AI 900

Contents

[WHAT IS MACHINE LEARNING 2](#_Toc192672003)

[MACHINE LEARNING MODEL 2](#_Toc192672004)

[MACHINE LEARNING PROCESS 2](#_Toc192672005)

[MACHINE LEARNING MODEL AND DATASET 3](#_Toc192672006)

[DATA SET 3](#_Toc192672007)

[FEATURES AND LABELS 4](#_Toc192672008)

[FEATURE ENGINEERING 5](#_Toc192672009)

[MACHINE LEARNING ALGORITHMS 5](#_Toc192672010)

[MACHINE LEARNING MODELS 5](#_Toc192672011)

[MACHINE LEARNING ALGORITHMS CLASSIFICATION 7](#_Toc192672012)

[AZURE MACHINE LEARNING WORKSPACE 8](#_Toc192672013)

[CREATING AZURE MACHINE LEARNING WORKSPACE 8](#_Toc192672014)

[WORKING WITH ML WORKSPACE 10](#_Toc192672015)

[BUILDING CLASSIFICATION ML PIPELINE 11](#_Toc192672016)

[STEP 1: SPLIT DATA 12](#_Toc192672017)

[STEP 2: TRAIN MODEL 18](#_Toc192672018)

[STEP 3: SCORE MODEL 20](#_Toc192672019)

[STEP 4: EVALUATE MODEL 21](#_Toc192672020)

[STEP 5: RUN THE PIPELINE – EXPERIMENT 21](#_Toc192672021)

[EVALUATION OF MODEL RESULT 21](#_Toc192672022)

[DEPLOYING THE ML MODEL 23](#_Toc192672023)

[INFERENCE PIPELINE 24](#_Toc192672024)

[TESTING THE MODEL 31](#_Toc192672025)

[CONSUMING MODEL USING ENDPOINT 31](#_Toc192672026)

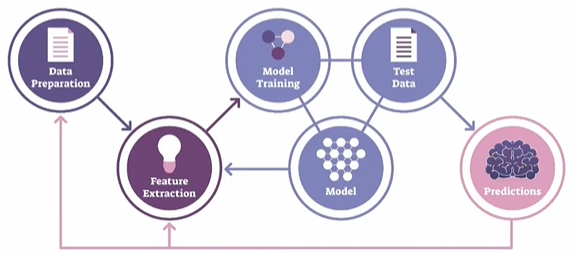
# WHAT IS MACHINE LEARNING

* Machine learning is a subfield of artificial intelligence.
* It involves training computers on large amounts of data.
* Computers learn from the data and make predictions or decisions.
* It allows computers to recognize patterns and improve their performance over time.
* Machine learning is used in various applications like image and speech recognition, recommendation systems, and predictive analytics.

## MACHINE LEARNING MODEL

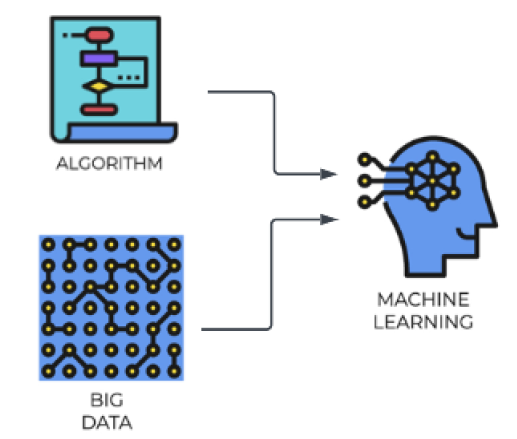
* This is an object that has been trained beforehand on a dataset. The model can then give output based on input values
* The machine learning object is nothing but a file or set of files. It would have the desired logic that would take in input and spur output data

## MACHINE LEARNING PROCESS



1. **DATA PREPARATION**:
   1. This step involves gathering and cleaning the data to make it suitable for analysis.
   2. It includes tasks like handling missing values, removing outliers, normalizing or standardizing the data, and splitting the data into training and testing sets.
2. **FEATURE EXTRACTION**:
   1. Feature extraction is the process of selecting or creating relevant features from the data that will be used as input for the machine learning model.
   2. It involves identifying the most informative attributes or transforming the data into a more meaningful representation that captures the underlying patterns and relationships.
3. **MODEL TRAINING**:
   1. Model training is the process of using the prepared dataset to teach the machine learning model to recognize patterns and make predictions.
   2. During this step, the model learns from the input data and adjusts its internal parameters to optimize its performance.
4. **MODEL**:
   1. The model is the representation of the learned patterns and relationships from the training data.
   2. It encapsulates the knowledge gained during the training process and can be used to make predictions or decisions on new, unseen data.
5. **TEST DATA**:
   1. Test data is a separate set of data that is not used during the model training. It is used to evaluate the performance of the trained model and measure its accuracy.
   2. The test data should be representative of the real-world scenarios the model will encounter.
6. **PREDICTION**:
   1. Once the model is trained and evaluated, it can be used to make predictions on new, unseen data.
   2. The model takes the input data, processes it using the learned patterns, and generates predictions or decisions based on the trained knowledge.
   3. The accuracy and reliability of the predictions depend on the quality of the model and the suitability of the input data.

## MACHINE LEARNING MODEL AND DATASET



### DATA SET

* A dataset is a collection of structured or unstructured data that is used for analysis, machine learning, or other data-related tasks. It typically consists of multiple data points or instances, where each data point represents an observation or sample.
* A dataset can include various types of data such as numerical, categorical, text, images, or time series.
* Datasets are crucial for training and evaluating machine learning models. They provide the necessary information for the model to learn patterns and relationships.
* Datasets can be sourced from various places, including public repositories, research institutions, or generated internally within an organization
* Common operations on datasets include cleaning, transforming, splitting into training and testing sets, and feature engineering.

#### EXAMPLE

A screenshot of a white grid

AI-generated content may be incorrect.

* Let's say we are building a ML model that can predict the house prices over time. For that we need to have prior information in place about the houses that were already sold over the previous years.
* The information can be the location of the house, the number of rooms, the agent involved in selling the house,

the area in square feet, and what was the price in USD. (as shown in above table)

* To train the model using historical data, we need to select the relevant features/column. ( for example, Agent column is irrelevant for price prediction)
* **This is the historical data is then used to training the ML model. Hence the data like - location, the number of rooms, and the area will become the features. Features are the measurable property within the dataset and the price is going be the label.**

### FEATURES AND LABELS

#### FEATURES

* Features, also known as input variables or independent variables, are the measurable properties or characteristics of the data that are used to make predictions or classifications.
* They represent the input data that the model will analyze and learn patterns from. For example, in the context of predicting house prices, features could include the location, number of rooms, and area of a house.
* Each feature is represented by a specific value or attribute.

