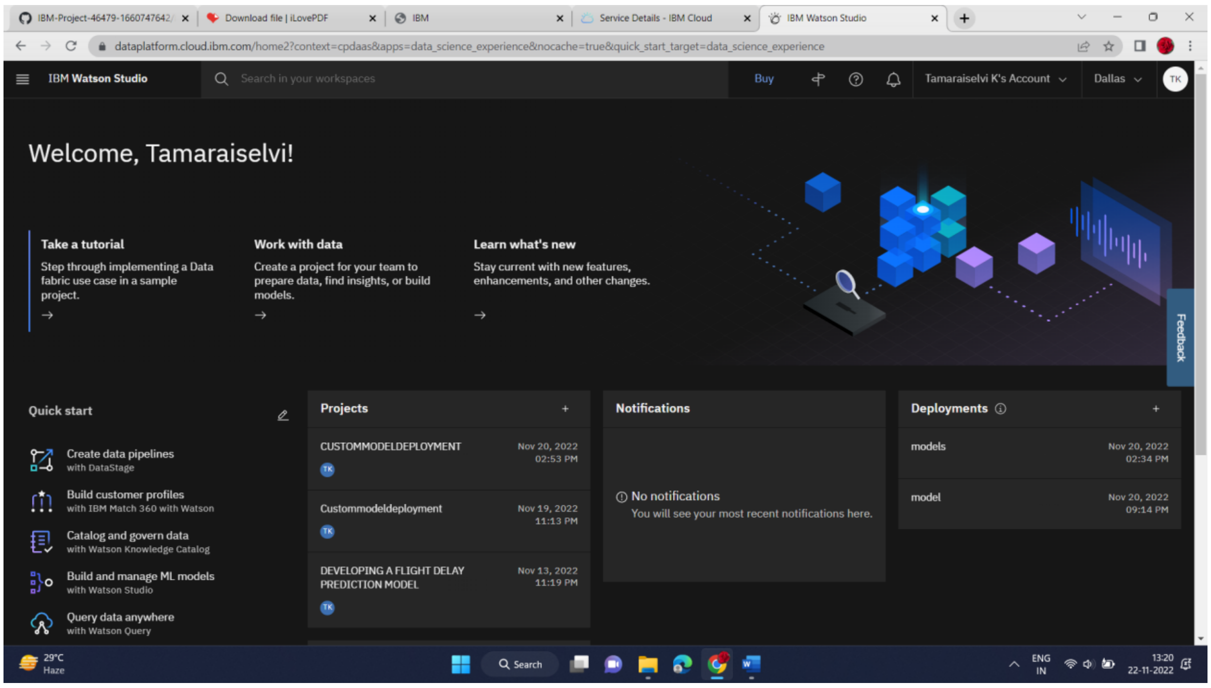
TRAIN THE MODEL ON IBM

DATE	01 NOVEMBER 2022
TEAM ID	PNT2022TMID17753
PROJECT NAME	DEVELOPING A FLIGHT DELAY PREDICTION MODEL BY USING MACHINE LEARNING


IBM-Project-46479-1660747642/
Download file | iLovePDF
IBM
Service Details - IBM Cloud
IBM Watson Studio
dataplatform.cloud.ibm.com/home2?context=cpdaas&apps=data_science_experience&nocache=true&quick_start_target=data_science_experience
IBM Watson Studio
Search in your workspaces
Buy
Tamaraiselvi K's Account
Dallas
TK
Feedback
29°C
Haze
Search
ENG
IN
13:20
22-11-2022

Welcome, Tamaraiselvi!

Take a tutorial
Step through implementing a Data fabric use case in a sample project.

Work with data
Create a project for your team to prepare data, find insights, or build models.

Learn what's new
Stay current with new features, enhancements, and other changes.

Quick start
- Create data pipelines with DataStage
- Build customer profiles with IBM Match 360 with Watson
- Catalog and govern data with Watson Knowledge Catalog
- Build and manage ML models with Watson Studio
- Query data anywhere with Watson Query

