Project Documentation

The project focuses on learning to use tools for conducting data set analysis.
The selected dataset was downloaded from the Kaggle platform

- Indicators of Heart Disease (2022 UPDATE)

The task was implemented using VSCode, Kaggle, MLFlow, and a local installation on the computer's hard drive. (The rationale was access to a laptop with good technical specifications: processor, RAM, fast high-capacity SSD drives.)

On GitHub, the repository Pawel20240101/MLOps_PZ contains all the necessary information to reproduce and run the project.

Project Structure

- Data – directory for storing the dataset downloaded from Kaggle
- Doc – project documentation in .docx and .pdf formats
- Eksploracja – folder for .png files generated during input data analysis
- Models – directory for locally saved classifier models and result files (best parameters) from training on the test set
- Notebooks – required Jupyter notebook files for executing the project
- Reports – .csv files containing results for each classifier
- Results – graphical presentation of results, i.e., confusion matrix, ROC curve, and for models supporting SHAP – SHAP results visualization

Additional Files

- settings.json – file specifying the environment used in the project.
  Due to occasional issues with incorrect environment selection or configuration, this file was prepared to solve the problem.
  The environment is automatically selected when launching the project.
- requirements.txt – file listing the required libraries
- README.md – general project information displayed on the main GitHub page
- Przydatne_polecenia.txt – file containing commands necessary for project execution

The project consists of three files located in the notebooks directory.

Project Reproduction

1. **Download the project from GitHub**

   In the VSCode terminal, type: **git clone https://github.com/Pawel20240101/MLOps_PZ**

2. **Create the appropriate environment on your local drive**

   Refer to the file Przydatne_polecenia.txt

3. **Activate your environment**

   Refer to the file Przydatne_polecenia.txt

4. **Install the required libraries from the requirements.txt file**

   In the VSCode terminal, type: **pip install -r requirements.txt**

5. **Start the MLflow server**

   In the VSCode terminal, type: **mlflow ui**

6. **Run the files in the notebooks directory in the following order:**

   a. 1_Import_i_eksploracja_danych_ML.ipynb

   b. 2_Przygotowanie_danych_ML.ipynb

   c. 3_ML_Workflow_pipeline.ipynb

7. **Check the results in MLflow**

   Open a browser and go to: **http://localhost:5000**

Description of the .ipynb Files Used in the Project

1_Import_i_eksploracja_danych_ML.ipynb

This notebook is responsible for downloading data from Kaggle and performing exploratory data analysis on the .csv dataset.

2_Przygotowanie_danych_ML.ipynb

This file prepares the dataset for the classification process through preprocessing and data formatting.

3_ML_Workflow_pipeline.ipynb

This is the core notebook that automates the full machine learning workflow: training models and logging results to an MLflow server. The process is executed fully automatically and consists of the following stages:

1. Installing and importing necessary libraries
2. Loading previously prepared data
3. Data preparation – identifying feature types
4. Preparing a pipeline for data preprocessing
5. Training models and logging results to MLflow
6. Comparing model performance
7. Hyperparameter tuning of the best model (using Optuna)
8. Saving the best model to a file
9. Prediction pipeline – generating predictions
10. Project summary
11. Completion

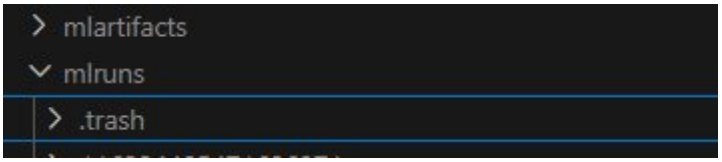
You can modify classifier parameters, experiment names, and more by editing the notebook:

Step 1, you can change the experiment name.

