Auto ML - praca domowa 2

Maciej Malewicz, Antoni Chudy

1 Wstęp

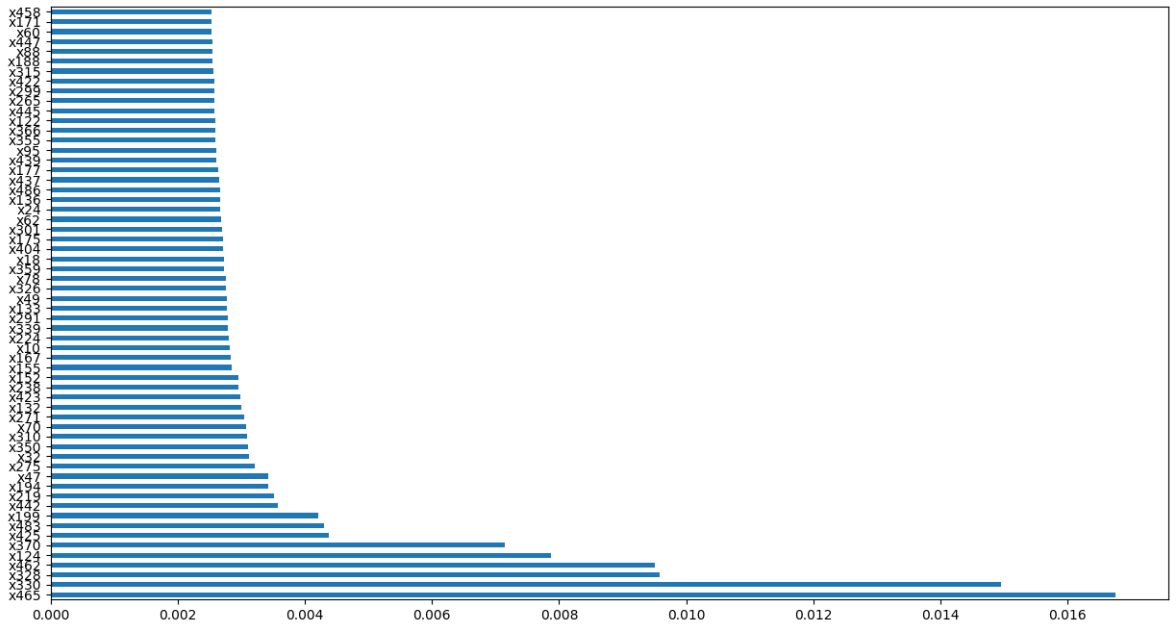
Celem projektu jest stworzenie metody klasyfikacji, która pozwoli na stworzenie modelu o najlepszej mocy predykcyjnej. Model będzie uczony na sztucznie wygenerowanym zbiorze danych o pewnych ukrytych zmiennych istotnych. Chcemy dokonać klasyfikacji do dwóch klas.
Zbiór danych nazwany w zadaniu jako *artificial_ train.data* podzieliliśmy na zbiór treningowy oraz testowy. Wspomniana w projekcie kroswalidacja obliczana jest na zbiorze treningowym, natomiast na zbiórze testowym sprawdzać będziemy jak dobrze radzą sobie modele wytrenowane na zbiorze treningowym. Dokładność modelu w każdym wypadku będzie mierzona za pomocą miary zrównoważonej dokładności balanced accuracy.

2 Przebieg projektu – wariant ręczny

1.Istotność zmiennych – korelacja. W pierwszym etapie zbadaliśmy korelacje między zmiennymi objaśniającymi. W przypadku silnego skorelowania kilku kolumn (powstała naturalna granica odcięcia o korelacji powyżej 0.9) pozostawialiśmy tylko jedną, dowolną z nich, resztę usuwając.

Kolumna 1	Kolumna 2	Korelacja
64	336	0.990429
442	472	0.990300
28	318	0.989628
105	128	0.989410
153	433	0.989355
241	475	0.989272
28	451	0.989168
281	433	0.989047
153	281	0.988695
318	451	0.988690
453	493	0.988458
48	378	0.988338
241	336	0.735253
64	241	0.734728

2.Istotność zmiennych za pomocą lasu losowego. W tym kroku na kolumnach pomniejszonych o zmienne linowo zależne wskazane przez korelacje, wytrenowaliśmy las losowy, aby zbadać, które zmienne są najistotniejsze w rozważanym problemie. Poniższy wykres przedstawia 50 najbardziej istotnych zmiennych wskazanych przez model (większa wartość oznacza, że zmienna jest bardziej istotna).



