

STROJOVÉ UČENIE

Type of crime prediction - Report

Jaroslav Ištók
mAIN

January 12, 2018

Contents

1	Abstrakt	2
2	Úvod do problematiky	2
3	Podobné práce a projekty	2
4	Dáta	3
5	Metódy a výsledky	4
6	Experimentálne vyhodnotenie	8
7	Technické detaily implementácie	9
8	Záver	10

1 Abstrakt

V projekte analyzujem záznamy kriminálnych činov spáchaných v americkom meste Chicago. Hlavnou úlohou je predpovedať typ trestného činu na základe rôznych atribútov, ako je miesto spáchania kriminálneho činu či jeho čas. Neskôr som pôvodný dataset neobsahoval dostatočné množstvo relevantných atribútov, preto som ho rozšíril o ďalšie dáta, aby som dostal viac potrebných atribútov a teda aj lepšie výsledky. Chicago rozlišuje až 29 typov trestných činov. V projekte som robil multiklasifikáciu na pôvodných 29 kategóriach kriminálnych činov, neskôr som kategórie rozdelil na násilné(violent) a nenásilné(non-violent) kriminálne činy a tým preformuloval problém na binárnu klasifikáciu. V ďalšej časti projektu som sa venoval najmä analýze dát a hľadanie zaujímavých skutočností, ktoré vyplývali zo získaných výsledkov. Nakoniec som napísal krátke porovnanie výkonnosti jednotlivých modelov na trénovacej a testovacej množine a zhrnul, čo som sa naučil.

2 Úvod do problematiky

S príchodom éry "veľkých dát" a efektívnych algoritmov na analýzu dát, sa stalo hľadanie rôznych vzorov v kriminálnych dátach veľmi populárnou oblasťou výskumu v oblasti strojového učenia. V mojom projekte sa na základe času, miesta a demografických údajov budem snažiť predpovedať o aký typ trestného činu pravdepodobne išlo. Hlavnou úlohou je vyskúšať si klasifikáciu na reálnych dátach, analyzovať výsledky a vyvodiť z nich závery. Vstupom sú dáta s rôznymi atribútmi a výstupom je typ kriminálneho činu, o aký pravdepodobne išlo, predpovedaný na základe týchto dát.

3 Podobné práce a projekty

Našiel som niekoľko projektov <https://github.com/RandomFractals/ChicagoCrimes/tree/master/notebooks> a <https://www.kaggle.com/currie32/crimes-in-chicago/kernels>, ktoré skúmajú dataset kriminálnych činov v meste Chicago. Prvý projekt sa zaoberá analýzou dát kriminálnych činov. V základe je to sada jupyter notebookov, v ktorých analyzujú a vizualizujú tieto dáta. Pred tým ako som začal robiť môj projekt, niektoré z nich som si stiahol a prezrel, aby som mal lepšiu predstavu o dátach s ktorými som sa chystal pracovať. V druhom projekte je zoznam rôznych prác, ktoré rôznym spôsobom analyzujú kriminálne dáta v Chicagu. Na začiatku som si niektoré z nich prezrel, aby som mal predstavu o tom, čo všetko sa dá s takýmito

dátami robiť a tiež aby som získal inšpiráciu.

4 Dáta

Základné dáta som získal z verejne prístupného datasetu <https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2/data>. Keďže atribútov bolo priveľa a mnohé mali veľmi nízku informačnú hodnotu, niektoré som odstránil. Po preskúmaní dát som vybral nasledujúce atribúty, ktoré sú najviac informatívne:

- **Primary Type** - typ kriminálneho činu
- **Arrest** - či bol páchateľ zatknutý
- **Domestic** - či bol spáchaný obyvateľom Chicaga
- **Beat** - časť mesta
- **Latitude** - súradnica približného miesta spáchania trestného činu
- **Longitude** - súradnica približného miesta spáchania trestného činu
- **Date and time** kedy sa kriminálny čin stal