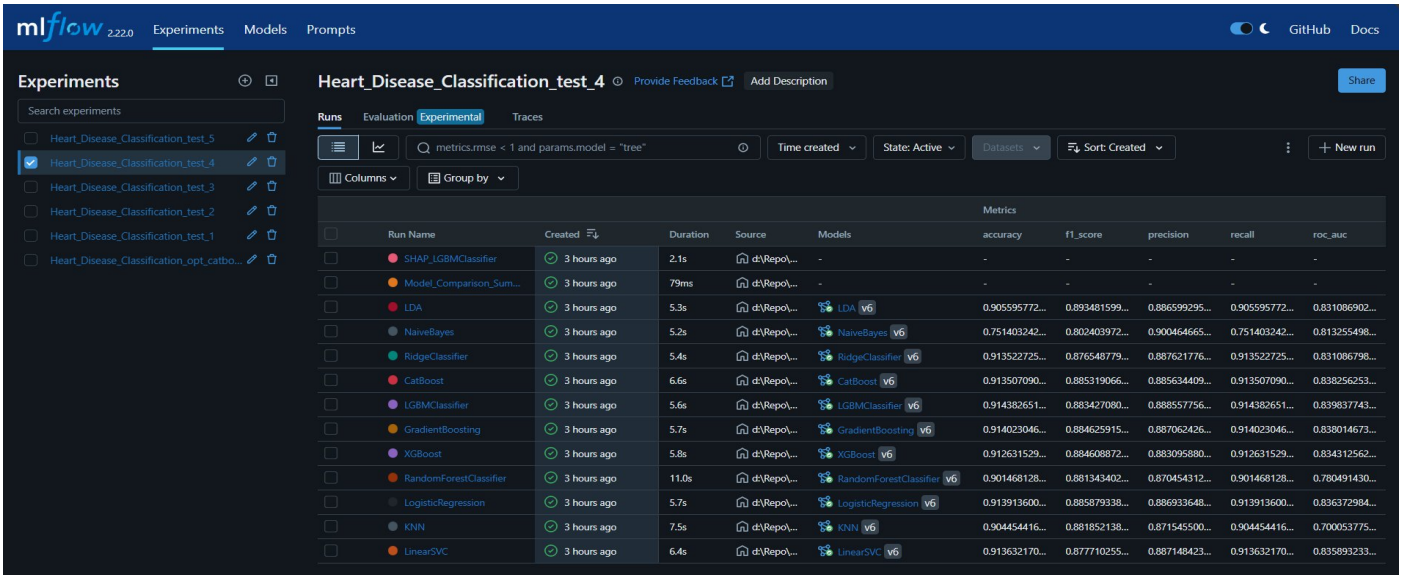
Step 5, you can include or exclude classifiers from training by commenting them out with #, or you can add new ones. In the latter case, corresponding sections later in the notebook must also be updated.

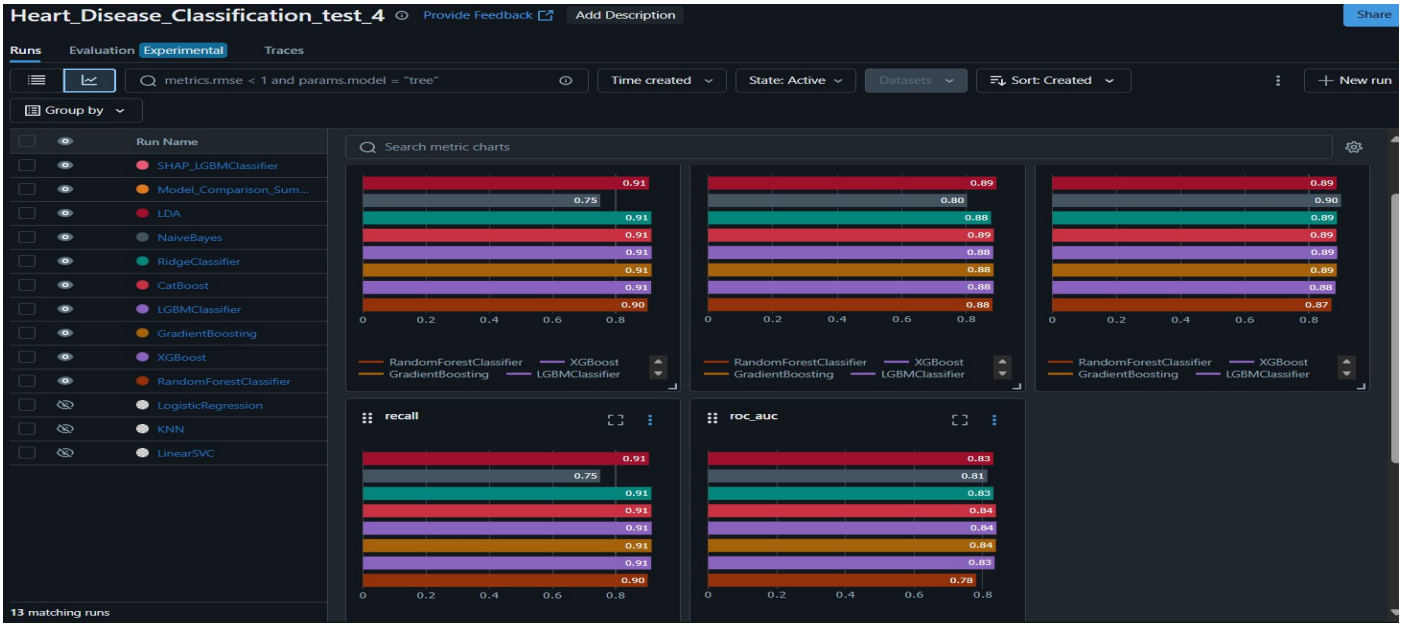
Note:

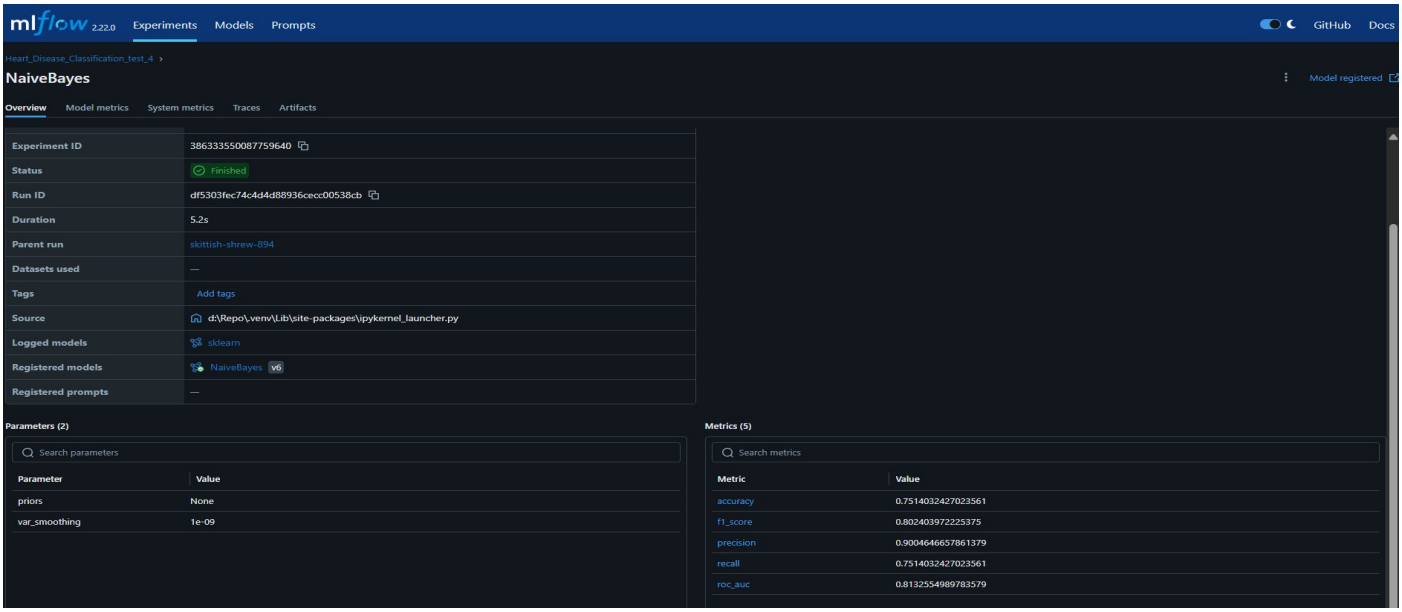
Sometimes, the .trash subdirectory is not automatically created inside the mlruns directory. If this happens, you should manually create the .trash folder to avoid runtime errors.

