Uživatelská příručka A

Celý projekt je veřejně dostupný na platformě **GitHub** na adrese `https://github.com/SpeekeR99/DP_2024_2025_Zappe`

> Smlouva o poskytnutí dat pro akademické účely od **Deutsche Börse AG** ze dne 14. 11. 2022 nedovoluje zveřejnění vstupních dat (*Non-Disclosure Agreement*). Případná aktualizace podmínek by byla zveřejněna v repozitáři v adresáři `data`.

Většina uvedených výpisů v této kapitole je pro operační systém **Linux**. Pro jiné operační systémy (**Windows**, **macOS**) je však postup analogický.

Předpoklady

Pro správné spuštění programu je zapotřebí mít nainstalovaný:

- **Python 3** (doporučená verze **Python 3.11**)
- Uživatelem preferovaný ***správce balíčků*** (doporučen je standardní **pip**)

Interpret jazyka **Python** si lze stáhnout z oficiálních stránek `https://www.python.org`. Standardní *správce balíčků* **pip** je nedílnou součástí instalace.

Naklonování repozitáře

Jelikož repozitář obsahuje tzv. *podmoduly*, je zapotřebí projekt správně naklonovat tzv. *rekurzivním klonováním*. Ukázka správného postupu je uvedena ve Výpisu A.1.

Výpis A.1: Ukázkový výpis při klonování repozitáře

```
uzivatel@pocitac:~$ git clone --recursive
    https://github.com/SpeekeR99/DP_2024_2025_Zappe.git
... (výpis průběhu klonování)
uzivatel@pocitac:~$
```

Vytvoření a aktivace virtuálního prostředí

Doporučuje se vytvořit si pro projekt samostatné virtuální prostředí, ve kterém budou instalovány všechny závislosti. Tím se předejde možným konfliktům s jinými projekty či globálně nainstalovanými balíčky.

Ukázka vytvoření a aktivace virtuálního prostředí pomocí standardního modulu `venv` je uvedena ve Výpisech A.2 a A.3.

Výpis A.2: Ukázkový výpis tvorby a aktivace virtuálního prostředí (**Windows**)

```
C:\Users\Uzivatel\DP_2024_2025_Zappe>python -m venv venv

C:\Users\Uzivatel\DP_2024_2025_Zappe>call venv/Scripts/activate.bat

(venv) C:\Users\Uzivatel\DP_2024_2025_Zappe>
```

Výpis A.3: Ukázkový výpis tvorby a aktivace virtuálního prostředí (**Linux**)

```
uzivatel@pocitac:~/DP_2024_2025_Zappe$ python3 -m venv venv
uzivatel@pocitac:~/DP_2024_2025_Zappe$ source venv/bin/activate
(venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$
```

Linux: aktivací virtuálního prostředí je umožněno používání příkazů `python` a `pip` namísto `python3` a `pip3`.

Instalace závislostí

Po úspěšném naklonování se v *kořenovém adresáři* projektu nachází soubor `requirements.txt`, který obsahuje potřebné knihovny. Jejich instalaci lze provést příkazem uvedeným ve Výpisu A.4.

Výpis A.4: Ukázkový výpis instalace závislostí

```
(venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$ pip install -r
    requirements.txt
... (výpis průběhu instalace závislostí)
(venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$
```

Spuštění

Jelikož je aplikace napsána v interpretovaném jazyce **Python**, není nutné program nejdříve kompilovat.

Veškeré skripty předpokládají spuštění z *kořenového adresáře* projektu! Při spuštění z jiného adresáře může dojít k chybám kvůli relativním cestám.

Stažení dat

Nejprve je nutné získat vstupní data – v rámci této práce byl využit proprietární nástroj **A7**. Skript `/src/A7/download_eobi.py` slouží ke stažení požadovaného souboru ve formátu *JSON* a očekává čtyři parametry v následujícím pořadí:

1. ***Market ID*** – např. `XEUR`
2. ***Datum*** ve formátu `YYYYMMDD` – např. `20191202`
3. ***Market Segment ID*** – např. `688`
4. ***Security ID*** – např. `4128839`

Ukázka úspěšného spuštění je uvedena ve Výpisu A.5. Po úspěšném dokončení by se měl v adresáři `/data` objevit stažený soubor.

Pro korektní fungování skriptů komunikujících s A7 API je nutné upravit zdrojový kód na řádcích 25 a 26, kde se nachází uživatelské ID a API klíč. Alternativně lze tyto údaje uložit do souboru `a7token.txt` v kořenovém adresáři projektu.

Výpis A.5: Ukázkový výpis spuštění stažení dat

```
(venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$ python
    src/A7/download_eobi.py XEUR 20191202 688 4128839
... (výpis průběhu stahování)
(venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$
```

Předzpracování dat

Po úspěšném stažení vstupního *JSON* souboru je nutné provést jeho převod a případné rozšíření dat. K tomu slouží dva skripty umístěné ve složce `/src/data_preprocess`:

- `json-detailed2lobster.py` – převádí vstupní *JSON* soubor surových zpráv do formátu *LOBSTER*.
- `augment_lobster.py` – provádí doplnění a rozšíření dat – výpočet dodatečných metrik.

