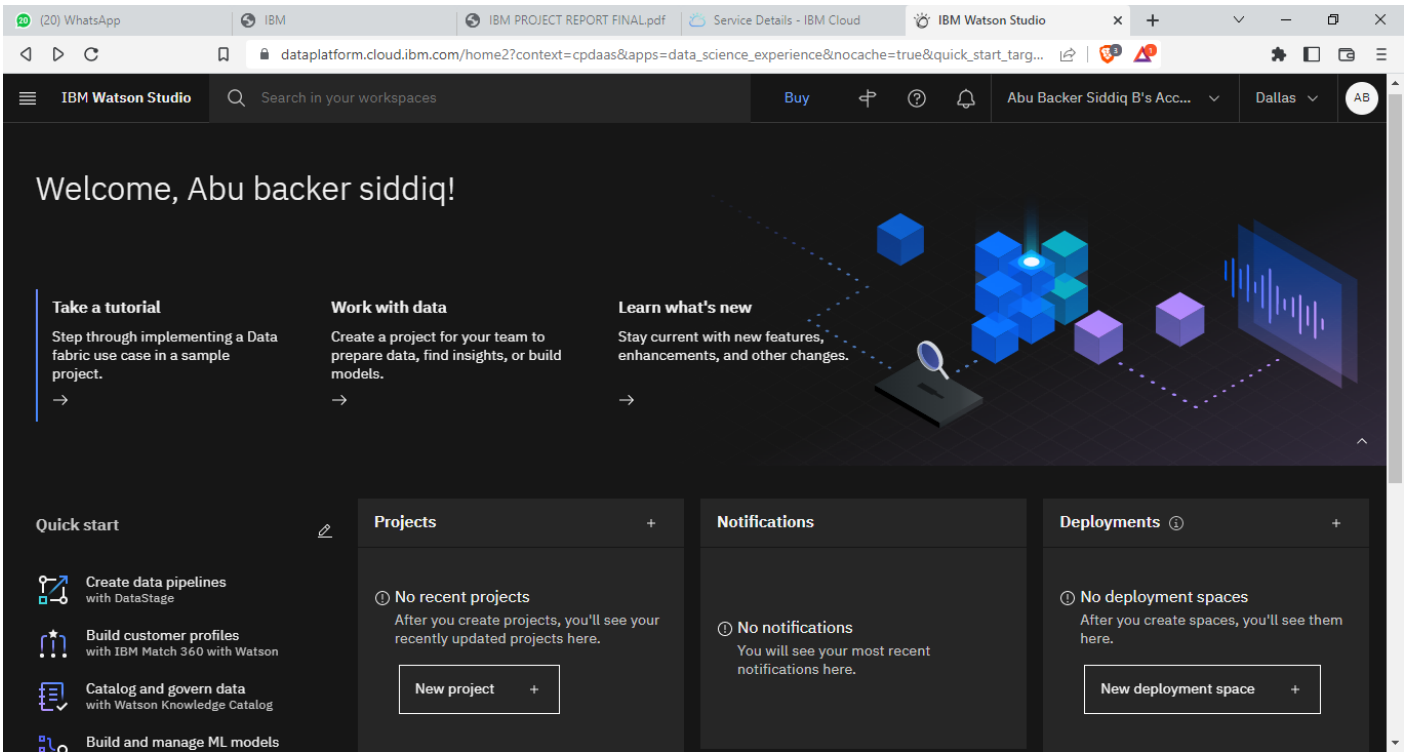
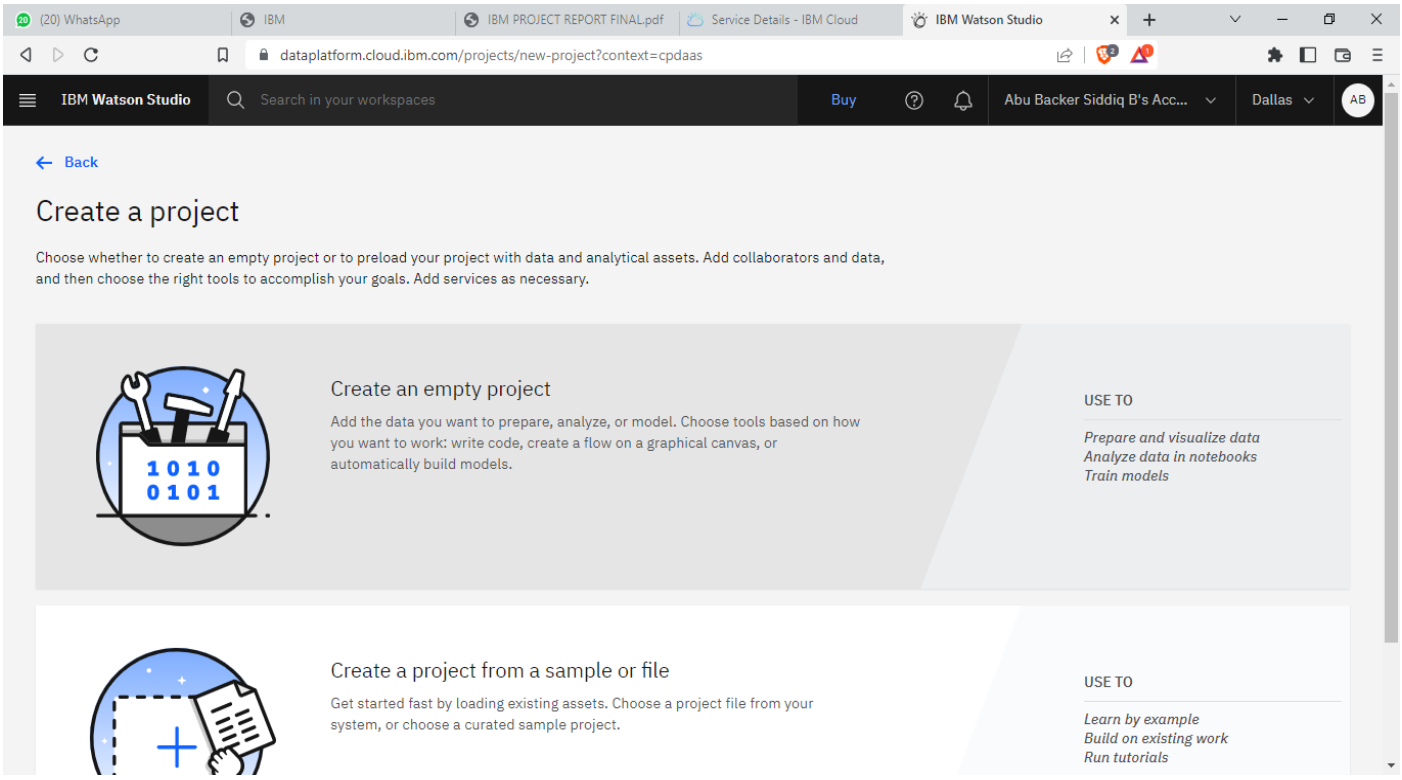
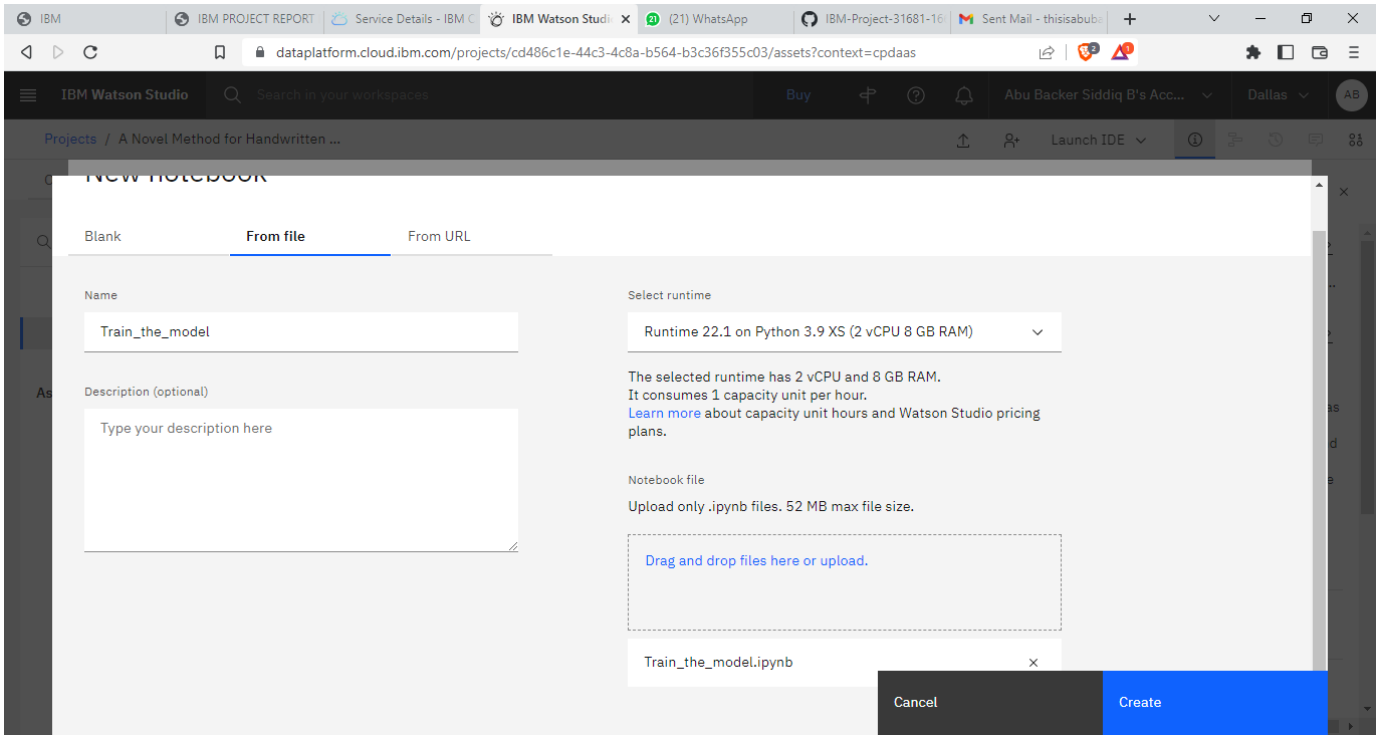
CREATING AN IBM WATSON STUDIO:



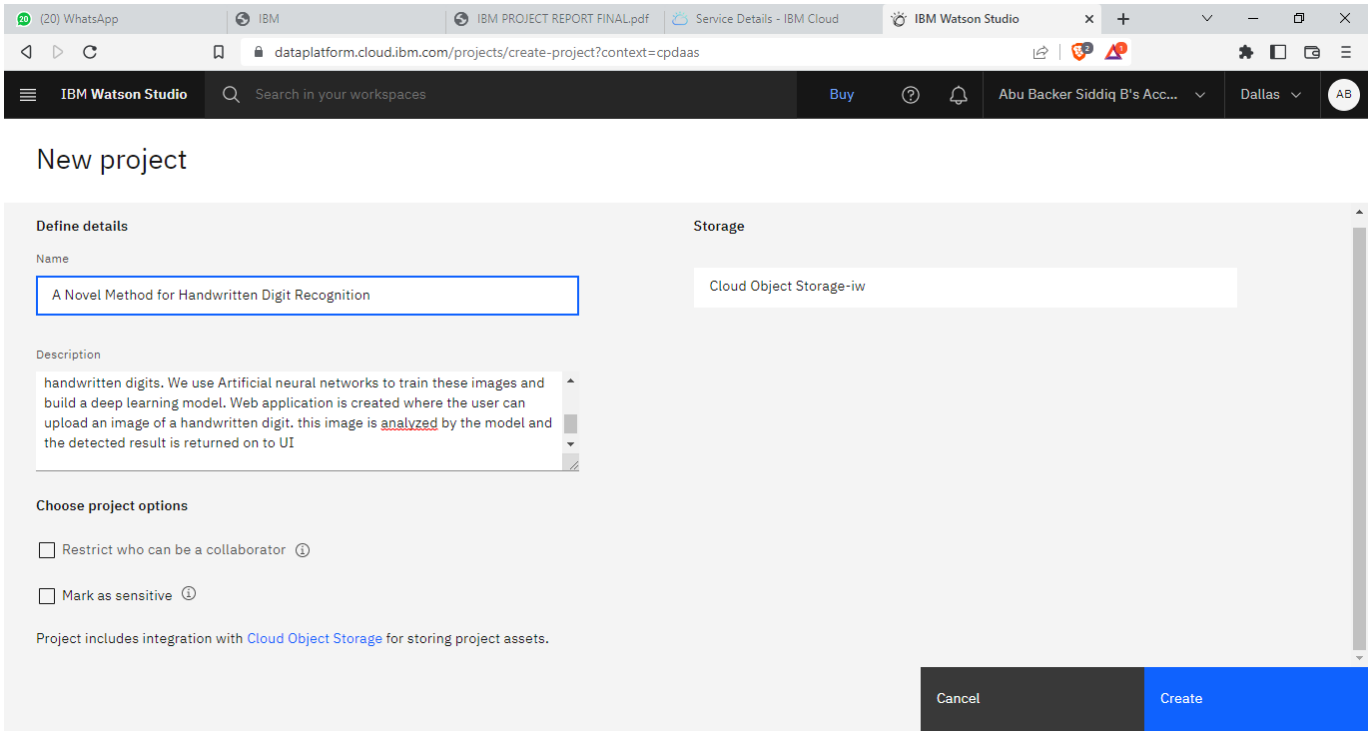
CREATING A PROJECT:



