PROJECT TITLE : UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

TEAM ID : PNT2022TMID42400

TEAM MEMBERS:

S. Praveen kumar, D. Prakash,G.Usha Rajeshwari, R. Sundari

IMPORT STATEMENTS

Import numpy as np

Import pandas as pd

Import seaborn as sns

Import matplotlib.pyplot as plt

%matplotlib inline

LOAD THE DATASET

Import os, types

Import pandas as pd

From botocore.client import Config

Import ibm\_boto3

Def \_\_iter\_\_(self): return 0

# @hidden\_cell

# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.

# You might want to remove those credentials before you share the notebook.

Cos\_client = ibm\_boto3.client(service\_name=’s3’,

Ibm\_api\_key\_id=’T6FhPnWEPrnR91XKAfpiopbqTZ8j-gbLtjakMGexd6v0’,

Ibm\_auth\_endpoint=<https://iam.cloud.ibm.com/oidc/token>,

Config=Config(signature\_version=’oauth’),

Endpoint\_url=’https://s3.private.us.cloud-object-storage.appdomain.cloud’)

Bucket = ‘university-donotdelete-pr-1ijujvyruwxy5c’

Object\_key = ‘Admission\_Predict.csv’

Body = cos\_client.get\_object(Bucket=bucket,Key=object\_key)[‘Body’]

# add missing \_\_iter\_\_ method, so pandas accepts body as file-like object

If not hasattr(body, “\_\_iter\_\_”): body.\_\_iter\_\_ = types.MethodType( \_\_iter\_\_, body )

Data = pd.read\_csv(body)

Data.head()

Serial No. GRE Score TOEFL Score University Rating SOP LOR CGPA Research Chance of Admit

0 1 337 118 4 4.5 4.5 9.65 1 0.92

1 2 324 107 4 4.0 4.5 8.87 1 0.76

2 3 316 104 3 3.0 3.5 8.00 1 0.72

3 4 322 110 3 3.5 2.5 8.67 1 0.80

4 5 314 103 2 2.0 3.0 8.21 0 0.65

Data.drop([“Serial No.”], axis=1, inplace=True)

Data.describe()

GRE Score TOEFL Score University Rating SOP LOR CGPA Research Chance of Admit

Count 400.000000 400.000000 400.000000 400.000000 400.000000 400.000000 400.000000 400.000000

Mean 316.807500 107.410000 3.087500 3.400000 3.452500 8.598925 0.547500 0.724350

Std 11.473646 6.069514 1.143728 1.006869 0.898478 0.596317 0.498362 0.142609

Min 290.000000 92.000000 1.000000 1.000000 1.000000 6.800000 0.000000 0.340000

25% 308.000000 103.000000 2.000000 2.500000 3.000000 8.170000 0.000000 0.640000

50% 317.000000 107.000000 3.000000 3.500000 3.500000 8.610000 1.000000 0.730000

75% 325.000000 112.000000 4.000000 4.000000 4.000000 9.062500 1.000000 0.830000

Max 340.000000 120.000000 5.000000 5.000000 5.000000 9.920000 1.000000 0.970000

Data.info()

RangeIndex: 400 entries, 0 to 399

Data columns (total 8 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 GRE Score 400 non-null int64

1 TOEFL Score 400 non-null int64

2 University Rating 400 non-null int64

3 SOP 400 non-null float64

4 LOR 400 non-null float64

5 CGPA 400 non-null float64

6 Research 400 non-null int64

7 Chance of Admit 400 non-null float64

Dtypes: float64(4), int64(4)

Memory usage: 25.1 KB

Data.isnull().sum()

GRE Score 0

TOEFL Score 0

University Rating 0

SOP 0

LOR 0

CGPA 0

Research 0

Chance of Admit 0

Dtype: int64

VISUALIZATION

Plt.scatter(data[‘GRE Score’],data[‘CGPA’])

Plt.title(‘CGPA vs GRE Score’)

Plt.xlabel(‘GRE Score’)

Plt.ylabel(‘CGPA’)

Plt.show()

Plt.scatter(data[‘CGPA’],data[‘SOP’])

Plt.title(‘SOP for CGPA’)

