Politechnika Wrocławska

Wydział Matematyki
Kierunek studiów: Matematyka stosowana
Specjalność: –

Praca dyplomowa – inżynierska

WYKRYWANIE OSZUSTW NA KARTACH PŁATNICZYCH Z WYKORZYSTANIEM METOD WRAŻLIWYCH NA KOSZT

Patryk Wielopolski

Słowa kluczowe:
Statystyka; Uczenie maszynowe; Klasyfikacja wrażliwa na koszt; Wykrywanie oszustw na kartach kredytowych

Krótkie streszczenie:
Celem pracy jest implementacja eksperymentu, który pozwoli porównać klasyczne oraz wrażliwe na koszt metody predykcyjne w problemie detekcji oszustw na kartach płatniczych. Oszustwa na kartach płatniczych wynikają z przekazania numeru karty nieznajomym, utraty lub kradzieży karty, kradzieży przesyłki lub skopiowania danych karty. Zwykle kończą się pojawieniem się na koncie nieautoryzowanych płatności. W pracy wykorzystane zostaną metody statystyczne oraz uczenia maszynowego oraz ich modyfikacje wrażliwe na koszt.

Opiekun pracy dyplomowej	dr inż. Andrzej Giniewicz	.....	.....
	Tytuł/stopień naukowy/imię i nazwisko	ocena	podpis

*Do celów archiwalnych pracę dyplomową zakwalifikowano do:**

a) kategorii A (akta wieczyste)

b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)

** niepotrzebne skreślić*

pieczątka wydziałowa

Wrocław, rok 2019

Wrocław University
of Science and Technology

Faculty of Pure and Applied Mathematics
Field of study: Applied Mathematics
Specialty: –

Engineering Thesis

CREDIT CARD FRAUD DETECTION USING COST SENSITIVE METHODS

Patryk Wielopolski

Keywords:
Statistics; Machine Learning; Cost Sensitive Classification; Credit Card Fraud Detection

Short summary:
The aim of this thesis is an implementation of an experiment, which will allow us to compare classical and cost-sensitive methods in credit card fraud detection. Credit card fraud occurs when card number is passed to someone else, when card is lost or stolen, when mail is stolen or card is copied by someone. Usually credit card fraud ends in unauthorised transactions on banking account. In this thesis statistical and machine learning methods will be utilized, in classical and their cost-sensitive modifications.

Supervisor	dr inż. Andrzej Giniewicz	.....	.....
	Title/degree/name and surname	grade	signature

*For the purposes of archival thesis qualified to:**

a) category A (perpetual files)

b) category BE 50 (subject to expertise after 50 years)

** delete as appropriate*

stamp of the faculty

Wrocław, 2019

Spis treści

Wstęp **1**

1 Wprowadzenie teoretyczne **5**
1.1 Miary skuteczności modeli 6
1.1.1 Macierz pomyłek 6
1.1.2 Miary skuteczności niewrażliwe na koszt 6
1.1.3 Krzywa charakterystyki pracy odbiornika 8
1.1.4 Macierz kosztu 8
1.1.5 Miary skuteczności modeli wrażliwe na koszt 9
1.2 Standardowe modele 11
1.2.1 Regresja logistyczna 11
1.2.2 Drzewo decyzyjne 12
1.2.3 Las losowy 14
1.2.4 XGBoost 15
1.3 Cost Dependent Classification 16
1.3.1 Optymalizacja progu 16
1.3.2 Minimalizacja ryzyka bayesowskiego 17
1.4 Cost Sensitive Training 18
1.4.1 Regresja logistyczna wrażliwa na koszt 18
1.4.2 Drzewo decyzyjne wrażliwe na koszt 19

2 Eksperyment **21**
2.1 Dane 21
2.2 Opis eksperymentu 23
2.3 Wyniki 23

Bibliografia **27**

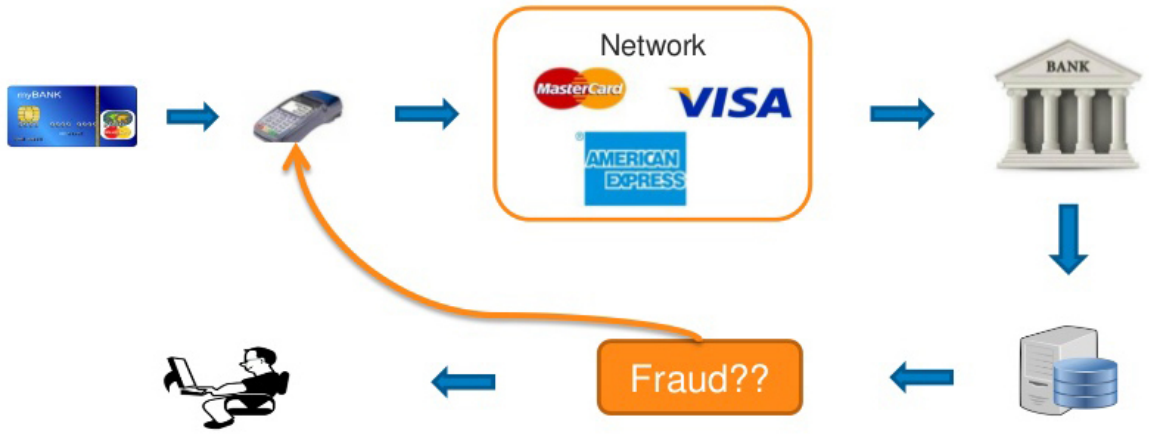
Wstęp

Historia kart płatniczych zaczyna się już w latach 80. XIX wieku, kiedy Edward Bellamy w swojej powieści *Looking Backward* wspomina o możliwości skorzystania z karty przedpłaconej (karta umożliwiająca dokonanie płatności z wcześniej wpłaconych środków).[1] Niestety pomysł takiej formy płatności nie został pozytywnie przyjęty w tamtych czasach i dopiero w 1914 roku amerykańska firma Western Union umożliwiła korzystanie z kart obciążeniowych (bank udziela klientowi odpowiedniego limitu płatności, za które należy zapłacić później). Koncepcja karty bardzo szybko rozpowszechniła się i mimo wybuchu II Wojny Światowej, która wstrzymała jej rozwój, to już w 1950 roku firma Diners Club umożliwiła swoim klientom dokonywania płatności w sklepach, restauracjach lub stacjach benzynowych bez użycia gotówki.[2] Kilka lat później wówczas największy bank w Stanach Zjednoczonych – Bank of America, wydał pierwszą prawdziwą kartę kredytową, która umożliwiała spłatę zadłużenia w całości lub tylko pewnej wyliczonej kwoty minimalnej. Kolejnym przełomem było wprowadzenie w roku 1972 paska magnetycznego, który umożliwiał korzystanie z PINu i dzięki temu zwiększał bezpieczeństwo klientów.[3] W roku 1973 i 1974 wprowadzono odpowiednio system BASE I oraz BASE II, które były pierwszymi systemami elektronicznymi. Następnie w latach 90. pojawiają się technologie takie jak karty chipowe EMV, Visa 3D Secure oraz Master Card SPA, które mają zapewnić zwiększenie bezpieczeństwa oraz zmniejszenie skali oszustw.

Dalszy rozwój technologiczny umożliwił nam nie tylko dokonywanie płatności kartą, lecz także wykorzystywanie do tego celu telefonu komórkowego lub zegarka. Ponadto transakcji nie dokonujemy już tylko w sklepach stacjonarnych, lecz także coraz częściej korzystamy ze sklepów internetowych. Wszystkie te czynniki powodują, że dokonywanie zakupów jest jeszcze prostsze, a ilość dokonywanych transakcji z roku na rok rośnie. [4] Niestety ten wzrost spowodował również rozwój sposobów dokonywania przestępstw związanych z kartami płatniczymi, przykładowo przekazanie numeru karty nieznajomym, utrata lub kradzież karty, skopiowania danych karty lub kradzież przesyłki z kartą, które najczęściej kończą się nieautoryzowanymi płatności na koncie.

Banki oraz instytucje finansowe w celu przeciwdziałania tym przestępstwom korzystają z systemów detekcji oszustw. Jest to jeden z elementów całego procesu dokonywania płatności, który jest przedstawiony na rysunku 1 na stronie 2. W momencie przyłożenia karty kredytowej do terminala nawiązywane jest połączenie z operatorem karty (np. Visa, MasterCard), który dalej komunikuje się z bankiem klienta. Następnie w odpowiednich bazach danych sprawdzane są dostępne środki, a system detekcji oszustw sprawdza, czy transakcja nie jest podejrzana. W przypadku wykrycia niezgodności system ją odrzuca i kieruje sprawę do odpowiedniego analityka. W przeciwnym razie płatność jest akceptowana

[1] Źródło: `https://pl.wikipedia.org/wiki/Karta_p%C5%82atnicza` (dostęp: 27.12.2019)
[2] Źródło: `https://www.kartyonline.pl/historia-kart-platniczych.php` (dostęp: 27.12.2019)
[3] Źródło: `https://en.wikipedia.org/wiki/Payment_card` (dostęp: 27.12.2019)
[4] Źródło: `https://worldpaymentsreport.com/non-cash-payments-volume/` (dostęp: 27.12.2019)



Rysunek 1: Schemat płatności kartą kredytową. Rysunek pochodzi z prezentacji A.Bahnsena podczas Konferencji Analytics 2013. Źródło: `https://www.slideshare.net/albahnsen/correa-bahnsen-alejandroanalytics2013slideshare`

i na terminalu wyświetla się odpowiedni komunikat.

