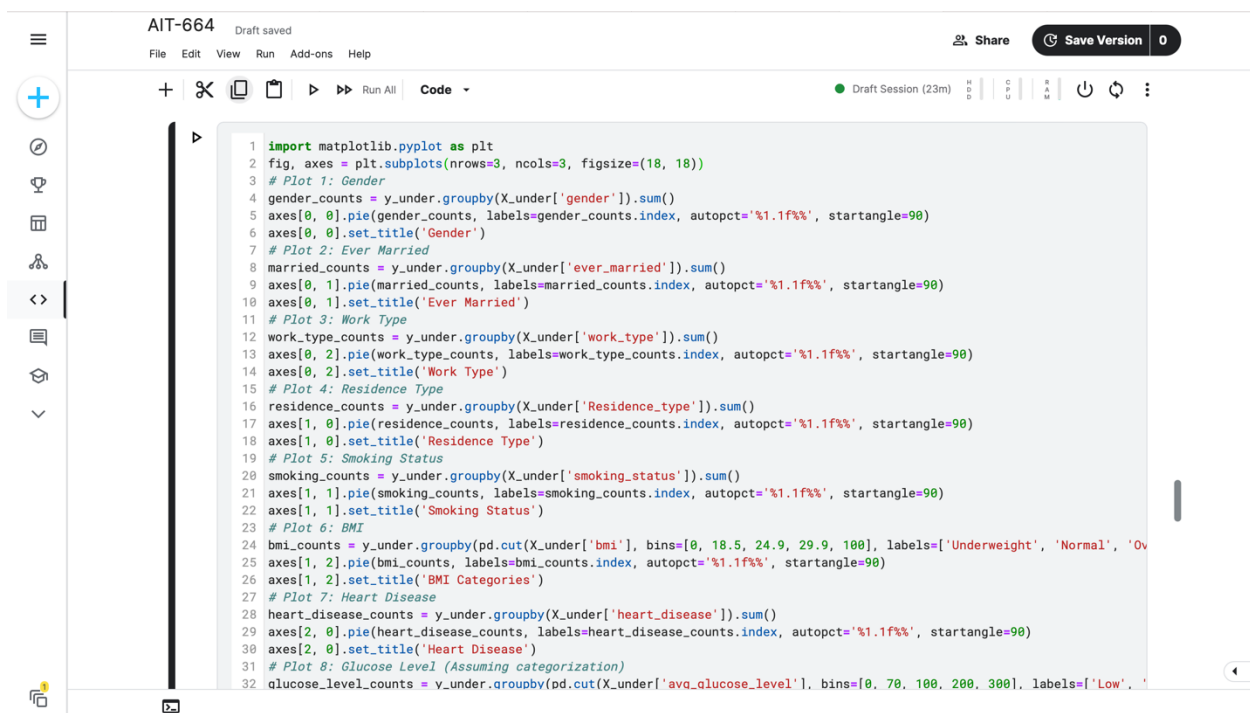
- ☐ **Pie charts:** I used pie charts for the relations for categorical values with stroke cases as 'yes' i.e., who had stroke. Pie charts are suitable for showing parts of a whole and are effective when we want to visualize the relative proportions of categories within a single feature. In this case, they help provide a clear comparison of stroke cases across different categorical features.

For correlations with target value as stroke:

- ☐ **Count plot: (Gender, Hypertension, Heart Disease, Ever Married, Work Type, Residence Type, and Smoking Status).** Count plots are useful for displaying the count of categorical variables. In this context, they show the distribution of stroke and no-stroke cases within different categorical features. For example, it helps visualize how many people had strokes based on their gender, marital status, work type, etc.

- **Histogram: (age)** A histogram provides a visual representation of the distribution of age. Using multiple stacked distributions allows to compare the age distribution for people with and without strokes.
- **Boxplot: (Average Glucose Level and BMI).** Boxplots are effective for displaying the distribution, central tendency, and spread of numerical variables. They help to identify potential outliers and visualize differences in the average glucose level and BMI between stroke and no-stroke cases.

Visualizations and Interpretations:



```

1 import matplotlib.pyplot as plt
2 fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(18, 18))
3 # Plot 1: Gender
4 gender_counts = y_under.groupby(X_under['gender']).sum()
5 axes[0, 0].pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', startangle=90)
6 axes[0, 0].set_title('Gender')
7 # Plot 2: Ever Married
8 married_counts = y_under.groupby(X_under['ever_married']).sum()
9 axes[0, 1].pie(married_counts, labels=married_counts.index, autopct='%1.1f%%', startangle=90)
10 axes[0, 1].set_title('Ever Married')
11 # Plot 3: Work Type
12 work_type_counts = y_under.groupby(X_under['work_type']).sum()
13 axes[0, 2].pie(work_type_counts, labels=work_type_counts.index, autopct='%1.1f%%', startangle=90)
14 axes[0, 2].set_title('Work Type')
15 # Plot 4: Residence Type
16 residence_counts = y_under.groupby(X_under['Residence_type']).sum()
17 axes[1, 0].pie(residence_counts, labels=residence_counts.index, autopct='%1.1f%%', startangle=90)
18 axes[1, 0].set_title('Residence Type')
19 # Plot 5: Smoking Status
20 smoking_counts = y_under.groupby(X_under['smoking_status']).sum()
21 axes[1, 1].pie(smoking_counts, labels=smoking_counts.index, autopct='%1.1f%%', startangle=90)
22 axes[1, 1].set_title('Smoking Status')
23 # Plot 6: BMI
24 bmi_counts = y_under.groupby(pd.cut(X_under['bmi'], bins=[0, 18.5, 24.9, 29.9, 100], labels=['Underweight', 'Normal', 'Overweight', 'Obese'])).sum()
25 axes[1, 2].pie(bmi_counts, labels=bmi_counts.index, autopct='%1.1f%%', startangle=90)
26 axes[1, 2].set_title('BMI Categories')
27 # Plot 7: Heart Disease
28 heart_disease_counts = y_under.groupby(X_under['heart_disease']).sum()
29 axes[2, 0].pie(heart_disease_counts, labels=heart_disease_counts.index, autopct='%1.1f%%', startangle=90)
30 axes[2, 0].set_title('Heart Disease')
31 # Plot 8: Glucose Level (Assuming categorization)
32 glucose_level_counts = y_under.groupby(pd.cut(X_under['avg_glucose_level'], bins=[0, 70, 100, 200, 300], labels=['Low', 'Medium', 'High'])).sum()

