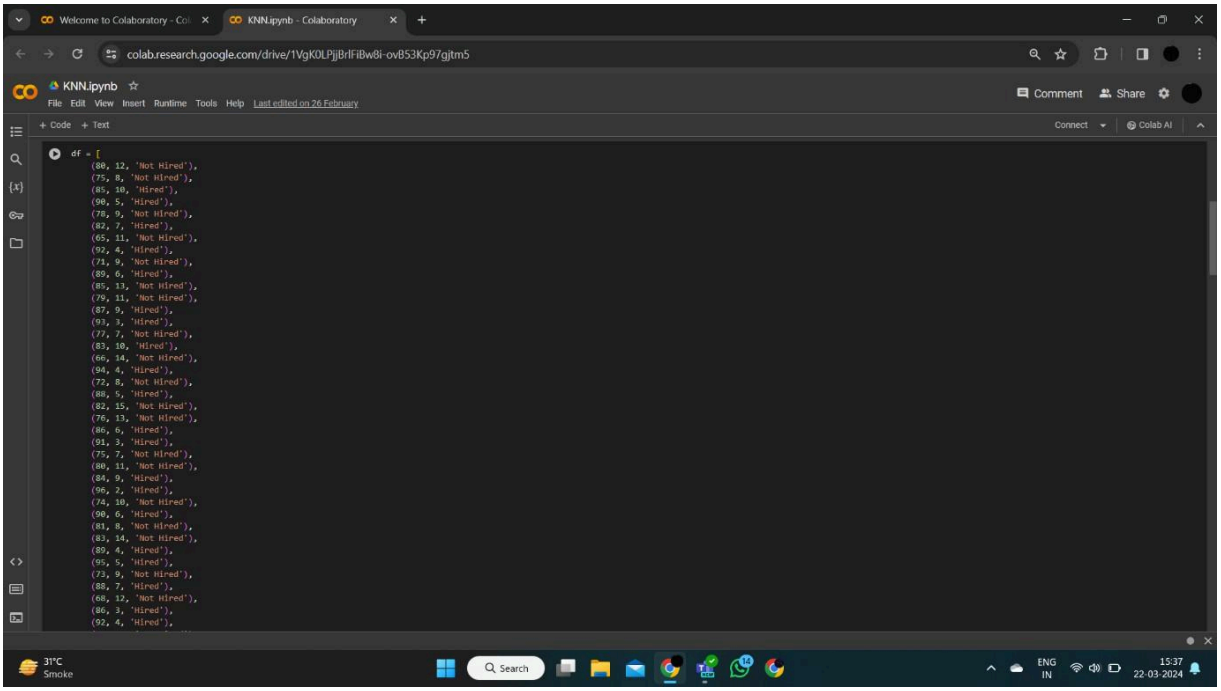
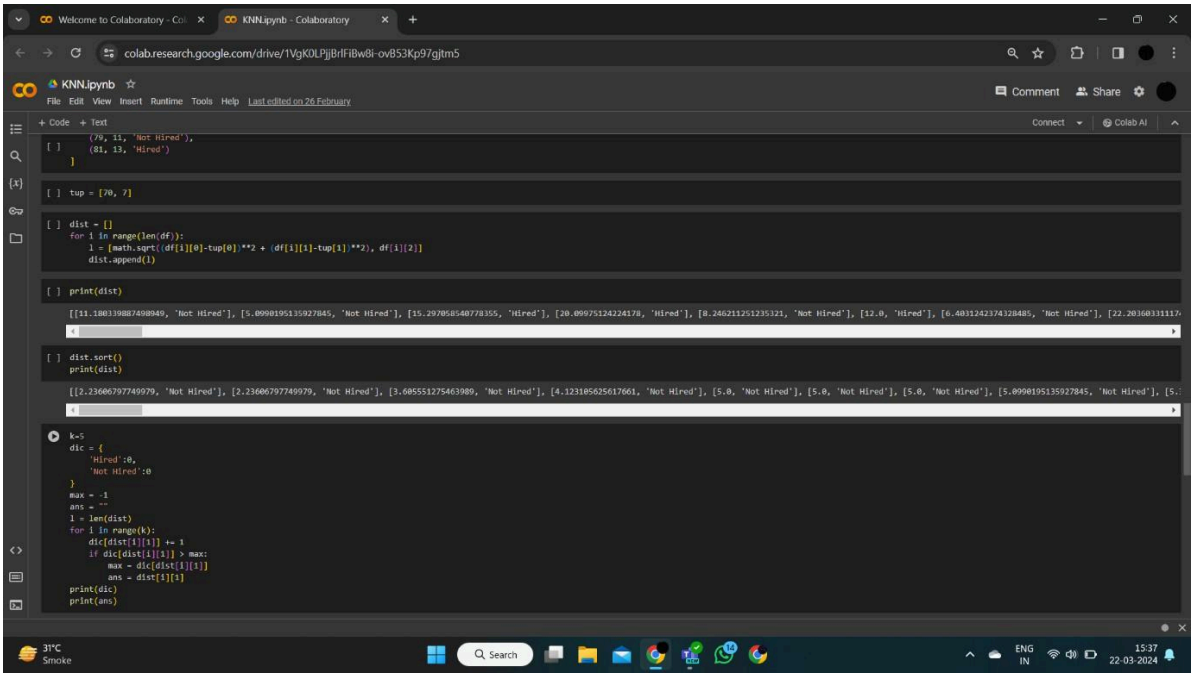
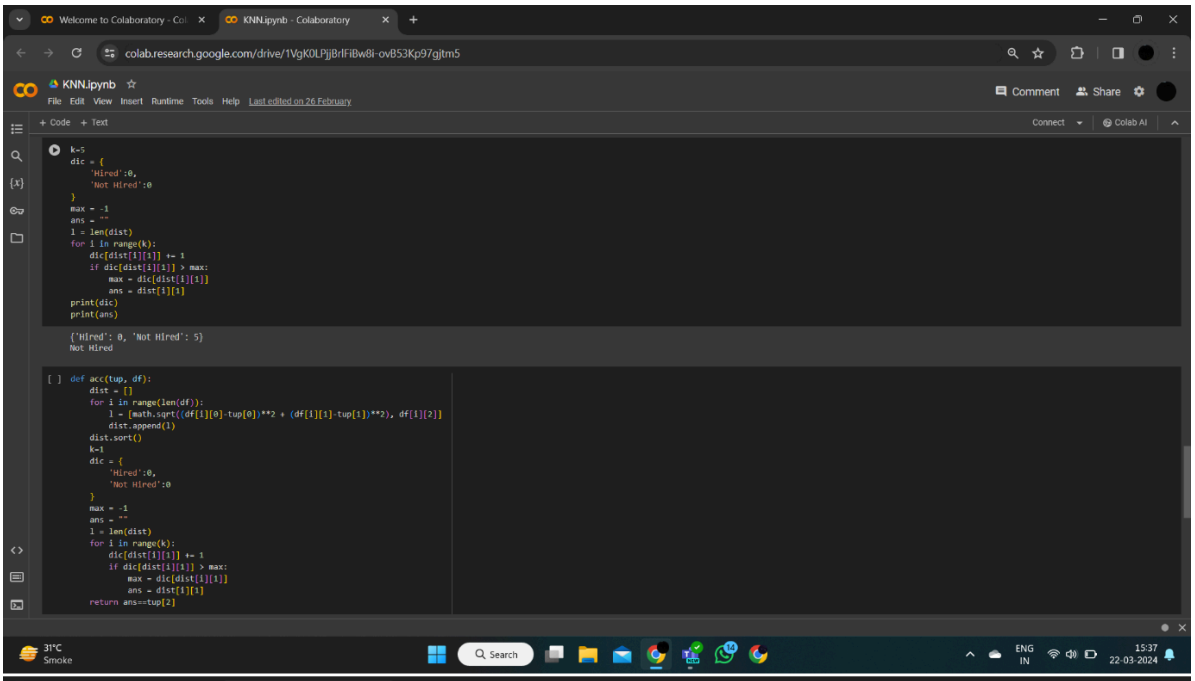
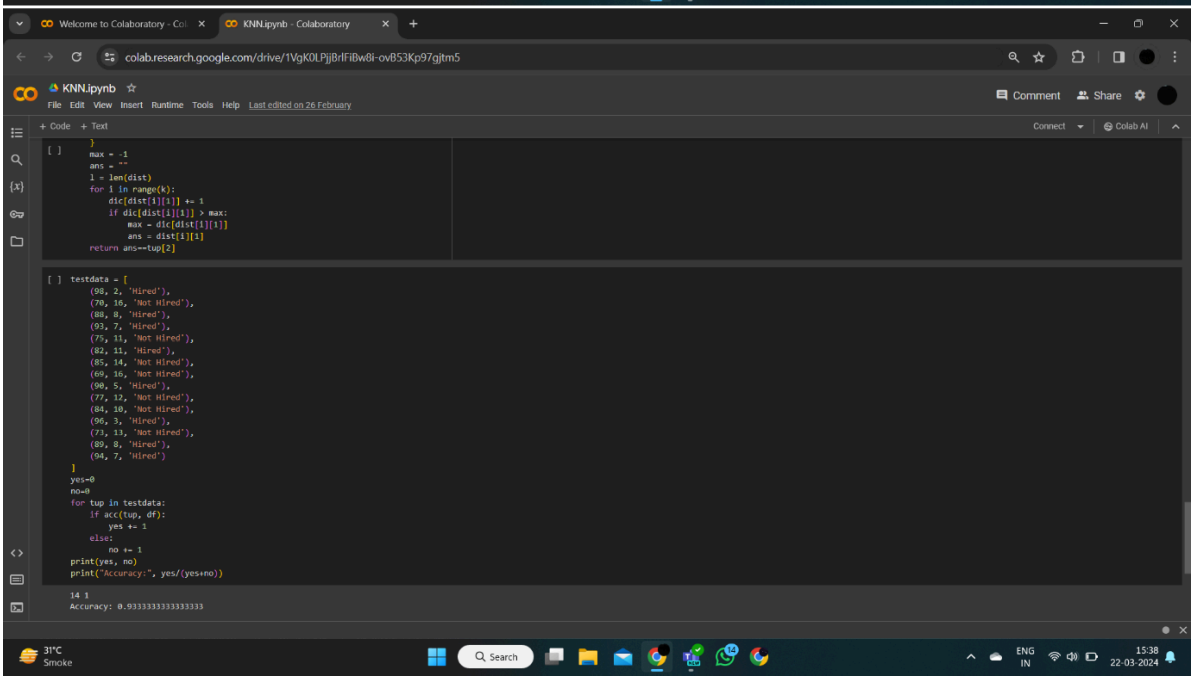
ML Experiment 4






