Prodigy_infotech

Task_1

In [ ]:

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

Importing the Dataset

In [3]:
```
Dataset = pd.read_csv(r"C:\Users\acer\Desktop\Machine Learning PDF\healthcare.
Dataset
```

Out[3]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_typ
0	9046	Male	67.0	0	1	Yes	Private	Urba
1	51676	Female	61.0	0	0	Yes	Self-employed	Rura
2	31112	Male	80.0	0	1	Yes	Private	Rura
3	60182	Female	49.0	0	0	Yes	Private	Urba
4	1665	Female	79.0	1	0	Yes	Self-employed	Rura
...	...	...	...	...	...	...	...	.
5105	18234	Female	80.0	1	0	Yes	Private	Urba
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urba
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rura
5108	37544	Male	51.0	0	0	Yes	Private	Rura
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urba

5110 rows × 12 columns

In [4]:
```
Dataset.head()
```

Out[4]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type
0	9046	Male	67.0	0	1	Yes	Private	Urban
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural
2	31112	Male	80.0	0	1	Yes	Private	Rural
3	60182	Female	49.0	0	0	Yes	Private	Urban
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural

In [5]:
```
Dataset.tail()
```

Out[5]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_typ
5105	18234	Female	80.0	1	0	Yes	Private	Urba
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urba
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rura
5108	37544	Male	51.0	0	0	Yes	Private	Rura
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urba

In [6]:
```
Dataset.describe()
```

Out[6]:

	id	age	hypertension	heart_disease	avg_glucose_level	bmi
count	5110.000000	5110.000000	5110.000000	5110.000000	5110.000000	4909.000000
mean	36517.829354	43.226614	0.097456	0.054012	106.147677	28.893237
std	21161.721625	22.612647	0.296607	0.226063	45.283560	7.854067
min	67.000000	0.080000	0.000000	0.000000	55.120000	10.300000
25%	17741.250000	25.000000	0.000000	0.000000	77.245000	23.500000
50%	36932.000000	45.000000	0.000000	0.000000	91.885000	28.100000
75%	54682.000000	61.000000	0.000000	0.000000	114.090000	33.100000
max	72940.000000	82.000000	1.000000	1.000000	271.740000	97.600000