Do niedawna najbardziej popularnym sposobem do wykrywania oszustw był system oparte na regułach eksperckich.[5] Powstały one na bazie doświadczenia analityków, którzy podczas swojej długoletniej pracy wielokrotnie sprawdzali transakcje pod kątem oszustw i byli w stanie dostrzec pewne zależności. Przykładami takich reguł mogą być: więcej niż cztery wypłaty z bankomatu w ciągu godziny, więcej niż dwie transakcje w ciągu 5 minut, dwie transakcje w ciągu 5 minut różnymi formami płatności (karta, internet). W przypadku, gdy rozpatrywana transakcja spełnia chociaż jedną z reguł, to jest ona blokowana i użytkownik jest informowany o odrzuceniu transakcji. To podejście jest stosunkowo łatwe do implementacji oraz jest bardzo proste do interpretacji, lecz niestety ma swoje wady. Oszuści wraz z upływem czasu zmieniają swoje sposoby dokonywania oszustw, natomiast systemy złożone z niemodyfikowalnych reguł nie są w stanie wykrywać nowych zachowań i dostosowywać się do zmian. Tę niedogodność są w stanie rozwiązać modele uczenia maszynowego, które tworzą pewnego rodzaju reguły, które nie są bezpośrednio wprowadzane przez użytkownika, lecz automatycznie wykrywane przez algorytmy na podstawie dostarczonych danych historycznych. Wraz z dynamicznym rozwojem tej dziedziny nauki zaczęły powstawać coraz bardziej wyrafinowane modele, które coraz lepiej radzą sobie z wykrywaniem takich zależności. Obecnie są one już wykorzystywane w produkcyjnych środowiskach instytucji finansowych i wspomagają walkę z przestępcami. Niestety standardowe podejście do modelowania predykcyjnego zakłada równy koszt popełnienia błędu, co nie jest prawdą w tym, jak i w wielu innych praktycznych zastosowaniach. Stąd powstała motywacja do rozwoju metod, które biorą pod uwagę tę różnicę.

Wybór tematu pracy był związany z codzienną pracą nad modelami predykcyjnymi wykorzystywanymi do detekcji oszustw w szkodach komunikacyjnych. Zauważony przeze mnie w trakcie prac problem nierównego kosztu pomyłek doprowadził mnie do poszukiwania metod, które będą adekwatne do tego rodzaju wyzwań. W ten sposób trafiłem na prace

[5] Wystąpienie A.Bahnsena podczas Konferencji Analytics 2013. Źródło: `https://www.youtube.com/watch?v=YCNkxaVDiA0`

dr Alejandro Correa Bahnsena, które poruszały to zagadnienie dla modelowania ryzyka kredytowego, prognozy rezygnacji z usług przez klientów, marketingu bezpośredniego oraz detekcji oszustw w transakcjach kartami kredytowymi. Z uwagi na bliską analogię ostatniego przykładu z moim problemem zdecydowałem się na bliższe zapoznanie z tym tematem.

Celem pracy jest opisanie i porównanie standardowych oraz wrażliwych na koszty modeli predykcyjnych w problemie detekcji oszustw na kartach płatniczych. W tym celu zostanie przeprowadzony eksperyment porównujący skuteczność poszczególnych metod. Wykorzystane zostaną modele takie jak regresja logistyczna, drzewo decyzyjne, las losowy oraz algorytm XGBoost wraz z odpowiednimi modyfikacjami oraz metody optymalizacji progu i minimalizacji ryzyka bayesowskiego.

Praca jest zorganizowana w następujący sposób. Rozdział pierwszy jest wstępem teoretycznym, który wprowadza odpowiednie pojęcia i modele z dziedziny statystyki oraz uczenia maszynowego. Rozdział drugi prezentuje opis eksperymentu na danych rzeczywistych wraz z wynikami oraz wnioskami. Ostatnia część podsumowuje całość pracy.

Rozdział 1

Wprowadzenie teoretyczne

Wprowadzenie teoretyczne zawiera przegląd metod z dziedziny statystyki oraz uczenia maszynowego, które wykorzystuje się w zagadnieniach klasyfikacyjnych. W szczególności zaczniemy od opisania macierzy pomyłek, macierzy kosztu oraz związane z nimi miary skuteczności modeli predykcyjnych, które służą do badania jakości predykcji. Następnie przejdziemy do modeli standardowych, które nie biorą pod uwagi różnych kosztów popełniania błędu klasyfikacyjnego. W ramach tej grupy opiszemy regresję logistyczną, drzewo decyzyjne, las losowy oraz algorytm XGBoost. Na końcu poruszymy temat modeli wrażliwych na koszt, które dzielimy na dwie podkategorie: Trening wrażliwy na koszt (*ang. Cost Sensitive Training*) oraz Klasyfikacja wrażliwa na koszt (*ang. Cost Sensitive Classification*).[1] Pierwsza z nich zawiera metody, które w trakcie uczenia modelu zwracają uwagę na koszt błędnej klasyfikacji. Są to rozszerzone wersje standardowych modeli, które w trakcie uczenia się mają zmienioną postać optymalizowanej funkcji. W tej pracy będziemy omawiać regresję logistyczną wrażliwą na koszt oraz drzewo decyzyjne wrażliwe na koszt. Druga podkategoria składa się z metod, które wykorzystują estymowane prawdopodobieństwo ze standardowych modeli i na jego podstawie tworzą dodatkowy model, który bierze pod uwagę koszt niepoprawnej klasyfikacji danego przypadku. Z tej grupy omówimy optymalizację progu oraz minimalizację ryzyka bayesowskiego.

W pracy będziemy wykorzystywać następujące oznaczenia:

- $\mathcal{S} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ – zbiór wszystkich N obserwacji,
- $\boldsymbol{x}_i = \left(x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(j)}\right)$ – i-ty wektor obserwacji składający się z j atrybutów,
- $x_i^{(k)}$ – k-ty atrybut i-tego wektora obserwacji,
- $\boldsymbol{y}_{\text{true}} = (y_1, y_2, \ldots, y_N)$ - wektor prawdziwych stanów klasyfikacji,
- $y_i \in \{0, 1\}$ – prawdziwy stan klasyfikacji dla i-tej obserwacji. Wartość 0 oznacza klasyfikację negatywną, natomiast wartość 1 oznacza klasyfikację pozytywną.
- $p_i \in [0, 1]$ – estymowana wartość prawdopodobieństwa dla i-tej obserwacji,
- $c_i \in \{0, 1\}$ – przewidywana klasa dla i-tej obserwacji. Oznaczenia analogicznie jak dla y_i.

[1] Źródło: `https://towardsdatascience.com/fraud-detection-with-cost-sensitive-machine-learning-24b8760d35d9`

1.1 Miary skuteczności modeli

1.1.1 Macierz pomyłek

Jedną z metod wykorzystywanych do oceny skuteczności modeli w zadaniach klasyfikacyjnych jest macierz pomyłek. Prawidłowe klasyfikacje zestawiamy z klasami nadanymi przez model, a następnie sprawdzamy, czy poprawnie zaklasyfikowaliśmy poszczególne obserwacje. W przypadku klasyfikacji binarnej popełniamy dwa rodzaje błędów, a wszystkie możliwe wyniki prezentują się w następujący sposób:

-Prawdziwie pozytywny wynik klasyfikacji (TP, z angielskiego *True Positive*) – obserwacja była pozytywna i zaklasyfikowaliśmy ją jako pozytywną;

-Fałszywie pozytywny wynik klasyfikacji (FP, z angielskiego *False Positive*) – obserwacja była negatywna i zaklasyfikowaliśmy ją jako pozytywną. W nomenklaturze statystycznej nazywamy ją również błędem pierwszego rodzaju;

-Fałszywie negatywny wynik klasyfikacji (FN, z angielskiego *False Negative*) – obserwacja była pozytywna i zaklasyfikowaliśmy ją jako negatywną. W nomenklaturze statystycznej nazywamy ją również błędem drugiego rodzaju;

-Prawdziwie negatywny wynik klasyfikacji (TN, z angielskiego *True Negative*) – obserwacja była negatywna i zaklasyfikowaliśmy ją jako negatywną.

Liczbę obserwacji należących do poszczególnych kategorii możemy reprezentować tak jak w tabeli 1.1.

	Stan pozytywny $y_i = 1$	Stan negatywny $y_i = 0$
Predykcja pozytywna $c_i = 1$	TP	FP
Predykcja negatywna $c_i = 0$	FN	TN

Tabela 1.1: Macierz pomyłek

1.1.2 Miary skuteczności niewrażliwe na koszt

Na podstawie macierzy pomyłek (tabela 1.1), którą omawialiśmy w poprzedniej podsekcji, możemy zdefiniować następujące miary skuteczności modeli, które nie biorą pod uwagę kosztu popełnienia błędu.

Dokładność

Dokładność (*ang. Accuracy*) mierzy stosunek poprawnie zaklasyfikowanych przypadków do ilości wszystkich obserwacji. Bierze ona pod uwagę zarówno obserwacje pozytywne, jak i negatywne. Z tego powodu bardzo dobrze radzi sobie w sytuacji, gdy interesują nas obserwacje z obu klas, natomiast dużo gorzej, gdy istotna jest tylko jedna klasa. W szczególności nie nadaje się ona do problemów z dużą dysproporcją klas, która pojawia się w przypadku detekcji oszustw. Dokładność zdefiniowana jest następującym wzorem

$$\text{Dokładność} = \frac{TP + TN}{TP + FP + FN + TN}.$$

Precyzja

Precyzja (*ang. Precision*) to stosunek poprawnie zaklasyfikowanych pozytywnych przypadków do wszystkich obserwacji zaklasyfikowanych jako pozytywne. Mierzy ona, jaki procent próbek prawidłowo zaklasyfikowaliśmy jako pozytywne. Warto zauważyć, że nie bierze ona pod uwagę obserwacji negatywnych. Jest ona bardzo często wykorzystywana podczas rozwiązywania większości problemów z zakresu uczenia maszynowego, ponieważ najczęściej interesuje nas, aby nasze typowania były możliwie najbardziej skuteczne. Z tego też powodu precyzja jest często nazywana skutecznością klasyfikacji prawdziwie pozytywnych (*ang. True Positive Accuracy*). Zapisana wzorem przyjmuje postać

$$\text{Precyzja} = \frac{TP}{TP + FP}.$$

Czułość

Czułość to stosunek prawdziwie pozytywnie zaklasyfikowanych przypadków do wszystkich pozytywnych obserwacji. Innymi słowy, jest to miara, która mówi nam o procencie znalezionych przez model przypadków pozytywnych. Podobnie jak precyzja skupia się jedynie na poprawnie zaklasyfikowanych obserwacjach pozytywnych. Jest ona szczególnie ważna w zastosowaniach medycznych, gdzie naszym głównym celem jest znalezienie wszystkich chorych osób. Używana jest także do wyznaczania krzywej charakterystyki pracy odbiornika (ROC, z angielskiego *Receiver operating characteristic*) opisanej w sekcji 1.1.3 na stronie 8. Czułość jest wyrażona wzorem

