

<https://www.kaggle.com/datasets/abdallamahgoub/diabetes>

```
'''Implement K-Nearest neighbour algorithm on diabetes.csv dataset. compute
confusion matrix, accuracy, error rate, precision and recall on the given
dataset.'''
```

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn import metrics

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

df.columns

df.isnull().sum()

# outcome is the label/target, other columns are features
x = df.drop('Outcome', axis = 1)
y = df['Outcome']

from sklearn.preprocessing import scale
x = scale(x)
# split into train and test
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3,
random_state = 42)

from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(x_train, y_train)
y_pred = knn.predict(x_test)

print("confusion matrix: ")
cs = metrics.confusion_matrix(y_test,y_pred)
print(cs)

print("accuracy ",metrics.accuracy_score(y_test,y_pred))

'''classification error rate: proportion of instances misclassified over the
whole set of instances. Error rate is calculated
as the total no. of two incorrect predictions (FN + FP) divided by the total
number of a dataset (examples in the dataset. Also
error_rate = 1 - accuracy )'''

total_misclassified = cs[0,1] + cs[1,0]
print(total_misclassified)
total_examples = cs[0,0]+cs[0,1]+cs[1,0]+cs[1,1]
print(total_examples)
```

```
print("Error rate",total_misclassified/total_examples)
print("Error rate ",1-metrics.accuracy_score(y_test,y_pred))

print("Precision score",metrics.precision_score(y_test,y_pred))

print("Recall score ",metrics.recall_score(y_test,y_pred))

print("classification report ",metrics.classification_report(y_test,y_pred))
```