



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 nieznanym, 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*

pieczęćka 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 modeli wrażliwe na koszt | 7 |
| 1.2 Standardowe modele | 8 |
| 1.2.1 Regresja logistyczna | 8 |
| 1.2.2 Drzewo klasyfikacyjne | 9 |
| 1.2.3 Las losowy | 10 |
| 1.2.4 XGBoost | 11 |
| 1.3 Cost Dependent Classification | 12 |
| 1.3.1 Optymalizacja progu | 12 |
| 1.3.2 Minimalizacja ryzyka bayesowskiego | 13 |
| 1.4 Cost Sensitive Training | 14 |
| 1.4.1 Regresja logistyczna wrażliwa na koszt | 14 |
| 1.4.2 Drzewo decyzyjne wrażliwe na koszt | 16 |
| 2 Eksperyment | 17 |
| 2.1 Dane | 17 |
| 2.2 Opis eksperymentu | 18 |
| 2.3 Wyniki | 18 |
| Bibliografia | 23 |

Wstęp

Historia kart płatniczych zaczyna się już w latach 80. XIX wieku, kiedy Edward Belamy 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 dokonywanie 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 nieznanemu, utrata lub kradzież karty, skopiowanie danych karty lub kradzież przesyłki z kartą, które najczęściej kończą się nieautoryzowanymi płatnościami na koncie.

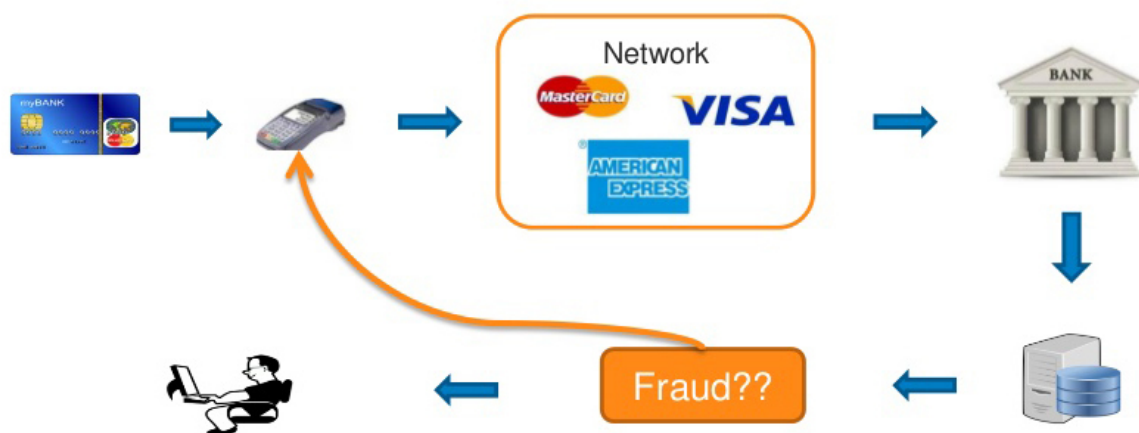
Banki oraz instytucje finansowe w celu przeciwdziałania tym przestępstwom korzystają z systemów detekcji oszustw. Jest to jeden z elementów całego procesu dokonywania płatności, który jest przedstawiony na rysunku 1 na stronie 2. W momencie przyłożenia karty kredytowej do terminala nawiązywane jest połączenie 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

¹Źródło: https://pl.wikipedia.org/wiki/Karta_p%C5%82atnicza (dostęp: 27.12.2019)

²Źródło: <https://www.kartyonline.pl/historia-kart-platniczych.php> (dostęp: 27.12.2019)

³Źródło: https://en.wikipedia.org/wiki/Payment_card (dostęp: 27.12.2019)

⁴Źródło: <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 predyktorycznego 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 predyktorycznymi 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

⁵Wystąpienie A.Bahnsena podczas Konferencji Analytics 2013. Źródło: <https://www.youtube.com/watch?v=YCNkxaVDiAO>

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 opisanie macierzy pomyłek, macierzy kosztu oraz związanymi z nimi miarami skuteczności modeli predykcyjnych, które służą do badania jakości predykcji. Następnie przejdziemy do modeli standardowych, które nie biorą pod uwagę różnych kosztów popełniania błędu klasyfikacyjnego. W ramach tej grupy opiszemy regresję logistyczną, drzewo decyzyjne, las losowy oraz algorytm XGBoost. Na końcu opiszemy modele wrażliwe 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 Pierwsze z nich zawierają 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} - zbiór wszystkich obserwacji,
- \mathbf{x}_i - i -ty wektor obserwacji,
- $y_i \in \{0, 1\}$ - prawdziwa klasyfikacja dla i -tej obserwacji. Wartość 0 oznacza klasyfikację negatywną, natomiast wartość 1 oznacza klasyfikację pozytywną.
- $p_i \in [0, 1]$ - przewidywana wartość prawdopodobieństwa dla i -tej obserwacji,
- $c_i \in \{0, 1\}$ - predykcja klasyfikacji dla i -tej obserwacji. Oznaczenia analogicznie jak dla y_i .

