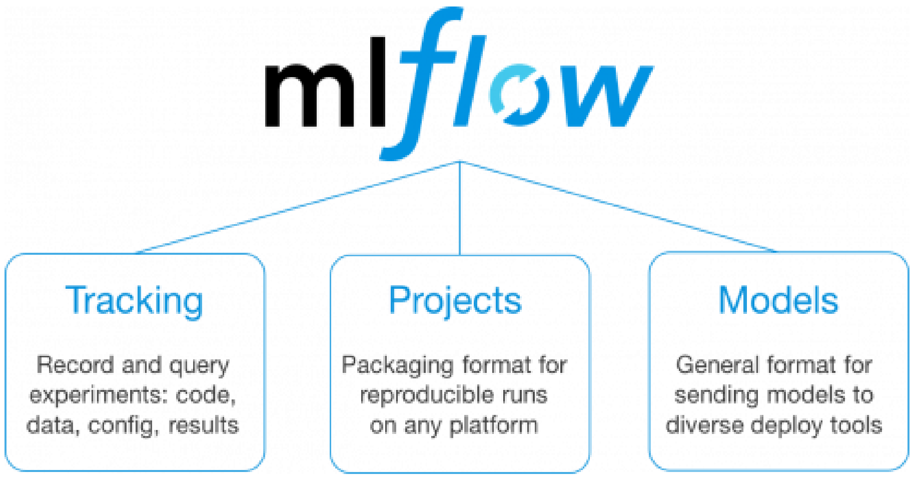
Experiment Tracking and Model Management

In this project, we create an experiment tracking our dataset i.e diamond.csv using ML Flow.

ML Flow


mlflow
Tracking
Record and query experiments: code, data, config, results
Projects
Packaging format for reproducible runs on any platform
Models
General format for sending models to diverse deploy tools

What is MLFlow?

MLFlow is a framework that supports the machine learning lifecycle. This means that it has components to monitor your model during training and running, the ability to store models, load the model in production code, and create a pipeline.

MLFlow Tracking—

Tracking is maybe the most interesting feature of the framework. It allows you to create an extensive logging framework around your model. You can define custom metrics so that after a run you can compare the output to previous runs.

MlFlow Projects—

This feature allows you to create a pipeline if you so desire. This feature uses its own template to define how you want to run the model on a cloud environment. As most companies have a way to run code in production this feature might be of less interest to you.

MlFlow Models—

Finally we have the Models feature. An MLFlow Model is a standard format for packaging machine learning models that can be used in a variety of downstream tools — for example, real-time serving through a REST API or batch inference on Apache Spark.

Command to Install MLFlow—

```
pip install mlflow
mlflow ui --backend-store-uri sqlite:///mlflow.db
```

How to implement MLFlow in our application—

Step-1

```
import mlflow
```

Step 2 - Set the tracker and experiment

```
mlflow.set_tracking_uri(DATABASE_URI)
mlflow.set_experiment("EXPERIMENT_NAME")
```

Step 3 - Start a experiment run

```
with mlflow.start_run():
```

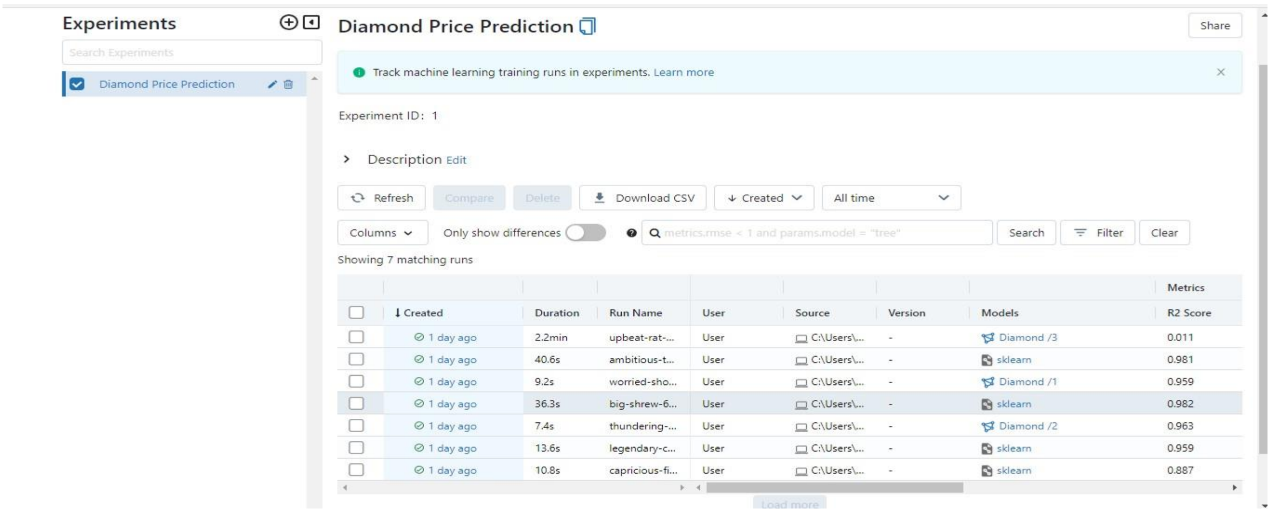
Step 4 - Logging the metadata

```
mlflow.set_tag(KEY, VALUE)
mlflow.log_param(KEY, VALUE) mlflow.log_metric(KEY, VALUE)
```

