### **B.M.S. COLLEGE OF ENGINEERING BENGALURU**

Autonomous Institute, Affiliated to VTU



### Lab Record

### **Machine Learning**

Submitted in partial fulfillment for the 6<sup>th</sup> Semester Laboratory

Bachelor of Engineering in Computer Science and Engineering

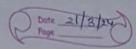
Submitted by:

**R SHREYA** 

1BM21CS152

Department of Computer Science and Engineering B.M.S. College of Engineering Bull Temple Road, Basavanagudi, Bangalore 560 019 Mar-June 2024

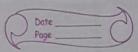
Lab	Program
1	Write a python program to import and export data using Pandas library functions
2	Demonstrate various data pre-processing techniques for a given dataset
3	Demonstrate various data pre-processing techniques for a given dataset
4	Implement Linear and Multi-Linear Regression algorithm using appropriate dataset
5	Use an appropriate data set for building the decision tree (ID3) and apply this knowledge to classify a new sample.
6	Build Logistic Regression Model for a given dataset and 6 Build KNN Classification model for a given dataset.
7	Build Support vector machine model for a given dataset and Build k-Means algorithm to cluster a set of data stored in a .CSV file.  And Implement Dimensionality reduction using Principle Component Analysis (PCA)
8	Implement Random forest ensemble method on a given dataset And Implement Boosting ensemble method on a given dataset.
9	Build Artificial Neural Network model with back propagation on a given dataset



	LA	<u>B-1</u>	Date 2 Page	1/3/14	
Write	a python colote c	programming pards	to import	and function	,,
Code:	1 11 12				
X = X co		names			alline)
Output					
	Sepal length is	non supolated	thin-con peta	L pdd	do
			length	n Lid	th
0	49	3-0	l-u		gris- Setosi
0			- Grand	0-2	
	49	3-0	ŀu	0-2	gris- Setos
1	49	3-0 3-2	1.3	0-2	Snis- Sctos Snis- Schor
1 2	4 9 4-7 4-6 5-0	3-0 3-2 3-1	1·u	0-2	Snis- Sctos Snis- Schor Snis- Subsci
1 2 3 4	4 9 4-7 4-6 5-0 5-4	3-0 3-1 3-6 3-9	1·4 1·7	0-2 0-2 0-2 0-2	Snis- Sctos Snis- Sctos Snis- Sctos
1 2 3 4	49 4-7 4-6 5-0 5-4	3-0 3-1 3-6 3-9	1.4 1.7	0-2 0-2 0-2 0-2	Snis- Sctos Snis- Sctos Snis- Sctos
1 2 3 4 Code,	49 4-7 4-6 5-0 5-4	3-0 3-1 3-6 3-9	1.3 1.5 1.7 1.7	0-2 0-2 0-2 0-2 0-4	Smis- School Smis- School Smis- School Smis- School

itis data = pd. read.cev (url, nomes = col.nomes)

output. Sepallingth in com Sepal ridthin - com petal length petal in class widthincm 3.5 5-1 1.4 0 Schic iris-date to-csv (" deened ins-data cov") P · Lock of the season of the season of the



11-	Get the Data
	Download the Data
	The state of the s
	import or
	import tarfile
	import urlib
	Use these libraries to download fextract
	web-based dateset that is housing dataset
	Retrieve the date into "housing top"  Load the date into housing last
	load the date into harring car
	housing heads
	housing +info()
	housing describe()
	-> Using these commands, enspect the attributes
C. Land	of the dataset
- 49.4	import matplotlib pyplot as pt
119.00	import Geoborn as sns
	and set well the stage
	housing hist (bins=50, figsize = (20,11))
	plt. show()
	Total Control of the States
	voing motplottib & Seoborn libraries detecting
	The toutteen
	Creeke 1

## Create Test Set

- -> Splitting the dataset on test ratio = 0.2, i.e.,
  training data is 80% of dataset and
  testing data is 20% of dataset
- → Stratified compling is when mordom chosen

  date are representation of a whole target

  population. Each homogenous subgroup is

  called Strate.

## Discover and Vicuolize the Date to gain ineights

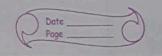
- -> Visualize the data using motplotlib and seaborn libraries
  - → Colcubring the standard correlation coefficient of every pair of column.

