Dokumentacja do projektu

Projekt dotyczy nauki wykorzystania narzędzi do przeprowadzenia analizy zbiorów danych.

Wybrany zbiór danych został pobrany z platformy Kaggle

- Indicators of Heart Disease (2022 UPDATE)

Zadanie zostało zrealizowane z wykorzystaniem VSCode, kaggle MLFlow oraz instalacji lokalnej na dysku komputera. (Argumentacją za tym był dostęp do laptopa o dobrych parametrach technicznych, procesor, pamięć RAM, Dyski SSD szybkie o dużej pojemności)

Na Github znajduje się Pawel20240101/MLOps_PZ w którym zawarte zostały wszystkie niezbędne informacje do odtworzenia i uruchomienia projektu.

Struktura projektu

- Data – katalog do przechowywania zbioru danych pobranego z Kaggle
- Doc – Dokumentacja projektu w postaci plików .docx oraz .pdf
- Eksploracja – Katalog w którym gromadzone są pliki .png powstałe w wyniku analizy danych wejściowych
- Models – modele klasyfikatorów pobrane lokalnie oraz plik wynikowy (najlepsze parametry) z treningu zbioru testowego
- Notebooks – Niezbędne pliku jupyter notebook wymagane do realizacji projektu
- Reports – pliki w formacie .csv – wyniki dla poszczególnych klasyfikatorów
- Results – Graficzna prezentacja wyników – tj. confusion matrix, roc curve oraz dla modeli które obsługują SHAP – prezentacja wyniku SHAP.

Dodatkowe pliki

- settings.json – plik ze wskazaniem środowiska wykorzystywanego w projekcie. Ze względu na zdarzające się sytuacje kiedy środowiska były wybrane nieprawidłowo lub był problem z ich wskazaniem przygotowałem plik który rozwiązuje ten kłopot. Środowisko jest wskazane automatycznie po uruchomieniu projektu.
- requirements.txt – Plik z wymaganymi bibliotekami.
- README.md – ogólne informacje wyświetlające się na głównej stronie projektu w serwisie Github.
- Przydatne_polecenia.txt – Plik z poleceniami niezbędnymi do realizacji projektu

Projekt składa się z trzech plików znajdujących się w katalogu notebooks.

Odtworzenie projektu

1. **Pobierz projekt z Github**

   W terminalu VScode wpisz: **git clone https://github.com/Pawel20240101/MLOps_PZ**

2. **W projekcie na Twoim dysku utwórz odpowiednie środowisko**

   Zajrzyj do pliku **Przydatne_polecenia.txt**

3. **Aktywuj swoje środowisko**

   Zajrzyj do pliku **Przydatne_polecenia.txt**

4. **Zainstaluj biblioteki z pliku requirements.txt**

   W terminalu VScode wpisz: **pip install -r requirements.txt**

5. **Uruchom serwer MLflow.**

   W terminalu VScode wpisz: **MLflow UI**

6. **Z katalogu notebooks uruchom po kolei pliki**

   a. 1_Import_i_eksploracja_danych_ML.ipynb

   b. 2_Przygotowanie_danych_ML.ipynb

   c. 3_ML_Workflow_pipeline.ipynb

7. **Sprawdź wynik w MLflow**

   Otwórz przeglądarkę i wpisz: **http://localhost:5000**

Opis plików .ipynb wykorzystanych w projekcie

1_Import_i_eksploracja_danych_ML.ipynb – plik służy do pobrania danych z Kaggle oraz wykonuje eksplorację danych ze zbioru .csv

2_Przygotowanie_danych_ML.ipynb – plik służy do wstępnego przygotowania danych do procesu kalsyfikacji

3_ML_Workflow_pipeline.ipynb – plik służy do realizacji głównego zadania czyli trenowania modeli i zapisywania wyników na serwerze MLflow – Proces jest wykonywany całkowicie automatycznie. I składa się z następujących etapów

1: Instalacja i import niezbędnych bibliotek

2: Wczytywanie danych przygotowanych wcześniej

3: Przygotowanie danych – identyfikacja kolumn

4: Przygotowanie pipeline do przetwarzania danych

5: Trenowanie modeli i logowanie wyników do MLflow

6: Porównanie wyników modeli

7: Tuning hiperparametrów najlepszego modelu (Optuna)