#### LABELS

* Labels, also known as target variables or dependent variables, are the values that the model aims to predict or classify.
* Labels represent the desired output or outcome based on the input data. In the house price prediction example, the label would be the actual price of the house.
* The model analyzes the provided features and attempts to learn the patterns that will help predict or estimate the label.

|  |
| --- |
| * During the training process, the model is presented with a dataset that contains both the features and the corresponding labels. * The model learns the relationship between the features and labels by adjusting its internal parameters to minimize the difference between its predicted output and the actual labels. * Once trained, the model can then be used to make predictions or classifications on new, unseen data by analyzing the features and producing a predicted label. * In summary, features are the measurable properties of the data that serve as input for the model, while labels are the desired output or outcome that the model aims to predict or classify. The relationship between the features and labels is learned by the model during the training process. |

**IN A NUTSHELL, the summary is**

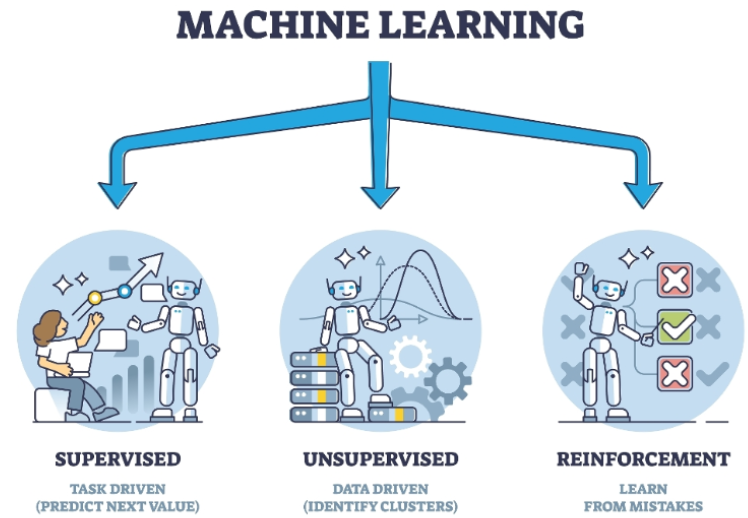
|  |
| --- |
| * Building a model to predict house prices over time * Prior information about sold houses including location, number of rooms, agent, area, and price in USD * Training the model using historical data, selecting relevant features and the price as the label * Cleaning the data, removing missing values, and performing feature engineering * Splitting the dataset into training and testing sets, using the training set to train the model and the testing set to evaluate its accuracy * Testing the accuracy and precision of the model by comparing its predictions with the actual values * Deploying the model and allowing users to make predictions based on new input data, such as location, number of rooms, and area. |

### FEATURE ENGINEERING

### MACHINE LEARNING ALGORITHMS

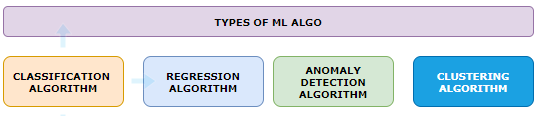
|  |  |
| --- | --- |
|  | * When building a machine learning model, we need to train it on a dataset that consists of input features (also known as independent variables) and corresponding output labels (also known as dependent variables). The dataset serves as the training data for the model to learn patterns and relationships between the input features and output labels. * To enable the model to learn from the dataset, we need to choose and apply a suitable machine learning algorithm. These algorithms are designed to process the input features and labels and make predictions or classifications based on the patterns they discover in the data. * The chosen algorithm takes the dataset as input and applies its specific mathematical or statistical techniques to train the model. It learns from the provided dataset by adjusting its internal parameters to minimize the difference between the predicted output and the actual output labels. * Once the model has been trained using the chosen algorithm, it can be used to make predictions or classifications on new, unseen data. The trained model has learned the patterns in the dataset and can apply this knowledge to make predictions on similar data instances. |

### MACHINE LEARNING MODELS



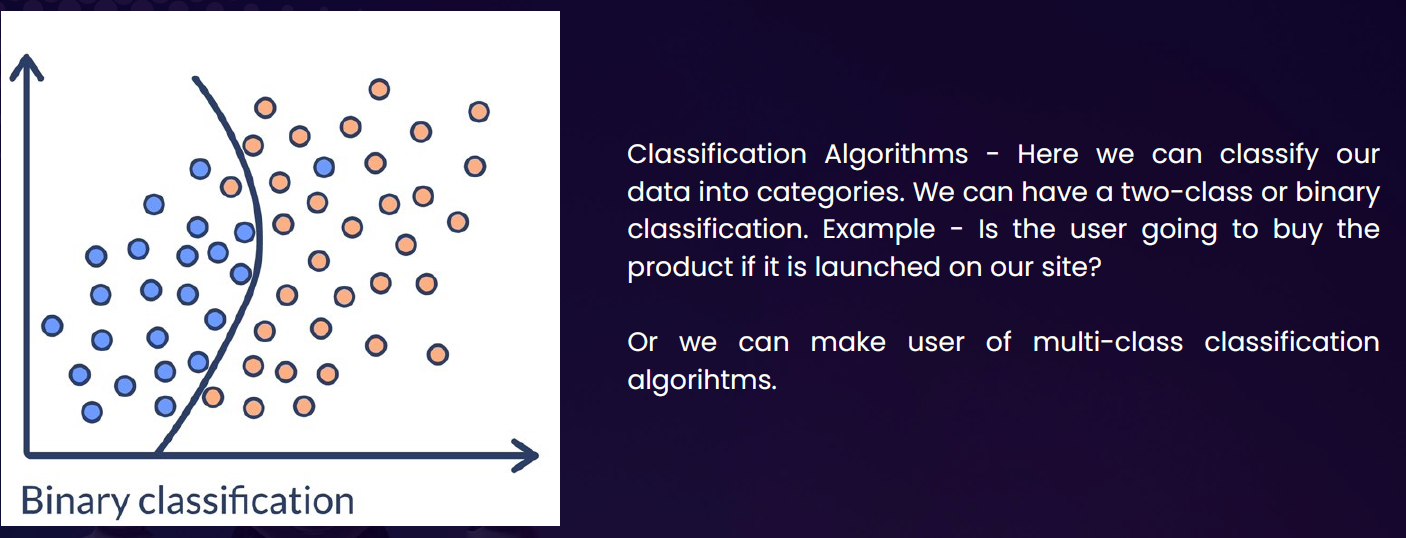
1. **SUPERVISED LEARNING MODELS**
   1. In supervised learning, for training data set that we feed in the features along with the labels(output)
2. **UNSUPERVISED LEARNING MODELS**:
   1. In unsupervised learning – we don’t know what the output will be, hence we cannot provide labels in training data
   2. Here we just want to the model to split our data into different groups or cluster
3. **REINFORCEMENT LEARNING MODELS**:
   1. These models learn through interaction with an environment. They receive feedback in the form of rewards or penalties, allowing them to learn optimal strategies for decision-making.
   2. Example :
      1. Let's take the example where we trying to teach a system how to play a game of chess against a human. So now, instead of we feeding in data about the different chess moves, we ask the model to play against a human.
      2. Hence, the system would go ahead and make a set of moves, and let's say it loses. So now it understands that if it performs particular set of actions, it results in a loss,then the next time around it will go and change its actions.
      3. It'll keep on doing this, so it's performing reinforcement of its actions. At the end when it does actually win a game,that's the reward. So now it understands that based on certain set of actions that it takes, it's gonna get a reward.

