Dydaktyka Informatyki 16(2021)
ISSN 2083-3156; e-ISSN 2543-9847
DOI: 10.15584/di.2021.16.14
http://www.di.univ.rzeszow.pl

Uniwersytet Rzeszowski

Jacek WOŁOSZYN

ORCID: 0000-0003-4340-9853. Dr inż., Uniwersytet Technologiczno-Humanistyczny w Radomiu, Wydział Informatyki i Matematyki, Katedra Informatyki, ul. Malczewskiego 20A; 26-600 Radom; e-mail: jacek.woloszyn@uthrad.pl

METODY DOBORU ZMIENNYCH DO MODELU Z WYKORZYSTANIEM BIBLIOTEK SZTUCZNEJ INTELIGENCJI

SUBSET SELECTION VARIABLES OF THE MODEL USING AI LIBRARIES

Słowa kluczowe: model, sztuczna inteligencja, dobór zmiennych.
Keywords: model, artificial intelligence, subset selection.

Streszczenie

Inwestycje w struktury informatyczne firm zaowocowały niespotykanym wzrostem posiadanych danych. Ten olbrzymi przyrost danych gromadzony praktycznie w każdym aspekcie dziedziny życia doprowadził do wzrostu zainteresowania metodami wydobywania informacji, wiedzy czy zależności. Przeprowadzając rozmyślania w kategorii analityki danych prawie zawsze należy dokonać wyboru zmiennych tak, aby każdy model w swojej końcowej postaci jak najprecyzyjniej odzwierciedlał rozważany proces. W artykule tym przedstawione zostaną najczęściej stosowane metody doboru zmiennych do modelu. Proces ten jest jednym z etapów budowy modelu i od jego przebiegu zależy w dużym stopniu końcowy efekt działania modelu.

Abstract

Investments in IT structures of companies resulted in an unprecedented increase in the collected data. This enormous increase in data collected in practically every aspect of the sphere of life has led to an increased interest in the methods of extracting information, knowledge and dependencies. When thinking about data analytics, you should almost always select the data so that each model in its final form reflects the process under study as accurately as possible. In this article, the most common methods of selecting variables for the model will be presented. This process is one of the stages of model building and the final effect of the model to a large extent depends on its course.

Wstęp

Dysponując ogromnymi ilościami danych staramy się pozyskać z nich jak najwięcej informacji pod kątem analizowanego problemu. Techniki eksploracji wiedzy znacznie się rozwinęły na przestrzeni ostatnich lat. W przeszłości zespoły analityków, statystyków, ekonometryków prowadziły analizy za pomocą ręcznych metod. Obecnie tak ogromny przyrost danych doprowadziłby do całkowitego zablokowania tradycyjnych metod. Dostępność i wzrost wydajności komputerów wymusił rozwój algorytmów sztucznej inteligencji. Przy tak dużej mocy obliczeniowej powstały metody łączenia danych, co spowodowało znaczny przyrost jakościowy technik eksploracji. Jednak, czy to w starej technologii, czy w nowej bardzo istotny jest proces doboru zmiennych do modelu. Prawidłowe przeprowadzenie tego procesu pozwoli na uniknięcie zjawiska korelacji, współliniowości, heteroscedastyczności, które są przyczyną niepoprawnego działania modelu. Na początku opracowania przedstawiono krótki opis danych wykorzystywanych w metodach doboru zmiennych do modelu. W dalszej części przedstawiono same metody i ich odpowiedzi na działanie danych wejściowych. Opisano regresję lasso, rekurencyjną eliminację cech, informację wzajemną, cykliczną eliminację cech, jak i klasyczną korelację opisującą powiązanie pomiędzy zmiennymi.