8: Zapis najlepszego modelu do pliku

9: Predict pipeline – generowanie predykcji

10: Podsumowanie projektu

11: Zakończenie

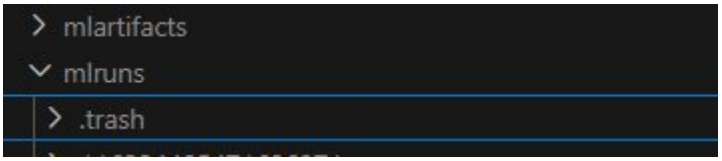
Oczywiście edytując plik można zmieniać parametry klasyfikatorów nazwy eksperymentów etc.

W punkcie 1 możemy zmieniać nazwy eksperymentu

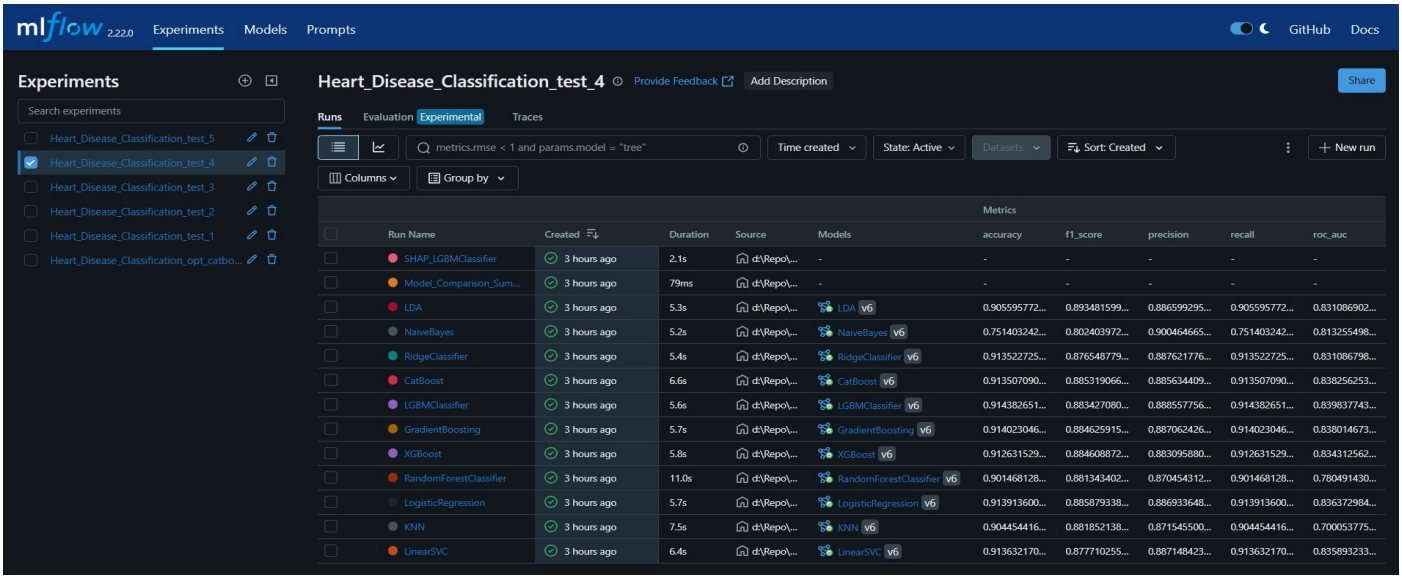
W punkcie 5 umieszczone są klasyfikatory – możemy je wyłączać z uczenie wstawiając znak # przed klasyfikator lub dodać nowe – w takim przypadku trzeba zmodyfikować kod znajdujący się w dalszej części notebooka.

