A Report on Technical Internship at

Numaligarh Refinary Limited

In partial fulfillment of the requirements for the degree of Bachelor of Technology

Submitted by

PANCHALI BUZAR BARUAH

7TH Semester (CSE-80/20)

Department of Computer Science & Engineering

DIBRUGARH UNIVERISTY INSITITUTE OF ENGINEERING & TECHNOLOGY

Submitted to

Department of Computer Science & Engineering

DIBRUGARH UNIVERISTY INSITITUTE OF ENGINEERING & TECHNOLOGY

Dibrugarh, Assam, 786004

CERTIFICATE

TO WHOM IT MAY CONCERN

This is to certify that the industrial training on Information Technology at NUMALIGARH REFINERY LIMITED (NRL) was completed by Miss. Panchali Buzar Baruah student of 6th semester. B.tech (CSE) of Dibrugarh University Institute of Engineering and Technology (DUIET) under my guidance at NRL Information Technology Department for a period of 1 month starting from 3th July 2023 to 3th August 2023.

DECLARATION

I, Panchali Buzar Baruah, hereby declare that this Project entitled **"A BriefStudy Of Machine Learning (ML) & Deep Learning (DL) "** have been prepared by me during the year 2023 under the Summer Internship Programme at Learning Centre and IT Dept, Numaligarh Refinery Limited.

I further Declare that to my knowledge the structure and content of this project is original and have not been submitted before for any purpose.

Panchali Buzar Baruah

B-TECH, 6th SEMESTER

Computer Science & Engineering

Contents

NUMALIGARH REFINERY LIMITED

A GOVERNMENT OF INDIA ENTERPRISE

CIN: U11202AS1993GOI003893

P.O.: N.R. Project, District: Golaghat, Assam, PIN: 785699

Internship Programme

CERTIFICATE OF COMPLETION

To Whom It May Concern

This is to certify that Mr./Ms. Panchali Buzar Baruah a student of DUIET, Dibrugarh University undergone Winter/Summer Internship Training under IIS department Numaligarh Refinery Limited from 03-07-2023 to 02-08-2023

He / She has successfully completed Internship training and submitted a report titled A Brief Study of Machine Learning (ML) & Deep Learning (DL) and it's Application on (CNN)

During the training period Mr./Ms. Panchali Buzar Baruah as found to be sincere and good in his / her conduct.

We wish him/her all success in life.

Date : 03-08-23

NRL/T&D/2023-156

(Dr. Kajal Saikia)
Chief General Manager (HR & L)
NRL

(Asish Baruah)
Deputy General Manager (T&D)
NRL

Abstract

Numaligarh Refinery Limited (NRL) is a state-owned oil refinery located in Numaligarh, Assam, India. It is the first and only refinery in India to use indigenous crude oil. During my internship at NRL, I had the opportunity to work on a variety of projects related to Machine Learning. I gained valuable experience, and I also learned about the important work that NRL does to produce essential products for India. Overall my internship at NRL was a valuable experience that allowed me to apply my classroom knowledge to real-world problems. I am grateful for the opportunity to have worked with such talented and dedicated engineers.

Acknowledgements

I would like to take this opportunity to thank all the person without whose constant guidance, support and help, this project would not have been to completion. I avail training and development cell and Mr. Ashish Baruah - DGM (T&D) and Mr.Sangam Panchanan- Officer(T&D) for allowing me to undergo Summer Training at Numaligarh Refinery Ltd., Golaghat, Assam and also for their generous guidance and helping in the preparation of this report.

I acknowledge my indebtedness to our HOD of IT Department Mr.Sanjeev Dibragede- GM(IIS), for his guidance and constant supervision as well as for providing me necessary information regarding the project and familiarize me with practical industrial environment and our Summer Training project guide, Miss Hemanti Pegu Officer(IIS) who in spite of her busy schedule, spared timeto guide me with enlightening discussion, helpful suggestions and a lot of encouragement throughout the training period. I would like to thank Miss Padmakshi Mazumdar – Officer (IIS). I would also like to express my sincere gratitude to them. The Mentors along with other Officers were very cordialand helpful, whenever approached.

INTRODUCTION

1.1 OVERVIEW OF NUMALIGARH REFINERY LIMITED (NRL)

Numaligarh Refinery Limited (NRL) is a Miniratna PSU under Ministry of Petroleum& Natural Gas, was established as a Company on 22nd April 1993in accordance with the provisions made in the historic Assam Accord signed on 15th August 1985. The 3 MMTPA Refinery was dedicated to the nation byformer Prime Minister ofIndia Shri Atal Bihari Vajpayee on 9th July, 1999. NRL has been able to display Credible performance since commencement of commercial production in October 2000. With its concern, commitment and contribution to socio-economic development of the state of Assam and North East India, it is considered as one of the glowing manifestations of successful business enterprise in the region, with its footprints across the globe where its products, specially Paraffin Wax continue to be exported

NRL has embarked on a major integrated Refinery Expansion Project to treble its capacity from 3 MMTPA to 9 MMTPA at an estimated investment ofmore than Rs 28,000 Crore. One of the highest in the region. The project also includes setting up of a Crude Oil Import Terminal at Paradip Port in Odisha and laying off about 1640 KM of pipelines for transportation of imported Crude Oil to Numaligarh.

SHAREHOLDING PATTERN OF NRL:

NAME OF SHAREHOLDER	PERCENTAGE OF HOLDING
OIL INDIA LIMITED	69.93
Govt. of Assam	26.00
ENGINEERS INDIA LIMITED	4.37

PRODUCT RANGE:

NRL product range includes LPG, Naphtha, Motor Spirit (MS), Aviation TurbineFuel (ATF), Superior Kerosene Oil (SKO), High Speed Diesel (HSD), Raw Petroleum Coke (RPC), Calcined Petroleum Coke (CPC), Sulphur, Wax, Nitrogen, Mineral Turpentine Oil (MTO), Special Boiling Point Spirit (SBPS) and liquid Sulphur.

Refinery Process unit:-

- Crude Distillation Unit (CDU)
- Vacuum Distillation Unit (VDU)
- Delayed Coker Unit (DCU)
- Hydrogen Unit (H2U)
- Coke Calcination Unit (CCU)
- Sulphur Recovery Block (SRB)
- Motor Spirit Plant (MSP)
- Naphtha Splitter Unit (NSU)
- Wax plant

Refinery Product:-

• LPG
• Naphtha
• Motor spirit Block
• High Speed Diesel
• Diesel Hydro Treater Plant
• Superior Kerosene Oil
• Raw Petroleum Coke
• Calcined Petroleum Coke
• Sulphur Recovery Block
• Wax
• Liquid Nitrogen
• MTO

1.2 INFORMATION TECHNOLOGY

The company has made significant progress in the use of Information Technology for managing its resources. The Company implemented an ERP solution of M/s Ramco Systems, first in oil sector PSUs, in October 1998 to integrate all the business functions of the Company. With the changing business requirements, a need was felt to upgrade the ERP System to cater to new challenges emerging due to new business processes being adopted by the Company. The company had implemented SAP R/3 4.7 ERP suite in August 2005 to cater to the new business processes. Presently the company is using SAP S/4 HANA. The company has also implemented B2B (Business To Business) over SAP for seamless integration of exchange of business information with IOCL, BPCL, HPCL, OIL and ONGC, Nayara Energy.

NRL is using Aspen Tech suite of software for different applications in the refinery Presently, NRL is using PIMS for Refinery Production Planning. DMC3 for Process Control, AtOM / AORA for

NRL has been extensively using e-Procurement portals (CPPP – Central Public Procurement Portal & GeM - Govt e-Market Place) in procurement of materials and services, also utilizing e- auction platform for selling products and materials

Inventory Reconciliation and Yield Accounting and IP21as historian of plant dataand dashboard under Manufacturing & Supply Chain (MSC) suite of software licenses. NRL is also using Aspen Hysys for process simulation and analysis under Aspen One Engineering suite (AES) of software license.

For Quality Control, Thermofisher Scientific, Sample Manager LIMS manages the laboratory, data and procedural workflows, connecting with enterprise systems (Aspen Tech IP21 and SAP), instruments andequipment to deliver increased compliance and productivity across NABL certified lab at Refinery, Numaligarh and Marketing Terminal, Siliguri.

1.3 A BRIEF STUDY OF MACHINE LEARNING AND ITS APPLICATIONS

Machine learning is a branch of Artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Supervised Learning

Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of label datasets to train algorithms that to classify data or predict outcomes accurately.

Types Of Supervised Learning

Various types of algorithms and computation methods are used in the supervised learning process. Below are some of the common types ofsupervised learning algorithms: 1.Regression is used to understand the relationship between dependable and independent variables. Moreover, it is a type of supervised learning that learns from labeled data sets to predict continuous output for different data in an algorithm

1.3.1 There two types of regression they are as follows:

Linear regression:

It is used to identify the relationship between two variables, typically used formaking future predictions. Moreover, linear regression is subdivided based on the number of Independent and dependent variables

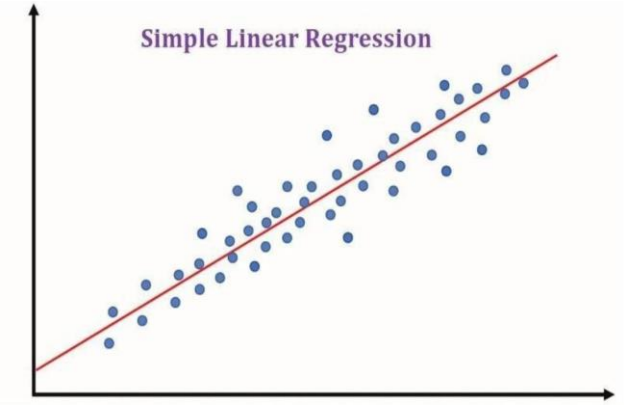

Fig-1

Logistic Regression

Logistic regression is used when the dependent variable is categorical or has binary outputs like ‘yes’ or ‘no’. Moreover, logistic regression is used to solve binary classification problems; that’s why it predicts discreet values for variables.