```

Fig (3a)

AIT-664Draft saved

FileEditViewRunAdd-onsHelp

ShareSave Version0

Run AllCode

Draft Session (22m)

+

```
8 married_counts = y_under.groupby(X_under['ever_married']).sum()
9 axes[0, 1].pie(married_counts, labels=married_counts.index, autopct='%1.1f%%', startangle=90)
10 axes[0, 1].set_title('Ever Married')
11 # Plot 3: Work Type
12 work_type_counts = y_under.groupby(X_under['work_type']).sum()
13 axes[0, 2].pie(work_type_counts, labels=work_type_counts.index, autopct='%1.1f%%', startangle=90)
14 axes[0, 2].set_title('Work Type')
15 # Plot 4: Residence Type
16 residence_counts = y_under.groupby(X_under['Residence_type']).sum()
17 axes[1, 0].pie(residence_counts, labels=residence_counts.index, autopct='%1.1f%%', startangle=90)
18 axes[1, 0].set_title('Residence Type')
19 # Plot 5: Smoking Status
20 smoking_counts = y_under.groupby(X_under['smoking_status']).sum()
21 axes[1, 1].pie(smoking_counts, labels=smoking_counts.index, autopct='%1.1f%%', startangle=90)
22 axes[1, 1].set_title('Smoking Status')
23 # Plot 6: BMI
24 bmi_counts = y_under.groupby(pd.cut(X_under['bmi'], bins=[0, 18.5, 24.9, 29.9, 100], labels=['Underweight', 'Normal', 'Overweight', 'Obese'])).sum()
25 axes[1, 2].pie(bmi_counts, labels=bmi_counts.index, autopct='%1.1f%%', startangle=90)
26 axes[1, 2].set_title('BMI Categories')
27 # Plot 7: Heart Disease
28 heart_disease_counts = y_under.groupby(X_under['heart_disease']).sum()
29 axes[2, 0].pie(heart_disease_counts, labels=heart_disease_counts.index, autopct='%1.1f%%', startangle=90)
30 axes[2, 0].set_title('Heart Disease')
31 # Plot 8: Glucose Level (Assuming categorization)
32 glucose_level_counts = y_under.groupby(pd.cut(X_under['avg_glucose_level'], bins=[0, 70, 100, 200, 300], labels=['Low', 'Medium', 'High'])).sum()
33 axes[2, 1].pie(glucose_level_counts, labels=glucose_level_counts.index, autopct='%1.1f%%', startangle=90)
34 axes[2, 1].set_title('Glucose Level Categories')
35 # Plot 9: Hypertension
36 hypertension_counts = y_under.groupby(X_under['hypertension']).sum()
37 axes[2, 2].pie(hypertension_counts, labels=hypertension_counts.index, autopct='%1.1f%%', startangle=90)
38 axes[2, 2].set_title('Hypertension')
39 plt.tight_layout()
40 plt.show()
41
```

Fig (3b)

