Machine Learning Model Deployment with IBM Cloud Watson Studio

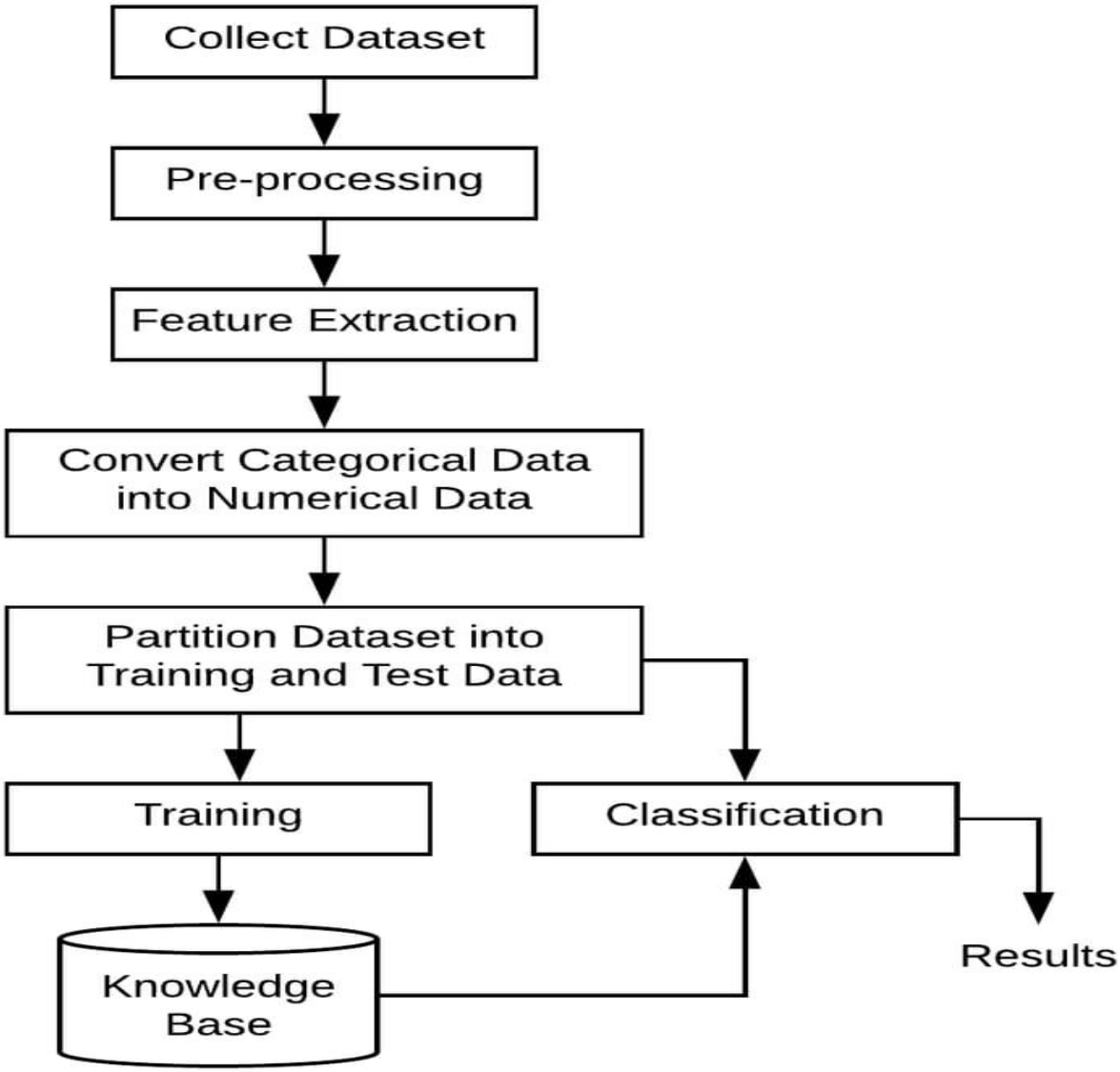
Phase 5: Project Documentation & Submission

Objective :

The objective of the Mission Learning model with IBM Cloud Watson Studio is to create an interactive and engaging learning experience. It aims to provide a platform where users can learn new skills, acquire knowledge, and solve problems through immersive missions. By leveraging the power of IBM Cloud Watson Studio, the model offers a dynamic and customizable learning environment that adapts to the user's progress and preferences. It combines elements of gamification, artificial intelligence, and data analytics to deliver personalized learning paths and real-time feedback. The ultimate goal is to empower users to become active learners, fostering curiosity, critical thinking, and collaboration. With the Mission Learning model, users can embark on exciting educational journeys, unlock achievements, and gain practical skills that can be applied in various domains. It's all about making learning fun, engaging, and effective!

Design Thinking Process :

1. **Predictive Use Case**: IPL Score Prediction



Predicting the total score of an IPL cricket match based on historical match data, team performance, player statistics, pitch conditions, and other relevant factors.

The main objective is to Provide cricket enthusiasts with real-time score predictions during IPL matches to enhance their viewing experience and engagement.

Technology used:

- Machine Learning.
- Deep Learning
- Flask (Front-end integration).
- Well, for the smooth running of the project we've used few libraries like NumPy, Pandas, Scikit-learn, TensorFlow, and Matplotlib.

2. **Dataset Selection:**

Gather historical IPL match data from reliable sources, including details such as team compositions, player statistics, venue information, weather conditions, and final match scores.

3. **Model Training**:

   Choose a suitable machine learning algorithm for regression, such as Random Forest Regression, Gradient Boosting, or even a Neural Network.

   Create relevant features like player form, team form, past performance at the venue, and head-to-head statistics. Split the dataset into training and testing sets.

4. **Model Deployment:**

   By Utilizing IBM Cloud Watson Studio for model development and deployment.