### MACHINE LEARNING ALGORITHMS CLASSIFICATION



#### CLASSIFICATION ALGORITHM

* This is simply used to class the data
* Example: Will the customer purchase the new product – Simple “yes”/ “No” answer



#### REGRESSION ALGORITHM

* This is used to predict new data point based on historical data
* Example: Calculate the forecasted revenue of a new product over coming week based on historical data

A graph with arrows and dots

AI-generated content may be incorrect.

#### ANOMALY DETECTION ALGORITHM

* This is used to detect if data deviates from the norm
* Example: used to detect fraudulent credit card purchase

#### CLUSTERING ALGORITHM

A computer screen with text and a diagram

AI-generated content may be incorrect.

# AZURE MACHINE LEARNING WORKSPACE

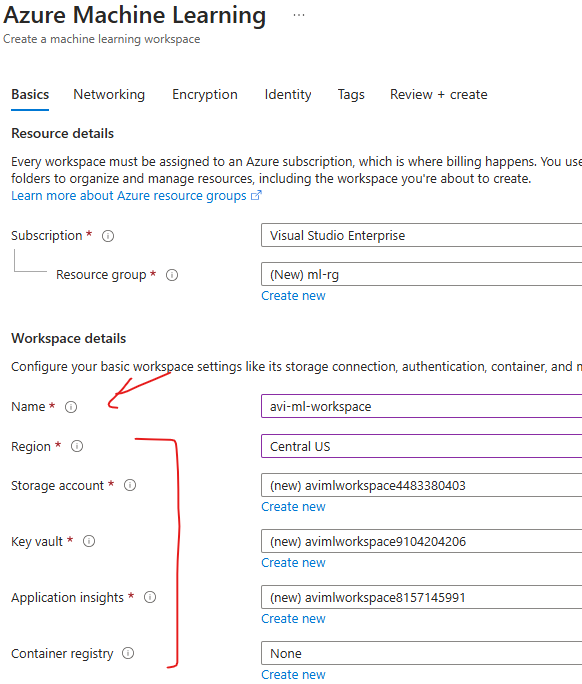
* To work on ML in Azure we need to deploy Azure ML workspace.

## CREATING AZURE MACHINE LEARNING WORKSPACE

* The Azure ML workspace encapsulates the ML artifacts that includes datasets, ML models etc.
* When we deploy ML workspace, there are other resources also get deployed along with it

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | | **STORAGE ACCOUNT** | * When we run the Job to train the machine learning model, these storage accounts are used to store the logs | | **CONTAINER REGISTRY** | * Used to store the images of created docker container | | **KEY VAULT** | * To score secrets needed by workspace or other resources | | **APPLICATION INSIGHT** | * To collect diagnostic information from the inference endpoints | |

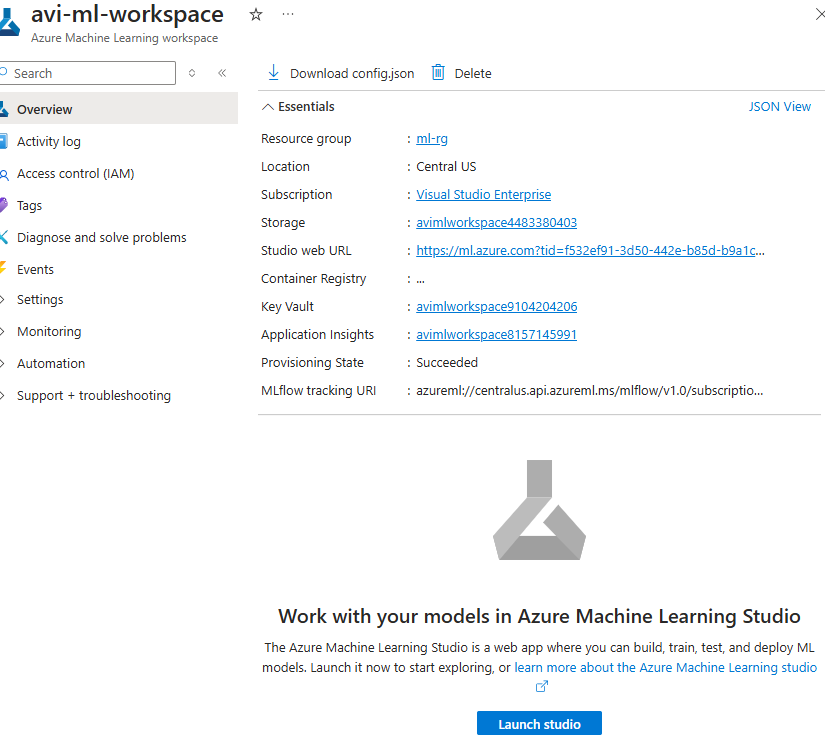
* Provide the name of ML workspace 🡪 Review & Create (*We will keep default selection of other options in the wizard*)