$$\text{Czułość} = \frac{TP}{TP + FN}.$$

Swoistość

Swoistość to stosunek prawdziwie negatywnie zaklasyfikowanych przypadków do wszystkich negatywnych obserwacji. Inaczej nazywana też odwrotną czułością, z uwagi na analogiczną postać do czułości, ponieważ mierzy ona również odsetek znalezionych wszystkich obserwacji, lecz dla negatywnej klasy. Wyraża się następującym wzorem

$$\text{Swoistość} = \frac{TN}{TN + FP}.$$

Ujemna wartość predykcyjna

Ujemna wartość predykcyjna to stosunek prawdziwie negatywnie zaklasyfikowanych przypadków do wszystkich negatywnie zaklasyfikowanych obserwacji. Nazywana jest również odwrotną precyzją, ponieważ analogicznie do precyzji skupia się na mierzeniu skuteczności klasyfikowania obserwacji negatywnych. Miara ta jest także używana do wyrysowania krzywej ROC (patrz sekcja 1.1.3 na stronie 8). Dana jest wzorem

$$\text{Ujemna wartość predykcyjna} = \frac{TN}{TN + FN}.$$

Zbalansowana skuteczność

Zbalansowana skuteczność to średnia arytmetyczna czułości oraz swoistości. W przypadku, gdy klasyfikator działa równie dobrze dla obu klas, to sprowadza się ona do

dokładności, natomiast gdy standardowa dokładność jest wysoka tylko z uwagi na niezbalansowaną liczość próbek w klasach, to wartość zbalansowanej skuteczności spada. Jest ona również ważna, ponieważ maksymalizacja tej miary odpowiada standardowemu szukaniu optymalnego punktu na krzywej ROC (patrz sekcja 1.1.3 na stronie 8) wg. metryki taksówkowej. Zadana jest wzorem

$$\text{Zbalansowana skuteczność} = \frac{\text{Czułość} + \text{Swoistość}}{2}.$$

F Score

F_1 Score to miara łącząca w sobie precyzję oraz czułość za pomocą średniej harmonicznej. Z uwagi na wykorzystanie tych dwóch wartości nie bierze ona pod uwagę prawidłowo zaklasyfikowanych obserwacji negatywnych. Wykorzystanie średniej harmonicznej powoduje, że kładziemy równy nacisk na obie użyte miary. Standardowy wzór wygląda następująco

$$\text{F}_1 \text{ Score} = \left(\frac{2}{\text{Precyzja}^{-1} + \text{Czułość}^{-1}} \right) = 2 \cdot \frac{\text{Precyzja} \cdot \text{Czułość}}{\text{Precyzja} + \text{Czułość}}.$$

W przypadku, gdy interesuje nas bardziej precyzja lub czułość, to definiujemy bardziej ogólną formułę na F_β Score:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precyzja} \cdot \text{Czułość}}{(\beta^2 \cdot \text{Czułość}) + \text{Precyzja}},$$

gdzie parametr $\beta > 0$ odpowiada na pytanie, ile razy bardziej czułość jest ważniejsza niż precyzja. Najczęściej wykorzystuje się parametr $\beta = 2$ lub $\beta = \frac{1}{2}$, który odpowiednio zwiększa nacisk na czułość lub zmniejsza wpływ fałszywie negatywnie zaklasyfikowanych obserwacji, skupiając się bardziej na precyzji.

1.1.3 Krzywa charakterystyki pracy odbiornika

Krzywa ROC

1.1.4 Macierz kosztu

W celu wprowadzenia potrzebnych miar skuteczności wrażliwych na koszt najpierw definiujemy macierz kosztu C. Ma ona postać analogiczną do macierzy pomyłek, lecz zamiast ilość obserwacji w odpowiednich komórkach wpisujemy koszt pomyłki dla i-tej obserwacji. Warto zaznaczyć, że koszt niekoniecznie musi być kwota pieniężną, lecz także możemy rozważać ilość poświęconego czasu, bądź inną dowolną mierzalną jednostkę. W szczególności dla przypadku klasyfikacji binarnej wyróżniamy cztery możliwe wartości:

- C_{TP_i} – koszt prawdziwie pozytywnego zaklasyfikowania i-tej obserwacji,
- C_{FP_i} – koszt fałszywie pozytywnego zaklasyfikowania i-tej obserwacji,
- C_{FN_i} – koszt fałszywie negatywnego zaklasyfikowania i-tej obserwacji,
- C_{TN_i} – koszt prawdziwie negatywnego zaklasyfikowania i-tej obserwacji.

W zależności od praktycznego zastosowania wartości w macierzy kosztu przyjmują różne wartości. Poniżej przedstawiamy kilka intuicji na podstawie macierzy kosztu zdefiniowanych dla różnych eksperymentów w [2].

- W przypadku, gdy predykcja klasy pozytywnej wiąże się z podjęciem jakiejś akcji, np. sprawdzenie transakcji kartą kredytową, wysłanie oferty reklamowej, to koszt pozytywnej predykcji jest dodatni, to znaczy C_{TP_i}, $C_{FP_i} > 0$. Ponadto najczęściej te koszty są równe i przyjmują stałą wartość dla wszystkich przypadków, to znaczy $C_{TP_i} = C_{FP_i} = \text{const.}$, ponieważ podejmowana zawsze jest ta sama akcja dla wszystkich obserwacji. Dodatkowo koszt pozytywnego zaklasyfikowania dla każdej obserwacji negatywnej jest zerowy, to znaczy $C_{TN_i} = 0$, ponieważ słusznie nie została podjęta żadna akcja. Zatem ostatecznie jedyną wartością, która różni się między przypadkami, jest koszt fałszywie negatywnej klasyfikacji i on również jest większy od zera.

- W przypadku, gdy predykcja dokonywana jest przez zautomatyzowany system, to koszt prawidłowej klasyfikacji jest równy zero, to znaczy $C_{TP_i} = C_{TN_i} = 0$, ponieważ maszynie nie musimy płacić pensji, która obsługuje daną sprawę. Natomiast koszty pomyłek są różne i zazwyczaj różnią się między obserwacjami. Biorąc jako przykład ryzyko kredytowe, możemy zauważyć, że koszty mogą być zarówno dodatnie, jak i ujemne. W tym konkretnym przypadku, gdy stwierdzimy, że dana osoba spłaci kredyt, natomiast w rzeczywistości nie zrobi tego, to będziemy stratni pewną sumę pieniędzy, co da nam dodatni koszt. Z drugiej strony, gdy uznamy, że dana osoba nie spłaci kredytu, a w rzeczywistości wywiązałaby się z zobowiązań, to utracimy pewien zarobek, zatem koszt będzie negatywny.

Koncepcyjnie koszt popełnienia błędu dla każdej obserwacji powinien być większy niż koszt poprawnej klasyfikacji, to znaczy

$$C_{FP_i} > C_{TP_i} \text{ i } C_{FN_i} > C_{TN_i} \quad \forall_{i \in \{1,2,\ldots,N\}}.$$

Natomiast jak możemy zauważyć na podstawie wyżej opisanych intuicji, w praktycznych zastosowaniach jest to bardzo często nierealistyczne założenie. Przykład takiej macierzy kosztu znajduje się w tabeli 1.2.

	Stan pozytywny $y_i = 1$	Stan negatywny $y_i = 0$
Predykcja pozytywna $c_i = 1$	C_{TP_i}	C_{FP_i}
Predykcja negatywna $c_i = 0$	C_{FN_i}	C_{TN_i}

Tabela 1.2: Macierz kosztu

1.1.5 Miary skuteczności modeli wrażliwe na koszt

Motywacją do powstania miar skuteczności wrażliwych na koszt jest potrzeba ewaluacji modeli, która miarodajnie odda złożoność rozwiązywanego problemu. Przykładowo w przypadku detekcji oszustw będzie w stanie odpowiedzieć na pytanie dotyczące zaoszczędzonej kwoty dzięki modelowi predykcyjnemu. Kolejny powodem jest fakt, że podstawowe metryki nie uwzględniają różnicy w kosztach pomyłki dla fałszywie pozytywnych oraz fałszywie negatywnych klasyfikacji, traktując oba błędy jednakowo. Ponadto warto zauważyć, że

omawiane problemy wymagają uwzględnienia nie tylko różnych kosztów per rodzaj pomyłki, ale także per konkretna obserwacja. Bazując na wprowadzonej w sekcji 1.1.4 macierzy kosztu C, zdefiniujemy kilka miar.

Koszt

Rozpoczniemy od matematycznego wprowadzenia kosztu, który wyznacza jaką wartość powinniśmy wziąć z macierzy kosztu C_i dla i-tej obserwacji mając daną predykcję oraz prawdziwą wartość klasyfikacji.

$$\text{Koszt}(y_i, c_i, C_i) = y_i \left(c_i C_{TP_i} + (1 - c_i) C_{FN_i}\right) + (1 - y_i) \left(c_i C_{FP_i} + (1 - c_i) C_{TN_i}\right),$$

gdzie C_{TP_i},C_{FP_i},C_{FN_i},C_{TN_i} to odpowiednie wartości z macierzy kosztu C_i.

Koszt całkowity

Koszt całkowity (TC, z angielskiego *ang. Total Cost*) to suma kosztów poniesionych dla danego zestawy predykcji $\boldsymbol{c}$, który zostały wygenerowany przez pewien model predykcyjny. Definiowany jest wzorem

$$\text{TC}(\boldsymbol{y}_{\text{true}}, \boldsymbol{c}, \boldsymbol{C}) = \sum_{i=1}^{N} \text{Koszt}(y_i, c_i, C_i), \tag{1.1}$$

gdzie:

- $\boldsymbol{c} = (c_1, c_2, \ldots, c_N)$ - wektor przewidywanych klas,
- $\boldsymbol{C} = (C_1, C_2, \ldots, C_N)$ - wektor macierzy kosztu.

