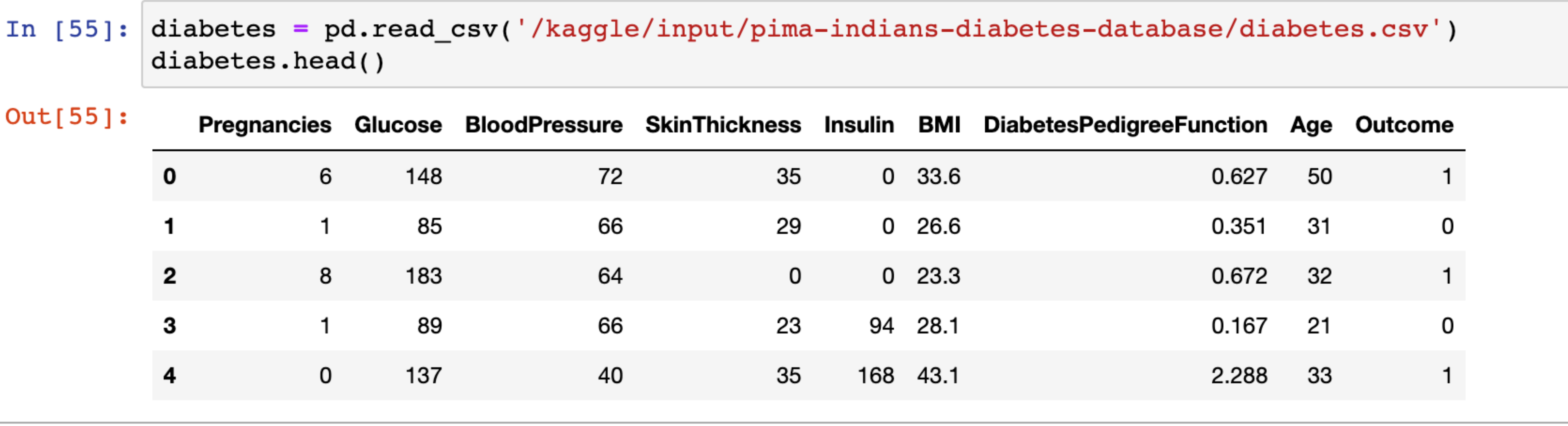
1. Introduction

DTSA 5510 - Unsupervised Algorithms in Machine Learning Final Project

