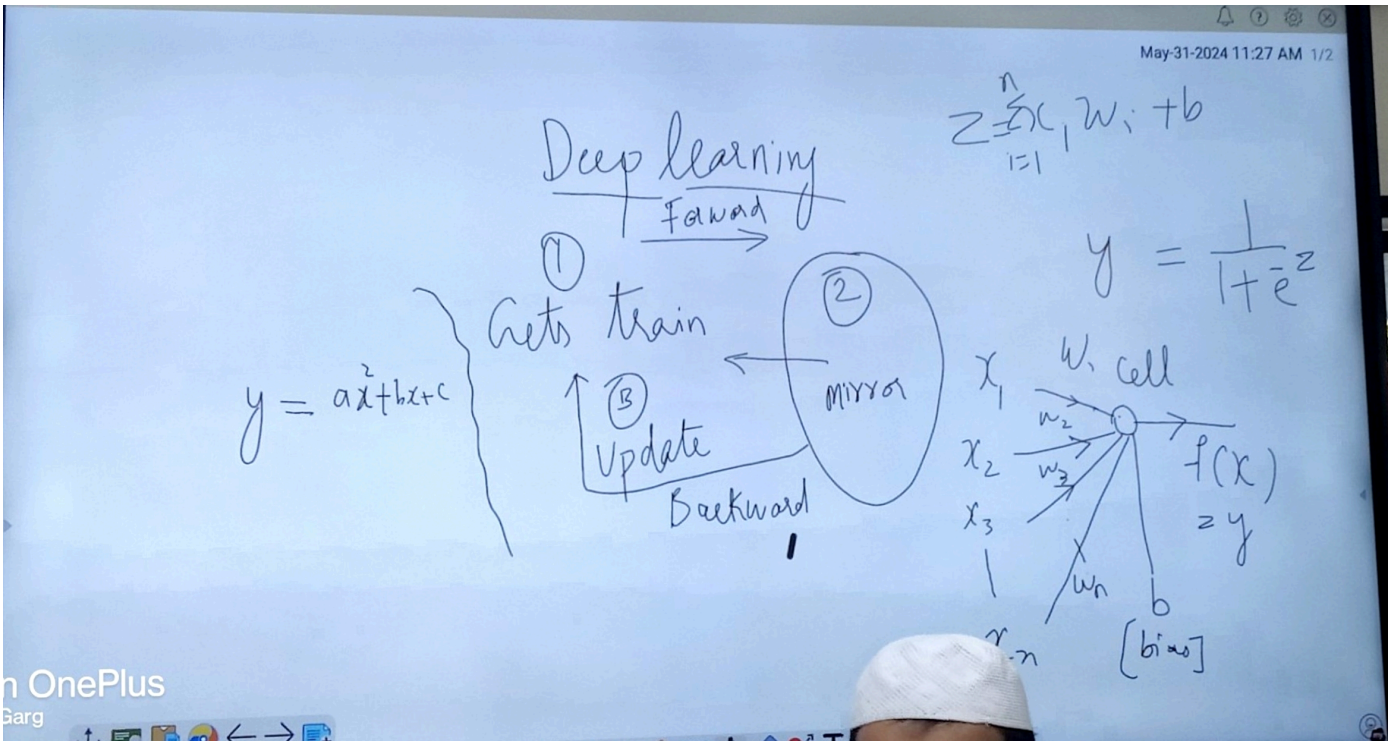
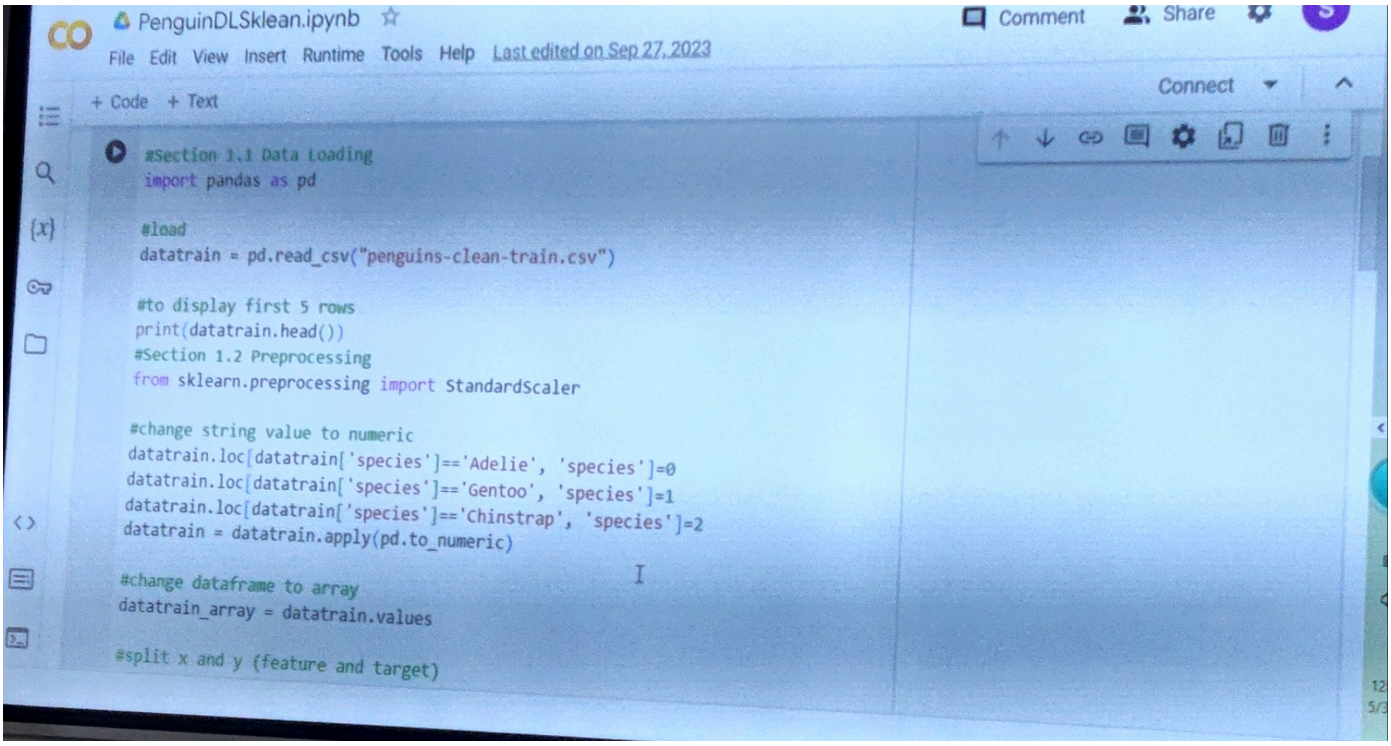
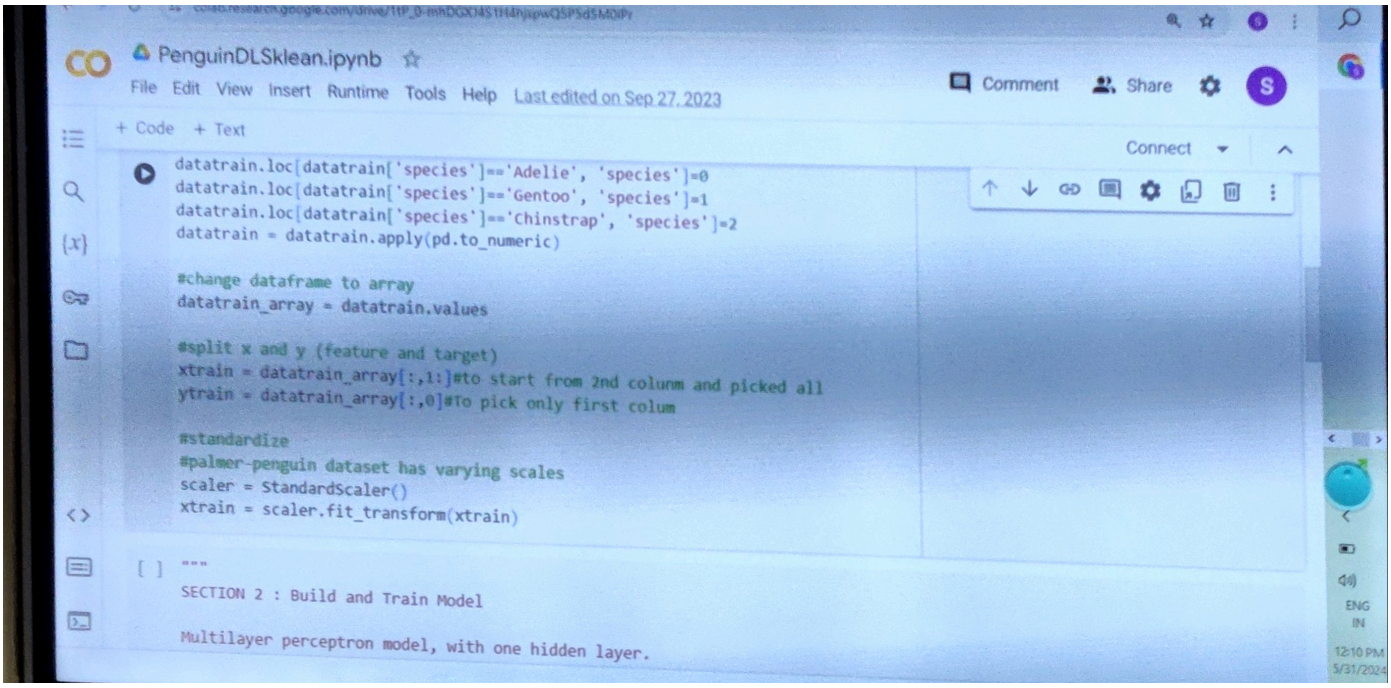
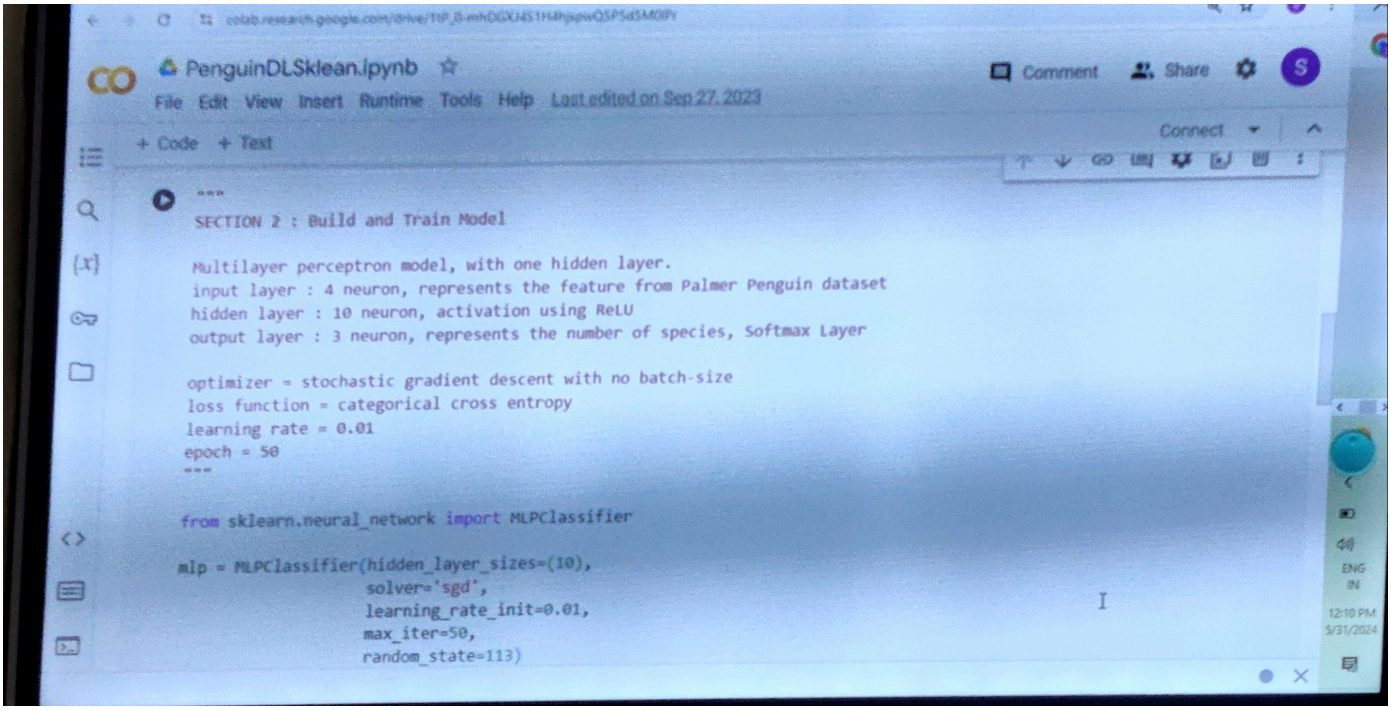
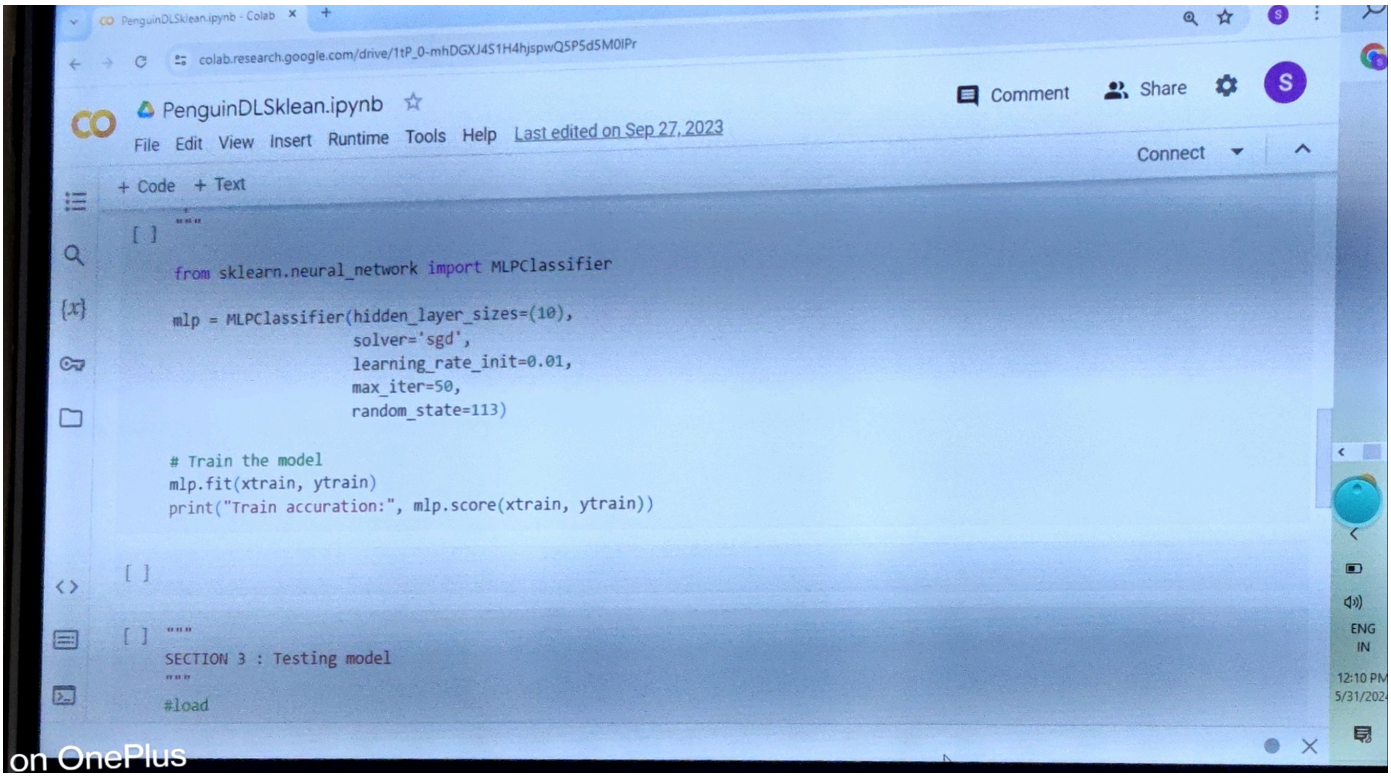
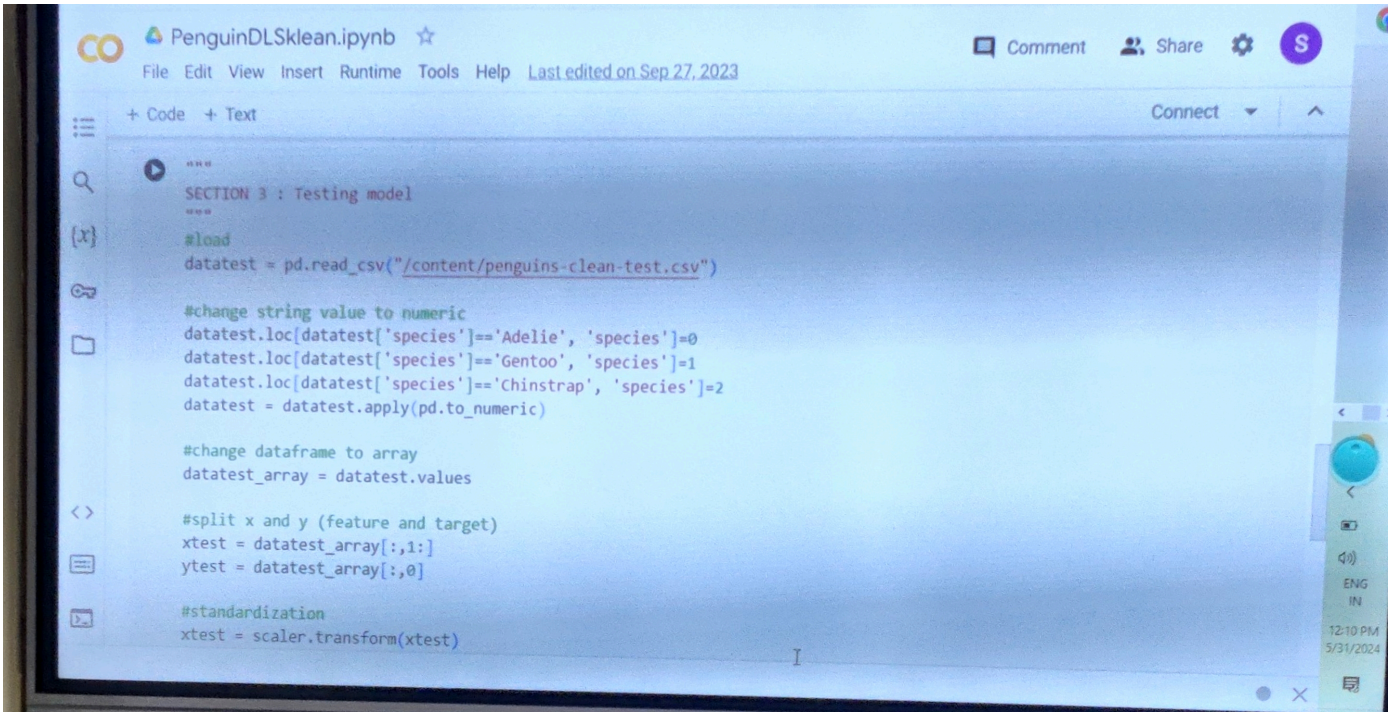

May-31-2024 11:27 AM 1/2
Deep learning
Fawad
$z = \sum_{i=1}^{n} x_i w_i + b$
