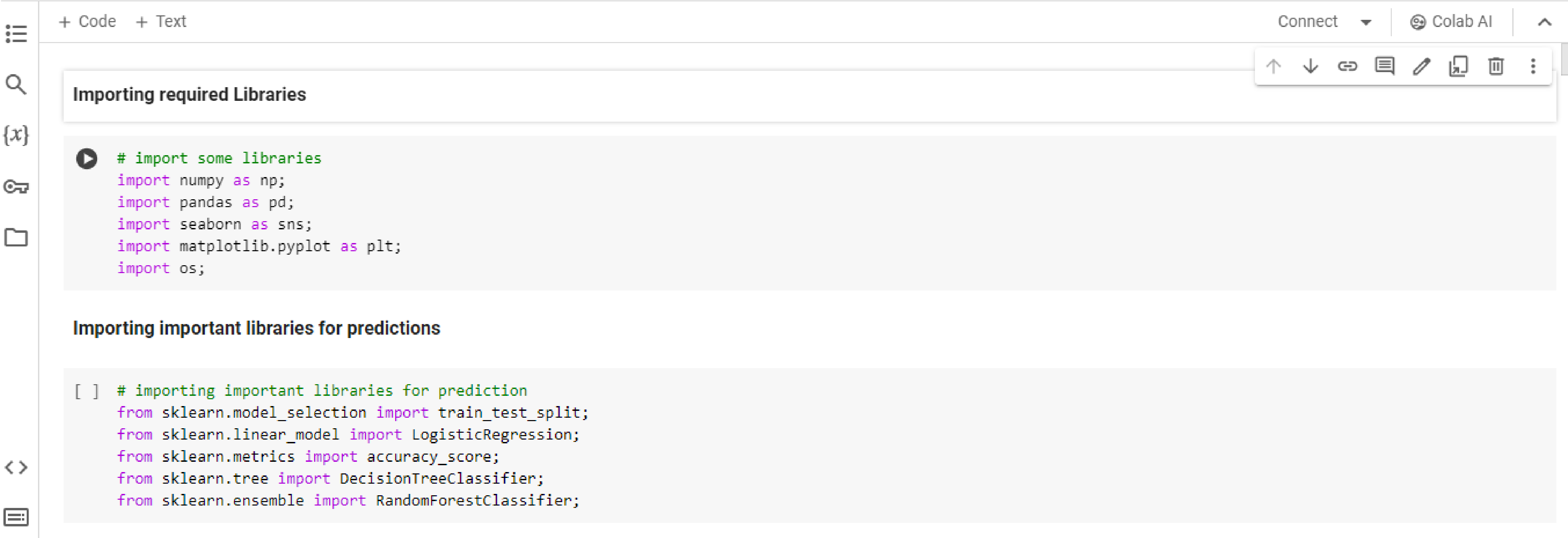
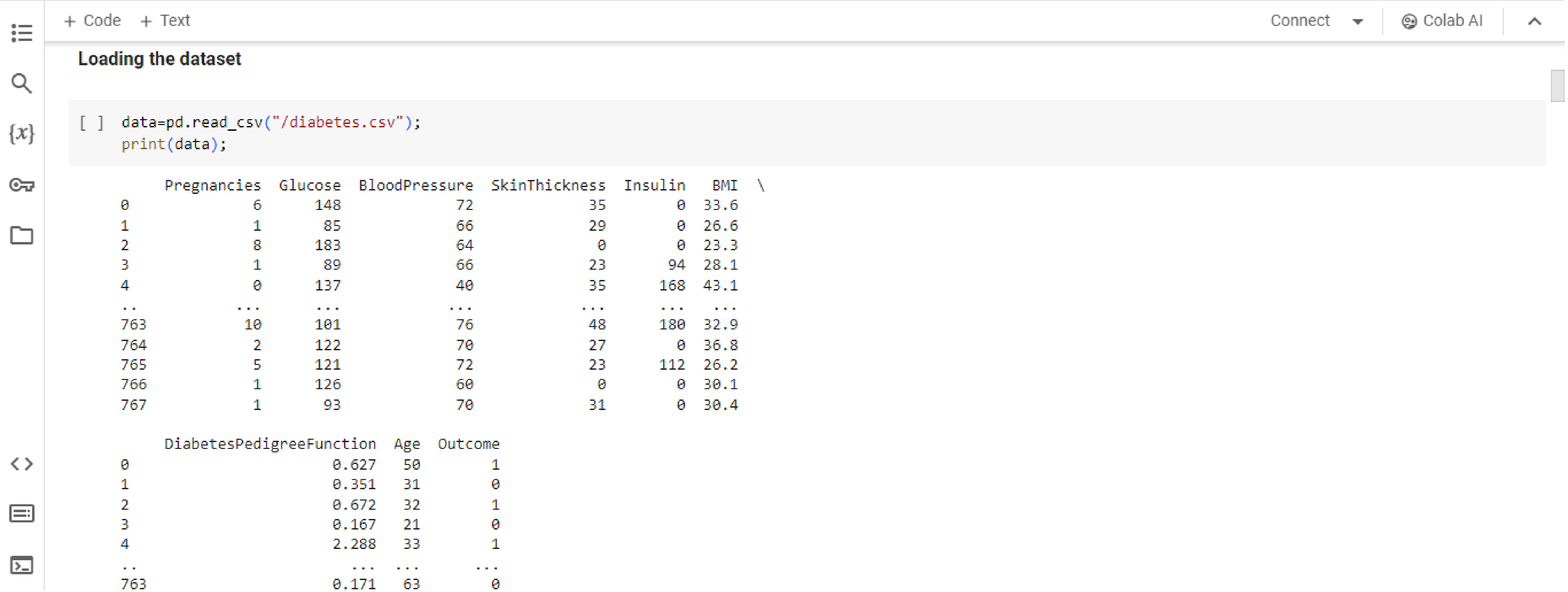


Importing required Libraries

```
# import some libraries
import numpy as np;
import pandas as pd;
import seaborn as sns;
import matplotlib.pyplot as plt;
import os;
```

Importing important libraries for predictions

```
[ ] # importing important libraries for prediction
    from sklearn.model_selection import train_test_split;
    from sklearn.linear_model import LogisticRegression;
    from sklearn.metrics import accuracy_score;
    from sklearn.tree import DecisionTreeClassifier;
    from sklearn.ensemble import RandomForestClassifier;
```