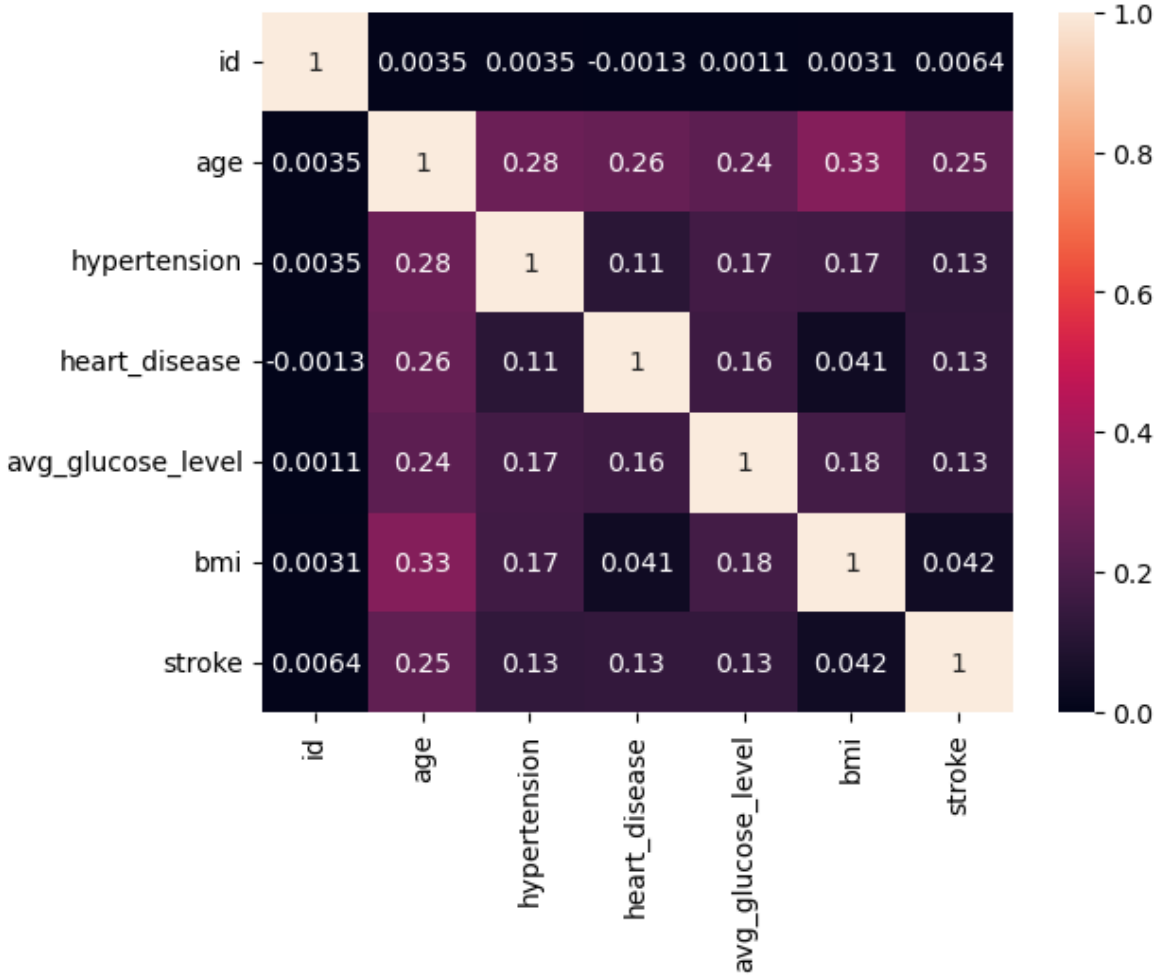
In [7]:
```
Dataset.corr()
```

Out[7]:

	id	age	hypertension	heart_disease	avg_glucose_level	bmi
id	1.000000	0.003538	0.003550	-0.001296	0.001092	0.003084
age	0.003538	1.000000	0.276398	0.263796	0.238171	0.333398
hypertension	0.003550	0.276398	1.000000	0.108306	0.174474	0.167811
heart_disease	-0.001296	0.263796	0.108306	1.000000	0.161857	0.041357
avg_glucose_level	0.001092	0.238171	0.174474	0.161857	1.000000	0.175502
bmi	0.003084	0.333398	0.167811	0.041357	0.175502	1.000000
stroke	0.006388	0.245257	0.127904	0.134914	0.131945	0.042374

