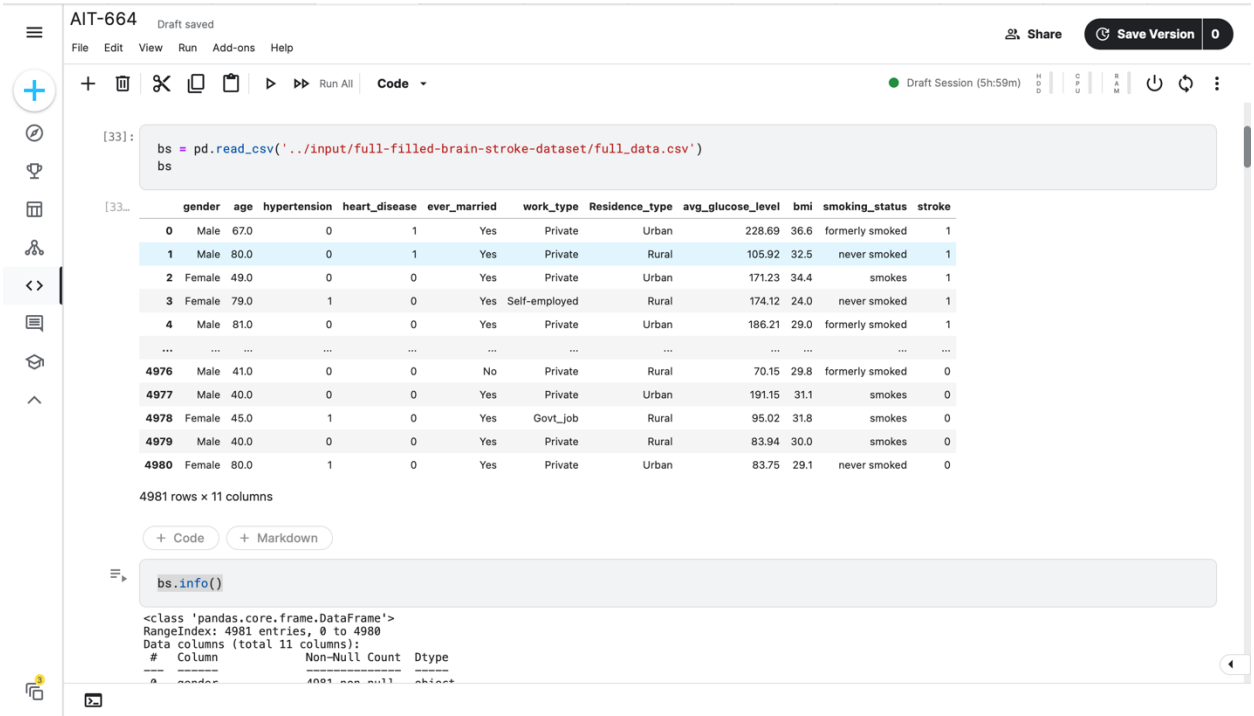
AIT 664 Part 2- Data Preparation & Information Modeling Report
G01394251
sai shivani rakam

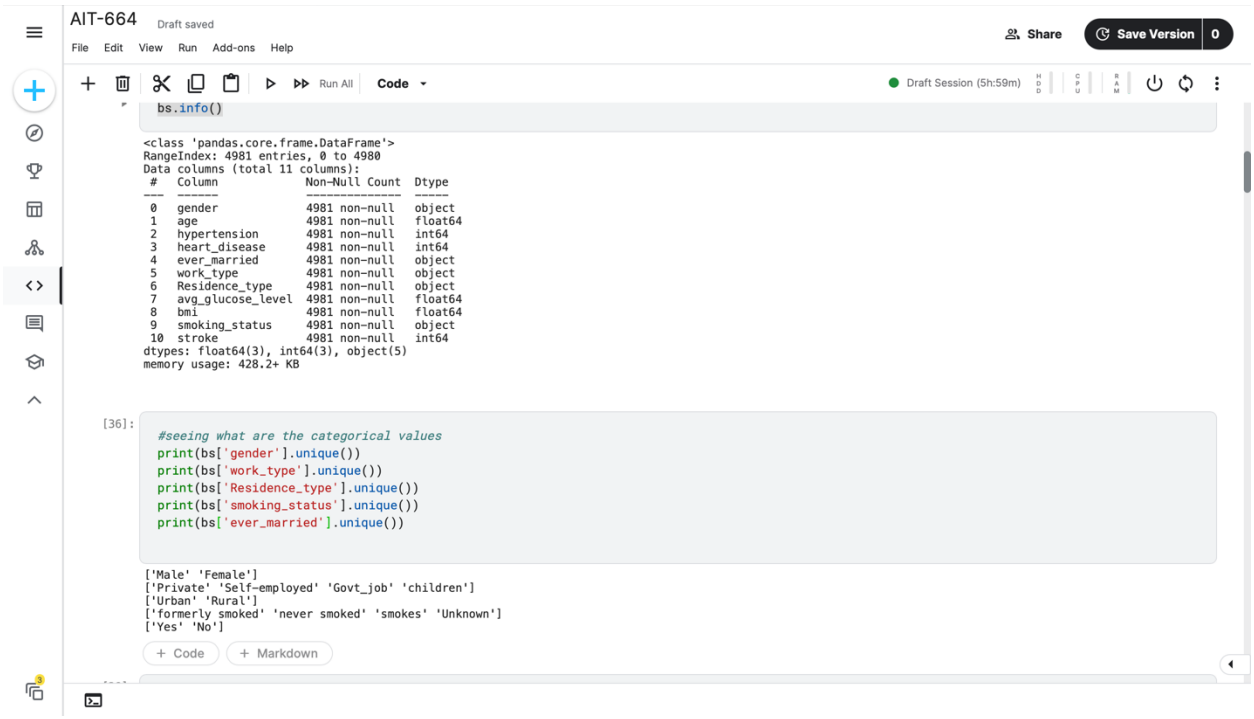
Brain stroke prediction

In this part of data preparation and information modeling report, I will be collecting the data that is related to my search area and exploring it to clean the data, perform analysis and generate results.