Welcome to Colaboratory - Col
KNN.ipynb - Colaboratory
colab.research.google.com/drive/1VgK0LPjjBrlFiBw8i-ovB53Kp97gjtm5
KNN.ipynb
File Edit View Insert Runtime Tools Help Last edited on 26 February
Comment
Share
+ Code + Text
Connect
Colab AI
k=5
dic = {
'Hired':0,
'Not Hired':0
}
max = -1
ans = ""
l = len(dist)
for i in range(k):
dic[dist[i][1]] += 1
if dic[dist[i][1]] > max:
max = dic[dist[i][1]]
ans = dist[i][1]
print(dic)
print(ans)
{'Hired': 0, 'Not Hired': 5}
Not Hired
def acc(tup, df):
dist = []
for i in range(len(df)):
l = [math.sqrt((df[i][0]-tup[0])**2 + (df[i][1]-tup[1])**2), df[i][2]]
dist.append(l)
dist.sort()
k=1
dic = {
'Hired':0,
'Not Hired':0
}
max = -1
ans = ""
l = len(dist)
for i in range(k):
dic[dist[i][1]] += 1
if dic[dist[i][1]] > max:
max = dic[dist[i][1]]
ans = dist[i][1]
return ans==tup[2]
31°C
Smoke
Search
ENG
IN
15:37
22-03-2024