1. Dane do testowania

Do przeprowadzenia procesu wyboru cech do budowy modelu zostaną wykorzystane dane jak na rys. 1 zgromadzone w bazie repozytorium UCI Machine Learning Repository titanic. Zbiór ten zawiera listę pasażerów Titanica. Dane umieszone są w kolumnach i zawierają następujące informacje:

pclass – klasa (1 – pierwsza, 2 – druga, 3 – trzecia),
survived – pasażer ocalał (0 – nie, 1 – tak).
name – imię i nazwisko,
sex – płeć (male —– mężczyzna, female – kobieta),
age – wiek,
sibsp – towarzysząca żona/mąż lub liczba bliźniaków,
parch – liczba towarzyszących dzieci/rodziców,
ticket – numer biletu,
fare – cena biletu,
cabin – numer kajuty,
embarked – miejsce zaokrętowania (C = Cherbourg, Q = Queenstown, S = Southampton),

boat – numer szalupy ratunkowej,
body – identyfikator zwłok,
home.dest – miejsce zamieszkania/cel podróży.

Analiza zostanie przeprowadzona z wykorzystaniem języka Python[1] w wersji 3 z wykorzystaniem bibliotek Pandas, Matplotlib, Sklearn[2], Rfpimp, Yellowbrick.

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
0	1	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S	2	NaN	St Louis, MO
1	1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville, ON
2	1	0	Allison, Miss. Helen Loraine	female	2.0000	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
3	1	0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1	2	113781	151.5500	C22 C26	S	NaN	135.0	Montreal, PQ / Chesterville, ON
4	1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
5	1	1	Anderson, Mr. Harry	male	48.0000	0	0	19952	26.5500	E12	S	3	NaN	New York, NY

Rys. 1. Fragment wykorzystywanego zbioru danych

Źródło: opracowanie własne.

2. Metody wyboru cech do modelu

Proces tworzenia modelu wymaga w początkowym etapie, aby wybrać zmienne do modelu. Idealne rozwiązanie to takie, gdyby można było do jego utworzenia użyć wszystkich zgromadzonych danych. Jednak w praktyce tak zdarza się niezmiernie rzadko. Uwzględnienie cech niepożądanych z punktu widzenia modelu może negatywnie wpływać na jego funkcjonowanie. Dla przykładu silna korelacja powoduje, że wskaźniki regresji są obarczone dużymi błędami, co wiąże się z tym, że zastosowanie takiego modelu w praktyce mija się z celem, ponieważ generuje on duże błędy.

2.1. Regresja lasso

Wykorzystując parametr regularyzycyjny alfa w regresji lasso w klasie LassoLarsCV modelu linear_model z biblioteki scikit-learn można regulować wagę cech. Im mniejsza jest nadawana waga mniej istotnym cechom, tym większa jest jego wartość. Klasa LassoLarsCV wylicza współczynniki regresji cech dla poszczególnych parametrów alfa wpływając tym samym na końcowy wynik generowany na wyjściu.

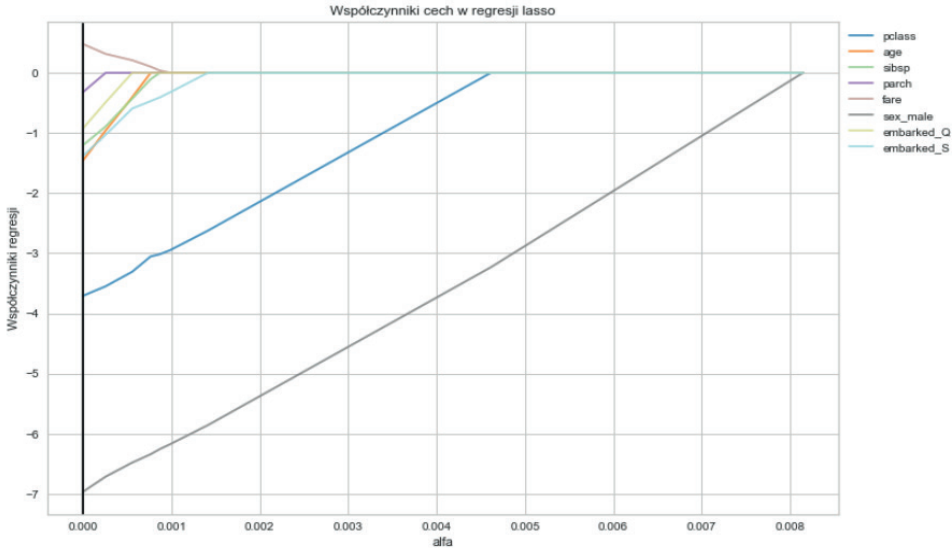
Można zwizualizować zależność współczynników regresji cech od parametru alfa. Należy zaimportować metodę linear_model z biblioteki sklearn i utworzyć model dla poszczególnych parametrów.

