Klasifikační model pro predikci

Tento program obsahuje dva skripty, které slouží k trénování a použití klasifikačního modelu pro predikci Segmentace na základě dat uložených v csv souboru. První skript natrénuje a uloží klasifikační model pomocí Train.csv datového souboru. Druhý skript následně načte natrénovaný model a provede inferenci na csv souboru zadaném v parametru příkazové řádky (viz sys.argv). Pokud csv soubor obsahuje sloupec "Segmentation", program vypíše na standardní výstup přesnost predikce ve formátu "accuracy: {počet_správných} / {počet_celkem}".

Cílem tohoto programu je umožnit trénování a použití klasifikačního modelu pro predikci Segmentace na základě csv souborů. Dále program umožňuje vypsat přesnost predikce na standardní výstup pro csv soubory obsahující sloupec "Segmentation".

Instalace do virtuálního prostředí

```
pip install -r requirements.txt
```

Hlavní použité knihovny

- Pandas
- Seaborn
- Matplotlib
- Scipy
- Scikit-learn

Popis souborů

- **knn.ipynb** obsahuje prototypový kód, kde byla otestována funčnost, obsahuje kód pro načtení dat, předzpracování a trénování modelu. Také obsahuje grafy, které budou uvedeny v dalším textu
- **train.py** obsahuje část kódu ze souboru **knn.ipynb**. Tato část slouží pro natrénování dat, zobrazení výsledku trénování a uložení modelu.
- **load.py** slouží pro validaci nad uloženým, natrénovaným modelem. Pokud je v datech sloupec Segmentaion, pak se zobrazí pouze "accuracy". Jinak je vytvořen soubor s predikcí dat a následné je uložen do složky "results".

Vstupní data

Vstupními daty jsou soubory s daty pro trénování modelu. Tyto soubory musí být ve formátu CSV.

Výstupní data

Výstupem programu je natrénovaný model, soubor s daty a jejich predikcí nebo standartní výstup v terminálu.

Příklad použití

```
python train.py data/Train.csv

python load.py data/Train.csv models/20230422_2035_kkn_model.joblib
```

Metoda KNN

KNN (k-Nearest Neighbors) je jedna z nejjednodušších a nejčastěji používaných metod pro klasifikaci a regresi v oblasti strojového učení. V balíku scikit-learn (sklearn) je tato metoda implementována v modulu neighbors, kde lze nalézt třídu KNeighborsClassifier pro klasifikaci a třídu KNeighborsRegressor pro regresi.

KNN je založen na jednoduché myšlence, že podobné příklady mají tendenci být kategorizovány stejně. Tedy pokud máme bod v prostoru, tak mu přiřadíme třídu podle tříd jeho nejbližších sousedů. KNN algoritmus vyžaduje, aby byla definována vzdálenostní funkce, pomocí které se určuje vzdálenost mezi body v prostoru.

Pro úspěšné použití KNN algoritmu je důležité vhodně zvolit parametr k, což je počet nejbližších sousedů, kteří mají být bráni v potaz při klasifikaci či regresi. Tento parametr lze určit pomocí křivky cross-validace, kde se zvolí hodnota k s nejlepším výsledkem.

KNN je velmi jednoduchý a intuitivní algoritmus, ale má několik nevýhod, jako je například náchylnost k šumu v datech a velké nároky na výpočetní výkon při velkém množství trénovacích dat. Více o metodě

Dalšími metodami použitými v projektu jsou metoda SVC (Support Vector Classification) z balíku scikit-learn a neuronová síť. Metoda SVC je založena na vytvoření nadrovin, které rozdělují data do různých tříd. Výhodou metody je, že dokáže pracovat s velkým množstvím dat a s více než dvěma třídami. Neuronová síť je další často používanou metodou strojového učení. Jedná se o soubor propojených neuronů, které se učí na datech a umí odhadovat neznámé vstupy. V balíku Keras lze vytvořit různé typy neuronových sítí, jako jsou například plně propojené sítě, konvoluční sítě nebo rekurentní sítě.