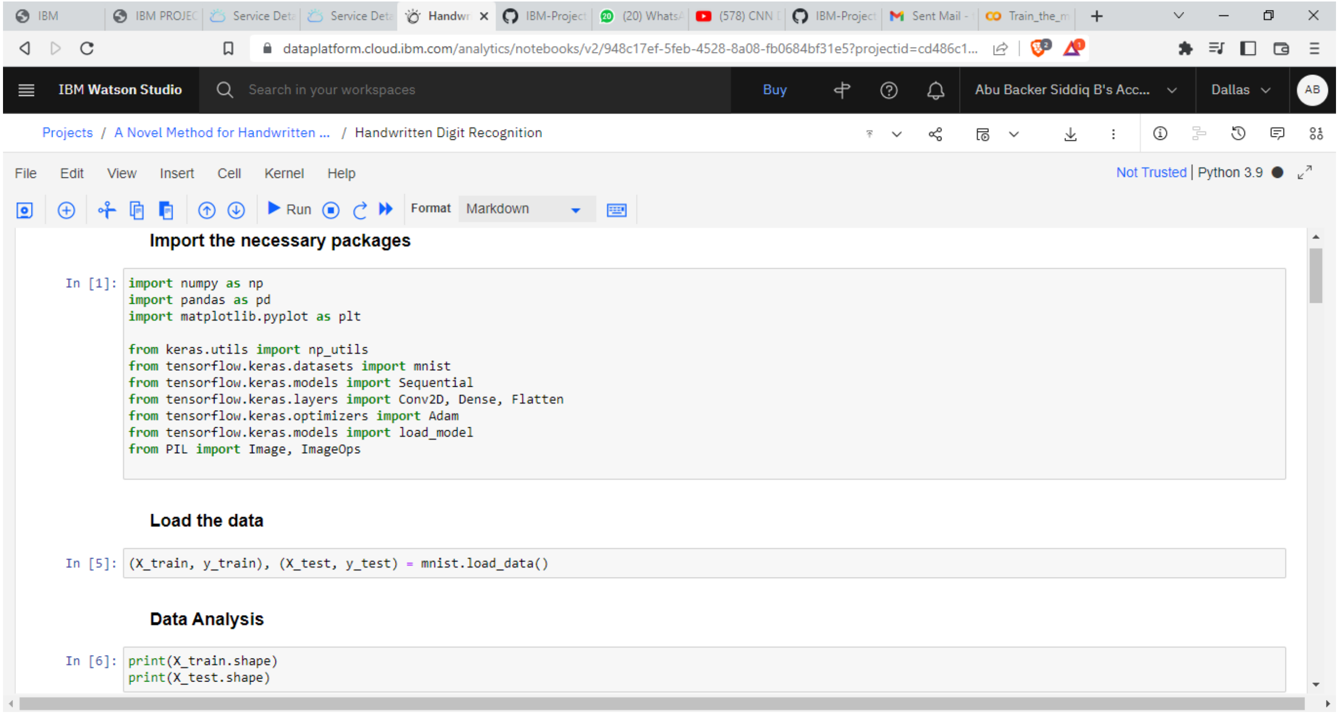
CREATING A NEW EVIRONMENT:

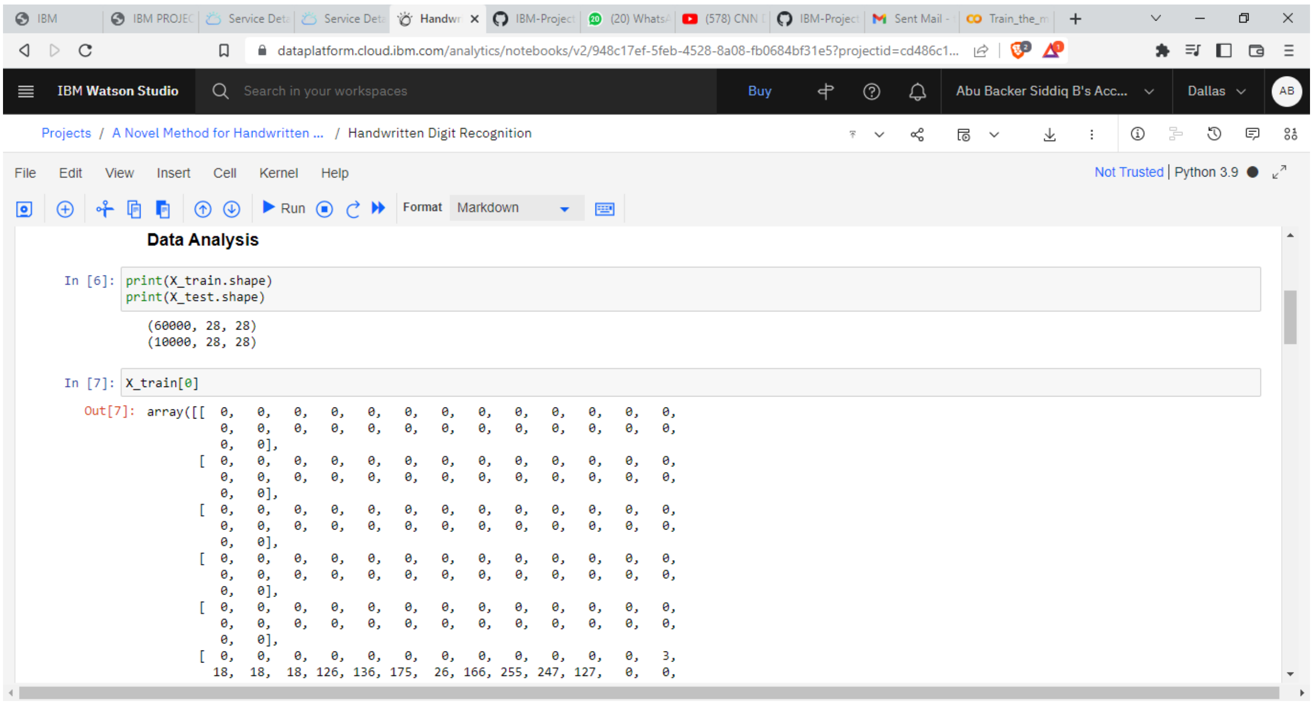


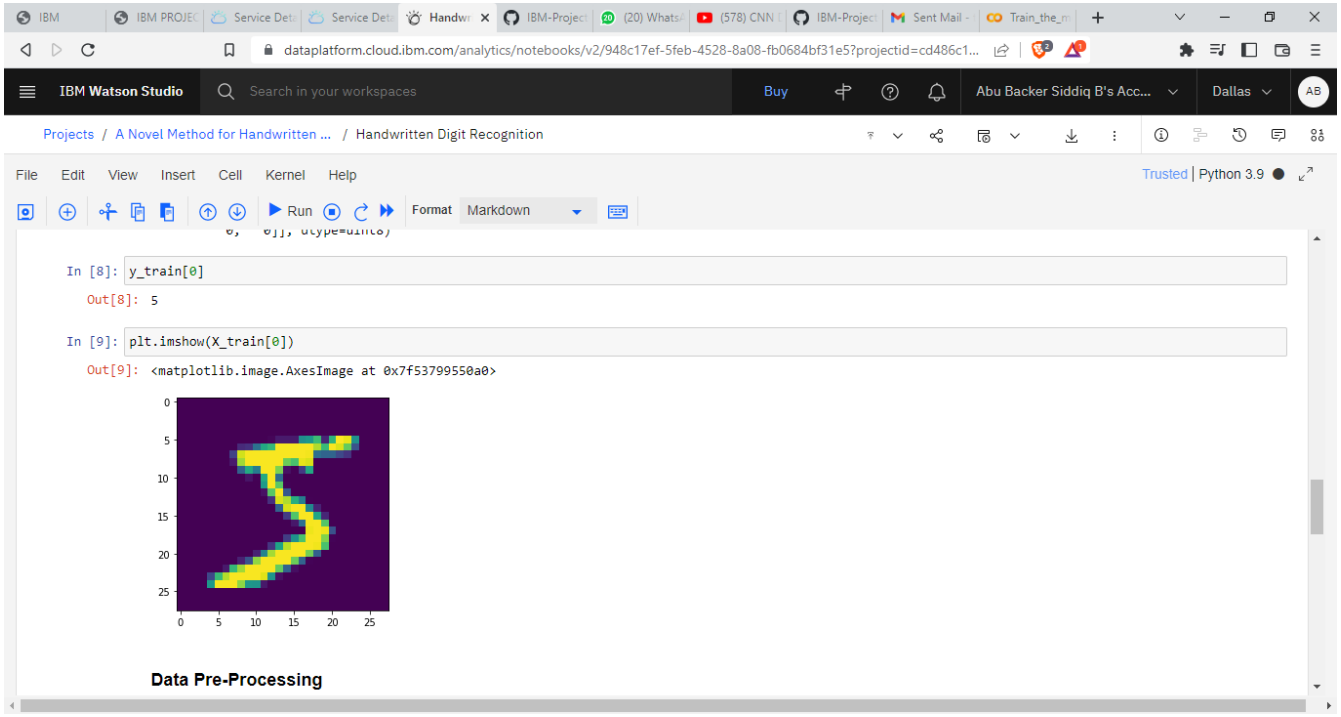
CREATING CLOUD SPACE:

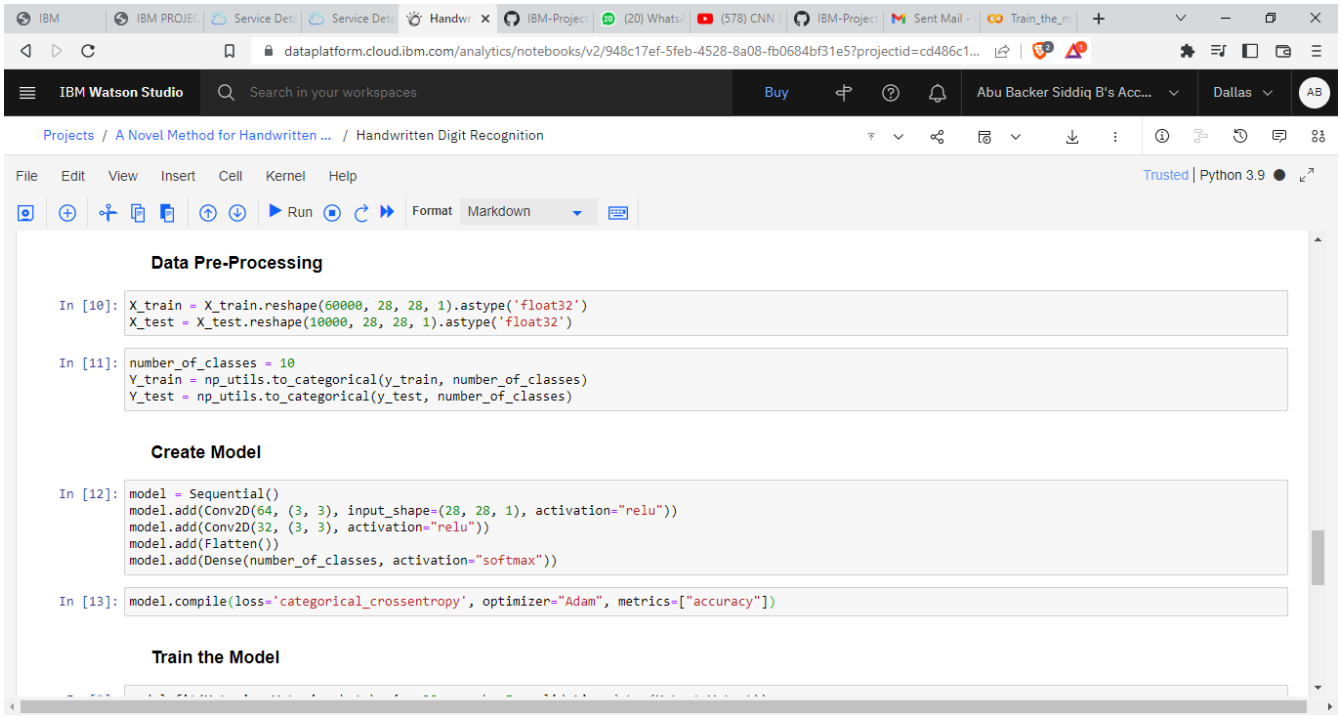


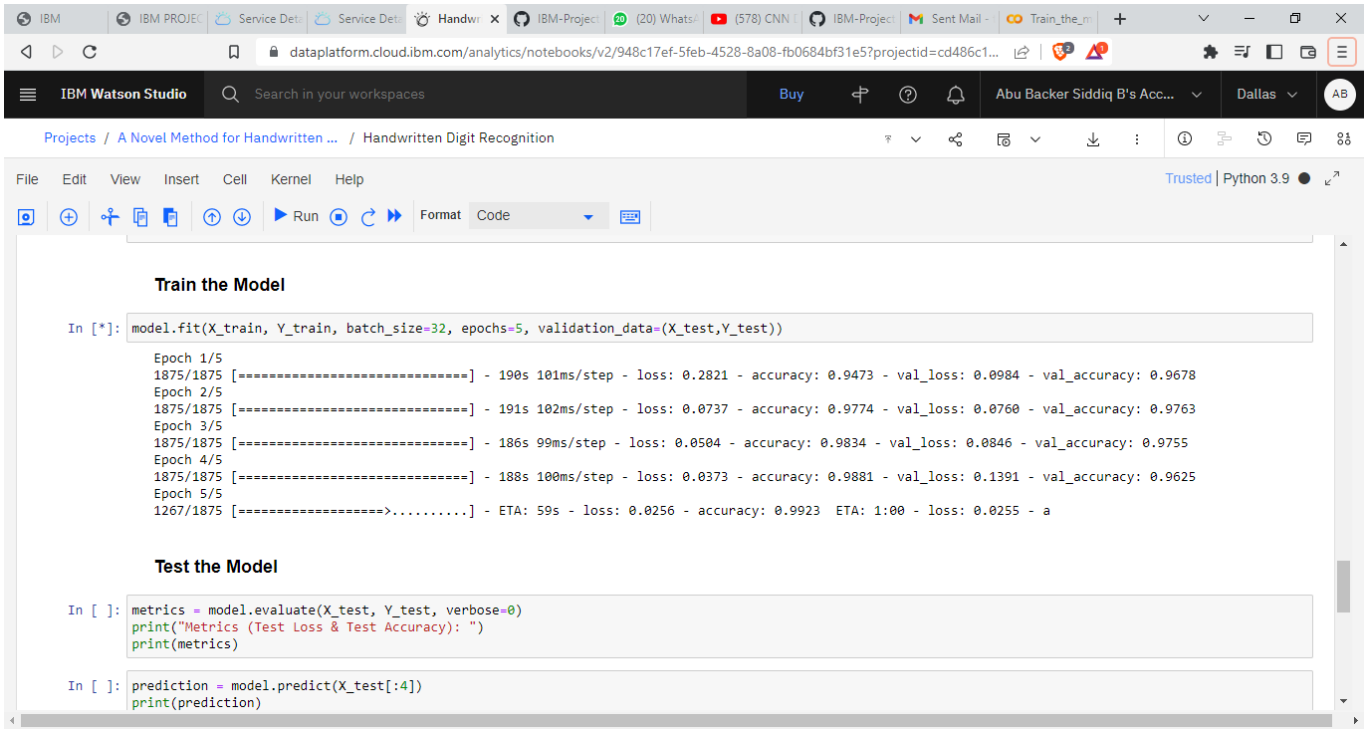
TRAINING THE MODEL ON IBM CLOUD:

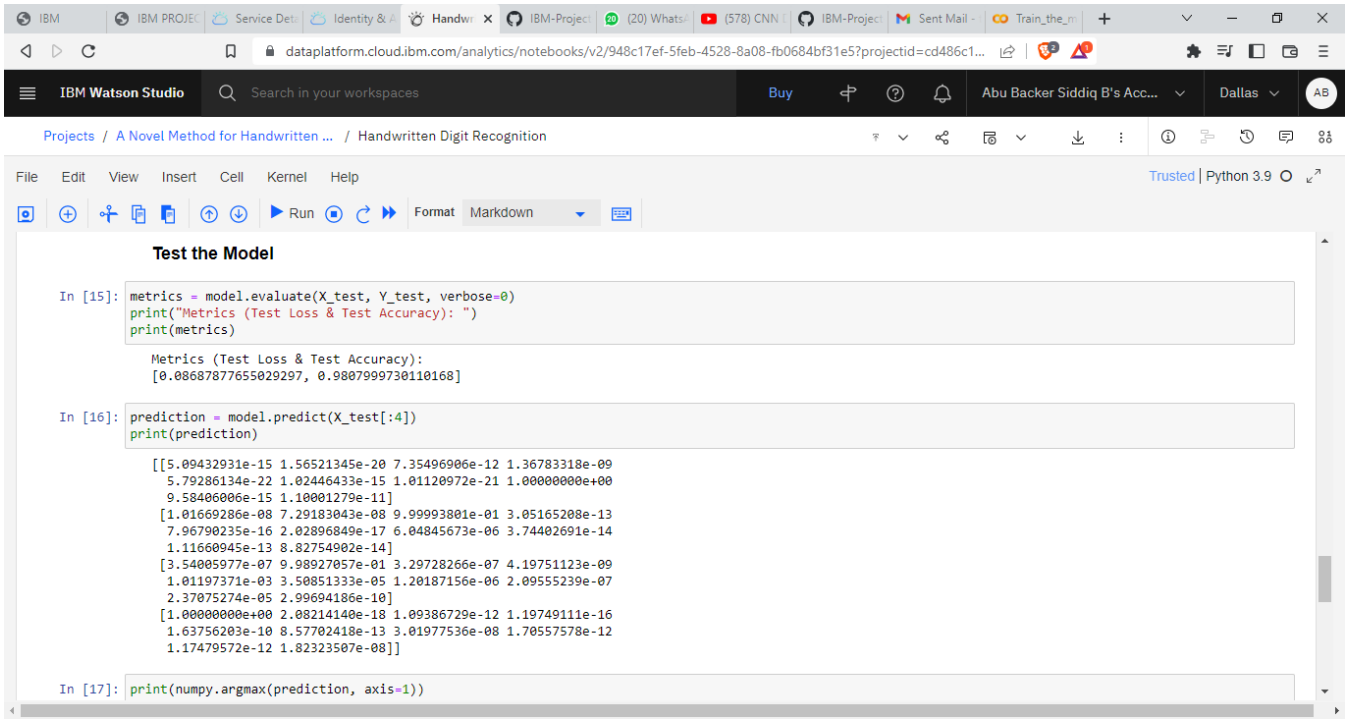


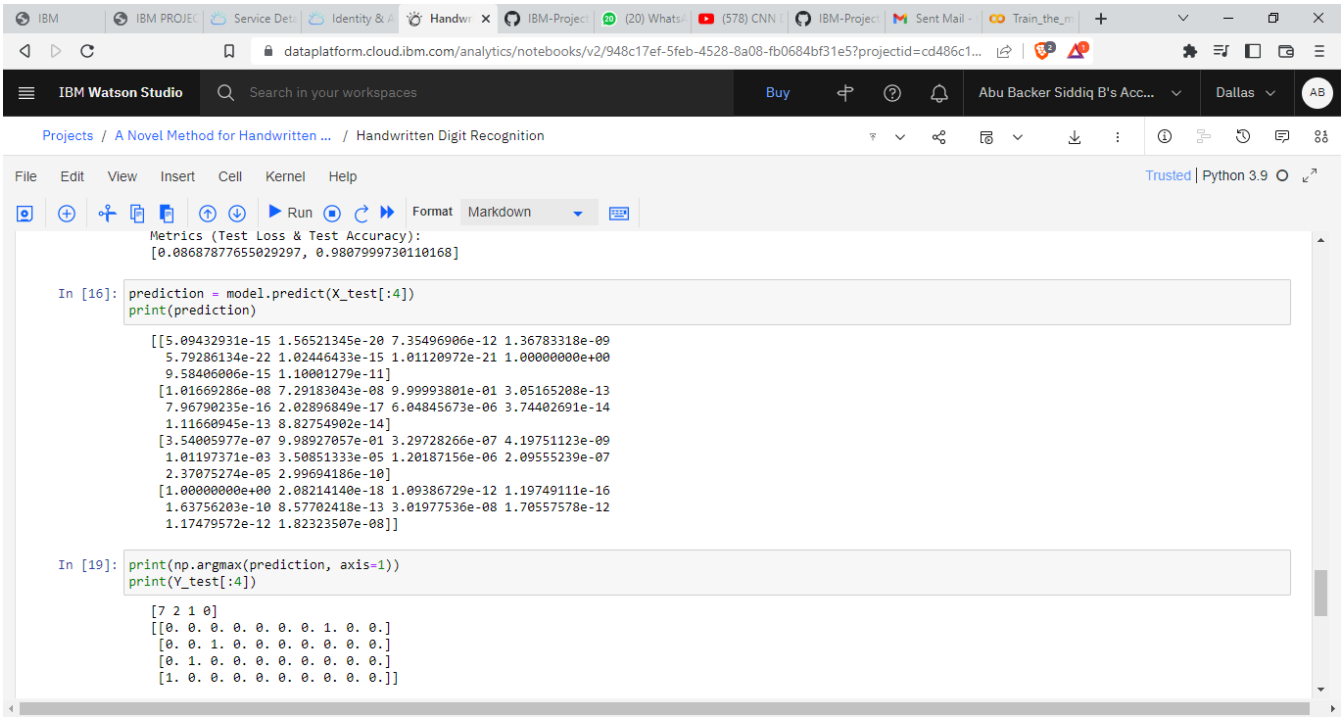


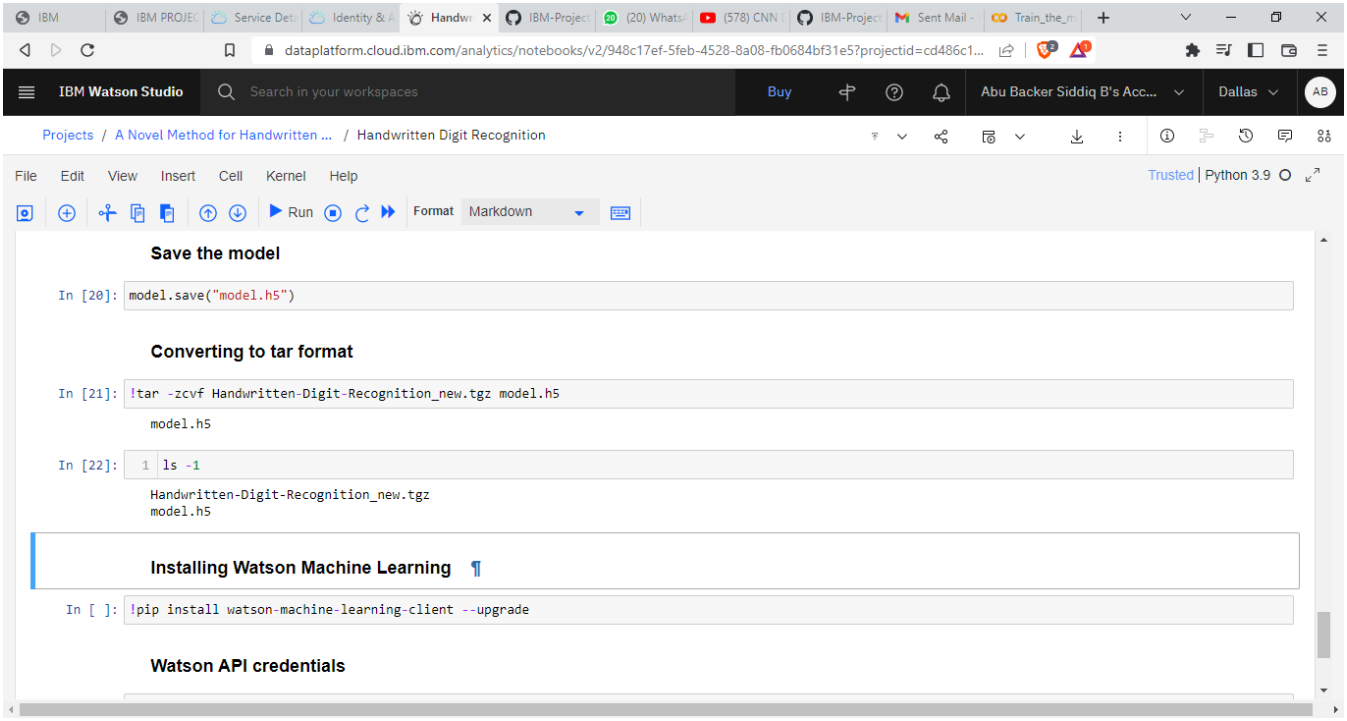

IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run
Format Markdown
In [8]: y_train[0]
Out[8]: 5
In [9]: plt.imshow(X_train[0])
Out[9]: <matplotlib.image.AxesImage at 0x7f53799550a0>
Data Pre-Processing

