

Design Proposal

Zespół 10

Julia Czyżewska, Szymon Ryszka, Michał Krasnodębski

Aplikacja analizująca wpływ infrastruktury miejskiej na ceny nieruchomości

Celem projektu jest stworzenie aplikacji internetowej, która wykorzystuje dane geolokalizacyjne i rynkowe do oceny, jak infrastruktura miejska wpływa na ceny mieszkań. Pozwala przewidywać wartość nieruchomości na podstawie lokalizacji oraz wizualizować przestrzenne zależności dla całego miasta.

Planowana funkcjonalność programu

Analiza lokalna

- Użytkownik może wprowadzić lokalizację poprzez:
 - wpisanie adresu w polu wyszukiwania, lub
 - wskazanie punktu na mapie (pinezka interaktywna).
- Po zaakceptowaniu lokalizacji aplikacja:
 - pobiera dane dotyczące elementów infrastruktury miejskiej: teatry, atrakcje turystyczne, placówki oświatowe, krzewy, drzewa, przystanki, bankomaty, apteki, stacje rowerowe, akademiki, hotele, komisariaty policji, urządzenia AED
 - oblicza przewidywaną cenę za metr kwadratowy,
 - scrapuje dane i oblicza rzeczywistą, uśrednioną cenę ofertową z dostępnych danych rynkowych,
 - wyświetla oba wyniki
- Dodatkowo użytkownik może zobaczyć krótkie podsumowanie czynników, które miały największy wpływ na wyliczoną prognozę

Analiza globalna

- Aplikacja prezentuje dane zagregowane dla całej Warszawy
 - wykresy i wizualizacje pokazujące wpływ poszczególnych elementów infrastruktury na ceny,

- możliwość filtrowania lub przełączania między różnymi typami infrastruktury, aby zobaczyć ich indywidualny wpływ.

- Analiza nie wymaga żadnego inputu od użytkownika.

Planowany stack technologiczny

Backend: FastAPI, Pydantic, HTTPX, BeautifulSoup4, geopy

Frontend: Dash (Plotly), Dash Leaflet, Plotly, pydeck (heatmap), HTTPX, Playwright (e2e)

Baza danych: PostgreSQL z rozszerzeniem PostGIS

Modele predykcyjne: sklearn, xgboost, shap, pygam

Źródła danych: Warszawski Portal Otwartych Danych (API), Otodom (oferty mieszkań) ,
Nominatim (geokodowanie)

Środowisko: Docker

Planowany zakres eksperymentów

TODO

- Porównanie modeli pod względem dokładności predykcji i wyjaśnialności wpływu poszczególnych cech na wynik końcowy. Zaplanowane do porównania modele: linear regression, random forest regressor, XGB regressor, generalized additive model.
- Porównanie dokładności modeli dla różnych promieni, w obrębie których mogą znajdować się obiekty infrastruktury wpływające na cenę.
- Walidacja poprawności wizualizacji danych na mapie i wykresach (czy wszystkie punkty i wartości są spójne z wynikami API).

Uzasadnienie wybranych modeli

- Model regresji liniowej - prosty model, może być wyznacznikiem dokładności dla pozostałych, łatwo interpretowalny liniowy wpływ cech
- Model random forest regressor - wprowadza nieliniowość, waga cech poprzez *feature importance* lub SHAP
- Model xgboost regressor - często wykazuje lepszą dokładność, interpretowalny dzięki SHAP
- Generative additive model - pokazuje wynik jako wpływ poszczególnych cech, ale może być nieliniowy,

Planowany harmonogram pracy

TODO

Tydzień 27.10 – 02.11

- Stworzenie ogólnej struktury folderów projektu
- Konfiguracja środowiska Python - instalacja i test podstawowych bibliotek
- Wstępna konfiguracja konteneryzacji
- Wstępna konfiguracja bazy danych PostgreSQL + PostGIS
- Inicjalizacja aplikacji frontendowej (Dash)
- Implementacja interaktywnej mapy (Dash Leaflet) z możliwością wskazania punktu lub wyszukania adresu
- Przygotowanie podstawowego layoutu (formularz, mapa, placeholder wyników)

