IMPORTING LIBRARIES

```
[23] import numpy as np
     import pandas as pd
     import seaborn as sns
     import matplotlib.pyplot as plt
```

LOADING DATASET

```
[24] df=pd.read_csv('/content/diabetes_prediction_dataset.csv')
     df
```

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	Female	80.0	0	1	never	25.19	6.6	140	0
1	Female	54.0	0	0	No Info	27.32	6.6	80	0
2	Male	28.0	0	0	never	27.32	5.7	158	0
3	Female	36.0	0	0	current	23.45	5.0	155	0

```
df=pd.read_csv('/content/diabetes_prediction_dataset.csv')
df
```

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	Female	80.0	0	1	never	25.19	6.6	140	0
1	Female	54.0	0	0	No Info	27.32	6.6	80	0
2	Male	28.0	0	0	never	27.32	5.7	158	0
3	Female	36.0	0	0	current	23.45	5.0	155	0
4	Male	76.0	1	1	current	20.14	4.8	155	0
...	...	...	...	...	...	...	...	...	...
99995	Female	80.0	0	0	No Info	27.32	6.2	90	0
99996	Female	2.0	0	0	No Info	17.37	6.5	100	0
99997	Male	66.0	0	0	former	27.83	5.7	155	0
99998	Female	24.0	0	0	never	35.42	4.0	100	0
99999	Female	57.0	0	0	current	22.43	6.6	90	0

100000 rows × 9 columns

DIABETES PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

```
df.head()
```

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	Female	80.0	0	1	never	25.19	6.6	140	0
1	Female	54.0	0	0	No Info	27.32	6.6	80	0
2	Male	28.0	0	0	never	27.32	5.7	158	0
3	Female	36.0	0	0	current	23.45	5.0	155	0
4	Male	76.0	1	1	current	20.14	4.8	155	0

Next steps: Generate code with df View recommended plots

```
[26] df.tail()
```

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
99995	Female	80.0	0	0	No Info	27.32	6.2	90	0
99996	Female	2.0	0	0	No Info	17.37	6.5	100	0
99997	Male	66.0	0	0	former	27.83	5.7	155	0
99998	Female	24.0	0	0	never	35.42	4.0	100	0

0s completed at 21:32

```
df.columns
```

```
Index(['gender', 'age', 'hypertension', 'heart_disease', 'smoking_history',
       'bmi', 'HbA1c_level', 'blood_glucose_level', 'diabetes'],
      dtype='object')
```

```
df.shape
```

```
(100000, 9)
```

[29]

```
df.describe()
```

	age	hypertension	heart_disease	bmi	HbA1c_level	blood_glucose_level	diabetes
count	100000.000000	100000.00000	100000.000000	100000.000000	100000.000000	100000.000000	100000.000000
mean	41.885856	0.07485	0.039420	27.320767	5.527507	138.058060	0.085000
std	22.516840	0.26315	0.194593	6.636783	1.070672	40.708136	0.278883
min	0.080000	0.00000	0.000000	10.010000	3.500000	80.000000	0.000000
25%	24.000000	0.00000	0.000000	23.630000	4.800000	100.000000	0.000000
50%	43.000000	0.00000	0.000000	27.320000	5.800000	140.000000	0.000000
75%	60.000000	0.00000	0.000000	29.580000	6.200000	159.000000	0.000000
max	80.000000	1.00000	1.000000	95.690000	9.000000	300.000000	1.000000

DIABETES PREDICTION USING ML

File Edit View Insert Runtime Tools Help Saving...