Oba skripty očekávají čtyři základní parametry ve stejném pořadí jako u předchozího kroku:

1. ***Market ID*** — např. `XEUR`
2. ***Datum*** ve formátu `YYYYMMDD` — např. `20191202`
3. ***Market Segment ID*** — např. `688`
4. ***Security ID*** — např. `4128839`

Spuštění těchto skriptů je obdobné jako v předchozím Výpisu A.5. Výsledné soubory budou uloženy do složky `/data`, kde slouží jako vstup pro další fázi zpracování – trénování modelů.

Spuštění trénování modelů

Pro trénování detekčních modelů slouží dva hlavní skripty umístěné ve složce `/src/anomaly_detection/models`:

- `autoencoder.py` – trénuje varianty *autoenkodérů* (plně propojený, konvoluční, transformer).
- `if_ocsvm_lof.py` — spouští modely typu *izolační les*, *jednotřídní SVM* a *lokální faktor odlehlosti*.

Spuštění skriptů vyžaduje několik parametrů, které určují jednak konkrétní datový soubor, jednak konfiguraci samotného modelu.

Mezi společné parametry patří:

- `--market_id` — např. `XEUR`
- `--date` — datum ve formátu `YYYYMMDD`, např. `20191202`
- `--market_segment_id` — např. `688`
- `--security_id` — např. `4128839`

Dále se parametry liší podle typu modelu.

Příklad parametrizace pro autoenkodér.

- `--model_type` – typ modelu: `ffnn`, `cnn` nebo `transformer`
- `--epochs`, `--kfolds`, `--batch_size`, `--lr` – běžné trénovací parametry
- `--seq_len`, `--latent_dim` – specifické pro sekvenční modely
- `--seed` – pro zajištění reprodukovatelnosti

Příklad parametrizace pro IF, OCSVM a LOF.

- `--model_type` – typ modelu: `if`, `ocsvm` nebo `lof`
- `--kfolds` – počet iterací pro křížovou validaci
- Parametry specifické pro jednotlivé modely:
  - IF: `--n_estimators`, `--max_samples`, `--max_features`
  - OCSVM: `--gamma`
  - LOF: `--n_neighbors`
- `--seed` – pro zajištění reprodukovatelnosti

Výsledky experimentů se automaticky ukládají do složky `/res`, zatímco natrénované modely jsou serializovány a ukládány do složky `/models` ve formátu `.pckl`.

Ukázkové spuštění obou skriptů může vypadat např. jako ve Výpisech A.6 a A.7.

Výpis A.6: Ukázkové spuštění modelu *CNN Autoencoder*

```
(venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$ python
    src/anomaly_detection/models/autoencoder.py --market_id XEUR --date
    20191202 --market_segment_id 688 --security_id 4128839 --model_type
    cnn --epochs 500 --kfolds 5 --batch_size 32 --lr 1e-3 --seq_len 64 --
    latent_dim 4
... (výpis průběhu trénování)
(venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$
```

Výpis A.7: Ukázkové spuštění modelu *Isolation Forest*

```
(venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$ python
    src/anomaly_detection/models/if_ocsvm_lof.py --market_id XEUR --date
    20191202 --market_segment_id 688 --security_id 4128839 --model_type
    if --kfolds 5 --n_estimators 100 --max_samples auto --max_features
    1.0 --seed 42
... (výpis průběhu trénování)
(venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$
```

Soubor modelů a vizualizace

Kombinaci výsledků všech modelů spolu s jejich vizualizací zajišťuje skript `/src/anomaly_detection/models/ensemble.py`. Tento skript umožňuje načíst uložené výsledky z předchozího kroku, dále je kombinovat a generovat vizuální výstupy pro snadnější interpretaci chování jednotlivých modelů i celého *souboru modelů*.

Skript akceptuje široké spektrum parametrů, které zahrnují jak identifikaci konkrétního datového souboru, tak konfigurace jednotlivých modelů. Většina parametrů odpovídá těm, které byly použity při trénování modelů. Nově zde přibývají následující parametry:

- `--no_if`
- `--no_ocsvm`
- `--no_lof`
- `--no_ffnn`
- `--no_cnn`
- `--no_transformer`

Parametry s prefixem `no_` slouží k deaktivaci daného modelu. Díky tomu je možné snadno testovat různé kombinace metod bez nutnosti úprav zdrojového kódu.

Skript načítá vstupní data ze složky `/data` a mezivýsledky z trénování ze složky `/res`. Výstupní vizualizace jsou automaticky ukládány do příslušných podadresářů složky `/img`, např. `/img/anomaly_detection`, `/img/eval` apod. Příklad spuštění `ensemble.py` na výše popsaných natrénovaných modelech vypadá dle vzoru Výpisu A.8.

Výpis A.8: Ukázkové spuštění souboru modelů

```
(venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$ python
    src/anomaly_detection/models/ensemble.py --market_id XEUR --date
    20210319 --market_segment_id 688 --security_id 5578483 --epochs 500 --
    kfolds 5 --if_n_estimators 100 --if_max_samples 0.1 --
    if_max_features 0.5 --no_ocsvm true --no_lof true --cnn_batch_size
    32 --cnn_lr 1e-3 --cnn_seq_len 64 --no_ffnn true --no_transformer
    true
(venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$
```