Raport Homework 2

Bartosz Jadczak
Maciej Kordyaczny

15 stycznia 2024

1 Wstęp

Celem niniejszego projektu było opracowanie skutecznej metody klasyfikacji, umożliwiającej zbudowanie modelu o jak największej mocy predykcyjnej. W ramach zadania dysponowaliśmy sztucznie wygenerowanym zbiorem danych o nazwie "artificial", w którym ukryte były istotne zmienne wymagające klasyfikacji dwóch klas. Kluczowym kryterium oceny skuteczności modelu była zrównoważona dokładność (balanced accuracy). Model należało przygotować w dwuch wariantach: wybierając model ręcznie oraz wykorzystując frameworki AtoMLowe.

2 Ręcznie wybrany model

W tym rozdziale opisujemy jak krok po kroku budowaliśmy i udoskonalaliśmy nasz model klasyfikacyjny. Skupiliśmy się na iteracyjnym procesie, gdzie każda zmiana miała na celu podbicie balanced accuracy - naszej głównej miary efektywności. Zaczynaliśmy od prostego klasyfikatora RandomForest, przechodziliśmy przez selekcję cech, dopasowywanie ich liczby, aż do rozbudowania modelu o dodatkowe klasyfikatory. W każdym etapie ważne było dla nas, aby balanced accuracy rosło, co kierowało naszymi decyzjami o wyborze narzędzi i metod.

2.1 Klasyfikacja przy użyciu RandomForest

W pierwszym kroku zaimplementowaliśmy ręczną klasyfikację danych przy użyciu algorytmu RandomForest. Użyliśmy biblioteki scikit-learn do tego celu. Ustaliliśmy hiperparametry modelu, takie jak liczba drzew decyzyjnych (n_estimators) i ziarno losowości (random_state). Następnie dokonaliśmy treningu modelu na zbiorze treningowym i przeprowadziliśmy predykcję na zbiorze walidacyjnym. Ta metoda była użyta jako punkt odniesienia do porównań z późniejszymi modelami.

Wynik BA otrzymany w tym etapie: 0.654

2.2 Tworzenie komitetu modeli

Kolejnym etapem było stworzenie komitetu modeli, który miał na celu poprawienie jakości klasyfikacji. Do komitetu wybraliśmy trzy różne algorytmy: RandomForest,

GradientBoosting oraz SVM. Skorzystaliśmy z paczki VotingClassifier z ustawieniem na 'soft', co oznaczało używanie prawdopodobieństw zamiast klas podczas głosowania.

Wynik BA otrzymany w tym etapie: 0.726

2.3 Wybór najważniejszych cech za pomocą RandomForest

W trzecim etapie zdecydowaliśmy się na wybór najważniejszych cech za pomocą algorytmu RandomForest. Użyliśmy funkcji SelectFromModel z limitem maksymalnie 50 cech. Następnie stworzyliśmy komitet modeli, podobnie jak w punkcie 2, ale tym razem na danych z wybranymi cechami. To pozwoliło nam sprawdzić, czy redukcja cech może poprawić jakość klasyfikacji.

Wynik BA otrzymany w tym etapie: 0.800

2.4 Optymalizacja ilości cech

W koljenej części skoncentrowaliśmy się na optymalizacji liczby cech. Ustaliliśmy zakres od 10 do 200 cech z krokiem co 10, testując na każdym etapie skuteczność komitetu modeli. Dla każdej iteracji ponownie stosowaliśmy SelectFromModel i wytrenowaliśmy komitet modeli na nowo wyselekcjonowanym zbiorze cech. Naszym celem było znalezienie optymalnej liczby cech, dla której osiągaliśmy najwyższą wartość balanced_accuracy. Liczba cech która przyniosła najlepszy wynik to: 20.

