Q1. Install MLflow

To get started with MLflow you'll need to install the MLflow Python package.
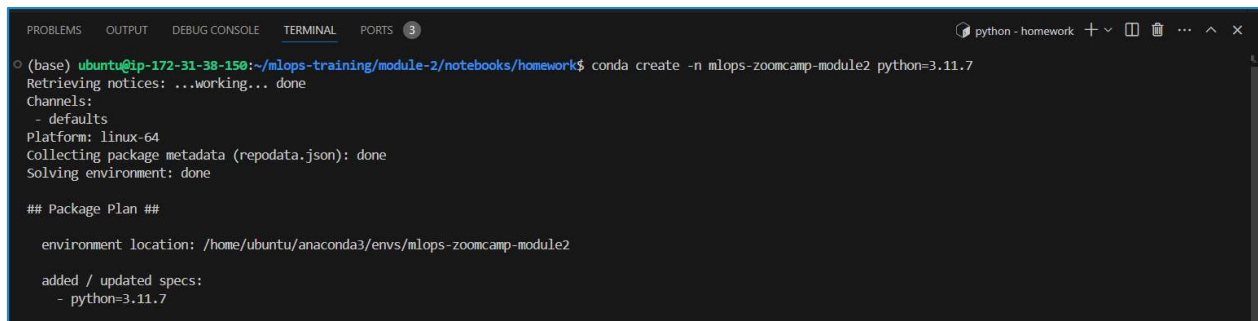
For this we recommend creating a separate Python environment, for example, you can use <u>conda environments</u>, and then install the package there with `pip` or `conda`.

Once you installed the package, run the command `mlflow --version` and check the output.

What's the version that you have?
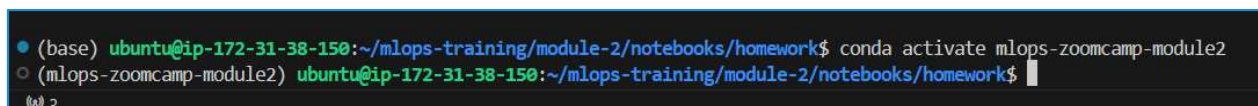
Answer

- **2.14.1**

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS 3                                           python - homework  + ∨  ⬚  🗑  ⋯  ∧  ✕
○ (base) ubuntu@ip-172-31-38-150:~/mlops-training/module-2/notebooks/homework$ conda create -n mlops-zoomcamp-module2 python=3.11.7
Retrieving notices: ...working... done
Channels:
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/ubuntu/anaconda3/envs/mlops-zoomcamp-module2

  added / updated specs:
    - python=3.11.7
```

```
● (base) ubuntu@ip-172-31-38-150:~/mlops-training/module-2/notebooks/homework$ conda activate mlops-zoomcamp-module2
○ (mlops-zoomcamp-module2) ubuntu@ip-172-31-38-150:~/mlops-training/module-2/notebooks/homework$
 3
```

module-2 > notebooks > homework > ≡ requirements.txt

