Assignment_5

August 21, 2024

[1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
```

[2]:

```
df = pd.read_csv('diabetes.csv')
df.head()
```

[2]:

```
   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

   Pedigree  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
```

[3]:

```
df.isnull().sum()
```

[3]:

```
Pregnancies      0
Glucose          0
BloodPressure    0
SkinThickness    0
Insulin          0
BMI              0
Pedigree         0
Age              0
Outcome          0
```

```
dtype: int64
```

[4]:
```
x = df.drop('Outcome', axis=1)
y = df['Outcome']
```

[5]:
```
scaler=MinMaxScaler()
x=scaler.fit_transform(x)
```

[6]:
```
x_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.
 ↪2,random_state=42)
from sklearn.neighbors import KNeighborsClassifier
clf = KNeighborsClassifier(n_neighbors=3)
```

[7]:
```
KNN=clf.fit(x_train,y_train)
prediction=KNN.predict(X_test)
```

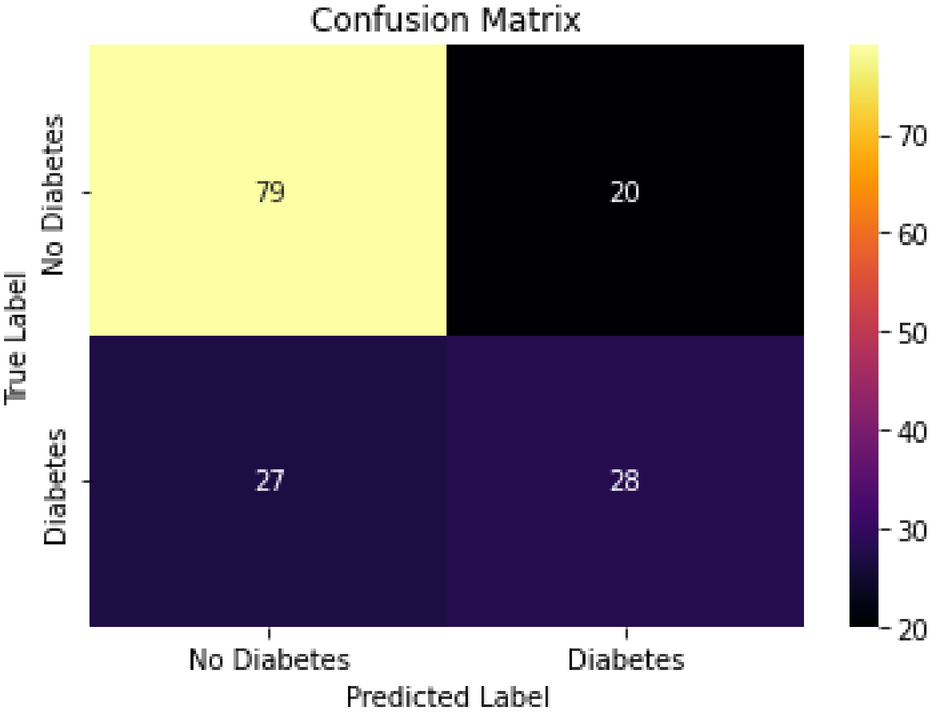
[8]:
```
accuracy = metrics.accuracy_score(y_test, prediction)
error_rate = 1 - accuracy
confusion_matrix = metrics.confusion_matrix(y_test, prediction)
classification_report = metrics.classification_report(y_test, prediction,␣
 ↪target_names=['No Diabetes', 'Diabetes'])
```

[9]:
```
print("Confusion Matrix:")
print(confusion_matrix)
print("\nClassification Report:")
print(classification_report)
print("\nAccuracy:", accuracy)
print("Error Rate:", error_rate)
```

```
Confusion Matrix:
[[79 20]
 [27 28]]

Classification Report:
              precision    recall  f1-score   support

 No Diabetes       0.75      0.80      0.77        99
    Diabetes       0.58      0.51      0.54        55

    accuracy                           0.69       154
   macro avg       0.66      0.65      0.66       154
weighted avg       0.69      0.69      0.69       154


Accuracy: 0.6948051948051948
Error Rate: 0.30519480519480524
```

[10]:

```
sns.heatmap(confusion_matrix, annot=True, fmt='d', cmap='inferno',␣
 ↪xticklabels=['No Diabetes', 'Diabetes'], yticklabels=['No Diabetes',␣
 ↪'Diabetes'])
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```


Confusion Matrix
True Label
No Diabetes
Diabetes
No Diabetes
Diabetes
Predicted Label