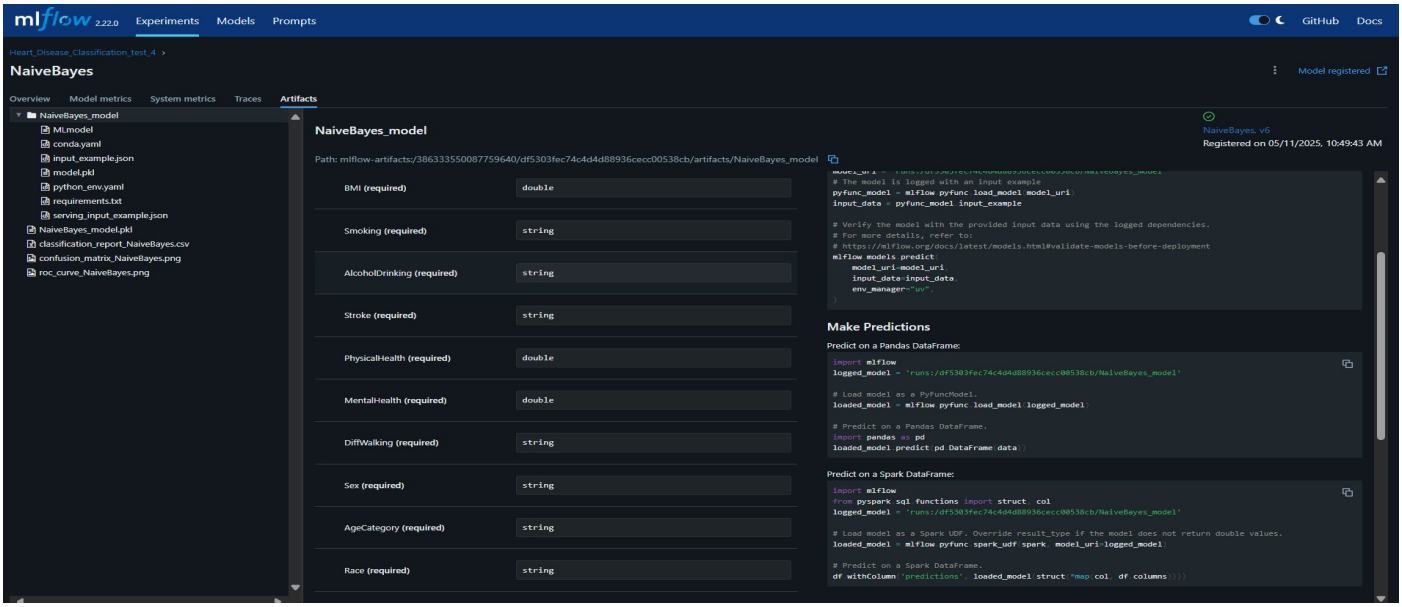


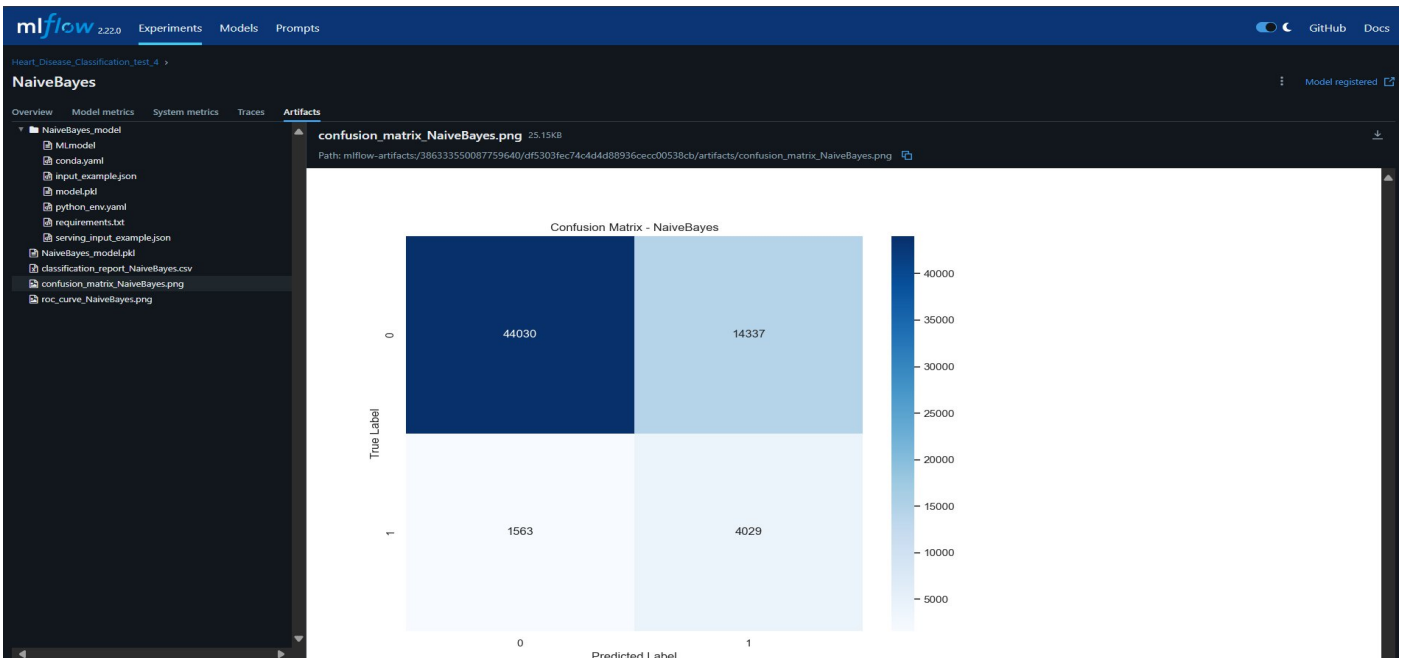
Visualization of Results Using MLflow (Examples)