¹Źródło: <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 klasyfikacji 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;
 - Fałszywie negatywny wynik klasyfikacji (FN, z angielskiego *False Positive*) – obserwacja była pozytywna i zaklasyfikowaliśmy ją jako negatywną.
- Liczbę obserwacji należących do każdej z klas możemy zapisać w tabeli, tak jak tabela 1.1.

| | | Prawdziwy stan | |
|-----------|-----------|----------------|-----------|
| | | Pozytywna | Negatywna |
| Predykcja | Pozytywna | TP | FP |
| | Negatywna | FN | TN |

Tabela 1.1: Macierz pomyłek

Dodatkowo na podstawie macierzy pomyłek możemy zdefiniować następujące miary skuteczności modeli:

Dokładność, która określa jaki procent wszystkich obserwacji zaklasyfikowaliśmy poprawnie.

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

Precyzja, która mierzy ile procent zaklasyfikowanych próbek jako pozytywne faktycznie były pozytywne.

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

Czułość, która mówi Nam o procencie znalezionych wszystkich pozytywnych przypadków.

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

F1 Score, który jest średnią harmoniczną precyzji oraz czułości.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precyzja} \cdot \text{Czułość}}{\text{Precyzja} + \text{Czułość}}$$

Wszystkie wyżej wspomniane miary skuteczności przyjmują wartości z zakresu $[0, 1]$ oraz wyższa wartość oznacza lepszy model.

1.1.2 Miary skuteczności modeli wrażliwe na koszt

Motywacją do powstania miar wrażliwych na koszt jest rzeczywista ewaluacja modeli. Podstawowe metryki nie uwzględniają różnicy w kosztach pomyłki dla FP i FN. Ponadto koszt oszustwa znacząco różni się w zależności od przypadku.

W celu wprowadzenia potrzebnych miar skuteczności potrzebujemy wprowadzić tzw. macierz kosztu, która jest zaprezentowana w Tabeli 1.2, gdzie poszczególne komórki oznacza odpowiadające wartość kosztu predykcji. Następnie definiujemy następującą wartość:

| Prawda | Predykcja | |
|--------|------------|------------|
| | Oszustwo | Normalna |
| | Oszustwo | Normalna |
| | C_{TP_i} | C_{FN_i} |
| | C_{FP_i} | C_{TN_i} |

Tabela 1.2: Macierz kosztu

$$\text{Koszt}(f(\mathbf{x}_i^*)) = y_i(c_i C_{TP_i} + (1 - c_i) C_{FN_i}) + (1 - y_i)(c_i C_{FP_i} + (1 - c_i) C_{TN_i}),$$

gdzie

- $\mathbf{x}_i^* = [\mathbf{x}_i, C_{TP_i}, C_{FP_i}, C_{FN_i}, C_{TN_i}]$ - wektor zawierający wartości cech i -tej obserwacji rozszerzony o koszt klasyfikacji,
- C_i - koszt klasyfikacji i -tej obserwacji,
- $f(\cdot)$ - model predykcyjny,

Następnie wprowadzamy następujące miary skuteczności modeli:

$$\text{Koszt całkowity}(f(\mathcal{S})) = \sum_{i=1}^N \text{Cost}(f(\mathbf{x}_i^*)), \quad (1.1)$$

$$\text{Oszczędności} = \frac{\text{Koszt}_n(\mathcal{S}) - \text{Koszt}(f(\mathcal{S}))}{\text{Koszt}_n(\mathcal{S})}, \quad (1.2)$$

gdzie:

- $\text{Koszt}_n = \min\{\text{Koszt}(f_0(\mathcal{S})), \text{Koszt}(f_1(\mathcal{S}))\}$,
- $f_a(\mathcal{S}) = \mathbf{a}$ gdzie $a \in \{0, 1\}$.

Wartość $f_a(\mathcal{S})$ możemy rozumieć jako przypadek naiwnego modelu, który wszystkim obserwacjom przyznaje wartość a . Natomiast Koszt_0 oznacza wybranie naiwnego klasyfikatora, który generuje mniejsze koszty. Ostatecznie oszczędności możemy rozumieć jako procentową wartość o ile testowany model jest lepszy od naiwnego klasyfikatora.