Oszczędności

Oszczędności (*ang. Savings*) to miara, którą intuicyjnie możemy rozumieć jako procentową wartość, o ile testowany model jest lepszy od bazowego klasyfikatora. Przy czym bazowym klasyfikatorem nazywamy model, który zwraca same predykcje pozytywne lub same predykcje negatywne, w zależności od tego, co bardziej się opłaca dla danego zestawu danych.

$$\text{Oszczędności}(\boldsymbol{y}_{\text{true}}, \boldsymbol{c}, \boldsymbol{C}) = \frac{\text{Koszt bazowy}(\boldsymbol{y}_{\text{true}}, C) - \text{TC}(\boldsymbol{y}_{\text{true}}, \boldsymbol{c}, \boldsymbol{C})}{\text{Koszt bazowy}(\boldsymbol{y}_{\text{true}}, C)}, \tag{1.2}$$

gdzie:

- $\text{Koszt bazowy}(\boldsymbol{y}_{\text{true}}, C) = \min\{\text{TC}(\boldsymbol{y}_{\text{true}}, \boldsymbol{c}_0, \boldsymbol{C}), \text{TC}(\boldsymbol{y}_{\text{true}}, \boldsymbol{c}_1, \boldsymbol{C})\}$ - koszt poniesiony w przypadku używania bazowego klasyfikatora,
- $\boldsymbol{c}_0 = (0, 0, \ldots, 0)$ - wektor N-elementowy predykcji równych 0,
- $\boldsymbol{c}_1 = (1, 1, \ldots, 1)$ - wektor N-elementowy predykcji równych 1.

1.2 Standardowe modele

1.2.1 Regresja logistyczna

Regresja logistyczna jest bardzo ważnym modelem statystycznym, którego celem jest modelowanie prawdopodobieństwa, że zmienna losowa Y przyjmie wartość 0 lub 1, na podstawie danych empirycznych. W celu wyprowadzenia rozważmy funkcję uogólnionego modelu liniowego parametryzowaną wektorem współczynników $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$ daną w następujący sposób

$$h_\theta(\boldsymbol{x_i}) = \frac{1}{1 + e^{-\boldsymbol{x}_i \theta^T}} = P(Y = 1 | X = \boldsymbol{x}_i; \theta). \tag{1.3}$$

Jednocześnie zauważmy, że

$$P(Y = 0 | X = \boldsymbol{x}_i; \theta) = 1 - P(Y = 1 | X = \boldsymbol{x}_i; \theta) = 1 - h_\theta(\boldsymbol{x_i})$$

oraz korzystając z faktu, że $y_i \in \{0, 1\}$, możemy zapisać

$$P(Y = y_i | \boldsymbol{x}_i; \theta) = h_\theta(\boldsymbol{x_i})^{y_i} (1 - h_\theta(\boldsymbol{x_i}))^{(1-y_i)}.$$

Następnie zakładając, że wszystkie obserwacje są niezależne i pochodzą z rozkładu Bernoulliego, wyznaczamy funkcję największej wiarygodności, aby wyestymować parametry θ.

$$L(\theta | \mathcal{S}) = \prod_{i=1}^{N} P(Y = y_i | X = \boldsymbol{x}_i; \theta) = \prod_{i=1}^{N} h_\theta(\boldsymbol{x_i})^{y_i} (1 - h_\theta(\boldsymbol{x_i}))^{(1-y_i)}$$

W celu znalezienia maksimum funkcji wiarygodności logarytmujemy funkcję $L(\theta|\mathcal{S})$ i otrzymujemy

$$J(\theta) := \log L(\theta | \mathcal{S}) = \frac{1}{N} \sum_{i=1}^{N} J_i(\theta),$$

gdzie $J_i(\theta)$ nazywana jest entropią krzyżową i przyjmuje postać

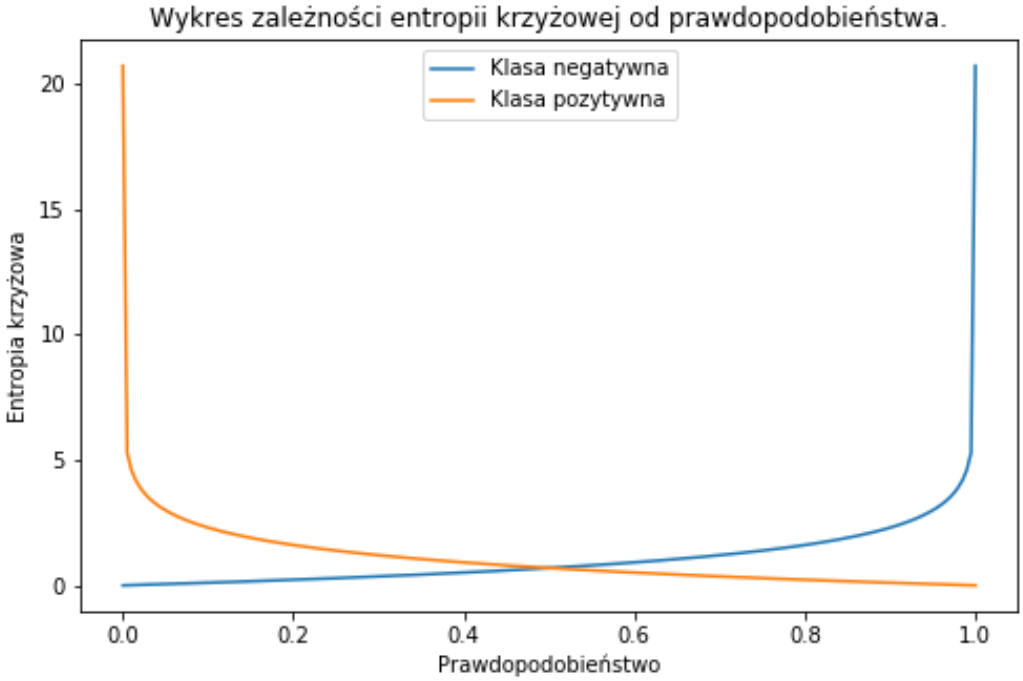
$$J_i(\theta) = -y_i \log(h_\theta(\boldsymbol{x_i})) - (1 - y_i) \log(1 - h_\theta(\boldsymbol{x_i})). \tag{1.4}$$

Niestety nie jest możliwe znalezienie bezpośredniego wzoru na współczynniki, które maksymalizują funkcję wiarygodności, zatem wykorzystuje się w tym celu algorytmy optymalizacji numerycznej, np. metodę Newtona.

Zauważmy, że maksymalizowana funkcja wiarygodności jest sumą wartości entropii krzyżowej dla poszczególnych obserwacji. W celu lepszego zrozumienia całości przyjrzymy się bliżej własnościom funkcji $J_i(\theta)$.

Wykres 1.1 ze strony 12 przedstawia przyjmowane wartości funkcji w zależności od estymowanego prawdopodobieństwa $h_\theta(\boldsymbol{x_i})$. Kolorem niebieskim jest wyznaczony przebieg funkcji dla klasy negatywnej, a pomarańczowym dla pozytywnej. Możemy zauważyć, że funkcja jest symetryczna względem wartości prawdopodobieństwa 0.5 dla obu klas. Oznacza to, że ten klasyfikator przykłada taką samą wagę dla obu rodzajów błędów. Ponadto koszt klasyfikacji dla pojedynczej obserwacji przyjmuje następujące wartości:

$$J_i(\theta) \approx \begin{cases} 0, & \text{jeżeli } y_i \approx h_\theta(\boldsymbol{x_i}), \\ \infty, & \text{jeżeli } y_i \approx (1 - h_\theta(\boldsymbol{x_i})). \end{cases}$$



Rysunek 1.1: Wykres zależności funkcji entropii krzyżowej od prawdopodobieństwa predykcji danej obserwacji. Kolor niebieski przedstawia wykres dla próbki o prawdziwej klasie pozytywnej, natomiast kolor pomarańczowy dla negatywnej.

To znaczy, że w przypadku prawidłowego zaklasyfikowania danej obserwacji funkcja przyjmuje wartość bliską zeru, natomiast w przypadku pomyłki nieskończoności. Stąd możemy wywnioskować, że koszty popełnienia bądź niepopełnienia błędu są następujące:

$$C_{TP_i} = C_{TN_i} \approx 0,$$

$$C_{FP_i} = C_{FN_i} \approx \infty.$$

... Stąd powstaje motywacja, aby zmodyfikować podaną funkcję i stworzyć regresję logistyczną, która będzie w stanie modyfikować koszty poszczególnych klasyfikacji.

1.2.2 Drzewo decyzyjne

Niech $\mathcal{Q}$ będzie zbiorem obserwacji dla danego węzła m. Ponadto niech podział będzie reprezentowany przez parę $\theta = (j, t_m)$, gdzie j to j-ty atrybut wektora $\boldsymbol{x}_i$, a t_m to warunek podziału danych na podzbiory $\mathcal{Q}_l$ oraz $\mathcal{Q}_r$, gdzie

$$\mathcal{Q}_l(\theta) = \{\boldsymbol{x}_i : \boldsymbol{x}_i \in \mathcal{Q} \wedge x_i^j \leq t_m\},$$

$$\mathcal{Q}_r(\theta) = \mathcal{Q} \setminus \mathcal{Q}_l(\theta).$$

Czystość danego węzła jest wyznaczana w następujący sposób

$$G(\mathcal{Q}, \theta) = \frac{|\mathcal{Q}^l|}{|\mathcal{Q}|} I(\mathcal{Q}_l(\theta)) + \frac{|\mathcal{Q}^r|}{|\mathcal{Q}|} I(\mathcal{Q}_r(\theta)),$$

gdzie $I(\cdot)$ może być dowolną miarą zanieczyszczenia. Ostatecznie w danym węźle wybierany jest ten podział, który minimalizuje funkcję czystości

$$\theta^* = \arg\min_{\theta} G(\mathcal{Q}, \theta).$$

Dalej następuje podział na podzbiory $\mathcal{Q}_l(\theta^*)$ i $\mathcal{Q}_r(\theta^*)$ i algorytm działa rekurencyjnie, aż do osiągnięcia maksymalnej głębokości,

Miary zanieczyszczenia dla zadań klasyfikacyjnych

- Misclassification: $I_m(\mathcal{Q}) = 1 - \max(p, 1-p)$,
- Entropy: $I_e(\mathcal{Q}) = -p\log_2 p - (1-p)\log_2(1-p)$,
- Gini: $I_g(\mathcal{Q}) = 2p(1-p)$,

oraz zadań regresyjnych

- $I_{mse}(\mathcal{Q}) = \frac{1}{|\mathcal{Q}|}\sum_{i\in\mathcal{Q}}(y_i - q)^2$,
- $I_{mae}(\mathcal{Q}) = \frac{1}{|\mathcal{Q}|}\sum_{i\in\mathcal{Q}}|y_i - q|$,

gdzie:

$$p = \frac{|\{\boldsymbol{x}_i : \boldsymbol{x}_i \in \mathcal{Q} \wedge y_i = 1\}|}{|\mathcal{Q}|},$$

$$q = \frac{\sum_{i\in\mathcal{Q}} y_i}{|\mathcal{Q}|}.$$

Pruning

Wyróżniamy trzy rodzaj algorytmów, które są odpowiedzialne za proces uczenia drzew. Poniżej krótki opis każdego z nich.

