my-project

April 1, 2024

1.CHOOSING A DATASET : I'VE CHOOSEN DIABETES PREDICTION DATASET WHICH FALLS UNDER THE CATEGORY OF MEDICAL DIAGONASIS ———————————————————————————— Title: Predicting Diabetes Risk Using Glucose and Blood Pressure: A Comparative Analysis of Decision Tree, Logistic Regression, and Naive Bayes

Introduction: This project aims to predict diabetes risk utilizing glucose and blood pressure as inputs, employing Decision Tree, Logistic Regression, and Naive Bayes algorithms. By assessing the performance of these methods on a standardized dataset, I aim to determine the most effective approach for accurate and timely diabetes risk prediction, facilitating proactive healthcare interventions and personalized patient care strategies.i've took this dataset from kaggle, let's see which algorithm accuries highest accuracy rate.

[290]:
```python
#2.IMPORTING ALL NECESSARY LIBRARIES
#---------------------------------

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix,␣
 ↪classification_report
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB,MultinomialNB
from sklearn.linear_model import LogisticRegression
```

[291]:
```python
#3.LOADING THE DATASET USING PANDAS MODULE
#----------------------------------------

data=pd.read_csv(r"C:\Users\gugan\Desktop\machine learning\GLUCOSE\GLUCOSE␣
 ↪LEVEL.csv")
data[:5]
```

[291]:

	glucose	bloodpressure	diabetes
0	40	85	0
1	40	92	0

```
2        45              63          1
3        45              80          0
4        40              73          1
```

[292]:
```
data.info()
print('')
data.shape
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 995 entries, 0 to 994
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   glucose        995 non-null    int64
 1   bloodpressure  995 non-null    int64
 2   diabetes       995 non-null    int64
dtypes: int64(3)
memory usage: 23.4 KB
```

[292]:
```
(995, 3)
```

[293]:
```
#4.FEATURE SELECTION (X,y) AND SCALING DATA (STANDARD SCALAR)
#-------------------------------------------------------------

X= data.iloc[:,0:2].values
y= data.iloc[:,2].values
```

[294]:
```
print(X.shape)
print(y.shape)

X[:5]

#y[:5]
```

```
(995, 2)
(995,)
```

[294]:
```
array([[40, 85],
       [40, 92],
       [45, 63],
       [45, 80],
       [40, 73]], dtype=int64)
```

[295]:
```
#DATA SPLITTING
#--------------
```

```python
Xtrain,Xtest,ytrain,ytest = train_test_split(X,y,test_size=0.20,random_state=2)
```

[296]:
```python
print('TARINING INPUT SAMPLES COUNT ==>',Xtrain.shape)
print('TRAINING OUTPUT SAMPLES COUNT ==>',ytrain.shape)
print('TESTING INPUT SAMPLE COUNT ==>',Xtest.shape)
print('TESTING OUTPUT SAMPLE COUNT ==>',ytest.shape)
```

```
TARINING INPUT SAMPLES COUNT ==> (796, 2)
TRAINING OUTPUT SAMPLES COUNT ==> (796,)
TESTING INPUT SAMPLE COUNT ==> (199, 2)
TESTING OUTPUT SAMPLE COUNT ==> (199,)
```

[297]:
```python
#IMPLEMENTING THE ALGORITHM
```

1 DIABETES PREDICTION USING DECISION TREE

[298]:
```python
#5.model creation by invoking the algorithm
#-----------------------------------------


dtree= DecisionTreeClassifier(max_depth=3,criterion='gini',random_state=3)
```

[299]:
```python
#6.model training by fitting the X and y data(X_train and y_train)
#-----------------------------------------------------------------

dtree.fit(Xtrain,ytrain)
```

[299]: DecisionTreeClassifier(max_depth=3, random_state=3)

[300]:
```python
#7.model prediction (ypre) -'using x_test'
#----------------------------------------

ypre = dtree.predict(Xtest)
```

[301]:
```python
#8.calculate performance accuracy using output matrix
#----------------------------------------------------

accuracy_score(ytest,ypre)
```

[301]: 0.9246231155778895

[302]:
```python
entro= DecisionTreeClassifier(max_depth=3,criterion='entropy',random_state=1)
```

[303]:
```python
entro.fit(Xtrain,ytrain)
```

[303]: DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=1)