Feb, 22, 2023

1. Exploratory Data Analysis

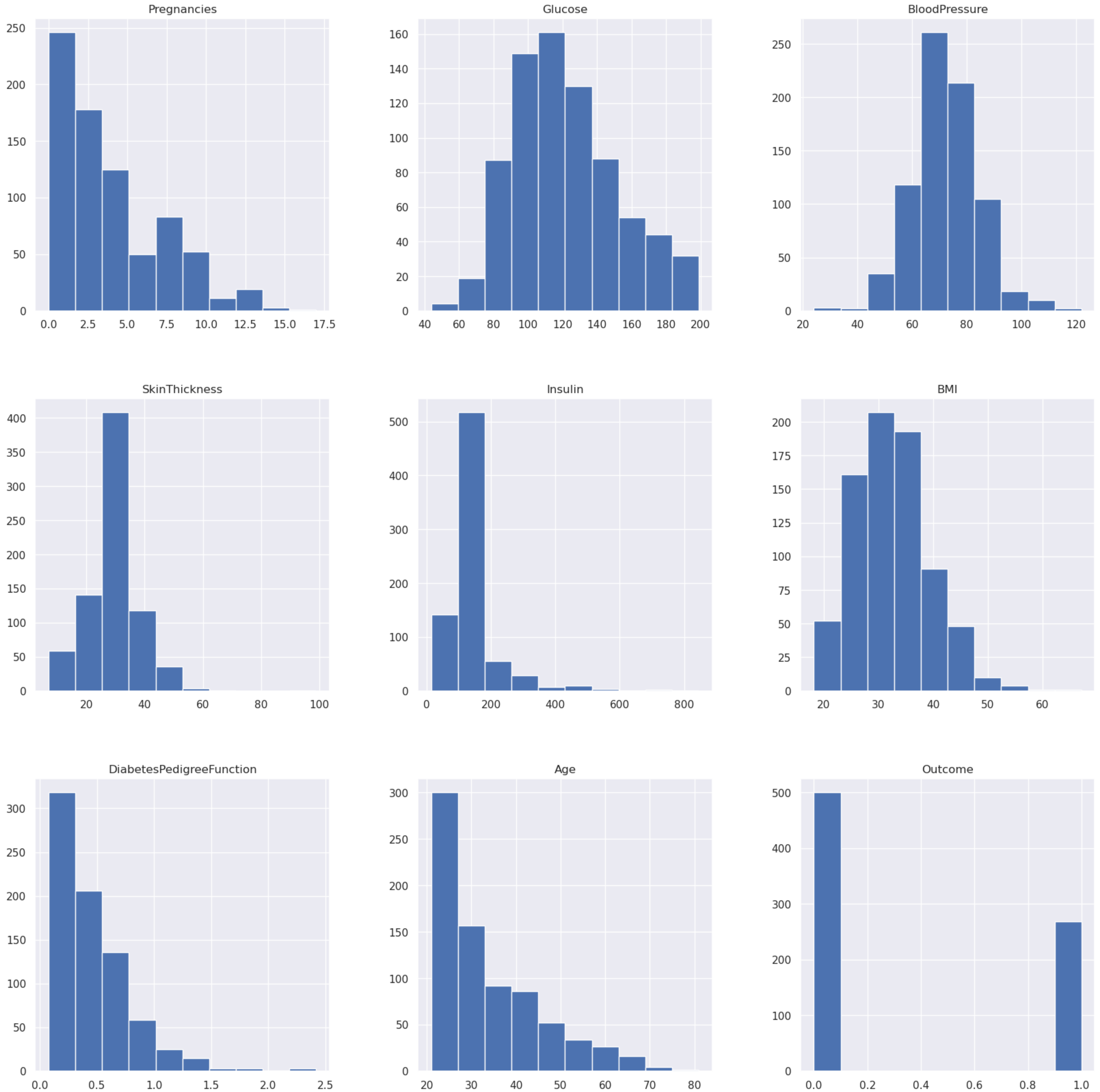


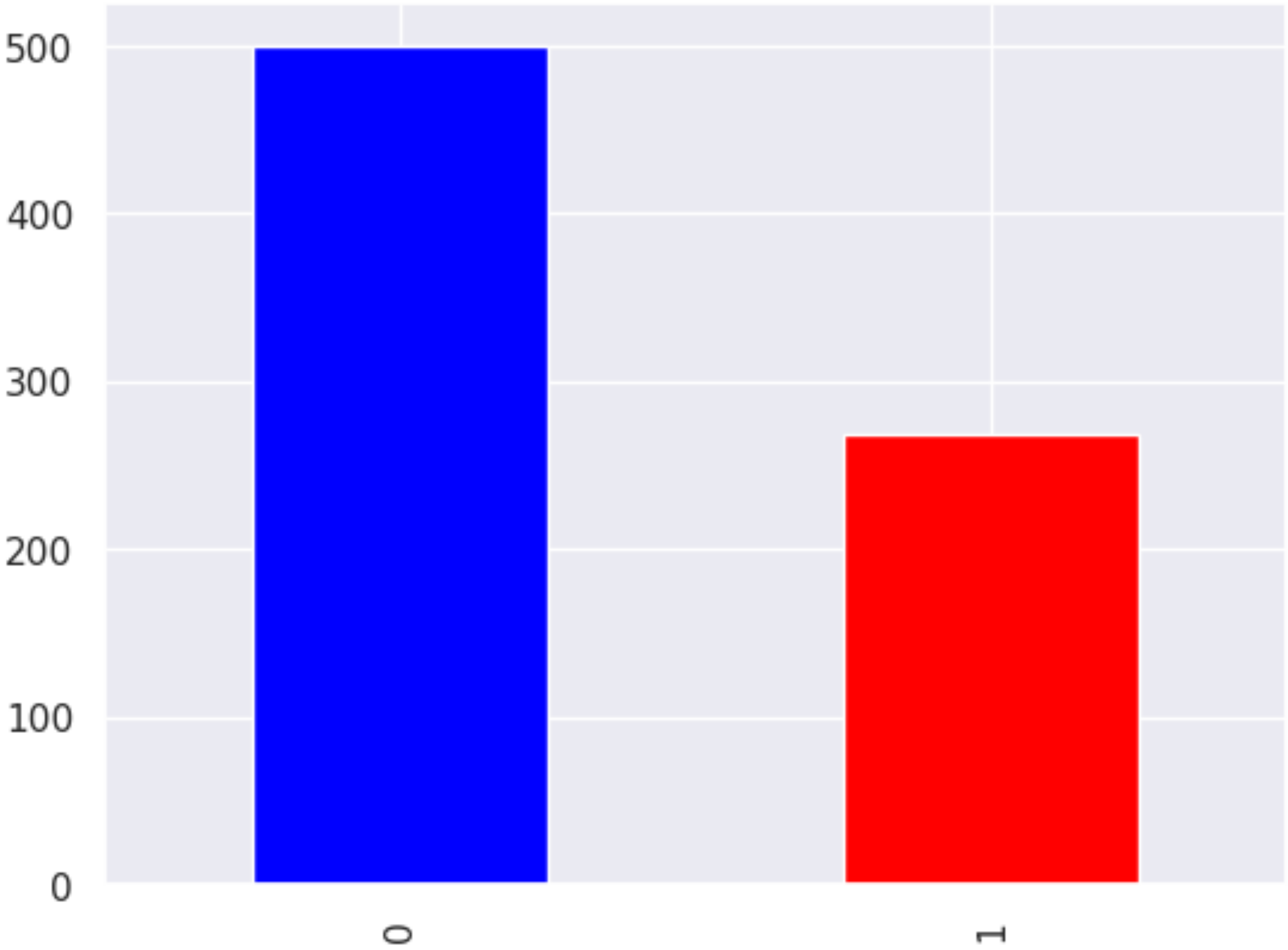
In [55]:

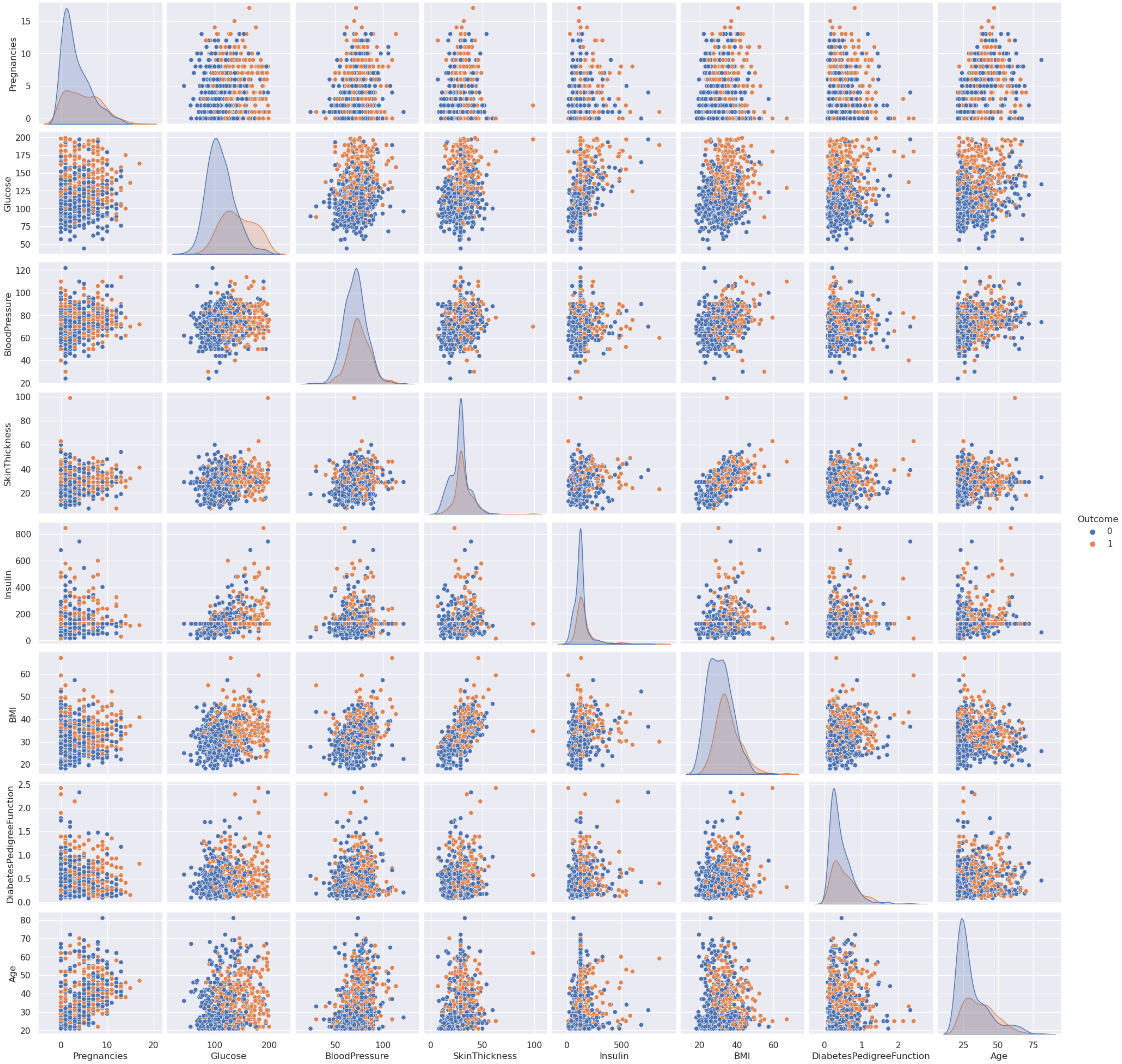
```
diabetes = pd.read_csv('/kaggle/input/pima-indians-diabetes-database/diabetes.csv')
diabetes.head()
```

