Opracowanie aplikacji przewidującej wartość rynkową nieruchomości w Polsce na podstawie modelu predykcyjnego

Mateusz Jurek, Arkadiusz Sałaj, Dominik Gajda, Jeremi Korowajski, Konrad Kruczek

dr inż. Paweł Drzymała

Politechnika Łódzka, WEEIA, Informatyka, Łódź, Polska

STRESZCZENIE

W artykule przedstawiono proces budowy modelu predykcyjnego na potrzeby przewidywania wartości rynkowej nieruchomości na obszarze Polski. Celem projektu jest utworzenie nowoczesnej aplikacji internetowej oferującej intuicyjne narzędzie z szerokim wachlarzem parametrów, umożliwiających użytkownikowi systemu generowanie precyzyjnych wyników. Na podstawie dostępnych cech modelu, zaproponowano rozwiązanie polegające na zbudowaniu trzech różnych modeli regresji o różnej liczbie parametrów wejściowych, w tym jeden automatycznie uzupełniający brakujące wartości o medianę danego atrybutu obliczonej w oparciu w zestaw danych treningowych. Wytypowano wskaźniki oceny modeli regresji, które najdokładniej obrazują wynikowe modele dla zbioru danych jakim dysponowano. Wytrenowano oraz przetestowano popularne modele regresji, z których wytypowano las losowy oraz XGBoosting jako modele uczenia zespołowego, które osiągały najniższe wartości błędu oraz najwyższe wartości współczynnika determinacji, świadczącego o zadowalających prognozach systemu. Modele zostały odpowiednio dostrojone metodą kroswalidacji, a następnie połączone kolejną warstwą uczenia zespołowego w celu utworzenia finalnego, najdokładniejszego modelu. Eksperymenty wykazały bardzo dokładne prognozy dla miast i obszarów o niewielkim względnym rozrzucie ceny mieszkań.

KEYWORDS drzewo decyzyjne, las losowy, mieszkania, nieruchomości, predykcja, regresja, rynek nieruchomości, uczenie maszynowe, uczenie zespołowe

I. WSTĘP

A. Obszar działań

Zagadnienie koncentruje się wokół polskiego rynku nieruchomości, zakupu i najmu mieszkań oraz zabudowań mieszkalnych. Obejmuje zarówno operacje na nieruchomościach dokonywane na rynku wtórnym jak i rynku pierwotnym.

B. Cel Projektu

Potrzebą realizacji projektu jest utworzenie systemu prognozującego wartość nieruchomości, znajdujących się w największych miastach na terenie Polski, w stopniu bardzo dobrym, zgodnie z definicją zawartą w wybranych wskaźnikach oceny jakości modelu.

II. STAN WIEDZY

Dane GUS z lat 2021-2022 obrazują, że ceny mieszkań na polskim rynku nieruchomości rosną szybciej niż przeciętne ceny w gospodarce (Zaremba, 2023). W roku 2023 średnia cena transakcyjna za m2 w Warszawie przekroczyła 13 tys. złotych (Narodowy Bank Polski [NBP], 2023). Dynamikę rynku może również obrazować znacząca zmiana stopy procentowej kredytu między rokiem 2022 a 2023, spadek kredytowej dostępności mieszkania badanej dla siedmiu największych miast oraz wzrost marż bankowych przy kredytach mieszkaniowych na przełomie lat 2022 i 2023 (NBP, 2023).

Zestawienia wskaźników rynkowych minionej dekady wyraźnie obrazują wysoką zmienność w okresie ostatnich dwóch lat, uzasadniać może wysokie zapotrzebowanie na zarówno aktualne dane cen mieszkań, jak i modele predykcji wartości rynkowej nieruchomości na tych danych utworzone.

Problem budowy modelu predykcji wartości nominalnej danego dobra na podstawie jego współrzędnych geograficznych oraz dodatkowych cech opisujących to dobro, sprowadza się najczęściej do utworzenia modelu regresji (Truong et al., 2019; Yang & Cao, 2018; Wu, 2020). Spośród testowanych modeli, najdokładniejsze rezultaty przynoszą: Gradient Boosting Decision Tree (GBDT), Las losowy w zadaniu regresji, Extreme Gradieng Boosting (XGB), Light Gradient Boosting Machine (LGBM) oraz wielorakie kombinacje modeli zespołowych, zbudowanych z wcześniej wymienionych.