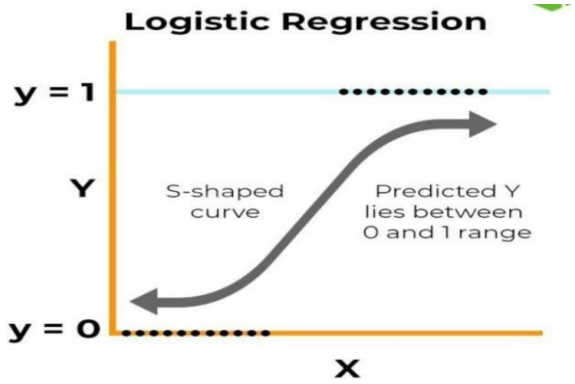

Fig-2

1.3.2 Classification

It is a type of supervised learning algorithm that accurately assigns data into different categories or classes. It recognises specific entities and analyzes them to conclude where those entities mustbe categorized. Some of the classification algorithms are as follows:

- K-nearest neighbor
- Random forest
- Support vector machines
- Decision tree
- Linear classifiers

Unsupervised Learning

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabelled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition.

1.3.3 Clustering Types

Following are the clustering types of Machine Learning:

- Hierarchical clustering
- K-means clustering
- K-NN (k nearest neighbors)
- Principal Component Analysis
- Singular Value Decomposition
- Independent Component Analysis

Fig-3

Clustering is an important concept when it comes to unsupervised learning. It mainly deals with finding a structure or pattern in a collection of uncategorised data. Unsupervised Learning Clustering algorithms will process your data and find natural clusters(groups) if they exist in the data. You can also modify how many clusters your algorithms should identify. It allows you to adjust the granularity of these groups.

1.3.4 What Is Cost Function?

Cost function measures the performance of a machine learning model for given data. Cost function quantifies the error between predicted and expected values andpresent thaterror in the form of a single real number. Depending on the problem,cost function can be formed in many different ways. The purpose of cost function is to be either:

Minimised: The returned value is usually called cost, loss or error. The goal is to find the values of model parameters for which cost function return as small a number as possibleMaximised: In this case the value yields is named as reward. The Goal is to find values ofmodel parameters for which the returned number is as large as possible

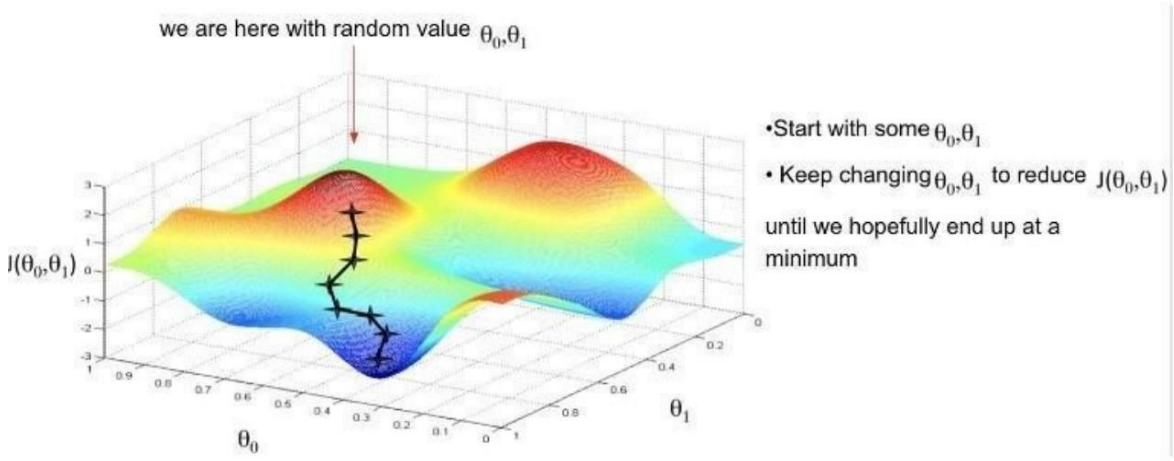
we are here with random value θ_0,θ_1
$J(\theta_0,\theta_1)$
θ_0
θ_1
•Start with some θ_0,θ_1
• Keep changing θ_0,θ_1 to reduce $J(\theta_0,\theta_1)$
until we hopefully end up at a minimum

Fig4

In machine learning and statistics, the learning rate is a tuning parameter inan optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function.

1.3.5 **Gradient Descent:** Gradient descent is an optimisation algorithm which is commonly-used to train machine learning models and neural networks. Training data helps these models learn over time, and the cost function within gradient descent specifically acts as a barometer, gauging its accuracy with each iteration of parameter updates.Until the function is close to or equal to zero, the model will continue to adjust its parameters to yield the smallest possible error. Once machine learning models are optimised for accuracy, they can be powerful tools for Artificial Intelligence (AI) and computer science applications.

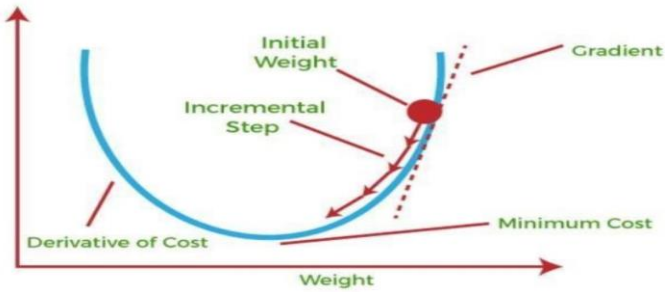

Fig-5

1.3.6 Support Vector Machine Algorithm (SVM)

Support Vector Machine or SVM which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line ordecision boundary that can segregate n- dimensional space into Classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.SVM Chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors,and hence algorithm is termed as Support Vector Machine**.**

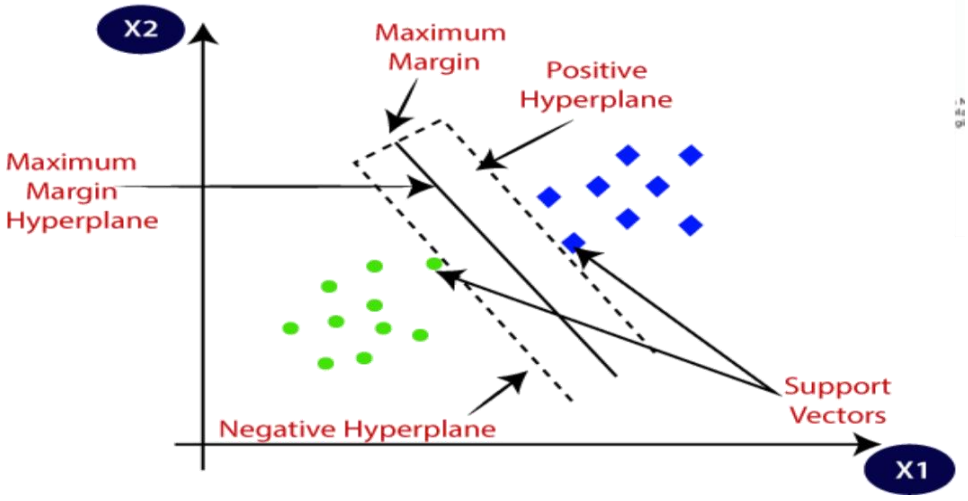

Fig-6

1.3.7 Half spaces

The half space hypothesis space is the set of hypotheses that consist of a hyperplane in a d-dimensional coordinate space that classifies a feature vector $\varphi(x) \in R^{d+1}$ as either -1 or 1 based on which side of the hyperplaneit lies. Here d represents the number of features of item x. A hypothesis in this space is often called a linear classifier or a perceptron.

2.1 **Convolutional Neural Network(CNN);**

A Convolutional Neural Network (CNN) is a type of deep learning algorithm that is particularly well-suited for image recognition and processing tasks. It is made up of multiple layers, including convolutional layers, pooling layers, and fully connected layers.

CNNs are trained using a large dataset of labeled images, where the network learns to recognize patterns and features that are associated with specific objects or classes. Once trained, a CNN can be used to classify new images, or extract features for use in other applications such as object detection or image segmentation.

Convolution Neural Network Architecture Model

The CNN architecture comprises three main layers: convolutional layers, pooling layers, and a fully connected (FC) layer.

2.1.1 The Convolutional Layer

Convolutional layers are the key building block of the network, where most of the computations are carried out. It works by applying a filter to the input data to identify features. This filter, known as a feature detector, checks the image input's receptive fields for a given feature. This operation is referred to as convolution. The final output of all the filter processes is called the **feature map**.

The CNN typically applies the **ReLU (Rectified Linear Unit**) transformation to each feature map after every convolution to introduce nonlinearity to the ML model. A convolutional layer is typically followed bya pooling layer. Together, the convolutional and pooling layers make up a convolutional block.

2.1.2 The Pooling Layers:

A pooling or downsampling layer reduces the dimensionality ofthe input. Like a convolutional operation, pooling operations use a filter to sweep the whole input image, but it doesn't use weights. The filter instead uses an aggregation function to populate the output array based on the receptive field's values.

There are two key types of pooling:

- **Average pooling:** The filter calculates the receptive field's average value when it scans the input.
- **Max pooling:** The filter sends the pixel with the maximum value to populate the output array. This approach is morecommon than average pooling.

2.1.3 The Fully Connected Layer

The final layer of a CNN is a fully connected layer.

The FC layer performs classification tasks using the features that the previous layers and filters extracted. Instead of ReLu functions, the FC layer typically uses a softmax function that classifies inputs more appropriately and produces a probabilityscore between 0 and 1.