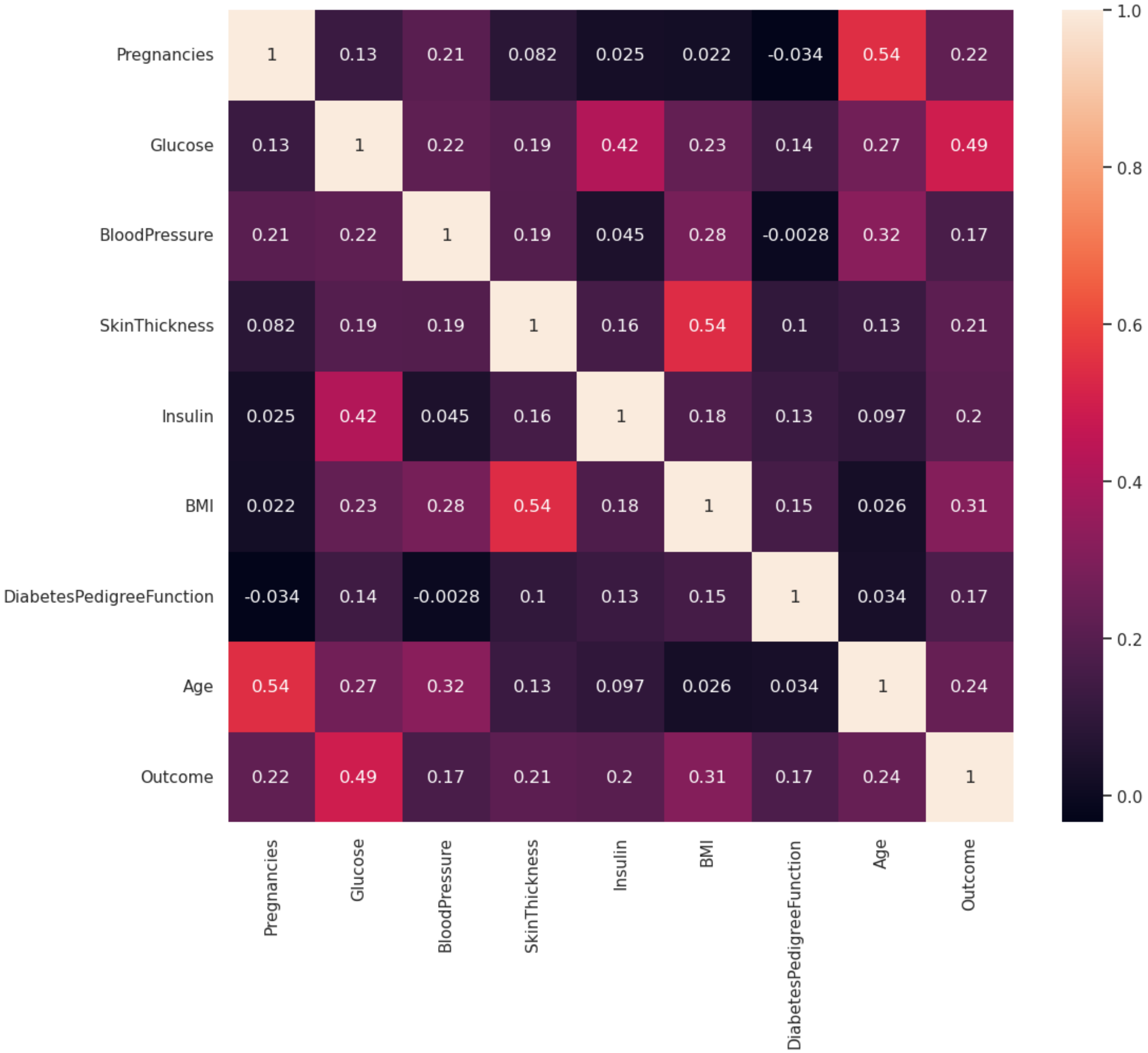
Out[55]:

	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


Pregnancies
250
200
150
100
50
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Glucose
160
140
120
100
80
60
40
20
0
40
60
80
100
120
140
160
180
200
BloodPressure
250
200
150
100
50
0
20
40
60
80
100
120
SkinThickness
400
350
300
250
200
150
100
50
0
20
40
60
80
100
Insulin
500
400
300
200
100
0
0
200
400
600
800
BMI
200
175
150
125
100
75
50
25
0
20
30
40
50
60
DiabetesPedigreeFunction
300
250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
Age
300
250
200
150
100
50
0
20
30
40
50
60
70
80
Outcome
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0


500
400
300
200
100
0
0
1


Pregnancies
0
5
10
15
Glucose
50
75
100
125
150
175
200
BloodPressure
20
40
60
80
100
120
SkinThickness
20
40
60
80
100
Insulin
0
200
400
600
800
BMI
20
30
40
50
60
DiabetesPedigreeFunction
0.0
0.5
1.0
1.5
2.0
2.5
Age
20
30
40
50
60
70
80
0
10
20
Pregnancies
100
200
Glucose
50
100
BloodPressure
0
50
100
SkinThickness
0
500
Insulin
20
40
60
BMI
0
1
2
DiabetesPedigreeFunction
25
50
75
Age
Outcome
● 0
● 1

