

```
import pandas as pd
from sklearn.metrics import classification_report
ds=pd.read_csv("/content/diabetes.csv")
ds.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diab
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	

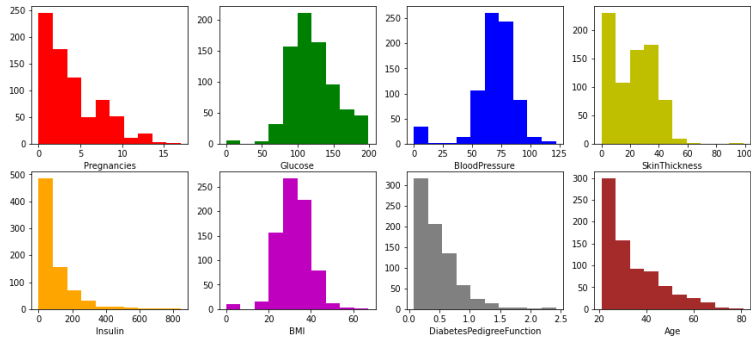
```
ds.isnull().sum()

Pregnancies      0
Glucose           0
BloodPressure     0
SkinThickness     0
Insulin           0
BMI              0
DiabetesPedigreeFunction  0
Age              0
Outcome          0
dtype: int64
```

```
ds.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	
count	768.000000	768.000000	768.000000	768.000000	768.000000	7
mean	3.845052	120.894531	69.105469	20.536458	79.799479	
std	3.369578	31.972618	19.355807	15.952218	115.244002	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	

```
import matplotlib.pyplot as plt
color=['r','g','b','y','orange','m','grey','brown']
plt.figure(figsize=(15,10))
for i in range(len(ds.keys())[:-1]):
    plt.subplot(3,4,i+1)
    plt.hist(ds[ds.keys()[i]],color=color[i])
    plt.xlabel(ds.keys()[i])
```



```
X=ds.drop('Outcome',axis=1)
Y=ds['Outcome']
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

```
scaler=StandardScaler()
X=scaler.fit_transform(X)
```

```
X_train,X_test,y_train,y_test=train_test_split(X,Y,random_state=10)
```

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
```

```
score=cross_val_score(DecisionTreeClassifier(),X,Y,cv=5)
score.mean()
```

```
0.6967490026313555
```

```
from sklearn.ensemble import BaggingClassifier
```

```
bagging=BaggingClassifier(
    base_estimator=DecisionTreeClassifier(),
    n_estimators=100,
    max_samples=0.8,
    oob_score=True,
    random_state=0
)
```

```
bagging.fit(X_train,y_train)
```

```
bagging.oob_score_
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/ensemble/_base.py:166: FutureWarning: `base_estimator` was renamed to `estimator` in
warnings.warn(
0.7604166666666666
```

```
decision_acc_bagg=bagging.score(X_test,y_test)
decision_acc_bagg
```

```
0.7760416666666666
```

```
from sklearn.linear_model import LogisticRegression
```

```
bagging=BaggingClassifier(
    base_estimator=LogisticRegression(),
    n_estimators=100,
    max_samples=0.8,
    oob_score=True,
    random_state=0
)
```

```
bagging.fit(X_train,y_train)
```

```
logistic_bagg=bagging.oob_score_
logistic_bagg
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/ensemble/_base.py:166: FutureWarning: `base_estimator` was renamed to `estimator` in
warnings.warn(
0.7725694444444444
```

```
from sklearn.metrics import classification_report, roc_curve
y_pred=bagging.predict(X_test)
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.74	0.88	0.81	121
1	0.71	0.48	0.57	71
accuracy			0.73	192
macro avg	0.73	0.68	0.69	192
weighted avg	0.73	0.73	0.72	192

```
from sklearn.svm import SVC
```

```
bagging=BaggingClassifier(
    base_estimator=SVC(),
    n_estimators=100.
```

```

max_samples=0.8,
oob_score=True,
random_state=0
)

bagging.fit(X_train,y_train)

svc_bagg=bagging.oob_score_
svc_bagg

/usr/local/lib/python3.9/dist-packages/sklearn/ensemble/_base.py:166: FutureWarning: `base_estimator` was renamed to `estimator` in
warnings.warn(
0.7829861111111112

```

```
from sklearn.ensemble import AdaBoostClassifier
```

```

boosting=AdaBoostClassifier(
    base_estimator=DecisionTreeClassifier(),
    n_estimators=10,
    random_state=0
)

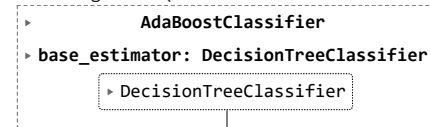
```

```
boosting.fit(X_train,y_train)
```

```

/usr/local/lib/python3.9/dist-packages/sklearn/ensemble/_base.py:166: Futu
warnings.warn(

```



```

decision_boost=boosting.score(X_test,y_test)
decision_boost

```

```
0.6979166666666666
```

```

boosting=AdaBoostClassifier(
    base_estimator=LogisticRegression(),
    n_estimators=10,
    random_state=0
)

```

```
boosting.fit(X_train,y_train)
```

```
logistic_boost=boosting.score(X_test,y_test)
```

```

/usr/local/lib/python3.9/dist-packages/sklearn/ensemble/_base.py:166: FutureWarning: `base_estimator` was renamed to `estimator` in
warnings.warn(

```

```

y_pred=boosting.predict(X_test)
print(classification_report(y_test,y_pred))

```

	precision	recall	f1-score	support
0	0.73	0.90	0.81	121
1	0.72	0.44	0.54	71
accuracy			0.73	192
macro avg	0.73	0.67	0.68	192
weighted avg	0.73	0.73	0.71	192

```
import matplotlib.pyplot as plt
```

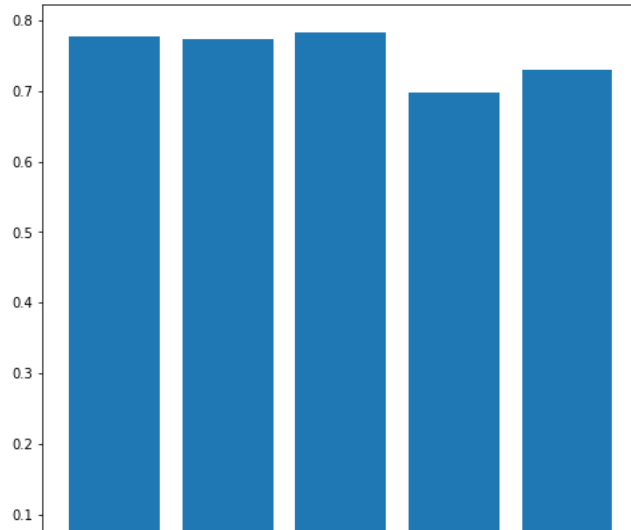
```
plt.figure(figsize=(8,8))
```

```

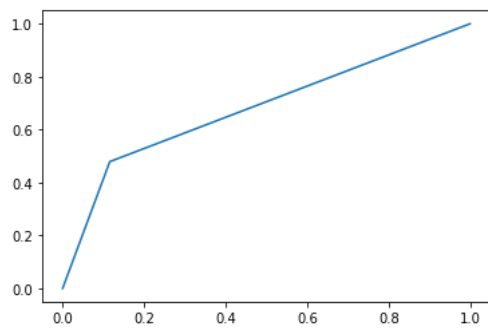
labels=['decision_bagg','logistic_bagg','svc_bagg','decision_boost','logistic_boost']
val=[decision_acc_bagg,logistic_bagg,svc_bagg,decision_boost,logistic_boost]
plt.bar(labels,val)

```

<BarContainer object of 5 artists>



```
fpr,tpr,_=roc_curve(y_test,y_pred)
plt.plot(fpr,tpr,label="roc")
plt.show()
```



[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 19:33