- ID3 (Iterative Dichotomiser 3) – algorytm pozwala na tworzenie drzew decyzyjnych, które w poszczególnych węzłach mogą mieć więcej niż dwa podziały. Działa jedynie dla zmiennych kategorycznych. Drzewo rośnie do maksymalnego rozmiaru, a następnie jest przycinane;
- C4.5 – następca ID3, który znosi restrykcję używania tylko kategorycznych atrybutów poprzez dyskretyzację zmiennych ciągłych. Następnie nauczone drzewo zamieniane jest w zestaw reguł, które są szeregowane względem poprawy skuteczności. Przycinanie w tym wypadku polega na usuwanie reguł, które nie poprawiają wyniku modelu;
- CART (z angielskiego *Classification and Regression Tree*) – algorytm bardzo podobny do C4.5, który umożliwia tworzenie drzew regresyjnych (w poprzednich algorytmach mamy do czynienia jedynie z zadaniami klasyfikacyjnymi) oraz nie tworzy zestawu reguł. Konstruuje on binarne drzewa, które do podziału używa atrybutu oraz progu odcięcia, który maksymalizuje przyrost informacji.

Warto zaznaczyć, że implementacja w bibliotece *sklearn* w Pythonie wykorzystuje zoptymalizowaną wersję algorytmu CART, który nie obsługuje zmiennych kategorycznych.

1.2.3 Las losowy

Las losowy oraz algorytm XGBoost są przedstawicielami szerokiej klasy modeli typu *ensemble*. Głównym celem tej metodologii jest wprowadzenie losowych perturbacji do zbioru treningowego w celu utworzenia różnych klasyfikatorów bazowych, których wspólne predykcje będą lepsze niż każdego z modeli bazowych osobno. Istnieją ponadto trzy główne powody, dlaczego te metody działają znacznie lepiej niż pojedyncze modele.

- Statystyczny – w przypadku, gdy mamy do czynienia z małym zbiorem danych, istnieje możliwość stworzenia kilku bardzo dobryh klasyfikatorów, które na zbiorze treningowym mają taką samą skuteczność. W przypadku dodatkowego zbioru testowego istnieje szansa wybrania nieoptymalnego;

- Obliczeniowy – większość algorytmów polega na optymalizacji numerycznej, która może zatrzymać się w minimum lokalnym. Dzięki zastosowaniu kilku modeli uzyskujemy szansę przeszukania większej części przestrzeni parametrów modeli;

- Reprezentacyjny – często funkcja f reprezentująca prawdziwy model jest bardzo skomplikowana i niemożliwe jest przedstawienie jej za pomocą jednego modelu. Wykorzystując złożenie kilku modeli bazowych, mamy szansę lepiej przybliżać prawdziwą funkcję f.

W celu utworzenia T klasyfikatorów bazowych należy ze zbioru treningowego $\mathcal{S}$ wybrać S_j dla $j = 1, 2, \ldots, T$ losowych podzbiorów obserwacji, które wykorzystamy do uczenia. W tym celu wykorzystuje się następujące procedury losowania.

- Wklejanie (*Pasting*) – Polega na niezależnym losowaniu $K < N$ próbek ze zbioru $\mathcal{S}$ bez powtórzeń. Najczęściej wykorzystywana jest na bardzo dużych zbiorów danych, gdzie zasoby obliczeniowe są ograniczone i nie jest możliwe wczytanie wszystkich danych do pamięci komputera. Ponadto dzięki możliwości wykorzystania obliczeń równoległych skraca się czas tworzenia całego modelu. Próbki mogą być losowane z rozkładu jednostajnego bądź wykorzystując metody biorące pod uwagę ważność poszczególnych obserwacji. [6]

- Workowanie (*ang. Bagging*) – Polega na niezależnym losowaniu $K \leq N$ próbek ze zbioru $\mathcal{S}$ z powtórzeniami. Najczęściej jest wykorzystywany w przypadku małych zbiorów danych, kiedy chcemy wykorzystać maksymalnie dużą ilość obserwacji, lecz także wprowadzić losowość do kolejnych klasyfikatorów bazowych.

- Losowe podprzestrzenie (*ang. Random Subspaces*) – Metoda polegająca na wybraniu do każdego klasyfikatora bazowego wszystkich obserwacji, lecz losuje się tylko część atrybutów opisujące dane. Głównym celem tego zabiegu jest poprawa skuteczności poprzez zwiększenie różnorodności klasyfikatorów bazowych. [16]

- Losowe łaty (*ang. Random Patches*) – Procedura łącząca metodę wklejania z losowymi podprzestrzeniami. Jej celem jest połączenie zalet obu metod i ostatecznie poprawienie skuteczności modelu z jednoczesną optymalizacją wykorzystywanego czasu oraz pamięci. [13]

Warto zauważyć, że mimo tego, że powyższe metody były oryginalnie tworzone z myślą o wykorzystaniu drzewa decyzyjnego jako klasyfikatora bazowego, to nic nie stoi na

przeszkodzie wykorzystać inny model. Ten fakt został między innymi wykorzystany w [2], gdzie wykorzystując omawiane w sekcji 1.4.2 drzewa decyzyjne wrażliwe na koszt, stworzono lasy losowe wrażliwe na koszt.

Procedura tworzenia lasu losowego jest podzielona na dwie fazy. [7] Pierwsza z nich wykorzystuje workowanie do wygenerowania odpowiednich próbek dla klasyfikatorów bazowych. Następnie trenujemy zmodyfikowane drzewa decyzyjne, które na etapie tworzenia każdego kolejnego węzła w drzewie losują podzbiór zmiennych do wyboru podziału. Finalnie podczas predykcji każdy model dokonuje klasyfikacji i ostatecznym wynikiem modelu jest większość głosów. Nie jest to jedyna możliwość agregowania wyników, przykładowo wykorzystywana w bibliotece *sklearn* w Python implementacja uśrednia wyniki prawdopodobieństwa z każdego klasyfikatora. [9]

1.2.4 XGBoost

Algorytm XGBoost jest przykładem modelu, który w iteracyjny sposób tworzy kolejne bazowe klasyfikatory. W przypadku tego algorytmu jako klasyfikator bazowy wykorzystujemy implementację drzew decyzyjnych typu CART, które nieznacznie różnią się od standardowych drzew decyzyjnych opisywanych w 1.2.2, ponieważ liść drzewa jest rozszerzony o wartość rzeczywistą, która reprezentuje decyzję modelu. Ponieważ jest to złożenie wielu klasyfikatorów, to możemy zapisać nasz model w następującej postaci:

$$\hat{y_i} = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F},$$

gdzie K oznacza liczbę drzew, f funkcje z przestrzeni $\mathcal{F}$ wszystkich możliwych drzew CART. Zatem funkcją, którą będziemy optymalizować przyjmuje następującą postać

$$\text{Obj}(\theta) = \sum_{i=1}^{n} l(y_i, y_i^{(t)}) + \sum_{k=1}^{K} \Omega(f_k) \tag{1.5}$$

W przypadku klasyfikacji przyjmujemy tak samo jak poprzednio funkcję entropii krzyżowej (1.4). Patrząc na postać modelu nie różni się ona niczym od lasu losowego. Zatem różnica między tymi modelami polega na sposobie trenowania drzew, który pokrótce opiszemy.

Ponieważ naszym modelem bazowym jest drzewo, to nie jesteśmy w stanie wprost rozwiązać zagadnienia optymalizacyjnego poprzez obliczenie gradientu funkcji i iteracyjne znalezienie rozwiązania. W tym przypadku posłużymy się treningiem addytywnym, który polega na iteracyjnym poprawianiem błędów z poprzednich modeli poprzez odpowiednie przydzielanie wag próbkom w kolejnych krokach algorytmu. Oznaczmy wartość predykcji w kroku t jako $y_i^{(t)}$.

$$y_i^{(0)} = 0$$

$$y_i^{(1)} = f_1(x_i) = y_i^{(0)} + f_1(x_i)$$

$$y_i^{(2)} = f_1(x_i) + f_2(x_i) = y_i^{(1)} + f_2(x_i)$$

$$\dots$$

$$y_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = y_i^{(t-1)} + f_t(x_i)$$

Pozostaje zagadnienie jakie drzewo chcemy wybrać w każdym z kroków. Oczywiście takie, które optymalizuje naszą funkcję Obj. Korzystając z powyższych wzorów oraz 1.5 otrzymujemy następującą postać tej funkcji:

$$\sum_{i=1}^{n} l(y_i, y_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + \text{constant} \ .$$

Korzystając z rozwinięcia szeregu Taylora dla funkcji straty otrzymujemy ogólny wzór:

$$\text{Obj}^{(t)} = \sum_{i=1}^{n} \left[l(y_i, y_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + \text{const.} \ ,$$

gdzie

$$g_i = \partial_{y_i^{(t-1)}} l(y_i, y_i^{(t-1)})$$
$$h_i = \partial^2_{y_i^{(t-1)}} l(y_i, y_i^{(t-1)})$$

Zatem po redukcji stałych, które są nieistotne z punktu widzenia optymalizacji otrzymujemy:

$$\sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)] + \Omega(f_t)$$

1.3 Cost Dependent Classification

Metody *Cost Dependent Classification* są przykładem pierwszego rodzaju modeli wrażliwych na koszt. W przypadku tych metod trening odbywa się dopiero po etapie stworzenia modelu zwracającego prawdopodobieństwa i informacja o koszcie jest uwzględniana dopiero w tej fazie modelowania. Cały proces jest przedstawiony na Rysunku 1.2. Górna część diagramu przedstawia standardowy proces uczenia modelu predykcyjnego, natomiast dolna część reprezentuje schemat dokonywania predykcji, w którym najpierw estymujemy prawdopodobieństwa dla zbioru testowego, a następnie wykorzystując te wartości oraz koszt klasyfikacji dokonujemy ostatecznej decyzji, czy dana obserwacja jest pozytywna czy negatywna. Warto wspomnieć, że w celu dokonania kolejnego treningu modelu potrzebujemy podzbioru danych, który nie był wykorzystywany do uczenia podstawowego modelu. Najczęściej taki zbiór nazywamy walidacyjnym bądź developerskim.