[304]:
```
ypre_ent= entro.predict(Xtest)
```

[305]:
```
accuracy_score(ytest,ypre_ent)
```

[305]: 0.9195979899497487

2 DIABETES PREDICTION USING LOGISTIC REGRESSION

[306]:
```
logreg = LogisticRegression()
```

[307]:
```
logreg.fit(Xtrain,ytrain)
```

[307]: LogisticRegression()

[308]:
```
ypre_log = logreg.predict(Xtest)
```

[309]:
```
accuracy_score(ytest,ypre_log)
```

[309]: 0.9296482412060302

3 DIABETES PREDICTION USING NAIVE BAYES CLASSIFIER

[310]:
```
gu = GaussianNB()
```

[311]:
```
gu.fit(Xtrain,ytrain)
```

[311]: GaussianNB()

[312]:
```
test_gpred=gu.predict(Xtest)
```

[313]:
```
accuracy_score(ytest,test_gpred)
```

[313]: 0.9396984924623115

[314]:
```
d=MultinomialNB()
```

[315]:
```
d.fit(Xtrain,ytrain)
```

[315]: MultinomialNB()

[316]:
```
test_mulpred = d.predict(Xtest)
```

[317]:
```
accuracy_score(ytest,test_mulpred)
```

[317]: 0.7336683417085427

[318]:
```
compare=pd.DataFrame({'actual output':ytest,'gini_dt':ypre,'entro_dt':
↪ypre_ent,'logreg':ypre_log ,'GaussianNB':test_gpred,'MultinomialNB':
↪test_mulpred})
```

[319]:
```
compare
```

[319]:
```
     actual output  gini_dt  entro_dt  logreg  GaussianNB  MultinomialNB
0                1        1         1       1           1              1
1                0        0         0       0           0              1
2                0        0         0       0           0              0
3                1        0         0       0           0              0
4                0        0         0       0           0              1
..             …      …       …     …         …           …
194              0        0         0       0           0              1
195              1        1         1       1           1              1
196              1        1         1       1           1              1
197              1        1         1       1           1              1
198              0        0         0       0           0              0

[199 rows x 6 columns]
```

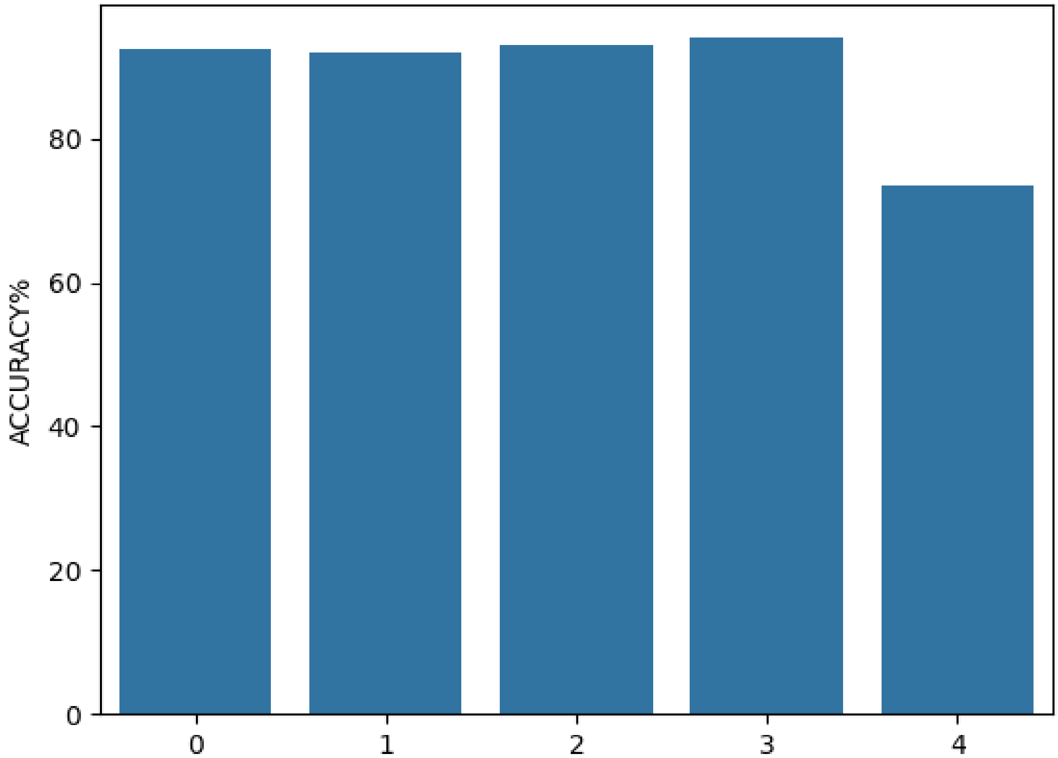
[320]:
```
report=pd.DataFrame({'MODEL':
↪['giniDT','entropyDT','logreg','guassNB','multinoNB'],'ACCURACY%':
↪[accuracy_score(ytest,ypre)*100,accuracy_score(ytest,ypre_ent)*100,accuracy_score(ytest,ypr
```

