



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
Wydział Zarządzania

Praca dyplomowa

*Zastosowanie uczenia maszynowego do prognozowania
zmian cen akcji*

Autor:
Kierunek studiów:
Opiekun pracy:

Łukasz Chuchra
Informatyka i Ekonometria
dr hab. Tomasz Wójtowicz

Kraków, 2023

Oświadczenie studenta

Upředzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz. U. z 2018 r. poz. 1191 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.”, a także upředzony(-a) o odpowiedzialności dyscyplinarnej na podstawie art. 307 ust. 1 ustawy z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2018 r. poz. 1668 z późn. zm.) „Student podlega odpowiedzialności dyscyplinarnej za naruszenie przepisów obowiązujących w uczelni oraz za czyn uchylający godności studenta.”, oświadczam, że niniejszą pracę dyplomową wykonałem osobiście i samodzielnie i nie korzystałem ze źródeł innych niż wymienione w pracy.

Jednocześnie Uczelnia informuje, że zgodnie z art. 15a ww. ustawy o prawie autorskim i prawach pokrewnych Uczelnia przysuguje pierwszeństwo w opublikowaniu pracy dyplomowej studenta. Jeżeli Uczelnia nie opublikowała pracy dyplomowej w terminie 6 miesięcy od dnia jej obrony, autor może ją opublikować, chyba że praca jest częścią utworu zbiorowego. Ponadto Uczelnia jako podmiot, o którym mowa w art. 7 ust. 1 pkt 1 ustawy z dnia 20 lipca 2018 r. — Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2018 r. poz. 1668 z późn. zm.), może korzystać bez wynagrodzenia i bez konieczności uzyskania zgody autora z utworu stworzonego przez studenta w wyniku wykonywania obowiązków związanych z odbywaniem studiów, udostępniać utwór ministrowi właściwemu do spraw szkolnictwa wyższego i nauki oraz korzystać z utworów znajdujących się w prowadzonych przez niego bazach danych, w celu sprawdzania z wykorzystaniem systemu antyplagiatowego. Minister właściwy do spraw szkolnictwa wyższego i nauki może korzystać z prac dyplomowych znajdujących się w prowadzonych przez niego bazach danych w zakresie niezbędnym do zapewnienia prawidłowego utrzymania i rozwoju tych baz oraz współpracujących z nimi systemów informatycznych.

.....
(czytelny podpis)

Spis treści

Wprowadzenie	4
1 Przegląd literatury naukowej	6
2 Metodyka badań	10
2.1 Język Python	11
2.2 Uczenie maszynowe	11
2.3 Drzewa decyzyjne	12
2.3.1 Bagging	18
2.3.2 Lasy losowe	19
2.3.3 Boosting	20
2.4 Regresja logistyczna	22
2.5 Maszyna wektorów nośnych (SVM)	27
3 Przygotowanie danych	35
3.1 Analiza techniczna i jej wskaźniki	36
3.2 Wybór zmiennych	41
3.3 Zbiór testowy i treningowy	43
3.4 Walidacja krzyżowa	43
3.5 Ocena poprawności modelu	44
4 Wyniki	46
4.1 Drzewa decyzyjne	46
4.2 Bagging	50
4.3 Lasy losowe	55
4.4 Boosting	58
4.5 Regresja logistyczna	62
4.6 Maszyna wektorów nośnych (SVM)	66
4.7 Ocena istotności zmiennych	71
4.8 Podsumowanie	74
Zakończenie	77

Wprowadzenie

Przełom XX i XXI wieku przyniósł znaczny rozwój komputeryzacji. Dzisiaj jest ona wszechobecna i w znacznym stopniu ułatwia codzienne funkcjonowanie. Nie inaczej jest w świecie nauki. Znaczny stopień komplikacji modeli matematycznych spowodował, że praktycznie nie ma możliwości prowadzenia obliczeń bez wykorzystania komputera. Algorytmy uczenia maszynowego przenoszą rozważania na zupełnie inny poziom. Ich założeniem jest automatyczne podejmowanie optymalnych decyzji w drodze uczenia się na własnym doświadczeniu. Jednym z zakresów, w których mogą zostać wykorzystane jest dziedzina finansów i ekonomii. Na rynkach giełdowych szybkie i poprawne podejmowanie decyzji jest kluczowe. Notowania akcji charakteryzują się bardzo dużą zmiennością nawet rozpatrując dane dzień po dniu. Warto zaznaczyć, że zmienność ta jest w dodatku bardzo trudna do wyjaśnienia, a żeby podjąć właściwą decyzję należy wziąć pod uwagę nie tylko dane numeryczne, ale również wydarzenia ze świata, które mogą mieć na nie wpływ. Znaczna komplikacja problemu powoduje, że użyteczne mogą się stać algorytmy uczenia maszynowego, które na własnym doświadczeniu mogłyby podejmować odpowiednie decyzje dotyczące kupna, sprzedaży lub zachowania papierów wartościowych w portfelu. Odpowiednie oprogramowanie oczywiście nie byłoby w stanie całkowicie wykluczyć elementu ludzkiego jeśli chodzi o podejmowanie decyzji giełdowych, ale mogłoby w znaczącym stopniu ją ułatwić.

Celem pracy jest zbadanie, czy metody uczenia maszynowego mogą być użyteczne w zagadnieniach finansowych, a konkretnie do prognozowania notowań na giełdzie na podstawie ich danych historycznych oraz wskaźników analizy technicznej. Do badań zostały wykorzystane historyczne dane dotyczące indeksów giełdowych WIG20, S&P 500, DAX, Nikkei 225, BSE SENSEX oraz FTSE 100 z lat 2010 - 2022, a prognozy wykonano na podstawie modeli drzew decyzyjnych, maszyny wektorów nośnych oraz regresji logistycznej.

Struktura pracy prezentuje się następująco. W pierwszym rozdziale zaprezentowano opis literatury naukowej z zakresu wykorzystania uczenia maszynowego do prognozowania cen akcji. W drugim rozdziale opisana została metodyka badań, a także przedstawiono teoretyczne opisy modeli wybranych do estymacji. W kolejnym rozdziale za-

prezentowano sposób przygotowania danych, a także zawarto opis wskaźników analizy technicznej wykorzystanych w pracy. W czwartym rozdziale zaprezentowane zostały wyniki każdego z modeli, a także wybrano najlepszy z nich na podstawie m.in. wskazania trafności prognoz. Na końcu przedstawiono również podsumowanie wyników pracy oraz dalsze plany rozwoju.

Rozdział 1

Przegląd literatury naukowej

Metody użyte do prognozowania cen akcji nie są jedynymi, które można wykorzystać do problemu rozważanego w pracy. Obthong i in., 2020 wskazują wady i zalety poszczególnych technik uczenia maszynowego, a także wykonali przegląd literatury dotyczącej ich wykorzystania w innych zagadnieniach finansowych. Warto zauważyć, że wskazują oni nie tylko na użycie technik do prognozowania, ale również do klasteryzacji i klasyfikacji oraz zaproponowali użycie poszczególnych technik w zależności od badanych instrumentów finansowych. Szczególnie ciekawe wydaje się również zbiorcze przedstawienie wyników z kilku prac o podobnej tematyce. W samej pracy autorzy wskazują również sposoby wyboru zmiennych do modelu. Zdaniem autorów najlepiej odzwierciedlające rzeczywistość wyniki można uzyskać w przypadku wykorzystania zmiennych pochodzących zarówno z analizy technicznej oraz fundamentalnej.

Podobne zdanie zdają się podzielać Beyaz i in., 2018 w swoim artykule. Choć modele stworzone na bazie analizy fundamentalnej zdają się przewyższać te bazujące tylko na analizie technicznej, to już kombinacja obydwu podejść daje lepsze rezultaty i zmniejsza wartość RMSE (*Root-Mean-Square error*) modelu. Beyaz i in. w swoim artykule posłużyli się danymi historycznymi w celu uzyskania na ich podstawie wskaźników analizy technicznej. Oprócz tego zastosowali również predyktory pochodzące z analizy fundamentalnej jak wskaźniki makroekonomiczne, czy notowania bezpośrednich rynkowych konkurentów badanych spółek. Warto zauważyć, że w swoich badaniach autorzy pokazali również, że użycie tego samego modelu do spółek z różnych sektorów gospodarczych również wpływa na poprawność modelu (np. modele lepiej dokonywały predykcji dla spółek z sektora energetyki). Ostatecznie z opublikowanych wyników można wywnioskować, że w przypadku badanych 140 spółek z indeksu S&P 500 lepiej sprawdzała się metoda wektorów nośnych (SVM) niż sieci neuronowe (ANN).

Inne podejście do problemu predykcji cen akcji zaprezentowali Chou and Nguyen,

2018. W swoim artykule stworzyli model predykcyjny LSSVR (*Least Squares Support Vector Regression*), czyli metodę opartą na algorytmie maszyny wektorów nośnych (SVM). Dzięki temu byli w stanie zaaplikować model do cen akcji o nieliniowej charakterystyce. Opisując sam model wskazują jednocześnie jak istotne w przypadku jego wykorzystania jest odpowiednie przetestowanie hiperparametrów modelu. Jak sami nadmienili model LSSVR charakteryzuje się mniejszą złożonością obliczeniową, ale jednocześnie wymaga odpowiedniego doboru hiperparametrów. Autorzy wskazują na złożoność problemu odpowiedniego doboru hiperparametrów do modelu, a sami stworzyli autorską optymalizację opartą na algorytmie metaheurystycznym.

Patel i in., 2015 również pracowali nad modelem opartym na uczeniu maszynowym do prognozowania zmian cen akcji. W swojej pracy wykorzystali historyczne dane pochodzące z indeksów i spółek z indyjskiej giełdy papierów wartościowych, a do predykcji stworzyli 4 różne modele predykcyjne: sieć neuronową (ANN), maszynę wektorów nośnych (SVM), lasy losowe (RF) oraz metodę naiwną Bayesa. Jako predyktorów użyli wskaźników analizy technicznej obliczonych na podstawie danych historycznych. Warto odnotować, że obliczone wskaźniki przed budową modelu zostały znormalizowane do zakresu $[-1, 1]$, aby nie doprowadzić do zmniejszenia wpływu poszczególnych predyktorów w modelu. Dodatkowo autorzy zaprezentowali interesujący sposób dyskretyzacji wskaźników analizy technicznej. Z uwagi na to, że użyte predyktory mają charakter ciągły autorzy postanowili dokonać ich konwersji uzyskując w ten sposób predyktory o charakterze binarnym przyjmujące jedynie dwie wartości: 1 oraz -1. Do pomiaru poprawności modelu użyli trafności (*accuracy*) oraz wskaźnika F1 (*F-measure*). Bazując na predyktorach o charakterze ciągłym najlepsze predykcje wykazał model lasów losowych uzyskując średnią trafność ze wszystkich badanych indeksów na poziomie 0,836. Najgorsze predykcje wykazał z kolei klasyfikator naiwny Bayesa uzyskując trafność na poziomie 0,733. Jak pokazali autorzy użyta, dyskretyzacja predyktorów również wpłynęła na poprawność modeli. Na podstawie nowych zmiennych istotnie poprawili jakość prognoz w każdym z modeli - najwyższą trafność wykazał klasyfikator naiwny Bayesa (0,902), zaś najniższą model sieci neuronowej (0,867).

Z kolei Attigeri i in., 2015 do predykcji zmian cen akcji zaproponowali rozwiązanie z wykorzystaniem regresji logistycznej. Zmienną objaśnianą była różnica między danymi zamknięcia dwóch kolejnych dni, która została przekształcona na zmienną binarną. Przyjęcie zmiennej binarnej pozwoliło na wykorzystanie do modelowania regresji logistycznej. Ciekawym podejściem był jednak wybór predyktorów do modelu. Za pomocą technik *big data* gromadzili oni informacje z mediów społecznościowych, które mogły wpływać na wahania cen akcji. Następnie przypisywali każdej wiadomości efekt na wahanie akcji - pozytywny (wzrost akcji), negatywny (spadek) lub neutralny. Tak stworzo-

ną zmienną wykorzystali jako jeden z predyktorów, razem z historycznymi wartościami dotyczącymi akcji.

Selvin i in., 2017 zaprezentowali rozwiązanie bazujące tylko na różnych typach sieci neuronowych - rekurencyjnej, konwolucyjnej oraz LSTM. W swojej pracy wykorzystali dane spółek z indeksu NIFTY 50 pochodzące z indyjskiej giełdy papierów wartościowych. Autorzy porównali również wyniki dla każdej z sieci neuronowych z wynikami uzyskanymi z modelu ARIMA. W ich przypadku sieci neuronowe uzyskiwały znacznie niższe wartości RMSE w porównaniu z modelem ARIMA.

Huang i in., 2005 zastosowali metody uczenia maszynowego do predykcji zmian notowań indeksu NIKKEI 225. W pracy posłużyli się tygodniowymi danymi wspomnianego indeksu, a jako predyktory zastosowali oni wskaźniki makroekonomiczne. Do badań posłużył model SVM, a same wyniki wskaźnika trafności zostały porównane z tym samym wskaźnikiem uzyskanym z modeli: błędzenia losowego, liniowej analizy dyskryminacyjnej, kwadratowej analizy dyskryminacyjnej i propagacji wstecznej opartej na sieci neuronowej. Jak wynika z zaprezentowanych rezultatów najwyższą wartość współczynnika trafności predykcji uzyskał model SVM (73% poprawnych oszacowań zmian indeksu). Autorzy zaproponowali również kombinowany model bazujący na wszystkich zbudowanych wcześniej modelach podwyższając trafność do 75%.

Vijh i in., 2020 w swojej pracy postanowili użyć modeli uczenia maszynowego do predykcji dokładnej ceny, a nie tylko kierunku jej ruchu. Za zbiór danych posłużyły notowania dzienne 5 spółek z lat 2009-2019 - Nike, Goldman Sachs, Johnson and Johnson, Pfizer, JP Morgan Chase and Co. Przed budową modelu przygotowali oni również odpowiednio zmienne. Jako predyktory wybrali oni średnią ruchomą z 7, 14 i 21 dni, odchylenie standardowe z 7 dni, a także różnicę między najwyższą ceną z dnia i najniższą oraz różnicę między ceną zamknięcia i otwarcia. Widać więc, że autorzy oparli swoje modele tylko o zmienne wynikające z analizy technicznej. Badania zostały przeprowadzone na modelu sieci neuronowej, a także lasów losowych, a do testowania wyników zostały wykorzystane wartości błędów RMSE, MAPE (*Mean Absolute Percentage Error*) oraz MBE (*Mean Bias Error*). Wyniki, które uzyskali pokazują, że model sieci neuronowej lepiej poradził sobie z problemem niż model lasów losowych.

Ciekawe wnioski można również wysnuć z pracy Ampomah i in., 2020. Autorzy również skupili się na budowie modelu do przewidywania stóp zwrotu dzień po dniu, jednak testowali oni jedynie modele drzew decyzyjnych - lasy losowe (RF), AdaBoost, XGBoost, Bagging, Extra Trees oraz łączony model oparty na średniej ważonej ze wszystkich modeli. Badania zostały wykonane na historycznych notowaniach indeksów i spółek z nowojorskiej i indyjskiej giełdy papierów wartościowych. Wśród nich znalazły się między innymi S&P 500, Microsoft Corporation, Tata Steel Ltd. oraz Dow Jones Industrial

Average. Również w tej pracy autorzy zdecydowali się zastosować walidację krzyżową na danych treningowych, żeby lepiej określić zdolności przewidywania każdego z modeli. Warto zaznaczyć, że całe badania oparte były na wcześniej obliczonych wskaźnikach technicznych. Łącznie autorzy zastosowali 40 wskaźników. Taka liczba zmiennych znacząco komplikuje budowę modelu, więc zastosowana została również metoda PCA (Principal Component Analysis) do redukcji wymiarowości danych. Autorzy zastosowali również kilka kryteriów oceny modelu. Tak jak w poprzednich pracach modele oceniane były na podstawie wskaźnika trafności, F1, czułości, specyficzności, a także krzywej ROC i wartości AUC. Ostatecznie na podstawie wyników każdego z modeli autorom udało się ustalić, że najlepszym modelem był Extra Tress z wskaźnikiem trafności 0.8375. Warto jednak zauważyć, że każdy z testowanych modeli uzyskał wartość tego wskaźnika większą niż 0,8, co może świadczyć o bardzo wysokich zdolnościach predykcyjnych modeli drzew decyzyjnych do badanego problemu.

Podsumowując wszystkie przedstawione prace, można zauważyć mnogość podejść do problemu prognozowania cen akcji. Występują podejścia bazujące na elementach analizy fundamentalnej, jak również analizy technicznej. Również w kwestii wyboru zmiennych do modelu istnieje kilka podejść. Na bazie analizy technicznej autorzy zdecydowali się zarówno na użycie surowych danych historycznych (dotyczących kilku wcześniejszych dni), jak również użycie wskaźników analizy technicznej. Jeśli chodzi o elementy analizy fundamentalnej, to autorzy wykorzystywali m.in. wskaźniki makroekonomiczne (wysokość stóp procentowych), czy notowania rynkowych konkurentów badanych spółek. Choć wszystkie przedstawione publikacje opierały się na założeniach uczenia maszynowego, to trudno wyłonić jedyny słuszny model do prognozowania zmian cen akcji. Autorzy wskazują zarówno na modele drzew decyzyjnych (m.in. lasów losowych, bagging, AdaBoost), regresji logistycznej, czy wariantów modelu maszyny wektorów nośnych. Znaczne zróżnicowanie podejść do problemu świadczy tylko o braku idealnego rozwiązania i jego złożoności, a także pozostawia pole do dalszych badań w tej materii.

Rozdział 2

Metodyka badań

Założeniem pracy jest zbudowanie modelu do przewidywania ruchów cen akcji na podstawie danych historycznych i zmiennych przygotowanych za ich pomocą. W badaniach posłużono się notowaniami historycznymi indeksów giełdowych WIG20, S&P 500, DAX, Nikkei 225, BSE SENSEX oraz FTSE 100. Całe badania oparto o założenia analizy technicznej bazując na danych dziennych wymienionych indeksów w zakresie ceny otwarcia, zamknięcia, najwyższej i najniższej ceny danego dnia oraz wolumenu. Na podstawie tych danych przygotowano wskaźniki analizy technicznej, które posłużyły następnie jako zmienne w modelach uczenia maszynowego. Same ruchy cen akcji rozprezentowane były przez zmienną binarną, która przyjmowała wartość 0, gdy odnotowano spadek ceny w danym dniu w porównaniu z dniem poprzednim i 1, gdy odnotowano wzrost lub brak zmiany. Dane dotyczące każdego indeksu podzielono następnie na zbiór treningowy oraz testowy. W celu dokładnego określenia zdolności prognostycznych modelu na zbiorze treningowym zastosowano również walidację krzyżową. Na tak przygotowanych danych wykonano prognozy. Do badań posłużono się modelami lasów losowych, maszyny wektorów nośnych oraz regresji logistycznej. Do wyłonienia najlepszych modeli zastosowano również metody strojenia hiperparametrów modeli. Na koniec zdolności prognostyczne modeli zostały porównane z sobą na podstawie wskaźników trafności, F1, a także wskazań krzywych ROC i wartości AUC.

Przygotowanie danych, budowa modelu i wykonanie prognoz zostało przeprowadzone w języku Python (wersja 3.9), a dane do pracy zostały pobrane ze strony stooq.pl.¹

¹ *Stooq*, [dostęp: 17.06.2022], <https://stooq.pl>

2.1. Język Python

Język Python jest interpretowanym językiem programowania wysokiego poziomu. Ze względu na przejrzystość i zwięzłość swojej składni, a także na możliwości, które oferuje jest szeroko wykorzystywany w nauce i finansach². Na liście bibliotek języka Python można znaleźć biblioteki takie jak pandas, numpy, scikit-learn, czy TA. Wszystkie wymienione biblioteki zostały użyte w niniejszej pracy i to na nich oparto wszystkie obliczenia. Biblioteka pandas została użyta do przechowania i wykonywania operacji na bazie danych zawierających notowania akcji. Biblioteka numpy pozwoliła na wykonanie operacji matematycznych na tabelach danych. Przygotowane dane posłużyły do zbudowania modeli uczenia maszynowego. W tym celu użyto biblioteki scikit-learn, która oprócz możliwości tworzenia modeli uczenia maszynowego oferuje również przeprowadzenie dokładnych analiz końcowego modelu i predykcji. Do przygotowania zmiennych posłużyła z kolei biblioteka TA, która oferuje szeroką gamę wbudowanych funkcji do obliczania wskaźników analizy technicznej.

2.2. Uczenie maszynowe

Termin "uczenie maszynowe" został użyty po raz pierwszy w 1959 roku przez Arthura Samuela. Sam termin był definiowany na wiele sposobów w różnych dziełach. Jego istotą jest umożliwienie komputerom myślenia i własnego podejmowania decyzji [Alzubi i in., 2018]. Samo uczenie maszynowe polega na zmienianiu decyzji przez komputer w celu poprawienia dokładności, która mierzona jest jako liczba poprawnych decyzji w stosunku do wszystkich decyzji.

Dzisiaj uczenie maszynowe jest wykorzystywane w wielu obszarach, począwszy od rozwiązywania problemów matematycznych, a skończywszy na generacji silników gier komputerowych i planszowych. Powszechnie znanym przykładem był wygenerowany za pomocą uczenia maszynowego silnik szachowy Deep Blue, który w 1997 roku był w stanie pokonać arcymistrza szachowego Garriego Kasparova.

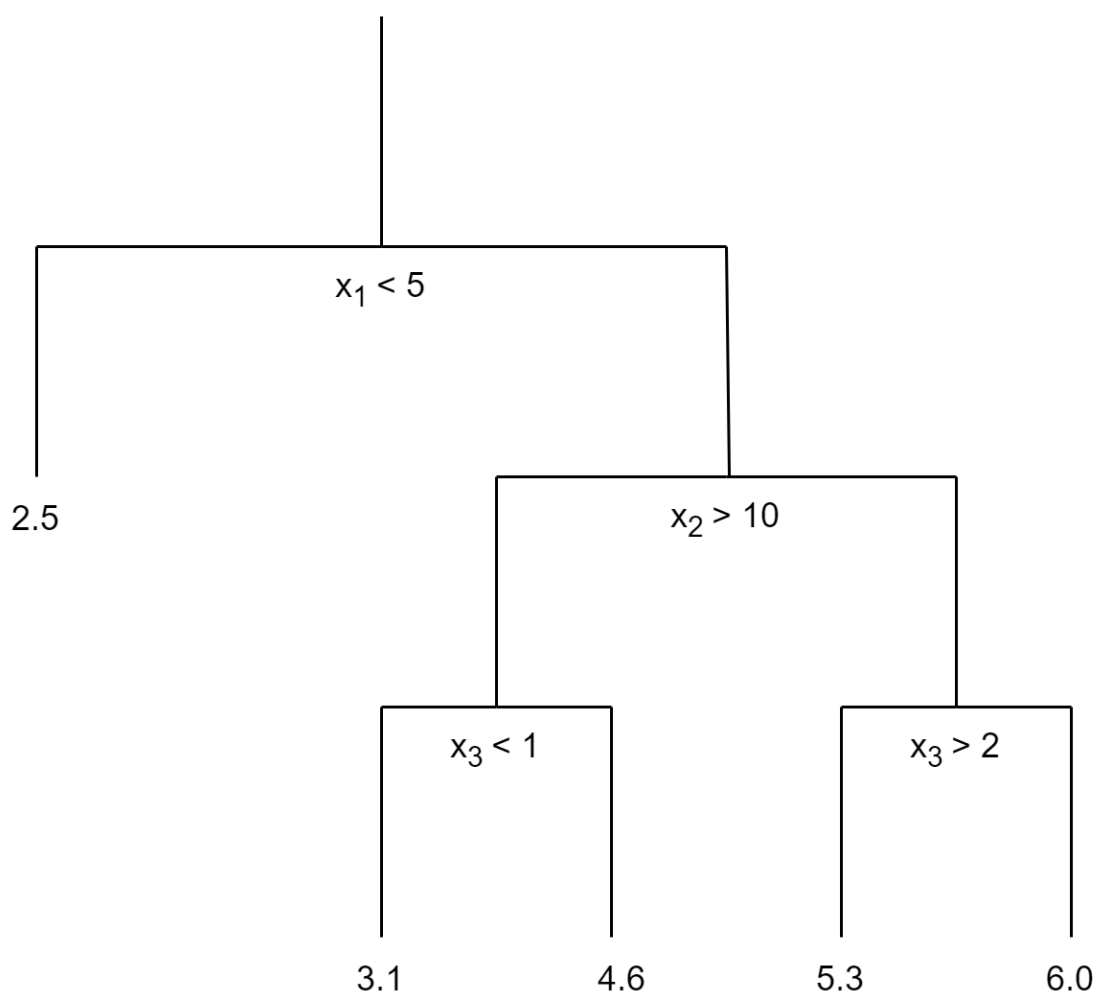
Do popularności technik uczenia maszynowego przyczynia się również ich dostępność. Wiele języków programowania i samych programów oferuje wbudowane narzędzia do tworzenia modeli. Również nietrudno natrafić na publikacje i materiały szkoleniowe z tej dziedziny, co znacznie ułatwia naukę i zastosowanie uczenia maszynowego w praktycznych przykładach.

²What is Python? Executive Summary, [dostęp: 19.01.2022], <https://www.python.org/doc/essays/blurb/>

Uczenie maszynowe może być również pomocne do rozwiązywania zagadnień finansowych. Jak pokazują liczne publikacje z powodzeniem można używać jego algorytmów do rozwiązywania problemów regresyjnych i klasyfikacyjnych.

2.3. Drzewa decyzyjne

Drzewa decyzyjne (DT - ang. Decision Trees) to przykład nadzorowanego modelu uczenia maszynowego. Jest to powszechnie używany model ze względu na swoją uniwersalność - ma zastosowanie zarówno w rozwiązywaniu zagadnień wymagających modeli regresyjnych, ale także do rozwiązywania problemów klasyfikacyjnych. Istotą modelu DT jest podział zbioru danych na regiony zgodnie z regułami określonymi na podstawie zmiennych objaśniających [James i in., 2014]. Jako przykład może posłużyć proste drzewo decyzyjne przedstawione na Rys. 2.1.



Rys. 2.1. Przykład drzewa decyzyjnego

Źródło: opracowanie własne

Z graficznej wizualizacji drzewa możemy odczytać, że zbiór danych został podzielony

na 5 regionów za pomocą warunków nałożonych na zmienne x_1 , x_2 oraz x_3 . Poniżej wypisano wszystkie z nich:

- Region 1: $x_1 < 5$
- Region 2: $x_1 \geq 5 \wedge x_2 > 10 \wedge x_3 < 1$
- Region 3: $x_1 \geq 5 \wedge x_2 > 10 \wedge x_3 \geq 1$
- Region 4: $x_1 \geq 5 \wedge x_2 \leq 10 \wedge x_3 > 2$
- Region 5: $x_1 \geq 5 \wedge x_2 \leq 10 \wedge x_3 \leq 2$

Warto zwrócić uwagę na reguły określające poszczególne regiony. Region 1 został wyodrębniony po prostym podziale zmiennych objaśniających ze względu na wartość tylko jednej zmiennej x_1 . Pozostałe 4 regiony są bardziej złożone, bo do ich wyodrębnienia użyto 3 warunków na zmienne x_1 , x_2 oraz x_3 . Nietrudno wyobrazić sobie, że w rzeczywistości drzewa decyzyjne są często znacznie bardziej skomplikowane, a wyodrębnienie poszczególnych regionów wymaga nałożenia reguł na wiele zmiennych. Sama graficzna prezentacja regionów jest również prosta. W przypadku, gdy zbiór danych jest podzielony na regiony tylko za pomocą reguł nałożonych na 2 zmienne każdy region będzie reprezentowany prostokątem na 2-wymiarowej przestrzeni. Oczywiście w przypadku, gdy zbiór danych zostanie rozdzielony regułami na 3 regiony reprezentacja regionów również musi być przedstawiona w 3-wymiarach itd.

Skojarzenie modelu z drzewem jest zasadne. Graficzna prezentacja nasuwa dokładnie taką interpretację. Korzeń to miejsce zawierające wszystkie możliwe decyzje. Każda reguła tworzy nowy węzeł (rozgałęzienie), a na końcu gałęzi znajdują się przewidziane w modelu wartości (liście).

Celem stosowania modelu DT jest predykcja konkretnych wartości zmiennej objaśnianej. W pokazanym wcześniej przykładzie uzyskaliśmy 5 wartości zmiennej - każda odpowiada jednemu regionowi. Warto zastanowić się jak właściwie model wylicza te wartości. Jak już zostało powiedziane modele DT mogą być wykorzystywane do problemów regresyjnych (przewidywanie konkretnych wartości zmiennej, np. cen akcji) i klasyfikacyjnych (przynależność do danej grupy reprezentowana w danych przez konkretną wartość całkowitą). W badanym przykładzie mamy do czynienia z modelem regresyjnym. Konkretnie przewidziane wartości są w nim średnimi wartościami ze zbioru treningowego zmiennej objaśnianej w obrębie danego regionu. W przypadku zagadnień klasyfikacyjnych następuje modyfikacja. W takim wypadku model zwraca najczęściej występującą wartość, a nie średnią.

W kontekście drzew decyzyjnych kluczowym zagadnieniem jest zrozumienie jak dochodzi do podziału drzewa na każdym etapie. Zaczniemy od drzew regresyjnych.

