ORIS data analyser

Ondřej Měšťan

ČVUT FIT

mestaond@fit.cvut.cz

2022-01-02

1 Úvod

Tento report shrnuje semestrální práci z předmětu BI-PYT v zimním semestru B211.

Semestrální práce se zabývala zpracováním dat z informačního systému Českého svazu orientačních sportů ORIS, konkrétně na výsledky závodů a závodníků.

Cílem bylo vytvořit webový nástroj, který na základě informací od uživatele stáhne data, ty dle parametrů zpracuje a poté je zobrazí v přehledné podobě uživateli. V zadání byly specifikovány dva okruhy dat ke zpracování – mezičasy ze závodů a výsledky registrovaných závodníků za určené časové období. Tato data mělo být možné vyexportovat ven do souboru vhodného formátu.

2 Použité postupy

Zde krátce představím postupy a myšlenky, které jsem při tvorbě semestrální práce používal. Rozdělím je dle dvou funkcionalit a na ty použité v obou částech.

2.1 Obecné

Jako základ semestrální práce jsem se rozhodl použít knihovnu Streamlit a její jednoduchý způsob vizualizace dat hlavně ze dvou důvodů. Prvním bylo, že knihovna poskytuje přívětivou webovou tvář nástroje jak pro uživatele, tak pro programátora, a druhým chytré funkce, které knihovna nabízí, například cache dat.

Pro získávání dat jsem původně uvažoval o web-scrapingu, ale nakonec jsem se přiklonil k metodám, které poskytuje API systému ORIS[1].

První výhodou je, že aspoň z mého úhlu pohledu je elegantnější řešení tyto metody použít, když jsou k dispozici, a hlavně je snazší celkové použití, protože všechny metody vrací přímo data v JSON, která lze pak jednoduše zpracovat.

Pro vnitřní reprezentaci dat jsem použil převážně tabulky z knihovny Pandas, ke grafickému zobrazení dat pak knihovnu Matplotlib s rozšířením Kaleido pro export grafů do obrázků. Toto rozšíření je bohužel poměrně velké, ale nepodařilo se mi najít jiný rozumný způsob, jak převést grafy do obrázkové podoby.

Aby bylo možné aplikaci spouštět přímo jako *python xxx.py*, inspiroval jsem se kódem se StackOverflow [1].

2.2 Analýza mezičasů

Část s analýzou mezičasů by nejspíš pro nějaké praktické užití byla užitečnější, takže jsem ji rozšířil i o export do PDF. K vytvoření jsem použil knihovnu FPDF a k vygenerování odkazu pro stažení jsem se inspiroval na fóru Streamlitu [2].

Při generování PDF jsem řešil problém s českými znaky, neboť všechny základní fonty z FPDF používají kódování latin-1 (západoevropské). Přidal jsem si tedy z oficiálních stránek DejaVu Fonts[2] stažené fonty s volnou licencí kódované v Unicode.

Pro vyhledání požadovaného závodu má uživatel k dispozici kalendář závodů v sezóně, který je možné filtrovat a vyhledávat v něm pomocí možností v sekci Advanced Filtering.

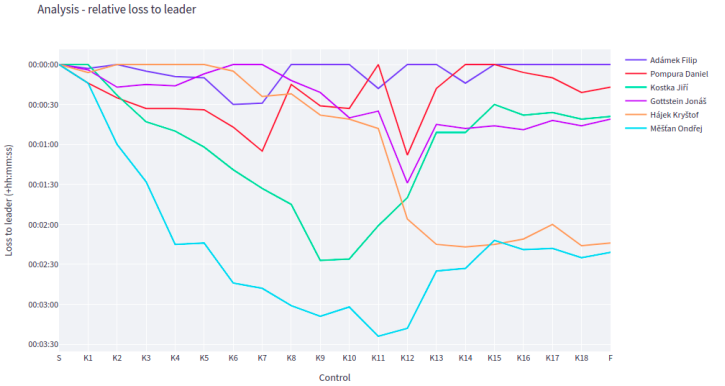
[1] https://oris.orientacnisporty.cz/API

[2] https://dejavu-fonts.github.io

Samotná stránka s analýzou uživateli po vyplnění ID závodu a jména kategorie zobrazí stránka s grafem, tabulkami s mezičasy a poli pro filtrovaní podle počtu nebo konkrétních závodníků (ti jsou poté v tabulce podbarveni [3]).

Obrázek 1: Analýza mezičasů – horní část stránky

Pro hezčí zobrazení lze graf přepnout do relativního režimu porovnání, aby zobrazoval ztráty na průběžně vedoucího závodníka, po najetí kurzorem se zobrazí informace o všech závodnících na dané kontrole.



