```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matr
import seaborn as sns
import matplotlib.pyplot as plt


data = pd.read_csv('/content/diabetes_prediction_dataset.csv')
print(data.head())
print(data.info())
print(data.describe())
```

```
   gender   age  hypertension  heart_disease smoking_history    bmi  \
0  Female  80.0             0              1           never  25.19
1  Female  54.0             0              0         No Info  27.32
2    Male  28.0             0              0           never  27.32
3  Female  36.0             0              0         current  23.45
4    Male  76.0             1              1         current  20.14

   HbA1c_level  blood_glucose_level  diabetes
0          6.6                  140         0
1          6.6                   80         0
2          5.7                  158         0
3          5.0                  155         0
4          4.8                  155         0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 9 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   gender               100000 non-null  object
 1   age                  100000 non-null  float64
 2   hypertension         100000 non-null  int64
 3   heart_disease        100000 non-null  int64
 4   smoking_history      100000 non-null  object
 5   bmi                  100000 non-null  float64
 6   HbA1c_level          100000 non-null  float64
 7   blood_glucose_level  100000 non-null  int64
 8   diabetes             100000 non-null  int64
dtypes: float64(3), int64(4), object(2)
memory usage: 6.9+ MB
None
                 age  hypertension  heart_disease            bmi  \
count  100000.000000  100000.00000  100000.000000  100000.000000
mean       41.885856       0.07485       0.039420      27.320767
std        22.516840       0.26315       0.194593       6.636783
min         0.080000       0.00000       0.000000      10.010000
25%        24.000000       0.00000       0.000000      23.630000
50%        43.000000       0.00000       0.000000      27.320000
75%        60.000000       0.00000       0.000000      29.580000
max        80.000000       1.00000       1.000000      95.690000

         HbA1c_level  blood_glucose_level       diabetes
count  100000.000000        100000.000000  100000.000000
mean        5.527507           138.058060       0.085000
std         1.070672            40.708136       0.278883
min         3.500000            80.000000       0.000000
25%         4.800000           100.000000       0.000000
50%         5.800000           140.000000       0.000000
75%         6.200000           159.000000       0.000000
max         9.000000           300.000000       1.000000
```

```
data['blood_glucose_level_category'] = pd.cut(data['blood_glucose_level'], bins=3,

scaler = StandardScaler()
data[['age', 'bmi', 'blood_glucose_level']] = scaler.fit_transform(data[['age', 'bm

X = data[['age', 'bmi', 'hypertension']]
y = data['blood_glucose_level_category']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_sta

model = LogisticRegression()
model.fit(X_train, y_train)

cv_scores = cross_val_score(model, X_train, y_train, cv=5)
print(f'Cross-validation scores: {cv_scores}')
print(f'Mean cross-validation score: {np.mean(cv_scores)}')

y_pred = model.predict(X_test)
print(f'Accuracy: {accuracy_score(y_test, y_pred)}')
print('Classification Report:')
print(classification_report(y_test, y_pred))
print('Confusion Matrix:')
print(confusion_matrix(y_test, y_pred))

coefficients = pd.DataFrame(model.coef_[0], X.columns, columns=['Coefficient'])
print(coefficients)

sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

```
Cross-validation scores: [0.58978571 0.5895     0.58964286 0.58985714 0.589857
Mean cross-validation score: 0.5897285714285714
Accuracy: 0.5915666666666667
Classification Report:
              precision    recall  f1-score   support

           0       0.59      1.00      0.74     17754
           1       0.00      0.00      0.00     11470
           2       0.05      0.00      0.00       776

    accuracy                           0.59     30000
   macro avg       0.21      0.33      0.25     30000
weighted avg       0.35      0.59      0.44     30000
```

```
weighted avg       0.53      0.59      0.41     30000

Confusion Matrix:
[[17746     0     8]
 [11459     0    11]
 [  775     0     1]]
             Coefficient
age            -0.354876
bmi            -0.183636
hypertension   -0.259470
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:134
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:134
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:134
  _warn_prf(average, modifier, msg_start, len(result))
```