

1.Cluster

<https://community.cloud.databricks.com/?o=853096632416255#notebook/546359089057236/command/546359089057272>

2.Published Link : Valid till 6 Months

<https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bfcf/853096632416255/546359089057236/6384386000767703/latest.html>

User Path : /Users/vijaygurung555@gmail.com/ML_Flow 01:17:30

Experiment ID : 546359089057236

Artifact Location : dbfs:/databricks/mlflow-tracking/546359089057236

Description : **THIS IS MY REGRESSOR ML FLOW MODEL **

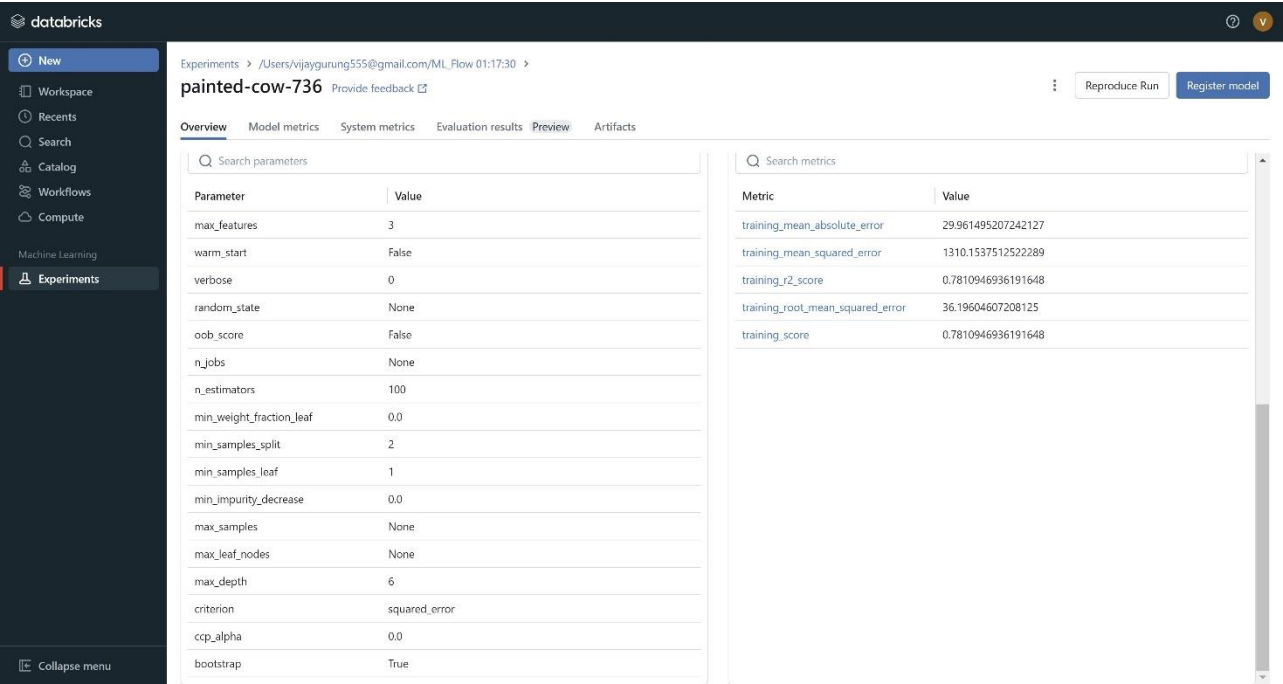
Matrix Run ID : [3383db4dd63a49ea8b6ef88f0255d424]

Overview :

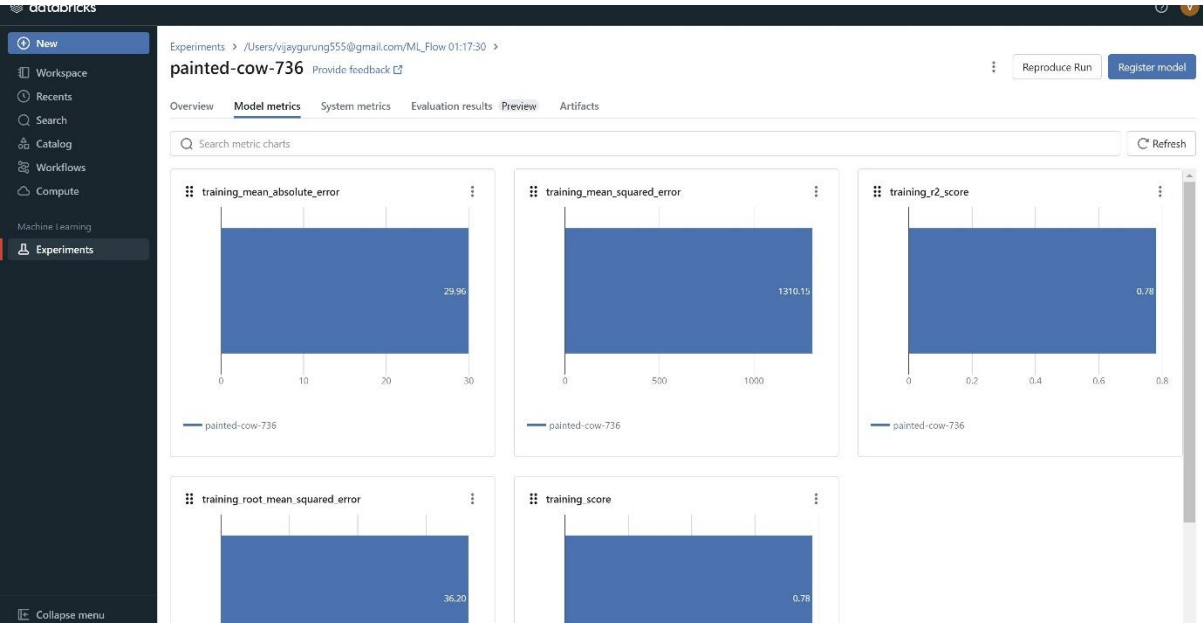
The screenshot displays the Databricks MLflow interface. On the left is a sidebar with navigation options: New, Workspace, Recents, Search, Catalog, Workflows, Compute, Machine Learning, and Experiments (selected). The main panel shows the 'classifier' experiment details. At the top, there's a breadcrumb trail: Experiments > /Users/vijaygurung555@gmail.com/ML_Flow 01:17:30 > classifier. Below this are tabs for Overview, Model metrics, System metrics, Evaluation results, Preview, and Artifacts. The 'Overview' tab is active, showing a description field (currently empty) and a 'Details' table. The table lists the following information: Created at (2024-08-17 01:49:01), Created by (vijaygurung555@gmail.com), Experiment ID (546359089057236), Status (Finished), Run ID (3383db4dd63a49ea8b6ef88f0255d424), Duration (7.4s), Datasets used (dataset (6580b268) Train +1), Tags (estimator_name: KNeighborsClassifier, estimator_class: sklearn.neighbors_classification.K...), Source (ML_Flow 01:17:30), Logged models (sklearn), and Registered models (—). Below the table are sections for Parameters (8) and Metrics (8), each with a search bar and a table header (Parameter/Value and Metric/Value respectively).