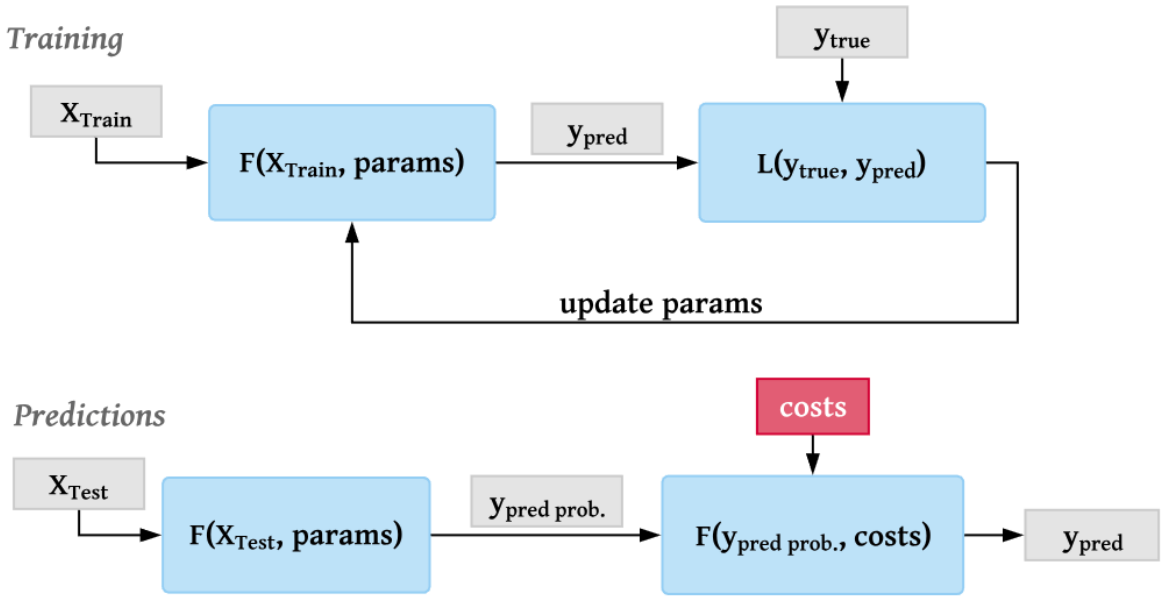
1.3.1 Optymalizacja progu

Metoda optymalizacji progu jest popularną metodą wyznaczania odpowiedniego progu prawdopodobieństwa, powyżej którego wszystkie obserwacje oznaczamy jako pozytywnie zaklasyfikowane. Może być ona wykorzystywana nie tylko do problemów wrażliwych na koszt, lecz do dowolnie zdefiniowanego zagadnienia, w którym wyznaczenie progu jest potrzebne. Jej sformułowanie wygląda następująco

$$\arg\min_{t \in [0,1]} f(\boldsymbol{y}_{\text{true}}, y_{\text{decision}}),$$

gdzie

- $f(\cdot)$ - miara skuteczność modelu, np. dokładność,
- $y_{\text{decision}} = (p_i > t)_{i=1}^{n}$ - wektor binarnych odpowiedzi modelu,



Rysunek 1.2: Schemat przedstawiający proces uczenia klasyfikatora wrażliwego na koszt. Źródło: `https://towardsdatascience.com/fraud-detection-with-cost-sensitive-machine-learning-24b8760d35d9`

- t - wartość progu decyzyjnego.

W naszym przypadku funkcja f to funkcja kosztu całkowitego (1.1). Innymi słowy, w tym przypadku będziemy szukać takiego progu decyzyjnego, który zminimalizuje sumaryczną wartość kosztów.

1.3.2 Minimalizacja ryzyka bayesowskiego

Metoda minimalizacji ryzyka bayesowskiego (*ang. Bayesian Minimum Risk*) polega na przypisaniu odpowiedniej wartości reprezentującej ryzyko zaklasyfikowania danej obserwacji jako pozytywnej lub negatywnej, które definiujemy w następujący sposób:

$$R(p_1|\boldsymbol{x}_i) = L(p_1|y_1)P(y_1|\boldsymbol{x}_i) + L(p_1|y_0)P(y_0|\boldsymbol{x}_i),$$

$$R(p_0|\boldsymbol{x}_i) = L(p_0|y_0)P(y_0|\boldsymbol{x}_i) + L(p_0|y_1)P(y_1|\boldsymbol{x}_i),$$

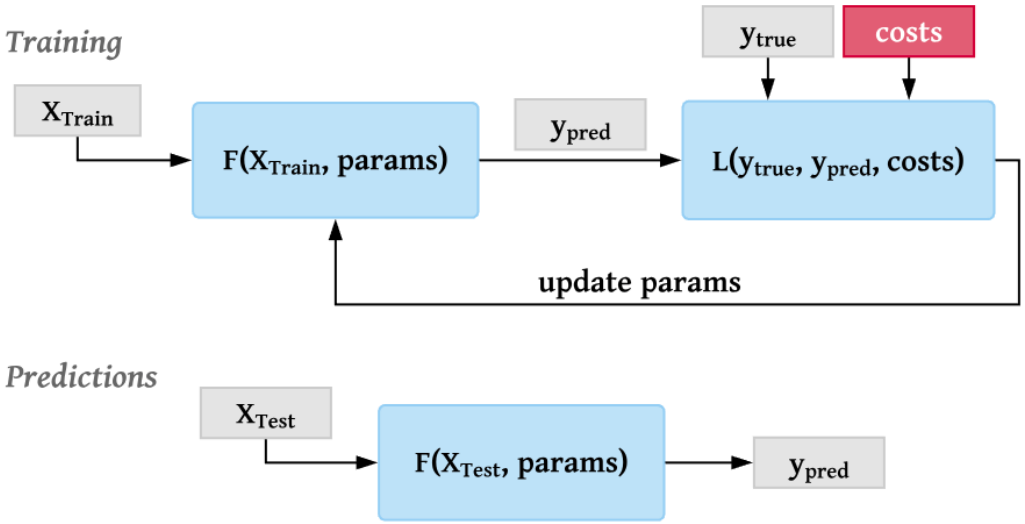
gdzie

- $P(p_1|\boldsymbol{x}_i)$, $P(p_0|\boldsymbol{x}_i)$ - oznaczają estymowane przez model prawdopodobieństwa zaklasyfikowania obserwacji jako odpowiednio pozytywna oraz negatywna klasa i $P(p_0|\boldsymbol{x}_i) = 1 - P(p_1|\boldsymbol{x}_i)$,
- $L(p_i|y_j)$ oraz $i, j \in \{l, f\}$ - funkcja kosztu.

Klasyfikacja danej obserwacji jest opisana następującą nierównością:

$$R(p_1|\boldsymbol{x}_i) \leq R(p_0|\boldsymbol{x}_i)$$

I oznacza ona, że klasyfikujemy dany przykład jako pozytywny, jeżeli ryzyko takiej decyzji jest mniejsze niż zaklasyfikowania danej obserwacji jako negatywną. Po przeprowadzaniu



Rysunek 1.3: Schemat przedstawiający proces uczenia modelu wrażliwego na koszt. Źródło: `https://towardsdatascience.com/fraud-detection-with-cost-sensitive-machine-learning-24b8760d35d9`

odpowiednich przekształceń algebraicznych otrzymujemy następujący wzór na klasyfikację przykładu jako pozytywny:

$$P(p_1|\boldsymbol{x}_i) \geq \frac{L(p_1|y_0) - L(p_0|y_0)}{L(p_0|y_1) - L(p_1|y_1) - L(p_0|y_0) + L(p_1|y_0)}.$$

W przypadku gdy za funkcję kosztu przyjmiemy odpowiednie wartość z macierzy kosztu, to otrzymujemy następujący prób decyzyjny:

$$p \geq \frac{C_{FP_i} - C_{TN_i}}{C_{FN_i} - C_{TP_i} - C_{TN_i} + C_{FP_i}}.$$