[1] A. Boschetti, L. Massaron, *Python. Podstawy nauki o danych*, Helion, Gliwice 2017.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, *12 Scikit-learn: Machine Learning in Python*, „Journal of Machine Learning Research" 2011, 12: 2825-2830.

```
from model import linear_model
model = linear_model.LassoLarsCV(cv=10, max_n_alphas=10).fit(X, Y)
fragment kodu
```

Wynikiem działania algorytmu jest wygenerowany wykres (rys. 2).



Rys. 2. Zależności współczynników regresji cech od parametrów modelu

Źródło: opracowanie własne.

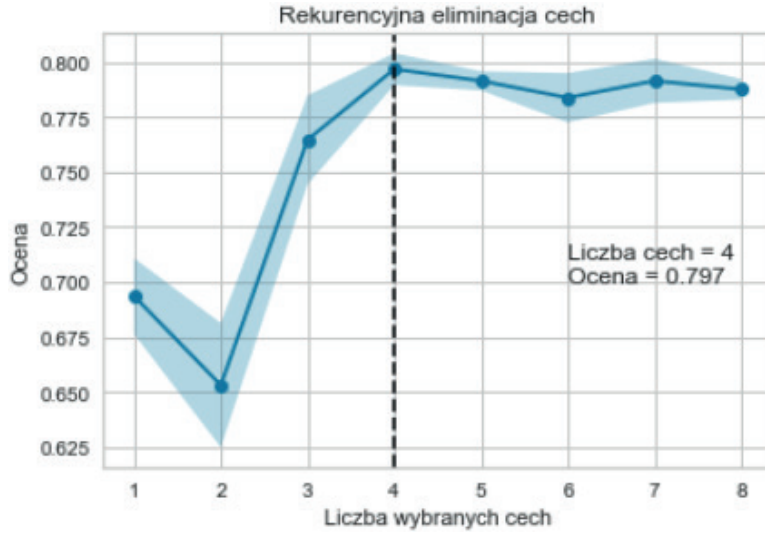
2.2. Rekurencyjna eliminacja cech

Interesującym podejściem jest też rekurencyjna eliminacja cech, która polega na kolejnym usuwaniu cech i dopasowaniu do modelu. Wykorzystywana jest do tego celu biblioteka Yellowbrick, a konkretnie funkcja RFECV

```
from yellowbrick.features import RFECV
fig, ax = plt.subplots(figsize=(6, 4))
rfe = RFECV(ensemble.RandomForestClassifier(n_estimators=100),  cv=4)
…
rfe.rfe_estimator_.ranking_
rfe.rfe_estimator_.n_features_
rfe.rfe_estimator_.support_
fragment kodu
```

```
array([True, True, False, False, True, True, False, False])
```

Jak widać, do modelu[3] zostały zakwalifikowane zmienne 1, 2, 5, 6 – rys. 3. Oznacza to, że do budowy końcowej postaci modelu należy wybrać te zmienne, gdyż właśnie ich użycie zapewnia najlepsze działanie modelu. Optymalne działanie modelu wykorzystuje oczywiście fakt, że zostawia on najmniejsze reszty. A co, jeśli chcemy użyć koniecznie innych zmiennych? Oczywiście można i nawet trzeba ich użyć, jeśli zależy nam na umieszczeniu ich w modelu. Należy jednak na uwadze mieć fakt, że taki model będzie generował większe błędy, co należy uwzględnić przy formułowaniu końcowych wniosków.