1.2 Standardowe modele

1.2.1 Regresja logistyczna

Regresja logistyczna należy do jednego z najbardziej podstawowych modeli statystycznych używanych do problemów klasyfikacyjnych. Jest to model, w którym zakładamy, że logarytm szans jest kombinacją liniową predyktorów, co możemy zapisać jako:

$$\log \frac{\hat{p}}{1 - \hat{p}} = \sum_{j=1}^k \theta^{(j)} x_i^{(j)}$$

Po nałożeniu odpowiednich przekształceń ostatecznie otrzymujemy, że estymowane prawdopodobieństwo możemy zapisać w następującej postaci

$$\hat{p} = P(y = 1 | \mathbf{x}_i) = h_{\theta}(\mathbf{x}_i) = g \left(\sum_{j=1}^k \theta^{(j)} x_i^{(j)} \right) \quad (1.3)$$

gdzie funkcja $g(z)$ to sigmoidalna funkcja łącząca, która jest opisana wzorem

$$g(z) = \frac{1}{1 + e^{-z}}.$$

W celu znalezienia odpowiednich parametrów definiujemy standardową funkcję straty, która przyjmuje następującą postać:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N J_i(\theta),$$

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

$$J_i(\theta) = -y_i \log(h_{\theta}(\mathbf{x}_i)) - (1 - y_i) \log(1 - h_{\theta}(\mathbf{x}_i)). \quad (1.4)$$

Następnie wykorzystując algorytm optymalizacyjny, który będzie w stanie znaleźć minimum funkcji straty, znajdujemy odpowiednie parametry modelu.

Korzystając z Rysunku 1.1 możemy zauważyć, że koszt klasyfikacji przyjmuje następujące wartości:

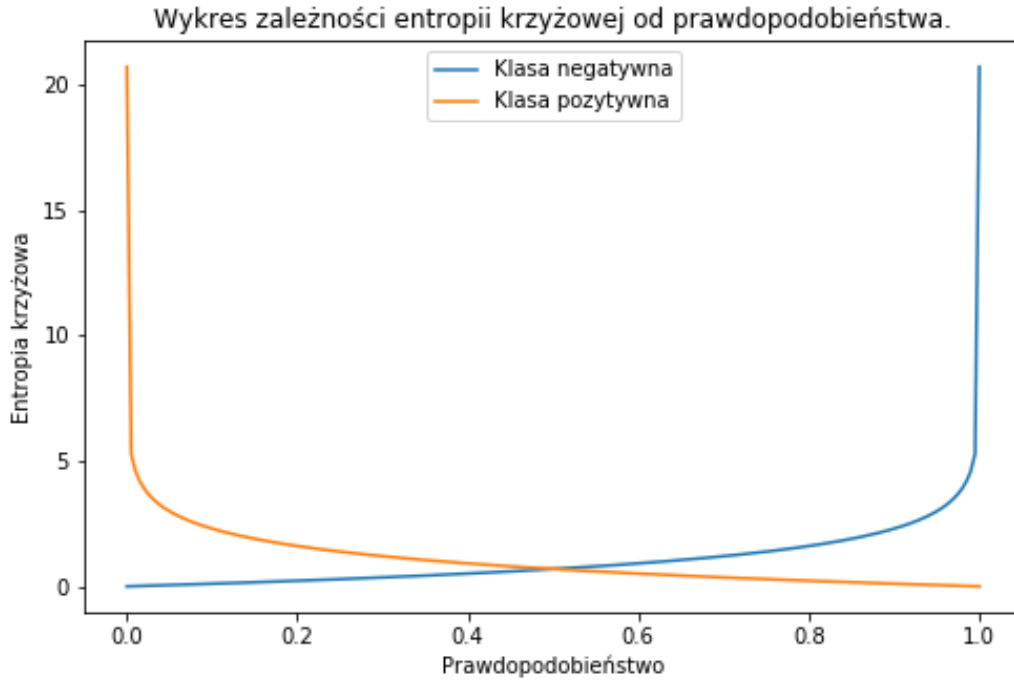
$$J_i(\theta) \approx \begin{cases} 0, & \text{jeżeli } y_i \approx h_{\theta}(\mathbf{x}_i), \\ \infty, & \text{jeżeli } y_i \approx (1 - h_{\theta}(\mathbf{x}_i)). \end{cases}$$

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

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

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

. Porównując otrzymane rezultaty z macierzą kosztów możemy zauważyć, że te wartości znacząco odbiegają od tego co w rzeczywistości chcemy otrzymać. Stąd powstaje motywacja, aby zmodyfikować podaną funkcję celu i stworzyć regresję logistyczną wrażliwą na koszt, która będzie optymalizować właściwe dla Nas wartości.