   Deploy the trained model as a web service on IBM Cloud Watson Studio's deployment capabilities, allowing it to be accessible via RESTful APIs.

5. **Integration:**

   Integrate the deployed model into websites allowing users to access real-time score predictions during live IPL matches. Design a user-friendly interface where users can input match details (teams, players, pitch conditions) to get score predictions.

Integration Steps:

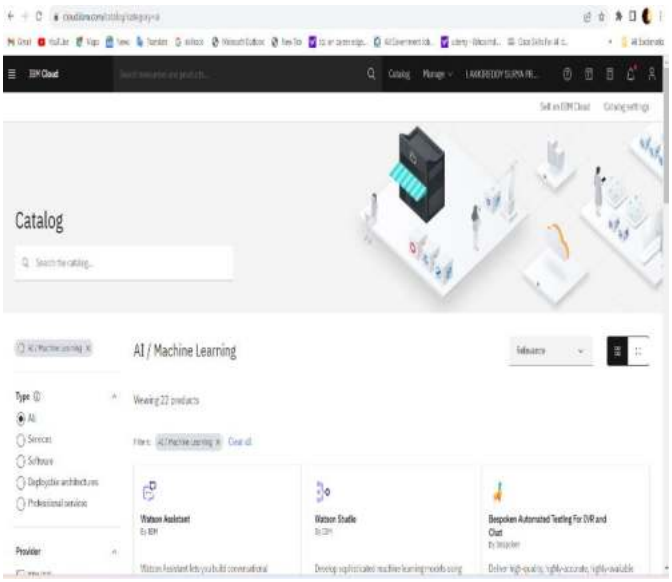
- Watson Studio seamlessly integrates with other IBM Cloud services, including data storage, databases, and cloud functions.

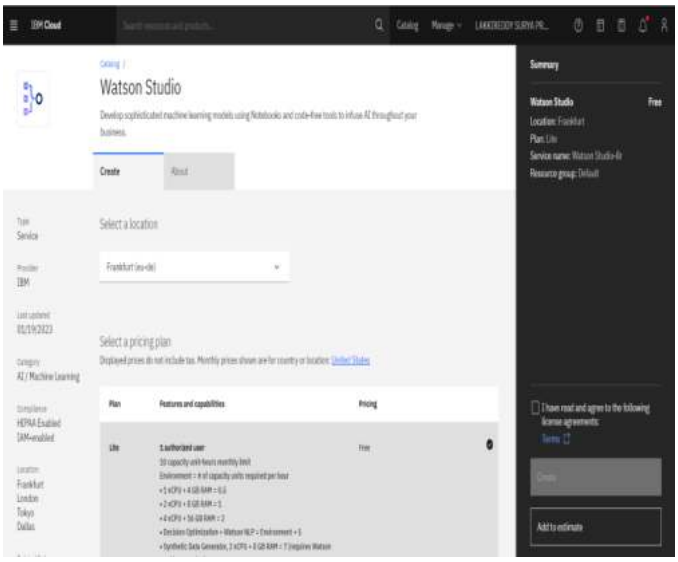
Now we are going to create the machine learning model with Watson Studio for that we will do the primary steps now :

STEP1:

- Login To The IBM account and click on the Catalog and then search for Watson Studio and give enter.

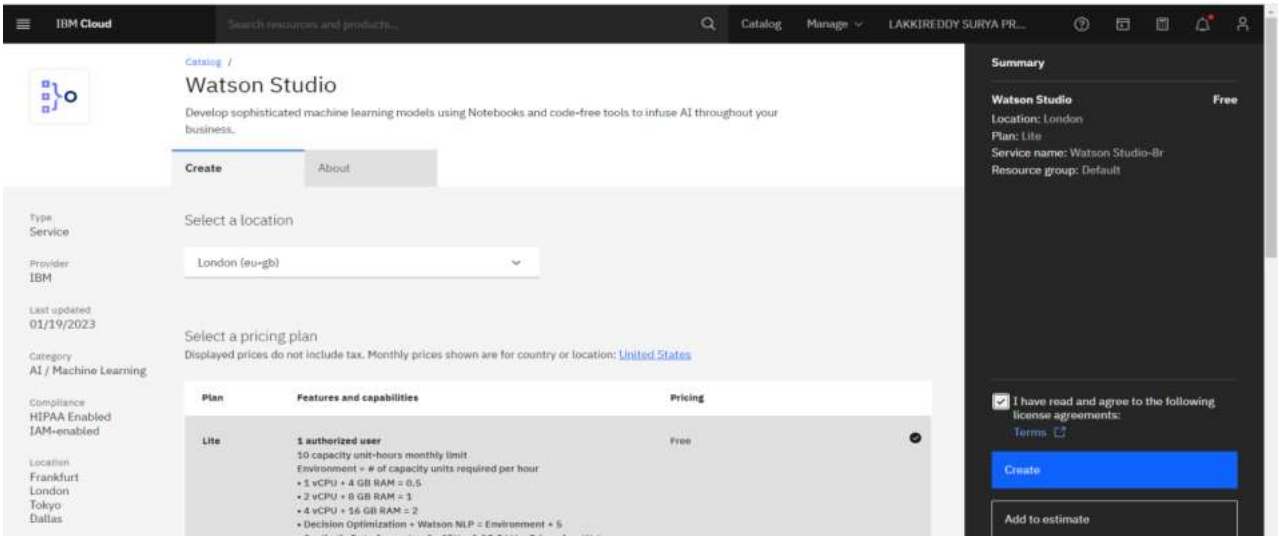
- You will get the Watson Assistant There By default you will have this



