# **Dokumentace: Omega**

Josef Vošvrda C4a

## **1. Úvod**

Tento projekt slouží k predikci výsledků závodů MotoGP pomocí modelů strojového učení. Systém umožňuje predikovat, zda jezdec závod dokončí, jaký bude mít výsledný čas a na jakém místě pravděpodobně skončí.

## **2. Architektura systému**

Systém se skládá ze tří hlavních částí:

1. **Predikční jádro** (main.py): pracuje s modely strojového učení
2. **Uživatelské rozhraní** (UI.py): GUI pro snadné zadávání údajů
3. **Scraper** (scraper.py): sběr dat z webu motogp.com

## **3. Soubory projektu**

### **3.1 main.py**

* Načítá tři modely: ml\_dnf.dat, ml\_time.dat, ml\_pos.dat
* Funkce:
  + dnf(inputs): predikce nedokončení závodu (0 nebo 1)
  + time(inputs): predikovaný čas v cíli
  + prediction(inputs): predikce celkového umístění
* Vstupní formát: slovník s klíči rider\_number, air\_temp, track\_cond, humidity, season, race\_id
* Využívá numpy pole a knihovnu pickle k práci s modely

### **3.2 UI.py**

* Vytváří GUI pomocí tkinter
* Nabízí:
  + Výběr jezdce a závodu (přes Combobox)
  + Zadání teploty, vlhkosti, stavu trati, sezóny
  + Tlačítko pro spuštění predikce
  + Zobrazení výsledků predikce (DNF, čas, pozice)
* Mapa jezdců a závodů převedena na číselné ID pro vstup do modelů

### **3.3 scraper.py**

* Používá selenium a chromedriver k získání dat ze stránky MotoGP
* Funkce:
  + scrape\_motogp\_classification(url): z webu získá výsledky závodu a údaje o počasí
  + main(race): iteruje přes sezóny 2021–2024, ukládá data do CSV
* Vrací:
  + results\_list: list jezdců s pozicí, body, číslem, jménem, časem
  + weather\_dict: slovník s počasím (teplota, vlhkost, stav trati, teplota země)

## **4. Modely strojového učení**

Modely byly natrénování na historických datech v [Google Collab](https://colab.research.google.com/drive/1pyhs581kTKYhssUuvZmTOXS9Gq0mXEWy). Uloženy ve formátu .dat pomocí pickle:

* ml\_dnf.dat: klasifikační model (pravděpodobnost nedokončení závodu)
* ml\_time.dat: regrese (předpověď času v cíli)
* ml\_pos.dat: regrese (předpověď výsledné pozice)

## **5. Uživatelské rozhraní**

GUI vytvořené v tkinter:

* Předvyplněné seznamy jezdců a závodních tratí
* Možnost zadání environmentálních proměnných
* Výstupy formátovány srozumitelně pro uživatele
* Ošetření chyb pomocí messagebox.showerror

## **6. Web scraping a datová příprava**

* Používá se selenium s headless režimem pro načtení stránky
* Extrahují se pouze relevantní data:
  + Z tabulky výsledků závodu
  + Z panelu s počasím
* Data se ukládají do CSV, které lze použít pro trénink nových modelů
* Dále jsem to celé zkompletoval do jednoho csv a dále s tím pracoval v google collab kde jsem data upravoval a čistil. [uprava](https://colab.research.google.com/drive/1g7KFnSSe10W1YAQrmD0YMxecJEn4s4GL)

## **7. Spuštění projektu**

**Instalace requirements**

Pomocí: pip install -r requirements.txt

### **Spuštění GUI**

python UI.py

### **Spuštění scraperu**

python scraper.py

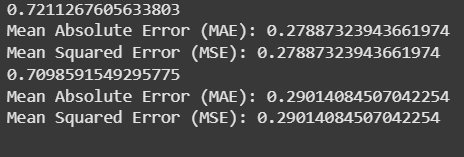
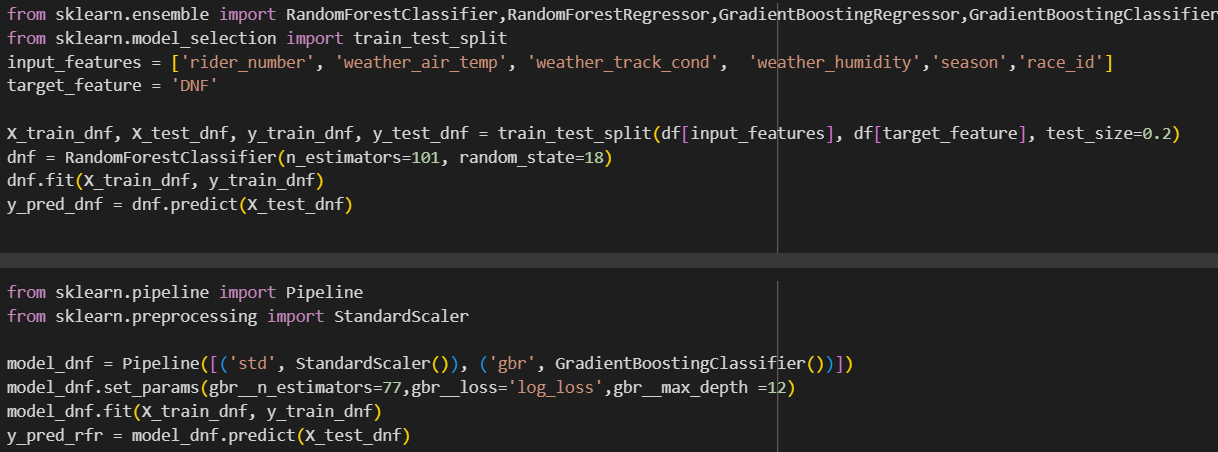
Výsledky budou uloženy do motogp\_all\_results\_\_<zkratka\_závodu>.csv

## **8. Závislosti a instalace**

pip install numpy scikit-learn selenium

* tkinter je součástí standardní instalace Pythonu
* Pro scraper.py je nutné mít nainstalovaný chromedriver

**9.Modely ML**

**Zde jsou chyby modelu na predikci dnf:  
  
A jejich kod:**

A takto jsem pokračoval s dalšími.