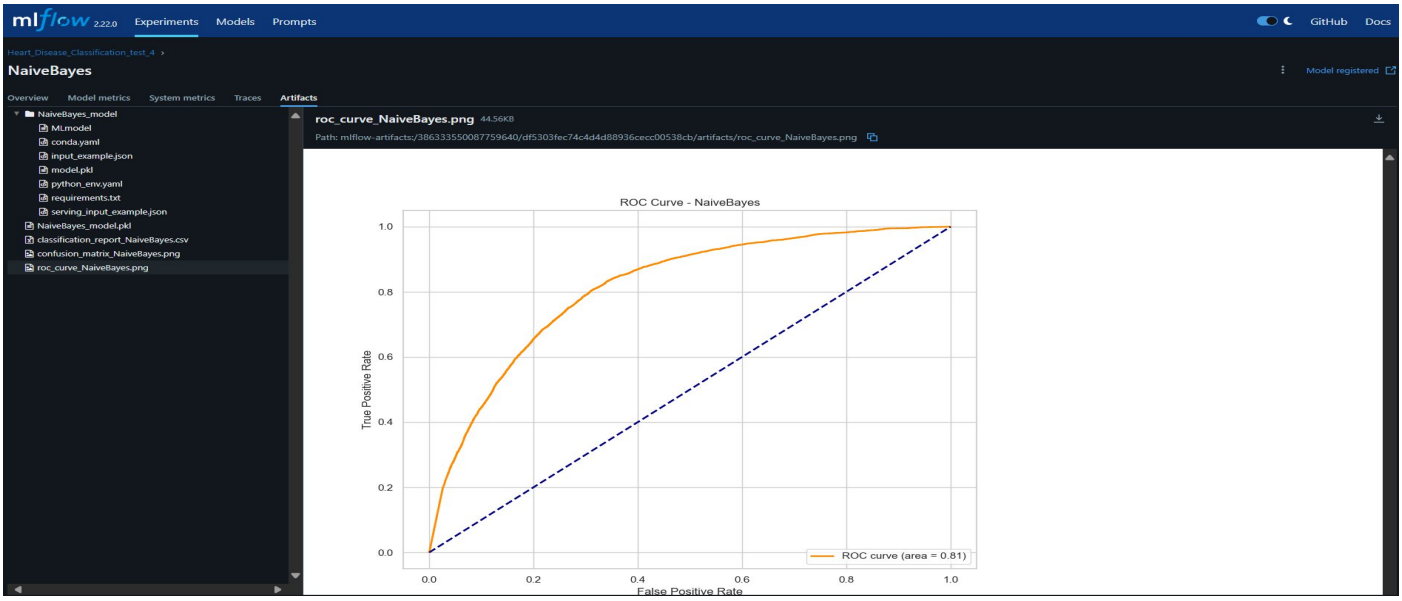


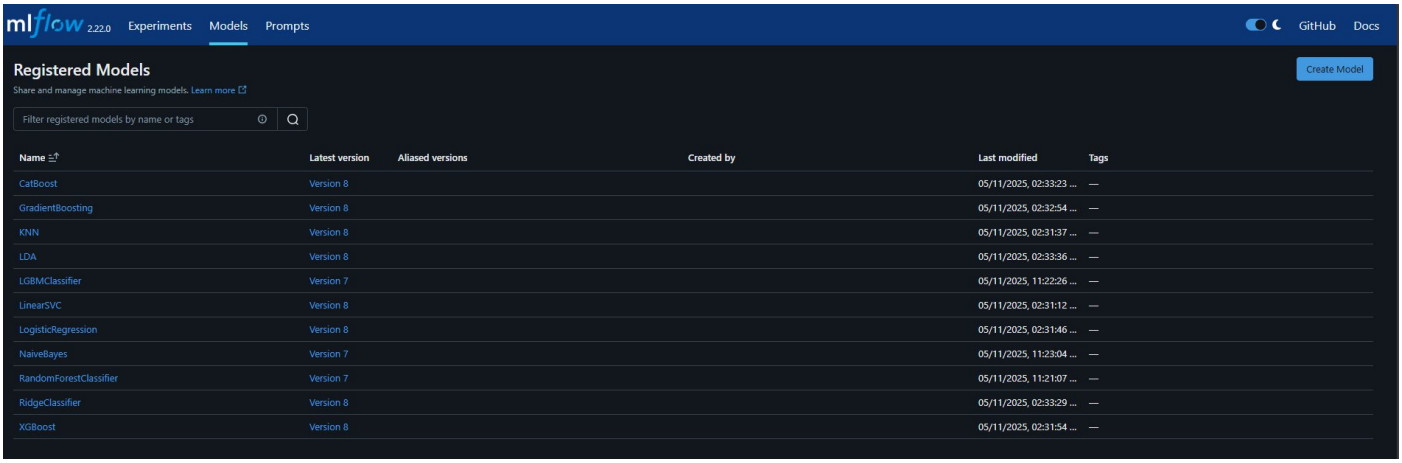


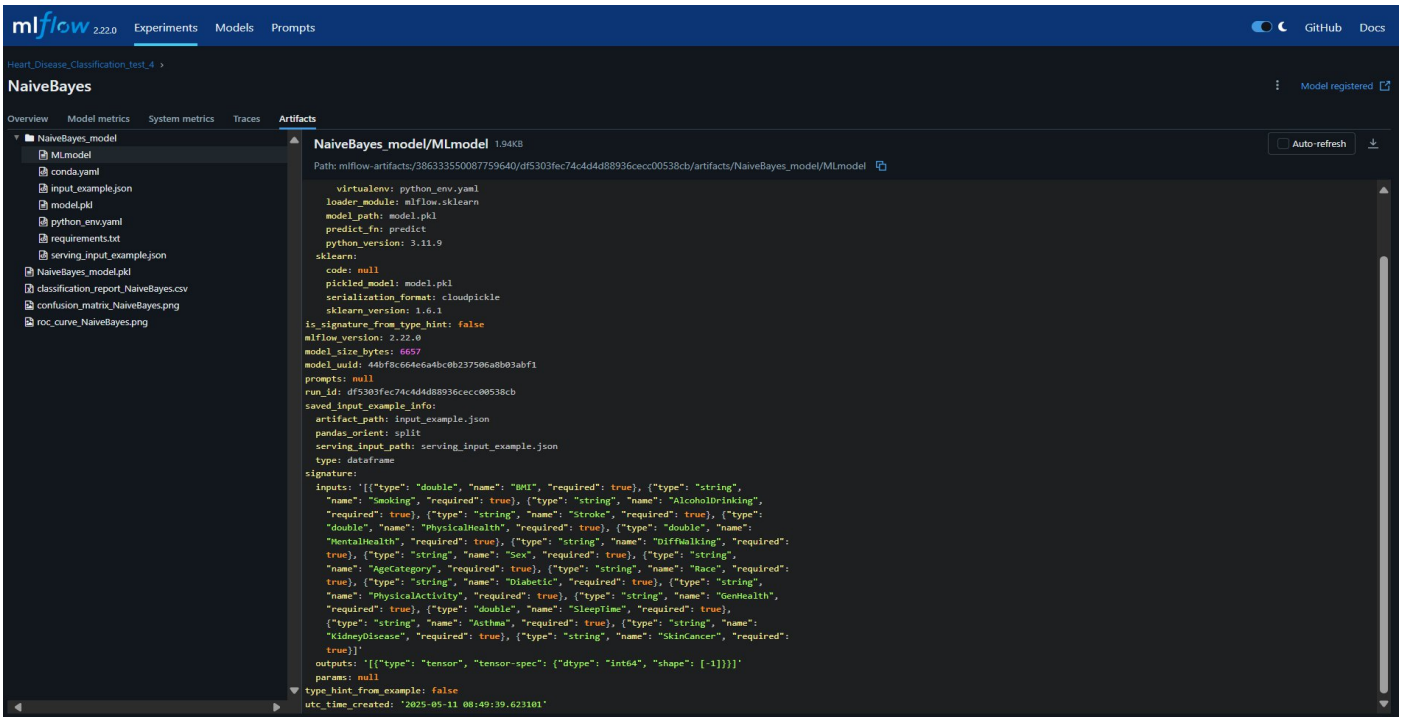



mlflow 2.22.0
Experiments
Models
Prompts
GitHub
Docs
Heart_Disease_Classification_test_4
NaiveBayes
Model registered
Overview
Model metrics
System metrics
Traces
Artifacts
NaiveBayes_model
MLmodel
conda.yaml
input_example.json
model.pkl
python_env.yaml
requirements.txt
serving_input_example.json
NaiveBayes_model.pkl
classification_report_NaiveBayes.csv
confusion_matrix_NaiveBayes.png
roc_curve_NaiveBayes.png
NaiveBayes_model
Path: mlflow-artifacts:/386333550087759640/df5303fec74c4d4d88936cecc00538cb/artifacts/NaiveBayes_model
NaiveBayes, v6
Registered on 05/11/2025, 10:49:43 AM
BMI (required) double
Smoking (required) string
AlcoholDrinking (required) string
Stroke (required) string