A. Konstruktywna analiza istniejących rozwiązań

Wyłonienie najdokładniejszego modelu dla ustalonego zestawu danych, jest w wielu pracach naukowych uzależnione od zbadania wyniku jednego wskaźnika – MSE (Yang & Cao, 2018), RMSLE (Truong et al., 2019). Wynik ewaluacji może być interpretowany wielorako, w zależności od dziedziny nauki, wskaźnika liczby wykorzystanych regresorów oraz posiadanego zbioru danych (SAS/STAT, 2015, s101). W celu kompleksowej oceny modelu warto w konsekwencji wykorzystać wiele wskaźników równolegle, które można ze sobą zestawić i zwiększyć wiarygodność wyników.

B. Uzasadnienie i formy rozwiązania

W celu odnalezienia najdokładniejszego modelu dla opracowywanego problemu, istotne jest posłużenie się wieloma modelami regresji oraz dostrojenie ich hiperparametrów, bazując na wybranym wskaźniku oceny modelu, a następnie zbadanie ich dokładności przy użyciu więcej niż jednego wskaźnika. Wyniki modelu osiągną wówczas większą wiarygodność.

W tym celu wykorzystany zostanie wskaźnik błędu *RMSLE* (Root Mean Squared Log Error), który, ze względu na operację logarytmowania, jest przydatny dla nieznormalizowanych danych i mogą mieć kilka rzędów wielkości rozrzutu.

Kolejnym wykorzystanym wskaźnikiem będzie współczynnik determinacji - *R2* (R-Square), interpretowany jako część odpowiedzi, która może być wyjaśniona przez prognozę regresora. Staje się przydatnym wskaźnikiem w momencie, gdy reprezentacja populacji, służąca do badań, jest pobierana losowo, co jesteśmy w stanie zagwarantować przy użyciu odpowiednich operacji podziału zbioru danych.

Ze względu na mnogość cech, jeden model może być niewystarczający do oferowanego użytkownikom systemu predykcyjnego. Powstaną zatem trzy różne modele różniące się liczbą cech wejściowych, w tym jeden automatycznie uzupełniający liczbę cech wejściowych o medianę ze zbioru treningowego, jeżeli użytkownik systemu nie ma wystarczającej liczby informacji, by samemu określić jej wartość.

III. OPIS ROZWIĄZANIA

**A. Opis techniczny rozwiązania**

Utworzona aplikacja składa się z części klienckiej oraz serwerowej. Część serwerowa ma dostęp do bazy danych zrejestrowanych użytkowników oraz do utworzonego modelu predykcyjnego.

Proces tworzenia modelu predykcyjnego rozpoczął się od kolekcjonowania i analizy danych wyceny mieszkań w piętnastu największych miastach Polski (Jamroz, 2024). Zbiór danych zawiera od 15 tysięcy do 21 tysięcy próbek w zależności od badanego miesiąca. Do trenowania i testowania modelu zostały wykorzystane dane z roku 2024 (od stycznia do maja), co, po usunięciu duplikatów, dało w wyniku ponad 48 tysięcy próbek.

Zebrana kolekcja została odpowiednio oczyszczona poprzez usunięcie kolumn, których wartość *null* stanowiła więcej niż 5% zbioru. Następnie dokonano analizy wartości dla każdej z cech, wizualizacji lokalizacji mieszkań dla każdego z miast oraz obliczeń współczynnika korelacji między parami cech w celu wyłonienia najistotniejszych atrybutów do utworzenia najmniejszego (pod względem liczby cech) modelu.

Modele uczenia maszynowego zostały utworzone w środowisku Jupyter Notebook w oparciu o bibliotekę scikit-learn.

Najmniejszy model został utworzony na podstawie czterech cech: powierzchni mieszkania, szerokości geograficznej, długości geograficznej i nazwy miasta. Nazwa miasta na potrzeby uczenia modelu została zakodowania gorąco-jedynkowo (one-hot), rozszerzając zbiór danych o liczbę kolumn równej liczbie miast.

Drugi z kolei model do predykcji wymaga dodatkowych trzech cech: liczba pokoi, liczba pięter budynku oraz odległość od centrum.