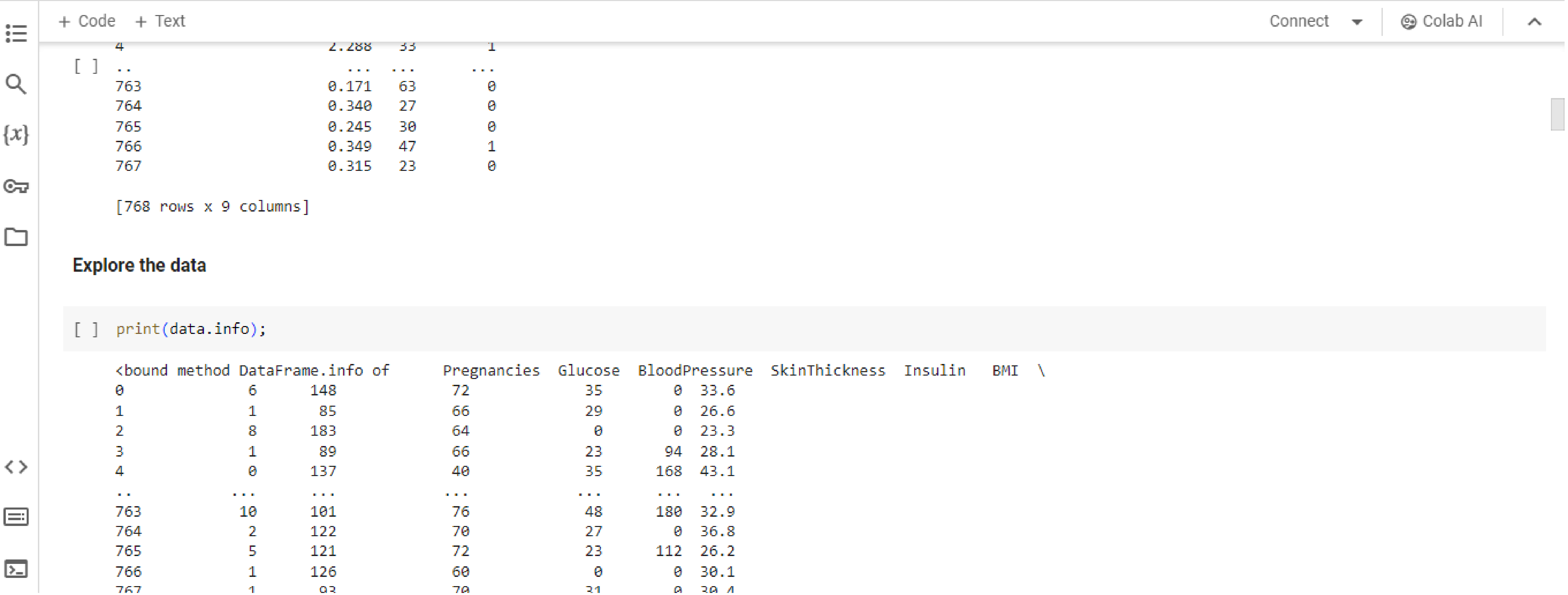


Loading the dataset

```
[ ] data=pd.read_csv("/diabetes.csv");
    print(data);
```

```
     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  Age  Outcome  
0                       0.627   50        1  
1                       0.351   31        0  
2                       0.672   32        1  
3                       0.167   21        0  
4                       2.288   33        1  
..                        ...  ...      ...  
763                     0.171   63        0  
```



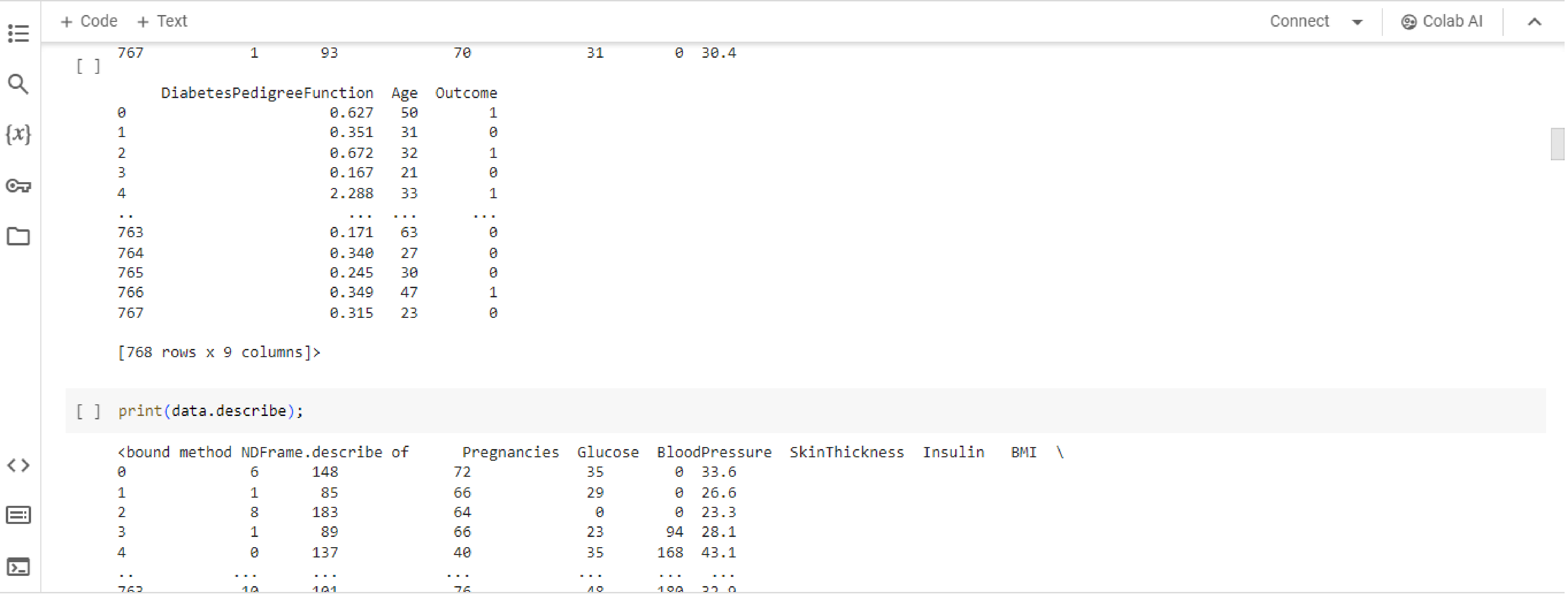
```
4                        2.288   33         1
..                         ...  ...       ...
763                      0.171   63         0
764                      0.340   27         0
765                      0.245   30         0
766                      0.349   47         1
767                      0.315   23         0

[768 rows x 9 columns]
```

Explore the data

```
[ ] print(data.info);
```

```
<bound method DataFrame.info of      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
```



