Politechnika Warszawska
Wydział Matematyki i Nauk Informacyjnych

Automatyczne uczenie maszynowe

Starcie drugie, czyli jak stunować modele i nie oszaleć w trakcie

Autorzy:
Siwak Paweł
Wujkowski Daniel

16.01.2024

Spis treści

1 Wstęp

Poniższy dokument jest sprawozdaniem z realizacji drugiego zadania domowego w ramach przedmiotu **Automatyczne uczenie maszynowe** na wydziale **MiNI PW**. Celem zadania było stworzenie modeli o największej mocy predykcyjnej dla zadania binarnej klasyfikacji, zarówno z wykorzystaniem "klasycznego" uczenia maszynowego oraz poznanych frameworków AutoML.

2 Podstawowa analiza danych

W celu lepszego zrozumienia problemu rozpoczęto od sprawdzenia podstawowych właściwości zbioru danych treningowych i testowych. Wyciągnięto następujące wnioski:

1. Na dane składa się 500 zmiennych.
2. W zbiorze treningowym znajduje się 2000 obserwacji, a w testowym - 600.
3. Zbiory danych składają się wyłącznie z danych numerycznych, bez brakujących wartości.
4. Nie występuje żadna klasa przeważająca w klasyfikacji, w zbiorze treningowym znajduje się dokładnie 1000 obserwacji dla klasy 1 i 1000 dla klasy -1.

3 Eksploracja danych

W związku z bardzo dużą liczbą zmiennych zdecydowano skupić się na jej redukcji. Po pierwsze pozwala to na późniejsze sprawdzenie większej liczby modeli (gdyż zmniejsza czas potrzebny na ich uczenie), a po drugie - wpływa pozytywnie na ich jakość.

3.1 PCA

Jedną z pierwszych znalezionych metod było PCA, czyli Principal Component Analysis. Niestety nie najlepsza umiejętność czytania ze zrozumieniem jednego z autorów spowodowała błędne jego zastosowanie. Doprowadziło to do zaskakującego rezultatu, gdzie po usunięciu z danych 20 ostatnich zmiennych udało się osiągnąć wynik 0.933 dla próbki 5% zbioru testowego. Późniejsze próby zastosowania tej metody we właściwy sposób skończyły się niepowodzeniem. Uzyskiwane wyniki okazywały się być z reguły bardzo niesatysfakcjonujące, tym samym zaniechano dalszych prób stosowania PCA w ramach wstępnego przetwarzania.

3.2 Autorska metoda

W wyniku wyżej opisanego incydentu podjęto próbę lepszego zrozumienia podstaw eksploracji danych. W efekcie powstał filtr zmiennych, który usuwał zmienne, dla których dowolny

z następujących 3 testów zakończył się niepowodzeniem:

1. Sprawdzana była korelacja danej zmiennej ze zmienną odpowiedzi; odrzucane były wszystkie zmienne, których wartość bezwzględna korelacji była mniejsza niż 0.001.
2. Badana była wartość miary Mutual Info (1); usuwano wszystkie zmienne o wartości mniejszej niż 0.00001.
3. Wykonano na danych test chi-kwadrat (4); odrzucano wszystkie zmienne, dla których p-value wyniosło więcej niż 0.1.

Konkretne wartości punktów odcięcia dla tych testów były wyznaczane eksperymentalnie. Korzystając z opisanych powyżej filtrów, udało się zredukować liczbę zmiennych do około 50. Pozwoliło to osiągnąć zauważalnie lepszy wynik niż w przypadku testów z zastosowaniem metod opisanych w poprzednim podrozdziale.

3.3 Sequential Feature Selection

Następnym krokiem było przetestowanie metody Sequential Feature Selection. Eksperymenty przeprowadzone zostały w kilku wariantach, zarówno w trybie *forward*, jak i *backward*. Algorytmu użyto na wstępnie ograniczonym przez opisane powyżej filtry zbiorze zmiennych z uwagi na niezwykle duży nakład czasowy konieczny do uruchomienia go na pełnych danych. Mimo to sam proces i tak okazał się dość żmudny, a wyniki, które udało się finalnie osiągnąć i tak były zauważalnie gorsze niż w przypadku zastosowania opisanych w dalszej części raportu metod, tym samym dalsze prace z tym narzędziem porzucono.

3.4 ANOVA

Dalsze poszukiwania doprowadziły do artykułu (3), sugerującego zastosowanie analizy wariancji (ANOVA) w przypadku rozważanego problemu. W tym celu użyto metody `SelectKBest` z funkcją `f_classif` jako parametrem. Aby wyznaczyć optymalną liczbę najlepszych zmiennych, które powinny zostać wybrane, przetestowano powyższy algorytm dla szeregu różnych wartości. Ostatecznie redukcja do trzynastu parametrów zapewniła najlepsze wyniki i tym samym uznano to za optymalną liczbę.

3.5 Boruta

Ostatnią sprawdzoną metodą eksploracji danych był algorytm Boruta (2), w niektórych zakątkach Internetu określany jako *game-changer*. Postanowiono zatem zbadać jego potencjał w kontekście rozważanego problemu. Metoda pozwoliła ograniczyć zbiór do 21 zmiennych, równocześnie zapewniając w większości sytuacji istotną przewagę mocy predykcyjnej modelu względem przypadków, w których skorzystano z innych, wcześniej testowanych sposobów wstępnego przetwarzania danych.