1. Data Collection and importing.

The data for this analysis was collected from Kaggle. The dataset focuses on the key areas related to brain stroke, i.e., the main domains that cause brain strokes which is relevant to my research area.

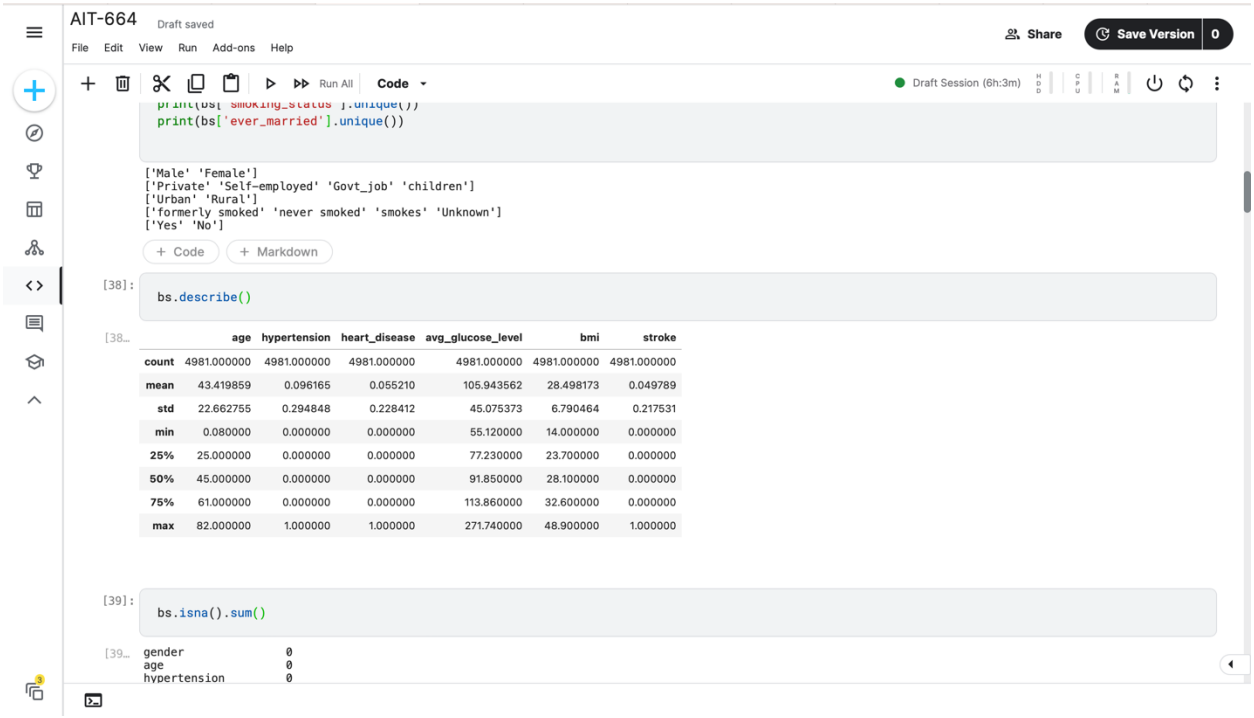




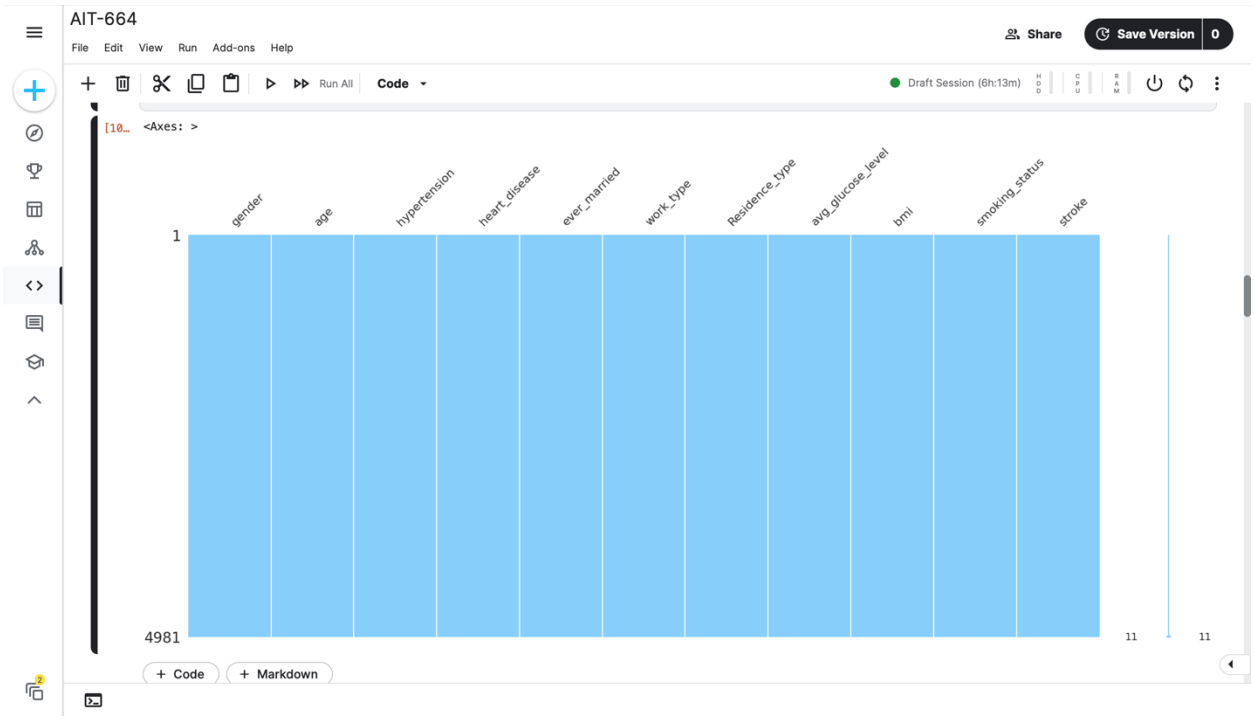
I used Kaggle notebook to explore and run required python code for my analysis. I imported the dataset with all the libraries. We can see that the dataset contains 4981 records and 11 features. The dataset has 11 columns. The above output tells the column name and the datatype of the column and the categorical values. The columns in this dataset are gender, age, hypertension, heart_disease, ever_married, work_type, residence_type, avg_glucose_level, bmi, smoking_status and stroke.

2. Data Preparation/cleaning