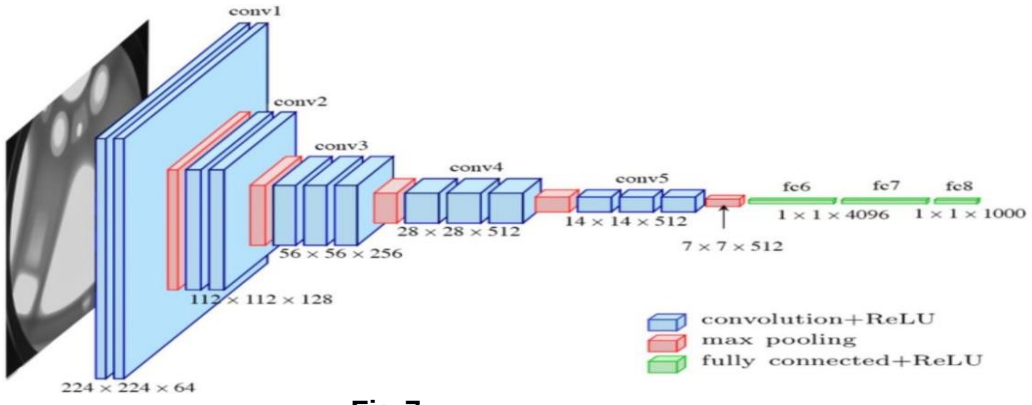

Fig-7

3.1 ASSIGNMENT-1:

PROBLEM STATEMENT: Implement the following -

Task1: Use DataFrame constructor in Pandas to create Dataframe.

Task2: Print a chosen row or a chosen column (data field).

Task3: Print rows that satisfy a condition on one of the column entries.

Task4: Use vectorised operations.

Task5: Add a new column to the DataFrame.

Task6: Export the DataFrame to a file in pickle format and save it as a file.Read thePickle file back to the program.

Task7: Use matplotlib library to make a scatter plot of columns that containNumeric Data, provide labels to the axes.

Task8: Implement Linear Regression to model the dependency between twovariables the predictor x and target y.

3.1.1 CODES:

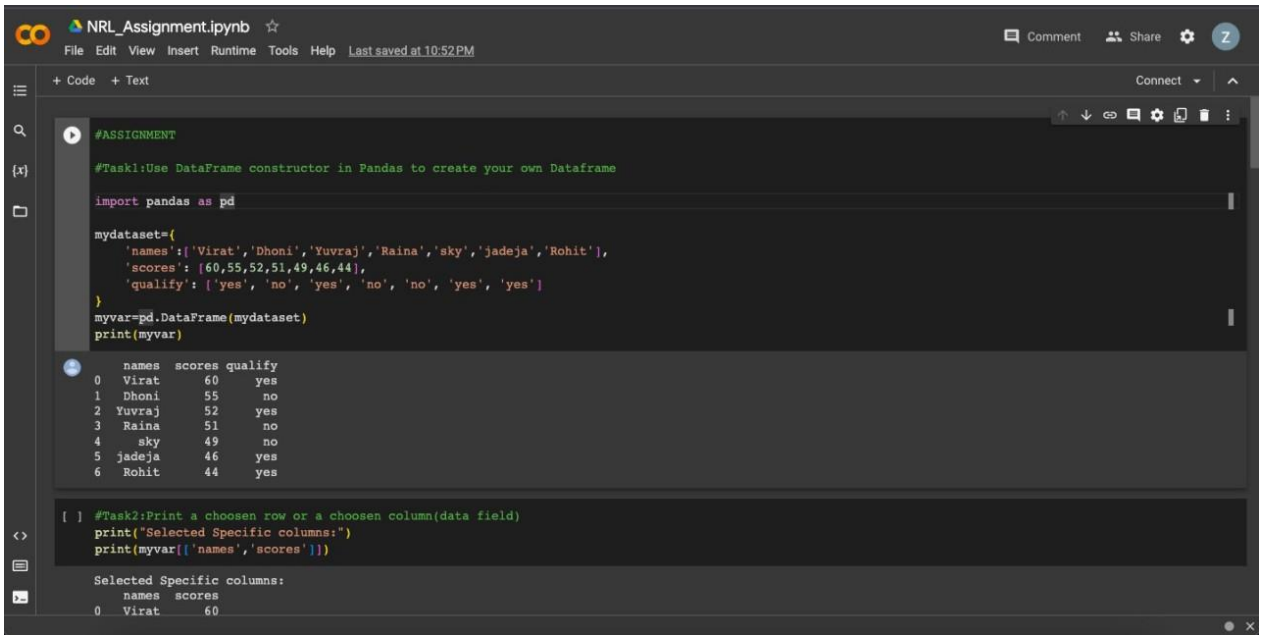

Fig - 8

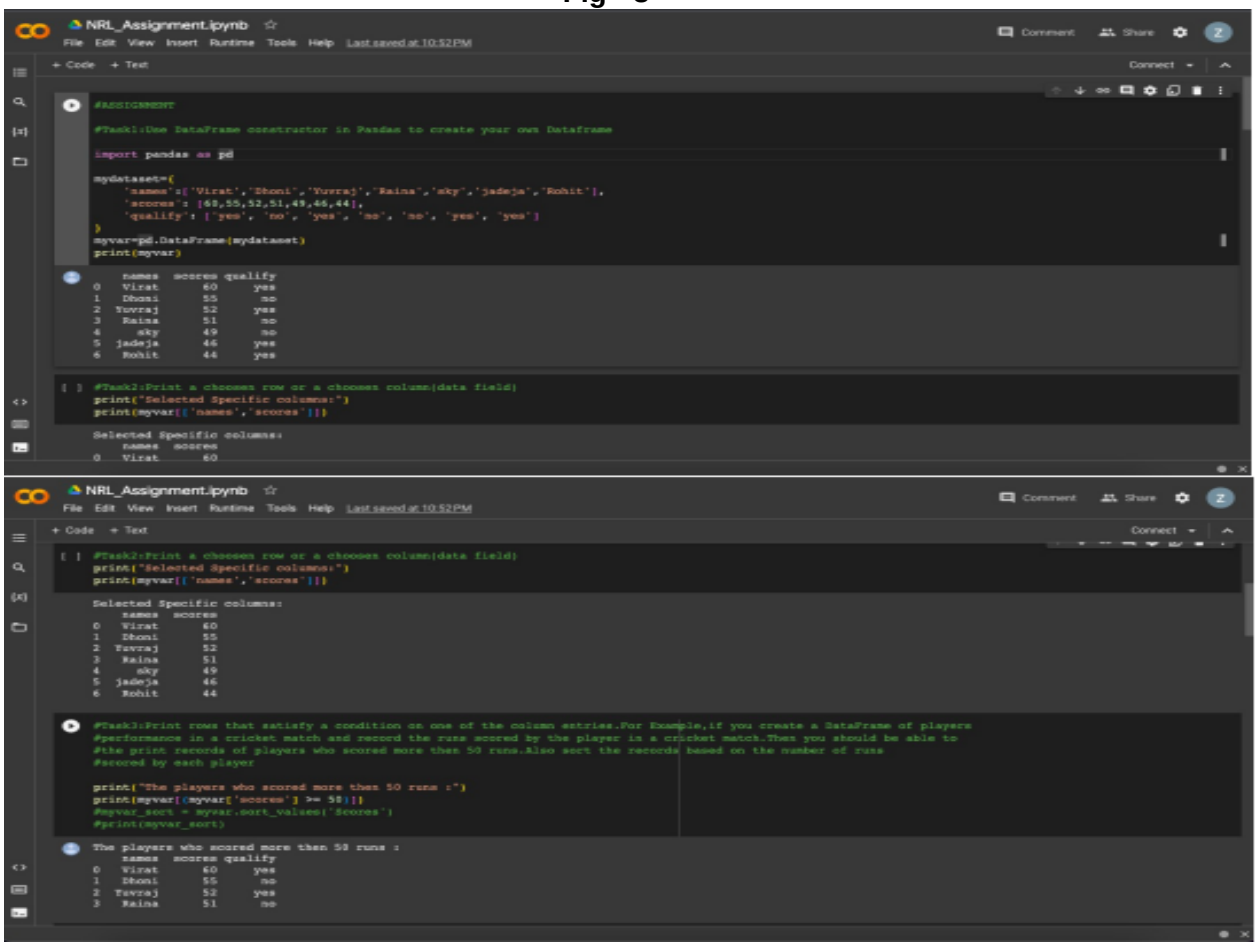

Fig-9

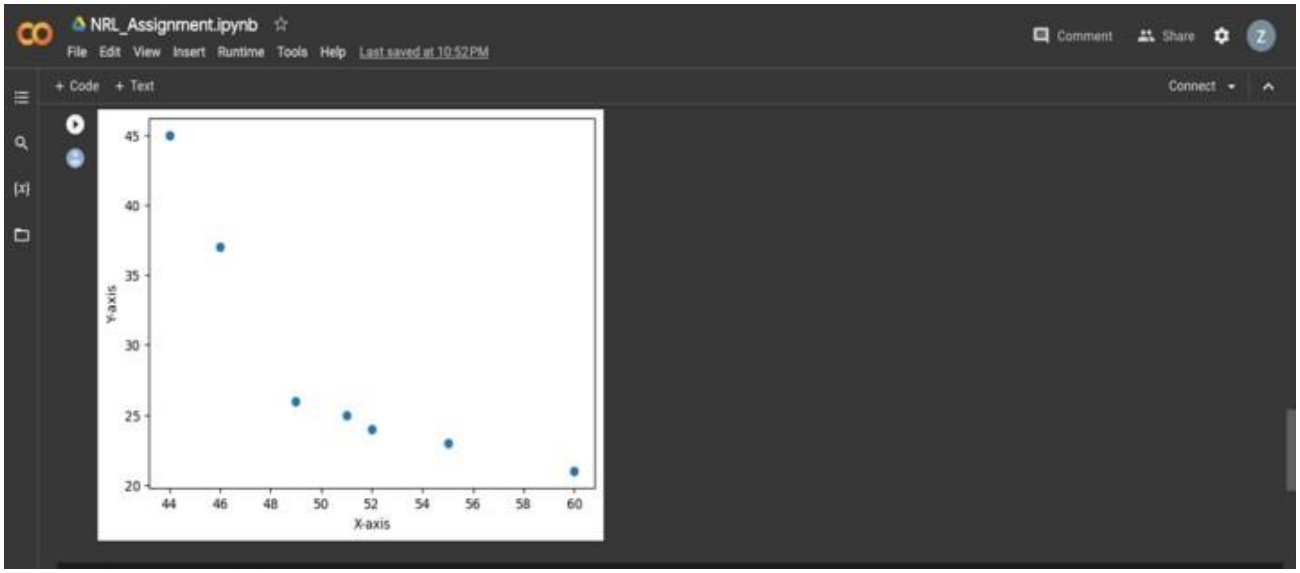
NRL_Assignment.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:52 PM
Comment
Share
+ Code + Text
Connect
Y-axis
X-axis
45
40
35
30
25
20
44
46
48
50
52
54
56
58
60