Plt.xlabel(‘CGPA’)

Plt.ylabel(‘SOP’)

Plt.show()

Data[data.CGPA >= 8.5].plot(kind=’scatter’, x=’GRE Score’, y=’TOEFL Score’,color=”BLUE”)

Plt.xlabel(“GRE Score”)

Plt.ylabel(“TOEFL SCORE”)

Plt.title(“CGPA>=8.5”)

Plt.grid(True)

Plt.show()

Data[“GRE Score”].plot(kind = ‘hist’,bins = 200,figsize = (6,6))

Plt.title(“GRE Scores”)

Plt.xlabel(“GRE Score”)

Plt.ylabel(“Frequency”)

Plt.show()

P = np.array([data[“TOEFL Score”].min(),data[“TOEFL Score”].mean(),data[“TOEFL Score”].max()])

R = [“Worst”,”Average”,”Best”]

Plt.bar(p,r)

Plt.title(“TOEFL Scores”)

Plt.xlabel(“Level”)

Plt.ylabel(“TOEFL Score”)

Plt.show()

G = np.array([data[“GRE Score”].min(),data[“GRE Score”].mean(),data[“GRE Score”].max()])

H = [“Worst”,”Average”,”Best”]

Plt.bar(g,h)

Plt.title(“GRE Scores”)

Plt.xlabel(“Level”)

Plt.ylabel(“GRE Score”)

Plt.show()

Plt.figure(figsize=(10, 10))

Sns.heatmap(data.corr(), annot=True, linewidths=0.05, fmt= ‘.2f’,cmap=”magma”)

Plt.show()

Data.Research.value\_counts()

Sns.countplot(x=”University Rating”,data=data)

Sns.barplot(x=”University Rating”, y=”Chance of Admit “, data=data)

TRAINING AND TESTING SPLIT

X=data.drop([‘Chance of Admit ‘],axis=1) #input data\_set

Y=data[‘Chance of Admit ‘] #output labels

From sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.15)

MODELING AND TRAINING

From sklearn.ensemble import GradientBoostingRegressor

Rgr = GradientBoostingRegressor()

Rgr.fit(X\_train,y\_train)

GradientBoostingRegressor()

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

Rgr.score(X\_test,y\_test)

0.7214021715194154

Y\_predict=rgr.predict(X\_test)

From sklearn.metrics import mean\_squared\_error, r2\_score,mean\_absolute\_error

Import numpy as np

Print(‘Mean Absolute Error:’, mean\_absolute\_error(y\_test, y\_predict))

Print(‘Mean Squared Error:’, mean\_squared\_error(y\_test, y\_predict))

Print(‘Root Mean Squared Error:’, np.sqrt(mean\_squared\_error(y\_test, y\_predict)))

Mean Absolute Error: 0.061115035673946834

Mean Squared Error: 0.007194293635482686

Root Mean Squared Error: 0.08481918200196631

Y\_train = (y\_train>0.5)

Y\_test = (y\_test>0.5)

From sklearn.linear\_model.\_logistic import LogisticRegression

Lore = LogisticRegression(random\_state=0, max\_iter=1000)

Lr = lore.fit(X\_train, y\_train)

Y\_pred = lr.predict(X\_test)

From sklearn.metrics import accuracy\_score, recall\_score, roc\_auc\_score, confusion\_matrix

Print(‘Accuracy Score:’, accuracy\_score(y\_test, y\_pred))

Print(‘Recall Score:’, recall\_score(y\_test, y\_pred))

Print(‘ROC AUC Score:’, roc\_auc\_score(y\_test, y\_pred))

Print(‘Confussion Matrix:\n’, confusion\_matrix(y\_test, y\_pred))

Accuracy Score: 0.95

Recall Score: 1.0

ROC AUC Score: 0.5

Confussion Matrix:

[[ 0 3]

[ 0 57]]

SAVING THE MODEL

Import pickle

Pickle.dump(lr, open(“university.pkl”, “wb”)) #logistic regression model

HOSTING THE MODEL

Import pickle

Lr = pickle.load(open(“university.pkl”, “rb”)) #logistic regression model

Pip install -U ibm-watson-machine-learning

Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.255)