Comment Share

+ Code + Text

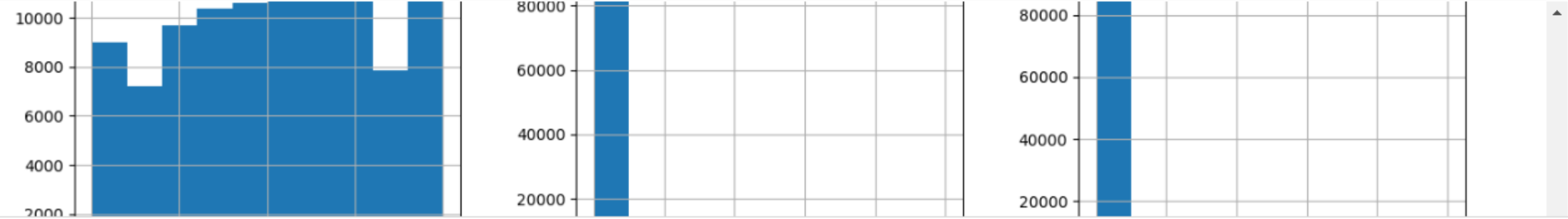
RAM Disk Colab AI

max	80.000000	1.00000	1.000000	95.690000	9.000000	300.000000	1.000000

```
df.isna().sum()
```

```
gender               0
age                  0
hypertension         0
heart_disease        0
smoking_history      0
bmi                  0
HbA1c_level          0
blood_glucose_level  0
diabetes             0
dtype: int64
```

```
df.hist(figsize=(16,13))
plt.show
```



0s completed at 21:32

DIABETES PREDICTION USING ML

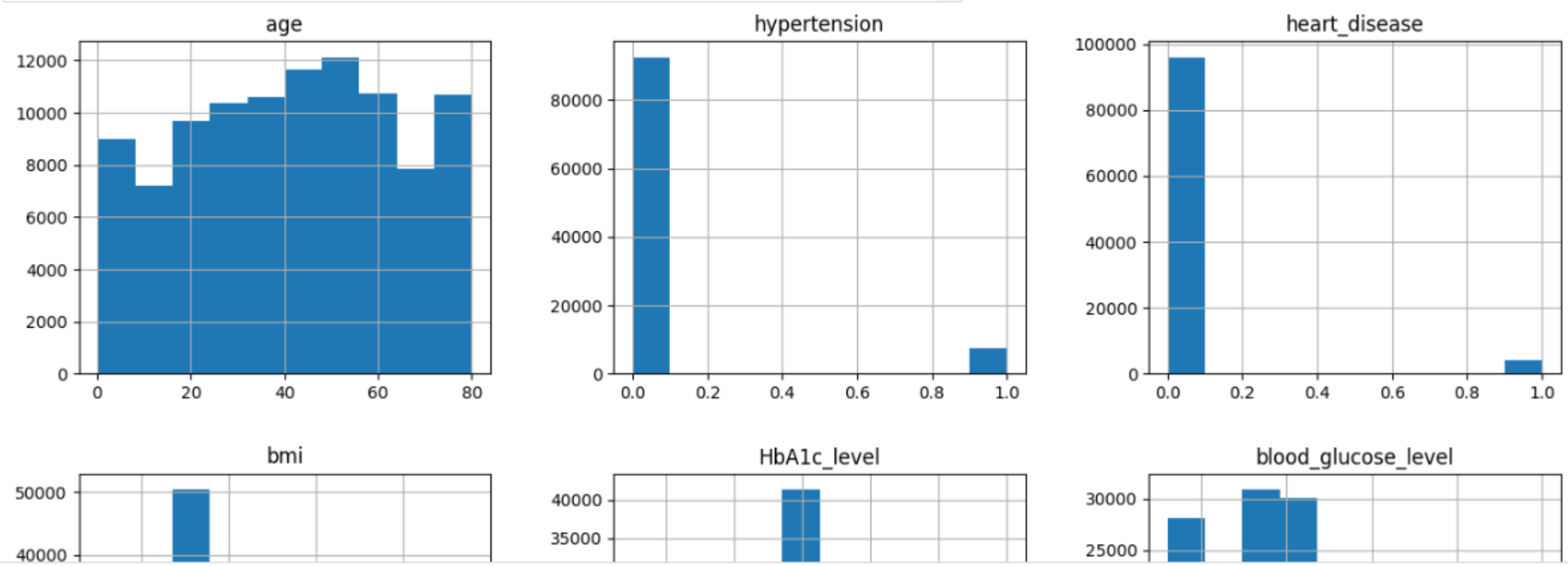
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