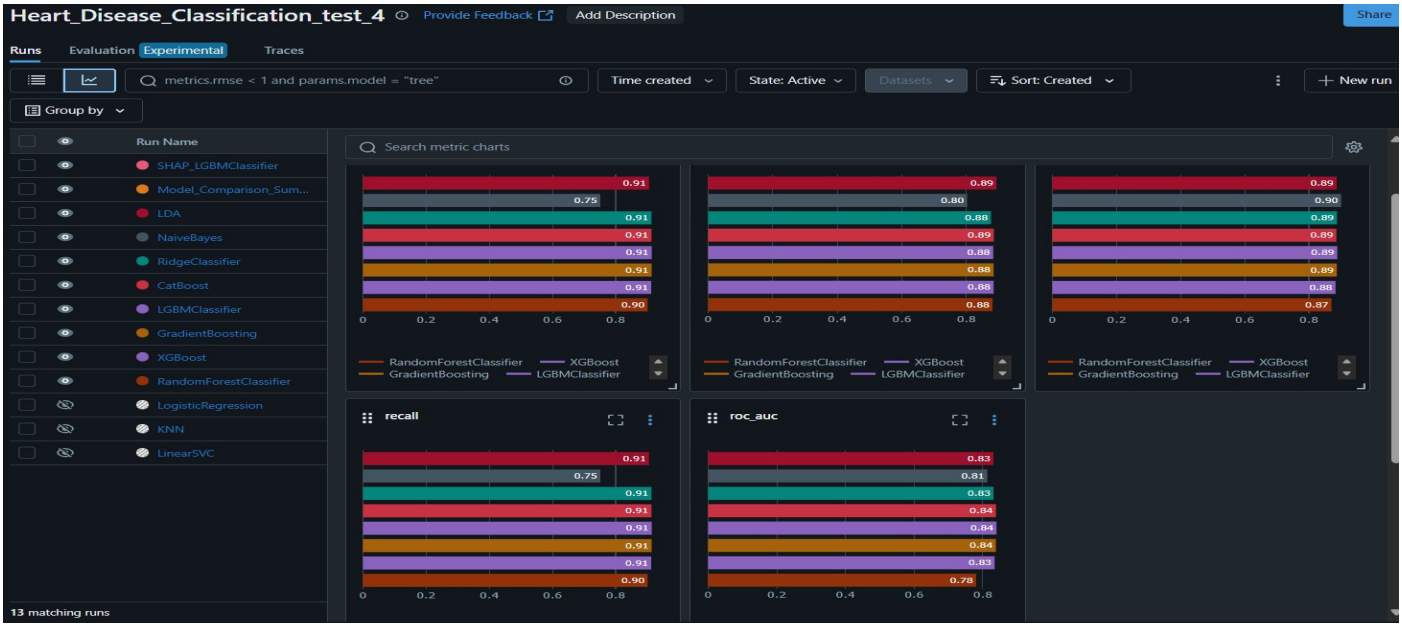
Uwaga.

