|  |  |  |  |
| --- | --- | --- | --- |
|  | Politechnika Bydgoska im. J. J. Śniadeckich  Wydział Telekomunikacji, Informatyki i Elektrotechniki  **Zakład Systemów Teleinformatycznych** | |  |
| **Przedmiot** | Wdrażanie Technik Uczenia Maszynowego | | |
| **Prowadzący** | mgr inż. Martyna Tarczewska | | |
| **Temat** | Klasyfikacja Obrazów – Pizza vs Niepizza | | |
| **Student** | Zuzanna Tarazewicz | **Nr Indeksu** | 116954 |
| **Ocena** |  | **Data Oddania** |  |

Spis Treści

**No table of contents entries found.**

# Specyfikacja Projektu

* Dane: <https://www.kaggle.com/datasets/carlosrunner/pizza-not-pizza>
* Repozytorium:
* Temat: Klasyfikacja Obrazów: Pizza vs Niepizza
* Chmura: Amazon Web Services

Celem projektu jest stworzenie modelu uczenia maszynowego klasyfikującego obrazy. Dany model będzie rozpoznawał czy na obrazie będzie znajdować się pizza, czy nie. Zbiór danych składa się z 787 obrazów pizzy w zbiorze treningowym, 98 obrazów pizzy w zbiorze walidacyjnym, 99 obrazów pizzy w zbiorze testowym, 786 zdjęć na których nie znajduje się pizza w zbiorze treningowym, 98 obrazów na których nie znajduje się pizza w zbiorze walidacyjnym I 99 obrazów na których nie znajduje się pizza w zbiorze testowym.

Kolejnym celem projektu jest stworzenie CI/CD projektu i wyszkolenie modelu na chmurze wykorzystując techniki MLOps.

Do wykonania projektu zostały wykorzystane:

* Python 3
* Amazon SageMakerLab
* Docker
* GitHub

# Zbiór Danych

## Pobranie Danych

W celu pobrania danych z kaggle.com wykorzystana została biblioteka opendatasets w której znajduje się funkcja download, która pobiera dane znajdujące się pod podanym linkiem. Po uruchomieniu programu należy podać login konta na kaggle i specjalnie wygenerowany klucz, który znajduje się w pliku .json po pobraniu.

Wygląda to w następujący sposób:

A black screen with red text

Description automatically generated

downloadData.py

from opendatasets import download

link = 'https://www.kaggle.com/datasets/carlosrunner/pizza-not-pizza'

download(link)

print('Download complete!')

## Podział danych

Zbiór znajdujący się na kaggle nie dzieli się na konkretne zbiory, a klasy. W celu podziału na zbiór treningowy, walidacyjny i testowy wykorzystano bibliotekę split-folder. W argumentach funkcji ratio pobierane są m.in ścieżka do zbioru, który ma zostać podzielony, ścieżka w której ma zostać zapisany i w jakich proporcjach mają zostać utworzone foldery. Jeśli ratio ma długość krotki równą dwa, to zbiór dzieli się na zbiór treningowy i testowy, a jeśli ma długość krotki równą trzy – to zbiór dzieli się na zbiór treningowy, testowy i walidacyjny.

split-folders.py

import splitfolders as s

s.ratio('pizza-not-pizza/pizza\_not\_pizza', output='pizza-notpizza', seed=1337, ratio=(.8, .1, .1), group\_prefix=None, move=True)

# Eksploracyjna Analiza Danych

# Tworzenie Modelu

# Tworzenie Frontendu aplikacji

# Uruchomienie programu przez Docker

A screenshot of a computer

Description automatically generated

# CI/CD

# Chmura

# Wnioski