Sprint-3 objectives:
-Train the model
-Save the model
-Deploy the model in ibm

Refer ipynb for the model.

Model:

Split train test data

In [40]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x1,y2,test_size=0.2,random_state=42)
```

In [41]:
```
x_train.shape,x_test.shape
```

Out[41]: ((9856, 12), (2464, 12))

Random Forest

In [43]:
```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(n_estimators=50,random_state=42)
rf.fit(x_train,y_train)
rf.score(x_test,y_test)
```

Out[43]: 0.8368506493506493

Evaluation of Random Forest

In [53]:
```
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
pred=rf.predict(x_test)
cm=confusion_matrix(y_test, pred)
#plt.figure(figsize=(10,6))
#sns.heatmap(cm, annot=True,cmap='winter',linewidths=0.3, linecolor='black',annot_kws={"size": 20})
TP=cm[0][0]
TN=cm[1][1]
FN=cm[1][0]
FP=cm[0][1]
#print(round(accuracy_score(prediction3,y_test)*100,2))
#print('Testing Accuracy for knn',(TP+TN)/(TP+TN+FN+FP))
print('Testing Sensitivity for Random Forest',(TP/(TP+FN)))
print('Testing Specificity for Random Forest',(TN/(TN+FP)))
print('Testing Precision for Random Forest',(TP/(TP+FP)))
print('Testing accuracy for Random Forest',accuracy_score(y_test, pred))
```

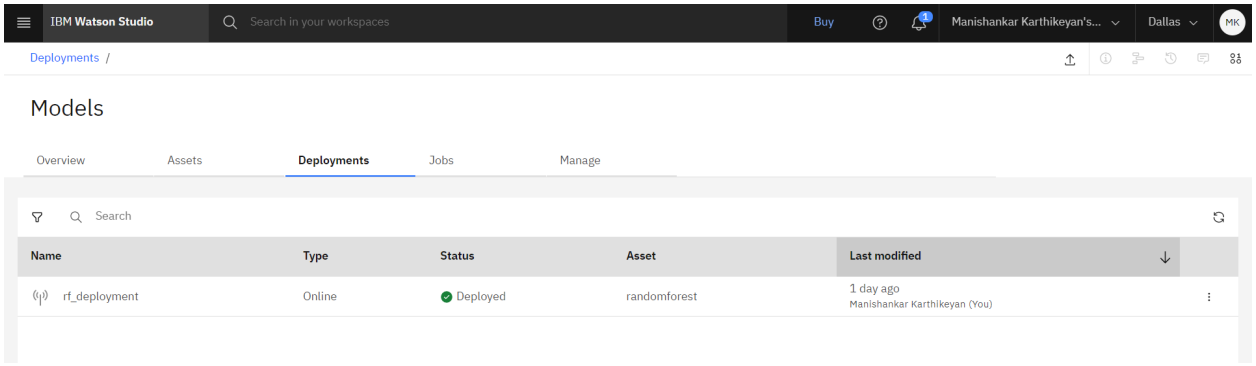
```
Testing Sensitivity for Random Forest 0.9360230547550432
Testing Specificity for Random Forest 0.8716577540106952
Testing Precision for Random Forest 0.9854368932038835
Testing accuracy for Random Forest 0.8368506493506493
```

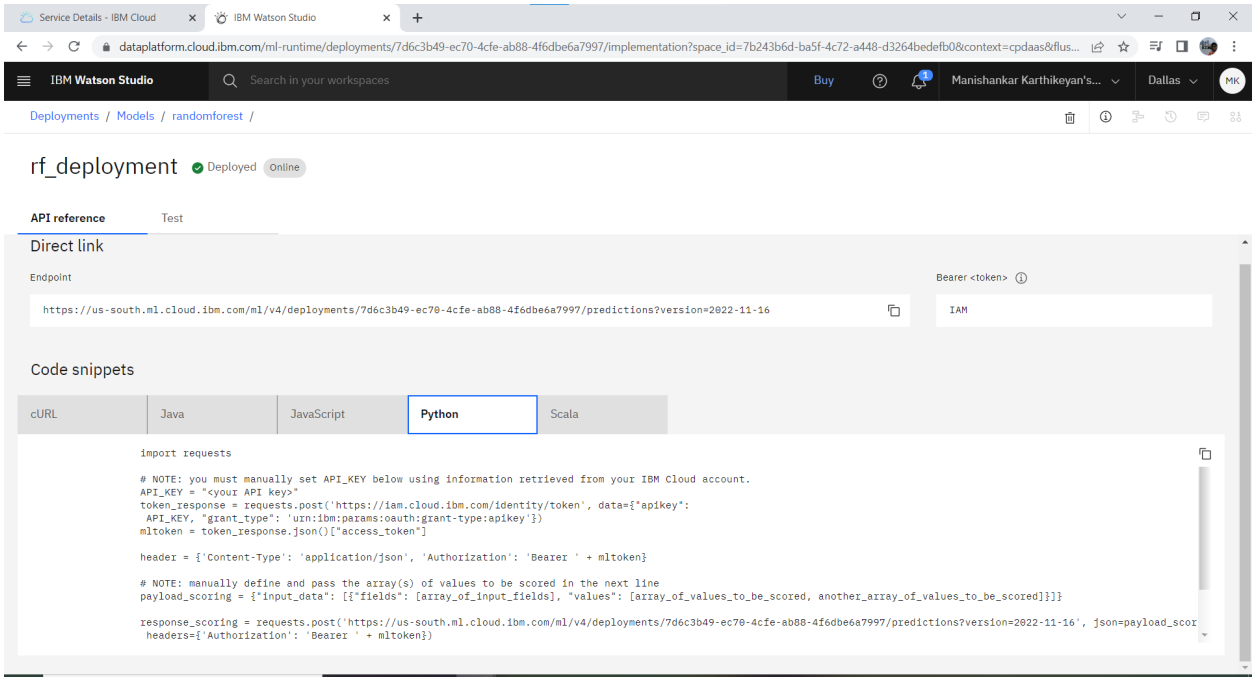
In [54]:
```
print(classification_report(y_test,pred))#RandomForest
```