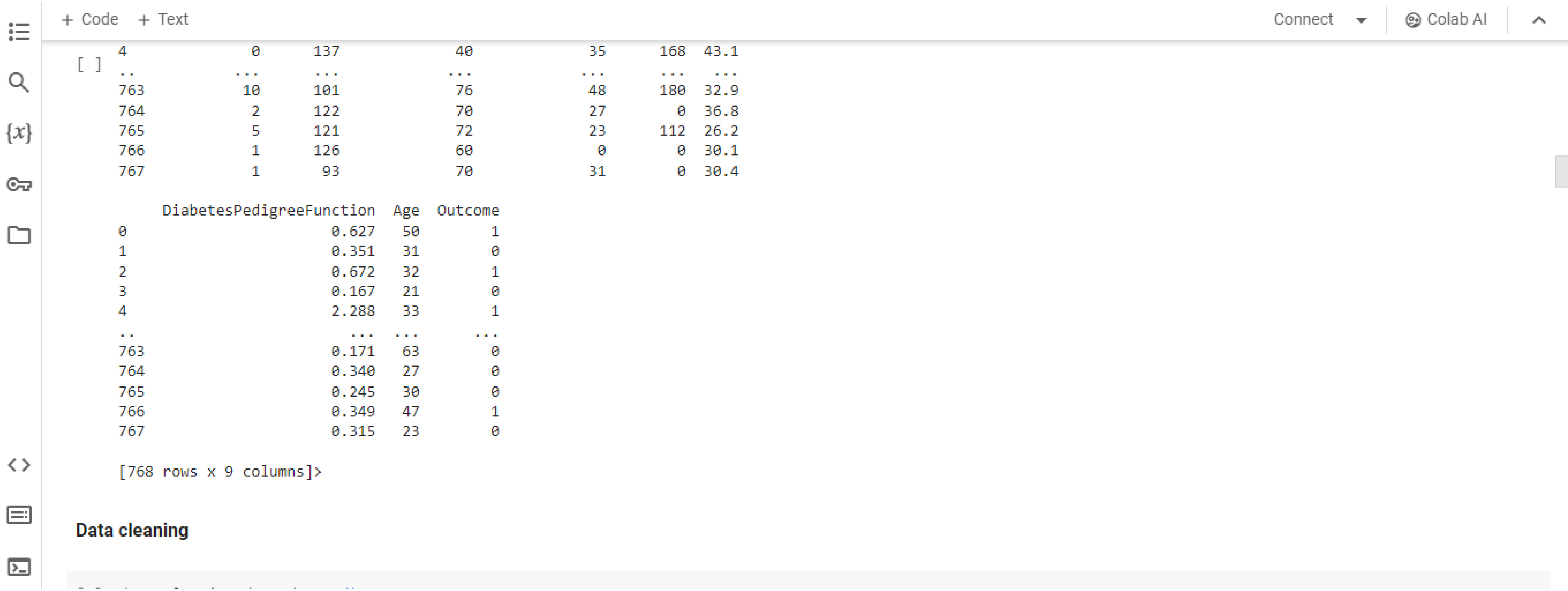
```
767            1       93           70           31        0  30.4
[ ]

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
..                        ...  ...      ...
763                     0.171   63        0
764                     0.340   27        0
765                     0.245   30        0
766                     0.349   47        1
767                     0.315   23        0

[768 rows x 9 columns]>
```

```
[ ] print(data.describe);
```

```
<bound method NDFrame.describe of      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
```

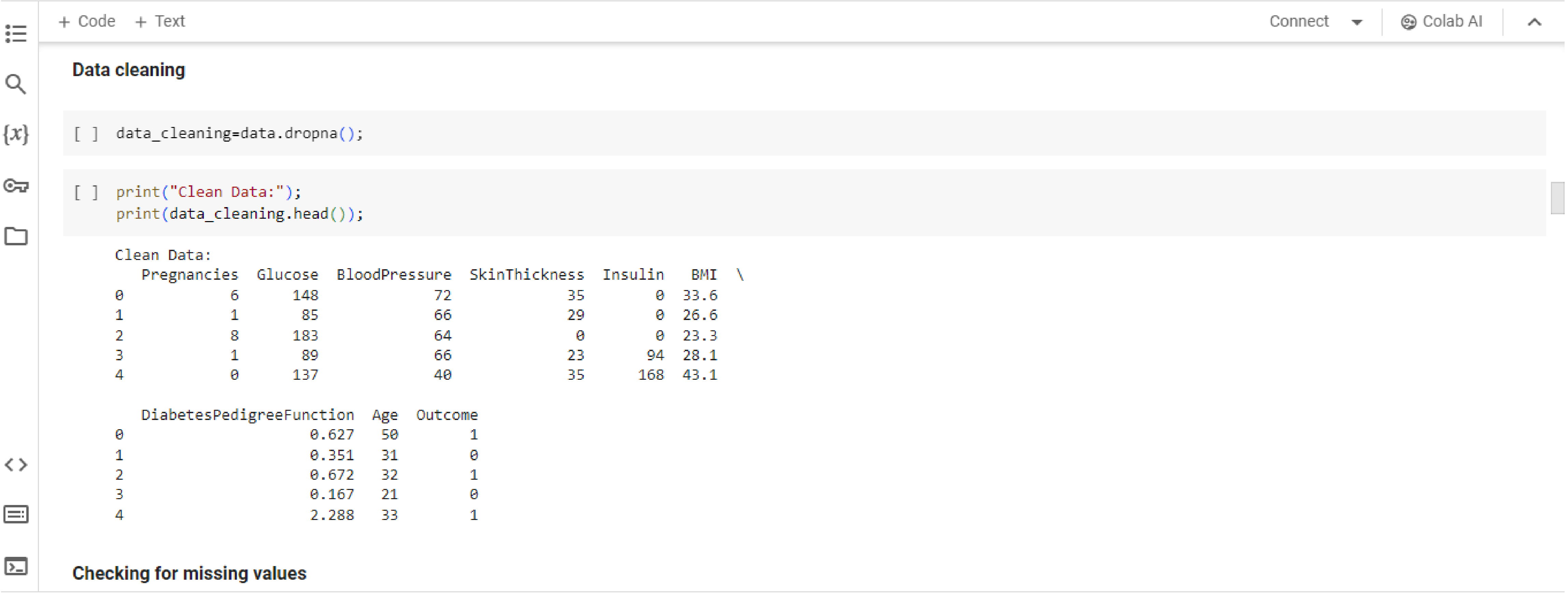


```
4              0      137             40             35      168  43.1
..           ...      ...            ...            ...      ...   ...
763           10      101             76             48      180  32.9
764            2      122             70             27        0  36.8
765            5      121             72             23      112  26.2
766            1      126             60              0        0  30.1
767            1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
..                        ...  ...      ...
763                     0.171   63        0
764                     0.340   27        0
765                     0.245   30        0
766                     0.349   47        1
767                     0.315   23        0

[768 rows x 9 columns]>
```