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.

1.2.2 Drzewo klasyfikacyjne

Drzewo klasyfikacyjne to przykład jednego z rodzaju drzew decyzyjnych, którego celem jest znalezienie najlepszego rozróżnienia pomiędzy klasami. W ogólności drzewo decyzyjne składa się z zestawu reguł.

Drzewo składa się z węzłów, które są reprezentowane przez parę (\mathbf{x}^j, l^j) , która oznacza podział zbioru obserwacji \mathcal{S} na dwa zbiory: \mathcal{S}^l oraz \mathcal{S}^r względem wektora obserwacji \mathbf{x} oraz progu decyzyjnego l^j w następujący sposób:

$$\mathcal{S}^l = \{\mathbf{x}^* : \mathbf{x}^* \in \mathcal{S} \wedge x_i^j \leq l^j\},$$

$$\mathcal{S}^r = \{\mathbf{x}^* : \mathbf{x}^* \in \mathcal{S} \wedge x_i^j > l^j\},$$

gdzie \mathbf{x}^j reprezentuje j -ty atrybut wektora \mathbf{x} . Ponadto l^j jest wartością taką, że $\min \mathbf{x}^j \leq l^j < \max \mathbf{x}^j$. Ponadto warto zauważyć, że $\mathcal{S}^l \cup \mathcal{S}^r = \mathcal{S}$, co oznacza, że nasz podział rozdziela wektor obserwacji na dokładnie dwa rozłączne zbiory. Po znalezieniu optymalnego podziału zliczamy ilość pozytywnych próbek:

$$\mathcal{S}_1 = \{\mathbf{x}^* : \mathbf{x}^* \in \mathcal{S} \wedge y_i = 1\},$$

a następnie zliczamy procent pozytywnych próbek jako:

$$\pi_1 = \frac{|\mathcal{S}_1|}{|\mathcal{S}|}.$$

Następnie dla każdego z liści jest obliczana wielkość jego zanieczyszczenia.

- Misclassification: $I_m(\pi_1) = 1 - \max(\pi_1, 1 - \pi_1)$
- Entropy: $I_e(\pi_1) = -\pi_1 \log(\pi_1) - (1 - \pi_1) \log(1 - \pi_1)$
- Gini: $I_g(\pi_1) = 2\pi_1(1 - \pi_1)$

Następnie dla każdego proponowanego podziału dla danej reguły (\mathbf{x}^j, l^j) liczony jest przyrost czystości w następujący sposób:

$$\text{Gain}(\mathbf{x}^j, l^j) = I(\pi_1) - \frac{|\mathcal{S}^l|}{|\mathcal{S}|} I(\pi_1^l) - \frac{|\mathcal{S}^r|}{|\mathcal{S}|} I(\pi_1^r),$$

gdzie $I(\pi_1)$ może być dowolną z zaproponowanych miar zanieczyszczenia. Ostatecznie wybiera się ten podział, który maksymalizuje przyrost czystości, a następnie dzieli się zbiór \mathcal{S} na podzbiory \mathcal{S}^l i \mathcal{S}^r . W taki sposób jest pokonywany pojedynczy podział zbioru dla konkretnego węzła. Całe drzewo tworzy się poprzez kolejne podziały węzłów aż do momentu dotarcia przez algorytm do kryterium stopu.

1.2.3 Las losowy

Kolejne modele są przedstawicielami szerokiej klasy metod *ensemble*, które wykorzystują metody próbkowania w celu utworzenia różnych klasyfikatorów bazowych, aby następnie dokonać ostatecznej predykcji. Metody te są używane, aby zredukować wariancję klasyfikatora bazowego (zazwyczaj drzewa decyzyjnego) poprzez losowe dobieranie próbek i/lub zmiennych, na których model będzie uczony. W wielu przypadkach stworzenie modelu opartego o *bagging* jest znacznie prostsze, ponieważ wymaga jedynie zmiany próbkowania danych bez naruszania modelu bazowego, natomiast metody typu boosting wymagają zmiany całego algorytmu. Z licznych obserwacji wynika, że pierwsza z metod dużo lepiej radzi sobie mają jako bazowe klasyfikatory złożone modele, w przeciwieństwie do drugiej, która zazwyczaj osiąga najlepsze wyniki wykorzystując proste modele (np. płytkie drzewa decyzyjne, tzn. o małej głębokości).