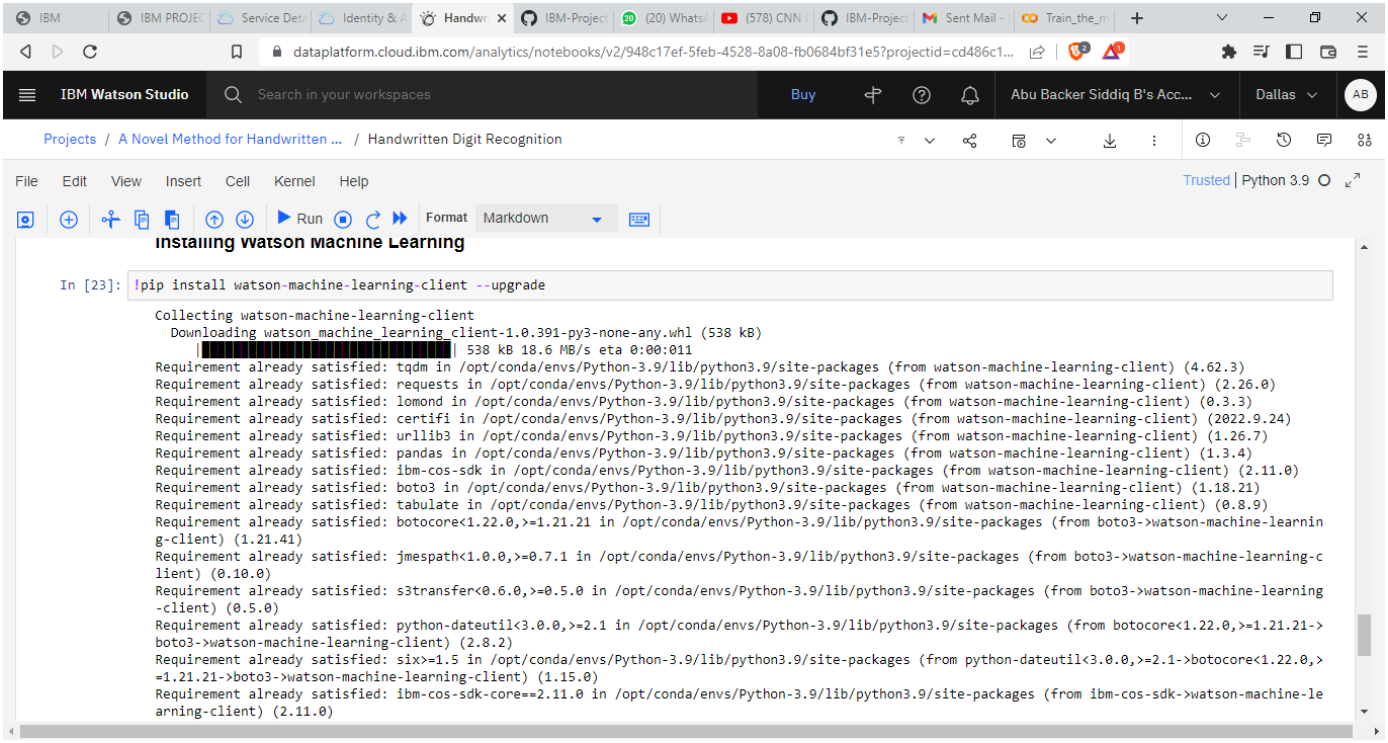

IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run
Format Markdown
Data Pre-Processing
In [10]: X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')
In [11]: number_of_classes = 10
Y_train = np_utils.to_categorical(y_train, number_of_classes)
Y_test = np_utils.to_categorical(y_test, number_of_classes)
Create Model
In [12]: model = Sequential()
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation="relu"))
model.add(Conv2D(32, (3, 3), activation="relu"))
model.add(Flatten())
model.add(Dense(number_of_classes, activation="softmax"))
In [13]: model.compile(loss='categorical_crossentropy', optimizer="Adam", metrics=["accuracy"])
Train the Model


IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run Format Code
Train the Model
In [*]: model.fit(X_train, Y_train, batch_size=32, epochs=5, validation_data=(X_test,Y_test))
Epoch 1/5
1875/1875 [==============================] - 190s 101ms/step - loss: 0.2821 - accuracy: 0.9473 - val_loss: 0.0984 - val_accuracy: 0.9678
Epoch 2/5
1875/1875 [==============================] - 191s 102ms/step - loss: 0.0737 - accuracy: 0.9774 - val_loss: 0.0760 - val_accuracy: 0.9763
Epoch 3/5
1875/1875 [==============================] - 186s 99ms/step - loss: 0.0504 - accuracy: 0.9834 - val_loss: 0.0846 - val_accuracy: 0.9755
Epoch 4/5
1875/1875 [==============================] - 188s 100ms/step - loss: 0.0373 - accuracy: 0.9881 - val_loss: 0.1391 - val_accuracy: 0.9625
Epoch 5/5
1267/1875 [===================>.....] - ETA: 59s - loss: 0.0256 - accuracy: 0.9923 ETA: 1:00 - loss: 0.0255 - a
Test the Model
In [ ]: metrics = model.evaluate(X_test, Y_test, verbose=0)
print("Metrics (Test Loss & Test Accuracy): ")
print(metrics)
In [ ]: prediction = model.predict(X_test[:4])
print(prediction)

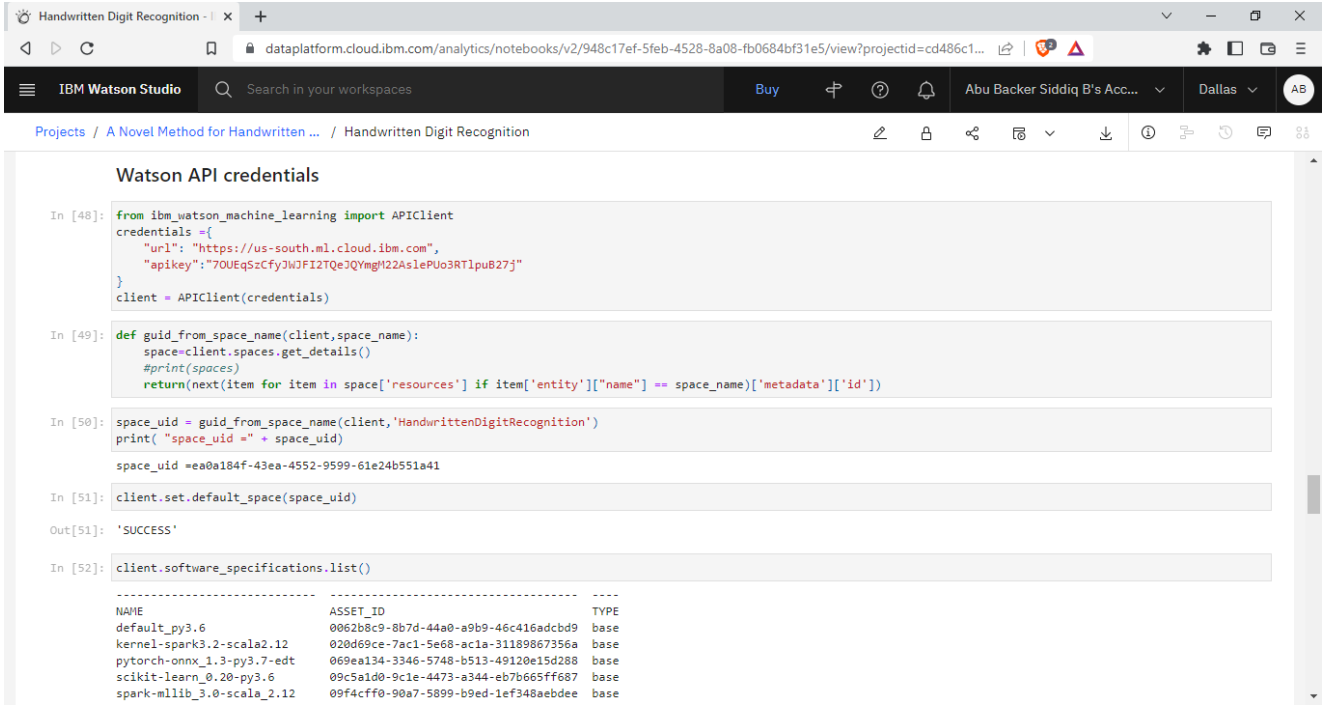

IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run Format Markdown
Test the Model
In [15]: metrics = model.evaluate(X_test, Y_test, verbose=0)
print("Metrics (Test Loss & Test Accuracy): ")
print(metrics)
Metrics (Test Loss & Test Accuracy):
[0.08687877655029297, 0.9807999730110168]
In [16]: prediction = model.predict(X_test[:4])
print(prediction)
[[5.09432931e-15 1.56521345e-20 7.35496906e-12 1.36783318e-09
5.79286134e-22 1.02446433e-15 1.01120972e-21 1.00000000e+00
9.58406006e-15 1.10001279e-11]
[1.01669286e-08 7.29183043e-08 9.99993801e-01 3.05165208e-13
7.96790235e-16 2.02896849e-17 6.04845673e-06 3.74402691e-14
1.11660945e-13 8.82754902e-14]
[3.54005977e-07 9.98927057e-01 3.29728266e-07 4.19751123e-09
1.01197371e-03 3.50851333e-05 1.20187156e-06 2.09555239e-07
2.37075274e-05 2.99694186e-10]
[1.00000000e+00 2.08214140e-18 1.09386729e-12 1.19749111e-16
1.63756203e-10 8.57702418e-13 3.01977536e-08 1.70557578e-12
1.17479572e-12 1.82323507e-08]]
In [17]: print(numpy.argmax(prediction, axis=1))