Keďže som pri multiklasifikácii nedostával dobré výsledky, neskôr som dáta obohatil o census údaje mesta Chicago aby som získal viac relevantných atribútov, ktoré výsledky zlepšili. Census dáta som získal z verejne dostupného zdroja na stránke <https://catalog.data.gov/dataset/census-data-selected-socioeconomic-indicators-in-chicago-2008-2012-36e55>. Tieto dáta som napároval na základe atribútu **Community Area** na údaje kriminálnych činov. Atribúty, ktoré som pridal do datasetu po najoinovaní nových dát sú nasledovné:

- **Percent of housing crowded**
- **Percent households below poverty**
- **Percent aged 16+ unemployed**
- **Percent aged 25+ without high school diploma**
- **Percent aged under 18 or over 64**
- **Per capita income**

- **Hardship index**

Pred tým ako som mohol začať trénovať modely na klasifikáciu dát, potreboval som dáta najskôr spracovať do použiteľného formátu. Zdroj odkiaľ som čerpal dáta umožňuje stiahnuť vyfiltrované dáta z určitého obdobia v CSV formáte. Pre potreby môjho projektu som si stiahol dataset, ktorý obsahuje záznamy kriminálnych činov za posledný rok, čo je 60619 príkladov z ktorých však používam 10000. Je to dostatočné množstvo aby som dostal dobré výsledky a zároveň aby tréning netrvalo príliš dlho (najmä pri SVM multiklasifikácii). Dáta som rozdelil na tréningovú a testovaciu množinu v nasledujúcom pomere: 7000 tréningových príkladov (703000 testovacích príkladov (30Numerické atribúty som nemienil. Textové atribúty som transformoval na číselné hodnoty (z toho dôvodu, že niektoré modely nevedia pracovať s nečíselnými atribútmi, ako napríklad SVM). Atribút "dátum a čas" som rozdelil na niekoľko nových číselných atribútov, aby som z pôvodného atribútu dostal čo najviac informácií.

- month
- day
- hour
- dayofyear
- week
- weekofyear
- dayofweek
- weekday
- quarter

5 Metódy a výsledky

Na klasifikáciu som použil nasledujúce modely strojového učenia. Výsledky uvádzam v accuracy skóre a tiež cross validáciu na 10 splitov.

- **Support Vector Machine** Support vector machines patria medzi klasifikátory s najväčším odstupom. V mojom projekte som konkrétne použil kernelové SVM s RBF (radial basis function) kernelom a vyskúšal

som aj polynomiálny kernel. Nepoužil som lineárne SVM pretože moje dáta zjavne nie sú lineárne a teda bolo by nemožné získať dobré výsledky. Miernou nevýhodou SVM je vyššia časová zložitosť pri väčšom množstve tréningových príkladov (oproti iným metódam), najmä pri multiklasifikácii one-to-many to je vidieť.

Výsledky multiklasifikácie s RBF kernelom:

set	accuracy	cross validation
training set	0.529142857143	0.280158206977
test set	0.273	0.260996574649

Výsledky multiklasifikácie s polynomiálnym kernelom:

set	accuracy	cross validation
training set	0.310857142857	0.231576589874
test set	0.217333333333	0.260912594716

Výsledky binárnej klasifikácie s RBF kernelom:

set	accuracy	cross validation
training set	0.937857142857	0.921721125378
test set	0.922333333333	0.879343092701

Výsledky binárnej klasifikácie s polynomiálnym kernelom:

set	accuracy	cross validation
training set	0.922285714286	0.914289484264
test set	0.913	0.889671981541

- **Binary Tree Classifier** Binárne rozhodovacie stromy patria medzi veľmi výpočtovo efektívne klasifikátory. Pomocou nich som dokázal dosiahnuť pri multiklasifikácii na tréningovej množine veľmi dobré výsledky, avšak na testovacej množine bolo mizerné skóre, čo svedčí o veľkom overfitingu dát (pre rozhodovacie stromy typické). Pri binárnej klasifikácii som dosiahol takmer perfektné výsledky pri dostatočnom množstve tréningových dát aj na tréningovej aj testovacej množine.