Collecting ibm-watson-machine-learning

Downloading ibm\_watson\_machine\_learning-1.0.256-py3-none-any.whl (1.8 MB)

|████████████████████████████████| 1.8 MB 23.9 MB/s eta 0:00:01

Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.3.4)

Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.8.9)

Requirement already satisfied: ibm-cos-sdk==2.11.\* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.11.0)

Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.3.3)

Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (21.3)

Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.26.7)

Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2022.9.24)

Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.26.0)

Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (4.8.2)

Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.\*->ibm-watson-machine-learning) (2.11.0)

Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.\*->ibm-watson-machine-learning) (2.11.0)

Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.\*->ibm-watson-machine-learning) (0.10.0)

Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.\*->ibm-watson-machine-learning) (2.8.2)

Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-learning) (2021.3)

Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-learning) (1.20.3)

Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.\*->ibm-watson-machine-learning) (1.15.0)

Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm-watson-machine-learning) (2.0.4)

Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm-watson-machine-learning) (3.3)

Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-metadata->ibm-watson-machine-learning) (3.6.0)

Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm-watson-machine-learning) (3.0.4)

Installing collected packages: ibm-watson-machine-learning

Attempting uninstall: ibm-watson-machine-learning

Found existing installation: ibm-watson-machine-learning 1.0.255

Uninstalling ibm-watson-machine-learning-1.0.255:

Successfully uninstalled ibm-watson-machine-learning-1.0.255

Successfully installed ibm-watson-machine-learning-1.0.256

Note: you may need to restart the kernel to use updated packages.

From ibm\_watson\_machine\_learning import APIClient

Import json

Uml\_credentials = {

“url”: <https://us-south.ml.cloud.ibm.com>,

“apikey”: “poJ22ua6BCG9qY33B8fkgnz1bnP1f9DZqUlF9NkBM1bZ”

}

Client = APIClient(uml\_credentials)

Def guid\_from\_space\_name(client, space\_name):

Space = client.spaces.get\_details()

Idr = []

For I in space[‘resources’]:

Idr.append(i[‘metadata’][‘id’])

Return idr

Space\_uid = guid\_from\_space\_name(client, “university”)

Print(space\_uid[0])

4f0253e5-f162-4eec-84ba-72e01fb69ab9

Client.set.default\_space(space\_uid[0])

‘SUCCESS’

Client.software\_specifications.list()

----------------------------- ------------------------------------ ----

NAME ASSET\_ID TYPE

Default\_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base

Kernel-spark3.2-scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base

Pytorch-onnx\_1.3-py3.7-edt 069ea134-3346-5748-b513-49120e15d288 base

Scikit-learn\_0.20-py3.6 09c5a1d0-9c1e-4473-a344-eb7b665ff687 base

Spark-mllib\_3.0-scala\_2.12 09f4cff0-90a7-5899-b9ed-1ef348aebdee base

Pytorch-onnx\_rt22.1-py3.9 0b848dd4-e681-5599-be41-b5f6fccc6471 base

Ai-function\_0.1-py3.6 0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda base

