Microsoft Azure

Manage and review models in Azure Machine Learning

Agenda

- Register an MLflow model in Azure Machine Learning
- Manage and compare models in Azure Machine Learning

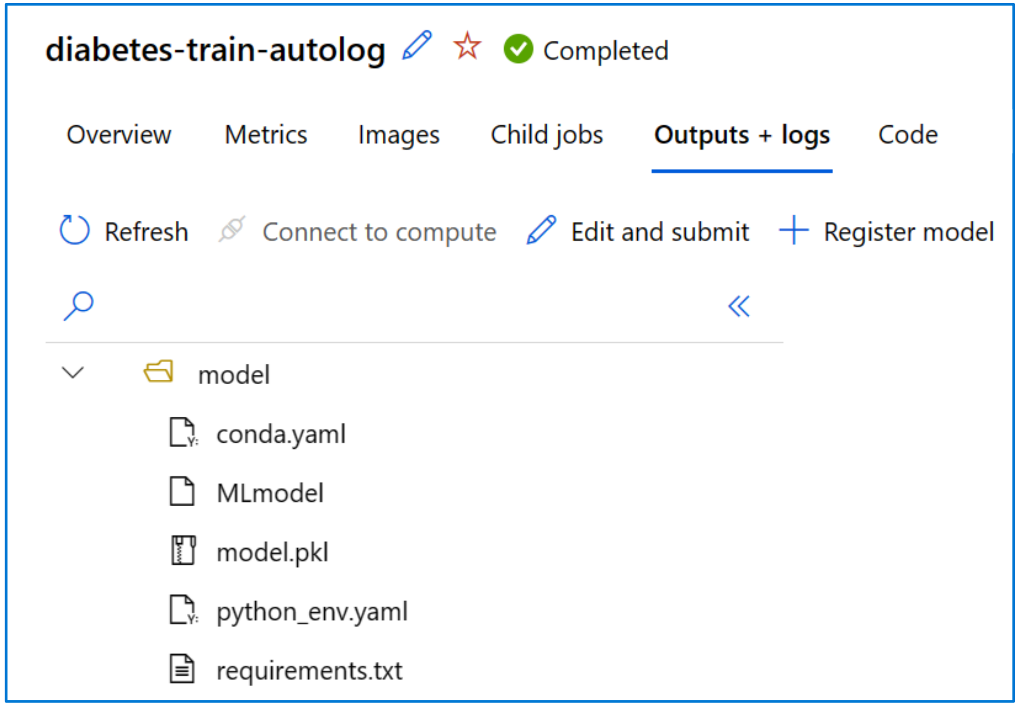
Register an MLflow model in Azure Machine Learning

Log a model with MLflow

MLflow allows you to log a model as an **artifact**, or as a **model**.

- When you log a model as an artifact, the model is treated as a file.
- When you log a model as a model, you're adding information to the registered model that enables you to use the model directly in pipelines or deployments.

When you log as a model, an MLmodel file is created in the output directory. The MLmodel file contains the model's metadata, which allows for model traceability.



Understand the MLmodel file format

The MLmodel file may include:

- **artifact_path**: During the training job, the model is logged to this path.
- **flavor**: The machine learning library with which the model was created.
- **model_uuid**: The unique identifier of the registered model.
- **run_id**: The unique identifier of job run during which the model was created.
- **signature**: Specifies the schema of the model's inputs and outputs:
  - **inputs**: Valid input to the model. For example, a subset of the training dataset.
  - **outputs**: Valid model output. For example, model predictions for the input dataset.

Use autologging to log a model

When you train a model, you can include **mlflow.autolog()** to enable autologging.

The model is logged when the **.fit()** method is called. The framework you use to train your model is identified and included as the **flavor** of your model.

Optionally, you can specify which flavor you want your model to be identified as by using **mlflow.<flavor>.autolog()**.

Manually log a model (1/2)

You can customize the signature by inferring the schema from the training dataset and model predictions.