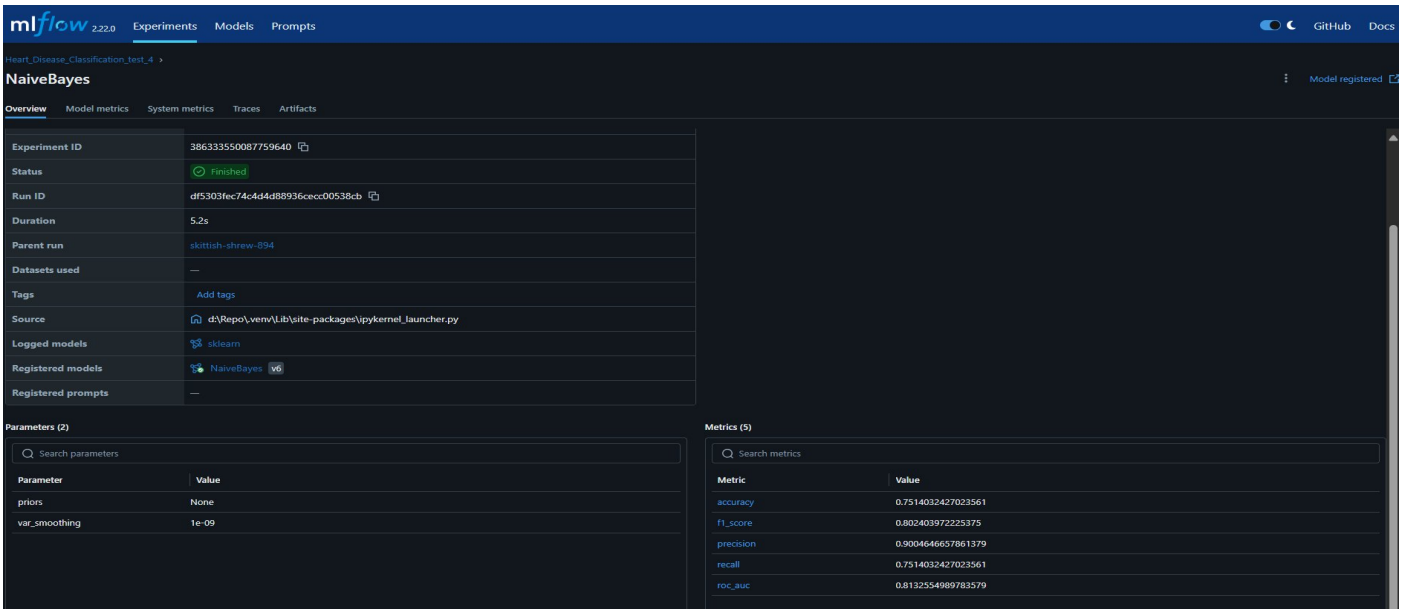
Czasami w trakcie automatycznego tworzenia katalogu mlruns w środku niego nie jest tworzony katalog .trash. W takim przypadku należy ręcznie stworzyć taki katalog.