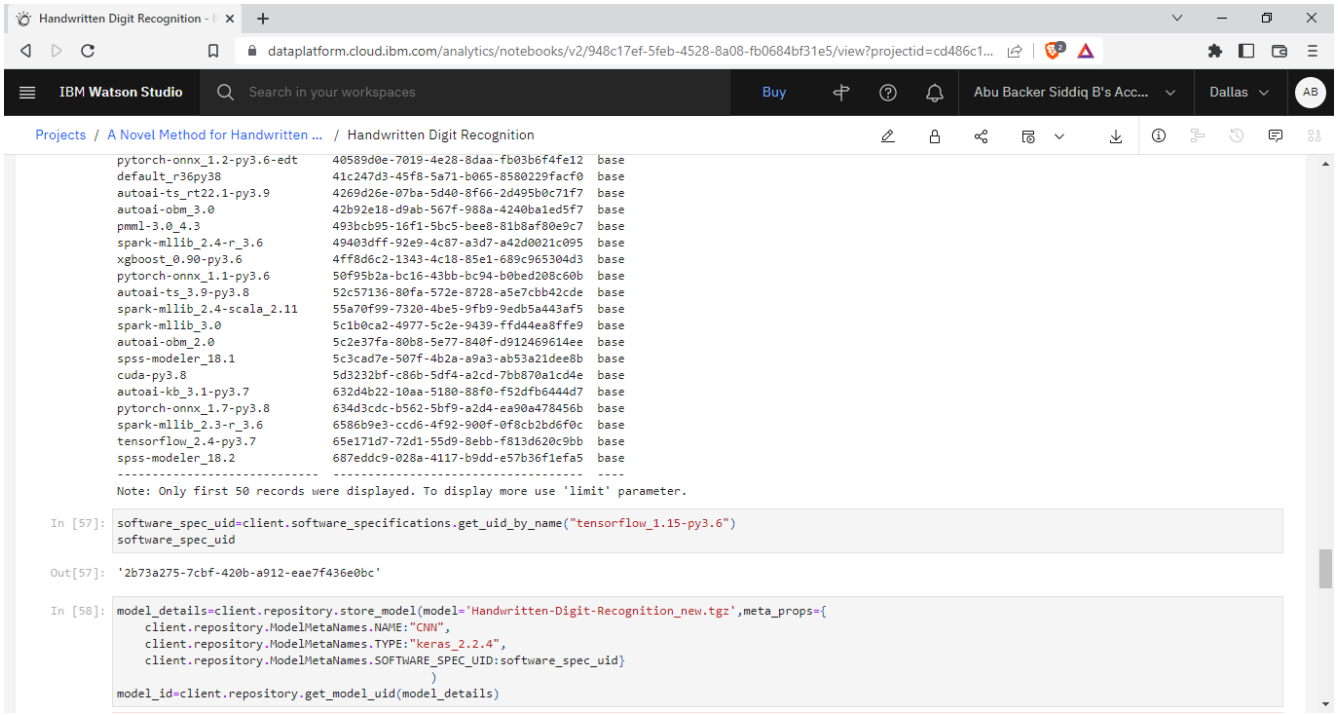

IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run
Format Markdown
Metrics (Test Loss & Test Accuracy):
[0.08687877655029297, 0.9807999730110168]
In [16]:
prediction = model.predict(X_test[:4])
print(prediction)
[[5.09432931e-15 1.56521345e-20 7.35496906e-12 1.36783318e-09
5.79286134e-22 1.02446433e-15 1.01120972e-21 1.00000000e+00
9.58406006e-15 1.10001279e-11]
[1.01669286e-08 7.29183043e-08 9.99993801e-01 3.05165208e-13
7.96790235e-16 2.02896849e-17 6.04845673e-06 3.74402691e-14
1.11660945e-13 8.82754902e-14]
[3.54005977e-07 9.98927057e-01 3.29728266e-07 4.19751123e-09
1.01197371e-03 3.50851333e-05 1.20187156e-06 2.09555239e-07
2.37075274e-05 2.99694186e-10]
[1.00000000e+00 2.08214140e-18 1.09386729e-12 1.19749111e-16
1.63756203e-10 8.57702418e-13 3.01977536e-08 1.70557578e-12
1.17479572e-12 1.82323507e-08]]
In [19]:
print(np.argmax(prediction, axis=1))
print(Y_test[:4])
[7 2 1 0]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
[0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
[0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
[1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]


IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run
Format Markdown
Save the model
In [20]:
model.save("model.h5")
Converting to tar format
In [21]:
!tar -zcvf Handwritten-Digit-Recognition_new.tgz model.h5
model.h5
In [22]:
ls -1
Handwritten-Digit-Recognition_new.tgz
model.h5
Installing Watson Machine Learning ¶
In [ ]:
!pip install watson-machine-learning-client --upgrade
Watson API credentials



WATSON API CONFIGURATION:




Handwritten Digit Recognition -
dataplatform.cloud.ibm.com/analytics/notebooks/v2/948c17ef-5feb-4528-8a08-fb0684bf31e5/view?projectid=cd486c1...
IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
AB
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
pytorch-onnx_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe12 base
default_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 base
autoai-ts_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base
autoai-obm_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base
pmml-3.0_4.3 493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base
spark-mllib_2.4-r_3.6 49403dff-92e9-4c87-a3d7-a42d0021c095 base
xgboost_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base
pytorch-onnx_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base
autoai-ts_3.9-py3.8 52c57136-80fa-572e-8728-a5e7cbb42cde base
spark-mllib_2.4-scala_2.11 55a70f99-7320-4be5-9fb9-9edb5a443af5 base
spark-mllib_3.0 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9 base
autoai-obm_2.0 5c2e37fa-80b8-5e77-840f-d912469614ee base
spss-modeler_18.1 5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b base
cuda-py3.8 5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e base
autoai-kb_3.1-py3.7 632d4b22-10aa-5180-88f0-f52dfb6444d7 base
pytorch-onnx_1.7-py3.8 634d3cdc-b562-5bf9-a2d4-ea90a478456b base
spark-mllib_2.3-r_3.6 6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c base
tensorflow_2.4-py3.7 65e171d7-72d1-55d9-8ebb-f813d620c9bb base
spss-modeler_18.2 687eddc9-028a-4117-b9dd-e57b36f1efa5 base
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
In [57]:
software_spec_uid=client.software_specifications.get_uid_by_name("tensorflow_1.15-py3.6")
software_spec_uid
Out[57]:
'2b73a275-7cbf-420b-a912-eae7f436e0bc'
In [58]:
model_details=client.repository.store_model(model='Handwritten-Digit-Recognition_new.tgz',meta_props={
client.repository.ModelMetaNames.NAME:"CNN",
client.repository.ModelMetaNames.TYPE:"keras_2.2.4",
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid}
)
model_id=client.repository.get_model_uid(model_details)