Python

```python
import mlflow.sklearn
from mlflow.models.signature import infer_signature

signature = infer_signature(iris_train, clf.predict(iris_train))

mlflow.sklearn.log_model(clf, "iris_rf", signature=signature)
```

Manually log a model (2/2)

You can define the schema for the input and output data of your model.

Python

```python
from mlflow.models.signature import ModelSignature
from mlflow.types.schema import Schema, ColSpec

input_schema = Schema([
        ColSpec("double", "sepal length (cm)"),
        ColSpec("double", "sepal width (cm)"),
        ColSpec("double", "petal length (cm)"),
        ColSpec("double", "petal width (cm)"), ])
output_schema = Schema([ColSpec("long")])

signature = ModelSignature(inputs=input_schema, outputs=output_schema)
```

Exercise - Log and register models with MLflow

In this exercise, you will:

Task 1:

Log models with MLflow.

Task 2:

Register an MLflow model in the Azure Machine Learning model registry.

Instructions

Follow these instructions to complete the exercise:

1. View the exercise repo at https://microsoftlearning.github.io/mslearn-azure-ml/.
2. Complete the **Log and register models with MLflow** exercise.

Knowledge check

A data scientist trains and logs a model with MLflow. When the data scientist deploys the model, the schema of the model's input and output isn't correct. What should the data scientist customize to fix the issue?

A data scientist trained a deep learning model with TensorFlow. The deployed model is compute-intensive and needs to use the most optimal inference server for similar workloads. Which model type is compatible with compute-intensive and no-code deployments?

Knowledge check

A data scientist trains and logs a model with MLflow. When the data scientist deploys the model, the schema of the model's input and output isn't correct. What should the data scientist customize to fix the issue?

- [ ] Customize the model's environment.
- [ ] Change the model's flavor.
- [x] Customize the model's signature.