In [9]:
```
sns.heatmap(data=Dataset.corr() , annot=True)
```

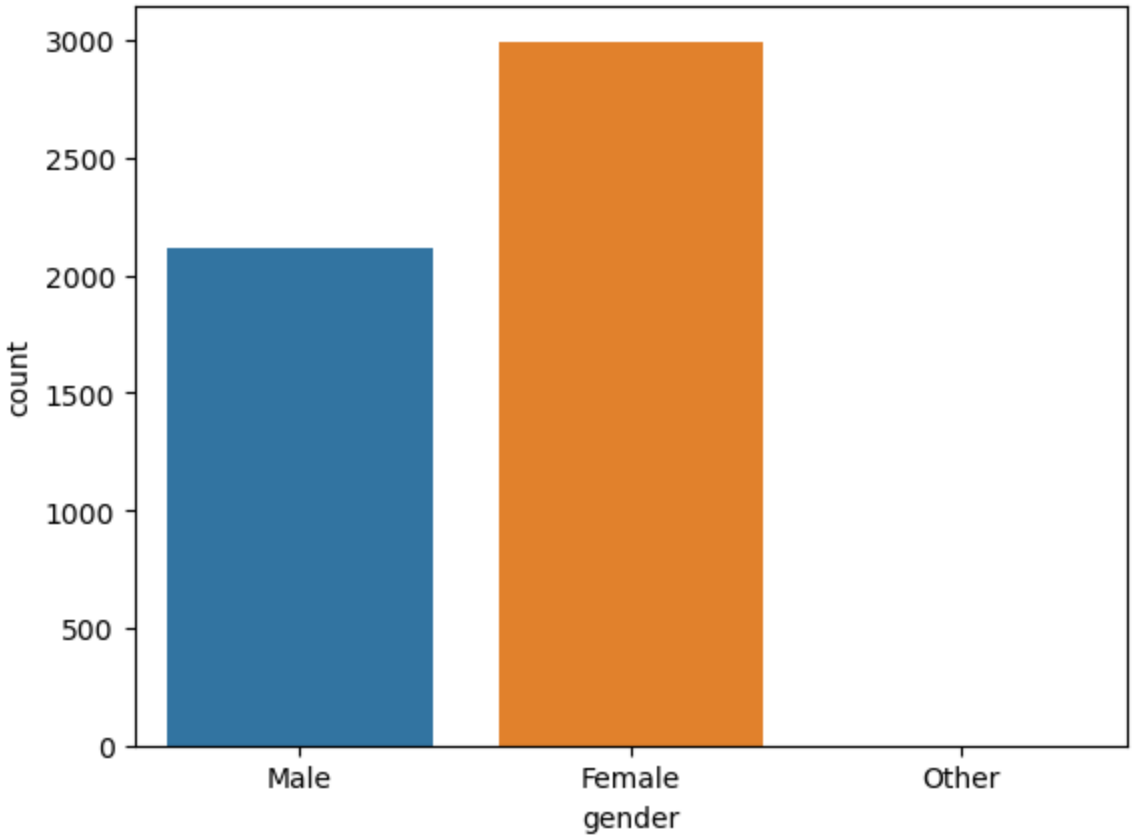
Out[9]: <AxesSubplot:>



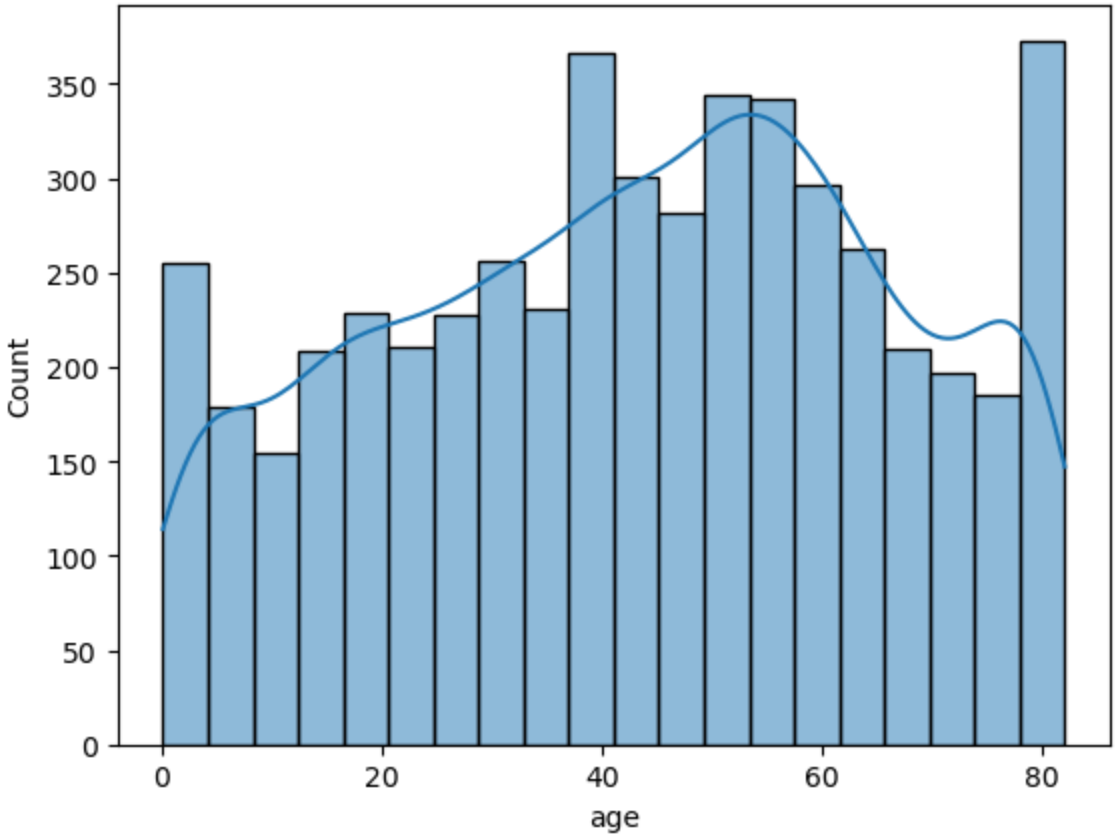
In [13]:
```
Dataset['gender'].value_counts()
```

