```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import StandardScaler


data = pd.read_csv('/content/diabetes.csv')
```

```
data.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Next steps: Generate code with `data` View recommended plots

```
data.shape
```

```
(768, 9)
```

```
data.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000

```
data.groupby('Outcome').mean()
```

Outcome	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
0	3.298000	109.980000	68.184000	19.664000	68.792000	30.304200
1	4.865672	141.257463	70.824627	22.164179	100.335821	35.142537

```
X = data.drop('Age', axis=1)
y = data['Age']
print(X)
print(y)
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0              6      148             72             35        0  33.6
1              1       85             66             29        0  26.6
2              8      183             64              0        0  23.3
3              1       89             66             23       94  28.1
4              0      137             40             35      168  43.1
..           ...      ...            ...            ...      ...   ...
763           10      101             76             48      180  32.9
764            2      122             70             27        0  36.8
765            5      121             72             23      112  26.2
766            1      126             60              0        0  30.1
767            1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Outcome
0                       0.627        1
1                       0.351        0
2                       0.672        1
```

```
3                       0.167        0
4                       2.288        1
..                        ...      ...
763                     0.171        0
764                     0.340        0
765                     0.245        0
766                     0.349        1
767                     0.315        0

[768 rows x 8 columns]
0      50
1      31
2      32
3      21
4      33
       ..
763    63
764    27
765    30
766    47
767    23
Name: Age, Length: 768, dtype: int64
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```
▾ RandomForestClassifier
RandomForestClassifier(random_state=42)
```

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

```
accuracy = accuracy_score(y_test, predictions)
report = classification_report(y_test, predictions)
```

```
e are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

re ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.

e are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

re ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.

e are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

re ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
```

```
print("Accuracy:", accuracy)
print("Classification Report:\n", report)
```

```
Accuracy: 0.03896103896103896
Classification Report:
              precision    recall  f1-score   support

          21       0.10      0.17      0.12        12
          22       0.00      0.00      0.00        13
          23       0.00      0.00      0.00         7
          24       0.00      0.00      0.00        10
          25       0.12      0.25      0.16         8
          26       0.14      0.33      0.20         3
          27       0.00      0.00      0.00         5
          28       0.00      0.00      0.00         9
          29       0.00      0.00      0.00        10
          30       0.00      0.00      0.00         4
          31       0.00      0.00      0.00         3
          32       0.00      0.00      0.00         4
          33       0.00      0.00      0.00         2
          34       0.00      0.00      0.00         2
          35       0.00      0.00      0.00         0
          36       0.00      0.00      0.00         4
          37       0.00      0.00      0.00         3
          38       0.00      0.00      0.00         6
          39       0.00      0.00      0.00         4
          40       0.00      0.00      0.00         2
          41       0.20      0.33      0.25         3
```

42	0.00	0.00	0.00	4
43	0.00	0.00	0.00	4
44	0.00	0.00	0.00	3
45	0.00	0.00	0.00	2
46	0.00	0.00	0.00	0
47	0.00	0.00	0.00	0
48	0.00	0.00	0.00	1
49	0.00	0.00	0.00	1
50	0.00	0.00	0.00	2
51	0.00	0.00	0.00	1
52	0.00	0.00	0.00	0
53	0.00	0.00	0.00	2
54	0.00	0.00	0.00	2
55	0.00	0.00	0.00	1
56	0.00	0.00	0.00	1
57	0.00	0.00	0.00	1
58	0.00	0.00	0.00	4
60	0.00	0.00	0.00	3
62	0.00	0.00	0.00	3
63	0.00	0.00	0.00	2
65	0.00	0.00	0.00	2
67	0.00	0.00	0.00	1
69	0.00	0.00	0.00	0
accuracy			0.04	154
macro avg	0.01	0.02	0.02	154
weighted avg	0.02	0.04	0.03	154