I checked for missing data and duplicates, there is no missing or duplicate data in the dataset. There is a categorical data in the smoke column that says UNKNOWN, I had a thought of changing it to never_smoked status but it could be misleading the original data.
I also generated the summary statistics for the numerical data.

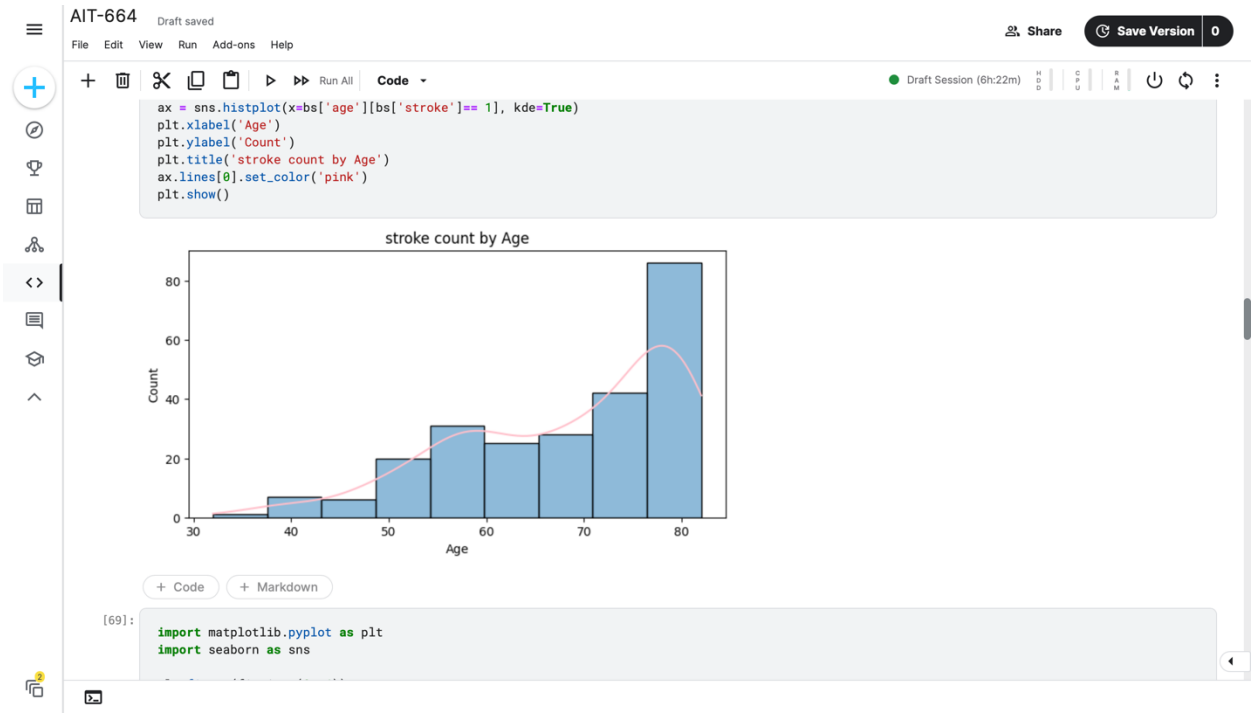


	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	4981.000000	4981.000000	4981.000000	4981.000000	4981.000000	4981.000000
mean	43.419859	0.096165	0.055210	105.943562	28.498173	0.049789
std	22.662755	0.294848	0.228412	45.075373	6.790464	0.217531
min	0.080000	0.000000	0.000000	55.120000	14.000000	0.000000
25%	25.000000	0.000000	0.000000	77.230000	23.700000	0.000000
50%	45.000000	0.000000	0.000000	91.850000	28.100000	0.000000
75%	61.000000	0.000000	0.000000	113.860000	32.600000	0.000000
max	82.000000	1.000000	1.000000	271.740000	48.900000	1.000000



3. Exploratory Analysis

I performed the EDA correlating the features of the dataset with the target variable(stroke) this gave me surprising results.