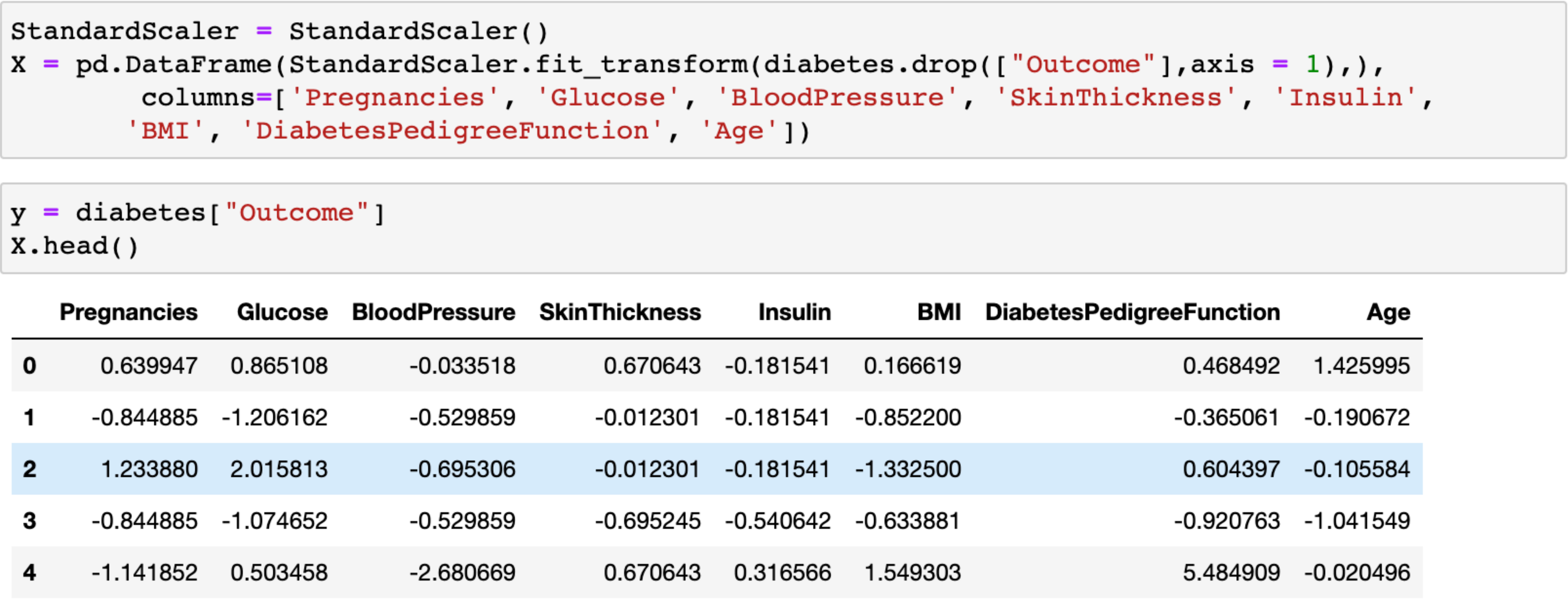

Pregnancies 1 0.13 0.21 0.082 0.025 0.022 -0.034 0.54 0.22
Glucose 0.13 1 0.22 0.19 0.42 0.23 0.14 0.27 0.49
BloodPressure 0.21 0.22 1 0.19 0.045 0.28 -0.0028 0.32 0.17
SkinThickness 0.082 0.19 0.19 1 0.16 0.54 0.1 0.13 0.21
Insulin 0.025 0.42 0.045 0.16 1 0.18 0.13 0.097 0.2
BMI 0.022 0.23 0.28 0.54 0.18 1 0.15 0.026 0.31
DiabetesPedigreeFunction -0.034 0.14 -0.0028 0.1 0.13 0.15 1 0.034 0.17
Age 0.54 0.27 0.32 0.13 0.097 0.026 0.034 1 0.24
Outcome 0.22 0.49 0.17 0.21 0.2 0.31 0.17 0.24 1
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
1.0
0.8
0.6
0.4
0.2
0.0