3.Wybór zmiennych za pomocą SequentialFeatureSelector(metodą „backward"). Po wyborze 50 najważniejszych zmiennych według lasu losowego, postanowiliśmy sprawdzić, jak usuwanie jeszcze kolejnych zmiennych za pomocą funkcjonalności *SequentialFeatureSelector* wpłynie na *balanced accuracy* używając wybranego modelu. Ze względu na fakt, że KNN trenuje się dość szybko wybraliśmy właśnie ten model. Poniższa tabela przedstawia wyniki walidacji krzyżowej na zbiorze treninowym tego modelu po usunięciu pewnej liczby zmiennych wskazanych przez SequentialFeatureSelector. Na

Liczba usuniętych zmiennych	1	7	13	19	25	31	36	38	40	42	44	45
Wynik	0.623	0.674	0.694	0.710	0.714	0.758	0.811	0.836	0.872	0.894	0.900	0.887

podstawie wyników z powyższej metody wybraliśmy 9 zmiennych. Za ich pomocą oraz wybranych podzbiorów tych atrybutów opierają się dalsze obliczenia.

4.Wybór modeli. Następnie sprawdzaliśmy dla wybranego podzbioru kolumn jakość predykcji wybranych przez nas modeli, były to: regresja logistyczna, SVC, KNN, lasy losowe, gradient boosting, naiwny bayes, LDA oraz QDA. Wyniki uzyskane za pomocą kroswalidacji na zbiorze treningowym i zmierzeniu zrównoważonej dokładności na zbiorze testowym prezentuje poniższa tabela:

Model	Regresja logistyczna	SVC	KNN	lasy losowe	gradient boosting	naiwny bayes	LDA	QDA
zbiór testowy	0.590	0.849	0.901	0.879	0.811	0.606	0.590	0.717
kroswalidacja	0.616	0.839	0.896	0.870	0.800	0.597	0.615	0.702

Do dalszego etapu budowania modelu postanowiliśmy wybrać KNN, SVC oraz lasy losowe ze względu na największe wartości *balanced accuracy* uzyskane przez te modele.
Dla porównania, w poniższej tabeli można zobaczyć wyniki modeli wytrenowanych na wszystkich kolumnach (nie licząc zmiennych liniowo zależnych usuniętych po punkcie (2) tego rozdziału).

Model	Regresja logistyczna	SVC	KNN	lasy losowe	gradient boosting	naiwny bayes	LDA	QDA
zbiór testowy	0.513	0.534	0.541	0.611	0.759	0.574	0.503	0.513
kroswalidacja	0.534	0.549	0.516	0.629	0.703	0.565	0.536	0.503

3 Uzyskane wyniki

Na podstawie zmiennych wybranych w punkcie o przebiegu projektu, spróbowaliśmy za pomocą metody *RandomizedSearchCV*, dla KNN, SVC, lasu losowego oraz metod Baggingu i Votingu dla tych modeli, znaleźć najbardziej optymalne hiperparametry, które mogłyby dać potencjalnie najlepszą moc predykcyjną. Poniższa tabela przedstawia uzyskane wyniki na różnych modelach wytrenowanych za pomocą hiperametrów wskazanych przez *RandomizedSearchCV*.

model	kroswalidacja	zbiór testowy
SVC	0.9032	0.9000
KNN	0.9055	0.9081
Las losowy	0.8870	0.8860
Bagging(SVC)	0.9064	0.8976
Bagging(KNN)	0.8988	0.8876
Voting(Las, KNN, SVC)	0.8998	0.9053
Voting(Las, Bagging(KNN), Bagging(SVC))	0.9042	0.9005

Biorąc pod uwagę wyniki oraz złożoności rozważanych modeli z powyższej tabeli, zdecydowaliśmy się na wybranie modelu Voting(Las, KNN i SVC). Według nas model ten będzie się bardziej stabilnie zachowywał na nowych danych. Oczywiście w ogólności wyniki w powyższej tabeli są bardzo zbliżone, zatem nie mamy pewności, który model jest tutaj najlepszy.

4 Przebieg projektu Auto–Ml

Jako bibliotekę automatycznego uczenia maszynowego wybraliśmy bibliotekę AutoGluon. W pierwszym kroku postąpiliśmy analogicznie jak w punkcie nr 1 istotność zmiennych – korelacja w wariancie ręcznym. Dzięki temu pozbyliśmy się silnie skorelowanych kolumn.

Następnie na zbiorze treningowym za pomocą pakietu AutoGluon szukamy najlepszego modelu. Obliczona dla niego *balanced accuracy* na zbiorze testowym wynosi 0.823. W ostatnim kroku dla stworzonego przez AutoGluon modelu wykonujemy predykcje na otrzymanym zbiorze.

5 Podsumowanie

Widzimy jak w rozważanym problemie ważna była selekcja zmiennych. W tabeli kończącej sekcję numer 2 większość wyników jest zbliżona do dokładności modelu losowego, natomiast wyniki po selekcji zmiennych, dla prawie wszystkich modeli znacznie wzrosły.
Natomiast skupiając się wyłącznie na wynikach po selekcji zmiennych możemy zauważyć, że w rozważanym problemie podstawowe modele KNN oraz SVC dają dobrą jakość predykcji. Zastosowanie bardziej skomplikowanych metod, takich jak bagging czy voting nie przyczynia się do istotnej poprawy jakości predykcji. Należy jednak wziąć pod uwagę fakt, że model oparty na votingu oraz votingu połączonego z baggingiem może być bardziej odporny na potencjalne różnice w nowym zbiorze.