Tydzień 03.11 – 09.11

- Implementacja pobierania danych z API Urzędu m.st. Warszawy – infrastruktura miejska i przystanki
- Stworzenie zbioru (lub sposobu tworzenia zbioru w czasie rzeczywistym) treningowego i testowego wykorzystując dane z API Urzędu m.st. Warszawy
- Stworzenie pierwszego, prostszego modelu regresji liniowej, który do uczenia będzie wykorzystywał dane API, a jako przewidywana cena wykorzystana będzie uśredniona dla każdej dzielnicy / osiedla
- Dodanie obsługi interakcji użytkownika w mapie (wybór punktu, adresu)
- Wstępna integracja frontendu z lokalnymi mockami API (testy UI bez backendu)

Prototyp aplikacji: 07.11**Tydzień 10.11 – 16.11**

- Implementacja scrapowania ofert mieszkań z Otodom (cena, metraż, adres)
- Obsługa błędów przy scrapowaniu
- Dodanie pierwszych wykresów (Plotly) – porównanie przewidywanej i rzeczywistej ceny m²

Tydzień 17.11 – 23.11

- Implementacja modułu geokodowania adresów z Otodom
- Implementacja obliczania średnich cen mieszkań w danym obszarze
- Stworzenie zbioru w oparciu o scrapowane dane - zamiast uśrednionej wartości dla dzielnic

Tydzień 24.11 – 30.11

- Stworzenie podstawowego FastAPI do udostępniania danych dla modelu regresyjnego
- Testy przesyłania danych do modelu ML i odbierania odpowiedzi
- Wstępna dokumentacja endpointów API
- Trening i dobór parametrów modelu regresji liniowej

- Integracja frontendu z prawdziwym API FastAPI (Analiza lokalna)
- Testy poprawności danych i wizualizacji (spójność punktów, skala mapy)

Tydzień 01.12 – 07.12

- Integracja API FastAPI z frontem (przesyłanie zapytań i odbiór wyników predykcji)
- Testowanie pełnego przepływu danych: model ↔ API ↔ frontend
- Trening i dobór parametrów modelu random forest regressor

Tydzień 08.12 – 14.12

- Zapewnienie mechanizmu tworzenia logów
- Przygotowanie pełnego środowiska Docker do uruchomienia backendu + bazy danych.
- Trening i dobór parametrów modelu xgb regressor
- Testy e2e frontendu (Playwright) – ścieżka Analiza lokalna i Analiza globalna

Tydzień 15.12 – 21.12

- Dokumentacja końcowa modułów backendowych
- Trening i dobór parametrów modelu generalized additive model
- Dokumentacja komponentów frontendu (layouty, callbacki, interakcje)

Tydzień 22.12 – 28.12 - święta

- Przerwa

Tydzień 29.12 – 04.01

- Testy wydajnościowe backendu (czas odpowiedzi API, przetwarzanie batch danych).
- Wybór najlepszego z modeli
- Testy wydajności frontendu (czas ładowania wyników)
- Poprawki UX/UI po testach użytkowników (czytelność mapy, opisy, etykiety)

Tydzień 05.01 – 11.01

- Finalne poprawki, przygotowanie prezentacji i nagranie działania

Bibliografia

- dokumentacja poszczególnych grup danych z <https://api.um.warszawa.pl/>
- <https://postgis.net/documentation/>
- <https://geopy.readthedocs.io/>
- <https://nominatim.org/release-docs/latest/>

- <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- <https://docs.pydantic.dev/latest/>
- <https://www.python-httpx.org/>
- **Draper, N. R., & Smith, H. (1998).** *Applied Regression Analysis*
- **Breiman, L. (1996).** *Bagging predictors. Machine Learning*
- **Chen, T., & Guestrin, C. (2016).** *XGBoost: A Scalable Tree Boosting System*
- **Hastie, T., & Tibshirani, R. (1986).** *Generalized Additive Models*