Úprava dat

Před samotným trénováním modelu bylo nutné upravit data. Nejčastěji se data upravují především normalizací či standardizací dat, aby byla zajištěna stejná váha každého z atributů. Dále byla data rozdělena do testovacích, trénovacích, aby bylo možné ověřit úspěšnost natrénovaného modelu na neviděných datech a předejít tak přetrénování. V projektu byl použit poměr 80:20 pro trénovací a testovací data.

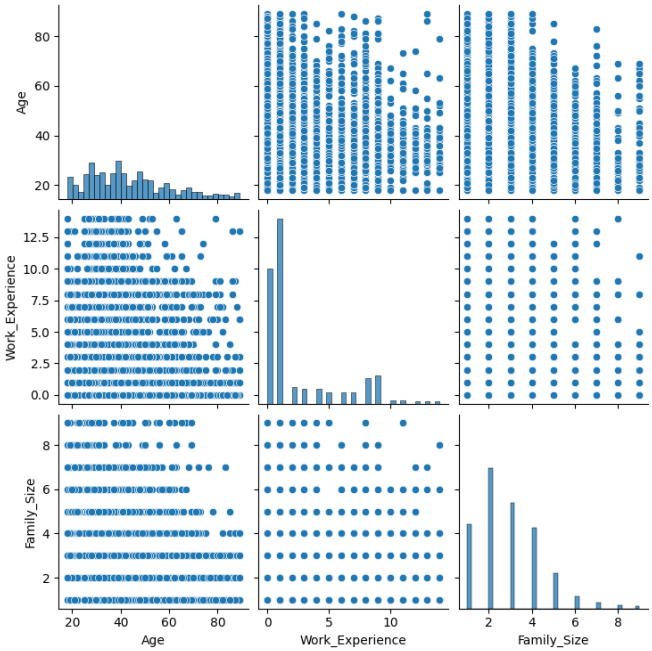
Byla použita metoda PCA (Principal Component Analysis) z analýzy dat před trénováním, která umožňuje redukci dimenzionality dat a získání nových příznaků, které lépe popisují původní data. Tento postup může vést k lepší výkonnosti modelu a efektivnějšímu využití zdrojů.

Byly také detekovány a odstraněny odlehlé hodnoty, aby neovlivnily výsledky trénování a predikce modelu. Tato detekce byla provedena pomocí statistických metod, jako je například z-score, a odstraněny byly hodnoty s extrémními hodnotami z-score.

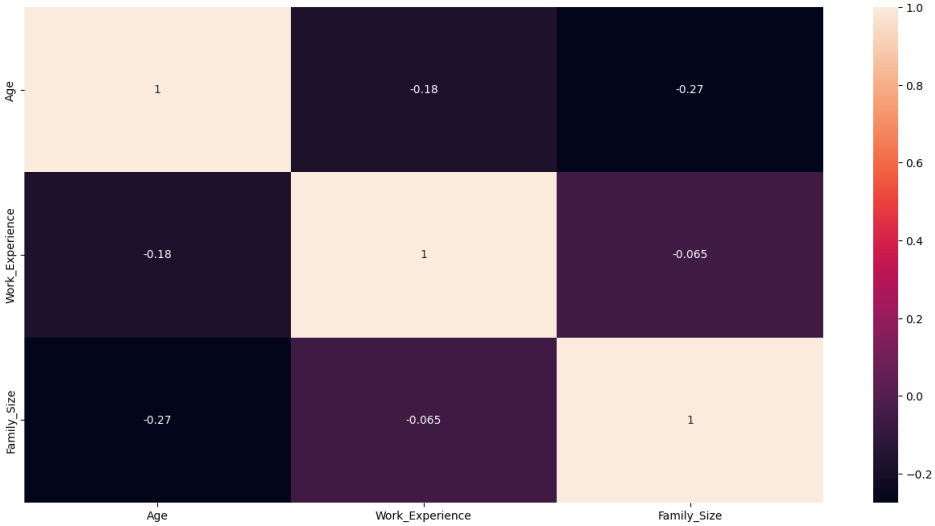
Výsledky

V následující sekci budou uvedeny výsledky projektu. Kromě popisu výsledků bude uvedeno také několik vizualizací, které nám pomohou lépe porozumět zjištěným výsledkům. Tyto vizualizace budou sloužit jako podpora.

