```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from sklearn.preprocessing import StandardScaler
5 from sklearn.model_selection import train_test_split
6 from sklearn.linear_model import LogisticRegression
7 from sklearn.metrics import confusion_matrix
8 from sklearn.metrics import accuracy_score
```

```
1 df = pd.read_csv('/stroke-data.csv')
```

```
1 df.head()
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes

Next steps: View recommended plots

```
1 df.dropna('id', axis=0,inplace=true)
```

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

```
1 df.head()
```

	gender	age	hypertension	heart_disease	ever_married	work_type	avg_glucose_level	bmi	smoking_status	stroke
0	Male	67.0	0	1	Yes	Private	228.69	36.6	formerly smoked	1
2	Male	80.0	0	1	Yes	Private	105.92	32.5	never smoked	1
3	Female	49.0	0	0	Yes	Private	171.23	34.4	smokes	1
4	Female	79.0	1	0	Yes	Self-employed	174.12	24.0	never smoked	1
5	Male	81.0	0	0	Yes	Private	186.21	29.0	formerly smoked	1

Next steps: View recommended plots

```
1 #one hot encoding.
2 gender = pd.get_dummies(df['gender'], dtype=int)
3 gender.drop(['Male'], axis=1, inplace=True)
4 df.drop('gender', axis=1, inplace=True)
5 df = pd.concat([gender, df], axis=1)
6 df.head()
```

	Female	Other	age	hypertension	heart_disease	ever_married	work_type	avg_glu
0	0	0	67.0	0	1	Yes	Private	
2	0	0	80.0	0	1	Yes	Private	
3	1	0	49.0	0	0	Yes	Private	
4	1	0	79.0	1	0	Yes	Self-employed	

Next steps: View recommended plots

```
1 df[['ever_married', 'work_type', 'smoking_status']] = df[['ever_married', 'work_type', 'smoking_status']].astype("c
2
```

```
def oneHot(name, X, output_features, dropFirst):
  dataframe = df[name]
  encoded = pd.get_dummies(dataframe, drop_first=dropFirst,dtype=int)
  print(encoded.head())
  encoded.columns = output_features
  X.drop(name, axis=1, inplace=True)
  X = pd.concat([encoded, X], axis=1)
  return X
```

```
featureCol = df['ever_married'].cat.categories
print(featureCol)
```

```
Index(['No', 'Yes'], dtype='object')
```

```
df = oneHot('ever_married', df, ['Married'], True)
df.head()
```

```
   Yes
0    1
2    1
3    1
4    1
5    1
```

	Married	Female	Other	age	hypertension	heart_disease	work_type	avg_glucose_
0	1	0	0	67.0	0	1	Private	2
2	1	0	0	80.0	0	1	Private	1
3	1	1	0	49.0	0	0	Private	1
4	1	1	0	79.0	1	0	Self-employed	1

Next steps: View recommended plots

```
featureCol = df['work_type'].cat.categories[1:]
df = oneHot('work_type', df, featureCol, True)
```

```
   Never_worked  Private  Self-employed  children
0             0        1              0         0
2             0        1              0         0
3             0        1              0         0
4             0        0              1         0
5             0        1              0         0
```

```
df.head()
```

	Never_worked	Private	Self-employed	children	Married	Female	Other	age	hypertens
0	0	1	0	0	1	0	0	67.0	
2	0	1	0	0	1	0	0	80.0	
3	0	1	0	0	1	1	0	49.0	
4	0	0	1	0	1	1	0	79.0	

Next steps: View recommended plots

```
featureCol = df['smoking_status'].cat.categories
print(featureCol)
df = oneHot('smoking_status', df, featureCol, False)
```

```
Index(['Unknown', 'formerly smoked', 'never smoked', 'smokes'], dtype='object')
   Unknown  formerly smoked  never smoked  smokes
0        0                1             0       0
2        0                0             1       0
3        0                0             0       1
4        0                0             1       0
5        0                1             0       0
```

```
df.head()
```

	Unknown	formerly smoked	never smoked	smokes	Never_worked	Private	Self-employed	children	Marr
0	0	1	0	0	0	1	0	0	
2	0	0	1	0	0	1	0	0	
3	0	0	0	1	0	1	0	0	
4	0	0	1	0	0	0	1	0	

Next steps: View recommended plots

```
df = df[df['Unknown'] == 0]
```

```
df.head()
```

	Unknown	formerly smoked	never smoked	smokes	Never_worked	Private	Self-employed	children	Marr
0	0	1	0	0	0	1	0	0	
2	0	0	1	0	0	1	0	0	
3	0	0	0	1	0	1	0	0	
4	0	0	1	0	0	0	1	0	

Next steps: View recommended plots

```
df.drop(['Unknown', 'formerly smoked'], axis=1, inplace = True)
```

```
df.head()
```

	never smoked	smokes	Never_worked	Private	Self-employed	children	Married	Female	Other
0	0	0	0	1	0	0	1	0	0
2	1	0	0	1	0	0	1	0	0
3	0	1	0	1	0	0	1	1	0
4	1	0	0	0	1	0	1	1	0

Next steps: View recommended plots

```
df['age'].describe()
df['age'] /= 100
```

```
standard = StandardScaler()
standardized_features = standard.fit_transform(df.iloc[:, [12,13]])
df.iloc[:, 12:14] = standardized_features
df.head()
```

	never smoked	smokes	Never_worked	Private	Self-employed	children	Married	Female	Other
0	0	0	0	1	0	0	1	0	0
2	1	0	0	1	0	0	1	0	0
3	0	1	0	1	0	0	1	1	0
4	1	0	0	0	1	0	1	1	0

Next steps: View recommended plots

```
X = df.drop('stroke', axis=1)
y = df['stroke']
```

```
X
```

	never smoked	smokes	Never_worked	Private	Self-employed	children	Married	Female	Oth
0	0	0	0	1	0	0	1	0	
2	1	0	0	1	0	0	1	0	
3	0	1	0	1	0	0	1	1	
4	1	0	0	0	1	0	1	1	
5	0	0	0	1	0	0	1	0	
...	...	...	...	...	...	...	...	...	
5100	1	0	0	0	1	0	1	0	
5102	1	0	0	1	0	0	1	1	
5106	1	0	0	0	1	0	1	1	
5107	1	0	0	0	1	0	1	1	
5108	0	0	0	1	0	0	1	0	

Nex View recommended plots

```
1 y
```

```
0       1
2       1
3       1
4       1
5       1
       ..
5100    0
5102    0
5106    0
5107    0
5108    0
Name: stroke, Length: 3426, dtype: int64
```

```
1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
1 model = LogisticRegression()
```

```
1 model.fit(X_train, y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
1 predictions = model.predict(X_test)
```

```
1 cnf_matrix = confusion_matrix(y_test, predictions)
2 cnf_matrix
```

```
array([[969,   0],
       [ 59,   0]])
```

```
1 score = accuracy_score(y_test, predictions)
2 print(score)
```

```
0.9426070038910506
```