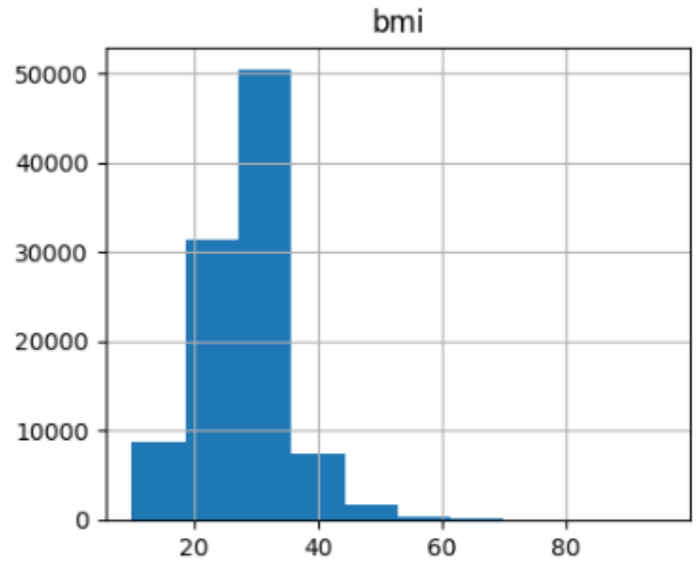
+ Code + Text

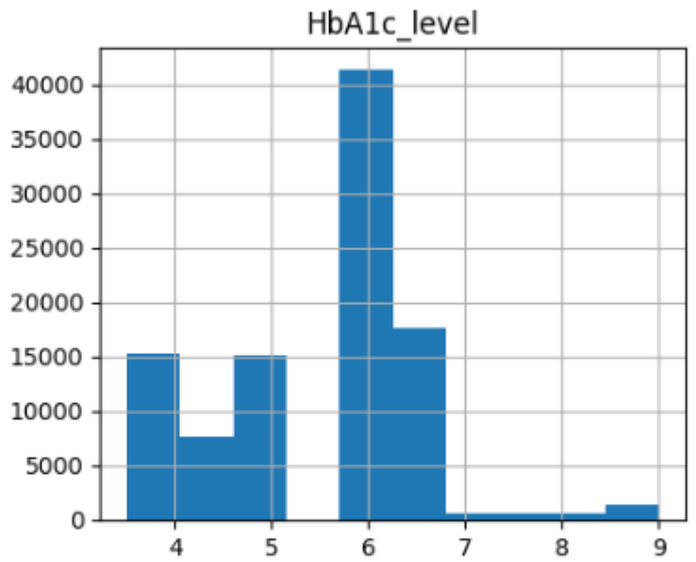
RAM Disk Colab AI

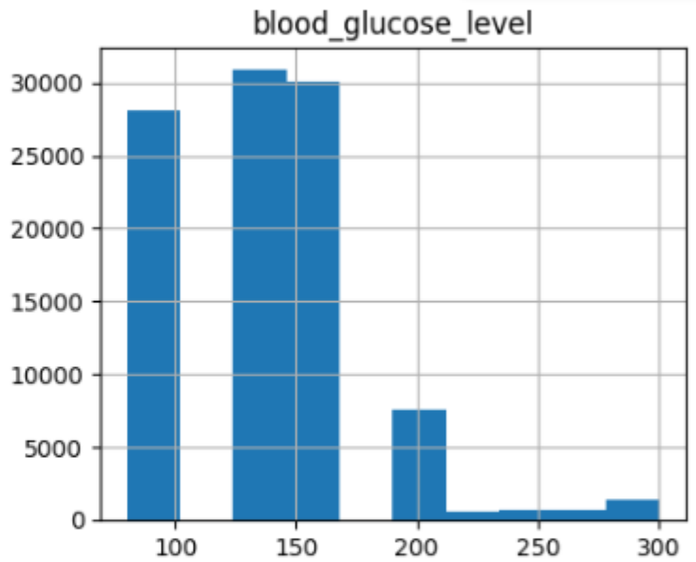
```
df.hist(figsize=(16,13))
plt.show
```