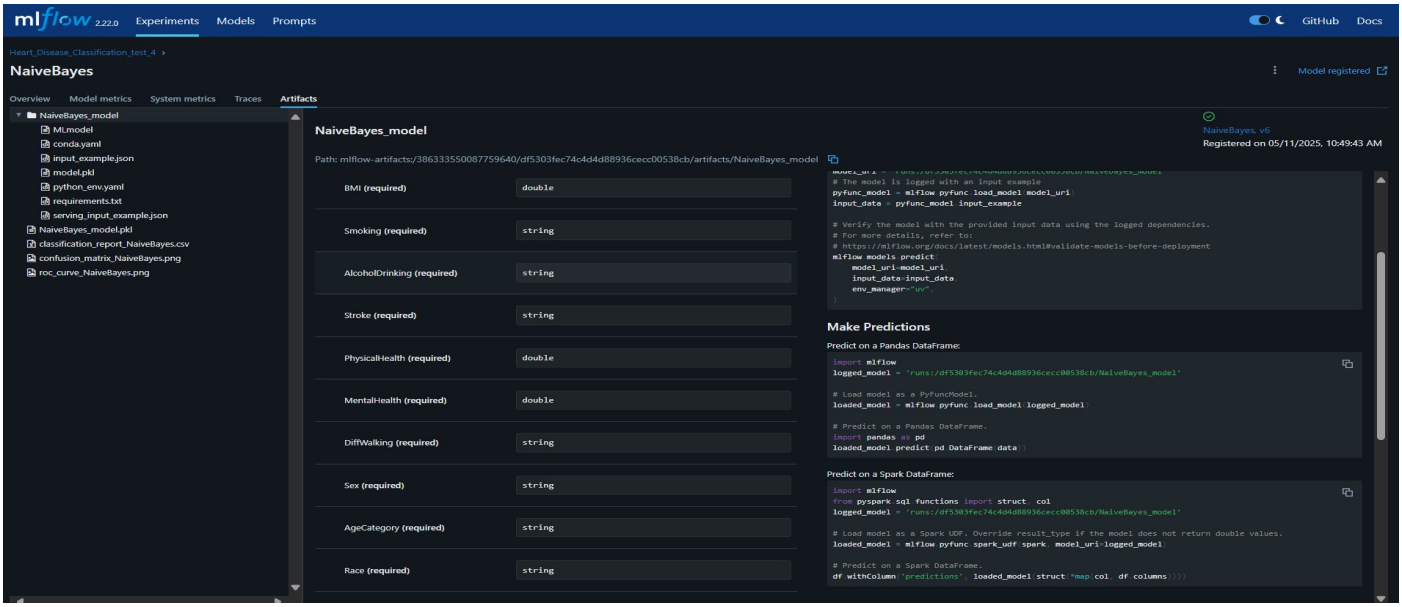


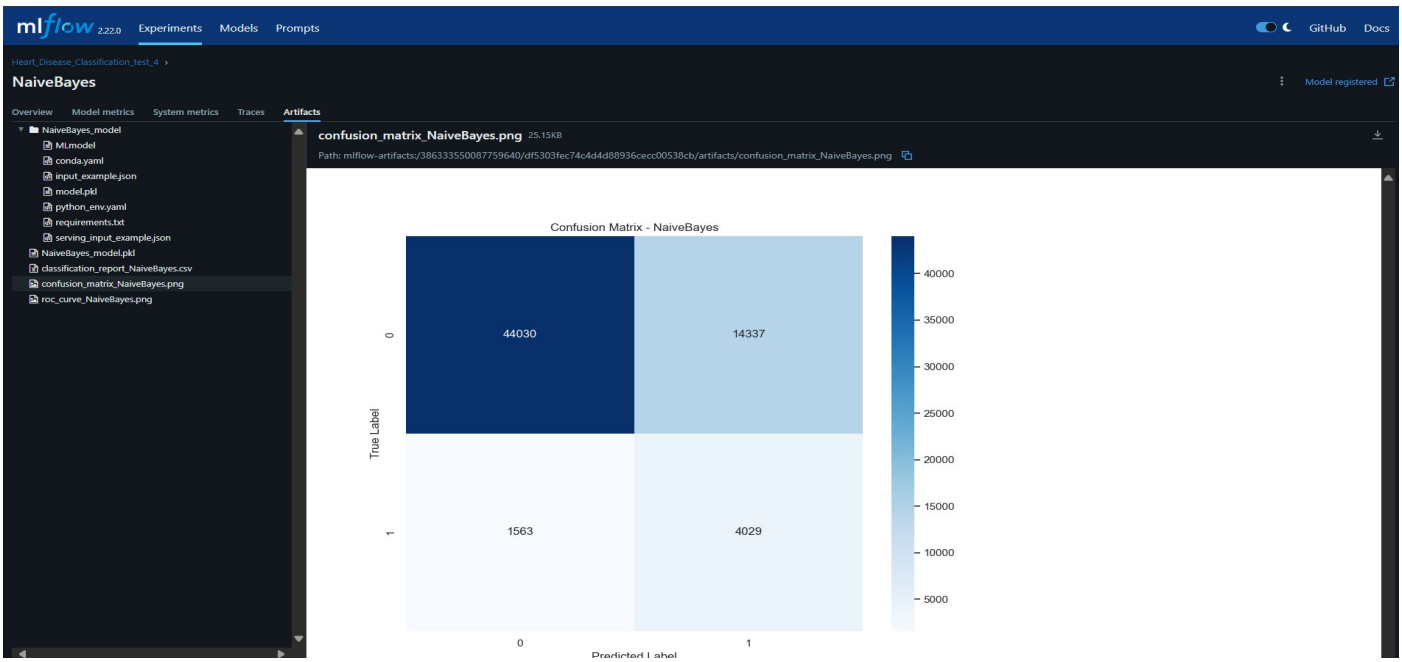
Wizualizacja wyników z wykorzystaniem MLflow (przykłady)