Data cleaning



Data cleaning

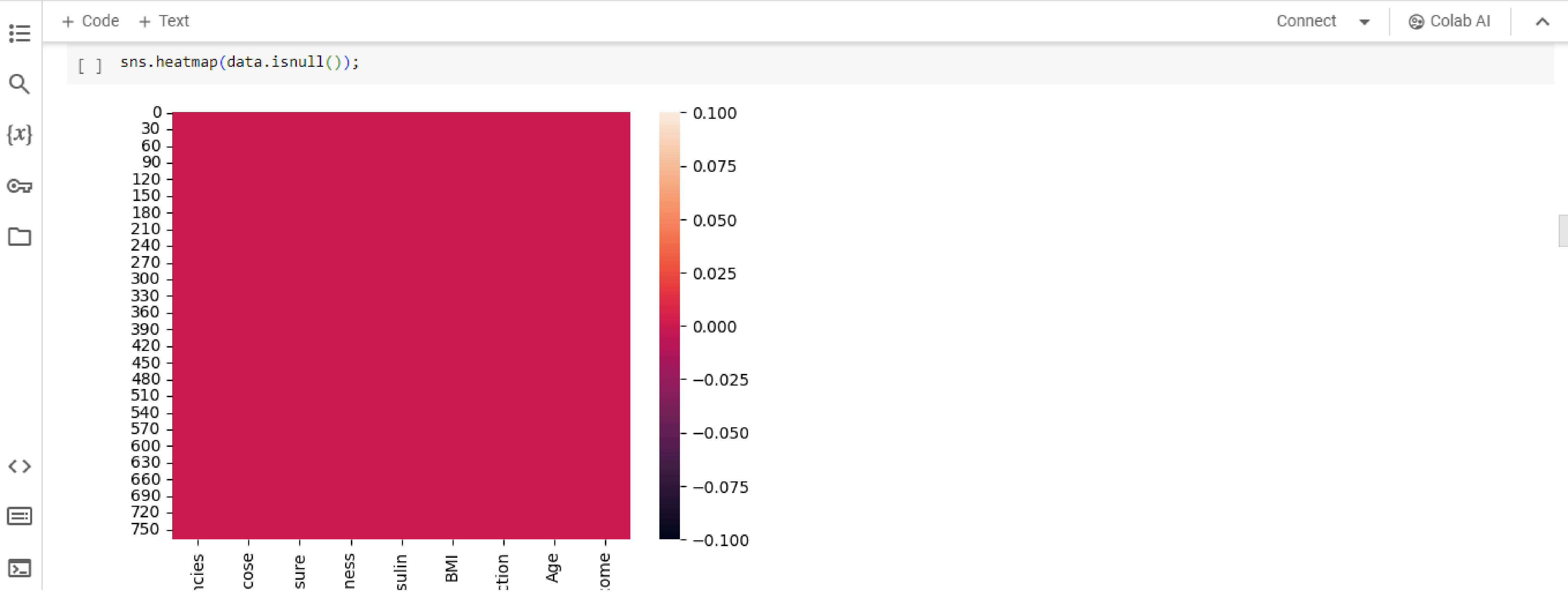
```
[ ]  data_cleaning=data.dropna();
```

```
[ ]  print("Clean Data:");
     print(data_cleaning.head());
```

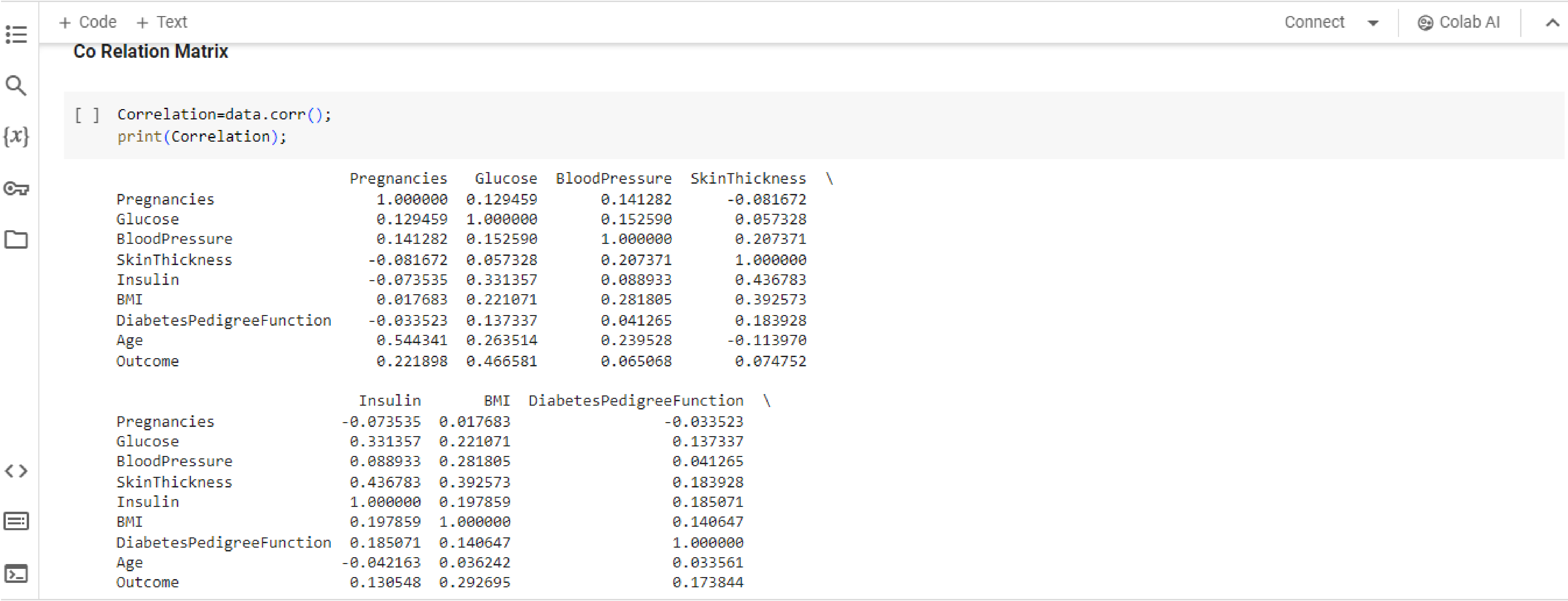
```
Clean Data:
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6   
1            1       85             66             29        0  26.6   
2            8      183             64              0        0  23.3   
3            1       89             66             23       94  28.1   
4            0      137             40             35      168  43.1   

   DiabetesPedigreeFunction  Age  Outcome  
0                     0.627   50        1  
1                     0.351   31        0  
2                     0.672   32        1  
3                     0.167   21        0  
4                     2.288   33        1  
```

