

SPRINT – 2 PROJECT DOCUMENT

Date	5 November 2022
Team ID	PNT2022TMID54451
Project Name	Developing a Flight Delay Prediction Model using Machine Learning

DEVELOPMENT PHASE:

SPRINT-2:

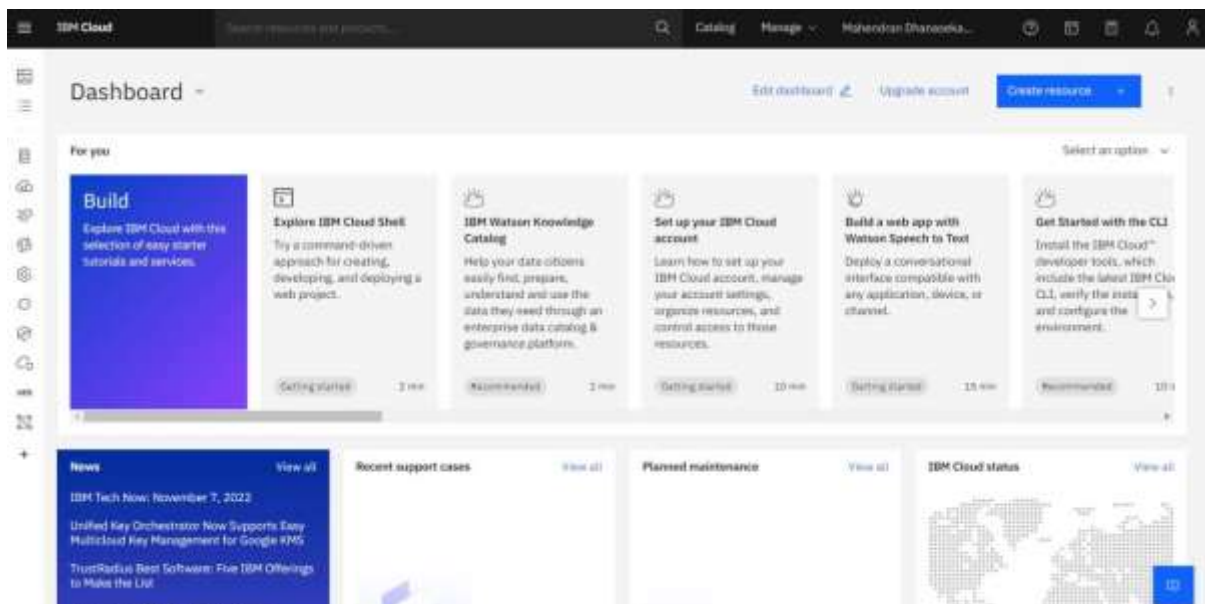
- Creating IBM cloud account & Required Resources
- Deploy our model in IBM Watson
- Creating Dashboard using HTML/CSS
- Create web app and Hosting in falsk
- Testing web app

Creating IBM cloud account & Required Resources:

Creating IBM cloud account:

Frist, need to create IBM Cloud account by using SI Mail Id and SI Password which is provided by IBM in profile.

Below dashboard of an account after created,



Creating IBM Cloud Required Resources:

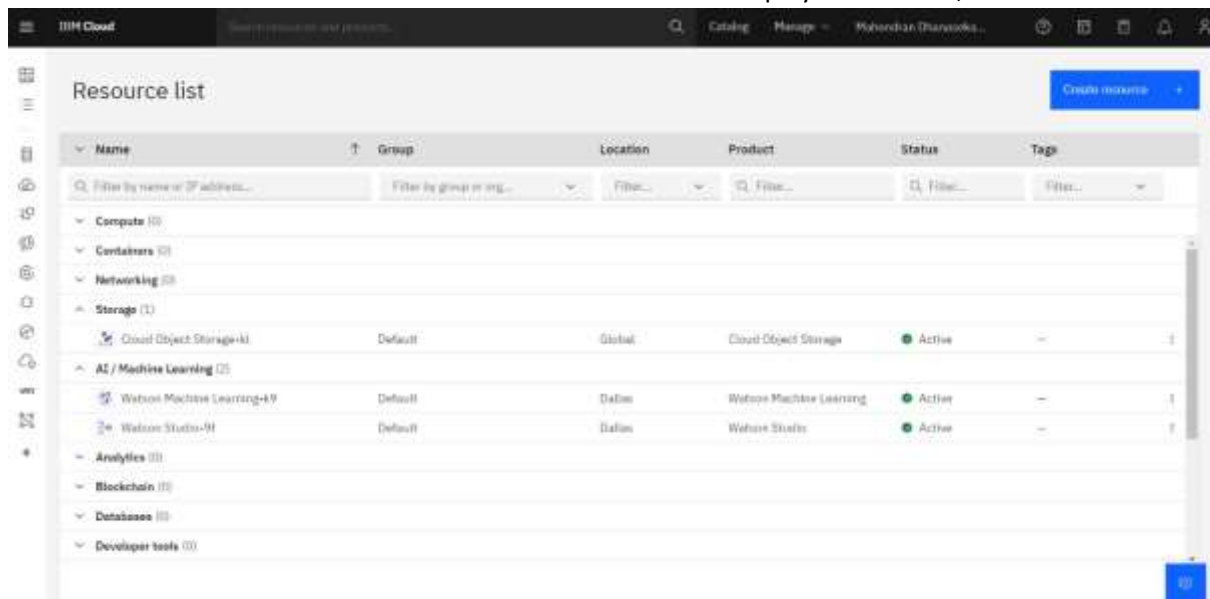
After creating IBM cloud account, to deploy ML model, need to create following resources such as,

Cloud Object Storage

Watson Machine Learning

Watson Studio

After created above resources Resource List of an account is displayed as follow,



All the resource are in active state.

All the required cloud resources are created successfully.

Deploy our model in IBM Watson:

To deploy ML model in IBM cloud, need to create project in IBM Watson. After successful creation of project import .ipynb file of sprint-1 which ML models are build in Jupyter notebook.

Upload required datasets and import it.

Deploy model using following code,

```
!pip install -U ibm-watson-machine-learning from
ibm_watson_machine_learning import APIClient
import json import numpy as np wml_cred={
    "apikey":"okbr7ARnOQjyplTOyvNFC2QVvCF6q7afpci065Hucby8",
    "url":"https://us-south.ml.cloud.ibm.com"
}
wml_clients=APIClient(wml_cred) wml_clients.spaces.list()
space_id="6d7c1218-3aca-4256-be3d-d610732530b1"
wml_clients.set.default_space(space_id)
wml_clients.software_specifications.list(500)
```

```

MODEL_NAME="randomforest"
DEPLOYMENT_NAME="rf_deployment"
DEMO_MODEL=rf
soft_sepc_id=wml_clients.software_specifications.get_id_by_name("runtime-22.1-py3.9")

```

In [115]:

```

model_props={ wml_clients.repository.ModelMetaNames.NAME:MODEL_NAME,
wml_clients.repository.ModelMetaNames.TYPE:"scikit-learn_1.0",
wml_clients.repository.ModelMetaNames.SOFTWARE_SPEC_UID: soft_sepc_id

}

```

In [116]:

```

model_details=wml_clients.repository.store_model(model=DEMO_MODEL,meta_props=model_props,training_data=x_train,
training_target=y_train.values.ravel())