Przykładowe metody losowania próbek do modeli bazowych:

- *Pasting* – losowanie obserwacji bez powtórzeń
- *Bagging* – losowanie obserwacji z powtórzeniami
- *Random Subspaces* – losowanie podzbioru zmiennych
- *Random Patches* – losowanie podzbioru zmiennych oraz obserwacji

W przypadku lasu losowego proces losowania próbek jest podzielony na dwie fazy. Pierwsza z nich polega na próbkowaniu z powtórzeniami obserwacji ze zbioru treningowego dla każdego z osobnych estymatorów bazowych. Następnie podczas fazy tworzenia kolejnych węzłów w drzewach wybierany jest losowy podzbiór zmiennych, które mogą być w tym kroku wykorzystane.

Celem tych dwóch różnych źródeł losowości jest redukcja wariancji lasu. Pojedyncze drzewa mają tendencję do zbytowego dopasowywania się do danych, zatem losowanie poszczególnych zmiennych w każdym kolejnym węźle pomaga to zredukować. Natomiast losowanie różnych próbek do każdego z klasyfikatorów pozwala na stworzenie lekko odmiennych modeli bazowych.

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 funkcję, którą będziemy optymalizować przyjmuje następującą postać

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

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 poprawianiu błędów z poprzednich modeli poprzez odpowiednie przydzielanie wag próbkom w kolejnych krokach algorytmu. Oznaczmy wartość predykcji w kroku t jako $y_i^{(t)}$.

$$\begin{aligned} y_i^{(0)} &= 0 \\ y_i^{(1)} &= f_1(x_i) = y_i^{(0)} + f_1(x_i) \\ y_i^{(2)} &= f_1(x_i) + f_2(x_i) = y_i^{(1)} + f_2(x_i) \\ &\dots \\ y_i^{(t)} &= \sum_{k=1}^t f_k(x_i) = y_i^{(t-1)} + f_t(x_i) \end{aligned}$$

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:

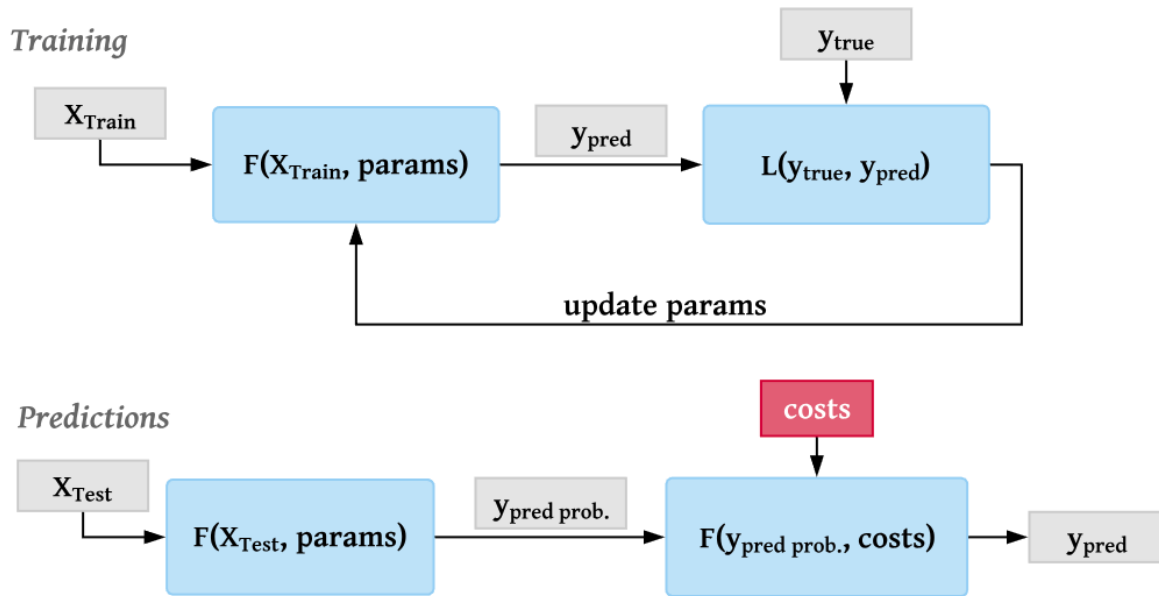
$$\sum_{i=1}^n l(y_i, y_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + \text{constant}.$$

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

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



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

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

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

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

gdzie

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

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

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

gdzie

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

Klasyfikacja danej obserwacji jest opisana następującą nierównością:

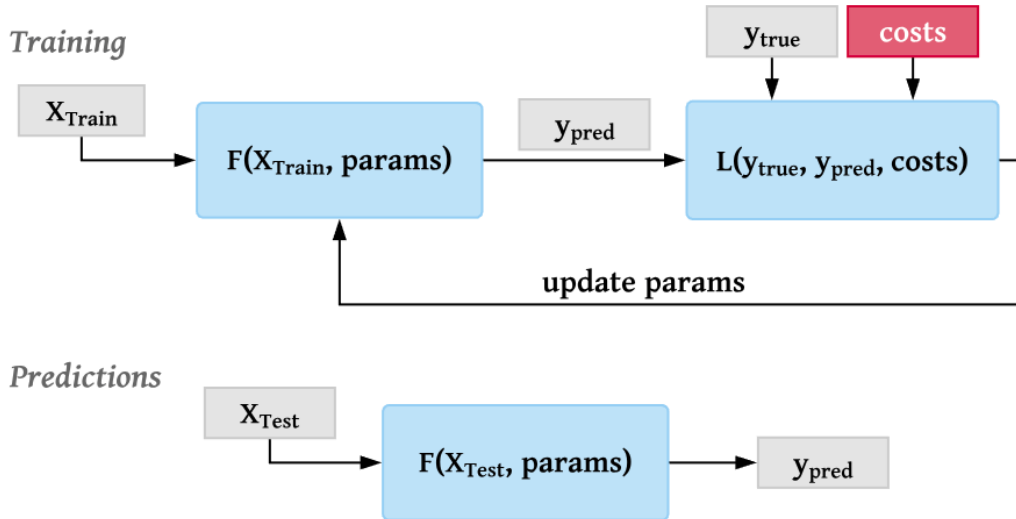
$$R(p_1|\mathbf{x}_i) \leq R(p_0|\mathbf{x}_i)$$

I oznacza ona, że klasyfikujemy dany przykład jako pozytywny, jeżeli ryzyko takiej decyzji jest mniejsze niż zaklasyfikowania danej obserwacji jako negatywną. Po przeprowadzaniu odpowiednich przekształceń algebraicznych otrzymujemy następujący wzór na klasyfikację przykładu jako pozytywny:

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

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

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



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

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

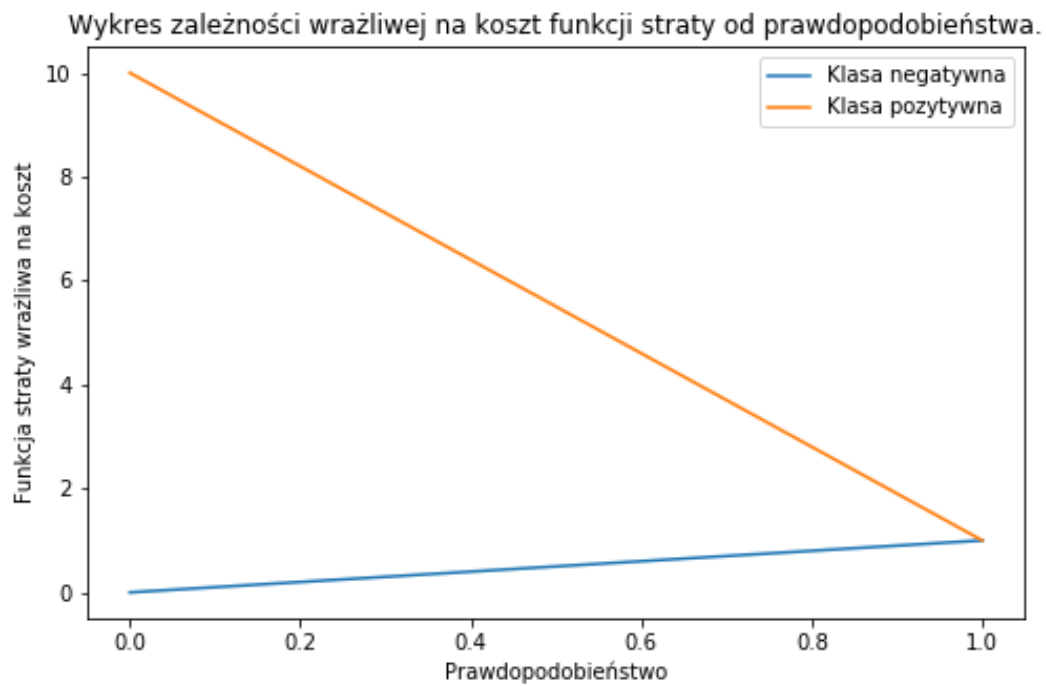
Pierwszym modelem, który stosunkowo łatwo przystosować do bycia wrażliwym na koszt jest regresja logistyczna. Kontynuując rozważania z podrozdziału 1.2.1 chcemy, aby funkcja straty przyjmowała następujące wartości, które odpowiadają wartościom z macierzy kosztu:

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

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

$$J^c(\theta) = \frac{1}{N} \sum_{i=1}^N \left(y_i \left(h_\theta(\mathbf{x}_i) C_{TP_i} + (1 - h_\theta(\mathbf{x}_i)) C_{FN_i} \right) + (1 - y_i) \left(h_\theta(\mathbf{x}_i) C_{FP_i} + (1 - h_\theta(\mathbf{x}_i)) C_{TN_i} \right) \right)$$

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



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: $C_{TP} = 1$, $C_{FN} = 10$, $C_{FP} = 1$, $C_{TN} = 0$.

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:

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

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

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

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 Detection¹ zawierający 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 ?? możemy domyślać się z jakimi zmiennymi mieliśmy do czynienia przed transformacjami. Podczas procesu dokonywania transakcji standardowo zbierane są: ID klienta, data dokonania transakcji, kwota, lokalizacja, typ transakcji (przykładowo płatność internetowa, płatność stacjonarna, wypłata z bankomatu), beneficjent transakcji (przykładowo linie lotnicze, hotel, wypożyczalnia samochodów itp.), historyczna informacja dot. czy dana transakcja była oszustwem, wiek klienta, kraj zamieszkania, kod pocztowy, typ karty (przykładowo VISA, MasterCard itp.). 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.

¹Źródło: <https://www.kaggle.com/mlg-ulb/creditcardfraud>

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 inwestycji oraz strat z tego wynikających.

| Prawda | Predykcja | |
|--------|-----------|----------|
| | Oszustwo | Normalna |
| | Oszustwo | Normalna |
| | Oszustwo | C_a |
| | Normalna | Amt_i |
| | | 0 |

Tabela 2.1: Macierz kosztu dla 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 predykcji 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 tę 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.

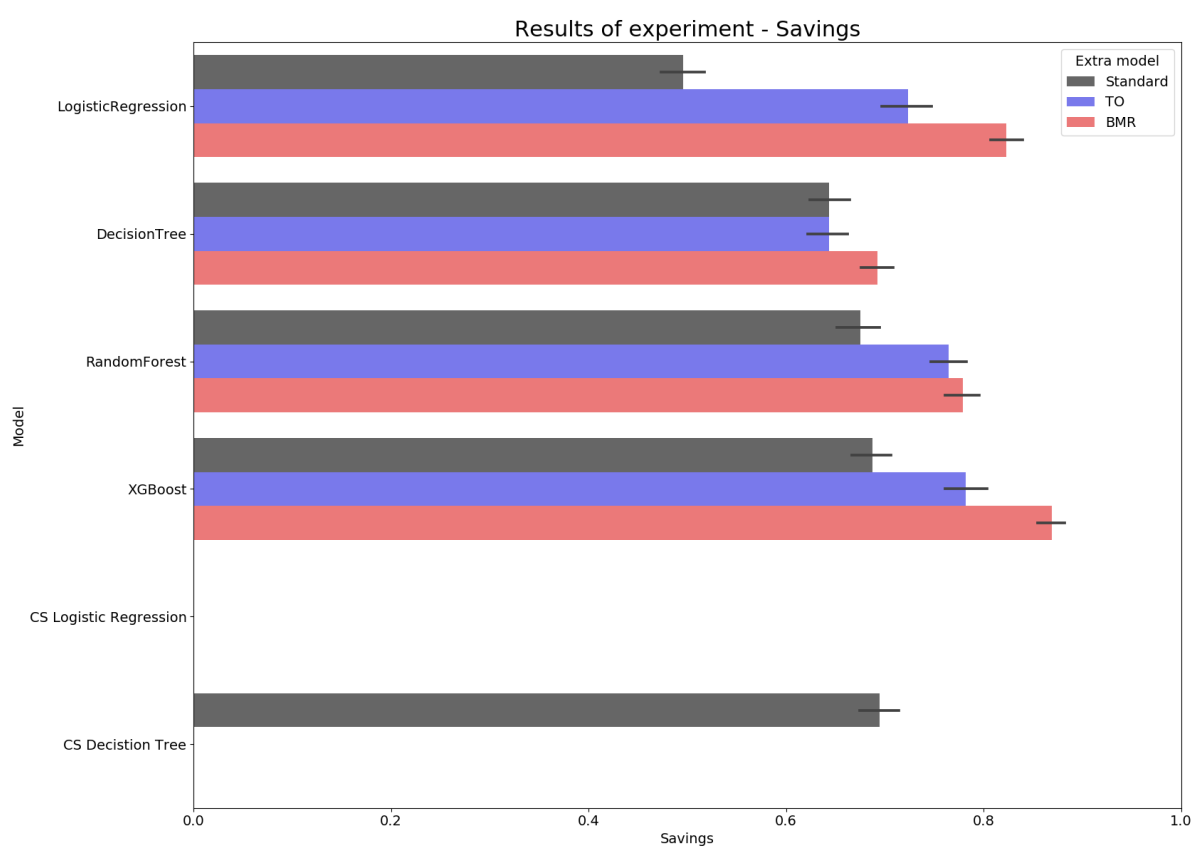
Do implementacji skryptów został wykorzystany język programowania Python wraz z bibliotekami costela, sklearn, pandas, numpy, matplotlib oraz język programowania bash.