$y = \frac{1}{1+e^{-z}}$
$y = ax^2 + bx + c$
1 Gets train
2 mirror
3 Update
Backward
x_1
x_2
x_3
x_n
w_1 cell
w_2
w_3
w_n
b
[bias]
f(x)
= y

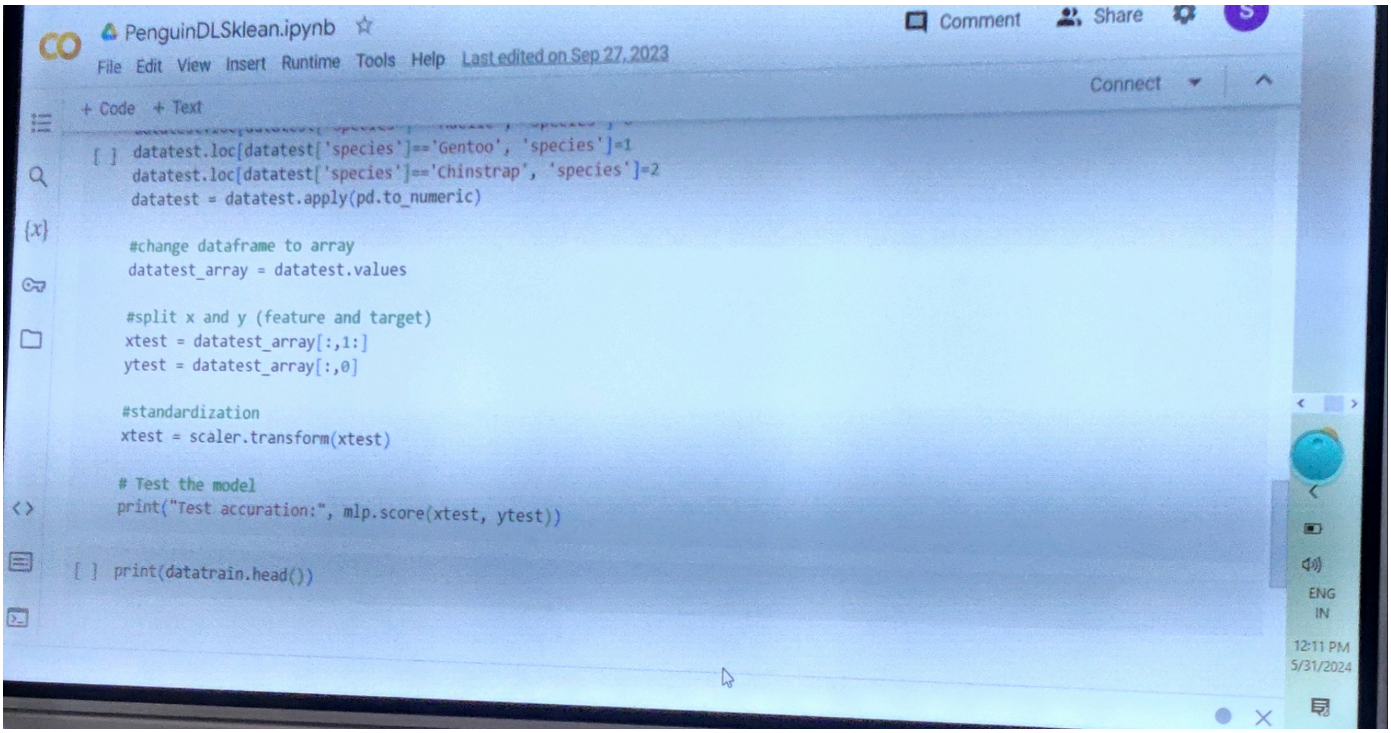

PenguinDLSklean.ipynb
File Edit View Insert Runtime Tools Help Last edited on Sep 27, 2023
Comment
Share
Connect
+ Code + Text
#Section 1.1 Data Loading