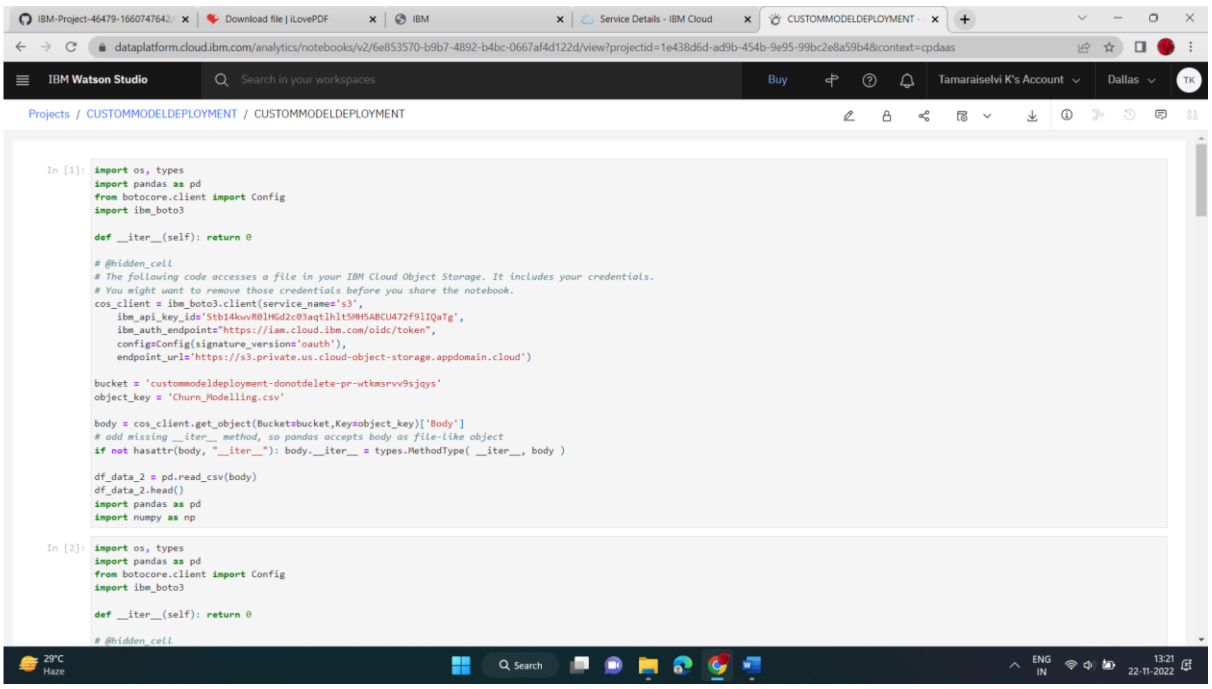
Projects

CUSTOMMODELDEPLOYMENT	Nov 20, 2022 02:53 PM
Custommodeldeployment	Nov 19, 2022 11:13 PM
DEVELOPING A FLIGHT DELAY PREDICTION MODEL	Nov 13, 2022 11:19 PM

Notifications
No notifications
You will see your most recent notifications here.

Deployments

models	Nov 20, 2022 02:34 PM
model	Nov 20, 2022 09:14 PM



```
In [1]: 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='Stb14kwvR0lHGd2c03aqtlhlt5MH5ABCU472f9lIQaTg',
            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 = 'custommodeldeployment-donotdelete-pr-wtkmsrvv9sjqys'
        object_key = 'Churn_Modelling.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 )

        df_data_2 = pd.read_csv(body)
        df_data_2.head()
        import pandas as pd
        import numpy as np

In [2]: import os, types
        import pandas as pd
        from botocore.client import Config
        import ibm_boto3

        def __iter__(self): return 0

        # @hidden_cell
```

```
import pandas as pd
import numpy as np
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0
data = pd.read_csv(body)
data.head()
type(data)
data.head(10)
data['Gender'],fillna(data['Gender'],mode()[0],inplace = True)
data['Age'],fillna(data['Age'],mean(),inplace = True)
data['CreditScore'],fillna(data['CreditScore'],median(),inplace = True)
data.isnull().any()
data.head(10)
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data["Geography"] = le.fit_transform(data["Geography"])
data["Gender"] = le.fit_transform(data["Gender"])
x = data.iloc[:,3:13].values
y = data.iloc[:,13].values
data
x
from sklearn.preprocessing import OneHotEncoder
one = OneHotEncoder()
z = one.fit_transform(x[:,1:2]).toarray()
x = np.delete(x,1,axis = 1)
x = np.concatenate((z,x),axis = 1)
z
x
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.2,random_state = 42)
x.shape
x_train.shape
x_test.shape
from sklearn.model_selection import train_test_split
x1 = [1,2,3,4,5,6,7,8,9,10]
y1 = [1,0,1,0,1,0,1,0,1,0]
for l in range(5):
    x_train1,x_test1,y_train1,y_test1 = train_test_split(x1,y1,test_size = 0.2,random_state = 2)
    print(x_train1, "with random state")
for l in range(5):
    x_train1,x_test1,y_train1,y_test1 = train_test_split(x1,y1,test_size = 0.2)
    print(x_train1,"without random state")
from sklearn.ensemble import RandomForestClassifier
```

```
forest_reg = RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=42)
forest_reg.fit(x_train,y_train)
x_train[0]
from ibm_watson_machine_learning import APIClient
wml_credentials = {
                  "url":"https://us-south.ml.cloud.ibm.com",
                  "apikey":"Wv3aXu7-agz7OrqIjR-btR10N_5Zncy7TqDdiM55xfyN"
                }
client = APIClient(wml_credentials)
def guid_from_space_name(client,space_name):
    space = client.spaces.get_details()
    return(next(item for item in space['resources'] if item['entity']["name"] == space_name)['metadata']['id'])
space_uid = guid_from_space_name(client, "models")
print("Space UID = " + space_uid)
client.set.default_space(space_uid)
client.software_specifications.list()
  software_spec_uid = client.software_specifications.get_uid_by_name("default_ py3.7")
software_spec_uid
df_data_1 = pd.read_csv(body)
df_data_1.head()
etaNames.NAME:"Churn_modelling",
client.repository.ModelMetaNames.TYPE:"scikit-learn_0.22",
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid}
                                          )
    model_id = client.repository.get_model_uid(model_details)
model_id
x_train[0]
forest_reg.predict([[1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 6.8600000e+02,
       1.0000000e+00, 3.2000000e+01, 6.0000000e+00, 0.0000000e+00,
       2.0000000e+00, 1.0000000e+00, 1.0000000e+00, 1.7909326e+05]])
```