Shiny-r3.6 0e6e79df-875e-4f24-8ae9-62dcc2148306 base

Tensorflow\_2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base

Pytorch\_1.1-py3.6 10ac12d6-6b30-4ccd-8392-3e922c096a92 base

Tensorflow\_1.15-py3.6-ddl 111e41b3-de2d-5422-a4d6-bf776828c4b7 base

Runtime-22.1-py3.9 12b83a17-24d8-5082-900f-0ab31fbfd3cb base

Scikit-learn\_0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base

Default\_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base

Pytorch-onnx\_1.3-py3.6 1bc6029a-cc97-56da-b8e0-39c3880dbbe7 base

Kernel-spark3.3-r3.6 1c9e5454-f216-59dd-a20e-474a5cdf5988 base

Pytorch-onnx\_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f base

Tensorflow\_2.1-py3.6 1eb25b84-d6ed-5dde-b6a5-3fbdf1665666 base

Spark-mllib\_3.2 20047f72-0a98-58c7-9ff5-a77b012eb8f5 base

Tensorflow\_2.4-py3.8-horovod 217c16f6-178f-56bf-824a-b19f20564c49 base

Runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-da66306ce658 base

Do\_py3.8 295addb5-9ef9-547e-9bf4-92ae3563e720 base

Autoai-ts\_3.8-py3.8 2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base

Tensorflow\_1.15-py3.6 2b73a275-7cbf-420b-a912-eae7f436e0bc base

Kernel-spark3.3-py3.9 2b7961e2-e3b1-5a8c-a491-482c8368839a base

Pytorch\_1.2-py3.6 2c8ef57d-2687-4b7d-acce-01f94976dac1 base

Spark-mllib\_2.3 2e51f700-bca0-4b0d-88dc-5c6791338875 base

Pytorch-onnx\_1.1-py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67e base

Spark-mllib\_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base

Spark-mllib\_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base

Xgboost\_0.82-py3.6 39e31acd-5f30-41dc-ae44-60233c80306e base

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.

Import sklearn

Sklearn.\_\_version\_\_

‘1.0.2’

MODEL\_NAME = ‘university’

DEPLOYMENT\_NAME = ‘uni’

DEMO\_MODEL = lr

Software\_spec\_uid = client.software\_specifications.get\_id\_by\_name(‘runtime-22.1-py3.9’)

Model\_props = {

Client.repository.ModelMetaNames.NAME: MODEL\_NAME,

Client.repository.ModelMetaNames.TYPE: ‘scikit-learn\_1.0 ‘,

Client.repository.ModelMetaNames.SOFTWARE\_SPEC\_UID: software\_spec\_uid

}

Model\_details = client.repository.store\_model(

Model = DEMO\_MODEL,

Meta\_props = model\_props,

Training\_data = X\_train,

Training\_target = y\_train

)

Model\_details

{‘entity’: {‘hybrid\_pipeline\_software\_specs’: [],

‘label\_column’: ‘Chance of Admit ‘,

‘schemas’: {‘input’: [{‘fields’: [{‘name’: ‘GRE Score’, ‘type’: ‘int64’},

{‘name’: ‘TOEFL Score’, ‘type’: ‘int64’},

{‘name’: ‘University Rating’, ‘type’: ‘int64’},

{‘name’: ‘SOP’, ‘type’: ‘float64’},

{‘name’: ‘LOR ‘, ‘type’: ‘float64’},

{‘name’: ‘CGPA’, ‘type’: ‘float64’},

{‘name’: ‘Research’, ‘type’: ‘int64’}],

‘id’: ‘1’,

‘type’: ‘struct’}],

‘output’: []},

‘software\_spec’: {‘id’: ‘12b83a17-24d8-5082-900f-0ab31fbfd3cb’,

‘name’: ‘runtime-22.1-py3.9’},

‘type’: ‘scikit-learn\_1.0’},

‘metadata’: {‘created\_at’: ‘2022-11-02T16:24:18.441Z’,

‘id’: ‘b3c03ac6-a17c-4eb2-b3de-a7e8b3a9e768’,

‘modified\_at’: ‘2022-11-02T16:24:21.210Z’,

‘name’: ‘university’,

‘owner’: ‘IBMid-6630041JHH’,

‘resource\_key’: ‘6da09709-1e17-4c03-8755-1b922005c52f’,

‘space\_id’: ‘4f0253e5-f162-4eec-84ba-72e01fb69ab9’},

‘system’: {‘warnings’: []}}

Model\_id = client.repository.get\_model\_id(model\_details)

Model\_id

‘b3c03ac6-a17c-4eb2-b3de-a7e8b3a9e768’

Deployment\_props = {

Client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT\_NAME,

Client.deployments.ConfigurationMetaNames.ONLINE: {}

}

Deployment = client.deployments.create(

Artifact\_uid = model\_id,

Meta\_props = deployment\_props

)

Synchronous deployment creation for uid: ‘b3c03ac6-a17c-4eb2-b3de-a7e8b3a9e768’ started

Initializing

Note: online\_url is deprecated and will be removed in a future release. Use serving\_urls instead.

Ready

Successfully finished deployment creation, deployment\_uid=’40623eea-21ed-4cb2-8736-7f593828a8b4’