4 Klasyczne uczenie maszynowe

W przypadku klasycznych modeli najlepsze wyniki zostały osiągane w przypadku redukcji zmiennych z wykorzystaniem algorytmu Boruty. Na potrzeby testów zdecydowano się również odpowiednio przeskalować wartości zmiennych przy pomocy narzędzia *StandardScaler* z uwagi na niedrzewiasty charakter niektórych rozważanych modeli. Ostatecznie zbadano 6 różnych algorytmów, które wytrenowane zostały na zredukowanym zbiorze, a następnie porównano ich jakość po zastosowaniu optymalizacji Bayesowskiej z 5-stopniową crosswalidacją i 50 iteracjami. Podjęto również próbę przeprowadzenia testów z użyciem techniki random search, jednakże dawała ona zdecydowanie mniej przewidywalne i z reguły zauważalnie gorsze rezultaty.

Wyniki zebrano w tabeli 1. Dla każdego przypadku obliczona została miara balanced accuracy dla wydzielonego z danych treningowych zbioru testowego ($\frac{1}{3}$ zbioru uczącego). W przypadku sieci neuronowej (MLP) przeprowadzono szereg eksperymentów, jednakże dość szybko zrezygnowano z dalszej optymalizacji. Dlatego też we wspomnianej tabeli nie została wymieniona ostatecznie uzyskana najlepsza wartość. Związane było to z mnogością parametrów, które można było dostosować i otrzymywanymi przeciętnymi wynikami. Zdecydowano, że jest to proces zbyt czasochłonny i potencjalnie niedający lepszych wyników niż inne rozważane modele.

Rodzaj modelu	Domyślne hiperparametry	Bayes
Random Forest	0.883	0.879
Gradient Boosting	0.762	0.871
SVM	0.827	0.878
Multi Layer Perceptron	0.756	-
Extra Trees	**0.886**	**0.903**
Histogram-based Gradient Boosting	0.856	0.873

Tabela 1: Zestawienie miary balanced accuracy dla testowanych modeli ML.

Jak widać, zdecydowanie najlepszy wynik udało się osiągnąć dla modelu Extra Trees. Mimo iż modele, takie jak SVM czy GB wykazywały bardzo dużą tunowalność, to ostatecznie otrzymane rezultaty i tak okazały się zauważalnie gorsze.

5 AutoML

W przypadku metod AutoML sprawdzono jakość dwóch frameworków: TabPFN oraz Autogluon. Dla wszystkich algorytmów zastosowano ten sam preprocessing, co w przypadku modeli klasycznych. Następnie dla każdego z modeli podjęto próbę nieznacznego tuningu hiperparametrów. Przez "automatyczną" charakterystykę używanych modeli z reguły niemożliwym było dostosowaniu dużej liczby opcji.

W pierwszej kolejności przetestowano framework TabPFN, gdyż wydawał się on dość

Framework	Boruta	ANOVA
TabPFN	0.832	0.856
Autogluon (default preset, 1h)	0.874	0.874
Autogluon (best preset, 1h)	-	0.898
Autogluon (default preset + stacking + bagging, 1h)	-	0.892
Autogluon (best preset + stacking + bagging, 1h)	-	**0.900**
Autogluon (best preset + stacking + bagging, 8h)	-	**0.900**

Tabela 2: Balanced accuracy dla różnych frameworków AutoML.

prostym i szybko działającym algorytmem. Istotne wydawało się również to, że działa on najlepiej dla zbiorów danych zawierających jedynie zmienne numeryczne. Podjęto próbę tunowania parametru N_ensemble_configurations, jednakże zmiana jego wartości nie wnosiła istotnych różnic w otrzymywanej mocy predykcyjnej.

Framework Autogluon od początku dawał zauważalnie lepsze rezultaty od poprzednika, dlatego zdecydowano się sprawdzić jego różne opcje konfiguracyjne. Z uwagi na pomijalnie krótki czas wykonania testu oraz zapewnienie determinizmu w przypadku zastosowania redukcji liczby zmiennych przy użyciu analizy wariancji ograniczono się do tej metody. Sprawdzono wpływ zmiany używanego ustawienia (`presets = best_quality`), umożliwienia zastosowania stackingu (`num_stack_levels = 3`) i baggingu (`num_bag_folds = 5`) oraz maksymalnego czasu uczenia.

Wyniki jakości sprawdzonych frameworków zebrano w tabeli 2. Tak samo, jak w poprzednim eksperymencie, dla każdego przypadku obliczona została miara balanced accuracy dla wydzielonego z danych treningowych zbioru testowego ($\frac{1}{3}$ zbioru uczącego).

6 Wnioski

Można zauważyć, że w przypadku użycia 'klasycznego' uczenia maszynowego udało się osiągnąć nieznacznie lepsze wyniki. W przypadku większego doświadczenia osób odpowiedzialnych za stworzenie modelu być może udałoby się tę różnicę powiększyć. Jednakże czas pracy potrzebny do uzyskania takich wyników jest wręcz nieporównywalny. W przypadku zwykłego modelu wymagane było przynajmniej kilka godzin w celu doboru, tunowania, uczenia i porównywania różnych rodzajów modeli. W przypadku automatycznego modelu, przy prawie zerowej ingerencji, udało się osiągnąć praktycznie równie wysokie wyniki. Pokazuje to, że automatyczne uczenie maszynowe jest bardzo obiecującym rozwiązaniem dla zadań predykcyjnych w przypadku mocno ograniczonych zasobów czasowych oraz ludzkich.

Literatura

[1] Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *Neural Networks, IEEE Transactions on*, 5:537 – 550, 08 1994.

[2] Miron B. Kursa and Witold R. Rudnicki. Feature selection with the boruta package. *Journal of Statistical Software*, 36(11):1–13, 2010.

[3] Jason Brownlee PhD. How to choose a feature selection method for machine learning. *Machine Learning Mastery*, 2020.

[4] Chi-squared test. https://en.wikipedia.org/wiki/Chi-squared_test.