Zjištění korelací

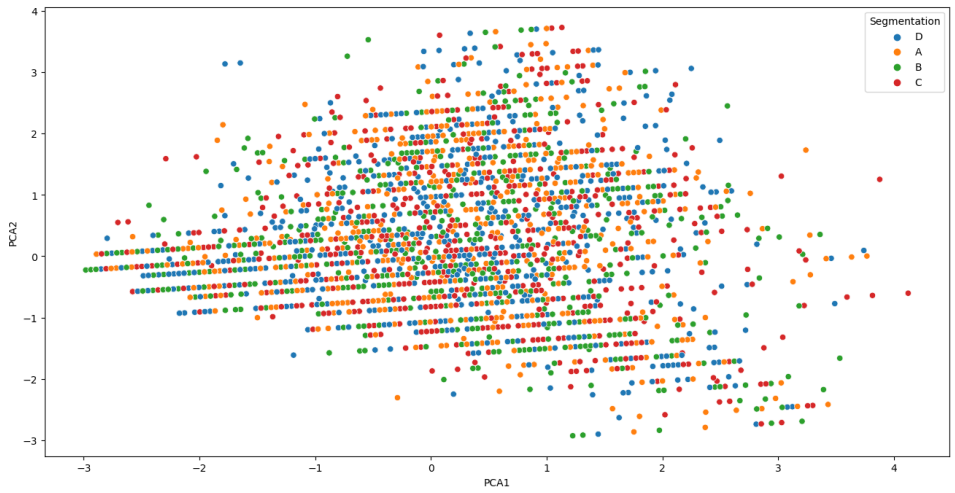


Obr. 1: Zjištění korelací, grafické zobrazení



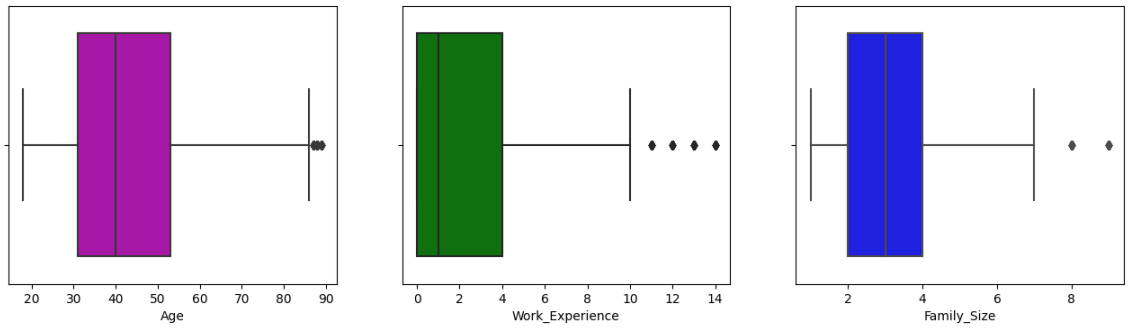
Obr. 2: Zjištění korelací, číselné zobrazení

Principal Component Analysis



Obr. 3: Principal Component Analysis

Detekce odlehlých hodnot



Obr. 4: Zobrazení odlehlých hodnot

Výsledky jednotlivých modelů

- KNN: 0.45
- SVC: 0.47
- NN: 0.48

Závěr

Bohužel, výsledky trénování modelů nebyly uspokojivé, a celková přesnost výsledků se pohybovala pod 0.5. Nejlépe si vedla neuronová síť s přesností 0.48, následovaná SVC s přesností 0.47 a nakonec KNN s přesností 0.45. Jedním z možných důvodů, proč se klasifikace nepodařila a nepovedlo se natrénovat model na vyšší přesnost, může být nedostatek relevantních dat pro trénování.

Při použití dat ke kontrole kvality (validaci) se přesnost ještě zhoršila a pohybovala se nad 0.33. Pro účely klasifikace se často používají data, která jsou dobře popsaná a strukturovaná, obsahují dostatečný počet datových bodů a jsou rovnoměrně rozložená ve všech kategoriích, aby se minimalizoval vliv náhodného výběru a šumů v datech.

Autor

Jan Zmrzlý, zmrzlyjan@gmail.com