Głównym parametrem regresyjnego modelu DT jest kryterium podziału zbioru danych. Najczęściej używanym w tym kontekście jest średni błąd kwadratowy (MSE - *mean squared error*), średni absolutny błąd (MAE - *mean absolute error*) oraz miara zanieczyszczenia Poissona. W przypadku kryterium MSE po dokonaniu podziału wyliczana jest średnia wartość zmiennej objaśnianej na zbiorze treningowym zgodnie ze wzorem

$$\bar{y}_m = \frac{1}{n_m} \sum_{y \in Q_m} y, \quad (2.1)$$

gdzie Q_m jest zawężonym w podziale regionem, n_m liczbą obserwacji w regionie, a y wartością zmiennej objaśnianej dla danej obserwacji w regionie. Korzystając z wyliczonej średniej jesteśmy w stanie policzyć w kolejnym kroku wartość kryterium MSE

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} (y - \bar{y}_m)^2. \quad (2.2)$$

Następnie wybierany jest ten podział, dla którego wartość MSE jest najmniejsza. Trzeba tutaj zaznaczyć, że model w tym przypadku korzysta z pewnego uproszczenia. Skoro model zawsze wybiera taki podział, który gwarantuje najmniejszą wartość MSE to automatycznie rezygnuje z tych, które dopiero w kolejnych krokach dają lepsze dopasowanie, ale za to w momencie dzielenia mają wyższą wartość MSE. Nie ma w końcu żadnej gwarancji, że taki wybór zawsze daje najlepsze wyniki [James i in., 2014].

W przypadku zastosowania minimalizacji MAE jako kryterium wyboru reguły podziału zamiast średniej jako punkt wyjścia używa się mediany obliczonej na obserwacjach z odpowiadających regionów. Na tej podstawie obliczana jest wartość kryterium MAE

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} |y - \text{mediana}(y)_m|. \quad (2.3)$$

Podobnie jak dla MSE wybieramy taki podział, dla którego kryterium wyboru jest najmniejsze.

Kolejnym kryterium jakie można zastosować w przypadku modelu regresyjnego DT jest minimalizacja odchylenia dla rozkładu Poissona. W tym przypadku należy posłużyć się wzorem

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} (y \log \frac{y}{\bar{y}_m} - y + \bar{y}_m). \quad (2.4)$$

Wszystkie trzy przedstawione kryteria mogą być użyte do modeli regresyjnych. Inaczej wygląda to w przypadku modeli klasyfikacyjnych DT. Nie możemy dla nich użyć po-

znanych już kryteriów, ponieważ przewidywaną wartością jest przynależność do konkretnej klasy reprezentowanej przez liczbę całkowitą, a nie średnia, czy mediana tak jak w przypadku modeli regresyjnych. Z tego powodu przewidziana wartość będzie wartością klasy z największą częstotliwością wystąpienia w danym regionie na zbiorze treningowym, a najczęściej używanymi kryteriami wskaźnik Giniego oraz entropii. W przypadku kryterium Giniego wartość obliczana jest wzorem

$$H(Q_m) = \sum_k^K \hat{p}_{m,k}(1 - \hat{p}_{m,k}) = 1 - \sum_k^K \hat{p}_{m,k}^2, \quad (2.5)$$

gdzie $\hat{p}_{m,k}$ jest liczbą wystąpień klasy k w regionie Q_m w stosunku do wszystkich obserwacji. Łatwo zauważyć, że faworyzowanymi podziałami będą te, które jednoznacznie rozdzielają zbiór treningowy - im liczniejsza dana klasa w regionie tym wartość wskaźnika bliższa 0 [James i in., 2014], [Geron, 2019].

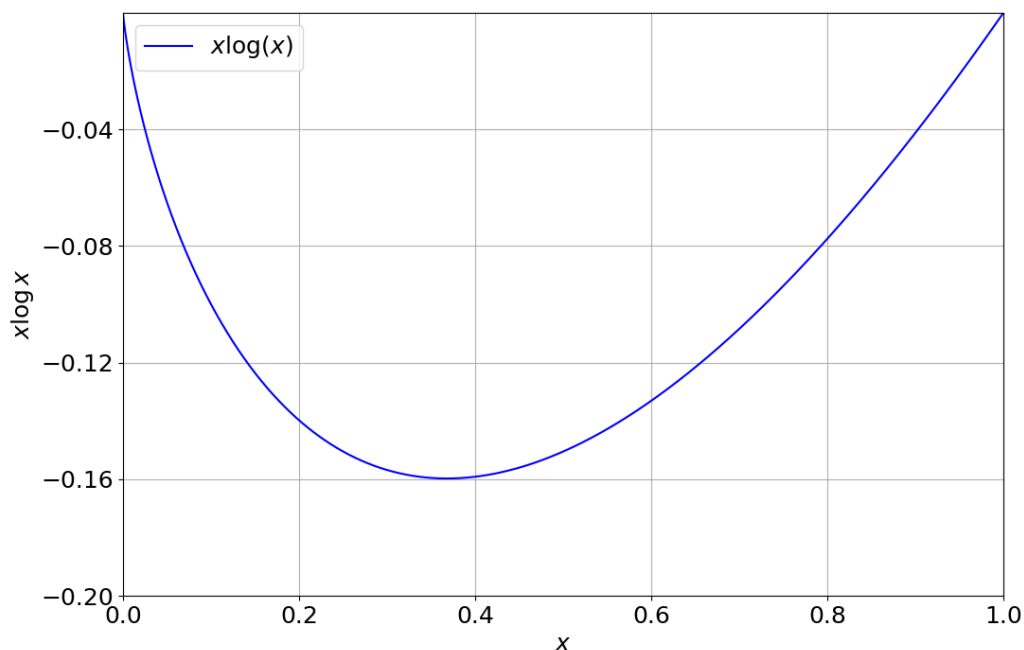
Alternatywą kryterium Giniego może być wskaźnik entropii. Jego interpretacja ma głębokie korzenie w fizyce. Fizyczną interpretacją byłaby miara uporządkowania cząstek - im niższa wartość entropii tym większe uporządkowanie. W kontekście szeroko pojętej informatyki pojęcie entropii ma zastosowanie w teorii informacji, a konkretnie we wzorze Shannona i reprezentuje zawartość informacji w wiadomości [Shannon, 1948], [Mazur, 1970]. Samo kryterium można przedstawić wzorem

$$H(Q_m) = - \sum_k^K \hat{p}_{m,k} \log \hat{p}_{m,k}. \quad (2.6)$$

Żeby dokładniej zrozumieć jak zachowuje się wskaźnik entropii spójrzmy na wykres funkcji $f(x) = x \log(x)$ (Rys. 2.2).

Z wykresu można odczytać, że dla wartości x bliskich 0 lub 1 $f(x) \approx 0$. To cenna informacja w kontekście interpretacji kryterium entropii, bo dowodzi, że dla wartości $\hat{p}_{m,k}$ bliskich 0 lub 1 jego wartość jest minimalna. Warto jednak zwrócić uwagę, że sama funkcja nie jest już symetryczna tak jak przy kryterium Giniego, a faworyzowane są szczególnie te podziały, w których występuje najwięcej zliczeń jednej klasy.

Niestety pomimo wielu zalet modeli DT trzeba zaznaczyć ich dużą wadę, czyli tendencję do przetrenowywania na dużych zbiorach danych z dużą liczbą zmiennych. W celu zmniejszenia wpływu tego zjawiska na model można zastosować metodę przycinania drzewa. W założeniach chcemy ograniczyć rozrost drzewa przez wprowadzenie nowego, kontrolującego to zjawisko parametru. Metoda przycinania jest realizowana w modelu przez wyliczenie kryterium wyboru dla przyciętych drzew, a następnie wy-



Rys. 2.2. Wykres funkcji $f(x) = x \log(x)$ dla $x \in [0, 1]$

Źródło: opracowanie własne

branie tego, który ma najmniejszą wartość tego kryterium $R_\alpha(T)$ ³. Realizowane jest to poniższym wzorem

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}|, \quad (2.7)$$

gdzie $R(T)$ jest całkowitą miarą zanieczyszczenia wszystkich węzłów końcowych (liści) w drzewie T , $|\tilde{T}|$ jest liczbą węzłów końcowych w drzewie, α wprowadzonym nieujemnym parametrem kontrolującym przycinanie drzewa. Interpretacja jest prosta - wraz ze wzrostem wartości α wartość całego wyrażenia $\alpha|\tilde{T}|$ staje się coraz większa, co prowadzi się do tego, że wartość $R_\alpha(T)$ jest większa dla drzew mających wiele węzłów końcowych.

W kontekście drzew decyzyjnych zaprezentowane zostały do tej pory dwa parametry regulujące model. W praktyce jest ich znacznie więcej. Tak jak to już zostało wspomniane model DT ma tendencję do przetrenowywania, dlatego warto przetestować dostępne hiperparametry mogące zniwelować ten efekt jednocześnie obniżając złożoność modelu. W tabeli 2.1 zostały zaprezentowane hiperparametry modelu DT.

³Decision Trees, [dostęp: 03.05.2023], <https://scikit-learn.org/stable/modules/tree.html>

Tabela 2.1. Wykaz hiperparametrów dostępnych w modelu drzew decyzyjnych

Parametr	Opis
criterion	Kryterium podziału węzła: Gini, entropia dla modelu klasyfikacyjnego; MSE, MAE, Poisson dla modelu regresyjnego
max_depth	Maksymalna głębokość drzewa
min_samples_split	Minimalna liczba obserwacji wymagana do przeprowadzenia podziału węzła
min_samples_leaf	Minimalna liczba obserwacji wymagana do uznania węzła za końcowy (liść)
max_leaf_nodes	Maksymalna liczba węzłów końcowych w rozpatrywanym drzewie
max_features	Maksymalna liczba zmiennych do przeprowadzenia podziału
ccp_alpha	Kryterium przycinania drzewa
min_impurity_decrease	Graniczna minimalna wartość zanieczyszczenia wymagana do przeprowadzenia podziału

Źródło: lista przygotowana na podstawie dokumentacji technicznej dotyczącej drzew decyzyjnych z biblioteki scikit-learn

Jak widać model DT jest bardzo elastyczny jeśli chodzi o jego strojenie. Mamy możliwość kontrolowania liczby obserwacji znajdujących się w liściu lub wymaganych do przeprowadzenia podziału, czy maksymalnej liczby węzłów i głębokości drzewa. Wszystkie te parametry z powodzeniem mogą zniwelować problem przetrenowania modelu oraz ograniczyć jego złożoność, co z kolei może ułatwić interpretację drzewa - im prostsze drzewo tym łatwiejsze w zrozumieniu [James i in., 2014].

Jak zostało zaprezentowane w tym rozdziale modele drzew decyzyjnych to bardzo elastyczne struktury o szerokim spektrum zastosowań. Oferują przy tym również bardzo przejrzystą interpretację oraz graficzną wizualizację rozwiązania. Trzeba jednak zauważyć, że ich prostota sprawia, że zwykle nie uzyskują najlepszych wyników jeśli chodzi o trafność prognoz w porównaniu z innymi modelami. W następnych podrozdziałach zostaną zaprezentowane inne modele wykorzystujące ideę drzew decyzyjnych.

2.3.1. Bagging

Bagging bazuje na przedstawionych w poprzednim rozdziale drzewach decyzyjnych, ale wykorzystuje również wielokrotną estymację modelu na próbkach wybranych ze zbioru treningowego zgodnie z ideą metod samowspornych (ang. *bootstrapping*). Początkowym krokiem jest w tym przypadku wybór podzbioru obserwacji ze zbioru treningowego. Bazując na małej próbce danych estymowany jest model drzew decyzyjnych. Podane dwa kroki są następnie wykonywane wielokrotnie, pamiętając o tym, że zawsze podzbiór zmiennych wybieramy ze zwracaniem. Na koniec w przypadku drzew regresyjnych wartości są uśredniane, a wynik zostaje przyjęty jako wartość przewidziana przez model. Dla drzew klasyfikacyjnych uśrednianie zostaje zastąpione głosowaniem - jako predykcja wybrana zostaje ta klasa, która została przewidziana najwięcej razy przez modele [Breiman, 1996].

Aby zrozumieć jaka korzyść wynika ze stosowania wielokrotnej estymacji modelu na podzbiorze danych musimy odnieść się do wariancji pojedynczej zmiennej. Załóżmy, że dysponujemy zbiorem n niezależnych obserwacji - x_1, x_2, \dots, x_n , z których każda ma wariancję $Var(x_i) = \sigma^2$. Chcąc wyliczyć wariancję wartości średniej uzyskamy wzór 2.8⁴

$$Var(\bar{x}) = \frac{\sigma^2}{n}. \quad (2.8)$$

Ze wzoru można wywnioskować, że wariancja wartości średniej z n obserwacji jest n -krotnie mniejsza w porównaniu do pojedynczej obserwacji. Oczywiście przy okazji tego twierdzenia trzeba zauważyć, że wzór ten dotyczy niezależnych obserwacji. W przypadku losowania zbioru obserwacji ze zwracaniem oczywiste jest to, że próby te nie zawsze będą niezależne, więc i wariancja wartości średniej nie będzie dokładnie opisywana tym wzorem, ale nadal powinna być mniejsza niż dla pojedynczego modelu [James i in., 2014]. Podsumowując uśrednianie wyników prowadzi do zmniejszenia wariancji przewidywanych wartości.

Skoro zakładamy, że dla każdego modelu próba wybierana jest oddzielnie z zwracaniem, to naturalne jest powielanie wybranych obserwacji, ale działa to też w drugą stronę i niektóre wartości nie zostaną wybrane wcale. Takie obserwacje nazywane są pozatreningowymi (OOB - *out-of-bag*) [Geron, 2019]. Pomimo, że nie są one użyteczne dla samych modeli to nadal można zrobić z nich pożytek i potraktować je jako nowy zbiór testowy. Wbudowane rozwiązanie w języku Python daje możliwość wyliczenia błędu MSE dla obserwacji OOB. Może to się wydać niepotrzebne skoro i tak dysponu-

⁴Wariancja, [dostęp 04.05.2023], <https://en.wikipedia.org/wiki/Variance>

jemy przygotowanym zbiorem testowym, ale dla dużych zbiorów danych wartość MSE OOB z powodzeniem może być wykorzystana jako ekwiwalent dla walidacji krzyżowej. Zyskiem jest w tym wypadku zaoszczędzony czas potrzebny do dodatkowej estymacji modelu [James i in., 2014].

Podsumowując korzystając z modelu bagging zmniejszamy wariancję zmiennej objaśnianej względem standardowego modelu drzewa decyzyjnego. Kluczowymi parametrami wydają się z kolei liczebność próby wyciągniętej ze zbioru treningowego dla pojedynczego modelu oraz liczba drzew potrzebnych do wyestymowania modelu dającego najlepsze prognozy w rozsądnym czasie.

2.3.2. Lasy losowe

Kolejnym modelem wykorzystującym ideę drzew decyzyjnych jest model lasów losowych (RF - *Random Forest*). W założeniach model ten jest bardzo podobny do baggingu, jednak wprowadza dodatkowy parametr kontrolujący dobór zmiennych. Mamy więc do czynienia nie tyle z samym wyborem mniejszej próby jako zbioru treningowego, ale również z samym ograniczeniem liczby zmiennych w modelu. Można się zastanawiać jakie korzyści z tego płyną. W kontekście modelu bagging rozpatrywaliśmy dobór małego zbioru treningowego wybieranego metodą bootstrappingu. Jego największą wadą było możliwe duplikowanie obserwacji dla różnych modeli, co skutkuje tym, że uzyskane drzewa nie były całkowicie niezależne od siebie. Wprowadzenie losowego doboru zmiennych, na podstawie których ma zostać zbudowany model gwarantuje mniejszą zależność między modelami [James i in., 2014], [Breiman, 2001]. Nie jest to jednak jedyna zaleta. Mając do czynienia ze zbiorem, który zawiera wiele zmiennych objaśniających można się spodziewać, że jedno zmienne będą wносить znacznie większy wkład (będą istotniejsze) dla modelu niż inne. Oznacza to tyle, że model nie zawierający mechanizmu ograniczania zmiennych będzie zawsze opierał się na bardziej istotnych zmiennych, co znacząco zmniejsza zróżnicowanie drzew decyzyjnych [James i in., 2014].

Jeśli założyć, że nie nadajemy żadnego warunku na wybór zmiennych do modelu RF to szybko można dojść do wniosku, że tak naprawdę stosujemy model bagging. Sam problem doboru optymalnej liczby zmiennych jest ciekawym zagadnieniem i nie ma jednego ogólnego podejścia. Breiman, 2001 argumentuje, że optymalna liczba parametrów może być określona jako $\log_2(m + 1)$, gdzie m to liczba wszystkich zmiennych w zbiorze. Hastie i in., 2009 sugerują z kolei, że liczba optymalnych parametrów wynosi \sqrt{m} i taką wartość domyślnie przyjmuje klasyfikacyjny model RF w pakiecie scikit-learn⁵.

⁵*Random Forest Classifier*, [dostęp: 04.05.2023], <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier>

W obydwu powyższych publikacjach można też znaleźć argumentację, że liczba zmiennych powinna być po prostu niska - 1 lub 2 zmienne.

2.3.3. Boosting

Niezwykle silnymi modelami bazującymi na drzewach decyzyjnych są modele wzmacniające (*boosting*). W ich przypadku modyfikacja polega na wielokrotnym sekwencyjnym tworzeniu nowych drzew decyzyjnych, z których każde niejako uczy się na błędach poprzednich. Prześledźmy cały proces na bazie algorytmu AdaBoost dla problemu klasyfikacji. Cały proces rozpoczyna się już na etapie tworzenia zbioru treningowego. Każdej obserwacji ze zbioru uczącego przypisywana jest taka sama waga zgodnie ze wzorem

$$w_i = \frac{1}{N}, i = 1, 2, \dots, N, \quad (2.9)$$

gdzie N jest liczbą obserwacji w zbiorze treningowym. Następnie ze zbioru wybierana jest reprezentacja obserwacji, na których podstawie budowane jest pierwsze drzewo decyzyjne. Na bazie przewidzianych przez model wartości zmiennej objaśnianej wyliczana jest na całym zbiorze uczącym wartość błędu prognozy klasyfikatora zgodnie ze wzorem

$$err_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N N w_i}. \quad (2.10)$$

Funkcja $I(y_i \neq G_m(x_i))$ użyta do wyliczenia błędu jest po prostu liczbą zliczeń błędnych prognoz. W naszym przypadku y_i to rzeczywista wartość, a $G_m(x_i)$ to wartość przewidziana przez model. Na podstawie wyliczonej wartości err_m wyliczana jest następnie waga dla całego drzewa zgodnie ze wzorem

$$\alpha_m = \log\left(\frac{1 - err_m}{err_m}\right). \quad (2.11)$$

Waga ta zostanie użyta na końcu przy generowaniu finalnej wersji klasyfikatora. Na koniec wagi wszystkich obserwacji zostają zaktualizowane zgodnie ze wzorem

$$w_i \leftarrow \exp(\alpha_m I(y_i \neq G_m(x_i))), i = 1, 2, \dots, N. \quad (2.12)$$

Tutaj kończy się pierwsza iteracja. Kolejna iteracja znowu rozpoczyna się od wyboru próby ze zbioru treningowego. Tym razem obserwacje z większą wagą mają naturalnie większą szansę na wybranie. Dzięki temu model jest w stanie "poprawiać" błędy

poprzedniego drzewa. Kolejne kroki przebiegają analogicznie. Całość jest powtarzana do czasu osiągnięcia założonej liczby iteracji M . Na koniec, mając już zbudowane M drzew decyzyjnych tworzony jest końcowy klasyfikator

$$G(x) = \text{sgn} \left[\sum_{m=1}^M \alpha_m G_m(x) \right]. \quad (2.13)$$

Tak jak to już zostało zapowiedziane drzewa są różnicowane między sobą wartością parametru α . Oczywiście drzewa, które uzyskały najmniejszą wartość err mają przypisaną większą wagę od pozostałych.

Zaprezentowany algorytm AdaBoost jest tak naprawdę uproszczoną wersją mającą zastosowanie, gdy zmienna objaśniana przyjmuje tylko dwie wartości. Rozszerzoną wersję tego algorytmu zaprezentowali Friedman i in., 2000. Wprowadzili oni modyfikację wartości przewidywanej. Zamiast przyporządkowywać klasę danej obserwacji model szacuje tylko prawdopodobieństwo przynależności. Dzięki temu w nowej postaci AdaBoost może być użyty do zmiennych przyjmujących więcej niż 2 wartości. Użyteczność modeli boosting nie ogranicza się tylko do problemów klasyfikacyjnych. Mogą być one z powodzeniem stosowane również jako modele regresyjne.

Na koniec warto przyjrzeć się jak można stroić model AdaBoost. Kluczowe są 3 parametry. Pierwszym z nich jest liczba iteracji, czyli liczba drzew jaka zostanie wygenerowana, na podstawie których zostanie stworzony końcowy model. Niestety inaczej niż bagging i RF, stosując modele boosting istnieje szansa przetrenowania. Zbyt duża liczba drzew może prowadzić do zbytniego dopasowania modelu do danych. Kolejnym parametrem jest liczba węzłów w drzewach decyzyjnych. James i in., 2014 sugerują stosowanie mało skomplikowanych drzew, a wręcz takich zawierających tylko jeden podział. Ostatnim z parametrów jest współczynnik uczenia (ang. shrinkage parameter). Jest to współczynnik, który dodatkowo można nałożyć na zmienną α . Wtedy obliczana jest ona wzorem

$$\alpha_m = \lambda \log \left(\frac{1 - err_m}{err_m} \right). \quad (2.14)$$

λ przyjmuje wartość większą od 0 i dzięki niej można kontrolować szybkość poprawy modelu. Zakładając, że $\lambda > 1$ to naturalnie wyliczone wagi w każdej iteracji są większe. Uzyskujemy w ten sposób zbiór danych, w których zmienne błędnie przewidziane przez model będą jeszcze częściej wybierane do kolejnych iteracji. Niestety w parze z tym pojawiają się wady. Model szybciej będzie się przetrenowywał, skoro już jego wersja z domyślną wartością $\lambda = 1$ wykazywała takie skłonności. Pomocą może być zmniejszenie liczby iteracji, ale to też nie zawsze będzie dobrym pomysłem, bo te same

drzewa mają decydować o kształcie końcowego modelu. Dokładnie odwrotną sytuację uzyskujemy dla $\lambda < 1$. Model co prawda wolniej będzie się poprawiał, między poszczególnymi iteracjami, ale zwiększenie ich liczby będzie korzystnie wpływać na końcową postać modelu - więcej drzew to mniejsza szansa na błędną prognozę. James i in., 2014 proponują początkowo użyć wartości 0.001 lub 0.01 jako λ .

2.4. Regresja logistyczna

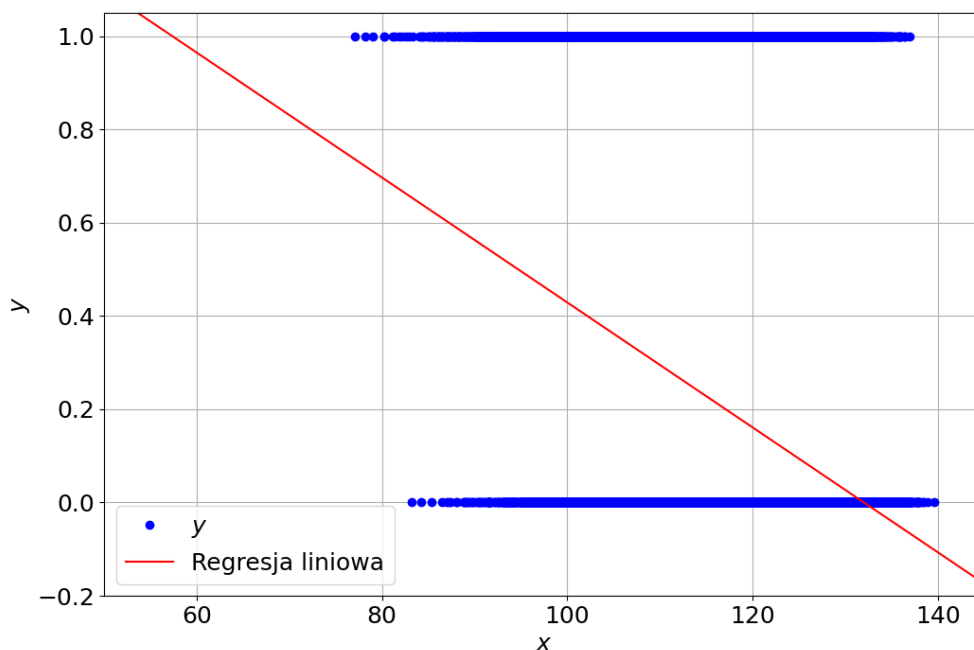
Niniejsza praca powstała z myślą o predykcji zmian cen akcji. Mamy więc do czynienia z problemem klasyfikacyjnym, w którym zmienna objaśniana będzie przyjmować tylko dwie wartości - 0, czyli spadek ceny w danym dniu i 1, czyli jej wzrost lub taka sama wartość jak dnia poprzedniego. Wymusza to użycie tylko modeli mających zastosowanie do klasyfikacji lub takich, które uniwersalnie mogą być użyte również do regresji (jak drzewa decyzyjne opisane w rozdziale 2.3). Z pewnością ciekawym modelem spełniającym to kryterium jest regresja logistyczna. Sama nazwa modelu może być myląca, ale służy ona właśnie do rozwiązywania problemów klasyfikacji. Co więcej jej głównym zastosowaniem jest prognozowanie zmiennych binarnych, więc można powiedzieć, że idealnie pasuje do badanego tematu.

Aby lepiej zrozumieć ideę regresji logistycznej musimy jednak cofnąć się do regresji liniowej. Na potrzeby rozważań założmy, że posiadamy zbiór danych składający się z dwóch zmiennych - objaśniającej x oraz objaśnianej y . Dodatkowo zmienna x jest zmienną ciągłą, a y binarną. Chcąc wyestymować model regresji liniowej na takich danych miałby on postać

$$y = p(x) = P(y = 1|x) = \beta_0 + \beta_1 x. \quad (2.15)$$

We wzorze celowo y został zastąpiony prawdopodobieństwem zdarzenia $y = 1$. Regresja liniowa nie prognozuje przynależności do danej klasy, a jedynie wartość z przedziału $(-\infty, +\infty)$. Jeżeli prognozowana wartość jest większa niż 0.5 to przyjmujemy wtedy, że model przewiduje dla danej obserwacji przynależność do klasy 1 i to właśnie rozumiemy jako prawdopodobieństwo tego zdarzenia. Trzeba jednak zauważyć pewne niebezpieczeństwo z tego wynikające. Spójrzmy na wykres modelu regresji liniowej wyestymowanego na tym zbiorze (Rys. 2.3).

Jak widać regresja liniowa oprócz wartości z przedziału $[0, 1]$ prognozowała również ujemne prawdopodobieństwa dla $x > 130$. Trudno zinterpretować takie wartości, dlatego naturalnym wydaje się potrzeba stworzenia nowego modelu, który prognozowałby tylko wartości z przedziału $[0, 1]$. Odpowiedzią na tę potrzebę jest właśnie regresja



Rys. 2.3. Dopasowanie regresji liniowej do zmiennej binarnej

Źródło: opracowanie własne

logistyczna.

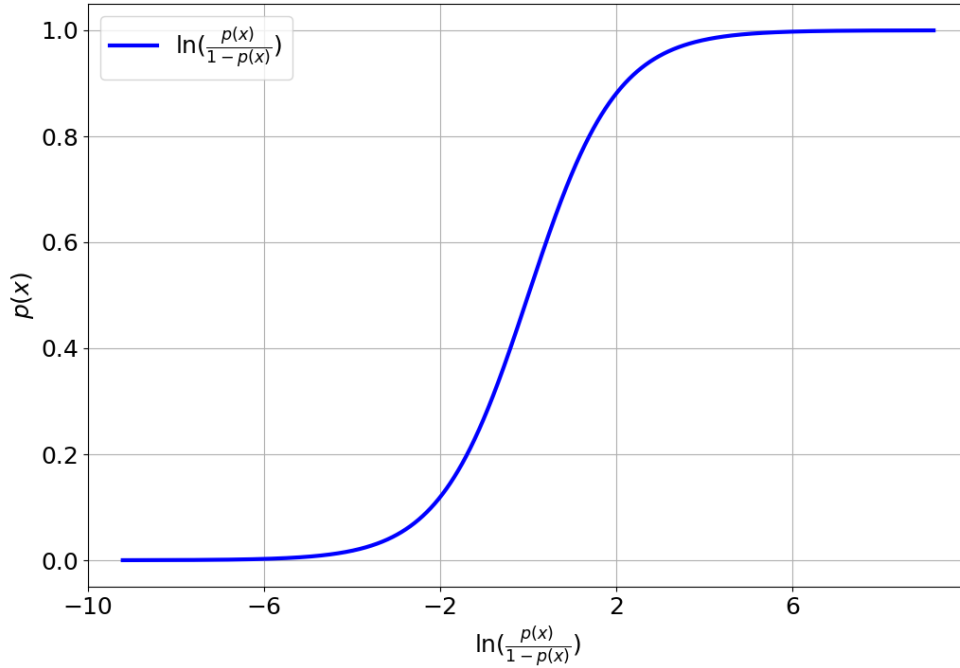
Prawdopodobieństwo w modelu definiowane jest jako funkcja logarytmiczna

$$p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}. \quad (2.16)$$

Wykonując kilka modyfikacji można dojść do wzoru

$$\ln\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 x. \quad (2.17)$$

Wzór ten jest nazywany funkcją logitową [Hosmer and Lemeshow, 2000]. Łatwo zauważyć, że mając zmienną binarną, $p(x)$ określa prawdopodobieństwo zdarzenia, że $y = 1$, podczas, gdy $1 - p(x)$ jest zdarzeniem przeciwnym. Stosunek $\frac{p(x)}{1 - p(x)}$ określa się jako szansę. Warto zauważyć, że otrzymana funkcja logit liniowo zależy od zmiennej x , to znaczy, że zmiana wartości x o jednostkę zwiększa wartość funkcji logit o β_1 . Nadal jednak relacja między szansą, a x nie jest liniowa i zmiana x o jeden powoduje zmianę funkcji szansy o e^{β_1} . Dodatkowo funkcja logit jest w tym momencie funkcją ciągłą przyjmującą wartości z przedziału $[0, 1]$ dla $x \in (-\infty, \infty)$ zgodnie z Rys. 2.4 (dla ułatwienia interpretacji osie na wykresie zostały zamienione).