Výsledky pri multiklasifikácii:

set	accuracy	cross validation
training set	0.997428571429	0.201167010467
test set	0.203666666667	0.184054750607

Výsledky binárnej klasifikácie:

set	accuracy	cross validation
training set	1.0	0.998571631196
test set	0.997666666667	0.999002214839

- **Random Forests** Náhodné lesy sú optimalizáciou rozhodovacích stromov. Namiesto vytvárania jedného stromu, spravíme viac verzií trérovacej množiny a natrénujeme simultánne viac stromov. Pri Random Forests som dostával najlepšie výsledky. Multiklasifikácia opäť trpela veľkým overfittingom dát, podobne ako pri jednom strome. Pri binárnej klasifikácii bol RandomForest konzistentný aj na trérovacej aj na testovacej množine, čo sú skvelé výsledky.

Výsledky multiklasifikácie:

set	accuracy	cross validation
training set	0.997428571429	0.334357381031
test set	0.331666666667	0.323377964103

Výsledky binárnej klasifikácie:

set	accuracy	cross validation
training set	1.0	0.999856938484
test set	0.999666666667	0.997667774086

- **Logistic regression** Logistická regresia je jedna zo základných klasifikačných metód, rozhodol som sa, že ju skúsím a porovnam jej výsledky s ostatnými metódami. Logistická regresia bola jedna z najhorších vyskúšaných metód. Pri multiklasifikácii boli zlé výsledky aj na trérovacej aj testovacej množine (čo značí underfitting dát), a pri binárnej klasifikácii mala horšie výsledky s porovnaním s ostatnými modelmi. Zjavne si logistická regresia nevie poradiť dobre s nelineárnymi dátami (potreboval by som zložitejšie hypotézy)

Výsledky multiklasifikácie:

set	accuracy	cross validation
training set	0.349857142857	0.347055569287
test set	0.347	0.339450907274

Výsledky binárnej klasifikácie:

set	accuracy	cross validation
training set	0.876857142857	0.87372433005
test set	0.873	0.867331966651

- **Multi layer perceptron** Multi layer perceptron je jednoduchá "feed forward" neurónová sieť s niekoľkými skrytými vrstvami. Chcel som vyskúšať, či komplexnejší model, akým je napríklad neurónová sieť, nebude schopný dosiahnuť lepšie výsledky. Pri tréňovaní som použil ReLu aktivačnú funkciu a 2 skryté vrstvy po 50 neurónov každá. Cross validačné skóre bolo nízke pri multiklasifikácii.

Výsledky multiklasifikácie:

set	accuracy	cross validation
training set	0.460142857143	0.30910230957
test set	0.296	0.292715926778

Výsledky binárnej klasifikácie:

set	accuracy	cross validation
training set	0.999857142857	0.99571407551
test set	0.996333333333	0.972668844469

Dôležitá poznámka: keďže samplujem určité množstvo dát (10000 príkladov) z pôvodného veľkého datasetu, môžu sa reálne výsledky trochu líšiť.