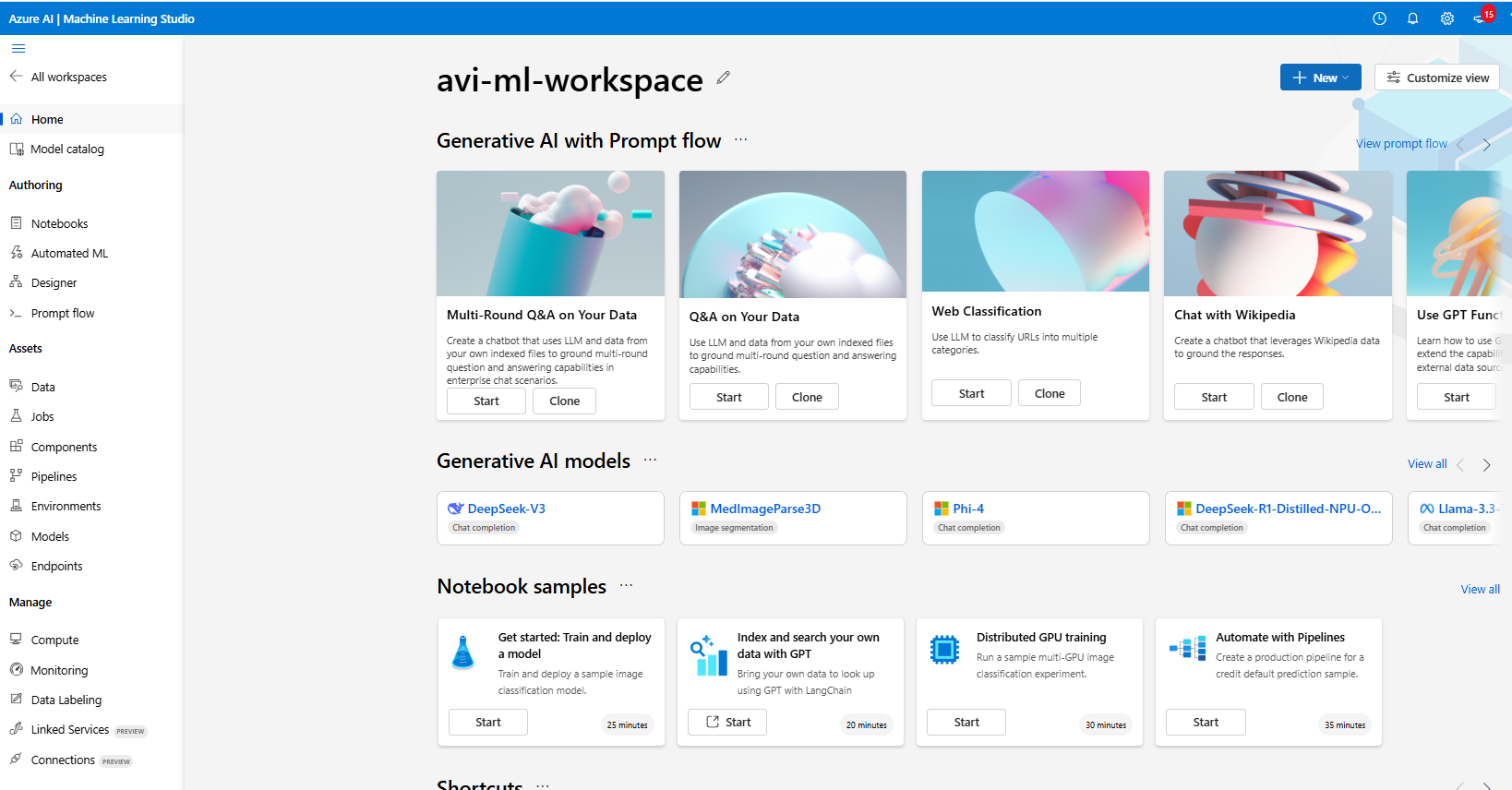


CREATED RESOURCES



* The ML workspace is a separate interface , hence to launch the workspace 🡪 Click on Launch Studio





## WORKING WITH ML WORKSPACE

1. **AZURE ML DESIGNER**
   1. Once the workspace is set up, we can access Azure ML Designer, a visual interface within Azure ML that allows us to build machine learning pipelines using pre-built components.
   2. This eliminates the need for extensive coding or development.
2. **CREATING A NEW PIPELINE**
   1. In Azure ML Designer, you create a new pipeline to start building your machine learning model. A pipeline in Azure ML is a series of connected components that define the workflow and operations performed on the data.
3. **USING PRE-BUILT COMPONENTS**
   1. Azure ML Designer provides a set of pre-built components that can be used in the pipeline.
   2. These components represent specific tasks or operations, such as data ingestion, data preprocessing, feature extraction, model training, and evaluation.
   3. We can choose from a variety of components based on your specific requirements.
4. **DRAGGING COMPONENTS ONTO THE CANVAS**
   1. In the Azure ML Designer canvas, we can drag and drop the pre-built components onto the canvas to build the model pipeline.
   2. Components can be connected to define the flow of data and the sequence of operations.
5. **CONFIGURING AND CONNECTING COMPONENTS**:
   1. After placing the components on the canvas, we can configure their properties and connect them to define the data flow and processing steps.
   2. This includes specifying input and output connections, setting parameters, and adjusting component settings.

A screenshot of a diagram

AI-generated content may be incorrect.

1. **TRAINING AND BUILDING THE MODEL**:
   1. Once the pipeline is set up, we can start the training and building process. This involves providing the necessary data to the pipeline and executing the components in the desired sequence.
   2. Data flows through the pipeline, and each component performs its assigned task, such as data transformation, model training, or model evaluation.
2. **MONITORING AND ITERATING**:
   1. Azure ML provides monitoring capabilities to track the progress and performance of the pipeline.
   2. We can monitor metrics, logs, and outputs to assess the model's performance and iterate on the pipeline configuration if necessary.

## BUILDING CLASSIFICATION ML PIPELINE

* The classification model is used to predict the values that fall in into simple category.
* In the below example we will use a pre-built dataset to train the model (Azure ML workspace has multiple pre-built data set). We will be using “***adult\_census\_income\_binary\_classification***”. In real world scenario we need to get the data set from one or multiple sources and then clean the data, which can then be used to train the ML model.
* Based in the below dataset model give the prediction of income (label)(*Income has two categories >50k or <=50k*)

|  |
| --- |
| Our aim is to build a machine learning model. This model will be used to classify the income of individuals into 2 classes - <=50K or > 50K |

A screenshot of a computer

AI-generated content may be incorrect.

### STEP 1: SPLIT DATA

A screenshot of a computer

AI-generated content may be incorrect.

#### DATA SET

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

#### SPLIT DATA

|  |  |
| --- | --- |
|  | * The dataset is then further split in certain ratio e.g. 70% of data can be used to train the model and 30% for validation * For the drag & drop the “Split data” component and configure the desired ratio of data split |

A screenshot of a computer

AI-generated content may be incorrect.

#### SETTING UP THE COMPUTE INSTANCE

|  |  |
| --- | --- |
|  | * To build/ train the ML model, we need to run the pipeline. * To execute the pipeline, we need a compute instance where the pipeline will execute. The compute instance is a Azure VM under the hood * Note: The ML algorithm is set of instructions / code. This instruction must iterate through large datasets to look at the pattern in the data. Hence, it needs a compute machine to run the instruction and build the ML model |