AIT-664 Draft saved

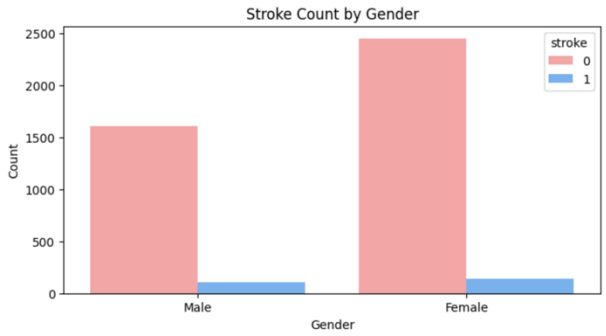
File Edit View Run Add-ons Help

Share Save Version 0

Run All Code Draft Session (6h:36m)

```
plt.ylabel('Count')
plt.title('Stroke Count by Gender')

plt.show()
```



+ Code + Markdown

```
import seaborn as sns
import matplotlib.pyplot as plt
```

AIT-664 Draft saved

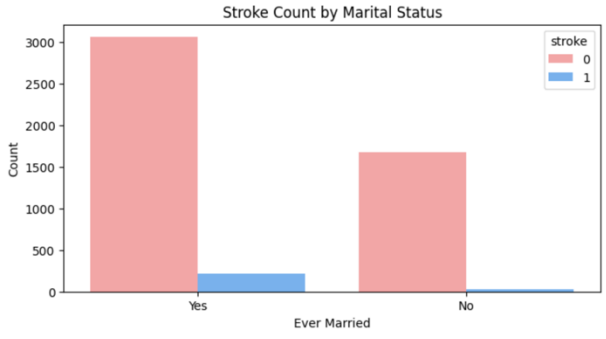
File Edit View Run Add-ons Help

Share Save Version 0

Run All Code Draft Session (6h:36m)

```
plt.xlabel('Ever Married')
plt.ylabel('Count')
plt.title('Stroke Count by Marital Status')

plt.show()
```



+ Code + Markdown

[78]:

```
import seaborn as sns
import matplotlib.pyplot as plt
```

AIT-664

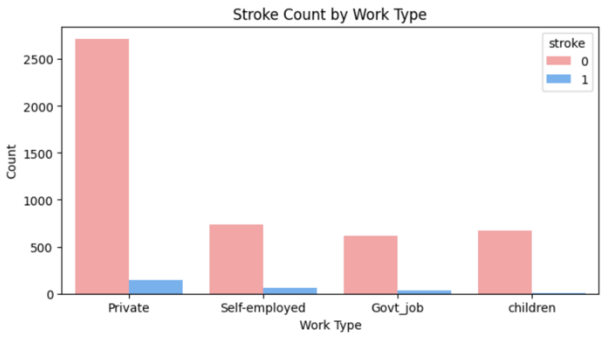
File Edit View Run Add-ons Help

Share Save Version 0

Run All Code Draft Session (6h:36m)

```
plt.xticks(rotation=0)

plt.show()
```




```
custom_palette = ["#FF9999", "#66B2FF"]


plt.figure(figsize=(8, ))

sns.countplot(x='Residence_type', data=df, hue='stroke', palette=custom_palette)
```

AIT-664 Draft saved

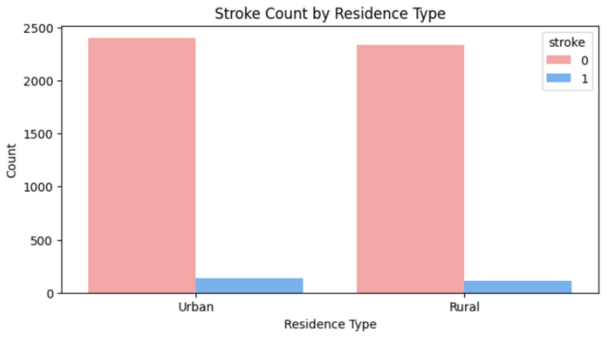
File Edit View Run Add-ons Help

Share Save Version 0

Run All Code Draft Session (6h:36m)

```
plt.ylabel('Count')
plt.title('Stroke Count by Residence Type')

plt.show()
```