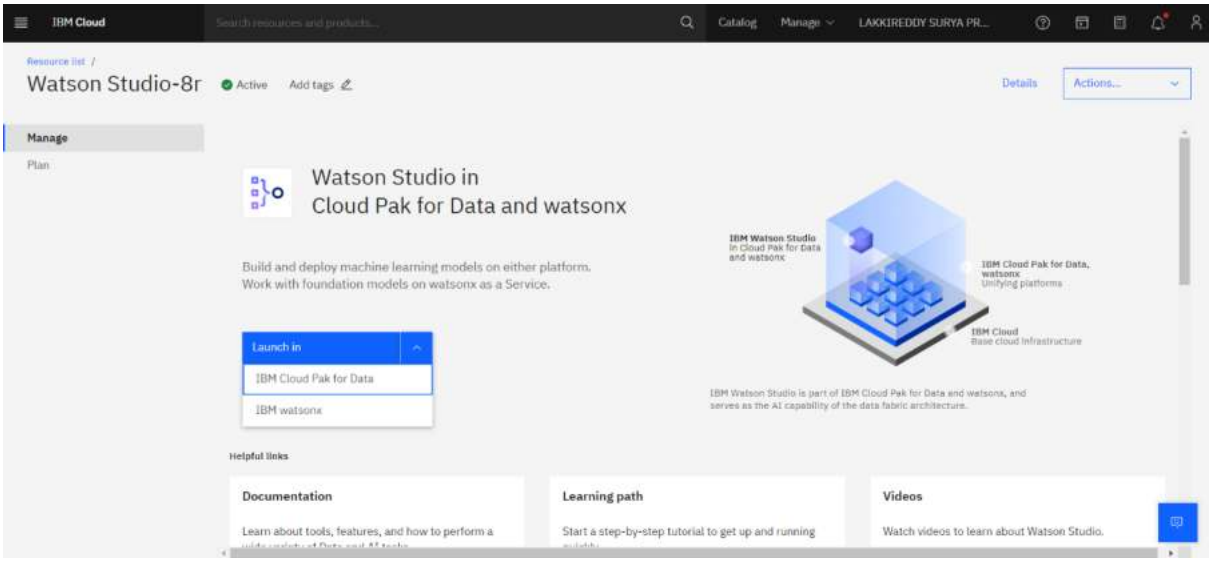


STEP 2:

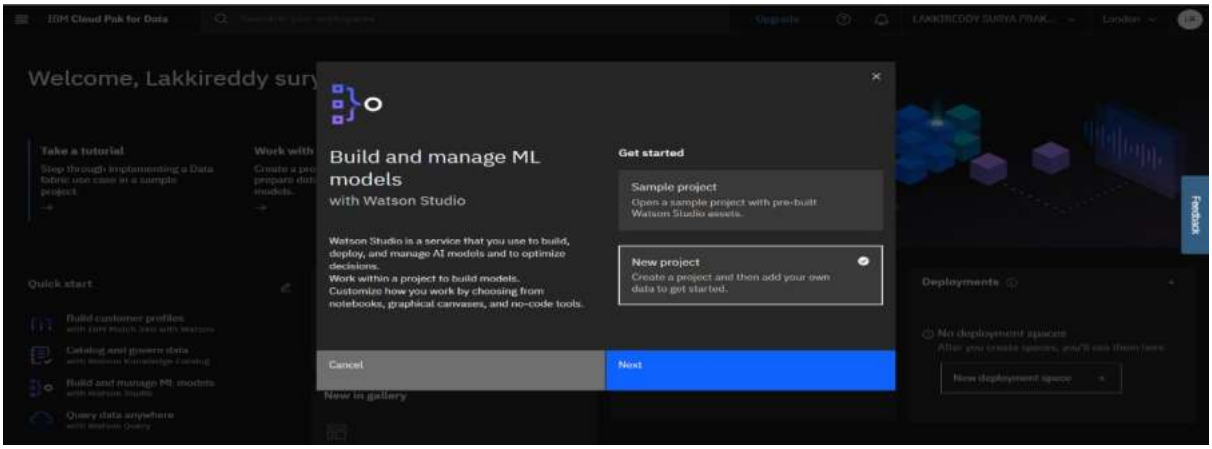
- Change the default location and give the location as London(eu-gb) and select the plan as Lite
- Give tick mark for I Have read and agree to the following license agreement
- Now click on create it will create an instance for you



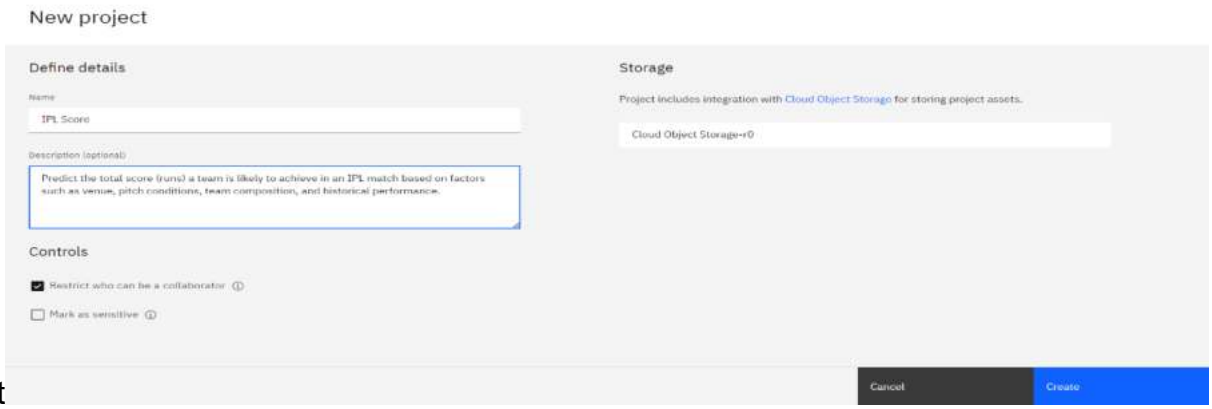
Step 3 : After creating an instance for Watson Studio you need to launch the Watson Studio by clicking the launch the IBM Cloud Pak for Data