Prepare the date for Machine loarning algorithm

Date cleaning, hardling test and categorical date, Crustom Transformers, feature scaling, transformation pipelines etc, are done here

## Select and Train model

- but the model is exefilting the date.
- -> To Tackle this, DecisionTruRegrands Model is used as it is appoble of finding non-linear relation ships within the date.



But, the decision tree model is also overfitting so badly that it performs worse than the linear regression model

At lost Rondomforest regress model is used. It is

fine tune your model:

- on the test set and then bounch, monitor and maintaining the system
  - Traluct your system on the tut set by wing mean-squered even method.

Lounar, Months & Manbain your syntim

No can automate this process by:

- Collecting frish date rigularly and labeling it

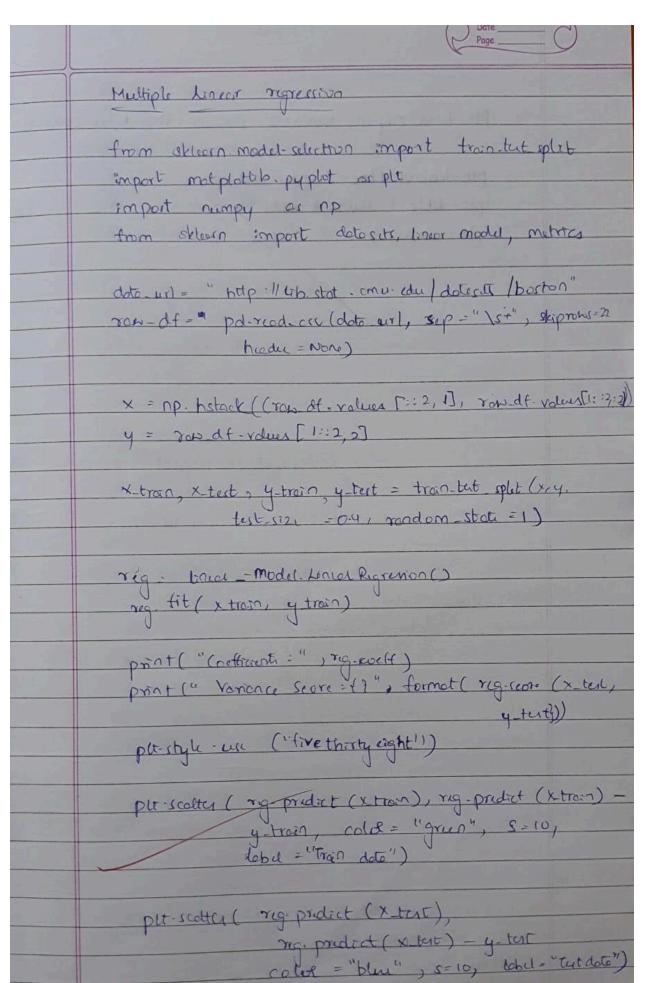
- briting script to train model and fine herethe

hype parameters

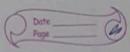
- Writing script to evaluate the model.

208-4 Python simplementation of Linear Regression import numpy as up import matplotth-pyplot as plt def estimate coef(2,y): n = np . (x) mx = np. mien(x) my = np men (4) ss ry = np. Sum(y+x) = n\*my \*mx 85\_22 = np - sum(x 2) - n = m2 = m2 bil = 88-24/55-22 b=0 = my - b1 + mx return (bo, bi) det plot regression line (2,4,6): plot-scotter (2,4, rola = "m", morker = "0", 5= 30) 4 prid - b[0] + b[1] +7 plt. plot (x,y, pred, cola="9") pt · xlobel ('x') plt. yebel ('4') det maines. 2 = np-oray ( [0,1,2, -. 9]) 4 = np. array ([1,3,2,...12]) print(b) blothedicuso - 1: ve (3.4. P) Output:

(b.0, b-1) = (1-2363, ... 1-16969, -



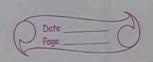
Plt-Lines (y=0, xmin=0) xmox=10, linewidth=2) plt-legend (loc= 'upper right')
plt-title ("Residual exert") plt- show().