Out[13]:
```
Female    2994
Male      2115
Other        1
Name: gender, dtype: int64
```

In [16]:
```
sns.countplot(x = 'gender' , data=Dataset)
plt.show()
```

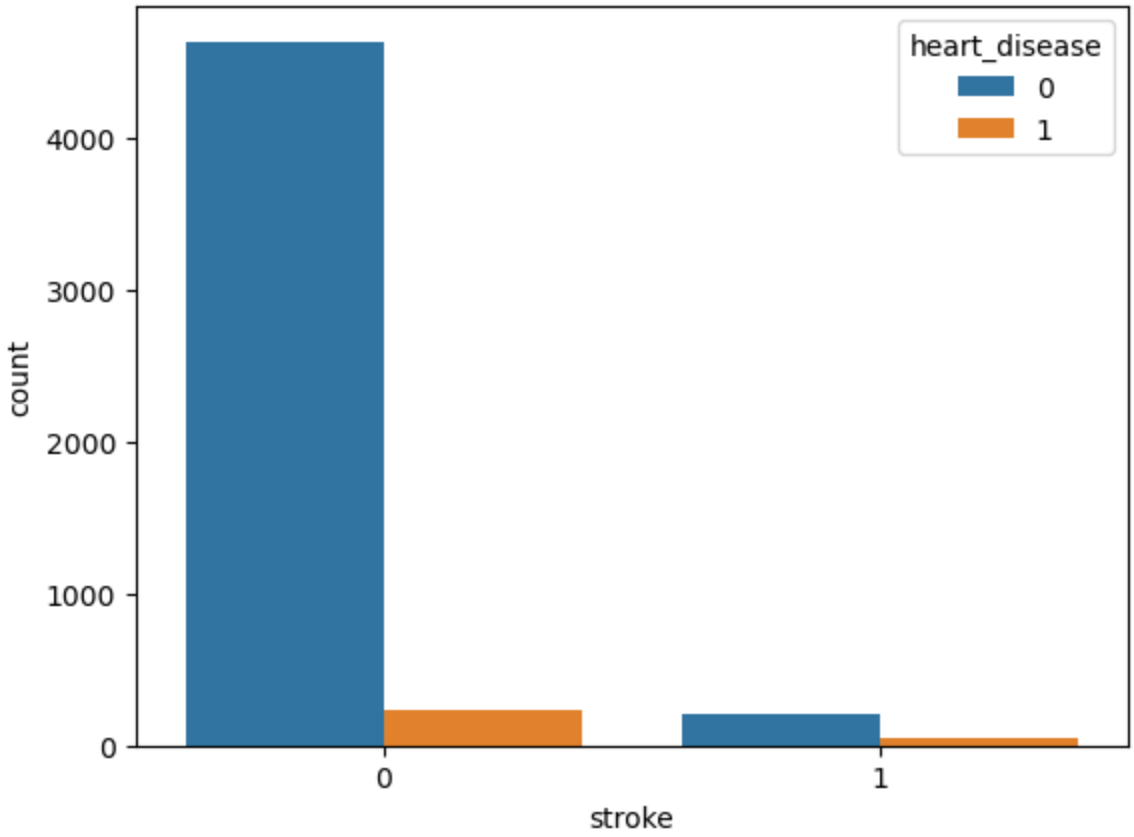