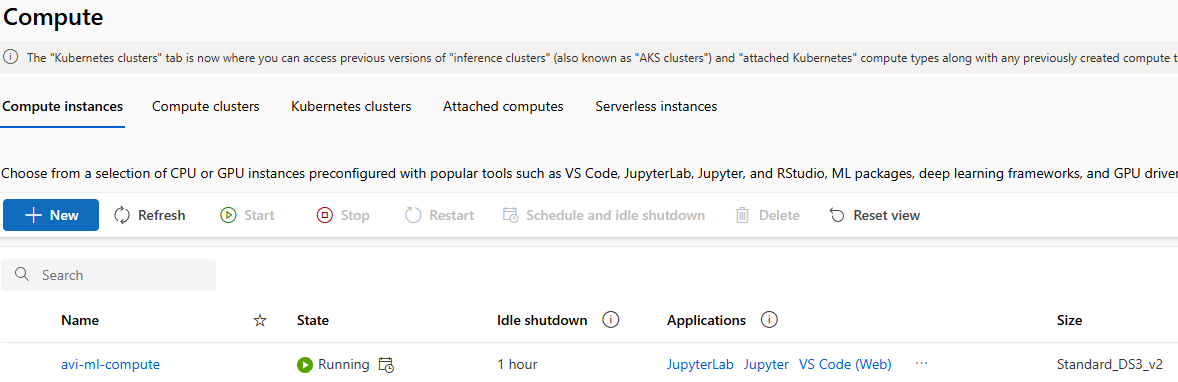
* The compute instance VM are available in different SKUs. For the below example- we will select ***Standard\_DS3\_v2*** VM
* For this example - we will keep other options as default and create the VM

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



#### RUNNING THE PIPELINE

* After the compute instance is setup, we can now run the pipeline (Note – the pipeline is not yet complete)
* To run the pipeline, we need to create an experiment
* The execution aka experiment will create a Job. This Job will execute on the compute machine
* Go to Designer 🡪 Configure & Submit

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

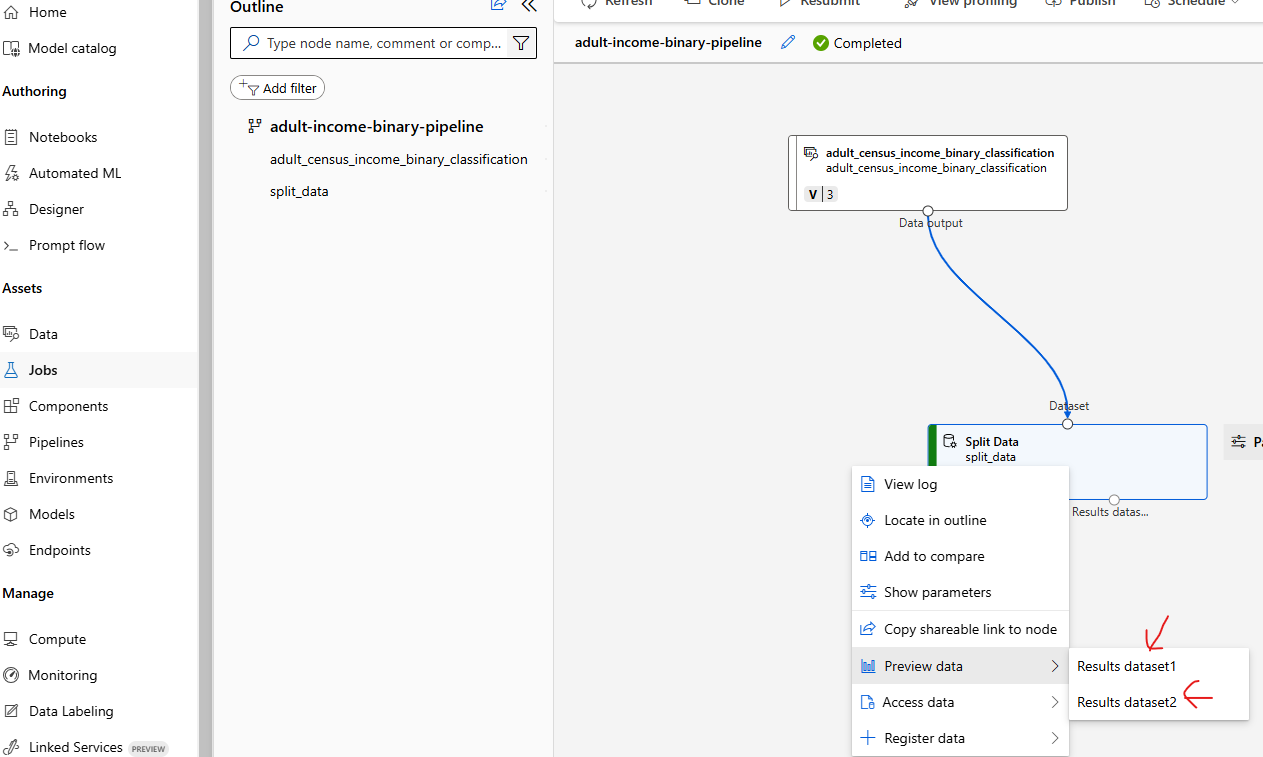
AI-generated content may be incorrect.

* The Job will download the entire dataset in the compute machine and then split the data

A screenshot of a computer

AI-generated content may be incorrect.

* The split data component will split the data in to 2 datasets in 70-30 ratio.



### STEP 2: TRAIN MODEL

A screenshot of a computer

AI-generated content may be incorrect.

#### CONFIGURE THE LABEL

To configure the model

1. Click on Pipeline Interface 🡪 Train Model Component 🡪 Edit Column

A screenshot of a computer

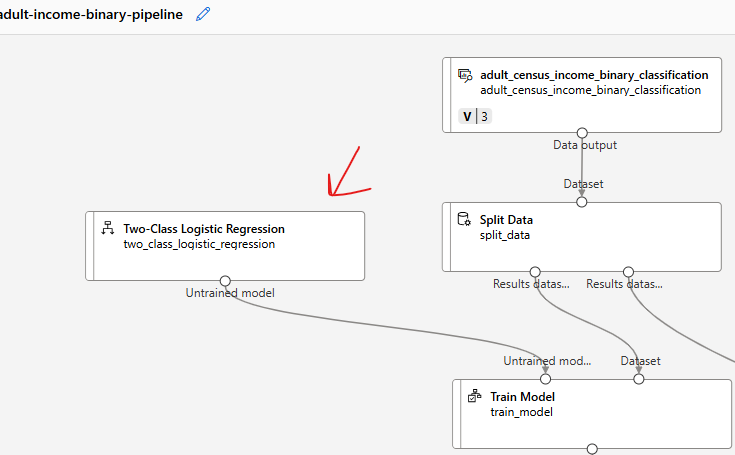
AI-generated content may be incorrect.

* Provide the name of the column name of the “**Label**” 🡪 Save

A screenshot of a computer

AI-generated content may be incorrect.

#### CONFIGURE THE ALGORITHM



### STEP 3: SCORE MODEL

* Score model is to test the model performance. For that we need to feed the 30% of the test data and Model as an input

A screenshot of a computer

AI-generated content may be incorrect.