Rys. 3. Cykliczna eliminacja cech

Źródło: opracowanie własne.

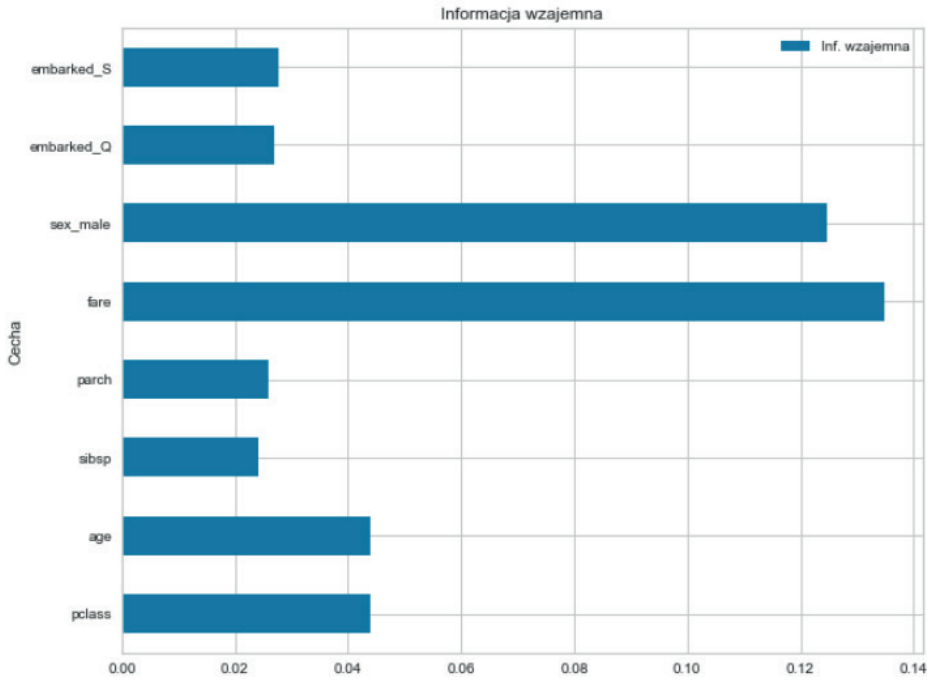
2.3. Informacja wzajemna

Kolejnym algorytmem, za pomocą którego można określać wzajemne powiązania o cechach i wartościach jest informacja wzajemna. Jest to miara zależności określająca liczbę bitów podobieństwa cechy i celu. Wartość zerowa oznacza brak powiązań. Wykorzystuje się do tego bibliotekę scikit-learn, a konkretnie test wykorzystujący algorytm k-najbliższych sąsiadów.

```
from sklearn import feature_selection
mic = feature_selection.mutual_info_classif(X, y)
fragment kodu
```

[3] M. Goodrich, R. Tamassia, M. Goldwasser, *Data Structures and Algorithms in Python*, Wiley 2013; Y. Hilpisch, *Derivatives Analytics with Python*, Wiley 2015.

Wynikiem działania tego algorytmu jest wykres uzyskany na rys. 4.



Rys. 4. Ważność cech wyliczona przez algorytm

Źródło: opracowanie własne.