Step 5 - Logging the model and other files (2 ways)

```
Way 1 - mlflow.<FRAMEWORK>.log_model(MODEL_OBJECT, artifact_path="PATH")
Way 2 - mlflow.log_artifact(LOCAL_PATH, artifact_path="PATH")
```

MLFlow Interface Experiment Tracking—


Experiments

Diamond Price Prediction

Track machine learning training runs in experiments. Learn more

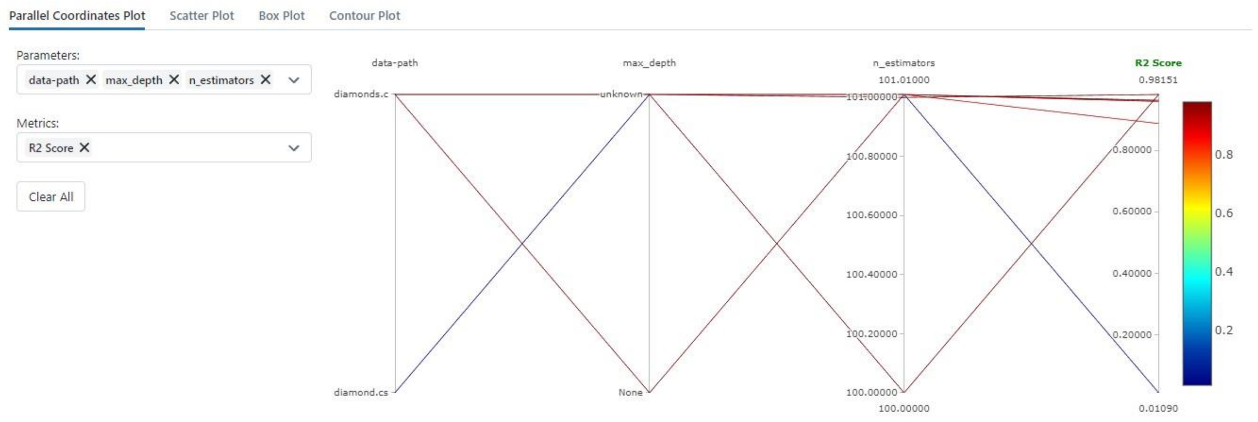
Experiment ID: 1

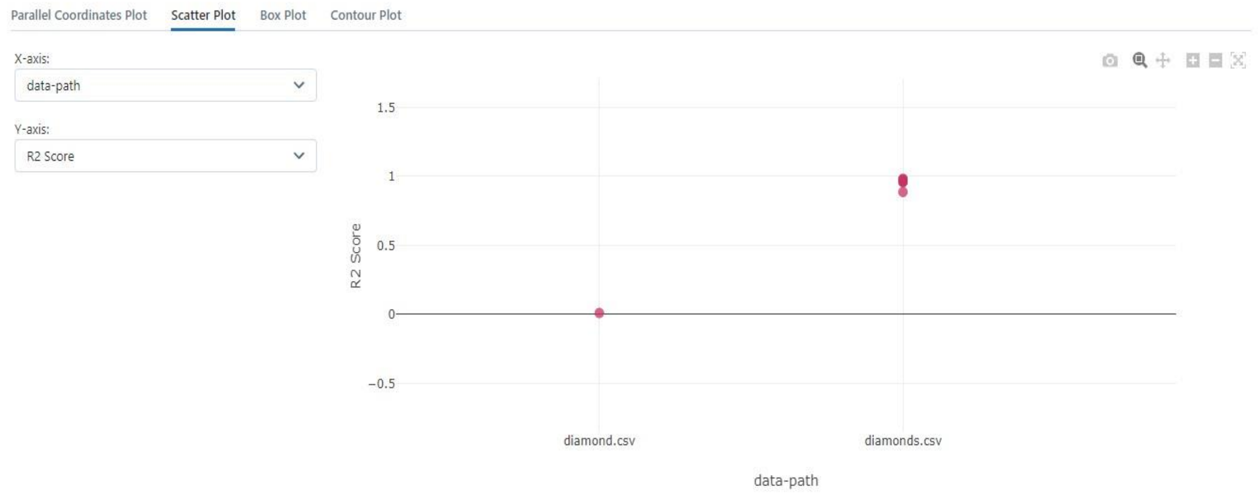
Description Edit

Showing 7 matching runs

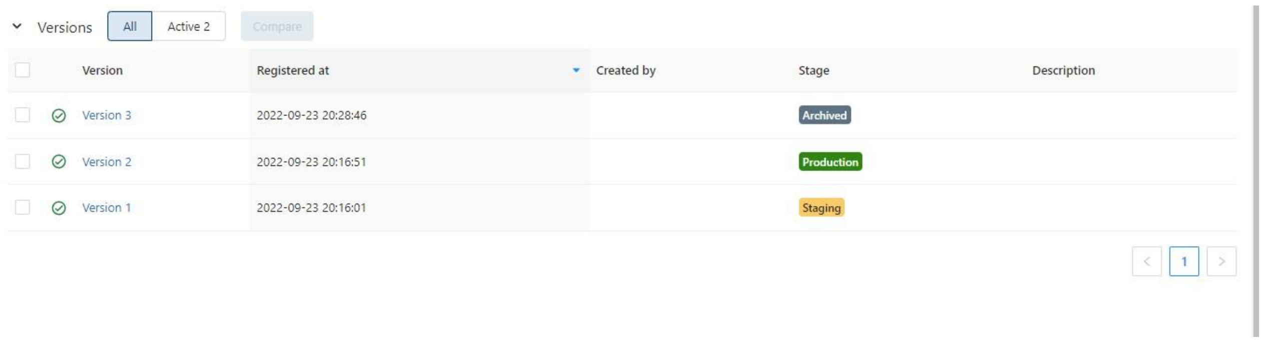
Created	Duration	Run Name	User	Source	Version	Models	R2 Score
1 day ago	2.2min	upbeat-rat-...	User	C:\Users\...	-	Diamond /3	0.011
1 day ago	40.6s	ambitious-t...	User	C:\Users\...	-	sklearn	0.981
1 day ago	9.2s	worried-sho...	User	C:\Users\...	-	Diamond /1	0.959
1 day ago	36.3s	big-shrew-6...	User	C:\Users\...	-	sklearn	0.982
1 day ago	7.4s	thundering-...	User	C:\Users\...	-	Diamond /2	0.963