2. Modeling

Scaling

$$z = \frac{x_i - \mu}{\sigma}$$

Scaling

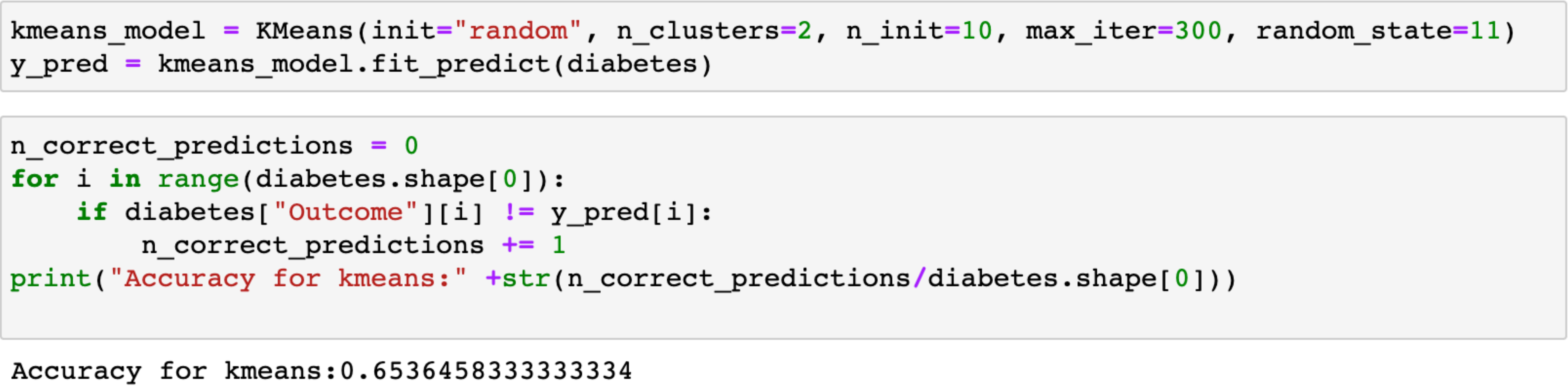


```python
StandardScaler = StandardScaler()
X = pd.DataFrame(StandardScaler.fit_transform(diabetes.drop(["Outcome"],axis = 1),),
        columns=['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age'])
```

```python
y = diabetes["Outcome"]
X.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	0.639947	0.865108	-0.033518	0.670643	-0.181541	0.166619	0.468492	1.425995
1	-0.844885	-1.206162	-0.529859	-0.012301	-0.181541	-0.852200	-0.365061	-0.190672