Fig-10

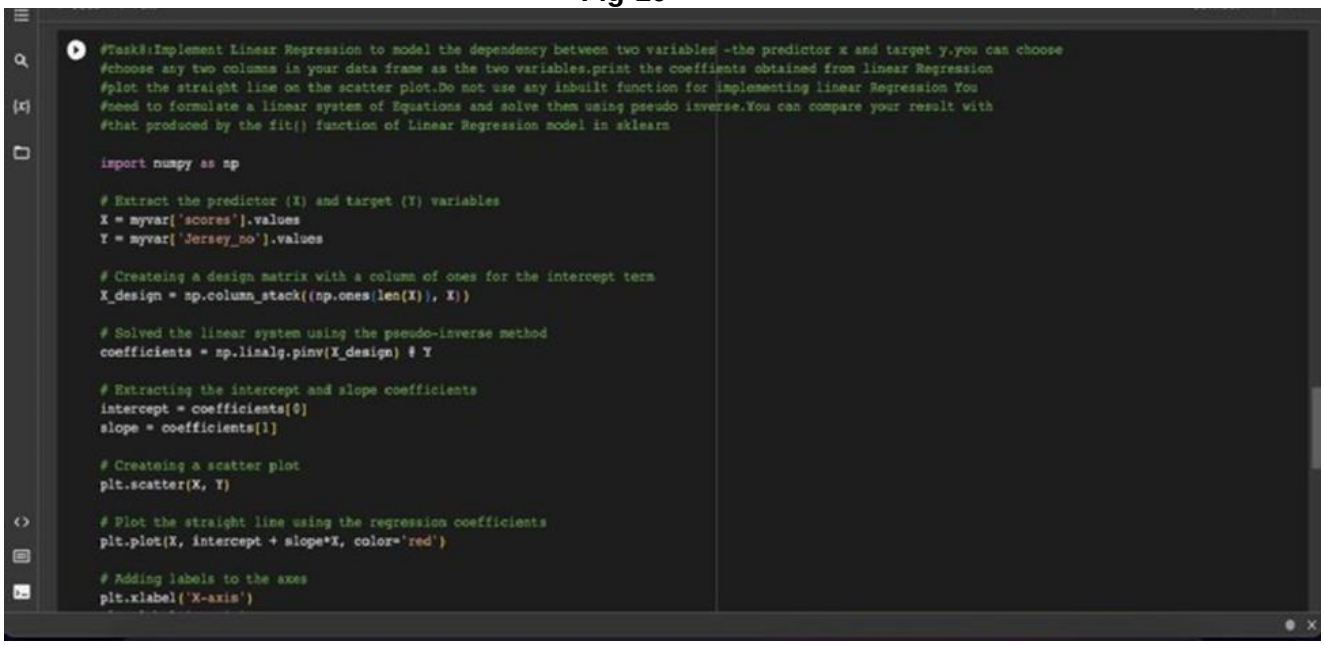
#Task8:Implement Linear Regression to model the dependency between two variables -the predictor x and target y.you can choose
#choose any two columns in your data frame as the two variables.print the coeffients obtained from linear Regression
#plot the straight line on the scatter plot.Do not use any inbuilt function for implementing linear Regression You
#need to formulate a linear system of Equations and solve them using pseudo inverse.You can compare your result with
#that produced by the fit() function of Linear Regression model in sklearn
import numpy as np
# Extract the predictor (X) and target (Y) variables
X = myvar['scores'].values
Y = myvar['Jersey_no'].values
# Creating a design matrix with a column of ones for the intercept term
X_design = np.column_stack((np.ones(len(X)), X))
# Solved the linear system using the pseudo-inverse method
coefficients = np.linalg.pinv(X_design) @ Y
# Extracting the intercept and slope coefficients
intercept = coefficients[0]
slope = coefficients[1]
# Creating a scatter plot
plt.scatter(X, Y)
# Plot the straight line using the regression coefficients
plt.plot(X, intercept + slope*X, color='red')
# Adding labels to the axes
plt.xlabel('X-axis')

Fig-11

3.2.ASSIGNMENT-2

PROBLEM STATEMENT:

Compare and analyze the results obtained by using the different Classifiers for thefollowing tasks.