Ostatni model składa się z piętnastu cech – wymienionych wcześniej siedmiu powiększonych o: odległość od uczelni, odległość od szkoły, odległość od przedszkola, odległość od apteki, odległość od szpitala, odległość od restauracji oraz liczba punktów zainteresowania w pobliżu (points of interest).

Do oceny modelu zostały wybrane dwa wskaźniki: *RMSLE* oraz *R2* (współczynnik determinacji). Celem było osiągnięcie wartości wskaźnika *RMSLE* jak najbliższej zeru oraz wskaźnika *R2* jak najbliższej jedynki. Modele zostały wytrenowane przy użyciu 80% dysponowanych próbek zbioru, a przetestowane przy wykorzystaniu pozostałych 20% próbek zbioru.

Przy ich użyciu przetestowano dokładność następujących modeli dostępnych w bibliotece scikit-learn w zadaniu regresji: SVM, K-Mean, Drzewo decyzyjne, Gradient Boosting, Las losowy oraz XGBososting. Hiperparametry każdego z modeli zostały dostrojone korzystając z metody przeszukiwania siatki względem ich parametrów domyślnych. Satysfakcjonujące wyniki zostały uzyskane jedynie przy użyciu metod uczenia zespołowego – XGBoosting oraz Las losowy, gdzie współczynnik determinacji przekroczył wartość 0.8, natomiast błąd RMSLE nie przekraczał wartości 0.003.

Tabela I zawiera zestawienie wartości wskaźników oceny wytrenowanych modeli.

TABELA I

Zestawienie dokładności modeli regresorów w zależności od liczby cech wejściowych

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Liczba cech wejściowych | Las losowy | XGBoosting | Model zespołowy | |
| 4 | R2 = 0.892  RMSLE = 0.021 | R2 = 0.872  RMSLE = 0.026 | R2 = 0.899  RMSLE = 0.019 | |
| 7 | R2 = 0.894  RMSLE = 0.020 | R2 = 0.900  RMSLE = 0.019 | R2 = 0.902  RMSLE = 0.018 | |
| 15 | R2 =0.899  RMSLE = 0.019 | R2 = 0.897  RMSLE = 0.020 | R2 = 0.901  RMSLE = 0.019 | |
|  |  | |  |

Dla siedmiu cech wejściowych model XGBoosting osiągał lepsze rezultaty w przypadku obu wskaźników, jednak w pozostałych przypadkach model Lasu losowego cechował się większą dokładnością. W konsekwencji dla każdego wariantu liczby cech wejściowych został utworzony model uczenia zespołowego złożony z dwóch wytrenowanych modeli – Lasu losowego i XGBoosting. Odpowiedź modelu wyłaniana jest na podstawie średniej arytmetycznej z predykcji każdego elementu zespołu. Wykorzystana metoda pozwoliła osiągnąć model cechujący się najbardziej wiarygodnymi prognozami.

Wytrenowane modele zostały wyeksportowane do plików PMML możliwych w następnym kroku do wykorzystania w środowisku Java utworzonego serwera. Do modeli zawierające największą liczbę cech została dodana dodatkowa warstwa preprocessingu, która uzupełnia brakujące wartości wejściowe przekazywane do modelu o medianę danej cechy obliczoną na podstawie danych treningowych.

Serwer został utworzony w języku Java przy wykorzystaniu platformy Spring. Obsługuje on rejestrację użytkownika, autentyfikację, wymianę danych przy użyciu standardu JSON Web Token, serwisowanie historii użytkownika, komunikację z bazą danych PostreSQL oraz API oferujące obliczanie prognoz cen mieszkania na podstawie wprowadzonych parametrów wejściowych dla trzech modeli predykcji.

Utworzony monolit został skonteneryzowany i uruchomiony na maszynie oferując swobodny dostęp do narzędzia.

Wykorzystana baza danych zawiera informację o zarejestrowanym użytkowniku, historii wyszukiwania oraz przypisanych do niego tokenach. *Rysunek 1* przedstawia diagram zaprojektowanej bazy danych.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, numer

Opis wygenerowany automatycznie

Rysunek 1.  Uproszczony diagram bazy danych utworzonej na potrzeby przechowywania informacji o zarejestrowanym użytkowniku, autentyfikacji i jego historii wyszukiwania.

Prototyp strony klienta został zaprojektowany w oparciu o sztuczną inteligencję (rys. 2). Zabieg ten umożliwił wygenerowanie wielu projektów w krótkim czasie oraz wyłonienie najlepszego stanowiącego fundament do dalszego budowania interfejsu, co jest zgodne z metodologią Design-Build.

