DWDM LAB Commands:-, MARKER, MARKES , GRADE

Sudo apt - get install pythons - veniv pythons -m venv dwdmlab source dwdmlab/bin/activate cd dwambb Source bin/active pip install numpy pip install skleam df = pd read. pip install pandas. (46) miss

1. # exp1. py:

from sklearn import datasets
from sklearn datasets import fetch_california_housing housing = fetch_california_housing(as_frame = TRUE) X = housing data

2. Most frequent value.

dil'marks 1' a dil ma

print(x) 3. Replace with a fixed value of

#expa.py:

from skleavn import datasets from skleavn datasets import fetch-california housing housing-data = fetch-california_housing().

Print (df) MARKS 1') (OJ)

x = housing - data data print (housing-data data)

dataset · csv MARKS1, MARKS2, MARKS3, GRADE 100, 90, 80, "Good" 70, 60, 70, "Average" 40, 45, 50, "fair" de la somo dambab bo import numpy as np
import pandas as pd

df = pd · read - csv ("dataset · csv")

print(df) # exp3. py. * Handling categorial data: DATA IMPUTATION: -: NOITATUAMI ATAO 1. Replacing the missing values with median. 2. Most frequent value. 3. Replace with a fixed value. import numpy as nptob tragani accesses and import pandas as pd

df = pd · read_csv('dataset · csv')

print(df)

18 [18] df['marks 1'] = df['Marks 2']. fillna (df['marksij mean()) print ("After DATA IMPUTATION") print(df) print (df["MARKS 1'] [0])

import numpy as np import pandas as po from sklean impute import Simple Imputer df = pd. read_esv('dataset.csv') print(df) imputer = Simple Imputer (Strategy = 'mean', missing-values imputer = imputer fit (df[MARKS1']]) df['MARKS1'] = imputer.transform(df[['MARKS1']]) print(df) import pandas as pd import numpy as np import copy df_flights = pd. read_csv("flights csv") Print(df. Hights. info()) cat_df_flights-df_flights.select_dbypes(include = Print(cat. df. flights ['UniqueCarrier']) [" object "]). copy() # Label Encoding cat. of_flights_lo; cat-flights copye) cat_df_flights_lefthiqueCarrier] = cat_df_flights k . print (cat df. flights_1c.dtypes) Conique Carrier J-astype Carrier cat_df_flights_lc['UniqueCassies']=cat_df_flights_lc ['Unique Carrier'] · cat · codes Print(cat_df_flights_lc)

7. cata.py import pand import pandas as pd ेर्राह्मा । । import numpy as np H- pd: sead_ esy import copy from skleam preprocessing import Label Encoder df_flights= pd. read_csv("flights.csv") 1 bobj = Label Encoder() MARKSO'] = i Print (df: flights info()) cat_skleann_flights = df_flights.copy() cat_skleagn_flights['Unique Carrier']=1bobj.fit_ transform (df_flights ['Unique Carrier] Print (cat_sklearn_flights ["thiqueCarrier"]) & cats-py Print (df. flights. import pandas as pd import numpy as np anticat, df. import copy of_flights = pd. read_csv ("flights.csv") Cat-onchot= df_flights. copy() cate one hot = pd. get_dummies (cat_one hot, - columns = ['UniqueCarrier'], prefix = ['Uniquelamor] Print(cate one hot)

g. import pandas as pd impost numpy as np import copy df-, flights = pd · read_csv("flights · csv") cat-onehot = df_flights.copy() 16-obj = Label Binarii Zeric) 1b_results = 1b-obj. fit_transform(df_flights ['UniqueCarrier']) Print (1b-results). from skleam. * To install jupy ter noteBOOK pip install jupyter notebook.

* jupyter notebook. python3 -m pip install -- upgrade pip python3 -m pip install -- upgrade pillow. *To install matplotlib use pip install matplotlibes, # Graph 1: - (topopostai · loboar, " : fgood tall) tring import numpy as np import pandos as pd import matplotlib pyplot as plt x = [1, 2, 3, 4, 5] Y = [10, 20, 30, 40, 50] plt. plot (x, y; linewidth=1.5; color = purple', manken = '*') plt.show()

Output: 45 35 1.0 15 2.0 25 303540455-0 20. #Linear Regression: import numpy as np import matplotlib.pyplot as plt from sklearn. linear model import Linear Regression x = p. array ([5,15, 25, 35, 45, 55]). reshape (-1,1) 7 = np. array ([5,20, 14,32,22,38]) ston 100 them Print(x,y) Pythons model = Lineau Regression () teni gig oneythons install result = model · score (x, y) dil tolotom 919 Print ("Score: ", result) olgran latent Print ("Intercept:", model intercept_) Graph 1 print ('Slope: ", model. coef_) y-pred = model · predict(x) Trogon print (Actual values of y: ', y) tropial print (Predicted values of y: ', y-pred) plt-scatter (x, y, color="black) pt. plot (x, y- pred, color= : pumple, manker= *, mark en face color = 'red)

Output: 10 20 30 40 50 # Linear Kegression of dataset: import numpy: as no doin doing x asternil import pandas as padedoile - test x automoil import matplotlib pyplot as plt from sklearn. linear model import Linear Regression n=pd. read_csv ('dataset.csv') X = np. array (n['MARKS1']). reshape(-1,1) y = np. array(n['MARKSa']) model=LineanRegression()
model-fit(x,y) r= model score (x,y) sot x sotodolo mottosa il print (x) Print ("Intercept:", model intercept_) print ("Slope:", model·coet.) y-pred = model predict(x)
print('Actual values of y:', y) print (Predicted values of y:', y-pred)
plt. scatter (x, y, color = 'black') plt. plot(x, y-pred, color='purple', linewidth=2, marker=x, markerfacecolor='red') plt. fitte ("Linean Regression")

#LineanRegression 2: import matplotlib pyplot as plt from sklearn import datasets, linear_model from sklearn metrics import mean_squared_error diabetes_x, diabetes_y = datasets · load_diabetes (return_x y = True diabetes_x = diabetes_x[:, np.newaxis, 2] trogeni diabetes_x_train = diabetes_x[:-20] trogoni diabetes_x_test = diabetes_x [-20:] troami diabetes-y-train= diabetes-y[:-20] diabetes_y_test = diabetes_y[-20:] 8.69=14 regr= linear_model. Linear Regression () regr. fit (diabetes - x - train, diabetes - y - train) diabetes-y-pred = regr. predict(diabetes-x-test). - print (diabetes-y-pred) plt. scatter (diabetes_x_test, diabetes_y_test, color = black) pit plot (diabetes_x test, diabetes_y - pred, color = blue linewidth = 2) print ("Mean squared croor: /2f"/ sa score (diabetes y-test diabetes-y-pred) pt. scatter (x, y, color - Back) Pit. plot(x, y-pred, color. purple; linewith = 2 imarker-x; markenta cecolos = 'sed') pt - Gille ("Linean Regression