6 Experimentálne vyhodnotenie

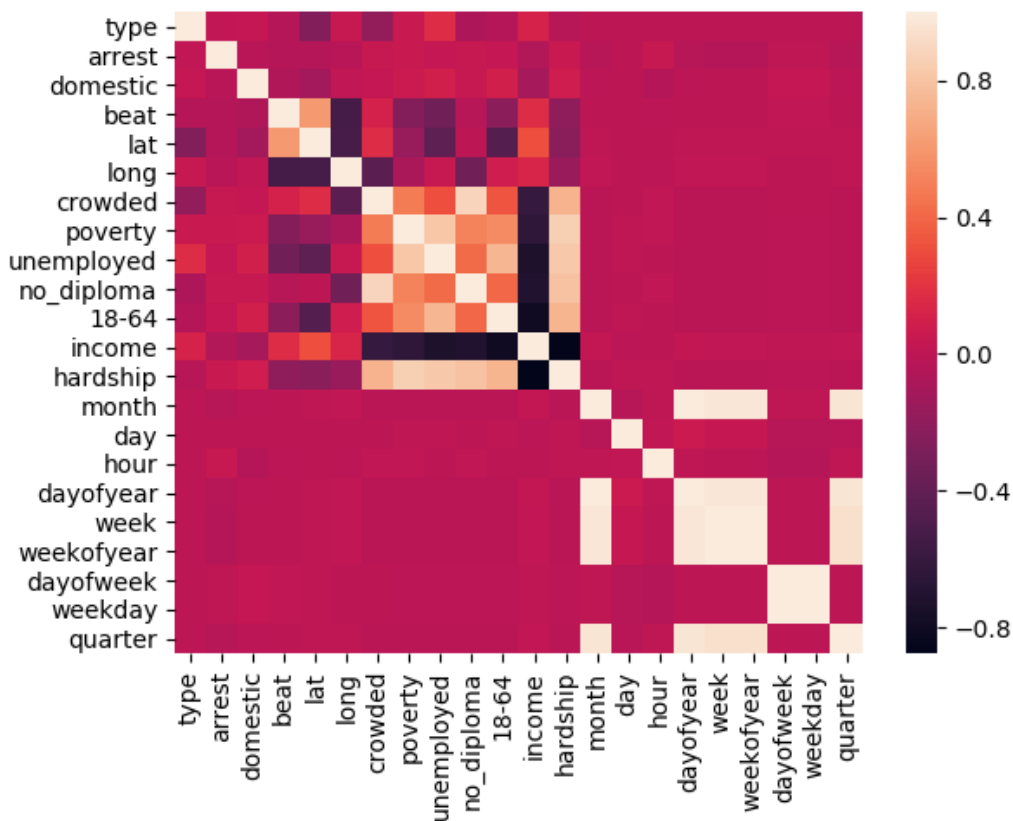


Figure 1: Attributes correlation heat map

Výsledky som experimentálne skúmal. Pri multiklasifikácii som dostával relatívne zlé výsledky pri všetkých vyskúšaných modeloch. Dôvody sú podľa mňa také, že tých kategórií je veľa, čiže by som potreboval asi hodne veľké množstvo trénovacích príkladov a pamäte (keď som skúšal väčšie datasety pri multiklasifikácii, Random Forests zabrali bezproblémov 4GB pamäte pri multiklasifikácii a SVM mali príliš veľkú časovú zložitosť - čo vyplýva z princípu ako pracuje OneVsRest multiklasifikácia). Nedokázal som sa dostať nad 0.4 accuracy skóre na testovacích množinách ani s jedným modelom. Skúšal som optimalizácie zmenou hyperparametrov jednotlivých modelov, ale tie veľmi nepomohli. Preto som sa rozhodol pre zmenu úlohy a zlúčením typov kriminálnych činov na 2 typy. Z korelačného grafu je vidno, že dáta ktoré som pridal do datasetu majú medzi sebou silné korelácie a korelácie

vytvárajú aj dátumovo časové údaje, čo pomohlo zlepšiť výsledky pri multi-klasifikácii. Ďalším problémom ktorý mohol byť dôsledkom zlých výsledkov, je že niektoré typy mali veľmi málo príkladov v porovnaní s ostatnými. Tento problém som neriešil, ale stačilo by získať viac príkladov pre typy ktoré mali málo zástupcov. Zaujímavým zistením je, že viac menej skoro všetky klasifikačné metódy boli schopné natrénovať pri binárnej klasifikácii (keď som zlúčil typy kriminálnych činov do dvoch hlavných: violent a non-violent) modely so 1.0*accuracy*. Podozrivo dobré výsledky pri binárnej klasifikácii ma viedli k tomu aby som sa viac zamyslel nad dátami a koreláciami jednotlivých atribútov. Vykreslil som si preto heatmapu korelácii jednotlivých atribútov ako je možno vidieť na Figure 1 obrázku. Skúsil som teda opačný extrém, koľko atribútov môžem odstrániť z datasetu pri binárnej klasifikácii, aby som stále dostal dobré výsledky?. Dá sa klasifikovať iba na základe nejakého konkrétneho atribútu? Odpoveď je, že **áno!!!** Napríklad Random-Forests dokážem na základe jediného atribútu(income) natrénovať klasifikátor so skóre 1.0*accuracy* na trénovacej aj testovacej množine!. Na heatmapu je vidno, že typy trestných činov (už zredukované na 2 hlavné) majú trochu vyššiu koreláciu s Income atribútom. Keď som podobnú vec skúsil s iným atribútom (domestic), ktorý je podľa heatmapy menej korelovaný s typom, tak som dostal 0.78*accuracy* na trénovacej aj testovacej množine, čo je o dosť horšie. Tiež je zaujímavé, že atribúty ktoré boli navyše nevytvárali zbytočný "noise" a aj s nimi boli výsledky rovnako dobré. Rovnako som skúšal aj cross validáciu, avšak priemery boli veľmi podobné accuracy skóre, preto ju neuvádzam.