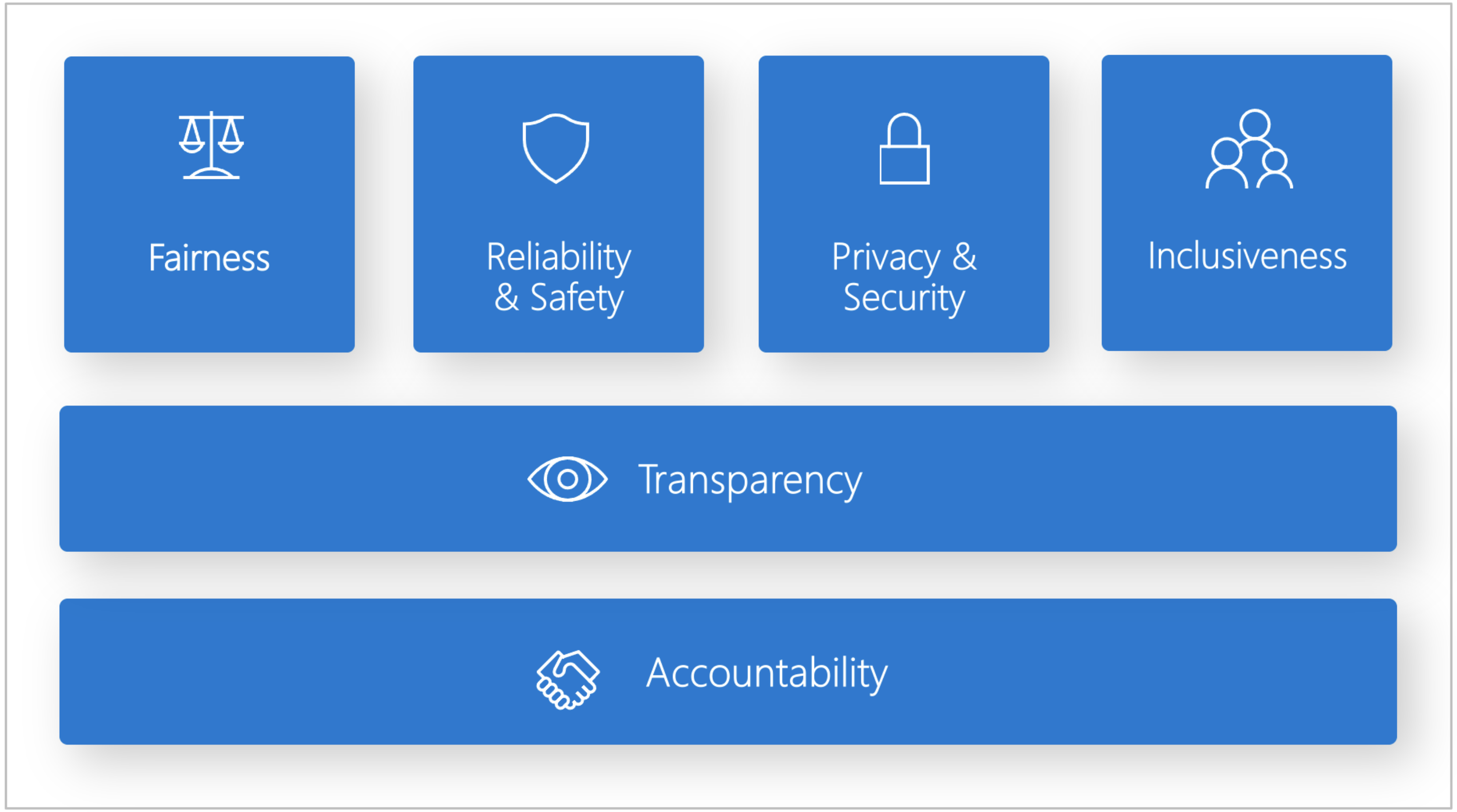
A data scientist trained a deep learning model with TensorFlow. The deployed model is compute-intensive and needs to use the most optimal inference server for similar workloads. Which model type is compatible with compute-intensive and no-code deployments?

- [ ] MLflow
- [x] Triton
- [ ] Custom

Manage and compare models in Azure Machine Learning

Understand Responsible Artificial Intelligence (AI)

Responsible Artificial Intelligence (**Responsible AI**) is an approach to developing, assessing, and deploying AI systems in a safe, trustworthy, and ethical way.



Reliability and safety: Understand your datasets

Use data analysis to identify issues or overrepresentation and underrepresentation and to see how the data is clustered in the dataset.

Use data analysis when you need to:

- Explore your dataset statistics by selecting different filters to slice your data into different dimensions (also known as cohorts).
- Understand the distribution of your dataset across different cohorts and feature groups.
- Determine whether your findings related to fairness, error analysis, and causality (derived from other dashboard components) are a result of your dataset's distribution.
- Decide in which areas to collect more data to mitigate errors that come from representation issues, label noise, feature noise, label bias, and similar factors.

Reliability and safety: Assess errors in your model

Identify erroneous cohorts of data. Explore whether your model underperforms for specific demographic groups.

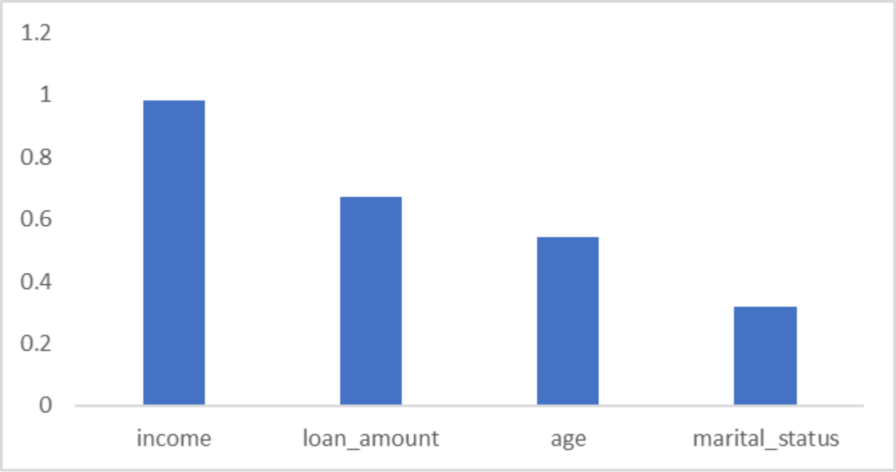
- **Error tree**: Partitions the data into interpretable subgroups.
- **Error heatmap**: Slices the data based on one- or two-dimensional grids of input features.