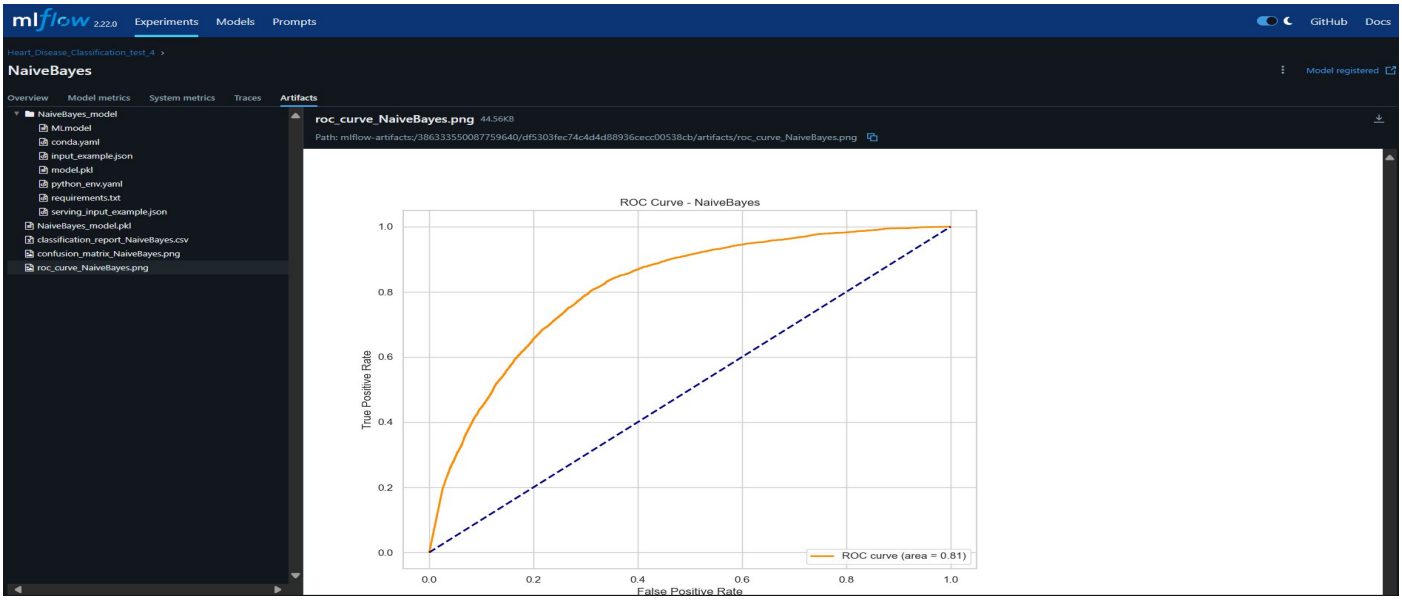


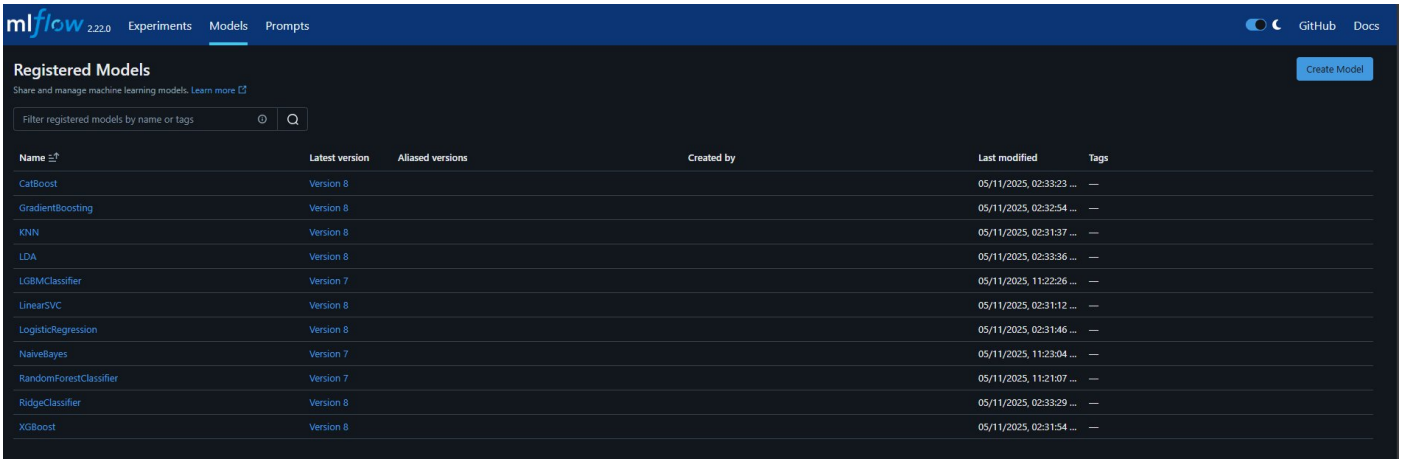


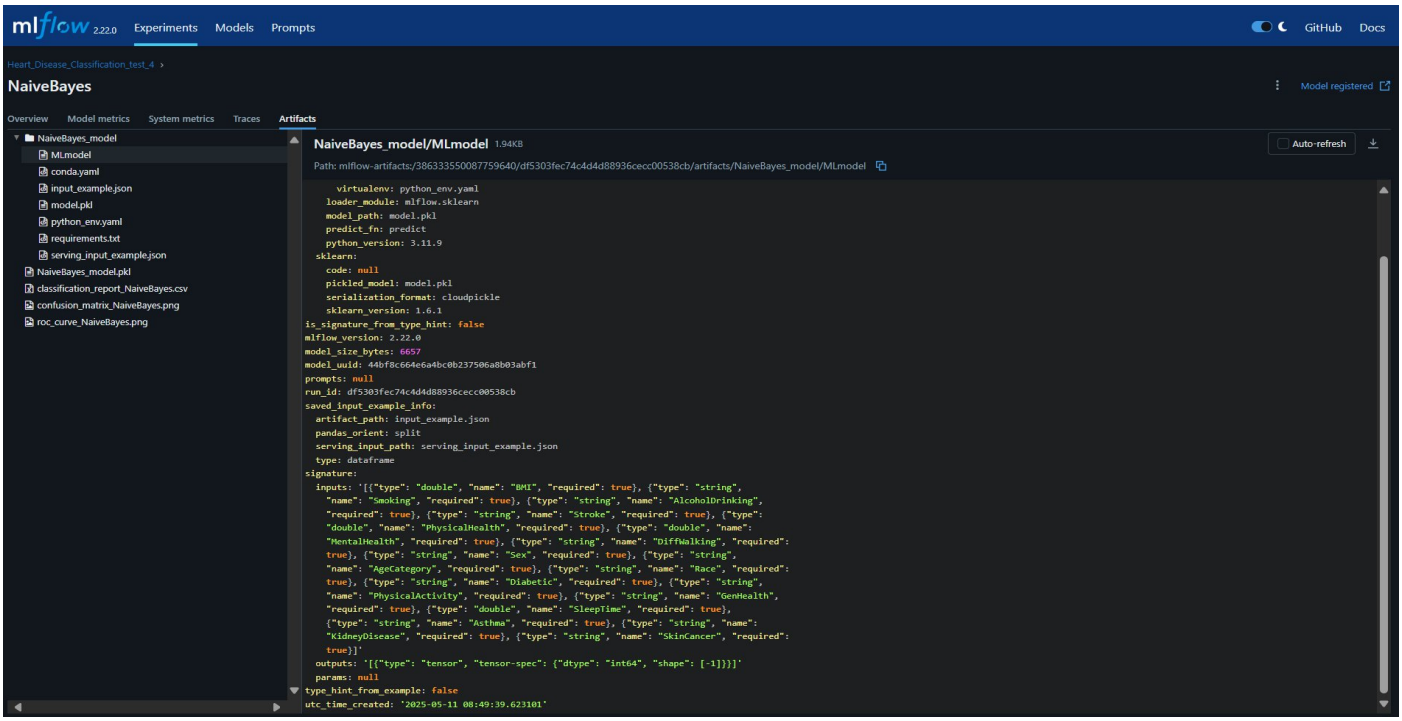



mlflow 2.22.0
Experiments
Models
Prompts
GitHub
Docs
Heart_Disease_Classification_test_4
NaiveBayes
Model registered
Overview
Model metrics
System metrics
Traces
Artifacts
NaiveBayes_model
MLmodel
conda.yaml
input_example.json
model.pkl
python_env.yaml
requirements.txt
serving_input_example.json
NaiveBayes_model.pkl
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