In [26]:
```
sns.histplot(data=Dataset , x = 'age' , kde=True)
plt.show()
```



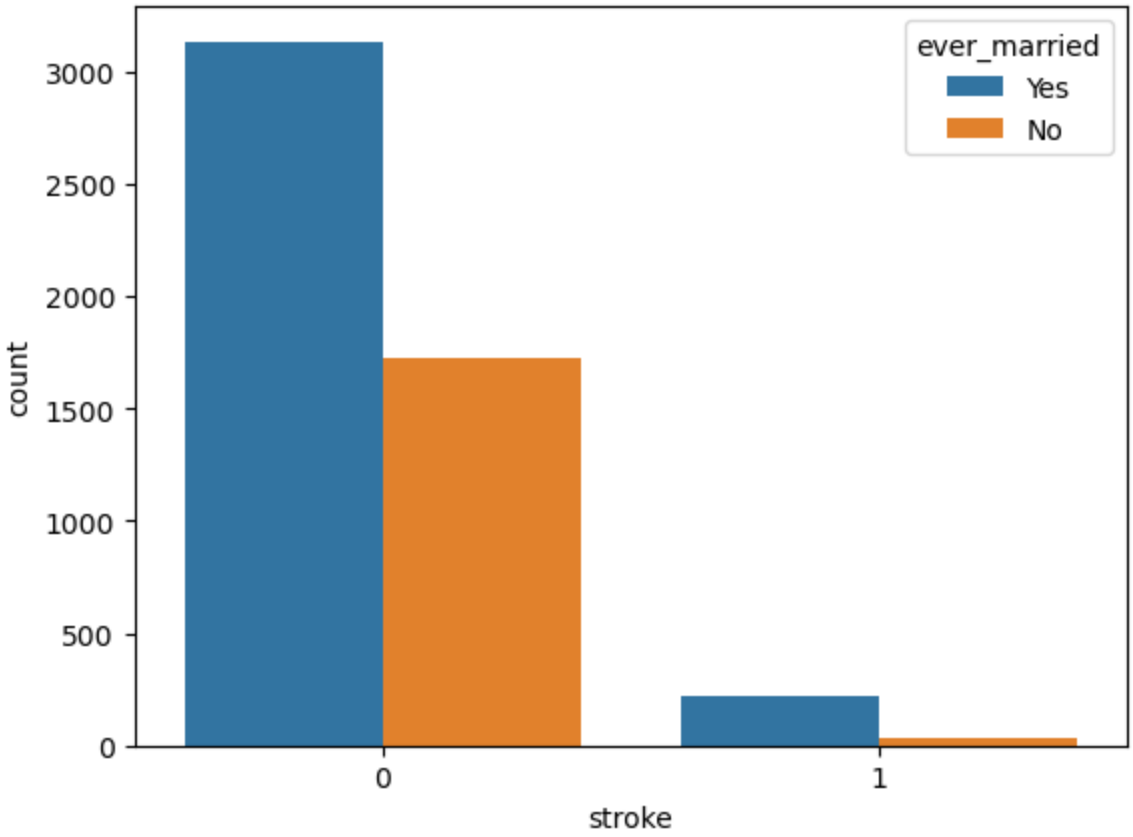
In [28]:
```
sns.countplot(data=Dataset , x = 'stroke' , hue='heart_disease')
```