2	1.233880	2.015813	-0.695306	-0.012301	-0.181541	-1.332500	0.604397	-0.105584
3	-0.844885	-1.074652	-0.529859	-0.695245	-0.540642	-0.633881	-0.920763	-1.041549
4	-1.141852	0.503458	-2.680669	0.670643	0.316566	1.549303	5.484909	-0.020496

Unsupervised Approach - KMeans

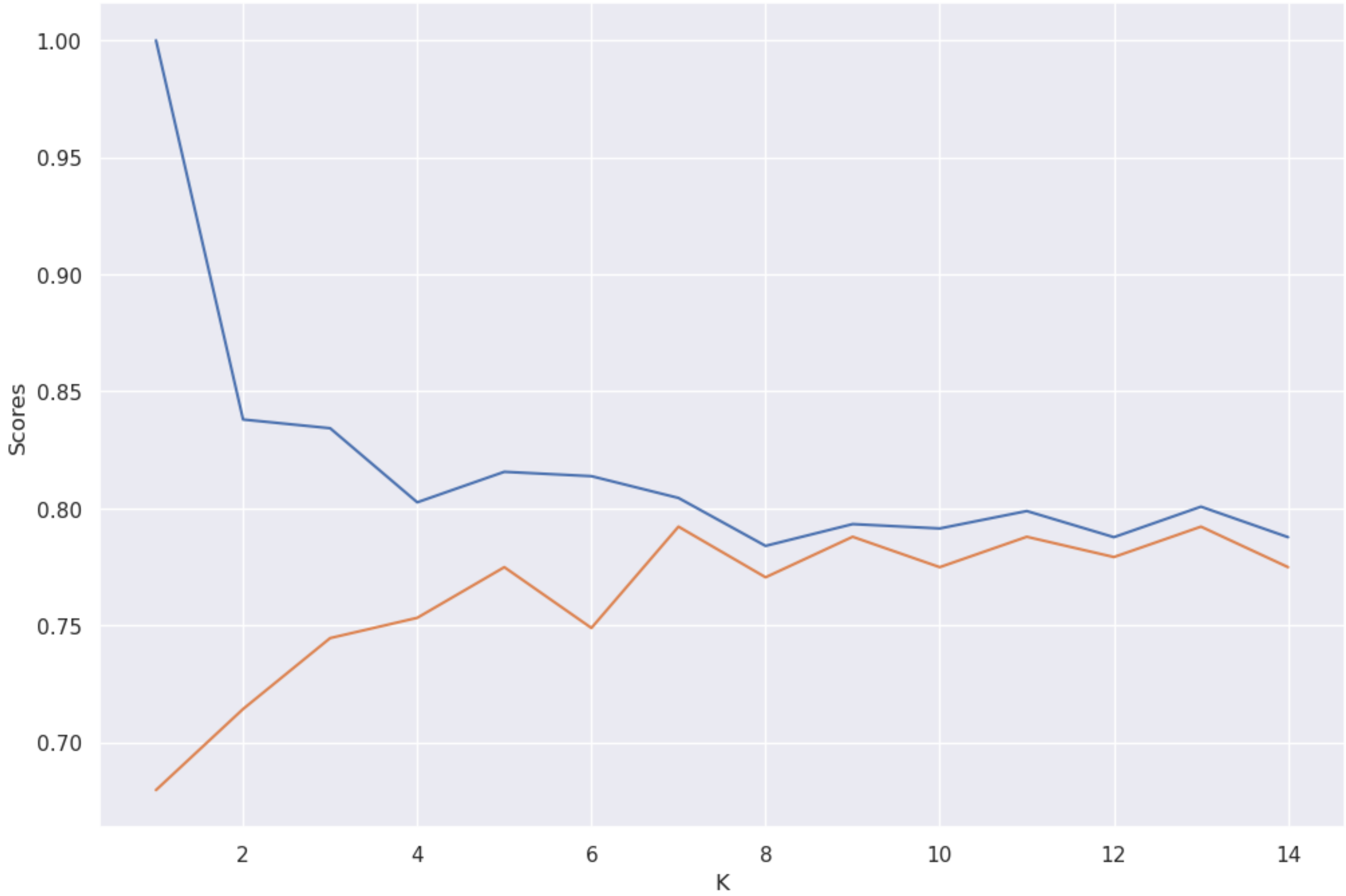


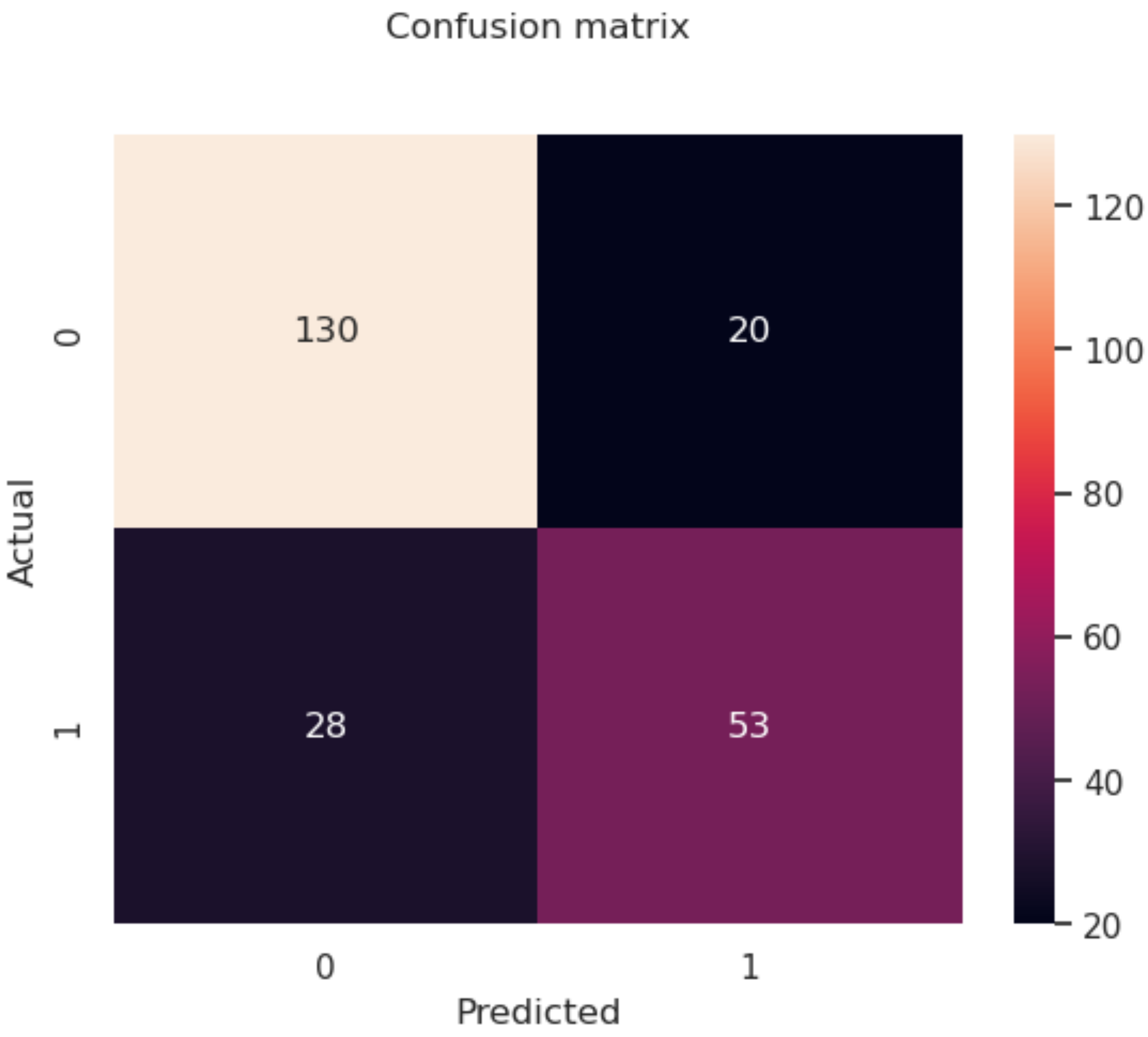
```
kmeans_model = KMeans(init="random", n_clusters=2, n_init=10, max_iter=300, random_state=11)
y_pred = kmeans_model.fit_predict(diabetes)
```