Details	
Created at	2024-08-17 01:49:01
Created by	vijaygurung555@gmail.com
Experiment ID	546359089057236
Status	Finished
Run ID	3383db4dd63a49ea8b6ef88f0255d424
Duration	7.4s
Datasets used	dataset (6580b268) Train +1
Tags	estimator_name: KNeighborsClassifier estimator_class: sklearn.neighbors_classification.K...
Source	ML_Flow 01:17:30
Logged models	sklearn
Registered models	—

Parameters :



Matrix Graph :



ML Flow Model : Sources Code : Apache Spark ML Flow:

The screenshot shows the Databricks MLflow Model page for an experiment named 'painted-cow-736'. The left sidebar contains navigation options: New, Workspace, Recents, Search, Catalog, Workflows, Compute, Machine Learning, and Experiments. The main content area is divided into several tabs: Overview, Model metrics, System metrics, Evaluation results, Preview, and Artifacts. The 'Artifacts' tab is selected, showing a list of artifacts under the 'model' folder, including metadata, MLmodel, conda.yaml, model.pkl, python_env.yaml, requirements.txt, and estimator.html. The 'model' artifact is selected, displaying its path and a description: 'MLflow Model'. Below this, the 'Model schema' is shown, detailing the input and output types. The 'Validate the model before deployment' section provides a code snippet for validating the model. The 'Make Predictions' section is also visible.

Experiments > /Users/vijaygurun555@gmail.com/ML_Flow 01:17:30 >

painted-cow-736 Provide feedback

Reproduce Run Register model

Overview Model metrics System metrics Evaluation results Preview **Artifacts**

model

Path: dbfs:/databricks/mlflow_tracking/546359089057236/3178e6548bb24034a2859d056d19f1c4/artifacts/model

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. You can also register it to the model registry to version control and deploy as a REST endpoint for real time serving.

Model schema

Input and output schema for your model. Learn more

Name	Type
Inputs (1)	
-(required)	Tensor (dtype: float64, shape: [-1,10])
Outputs (1)	
-(required)	Tensor (dtype: float64, shape: [-1])

Validate the model before deployment

Run the following code to validate model inference works on the example payload, prior to deploying it to a serving endpoint

```
from mlflow.models import validate_serving_input

model_uri = 'runs:/3178e6548bb24034a2859d056d19f1c4/model'

# The logged model does not contain an input_example.
# Manually generate a serving payload to verify your model prior to deployment.
from mlflow.models import convert_input_example_to_serving_input

# Define INPUT_EXAMPLE via assignment with your own input example to the model
# A valid input example is a data instance suitable for pyfunc prediction
serving_payload = convert_input_example_to_serving_input(INPUT_EXAMPLE)

# Validate the serving payload works on the model
validate_serving_input(model_uri, serving_payload)
```

Make Predictions

Estimator HTML

The screenshot shows the Databricks MLflow Estimator HTML page for the same experiment 'painted-cow-736'. The left sidebar is identical to the previous screenshot. The main content area shows the 'Artifacts' tab with a list of artifacts under the 'model' folder. The 'estimator.html' artifact is selected, displaying its path and a description: '4.84KB'. The code snippet for the estimator is shown, which is a RandomForestRegressor.

Experiments > /Users/vijaygurun555@gmail.com/ML_Flow 01:17:30 >

painted-cow-736 Provide feedback

Reproduce Run Register model

Overview Model metrics System metrics Evaluation results Preview **Artifacts**

estimator.html 4.84KB

Path: dbfs:/databricks/mlflow_tracking/546359089057236/3178e6548bb24034a2859d056d19f1c4/artifacts/estimator.html

```
RandomForestRegressor(max_depth=6, max_features=3)
```

ML Model : PKL File :

PKL stands for (Python pickle files)

These allow us to store the stages.

Use PKL ML used for efficiently saving, loading, sharing and reproducing models and other Python objects.