1 day ago	13.6s	legendary-c...	User	C:\Users\...	-	sklearn	0.959
1 day ago	10.8s	capricious-fi...	User	C:\Users\...	-	sklearn	0.887

Comparison Graph among different models—





MLFlow Interface for Model Management—



		Version	Registered at	Created by	Stage	Description
		Version 3	2022-09-23 20:28:46		Archived	
		Version 2	2022-09-23 20:16:51		Production	
		Version 1	2022-09-23 20:16:01		Staging	

Pipeline using Workflow Orchestration tool i.e. Prefect

Install Prefect On Machine—

```
pip install prefect
```

Run Prefect—

```
prefect orion start
```

What is prefect?

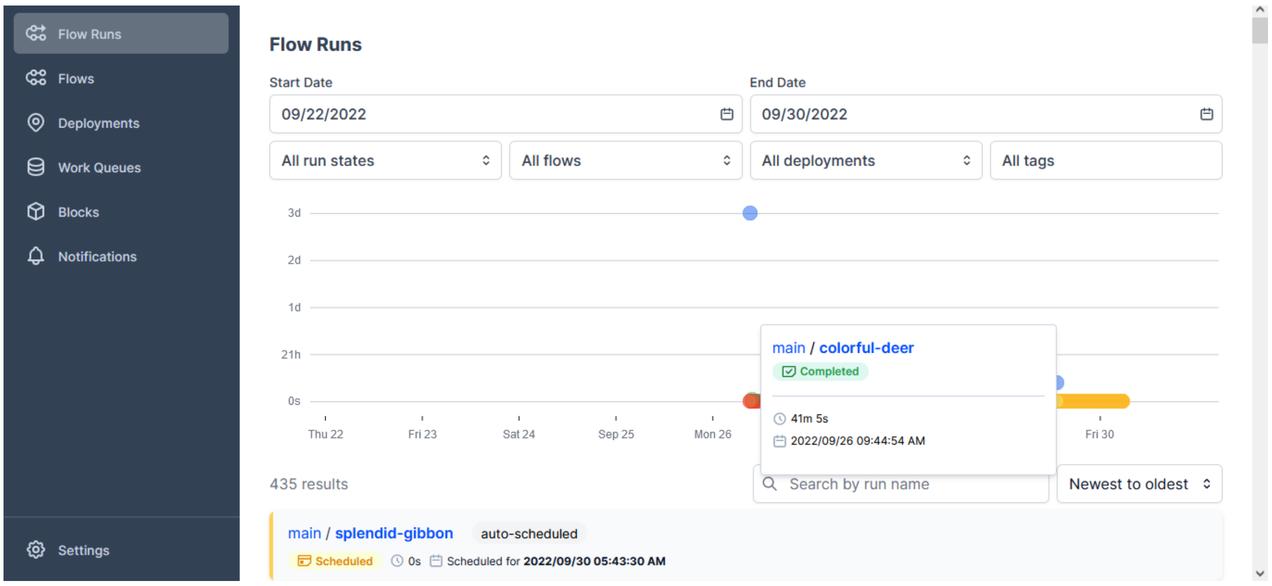
Up to this point, we've demonstrated running Prefect flows and tasks in a local environment using the ephemeral Prefect Orion API. As you've seen, it's possible to run flexible, sophisticated workflows in this way without any further configuration.