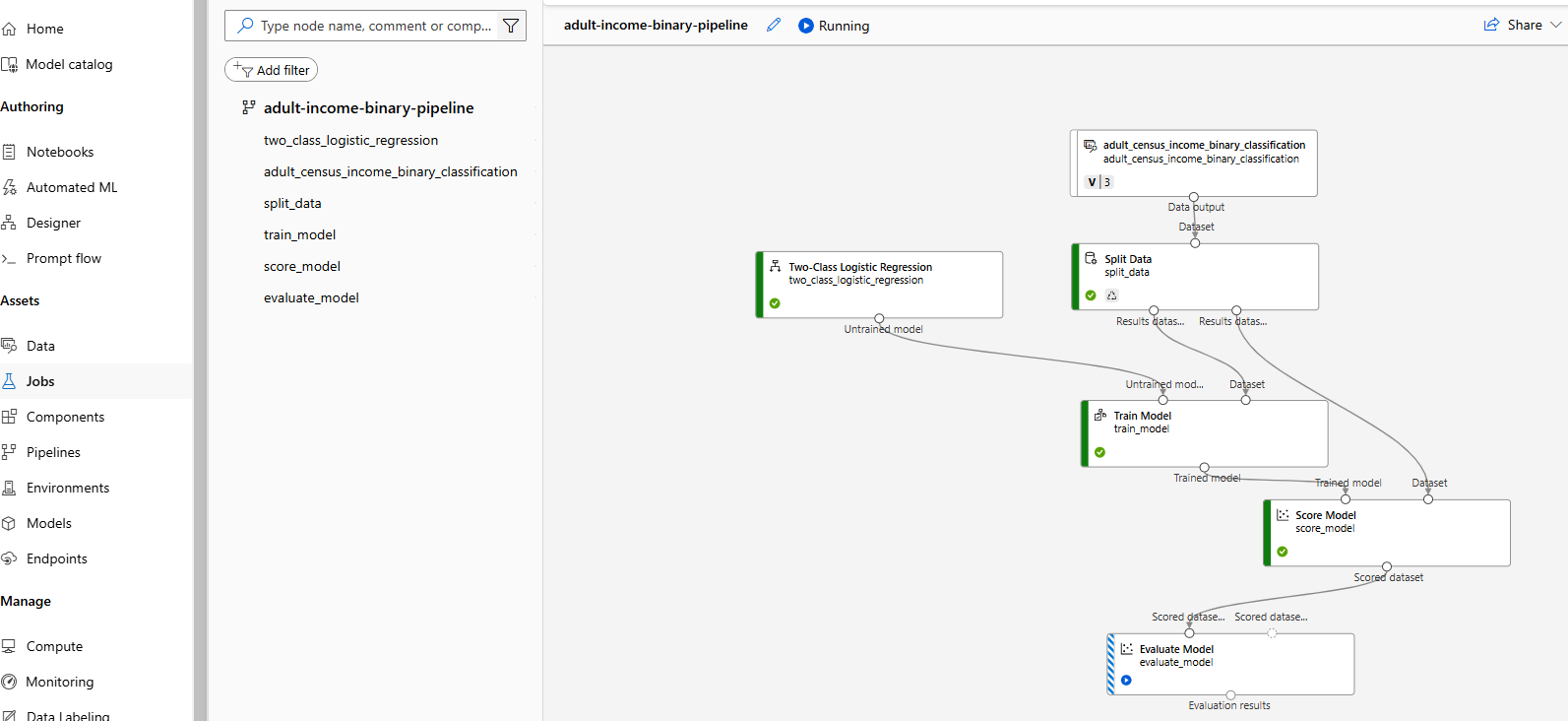
### STEP 4: EVALUATE MODEL

A screenshot of a computer

AI-generated content may be incorrect.

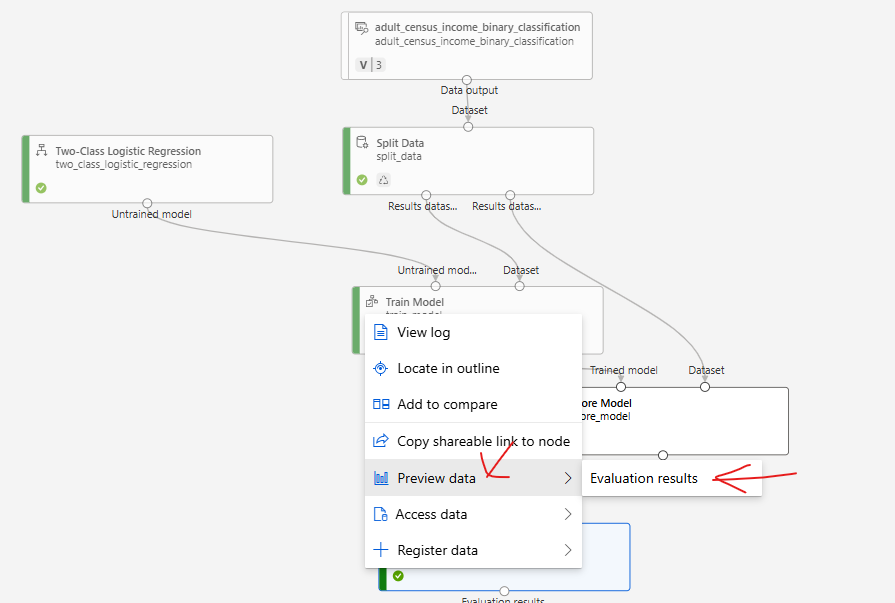
### STEP 5: RUN THE PIPELINE – EXPERIMENT

* Designer 🡪 Configure & Submit 🡪 Then Configure the Experiment 🡪 Review + Submit
* This will create & submit a new Job that will run the entire pipeline
* To view the status of Job . Go to Job and open the experiment



### EVALUATION OF MODEL RESULT

* **Every model has their specific evaluation metrices. Below are the details of metrices of classification model.**



* **There are different metrics that can be used to evaluate model performance, such as Accuracy and Precision. The closer these metrics are to 1, the more accurate the model is.**
* **One of the important metrics is “Confusion Metrics”**

A screenshot of a computer

AI-generated content may be incorrect.

#### CONFUSION MATRICES

* Confusion matrices are the way to evaluate the performance of **a classification model**.
* It is a table that shows the number of correct and incorrect predictions made by the model.
* The table is divided into four quadrants:
  1. *true positives (correctly predicted positive instances),*
  2. *true negatives (correctly predicted negative instances),*
  3. *false positives (incorrectly predicted positive instances)*
  4. *false negatives (incorrectly predicted negative instances).*

By analyzing the values in the confusion matrix, various performance metrics such as accuracy, precision, recall, and F1 score can be calculated to assess the model's effectiveness.

|  |  |
| --- | --- |
| **A screenshot of a graph  AI-generated content may be incorrect.** |  |

## DEPLOYING THE ML MODEL

Once the evaluation is done, we can go ahead and do the deployment so that request can be submitted with an input.

A diagram of a pipeline

AI-generated content may be incorrect.