1. Half Space

2. Logistic Regression (using inbuilt function)

3. SVM classifier (using a linear kernel)

4. SVM classifier (using a Polynomial kernel and a Gaussian kernel)

5.Logistic Regression using the SGD procedure

3.2.1:CODES-

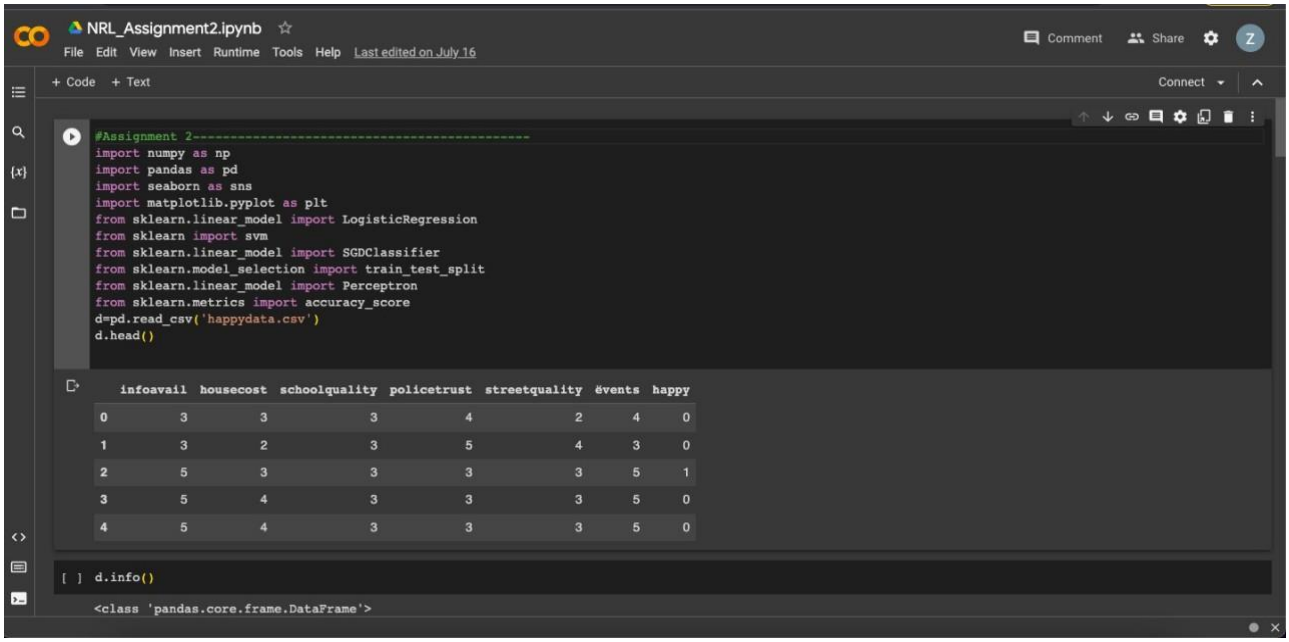

Fig-12

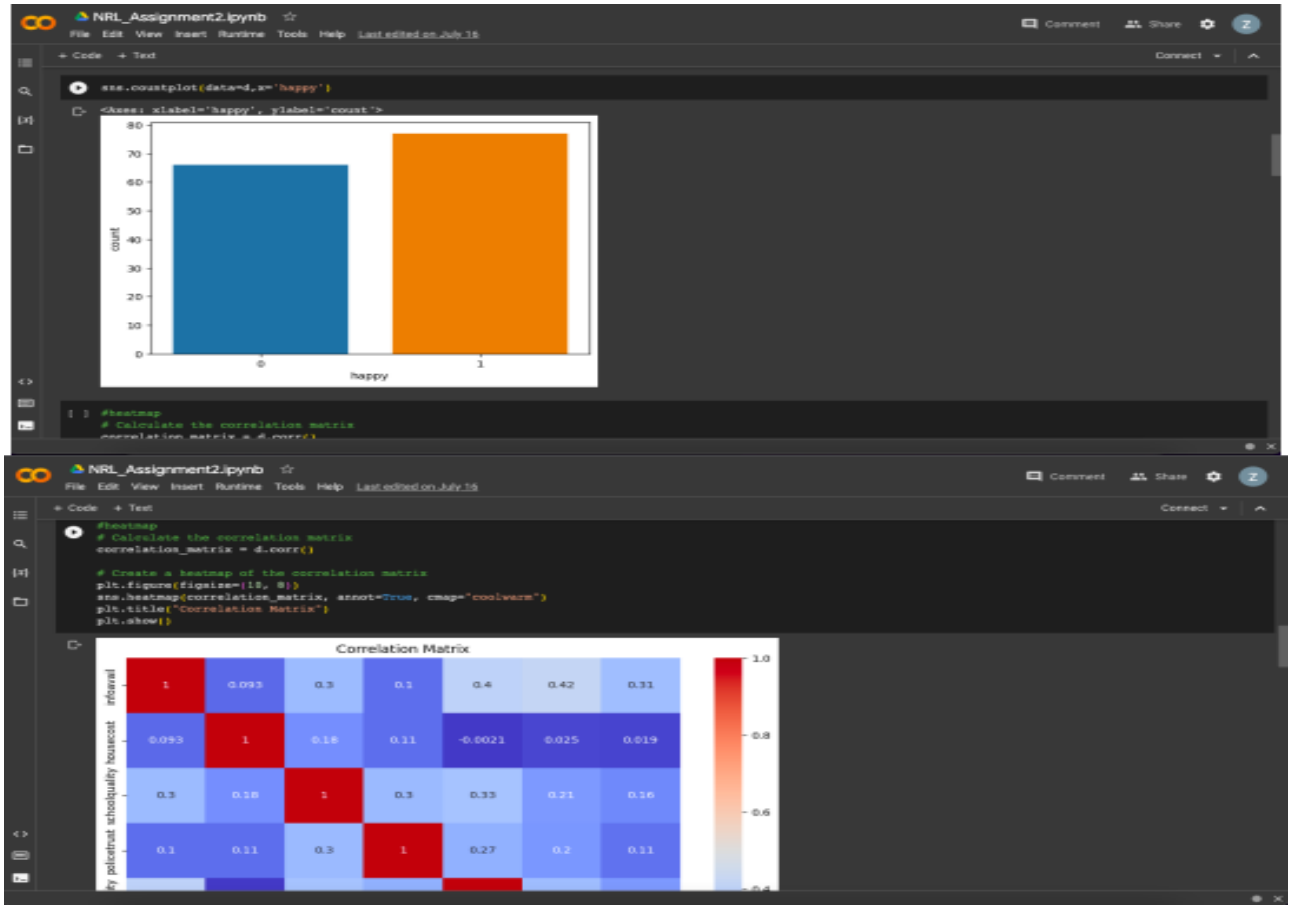

Fig-13

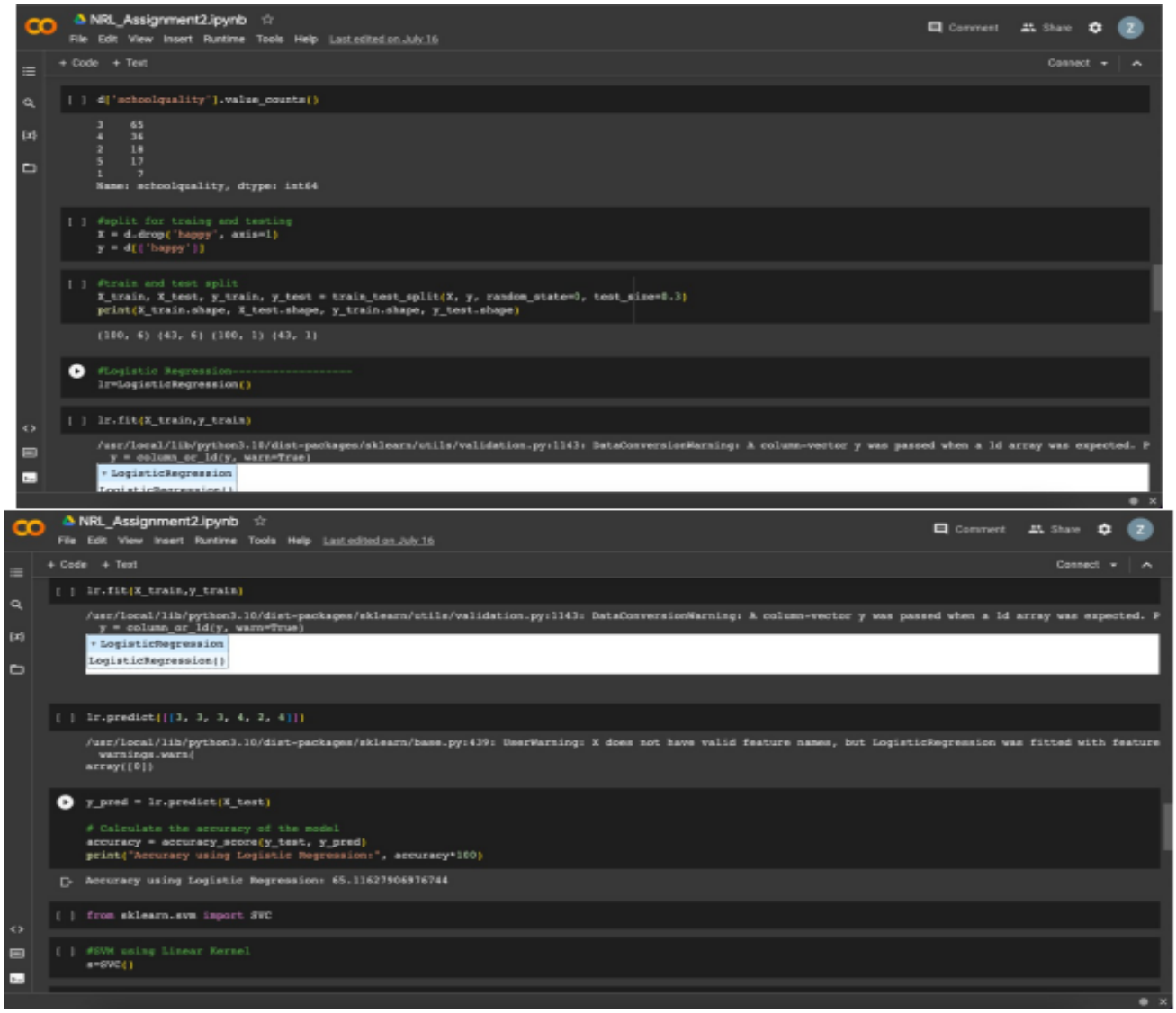
NRL_Assignment2.ipynb
File Edit View Insert Runtime Tools Help Last edited on July 16
Comment Share
+ Code + Text
Connect
[ ] d['schoolquality'].value_counts()
3 65
4 36
2 18
5 17
1 7
Name: schoolquality, dtype: int64
[ ] #split for train and testing
X = d.drop('happy', axis=1)
y = d[['happy']]
[ ] #train and test split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, test_size=0.3)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
(100, 6) (43, 6) (100, 1) (43, 1)
#Logistic Regression----------
lr=LogisticRegression()
[ ] lr.fit(X_train,y_train)
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. P
y = column_or_1d(y, warn=True)
LogisticRegression
LogisticRegression()
[ ] lr.predict([[3, 3, 3, 4, 2, 4]])
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LogisticRegression was fitted with feature
warnings.warn(
array([0])
y_pred = lr.predict(X_test)
# Calculate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy using Logistic Regression:", accuracy*100)
Accuracy using Logistic Regression: 65.11627906976744
[ ] from sklearn.svm import SVC
[ ] #SVM using Linear Kernel
s=SVC()

Fig-14

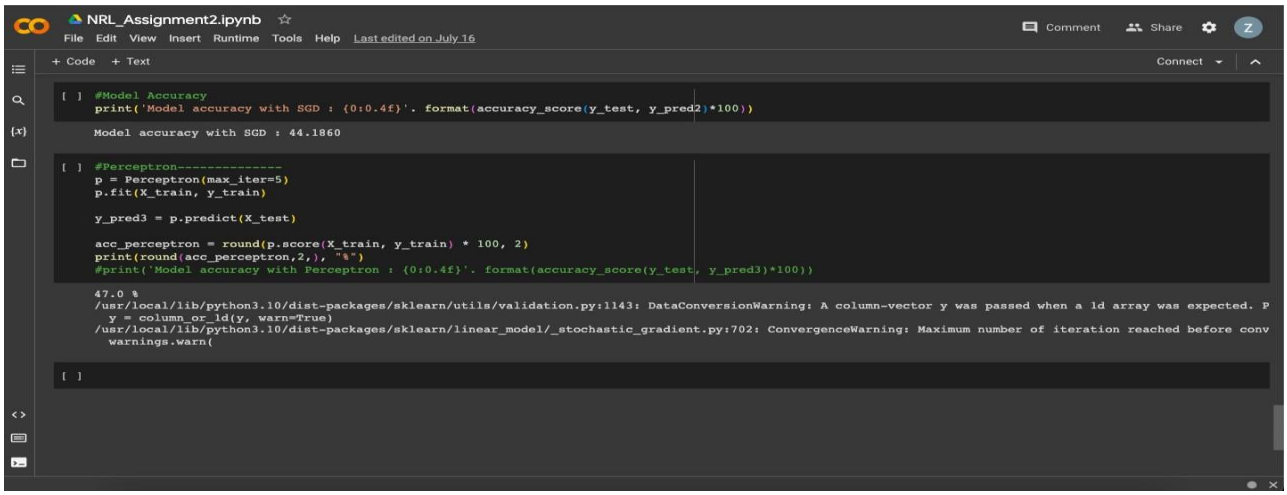
NRL_Assignment2.ipynb
File Edit View Insert Runtime Tools Help Last edited on July 16
Comment Share
+ Code + Text
Connect
[ ] #Model Accuracy
print('Model accuracy with SGD : {0:0.4f}'. format(accuracy_score(y_test, y_pred2)*100))
Model accuracy with SGD : 44.1860
[ ] #Perceptron----------
p = Perceptron(max_iter=5)
p.fit(X_train, y_train)
y_pred3 = p.predict(X_test)
acc_perceptron = round(p.score(X_train, y_train) * 100, 2)
print(round(acc_perceptron,2,), "%")
#print('Model accuracy with Perceptron : {0:0.4f}'. format(accuracy_score(y_test, y_pred3)*100))
47.0 %
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. P
y = column_or_1d(y, warn=True)
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_stochastic_gradient.py:702: ConvergenceWarning: Maximum number of iteration reached before conv
warnings.warn(
[ ]

Fig-5

4.1 **PROJECT:**

Implementation of a Fashion MNIST dataset classifier Using Tensor Flow in Python using CNN.Make a comparative study how the result gets affected if the percentage of the training data and Test data are changed. I.e. classification accuracy when training and test are taken as (60:40), (70:30), (80:20), (90:10)

4.1.1 About Dataset:

Fashion-MNIST is a dataset of Zalando's article images—consisting of a training set of60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. Zalando intends

Fashion-MNIST to serve as a direct drop-in replacement for the original MNIST datasetfor benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits.