```
n_correct_predictions = 0
for i in range(diabetes.shape[0]):
    if diabetes["Outcome"][i] != y_pred[i]:
        n_correct_predictions += 1
print("Accuracy for kmeans:" +str(n_correct_predictions/diabetes.shape[0]))
```

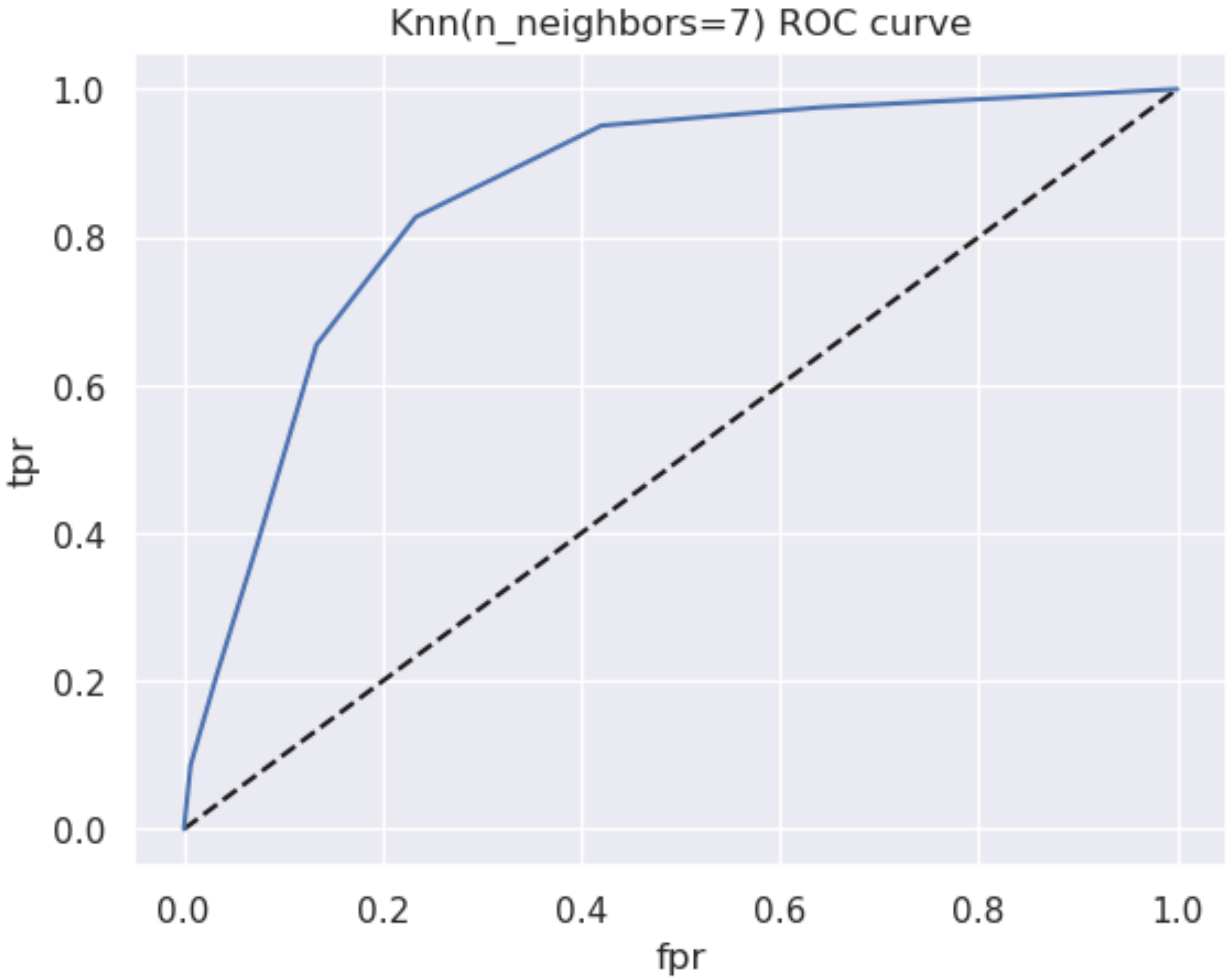
```
Accuracy for kmeans:0.6536458333333334
```

Supervised Approach - KNN





Confusion matrix


Knn(n_neighbors=7) ROC curve
tpr
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
fpr