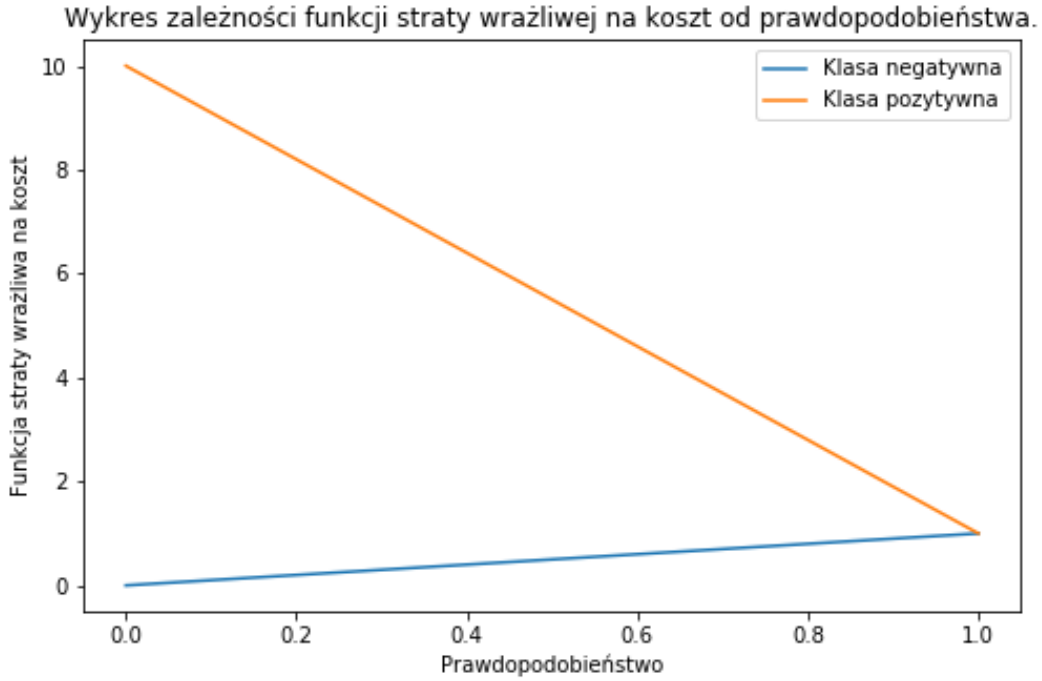
1.4 Cost Sensitive Training

Drugą podgrupą metod wrażliwych na koszt jest *Cost Sensitive Trainig*. Są to metody, które już na etapie treningu modelu biorą pod uwagę koszt związany z klasyfikacją danej obserwacji. Schemat uczenia modelu jest przedstawiony na Rysunku 1.3. Model jako wejście otrzymuje zbiór danych treningowych, następnie dokonuje predykcji i na bazie prawdziwych odpowiedzi oraz kosztów wyznaczana jest skuteczność modelu i kolejno aktualizowane są wagi, a cały proces jest iteracyjnie powtarzany aż do momentu osiągnięcia zadanego kryterium stopu.

1.4.1 Regresja logistyczna wrażliwa na koszt

Pierwszy modelem, który stosunkowo łatwo przystosować do bycia wrażliwym na koszt jest regresja logistyczna. Kontynuując rozważania z podrozdziału 1.2.1 chcemy, aby funkcja straty przyjmowała następujące wartości, które odpowiadają wartościom z macierzy kosztu:

$$J_i^c(\theta) = \begin{cases} C_{TP_i}, & \text{jeżeli } y_i = 1 \text{ i } h_\theta(\boldsymbol{x_i}) \approx 1, \\ C_{TN_i}, & \text{jeżeli } y_i = 0 \text{ i } h_\theta(\boldsymbol{x_i}) \approx 0, \\ C_{FP_i}, & \text{jeżeli } y_i = 0 \text{ i } h_\theta(\boldsymbol{x_i}) \approx 1, \\ C_{FN_i}, & \text{jeżeli } y_i = 1 \text{ i } h_\theta(\boldsymbol{x_i}) \approx 0. \end{cases}$$



Rysunek 1.4: Wykres zależności wrażliwej na koszt funkcji straty od prawdopodobieństwa predykcji danej obserwacji. Kolor niebieski przedstawia wykres dla próbki o prawdziwej klasie pozytywnej, natomiast kolor pomarańczowy dla negatywnej. Wykres dla przykładowych wartości: $C_{\text{TP}} = 1$, $C_{\text{FN}} = 10$, $C_{\text{FP}} = 1$, $C_{\text{TN}} = 0$.

W rezultacie funkcją, która zachowuje się w powyższy sposób jest:

$$J^c(\theta) = \frac{1}{N}\sum_{i=1}^{N}\Big(y_i\Big(h_\theta(\boldsymbol{x_i})C_{TP_i} + (1 - h_\theta(\boldsymbol{x_i}))C_{FN_i}\Big) + \\ + (1 - y_i)\Big(h_\theta(\boldsymbol{x_i})C_{FP_i} + (1 - h_\theta(\boldsymbol{x_i}))C_{TN_i}\Big)\Big)$$

Następnie standardowo wykorzystując algorytm optymalizujący, który znajdzie odpowiednie współczynniki modelu trenujemy model predykcyjny.

1.4.2 Drzewo decyzyjne wrażliwe na koszt

Analogicznie jak w przypadku regresji logistycznej w celu wprowadzenia kosztu do treningu drzewa decyzyjnego musimy uzależnić proces powstawania modelu od kosztu danej próbki. Naturalnym miejscem dla drzewa decyzyjnego jest proces podziału danego węzła na kolejne węzły bądź liście. Dlatego definiujemy następującą miarę zanieczyszczenia, która następujący wzór:

$$I_c(\mathcal{S}) = min\left\{Cost(f_0(\mathcal{S})), Cost(f_1(\mathcal{S}))\right\}.$$

Dodatkowo w przy dokonywaniu predykcji klasyfikacja na wartość pozytywną bądź negatywną następuje wg następującego kryterium:

$$f(\mathcal{S}) = \begin{cases} 0, & \text{jeżeli Cost}(f_0(\mathcal{S})) \leq \text{Cost}(f_1(\mathcal{S})), \\ 1, & \text{w przeciwnym wypadku,} \end{cases}$$

gdzie jak zawsze 1 oznacza wartość pozytywną, a 0 negatywną. Z poprzednich rozważań oczywiście możliwe jest stworzenie modeli typu *ensemble*, w których klasyfikatorem bazowym byłoby drzewo decyzyjne wrażliwe na koszt, natomiast jest to poza obszarem badań aktualnej pracy.

Rozdział 2

Eksperyment

W celu porównania skuteczności metod wrażliwych na koszt względem siebie wykonamy eksperyment, którego celem będzie sprawdzenie, który z modeli lub ich kombinacja przyniesie największe oszczędności na zbiorze danych dot. oszustw z wykorzystaniem kart kredytowych. Jako dodatkowy wskaźnik wykorzystamy miarę skuteczności modelu F1 Score, która będzie w stanie odpowiedzieć jak zmieniła się ogólna skuteczność modelu.

2.1 Dane

Do eksperymentu zostanie wykorzystany zbiór danych Credit Card Fraud Detection.[1] Zawiera on 284,807 transakcji w tym zaledwie 492 oszustw. Tabela składa się z 30 kolumn, w tym 28 z nich są to zanonimizowane zmienne numeryczne, które były wcześniej poddane transformacji PCA (*ang. Principal Component Analysis*), a dodatkowo posiadamy informacje dot. czasu transakcji oraz kwoty. Ponadto wśród danych nie ma brakujących wartości. Podczas modelowania pominiemy zmienną czasową, ponieważ sama w sobie nie zawiera ona istotnej informacji, natomiast stworzenie znaczących zmiennych nie jest istotą przeprowadzanego eksperymentu.

Mimo tego, że dane zostały poddane anonimizacji, to na podstawie [5] możemy domyślać się z jakimi zmiennymi mieliśmy do czynienia przed transformacjami. Podczas procesu dokonywania transakcji standardowo zbierane są:

- data dokonania transakcji,
- numer konta,
- numer karty,
- typ transakcji (płatność internetowa, płatność stacjonarna, wypłata z bankomatu),
- kwota,
- ID beneficjenta transakcji
- grupa beneficjenta transakcji (przykładowo linie lotnicze, hotel, wypożyczalnia samochodów itp.),
- kraj dokonania transakcji

[1] Źródło: `https://www.kaggle.com/mlg-ulb/creditcardfraud`

- kraj zamieszkania właściciela karty,
- typ karty (przykładowo VISA, MasterCard itp.),
- wiek klienta,
- płeć klienta,
- bank klienta,
- historyczna informacja czy dana transakcja była oszustwem.

Na podstawie wymienionych informacji tworzy się agregaty czasowe bazujące na historii, które składają się z następujących kombinacji zmiennych:

- Klient, karta kredytowa;
- Typ transakcji, beneficjent transakcji, kategoria beneficjenta, kraj;
- W ciągu ostatnich godzin/dni/tygodni/miesięcy;
- Ilość, średnia lub suma transakcji.

Przykładowymi zmiennymi, które powstają w tym procesie są: Ilość transakcji dla tego samego klienta w ciągu ostatnich 6/12/24 godzin, Średnia wartość transakcji dla danego klienta z ostatniego tygodnia.

Informacja, która jest najbardziej istotna z punktu widzenia naszego modelowania, to rozkład oraz charakterystyka zmiennej dot. kwoty transakcji. Na Rysunku **??** znajduje się wykres rozkładu ...

Warto zauważyć, że najpopularniejszymi wartościami transakcji są:

Na podstawie przeprowadzonej analizy definiujemy w Tabeli 2.1 macierz kosztu dla tego eksperymentu, gdzie C_a to koszt administracyjny podjęcia sprawy przez analityka, który sprawdza, czy dana obserwacja jest faktycznie oszustwem niezależnie od tego, jaki był jego ostateczny werdykt, Amt_i to wartość transakcji, jaką utracimy w przypadku niewskazanie danej obserwacji jako oszustwa, natomiast zerowa wartość kosztu dla normalnej transakcji, która była prawidłowo wskazana, wynika z braku konieczności podjęcia inwestygacji oraz strat z tego wynikających.

	Stan pozytywny $y_i = 1$	Stan negatywny $y_i = 0$
Predykcja pozytywna $c_i = 1$	$\mathrm{C_{TP_i}} = \mathrm{C_a}$	$\mathrm{C_{FP_i}} = \mathrm{C_a}$
Predykcja negatywna $c_i = 0$	$\mathrm{C_{FN_i}} = \mathrm{Amt_i}$	$\mathrm{C_{TN_i}} = 0$

Tabela 2.1: Macierz kosztu eksperymentu.