```
 1    mlflow
 2    jupyter
 3    scikit-learn
 4    pandas
 5    seaborn
 6    hyperopt
 7    xgboost
 8    fastparquet
 9    boto3
10
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   **TERMINAL**   PORTS **3**   py

```
○ (mlops-zoomcamp-module2) ubuntu@ip-172-31-38-150:~/mlops-training/module-2/notebooks/homework$ pip install -r requirements.txt
Collecting mlflow (from -r requirements.txt (line 1))
  Using cached mlflow-2.14.1-py3-none-any.whl.metadata (29 kB)
Collecting jupyter (from -r requirements.txt (line 2))
  Using cached jupyter-1.0.0-py2.py3-none-any.whl.metadata (995 bytes)
Collecting scikit-learn (from -r requirements.txt (line 3))
  Using cached scikit_learn-1.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (11 kB)
Collecting pandas (from -r requirements.txt (line 4))
  Using cached pandas-2.2.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (19 kB)
Collecting seaborn (from -r requirements.txt (line 5))
  Using cached seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Collecting hyperopt (from -r requirements.txt (line 6))
  Using cached hyperopt-0.2.7-py2.py3-none-any.whl.metadata (1.7 kB)
Collecting xgboost (from -r requirements.txt (line 7))
  Using cached xgboost-2.1.0-py3-none-manylinux_2_28_x86_64.whl.metadata (2.1 kB)
Collecting fastparquet (from -r requirements.txt (line 8))
  Using cached fastparquet-2024.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.1 kB)
Collecting boto3 (from -r requirements.txt (line 9))
  Downloading boto3-1.34.131-py3-none-any.whl.metadata (6.6 kB)
Collecting Flask<4 (from mlflow->-r requirements.txt (line 1))
  Using cached flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
Collecting alembic!=1.10.0,<2 (from mlflow->-r requirements.txt (line 1))
  Using cached alembic-1.13.1-py3-none-any.whl.metadata (7.4 kB)
Collecting cachetools<6,>=5.0.0 (from mlflow->-r requirements.txt (line 1))
  Using cached cachetools-5.3.3-py3-none-any.whl.metadata (5.3 kB)
Collecting click<9,>=7.0 (from mlflow->-r requirements.txt (line 1))
  Using cached click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
```

127.0.0.1:8888/notebooks/homework%2Fhomework_module2.ipynb

jupyter homework_module2 Last Checkpoint: 55 minutes ago   Trusted

File   Edit   View   Run   Kernel   Settings   Help

▶   ■   C   ▶▶   Code   ⌄   ⌄ Open in...   ⚙   Python 3 (ipykernel) ○

Q1. Install MLflow To get started with MLflow you'll need to install the MLflow Python package.

For this we recommend creating a separate Python environment, for example, you can use conda environments, and then install the package there with pip or conda.

Once you installed the package, run the command mlflow --version and check the output.

What's the version that you have?

```
[4]: !mlflow --version

     mlflow, version 2.14.1
```

Q2. Download and preprocess the data

We'll use the Green Taxi Trip Records dataset to predict the duration of each trip.

Download the data for January, February and March 2023 in parquet format from <u>here</u>.

Use the script `preprocess_data.py` located in the folder <u>homework</u> to preprocess the data.

The script will:

- load the data from the folder `<TAXI_DATA_FOLDER>` (the folder where you have downloaded the data),
- fit a `DictVectorizer` on the training set (January 2023 data),
- save the preprocessed datasets and the `DictVectorizer` to disk.

Your task is to download the datasets and then execute this command:

```
python preprocess_data.py --raw_data_path <TAXI_DATA_FOLDER> --dest_path ./output
```

Tip: go to `02-experiment-tracking/homework/` folder before executing the command and change the value of `<TAXI_DATA_FOLDER>` to the location where you saved the data.

How many files were saved to `OUTPUT_FOLDER`?

- 1
- 3
- 4
- 7

Answer

- **4**

Downloading data

```
[5]:  !wget https://d37ci6vzurychx.cloudfront.net/trip-data/green_tripdata_2023-01.parquet  -O ../data/green_tripdata_2023-01.parquet
      !wget https://d37ci6vzurychx.cloudfront.net/trip-data/green_tripdata_2023-02.parquet -O ../data/green_tripdata_2023-02.parquet
      !wget https://d37ci6vzurychx.cloudfront.net/trip-data/green_tripdata_2023-03.parquet -O ../data/green_tripdata_2023-03.parquet

      --2024-06-22 18:23:37--  https://d37ci6vzurychx.cloudfront.net/trip-data/green_tripdata_2023-01.parquet
      Resolving d37ci6vzurychx.cloudfront.net (d37ci6vzurychx.cloudfront.net)... 13.32.192.190, 13.32.192.2, 13.32.192.116, ...
      Connecting to d37ci6vzurychx.cloudfront.net (d37ci6vzurychx.cloudfront.net)|13.32.192.190|:443... connected.
      HTTP request sent, awaiting response... 200 OK
      Length: 1427002 (1.4M) [binary/octet-stream]
      Saving to: '../data/green_tripdata_2023-01.parquet'

      ../data/green_tripd 100%[===================>]   1.36M  --.-KB/s    in 0.03s

      2024-06-22 18:23:38 (41.7 MB/s) - '../data/green_tripdata_2023-01.parquet' saved [1427002/1427002]

      --2024-06-22 18:23:38--  https://d37ci6vzurychx.cloudfront.net/trip-data/green_tripdata_2023-02.parquet
      Resolving d37ci6vzurychx.cloudfront.net (d37ci6vzurychx.cloudfront.net)... 13.32.192.2, 13.32.192.116, 13.32.192.124, ...
      Connecting to d37ci6vzurychx.cloudfront.net (d37ci6vzurychx.cloudfront.net)|13.32.192.2|:443... connected.
      HTTP request sent, awaiting response... 200 OK
      Length: 1533740 (1.5M) [binary/octet-stream]
      Saving to: '../data/green_tripdata_2023-02.parquet'

      ../data/green_tripd 100%[===================>]   1.46M  --.-KB/s    in 0.02s

      2024-06-22 18:23:38 (78.1 MB/s) - '../data/green_tripdata_2023-02.parquet' saved [1533740/1533740]

      --2024-06-22 18:23:39--  https://d37ci6vzurychx.cloudfront.net/trip-data/green_tripdata_2023-03.parquet
      Resolving d37ci6vzurychx.cloudfront.net (d37ci6vzurychx.cloudfront.net)... 13.32.192.2, 13.32.192.116, 13.32.192.124, ...
      Connecting to d37ci6vzurychx.cloudfront.net (d37ci6vzurychx.cloudfront.net)|13.32.192.2|:443... connected.
      HTTP request sent, awaiting response... 200 OK
      Length: 1730999 (1.7M) [binary/octet-stream]
      Saving to: '../data/green_tripdata_2023-03.parquet'

      ../data/green_tripd 100%[===================>]   1.65M  --.-KB/s    in 0.02s

      2024-06-22 18:23:39 (86.4 MB/s) - '../data/green_tripdata_2023-03.parquet' saved [1730999/1730999]
```

Running preprocess file and printing the count of files in output folder

```
[11]:  !python preprocess_data.py --raw_data_path ../data --dest_path ./output

[13]:  import os
       for path in os.scandir('./output'):
           if path.is_file():
               print(path.name)

       val.pkl
       dv.pkl
       train.pkl
       test.pkl

[ ]:
```

Q3. Train a model with autolog

We will train a `RandomForestRegressor` (from Scikit-Learn) on the taxi dataset.

We have prepared the training script `train.py` for this exercise, which can be also found in the folder `homework`.

The script will:

- load the datasets produced by the previous step,
- train the model on the training set,
- calculate the RMSE score on the validation set.

Your task is to modify the script to enable **autologging** with MLflow, execute the script and then launch the MLflow UI to check that the experiment run was properly tracked.

Tip 1: don't forget to wrap the training code with a `with mlflow.start_run():` statement as we showed in the videos.

Tip 2: don't modify the hyperparameters of the model to make sure that the training will finish quickly.

What is the value of the `min_samples_split` parameter:

- 2
- 4
- 8
- 10

Answer

- 2

Jupyter **train.py** Last Checkpoint: 22 hours ago

File   Edit   View   Settings   Help

```python
1  import os
2  import pickle
3  import click
4  import mlflow
5
6  from sklearn.ensemble import RandomForestRegressor
7  from sklearn.metrics import mean_squared_error
8
9  # Adding tracking uri and creating experiment if does not exist
10 mlflow.set_tracking_uri("sqlite:///mlflow.db")
11 mlflow.set_experiment("homework_module2")
12
13 def load_pickle(filename: str):
14     with open(filename, "rb") as f_in:
15         return pickle.load(f_in)
16
```

```python
21     default="./output",
22     help="Location where the processed NYC taxi trip data was saved"
23 )
24 def run_train(data_path: str):
25
26     # Adding autolog
27     mlflow.sklearn.autolog()
28     X_train, y_train = load_pickle(os.path.join(data_path, "train.pkl"))
29     X_val, y_val = load_pickle(os.path.join(data_path, "val.pkl"))
30
31     # wrapping the training code with a  with mlflow.start_run()
32     with mlflow.start_run():
33         rf = RandomForestRegressor(max_depth=10, random_state=0)
34         rf.fit(X_train, y_train)
35         y_pred = rf.predict(X_val)
36
37         rmse = mean_squared_error(y_val, y_pred, squared=False)
38
39
40 if __name__ == '__main__':
41     run_train()
```

homework_module2 ›

honorable-shrike-648

Overview   Model metrics   System metrics   Artifacts

Parameters (18)

Metrics

Parameter	Value
bootstrap	True
ccp_alpha	0.0
criterion	squared_error
max_depth	10
max_features	1.0
max_leaf_nodes	None
max_samples	None
min_impurity_decrease	0.0
min_samples_leaf	1
min_samples_split	2
min_weight_fraction_leaf	0.0
monotonic_cst	None
n_estimators	100
n_jobs	None
oob_score	False
random_state	0
verbose	0
warm_start	False

Q4. Launch the tracking server locally

Now we want to manage the entire lifecycle of our ML model. In this step, you'll need to launch a tracking server. This way we will also have access to the model registry.

Your task is to:

- launch the tracking server on your local machine,
- select a SQLite db for the backend store and a folder called `artifacts` for the artifacts store.
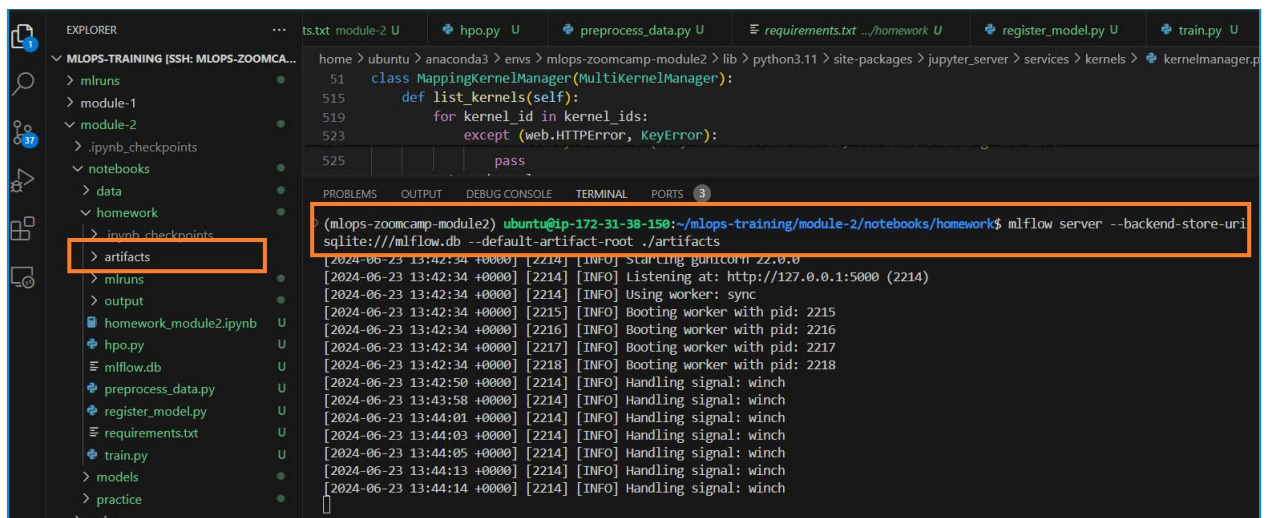
You should keep the tracking server running to work on the next two exercises that use the server.

In addition to `backend-store-uri,` what else do you need to pass to properly configure the server?

- `default-artifact-root`
- `serve-artifacts`
- `artifacts-only`
- `artifacts-destination`

Answer

default-artifact-root

Q5. Tune model hyperparameters

Now let's try to reduce the validation error by tuning the hyperparameters of the `RandomForestRegressor` using `hyperopt`. We have prepared the script `hpo.py` for this exercise.

Your task is to modify the script `hpo.py` and make sure that the validation RMSE is logged to the tracking server for each run of the hyperparameter optimization (you will need to add a few lines of code to the `objective` function) and run the script without passing any parameters.

After that, open UI and explore the runs from the experiment called `random-forest-hyperopt` to answer the question below.

Note: Don't use autologging for this exercise.

The idea is to just log the information that you need to answer the question below, including:

- the list of hyperparameters that are passed to the `objective` function during the optimization,
- the RMSE obtained on the validation set (February 2023 data).

What's the best validation RMSE that you got?

- 4.817
- 5.335
- 5.818
- 6.336

Answer

- 5.335

File   Edit   View   Settings   Help

```python
36
37    def objective(params):
38
39        # wrapping the training code with a  with mlflow.start_run()
40        with mlflow.start_run():
41
42            #logging params
43            mlflow.log_params(params)
44            rf = RandomForestRegressor(**params)
45            rf.fit(X_train, y_train)
46            y_pred = rf.predict(X_val)
47            rmse = mean_squared_error(y_val, y_pred, squared=False)
48            #logging metric rmse
49            mlflow.log_metric("rmse", rmse)
50
51        return {'loss': rmse, 'status': STATUS_OK}
52
53    search_space = {
54        'max_depth': scope.int(hp.quniform('max_depth', 1, 20, 1)),
55        'n_estimators': scope.int(hp.quniform('n_estimators', 10, 50, 1)),
56        'min_samples_split': scope.int(hp.quniform('min_samples_split', 2, 10, 1)),
57        'min_samples_leaf': scope.int(hp.quniform('min_samples_leaf', 1, 4, 1)),
58        'random_state': 42
59    }
60
61    rstate = np.random.default_rng(42)  # for reproducible results
62    fmin(
63        fn=objective,
64        space=search_space,
65        algo=tpe.suggest,
66        max_evals=num_trials,
67        trials=Trials(),
68        rstate=rstate
69    )
70
```

• 5.335

```
[1]: !python hpo.py
```

2024/06/23 13:55:33 INFO mlflow.tracking.fluent: Experiment with name 'random-forest-hyperopt' does not exist. Creating a new experiment.
0%| | 0/15 [00:00<?, ?trial/s, best loss=?]/home/ubuntu/anaconda3/envs/mlops-zoomcamp-module2/lib/python3.11/site-pa
ckages/sklearn/metrics/_regression.py:492: FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calculate the root mean s
quared error, use the function'root_mean_squared_error'.
  warnings.warn(

7%|▊ | 1/15 [00:12<02:50, 12.16s/trial, best loss: 5.370086069268862]/home/ubuntu/anaconda3/envs/mlops-zoomcamp-module2/lib/python3.11/site-pa
ckages/sklearn/metrics/_regression.py:492: FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calculate the root mean s
quared error, use the function'root_mean_squared_error'.
  warnings.warn(

13%|█▌ | 2/15 [00:12<01:09, 5.31s/trial, best loss: 5.370086069268862]/home/ubuntu/anaconda3/envs/mlops-zoomcamp-module2/lib/python3.11/site-pa
ckages/sklearn/metrics/_regression.py:492: FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calculate the root mean s
quared error, use the function'root_mean_squared_error'.
  warnings.warn(

20%|██▋ | 3/15 [00:13<00:38, 3.24s/trial, best loss: 5.370086069268862]/home/ubuntu/anaconda3/envs/mlops-zoomcamp-module2/lib/python3.11/site-pa
ckages/sklearn/metrics/_regression.py:492: FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calculate the root mean s
quared error, use the function'root_mean_squared_error'.
  warnings.warn(

27%|███▋ | 4/15 [00:21<00:57, 5.21s/trial, best loss: 5.357490752366866]/home/ubuntu/anaconda3/envs/mlops-zoomcamp-module2/lib/python3.11/site-pa
ckages/sklearn/metrics/_regression.py:492: FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calculate the root mean s
quared error, use the function'root_mean_squared_error'.
  warnings.warn(

33%|████▋ | 5/15 [00:25<00:46, 4.68s/trial, best loss: 5.357490752366866]/home/ubuntu/anaconda3/envs/mlops-zoomcamp-module2/lib/python3.11/site-pa
ckages/sklearn/metrics/_regression.py:492: FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calculate the root mean s
quared error, use the function'root_mean_squared_error'.
  warnings.warn(

Q6. Promote the best model to the model registry

The results from the hyperparameter optimization are quite good. So, we can assume that we are ready to test some of these models in production. In this exercise, you'll promote the best model to the model registry. We have prepared a script called `register_model.py`, which will check the results from the previous step and select the top 5 runs. After that, it will calculate the RMSE of those models on the test set (March 2023 data) and save the results to a new experiment called `random-forest-best-models`.

Your task is to update the script `register_model.py` so that it selects the model with the lowest RMSE on the test set and registers it to the model registry.

Tip 1: you can use the method `search_runs` from the `MlflowClient` to get the model with the lowest RMSE,

Tip 2: to register the model you can use the method `mlflow.register_model` and you will need to pass the right `model_uri` in the f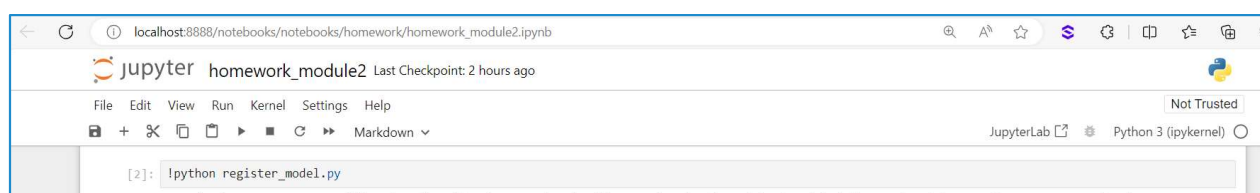orm of a string that looks like this: `"runs:/<RUN_ID>/model"`, and the name of the model (make sure to choose a good one!).
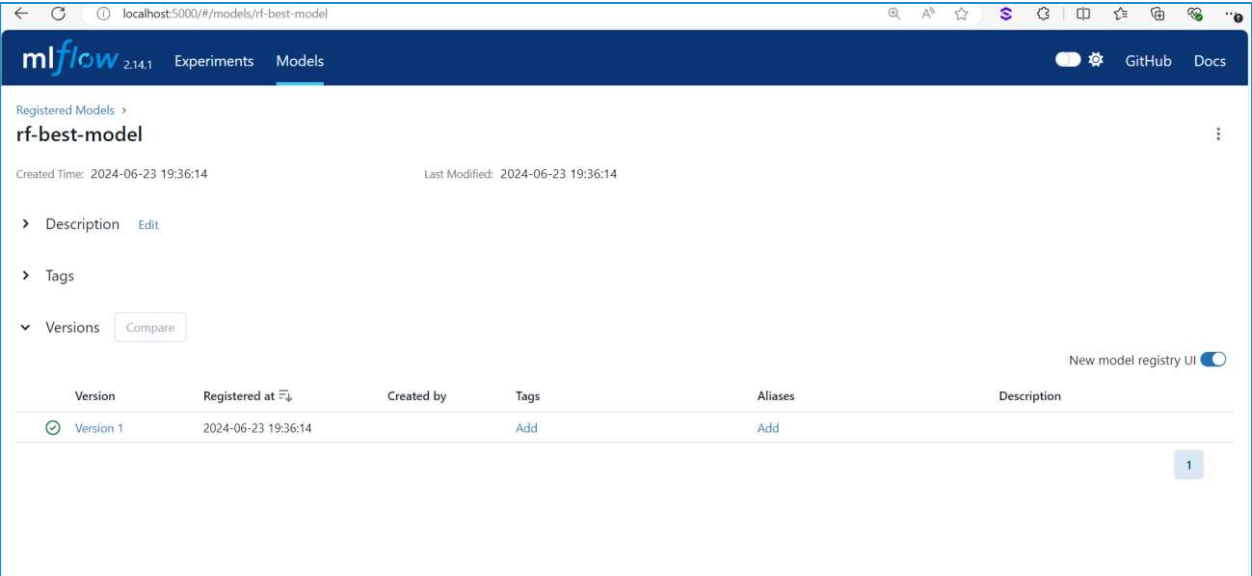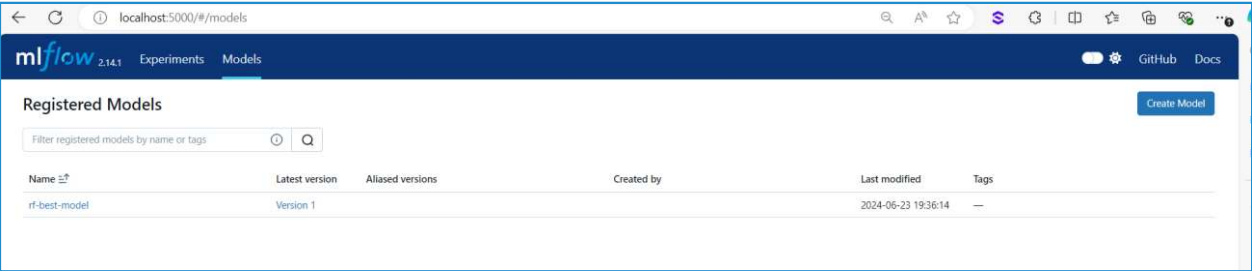
What is the test RMSE of the best model?

- 5.060
- 5.567
- 6.061
- 6.568

Answer

- 5.567

ml*flow* 2.14.1   Experiments   Models   GitHub   Docs

Experiments   random-forest-best-models ⓘ Provide Feedback ↗ Add Description   Share

Search Experiments

☐ Default
☐ homework_module2
☐ random-forest-hyperopt
☑ random-forest-best-mod...

metrics.rmse < 1 and params.model = "tree"   ⓘ   Time created ∨   State: Active ∨   Datasets ∨   ⋮ ↻ + New run

⇅ Sort: test_rmse ∨   ▦ Columns ∨   ☐ Expand rows   ▦ Group by ∨

Table   Chart   Evaluation [Experimental]   Traces [Experimental]

		Run Name	Created	Duration	Source	Models	Metrics test_rmse ⇅
☐		● resilient-grouse-31	⊘ 6 minutes ago	dataset (5a2...	15.8s	⌂ register_...	rf-best-model v1 +1	5.567408012...
☐		● auspicious-elk-893	⊘ 6 minutes ago	dataset (5a2...	22.1s	⌂ register_...	sklearn	5.585312218...
☐		● carefree-ox-493	⊘ 5 minutes ago	dataset (5a2...	15.8s	⌂ register_...	sklearn	5.589460017...
☐		● orderly-deer-948	⊘ 6 minutes ago	dataset (5a2...	15.4s	⌂ register_...	sklearn	5.592132279...
☐		● fortunate-doe-578	⊘ 5 minutes ago	dataset (5a2...	14.5s	⌂ register_...	sklearn	5.594160565...

---

random-forest-best-models ›

resilient-grouse-31   ⋮   Model registe

Overview   Model metrics   System metrics   Artifacts

| Logged models | ✕ sklearn |
| Registered models | ✕ rf-best-model v1 |

Parameters (18)

Search parameters

Parameter	Value
bootstrap	True
ccp_alpha	0.0
criterion	squared_error
max_depth	19
max_features	1.0
max_leaf_nodes	None
max_samples	None
min_impurity_decrease	0.0
min_samples_leaf	2
min_samples_split	2
min_weight_fraction_leaf	0.0
monotonic_cst	None
n_estimators	11
n_jobs	None
oob_score	False

Metrics (7)

Search metrics

Metric	Value
training_mean_squared_error	24.785660360279664
training_mean_absolute_error	3.23365652419934
training_r2_score	0.6956122194293166
training_root_mean_squared_error	4.978519896543516
training_score	0.6956122194293166
val_rmse	5.335419588556921
test_rmse	5.567408012462019