7 Technické detaily implementácie

Na implementáciu projektu som použil jazyk Python, ktorý poznám celkom dobre. Na spracovanie dát som použil knižnicu **pandas**, na vykresľovanie grafov používam **matplotlib** a na prácu s maticami a samotnými modelmi využívam **sklearn** a **numpy** knižnice, ktoré poskytujú všetky potrebné nástroje. Pri implementácii som sa nestretol so žiadnymi veľkými problémami. Keď som sa naučil pracovať s danými knižnicami, tak samotné trénovanie algoritmov bolo relatívne jednoduché spraviť. Rovnako bolo rýchle aj vyskúšať a porovnať niekoľko rôznych modelov. Najviac mi dalo zabráť spracovanie, spájanie a úprava data setu do správnych dátových štruktúr. V základe mám triedu pomocou ktorej spracovávam dáta (DataWrangler) a potom triedu Classifier, ktorá vie trénovať klasifikátory a vypísať výsledky. Setup je v main.py súbore, kde sú volania jednotlivých metód. Trieda DataWrangler obsahuje aj statické metódy ktorými som si vykresľoval grafy a heatmapu.

Jazyk Python má jednu veľkú nevýhodu (aspoň pre mňa) a tou je GIL(global interpreter lock) interpretera, čiže nemožnosť využívať plný výkon procesora (v základe všetko beží iba na jednom jadre, čo značne spomaľuje tréning algoritmov, ktoré je možné paralelizovať, napríklad Random Forests). Na menšie projekty je dobrý, ale na väčšie by som siahol určite po nejakom kompilovateľnom jazyku, napríklad Java, ktorá mi príde "čistejšia" a prehľadnejšia. Celkovo však zastávam názor, že na programovacom jazyku nezáleží, tie algoritmy sú implementovateľné v každom a vždy je vhodné vybrať ten, ktorý sa na daný problém hodí najlepšie. **Zdrojové kódy projektu sú zverejnené na githube** <https://github.com/jaroslavistok/ChicagoCrimes>

8 Záver

Projekt bol pre mňa veľmi dobrou skúsenosťou a praktickou aplikáciou techník strojového učenia na reálnych dátach. Naučil ma najmä to, že na aplikáciu strojového učenia do praxe nestačí teoreticky poznať jednotlivé modely, ale je veľmi dôležité položiť si správnu otázku, čo s danými dátami viem dosiahnuť. Spracovanie a "vyčistenie" dát od šumu, starostlivá analýza, pohľad na korelácie atribútov, a vhodná transformácia sú prínajmenšom rovnako dôležité ako zvolenie vhodného modelu a optimalizácia hyperparametrov na dosiahnutie a natréningovanie modelu s dostatočne malou chybou na nových dátach. V mojom projekte som sa dopracoval k rôznym zaujímavým výsledkom a záverom, vyskúšal niekoľko rôznych modelov a naučil sa pochopiť niektoré dôležité veci pri aplikácii strojového učenia do praxe. Zistil som, že žiadny model nie je vyslovene zlý alebo dobrý, každý je vhodný na iný typ úloh. Môj projekt bol čiastočne úspešný a čiastočne neúspešný. Po zlúčení kategórií a modifikácii multiklasifikácie na binárnu klasifikáciu som bol schopný natrénovať konzistentný klasifikátor, čo sa dá pokladať za úspech. Ak by som podobný projekt išiel robiť znovu, asi by som si starostlivejšie vybral úlohu, inak by som postupoval podobným spôsobom. Mohol som vyskúšať viac experimentovať s hyperparametrami a rôznymi optimalizačnými metódami, pravdepodobne by som výsledky o niečo vylepšil, pravdepodobne však nie signifikantne.