[ of Ctorget ] - barget rapidly

LAB-I Implementation of 103 import numpy as np Emport pondos as pd eps = nprinto (front). eps from numpy import logs as log from google colob import drive drive mount ('/content/drive') Path = 'drive / my Drive / ml date cets / Play Trace car' df = pd. rood\_csv(Poth) det find entropy (df): torget - of kyr ()[-i] ontropy = 0 volues = df(target). unique() for value on values: fraction = af [target] - value counts () [value]/lea (afflogs) entropy += - fraction + np. log2 (fraction) yeturn entropy det overage information (alf, attributer) target = df. keys ()[-1] torget-voriables = df [torget]. unique) Vortables = df[attribute].conique() entropy2 = 6 for variable in variables: entropy = 0 for target variable and target variables rum = lin ( of Cottibuty [ of Cottibuty] == vorably

der = lo (df [attribute] [df [attribute] == vorable) from = num / (dentips) entropy + = - fraction \* log (fraction teps) froution; - der (lor(dt) entropy 2 = - fraction 2 + cotropy return abs (entropys) tree = build Tru(de) import pprint pprint - pprint (tra) Dutput: ('auttook' : f'orarcost': 'ya', 'rony': ('sorey': (false:'yes) 'sunny': Y'humidity': Y'high': 'no',



# Logistic Regression

from maplettib import pyplet as plt

from Sklearn metrics import accuracy-score

Path = 'drive/rayprive/midala sets/ insurance data-civ's

dt: pd. reed-civ (Path)

from Sklearn model selection import train test split
from matplotleb import pyplots as plt
% matplotleb enline,

plt. Scotter (doto ['(Et Score'] 4 date ['Admitted'].

morker = 1.1, cold = 'purple')

x troin , x test, y-train, y-lut = train test split (date

[[CETSCORO]], data ['Admitted'], train size = 0-8)

- from sklern linear model import Logistic Regrenson

model - kogistic Regrenson ()

model - fit (x train, y-train)

y-predicted = model predict (x text)

model - Score (x text > y-text)

print (y-predicted)

print (xtext)

from sklern brus model import linerRigranion

model: brus Repression()

model fil (xtrain, ytrain)

print("coefficient coefficient (b):", model coeff)

print("Intercept (b):", model intercept)

output.

ha 0.999

KNN Implementation import numpy as no import pandar as pd from google-colab import drive drive mount ( '/content / drive') datoset = political care ( /content /drive / Hypore/ins.ca) date set . head ) dateset groupby ('species'). Size () feotiere-columns = ['sipallingth', 'sepol width', 'petallingth'] X = dotact [fecture columns]. values y = datosit [ spices ] · volus from Sklern Preprocening import lobel Encodes le: Lobel Encodus) y = le-fit transform (y) from skleern model selection import train-test split x train, x test, y train, y test = train test spit (x, x, tut-size = 0.2 grandom state = 0) import matplotlib pyplot as plt a import suborn of one % motplottib inline from allern nighbours import knighbour (braince

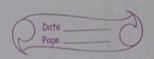
from okliera metrice import confusion-metrix, accuracy from skhorn model selection import cost-vol-core clanifica = Kneighbors Clanifica ( n. neighbors = 3) classifier - fit (xtrain, ytrain) 4 pred = classifier product (x-text) accuracy = accuracy - score (y test , y pred) \$ 100 print ( 'Accurracy of our model is equal + ctr ( mund (accurracy, 2)) + 1./-1) Output. Accuracy of our model is equal 96-67-6

SVM from okliern doteset import load - briest concu import motplottib. pyplot as plt from Skhorn inspection import Decision Boundary Duplay from oblean sum import suc conce = load brist concer) X = concy date [: 1:2] Y = concue target Svm = svc (keenel= "rbf", gamma=0.5, c=1.0) · svm. fit(xiy) Decision Boundary Display . from estimated ( sym, response \_ method = " predict") cmap = plt.cm. Spectral, alpho = 0.6, xbbd = conce feature namer (0), ylabel = concer. feeture\_nomentil, plt scatter (x [:, o], x[:, N, C= 41 S = 20, edge rolors = " "

plt show

K-Michs Clustering import pandar or pd import numpy as op import Seeborn as sns import motphotlib import skleen a dotoset import locdine import skleain. cluster import KHIONS x, y = lood\_ inis (roturn x-y = True) kmions = KMions (northister = 3, random state = 2) kmiens. fit (x) Knerns. cluster\_centers\_ pride kiniens - fit pridict (x)