Obrázek 2: Relativní porovnání závodníků dle ztráty

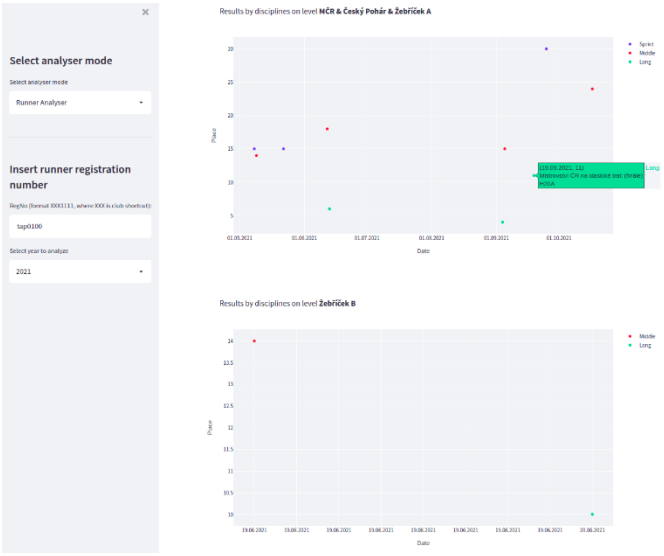
2.3 Analýza výsledků závodníka

Tato část byla lehce problematická, protože API ORISu bohužel neobsahuje žádnou metodu, kterou by bylo možné jednoduše získat výsledky závodníka, ale pouze seznam závodů, kam se přihlásil, což způsobilo dva problémy.

Prvním bylo, že jsem poté ve výsledcích každého závodu musel najít umístění závodníka, což znamená pro každý závod jeden request navíc a celkové zpomalení aplikace (zhruba na 1 vteřinu za každý načítaný závod). Problém jsem částečně vyřešil co největší optimalizací počtu requestů a zafixováním rozmezí závodů na jednu sezónu, aby načítání netrvalo tak dlouho.

Druhým byla dvoukolová mistrovství, protože závodník se vždy přihlašuje pouze na první den, a ještě do obecné kategorie (například H21), ale oba dny pak běží nějakou podkategorii (například H21B), v kvalifikaci losovanou a ve finále dle výsledků kvalifikace. Protože se jedná o maximálně dva závody ročně, problém jsem vyřešil zobrazováním pouze výsledků z finále, kde procházím všechny podkategorie a hledám závodníka ve výsledcích (podkategorie procházím v pořadí A-D, což znamená maximálně 8 requestů navíc).

Při zobrazení jsem se soustředil na co nejlepší předání možných informací – rozdělil jsem výsledky podle úrovní závodů (celostátní, žebříček B, oblastní a etapové) a v každém grafu pak dle individuálních disciplín (klasická trať, krátká trať, sprint). Nezpracovávám týmové (nelze porovnávat) a atypické disciplíny (z výsledků by nebylo možné vyčíst relevantní informace). Po najetí kurzorem na bod se zobrazí podrobnosti o daném závodě.



Obrázek 3: Analýza závodníka – prostřední část stránky

3 Shrnutí práce

V následujících odstavcích se pokusím shrnout výsledky semestrální práce

3.1 Dosažené výsledky

Vytvořil jsem nástroj k uživatelsky přívětivé vizualizaci dat, který splňuje zadání a moje představy – ukáže jiný pohled na výsledky než jen v textové nebo tabulkové podobě a umí poskytnout závodníkům zpětnou vazbu k jejich výkonům.

3.2 Možnosti ke zlepšení

Zde se nepochybně nabízí již zmiňovaná problematika s rychlostí načítání výsledků jednotlivých závodníků. K řešení by bylo nejspíš potřeba k celému problému přistoupit jinak (možná přes scrapování), nicméně tato funkce není určená pro rozbory, spíše pro zajímavost.

K analýze výsledků závodníka by také bylo možné doplnit možnost exportu do PDF. Tuto funkcionalitu jsem ale vynechal, protože se ztratí interaktivita grafů a ze samotných barevných teček toho nelze moc vyčíst.

3.3 Testování

Semestrální práci jsem otestoval nejvyšším možným způsobem za pomoci vlastních testů i Pytestu. Bohužel spousta částí (práce s grafy, tvoření pdf apod.) se nejsnáze testuje uživatelsky a vizuálně (stylem „vím, co se má zobrazit a jak to má vypadat“), leckde jsem ani nevěděl, jak by bylo možné test napsat (jak otestovat, zda graf obsahuje, co má), proto není kód z části pokrytý testy.

3.4 Další využití práce

Moje práce by si zasloužila zveřejnění někde na webu, aby vytvořených funkcionalit mohli využít i ostatní závodníci z komunity orientačních běžců, a i díky nim se zlepšovat ve svých výkonech. Už nyní jsem se snažil práci zprovoznit na serveru PythonAnywhere (zatím neúspěšně, musím si požádat o odblokování přístupu k API ORISu).

3.5 Můj komentář

Práci jsem dlouho odkládal, ale když už jsem začal, tak se z ní záhy stal můj asi za celé studium nejoblíbenější projekt (hlavně díky blízkému tématu a praktičnosti práce), celý proces tvorby jsem si užíval a s výsledkem jsem spokojený.

Reference – zdroje kódu

[1] Tvorba odkazu ke stažení pdf
https://discuss.streamlit.io/t/creating-a-pdf-file-generator/7613/2

[2] Spuštění Streamlitu přes Python
https://stackoverflow.com/questions/62760929/how-can-i-run-a-streamlit-app-from-within-a-python-script

[3] Barvení řádek v Pandas.DataFrame
https://stackoverflow.com/questions/31129076/python-pandas-highlight-row-in-dataframe

Dokumentace

Pandas:

https://pandas.pydata.org/docs/reference/index.html

Streamlit:

https://docs.streamlit.io/

FPDF:

https://pyfpdf.readthedocs.io/en/latest/