Interfejs koncentruje się na nowoczesnym designie, wysokiej czytelności, rozpoznawalności marki oraz wysokiej roli kolorów w odbiorze produktu. Wysoki nacisk postawiono na identyfikację wizualną na przykładzie projektowania logo produktu (rys. 3). Prototyp został przetestowany z udziałem reprezentantów docelowej grupy odbiorców – użytkowników aplikacji internetowych śledzących ceny nieruchomości.

Implementacji strony dokonano przy wykorzystaniu technologii Vue.js rozszerzonej o zestaw komponentów z platformy Vuetify. Wyodrębniono sekcję logowania i rejestracji użytkownika, utworzono interaktywny widget selekcji lokalizacji, zamieszczono statystyki najczęściej wyszukiwanych parametrów oraz obsłużono, przy wykorzystaniu utworzonego API, modele predykcji cen mieszkań.

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Rysunek 2.  Makieta strony klienckiej aplikacji utworzona przy użyciu programu Figma.

Obraz zawierający Grafika, symbol, Czcionka, krąg

Opis wygenerowany automatycznie

Rysunek 3. Zaprojektowane propozycje logo utworzonej aplikacji internetowej.

**B. Zarządzanie projektem i pracą zespołową**

Zespół zaadoptował przyrostowy model wytwarzania oprogramowania, realizując funkcjonalności przypisane do ról członków w tygodniowych cyklach. Każdy cykl kończył się Minutką, w ramach której zawarte były informacje opisujące dotychczasowy progres, zrealizowane zadania oraz plan działania na kolejny cykl.

Kod źródłowy projektu był konsekwentnie wersjonowany w środowisku Git i udostępniany w repozytorium GitHub (zał. 1). Każdy moduł posiadał swoją osobną gałąź, nad którą pracowali członkowie zespołu z przypisanymi odpowiednimi rolami.

IV. BADANIA/TESTY

Testy modelu przeprowadzane były już na etapie projektowania przed eksportowaniem do pliku PMML. Dokonano pomiaru wybranych wskaźników na podstawie danych testowych stanowiących 20% populacji zbioru i porównano je między różnymi modelami dążąc do osiągnięcia jak najlepszych rezultatów, które przedstawione zostały w *Tabeli I*.

Sporządzono wykresy wizualizujące różnicę wartości prognozy od wartości rzeczywistej (rys. 4) oraz macierz pomyłek dla 8 kwantyli (rys. 5).

Przeprowadzono również testy wyników największego modelu dla niepełnego wektora cech. Wiarygodność prognoz malała wraz ze spadkiem liczby wprowadzonych parametrów, czyli ze wzrostem liczby cech, które uzupełniane były automatycznie na podstawie mediany wartości cech zbioru treningowego.

Testy warstwy serwerowej przeprowadzane były przy użyciu platformy Postman. Weryfikowana została odpowiedź wszystkich endpointów oraz przekazywanie odpowiedzi modelu.

Obraz zawierający zrzut ekranu, diagram, linia

Opis wygenerowany automatycznie

Rysunek 4.  Wykres błędu względem wartości rzeczywistej dla nawiększego pod względem liczby cech modelu zespołowego.

Obraz zawierający tekst, zrzut ekranu, Prostokąt, numer

Opis wygenerowany automatycznie

Rysunek 5.  Macierz pomyłek utworzona dla największego pod względem liczby cech modelu zespołowego.

Zrealizowano testy integracyjne systemu, zweryfikowano czy wyświetlane informacje w odpowiednich sekcjach aplikacji odpowiadają danym pobranym przy użyciu prywatnego API. Przetestowano rejestrację użytkowników porównując dane wprowadzane przez użytkownika z informacjami wprowadzanymi do bazy. Zweryfikowano reakcję systemu na wprowadzanie błędnych danych, takich jak ujemna powierzchnia mieszkania, nieistniejący adres e-mail czy nieobsługiwana lokalizacja.

V. WYNIKI I ANALIZA

Zamieszczone w Tabeli I wyniki wskaźników oceny modelu świadczą, że dokonywane prognozy mają wysoką wiarygodność, co odczytać można na podstawie współczynnika determinacji przekraczającego wartość 0.9. Dla rynku nieruchomości charakteryzującego się wysoką zmiennością w badanym okresie, model w satysfakcjonujący sposób jest w stanie przewidzieć wartość mieszkania.