Transparency: Interpret your model

Aggregate feature importance

Overall feature importance for all test data

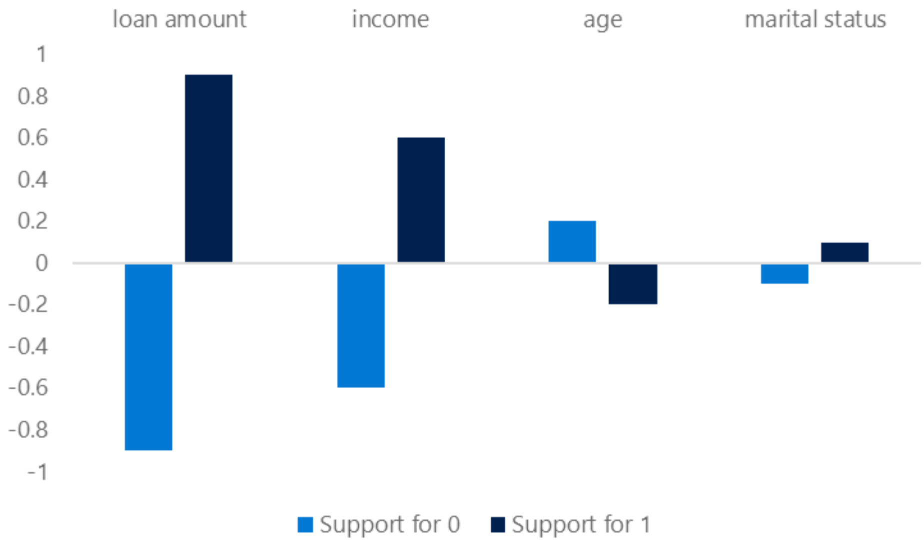
Indicates the relative influence of each feature on the predicted label



Individual feature importance

Feature importance for an individual prediction

In classification, this shows the relative support for each possible class per feature



Fairness: Mitigate disparity

The model overview and fairness assessment evaluates the performance of your model and evaluates your model's group fairness issues.

- **Disparity in model performance**: Do different cohorts of the data perform differently when comparing selected performance metrics?
- **Disparity in selection rate**: Are there cohorts of the data that more often get a favorable prediction?

Exercise – Compare and evaluate models

In this exercise, you will:

Task 1:

Prepare your data and create a Responsible AI dashboard.

Task 2:

Explore the Responsible AI dashboard.

Instructions

Follow these instructions to complete the exercise:

1. View the exercise repo at https://microsoftlearning.github.io/mslearn-azure-ml/.
2. Complete the **Compare and evaluate models** exercise.

Knowledge check

You are training a binary classification model to support admission approval decisions for a college degree program. How can you evaluate if the model is fair, and doesn't discriminate based on ethnicity?

- [x] Compare disparity between selection rates and performance metrics across ethnicities.
- [ ] Remove the ethnicity feature from the training dataset.
- [ ] Evaluate each trained model with a validation dataset and use the model with the highest accuracy score.

You have trained a model, and you want to quantify the influence of each feature on a specific individual prediction. What kind of feature importance should you examine?

- [ ] Aggregate feature importance
- [x] Individual feature importance

Microsoft Azure