2.3 Wyniki

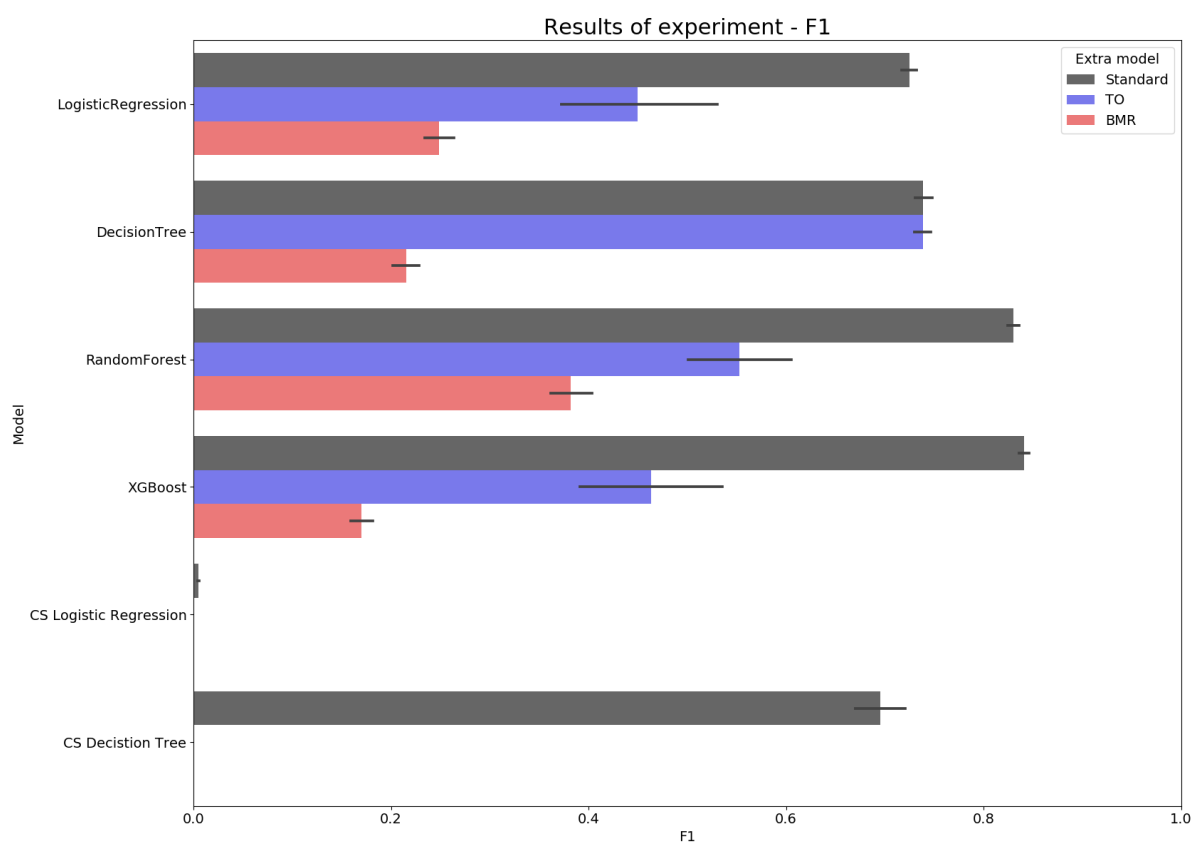
Lorem ipsum ...

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

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



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



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] 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.
- [7] 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.
- [8] 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.