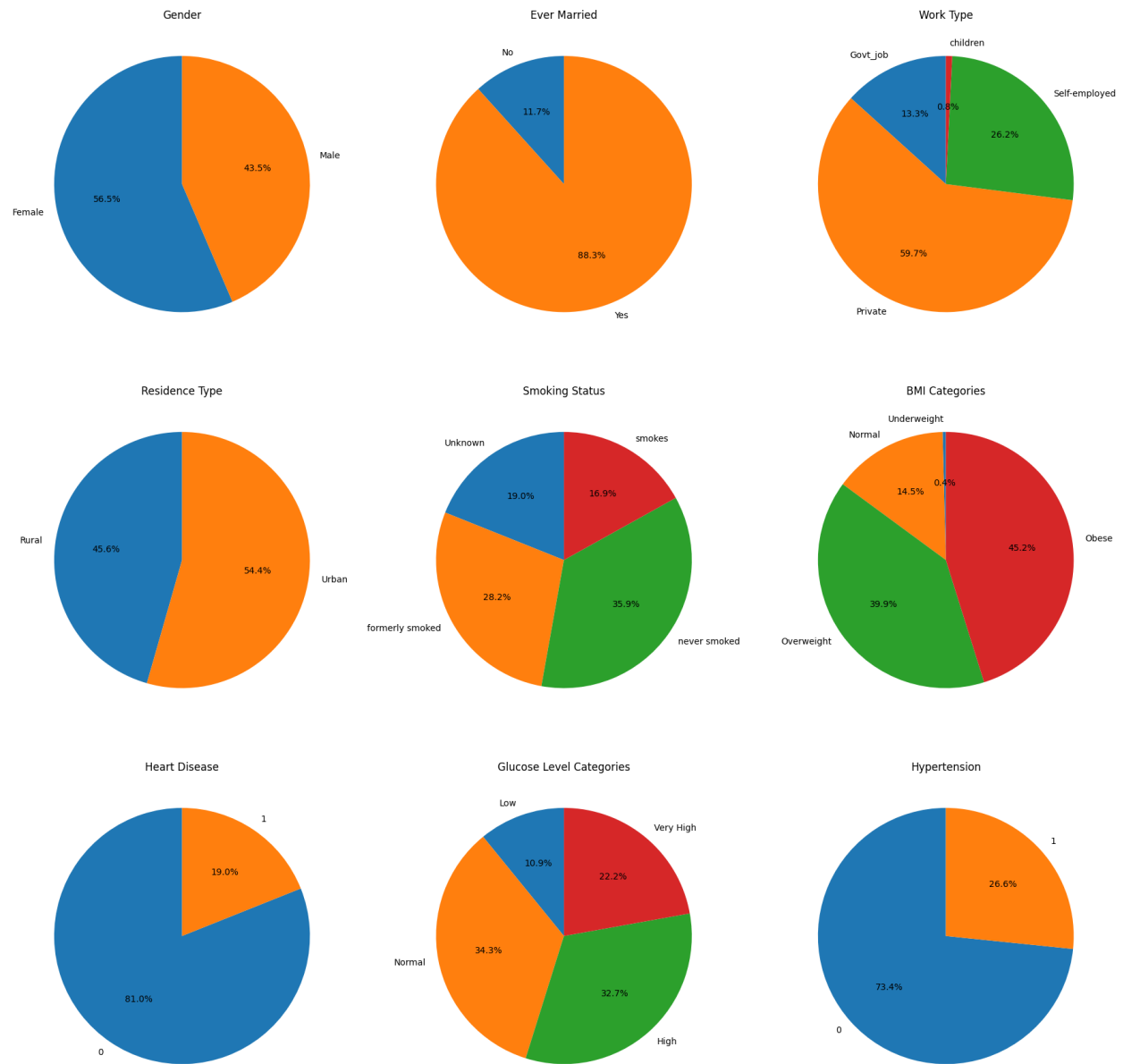


Fig (3c)

In Fig (3c) we can see that I generated pie charts for Gender, ever married, Work type, Residence type, Smoking status for the Stroke status as Yes.

These Charts gives us the following information,

- ☐ From the data collected, 56.5% Females have strokes and 43.5% Males.
- ☐ We can see that more than half i.e., 59.7% of strokes are from people who have private jobs compared to Govt jobs, self-employment, and children.
- ☐ People in urban areas are higher to have stroke.

- 28.2% of former smokers and 16.9 % of smokers have had strokes and 35.9% of non-smokers still tend to have stroke.
- The BMI information is in numerical data, to give a clear understanding I used bin values to categorize. These categories are defined by health organizations and are widely used to classify individuals into different weight status groups.
Underweight: BMI less than 18.5
Normal: BMI between 18.5 and 24.9
Overweight: BMI between 25.0 and 29.9
Obese: BMI 30.0 and above
- 45.2% of the people who have stroke are the ones who are obese.
- The values for the glucose level categories ('Low', 'Normal', 'High', 'Very High') and their corresponding bins were chosen based on commonly accepted ranges for blood glucose levels. These ranges are commonly used in healthcare to classify blood glucose levels.
Low: Glucose level less than 70 mg/dL
Normal: Glucose level between 70 and 100 mg/dL
High: Glucose level between 101 and 200 mg/dL
Very High: Glucose level 201 mg/dL and above
- 32.7% of people have higher glucose levels.