***The ML model will be deployed to AKS cluster(Compute)***

|  |  |
| --- | --- |
| Example | INPUT DATA TO MODEL AS POST REQUEST |
| * The expectation is to predict the income when the following input is provided to the model | **{"Inputs": {"WebServiceInput0": [**  **{ "age": 39,**  **"workclass": "State-gov",**  **"fnlwgt": 77516,**  **"education": "Bachelors",**  **"education-num": 13,**  **"marital-status": "Never-married",**  **"occupation": "Adm-clerical",**  **"relationship": "Not-in-family",**  **"race": "White",**  **"sex": "Male",**  **"capital-gain": 0,**  **"capital-loss": 0,**  **"hours-per-week": 60,**  **"native-country": "United-States",**  **"income": ""**  **}**  **]**  **}}** |

* After the successful run job- we need to create inference pipeline

### INFERENCE PIPELINE

* **An inference pipeline is a way to deploy and manage an end-to-end workflow for deploying machine learning models as web services**.
* It allows us to create a pipeline that consists of multiple steps, including data preprocessing, model scoring, and post-processing.
* The inference pipeline in Azure ML workspace provides a streamlined and automated process for deploying models and making predictions on new data. It helps in simplifying the deployment process, managing dependencies, and scaling the inference workload.
* With an inference pipeline, you can encapsulate the entire workflow and deploy it as a single endpoint. This makes it easy to integrate the model into applications or systems for real-time predictions.
* **The inference pipeline can be created and managed using Azure ML SDK, Azure Portal, or other Azure ML tools.**

#### CREATING AN INFERENCE PIPELINE

* Go to jobs 🡪 Open the Job

A screenshot of a computer

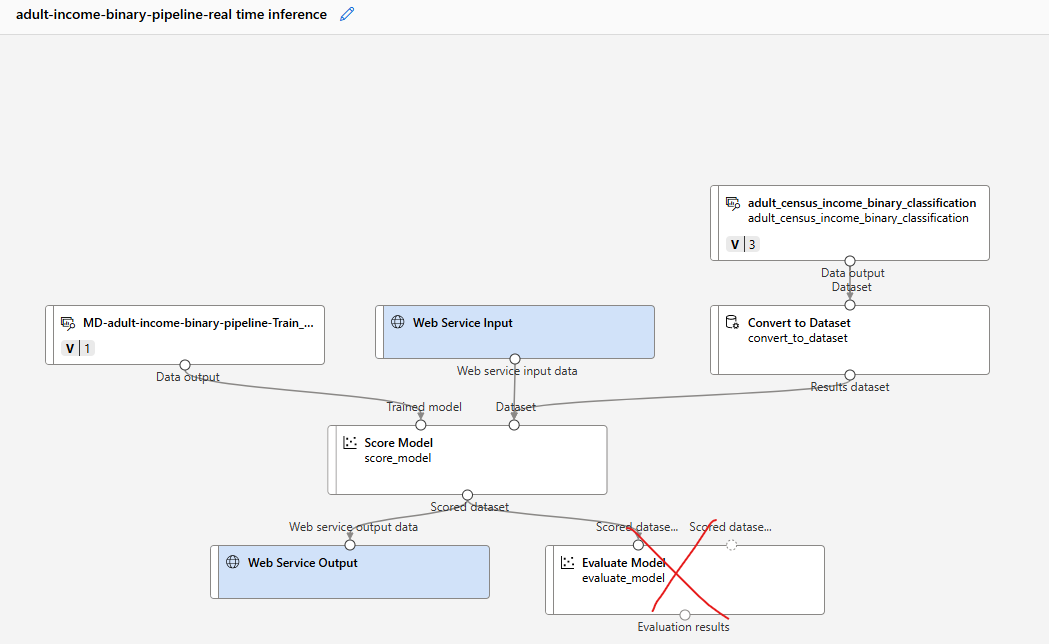
AI-generated content may be incorrect.

* Create a “Real time Inference Pipeline” .

A screenshot of a computer

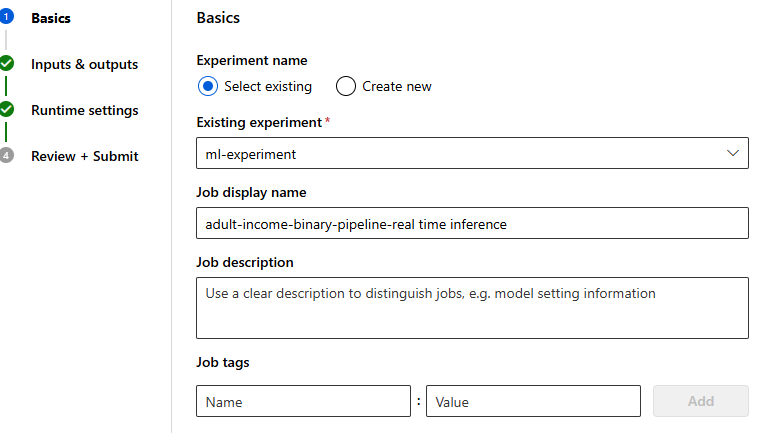
AI-generated content may be incorrect.

* This will create a new pipeline, with additional component “Web Service Input” and “Web Service Output”
* Note: We can remove the Evaluate Model component (as we have already evaluated the model)



#### RUN THE INFERENCE PIPELINE

* Save and click on “Configure and Submit” to configure the Job



A screenshot of a computer

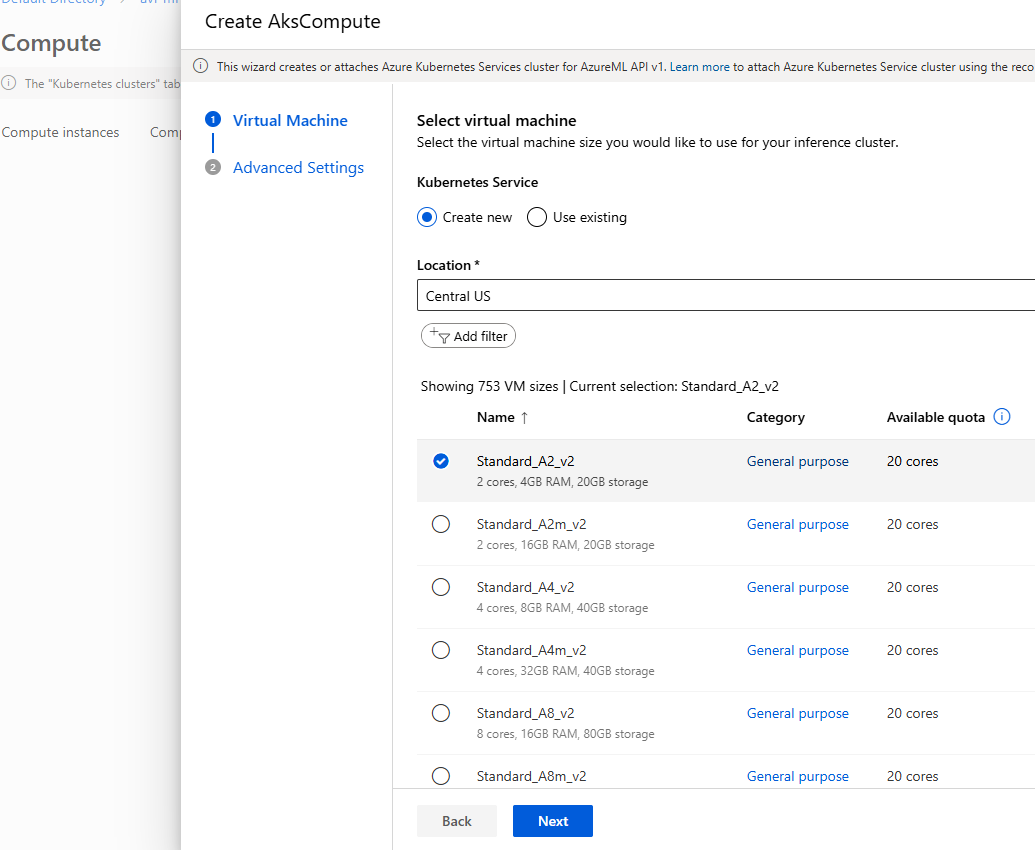
AI-generated content may be incorrect.