Aplikacja została zbudowana z myślą o wysokiej użyteczności, udostępniając ekskluzywne funkcjonalności na tle konkurencji. Zarówno umożliwia szybkie wyszukiwanie wyników dla minimalnej liczby wprowadzonych parametrów, jak i kompleksową analizę dla wielu zmiennych.

System zawiera benefity dla zarejestrowanych użytkowników, co zapewnia zarówno zbliżyć użytkowników do aplikacji oraz zaplanować plan subskrypcji w celu odblokowania kolejnych funkcjonalności.

VI. WNIOSKI I PERSPEKTYWY ROZWOJU

Rezultatem projektu jest spójny, ale również skalowalny i otwarty na rozwój system predykcji wartości rynkowej nieruchomości. W dużym stopniu aplikacja jest zależna od dostępu do aktualnych i szczegółowych danych, które w obecnym okresie zawężały się do obszaru Polski, jednak korzystając z wypracowanego systemu obsługi, w krótkim czasie można rozszerzyć zasięg ofert również o sąsiadujące kraje lub cały rynek europejski.

Proces aktualizacji danych nie jest zautomatyzowany, co może stanowić kierunek rozwoju projektu.

Zważając na wzrost zainteresowania aplikacją oraz rozszerzenie jej działania również na inne państwa, należy również dokonać odpowiednich zabezpieczeń i testów wydajnościowych na dużym obciążeniu serwera.

ZAŁĄCZNIKI

Zał. 1 <https://github.com/Nerekk/Projekt-kompetencyjny-2024>

PODZIĘKOWANIA

Twórcy artykułu pragną serdecznie podziękować dyrektorom, profesorom, doktorom i pracownikom dydaktycznym instytutów IMSI, IIS, DMCS oraz kadrze akademickiej wydziału Elektrotechniki Elektroniki Informatyki i Automatyki Politechniki Łódzkiej przyczyniającej się do przygotowania toku kształcenia studentów na kierunku informatyki.

BIBLIOGRAFIA

Truong, Q., Nguyen, M., Dang, H., & Mei, B. (2019). *Housing Price Prediction via Improved Machine Learning Techniques*. Elsevier.​

Yang, B., & Cao, B. (2018). *Research on Ensemble Learning based Housing Price Prediction Model. Big Geospatial Data and Data Science*, 1, 1-8. Clausius Scientific Press, Canada.​

Zixu Wu (2020). *Prediction of California House Price Based on Multiple Linear Regression*. Academic Journal of Engineering and Technology Science, ISSN 2616-5767 Vol.3, Issue 7: 11-15​

Lin, D., Wachter, S. (2019) . *The Effect of Land Use Regulation on Housing Prices: Theory and Evidence from California*. Wharton University of Pennsylvania.​

Cootes, T. F., Ionita, M. C., Lindner, C., & Sauer, P. (2012). *Robust and accurate shape model fitting using random forest regression voting.* In Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12 (pp. 278-291). Springer Berlin Heidelberg.

Sharma, A. K., Li, L. H., & Ahmad, R. (2022, November). *Default Risk Prediction Using Random Forest and XGBoosting Classifier*. In 2021 International Conference on Security and Information Technologies with AI, Internet Computing and Big-data Applications (pp. 91-101). Cham: Springer International Publishing.

Rafał Zbyrowski (2022). *Przyczynowość cen na rynku nieruchomości w Polsce*. Katedra Metod Ilościowych, Wydział Zarządzania, Uniwersytet Warszawski. Metody ilościowe w badaniach ekonomicznych, Tom XXIII/3, 2022, s. 67 – 77

SAS/STAT® 14.1 User’s Guide (2015). *Introduction to Regression Procedures*, SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414, s101

Jamroz, K. (2024). Kaggle, *Apartment Prices in Poland*, [www.kaggle.com/datasets/krzysztofjamroz/apartment-prices-in-poland/data](https://www.kaggle.com/datasets/krzysztofjamroz/apartment-prices-in-poland/data)

Webb, P., et al. (2013). *Spring Boot Reference Guide*

Nelson, B. (2018). *Getting to Know Vue.js,* Apress, ISBN-13 (pbk): 978-1-4842-3780-9

Stonebraker, M., Rowe, L. A., & Hirohama, M. (1990). *The implementation of POSTGRES*. IEEE transactions on knowledge and data engineering, 2(1), 125-142.