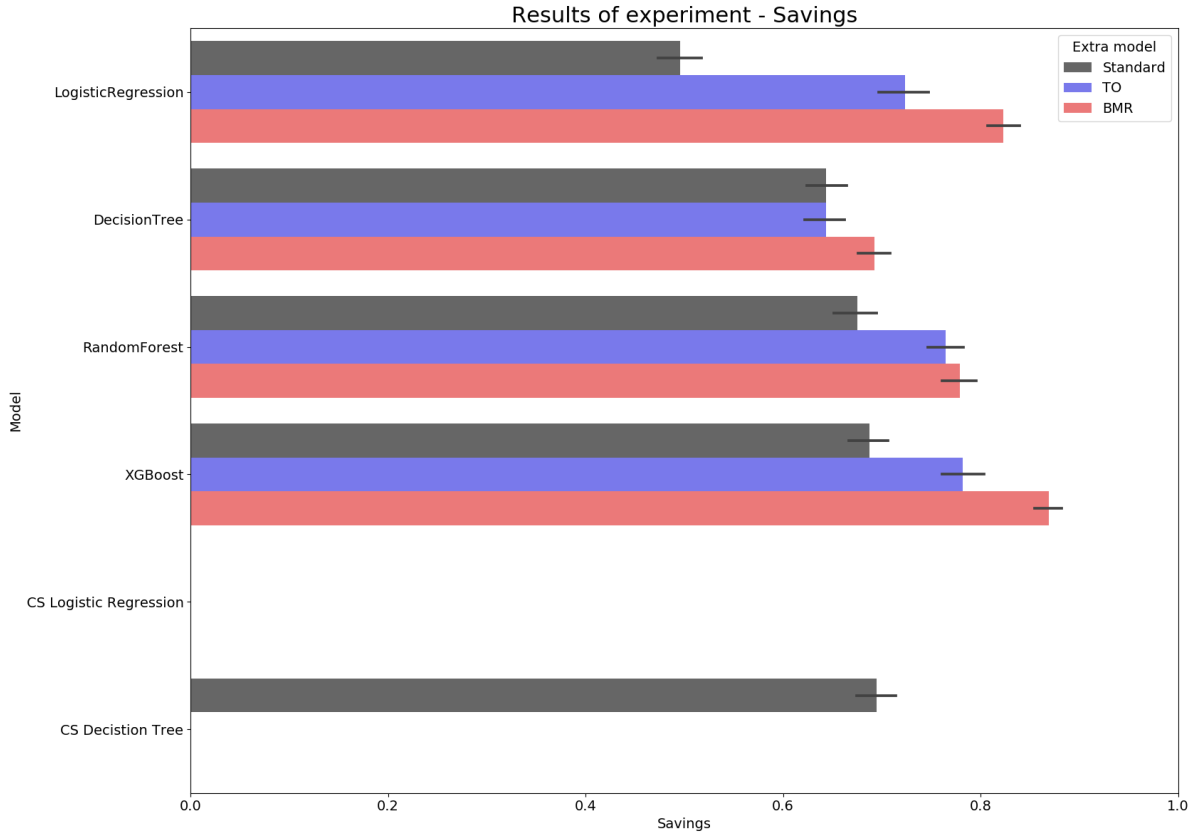
2.2 Opis eksperymentu

Eksperyment został przeprowadzony w następujący sposób: 50-krotnie dzielimy zbiór danych w proporcjach 50:17:33 na zbiór treningowy, walidacyjny oraz testowy. Następnie uczymy wszystkie modele na zbiorze treningowym. Dla modelu XGBoost wykorzystujemy zbiór walidacyjny do procesu wczesnego zatrzymywania (*ang. Early stopping*), natomiast dla modeli BMR oraz TO korzystamy z tego zbioru jako zbiór treningowy. Następnie dla wszystkich modeli dokonujemy predykcyji na zbiorze testowym i mierzymy skuteczność typowań. Warto wspomnieć, że 50-krotne losowanie nowych podziałów zbioru ma na celu zmniejszenie ryzyka, że wynik zależy od wylosowanej próbki. Ponadto wykorzystano tą technikę w opozycji do standardowej warstwowej walidacji krzyżowej (*ang. Stratified Cross Validation*) z uwagi na chęć uniknięcia zbyt małej próbki testowej w przypadku walidacji krzyżowej z dużą ilością podziałów. Następnie dla każdego modelu została obliczona wartość średnia oraz odchylenie standardowe dla wcześniej określonych miar.

Do implementacji skryptów został wykorzystany język programowania Python wraz z bibliotekami costcla, sklearn, pandas, numpy, matplotlib oraz język programowania bash.

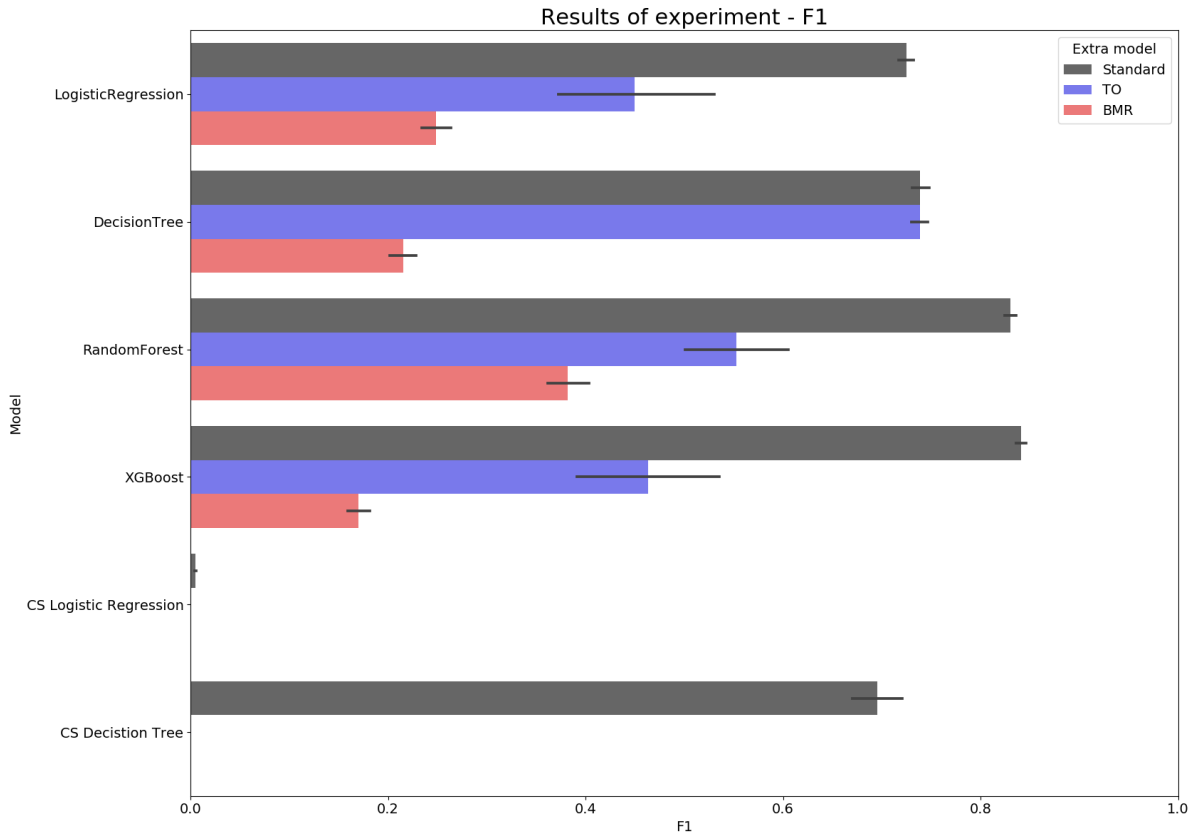
2.3 Wyniki

Lorem ipsum ...



Rysunek 2.1: Wyniki eksperymentu dla miary skuteczności Oszczędności. Źródło: Opracowanie własne.

Korzystając z Rysunku 2.1 możemy zauważyć, że: ...
Korzystając z Rysunku 2.2 możemy zauważyć, że: ...



Rysunek 2.2: Wyniki eksperymentu dla miary skuteczności F1-Score. Źródło: Opracowanie własne.

Podsumowanie

Bibliografia

[1] Bahnsen, A. C., Aouada, D., Ottersten, B. Example-dependent cost-sensitive logistic regression for credit scoring.

[2] Bahnsen, A. C., Aouada, D., Ottersten, B. Ensemble of example-dependent cost-sensitive decision trees, 2015.

[3] Bahnsen, A. C., Aouada, D., Ottersten, B. Example-dependent cost-sensitive decision trees. *Expert Syst. Appl. 42*, 19 (Nov. 2015), 6609–6619.

[4] Bahnsen, A. C., Stojanovic, A., Aouada, D., Ottersten, B. *Improving Credit Card Fraud Detection with Calibrated Probabilities*. pp. 677–685.

[5] Bahnsen, A. C., Stojanovic, A., Aouada, D., Ottersten, B. Cost sensitive credit card fraud detection using bayes minimum risk. In *2013 12th International Conference on Machine Learning and Applications* (Dec 2013), vol. 1, pp. 333–338.

[6] Breiman, L. Pasting small votes for classification in large databases and on-line. *Machine Learning 36*, 1 (Jul 1999), 85–103.

[7] Breiman, L. Random forests. *Machine Learning 45*, 1 (Oct 2001), 5–32.

[8] Brodersen, K. H., Ong, C. S., Stephan, K. E., Buhmann, J. M. The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition* (Aug 2010), pp. 3121–3124.

[9] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* (2013), pp. 108–122.

[10] Chen, T., Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2016), KDD '16, ACM, pp. 785–794.

[11] Dietterich, T. G. Ensemble methods in machine learning. In *Multiple Classifier Systems* (Berlin, Heidelberg, 2000), Springer Berlin Heidelberg, pp. 1–15.

[12] Elkan, C. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2* (San Francisco, CA, USA, 2001), IJCAI'01, Morgan Kaufmann Publishers Inc., pp. 973–978.

[13] Louppe, G., Geurts, P. Ensembles on random patches. pp. 346–361.

[14] Powers, D., Ailab. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *J. Mach. Learn. Technol 2* (01 2011), 2229–3981.

[15] Sheng, V. S., Ling, C. X. Thresholding for making classifiers cost-sensitive. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1* (2006), AAAI'06, AAAI Press, pp. 476–481.

[16] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 8 (Aug 1998), 832–844.