Many users find running flows locally is useful for development, testing, and one-off or occasional workflow execution.

Where you begin leveraging the full power of Prefect is when you begin using Prefect for flow coordination and orchestration — building, running, and monitoring your workflows at scale.

Orchestration with Prefect helps you schedule and deploy workflows that run in the environments best suited to their execution. Orchestration helps you prevent and recover from failures, and view and manage the status of workflows, making your workflow:

Prefect Home Page---



Flow Runs

Start Date: 09/22/2022
End Date: 09/30/2022

All run states | All flows | All deployments | All tags

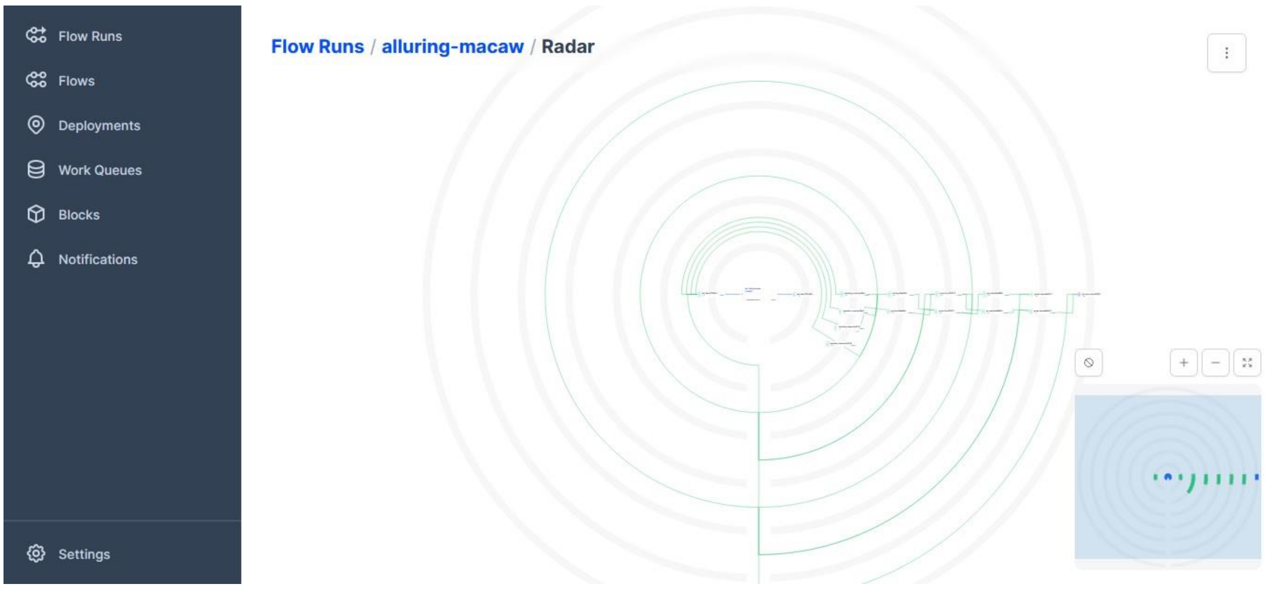
main / colorful-deer
Completed
41m 5s
2022/09/26 09:44:54 AM

435 results

Search by run name
Newest to oldest

main / splendid-gibbon auto-scheduled
Scheduled 0s Scheduled for **2022/09/30 05:43:30 AM**

Workflow Pipeline—



Flow Runs / alluring-macaw / Radar