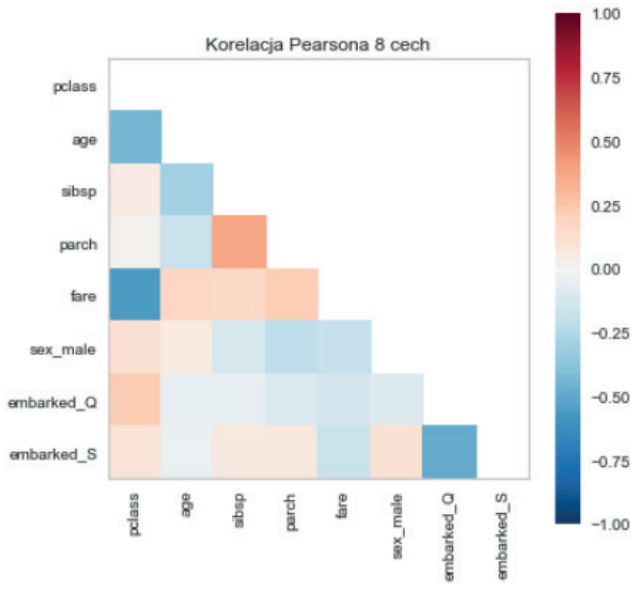
2.4. Analiza korelacji

Kolejnym sposobem jest wybór takich cech modelu, których wariancja jest największa. Oczywisty jest fakt, że zawierają one wówczas największą ilość informacji. Spowodowane jest to tym, że zmienne o większym rozrzucie mocniej wpływają na wynik końcowy, aniżeli zmienna zawierająca na przykład stałą wartość. Taka zmienna w żaden sposób nie wpływa na końcowy wynik i można ją śmiało z modelu usunąć.

Kolejnym tradycyjnym i najczęściej stosowanym sposobem jest wyliczenie korelacji pomiędzy wszystkimi zmiennymi i wybór tych cech, których wartość pomiędzy zmienną objaśnianą, a zmiennymi objaśniającymi jak największa. Wartość bezwzględna wyliczonej korelacji zawsze mieści się w przedziale od <0, 1>. Wartość zero oznacza brak korelacji, a wartość 1 wartość funkcyjną. Jednocześnie należy pamiętać o tym, że wartość korelacji pomiędzy samymi zmiennymi objaśniającymi powinna być jak najmniejsza. W przypadku wystąpienia dużych wartości pomiędzy nimi należy zdecydować się na usunięcie jed-

nej lub wielu z nich. Należy wówczas pozostawić w modelu tę zmienną, która według naszego uznania będzie nam najbardziej potrzebna. Wizualizację wyliczonych wartości korelacji pokazuje rys. 5.

```
from yellowbrick.features import Rank2D
fig, ax = plt.subplots(figsize=(6, 6))
pcv = Rank2D(features=X.columns, algorithm="pearson", title="Korelacja ..
fragment kodu
```



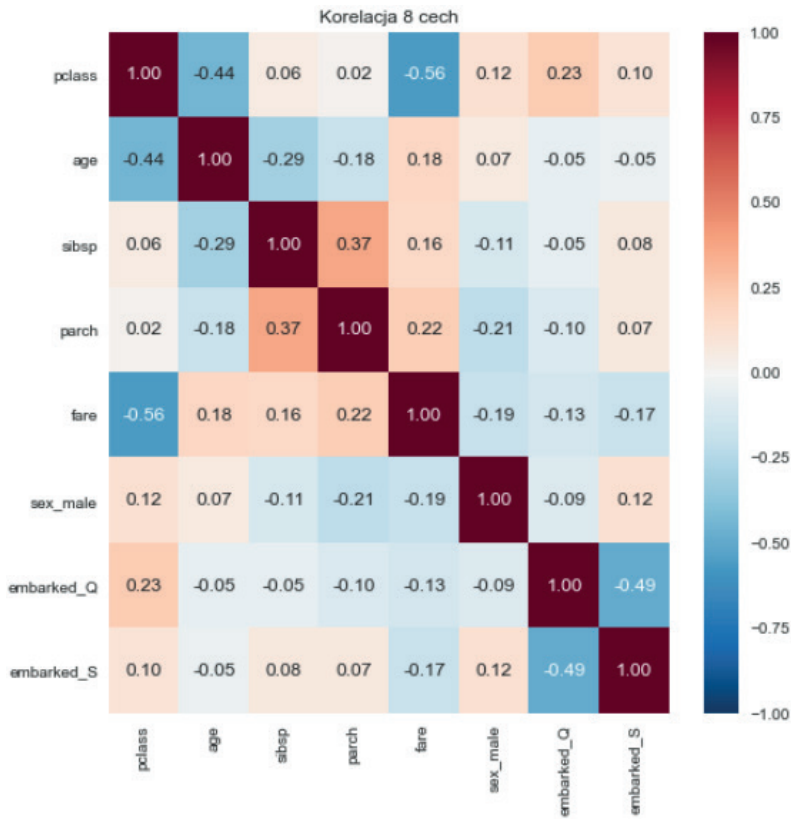
Rys. 5. Korelacja Pearsona wybranych cech

