```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


dfx = pd.read_csv("/content/Diabetes_XTrain.csv")
dfy = pd.read_csv("/content/Diabetes_YTrain.csv")
print(dfx.shape)
print(dfy.shape)
print(dfx.head())
print(dfy.head())
print(dfx.columns)
```

```
(576, 8)
(576, 1)
   Pregnancies  Glucose  BloodPressure  ...   BMI  DiabetesPedigreeFunction  A
0            7      168             88  ...  38.2                     0.787
1            8      110             76  ...  27.8                     0.237
2            7      147             76  ...  39.4                     0.257
3            2      100             66  ...  32.9                     0.867
4            4      129             86  ...  35.1                     0.231

[5 rows x 8 columns]
   Outcome
0        1
1        0
2        1
3        1
4        0
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age'],
      dtype='object')
```

```
X = dfx.values
Y = dfy.values.reshape(-1,)
print(X.shape)
print(Y.shape)
```

```
(576, 8)
(576,)
```

Step 2:---> spliting the dataset into training and testing

```
split = int(0.80*X.shape[0])
x_tr,y_tr = X[:split,:],Y[:split]
x_te,y_te = X[split:,:],Y[split:]


print(x_tr.shape,y_tr.shape)
print(x_te.shape,y_te.shape)
```

```
(460, 8) (460,)
(116, 8) (116,)
```

Step 3:---> Knn algorithm and it's implementation

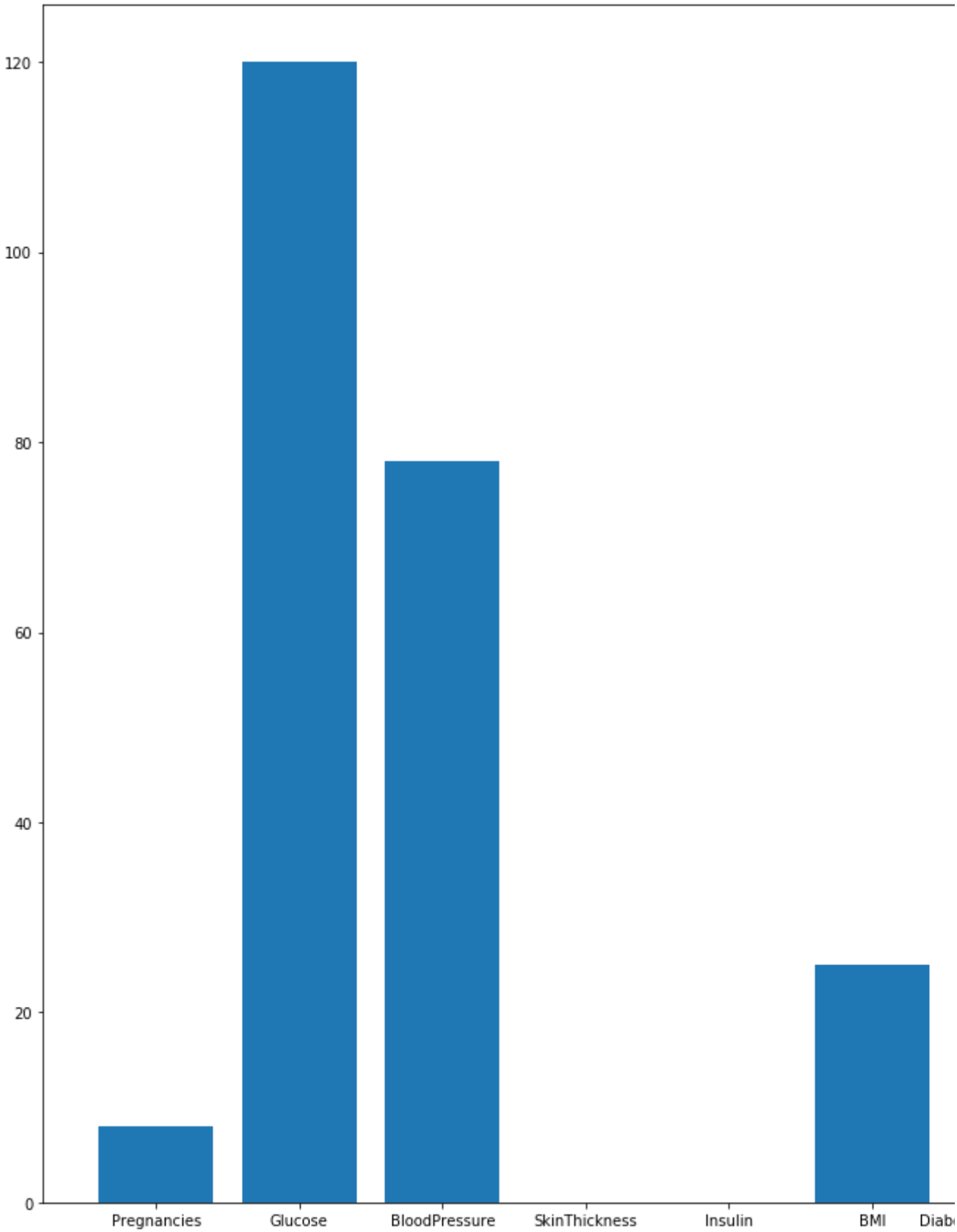
```
def dist(x1,x2):
  return np.sqrt(sum((x1-x2)**2))


def knn(x_tr,y_tr,q_pt,k=5):
  vals = []
  m = x_tr.shape[0]
  d = 0.0
  for i in range(m):
    d = dist(x_tr[i],q_pt)
    vals.append((d,y_tr[i]))
  vals = sorted(vals)
  vals = vals[:k]
  print(vals)
  vals = np.array(vals)
  new_vals = np.unique(vals[:,1],return_counts=True)
  print(new_vals)
  index = new_vals[1].argmax()
  return new_vals[0][index]



pred = knn(x_tr,y_tr,x_te[5])
org = y_te[5]
if pred==org:
  print("correct prediction:--")
  print(pred)
else:
  print("wrong prediction:---")
```

```
[(10.493913712242922, 0), (11.753542487267403, 0), (12.160163814686051, 0), (1
(array([0., 1.]), array([3, 2]))
correct prediction:--
0.0
```

```
print(x_tr[0].shape)
print(y_tr[0])
print(type(x_tr))
print(type(y_tr))
```

```
(8,)
1
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
```

```
plt.figure(figsize=(15,15))
plt.bar(dfx.columns,x_te[5],label=pred)
plt.legend()
plt.show()
```


120
100
80
60
40
20
0
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
Diab