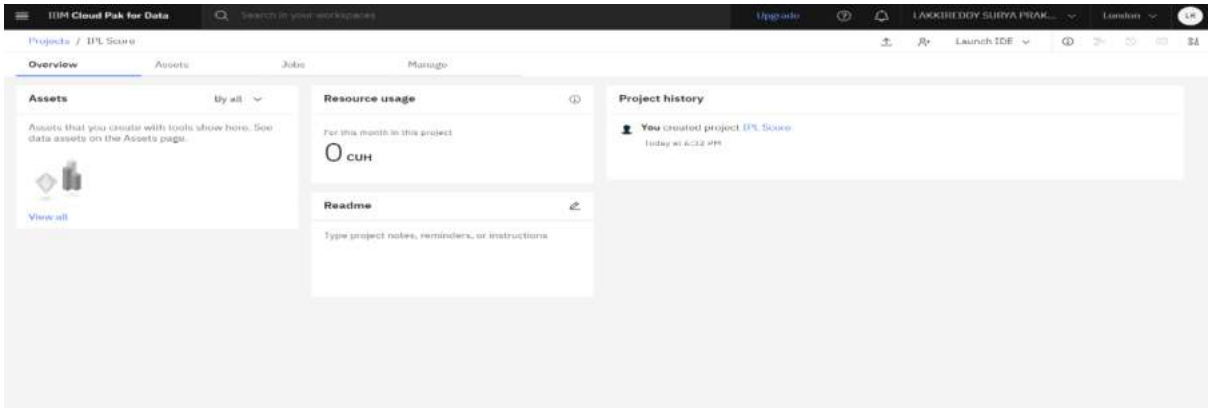
Step 4: After launching the IBM Cloud Pak for Data, it displays the below page then press the create new project **.**It ask press next for the further process by press next it creates new project



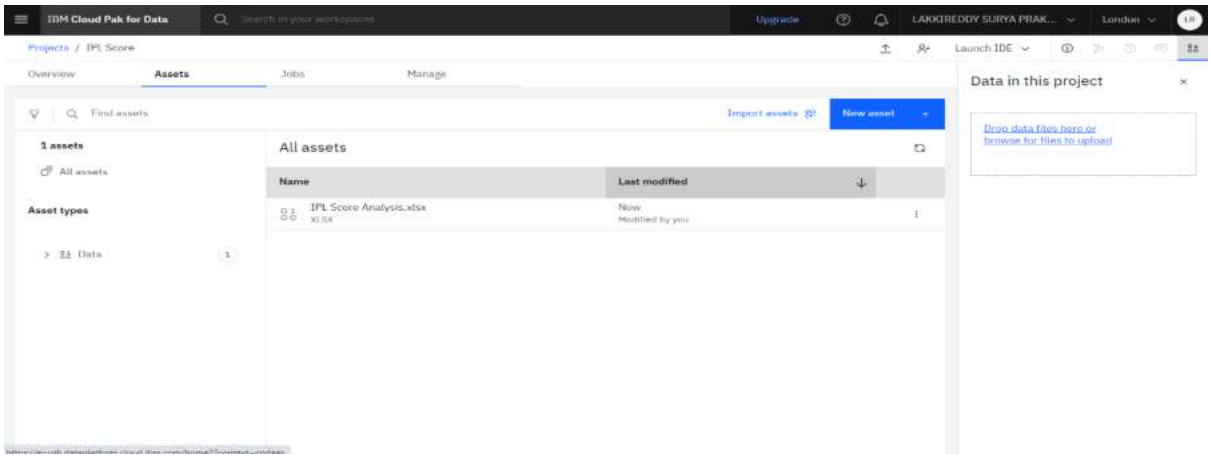
Step 5: It asks the project Name & Description to create a project, provide those to crate project



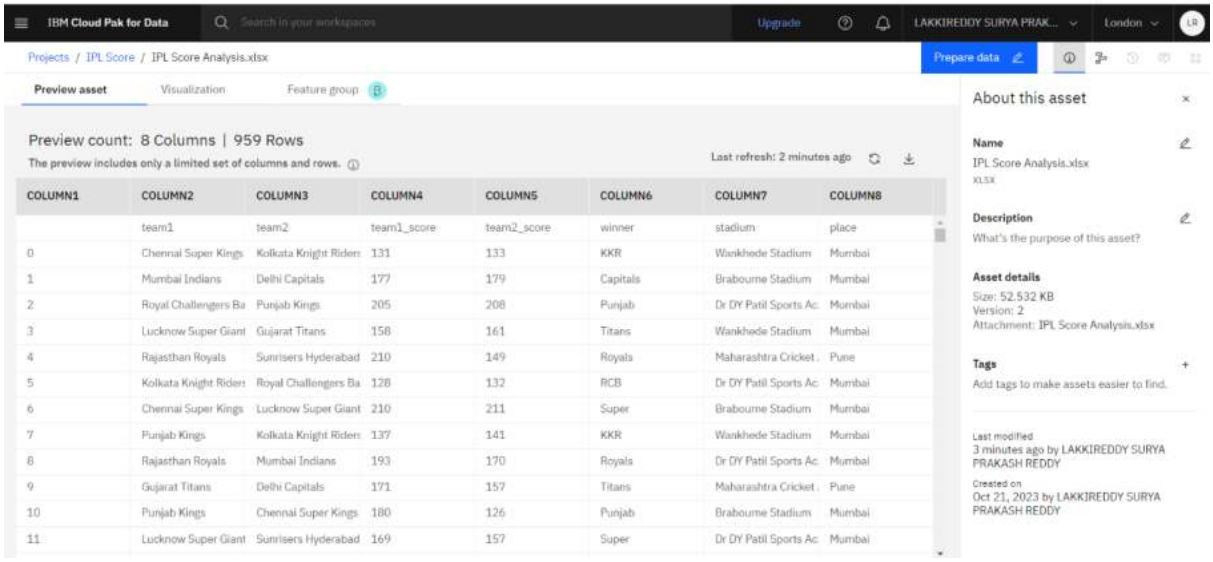
Step 6: By pressing create option the new project is created with the Machine Learning Model**.** It displays the below page for the further process