4.1.2 Content:

- Each image is 28 pixels in height and 28 pixels in width, for a total of 784pixels in total.
- Each pixel has a single pixel-value associated with it, indicating the lightnessor darkness of that pixel, with higher numbers meaning darker.
- This pixel-value is an integer between 0 and 255.
- The training and test data sets have 785 columns.
- The first column consists of the class labels (see above), and represents thearticle of clothing.
- The rest of 784 columns (1-785) contain the pixel-values of the associatedimage

4.2. CODES-

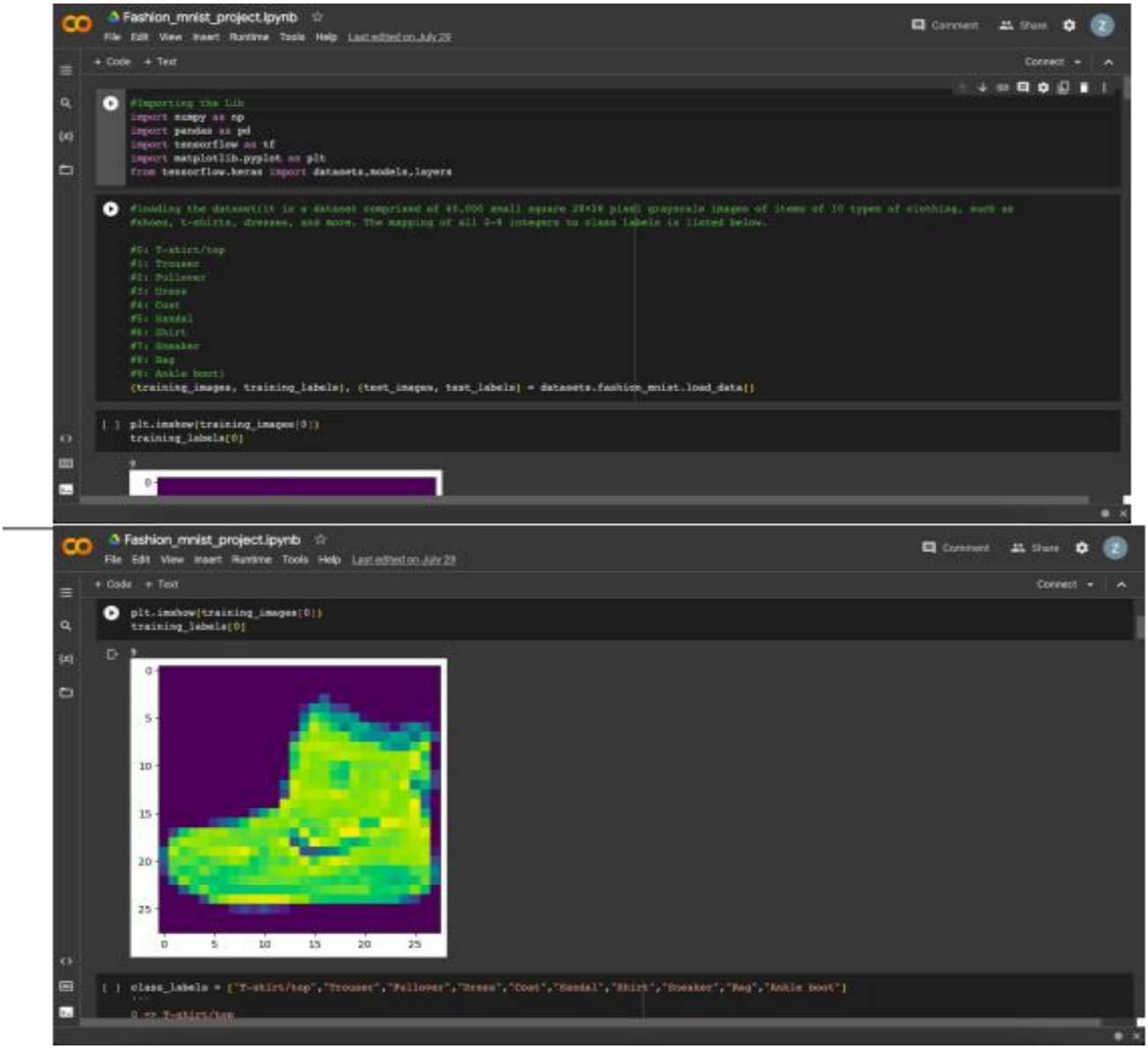

Fig-16

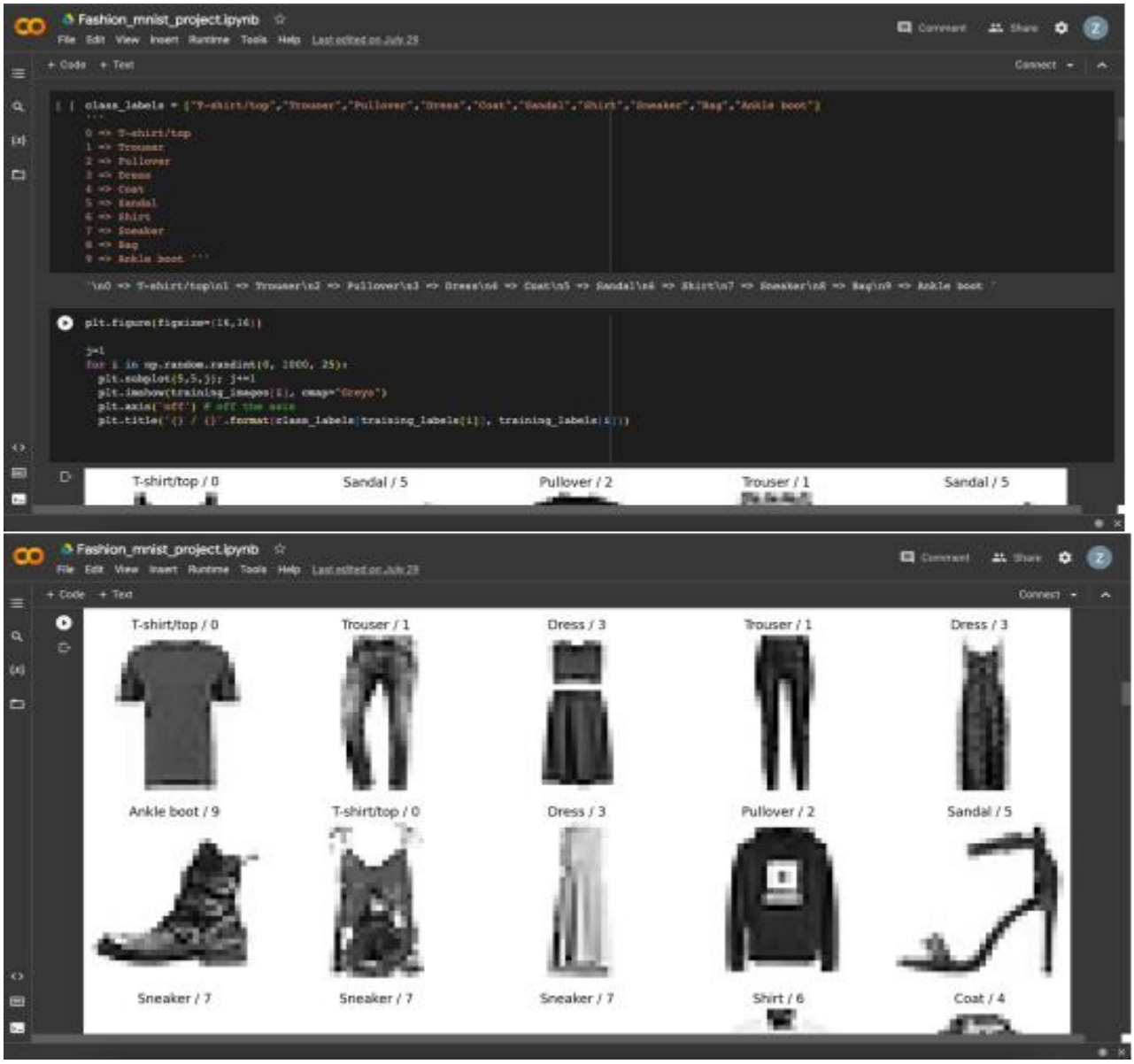
Fashion_mnist_project.ipynb
File Edit View Insert Runtime Tools Help Last edited on July 29
+ Code + Text
Connect
class_labels = ["T-shirt/top","Trouser","Pullover","Dress","Coat","Sandal","Shirt","Sneaker","Bag","Ankle boot"]
'''
0 => T-shirt/top
1 => Trouser
2 => Pullover
3 => Dress
4 => Coat
5 => Sandal
6 => Shirt
7 => Sneaker
8 => Bag
9 => Ankle boot '''
'\n0 => T-shirt/top\n1 => Trouser\n2 => Pullover\n3 => Dress\n4 => Coat\n5 => Sandal\n6 => Shirt\n7 => Sneaker\n8 => Bag\n9 => Ankle boot '
plt.figure(figsize=(16,16))
j=1
for i in np.random.randint(0, 1000, 25):
plt.subplot(5,5,j); j+=1
plt.imshow(training_images[i], cmap="Greys")
plt.axis('off') # off the axis
plt.title('{} / {}'.format(class_labels[training_labels[i]], training_labels[i]))
T-shirt/top / 0
Sandal / 5
Pullover / 2
Trouser / 1
Sandal / 5
Fashion_mnist_project.ipynb
File Edit View Insert Runtime Tools Help Last edited on July 29
+ Code + Text
Connect
T-shirt/top / 0
Trouser / 1
Dress / 3
Trouser / 1
Dress / 3
Ankle boot / 9
T-shirt/top / 0
Dress / 3
Pullover / 2
Sandal / 5
Sneaker / 7
Sneaker / 7
Sneaker / 7
Shirt / 6
Coat / 4