Rys. 2.4. Wykres $p(x)$ od wartości funkcji logit

Źródło: opracowanie własne

Funkcja logit asymptotycznie zbiega do wartości $-\infty$ dla $p(x) \rightarrow 0$ i do ∞ dla $p(x) \rightarrow 1$. Skoro model przewiduje wartości prawdopodobieństwa $p(x)$ to dla zmiennych binarnych musi ono zostać przeliczone na konkretną klasę zmiennej objaśnianej. Zwykle wartości $p(x) > 0.5$ oznaczają przyjęcie klasy 1, a $p(x) < 0.5$ klasy 0 [Geron, 2019].

Skoro model jest gotowy trzeba dobrać sposób estymacji parametrów β_0 i β_1 . W przypadku regresji liniowej stosuje się do tego metodę najmniejszych kwadratów (MNK). Niestety ta sama metoda nie ma zastosowania w przypadku regresji logistycznej [Hosmer and Lemeshow, 2000]. W jej przypadku trzeba zastosować metodę największej wiarygodności (MNW). MNW umożliwia estymację takich parametrów modelu, które będą maksymalizować wartość funkcji wiarygodności. Do tego trzeba jednak znać postać funkcji wiarygodności. Intuicja podpowiada, że parametry muszą być tak dobrane, aby model przewidywał wartość $p(x) \approx 0$ dla $y = 0$ i $p(x) \approx 1$ dla $y = 1$. Dla wygody dalszych rozważań funkcję wiarygodności dla jednej obserwacji można zapisać jako

$$p(x_i)^{y_i} [1 - p(x_i)]^{1-y_i}. \quad (2.18)$$

Teraz zastanówmy się, czy powyższa funkcja rzeczywiście spełnia wymagania. Rozważmy cztery przypadki:

$$1. \ p(x) \approx 0 \wedge y = 0 : p(x_i)^{y_i} \approx 1, [1 - p(x_i)]^{1-y_i} \approx 1 \rightarrow p(x_i)^{y_i} [1 - p(x_i)]^{1-y_i} \approx 1$$

2. $p(x) \approx 1 \wedge y = 1 : p(x_i)^{y_i} \approx 1, [1 - p(x_i)]^{1-y_i} \approx 1 \rightarrow p(x_i)^{y_i} [1 - p(x_i)]^{1-y_i} \approx 1$
3. $p(x) \approx 0 \wedge y = 1 : p(x_i)^{y_i} \approx 0, [1 - p(x_i)]^{1-y_i} \approx 1 \rightarrow p(x_i)^{y_i} [1 - p(x_i)]^{1-y_i} \approx 0$
4. $p(x) \approx 1 \wedge y = 0 : p(x_i)^{y_i} \approx 1, [1 - p(x_i)]^{1-y_i} \approx 0 \rightarrow p(x_i)^{y_i} [1 - p(x_i)]^{1-y_i} \approx 0$

Funkcja działa tak jak powinna i uzyskuje największe wartości tylko kiedy $p(x)$ osiąga zbliżone wartości do y .

Rozszerzając powyższe równanie na cały zbiór obserwacji można skonstruować funkcję wiarygodności

$$l = \prod_{i=1}^N \left[p(x_i)^{y_i} [1 - p(x_i)]^{1-y_i} \right]. \quad (2.19)$$

Funkcję l wyliczamy przez pomnożenie wartości wynikających ze wzoru 2.18 dla wszystkich obserwacji (N). Wykorzystano tutaj założenie regresji logistycznej, że poszczególne obserwacje są od siebie niezależne [Hosmer and Lemeshow, 2000].

Zgodnie z algorytmem MNW należy teraz stworzyć funkcję logarytmu z funkcji wiarygodności. Wykonując kilka przekształceń można otrzymać wzór

$$L = \ln(l) = \sum_{i=1}^N \left[y_i \ln(p(x_i)) + (1 - y_i) \ln[1 - p(x_i)] \right] = \sum_{i=1}^N \left[y_i \beta^T x_i - \ln \left[1 + e^{\beta^T x_i} \right] \right], \quad (2.20)$$

gdzie β^T jest transponowanym wektorem parametrów β_i dla wszystkich zmiennych w modelu. Dla rozpatrywanego uproszczonego przypadku $\beta^T = \{\beta_0, \beta_1\}$.

Jest to logarytmiczna funkcja wiarygodności. Kolejnym krokiem będzie znalezienie maksimum funkcji L . Nie jest to trywialne zagadnienie, bo sama funkcja nie jest liniowa. Z pomocą przychodzą jednak znowu wbudowane pakiety matematyczne, np. w języku Python, Umożliwiają one znalezienie maksimum funkcji, a co za tym idzie optymalnych parametrów β_0, β_1 .

W całym rozdziale założono, że funkcja logit liniowo zależy od zmiennej x . Nic jednak nie stoi na przeszkodzie, aby logit uzależnić liniowo od wielu zmiennych. Model regresji logistycznej można również użyć w przypadku, gdy X jest zmienną wielowymiarową.

W odróżnieniu od modeli drzew decyzyjnych regresja logistyczna nie daje tak szerokiego pola manewru jeśli chodzi o strojenie. W całym rozpatrywanym do tej pory algorytmie ciężko znaleźć takie parametry, na których warto się skupić testując różne warianty modelu. Istnieją jednak dwa inne warianty regresji liniowej, których stosowanie może wpłynąć na stopień dopasowania modelu do danych. Obydwa warianty bazują na idei

regularyzacji modelu. Regularyzację modelu stosuje się w celu zmniejszenia wpływu przetrenowywania modelu i polega ona na zmniejszaniu wartości parametrów w modelach regresyjnych. Dokładny mechanizm przedyskutujemy poniżej przy prezentacji sposobów regularyzacji. Najpopularniejszymi metodami są regresja metodą Lasso (*least absolute shrinkage and selection operator*), regresja grzbietowa i metoda elastycznej siatki (*elastic net*). Zaletą regularyzacji jest to, że może być ona stosowana do wielu modeli regresyjnych. Te same rozwiązania mogą być użyte do regresji liniowej jak i do regresji logistycznej.

Metoda Lasso polega na dodaniu członu regularyzującego do równania modelu, które chcemy zoptymalizować. Człon ten przyjmuje postać

$$\gamma_1 = \frac{1}{C} \sum_{i=1}^n |\beta_i|, \quad (2.21)$$

gdzie n jest liczbą zmiennych w zbiorze danych.

Po zastosowaniu nowego czynnika funkcja L przyjmie więc postać

$$L = \sum_{i=1}^N \left[y_i \ln(p(x_i)) + (1 - y_i) \ln[1 - p(x_i)] \right] - \frac{1}{C} \sum_{i=1}^n |\beta_i|. \quad (2.22)$$

W całym modelu nadal dążymy do maksymalizacji funkcji L . Z drugiej strony zyskujemy też możliwość bezpośredniej kontroli estymowanych parametrów β - im większe wartości β tym większa wartość funkcji L . Warto zauważyć, że w tym wariancie pojawiła się również nowa zmienna C . To właśnie ona odpowiada za odpowiednie wyważenie tych dwóch wpływów i może być testowana jako hiperparametr modelu. Dla dużych C wartość członu regularyzującego zbiega do zera i otrzymujemy standardowy model regresji logistycznej. Z kolei dla bardzo małych wartości C estymowane parametry β będą zbiegać do zera. Z tych krótkich rozważań można więc wysnuć wniosek, że strojąc parametr C tak naprawdę kontrolujemy czułość (wariancję) i obciążenie modelu (*bias*). Generalnie obciążenie modelu wynika z ograniczenia jego elastyczności. Objawiać się może na przykład tym, że zastostujemy model liniowy, do problemu o zupełnie innej zależności między zmiennymi. Wariancja wynika z nadmiernej czułości modelu na zmianę zbioru danych treningowych. Jeśli nieznacznie zmieniając zbiór danych uzyskujemy znacznie zróżnicowane wyniki predykcji to mówimy, że model jest bardzo czuły lub po prostu ma dużą wariancję. W kontekście regresji LASSO dla małych wartości C uzyskujemy model bardzo sztywny, a co za tym idzie prawdopodobnie o dużym obciążeniu. Dla dużych wartości C model będzie bardziej elastyczny, ale będzie również bardziej czuły na jakiegokolwiek zmiany w zbiorze danych. Nadmierna czułość

objawi się, kiedy będziemy chcieli użyć wygenerowanego modelu na zbiorze testowym. Możemy wtedy uzyskać model przetrenowany. Przy dużym obciążeniu częściej będziemy uzyskiwać model niedotrenowany. Celem testowania hiperparametru C będzie znalezienie takiego punktu, dla którego kombinacja obciążenia i wariancji będzie optymalna [Geron, 2019, James i in., 2014]. Parametr C przyjmuje wartości dodatnie z zakresu $(0, +\infty)$.

Zaletą stosowania metody Lasso jest również całkowite wyzerowanie parametrów przy mniej istotnych cechach. Z tego powodu model może być również wykorzystywany do selekcji cech.

Metoda grzbietowa w założeniach jest bardzo podobna do Lasso. Jedyna różnica polega na zmianie członu dodawanego do funkcji. Tym razem przyjmuje on postać

$$\gamma_2 = \frac{1}{2C} \sum_{i=1}^n \beta_i^2. \quad (2.23)$$

Po zastosowaniu czynnika funkcja L przyjmie więc postać

$$L = \sum_{i=1}^N \left[y_i \ln(p(x_i)) + (1 - y_i) \ln[1 - p(x_i)] \right] - \frac{1}{2C} \sum_{i=1}^n \beta_i^2. \quad (2.24)$$

Ostatnim wariantem jest zastosowanie obydwu członów jednocześnie. Uzyskujemy wtedy metodę siatki elastycznej, a funkcja L przyjmuje wtedy postać

$$L = \sum_{i=1}^N \left[y_i \ln(p(x_i)) + (1 - y_i) \ln[1 - p(x_i)] \right] - \frac{r}{C} \sum_{i=1}^n |\beta_i| - \frac{1-r}{2C} \sum_{i=1}^n \beta_i^2. \quad (2.25)$$

Warto zauważyć, że w modelu siatki elastycznej mamy dodatkowy parametr r . Dzięki niemu kontrolujemy wpływ od członu LASSO i regresji grzbietowej. Przyjmuje wartości z zakresu $[0, 1]$. Dla wartości 1 otrzymujemy regresję LASSO, a dla 0 regresję grzbietową.

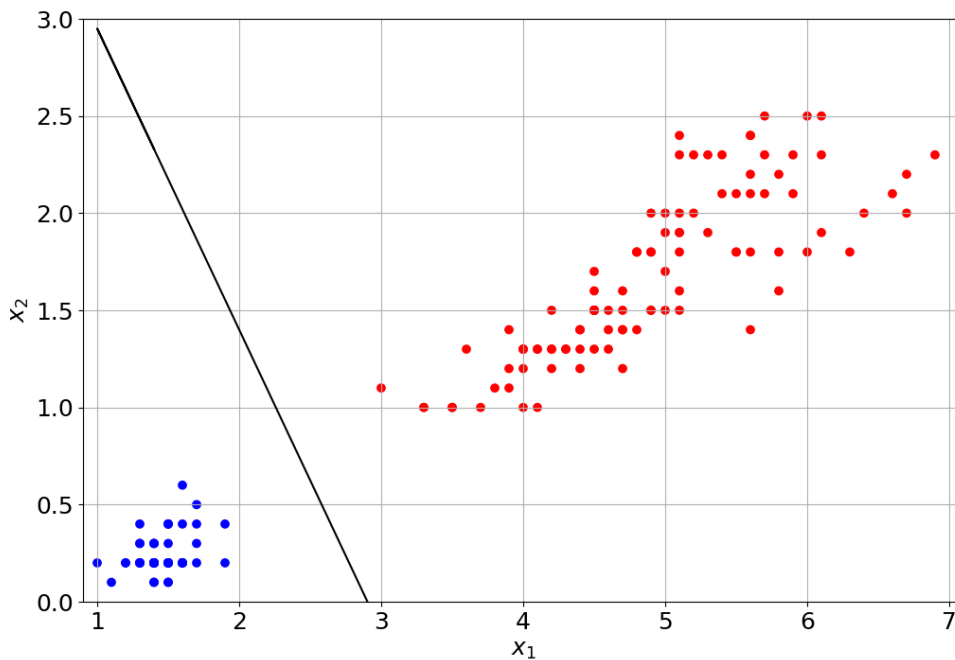
2.5. Maszyna wektorów nośnych (SVM)

Kolejnym modelem wykorzystywanym do zadań klasyfikacyjnych jest maszyna wektorów nośnych (ang. Support Vector Machine - SVM). Założeniem SVM jest znalezienie takiej hiperpłaszczyzny, która najlepiej będzie dzielić przestrzeń zmiennych X ze względu na przynależność do klasy y . W p -wymiarowej przestrzeni hiperpłaszczyzna ta będzie opisana równaniem

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p = 0. \quad (2.26)$$

Hiperpłaszczyzna dzieli przestrzeń na 2 części. Jeśli chodzi o wizualizację to najlepiej wyobrazić to sobie w 2-wymiarowej przestrzeni. Na potrzeby rozważań założmy, że dysponujemy zbiorem danych złożonym z dwóch zmiennych objaśniających x_1 oraz x_2 i objaśnianej y , która przyjmuje dwie klasy: 1 oraz -1.

Taki zbiór można przedstawić na 2-wymiarowej płaszczyźnie (Rys. 2.5)



Rys. 2.5. Przykładowy 2-wymiarowy zbiór danych podzielony ze względu na wartość klasy.

Źródło: opracowanie własne. Czerwone punkty odpowiadają obserwacjom, dla których $y = 1$, a niebieskie to te, dla których $y = -1$.

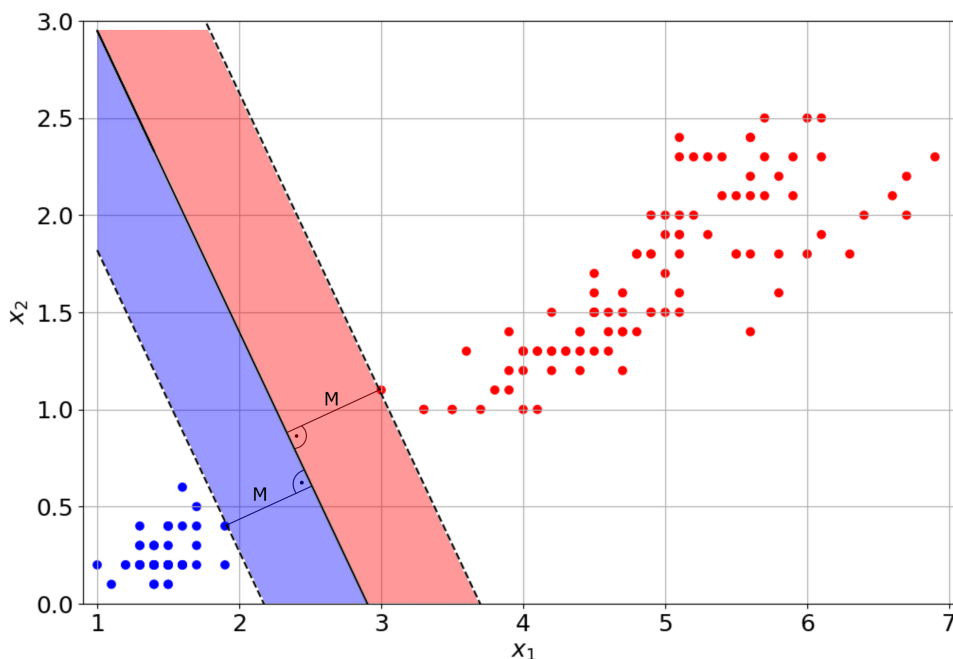
Obserwacje na wykresie zostały przyporządkowane do grup na podstawie odpowiadającej im wartości zmiennej y . Hiperpłaszczyzna rozdzielająca obserwacje na dwie grupy w tym przypadku jest linią prostą. Przykład takiej linii został zaznaczony na Rys. 2.5 czarnym kolorem. Warto zauważyć, że w rozpatrywanym przypadku wszystkie punkty leżące poniżej prostej są opisane równaniem

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 < 0, \quad (2.27)$$

zaś te leżące powyżej równaniem

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 > 0. \quad (2.28)$$

W ten sposób można zbudować model przewidujący przynależność do klasy na podstawie zmiennych objaśniających. Jeśli $\beta_0 + \beta_1 X_1 + \beta_2 X_2 < 0$ dla wyestymowanych parametrów $\beta_0, \beta_1, \beta_2$ to model powinien przyporządkować daną obserwację do klasy -1, w przeciwnym wypadku do klasy 1. Trzeba tutaj zauważyć pewną komplikację. W przypadku przestrzeni danych, które da się idealnie pogrupować, czyli tak, że wszystkie obserwacje poprawnie będą należały do odpowiadających im klas, istnieje nieskończona liczba możliwych hiperpłaszczyzn [James i in., 2014]. Konieczne jest w takim razie przyjęcie kryterium wyboru konkretnej hiperpłaszczyzny ze zbioru możliwych rozwiązań. Najbardziej intuicyjnym modelem, który realizuje powyższe założenie jest klasyfikator maksymalnego marginesu (MMC - *maximal margin classifier*). Margines w tym przypadku jest odległością najbliższych punktów (po obu stronach) do hiperpłaszczyzny, a punkty wyznaczające margines nazywamy wektorami nośnymi. Cały zamysł modelu przedstawiono na Rys. 2.6.



Rys. 2.6. Przykładowy 2-wymiarowy zbiór danych podzielony ze względu na wartość klasy z naniesionymi marginesami klasyfikatora.

Źródło: opracowanie własne. Czerwone punkty odpowiadają obserwacjom, dla których $y = -1$, a niebieskie to te, dla których $y = 1$. Wyznaczone marginesy po obydwu stronach hiperpłaszczyzny zostały zaznaczone odpowiadającymi im kolorami.

Zaznaczona na rysunku wartość M odpowiada szerokości wyznaczonego margine-

su, czyli odległości wektorów nośnych od hiperpłaszczyzny. Taki dobór modelu jest bardzo intuicyjny. Szerokość marginesu jest w tym przypadku miernikiem zaufania do hiperpłaszczyzny. Im szerszy margines tym większa pewność wyznaczonego modelu [James i in., 2014].

Pozostaje jeszcze kwestia optymalizacji. Wiemy już, że wybrany model będzie miał największą wartość M spośród możliwych rozwiązań. Ponadto w modelu zakładamy, że jesteśmy w stanie idealnie rozdzielić obserwacje na grupy. W takim wypadku można to zapewnić wzorem

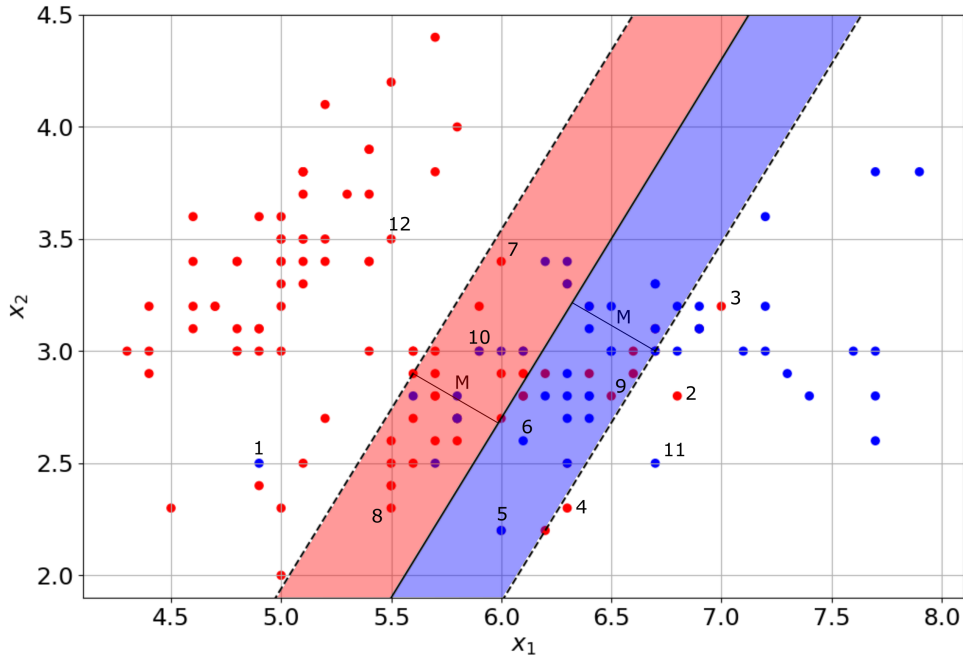
$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}) \geq M, \quad i = 1, 2, \dots, n. \quad (2.29)$$

Jeśli rzeczywista wartość y dla obserwacji i wynosi -1 to gwarantuje to, że punkt zawsze będzie się znajdował poniżej dolnej linii marginesu (poniżej niebieskiego obszaru na wykresie 2.6) zgodnie ze wzorem 2.27. Jeśli wartość y dla obserwacji i będzie wynosić 1 to taka obserwacja zawsze powinna się znaleźć powyżej górnej linii marginesu (powyżej czerwonego obszaru na wykresie 2.6). Rozpatrując dwa powyższe warunki jesteśmy w stanie wyznaczyć optymalny klasyfikator maksymalnego marginesu.

Przedstawiony na wykresie przykład jest bardzo prosty i nawet wizualnie jesteśmy w stanie wyznaczyć dwie grupy wartości. Co jednak w przypadku, gdy rozkład danych jest bardziej skomplikowany i nie jesteśmy w stanie wyznaczyć płaszczyzny, która idealnie rozdzieli obserwacje na grupy? Wtedy możemy skorzystać z klasyfikatora miękkiego marginesu, który dopuszcza występowanie obserwacji w obszarze marginesu, a nawet po złej stronie rozdzielającej hiperpłaszczyzny. Jest to więc uogólniona wersja klasyfikatora maksymalnego marginesu i to właśnie ona jest nazywana klasyfikatorem wektorów nośnych (ang. Support Vector Classifier (SVC)).

Założenie SVC jest bardzo podobne do klasyfikatora maksymalnego marginesu, z tą różnicą, że SVC dopuszcza możliwość przekroczenia marginesu przez obserwacje poszczególnych grup. Przykład nieseparowalnego zbioru danych i poglądowy sposób dopasowania hiperpłaszczyzny został pokazany na Rys. 2.7.

Tak jak poprzednim razem model wyznacza hiperpłaszczyznę dzielącą przestrzeń danych na dwie części. Można zauważyć, że większość punktów została poprawnie przypisana do klas - punkty 11 i 12. Nowością są teraz punkty leżące między płaszczyzną, a marginesem (punkty 5, 6, 7, 8). Nadal są one poprawnie sklasyfikowane, ale ich położenie nie daje gwarancji poprawnego przypisania. Dowodem na to są punkty 9 i 10. W ich przypadku model przypisuje złą klasę, a same punkty leżą w obrębie marginesów klas przeciwnych. Ostatnimi przypadkami są mocno wystające w głąb podprzestrzeni odpowiadających przeciwnym klasom punkty 1, 2, 3 oraz 4. W ich przypadku model



Rys. 2.7. Przykładowy 2-wymiarowy zbiór danych nieseparowalnych klasyfikatorem maksymalnego marginesu podzielony ze względu na wartość klasy.

Źródło: opracowanie własne. Czerwone punkty odpowiadają obserwacjom, dla których $y = -1$, a niebieskie to te, dla których $y = 1$. Wyznaczone marginesy po obydwu stronach hiperpłaszczyzny zostały zaznaczone odpowiadającymi im kolorami.

nie ma szans poprawnie przypisać im klasy. Błędne przypisania obniżają jakość modelu, ale z drugiej strony nadal jesteśmy w stanie poprawnie odseparować większość punktów. Klasyfikator maksymalnego marginesu nie byłby w stanie nic zdziałać w tym przypadku. SVC ma jeszcze jedną ogromną zaletę. Skoro dopuszcza miękki margines, to też jest bardziej odporny na zmiany w pojedynczych obserwacjach, a co za tym idzie rzadziej będzie w nim dochodzić do przetrenowania [James i in., 2014].

Możliwość przekraczania marginesów przez punkty wymusza konieczność kontrolowania tego zjawiska. W tym celu do modelu wprowadza się parametr C . Żeby wyjaśnić jego znaczenie spójrzmy na wzór opisujący położenie pojedynczego punktu względem hiperpłaszczyzny. Tym razem opisywane będzie wzorem

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}) \geq M(1 - \epsilon_i). \quad (2.30)$$

Pojawia się nowy parametr ϵ_i , który opisuje położenie pojedynczego punktu względem hiperpłaszczyzny. Jeśli dla obserwacji i , $\epsilon_i = 0$ oznacza to tyle, że leży ona po odpowiedniej stronie marginesu. Jeśli $\epsilon_i > 0$ dochodzi do przekroczenia marginesu, ale

zmienna nadal znajduje się po odpowiedniej stronie hiperpłaszczyzny. Jeśli $\epsilon_i > 0$ to zmienna jest po złej stronie hiperpłaszczyzny [James i in., 2014]. Parametr C jest sumą wartości ϵ ze wszystkich zmiennych

$$C = \sum_i^n \epsilon_i. \quad (2.31)$$

W celu uzyskania optymalnego modelu SVC nadal będziemy dążyć do maksymalizacji szerokości marginesu M , ale tym razem położenie pojedynczej zmiennej opisane jest wzorem 2.30, co dodatkowo wymusza warunek $\sum_i^n \epsilon_i \leq C$. Rozpatrując taki problem optymalizacyjny znajdujemy poszukiwany model SVC.

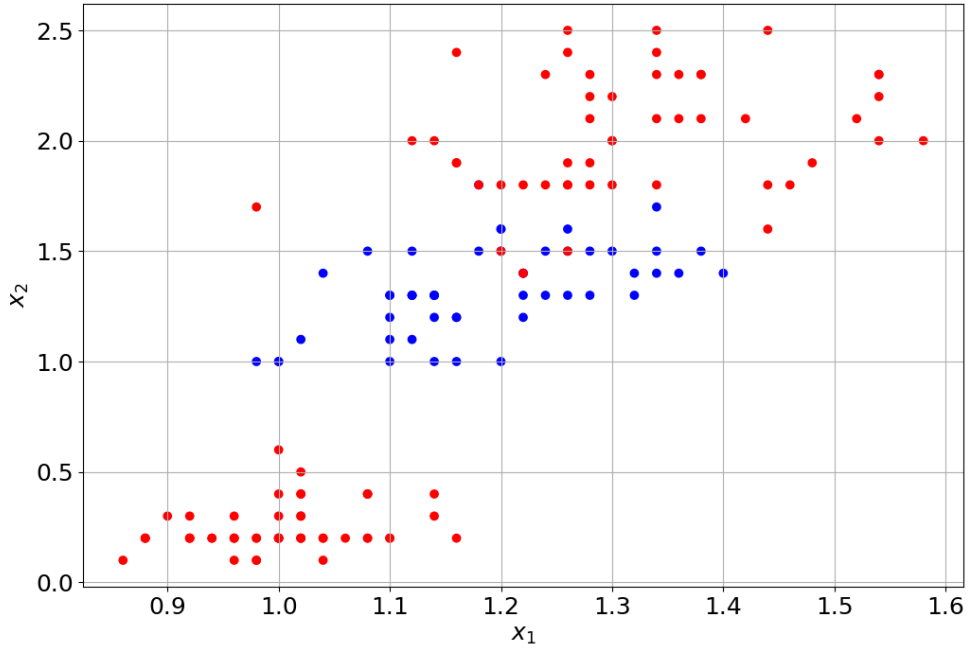
Warto jeszcze przyrzeć się bliżej parametrowi C , bo dzięki niemu kontrolujemy restrykcyjność modelu. Duża wartość C oznacza tyle, że dopuszczamy większą swobodę modelu jeśli chodzi o przekraczanie marginesów przez pojedyncze zmienne. Takie ustawienie powoduje, że model częściej będzie się mylił, ale wariancja obserwacji będzie mniejsza. Mała wartość ma działanie dokładnie odwrotne.