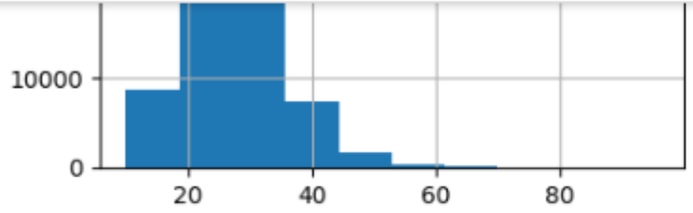
0s completed at 21:32

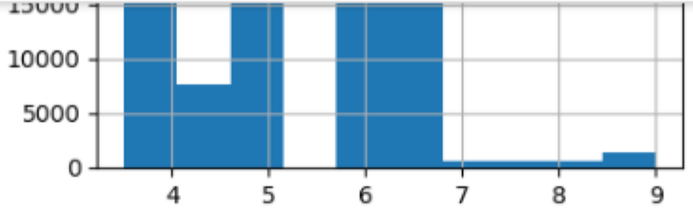

bmi
50000
40000
30000
20000
10000
0
20
40
60
80

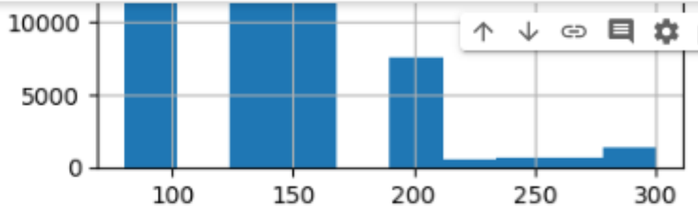

HbA1c_level
40000
35000
30000
25000
20000
15000
10000
5000
0
4
5
6
7
8
9

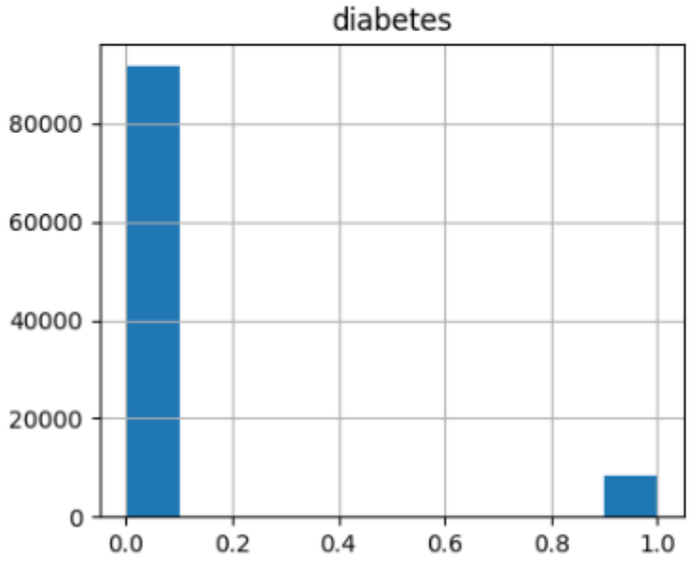