Źródło: opracowanie własne.

Wynik działania tego algorytmu generuje macierz korelacji. Można z niej wyczytać, jak silnie są ze sobą powiązane poszczególne zmienne i zadecydować o ich losie w modelu.

Podobną funkcję pełni tak zwana mapa ciepła (rys. 6), która została wygenerowana metodą heatmap z biblioteki seaborn. Zawiera ona dokładnie te same informacje, jednak są one przedstawione w nieco odmienny sposób.

```
from seaborn import heatmap
fig, ax = plt.subplots(figsize=(8, 8))
ax = heatmap(X.corr(),  fmt=".2f", annot=True,  ax=ax,  cmap="RdBu_r", ..
fragment kodu
```



Rys. 6. Mapa ciepła

Źródło: opracowanie własne.

Podsumowanie

Model to nic innego jak uproszczenie pewnej rzeczywistości. Pozwala on nam zrozumieć analizowane zjawiska i poznać, jakie zmienne wartości są jego głównymi składowymi. Oznacza to, że mając poprawnie zbudowany model możemy napisać algorytm[4], który będzie symulował analizowaną przez nas rzeczywistość. Wpływając na parametry modelu uzyskujemy odpowiedzi, które informują nas, jak zachowa się badane otoczenie, gdy będą na niego działać rozpatrywane zmienne. Problem w tym, że aby model działał poprawnie, to musi zostać zbudowany w oparciu o pewne zasady. Jedną z nich jest wybór odpowiednich zmiennych. Tylko wówczas, gdy będzie generowany mały błąd, wów-

[4] S. Raschka, V. Mirjalili, *Python Machine Learning*, Packt 2017.

czas odpowiedzi będą poprawne. Uproszenie samego modelu oparte na założeniach dotyczących tego co jest dostępne, a nie jest istotne dla prześledzenia samego procesu, jest kolejnym ważnym aspektem, na który trzeba zwrócić uwagę. Możemy powiedzieć, że model budowany przez nas model predykcyjny to wzór umożliwiający oszacowanie nieznanej wartości docelowej. Wybór zmiennych do modelu jest jednym z najbardziej istotnych i trudnych etapów w procesie budowy modelu. Opisane metody pomagają badaczowi w tym procesie sugerując najlepsze rozwiązanie, nie zawsze jest ono jednak zgodne z oczekiwaniami badacza. Ostateczna decyzja jednak zawsze należy do nas.

Bibliografia

Boschetti A, Massaron L, *Python. Podstawy nauki o danych*, Helion, Gliwice 2017.

Goodrich M., Tamassia R., Goldwasser M., *Data Structures and Algorithms in Python*, Wiley 2013.

Hilpisch Y., *Derivatives Analytics with Python*, Wiley 2015.

Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay É., *12 Scikit-learn: Machine Learning in Python*, „Journal of Machine Learning Research“ 2011, 12: 2825-2830.

Raschka S., Mirjalili V., *Python Machine Learning*, Packt 2017.