import pandas as pd
#load
datatrain = pd.read_csv("penguins-clean-train.csv")
#to display first 5 rows
print(datatrain.head())
#Section 1.2 Preprocessing
from sklearn.preprocessing import StandardScaler
#change string value to numeric
datatrain.loc[datatrain['species']=='Adelie', 'species']=0
datatrain.loc[datatrain['species']=='Gentoo', 'species']=1
datatrain.loc[datatrain['species']=='Chinstrap', 'species']=2
datatrain = datatrain.apply(pd.to_numeric)
#change dataframe to array
datatrain_array = datatrain.values
#split x and y (feature and target)


PenguinDLSklean.ipynb
File Edit View Insert Runtime Tools Help Last edited on Sep 27, 2023
Comment
Share
S
+ Code + Text
Connect
datatrain.loc[datatrain['species']=='Adelie', 'species']=0
datatrain.loc[datatrain['species']=='Gentoo', 'species']=1
datatrain.loc[datatrain['species']=='Chinstrap', 'species']=2
datatrain = datatrain.apply(pd.to_numeric)
#change dataframe to array
datatrain_array = datatrain.values
#split x and y (feature and target)
xtrain = datatrain_array[:,1:]#to start from 2nd columm and picked all
ytrain = datatrain_array[:,0]#To pick only first colum
#standardize
#palmer-penguin dataset has varying scales
scaler = StandardScaler()
xtrain = scaler.fit_transform(xtrain)
[ ] """
SECTION 2 : Build and Train Model
Multilayer perceptron model, with one hidden layer.