blood_glucose_level
30000
25000
20000
15000
10000
5000
0
100
150
200
250
300


diabetes
80000
60000










```
[39] sns.pairplot(df)
```

DIABETES PREDICTION USING ML

File Edit View Insert Runtime Tools Help

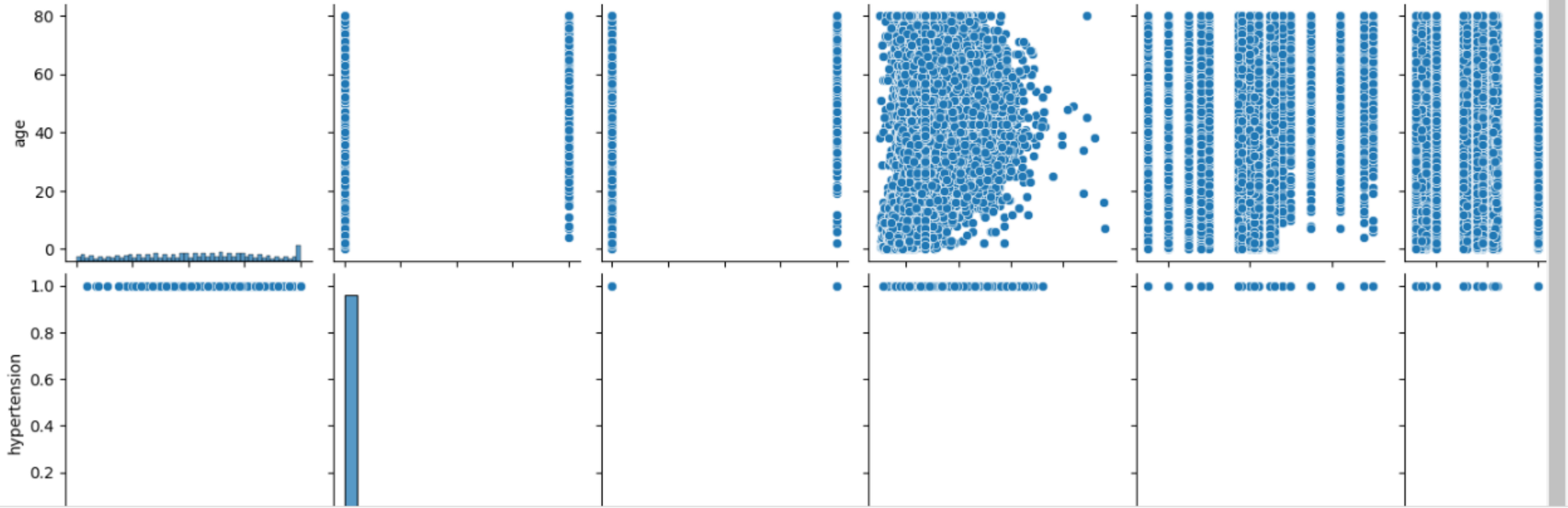
Comment Share

+ Code + Text

RAM Disk Colab AI