WKŁAD W PRACĘ NAD PROJEKTEM

Dr inż. Paweł Drzymała, Opiekun projektu, pracownik naukowy instytutu IMSI. Twórca ponad osiemdziesięciu wielokrotnie cytowanych publikacji naukowych w latach 1998-2024. Poza szerokim dorobkiem naukowym na rzecz informatyki, elektrotechniki i mechatroniki, prowadzi również wieloletnią działalność dydaktyczną na Politechnice Łódzkiej. Zapewnił zespołowi ekspercką wiedzę z zakresu projektowania systemów bazadanowych, transmisji danych w architekturze klient-serwer oraz bezpieczeństwa aplikacji webowych.

Mateusz Jurek, Student 6. semestru kierunku Informatyka. Odpowiadał za projektowanie architektury serwera, implementację serwera aplikacji, projektowanie bazy danych, łączenie bazy danych z silnikiem systemu, komunikację z warstwą kliencką, opracowywanie scenariuszy testów, konteneryzację aplikacji oraz metodę wdrażania systemu. Pełnił rolę koordynatora zespołu, specjalisty z dziedzin projektowania aplikacji internetowych, projektowania serwisów REST, konteneryzacji aplikacji i projektowania baz danych PostgreSQL Zapewnił zespołowi kompetencje z zakresu technologii i narzędzi Spring Boot, Java, JWT, Swagger, Postman, Docker.

Arkadiusz Sałaj, Student 6. semestru kierunku Informatyka. Odpowiadał za gromadzenie i analizę danych, procesowanie zbioru danych, projektowanie architektury serwera, implementację serwera aplikacji, komunikowanie serwera aplikacji z wytworzonym modelem, zarządzanie repozytorium projektu, implementację komunikacji z klientem aplikacji, testy komunikacji oraz łączenie bazy danych z silnikiem systemu. Pełnij rolę specjalisty z dziedzin projektowania aplikacji internetowych, projektowania serwisów REST, projektowania baz danych i analizy danych. Zapewnił zespołowi kompetencje z zakresu technologii i narzędzi Spring Boot, Java, JWT, Docker.

Jeremi Korowajski, Student 6. semestru kierunku Informatyka. Odpowiadał za warstwę kliencką aplikacji, implementację strony www, projektowanie interfejsu użytkownika, komunikację klienta z serwerem aplikacji, zabezpieczenia transmisji danych, konfigurację transmisji oraz autentyfikację użytkownika. Pełnił rolę specjalisty z dziedzin projektowania aplikacji klienckich, projektowania interfejsów użytkownika, user experience, transmisji danych klient-serwer. Zapewnił zespołowi kompetencje z zakresu technologii i narzędzi Spring Boot, Java, JavaScript, Vue.js, Vuetify, JWT.

Konrad Kruczek, Student 6. semestru kierunku Informatyka. Odpowiadał za warstwę kliencką aplikacji, implementację strony www, prototypowanie interfejsu użytkownika, proces projektowania i testowania interfejsu, estetykę i kreowanie marki projektu oraz zapewnienie obsługi aplikacji przez urządzenia mobilne. Pełnił rolę specjalisty z dziedzin projektowania aplikacji klienckich, projektowania interfejsów użytkownika, user experience, metodyki projektowania stron i prototypowania. Zapewnił zespołowi kompetencję z zakresu technologii i narzędzi JavaScript, Vue.js, Vuetify, Figma oraz wykorzystania AI w procesie tworzenia interfejsów.

Dominik Gajda, Student 6. semestru kierunku Informatyka. Odpowiadał za selekcję danych, gromadzenie zbiorów danych, analizę statystyczną, projektowanie procesu oczyszczania i przygotowywania danych, selekcję miar oceny modelu uczenia maszynowego, trenowanie i testowanie modeli regresji, wizualizację wyników i eksport modeli do zasobów serwera. Pełnił rolę specjalisty z zakresu analizy danych, uczenia maszynowego i komunikacji modelu z aplikacją. Zapewnił zespołowi kompetencję z zakresu technologii i narzędzi Jupyter Notebook, Python, scikit-learn, PMML.