CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (AUTONOMOUS), HYD – 500 075
(Autonomous)
I / II MID SESSIONAL EXAMINATION
MAIN ANSWER BOOK

Class& Branch BE-VI sem H1 (IT1) Last Page No.: 8

Subject: Artificial Intelligence Lab Internal Roll no. 160118737001

(1) Implementation of MINIMAX game search.

(2) Apply a linear Regression method to fit the training data & then predict the output using the test dataset using diabetes dataset

⇒ (1) MiniMax Game Search

⇒ It is a backtracking algorithm.

⇒ It is used in game theory for decision making to find the optimal move of a player. Assume that your opponent also makes optimal moves.

⇒ It is used in games such as tic-tac-toe, chess etc.

⇒ In MiniMax there are 2 players – Maximizer & Minimizer.

→ The Maximizer tries to get highest score while the Minimizer tries to get lowest score.

→ Since it is a backtracking algorithm, it tries all moves & then backtracks & makes a decision

Program :

```
import math.

def minimax ( cDepth , nIndex , maxTurn , points , tDepth):
    #base case : tDepth reached.
    if (cDepth == tDepth) :
        return points [nIndex]
    #If it is turn of maximizer
    if (maxTurn ):
        return max (minimax (cDepth+1 , nIndex * 2 ,
                             False, points , tDepth),
                    minimax (cDepth +1, nIndex * 2 +1,
                             False , points, tDepth ))
    #If it is turn of minimizer
    else :
        return min (minimax (cDepth+1, nIndex *2,
                             True , points, tDepth),
                    minimax ( cDepth +1 , nIndex * 2 +1,
                             True , points , tDepth))

points = [ 3, 5, 2, 9, 12, 5, 23 , 23 ]

tree Depth = math. log (len (scores), 2 )

print (" The optimal value is : " , end = "")
print ( minimax (0, 0, True , points , tree Depth) )
```

[2]

Output :

The optimal value is 12

```
The optimal value is : 12
```

(2) Linear Regression for diabetes dataset:

⇒ Linear Regression comes under Regression which is a type of Supervised Machine Learning.

⇒ Supervised Machine learning is a type of machine learning that employs the use of labelled datasets to predict the future/outcome values.

⇒ Regression is about predicting a quantity. It predicts a continuous quantity output for a dataset.

⇒ Linear regression quantifies the relationship between one or more predictor variables & one outcome variable.

⇒ It is a Linear model — a model that assumes a linear relationship between the input variables (x) & single output variable (y)

⇒ It establishes the relationship between dependent variable (y) & independent variable (x) using a best fit line (regression line).

Program

```
from sklearn import datasets, linear_model
import matplotlib.pyplot as plt
import numpy as np

# Load dataset
diabetes = datasets.load_diabetes()
# using one feature for training
diabetes_X = diabetes.data[:, np.newaxis, 2]

# Split data into training & testing datasets
diabetes_X_train = diabetes_X[:-20]

diabetes_X_test = diabetes_X[-20:]

# Split targets into training & testing datasets
diabetes_y_train = diabetes.target[:-20]
diabetes_y_test = diabetes.target[-20:]

# Creating object for Linear Regression.
reg = linear_model.LinearRegression()
# Train model using training datasets
reg.fit(diabetes_X_train, diabetes_y_train)
# Input data.
print("Input values")

print(diabetes_X_test)
# Making prediction using testing set
diabetes_y_pred = reg.predict(diabetes_X_test)
```

```
# Predicted Data
print ("Predicted output values")
print (diabetes_y_pred)
# Plotting outputs
plt.scatter (diabetes_X_test, diabetes_y_test, color='black')
plt.plot (diabetes_X_test, diabetes_y_pred, color='red',
          linewidth=1)

plt.show()
```