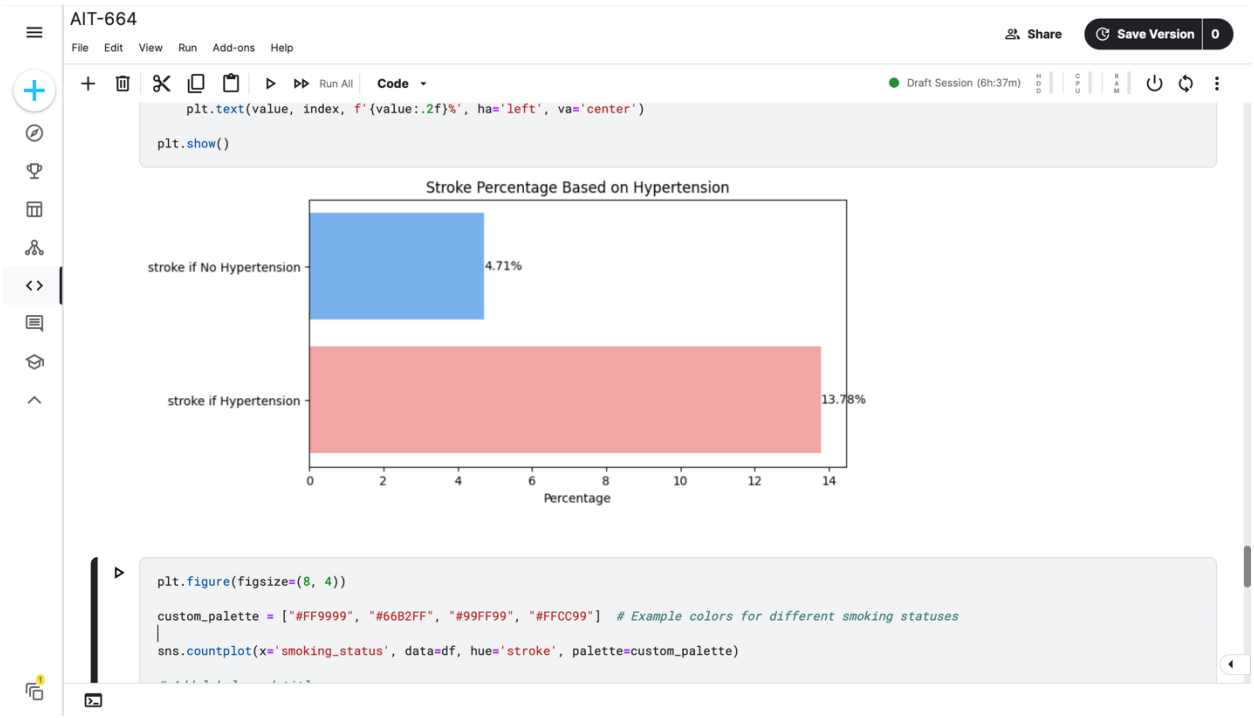


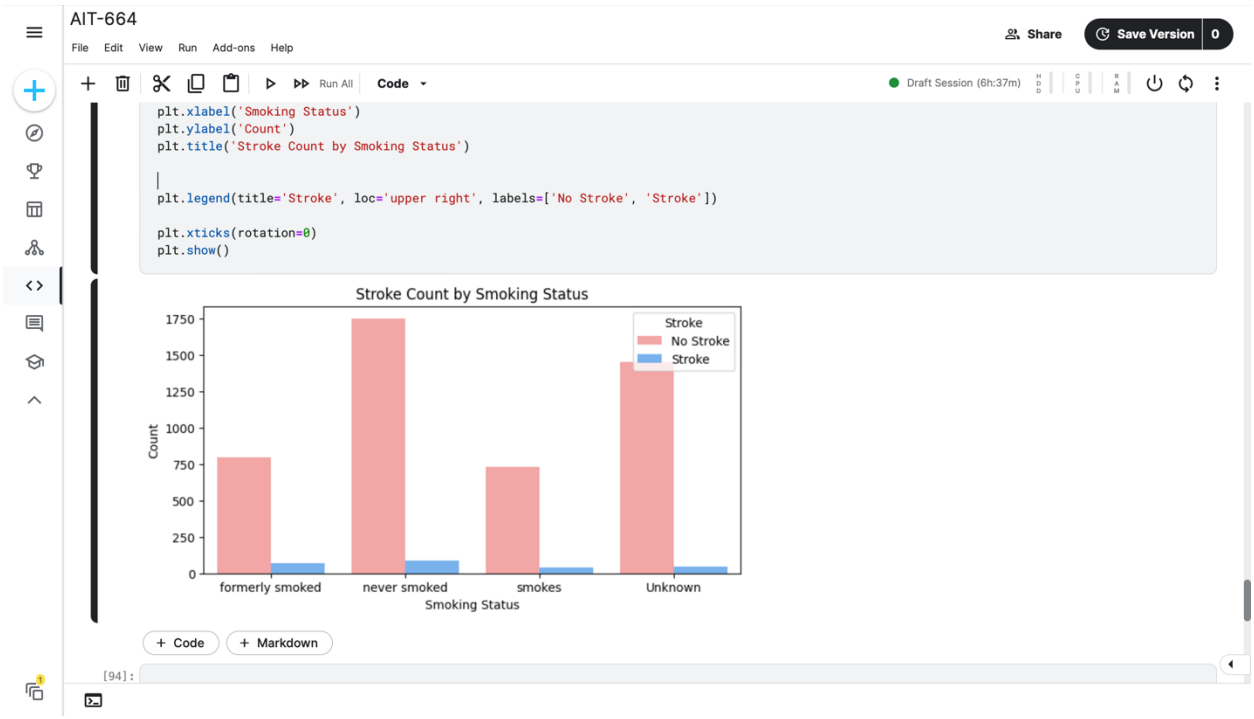
+ Code + Markdown

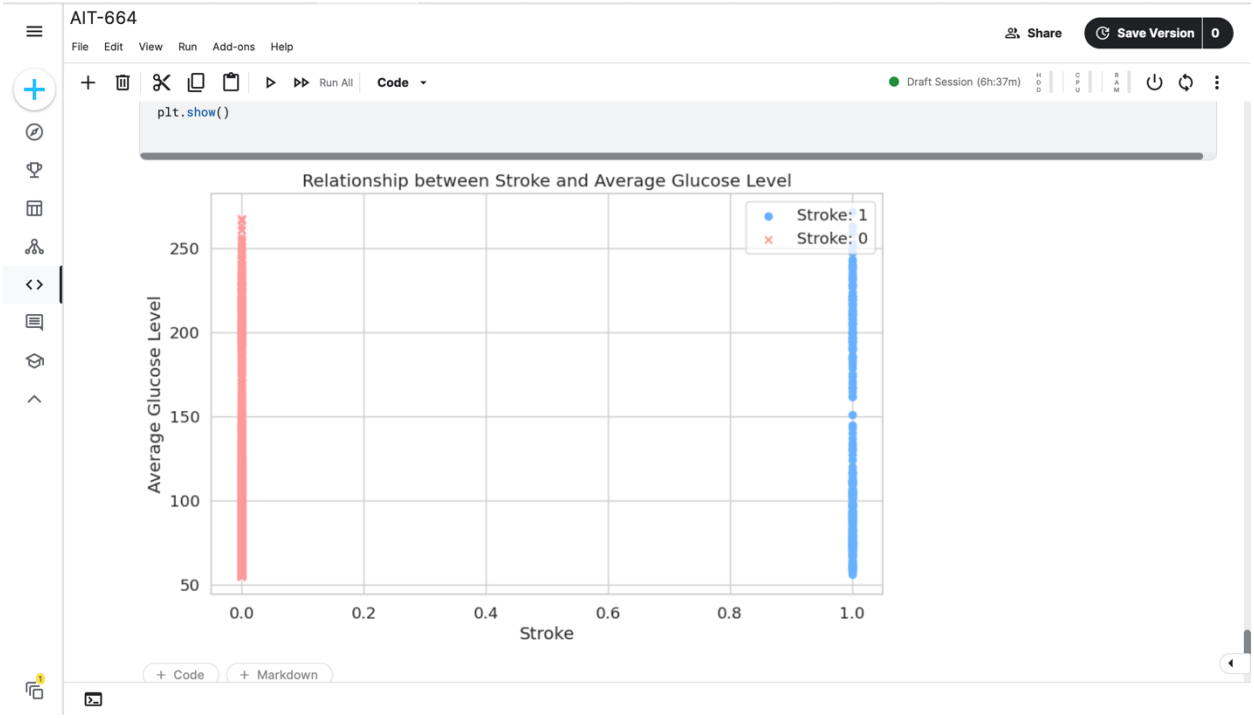
[83]:

```
# Calculate percentages
a = bs['stroke'][bs['heart_disease'] == 0].value_counts()
b = bs['stroke'][bs['heart_disease'] == 1].value_counts()
c, d = 100 * a[1] / (a[0] + a[1]), 100 * b[1] / (b[0] + b[1])
```


AIT-664
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (6h:36m)
plt.text(index, value, f'{value:.2f}%', ha='center', va='bottom')
plt.show()
Stroke Percentage Based on Heart Disease
17.15%
4.94%
Percentage
stroke if No Heart Disease
stroke if Heart Disease
+ Code
+ Markdown
[86]:
a = bs['stroke'][bs['hypertension'] == 0].value_counts()
b = bs['stroke'][bs['hypertension'] == 1].value_counts()
c, d = 100 * a[1] / (a[0] + a[1]), 100 * b[1] / (b[0] + b[1])


AIT-664
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (6h:37m)
plt.text(value, index, f'{value:.2f}%', ha='left', va='center')
plt.show()
Stroke Percentage Based on Hypertension
stroke if No Hypertension
stroke if Hypertension
4.71%
13.78%
Percentage
plt.figure(figsize=(8, 4))
custom_palette = ["#FF9999", "#66B2FF", "#99FF99", "#FFCC99"] # Example colors for different smoking statuses
sns.countplot(x='smoking_status', data=df, hue='stroke', palette=custom_palette)





Observation from outputs:

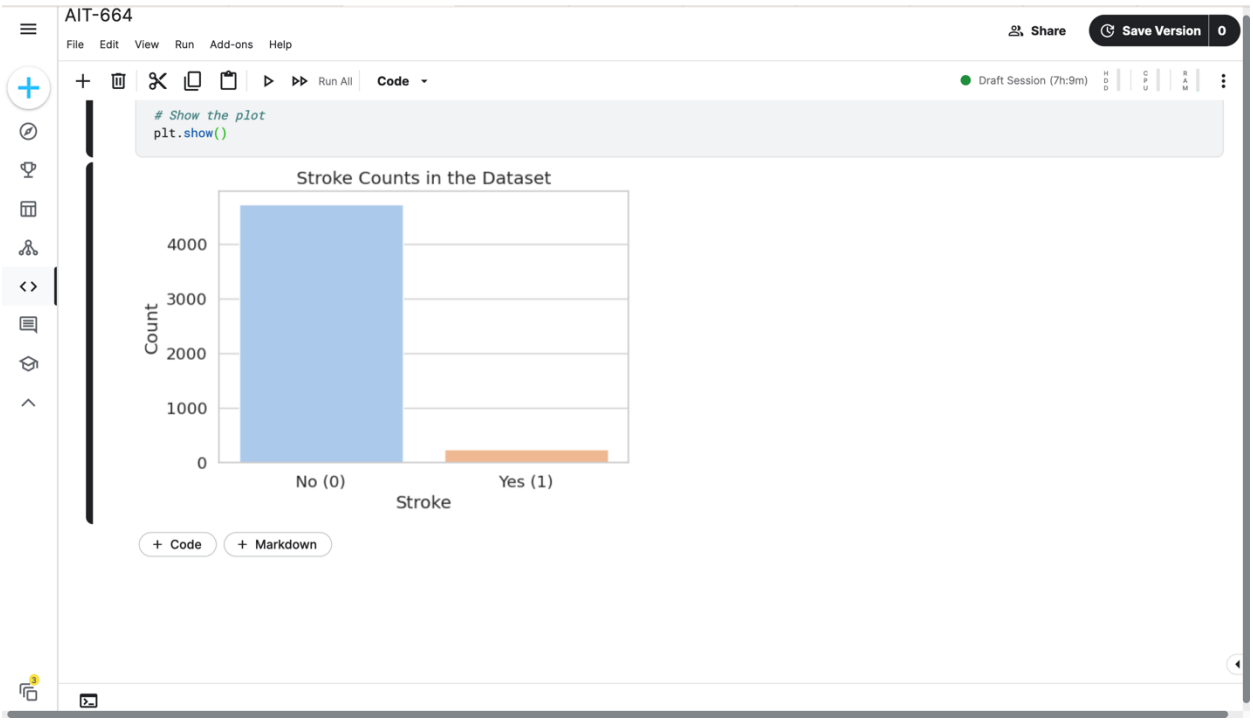
- It is observed that older people are more prone to stroke.
- Females are at risk of getting stroke than male.
- Strokes are also more likely to occur to people who have high glucose level compared to the ones with normal glucose level.

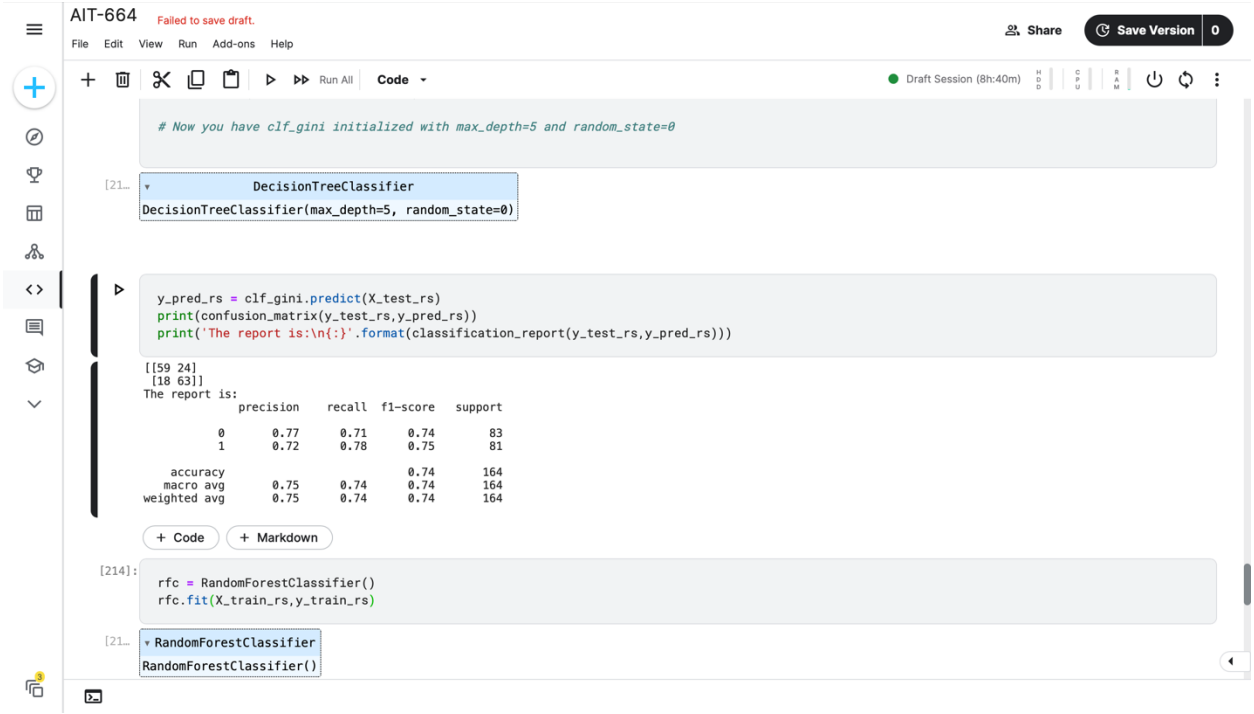
- Marriage also seems to be responsible for increasing the risk for stroke.
- there are also more percentage of people getting a stroke if they have hypertension and are suffering from any kind of heart disease.
- Smokers have higher risk of having a Brain Stroke.
- Urban people have a little more risk of stroke than rural.