Tak jak to zostało już wspomniane model SVC jest odporny na zmiany pojedynczych obserwacji. Przypatrując się wzorowi 2.30 można dojść do wniosku, że to zmienne przekraczające marginesy decydują o postaci modelu. Im bardziej oddalona obserwacja od hiperpłaszczyzny po jej złej stronie tym bardziej zwiększa ona sumę $\sum_i^n \epsilon_i$, a co za tym idzie zabiera możliwość przekroczenia marginesu innym obserwacjom.

Do tej pory cały czas rozpatrywaliśmy użycie klasyfikatora maksymalnego marginesu i klasyfikatora miękkiego marginesu. W obydwu modelach wyznaczaliśmy hiperpłaszczyznę, która określana była jako liniowa. Nietrudno wyobrazić sobie zbiór danych, który jest nierozdzielny klasyfikatorem liniowym. Spójrzmy na wykres danych Rys. 2.8.

W przypadku takiego zbioru danych hiperpłaszczyzna o charakterze liniowym nie jest niestety dobrym rozwiązaniem. Wyjściem w tej sytuacji może być skonstruowanie hiperpłaszczyzny nieliniowej. Najprościej można to zrealizować poprzez dodanie wyrazów wyższego rzędu do równania opisującego hiperpłaszczyznę. Nie wpływa to jednak na charakter modelu. Nieliniowość hiperpłaszczyzny jest niestety większym wyzwaniem technicznym, ale warunek optymalizacyjny pozostaje ten sam.

Rozpatrując nieliniowe hiperpłaszczyzny w modelu SVC dochodzimy do kolejnej generalizacji modelu, czyli zapowiedzianej na początku rozdziału maszyny wektorów nośnych (ang. Support Vector Machine (SVM)). SVM rozszerza funkcjonalność modelu SVC dzięki możliwości zmienienia jądra (ang. kernel) modelu [James i in., 2014]. Jest to bardzo wygodne, bo dzięki jednemu narzędziu jesteśmy w stanie klasyfikować liniowe i nieliniowe przypadki. Widać tutaj wielką uniwersalność modeli SVM, zwłaszcza



Rys. 2.8. Przykładowy 2-wymiarowy zbiór danych nieseparowalny liniową hiperpłaszczyzną

Źródło: opracowanie własne

jeśli dodamy do tego, że mogą służyć również jako modele regresyjne.

Wróćmy jednak do zagadnienia jądra funkcji. Można pokazać [James i in., 2014], [Hastie i in., 2009], [Press i in., 2007], że funkcja optymalizująca w modelu SVC zależy od iloczynu skalarnego obserwacji

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^p x_{ij} x_{i'j}, \quad (2.32)$$

gdzie x_i i $x_{i'}$ pochodzą ze zbioru treningowego. Zakładamy tutaj, że i oraz i' mogą być od siebie różne. j odpowiada indeksowi zmiennej dla danej obserwacji i . W rozpatrywanym przypadku z początku rozdziału $p = 2$, bo mamy dwie zmienne - x_1 oraz x_2 .

Uogólniając wzór na hiperpłaszczyznę modelu można ją zapisać jako

$$f(x) = \alpha + \sum_{i=1}^n \beta_i \langle x, x_i \rangle. \quad (2.33)$$

Rozpatrujemy teraz dowolną obserwację x i jej iloczyn $\langle x, x_i \rangle$ z obserwacjami x_i ze zbioru treningowego.

Jest to poprawny wzór dla jądra liniowego, ale rozpatrując model o innej strukturze jądra x będzie reprezentowany jako funkcja $h(x)$, co daje w efekcie wzór

$$f(x) = \alpha + \sum_{i=1}^n \beta_i \langle h(x), h(x_i) \rangle. \quad (2.34)$$

Iloczyn $\langle h(x), h(x_i) \rangle$ to właśnie funkcja jądra modelu. W ogólności może przybierać różne formy. Również może być rozpatrywany jako hiperparametr modelu SVM. W tabeli Tab. 2.2 zaprezentowano kilka najpopularniejszych funkcji jądra modelu SVM.

Tabela 2.2. Wykaz funkcji jądra dostępnych w modelu SVC

Nazwa	Funkcja
linear	$K(x_i, x_{i'}) = \sum_{j=1}^p x_{ij}x_{i'j}$
poly	$K(x_i, x_{i'}) = (1 + \sum_{j=1}^p x_{ij}x_{i'j})^d, d - \text{stopień wielomianu}$
rbf	$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^p (x_{ij} - x_{i'j})^2\right)$
sigmoid	$K(x_i, x_{i'}) = \tanh\left(\kappa_1 \sum_{j=1}^p x_{ij}x_{i'j} + \kappa_2\right)$
precomputed	Własna funkcja jądra - wymaga podania iloczynu $\langle h(x), h(x_i) \rangle$

Źródło: lista przygotowana na podstawie dokumentacji technicznej dotyczącej modelu SVC z biblioteki scikit-learn⁶

Na koniec analizy modelu SVM trzeba wspomnieć o mechanizmie zawiasowej funkcji straty (*hinge loss*). Wracając do problemu optymalizacyjnego modelu SVC znowu można zapisać go w inny sposób [James i in., 2014], [Press i in., 2007]

$$\sum_{i=1}^n \max[0, 1 - y_i f(x_i)] + \lambda \sum_{j=1}^p \beta_j^2. \quad (2.35)$$

Zadaniem optymalizacyjnym jest znalezienie parametrów, dla których powyższa funkcja jest minimalna [James i in., 2014]. Jest to bardzo podobny mechanizm, jeśli porównać go z funkcją straty L przedstawioną w rozdziale o regresji logistycznej (wzór 2.23). Człon $\lambda \sum_{j=1}^p \beta_j^2$ przypomina w tym przypadku czynnik metody grzbietowej. Mamy więc do czynienia z pewnego rodzaju regularyzacją modelu. Znowu kluczowy jest parametr λ , bo to on kontroluje regularyzację przez wymuszenie zmniejszenia parametrów β powyższej funkcji. Jeśli λ jest mała to parametry β optymalnego modelu będą miały większą wartość i częściej będzie dochodzić do przekraczania marginesu. Dokładnie odwrotna sytuacja jest w przypadku dużej wartości λ .

Rozdział 3

Przygotowanie danych

W tym rozdziale zaprezentowana zostanie charakterystyka danych użytych w niniejszej pracy oraz ogólna idea analizy technicznej. Zostaną również scharakteryzowane indeksy obliczane na podstawie podstawowych wartości dotyczących akcji tj. wartości zamknięcia, otwarcia, najwyższej oraz najniższej ceny dziennej oraz wolumenu. Następnie zostanie poddany dyskusji sposób doboru zmiennych, na podstawie których będzie estymowany model. Na koniec krótko zostaną scharakteryzowane pojęcia walidacji krzyżowej, podziału zbioru danych na zbiór testowy i treningowy oraz sposoby oceny poprawności modelu.

W badaniach zostały wykorzystane historyczne dane dotyczące indeksów giełdowych S&P 500, DAX, Nikkei 225, BSE SENSEX 30, UK 100 oraz WIG20 z lat 2010 - 2022. Dobór indeksów do badanego problemu nie był przypadkowy. Każdy indeks pochodził z innej wiodącej giełdy papierów wartościowych z całego świata. Dzięki temu powtarzalność w wynikach będzie mogła świadczyć o słuszności użycia uczenia maszynowego do prognozowania cen akcji bez względu na giełdę, z której pochodzą. To założenie ma niestety pewne ograniczenia, bo na same predykcje będą miały również wpływ czynniki związane z niską płynnością akcji. Dla wszystkich indeksów dane dotyczyły wartości dziennych. Wyjściowo w danych zawarte były wartości otwarcia, zamknięcia, najwyższa i najniższa cena z danego dnia oraz wolumen. Na podstawie danych dotyczących cen zamknięciowych wyliczono binarną zmienną y . Jeśli różnica między ceną z dnia t i $t - 1$ była nieujemna, to przypisowano wartość 1 dla dnia t . Jeśli była ujemna, to przypisywano wartość 0. W kolejnym rozdziale zostanie scharakteryzowana idea analizy technicznej, która będzie wstępem do wyboru zmiennych do modelu.

3.1. Analiza techniczna i jej wskaźniki

Analiza techniczna polega na prognozowaniu trendów na rynku papierów wartościowych przy użyciu historycznych danych. Ściśle rzecz ujmując na podstawie danych historycznych budowane są struktury matematyczne, które mają odzwierciedlać trendy na rynku, a raczej pomóc w oszacowaniu tych trendów. Tak ugruntowane struktury zapewniają uniwersalność użycia, czyli mogą być użyte do dowolnych produktów giełdowych w każdym momencie czasu. Są również łatwo interpretowalne i mają jasno określoną logikę [Colby, 2002].

Założeniem analizy technicznej jest powtarzalność zdarzeń. Cała analiza techniczna bazuje na zdarzeniach z przeszłości jako punkcie wyjścia do przewidywania przyszłości. To co staramy się przewidzieć to trend. Jego występowanie jest kolejnym, równie ważnym założeniem analizy technicznej. Występowanie trendu zgodnie z analizą techniczną jest następstwem mechaniki "owczego pędu", czyli działaniem mającym na celu naśladowanie aktualnych zdarzeń na rynku. Zakładamy więc, że inwestorzy w swoich decyzjach bardzo mocno bazują na emocjach i starają się dopasować do aktualnych realiów rynkowych. Wyróżnić można 3 rodzaje trendów: wzrostowy, spadkowy oraz horyzontalny. W zależności od tego jak długo trend występuje można również podzielić go na długo-, średnio- i krótkoterminowy. Długoterminowy trend rozpatruje się zwykle przypadku, gdy utrzymuje się dłużej niż rok. Krótkoterminowy to taki, który zwykle występuje przez kilka lub kilkanaście dni. Cała reszta to trend średnioterminowy [Colby, 2002].

Indeksy analizy technicznej można podzielić na 4 grupy: wskaźniki momentum, trendu, zmienności (oscylatory) oraz wolumenu. Wskaźniki momentum mierzą prędkość ruchów cen i zwykle korespondują z wskaźnikami trendu. Typowy cykl giełdowy rozpoczyna się od wzrostu cen. Mamy wtedy do czynienia z nowym trendem wzrostowym, któremu towarzyszy nagły wzrost momentum. Sytuacja trwa aż cena akcji zaczyna się zbliżać do swojej najwyższej wartości, wtedy momentum stopniowo maleje. W szczytowym punkcie momentum oscyluje wokół zera i mamy wtedy do czynienia z dodatnią dywergencją, czyli rozbieżnością między ceną akcji, a wskazaniem indeksów analizy technicznej. Jej następstwem jest zmiana trendu. Po osiągnięciu najwyższej wartości następuje powolny spadek wartości, któremu towarzyszy wzrost ujemnego momentum. Dochodzimy do najniższej wartości, gdzie mamy do czynienia z ujemną dywergencją i ponownym odwróceniem trendu. Następnie cały cykl zaczyna się od początku. Jeśli chodzi o oscylatory to pokazują one zmienność cen produktów. Większa zmienność cen oznacza zwykle większą niepewność inwestorów co do produktu. Z drugiej strony mała zmienność może być symptomem nadchodzącej zmiany trendu

[Bojańczyk, 2013]. Wskaźniki wolumenowe bazują na liczbie transakcji wykonanych na danym produkcie. Wzrastająca liczba transakcji przy jednoczesnym wzroście cen jest indykatorem siły produktu. Odwrotną sytuację mamy w przypadku dużego wolumenu i spadającej ceny. W przypadku, gdy cena akcji osiąga maksimum przy jednoczesnym spadku wolumenu może to świadczyć o zmianie trendu.

Wskaźniki analizy technicznej tworzą bardzo silne narzędzie w ręku inwestora. Mogą informować o nadchodzących zmianach nastrojów giełdowych, dzięki czemu jesteśmy w stanie wcześniej podjąć korzystne decyzje. Z uwagi na korzyści wynikające z ich stosowania zostały one użyte w niniejszej pracy jako zmienne objaśniające. Problemem, jaki pojawia się w przypadku analizy technicznej jest ogromna liczba istniejących wskaźników. Nie pomaga również fakt, że każdy z nich ma racjonalne podłoże matematyczne. Tworzy to nowy problem wyboru optymalnych wskaźników do badanego problemu. Można natknąć się w literaturze na określenie "zoo wskaźników" w kontekście tego problemu [Peng i in., 2021]. Wybór poszczególnych wskaźników do modelu został opisany w następnym podrozdziale, a poniżej znajduje się charakterystyka każdego z nich.

Średnia ruchoma (MA)

Jest to jeden z najstarszych wskaźników analizy technicznej i wyliczana jest następującą formułą

$$MA_t = \frac{1}{n} \sum_{i=t-n+1}^t C_i. \quad (3.1)$$

gdzie t jest numerem obserwacji, dla której wyliczamy średnią ruchomą, C_t jest ceną zamknięcia w momencie i , a n jest liczbą kolejnych obserwacji, które bierzemy pod uwagę przy obliczeniach. Trzeba tutaj zauważyć, że średnia ta jest wyliczana na podstawie n obserwacji poprzedzających dany moment. Oznacza to, że najstarsza obserwacja jest traktowana tak samo jak najwcześniejsza i będzie miała taki sam wkład przy wyliczaniu średniej ruchomej. To niewątpliwie duży problem, bo im dane bardziej zbliżone w czasie tym większa szansa, że będą zachowywały się podobnie. Ten argument podawany jest jako krytyka średniej ruchomej [Colby, 2002]. W celu wyeliminowania tego problemu stosuje się dwie inne wersje średniej ruchomej czyli ważoną i eksponencjalną. Ważona średnia przypisuje wagi poszczególnym cenom z poprzedzających dni. Najstarsze ceny otrzymują najniższe wagi. Jednym z jej wariantów jest eksponencjalna średnia ważona, która przypisuje wagi malejące eksponencjalnie dla coraz starszych obserwacji.

Konwergencja/dywergencja średnich kroczących (MACD)

Wskaźnik MACD bazuje na eksponencjalnej średniej ruchomej. Do jego obliczenia potrzebne są 2 średnie ruchome - jedna obliczona na podstawie 12 ostatnich obserwacji, a druga z 26. Następnie od średniej 12-dniowej odejmujemy 26-dniową. Kolejnym krokiem jest wyliczenie linii sygnałowej, która reprezentowana jest przez zwykłą eksponencjalną średnią ruchomą z ostatnich 9 dni. Ostatnim krokiem jest obliczenie różnicy między wcześniej obliczoną różnicą średnich, a linią sygnałową. Nie jest to jasno określona zasada, ale MACD można graficznie interpretować patrząc na punkty przecięcia różnicy średnich z linią sygnałową. W momencie, gdy różnica średnich przewyższa wartość linii sygnałowej, to jest to znak, że powinniśmy kupić dany instrument. Analogicznie jeśli różnica średnich jest poniżej linii sygnałowej, to jest znak do sprzedaży [Colby, 2002]. Należy tutaj jednak wspomnieć, że przyjęte zakresy średnich ruchomych podczas wyliczania MACD nie są stałymi wartościami i można je dostosowywać do własnych potrzeb. Podane zakresy są jedynie wartościami zwyczajowo przyjętymi jako domyślne przy takich wyliczeniach.

Wskaźnik siły względnej (RSI)

Wskaźnik RSI służy do szacowania wielkości i szybkości zmian cen. Wskaźnik ten może również służyć do wykrywania zjawiska przeszacowania akcji oraz szacowania prawdopodobieństwa zmiany trendu. RSI obliczany jest następującą formułą

$$RSI_t = 100 - \frac{100}{1 + \frac{EMA_t(DM^+)}{EMA_t(DM^-)}}, \quad (3.2)$$

gdzie $EMA_t(DM^+)$ jest eksponencjalną średnią ruchomą (EMA) z wzrostów cen z ostatnich n odczytów, a $EMA_t(DM^-)$ EMA z spadków cen z tego samego zakresu. DM^+ obliczana jest dla każdego dnia jako wartość różnicy cen. Jeśli jest ona ujemna to zastępujemy ją zerem. W przypadku DM^- każda dodatnia wartość zastępowana jest zerem. W ostatecznych obliczeniach EMA korzystamy tylko z absolutnych wartości spadków i wzrostów cen. Cała formuła jest stworzona tak, aby znormalizować wartość RSI do zakresu $<0, 100>$. Przyjmuje się, że wartość RSI powyżej 70 oznacza przeszacowanie akcji i może być sygnałem do sprzedaży przed zmianą trendu lub korektą. Im wartość bliższa 100 tym większe prawdopodobieństwo zmiany trendu. Wartość RSI poniżej 30 oznacza niedoszacowanie akcji i może być sygnałem do zakupu przed nadchodzącą zmianą trendu [Colby, 2002].

Wskaźnik zmiany (ROC)

Wskaźnik ROC mierzy stopę zmiany ceny akcji i obliczany jest następującą formułą

$$ROC_t = \frac{C_t - C_{t-n}}{C_{t-n}}. \quad (3.3)$$

Wskaźnik ten pokazuje więc względną zmianę ceny akcji z obserwacji t w porównaniu z ceną dla obserwacji $t - n$. ROC może być wykorzystany do wykrywania dywergencji. Przy trendzie wzrostowym może się zdarzyć, że wartość ROC maleje. Oznacza to również, że prędkość zmiany maleje, co może świadczyć o nadchodzącej zmianie trendu.

Wskaźnik kanałowy (CCI)

CCI służy do szacowania kierunku trendu oraz jego siły. Matematycznie reprezentowane jest to jako odchylenie wartości typowej dla danej obserwacji od wartości średniej z n poprzednich obserwacji

$$CCI_t = \frac{T_t - MA_t(T_t)}{0.015 \frac{1}{n} \sum_{i=1}^n |T_{t-i+1} - MA_t(T_t)|}. \quad (3.4)$$

Wartość typowa T_t jest w tym przypadku średnią z ceny zamknięcia oraz ceny najwyższej i najniższej. Większość obserwacji charakteryzuje się niskim wskaźnikiem CCI, co oznacza, że ich wartości typowe nie odbiegają w znacznym stopniu od wartości średniej. Duże wartości CCI z kolei alarmują o szansie na korzystne kupno lub sprzedaż papierów wartościowych.

Wskaźnik średniego ruchu kierunkowego (ADX)

Wskaźnik ADX służy do szacowania siły trendu. Przy jego wyliczaniu bazujemy na spadkach i wzrostach cen z danego zakresu. Matematyczna formuła do wyliczenia wskaźnika ADX wygląda następująco

$$ADX_t = 100 \frac{|DI_t^+ - DI_t^-|}{DI_t^+ + DI_t^-}. \quad (3.5)$$

Wartości DI^+ oraz DI^- są wyliczane w tym przypadku jako stosunek EMA z ostatnich n wzrostów lub spadków cen do wartości indeksu ATR. Matematycznie można to przedstawić następującym wzorem

$$DI_t^+ = 100 \frac{EMA_t(DM^+)}{ATR_t}, \quad (3.6)$$

zaś sam wskaźnik ATR poniższym wzorem

$$ATR_t = EMA_t(\max(H_t - L_t, |H_t - C_{t-1}|, |L_t - C_{t-1}|)). \quad (3.7)$$

Interpretacja wskaźnika ADX opiera się, więc na stosunku DI^+ do DI^- , czyli reprezentacji siły trendu w dwóch różnych kierunkach. Sama wartość wskaźnika jest zawsze dodatnia, a przyjęta interpretacja jest taka, że dla ADX poniżej 20 mamy do czynienia z niskim trendem, a powyżej 25 z wysokim. Dodatkowo, żeby dowiedzieć się jaki kierunek ma trend, trzeba porównać wartość DI^+ z DI^- . Wygodnie jest przedstawić całe rozumowanie na wykresie z narysowanymi liniami dla wartości ADX, DI^+ oraz DI^- .

Oscylator stochastyczny

Oscylator stochastyczny to wskaźnik momentum bazujący na porównaniu ceny zamknięcia do cen zamknięcia z n ostatnich obserwacji

$$\%K_t = 100 \frac{C_t - Ln_t}{Hn_t - Ln_t}. \quad (3.8)$$

Wygodnie jest go odczytywać na wykresie razem z 3-dniową średnią ruchomą wyliczoną na podstawie jego wartości. W momencie przecięcia się obydwu linii mówimy o sygnale transakcyjnym, czyli sytuacji korzystnej do kupna lub sprzedaży. Przy trendzie rosnącym wartość oscylatora jest zbliżona do 100, bo cena zamknięcia jest bliska cenie najwyższej z zadanego zakresu. Przy trendzie malejącym wartość oscylatora zbliża się do zera.

Wstęga Bollingera

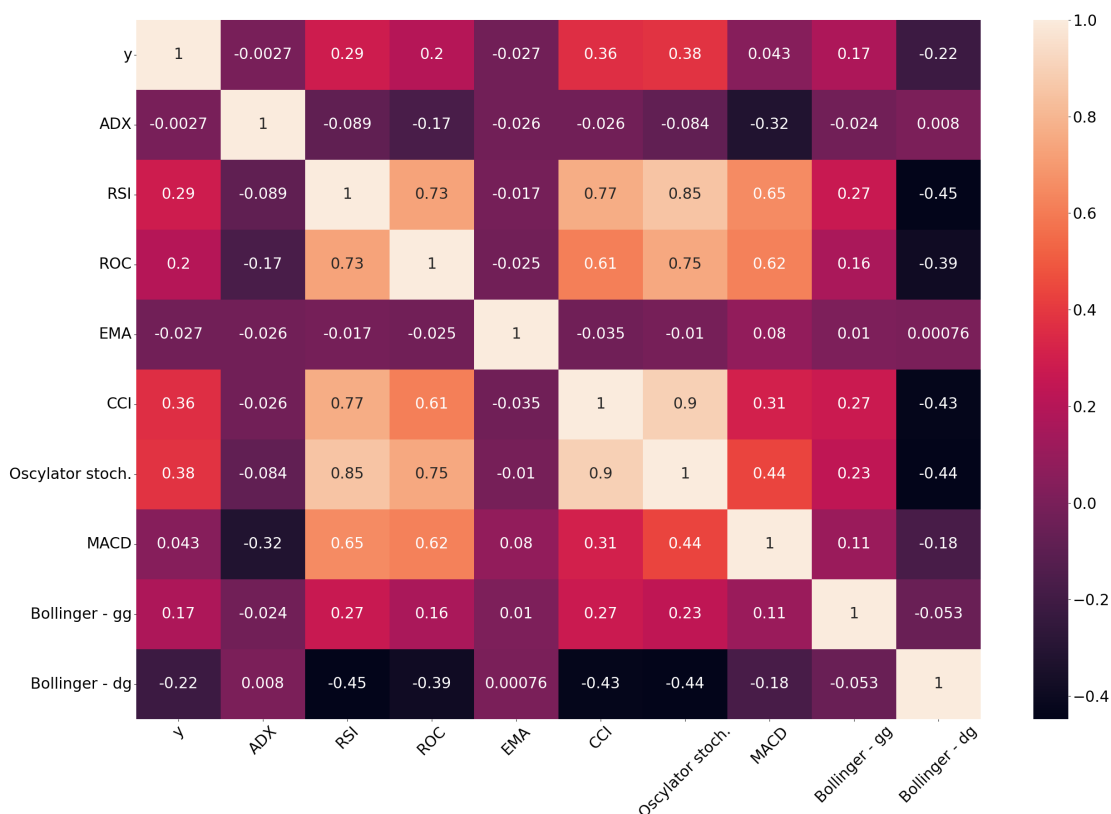
Wstęga Bollingera jest wskaźnikiem zmienności ceny. Jego konstrukcja opiera się na wyliczeniu średniej ruchomej wartości typowych z ostatnich n obserwacji. Kolejnym krokiem jest wyznaczenie górnej i dolnej granicy wstęgi, jako linii oddalonych o zadaną liczbę odchyłeń standardowych od średniej ruchomej. Kluczowe w samej konstrukcji wstęgi jest istnienie granic. Gdy cena zbliża się do dolnej lub górnej granicy może to być sygnałem o odpowiednio niedoszacowaniu lub przeszacowaniu akcji. Ważna jest również szerokość samej wstęgi. Im jest węższa tym mniejsze odchylenie standardowe, czyli niska zmienność ceny. Może to świadczyć o nadchodzącej zmianie trendu.

3.2. Wybór zmiennych

Jednym z kluczowych problemów przy estymacji modeli jest dobór odpowiednich zmiennych. W przypadku niniejszej pracy dane wyjściowe zawierają podstawowe dane dotyczące cen akcji, czy wolumenu. Na ich podstawie obliczane są indeksy analizy technicznej. Wcześniej zostało już wspomniane, że istnieje bardzo dużo takich indeksów, więc sam problem doboru zmiennych jest czasochłonny i nietrywialny. Na dodatek same zmienne mogą być silnie zależne od siebie. Trzeba tu wspomnieć również o ograniczeniach technicznych. Algorytmy uczenia maszynowego są dosyć wymagające w kontekście złożoności obliczeniowej. O ile drzewa decyzyjne są szybkimi algorytmami, o tyle modele SVM są znacznie wolniejsze i trzeba tu zaznaczyć, że to właśnie liczba cech zaraz obok rozmiaru próbki badawczej jest jednym z głównych parametrów wpływających na czas potrzebny na wykonanie obliczeń. Z tego powodu konieczny jest odpowiedni dobór zmiennych do generacji modeli.

Zmienne zostały wybrane bazując na metodach selekcji postępującej, rekursywnej eliminacji zmiennych, a na koniec sprawdzono również współczynniki korelacji zmiennych między sobą. Wszystkie rozważania przeprowadzono na podstawie pełnych danych indeksu S&P 500. Metoda selekcji postępującej polega na estymacji modelu na coraz większym zakresie zmiennych. W pierwszej iteracji model składa się z jednej zmiennej, a w następnych krokach ta liczba jest zwiększana. Algorytm trwa do momentu osiągnięcia zadanej wartości granicznej współczynnika trafności. Jeśli w kolejnej iteracji współczynnik trafności nie zmienił się przynajmniej o wartość graniczną to algorytm jest przerywany. Odwrotnie jest w algorytmie rekursywnej eliminacji zmiennych. W nim zaczynamy ze wszystkimi zmiennymi, a w każdym kolejnym kroku pozbywamy się zadanej liczby najmniej istotnych zmiennych dla modelu. Iteracje są powtarzane do momentu osiągnięcia zadanej liczby parametrów. Wzięto również pod uwagę korelację pomiędzy poszczególnymi cechami oraz zmienną objaśnianą. Ostatecznie wybrano takie zmienne, które jednocześnie zostały wskazane przez obydwa modele selekcji zmiennych, a dodatkowo cechowały się jak najniższą korelacją z innymi zmiennymi objaśniającymi i jak najwyższą korelacją ze zmienną objaśnianą. Lista zmiennych została ograniczona do 9: eksponencjalnej średniej ruchomej (EMA), wskaźników ROC, RSI, CCI, MACD, ADX, oscylatora stochastycznego oraz 2 zmiennych binarnych bazujących na wstędze Bollingera, czyli wskaźników przekroczenia górnej i dolnej granicy wstęgi (zmienne przyjmują wartość 1 w momencie przekroczenia). Na Rys. 3.1 zaprezentowano tabelę korelacji między zmiennymi wybranymi do modelu.

Można zauważyć, że niektóre zmienne są silnie skorelowane ze sobą, np. CCI i oscylator stochastyczny albo słabo skorelowane ze zmienną objaśnianą - ADX, EMA, MACD.



Rys. 3.1. Tabela korelacji między zmiennymi wybranymi do zbudowania modelu

Źródło: opracowanie własne

Stoi to w opozycji do przyjętego kryterium o eliminacji zmiennych, które wykazywały takie tendencje. Same modele wykazywały jednak zauważalnie niższą trafność prognoz po ich eliminacji. Z tego powodu ostatecznie uwzględniono je w rozważaniach.

Dodatkowo same zmienne zostały zestandaryzowane zgodnie ze wzorem

$$z = \frac{x - \bar{x}}{s_x}, \quad (3.9)$$

gdzie \bar{x} i s_x są odpowiednio wartością średnią i odchyleniem standardowym zmiennej x na badanym zbiorze danych. Sama standaryzacja nie jest wymagana przy rozpatrywaniu drzew decyzyjnych, ale może wpływać na wyniki w regresji logistycznej oraz SVM. Należy zauważyć konieczność jej użycia po dodaniu członu regularyzującego we wzorach 2.23 oraz 2.35. Zmienne charakteryzujące się niższymi średnimi wartościami mogą uzyskiwać wyższe wartości parametrów β , dlatego żeby wyeliminować ten efekt stosuje się standaryzację zmiennych.

3.3. Zbiór testowy i treningowy

Kolejnym ważnym krokiem poprzedzającym proces estymacji modeli jest podział danych na zbiór testowy oraz uczący. Jest to konieczne ze względu na możliwość jednoczesnego uczenia modelu, a następnie jego walidacji przy użyciu jednego zbioru danych. Algorytm jest w tym przypadku prosty i polega na losowym podziale danych. Stosuje się różne proporcje podziału, ale zwykle zbiór treningowy jest większy od testowego, dzięki temu jest mniejsza szansa, że model będzie obciążony błędami wynikającymi z użycia zbyt małej próby treningowej. W niniejszej pracy użyto do podziału proporcji 4:1, czyli zbiór treningowy był 4 razy większy od testowego. Oznacza to, że zbiór treningowy zawierał około 2600 obserwacji (w zależności od wybranego indeksu giełdowego), a testowy 650.