```

In [117]:

```

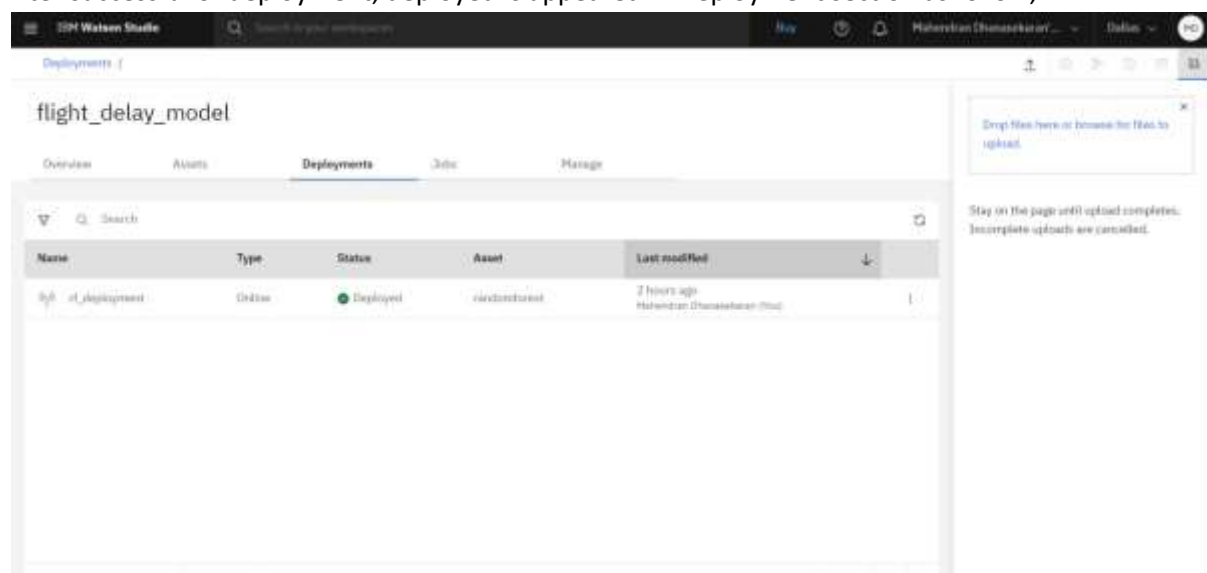
model_details
model_id=wml_clients.repository.get_model_id(model_details) dep_props={
wml_clients.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
wml_clients.deployments.ConfigurationMetaNames.ONLINE:{}
}

```

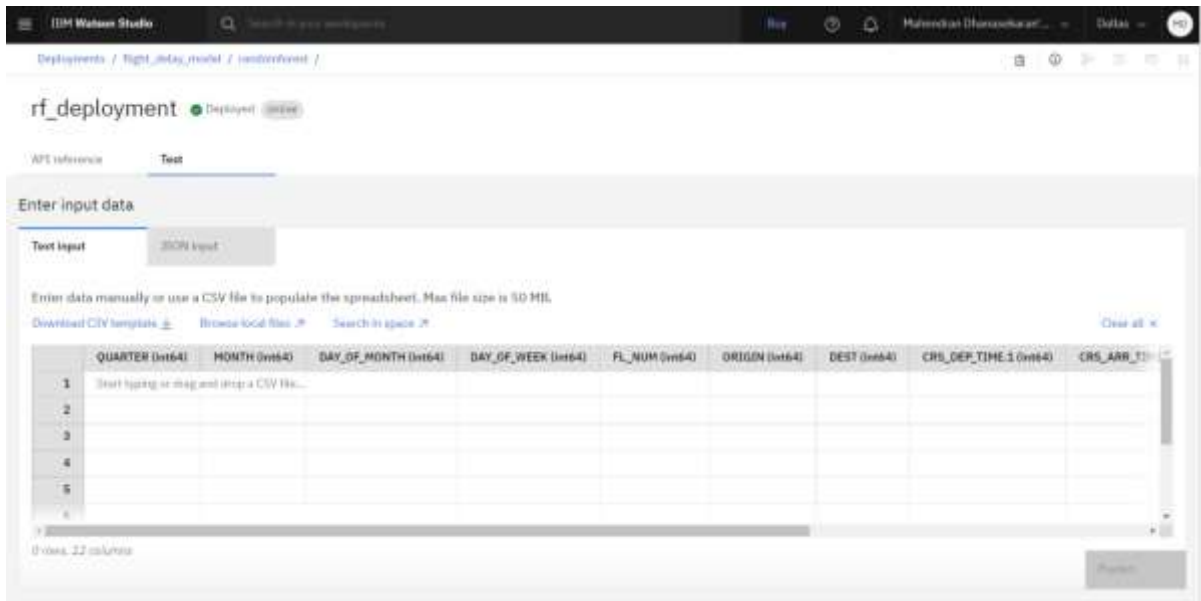
In [125]: deployment=wml_clients.deployments.create(artifact_uid=model_id,meta_props=dep_props)

