

```
import numpy as np
import pandas as pd
df=pd.read_csv('/content/stroke.csv')
df
```

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

5110 rows × 12 columns

Next steps:

Generate code with df

View recommended plots

```
df.columns
```

```
Index(['gender', 'age', 'hypertension', 'heart_disease', 'ever_married',
      'work_type', 'avg_glucose_level', 'bmi', 'smoking_status', 'stroke'],
      dtype='object')
```

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

	gender	age	hypertension	heart_disease	ever_married	work_type	avg_glucose_
0	Male	67.0	0	1	Yes	Private	
1	Female	61.0	0	0	Yes	Self-employed	
2	Male	80.0	0	1	Yes	Private	
3	Female	49.0	0	0	Yes	Private	
4	Female	79.0	1	0	Yes	Self-employed	
...
5105	Female	80.0	1	0	Yes	Private	
5106	Female	81.0	0	0	Yes	Self-employed	
5107	Female	35.0	0	0	Yes	Self-employed	
5108	Male	51.0	0	0	Yes	Private	

Next steps:

Generate code with df

View recommended plots

```
df.head()
```

	gender	age	hypertension	heart_disease	ever_married	work_type	avg_glucose_level
0	Male	67.0	0	1	Yes	Private	228
1	Female	61.0	0	0	Yes	Self-employed	202
2	Male	80.0	0	1	Yes	Private	105
3	Female	49.0	0	0	Yes	Private	171



Next steps:

[Generate code with df](#)[View recommended plots](#)

df.tail()

	gender	age	hypertension	heart_disease	ever_married	work_type	avg_glucose_level
5105	Female	80.0	1	0	Yes	Private	
5106	Female	81.0	0	0	Yes	Self-employed	
5107	Female	35.0	0	0	Yes	Self-employed	
5108	Male	51.0	0	0	Yes	Private	



df.isna().sum()

```
df['bmi']=df['bmi'].fillna(df['bmi'].mean())
df.isna().sum()
```

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

df.dtypes

```
gender          int64
age            float64
hypertension    int64
heart_disease   int64
ever_married    int64
work_type       int64
avg_glucose_level float64
bmi            float64
smoking_status  int64
stroke         int64
dtype: object
```

```
from sklearn.preprocessing import LabelEncoder
encode=LabelEncoder()
df['gender']=encode.fit_transform(df['gender'])
df['ever_married']=encode.fit_transform(df['ever_married'])
df['work_type']=encode.fit_transform(df['work_type'])
df['smoking_status']=encode.fit_transform(df['smoking_status'])
df.dtypes
```

```
gender          int64
age            float64
hypertension    int64
heart_disease   int64
ever_married    int64
work_type       int64
avg_glucose_level float64
bmi            float64
smoking_status  int64
stroke         int64
dtype: object
```

```
x=df.iloc[:, :-1].values
x
```

```

array([[ 1.      , 67.      , 0.      , ..., 228.69   ,
        36.6     , 1.      , ],
       [ 0.      , 61.      , 0.      , ..., 202.21   ,
        28.89323691, 2.      , ],
       [ 1.      , 80.      , 0.      , ..., 105.92   ,
        32.5     , 2.      , ],
       ...,
       [ 0.      , 35.      , 0.      , ..., 82.99    ,
        30.6     , 2.      , ],
       [ 1.      , 51.      , 0.      , ..., 166.29   ,
        25.6     , 1.      , ],
       [ 0.      , 44.      , 0.      , ..., 85.28    ,
        26.2     , 0.      , ]])

y=df.iloc[:, -1].values
y

array([1, 1, 1, ..., 0, 0, 0])

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=1)
x_train

from sklearn.preprocessing import StandardScaler
scale=StandardScaler()
scale.fit(x_train)
x_train=scale.transform(x_train)
x_test=scale.transform(x_test)
x_train

from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import BernoulliNB
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
knn=KNeighborsClassifier(n_neighbors=7)
base=BernoulliNB()
model=SVC()
lst=[knn,base,model]

for i in lst:
    i.fit(x_train,y_train)
    y_pred=i.predict(x_test)
    print("\n")
    print("Model is", i)
    print('*'*100)
    print(confusion_matrix(y_test,y_pred))
    print("Score is",accuracy_score(y_test,y_pred))
    print(classification_report(y_test,y_pred))

```

```

Model is KNeighborsClassifier(n_neighbors=7)
*****
[[1449   1]
 [ 82   1]]
Score is 0.9458577951728636

```

	precision	recall	f1-score	support
0	0.95	1.00	0.97	1450
1	0.50	0.01	0.02	83
accuracy			0.95	1533
macro avg	0.72	0.51	0.50	1533
weighted avg	0.92	0.95	0.92	1533

```

Model is BernoulliNB()
*****
[[1441   9]
 [ 80   3]]
Score is 0.9419439008480104

```

	precision	recall	f1-score	support
0	0.95	0.99	0.97	1450
1	0.25	0.04	0.06	83
accuracy			0.94	1533
macro avg	0.60	0.51	0.52	1533
weighted avg	0.91	0.94	0.92	1533

```

Model is SVC()
*****
[[1450    0]
 [  83    0]]
Score is 0.9458577951728636

```

	precision	recall	f1-score	support
0	0.95	1.00	0.97	1450
1	0.00	0.00	0.00	83
accuracy			0.95	1533
macro avg	0.47	0.50	0.49	1533
weighted avg	0.89	0.95	0.92	1533

```

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are i
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are i
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are i
_warn_prf(average, modifier, msg_start, len(result))

```