\_Flower\_Match\_

Dokumentacja wstępna projektu

Justyna Mątewka

# Opis ogólny i wybór tematu:

Projekt "Flower\_Match" ma ambitny cel - rozpoznawanie różnych gatunków kwiatów za pomocą zaawansowanych technik uczenia maszynowego. Ten innowacyjny projekt ma na celu ułatwienie użytkownikom identyfikacji różnych gatunków kwiatów, które mogą napotkać w swoim otoczeniu, za pomocą prostej fotografii.

Rozpoznawanie kwiatów jest niezwykle interesującym i zarazem skomplikowanym tematem. Na całym świecie istnieje wiele różnych gatunków kwiatów, a ich różnorodność i złożoność sprawiają, że ich odróżnienie może być trudna nawet dla doświadczonych botaników.

# Wykorzystane technologie

Python 3.12 + biblioteki: pandas, numPy, sklearn, joblib, Scibit-Learn

Jupyter Notebook i JupyterLab

Anaconda3

Mercury

# Ogólny plan działania:

① Podstawowa detekcja, czy to kwiat i jaki jego rodzaj → naucz model:

→ Znalezienie bazy danych - [www.kaggle.com](http://www.kaggle.com/)

→ Obróbka danych w bazie

→ Wybór algorytmu

→ Tworzenie modelu i trening

→ Walidacja danych

→ Testy danych

② Stronka www, na której będzie można zamieszczać zdjęcie i otrzymywać wynik  
 analizy → Mercury? (<https://mljar.com/mercury>)

Wybór tej technologii, bo najwygodniej jest podpiąć dynamiczny notatniki Jupytera i uzyskać estetyczny wygląd projektu po stronie klienta bez konieczności instalowania wielu systemów

③ Podstrona dedykowana accuracy modelu

Zawiera archiwalne wykresy dla bazy treningowej, walidacyjnej, testowej i wykresy dynamiczne dla pojedynczych danych dla wizualizacji progresu

Trochę teorii:

1. Pierwszym krokiem jest import danych, które będą podstawą do budowania i treningu modelu.

2. Następnie przeprowadzamy proces czyszczenia bazy z wszelkich powtórzeń, niepełnych, mylnych, zduplikowanych inputów i outlierów. Ten krok ma na celu na wstępie poprawić dokładność działania modelu, eliminując szum i błędy z danych.

3. Kolejnym etapem jest podział danych na część treningową, walidacyjną i testową. Zwykle stosuje się proporcję 60% danych na trening, 20% na walidację i 20% na testy. Ten podział jest kluczowy dla utrzymania wiarygodności modelu i zapewnienia, że jest on dobrze nauczony, ale nie przeuczony.

4. Teraz tworzymy i trenujemy model z wykorzystaniem wybranego algorytmu, dostosowując go do naszego zestawu danych i problemu, który próbujemy rozwiązać.

5. Po utworzeniu i wytrenowaniu modelu, przystępujemy do tworzenia przewidywań. Pytamy model, czy dany obiekt to kot, czy nie. Na tym etapie, zazwyczaj jego dokładność może być daleka od idealnej.

6. Ostatnim etapem jest rozwój i poprawa - cofamy się do etapu tworzenia modelu, poprawiamy fragmenty, które nie działają tak, jak powinny, a następnie powtarzamy kolejne kroki do momentu, aż osiągniemy satysfakcjonującą dokładność naszego modelu. W tym procesie korzystamy z części walidacyjnej danych do sprawdzenia dokładności modelu i wprowadzania poprawek. Na sam koniec, po wszystkich poprawkach i optymalizacjach, używamy części testowej danych do ostatecznej oceny modelu.

# Przykładowe źródła:

* <https://youtu.be/7eh4d6sabA0>
* <https://youtu.be/i_LwzRVP7bg>