3.4. Walidacja krzyżowa

Ostatnim ważnym etapem jest zastosowanie walidacji krzyżowej w całym procesie. Walidacja krzyżowa polega na podziale zbioru treningowego na mniejsze zbiory w celu lepszego oszacowania rzeczywistych możliwości modelu. Normalnie bez walidacji krzyżowej jesteśmy w stanie jednocześnie wygenerować jeden model i na nim sprawdzić możliwości predykcyjne na zbiorze testowym. Dzięki dodatkowemu podziałowi zbioru treningowego estymujemy model tyle razy, ile dodatkowych zbiorów walidacyjnych wygenerowaliśmy, a ostatecznie zdolności predykcyjne są po prostu szacowane jako średnia ze wszystkich iteracji. Walidacja znajduje jeszcze jedno ważne zastosowanie. Skoro dzięki niej jesteśmy w stanie lepiej oszacować możliwości modelu, to można ją również wykorzystać do porównania modeli z różnymi parametrami. Możemy to oczywiście zrobić i bez walidacji krzyżowej, ale również tutaj średnia wyciągnięta z poszczególnych iteracji zapewnia dokładniejszy szacunek. Jesteśmy więc w stanie z większą dokładnością wskazać model z najlepiej dobranymi hiperparametrami. Niestety to wszystko nie jest bez konsekwencji. Zastosowanie walidacji krzyżowej powoduje wydłużenie procesu generacji modelu. Ważne jest więc odpowiednie dobranie liczby zbiorów walidacyjnych. W przypadku niniejszej pracy zdecydowano się użyć podziału na 5 części. Podział zastosowano zarówno przy doborze optymalnych hiperparametrów jak również do końcowej estymacji modelu.

3.5. Ocena poprawności modelu

W przypadku problemów klasyfikacyjnych głównym wskaźnikiem oceny poprawności modelu jest trafność predykcji w stosunku do rzeczywistej przynależności zmiennej objaśnianej do klasy. Najłatwiej można to zobrazować za pomocą macierzy pomyłek. Poniżej zaprezentowano macierz pomyłek modelu klasyfikacyjnego dla dwóch możliwych klas.

Tabela 3.1. Macierz pomyłek dla modelu z dwiema możliwymi klasami zmiennej objaśnianej

		Wartość rzeczywista	
		1	0
Prognoza	1	TP	FP
	0	FN	TN

Wartości TP (*true positive*) i TN (*true negative*) występują w przypadku zgodności obydwu wyników - czyli wartość rzeczywista jest taka sama jak wartość predykcji. FN (*false negative*) i FP (*false positive*) występują w przypadku niezgodności klasy przewidzianej w stosunku do rzeczywistej wartości. Bazując na macierzy pomyłek można skonstruować wskaźniki zdolności predykcyjnych modeli.

Wskaźnik trafności (*accuracy*) jest obliczany jako stosunek trafnych prognoz (TP oraz TN) do liczby wszystkich prognoz

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP}. \quad (3.10)$$

Do budowy wskaźnika F1 wykorzystujemy miarę czułości (*sensitivity*) i precyzji (*precision*) modelu. Czułość definiowana jest jako

$$sensitivity = \frac{TP}{TP + FN}, \quad (3.11)$$

zaś precyzja jako

$$precision = \frac{TP}{TP + FP}. \quad (3.12)$$

Analizując wskaźnik czułości badamy jaką część klasy 1 jest objęta poprawnym przewidywaniem. Wskaźnik precyzji z kolei daje informację o tym w jakim stopniu prognozowana wartość klasy 1 pokrywa się z rzeczywistością.

Ostatecznie F1 obliczamy jako

$$F1 = 2 \frac{precision * sensitivity}{precision + sensitivity}. \quad (3.13)$$

Stosowanie wskaźnika F1 ułatwia porównywanie modeli. Zamiast używać precyzji i czułości sprowadzamy porównanie modeli do wartości jednego wskaźnika.

Przed zdefiniowaniem krzywej ROC musimy poznać jeszcze jedną miarę - specyficzność (*specificity*). Definiowana jest jako

$$specificity = \frac{FN}{TN + FP}. \quad (3.14)$$

Konstrukcja krzywej ROC opiera się na pojęciu progu odcięcia, czyli wartości prawdopodobieństwa, dla której przypisujemy obserwację do klasy 1. Dla danego modelu rozpatrujemy wiele wartości progu odcięcia i sprawdzamy jak w danym przypadku zachowują się miary specyficzności i czułości. Rozpatrując wiele punktów odcięcia możemy również przestawić ROC w postaci graficznej. Najlepszym punktem będzie taki, który jednocześnie będzie maksymalizował miarę specyficzności i czułości.

Wskaźnik AUC (*area under curve*) definiujemy jako pole pod wykresem ROC i przyjmuje wartość z przedziału [0, 1]. Generalnie im wyższa wartość wskaźnika AUC tym klasyfikator jest lepszy.

Rozdział 4

Wyniki

Rozdział 4 zawiera wyniki predykcji modeli uczenia maszynowego. Testowano modele opisane w rozdziale 2, czyli drzewa decyzyjne, bagging, lasy losowe, boosting, regresję logistyczną oraz maszynę wektorów nośnych. Celem było stworzenie modelu, który najlepiej będzie przewidywał zmiany cen akcji na 1 lub 2 sesje do przodu. W celu porównania zdolności predykcyjnych poszczególnych modeli posłużono się wskaźnikami trafności, F1 oraz krzywymi ROC i wartościami AUC.

4.1. Drzewa decyzyjne

Pierwszym modelem testowanym w kontekście predykcji zmian cen akcji są drzewa decyzyjne. W pierwszym kroku określono zestaw hiperparametrów modelu. Zestaw ten został użyty tylko przy pierwszej iteracji. W kolejnych krokach zawężano przedziały hiperparametrów, dzięki czemu model był generowany dla coraz bardziej szczegółowych wartości. W każdej iteracji zawsze wybierano taki zestaw hiperparametrów, dla których model osiągał najwyższą trafność. Ze względu na dużą złożoność obliczeniową wybór hiperparametrów przeprowadzono na podstawie zbioru treningowego dla indeksu S&P 500. Zestaw optymalnych parametrów użyto następnie do generacji modeli dla pozostałych indeksów.

Pierwszy zbiór hiperparametrów wraz z przedziałami testowania został przedstawiony w tabeli 4.1.

Korzystając z 5-krotnej walidacji krzyżowej wyznaczono zbiór parametrów, dla których model osiągnął najwyższą trafność - 0,685 ($F1 = 0,685$, $AUC = 0,753$):

- *ccp_alpha*: 0
- *criterion*: "entropy"

Tabela 4.1. Zbiór hiperparametrów testowanych w modelu drzew decyzyjnych (iteracja pierwsza)

Parametr	Zakres
<i>criterion</i>	"gini", "entropy"
<i>max_depth</i>	1, 3, 5, 10, 15, 20, 30, 40
<i>min_samples_split</i>	2, 3, 5, 10, 15
<i>min_samples_leaf</i>	1, 2, 3, 5, 10
<i>ccp_alpha</i>	0, 0.5, 1, 2, 5, 10, 20

- *max_depth*: 5
- *min_samples_leaf*: 10
- *min_samples_split*: 2

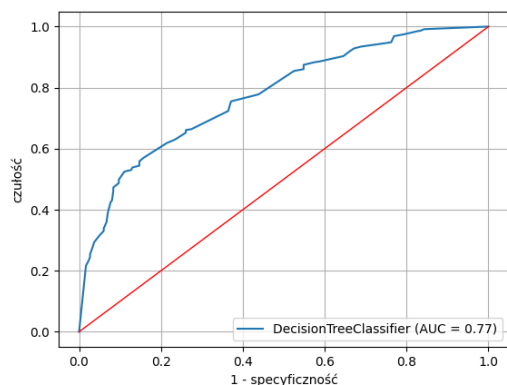
W kolejnej iteracji nie testowano już parametru kryterium podziału (*criterion*). Z uwagi na to, że parametry *ccp_alpha*, *min_samples_split* również zostały jasno określone nie były one testowane w następnej iteracji. Określony został jednak na nowo zbiór dla parametru *max_depth* - [4, 5, 6, 7, 8] oraz *min_samples_leaf* - [7, 10, 12, 15, 18, 21, 25, 30]. W ten sposób znaleziono dokładną wartość parametrów *max_depth* oraz *min_samples_leaf*. Ostatecznie wyznaczono optymalny zbiór parametrów (Tabela 4.2), dla których model osiągnął trafność - 0,692 (F1 = 0,714, AUC = 0,759). Widać więc wzrost trafności oraz wskaźników F1 oraz AUC.

Tabela 4.2. Zbiór hiperparametrów testowanych w modelu drzew decyzyjnych (iteracja końcowa)

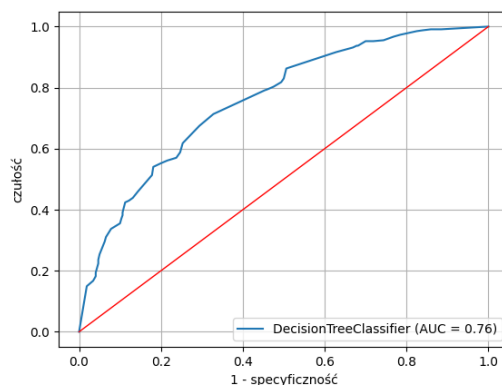
Parametr	Wartość
<i>criterion</i>	"entropy"
<i>max_depth</i>	8
<i>min_samples_split</i>	2
<i>min_samples_leaf</i>	21
<i>ccp_alpha</i>	0

Korzystając z określonego zestawu hiperparametrów wygenerowano modele drzew decyzyjnych na podstawie danych treningowych, dla indeksów WIG20, S&P 500, DAX, Nikkei 225, BSE SENSEX oraz FTSE 100. Uzyskane wyniki dla danych testowych zostały zaprezentowane w tabeli 4.3.

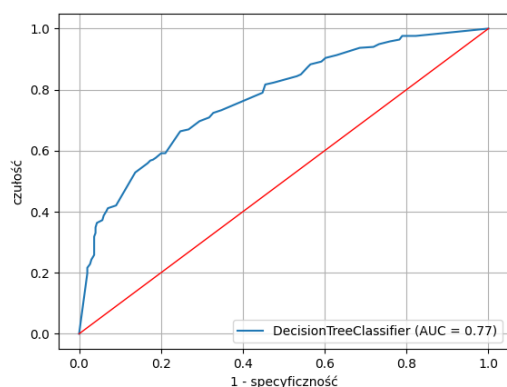
Na rysunku 4.1 zaprezentowano również wykresy krzywych ROC wraz z wartościami AUC dla poszczególnych indeksów.



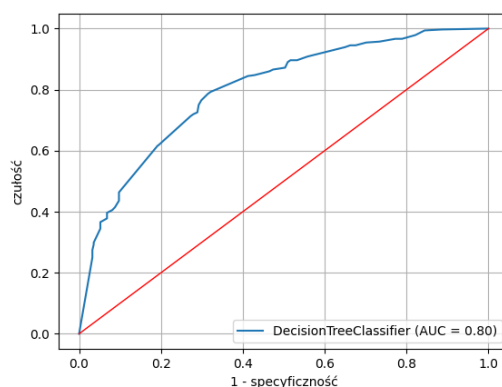
(a) S&P 500



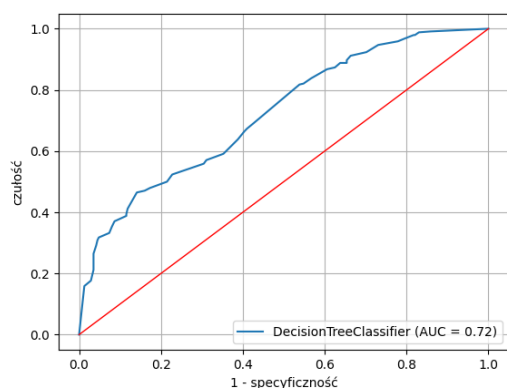
(b) DAX



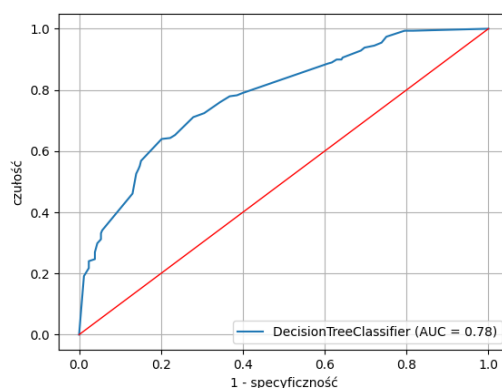
(c) Nikkei 225



(d) BSE Sensex 30



(e) UK 100



(f) WIG 20

Rys. 4.1. Krzywa ROC dla modelu drzew decyzyjnych dla poszczególnych indeksów na danych testowych (horyzont jednosesyjny)

Uzyskane modele charakteryzują się zbliżonymi mocami predykcyjnymi dla każdego z indeksów. Najwyższą trafność osiągnięto dla indeksu WIG 20 - 0,690. Równie wysokie

Tabela 4.3. Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu drzew decyzyjnych na danych testowych (horyzont jednosesyjny)

Indeks	Trafność	F1
S&P 500	0,666	0,693
DAX	0,640	0,660
Nikkei 225	0,680	0,688
BSE Sensex 30	0,689	0,707
UK 100	0,684	0,697
WIG 20	0,690	0,655

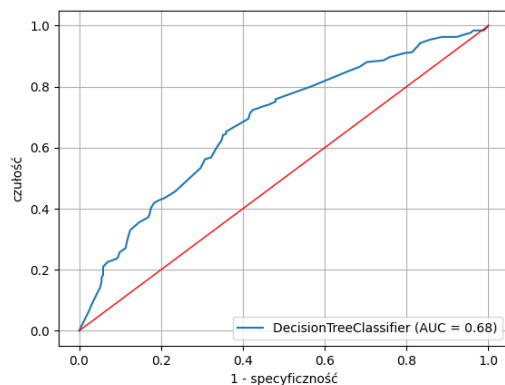
wartości uzyskano dla BSE Sensex 30, Nikkei 225 oraz UK 100. Najgorszą trafność odnotowano dla indeksu DAX. Podobną tendencję można zauważyć analizując uzyskane wartości wskaźnika F1. Jeśli chodzi o uzyskane krzywe ROC to charakteryzują się podobnym przebiegiem. Wartości AUC również to potwierdzają.

Korzystając z gotowego zestawu hiperparametrów wygenerowano dodatkowo model, w którym testowano zdolności predykcyjne na dwie sesje do przodu. W tabeli 4.4 przedstawiono uzyskane wyniki. Na rysunku 4.2 zaprezentowano również wykresy krzywych ROC wraz z wartościami AUC dla poszczególnych indeksów.

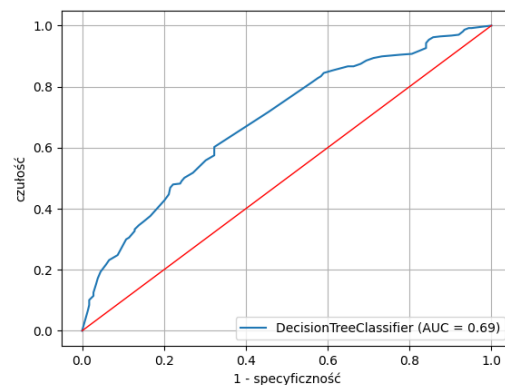
Tabela 4.4. Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu drzew decyzyjnych na danych testowych (horyzont dwusesyjny)

Indeks	Trafność	F1
S&P 500	0,609	0,659
DAX	0,597	0,634
Nikkei 225	0,623	0,627
BSE Sensex 30	0,616	0,638
UK 100	0,629	0,670
WIG 20	0,639	0,574

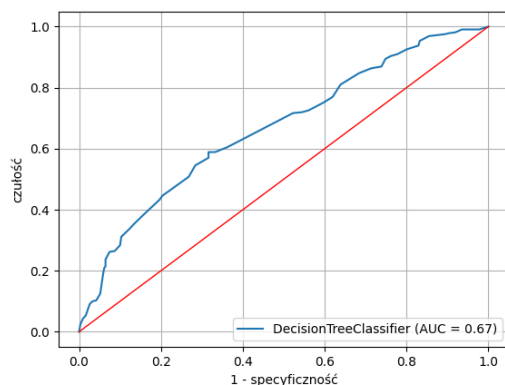
Widać wyraźny spadek wskaźników trafności i F1, nadal jednak trafność utrzymuje się na poziomie 0,59 - 0,64. Podobnie jest dla wskaźników F1. Najwyższą trafność uzyskano dla WIG 20, z kolei najwyższą wartość F1 dla BSE Sensex 30. Kształt krzywych ROC i wartość AUC również świadczą o spadku zdolności predykcyjnych. Dla porównania dla horyzontu jednosesyjnego odnotowano AUC w granicach 0,72 - 0,80, dla horyzontu dwusesyjnego było to już tylko 0,65 - 0,69.



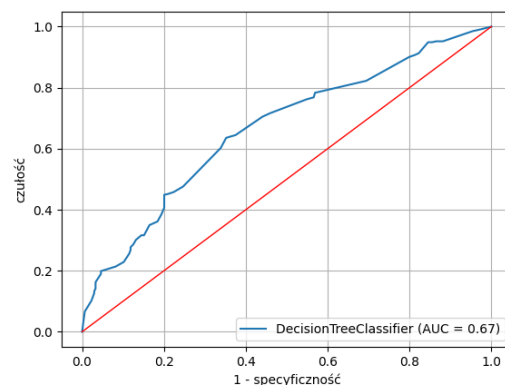
(a) S&P 500



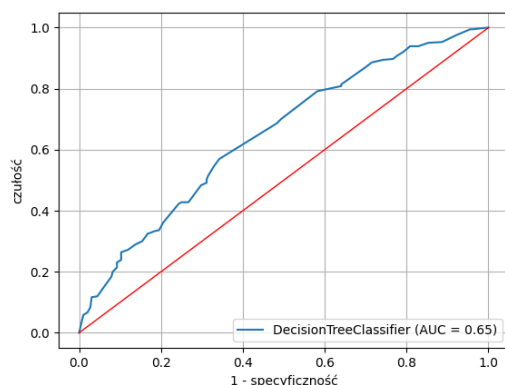
(b) DAX



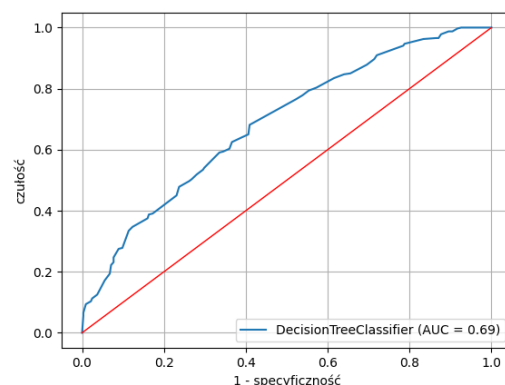
(c) Nikkei 225



(d) BSE Sensex 30



(e) UK 100



(f) WIG 20

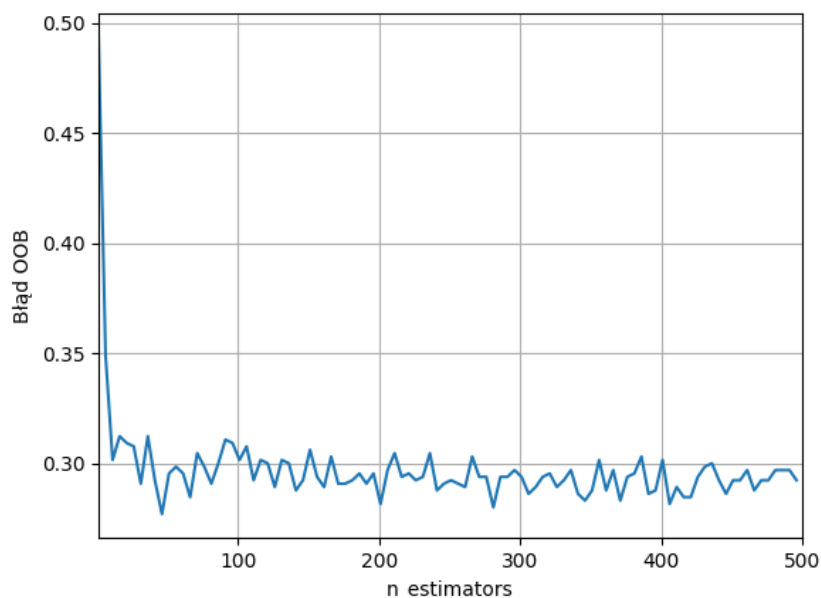
Rys. 4.2. Krzywa ROC dla modelu drzew decyzyjnych dla poszczególnych indeksów na danych testowych (horyzont dwusesyjny)

4.2. Bagging

Kolejnym modelem testowanym do predykcji zmian cen akcji był Bagging. Z uwagi na to, że opiera się na algorytmie drzew decyzyjnych, jako jądro estymacji wykorzystano

model drzew decyzyjnych dla optymalnych parametrów z poprzedniego rozdziału (tabela 4.2). Dodatkowo testowano również hiperparametr dostępny dla modelu bagging, czyli liczbę wygenerowanych drzew decyzyjnych (parametr $n_estimators$) w zakresie [5, 20, 40, 60, 80, 100, 150, 200, 300, 500]. Korzystając z 5-krotnej walidacji krzyżowej najwyższą trafność - 0,727 ($F1 = 0,751$) uzyskano dla $n_parameters = 100$. W kolejnej iteracji zawężono zbiór liczby drzew - [90, 95, 100, 105, 110, 115, 120, 125, 130, 135, 140]. W ten sposób trafność nieznacznie wzrosła do 0,731 ($F1 = 0,754$) dla 90 drzew. Poprawa nie jest więc tak zauważalna jak dla drzew decyzyjnych, ale trzeba mieć na uwadze to, że teraz testowano tylko jeden hiperparametr.

W rozdziale poświęconym modelowi bagging (2.3.1) wspomniane zostało pojęcie obserwacji OOB. Przy okazji modelu bagging wykorzystano je w celu analizy doboru optymalnej liczby drzew. Na wykresie 4.3 zaprezentowano wartość błędu OOB od liczby wygenerowanych drzew w modelu bagging.



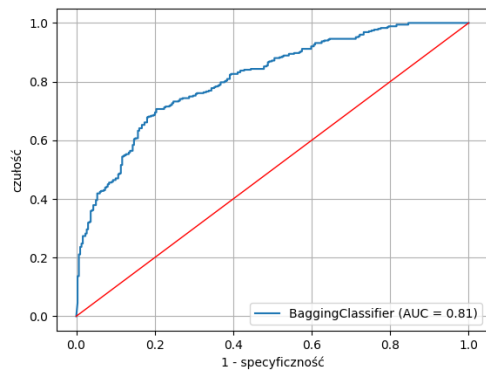
Rys. 4.3. Wykres wartości błędu OOB od hiperparametru $n_estimators$ dla modelu bagging

Wartość błędu została obliczona jako $1 - \text{trafność prognoz dla obserwacji OOB}$. Z wykresu można odczytać znacznie większe wartości błędu OOB dla małej liczby drzew (< 20). Co prawda minimum nie jest osiągnięte dokładnie dla 90 drzew, tak jak wskazywała na to metoda doboru parametrów za pomocą walidacji krzyżowej, ale wartości błędu OOB są do siebie bardzo zbliżone. Z uwagi na to przyjęto ostatecznie wartość wskazaną walidacją krzyżową jako optymalną.

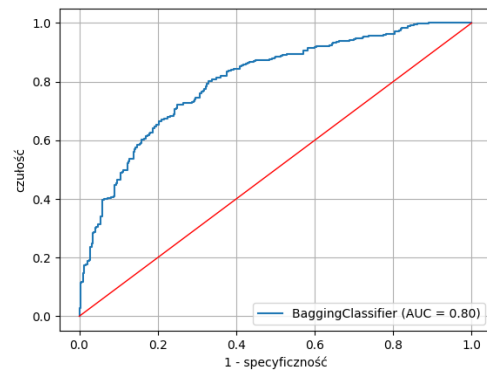
Również model bagging przetestowano dla wszystkich indeksów. Tak jak wcześniej optymalny zestaw parametrów został wybrany na podstawie zbioru treningowego dla

danych S&P 500. W tabeli 4.5 zaprezentowano wskaźniki trafności i F1 dla wszystkich indeksów wygenerowane na podstawie zbiorów testowych.

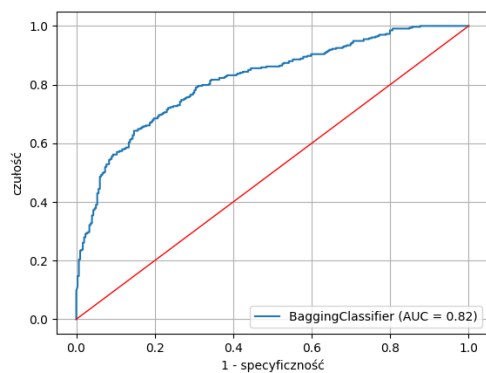
Na rysunku 4.4 zaprezentowano również wykresy krzywych ROC wraz z wartościami AUC dla poszczególnych indeksów.



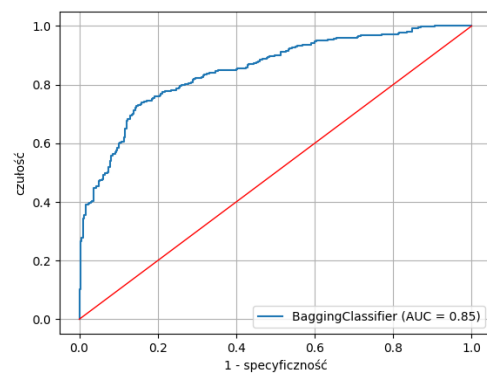
(a) S&P 500



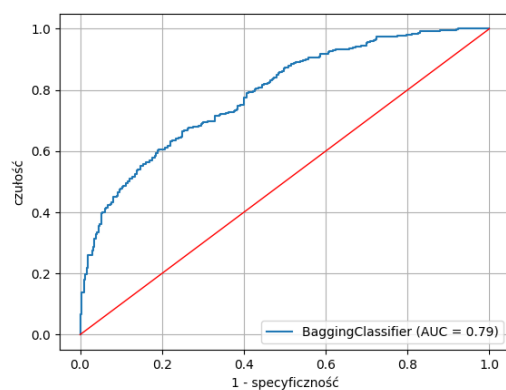
(b) DAX



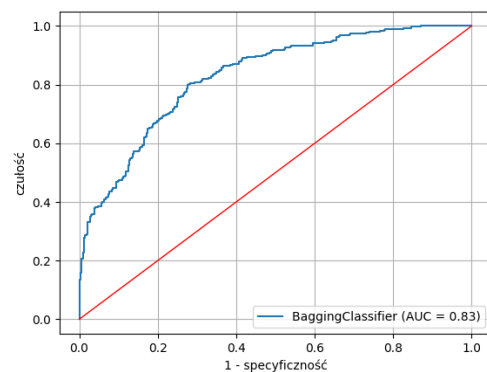
(c) Nikkei 225



(d) BSE Sensex 30



(e) UK 100



(f) WIG 20

Rys. 4.4. Krzywa ROC dla modelu bagging dla poszczególnych indeksów na danych testowych (horyzont jednosesyjny)

Tabela 4.5. Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu bagging na danych testowych (horyzont jednosesyjny)

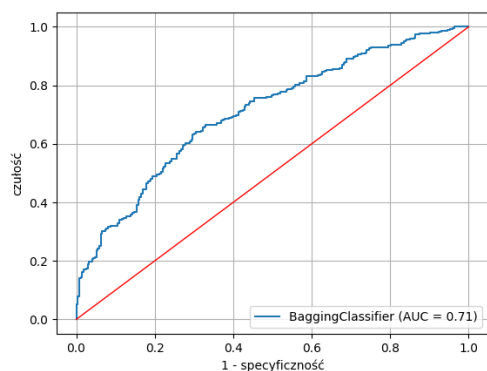
Indeks	Trafność	F1
S&P 500	0,720	0,747
DAX	0,670	0,651
Nikkei 225	0,718	0,726
BSE Sensex 30	0,722	0,721
UK 100	0,713	0,709
WIG 20	0,704	0,672

Uzyskane modele charakteryzują się zbliżonymi mocami predykcyjnymi dla każdego z indeksów. Najwyższą trafność osiągnięto dla indeksów S&P 500, BSE Sensex 30, UK 100 oraz Nikkei 225. Niewiele niższą trafność wykazał model dla indeksu WIG 20. Wyraźnie niższą wartość uzyskał znowu model dla indeksu DAX. Podobną tendencję można zauważyć analizując uzyskane wartości wskaźnika F1. Jeśli chodzi o krzywe ROC to charakteryzują się podobnym przebiegiem i wartościami AUC w granicach 0,8.

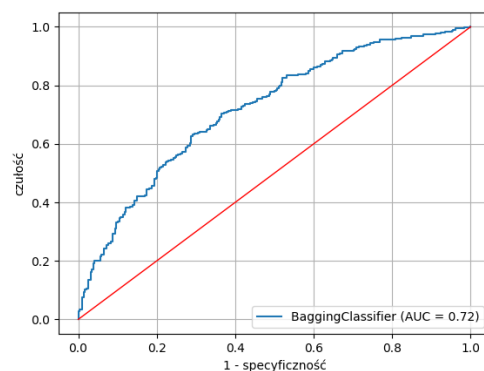
Korzystając z gotowego zestawu hiperparametrów wygenerowano dodatkowo model, w którym testowano zdolności predykcyjne na dwie sesje do przodu. W tabeli 4.6 przedstawiono uzyskane wyniki. Na rysunku 4.5 zaprezentowano również wykresy krzywych ROC wraz z wartościami AUC dla poszczególnych indeksów.