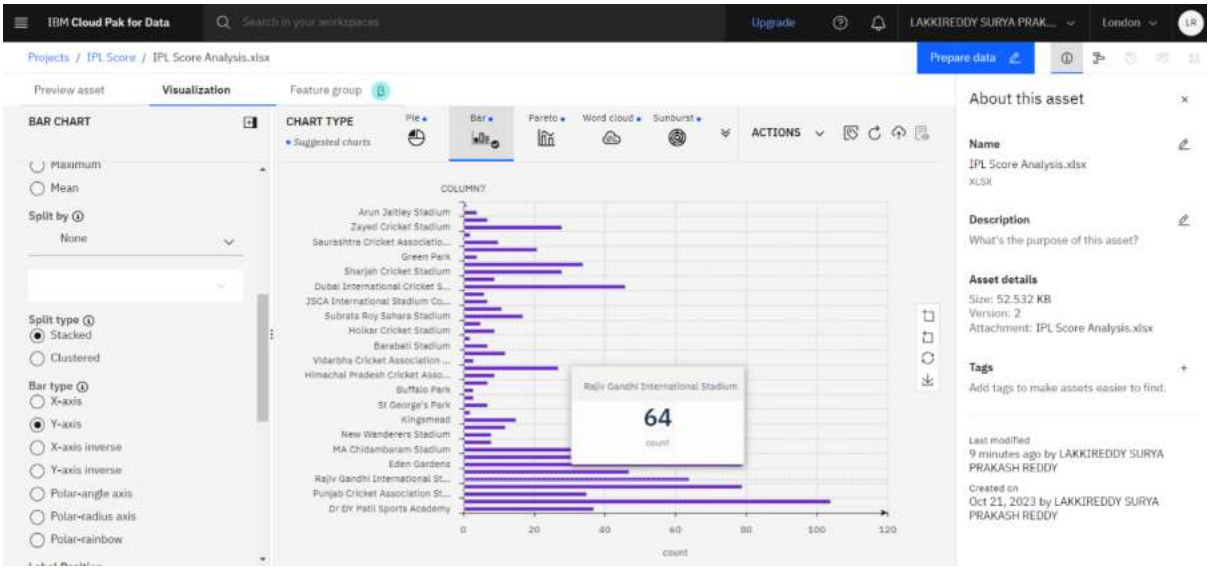
Step 7: From above go to the Assets then add the data set in the Assets section



Step 8: Press the add Assets then it displays the Data present in the Assets

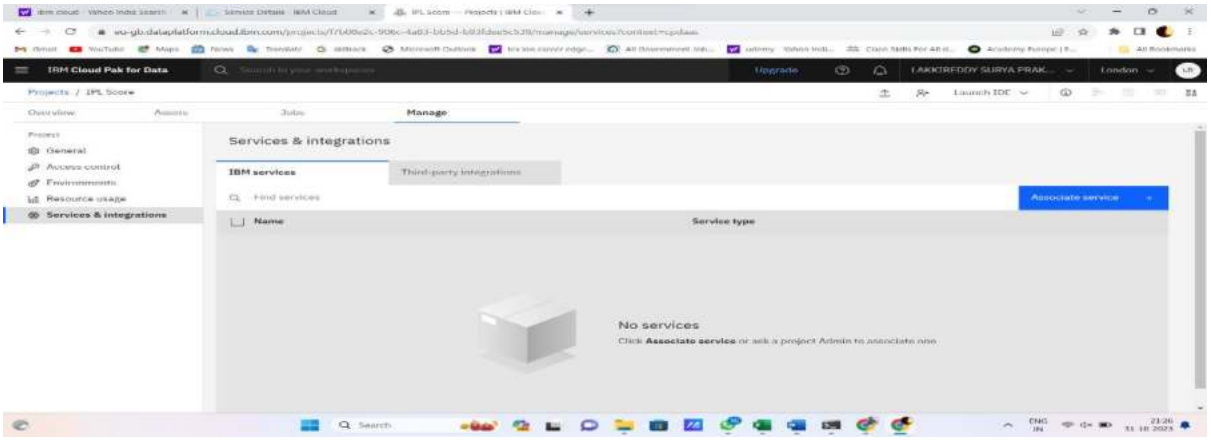


Step 9: Press the visualization to visualize the data set present in the Assets. In this visualization we see data in the various types of charts**.** As of now I am displaying the data in the from of Bar graph.