NOTE: APIKey must need to create to deploy and connect API

After successful of deployment, deployed is appeared in Deployment section as follow,



Testing of deployed model as follow, by giving values of all the features and it gives prediction.



After these, need to copy API requesting codes on required language(python).

Creating Dashboard using HTML/CSS:

Frontend Dashboard is created using HTML/CSS,

Result as web page like,

Flight Delay Prediction

Quarter of the year

Month in number

Day of the Month

Day of the week

Flight Number

Origin Airport:

Destination Airport:

Planned Departure Time(format hhmm)

Planned Arrival Time(format hhmm)

Create web app and Hosting in flask:

Name	Date Modified
static	08-11-2022 12:17 AM
</> cssstyle.css	07-11-2022 10:29 PM
templates	08-11-2022 02:53 AM
</> mainpage.html	08-11-2022 02:53 AM
app_ibm.py	09-11-2022 01:25 AM
app.py	08-11-2022 02:55 AM
rfmodel.pkl	08-11-2022 01:52 AM

The screenshot displays a Jupyter Notebook environment with a Flask application. The code defines a Flask app with several routes:

- `/health`: A simple health check endpoint returning a 200 status.
- `/token`: An endpoint that generates a JWT token for authentication.
- `/predict`: The main endpoint for sentiment prediction. It accepts a POST request with a JSON body containing a list of text samples. The code iterates over these samples, predicts the sentiment (positive, neutral, or negative), and returns the results as a JSON array.