Welcome to Colaboratory - Col
KNN.ipynb - Colaboratory
colab.research.google.com/drive/1VgK0LPjjBrlFiBw8i-ovB53Kp97gjtm5
KNN.ipynb
File Edit View Insert Runtime Tools Help Last edited on 26 February
Comment
Share
+ Code + Text
Connect
Colab AI
}
max = -1
ans = ""
l = len(dist)
for i in range(k):
dic[dist[i][1]] += 1
if dic[dist[i][1]] > max:
max = dic[dist[i][1]]
ans = dist[i][1]
return ans==tup[2]
testdata = [
(90, 2, 'Hired'),
(70, 16, 'Not Hired'),
(88, 8, 'Hired'),
(93, 7, 'Hired'),
(75, 11, 'Not Hired'),
(82, 11, 'Hired'),
(85, 14, 'Not Hired'),
(69, 16, 'Not Hired'),
(90, 5, 'Hired'),
(77, 12, 'Not Hired'),
(84, 10, 'Not Hired'),
(96, 3, 'Hired'),
(73, 13, 'Not Hired'),
(89, 8, 'Hired'),
(94, 7, 'Hired')
]
yes=0
no=0
for tup in testdata:
if acc(tup, df):
yes += 1
else:
no += 1
print(yes, no)
print("Accuracy:", yes/(yes+no))
14 1
Accuracy: 0.9333333333333333
31°C
Smoke
Search
ENG
IN
15:38
22-03-2024

ML Experiment 4

DATE:

Kreena Shah
60004210243
0179
C'32

Aim : To implement Principal Component Analysis (PCA)

Theory :

PCA works on the condition that while the data in a higher dimensional space is mapped to data in lower dimension space, the variance of the data in the lower dimensional space should be maximum.

PCA is a statistical procedure that uses an orthogonal transformation that converts a set of correlated variables to a set of uncorrelated variables.

It is an unsupervised learning algorithm used to examine the iterations among a set of variables.

Steps

(I) Standardization

$$z = \frac{x - \mu}{\sigma}$$

(II) Covariance Matrix Computation

$$cov(x1, x2) = \sum_{i=1}^{n} \frac{(x1_i - \hat{x1})(x2_i - \overline{x2})}{n-1}$$

DATE:

(III) Compute Eigen values & eigen vectors of covariance matrix to identify principal components

$$AX = \lambda X$$
$$AX - \lambda X = 0$$
$$(A - \lambda I)X = 0$$

Conclusion : Thus, we implemented PCA.