```
              precision    recall  f1-score   support

         0.0       0.86      0.96      0.91      1683
         1.0       0.73      0.53      0.61       308
         2.0       0.67      0.49      0.57       288
         3.0       0.88      0.65      0.75        55
         4.0       0.92      0.75      0.82       130

    accuracy                           0.84      2464
   macro avg       0.81      0.68      0.73      2464
weighted avg       0.83      0.84      0.83      2464
```

In [57]:
```
import pickle
pickle.dump(rf,open("rfmodel.pkl",'wb'))
```

Deployment:


IBM Watson Studio
Search in your workspaces
Buy
Manishankar Karthikeyan's...
Dallas
MK
Deployments /
Models
Overview
Assets
Deployments
Jobs
Manage
Search
Name
Type
Status
Asset
Last modified
rf_deployment
Online
Deployed
randomforest
1 day ago
Manishankar Karthikeyan (You)


Service Details - IBM Cloud
IBM Watson Studio
dataplatform.cloud.ibm.com/ml-runtime/deployments/7d6c3b49-ec70-4cfe-ab88-4f6dbe6a7997/implementation?space_id=7b243b6d-ba5f-4c72-a448-d3264bedefb0&context=cpdaas&flus...
IBM Watson Studio
Search in your workspaces
Buy
Manishankar Karthikeyan's...
Dallas
MK
Deployments / Models / randomforest /
rf_deployment
Deployed
Online
API reference
Test
Direct link
Endpoint
https://us-south.ml.cloud.ibm.com/ml/v4/deployments/7d6c3b49-ec70-4cfe-ab88-4f6dbe6a7997/predictions?version=2022-11-16
Bearer <token>
IAM
Code snippets
cURL
Java
JavaScript
Python
Scala
import requests
# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "<your API key>"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"fields": [array_of_input_fields], "values": [array_of_values_to_be_scored, another_array_of_values_to_be_scored]}]}
response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/7d6c3b49-ec70-4cfe-ab88-4f6dbe6a7997/predictions?version=2022-11-16', json=payload_scor
headers={'Authorization': 'Bearer ' + mltoken})