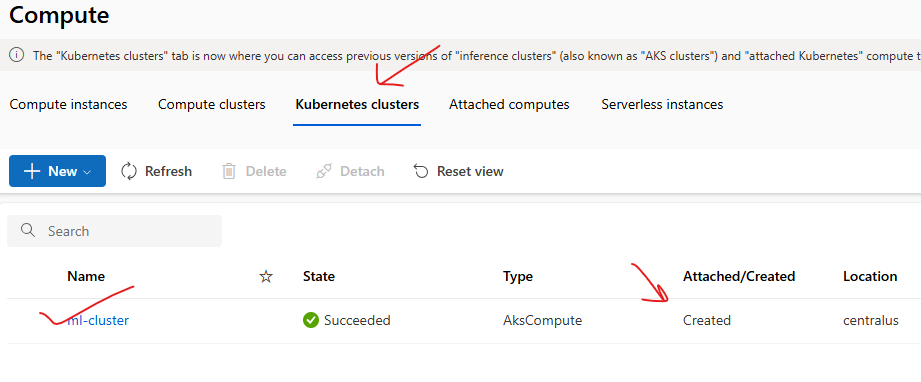
#### COMPUTE

* We will now create a compute to deploy the ML model

A screenshot of a computer

AI-generated content may be incorrect.





A screenshot of a computer

AI-generated content may be incorrect.

* Go to job 🡪 open the Inference pipeline Job 🡪 Deploy
* This will deploy the model to the compute instance (Kubernetes Cluster)

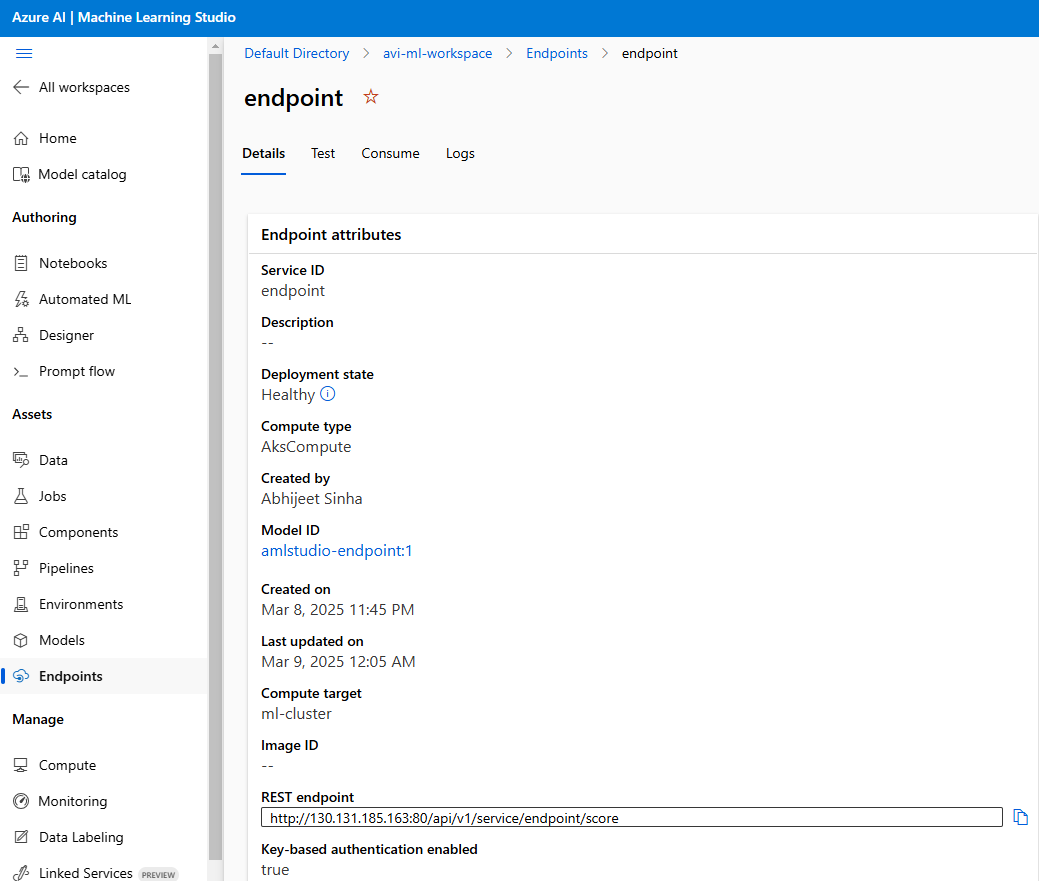
A screenshot of a computer

AI-generated content may be incorrect.

* After the deploy it will create an endpoint using which the ML model can be called. Note – The deployment status can be check using *Endpoint 🡪 Deployment State.*
* After the deployment the deployment state will be updated as “**Healthy**”

A screenshot of a computer

AI-generated content may be incorrect.



## TESTING THE MODEL

|  |  |
| --- | --- |
| INPUT DATA TO MODEL AS POST REQUEST |  |
| **{"Inputs": {"WebServiceInput0": [**  **{ "age": 39,**  **"workclass": "State-gov",**  **"fnlwgt": 77516,**  **"education": "Bachelors",**  **"education-num": 13,**  **"marital-status": "Never-married",**  **"occupation": "Adm-clerical",**  **"relationship": "Not-in-family",**  **"race": "White",**  **"sex": "Male",**  **"capital-gain": 0,**  **"capital-loss": 0,**  **"hours-per-week": 60,**  **"native-country": "United-States",**  **"income": ""**  **}**  **]**  **}}** |  |

### CONSUMING MODEL USING ENDPOINT

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

