

Training Component - Inputs and Outputs Explanation

Component Overview

The `train_model` component is a Kubeflow pipeline component that trains a Linear Regression model for predicting medical insurance charges.

Inputs

1. `X_train_path` - Training Features

- **Type:** `dsl.InputPath(str)`
- **Description:** Path to the serialized training feature matrix
- **Format:** Pickle file (`.pkl`)
- **Content:** Pandas DataFrame containing preprocessed features including:
 - `age`: Patient's age
 - `sex`: Gender (label encoded: 0 or 1)
 - `bmi`: Body Mass Index
 - `children`: Number of dependents
 - `smoker`: Smoking status (label encoded: 0 or 1)
 - `region`: Geographic region (label encoded: 0-3)
- **Source:** Output from the preprocessing component

2. `y_train_path` - Training Labels

- **Type:** `dsl.InputPath(str)`
 - **Description:** Path to the serialized training target variable
 - **Format:** Pickle file (`.pkl`)
 - **Content:** Pandas Series containing insurance charges (continuous numeric values)
 - **Source:** Output from the preprocessing component
-

Outputs

1. `model_output_path` - Trained Model

- **Type:** `dsl.OutputPath(str)`
 - **Description:** Path where the trained model artifact is saved
 - **Format:** Pickle file (`.pkl`)
 - **Content:** Serialized scikit-learn LinearRegression model object
 - **Usage:** This model can be loaded in subsequent components for:
 - Making predictions on test data
 - Evaluation and metrics calculation
 - Deployment in production environments
-

Why These Input/Output Types?

InputPath vs Regular Parameters

- **InputPath** is used instead of passing raw data as parameters because:
 - Handles large datasets efficiently
 - Automatically manages artifact storage in Kubeflow
 - Enables data lineage tracking
 - Supports proper dependency management between pipeline steps

OutputPath Benefits

- **OutputPath** automatically:
 - Creates a unique file path for the output
 - Stores the artifact in Kubeflow's artifact store
 - Makes the output available to downstream components
 - Enables versioning and reproducibility

Data Flow Example

```
Preprocessing Component
  ↓ (outputs)
X_train_path: /tmp/artifacts/X_train_1234.pkl
y_train_path: /tmp/artifacts/y_train_1234.pkl
  ↓ (inputs to train_model)
Training Component
  ↓ (output)
model_output_path: /tmp/artifacts/model_5678.pkl
```

Component Signature

```
@dsl.component(
    base_image="python:3.9",
    packages_to_install=["pandas", "scikit-learn"]
)
def train_model(
    X_train_path: dsl.InputPath(str),      # Input: Features
    y_train_path: dsl.InputPath(str),      # Input: Labels
    model_output_path: dsl.OutputPath(str) # Output: Trained model
):
    # Training logic here
    pass
```

Key Takeaways

1. **Inputs are paths, not data:** Components receive file paths to serialized data, not the actual data objects
2. **Automatic artifact management:** Kubeflow handles file storage, retrieval, and versioning
3. **Type safety:** Using typed paths ensures proper data flow between components
4. **Reusability:** This component can be used in any pipeline that provides compatible training data