Fig-18

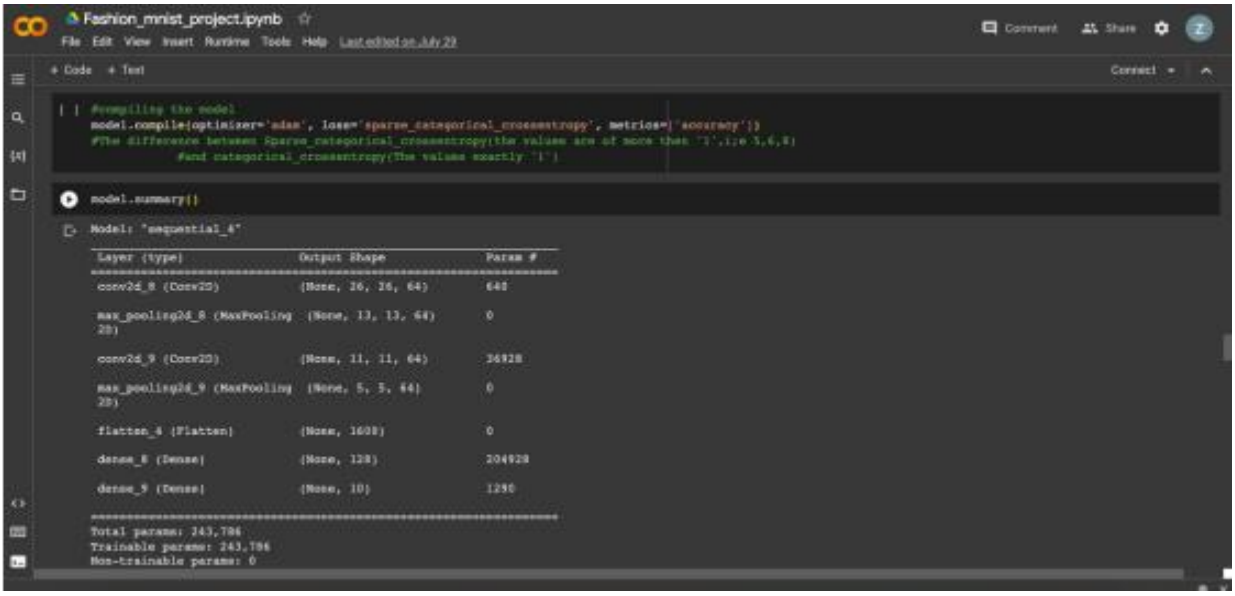
Fashion_mnist_project.ipynb
File Edit View Insert Runtime Tools Help Last edited on July 29
Comment Share
+ Code + Text
Connect
#compiling the model
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
#The difference between Sparse_categorical_crossentropy(the values are of more than '1',i;e 5,6,8)
#and categorical_crossentropy(The values exactly '1')
model.summary()
Model: "sequential_4"
Layer (type) Output Shape Param #
conv2d_8 (Conv2D) (None, 26, 26, 64) 640
max_pooling2d_8 (MaxPooling 2D) (None, 13, 13, 64) 0
conv2d_9 (Conv2D) (None, 11, 11, 64) 36928
max_pooling2d_9 (MaxPooling 2D) (None, 5, 5, 64) 0
flatten_4 (Flatten) (None, 1600) 0
dense_8 (Dense) (None, 128) 204928
dense_9 (Dense) (None, 10) 1290
Total params: 243,786
Trainable params: 243,786
Non-trainable params: 0

Fig- 19

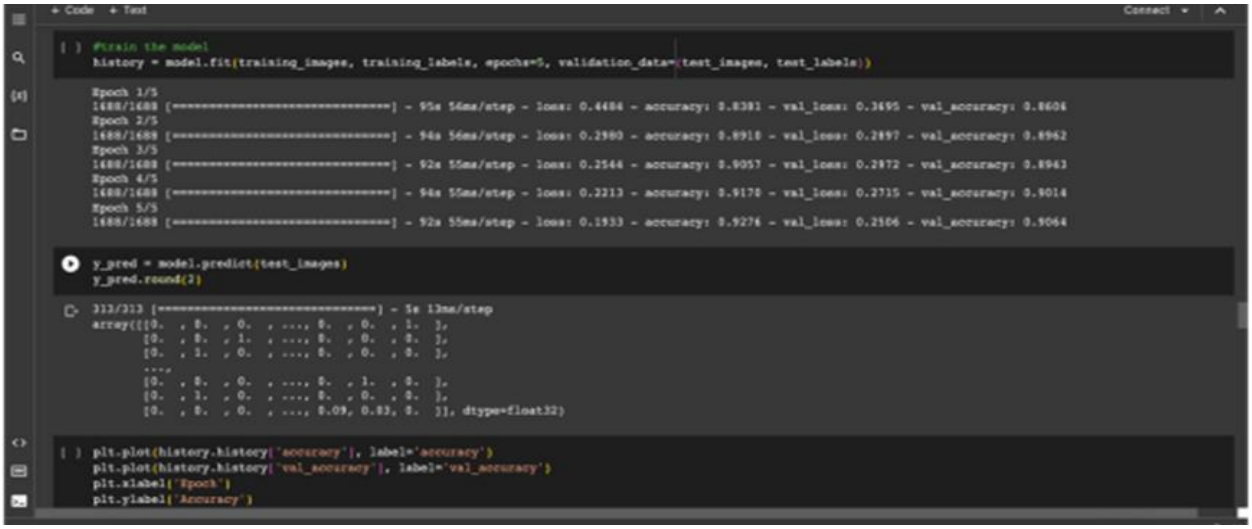
+ Code + Text
Connect
#train the model
history = model.fit(training_images, training_labels, epochs=5, validation_data=(test_images, test_labels))
Epoch 1/5
1688/1688 [==========] - 95s 56ms/step - loss: 0.4484 - accuracy: 0.8381 - val_loss: 0.3695 - val_accuracy: 0.8604
Epoch 2/5
1688/1688 [==========] - 94s 56ms/step - loss: 0.2980 - accuracy: 0.8918 - val_loss: 0.2897 - val_accuracy: 0.8962
Epoch 3/5
1688/1688 [==========] - 92s 55ms/step - loss: 0.2544 - accuracy: 0.9057 - val_loss: 0.2972 - val_accuracy: 0.8943
Epoch 4/5
1688/1688 [==========] - 94s 55ms/step - loss: 0.2213 - accuracy: 0.9178 - val_loss: 0.2715 - val_accuracy: 0.9014
Epoch 5/5
1688/1688 [==========] - 92s 55ms/step - loss: 0.1933 - accuracy: 0.9276 - val_loss: 0.2506 - val_accuracy: 0.9064
y_pred = model.predict(test_images)
y_pred.round(2)
313/313 [==========] - 1s 13ms/step
array([[0. , 0. , 0. , ..., 0. , 0. , 1.],
[0. , 0. , 1. , ..., 0. , 0. , 0.],
[0. , 1. , 0. , ..., 0. , 0. , 0.],
...,
[0. , 0. , 0. , ..., 0. , 1. , 0.],
[0. , 1. , 0. , ..., 0. , 0. , 0.],
[0. , 0. , 0. , ..., 0.09, 0.03, 0.]], dtype=float32)
plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label='val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')

Fig -20

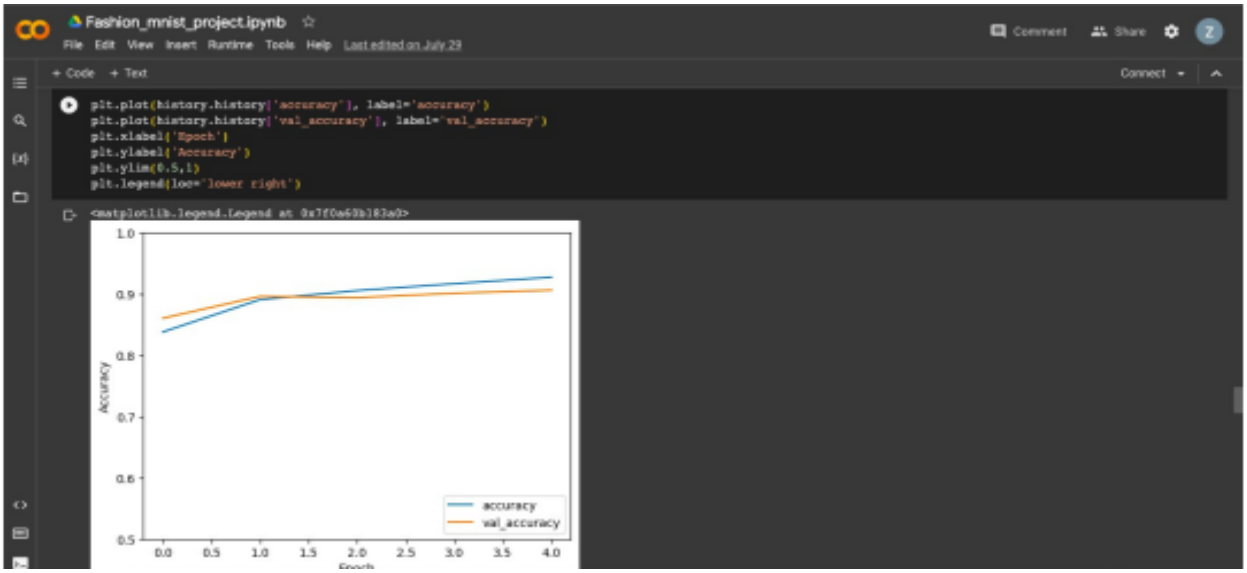
Fashion_mnist_project.ipynb
File Edit View Insert Runtime Tools Help Last edited on July 29
Comment
Share
Connect
+ Code + Text
plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label='val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim(0.5,1)
plt.legend(loc='lower right')
<matplotlib.legend.Legend at 0x7f0a63b183a0>
Accuracy
Epoch
accuracy
val_accuracy