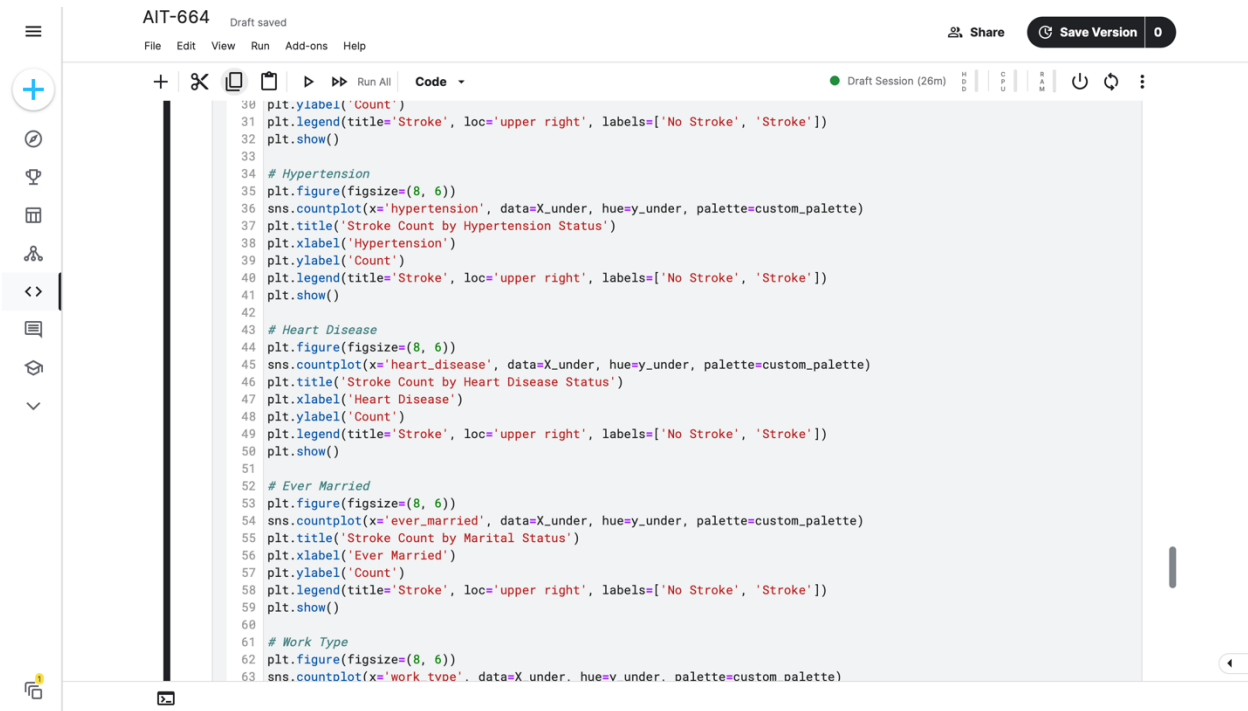
Visualization with target variable as stroke:

```

1 import pandas as pd
2 import seaborn as sns
3 import matplotlib.pyplot as plt
4 from imblearn.under_sampling import RandomUnderSampler
5 from collections import Counter
6
7 X = bs.drop(columns=['stroke'])
8 y = bs['stroke']
9 sampling_strategy = 1.0
10
11 undersample = RandomUnderSampler(sampling_strategy=sampling_strategy)
12 X_under, y_under = undersample.fit_resample(X, y)
13
14 custom_palette = {0: '#00FF00', 1: '#FF0000'}
15
16 # Gender
17 plt.figure(figsize=(8, 6))
18 sns.countplot(x='gender', data=X_under, hue=y_under, palette=custom_palette)
19 plt.title('Stroke Count by Gender')
20 plt.xlabel('Gender')
21 plt.ylabel('Count')
22 plt.legend(title='Stroke', loc='upper right', labels=['No Stroke', 'Stroke'])
23 plt.show()
24
25 # Age
26 plt.figure(figsize=(8, 6))
27 sns.histplot(data=X_under, x='age', hue=y_under, multiple="stack", kde=True, palette=custom_palette)
28 plt.title('Age Distribution by Stroke Status')
29 plt.xlabel('Age')
30 plt.ylabel('Count')
31 plt.legend(title='Stroke', loc='upper right', labels=['No Stroke', 'Stroke'])
32 plt.show()
33

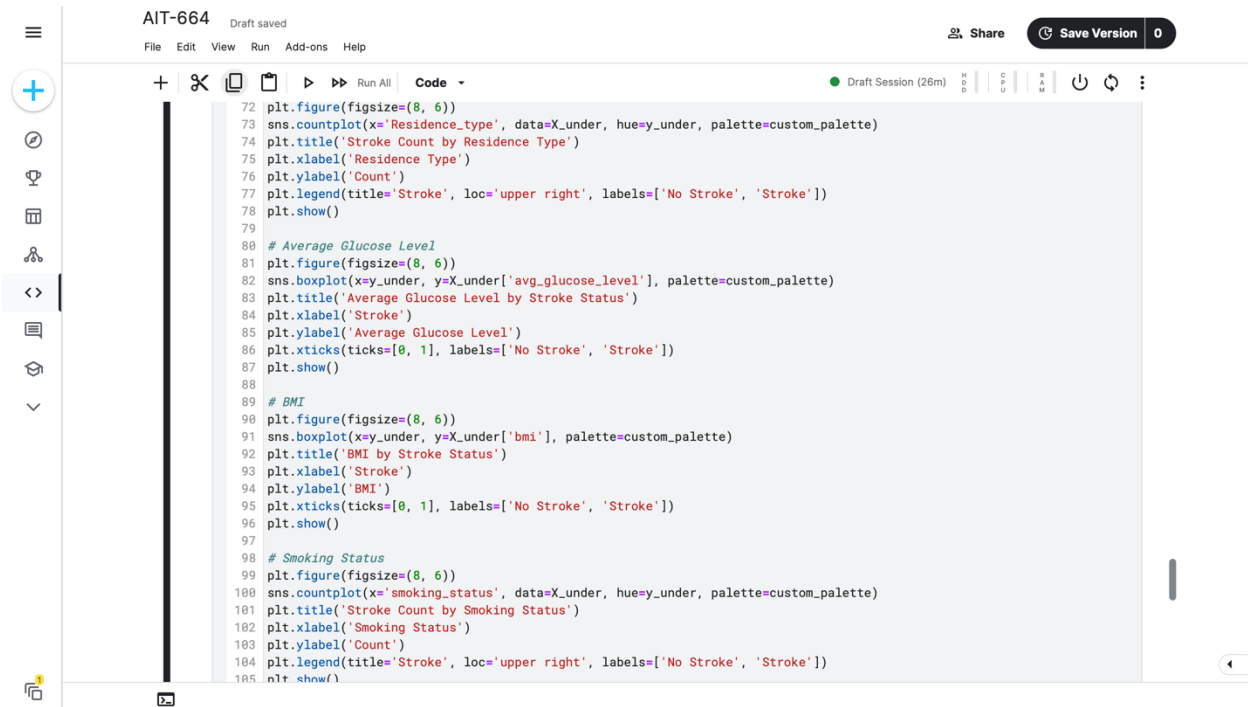
```

Fig (4a)



```
30 plt.ylabel('Count')
31 plt.legend(title='Stroke', loc='upper right', labels=['No Stroke', 'Stroke'])
32 plt.show()
33
34 # Hypertension
35 plt.figure(figsize=(8, 6))
36 sns.countplot(x='hypertension', data=X_under, hue=y_under, palette=custom_palette)
37 plt.title('Stroke Count by Hypertension Status')
38 plt.xlabel('Hypertension')
39 plt.ylabel('Count')
40 plt.legend(title='Stroke', loc='upper right', labels=['No Stroke', 'Stroke'])
41 plt.show()
42
43 # Heart Disease
44 plt.figure(figsize=(8, 6))
45 sns.countplot(x='heart_disease', data=X_under, hue=y_under, palette=custom_palette)
46 plt.title('Stroke Count by Heart Disease Status')
47 plt.xlabel('Heart Disease')
48 plt.ylabel('Count')
49 plt.legend(title='Stroke', loc='upper right', labels=['No Stroke', 'Stroke'])
50 plt.show()
51
52 # Ever Married
53 plt.figure(figsize=(8, 6))
54 sns.countplot(x='ever_married', data=X_under, hue=y_under, palette=custom_palette)
55 plt.title('Stroke Count by Marital Status')
56 plt.xlabel('Ever Married')
57 plt.ylabel('Count')
58 plt.legend(title='Stroke', loc='upper right', labels=['No Stroke', 'Stroke'])
59 plt.show()
60
61 # Work Type
62 plt.figure(figsize=(8, 6))
63 sns.countplot(x='work_type', data=X_under, hue=y_under, palette=custom_palette)
```

Fig (4b)



```
72 plt.figure(figsize=(8, 6))
73 sns.countplot(x='Residence_type', data=X_under, hue=y_under, palette=custom_palette)
74 plt.title('Stroke Count by Residence Type')
75 plt.xlabel('Residence Type')
76 plt.ylabel('Count')
77 plt.legend(title='Stroke', loc='upper right', labels=['No Stroke', 'Stroke'])
78 plt.show()
79
80 # Average Glucose Level
81 plt.figure(figsize=(8, 6))
82 sns.boxplot(x=y_under, y=X_under['avg_glucose_level'], palette=custom_palette)
83 plt.title('Average Glucose Level by Stroke Status')
84 plt.xlabel('Stroke')
85 plt.ylabel('Average Glucose Level')
86 plt.xticks(ticks=[0, 1], labels=['No Stroke', 'Stroke'])
87 plt.show()
88
89 # BMI
90 plt.figure(figsize=(8, 6))
91 sns.boxplot(x=y_under, y=X_under['bmi'], palette=custom_palette)
92 plt.title('BMI by Stroke Status')
93 plt.xlabel('Stroke')
94 plt.ylabel('BMI')
95 plt.xticks(ticks=[0, 1], labels=['No Stroke', 'Stroke'])
96 plt.show()
97
98 # Smoking Status
99 plt.figure(figsize=(8, 6))
100 sns.countplot(x='smoking_status', data=X_under, hue=y_under, palette=custom_palette)
101 plt.title('Stroke Count by Smoking Status')
102 plt.xlabel('Smoking Status')
103 plt.ylabel('Count')
104 plt.legend(title='Stroke', loc='upper right', labels=['No Stroke', 'Stroke'])
105 plt.show()
```

Fig (4c)

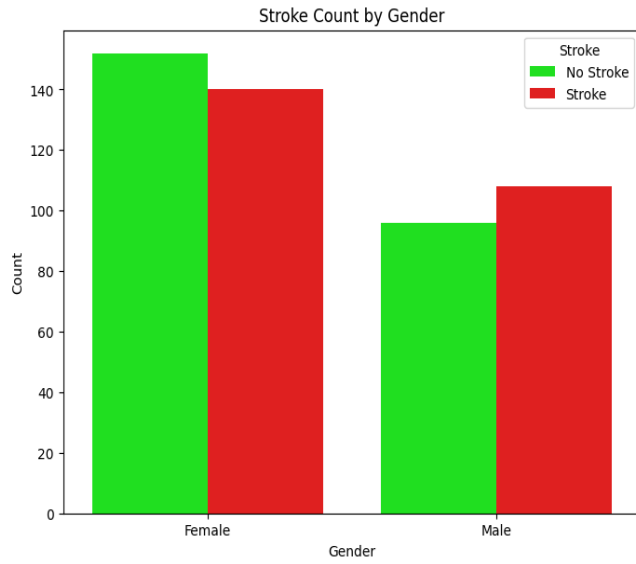


Fig 4.1

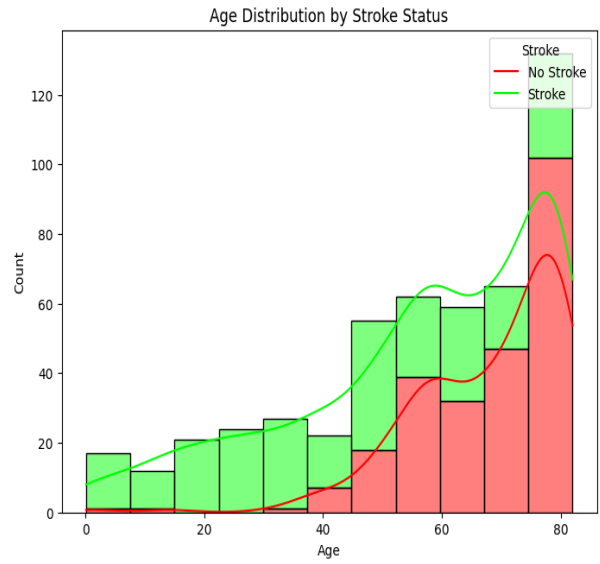


Fig 4.2

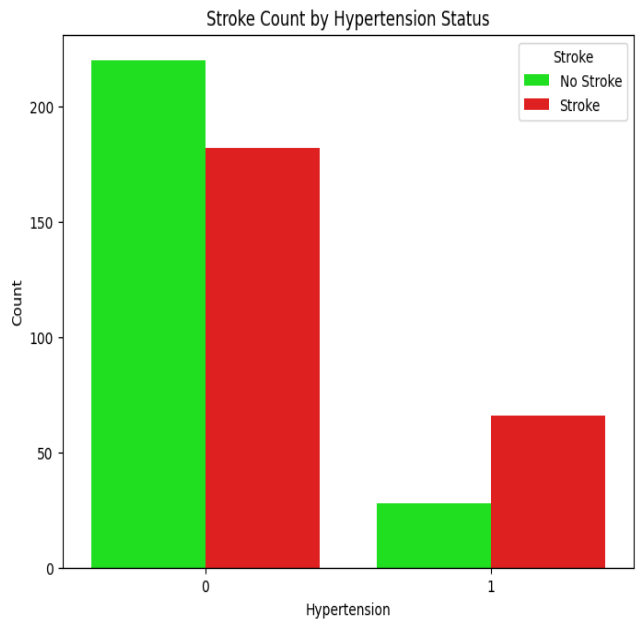


Fig 4.3

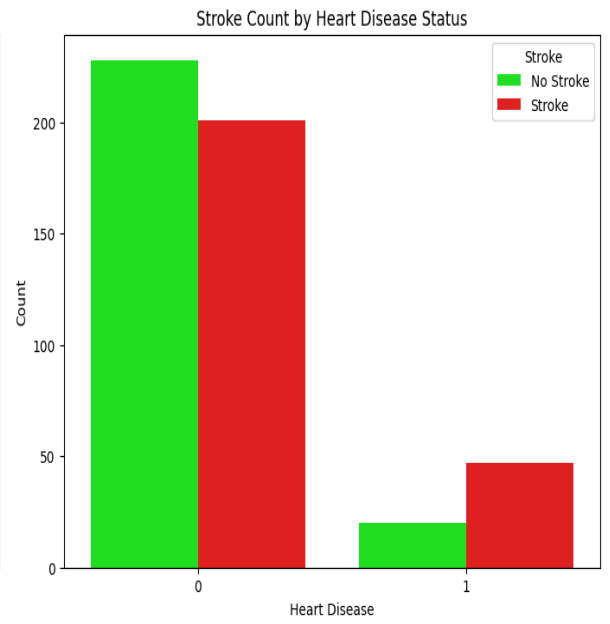


Fig 4.4

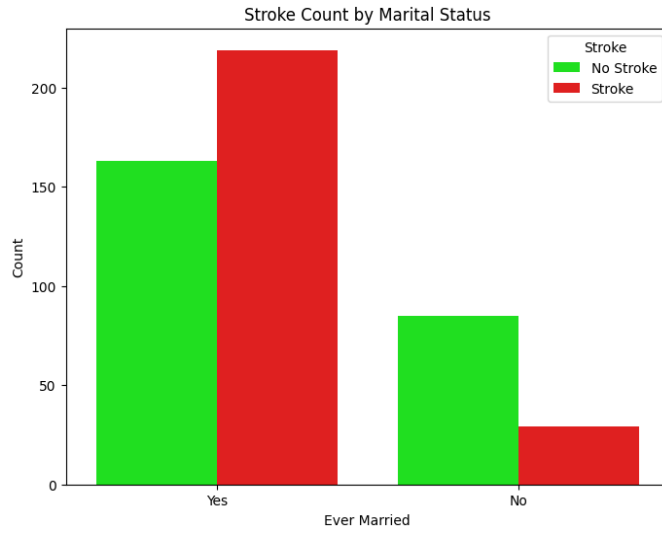


Fig 4.5

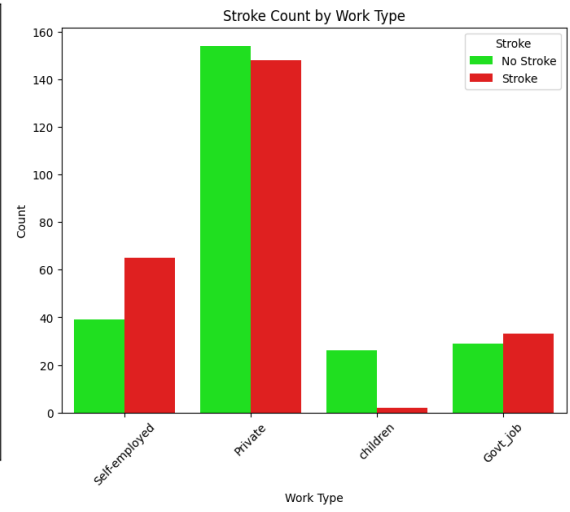


Fig 4.6

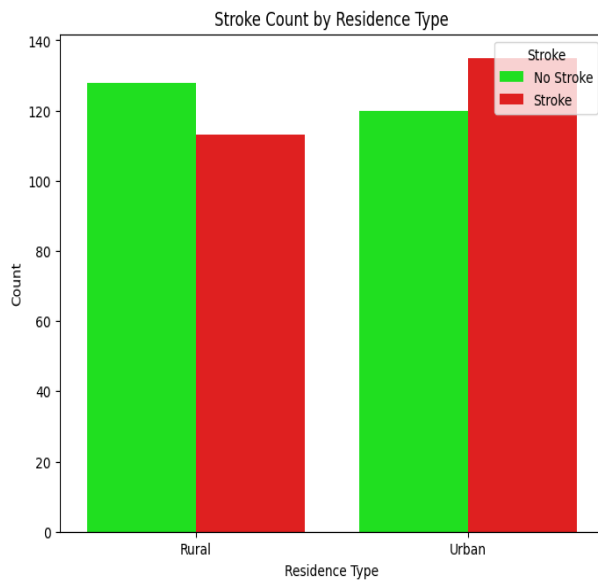


Fig 4.7

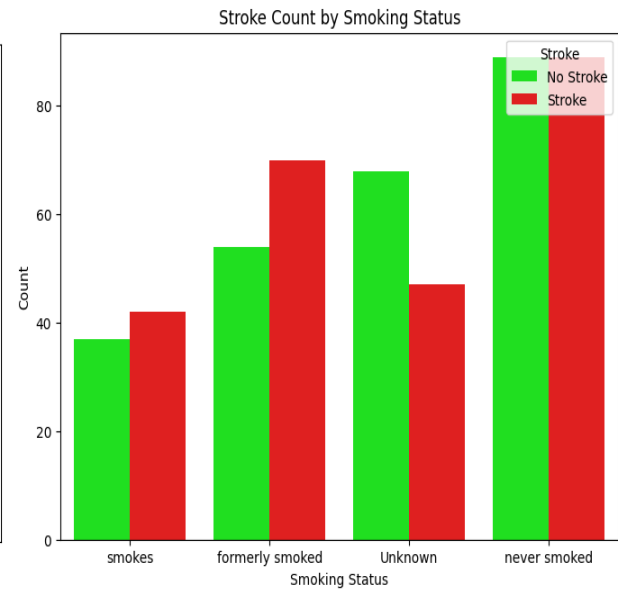


Fig 4.8

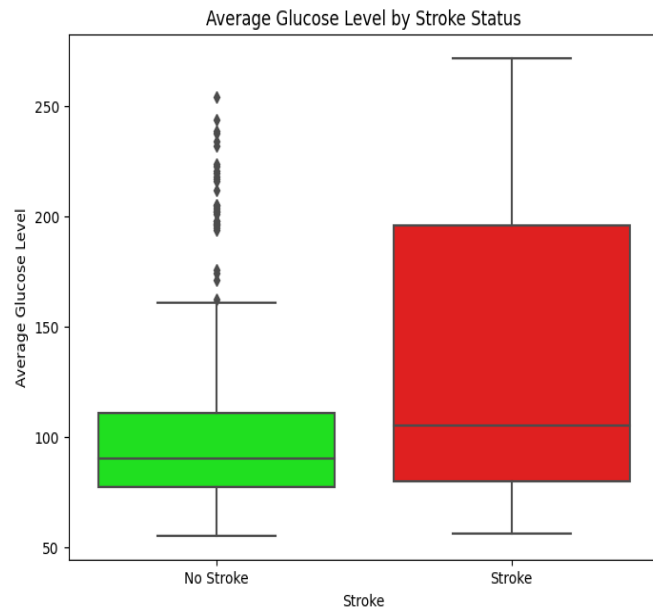


Fig 4.9

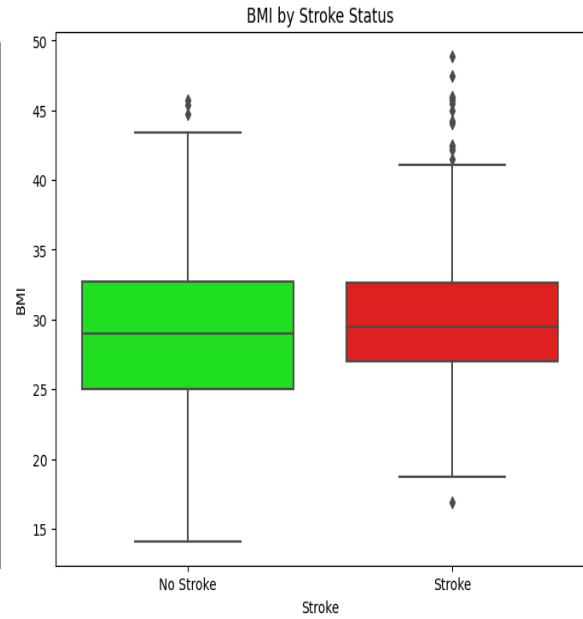


Fig 4.10

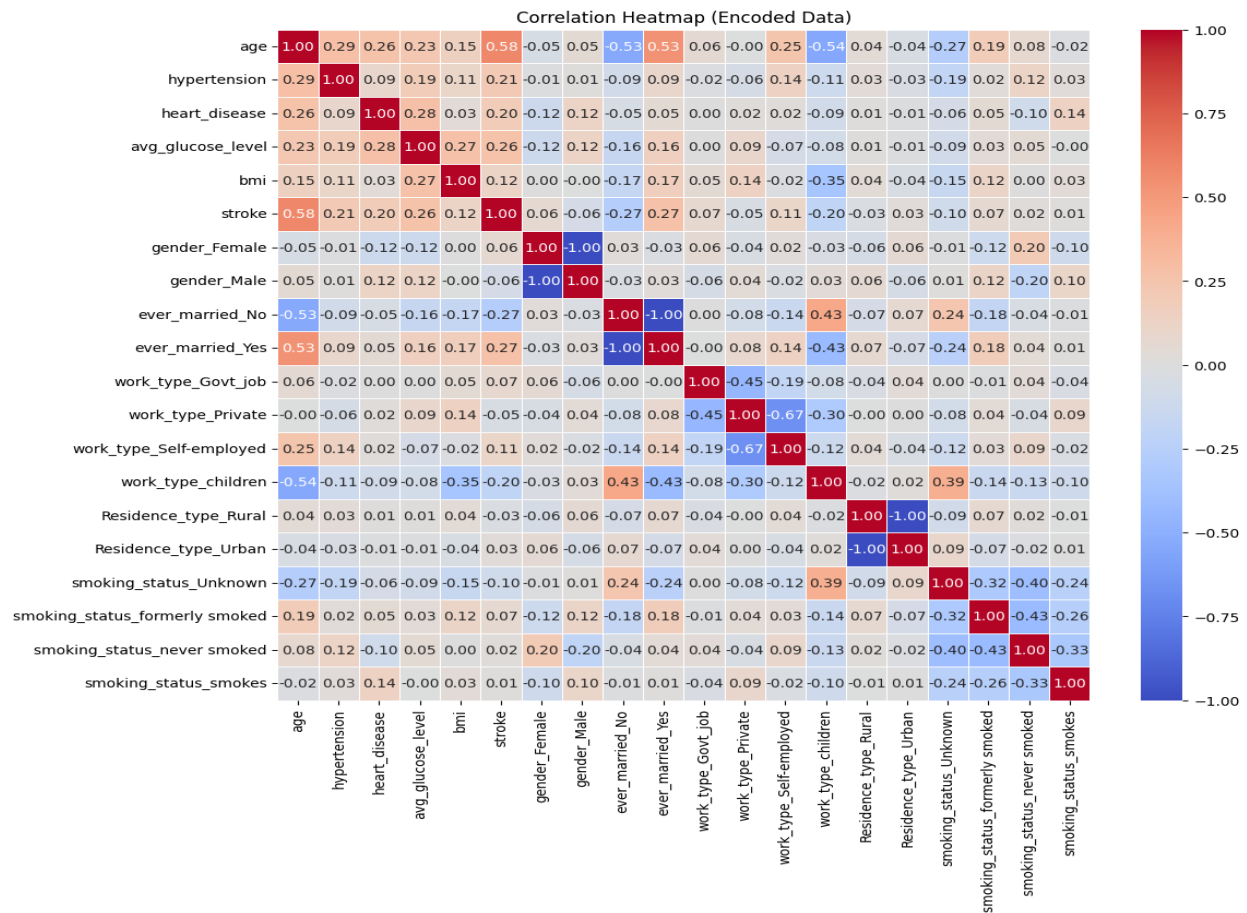


Fig 4.11

- ❑ From the visualizations we can conclude that, Age, Glucose level, smoking status, heart disease and Hypertension are the major factors for strokes.
- ❑ Gender and Stroke Incidence Shows a higher incidence of strokes in males compared to females. Males may have specific risk factors contributing to a higher stroke rate.
- ❑ Age, Average Glucose Level Indicates a positive correlation between age and average glucose level. Higher average glucose levels might be associated with aging, potentially contributing to stroke risk.
- ❑ Correlation between Features Highlights positive correlations between hypertension and age, as well as hypertension and heart disease.
- ❑ Stroke Cases by Smoking Status Illustrates a higher percentage of strokes in individuals who used to smoke.

Key take aways:

- ❑ Hypertension appears to be linked to both age and heart disease, indicating the importance of managing hypertension for preventing strokes in individuals with heart conditions.
- ❑ Former smokers might still face increased stroke risks even after quitting. Smoking cessation programs are essential for reducing stroke incidence.
- ❑ Monitoring glucose levels in elderly individuals is crucial for stroke prevention.
- ❑ Being overweight or obese can lead to other risk factors such as hypertension, diabetes, and heart disease.

Preventions measures to reduce the risk:

- ❑ Regular Health Check-ups
- ❑ Healthy Lifestyle Choices
- ❑ Medication Adherence
- ❑ Stress Management
- ❑ Awareness and Education

Conclusion:

In conclusion, through comprehensive data visualization, we have identified significant factors influencing stroke occurrences and outlined actionable preventive measures. By leveraging these insights, healthcare professionals can develop targeted interventions, leading to improved stroke prevention strategies and overall public health.

References:

- [1] Akbasli, I. T. (2022, July 16). *Brain stroke prediction dataset*. Kaggle.
<https://www.kaggle.com/datasets/zzettrkalpakbal/full-filled-brain-stroke-dataset>
- [2] World Health Organization. (n.d.). *World Health Organization (WHO)*. World Health Organization. <https://www.who.int/>
- [3] National Health Service. (n.d.). NHS choices. <https://www.nhs.uk/>