Checking for missing values



```
[ ] sns.heatmap(data.isnull());
```

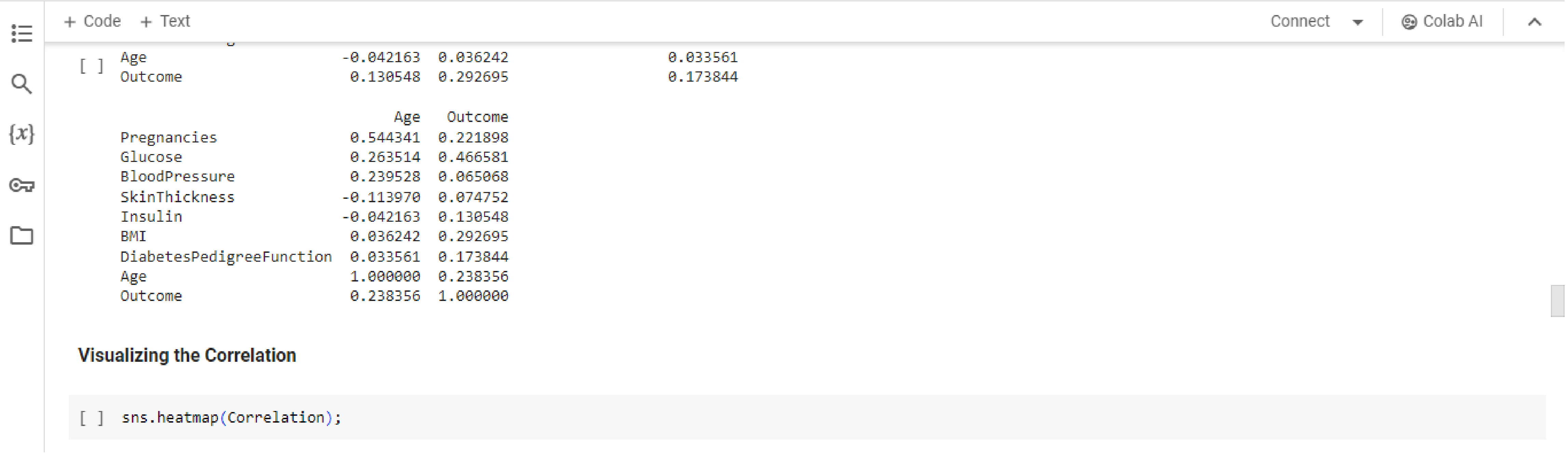


Co Relation Matrix

```
[ ] Correlation=data.corr();
    print(Correlation);
```

```
                          Pregnancies   Glucose  BloodPressure  SkinThickness  \
Pregnancies                  1.000000  0.129459       0.141282      -0.081672   
Glucose                      0.129459  1.000000       0.152590       0.057328   
BloodPressure                0.141282  0.152590       1.000000       0.207371   
SkinThickness               -0.081672  0.057328       0.207371       1.000000   
Insulin                     -0.073535  0.331357       0.088933       0.436783   
BMI                          0.017683  0.221071       0.281805       0.392573   
DiabetesPedigreeFunction    -0.033523  0.137337       0.041265       0.183928   
Age                          0.544341  0.263514       0.239528      -0.113970   
Outcome                      0.221898  0.466581       0.065068       0.074752   

                           Insulin       BMI  DiabetesPedigreeFunction  \
Pregnancies              -0.073535  0.017683                 -0.033523   
Glucose                   0.331357  0.221071                  0.137337   
BloodPressure             0.088933  0.281805                  0.041265   
SkinThickness             0.436783  0.392573                  0.183928   
Insulin                   1.000000  0.197859                  0.185071   
BMI                       0.197859  1.000000                  0.140647   
DiabetesPedigreeFunction  0.185071  0.140647                  1.000000   
Age                      -0.042163  0.036242                  0.033561   
Outcome                   0.130548  0.292695                  0.173844   
```