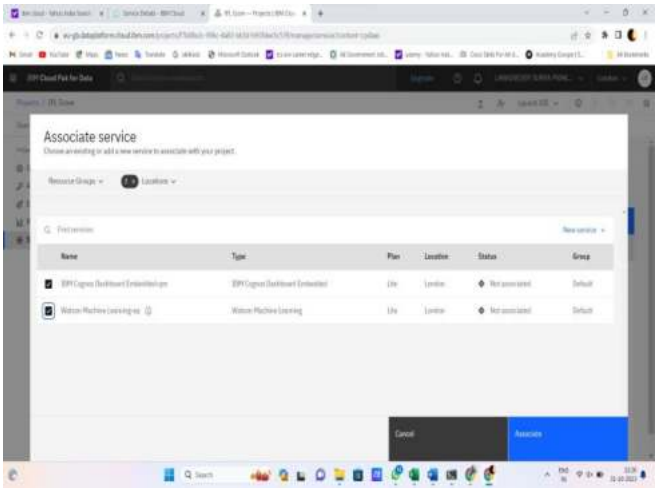


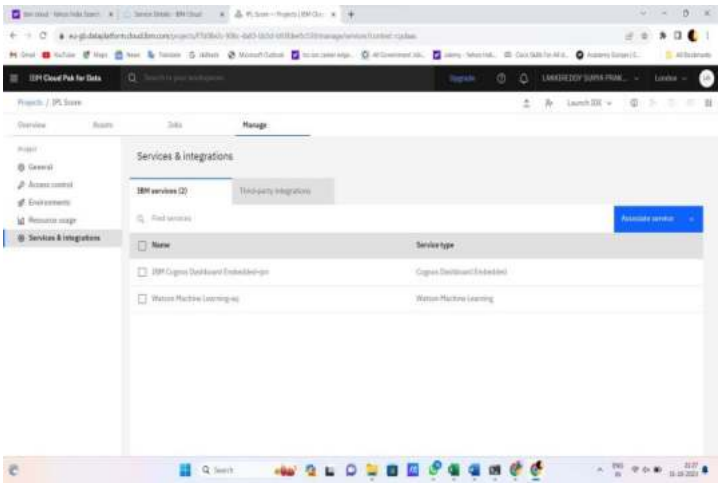
Deployment Process:

STEP1:

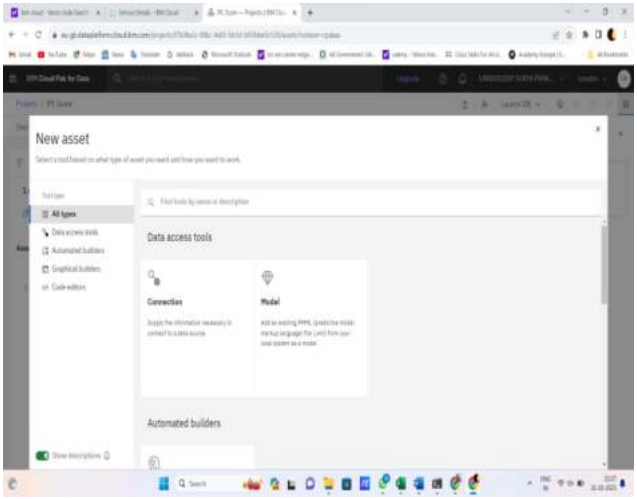


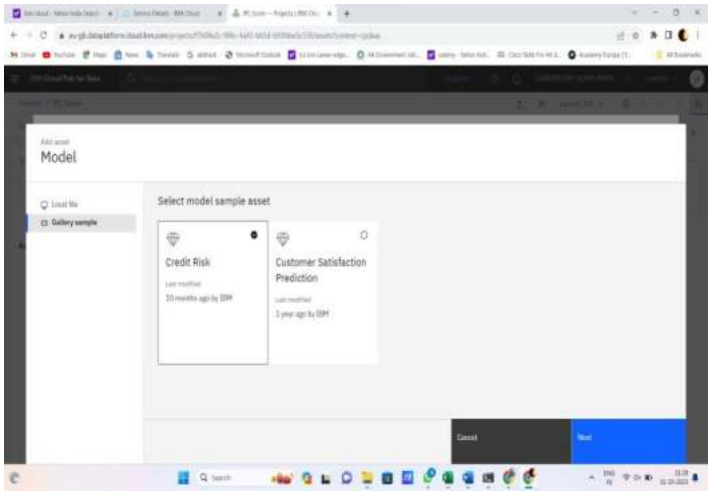
From the above picture we can see that previously created project, now here we associating services which is previously existed. Now press the Associate to Associate the services. after that we can see the list of Associated Services.



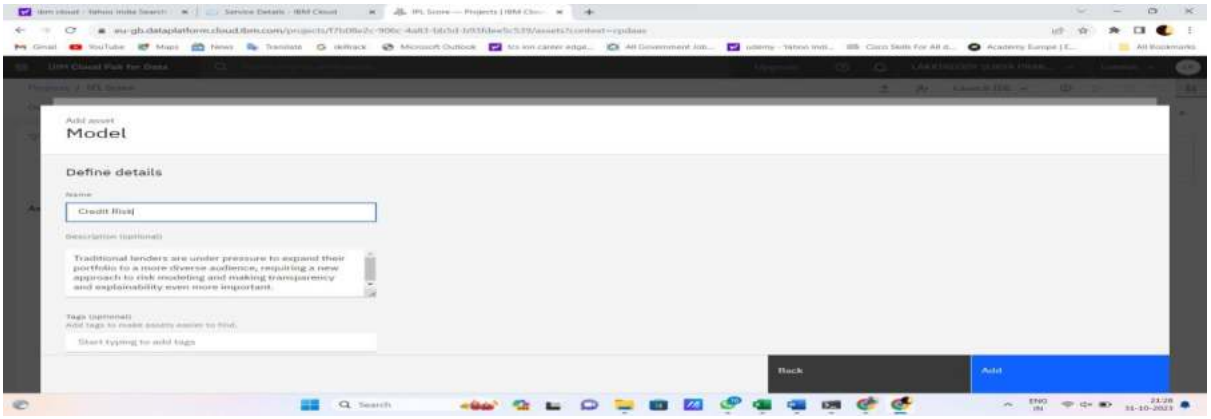


STEP2: It is process of creating model by New asset it displays like a below page from there we need select the Model. After pressing model it displays next page, Here we need to choose the Gallery sample. Next it gives two options in that we need to choose the credit Risk , Then press NEXT option for the further process