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7b05c55bc460>
```



0s completed at 21:32

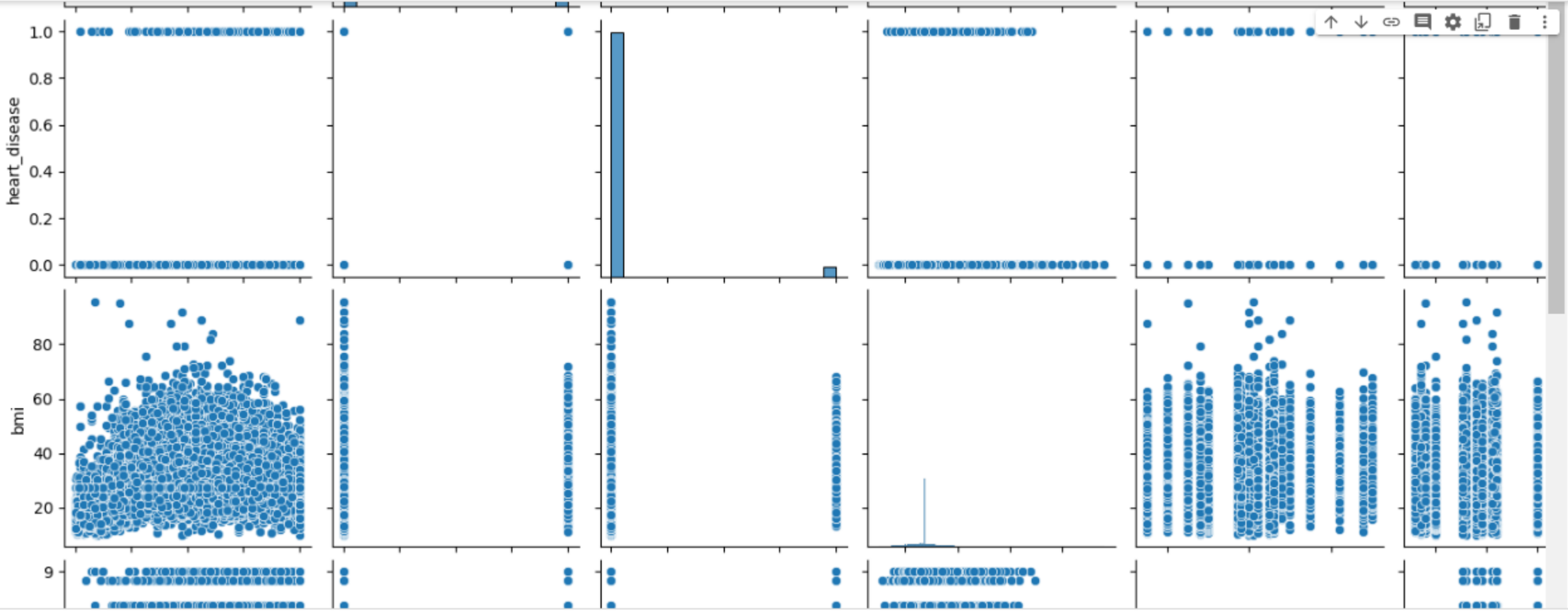
DIABETES PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

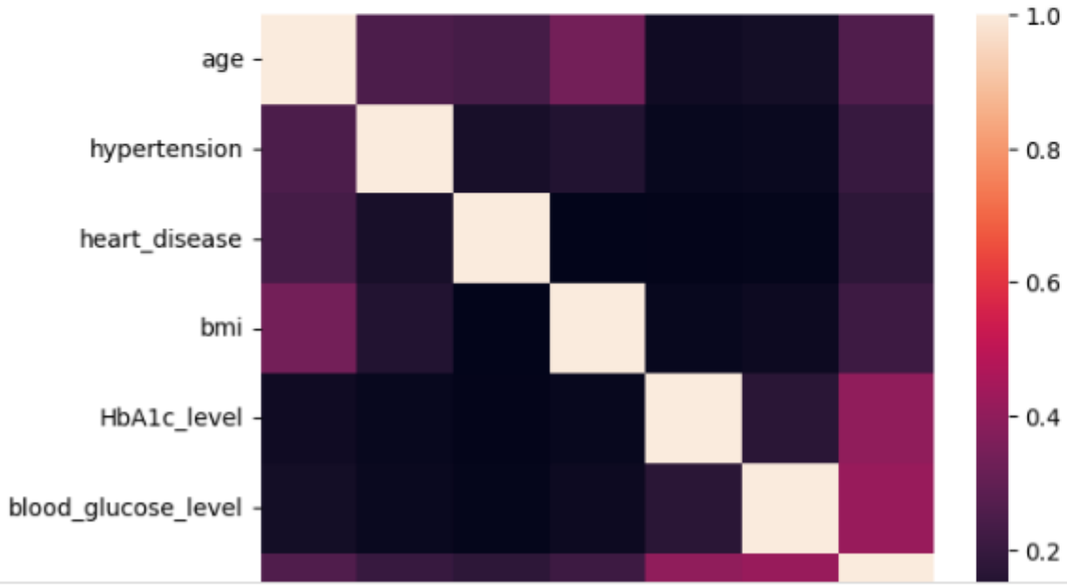


0s completed at 21:32

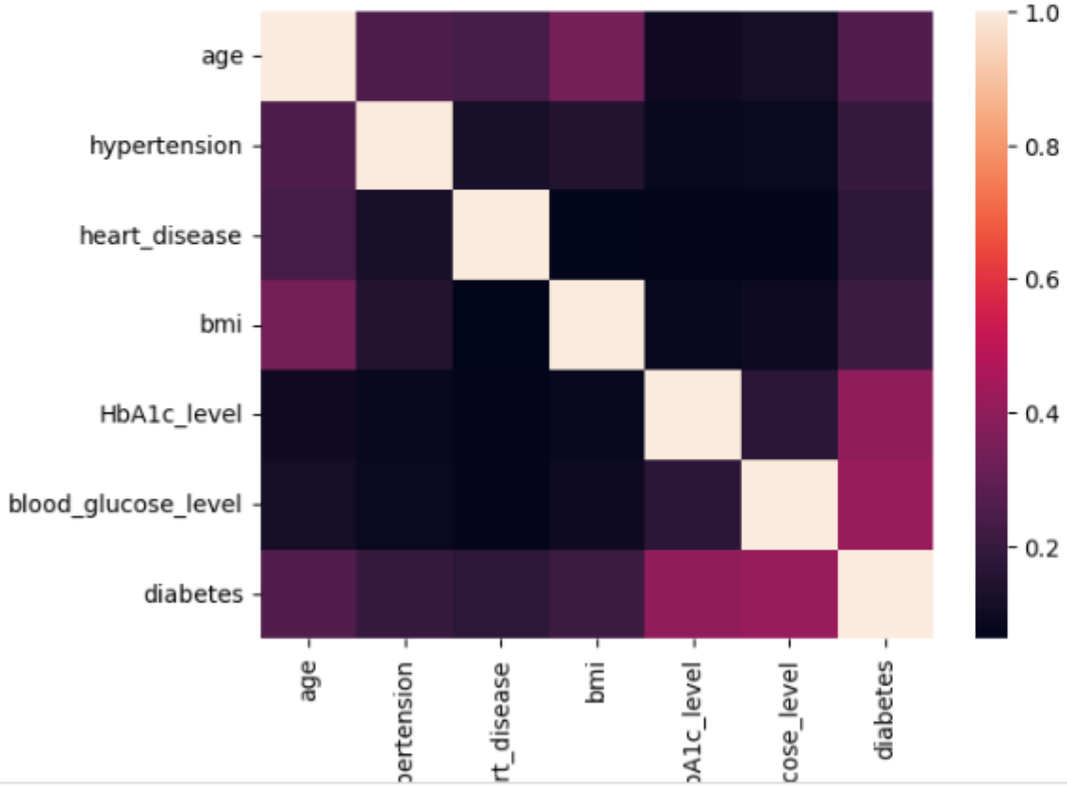
CORELATION