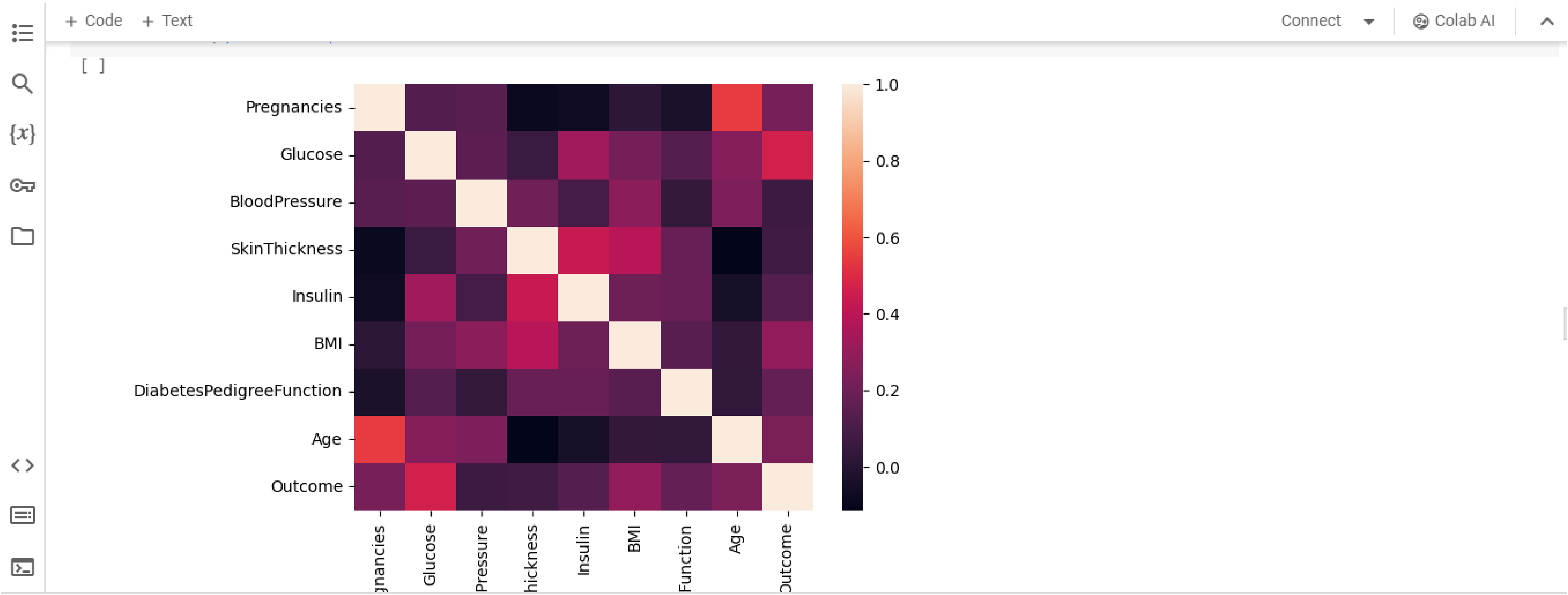


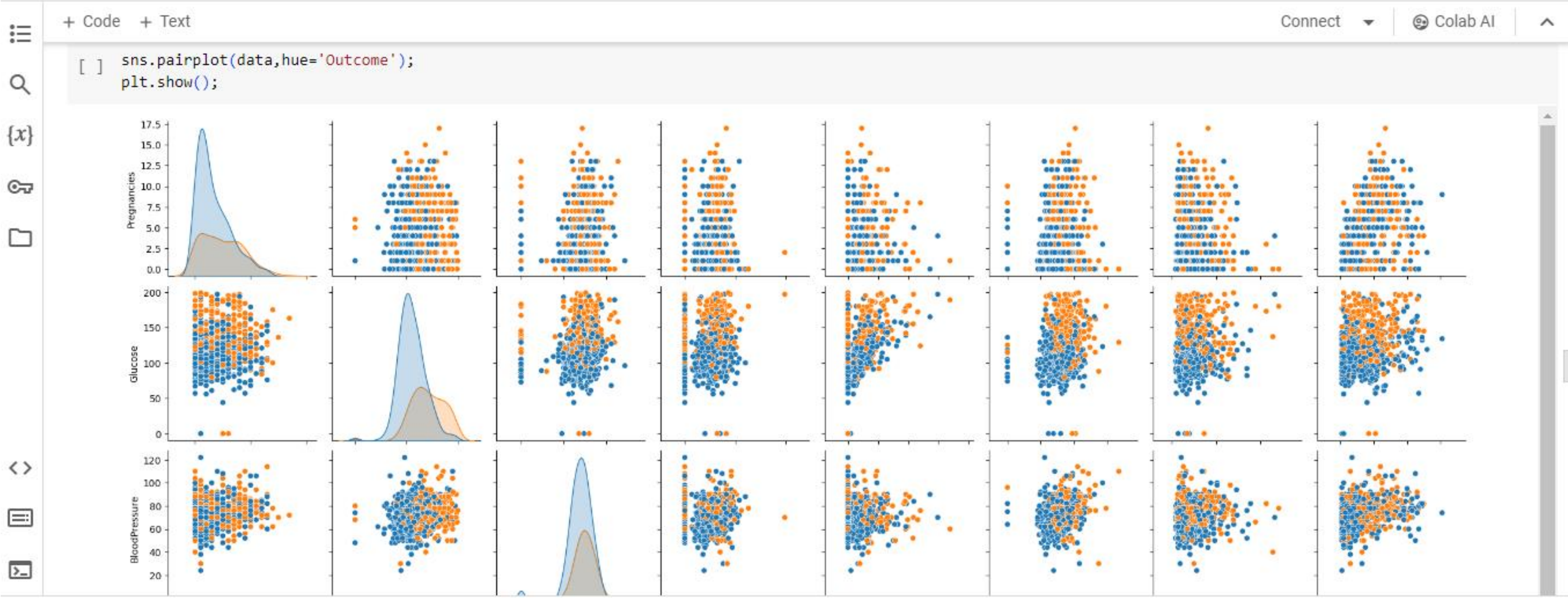
```
[ ]  Age                      -0.042163  0.036242                      0.033561
     Outcome                   0.130548  0.292695                      0.173844

                                Age   Outcome
     Pregnancies               0.544341  0.221898
     Glucose                   0.263514  0.466581
     BloodPressure             0.239528  0.065068
     SkinThickness            -0.113970  0.074752
     Insulin                  -0.042163  0.130548
     BMI                       0.036242  0.292695
     DiabetesPedigreeFunction  0.033561  0.173844
     Age                       1.000000  0.238356
     Outcome                   0.238356  1.000000
```

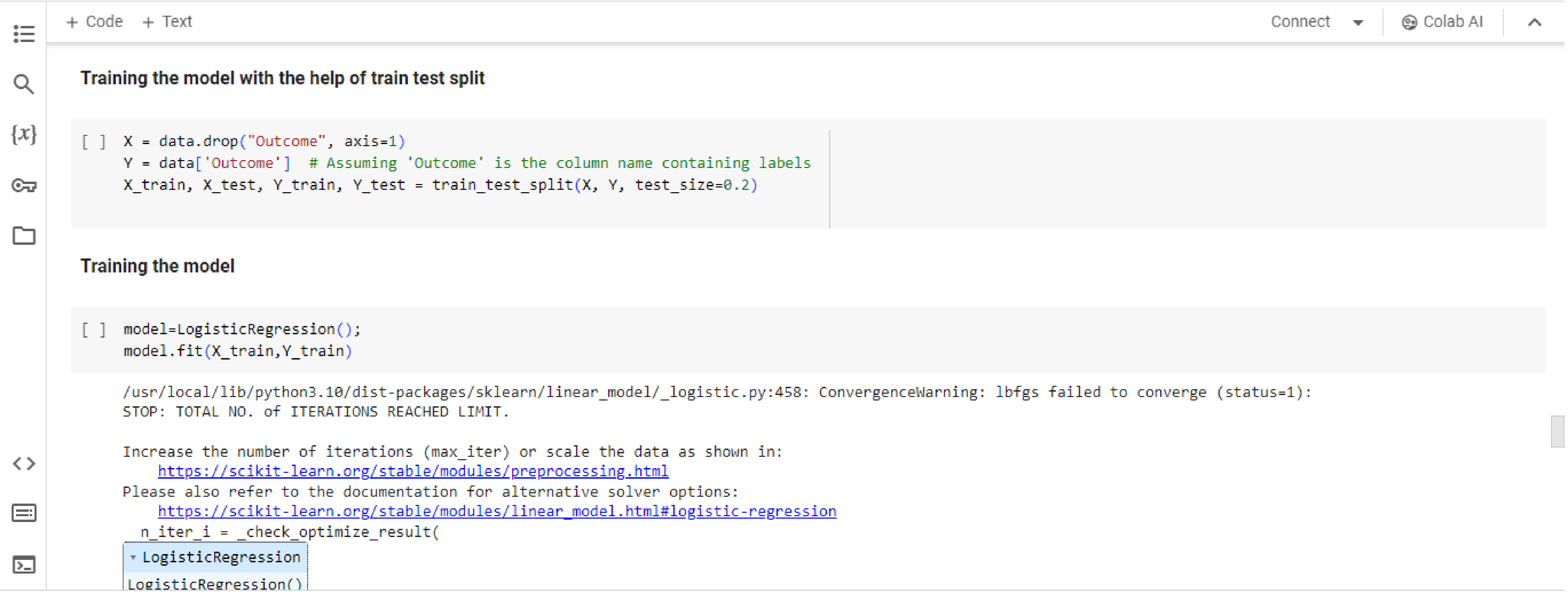
Visualizing the Correlation

```
[ ]  sns.heatmap(Correlation);
```


+ Code
+ Text
Connect
Colab AI
[ ]
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
gnancies
Glucose
Pressure
hickness
Insulin
BMI
Function
Age
Outcome
1.0
0.8
0.6
0.4
0.2
0.0



```
sns.pairplot(data,hue='Outcome');
plt.show();
```



