NAME: MOHAMMAD HUSSAM(2033.KHI.DEG.020)

PAIRING WITH : MAVIA ALAM KHAN (2303.KHI.DEG.017)

&

AQSA TAUHEED(2303.KHI.DEG.011)

ASSIGNMENT 3.4 (a + b)

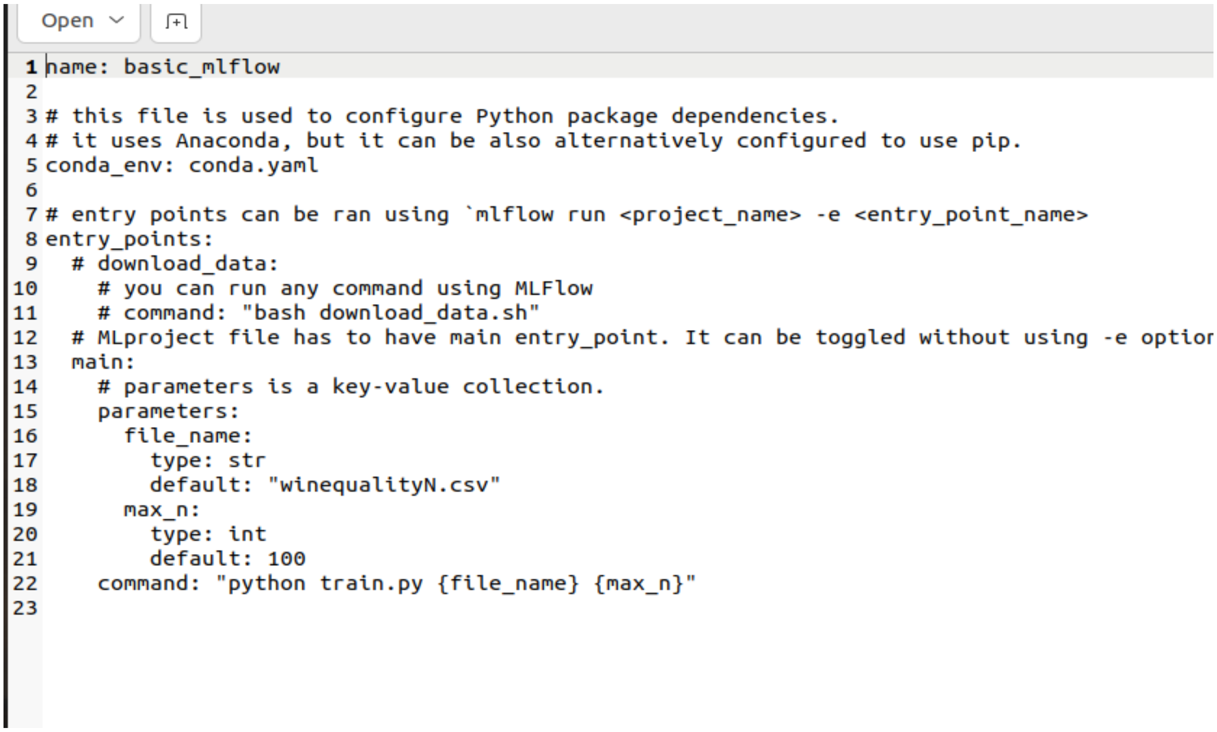
Write a component that will log metadata of your Classification model that you trained on the day dedicated to Supervised Learning. Remember to include all metadata that are important to track for this problem.

Run your Classification model that you trained on the day dedicated to Supervised Learning in MLFlow.

SOLUTION:

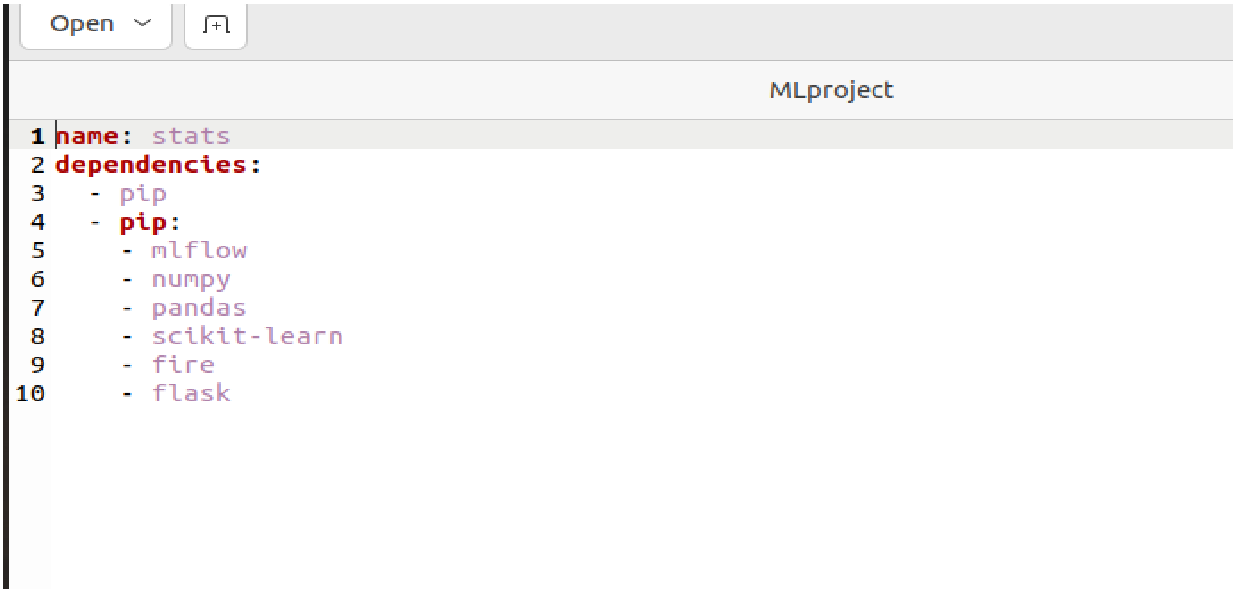
STEP:1:

First we downloaded the winequalityN.csv data set from kaggle and then mlproject file is a configuration file used by MLFlow to define how to run a project.so we did configiration in it.



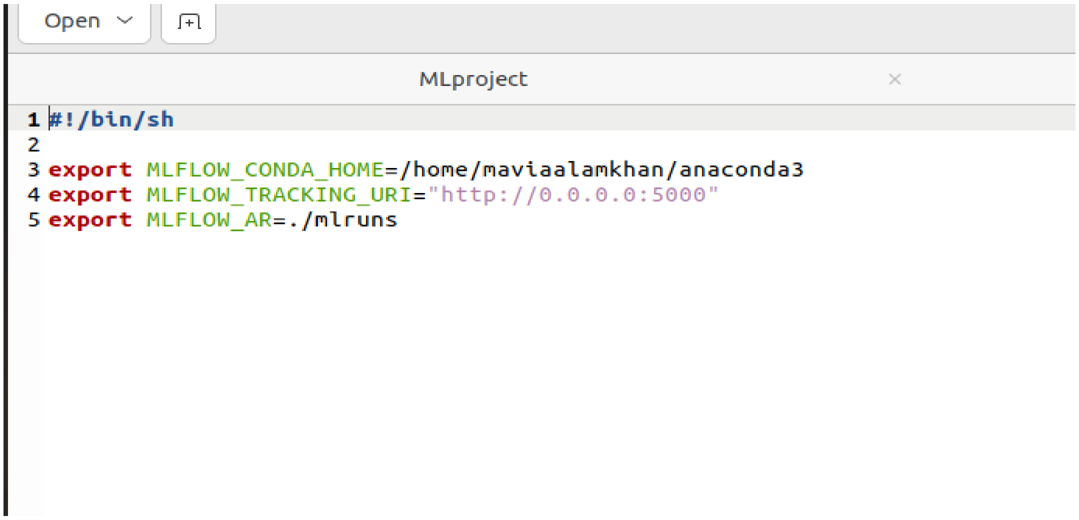
```
name: basic_mlflow

# this file is used to configure Python package dependencies.
# it uses Anaconda, but it can be also alternatively configured to use pip.
conda_env: conda.yaml

# entry points can be ran using `mlflow run <project_name> -e <entry_point_name>
entry_points:
  # download_data:
    # you can run any command using MLFlow
    # command: "bash download_data.sh"
  # MLproject file has to have main entry_point. It can be toggled without using -e optior
  main:
    # parameters is a key-value collection.
    parameters:
      file_name:
        type: str
        default: "winequalityN.csv"
      max_n:
        type: int
        default: 100
    command: "python train.py {file_name} {max_n}"
```

After that the conda package manager to specify the dependencies and configuration of a software environment.



```
name: stats
dependencies:
  - pip
  - pip:
    - mlflow
    - numpy
    - pandas
    - scikit-learn
    - fire
    - flask
```

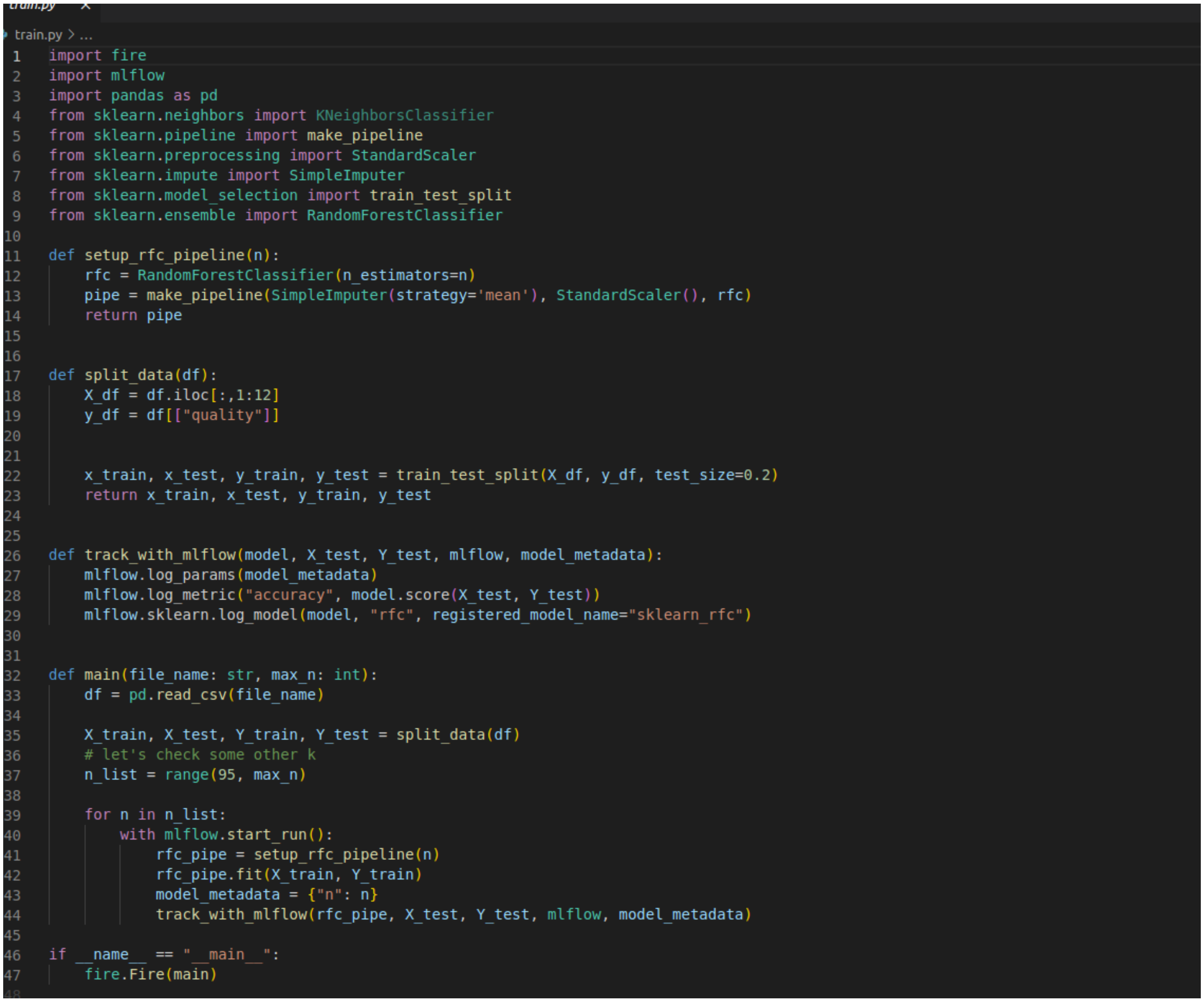
mlflow_env_vars.sh is a shell script that can be used to set environment variables that are used by MLFlow.



```
#!/bin/sh

export MLFLOW_CONDA_HOME=/home/maviaalamkhan/anaconda3
export MLFLOW_TRACKING_URI="http://0.0.0.0:5000"
export MLFLOW_AR=./mlruns
```

STEP:2:

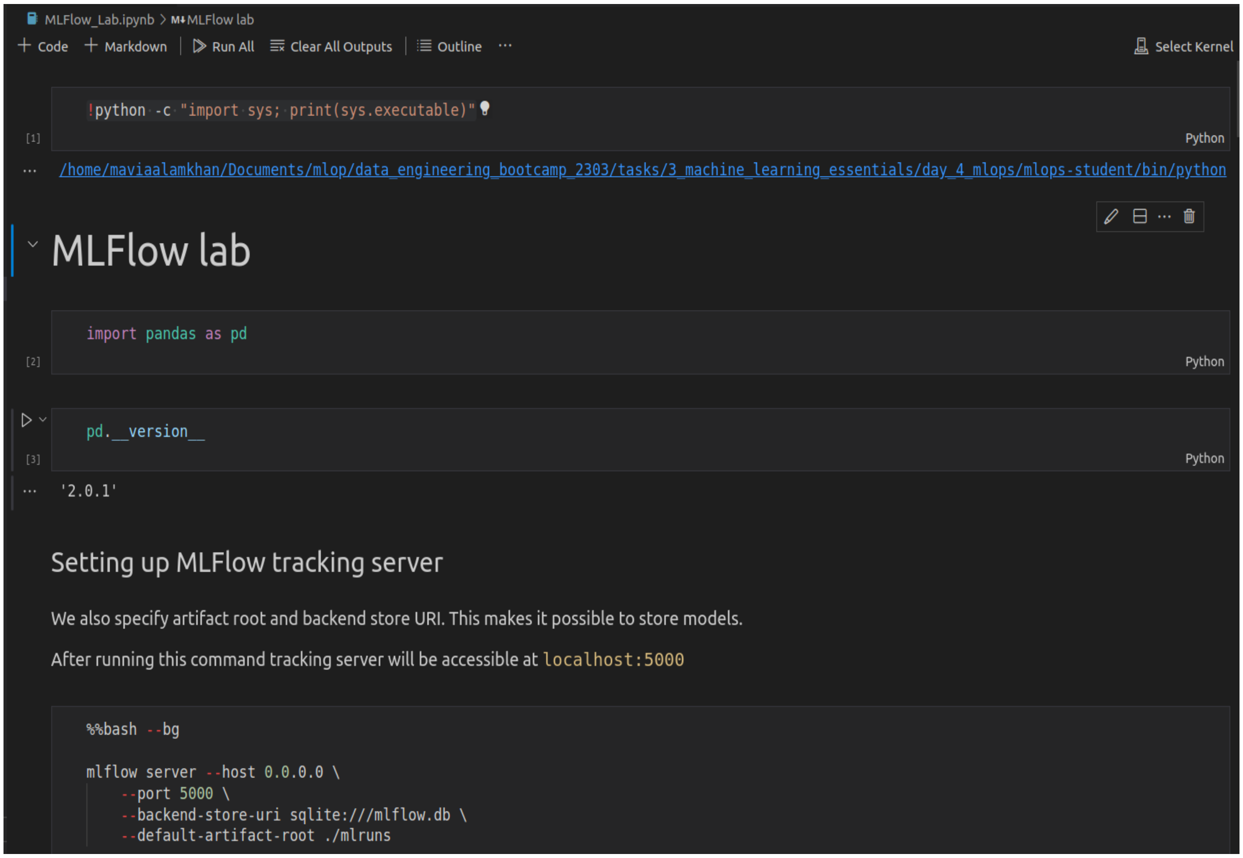
We setup the train.py file. To log this metadata, we can create a Python module that defines a function to train the classification model and log the metadata using MLFlow



```python
import fire
import mlflow
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

def setup_rfc_pipeline(n):
    rfc = RandomForestClassifier(n_estimators=n)
    pipe = make_pipeline(SimpleImputer(strategy='mean'), StandardScaler(), rfc)
    return pipe


def split_data(df):
    X_df = df.iloc[:,1:12]
    y_df = df[["quality"]]


    x_train, x_test, y_train, y_test = train_test_split(X_df, y_df, test_size=0.2)
    return x_train, x_test, y_train, y_test


def track_with_mlflow(model, X_test, Y_test, mlflow, model_metadata):
    mlflow.log_params(model_metadata)
    mlflow.log_metric("accuracy", model.score(X_test, Y_test))
    mlflow.sklearn.log_model(model, "rfc", registered_model_name="sklearn_rfc")


def main(file_name: str, max_n: int):
    df = pd.read_csv(file_name)

    X_train, X_test, Y_train, Y_test = split_data(df)
    # let's check some other k
    n_list = range(95, max_n)

    for n in n_list:
        with mlflow.start_run():
            rfc_pipe = setup_rfc_pipeline(n)
            rfc_pipe.fit(X_train, Y_train)
            model_metadata = {"n": n}
            track_with_mlflow(rfc_pipe, X_test, Y_test, mlflow, model_metadata)

if __name__ == "__main__":
    fire.Fire(main)
```

STEP 3 :

We used the MlFlow_lab.ipnyb file in conjunction with MLFlow to develop, test, and experiment with different models and hyperparameters.



```
!python -c "import sys; print(sys.executable)"
```

```
/home/maviaalamkhan/Documents/mlop/data_engineering_bootcamp_2303/tasks/3_machine_learning_essentials/day_4_mlops/mlops-student/bin/python
```

MLFlow lab

```
import pandas as pd
```

```
pd.__version__
```

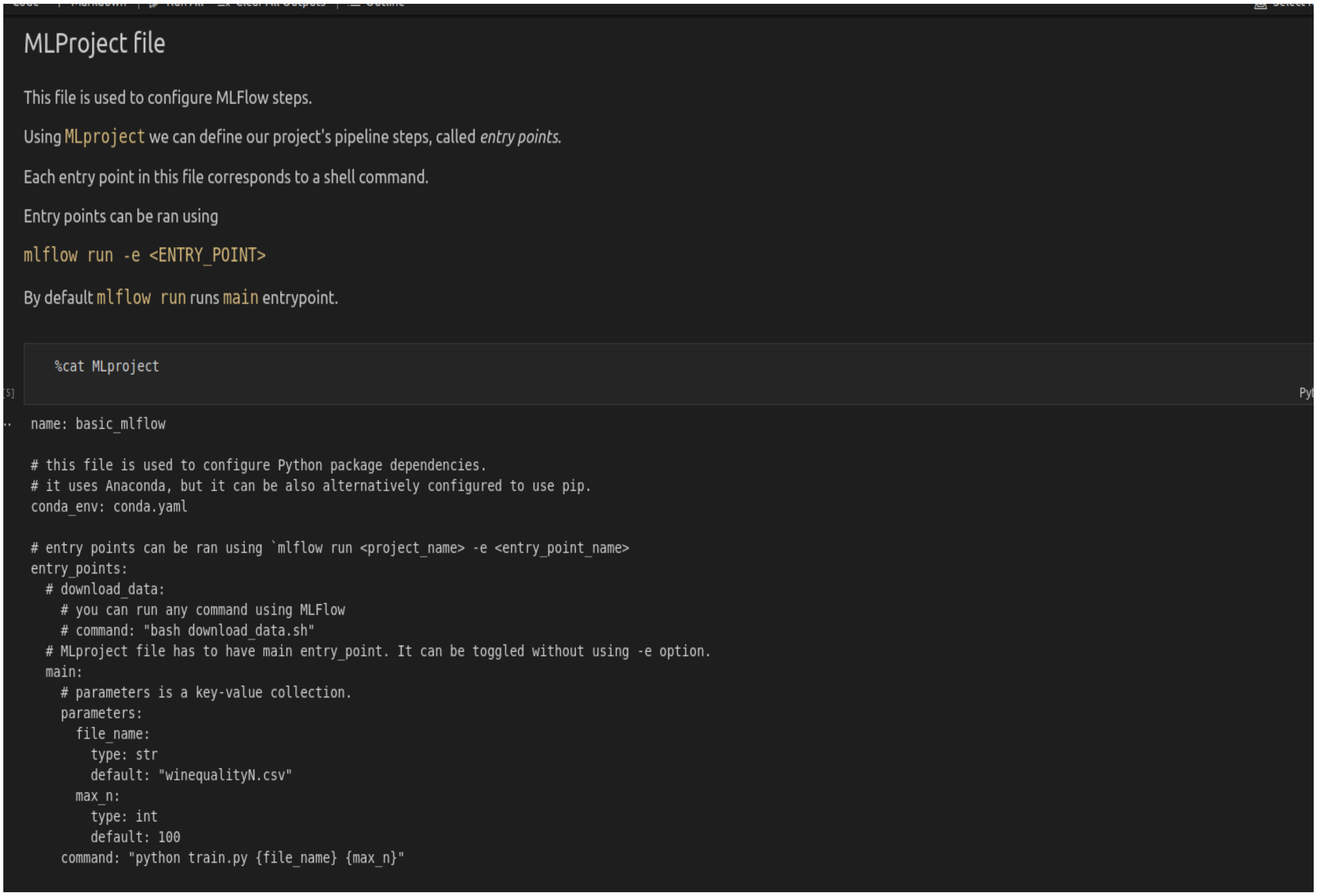
```
'2.0.1'
```

Setting up MLFlow tracking server

We also specify artifact root and backend store URI. This makes it possible to store models.

After running this command tracking server will be accessible at `localhost:5000`

```
%%bash --bg

mlflow server --host 0.0.0.0 \
    --port 5000 \
    --backend-store-uri sqlite:///mlflow.db \
    --default-artifact-root ./mlruns
```



MLProject file

This file is used to configure MLFlow steps.

Using `MLproject` we can define our project's pipeline steps, called *entry points*.

Each entry point in this file corresponds to a shell command.

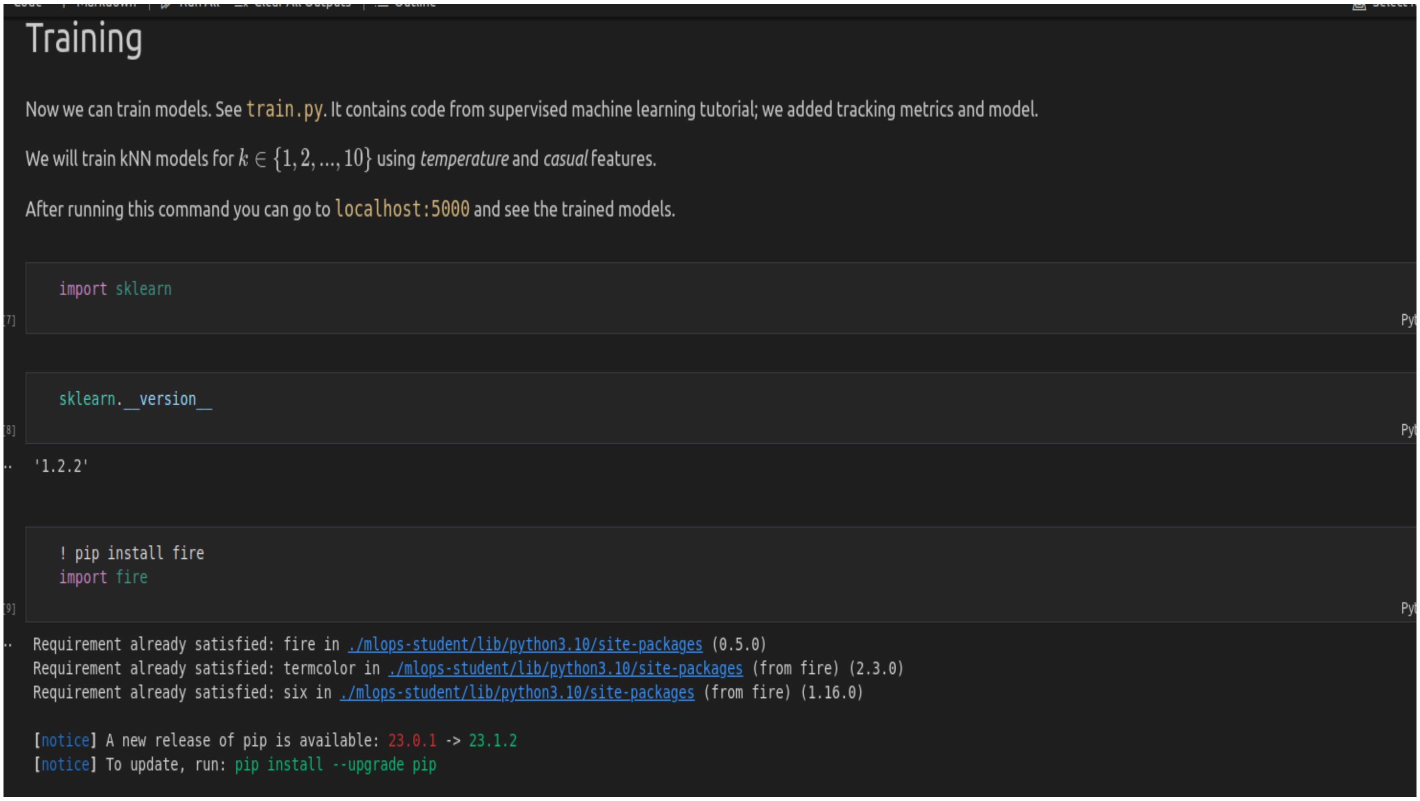
Entry points can be ran using

`mlflow run -e <ENTRY_POINT>`

By default `mlflow run` runs `main` entrypoint.

```
%cat MLproject
```

```
name: basic_mlflow

# this file is used to configure Python package dependencies.
# it uses Anaconda, but it can be also alternatively configured to use pip.
conda_env: conda.yaml

# entry points can be ran using `mlflow run <project_name> -e <entry_point_name>
entry_points:
  # download_data:
    # you can run any command using MLFlow
    # command: "bash download_data.sh"
  # MLproject file has to have main entry_point. It can be toggled without using -e option.
  main:
    # parameters is a key-value collection.
    parameters:
      file_name:
        type: str
        default: "winequalityN.csv"
      max_n:
        type: int
        default: 100
    command: "python train.py {file_name} {max_n}"
```



Training

Now we can train models. See `train.py`. It contains code from supervised machine learning tutorial; we added tracking metrics and model.

We will train kNN models for $k \in \{1, 2, ..., 10\}$ using *temperature* and *casual* features.

After running this command you can go to `localhost:5000` and see the trained models.

```
import sklearn
```

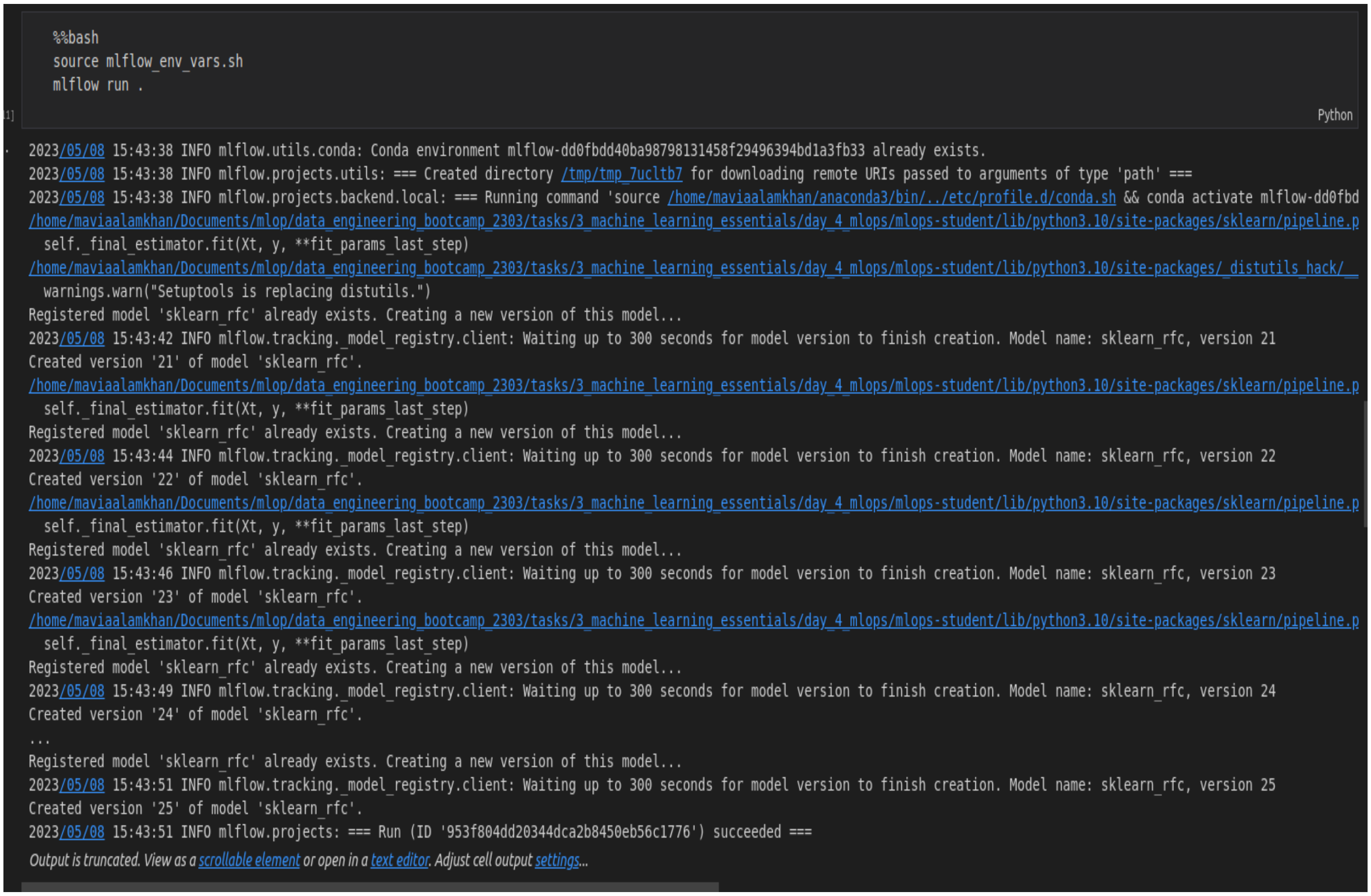
```
sklearn.__version__
```

```
'1.2.2'
```

```
! pip install fire
import fire
```

```
Requirement already satisfied: fire in ./mlops-student/lib/python3.10/site-packages (0.5.0)
Requirement already satisfied: termcolor in ./mlops-student/lib/python3.10/site-packages (from fire) (2.3.0)
Requirement already satisfied: six in ./mlops-student/lib/python3.10/site-packages (from fire) (1.16.0)

[notice] A new release of pip is available: 23.0.1 -> 23.1.2
[notice] To update, run: pip install --upgrade pip
```

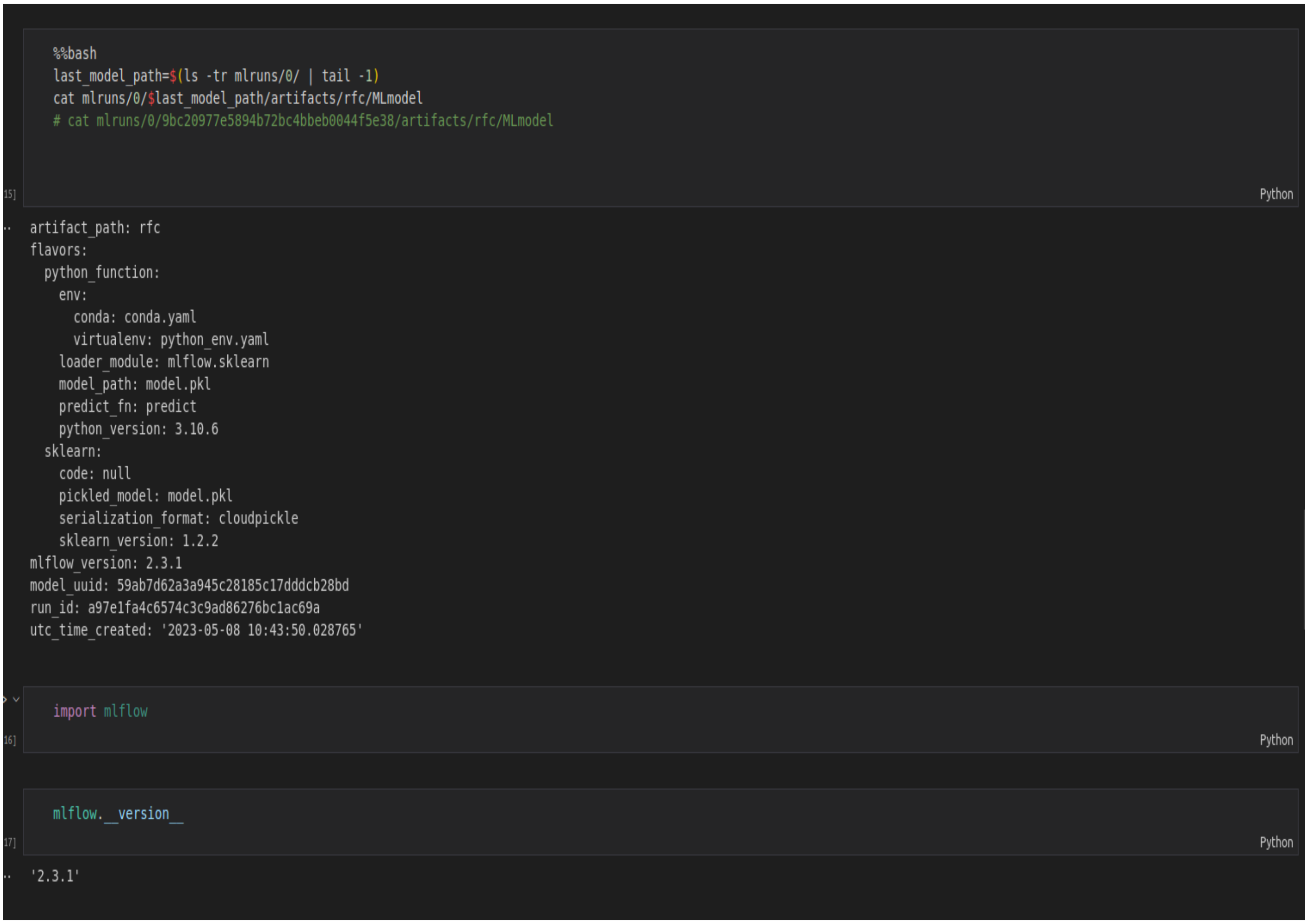
by running these commands, we are able to use MLFlow to manage our machine learning experiments, including tracking the performance of our models and organizing the results of multiple runs.



```
%%bash
source mlflow_env_vars.sh
mlflow run .
```

```
2023/05/08 15:43:38 INFO mlflow.utils.conda: Conda environment mlflow-dd0fbdd40ba98798131458f29496394bd1a3fb33 already exists.
2023/05/08 15:43:38 INFO mlflow.projects.utils: === Created directory /tmp/tmp_7ucltb7 for downloading remote URIs passed to arguments of type 'path' ===
2023/05/08 15:43:38 INFO mlflow.projects.backend.local: === Running command 'source /home/maviaalamkhan/anaconda3/bin/../etc/profile.d/conda.sh && conda activate mlflow-dd0fbd
/home/maviaalamkhan/Documents/mlop/data_engineering_bootcamp_2303/tasks/3_machine_learning_essentials/day_4_mlops/mlops-student/lib/python3.10/site-packages/sklearn/pipeline.p
  self._final_estimator.fit(Xt, y, **fit_params_last_step)
/home/maviaalamkhan/Documents/mlop/data_engineering_bootcamp_2303/tasks/3_machine_learning_essentials/day_4_mlops/mlops-student/lib/python3.10/site-packages/_distutils_hack/__
  warnings.warn("Setuptools is replacing distutils.")
Registered model 'sklearn_rfc' already exists. Creating a new version of this model...
2023/05/08 15:43:42 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation. Model name: sklearn_rfc, version 21
Created version '21' of model 'sklearn_rfc'.
/home/maviaalamkhan/Documents/mlop/data_engineering_bootcamp_2303/tasks/3_machine_learning_essentials/day_4_mlops/mlops-student/lib/python3.10/site-packages/sklearn/pipeline.p
  self._final_estimator.fit(Xt, y, **fit_params_last_step)
Registered model 'sklearn_rfc' already exists. Creating a new version of this model...
2023/05/08 15:43:44 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation. Model name: sklearn_rfc, version 22
Created version '22' of model 'sklearn_rfc'.
/home/maviaalamkhan/Documents/mlop/data_engineering_bootcamp_2303/tasks/3_machine_learning_essentials/day_4_mlops/mlops-student/lib/python3.10/site-packages/sklearn/pipeline.p
  self._final_estimator.fit(Xt, y, **fit_params_last_step)
Registered model 'sklearn_rfc' already exists. Creating a new version of this model...
2023/05/08 15:43:46 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation. Model name: sklearn_rfc, version 23
Created version '23' of model 'sklearn_rfc'.
/home/maviaalamkhan/Documents/mlop/data_engineering_bootcamp_2303/tasks/3_machine_learning_essentials/day_4_mlops/mlops-student/lib/python3.10/site-packages/sklearn/pipeline.p
  self._final_estimator.fit(Xt, y, **fit_params_last_step)
Registered model 'sklearn_rfc' already exists. Creating a new version of this model...
2023/05/08 15:43:49 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation. Model name: sklearn_rfc, version 24
Created version '24' of model 'sklearn_rfc'.
...
Registered model 'sklearn_rfc' already exists. Creating a new version of this model...
2023/05/08 15:43:51 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation. Model name: sklearn_rfc, version 25
Created version '25' of model 'sklearn_rfc'.
2023/05/08 15:43:51 INFO mlflow.projects: === Run (ID '953f804dd20344dca2b8450eb56c1776') succeeded ===
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...



```
%%bash
last_model_path=$(ls -tr mlruns/0/ | tail -1)
cat mlruns/0/$last_model_path/artifacts/rfc/MLmodel
# cat mlruns/0/9bc20977e5894b72bc4bbeb0044f5e38/artifacts/rfc/MLmodel
```

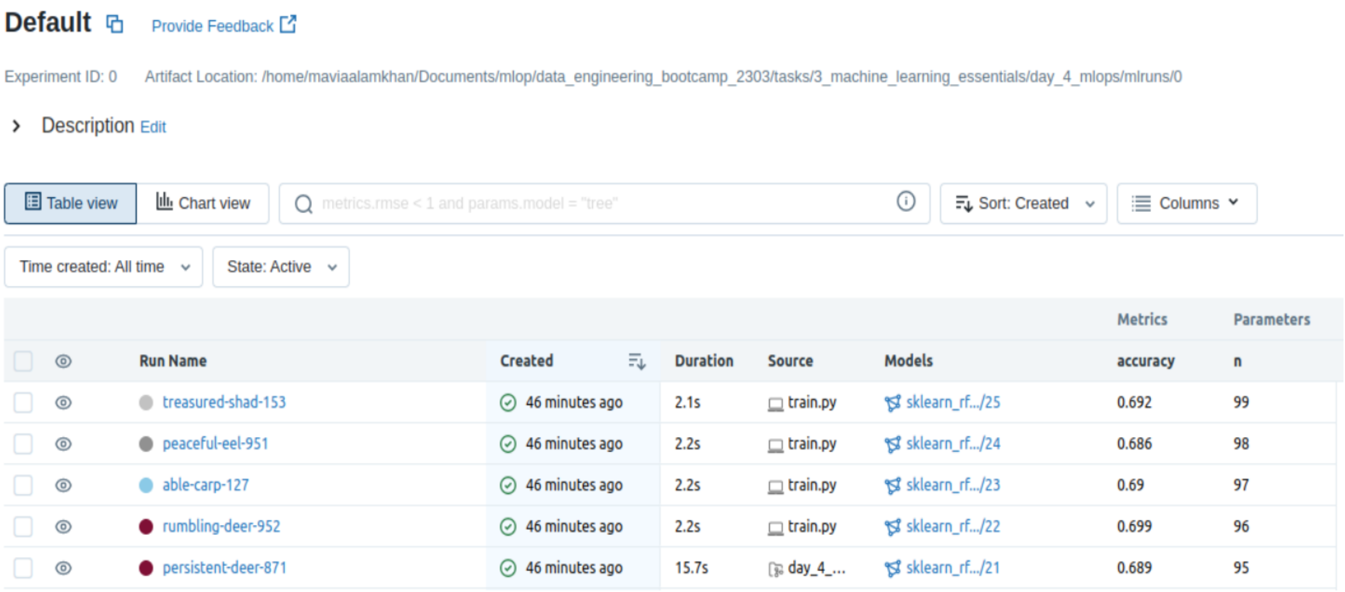
```
artifact_path: rfc
flavors:
  python_function:
    env:
      conda: conda.yaml
      virtualenv: python_env.yaml
    loader_module: mlflow.sklearn
    model_path: model.pkl
    predict_fn: predict
    python_version: 3.10.6
  sklearn:
    code: null
    pickled_model: model.pkl
    serialization_format: cloudpickle
    sklearn_version: 1.2.2
mlflow_version: 2.3.1
model_uuid: 59ab7d62a3a945c28185c17dddcb28bd
run_id: a97e1fa4c6574c3c9ad86276bc1ac69a
utc_time_created: '2023-05-08 10:43:50.028765'
```

```
import mlflow
```

```
mlflow.__version__
```

```
'2.3.1'
```

LOGS:

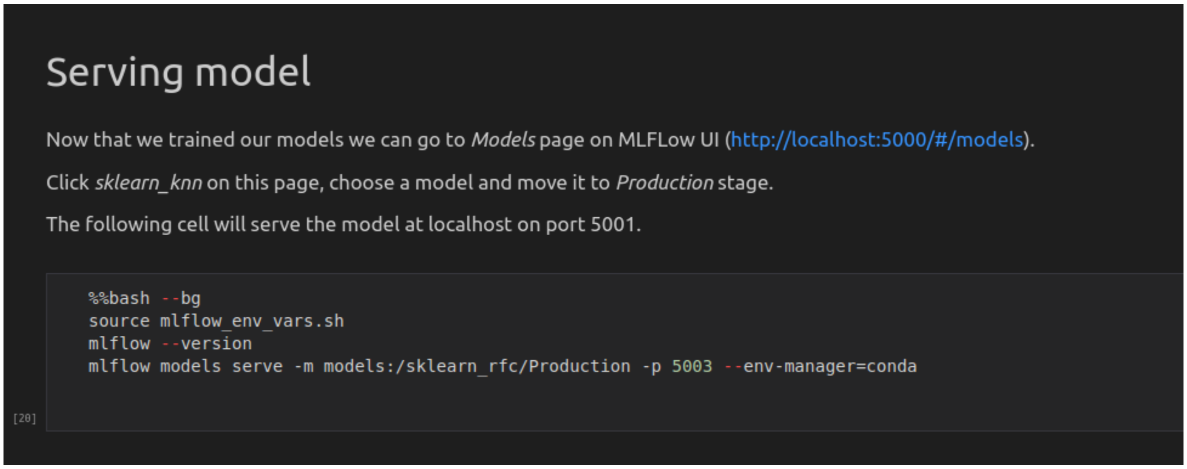


Default

Experiment ID: 0   Artifact Location: /home/maviaalamkhan/Documents/mlop/data_engineering_bootcamp_2303/tasks/3_machine_learning_essentials/day_4_mlops/mlruns/0

Time created: All time   State: Active

							Metrics	Parameters
□		Run Name	Created	Duration	Source	Models	accuracy	n
□		treasured-shad-153	✓ 46 minutes ago	2.1s	train.py	sklearn_rf.../25	0.692	99
□		peaceful-eel-951	✓ 46 minutes ago	2.2s	train.py	sklearn_rf.../24	0.686	98
□		able-carp-127	✓ 46 minutes ago	2.2s	train.py	sklearn_rf.../23	0.69	97
□		rumbling-deer-952	✓ 46 minutes ago	2.2s	train.py	sklearn_rf.../22	0.699	96
□		persistent-deer-871	✓ 46 minutes ago	15.7s	day_4_...	sklearn_rf.../21	0.689	95



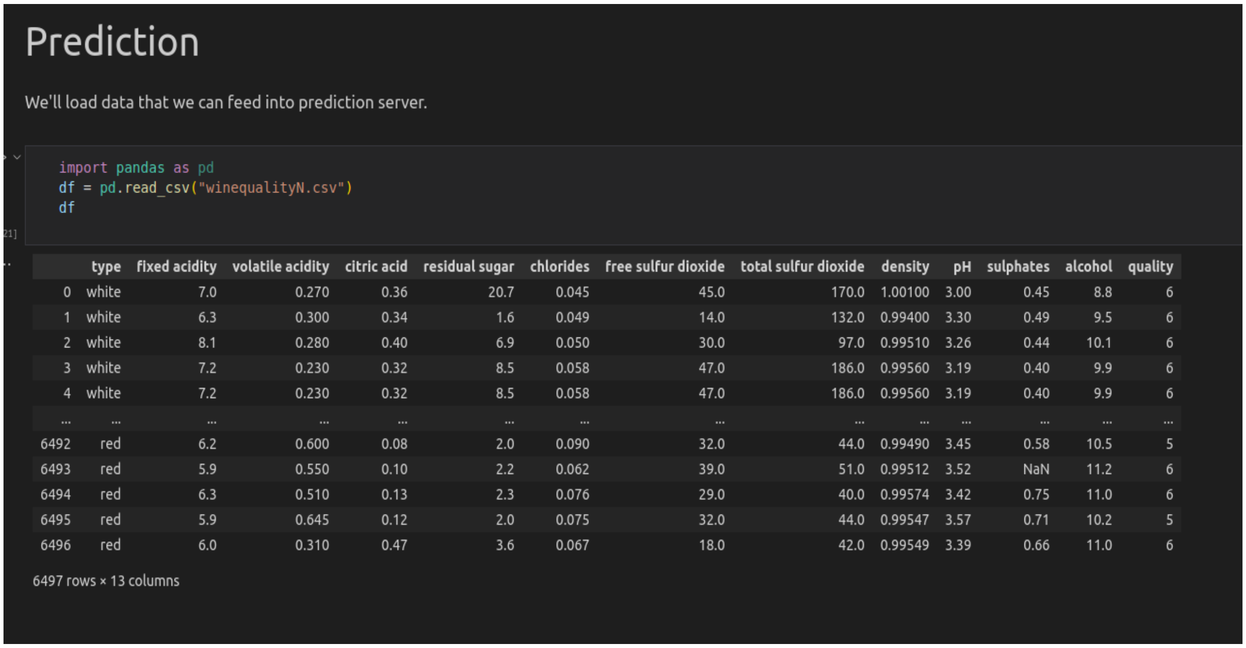
Serving model

Now that we trained our models we can go to *Models* page on MLFLow UI (http://localhost:5000/#/models).

Click *sklearn_knn* on this page, choose a model and move it to *Production* stage.

The following cell will serve the model at localhost on port 5001.

```
%%bash --bg
source mlflow_env_vars.sh
mlflow --version
mlflow models serve -m models:/sklearn_rfc/Production -p 5003 --env-manager=conda
```



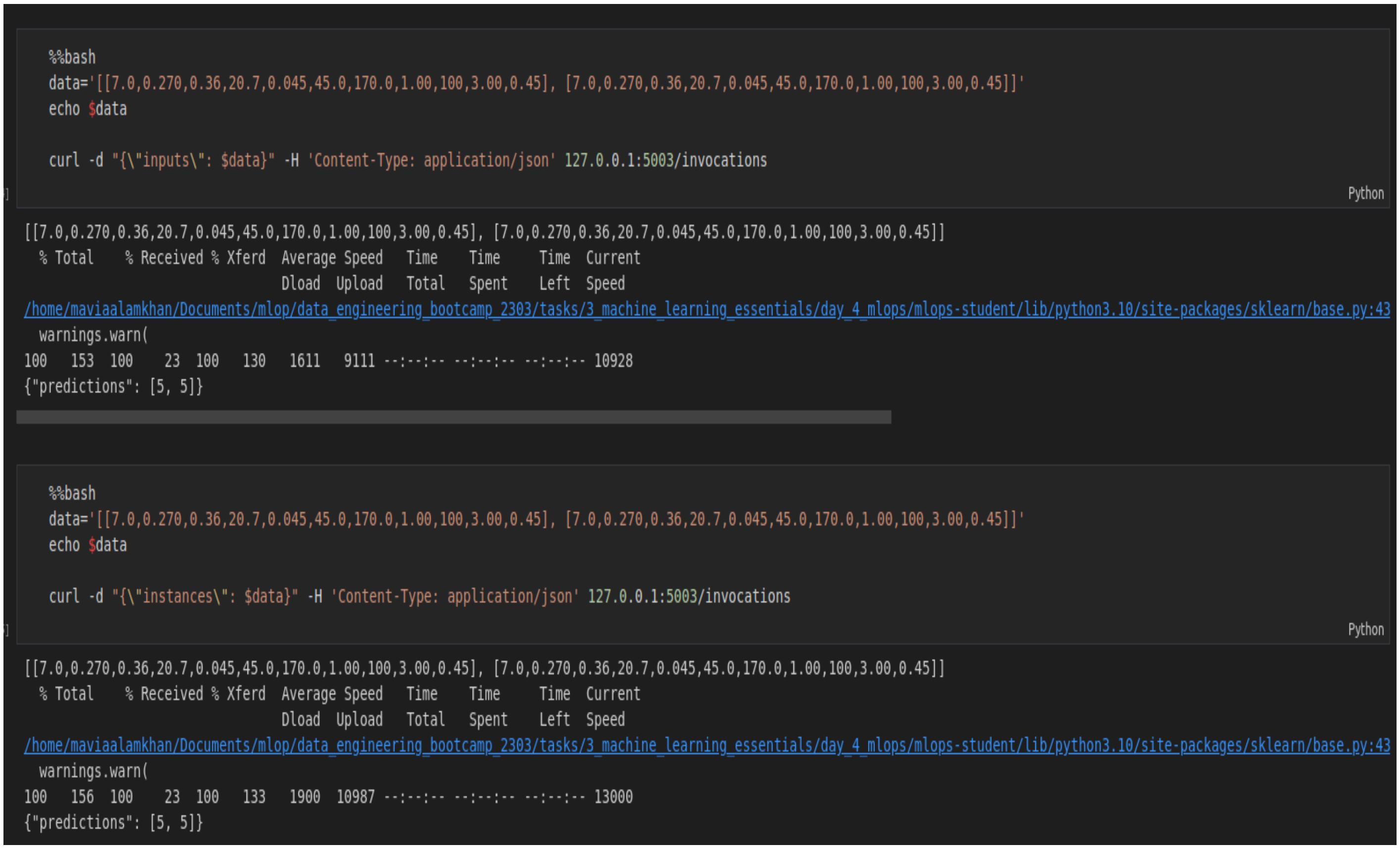
Prediction

We'll load data that we can feed into prediction server.

```
import pandas as pd
df = pd.read_csv("winequalityN.csv")
df
```

	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	white	7.0	0.270	0.36	20.7	0.045	45.0	170.0	1.00100	3.00	0.45	8.8	6
1	white	6.3	0.300	0.34	1.6	0.049	14.0	132.0	0.99400	3.30	0.49	9.5	6
2	white	8.1	0.280	0.40	6.9	0.050	30.0	97.0	0.99510	3.26	0.44	10.1	6
3	white	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	9.9	6
4	white	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	9.9	6
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6492	red	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
6493	red	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	NaN	11.2	6
6494	red	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
6495	red	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
6496	red	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

6497 rows × 13 columns



```
%%bash
data='[[7.0,0.270,0.36,20.7,0.045,45.0,170.0,1.00,100,3.00,0.45], [7.0,0.270,0.36,20.7,0.045,45.0,170.0,1.00,100,3.00,0.45]]'
echo $data

curl -d "{\"inputs\": $data}" -H 'Content-Type: application/json' 127.0.0.1:5003/invocations
```

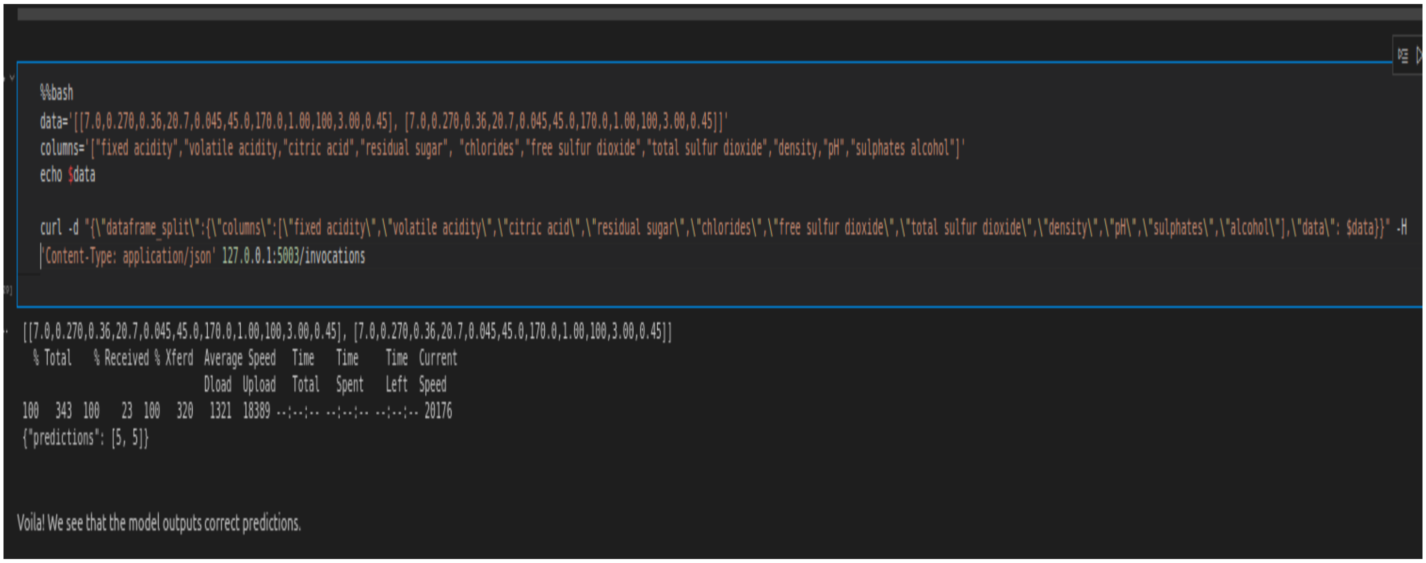
Python

```
[[7.0,0.270,0.36,20.7,0.045,45.0,170.0,1.00,100,3.00,0.45], [7.0,0.270,0.36,20.7,0.045,45.0,170.0,1.00,100,3.00,0.45]]
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
/home/maviaalamkhan/Documents/mlop/data_engineering_bootcamp_2303/tasks/3_machine_learning_essentials/day_4_mlops/mlops-student/lib/python3.10/site-packages/sklearn/base.py:43
  warnings.warn(
100   153  100    23  100   130   1611   9111 --:--:-- --:--:-- --:--:-- 10928
{"predictions": [5, 5]}
```

```
%%bash
data='[[7.0,0.270,0.36,20.7,0.045,45.0,170.0,1.00,100,3.00,0.45], [7.0,0.270,0.36,20.7,0.045,45.0,170.0,1.00,100,3.00,0.45]]'
echo $data

curl -d "{\"instances\": $data}" -H 'Content-Type: application/json' 127.0.0.1:5003/invocations
```

Python

```
[[7.0,0.270,0.36,20.7,0.045,45.0,170.0,1.00,100,3.00,0.45], [7.0,0.270,0.36,20.7,0.045,45.0,170.0,1.00,100,3.00,0.45]]
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
/home/maviaalamkhan/Documents/mlop/data_engineering_bootcamp_2303/tasks/3_machine_learning_essentials/day_4_mlops/mlops-student/lib/python3.10/site-packages/sklearn/base.py:43
  warnings.warn(
100   156  100    23  100   133   1900  10987 --:--:-- --:--:-- --:--:-- 13000
{"predictions": [5, 5]}
```



```
%%bash
data='[[7.0,0.270,0.36,20.7,0.045,45.0,170.0,1.00,100,3.00,0.45], [7.0,0.270,0.36,20.7,0.045,45.0,170.0,1.00,100,3.00,0.45]]'
columns='["fixed acidity","volatile acidity,"citric acid","residual sugar", "chlorides","free sulfur dioxide","total sulfur dioxide","density,"pH","sulphates alcohol"]'
echo $data

curl -d "{\"dataframe_split\":{\"columns\":[\"fixed acidity\",\"volatile acidity\",\"citric acid\",\"residual sugar\",\"chlorides\",\"free sulfur dioxide\",\"total sulfur dioxide\",\"density\",\"pH\",\"sulphates\",\"alcohol\"],\"data\": $data}}" -H
'Content-Type: application/json' 127.0.0.1:5003/invocations
```

```
[[7.0,0.270,0.36,20.7,0.045,45.0,170.0,1.00,100,3.00,0.45], [7.0,0.270,0.36,20.7,0.045,45.0,170.0,1.00,100,3.00,0.45]]
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   343  100    23  100   320   1321  18389 --:--:-- --:--:-- --:--:-- 20176
{"predictions": [5, 5]}
```

Voila! We see that the model outputs correct predictions.