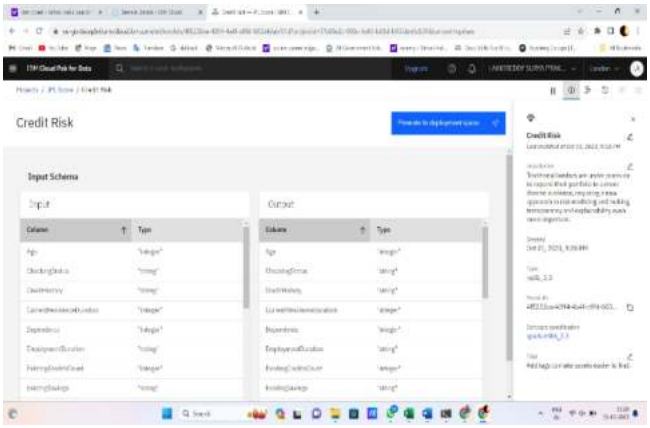


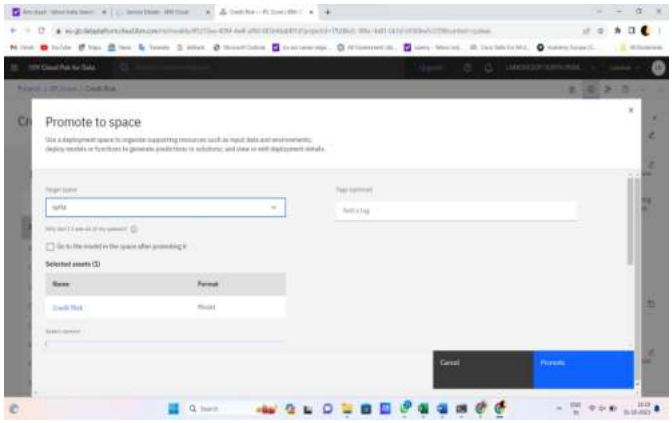


STEP3: It is the process of giving name of the model, here I am model as Credit Risk. Then press the Add to move next process

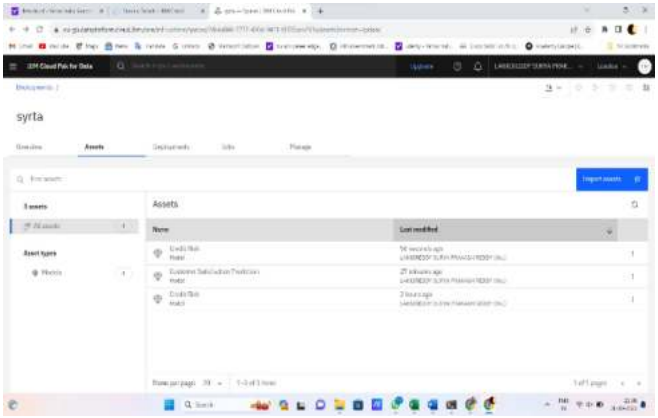


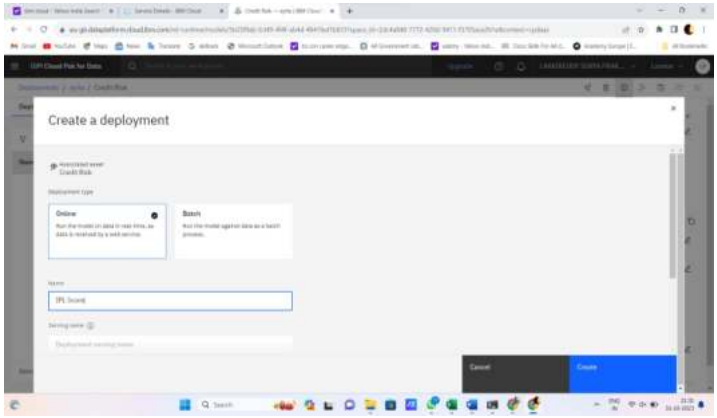
STEP4: It is process of deploying the developed model, Here press the promote to deployment space. It is process of deploying the developed model, Here press the promote to deployment space.



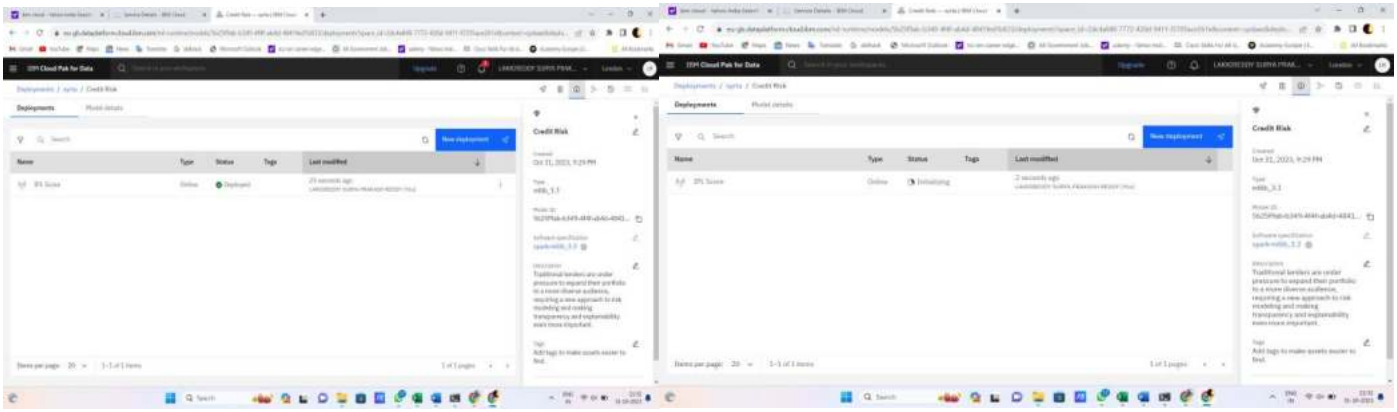


STEP5: Here we can see the list of models created using Watson Studio. Now choose the developed model from the above list of models.