ENG
IN
12:10 PM
5/31/2024


PenguinDLSklean.ipynb
File Edit View Insert Runtime Tools Help Last edited on Sep 27, 2023
Comment
Share
Connect
+ Code + Text
"""
SECTION 2 : Build and Train Model
Multilayer perceptron model, with one hidden layer.
input layer : 4 neuron, represents the feature from Palmer Penguin dataset
hidden layer : 10 neuron, activation using ReLU
output layer : 3 neuron, represents the number of species, Softmax Layer
optimizer = stochastic gradient descent with no batch-size
loss function = categorical cross entropy
learning rate = 0.01
epoch = 50
"""
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(10),
solver='sgd',
learning_rate_init=0.01,
max_iter=50,
random_state=113)


PenguinDLSklean.ipynb - Colab
colab.research.google.com/drive/1tP_0-mhDGXJ4S1H4hjspwQ5P5d5M0IPr
PenguinDLSklean.ipynb
File Edit View Insert Runtime Tools Help Last edited on Sep 27, 2023
Comment
Share
Connect
+ Code + Text
"""
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(10),
solver='sgd',
learning_rate_init=0.01,
max_iter=50,
random_state=113)
# Train the model
mlp.fit(xtrain, ytrain)
print("Train accuration:", mlp.score(xtrain, ytrain))
"""
SECTION 3 : Testing model
"""
#load
ENG
IN
12:10 PM
5/31/2024
on OnePlus


PenguinDLSklean.ipynb
File Edit View Insert Runtime Tools Help Last edited on Sep 27, 2023
Comment
Share
S
+ Code + Text
Connect
"""
SECTION 3 : Testing model
"""
#load
datatest = pd.read_csv("/content/penguins-clean-test.csv")
#change string value to numeric
datatest.loc[datatest['species']=='Adelie', 'species']=0
datatest.loc[datatest['species']=='Gentoo', 'species']=1
datatest.loc[datatest['species']=='Chinstrap', 'species']=2
datatest = datatest.apply(pd.to_numeric)
#change dataframe to array
datatest_array = datatest.values
#split x and y (feature and target)
xtest = datatest_array[:,1:]
ytest = datatest_array[:,0]
#standardization
xtest = scaler.transform(xtest)
ENG
IN
12:10 PM
5/31/2024


PenguinDLSklean.ipynb
File Edit View Insert Runtime Tools Help Last edited on Sep 27, 2023
Comment
Share
Connect
+ Code + Text
datatest.loc[datatest['species']=='Gentoo', 'species']=1
datatest.loc[datatest['species']=='Chinstrap', 'species']=2
datatest = datatest.apply(pd.to_numeric)
#change dataframe to array
datatest_array = datatest.values
#split x and y (feature and target)
xtest = datatest_array[:,1:]
ytest = datatest_array[:,0]
#standardization
xtest = scaler.transform(xtest)
# Test the model
print("Test accuration:", mlp.score(xtest, ytest))
print(datatrain.head())
ENG
IN
12:11 PM
5/31/2024