Fig –21

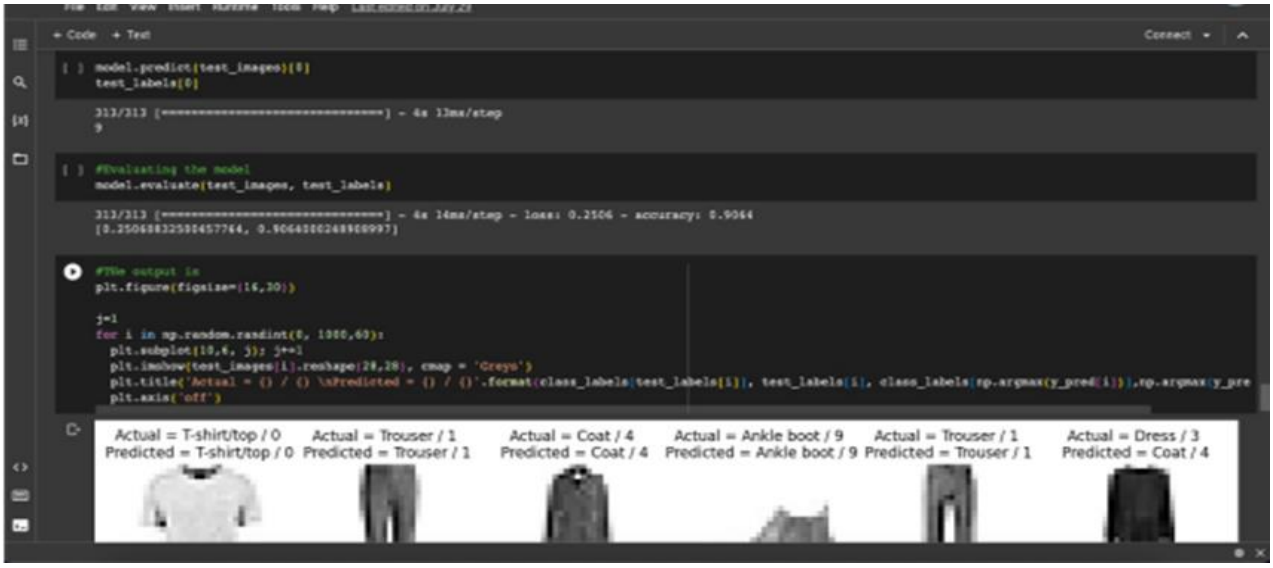
+ Code + Text
Connect
model.predict(test_images)[0]
test_labels[0]
313/313 [==========] - 4s 13ms/step
9
#Evaluating the model
model.evaluate(test_images, test_labels)
313/313 [==========] - 4s 14ms/step - loss: 0.2506 - accuracy: 0.9064
[0.25068832588195777, 0.9064000248908997]
#The output is
plt.figure(figsize=(16,20))
for i in np.random.randint(0, 1000,6):
plt.subplot(10,6, j); j+=1
plt.imshow(test_images[i].reshape(28,28), cmap = 'Greys')
plt.title('Actual = {} / {} \nPredicted = {} / {}'.format(class_labels[test_labels[i]], test_labels[i], class_labels[np.argmax(y_pred[i])],np.argmax(y_pre
plt.axis('off')
Actual = T-shirt/top / 0
Predicted = T-shirt/top / 0
Actual = Trouser / 1
Predicted = Trouser / 1
Actual = Coat / 4
Predicted = Coat / 4
Actual = Ankle boot / 9
Predicted = Ankle boot / 9
Actual = Trouser / 1
Predicted = Trouser / 1
Actual = Dress / 3
Predicted = Coat / 4

Fig -22

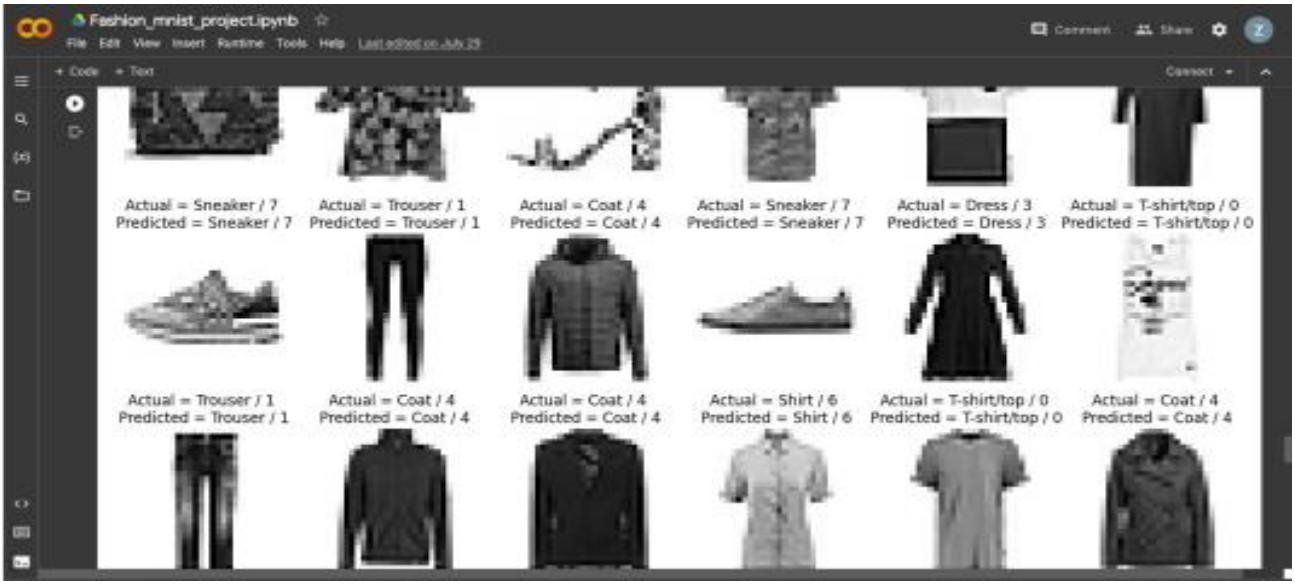

Fig – 23

Fig-24

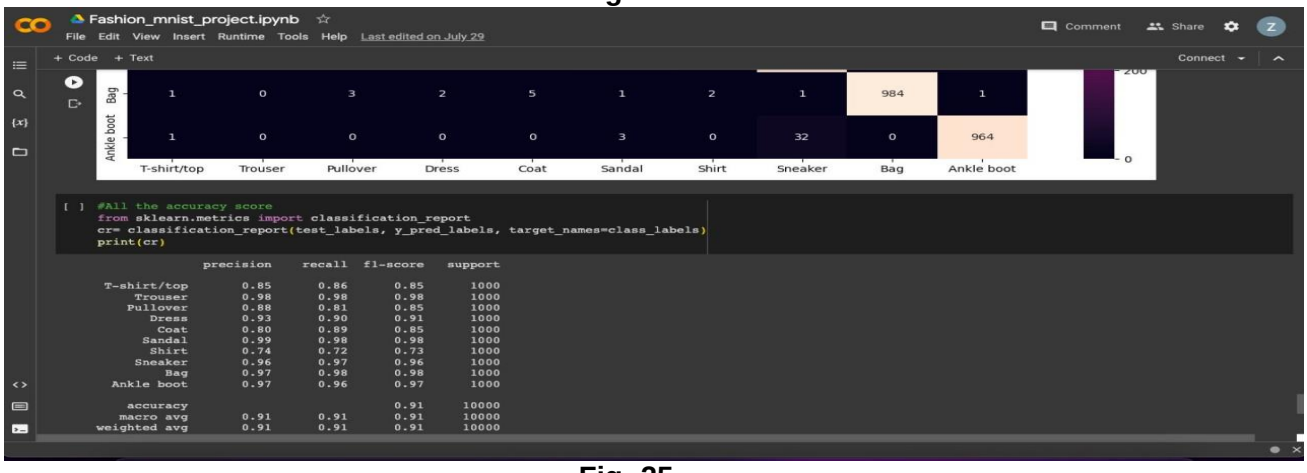

Fig- 25

CONCLUSION

In conclusion,the internship was a useful experience. I gained new Knowledge and skills and meet many new people during the internship period.I got insight into the work of an Industry(NRL) and learned the procedure of how a task is performed.I found out what my strengths and weaknesses are and how I can further improve myself. I would like to thank my parents,friends and the authorities of NRL who believed in me and helped me during difficult times. I also would like to special mentions Joy Anupol Neog,Arunangshu chetia and Panchali Buzar Baruah who interned along with me and were a part of this internship. Lastly, I thank each and every one who helped in any way during this internship and made this experience smooth and successful.

References

[1] Cricket Match, Dataset, from Kaggle,

[2] Scikit – learn,

[3] Binary dataset, Happydata , from kaggle,

[4] Fashion MNIST, An MNIST-like dataset of 70,000 28x28 labeled fashion images

[5] DanB,CollinMoris,DeepLearningFromScratch,

[6] DanB, Dropout and Strides for Larger Models,

[7] BGO, CNN with Keras,

[8] NAIN, EagerFMINST,

[9] Why Dropounts prevent overfitting in Deep Neural Networks

[10] Dropout: A Simple Way to Prevent Neural Networks from Overfitting