```
sns.heatmap(df.corr())
```

```
<ipython-input-40-aa4f4450a243>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid
  sns.heatmap(df.corr())
<Axes: >
```



```
<ipython-input-40-aa4f4450a243>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will defaul
  sns.heatmap(df.corr())
<Axes: >
```



IMPORTING LABEL ENCODER

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
columns=['gender','smoking_history']
for i in df[columns]:
  df[i]=encoder.fit_transform(df[i])
df
```

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	0	80.0	0	1	4	25.19	6.6	140	0
1	0	54.0	0	0	0	27.32	6.6	80	0
2	1	28.0	0	0	4	27.32	5.7	158	0
3	0	36.0	0	0	1	23.45	5.0	155	0
4	1	76.0	1	1	1	20.14	4.8	155	0
...	...	...	...	...	...	...	...	...	...
99995	0	80.0	0	0	0	27.32	6.2	90	0
99996	0	2.0	0	0	0	17.37	6.5	100	0

SPLITTING THE DATA INTO X AND Y VARIABLES

[42]
```
x=df.iloc[:,:-1].values
x
```

```
array([[  0.  ,  80.  ,   0.  , ...,  25.19,   6.6 , 140.  ],
       [  0.  ,  54.  ,   0.  , ...,  27.32,   6.6 ,  80.  ],
       [  1.  ,  28.  ,   0.  , ...,  27.32,   5.7 , 158.  ],
       ...,
       [  1.  ,  66.  ,   0.  , ...,  27.83,   5.7 , 155.  ],
       [  0.  ,  24.  ,   0.  , ...,  35.42,   4.  , 100.  ],
       [  0.  ,  57.  ,   0.  , ...,  22.43,   6.6 ,  90.  ]])
```

[43]
```
y=df.iloc[:,-1]
y
```

```
0        0
1        0
2        0
3        0
4        0
        ..
99995    0
99996    0
99997    0
```

IMPORTING TRAIN TEST SPLIT

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
x_train
```

```
array([[  1.  ,  49.  ,   0.  , ...,  27.32,   5.  , 155.  ],
       [  1.  ,  64.  ,   0.  , ...,  27.32,   3.5 , 145.  ],
       [  0.  ,  24.  ,   0.  , ...,  27.32,   3.5 , 130.  ],
       ...,
       [  1.  ,  42.  ,   0.  , ...,  26.14,   5.8 ,  85.  ],
       [  0.  ,  37.  ,   0.  , ...,  24.96,   6.2 , 158.  ],
       [  0.  ,  23.  ,   0.  , ...,  27.99,   5.  , 159.  ]])
```

```
x_test
```

```
array([[  0.  ,  13.  ,   0.  , ...,  20.82,   5.8 , 126.  ],
       [  0.  ,   3.  ,   0.  , ...,  21.  ,   5.  , 145.  ],
       [  1.  ,  63.  ,   0.  , ...,  25.32,   3.5 , 200.  ],
       ...,
       [  0.  ,  80.  ,   0.  , ...,  33.03,   6.6 , 100.  ],
       [  0.  ,  80.  ,   0.  , ...,  27.32,   6.2 , 200.  ],
       [  0.  ,  26.  ,   0.  , ...,  20.67,   3.5 , 100.  ]])
```

```
y_train
```

```
76513    0
60406    0
27322    0
53699    0
65412    0
        ..
6265     0
54886    0
76820    0
860      0
15795    0
Name: diabetes, Length: 70000, dtype: int64
```

```
[47] y_test
```

```
75721    0
80184    0
19864    0
76699    0
92991    1
        ..
42648    1
86306    0
45466    0
63724    0
34122    0
```

STANDARDISING WITH MINMAX SCALER

[48]
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
x_train=scaler.fit_transform(x_train)
x_test=scaler.fit_transform(x_test)
x_train
```

```
array([[0.5       , 0.61211211, 0.        , ..., 0.21158783, 0.27272727,
        0.34090909],
       [0.5       , 0.7997998 , 0.        , ..., 0.21158783, 0.        ,
        0.29545455],
       [0.        , 0.2992993 , 0.        , ..., 0.21158783, 0.        ,
        0.22727273],
       ...,
       [0.5       , 0.52452452, 0.        , ..., 0.19716416, 0.41818182,
        0.02272727],
       [0.        , 0.46196196, 0.        , ..., 0.1827405 , 0.49090909,
        0.35454545],
       [0.        , 0.28678679, 0.        , ..., 0.21977753, 0.27272727,
        0.35909091]])
```

MODEL CREATION

MODEL CREATION

[49]
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
knn_model=KNeighborsClassifier()
dt_model=DecisionTreeClassifier()
rf_model=RandomForestClassifier()
adb_model=AdaBoostClassifier()
full_model=[knn_model,dt_model,rf_model,adb_model]
```

PERFORMANCE EVALUATION (ACCURACY SCORE, CONFUSION MATRIX,CLASSIFICATION REPORT)

[50]
```
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
for i in full_model:
  print(i)
  i.fit(x_train,y_train)
  y_pred=i.predict(x_test)
  result=confusion_matrix(y_test,y_pred)
  print(result)
```

PERFORMANCE EVALUATION (ACCURACY SCORE, CONFUSION MATRIX,CLASSIFICATION REPORT)

```
[50] from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
     for i in full_model:
       print(i)
       i.fit(x_train,y_train)
       y_pred=i.predict(x_test)
       result=confusion_matrix(y_test,y_pred)
       print(result)
       print("*************************************************")
       print(accuracy_score(y_test,y_pred))
       print("*************************************************")
```

```
KNeighborsClassifier()
[[27319   134]
 [ 1010  1537]]
*************************************************
0.9618666666666666
*************************************************
DecisionTreeClassifier()
[[26750   703]
 [  672  1875]]
*************************************************
0.9541666666666667
*************************************************
```

```
KNeighborsClassifier()
[[27319   134]
 [ 1010  1537]]
**************************************************
0.9618666666666666
**************************************************
DecisionTreeClassifier()
[[26750   703]
 [  672  1875]]
**************************************************
0.9541666666666667
**************************************************
RandomForestClassifier()
[[27408    45]
 [  825  1722]]
**************************************************
0.971
**************************************************
AdaBoostClassifier()
[[27422    31]
 [  821  1726]]
**************************************************
0.9716
**************************************************
```

LINEAR REGRESSION MODEL

```
[50] 0.971
**********
AdaBoostClassifier()
[[27422    31]
 [  821  1726]]
**********
0.9716
**********
```

LINEAR REGRESSION MODEL

```
[51] from sklearn.linear_model import LinearRegression
     lr_model=LinearRegression()
     lr_model.fit(x_train,y_train)
     y_pred=lr_model.predict(x_test)
     y_pred
```

```
array([-0.01212545, -0.0475788 ,  0.07669402, ...,  0.2618634 ,
        0.30997328, -0.23999124])
```