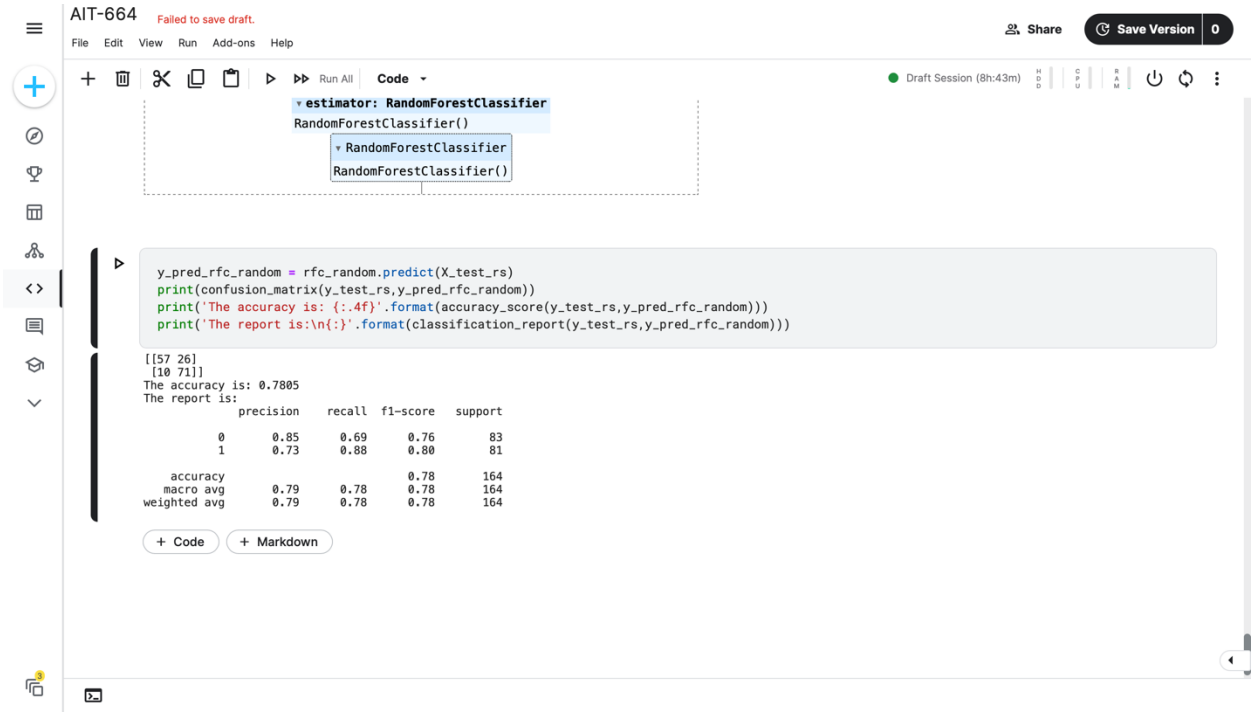
4. Preprocessing/Modeling

In this step of preprocessing and modelling the data I further analyzed the data. I observed that the data is not balanced. Modelling the data that is not balanced would be challenging, it is a good decision to balance the data.
To balance it we can either be done by down sampling or over sampling, here I
Performed down sampling on the majority class by randomly selecting 5% of the data. This down sampling technique is commonly used in machine learning to address class imbalance issues, where one class is significantly underrepresented compared to the other class. By creating a balanced subset, it can improve the performance of algorithms, especially in classification.

I utilized a Random Forest Classifier for my predictions, starting with a basic model and then refining it through hyperparameter tuning using RandomizedSearchCV. The dataset was split into training and testing sets for model evaluation.







Observation:

- Random Under-Sampling: After balancing the dataset, both classes (0 and 1) had 248 instances each.
- Basic Random Forest Model: The initial Random Forest model achieved an accuracy of 74% on the test set, with a precision of 72% for stroke patients and 77% for non-stroke

patients. Recall was 78% for stroke patients and 71% for non-stroke patients. F1-scores were 0.75 for stroke patients and 0.74 for non-stroke patients.

- Tuned Random Forest Model: The hyperparameter-tuned Random Forest model showed significant improvement. It achieved an accuracy of 78%, with a precision of 73% for stroke patients and 85% for non-stroke patients. Recall was 88% for stroke patients and 69% for non-stroke patients. F1-scores were 0.80 for stroke patients and 0.76 for non-stroke patients.

The Random Forest model, especially after hyperparameter tuning, demonstrated promising results in predicting strokes. With an accuracy of 78%, the model showed strong precision and recall values, indicating its ability to correctly identify stroke patients while minimizing misclassifications. Further fine-tuning and exploration of different algorithms could potentially enhance the model's performance, making it a valuable tool in identifying patients at risk of strokes.

Conclusion:

In this data preparation and information modeling report, a comprehensive analysis was conducted on a dataset sourced from Kaggle focusing on brain stroke-related domains. The dataset was meticulously processed, normalized, and cleaned to ensure its quality and reliability for analysis. Notable observations emerged, revealing crucial insights into stroke risk factors. Key findings indicated that older age, female gender, high glucose levels, marriage, hypertension, heart diseases, and smoking are all significant contributors to stroke risk. An initial Random Forest model achieved a respectable accuracy of 74%. However, significant enhancements were achieved through hyperparameter tuning.

Reference:

Akbasli, I. T. (2022, July 16). *Brain stroke prediction dataset*. Kaggle. https://www.kaggle.com/datasets/zzettrkalpakbal/full-filled-brain-stroke-dataset