Tabela 4.6. Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu bagging na danych testowych (horyzont dwusesyjny)

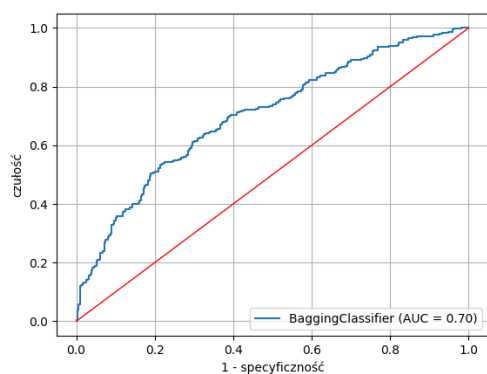
Indeks	Trafność	F1
S&P 500	0,634	0,699
DAX	0,627	0,671
Nikkei 225	0,680	0,672
BSE Sensex 30	0,676	0,675
UK 100	0,638	0,684
WIG 20	0,650	0,641



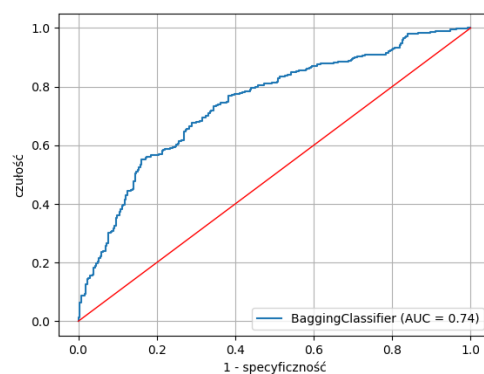
(a) S&P 500



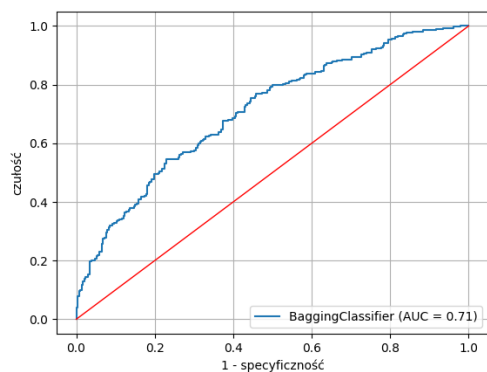
(b) DAX



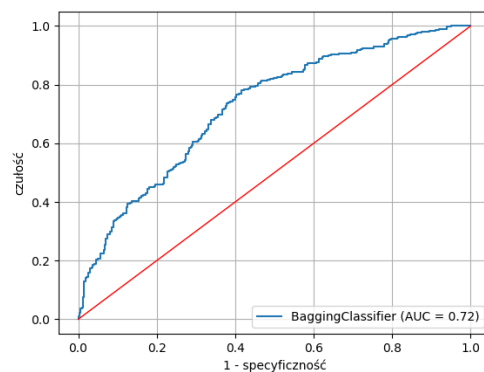
(c) Nikkei 225



(d) BSE Sensex 30



(e) UK 100



(f) WIG 20

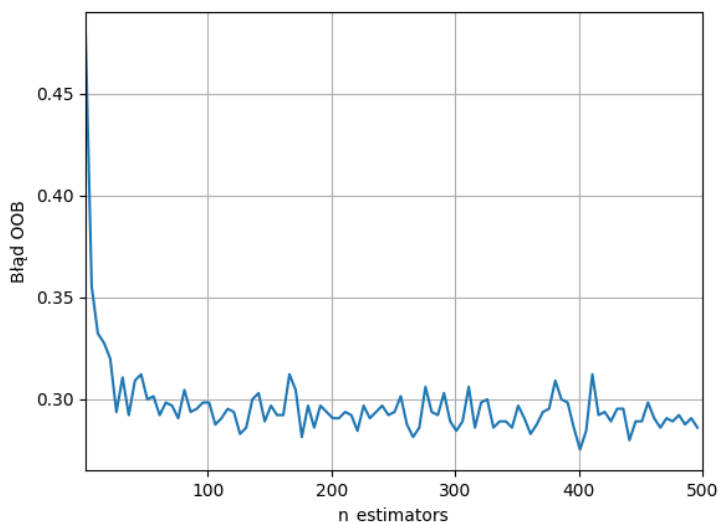
Rys. 4.5. Krzywa ROC dla modelu bagging dla poszczególnych indeksów na danych testowych (horyzont dwusesyjny)

Znowu widać wyraźny spadek wskaźników trafności i F1 jednak są one wyższe zwłaszcza dla indeksów Nikkei 225 oraz BSE Sensex 30 w porównaniu z drzewami decyzyjnymi. Takie same wnioski nasuwają się po analizie krzywych ROC. Dla wszystkich indeksów mają podobny kształt, ale wartość AUC oscyluje tym razem w granicach 0,70 - 0,74.

4.3. Lasy losowe

Model lasów losowych również opiera się na algorytmie drzew decyzyjnych, dlatego jako jądro estymacji wykorzystano model drzew decyzyjnych dla optymalnych parametrów (tabela 4.2). Dodatkowo testowano również hiperparametry dostępne dla modelu lasów losowych, czyli liczbę wygenerowanych drzew decyzyjnych (parametr *n_estimators*) w zakresie [5, 20, 40, 60, 80, 100, 150, 200, 300, 500] oraz liczbę cech w pojedynczej iteracji (parametr *max_features*) w zakresie ["sqrt", "log2", 2, 3]. Korzystając z 5-krotnej walidacji krzyżowej najwyższą trafność - 0,719 (F1 = 0,739) uzyskano dla *n_estimators* = 60 i *max_features* = "log2". W kolejnej iteracji zawężono zbiór liczby drzew - [42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78]. Parametr *max_features* nie testowano w drugiej iteracji i przyjęto dla niego wartość "log2". W ten sposób trafność nieznacznie wzrosła do 0.721 (F1 = 0.739) dla 42 drzew.

Również przy okazji modelu lasów losowych skorzystano z metody wyznaczania hiperparametrów modelu na podstawie analizy błędu OOB. Na wykresie 4.3 zaprezentowano wartość błędu OOB od liczby wygenerowanych drzew w modelu lasów losowych.



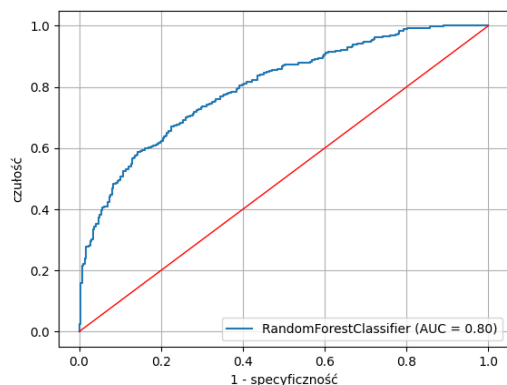
Rys. 4.6. Wykres wartości błędu OOB od hiperparametru *n_estimators* dla modelu lasów losowych

Na wykresie wyraźnie widać znacznie większe wartości błędu OOB dla małej liczby drzew (<50). Dla większej liczby drzew wartości błędu są do siebie bardzo zbliżone, chociaż najniższe wartości odnotowano dla około 400 drzew. Z uwagi na bardzo zbliżone wartości błędu OOB dla poszczególnych wartości parametru *n_estimators*, przyjęto ostatecznie wartość wskazaną walidacją krzyżową jako optymalną.

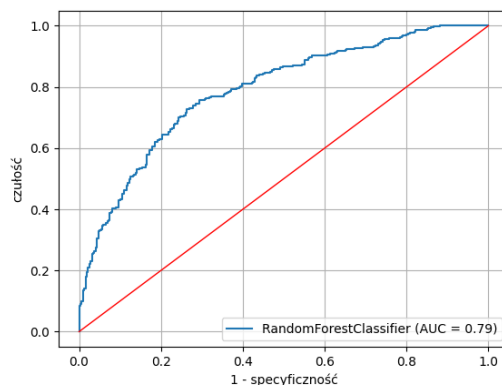
Również model lasów losowych przetestowano dla wszystkich indeksów. W tabeli 4.7

zaprezentowano wskaźniki trafności i F1 dla wszystkich indeksów, wygenerowane na podstawie zbiorów testowych.

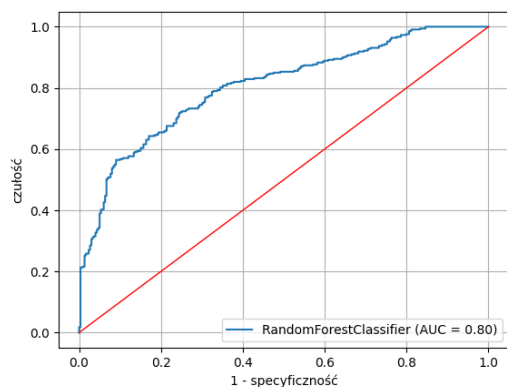
Na rysunku 4.7 zaprezentowano również wykresy krzywych ROC wraz z wartościami AUC dla poszczególnych indeksów.



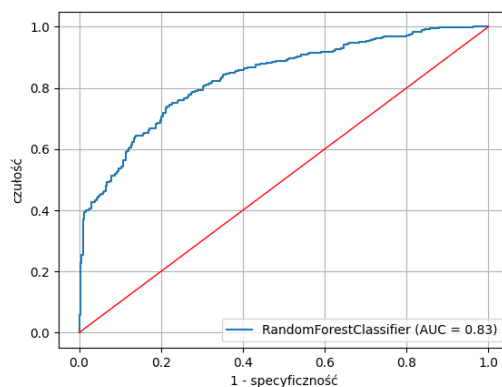
(a) S&P 500



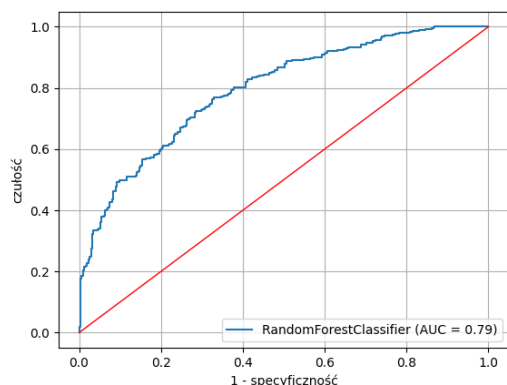
(b) DAX



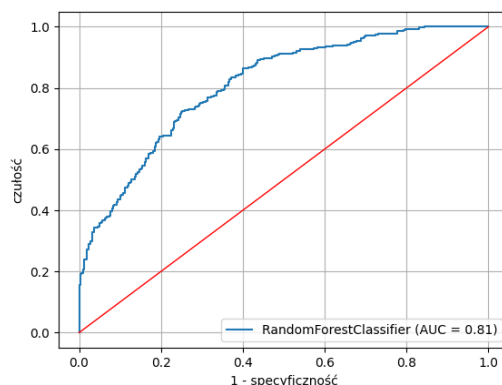
(c) Nikkei 225



(d) BSE Sensex 30



(e) UK 100



(f) WIG 20

Rys. 4.7. Krzywa ROC dla modelu lasów losowych dla poszczególnych indeksów na danych testowych (horyzont jednodesyjny)

Tabela 4.7. Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu lasów losowych na danych testowych (horyzont jednosesyjny)

Indeks	Trafność	F1
S&P 500	0,706	0,709
DAX	0,672	0,666
Nikkei 225	0,695	0,693
BSE Sensex 30	0,711	0,703
UK 100	0,706	0,708
WIG 20	0,687	0,646

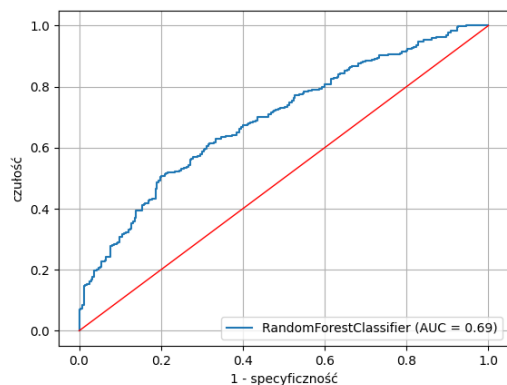
Chociaż znów zdolności predykcyjne dla każdego z indeksów są zbliżone, to sam model osiąga niższe trafności i F1 w porównaniu z modelem bagging. Najniższą trafność znów odnotowano dla indeksu DAX. Podobną tendencję można zauważyć analizując uzyskane wartości wskaźnika F1. Jeśli chodzi o uzyskane krzywe ROC to charakteryzują się podobnym przebiegiem i wartościami AUC w granicach 0,8.

Korzystając z gotowego zestawu hiperparametrów wygenerowano dodatkowo model, w którym testowano zdolności predykcyjne na dwie sesje do przodu. W tabeli 4.8 przedstawiono uzyskane wyniki. Na rysunku 4.8 zaprezentowano również wykresy krzywych ROC wraz z wartościami AUC dla poszczególnych indeksów.

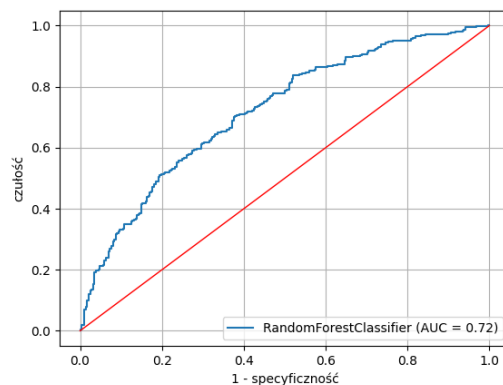
Tabela 4.8. Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu lasów losowych na danych testowych (horyzont dwusesyjny)

Indeks	Trafność	F1
S&P 500	0,651	0,703
DAX	0,617	0,688
Nikkei 225	0,645	0,635
BSE Sensex 30	0,662	0,663
UK 100	0,640	0,693
WIG 20	0,654	0,632

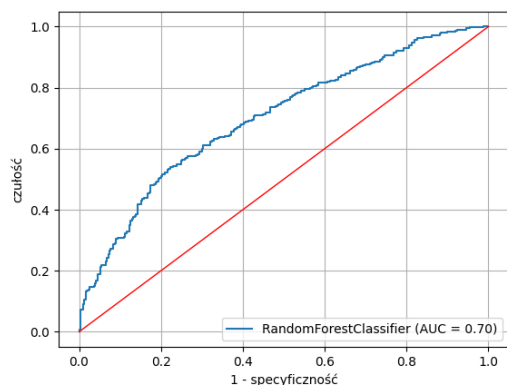
Znowu uzyskano trafność w okolicach 0,61 - 0,65. Najwyższe wartości trafności odnotowano dla S&P 500 oraz WIG 20. Z kolei najniższą trafność otrzymano dla indeksu DAX. Wykresy krzywych ROC są wyraźnie spłaszczone w porównaniu z wykresami dla horyzontu jednosesyjnego. Wartości AUC oscylują w granicach 0,70 - 0,74.



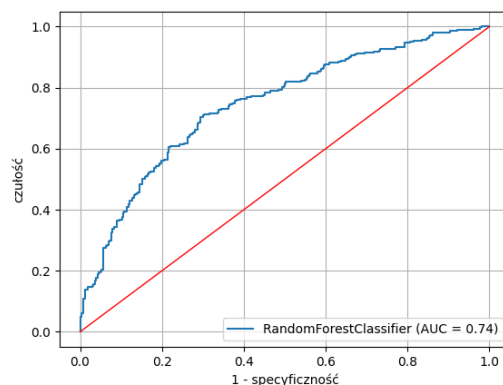
(a) S&P 500



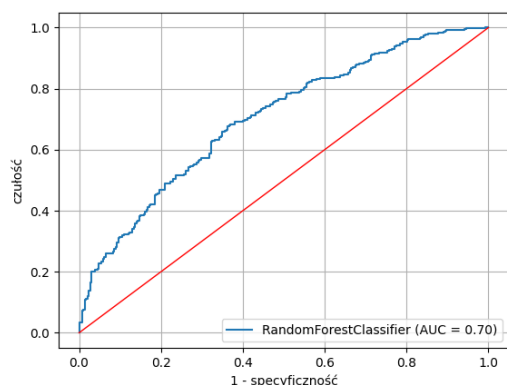
(b) DAX



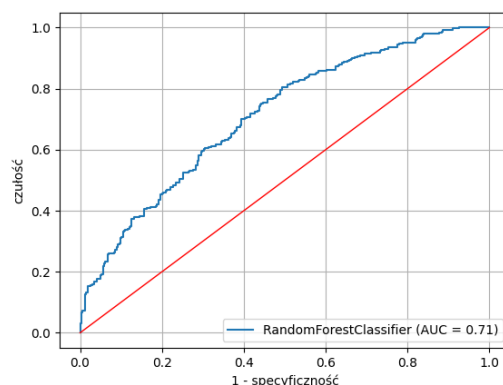
(c) Nikkei 225



(d) BSE Sensex 30



(e) UK 100



(f) WIG 20

Rys. 4.8. Krzywa ROC dla modelu lasów losowych dla poszczególnych indeksów na danych testowych (horyzont dwusesyjny)

4.4. Boosting

Ostatnim testowanym modelem opartym na idei drzew decyzyjnych jest boosting. Do predykcji wykorzystano klasyfikacyjny model AdaBoost, który jest jednym z wariantów

boostingu.

Tak jak dla modelu lasów losowych i baggingu, jako jądro estymacji wykorzystano model drzew decyzyjnych dla optymalnych parametrów (tabela 4.2). Dodatkowo testowano również hiperparametry dostępne dla modelu AdaBoost, czyli liczbę wygenerowanych drzew decyzyjnych (parametr *n_estimators*) w zakresie [50, 75, 100, 150, 200, 300, 500, 1000], parametr uczenia (*learning_rate*) w zakresie [0,0001; 0,001; 0,01; 0,1; 0,5; 1; 2; 5; 10] oraz liczbę węzłów końcowych w drzewie w każdej iteracji (parametr *max_leaf_nodes*) w zakresie [null, 2, 3, 5, 7, 10, 20] (wartość "null" parametru *max_leaf_nodes* oznacza brak ograniczenia na liczbę węzłów).

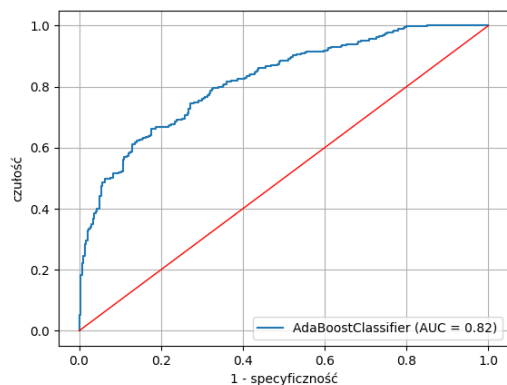
Korzystając z 5-krotnej walidacji krzyżowej najwyższą trafność - 0,734 (F1 = 0,756) uzyskano dla *n_estimators* = 1000, *learning_rate* = 0,1 oraz *max_leaf_nodes* = 2. W kolejnej iteracji zmieniono zakres liczby drzew w modelu - [600, 800, 1000, 1200, 1400, 1600, 1800, 2000, 2500] oraz parametr uczenia - [0,04; 0,07; 0,10; 0,015; 0,020; 0,025; 0,030; 0,035; 0,040; 0,045]. Parametr *max_leaf_nodes* nie był testowany w dalszych iteracjach, ponieważ został dokładnie określony. Trafność w drugiej iteracji nieznacznie wzrosła do 0,739 (F1 = 0,761) dla 2500 drzew i parametru uczenia 0,07. W końcowej iteracji testowano już tylko liczbę drzew w pojedynczym modelu. W tym celu określono nowy zbiór - [2250, 2500, 2750, 3000, 3500]. Najwyższą trafność uzyskano jednak dla 2500 drzew, czyli dla dokładnie takich samych parametrów jak w drugiej iteracji. Ostateczne wartości hiperparametrów zaprezentowano w tabeli 4.9.

Tabela 4.9. Zbiór hiperparametrów testowanych w modelu AdaBoost (iteracja końcowa)

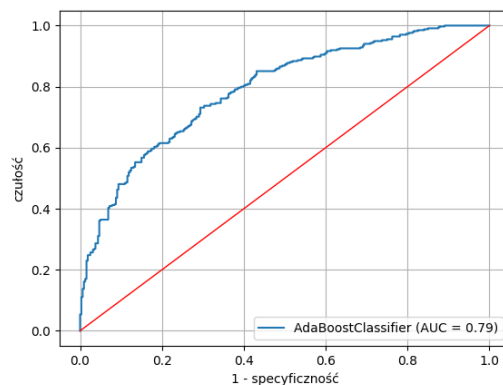
Parametr	Wartość
<i>criterion</i>	"entropy"
<i>max_depth</i>	8
<i>min_samples_split</i>	2
<i>min_samples_leaf</i>	21
<i>learning_rate</i>	0,07
<i>n_estimators</i>	2500
<i>max_leaf_nodes</i>	2

W tabeli 4.10 zaprezentowano wskaźniki trafności i F1 dla wszystkich indeksów, wygenerowane na podstawie zbiorów testowych.

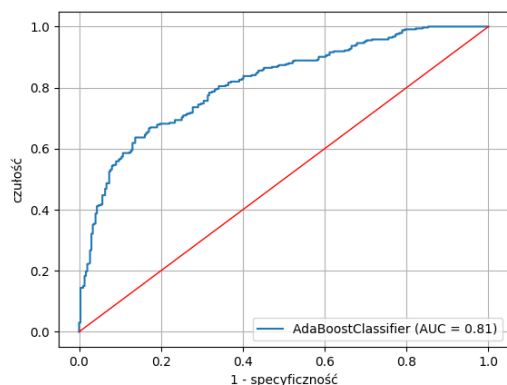
Na rysunku 4.9 zaprezentowano również wykresy krzywych ROC wraz z wartościami AUC dla poszczególnych indeksów.



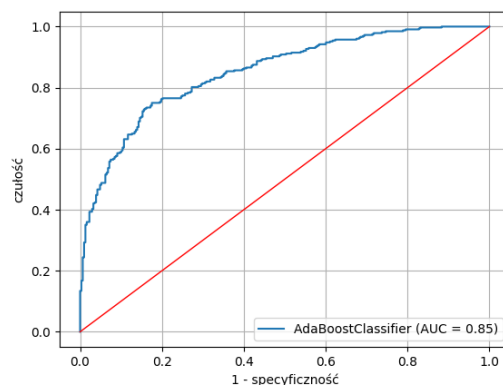
(a) S&P 500



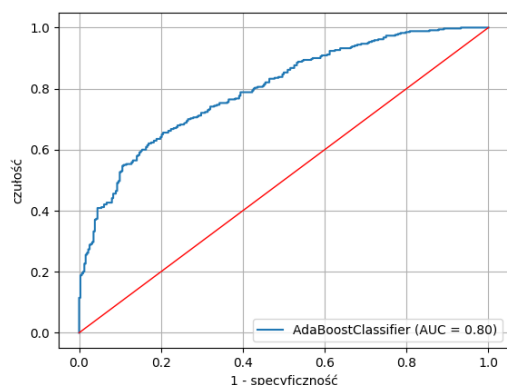
(b) DAX



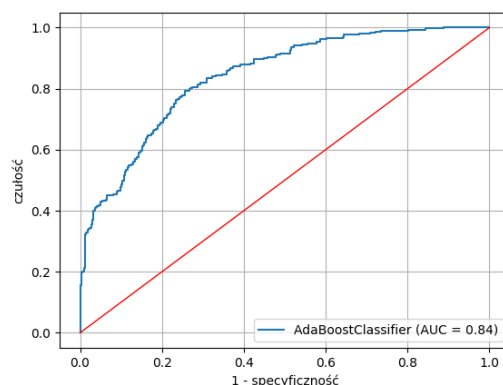
(c) Nikkei 225



(d) BSE Sensex 30



(e) UK 100



(f) WIG 20

Rys. 4.9. Krzywa ROC dla modelu boosting dla poszczególnych indeksów na danych testowych (horyzont jednosesyjny)

Wyniki z modelu AdaBoost są bardzo zbliżone do uzyskanych modelem lasów losowych. Jednocześnie znowu nie udało się osiągnąć trafności i F1 zbliżonych do baggingu. Najlepsze prognozy uzyskano dla indeksów S&P 500, BSE Sensex 30 oraz WIG 20. Nieco gorsze prognozy uzyskano dla Nikkei 225 i znowu najłabsze trafności uży-

Tabela 4.10. Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu boosting na danych testowych (horyzont jednosesyjny)

Indeks	Trafność	F1
S&P 500	0,708	0,728
DAX	0,661	0,672
Nikkei 225	0,695	0,713
BSE Sensex 30	0,712	0,715
UK 100	0,702	0,716
WIG 20	0,712	0,700

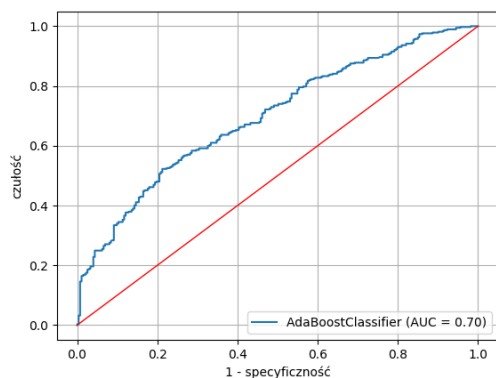
skłał indeks DAX. Wszystkie wykresy ROC są do siebie bardzo zbliżone tak jak dla poprzednich testowanych modeli. Tym razem jednak wartość AUC nie spadła poniżej 0,8 dla żadnego modelu. Najwyższe AUC uzyskano dla indeksów BSE Sensex oraz WIG 20.

Korzystając z gotowego zestawu hiperparametrów wygenerowano dodatkowo model, w którym testowano zdolności predykcyjne na dwie sesje do przodu. W tabeli 4.11 przedstawiono uzyskane wyniki. Na rysunku 4.10 zaprezentowano również wykresy krzywych ROC wraz z wartościami AUC dla poszczególnych indeksów.

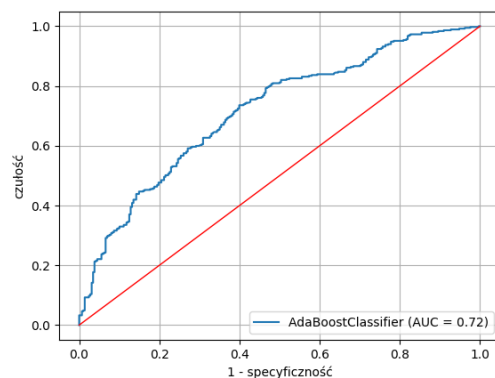
Tabela 4.11. Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu boosting na danych testowych (horyzont dwusesyjny)

Indeks	Trafność	F1
S&P 500	0,657	0,714
DAX	0,597	0,649
Nikkei 225	0,635	0,625
BSE Sensex 30	0,651	0,665
UK 100	0,624	0,680
WIG 20	0,612	0,611

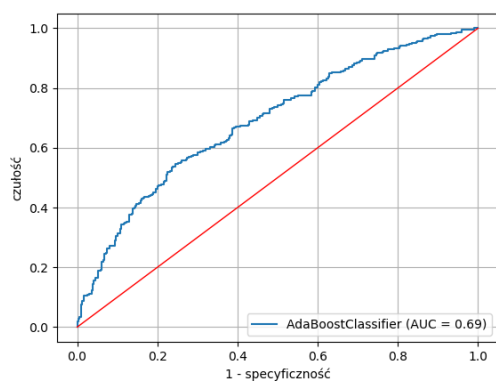
Trafności dla horyzontu 2-dniowego oscylują w granicach 0,63 - 0,68. Najwyższe wartości trafności odnotowano dla BSE Sensex 30 oraz S&P 500. Z kolei najniższą trafność otrzymano znowu dla indeksu DAX. Wykresy krzywych ROC są wyraźnie spłaszczone w porównaniu z wykresami dla horyzontu jednosesyjnego. Wartości AUC oscylują w granicach 0,69 - 0,75.