PhysicalHealth (required) double
MentalHealth (required) double
DiffWalking (required) string
Sex (required) string
AgeCategory (required) string
Race (required) string
Make Predictions
Predict on a Pandas DataFrame:
Predict on a Spark DataFrame:


mlflow 2.22.0
Experiments
Models
Prompts
GitHub
Docs
Heart_Disease_Classification_test_4
NaiveBayes
Model registered
Overview
Model metrics
System metrics
Traces
Artifacts
NaiveBayes_model
MLmodel
conda.yaml
input_example.json
model.pkl
python_env.yaml
requirements.txt
serving_input_example.json
NaiveBayes_model.pkl
classification_report_NaiveBayes.csv
confusion_matrix_NaiveBayes.png
roc_curve_NaiveBayes.png
confusion_matrix_NaiveBayes.png 25.15KB
Path: mlflow-artifacts:/386333550087759640/df5303fec74c4d4d88936cecc00538cb/artifacts/confusion_matrix_NaiveBayes.png
Confusion Matrix - NaiveBayes
44030
14337
1563
4029
True Label
Predicted Label
0
1
40000
35000
30000
25000
20000
15000
10000
5000


mlflow 2.22.0
Experiments
Models
Prompts
GitHub
Docs
Heart_Disease_Classification_test_4
NaiveBayes
Model registered
Overview
Model metrics
System metrics
Traces
Artifacts
NaiveBayes_model
MLmodel
conda.yaml
input_example.json
model.pkl
python_env.yaml
requirements.txt
serving_input_example.json
NaiveBayes_model.pkl
classification_report_NaiveBayes.csv
confusion_matrix_NaiveBayes.png
roc_curve_NaiveBayes.png
roc_curve_NaiveBayes.png 44.56KB
Path: mlflow-artifacts:/386333550087759640/df5303fec74c4d4d88936cecc00538cb/artifacts/roc_curve_NaiveBayes.png
ROC Curve - NaiveBayes
True Positive Rate
False Positive Rate
ROC curve (area = 0.81)
0.0
0.2
0.4
0.6
0.8
1.0