Diminisionalty Reduction using PCA. import pander as pol impost numpy as ap import motplotlib pyplot as plt 1/2 motplotlib intin from sklein - decomposition import PCA from oklern proprocessing import Standardscole from skleern dotosch import load-brioit-conce date = locd\_bright-concur) print (date [' target nameil) print (date [ feature names ]) df1 = pd. Datofrane (date ['date], (dumni > doto [ fecture-mmy]) 8 caling = Standard Scalus scoling fit (df1) scaled-date = scoling. transform (df) principal = PCA(1-component = 3) principal . fit ( Socied data) X = principal stransform (scaled-date)



Implement Roadom Forcit encemble method

import pordar as pd

import numpy or np

import morphotlib pyplot as plt.

import seaborn or sne

import Skillern

from other import dotesta

from skleam-model selection import train-test split

from Sklean ensemble import Random Forut Classifier

from skleam metrice import accuracy score,

confusion\_motorx, claustication\_report

inis = datescte. locd-ins()

inc date = pd-Dato fram (1

'Supal length' : ins. dato [o:, 0],

'sepol width': ins. date [:, i],

'puta length': inic date [:2],

· putal width : ins date [:,3],

Species : Mistarger

3)

X = iris\_date.iloc[:,:-1]. values

y = inis\_date.ilocl:, -1]. rolues

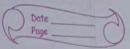
x-train, x-test, y-train, y-text = train\_test\_split(x, y,

text-8121 = 0.33, sondom state =0)

model = Random Forest Clanifical)

model fit (x-tran, y-tran)

occuracy-score (yout , y prid) Output : 0-56 Ado Boust from okleem detects import bood\_inc from skleen model selection import train-test splet from Skleen ensemble import AdoBoostClanifica from Sklerer : metrice import accuracy - score ins = load-ins() X = ins - date y = inc. target x-train, x-test, y train, y-test = train-test split (x, x, test-8121 = 04, random\_state = 32) adopost - clt = Ado Boost Classifice (n\_celimators - 30, learning - rat = 10, rondom\_state = 42) adaboost\_clf. fit (x-train, y train) Y-pred: colaboust\_clf-predict (x text) accuracy = accuracy (core (y tut, y pred) prot ("Accuracy ", occurracy) Output Accuracy = 0.96667



=	
	ANN Amplementation
	The state of the s
	import sumpy as up
	x = pp. array (([2,9], [1,5], [3,6]) = dtype= floot)
	y = np array (([a2], [86], [85]), dtype = flect)
	x = x /np amax (x, axis=0)
	y = y/100
	(Tara be appeared at
1	epoch : 5000
	Jack 1 - 6-1 har a medical a broaded
	input leyer neurons = 2
	hiddenlayer, neurons = 3
	output neurons =1
	The Construction total x + d.
	with = np. mordom. uniform (size = (inputlayer-neurone,
	hiddenlague neurone))
	bh = np-rondom uniform (oize= (1, hiddenleyee numare))
	Nout = np rondom · uniform ( Size = ( hiddenlayer neurone,
	otapia neuman
	bout = np. random · uniform (size = (1, output neurona))
	det sigmoid (2):  return 1/(1+np.exp(-2))
	return 1/(1+ np. (xp(-2))
	det denvolves sigmoid (x):
	return x (1-x)
	TOUGH CONTROL OF THE PARTY OF T
	for i in range (ipour):
	for i in range (ipoin):  hip = np. dot (x, +h)

hipp = hipp fbb

```
outings one dot (hoyer east, Nout)
     outing = outings + bout
     output = sigmoid (oatinp)
   to = y-output
    outgred = derivatives sigmoid (output).
    doutput = Eo * outgrad
    EH = d-output - dot (wout-T)
    hiddengrad = derivatives sigmaid (hlayel-act)
d-hiddenlayer = EH* hiddengrad
hout += hloyer. oct . T. dot (doutput) * Ir
wh += x-t-dot (d-hiddenloyer) * lr
print ("Input: 10" + str(x))
print ("Actual Output : 19" + str(y)
print ( Predicted output : In + output)
Input 1) - 100 mg my my more more more
  [ro.66667 1.1.-]
  [0-83333 O-STITE]
 [1... ] 0.666667]]
Actual Dutpert predicted outpert
  [10-92]
                          [ [0.86729245]
   [0.86]
                            [0.8451567]
   [089]]
           [0.8640413]
```