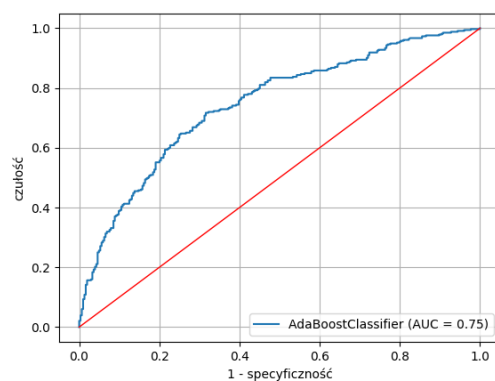
(a) S&P 500



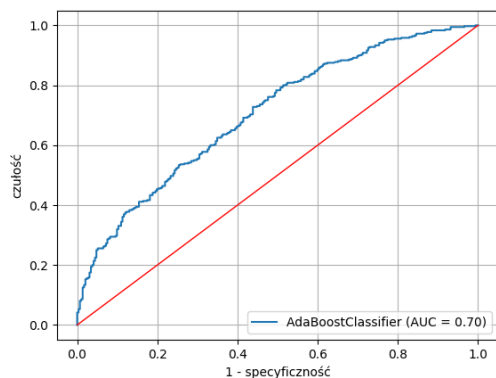
(b) DAX



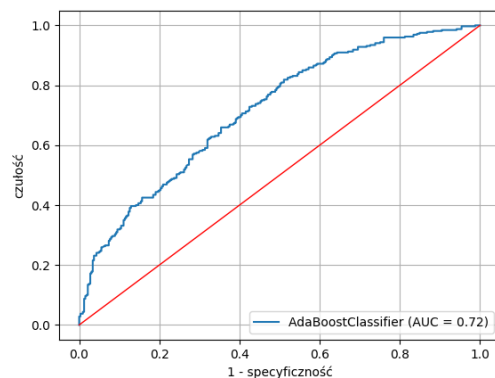
(c) Nikkei 225



(d) BSE Sensex 30



(e) UK 100



(f) WIG 20

Rys. 4.10. Krzywa ROC dla modelu boosting dla poszczególnych indeksów na danych testowych (horyzont dwusesyjny)

4.5. Regresja logistyczna

Następnym testowanym modelem jest regresja logistyczna. W przeciwieństwie do modeli drzew decyzyjnych nie dysponujemy dla niego tyloma hiperparametrami. Dzięki uwzględnieniu regularyzacji stroić będziemy parametr *penalty*, czyli wybór metody (re-

gresja grzbietowa, LASSO, sieć elastyczna lub brak regularyzacji), C czyli stopień regularyzacji oraz $l1_ratio$, który odpowiada wartości r we wzorze 2.25. W tabeli 4.12 zaprezentowano testowane przedziały hiperparametrów w pierwszej iteracji.

Tabela 4.12. Zbiór hiperparametrów testowanych w modelu regresji logistycznej (iteracja pierwsza)

Parametr	Zakres
<i>penalty</i>	"l1", "l2", "elasticnet", null
C	0,0001; 0,001; 0,01; 0,1; 1; 10; 100; 1000
<i>l1_ratio</i>	0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9

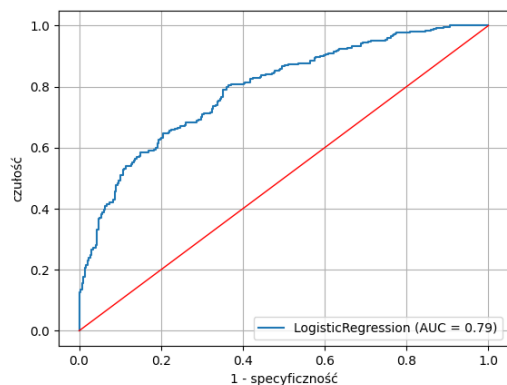
Wartość "null" dla parametru *penalty* oznacza brak regularyzacji. Korzystając z 5-krotnej walidacji krzyżowej najwyższą trafność - 0,707 (F1 = 0,747) uzyskano dla *penalty* = "elasticnet", C = 1 oraz $l1_ratio$ = 0,6. W kolejnej iteracji zmieniono zakres parametru C - [0,4; 0,7; 1; 1,5; 2; 3; 5; 7; 9] oraz $l1_ratio$ - [0,54; 0,57; 0,6; 0,63; 0,66]. Parametr *penalty* nie był testowany w dalszych iteracjach, ponieważ został dokładnie określony. W drugiej iteracji dla najlepszego modelu odnotowano takie same wartości trafności i F1 w przybliżeniu do 3 miejsca po przecinku. Ostatecznie jako optymalne przyjęto wartości wyznaczone w 1 iteracji, zaprezentowano je w tabeli 4.13

Tabela 4.13. Zbiór hiperparametrów testowanych w modelu regresji logistycznej (iteracja końcowa)

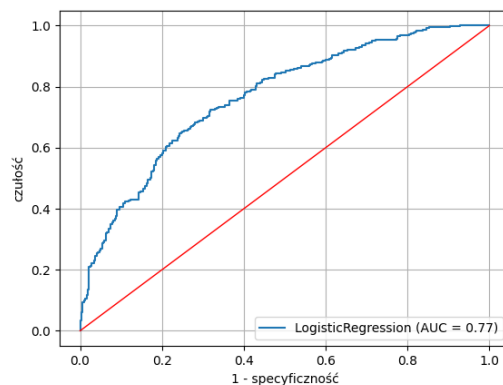
Parametr	Zakres
penalty	"elasticnet"
C	1
<i>l1_ratio</i>	0,6

W tabeli 4.14 zaprezentowano wskaźniki trafności i F1 dla wszystkich indeksów, wygenerowane na podstawie zbiorów testowych.

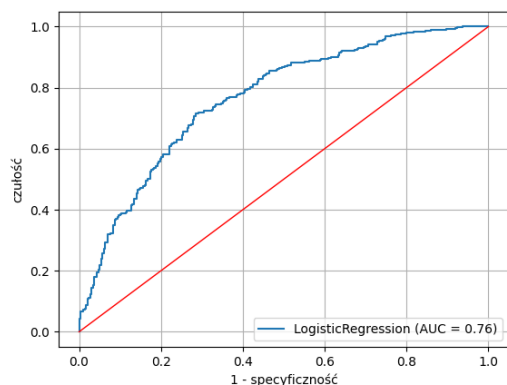
Na rysunku 4.11 zaprezentowano również wykresy krzywych ROC wraz z wartościami AUC dla poszczególnych indeksów.



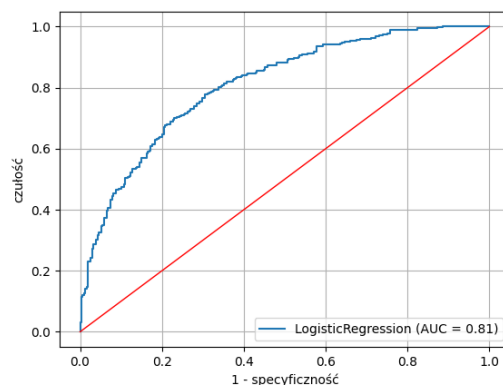
(a) S&P 500



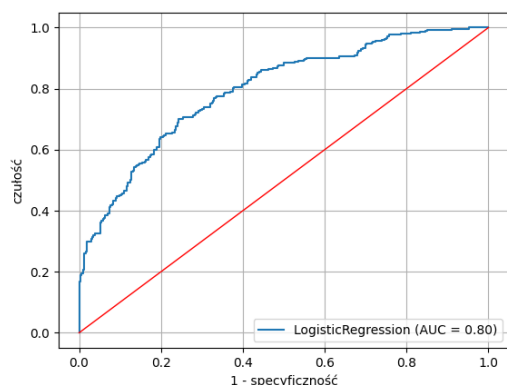
(b) DAX



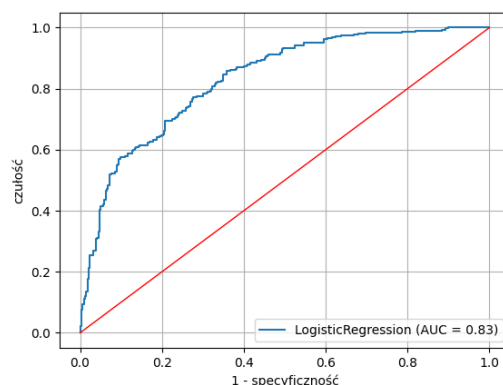
(c) Nikkei 225



(d) BSE Sensex 30



(e) UK 100



(f) WIG 20

Rys. 4.11. Krzywa ROC dla modelu regresji logistycznej dla poszczególnych indeksów na danych testowych (horyzont jednosesyjny)

Wyniki regresji logistycznej są na wysokim poziomie w porównaniu z wcześniejszymi modelami. Widać zdecydowanie większą trafność dla indeksu WIG 20. Pozostałe indeksy nie uzyskały tak wysokich trafności, ale nadal są na poziomie 0,70 - 0,72. Wyjątkiem jest DAX, dla którego uzyskano najniższą trafność - 0,676. Wszystkie wykresy są

Tabela 4.14. Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu regresji logistycznej na danych testowych (horyzont jednosesyjny)

Indeks	Trafność	F1
S&P 500	0,700	0,738
DAX	0,676	0,691
Nikkei 225	0,698	0,728
BSE Sensex 30	0,717	0,733
UK 100	0,716	0,732
WIG 20	0,746	0,735

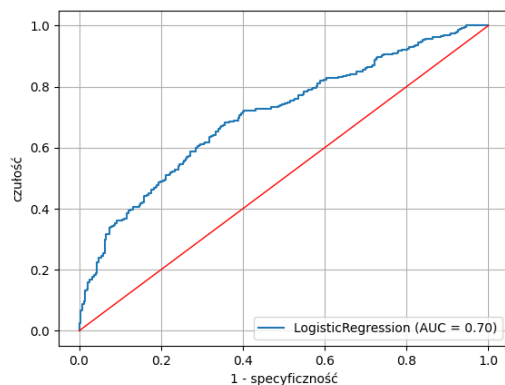
do siebie bardzo zbliżone, tak jak dla poprzednich testowanych modeli. Nieco gorsze wyniki AUC uzyskano dla indeksów Nikkei 225 oraz DAX, jednak nadal dla żadnego indeksu nie uzyskano AUC poniżej 0,75.

Korzystając z gotowego zestawu hiperparametrów wygenerowano dodatkowo model, w którym testowano zdolności predykcyjne na dwie sesje do przodu. W tabeli 4.15 przedstawiono uzyskane wyniki. Na rysunku 4.12 zaprezentowano również wykresy krzywych ROC wraz z wartościami AUC dla poszczególnych indeksów.

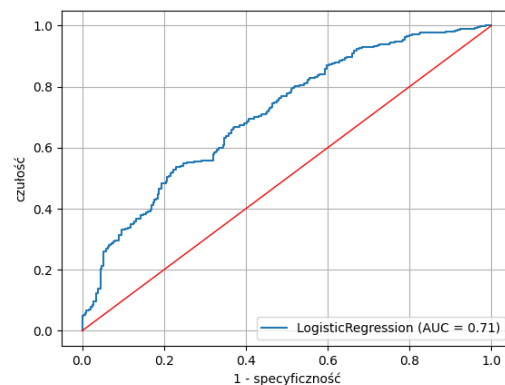
Tabela 4.15. Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu regresji logistycznej na danych testowych (horyzont dwusesyjny)

Indeks	Trafność	F1
S&P 500	0,655	0,730
DAX	0,632	0,700
Nikkei 225	0,647	0,660
BSE Sensex 30	0,682	0,711
UK 100	0,635	0,700
WIG 20	0,682	0,687

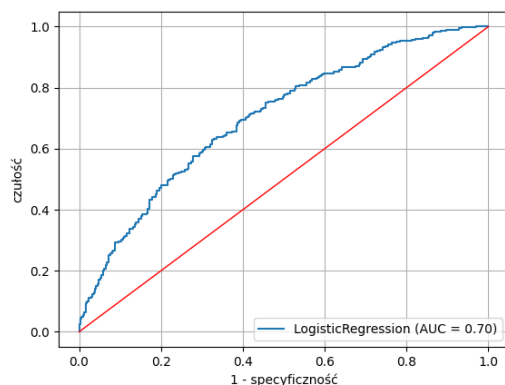
Trafności dla horyzontu dwusesyjnego oscylują w granicach 0,63 - 0,68. Widać więc wyraźną poprawę w porównaniu z wynikami dla poprzednich modeli. Najwyższe trafności odnotowano dla BSE Sensex 30 oraz WIG 20. Z kolei najniższą otrzymano znowu dla indeksu DAX. Wykresy krzywych ROC są wyraźnie spłaszczone w porównaniu z wykresami dla horyzontu jednosesyjnego. Wartości AUC oscylują w granicach 0,67 - 0,75.



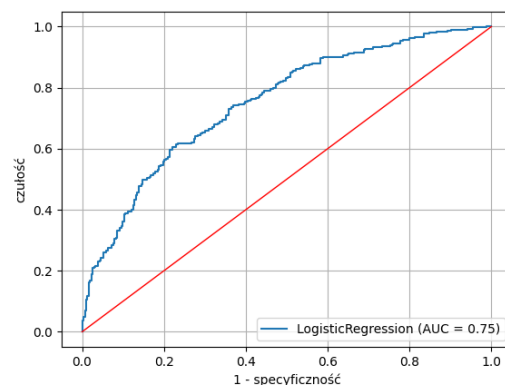
(a) S&P 500



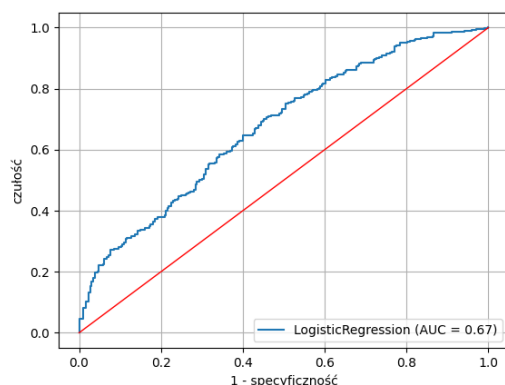
(b) DAX



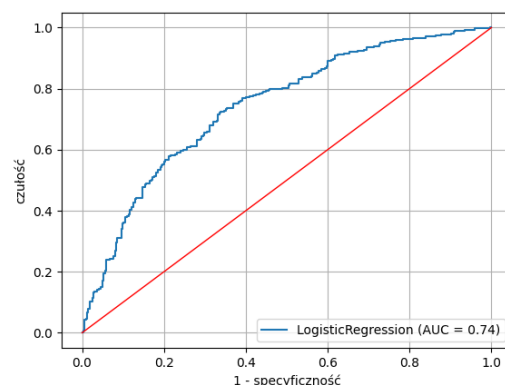
(c) Nikkei 225



(d) BSE Sensex 30



(e) UK 100



(f) WIG 20

Rys. 4.12. Krzywa ROC dla modelu regresji logistycznej dla poszczególnych indeksów na danych testowych (horyzont dwusesyjny)

4.6. Maszyna wektorów nośnych (SVM)

Ostatnim testowanym modelem jest maszyna wektorów nośnych. W przypadku modelu SVC kluczowymi parametrami są kształt jądra modelu (tabela 2.2) oraz stopień

regularyzacji, czyli parametr λ we wzorze 2.35. Dodatkowo testowano również stopień wielomianu dla modelu z jądrem wielomianowym oraz wartość parametru γ dla jądra gaussowskiego *rbf*. W tabeli 4.16 zaprezentowano testowane przedziały hiperparametrów w pierwszej iteracji.

Tabela 4.16. Zbiór hiperparametrów testowanych w modelu SVM (iteracja pierwsza)

Parametr	Zakres
<i>kernel</i>	"linear", "poly", "rbf", "sigmoid"
<i>degree</i>	2, 3
<i>gamma</i>	"scale", "auto"
<i>C</i>	0,0001; 0,001; 0,01; 0,1; 1; 10; 100; 1000

W przypadku jądra wielomianowego testowano tylko stopień 2 i 3. Jeśli chodzi o jądro gaussowskie (*rbf*), to w jego przypadku testowano dwie wartości parametru γ . Wartość "auto" oznacza przyjęcie $\gamma = 1 / (\text{liczba zmiennych})$, z kolei dla *scale*, $\gamma = 1 / (\text{liczba zmiennych} * \text{wariancja zmiennych objaśniających})$. W przypadku wariancji we wzorze mówimy o jej obliczeniu z całego zbioru danych, bez rozróżnienia na poszczególne cechy. Jeśli chodzi o parametr *C* to jest on równy odwrotności parametru λ ze wzoru 2.35. Czyli w praktyce będziemy testować ten parametr tak samo jak w regresji logistycznej.

Korzystając z 5-krotnej walidacji krzyżowej najwyższą trafność - 0,742 (F1 = 0,771) uzyskano dla jądra gaussowskiego (*rbf*) z γ określoną przez wartość "auto" i parametrem $C = 100$. W drugiej iteracji sprawdzano już tylko parametr *C* w zakresie [20, 40, 70, 100, 130, 160, 200, 300, 400, 500, 600, 700, 800, 900]. W efekcie trafność nieznacznie wzrosła pomimo, że w zaokrągleniu do 3 miejsca po przecinku wyniosła tyle samo (tak samo wartość F1) dla $C = 70$.

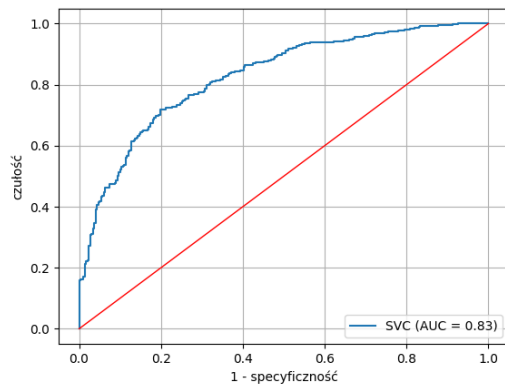
Końcowe wartości hiperparametrów zaprezentowano w tabeli 4.17.

Tabela 4.17. Zbiór hiperparametrów testowanych w modelu SVM (iteracja końcowa)

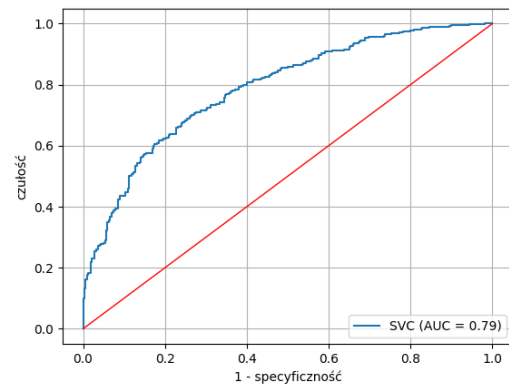
Parametr	Zakres
kernel	"rbf"
gamma	"auto"
C	70

W tabeli 4.18 zaprezentowano wskaźniki trafności i F1 dla wszystkich indeksów wygenerowane na podstawie zbiorów testowych.

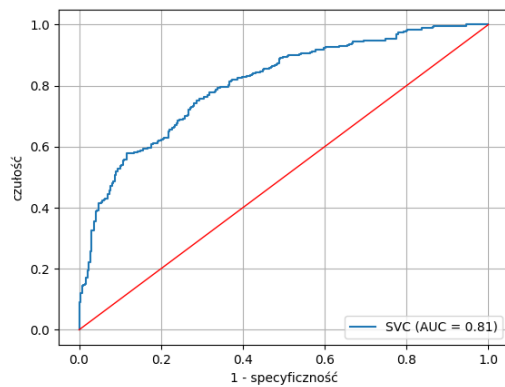
Na rysunku 4.13 zaprezentowano również wykresy krzywych ROC wraz z wartościami AUC dla poszczególnych indeksów.



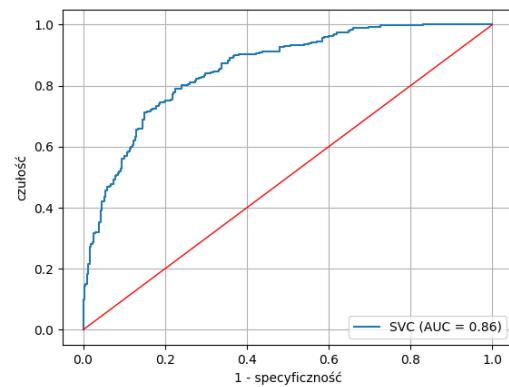
(a) S&P 500



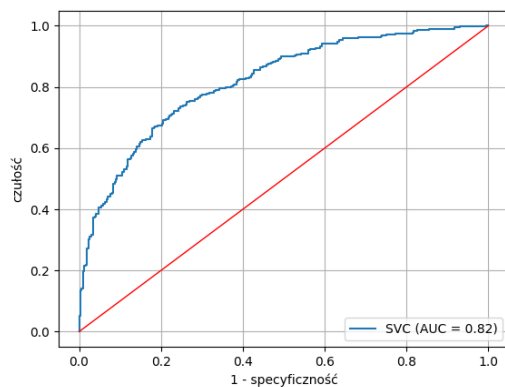
(b) DAX



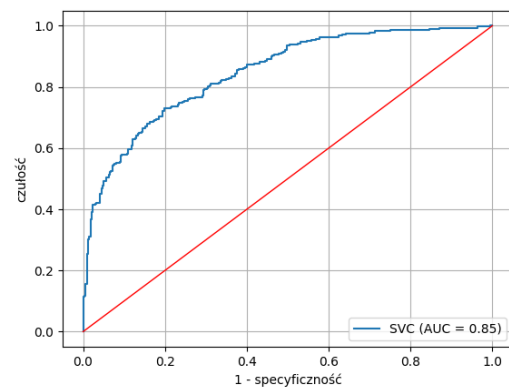
(c) Nikkei 225



(d) BSE Sensex 30



(e) UK 100



(f) WIG 20

Rys. 4.13. Krzywa ROC dla modelu SVM dla poszczególnych indeksów na danych testowych (horyzont jednosesyjny)

Tabela 4.18. Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu SVM na danych testowych (horyzont jednosesyjny)

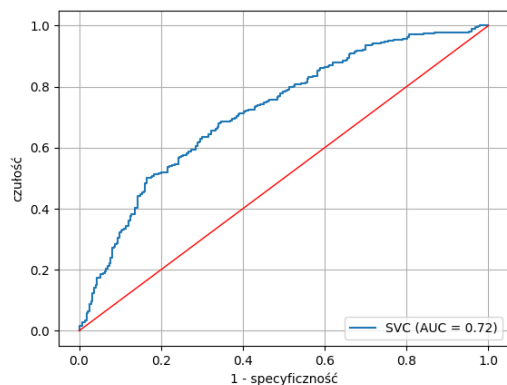
Indeks	Trafność	F1
S&P 500	0,718	0,742
DAX	0,661	0,659
Nikkei 225	0,700	0,719
BSE Sensex 30	0,752	0,757
UK 100	0,676	0,691
WIG 20	0,730	0,716

Zdecydowanie najwyższą trafność uzyskano dla indeksu BSE Sensex 30 - 0,752. Wysokie wartości wskaźników trafności oraz F1 uzyskano również dla S&P 500 oraz WIG 20. Najgorsze prognozy uzyskano po raz kolejny dla wskaźnika DAX oraz UK 100. Wszystkie wykresy ROC są do siebie bardzo zbliżone tak jak dla poprzednich testowanych modeli, ale tym razem tylko jeden indeks (DAX) uzyskał AUC poniżej 0,8. Na szczególną uwagę zasługują indeksy BSE Sensex 30 oraz WIG 20, które uzyskały AUC równe odpowiednio 0,85 i 0,86.

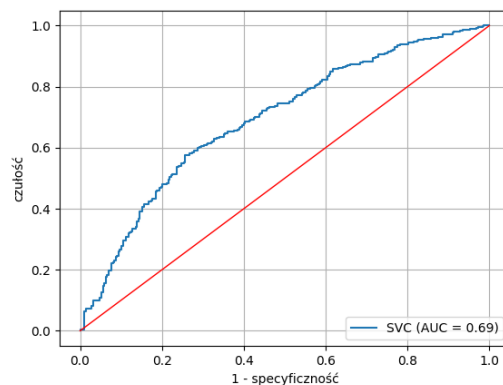
Korzystając z gotowego zestawu hiperparametrów wygenerowano dodatkowo model, w którym testowano zdolności predykcyjne na dwie sesje do przodu. W tabeli 4.19 przedstawiono uzyskane wyniki. Na rysunku 4.14 zaprezentowano również wykresy krzywych ROC wraz z wartościami AUC dla poszczególnych indeksów.

Tabela 4.19. Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu SVM na danych testowych (horyzont dwusesyjny)

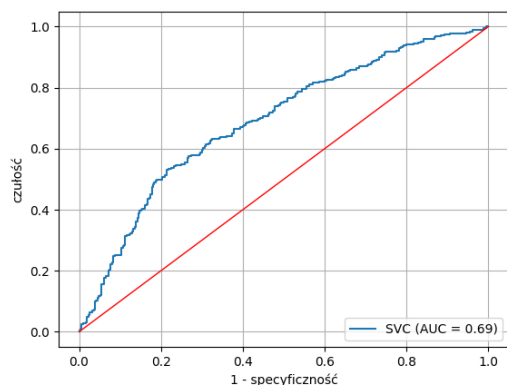
Indeks	Trafność	F1
S&P 500	0,637	0,708
DAX	0,629	0,683
Nikkei 225	0,599	0,614
BSE Sensex 30	0,646	0,653
UK 100	0,586	0,636
WIG 20	0,643	0,636



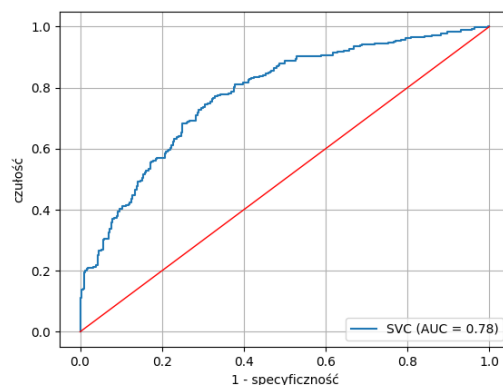
(a) S&P 500



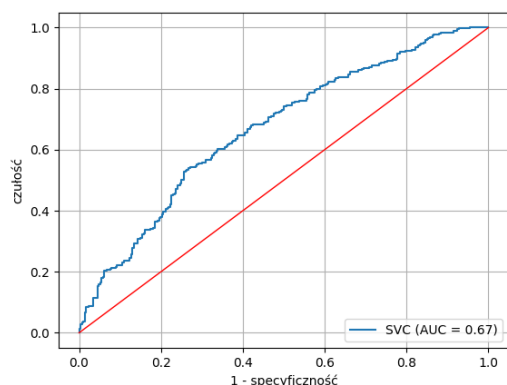
(b) DAX



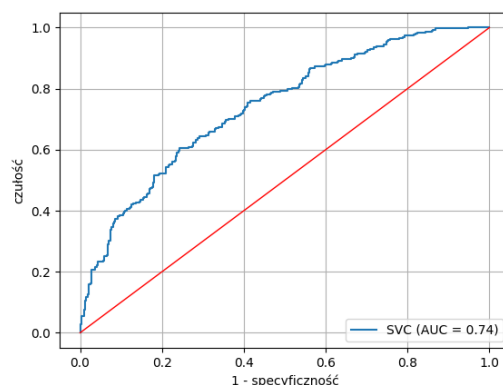
(c) Nikkei 225



(d) BSE Sensex 30



(e) UK 100



(f) WIG 20

Rys. 4.14. Krzywa ROC dla modelu SVM dla poszczególnych indeksów na danych testowych (horyzont dwusesyjny)


Uzyskano trafność w granicach 0,59 - 0,65. Najwyższe wartości trafności odnotowano dla BSE Sensex 30 oraz WIG 20. Z kolei najniższe otrzymano dla indeksu DAX. Dla S&P 500 wartość wskaźnika F1 była najwyższa i wyniosła 0,708. Najniższą wartość F1 odnotowano dla Nikkei 225 - 0,614. Wykresy krzywych ROC są wyraźnie spłaszczone

w porównaniu z wykresami dla horyzontu jednosesyjnego. Wartości AUC oscylują w granicach 0,67 - 0,78.

4.7. Ocena istotności zmiennych

Ważnym elementem oceny modelu jest analiza istotności poszczególnych zmiennych. Sprawdzamy w ten sposób jak obecność konkretnej zmiennej w zbiorze danych wpływa na wskaźniki zdolności predykcyjnych modeli. Jedną z metod stosowanych do tego jest permutacyjna ocena istotności zmiennych. Polega ona na eliminacji wpływu zmiennej dla konkretnego modelu i oszacowaniu jak przez to zmieniają się wskaźniki zdolności predykcyjnych. Najłatwiej zaprezentować jej zasadę działania na przykładzie. Załóżmy, że dysponujemy zbiorem danych złożonym z 3 zmiennych objaśniających - x_1 , x_2 , x_3 oraz zmiennej objaśnianej y . Mamy również wygenerowany model na zbiorze treningowym. Chcąc sprawdzić istotność poszczególnych zmiennych musimy najpierw określić zdolność predykcyjną modelu (np. jako trafność lub F1) na zbiorze testowym. Następnie wykonujemy permutacje na zbiorze testowym zmieniając porządek wartości tylko dla jednej zmiennej - np. x_1 . Sposób modyfikacji przedstawiono na rysunku 4.15. Używając wygenerowanego wcześniej modelu będziemy teraz prognozować bazując na zmiennej o innej relacji w stosunku do zmiennej objaśnianej. Ma to na celu symulować podanie losowej wartości x_1 zamiast rzeczywistej. W praktyce badamy więc różne warianty modelu przez zmianę zbioru danych, a nie parametrów. W następnym kroku obliczamy wartość wskaźnika predykcyjnego i sprawdzamy o ile się zmieniła w stosunku do wartości określonej na oryginalnym zbiorze testowym. Całość powtarzamy zadaną liczbę razy i obliczamy średnią wartość. W ten sposób szacujemy istotność zmiennej x_1 dla modelu. Analogiczne rozumowanie przeprowadzamy dla pozostałych zmiennych objaśniających.