mlflow 2.22.0
Experiments
Models
Prompts
GitHub
Docs
Registered Models
Share and manage machine learning models. Learn more
Create Model
Filter registered models by name or tags
Name
Latest version
Aliased versions
Created by
Last modified
Tags
CatBoost Version 8 05/11/2025, 02:33:23 ... —
GradientBoosting Version 8 05/11/2025, 02:32:54 ... —
KNN Version 8 05/11/2025, 02:31:37 ... —
LDA Version 8 05/11/2025, 02:33:36 ... —
LGBMClassifier Version 7 05/11/2025, 11:22:26 ... —
LinearSVC Version 8 05/11/2025, 02:31:12 ... —
LogisticRegression Version 8 05/11/2025, 02:31:46 ... —
NaiveBayes Version 7 05/11/2025, 11:23:04 ... —
RandomForestClassifier Version 7 05/11/2025, 11:21:07 ... —
RidgeClassifier Version 8 05/11/2025, 02:33:29 ... —
XGBoost Version 8 05/11/2025, 02:31:54 ... —


mlflow 2.22.0
Experiments
Models
Prompts
GitHub
Docs
Heart_Disease_Classification_test_4 ›
NaiveBayes
Model registered
Overview
Model metrics
System metrics
Traces
Artifacts
NaiveBayes_model
MLmodel
conda.yaml
input_example.json
model.pkl
python_env.yaml
requirements.txt
serving_input_example.json
NaiveBayes_model.pkl
classification_report_NaiveBayes.csv
confusion_matrix_NaiveBayes.png
roc_curve_NaiveBayes.png
NaiveBayes_model/MLmodel 1.94KB
Auto-refresh
Path: mlflow-artifacts:/386333550087759640/df5303fec74c4d4d88936cecc00538cb/artifacts/NaiveBayes_model/MLmodel
virtualenv: python_env.yaml
loader_module: mlflow.sklearn
model_path: model.pkl
predict_fn: predict
python_version: 3.11.9
sklearn:
code: null
pickled_model: model.pkl
serialization_format: cloudpickle
sklearn_version: 1.6.1
is_signature_from_type_hint: false
mlflow_version: 2.22.0
model_size_bytes: 6657
model_uuid: 44bf8c664e6a4bc0b237506a8b03abf1
prompts: null
run_id: df5303fec74c4d4d88936cecc00538cb
saved_input_example_info:
artifact_path: input_example.json
pandas_orient: split
serving_input_path: serving_input_example.json
type: dataframe
signature:
inputs: '[{"type": "double", "name": "BMI", "required": true}, {"type": "string",
"name": "Smoking", "required": true}, {"type": "string", "name": "AlcoholDrinking",
"required": true}, {"type": "string", "name": "Stroke", "required": true}, {"type":
"double", "name": "PhysicalHealth", "required": true}, {"type": "double", "name":
"MentalHealth", "required": true}, {"type": "string", "name": "DiffWalking", "required":
true}, {"type": "string", "name": "Sex", "required": true}, {"type": "string",
"name": "AgeCategory", "required": true}, {"type": "string", "name": "Race", "required":
true}, {"type": "string", "name": "Diabetic", "required": true}, {"type": "string",
"name": "PhysicalActivity", "required": true}, {"type": "string", "name": "GenHealth",
"required": true}, {"type": "double", "name": "SleepTime", "required": true},
{"type": "string", "name": "Asthma", "required": true}, {"type": "string", "name":
"KidneyDisease", "required": true}, {"type": "string", "name": "SkinCancer", "required":
true}]'
outputs: '[{"type": "tensor", "tensor-spec": {"dtype": "int64", "shape": [-1]}}]'
params: null
type_hint_from_example: false
utc_time_created: '2025-05-11 08:49:39.623101'


mlflow 2.22.0
Experiments
Models
Prompts
GitHub
Docs
Heart_Disease_Classification_test_3 ›
SHAP_LGBMClassifier
Overview
Model metrics
System metrics
Traces
Artifacts
shap_summary_LGBMClassifier.png
shap_summary_LGBMClassifier.png 98.45KB
Path: mlflow-artifacts:/116224405471686871/6ba093c2407d4641bc002068c7cc48c2/artifacts/shap_summary_LGBMClassifier.png
cat__AgeCategory_18-24
cat__Sex_Female
cat__AgeCategory_25-29
cat__GenHealth_Excellent
cat__AgeCategory_30-34
cat__Smoking_No
num__BMI
cat__GenHealth_Very good
cat__Stroke_No
cat__AgeCategory_35-39
cat__AgeCategory_80 or older
cat__GenHealth_Fair
cat__AgeCategory_40-44
cat__AgeCategory_45-49
cat__DiffWalking_No
cat__Diabetic_Yes
num__PhysicalHealth
cat__AgeCategory_75-79
High
Feature value