**Lab 5: Deploying Your Model as a Web Service in Azure Machine Learning Studio**

**Objective**

Deploy the best-performing model from the Automated ML run to Azure Container Instance (ACI) as a web service and learn how to make predictions using the deployed model.

**Instructions**

1. **Selecting the Best-Performing Model:**
   * Navigate to the **Models** section in Azure Machine Learning Studio once your Automated ML job is completed.
   * Identify the model with the best performance metrics. Look for high scores or low errors, depending on the specific metric used (accuracy, RMSE, etc.). The best model will also have feature explanations in the **Explained** column.
2. **Deploying the Model:**
   * Click on the model you’ve selected to open the model's details page.
   * Click on the **Deploy** button.
   * Choose **Web service (ACI)** as the deployment option.
   * Configure the deployment settings:
     + **Name:** Provide a unique name for the web service.
     + **Compute type:** Select **Azure Container Instance**.
     + **Enable authentication:** Set this toggle to true to secure your web service.
3. **Monitoring the Deployment:**
   * After initiating the deployment, monitor the progress in the **Endpoints** menu.
   * Ensure that the **Deployment state** shows **Healthy** and the **Operation state** is **Succeeded**.
4. **Making Predictions:**
   * Once the deployment is successful, go to the **Endpoints** menu and find your new endpoint.
   * Use the **Test** tab in the endpoint details to make a quick test prediction to verify that everything is working as expected.
   * Access the **Consume** tab to find the endpoint URL and authentication key, which are necessary for integrating the web service into your applications.

**Exploration and Integration:**

* Experiment with sending different data inputs to your model through the **Test** tab and observe how it predicts outcomes.
* Discuss how you might integrate this web service into an application or workflow. What kind of data would you send? How would you handle the responses?