Numer obserwacji	x_1	x_2	x_3	y
1	2	-3	8	0
2	4	1	7	1
3	6	5	6	1
4	8	9	5	0
5	10	13	4	1



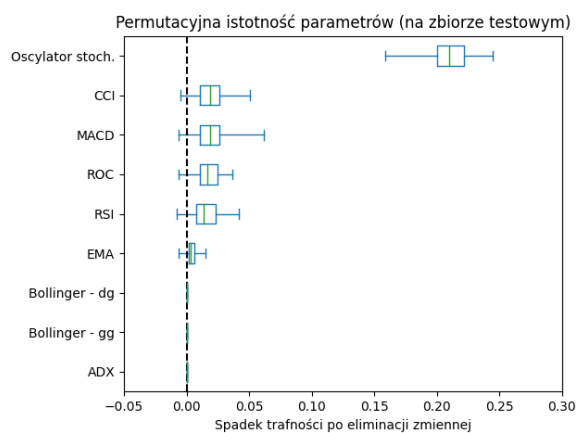
Numer obserwacji	x_1	x_2	x_3	y
1	2	-3	8	0
2	4	1	7	1
3	10	5	6	1
4	2	9	5	0
5	6	13	4	1

Rys. 4.15. Schemat metody permutacyjnej oceny istotności zmiennych w modelu. Po lewej stronie zaprezentowano oryginalny zbiór danych, a po prawej po wykonaniu permutacji. Na niebiesko zaznaczono kolumnę ze zmienną, dla której wykonujemy zamianę danych.

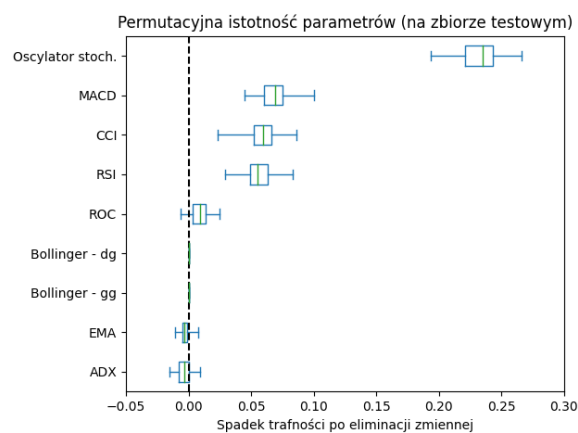
Można się zastanowić jaka korzyść płynie ze stosowania tej metody. Przede wszystkim nie wymaga ona generacji nowego modelu w każdej iteracji. Bazujemy jedynie na wykonaniu prostych permutacji, co znacząco skraca czas potrzebny na wykonanie obliczeń. Kolejną zaletą jest uniwersalność tej metody. Nigdzie nie precyzujemy jaki model rozpatrujemy. W ogólności można użyć permutacyjnej oceny istotności zmiennych dla dowolnego modelu.

Opisaną metodę przetestowano dla wszystkich testowanych w pracy modeli. Dla każdej ze zmiennych wykonano 100 permutacji mierząc w każdej iteracji spadek lub wzrost trafności. Wyniki przedstawiono na rysunku 4.16 w postaci wykresów pudełkowych dla danych zmiennych.

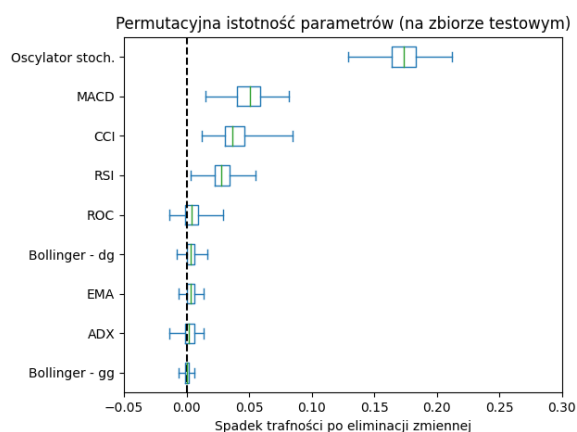
Dla wszystkich modeli z wyjątkiem SVM najistotniejsze okazały się wskazania oscylatora stochastycznego. Usunięcie tej zmiennej powoduje spadek trafności o 0,12 - 0,28. Jednocześnie dla tej zmiennej odnotowano też największy rozrzut danych, o czym świadczą długości pudełek i wąsów na wykresach. Mniej istotnymi zmiennymi okazały się CCI, MACD oraz RSI. Ostatnia z wymienionych co prawda była najistotniejsza dla SVM, ale dla pozostałych modeli permutacyjna ocena istotności wskazuje na spadek trafności o maksymalnie 0,15 w jej przypadku. Jeśli chodzi o CCI oraz MACD to okazały się one jednymi z najistotniejszych zmiennych w każdym z modeli notując spadek trafności nawet o 0,15, po wykonaniu permutacji na kolumnach z wartościami dla tych zmiennych. Warto odnotować, że binarne zmienne wskazujące na przekroczenie granic wstęgi Bollingera w niskim stopniu wpływały na zdolności predykcyjne badanych modeli. Wyjątkiem jest regresja logistyczna, dla której permutacje wykonane na tej zmiennej powodowały spadek trafności nawet o 0,05. Podobne wnioski nasuwają się po analizie spadków wartości dla zmiennych EMA oraz ADX. Ciekawy jest jednak przypadek modelu bagging. Po wykonaniu permutacji dla tych dwóch zmiennych wskaźnik trafności średnio ulegał wręcz nieznacznej poprawie.



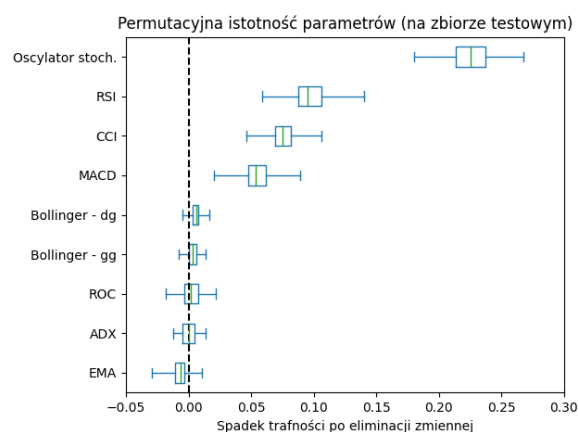
(a) Drzewa decyzyjne



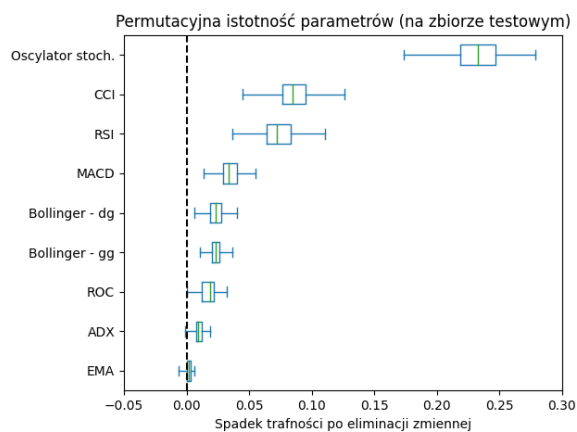
(b) Bagging



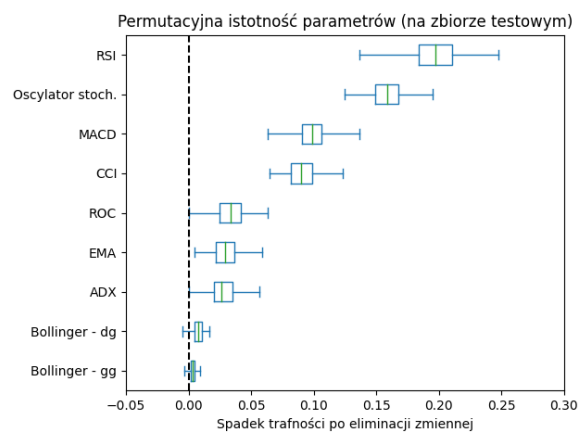
(c) Lasy losowe



(d) Boosting



(e) Regresja logistyczna



(f) SVM

Rys. 4.16. Wykres permutacyjnej oceny istotności zmiennych dla testowanych modeli na przykładzie indeksu S&P 500. Zieloną linię oznaczono medianę spadku trafności. Boczne krawędzie pudełka oznaczały wartości pierwszego i trzeciego kartyla. Końce wąsów oznaczają wartość minimalną i maksymalną. Analizę wykonano na zbiorze testowym.

4.8. Podsumowanie

W pracy testowano 6 modeli dla 6 zbiorów danych zawierających dane dzienne indeksów giełdowych. 4 modele pochodziły z grupy modeli drzew decyzyjnych - podstawowy model drzew decyzyjnych, bagging, lasy losowe oraz boosting. Dodatkowo przetestowano również model regresji logistycznej oraz maszyny wektorów nośnych. W tabelach 4.20 oraz 4.21 zaprezentowano końcowe wskaźniki trafności oraz F1 dla każdego z modeli z wyszczególnieniem testowanych indeksów.

Tabela 4.20. Zbiorcze wyniki trafności dla poszczególnych modeli z podziałem na indeksy (horyzont jednosesyjny).

	Drzewa decyzyjne	Bagging	Lasy losowe	Boosting	Regresja logistyczna	SVM	Średnia
S&P 500	0,666	0,720	0,706	0,708	0,700	0,718	0,703
DAX	0,640	0,670	0,672	0,661	0,676	0,661	0,663
Nikkei 225	0,680	0,718	0,695	0,695	0,698	0,700	0,698
BSE Sensex 30	0,689	0,722	0,711	0,712	0,717	0,752	0,717
UK 100	0,684	0,713	0,706	0,702	0,716	0,676	0,700
WIG 20	0,690	0,704	0,687	0,712	0,746	0,730	0,712
Średnia	0,675	0,708	0,696	0,698	0,709	0,706	

Źródło: opracowanie własne. W tabeli użyto formatowania warunkowego - czerwony kolor odpowiadał najniższym wartościom, a zielony najwyższym

Tabela 4.21. Zbiorcze wyniki F1 dla poszczególnych modeli z podziałem na indeksy (horyzont jednosesyjny).

	Drzewa decyzyjne	Bagging	Lasy losowe	Boosting	Regresja logistyczna	SVM	Średnia
S&P 500	0,693	0,747	0,709	0,728	0,738	0,742	0,726
DAX	0,660	0,651	0,666	0,672	0,691	0,659	0,667
Nikkei 225	0,688	0,726	0,693	0,713	0,728	0,719	0,711
BSE Sensex 30	0,707	0,721	0,703	0,715	0,733	0,757	0,723
UK 100	0,697	0,709	0,708	0,716	0,732	0,691	0,709
WIG 20	0,655	0,672	0,646	0,700	0,735	0,716	0,687
Średnia	0,683	0,704	0,688	0,707	0,726	0,714	

Źródło: opracowanie własne. W tabeli użyto formatowania warunkowego - czerwony kolor odpowiadał najniższym wartościom, a zielony najwyższym

Bazując na uzyskanych wynikach trafności można stwierdzić, że najlepszym modelem do predykcji zmian cen akcji na 1 sesję do przodu jest regresja logistyczna - średnia trafność na poziomie 0,709. Nieznacznie gorsze okazały się modele SVM oraz bagging - odpowiednio 0,708 i 0,706. Gorzej spisały się modele lasów losowych i boostingu, a zdecydowanie najslabszym był podstawowy model drzew decyzyjnych. Można

zauważyć również, że najlepsze prognozy uzyskano dla indeksu S&P 500, co jest spodziewnym rezultatem biorąc pod uwagę, że to na danych tego indeksu strojono hiperparametry modeli. Najniższe wskaźniki trafności uzyskano dla DAX i WIG 20. Ciekawe wnioski daje również analiza pojedynczych wartości dla poszczególnych modeli. Model SVM uzyskał najwyższą trafność dla indeksu BSE Sensex 30, ale jednocześnie jedną z najniższych wartości dla indeksów DAX oraz UK 100. Można więc stwierdzić, że SVM cechował się wysoką czułością (wariancją). Zupełnie odwrotnie jest dla modeli regresji logistycznej oraz baggingu, które były bardzo stabilne.

Analizując uzyskane wartości F1 można dojść do tych samych wniosków co dla wskaźników trafności. Również tutaj najlepszym modelem jest regresja logistyczna z F1 na poziomie 0,726. Znacznie lepiej wypada boosting, dla którego uzyskano wyższe F1 niż dla baggingu. Najniższe średnie F1 uzyskiwały modele drzew decyzyjnych oraz lasów losowych. Znowu najwyższe wartości F1 uzyskano dla S&P 500 - 0,726, ale nieznacznie gorsze otrzymano również dla BSE Sensex 30 (0,723). Regresja logistyczna znowu cechuje się wysoką stabilnością wyników.

Wszystkie modele sprawdzono również pod kątem prognozowania zmian cen akcji na dwie sesje do przodu. W tabelach 4.22 oraz 4.23 zaprezentowano końcowe wskaźniki trafności oraz F1 dla każdego z modeli z wyszczególnieniem testowanych indeksów.

Również przy dwusesyjnym horyzoncie predykcyjnym najlepszym modelem okazała się regresja logistyczna ze średnią trafnością na poziomie 0,656. Wysokie wyniki odnotowano również dla modelu lasów losowych i baggingu - odpowiednio 0,645 i 0,651. Najniższe trafności uzyskano dla drzew decyzyjnych. Jeśli chodzi o wskazania dla poszczególnych indeksów to najlepiej w tej kwestii modele prognozowały dla indeksu BSE Sensex 30. Najgorzej w zestawieniu wypada DAX. Zarówno regresja logistyczna jak i bagging są bardzo stabilnymi modelami. Analiza F1 dostarcza podobnych wniosków i potwierdza, że najlepiej spisywały się modele regresji logistycznej i baggingu.

Dla modeli przewidujących na 1 sesję do przodu otrzymano średnią trafność i F1 na poziomie odpowiednio 0,699 i 0,704. Dla horyzontu dwusesyjnego było to już 0,637 i 0,664. Trafność jest więc niższa o 6 punktów procentowych. Trzeba jednak przyznać, że daleko tym wartościom do wyników całkowicie losowych i badane modele mogą być pomocne w kontekście decyzji o kupnie lub sprzedaży. Ciekawa mogłaby być tutaj również analiza dla dłuższego okna predykcji, jednak to wykracza poza zakres tej pracy.

Tabela 4.22. Zbiorcze wyniki trafności dla poszczególnych modeli z podziałem na indeksy (horyzont dwusesyjny).

	Drzewa decyzyjne	Bagging	Lasy losowe	Boosting	Regresja logistyczna	SVM	Średnia
S&P 500	0,609	0,634	0,651	0,657	0,655	0,637	0,641
DAX	0,597	0,627	0,617	0,597	0,632	0,629	0,617
Nikkei 225	0,623	0,680	0,645	0,635	0,647	0,599	0,638
BSE Sensex 30	0,616	0,676	0,662	0,651	0,682	0,646	0,656
UK 100	0,629	0,638	0,640	0,624	0,635	0,586	0,625
WIG 20	0,639	0,650	0,654	0,612	0,682	0,643	0,647
Średnia	0,619	0,651	0,645	0,629	0,656	0,623	

Źródło: opracowanie własne. W tabeli użyto formatowania warunkowego - czerwony kolor odpowiadał najniższym wartościom, a zielony najwyższym

Tabela 4.23. Zbiorcze wyniki F1 dla poszczególnych modeli z podziałem na indeksy (horyzont dwusesyjny)

	Drzewa decyzyjne	Bagging	Lasy losowe	Boosting	Regresja logistyczna	SVM	Średnia
S&P 500	0,659	0,699	0,703	0,714	0,730	0,708	0,702
DAX	0,634	0,671	0,688	0,649	0,700	0,683	0,671
Nikkei 225	0,627	0,672	0,635	0,625	0,660	0,614	0,639
BSE Sensex 30	0,638	0,675	0,663	0,665	0,711	0,653	0,668
UK 100	0,670	0,684	0,693	0,680	0,700	0,636	0,677
WIG 20	0,574	0,641	0,632	0,611	0,687	0,636	0,630
Średnia	0,634	0,674	0,669	0,657	0,698	0,655	

Źródło: opracowanie własne. W tabeli użyto formatowania warunkowego - czerwony kolor odpowiadał najniższym wartościom, a zielony najwyższym

Jeśli chodzi o wpływ poszczególnych zmiennych na zdolności progностyczne to wykorzystano do tego metodę permutacyjnej oceny istotności. Analiza wskazuje na wysoką istotność zmiennych CCI, MACD, RSI oraz oscylatora stochastycznego dla wszystkich badanych modeli. Pozostałe zmienne były znacznie mniej istotne. Skrajnym przypadkiem jest model bagging, dla którego permutacje wykonane na zmiennych EMA oraz ADX powodowały wzrost trafności modelu.

Porównując uzyskane trafności z zaprezentowanymi rozwiązaniami innych autorów w rozdziale poświęconym przeglądowi literatury można dojść do wniosku, że wygenerowane modele osiągnęły przyzwoite wskazania. Co prawda Patel i in., 2015 oraz Ampomah i in., 2020 zaprezentowali modele osiągające trafności na poziomie 0,8 - 0,9, ale już Attigeri i in., 2015 dla modelu regresji logistycznej osiągnęli porównywalne z uzyskanymi trafności - 0,70.

Zakończenie

Założeniem pracy było zbudowanie i przetestowanie modeli uczenia maszynowego w kontekście predykcji zmian cen akcji na 1 lub 2 sesje do przodu. W pracy użyto modeli drzew decyzyjnych (bagging, lasy losowe, boosting), regresji logistycznej oraz maszyny wektorów nośnych. Modele testowane były na 6 indeksach giełdowych: S&P 500, DAX, Nikkei 225, BSE Sensex 30, UK 100, WIG 20. Wszystkie wymienione modele udało się poprawnie wygenerować dla zadanych indeksów, bazując na danych dotyczących historycznych cen akcji z lat 2010 - 2022.

Uzyskane pojedyncze i średnie trafności dla obydwu horyzontów predykcyjnych pozwalają sądzić, że stosowanie modeli uczenia maszynowego dla badanego problemu jest zasadne. Trzeba jednak zaznaczyć, że jesteśmy nadal daleko od dokładnej predykcji i badane rozwiązanie może być użyteczne w procesie decyzyjnym tylko jako pewnego rodzaju pomoc.

Przeprowadzone badania można traktować jako punkt wyjścia do dalszych badań. W tym kontekście należy zastanowić się nad rozszerzeniem horyzontu prognostycznego. Nie da się ukryć, że wykonywanie prognoz na wiele dni do przodu z należytą dokładnością byłoby bardzo użytecznym narzędziem giełdowym. Kolejnym elementem, na którym można się skupić w przyszłych badaniach jest wykorzystanie innych metod uczenia maszynowego. Już z literatury zaprezentowanej w rozdziale 1 można wywnioskować, że wykorzystanie chociażby sieci neuronowych może dać równie korzystne lub lepsze prognozy. Korzystnie na zdolności prognostyczne może wpłynąć również zmiana lub rozszerzenie rozpatrywanych zmiennych objaśniających. Ostatnim elementem wartym rozważenia jest uogólnienie badanego problemu do zagadnienia regresyjnego i prognozowanie nie zmian, a wprost cen akcji. Oczywiście zmienia się natura całego problemu, jednak zyskujemy dzięki temu dokładną prognozę, którą można wykorzystać przy szacunkach zysków i strat z transakcji. W tym kontekście ciekawym zagadnieniem byłaby również budowa narzędzia wybierającego najbardziej optymalną strategię biznesową.

Jak widać samo zagadnienie użycia uczenia maszynowego do prognozowania cen

akcji jest wdzięcznym tematem i daje bardzo dużo możliwości dalszego rozwoju. Aktualnie zwiększające się zainteresowanie uczeniem maszynowym pozwala sądzić, że wciąż powstające modele będą w przyszłości w większym stopniu oddawały rzeczywisty stan rzeczy.

Bibliografia

Literatura

- Alzubi, J., Nayyar, A., & Kumar, A. (2018). Machine learning from theory to algorithms: An overview. *Journal of Physics: Conference Series*, 1142(1), 012012. <https://doi.org/10.1088/1742-6596/1142/1/012012>
- Ampomah, E., Qin, Z., & Nyame, G. (2020). Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement. *Information*, 11(6). <https://doi.org/10.3390/info11060332>
- Attigeri, G. V., Manohara, P. M. M., Pai, R. M., & Nayak, A. (2015). Stock market prediction: A big data approach, 1–5. <https://doi.org/10.1109/TENCON.2015.7373006>
- Beyaz, E., Tekiner, F., Zeng, X.-j., & Keane, J. (2018). Comparing technical and fundamental indicators in stock price forecasting, 1607–1613. <https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00262>
- Bojańczyk, M. (2013). Od niestabilności do ryzyka – analiza zmienności historycznej indeksów giełdowych. *Kwartalnik Nauk o Przedsiębiorstwie*, 29, 4, 64–77. <https://doi.org/https://econjournals.sgh.waw.pl/KNoP/article/view/2150>
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24, 123–140. <https://doi.org/10.1007/BF00058655>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32. <https://doi.org/10.1023/A:1010933404324>
- Chou, J.-S., & Nguyen, T.-K. (2018). Forward forecast of stock price using sliding-window metaheuristic-optimized machine-learning regression. *IEEE Transactions on Industrial Informatics*, 14(7), 3132–3142. <https://doi.org/10.1109/TII.2018.2794389>

- Colby, R. W. (2002). *The encyclopedia of technical market indicators, second edition*. McGraw Hill LLC. <https://books.google.pl/books?id=f82LUFQVGOUUC>
- Friedman, J., Hastie, T., & Tibshirani, R. (2000). Special invited paper. additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2), 337–374.
- Geron, A. (2019). *Uczenie maszynowe z uzyciem scikit-learn i tensorflow wyd. ii*. Helion.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. Springer New York Inc. <https://doi.org/https://doi.org/10.1007/978-0-387-84858-7>
- Hosmer, D. W., & Lemeshow, S. (2000). *Introduction to the logistic regression model*. John Wiley & Sons, Ltd. <https://doi.org/https://doi.org/10.1002/0471722146.ch1>
- Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine [Applications of Neural Networks]. *Computers & Operations Research*, 32(10), 2513–2522. <https://doi.org/https://doi.org/10.1016/j.cor.2004.03.016>
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). *An introduction to statistical learning: With applications in r*. Springer Publishing Company, Incorporated.
- Mazur, M. (1970). *Jakościowa teoria informacji*. WNT.
- Obthong, M., Tantisantiwong, N., Jeamwattananachai, W., & Wills, G. (2020). A survey on machine learning for stock price prediction: Algorithms and techniques, 63–71. <https://eprints.soton.ac.uk/437785/>
- Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 42(1), 259–268. <https://doi.org/https://doi.org/10.1016/j.eswa.2014.07.040>
- Peng, Y., Albuquerque, P. H. M., Kimura, H., & Saavedra, C. A. P. B. (2021). Feature selection and deep neural networks for stock price direction forecasting using technical analysis indicators. *Machine Learning with Applications*, 5, 100060. <https://doi.org/https://doi.org/10.1016/j.mlwa.2021.100060>
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes 3rd edition: The art of scientific computing (3rd ed.)*. Cambridge University Press.

- Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017). Stock price prediction using lstm, rnn and cnn-sliding window model, 1643–1647. <https://doi.org/10.1109/ICACCI.2017.8126078>
- Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3), 379–423. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>
- Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques [International Conference on Computational Intelligence and Data Science]. *Procedia Computer Science*, 167, 599–606. <https://doi.org/https://doi.org/10.1016/j.procs.2020.03.326>

Strony internetowe

- Decision trees* [dostęp: 03.05.2023]. <https://scikit-learn.org/stable/modules/tree.html>
- Random forest classifier* [dostęp: 04.05.2023]. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier>
- Stooq* [dostęp: 17.06.2022]. <https://stooq.pl>
- Support vector classifier* [dostęp: 16.05.2023]. <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
- Wariancja* [dostęp 04.05.2023]. <https://en.wikipedia.org/wiki/Variance>
- What is python? executive summary* [dostęp: 19.01.2022]. <https://www.python.org/doc/essays/blurb/>

Spis rysunków

2.1	Przykład drzewa decyzyjnego	12
2.2	Wykres funkcji $f(x) = x \log(x)$ dla $x \in [0, 1]$ Źródło: opracowanie własne . . .	16
2.3	Dopasowanie regresji liniowej do zmiennej binarnej	23
2.4	Wykres $p(x)$ od wartości funkcji logit	24
2.5	Przykładowy 2-wymiarowy zbiór danych podzielony ze względu na wartość klasy	28
2.6	Przykładowy 2-wymiarowy zbiór danych podzielony ze względu na wartość klasy z naniesionymi marginesami klasyfikatora	29
2.7	Przykładowy 2-wymiarowy zbiór danych nieseparowalnych klasyfikatorem maksymalnego marginesu	31
2.8	Przykładowy 2-wymiarowy zbiór danych nieseparowalny liniową hiperpłaszczyzną	33
3.1	Tabela korelacji między zmiennymi wybranymi do zbudowania modelu .	42
4.1	Krzywa ROC dla modelu drzew decyzyjnych dla poszczególnych indeksów na danych testowych (horyzont jednosesyjny)	48
4.2	Krzywa ROC dla modelu drzew decyzyjnych dla poszczególnych indeksów na danych testowych (horyzont dwusesyjny)	50
4.3	Wykres wartości błędu OOB od hiperparametru $n_estimators$ dla modelu bagging	51
4.4	Krzywa ROC dla modelu bagging dla poszczególnych indeksów na danych testowych (horyzont jednosesyjny)	52
4.5	Krzywa ROC dla modelu bagging dla poszczególnych indeksów na danych testowych (horyzont dwusesyjny)	54
4.6	Wykres wartości błędu OOB od hiperparametru $n_estimators$ dla modelu lasów losowych	55
4.7	Krzywa ROC dla modelu lasów losowych dla poszczególnych indeksów na danych testowych (horyzont jednosesyjny)	56

4.8	Krzywa ROC dla modelu lasów losowych dla poszczególnych indeksów na danych testowych (horyzont dwusesyjny)	58
4.9	Krzywa ROC dla modelu boosting dla poszczególnych indeksów na danych testowych (horyzont jednosesyjny)	60
4.10	Krzywa ROC dla modelu boosting dla poszczególnych indeksów na danych testowych (horyzont dwusesyjny)	62
4.11	Krzywa ROC dla modelu regresji logistycznej dla poszczególnych indeksów na danych testowych (horyzont jednosesyjny)	64
4.12	Krzywa ROC dla modelu regresji logistycznej dla poszczególnych indeksów na danych testowych (horyzont dwusesyjny)	66
4.13	Krzywa ROC dla modelu SVM dla poszczególnych indeksów na danych testowych (horyzont jednosesyjny)	68
4.14	Krzywa ROC dla modelu SVM dla poszczególnych indeksów na danych testowych (horyzont dwusesyjny)	70
4.15	Schemat metody permutacyjnej oceny istotności zmiennych w modelu .	71
4.16	Wykres permutacyjnej oceny istotności zmiennych dla testowanych modeli na przykładzie indeksu S&P 500	73

Spis tabel

2.1	Wykaz hiperparametrów dostępnych w modelu drzew decyzyjnych . . .	17
2.2	Wykaz funkcji jądra dostępnych w modelu SVC	34
3.1	Macierz pomyłek dla modelu z dwiema możliwymi klasami zmiennej objaśnianej	44
4.1	Zbiór hiperparametrów testowanych w modelu drzew decyzyjnych (iteracja pierwsza)	47
4.2	Zbiór hiperparametrów testowanych w modelu drzew decyzyjnych (iteracja końcowa)	47
4.3	Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu drzew decyzyjnych na danych testowych (horyzont jednosesyjny)	49
4.4	Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu drzew decyzyjnych na danych testowych (horyzont dwusesyjny)	49
4.5	Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu bagging na danych testowych (horyzont jednosesyjny)	53
4.6	Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu bagging na danych testowych (horyzont dwusesyjny)	53
4.7	Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu lasów losowych na danych testowych (horyzont jednosesyjny)	57
4.8	Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu lasów losowych na danych testowych (horyzont dwusesyjny)	57
4.9	Zbiór hiperparametrów testowanych w modelu AdaBoost (iteracja końcowa)	59
4.10	Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu boosting na danych testowych (horyzont jednosesyjny)	61
4.11	Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu boosting na danych testowych (horyzont dwusesyjny)	61
4.12	Zbiór hiperparametrów testowanych w modelu regresji logistycznej (iteracja pierwsza)	63

4.13 Zbiór hiperparametrów testowanych w modelu regresji logistycznej (iteracja końcowa)	63
4.14 Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu regresji logistycznej na danych testowych (horyzont jednosesyjny) . . .	65
4.15 Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu regresji logistycznej na danych testowych (horyzont dwusesyjny)	65
4.16 Zbiór hiperparametrów testowanych w modelu SVM (iteracja pierwsza)	67
4.17 Zbiór hiperparametrów testowanych w modelu SVM (iteracja końcowa) .	67
4.18 Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu SVM na danych testowych (horyzont jednosesyjny)	69
4.19 Wartości trafności i F1 dla wybranych indeksów giełdowych dla modelu SVM na danych testowych (horyzont dwusesyjny)	69
4.20 Zbiorcze wyniki trafności dla poszczególnych modeli z podziałem na indeksy (horyzont jednosesyjny)	74
4.21 Zbiorcze wyniki F1 dla poszczególnych modeli z podziałem na indeksy (horyzont jednosesyjny)	74
4.22 Zbiorcze wyniki trafności dla poszczególnych modeli z podziałem na indeksy (horyzont dwusesyjny)	76
4.23 Zbiorcze wyniki F1 dla poszczególnych modeli z podziałem na indeksy (horyzont dwusesyjny)	76