[321]:
```
report
```

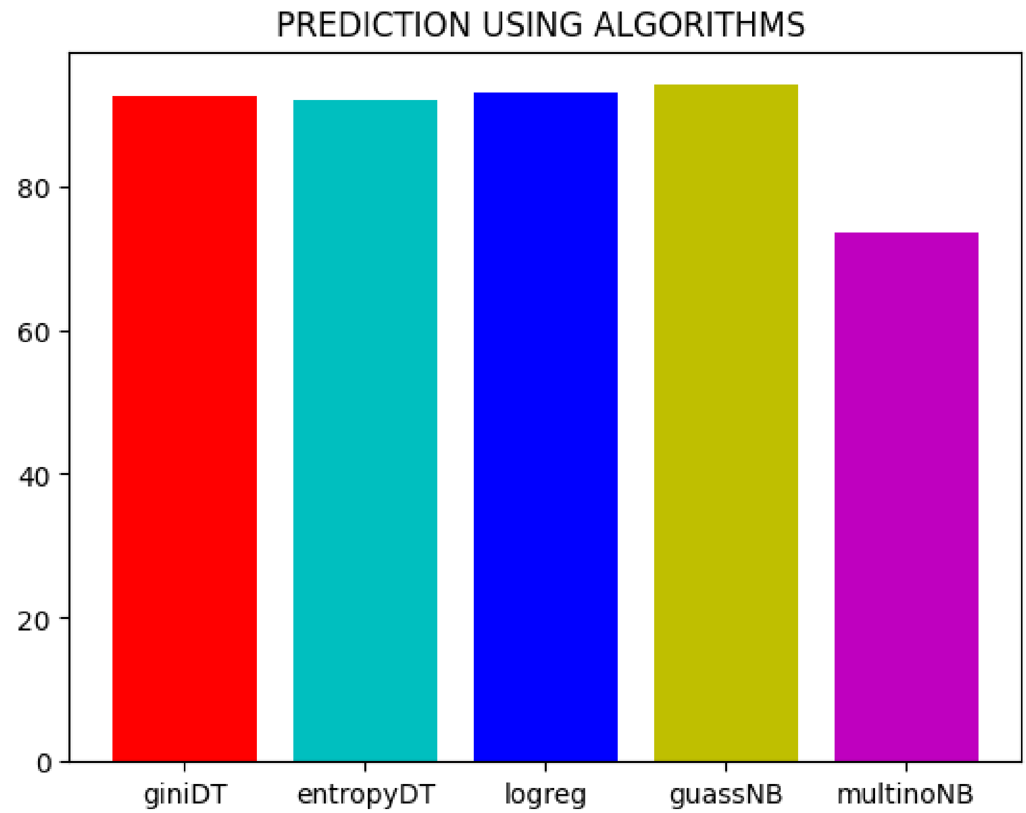
[321]:
```
       MODEL  ACCURACY%
0     giniDT  92.462312
1  entropyDT  91.959799
2     logreg  92.964824
3    guassNB  93.969849
4  multinoNB  73.366834
```

[323]:
```
sns.barplot(report['ACCURACY%'])
```

[323]: <Axes: ylabel='ACCURACY%'>



[330]:
```
count = [92.462312,91.959799,92.964824,93.969849,73.366834]
color_code = ['r','c','b','y','m']
plt.bar(['giniDT','entropyDT','logreg','guassNB','multinoNB'],count,color =␣
 ↪color_code)
plt.title('PREDICTION USING ALGORITHMS ')
plt.show()
```



Conclusion: In this study, we investigated the predictive capability of Decision Tree, Logistic Regression, and Naive Bayes algorithms in assessing diabetes risk based on glucose and blood pressure levels. Through rigorous evaluation, it has been demonstrated that the Naive Bayes classifier outperforms the other methods, achieving an impressive accuracy rating of 93