Out[28]:
```
<AxesSubplot:xlabel='stroke', ylabel='count'>
```



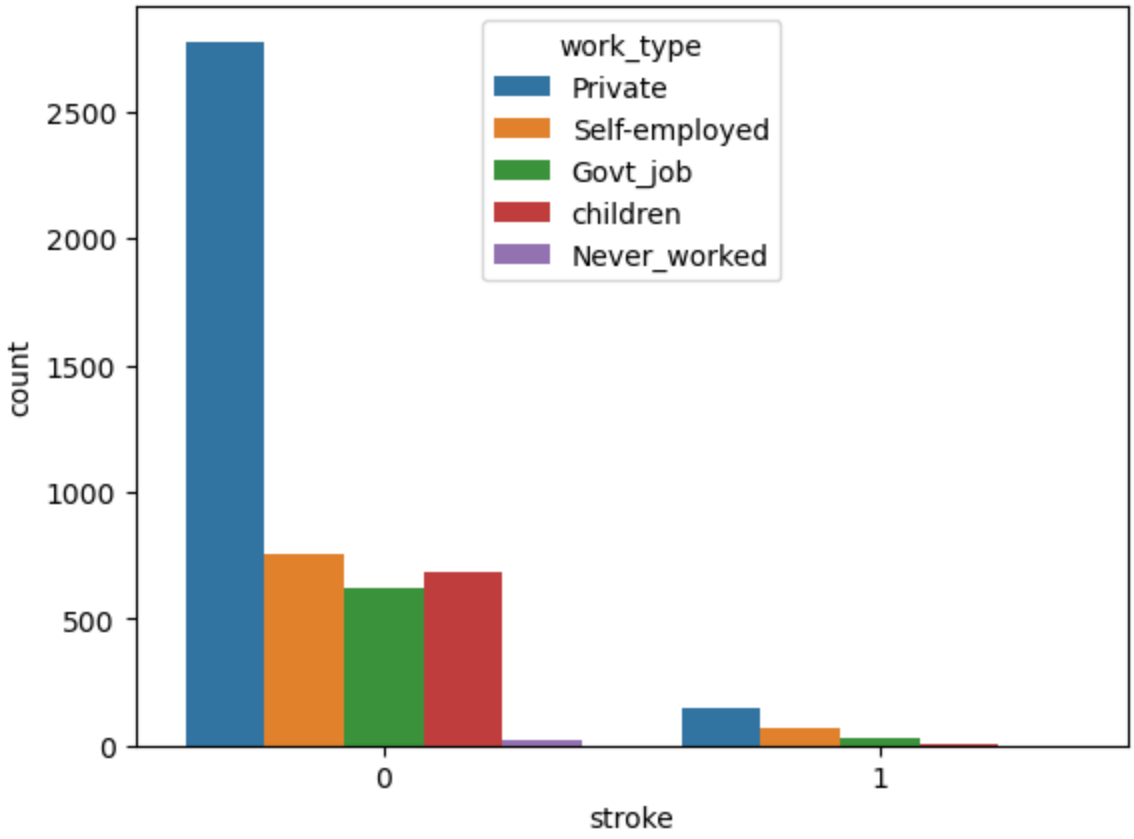
In [29]:
```
sns.countplot(data=Dataset , x = 'stroke' , hue='ever_married')
```

Out[29]:
```
<AxesSubplot:xlabel='stroke', ylabel='count'>
```



In [30]:
```
sns.countplot(data=Dataset , x = 'stroke' , hue='work_type')
```

Out[30]:
```
<AxesSubplot:xlabel='stroke', ylabel='count'>
```



In [ ]:
```

```