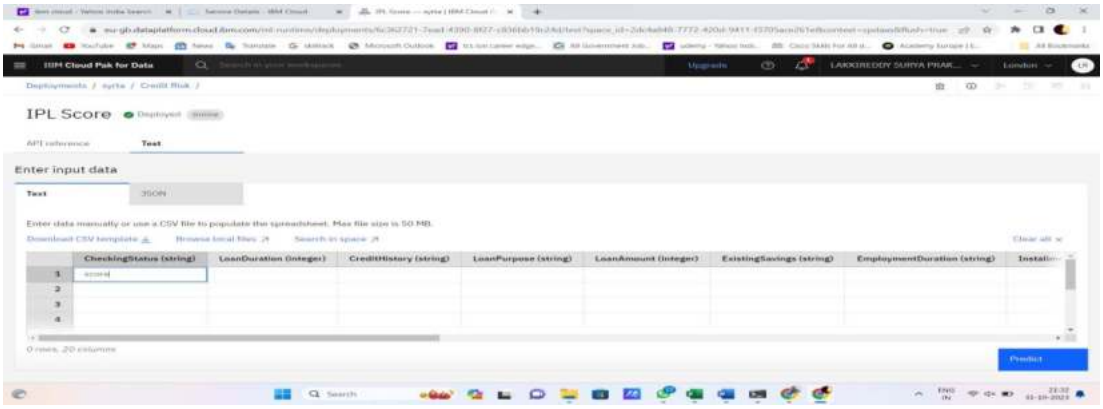


STEP6: It is the process of Initialising the Deployements of a model. After Initialising the model

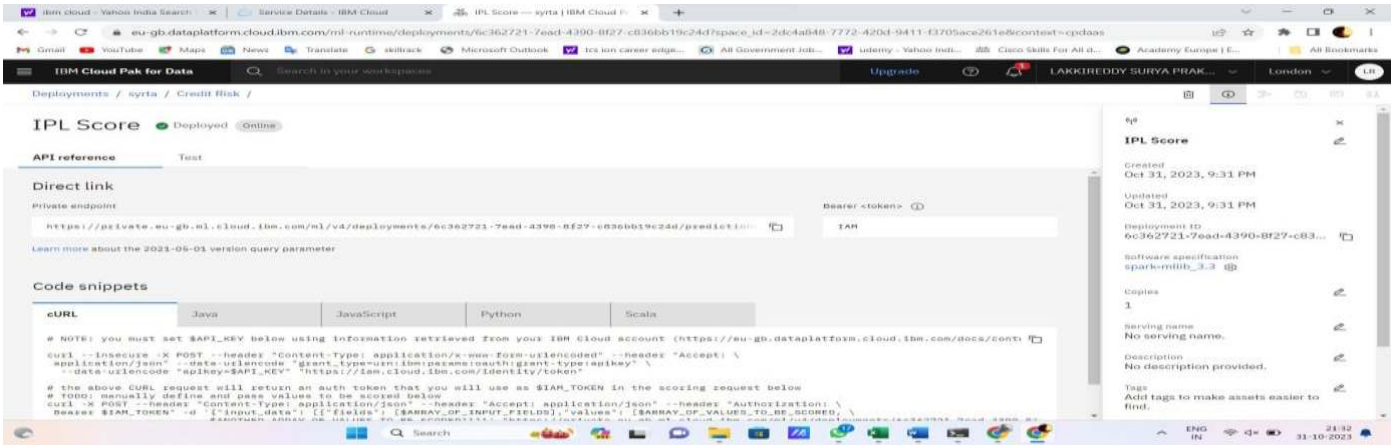


STEP7:

It is the process of testing the model



STEP8: The model is successfully deployed into the IBM Cloud



Web Application Development:

SOURCE CODE:

```
<!DOCTYPE html>
<html>
<head>
    <title>IPL Score Prediction</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            background-image: url("https://wallpapercave.com/wp/wp4059913.jpg");
            background-repeat: no-repeat;
            background-size: cover;
            height: 100px;
            }
            container {
            margin: 50px auto;
```

```
            max-width: 400px;
            padding: 20px;
            border: 1px solid #ccc;
            border-radius: 5px;
            background-color: black;
            color:blue;
        }
        #team1, #team2, #predictionResult {
            margin: 10px 0;
        }
        #predictButton {
            background-color: #007bff;
            color: #fff;
            border: none;
            padding: 10px 20px;
            cursor: pointer;
        }
        img{
            border-radius: 50%;              }
        h1{
            color:yellow;
            align-content: first baseline;
        }
    </style>
</head>
<body>
    <h1>IPL Score Prediction</h1>
   <img src="https://telecomtalk.info/wp-content/uploads/2022/06/viacom18-bags-three-big-wins-related-ipl.jpg" width="160" height="150">
    <div class="container">
        <label for="team1">Team 1:</label>
        <input type="text" id="team1" placeholder="Enter Team 1" required>
        <br>
        <label for="team2">Team 2:</label>
```

```
        <input type="text" id="team2" placeholder="Enter Team 2" required>
        <br>
        <label for="venue">Venue:  </label>
        <input type="text" id="venue" placeholder="Enter Venue" required>
        <br><br>
        <button id="predictButton">Predict Score</button>
        <div id="predictionResult"></div>
    </div>
    <script>
        document.getElementById('predictButton').addEventListener('click', function () {
            const team1 = document.getElementById('team1').value;
            const team2 = document.getElementById('team2').value;
            const venue = document.getElementById('venue').value;
            const randomScore = Math.floor(Math.random() * 200);
            document.getElementById('predictionResult').innerHTML = Predicted Score:
${randomScore};
        });
    </script>
</body>
</html>
```

OUTPUT: