Assignment # 4

Session Fall 2025 – BSAI

Submitted by:
Soban Athar 22i-0535

Submitted to:
Dr. Mateen Yaqoob

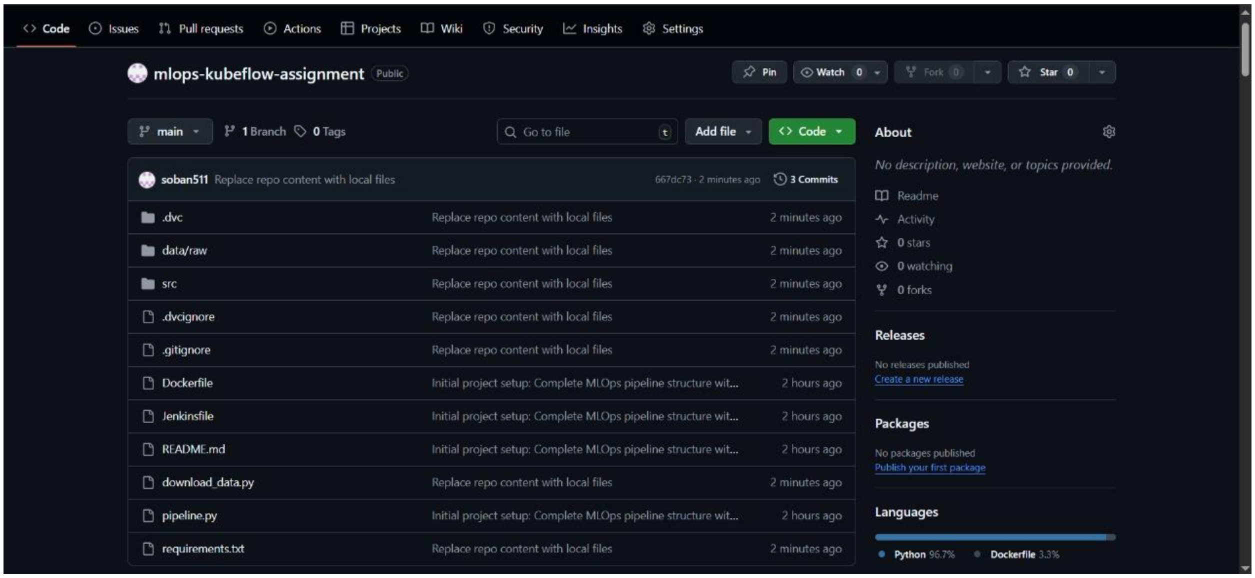
Department of Artificial Intelligence

National University of Computer and Emerging sciences,

FAST

Task 1:

Repository File Structure:



soban511 Replace repo content with local files 667dc73 · 2 minutes ago 3 Commits

.dvc	Replace repo content with local files	2 minutes ago
data/raw	Replace repo content with local files	2 minutes ago
src	Replace repo content with local files	2 minutes ago
.dvcignore	Replace repo content with local files	2 minutes ago
.gitignore	Replace repo content with local files	2 minutes ago
Dockerfile	Initial project setup: Complete MLOps pipeline structure wit...	2 hours ago
Jenkinsfile	Initial project setup: Complete MLOps pipeline structure wit...	2 hours ago
README.md	Initial project setup: Complete MLOps pipeline structure wit...	2 hours ago
download_data.py	Replace repo content with local files	2 minutes ago
pipeline.py	Initial project setup: Complete MLOps pipeline structure wit...	2 hours ago
requirements.txt	Replace repo content with local files	2 minutes ago

About

No description, website, or topics provided.

Readme
Activity
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

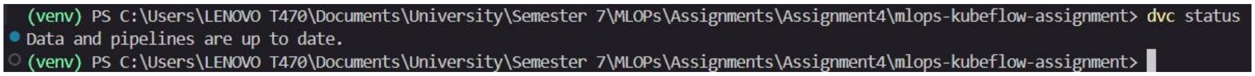
Packages

No packages published
Publish your first package

Languages

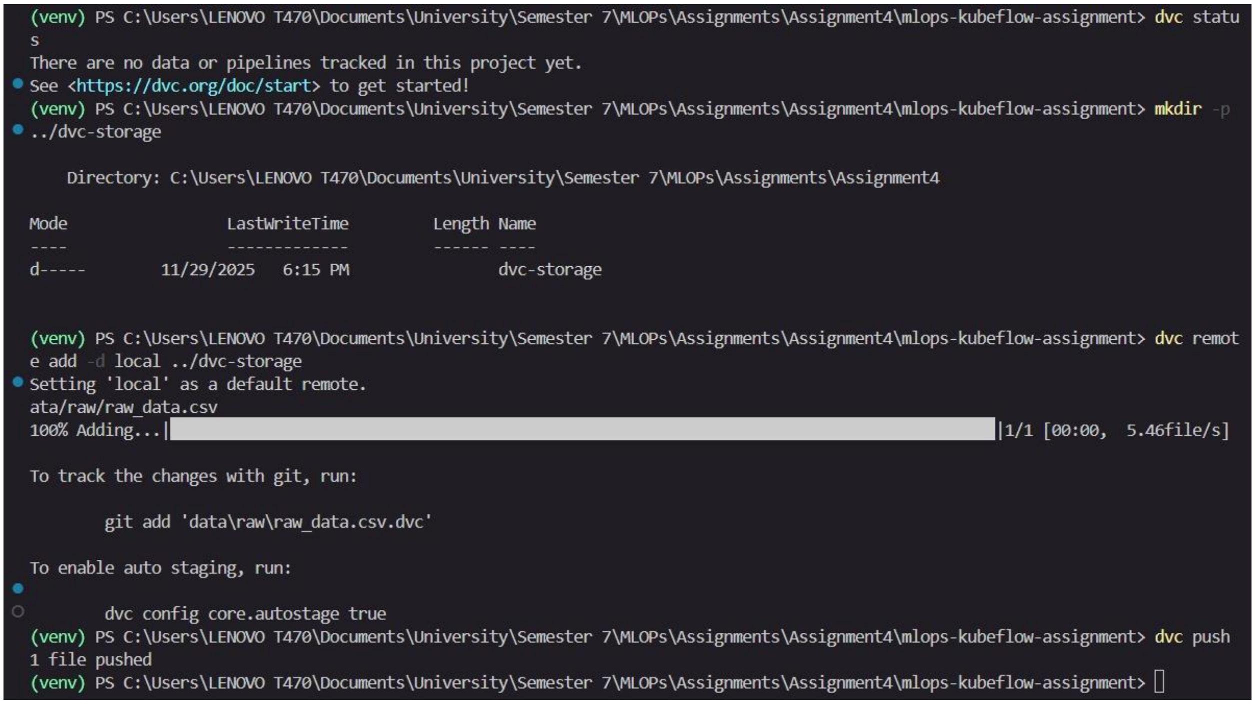
● Python 96.7% ● Dockerfile 3.3%

Command “dvc status”:



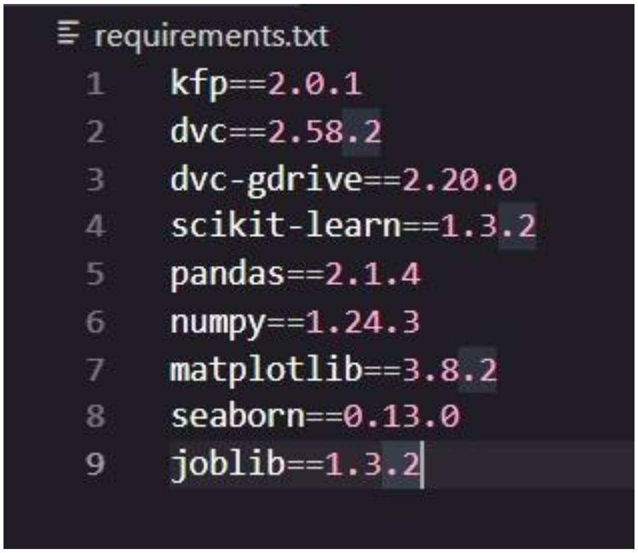
```
(venv) PS C:\Users\LENOVO T470\Documents\University\Semester 7\MLOPs\Assignments\Assignment4\mlops-kubeflow-assignment> dvc status
Data and pipelines are up to date.
(venv) PS C:\Users\LENOVO T470\Documents\University\Semester 7\MLOPs\Assignments\Assignment4\mlops-kubeflow-assignment>
```

Command “dvc push”:



```
(venv) PS C:\Users\LENOVO T470\Documents\University\Semester 7\MLOPs\Assignments\Assignment4\mlops-kubeflow-assignment> dvc statu
s
There are no data or pipelines tracked in this project yet.
See <https://dvc.org/doc/start> to get started!
(venv) PS C:\Users\LENOVO T470\Documents\University\Semester 7\MLOPs\Assignments\Assignment4\mlops-kubeflow-assignment> mkdir -p
../dvc-storage

    Directory: C:\Users\LENOVO T470\Documents\University\Semester 7\MLOPs\Assignments\Assignment4

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/29/2025   6:15 PM                dvc-storage


(venv) PS C:\Users\LENOVO T470\Documents\University\Semester 7\MLOPs\Assignments\Assignment4\mlops-kubeflow-assignment> dvc remot
e add -d local ../dvc-storage
Setting 'local' as a default remote.
ata/raw/raw_data.csv
100% Adding...|████████████████████████████████████████████████████████|1/1 [00:00,  5.46file/s]

To track the changes with git, run:

        git add 'data\raw\raw_data.csv.dvc'

To enable auto staging, run:

        dvc config core.autostage true
(venv) PS C:\Users\LENOVO T470\Documents\University\Semester 7\MLOPs\Assignments\Assignment4\mlops-kubeflow-assignment> dvc push
1 file pushed
(venv) PS C:\Users\LENOVO T470\Documents\University\Semester 7\MLOPs\Assignments\Assignment4\mlops-kubeflow-assignment>
```

Content of requirements.txt:

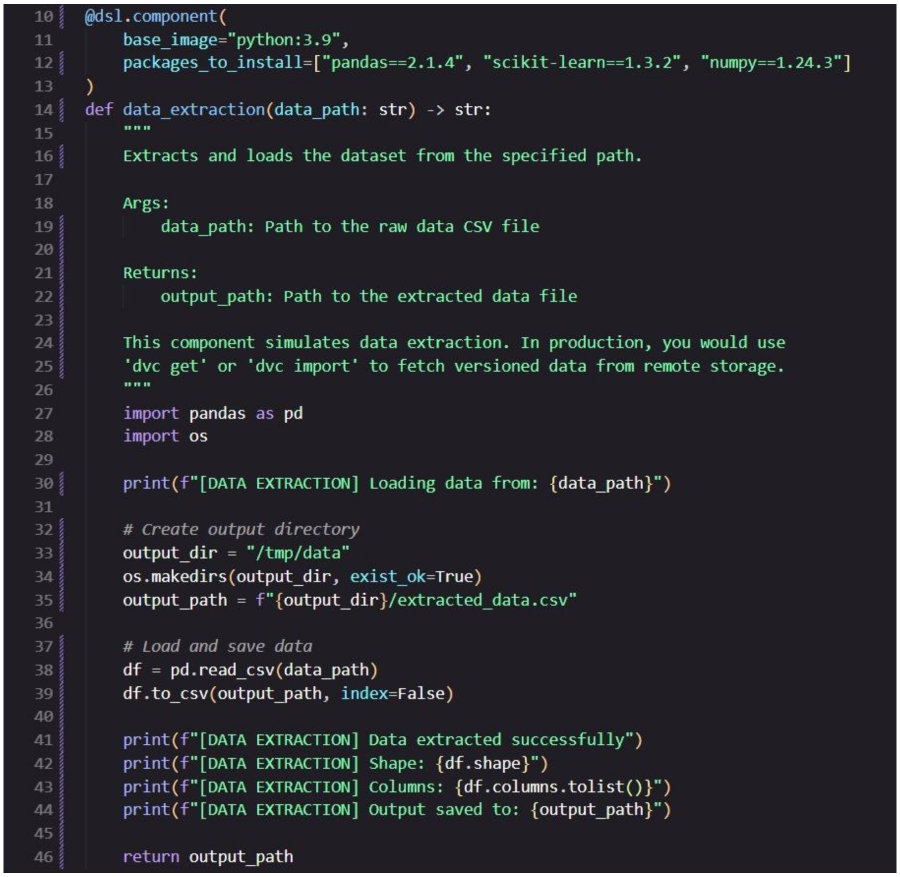


requirements.txt

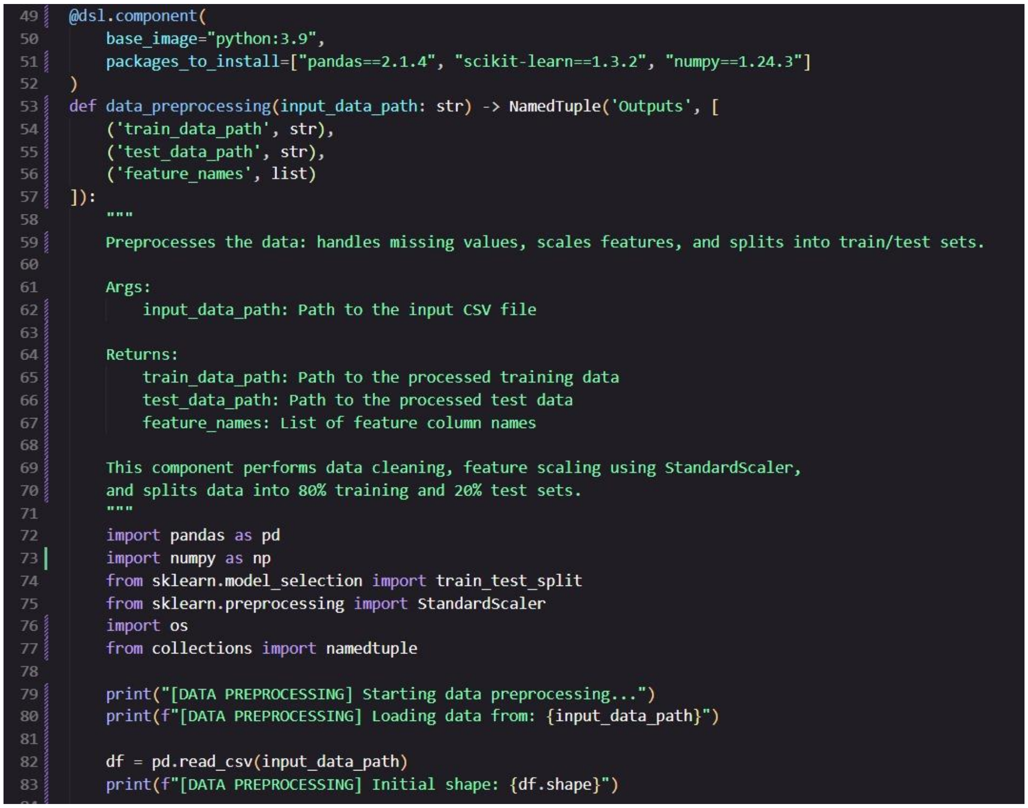
```
kfp==2.0.1
dvc==2.58.2
dvc-gdrive==2.20.0
scikit-learn==1.3.2
pandas==2.1.4
numpy==1.24.3
matplotlib==3.8.2
seaborn==0.13.0
joblib==1.3.2
```

Task 2:

src/pipeline_components.py file:

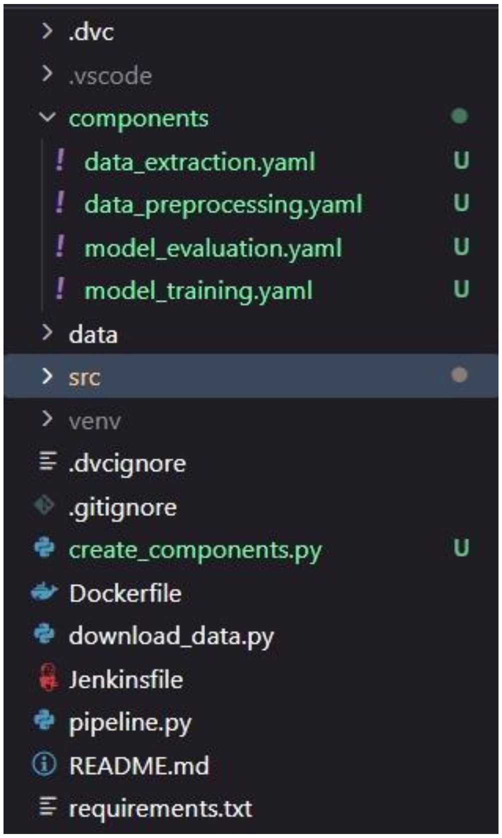


```python
@dsl.component(
    base_image="python:3.9",
    packages_to_install=["pandas==2.1.4", "scikit-learn==1.3.2", "numpy==1.24.3"]
)
def data_extraction(data_path: str) -> str:
    """
    Extracts and loads the dataset from the specified path.

    Args:
        data_path: Path to the raw data CSV file

    Returns:
        output_path: Path to the extracted data file

    This component simulates data extraction. In production, you would use
    'dvc get' or 'dvc import' to fetch versioned data from remote storage.
    """
    import pandas as pd
    import os

    print(f"[DATA EXTRACTION] Loading data from: {data_path}")

    # Create output directory
    output_dir = "/tmp/data"
    os.makedirs(output_dir, exist_ok=True)
    output_path = f"{output_dir}/extracted_data.csv"

    # Load and save data
    df = pd.read_csv(data_path)
    df.to_csv(output_path, index=False)

    print(f"[DATA EXTRACTION] Data extracted successfully")
    print(f"[DATA EXTRACTION] Shape: {df.shape}")
    print(f"[DATA EXTRACTION] Columns: {df.columns.tolist()}")
    print(f"[DATA EXTRACTION] Output saved to: {output_path}")

    return output_path
```



```python
@dsl.component(
    base_image="python:3.9",
    packages_to_install=["pandas==2.1.4", "scikit-learn==1.3.2", "numpy==1.24.3"]
)
def data_preprocessing(input_data_path: str) -> NamedTuple('Outputs', [
    ('train_data_path', str),
    ('test_data_path', str),
    ('feature_names', list)
]):
    """
    Preprocesses the data: handles missing values, scales features, and splits into train/test sets.

    Args:
        input_data_path: Path to the input CSV file

    Returns:
        train_data_path: Path to the processed training data
        test_data_path: Path to the processed test data
        feature_names: List of feature column names

    This component performs data cleaning, feature scaling using StandardScaler,
    and splits data into 80% training and 20% test sets.
    """
    import pandas as pd
    import numpy as np
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import StandardScaler
    import os
    from collections import namedtuple

    print("[DATA PREPROCESSING] Starting data preprocessing...")
    print(f"[DATA PREPROCESSING] Loading data from: {input_data_path}")

    df = pd.read_csv(input_data_path)
    print(f"[DATA PREPROCESSING] Initial shape: {df.shape}")
```

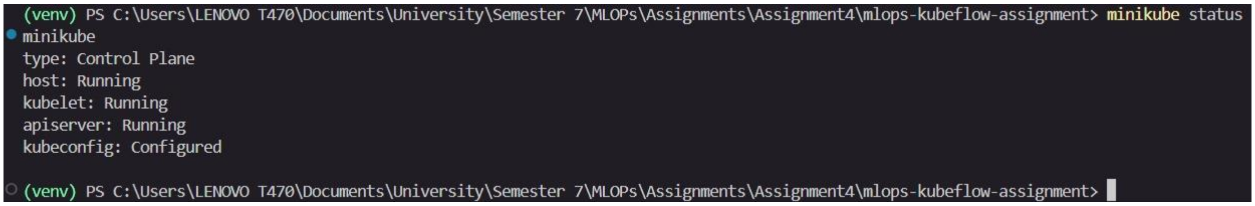
YAML files of components directory:



```
> .dvc
> .vscode
v components                     ●
    ! data_extraction.yaml       U
    ! data_preprocessing.yaml    U
    ! model_evaluation.yaml      U
    ! model_training.yaml        U
> data
> src                            ●
> venv
  .dvcignore
  .gitignore
  create_components.py           U
  Dockerfile
  download_data.py
  Jenkinsfile
  pipeline.py
  README.md
  requirements.txt
```

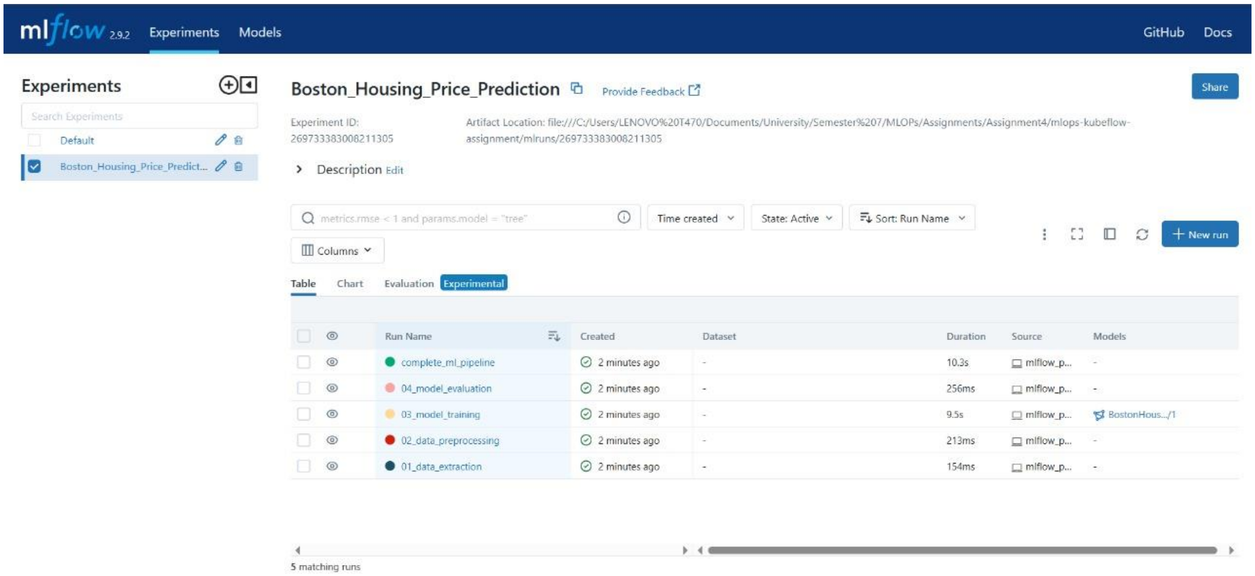
Task 3:

Command “minicube status”:



```
(venv) PS C:\Users\LENOVO T470\Documents\University\Semester 7\MLOPs\Assignments\Assignment4\mlops-kubeflow-assignment> minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

(venv) PS C:\Users\LENOVO T470\Documents\University\Semester 7\MLOPs\Assignments\Assignment4\mlops-kubeflow-assignment>
```

Mlflow Pipelines UI:



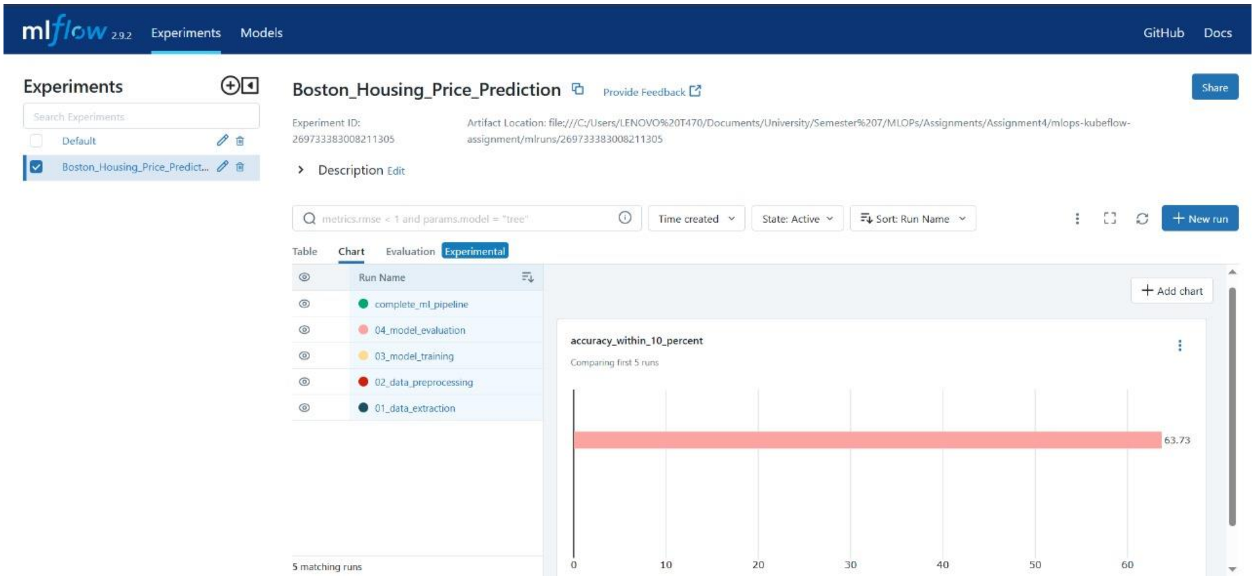
Boston_Housing_Price_Prediction

Experiment ID: 269733383008211305

Artifact Location: file:///C:/Users/LENOVO%20T470/Documents/University/Semester%207/MLOPs/Assignments/Assignment4/mlops-kubeflow-assignment/mlruns/269733383008211305

Run Name	Created	Dataset	Duration	Source	Models
complete_ml_pipeline	2 minutes ago	-	10.3s	mlflow_p...	-
04_model_evaluation	2 minutes ago	-	256ms	mlflow_p...	-
03_model_training	2 minutes ago	-	9.5s	mlflow_p...	BostonHous.../1
02_data_preprocessing	2 minutes ago	-	213ms	mlflow_p...	-
01_data_extraction	2 minutes ago	-	154ms	mlflow_p...	-

5 matching runs



Boston_Housing_Price_Prediction

Experiment ID: 269733383008211305

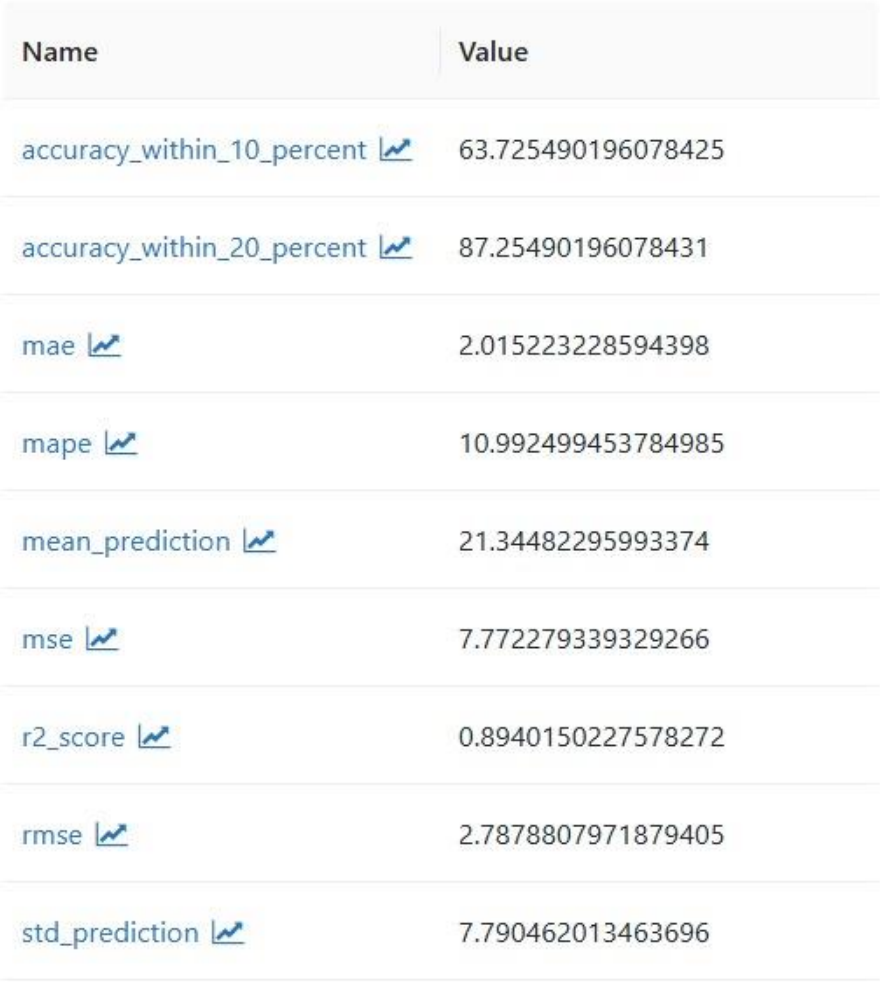
Artifact Location: file:///C:/Users/LENOVO%20T470/Documents/University/Semester%207/MLOPs/Assignments/Assignment4/mlops-kubeflow-assignment/mlruns/269733383008211305

Run Name
complete_ml_pipeline
04_model_evaluation
03_model_training
02_data_preprocessing
01_data_extraction

5 matching runs

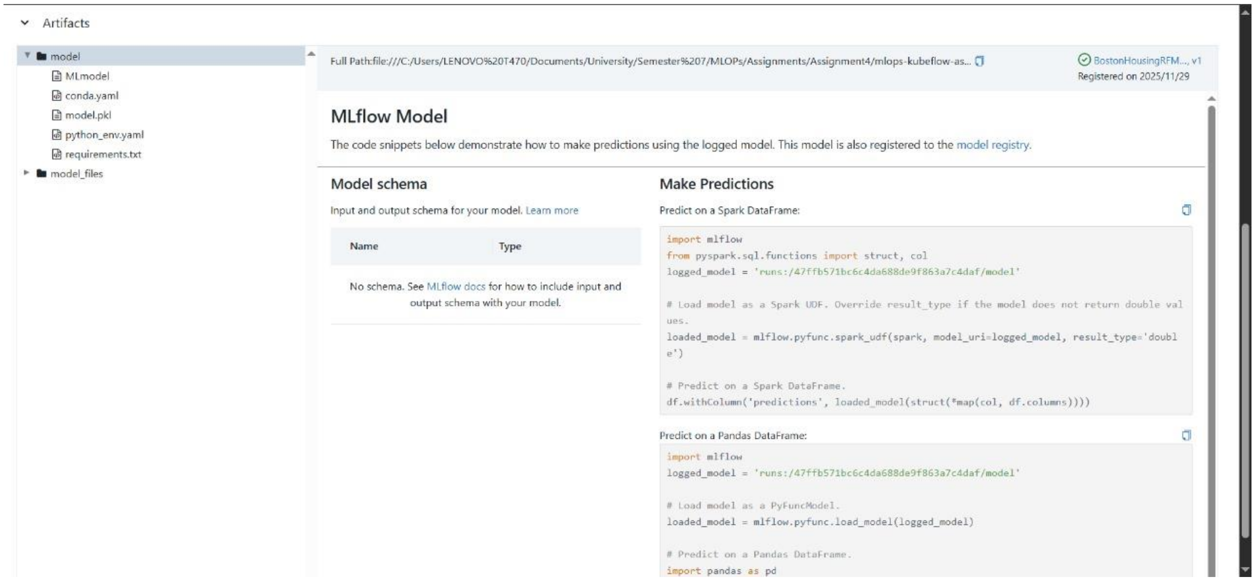
accuracy_within_10_percent

Showing the output (accuracy):



Name	Value
accuracy_within_10_percent	63.725490196078425
accuracy_within_20_percent	87.25490196078431
mae	2.015223228594398
mape	10.992499453784985
mean_prediction	21.34482295993374
mse	7.772279339329266
r2_score	0.8940150227578272
rmse	2.7878807971879405
std_prediction	7.790462013463696

Artifacts:



Full Path:file:///C:/Users/LENOVO%20T470/Documents/University/Semester%207/MLOPs/Assignments/Assignment4/mlops-kubeflow-as...

BostonHousingRFM..., v1
Registered on 2025/11/29

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. This model is also registered to the model registry.

Model schema

Input and output schema for your model. Learn more

Name	Type

No schema. See MLflow docs for how to include input and output schema with your model.

Make Predictions

Predict on a Spark DataFrame:

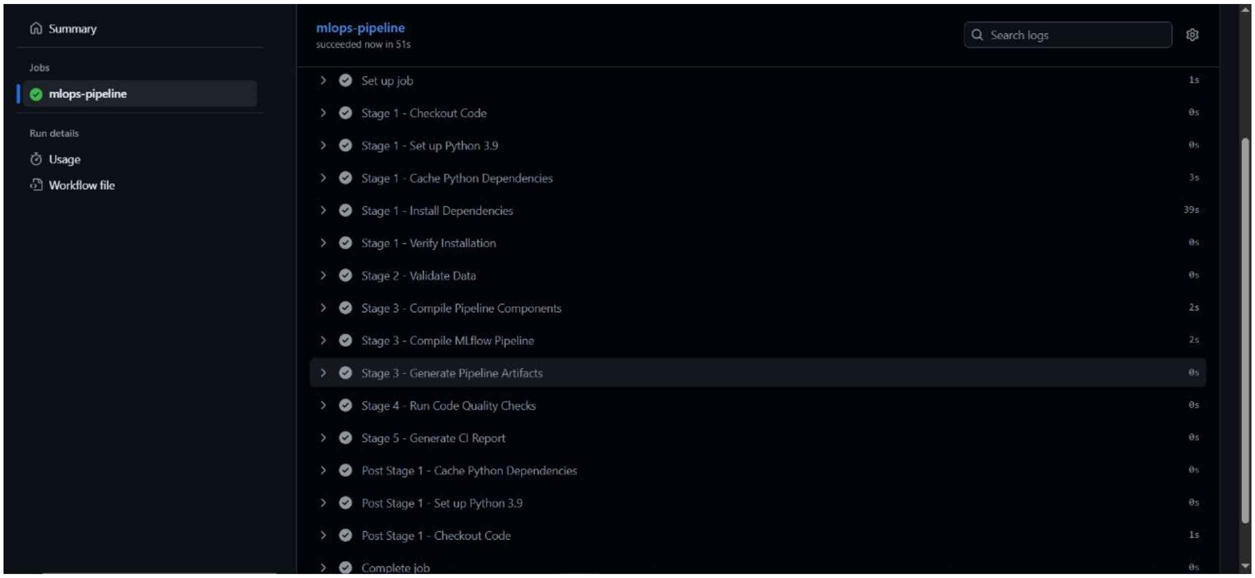
```
import mlflow
from pyspark.sql.functions import struct, col
logged_model = 'runs:/47ffb571bc6c4da688de9f863a7c4daf/model'

# Load model as a Spark UDF. Override result_type if the model does not return double val
ues.
loaded_model = mlflow.pyfunc.spark_udf(spark, model_uri=logged_model, result_type='doubl
e')

# Predict on a Spark DataFrame.
df.withColumn('predictions', loaded_model(struct(*map(col, df.columns))))
```

Predict on a Pandas DataFrame:

```
import mlflow
logged_model = 'runs:/47ffb571bc6c4da688de9f863a7c4daf/model'

# Load model as a PyFuncModel.
loaded_model = mlflow.pyfunc.load_model(logged_model)

# Predict on a Pandas DataFrame.
import pandas as pd
```

Task 4:

Successfull stages execution:



mlops-pipeline
succeeded now in 51s

```
Set up job                                   1s
Stage 1 - Checkout Code                      0s
Stage 1 - Set up Python 3.9                  0s
Stage 1 - Cache Python Dependencies          3s
Stage 1 - Install Dependencies              39s
Stage 1 - Verify Installation                0s
Stage 2 - Validate Data                      0s
Stage 3 - Compile Pipeline Components        2s
Stage 3 - Compile MLflow Pipeline            2s
Stage 3 - Generate Pipeline Artifacts        0s
Stage 4 - Run Code Quality Checks            0s
Stage 5 - Generate CI Report                 0s
Post Stage 1 - Cache Python Dependencies     0s
Post Stage 1 - Set up Python 3.9             0s
Post Stage 1 - Checkout Code                 1s
Complete job                                 0s
```

Task 5:

Repository URL:

https://github.com/soban511/mlops-kubeflow-assignment.git