Wynik BA otrzymany w tym etapie: 0.840

2.5 Rozszerzona optymalizacja cech

Po wyznaczeniu optymalnej liczby cech, zdecydowaliśmy się na jej precyzyjne dostrojenie. Rozszerzyliśmy zakres poszukiwań o 5 cech w górę i w dół od wcześniej ustalonej optymalnej wartości, zmniejszając krok do 1. Taki podejście pozwoliło na dokładniejszą analizę wpływu liczby cech na skuteczność modelu. Podobnie jak wcześniej, dla każdej liczby cech trenowaliśmy komitet modeli i obserwowaliśmy wyniki balanced_accuracy, co finalnie pozwoliło na wyselekcjonowanie jeszcze bardziej precyzyjnego zestawu cech. Ilośćcech wyniosła: 21.

Wynik BA otrzymany w tym etapie: 0.845

2.6 Końcowy model

W ostatniej części tego zadania, zdecydowaliśmy się na rozszerzenie komitetu modeli o dodatkowe klasyfikatory. Oprócz RandomForest, GradientBoostingClassifier i SVC, dołączyliśmy KNeighborsClassifier i ExtraTreesClassifier. Użyliśmy ExtraTreesClassifier do selekcji cech, co pozwoliło na uwzględnienie różnorodności w podejściach klasyfikacyjnych. Wytrenowaliśmy komitet modeli z tymi klasyfikatorami, korzystając z danych z wyselekcjonowanymi cechami. Wynik balanced_accuracy dla tego rozszerzonego modelu dostarczył nam końcowych wniosków dotyczących efektywności naszego podejścia w

kontekście różnorodności klasyfikatorów.

Wynik BA otrzymany w tym etapie: 0.870

3 Model wykorzystujący Autogluon

W tym rozdziale przedstawimy inne podejście do rozwiązania naszego zadania klasyfikacji, wykorzystując narzędzie AutoGluon. AutoGluon to zaawansowany framework automatycznego uczenia maszynowego, który umożliwia automatyzację wielu etapów procesu tworzenia modeli, co znacznie usprawnia pracę nad problemami klasyfikacji.

Wybraliśmy AutoGluon ze względu na jego zdolność do automatycznego dostosowywania wielu hiperparametrów modelu, co pozwala na uzyskanie efektywnego klasyfikatora bez konieczności ręcznego dostosowywania parametrów. To narzędzie jest szczególnie przydatne w przypadku dużych zbiorów danych, gdzie ręczna optymalizacja modelu może być trudna i czasochłonna. Dzięki temu możemy skupić się na eksploracji danych i analizie wyników, a AutoGluon zajmuje się resztą.

Aby użyć AutoGluon, musieliśmy przekształcić dane do odpowiedniego formatu. Do zbioru treningowego dodaliśmy kolumnę 'label', która zawierała etykiety klas. Teraz dane były gotowe do użycia w AutoGluon.

Następnie zainicjowaliśmy i przetrenowaliśmy model AutoGluon za pomocą klasy TabularPredictor. Określiliśmy, że kolumna 'label' jest naszym celem klasyfikacji, a metryką oceny jakości modelu była balanced accuracy (eval_metric='balanced_accuracy'). Użyliśmy również opcji presets='best_quality', co pozwoliło na wykorzystanie najlepszych dostępnych ustawień w celu osiągnięcia jak najwyższej jakości modelu. Limit czasu na przetrenowanie modelu wynosił 200 sekund.

Wynik BA uzyskany przy użyciu tego modelu to: 0.852

4 Podsumowanie

Podsumowując, nasz projekt pokazał, że staranne dobieranie cech i używanie różnych metod klasyfikacji w modelu, który sami tworzyliśmy, dało lepsze wyniki niż automatyczne narzędzia jak AutoGluon. Chociaż AutoGluon jest pomocny i łatwy w użyciu, to nasze własne podejście, gdzie krok po kroku poprawialiśmy model, okazało się skuteczniejsze. To ważna lekcja, że czasem własna praca i dostosowywanie modelu do konkretnych potrzeb może przynieść lepsze efekty, szczególnie gdy mamy konkretny cel, jak wysoka dokładność klasyfikacji.