Training the model with the help of train test split

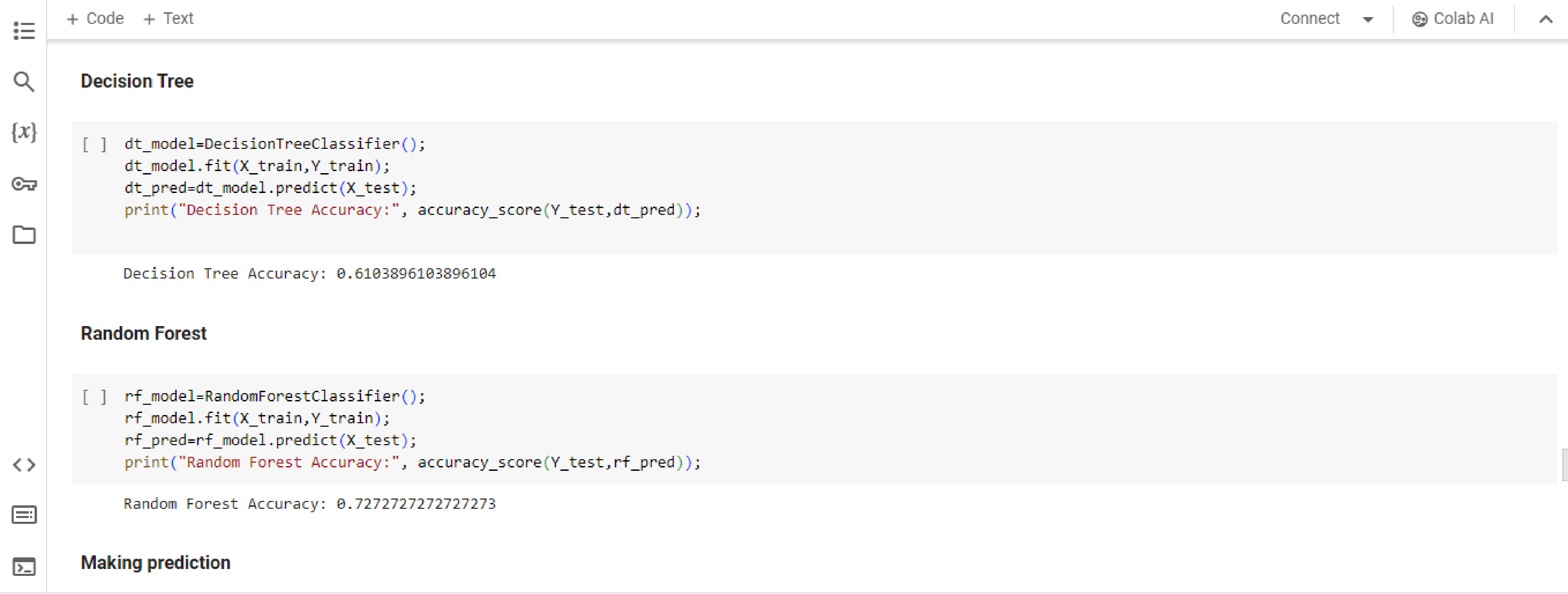
```
[ ] X = data.drop("Outcome", axis=1)
    Y = data['Outcome']  # Assuming 'Outcome' is the column name containing labels
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)
```

Training the model

```
[ ] model=LogisticRegression();
    model.fit(X_train,Y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
▾ LogisticRegression
LogisticRegression()
```



Decision Tree

```
[ ] dt_model=DecisionTreeClassifier();
    dt_model.fit(X_train,Y_train);
    dt_pred=dt_model.predict(X_test);
    print("Decision Tree Accuracy:", accuracy_score(Y_test,dt_pred));
```

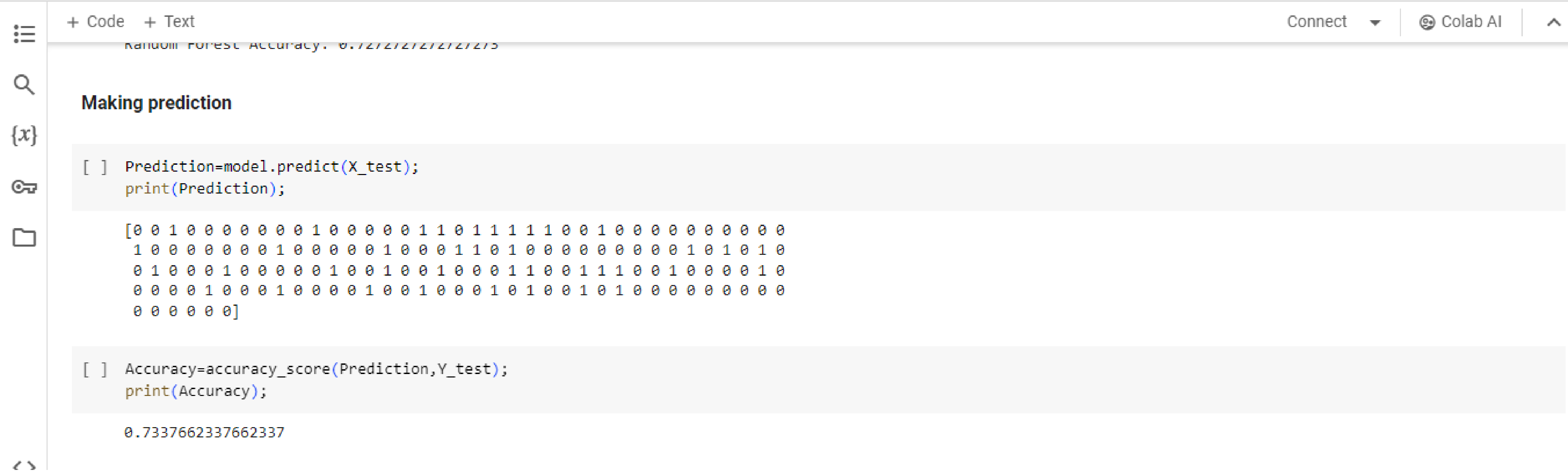
```
Decision Tree Accuracy: 0.6103896103896104
```

Random Forest

```
[ ] rf_model=RandomForestClassifier();
    rf_model.fit(X_train,Y_train);
    rf_pred=rf_model.predict(X_test);
    print("Random Forest Accuracy:", accuracy_score(Y_test,rf_pred));
```

```
Random Forest Accuracy: 0.7272727272727273
```