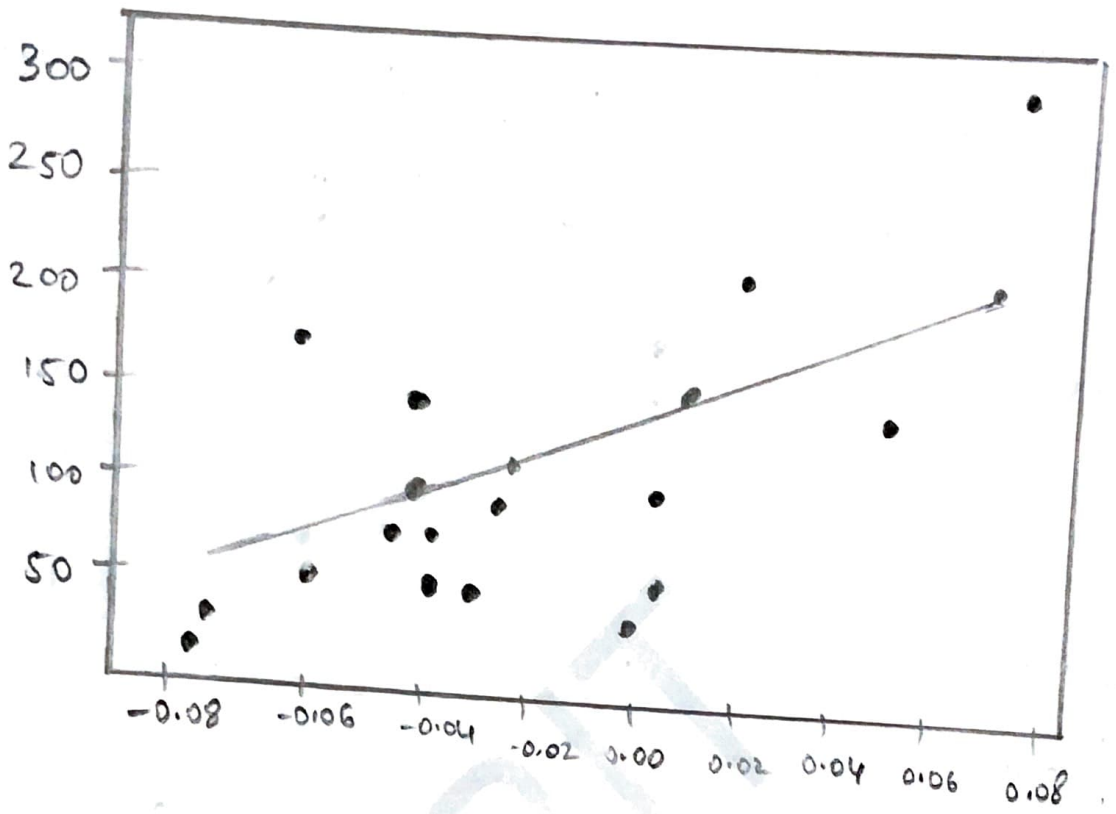
output :

Input values

```
[[ 0.07786339],[-0.03961813],[ 0.01103904],[-0.04069594],
 [-0.03422907],[ 0.00564998],[ 0.08864151],[-0.03315126],
 [-0.05686312],[-0.03099563],[ 0.05522933],[-0.06009656],
 [ 0.00133873],[-0.02345095],[-0.07410811],[ 0.01966154],
 [-0.01590626],[-0.01590626],[ 0.03906215],[-0.0730303 ]]
```

Predicted output values

```
[225.9732401  115.74763774  163.27610621  114.73638965  120.80385422
 158.21988574  236.08568105  121.81509832   99.56772822  123.83758651
 204.73711411   96.53399594  154.17490936  130.91629517   83.38783227
 177.36605897  137.99500384  137.99500384  189.56845268   84.39906687]
```


300
250
200
150
100
50
-0.08
-0.06
-0.04
-0.02
0.00
0.02
0.04
0.06
0.08

```
Input Values
[[ 0.07786339]
 [-0.03961813]
 [ 0.01103904]
 [-0.04069594]
 [-0.03422907]
 [ 0.00564998]
 [ 0.08864151]
 [-0.03315126]
 [-0.05686312]
 [-0.03099563]
 [ 0.05522933]
 [-0.06009656]
 [ 0.00133873]
 [-0.02345095]
 [-0.07410811]
 [ 0.01966154]
 [-0.01590626]
 [-0.01590626]
 [ 0.03906215]
 [-0.0730303 ]]
Predicted Output Values
[225.9732401  115.74763374 163.27610621 114.73638965 120.80385422
 158.21988574 236.08568105 121.81509832  99.56772822 123.83758651
 204.73711411  96.53399594 154.17490936 130.91629517  83.3878227
 171.36605897 137.99500384 137.99500384 189.56845268  84.3990668 ]
```