22.05.2024

Výběr klasifikátorů v úlohách strojového učení

Filip Švadlenka, FEL KyR

Na základě zadání 14. úlohy zde zpracuji tři úkoly a zároveň provedu jejich rozbor. Veškeré kódy jsou k nalezení v odkazu v příloze a veškeré grafy byly kresleny s pomocí Python knihovny scikit-learn (též sklearn). Všechny programy předpokládají, že je uživatel otevře a doplní své hodnoty a názvy souborů. Nejedná se o aplikace se vstupem, pouze o výpočetní programy. Výsledky jsou však komentované a textové.

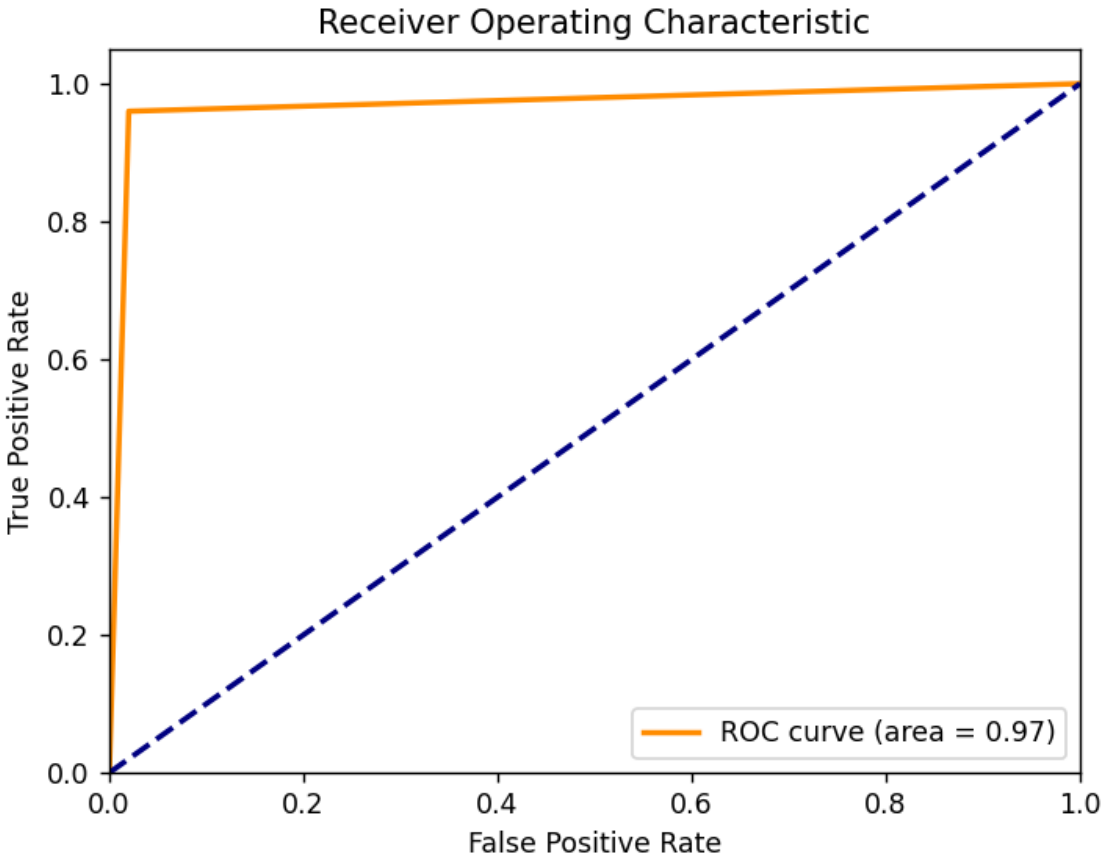
První úloha se zaměřuje na obecné určení nejvhodnějšího binárního klasifikátoru. To tedy obnáší nalezení takové funkce $C(x,\alpha)\in\{0,1\}$, kde vektor **x** popisuje námi klasifikovaný objekt, např. obrázek. Úloha nám však předkládá již natrénovaný klasifikátor C1, pro nějž byly vyzkoušeny všechny přípustné hodnoty parametru $\alpha\in\{\alpha0,\alpha1,\ldots,\alpha49\}$. Nalezení nejlepšího parametru a jeho odpovídajícího řádku můžeme provést vykreslením ROC (Receiver Operating Characteristic) křivky. Ta ukazuje vztah mezi mírou pravdivě positivních případů TPR, tedy správně určenými případy, a mírou falešně positivních případů FPR, tedy nesprávně vyhodnocenými. TRP též nazýváme senzitivita.

Obecně tedy jde při hledání nejlepšího parametru o vykreslení ROC křivky pro naše vzorová data, reprezentující objekt, a následné hledání zlomového bodu, tedy bodu na ROC křivce s nelepším poměrem mezi TPR a FPR, takzvaný ROC AUC. AUC je reprezentací plochy pod grafem ROC křivky na intervalu $FPT \in \langle 0,1\rangle$. Pokud bych použil už specifický příklad, ze zadaného klasifikátoru C1 až C2 a dat GT lze určit jako optimální klasifikátor C1, konkrétně parametr řádku 22. Ten má ROC AUC 0.97. Na straně 2 je k nahlédnutí dotyčný graf. Výpočet a vykreslení byly učiněny programem ROC.py (v odkazu).

Řešení druhého úkolu, „Přísně tajné!“, je odlišné od obecného řešení v prvním. Jedná se o situaci, kdy chceme, aby náš klíč, otisk prstu agentky 00111 reprezentovaný vektorem **x**, byl jediným možným klíčem k otevření dat. Hledaný klasifikátor, konkrétně jeho ROC křivka pro naše data musí tedy minimalizovat, ideálně anulovat FPR pro co největší nenulové TPR. Zadání nám dává neomezený čas pro odemčení, tedy nízká TPR nám nevadí. Hledání se nám trochu zkomplikuje ve chvíli, kdy více klasifikátorů disponuje nulovým FPR pro nenulové TPR. Pro takové případy je v přiloženém kódu fingerprints.py, který byl k vyřešení úlohy použit, podmínka navíc maximalizující TPR pro nulové FPR. Výsledky mého programu ukázaly, že agentka nejlépe udělá, když data zabezpečí pomocí klasifikátoru C4 a parametrem řádku 11, který má pro FPR = 0.0 maximální TPR = 0.46.

Poslední úkol, „Hlavně bezpečně“, na předchozí navazuje, jedná se o stejné jádro problému. Neznámý klasifikátor C6 od podezřelého agenta můžeme otestovat oproti našemu nejlepšímu výběru porovnáním hodnot FPR a TPR. Konkrétně pokud je můj FPR lepší (nižší) než od FPR klasifikátoru agenta, nebo se naše FPR rovnají ale mé TPR je lepší(vyšší), pak nám agent poskytl méně bezpečný klasifikátor, než je ten náš. V opačném případě je jeho klasifikátor lepší. Implementace vyčtených podmínek je v odevzdávaném programu compare.py, který zároveň poskytuje odpověď a data ohledně klasifikátoru od agenta.

Odkaz na Github repositář s programy: https://gitlab.fit.cvut.cz/svadlfil/vyber-klasifikatoru-kui


Receiver Operating Characteristic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.97)