Making prediction



```
Random Forest Accuracy: 0.7272727272727273
```

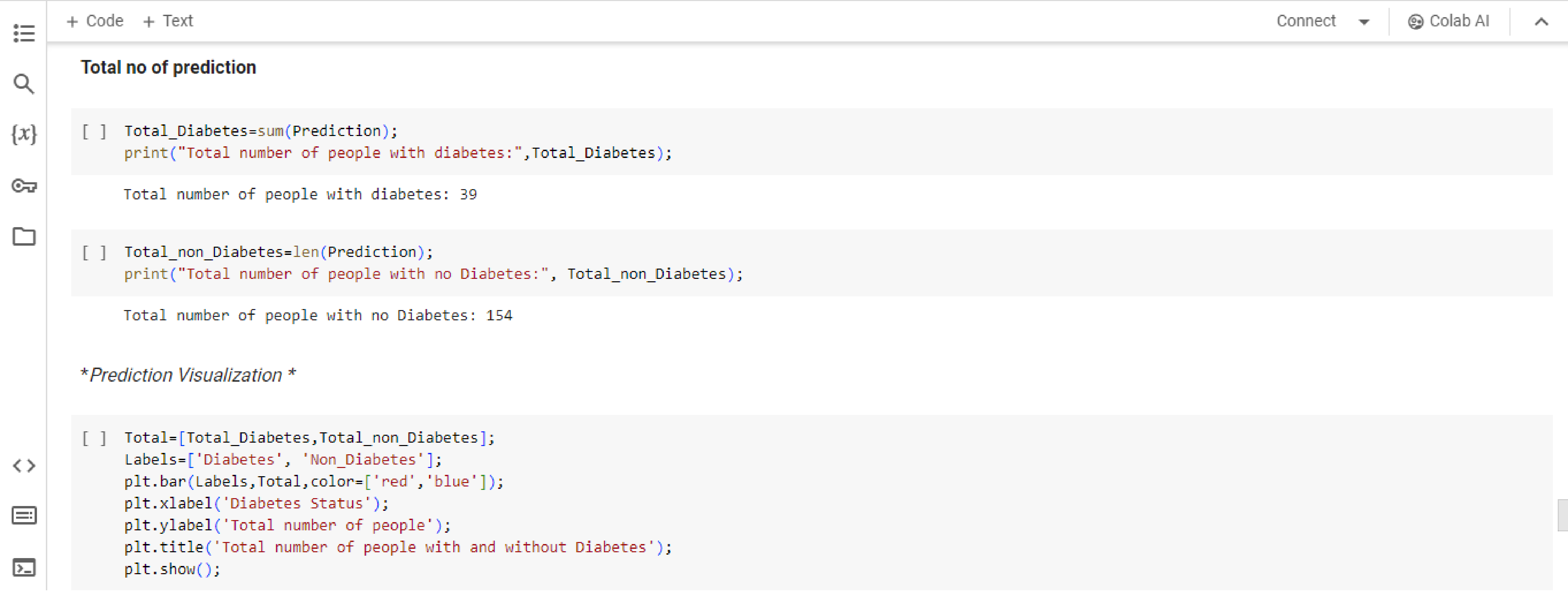
Making prediction

```
[ ] Prediction=model.predict(X_test);
    print(Prediction);
```

```
[0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0
 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 0
 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0]
```

```
[ ] Accuracy=accuracy_score(Prediction,Y_test);
    print(Accuracy);
```

```
0.7337662337662337
```



Total no of prediction

```
[ ] Total_Diabetes=sum(Prediction);
    print("Total number of people with diabetes:",Total_Diabetes);
```

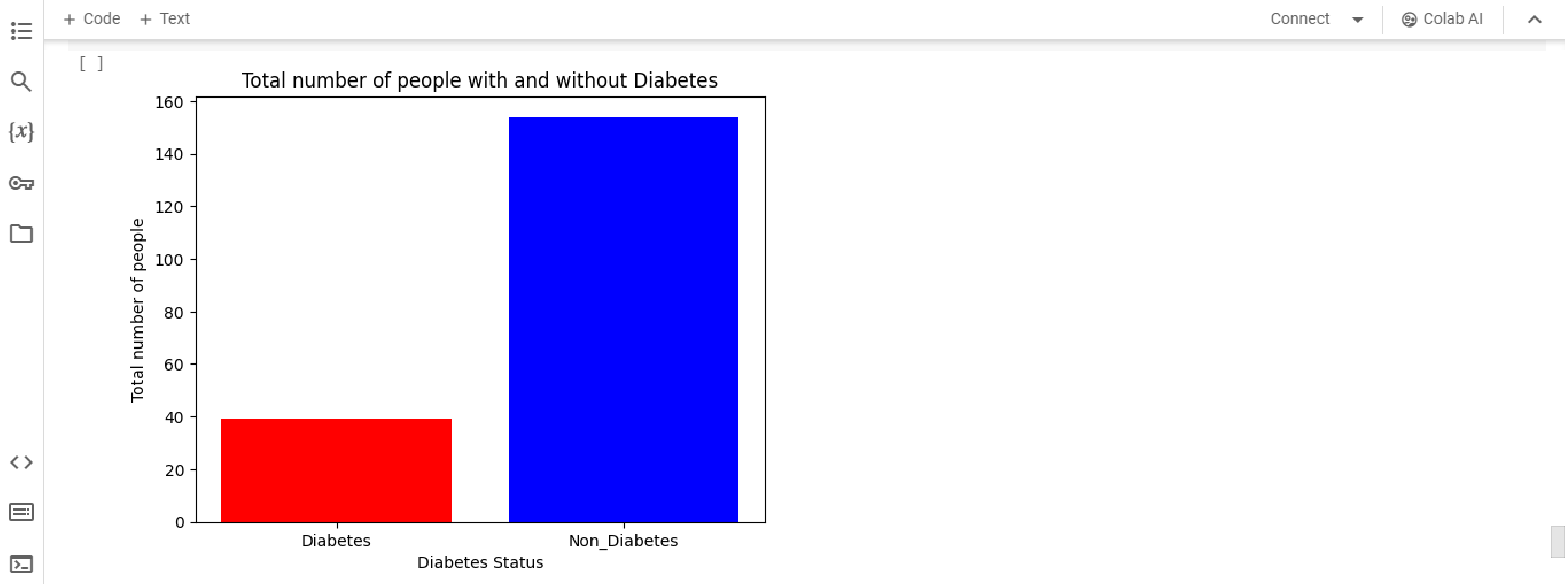
```
Total number of people with diabetes: 39
```

```
[ ] Total_non_Diabetes=len(Prediction);
    print("Total number of people with no Diabetes:", Total_non_Diabetes);
```

```
Total number of people with no Diabetes: 154
```

**Prediction Visualization **

```
[ ] Total=[Total_Diabetes,Total_non_Diabetes];
    Labels=['Diabetes', 'Non_Diabetes'];
    plt.bar(Labels,Total,color=['red','blue']);
    plt.xlabel('Diabetes Status');
    plt.ylabel('Total number of people');
    plt.title('Total number of people with and without Diabetes');
    plt.show();
```



+ Code
+ Text
Connect
Colab AI
[ ]
Total number of people with and without Diabetes
Total number of people
160
140
120
100
80
60
40
20
0
Diabetes
Non_Diabetes
Diabetes Status