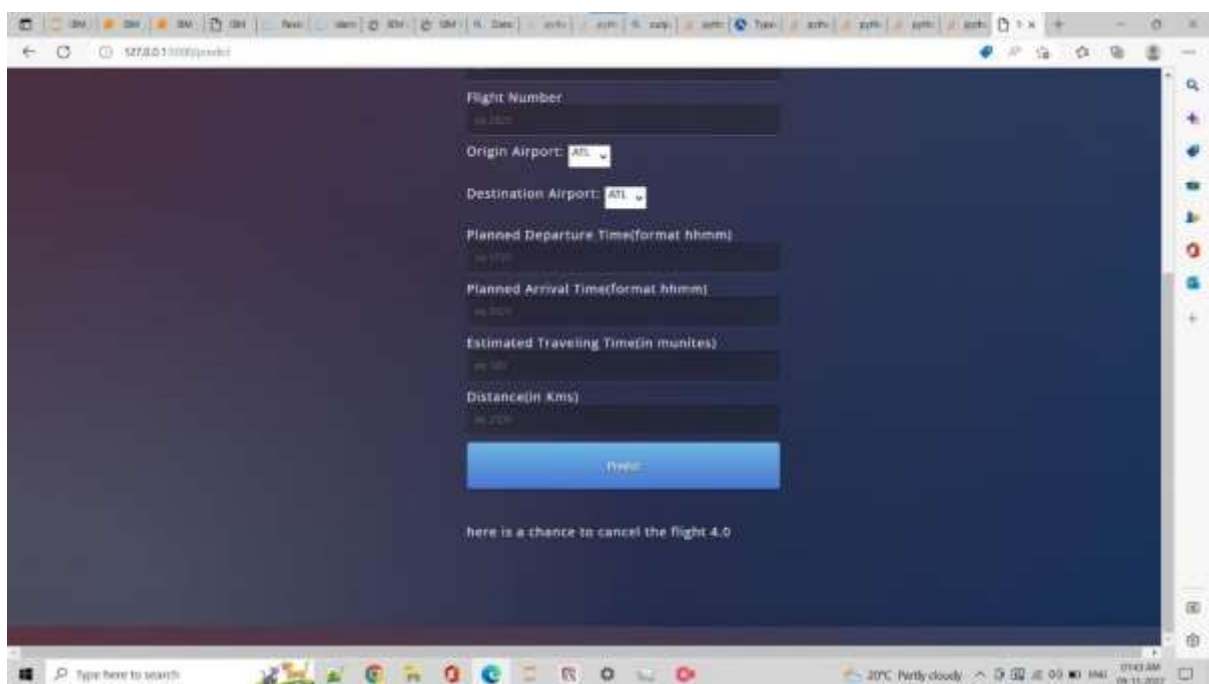
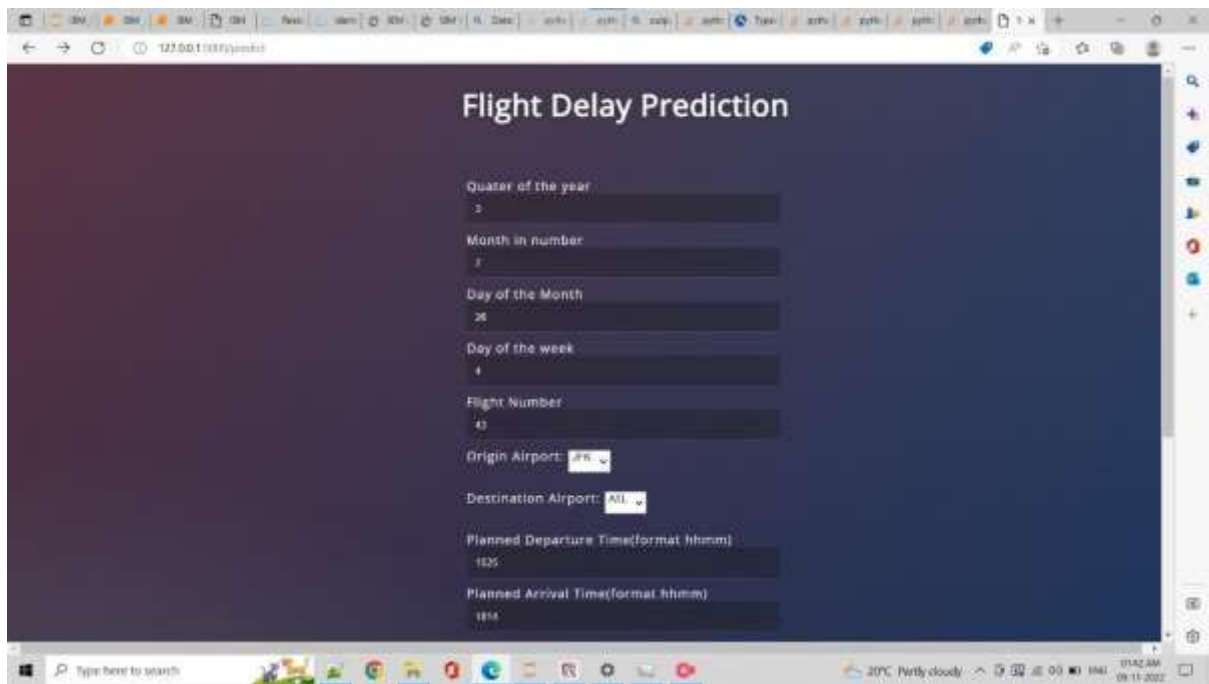
The output of the `/predict` endpoint is shown in the bottom right pane, displaying the predicted sentiment and probability for each input text sample.

Localhost url is displayed in console, copy and paste in browser then search it , frond end HTML?CSS page is displayed. Successfully created and hosted web app in flask.

Set FLASK_ENV=Development,

Testing web app:

Enter the data on the required fields,



Output is predicted by ML model successfully.