**Instytut Informatyki  
Wydział Nauk Ścisłych i Technicznych  
Uniwersytet Rzeszowski**

**Przedmiot:**

**Hurtownie danych**

**Dokumentacja projektu:**

**HomeValue-Analysis**

**Wykonali: Mateusz Bocak i Gabriela Bieniek**

**Prowadzący: mgr inż. Adam Szczur**

**Rzeszów 2025**

Spis treści

[1. Temat i cel projektu 3](#_Toc198701894)

[2. Techniczne aspekty projektu 3](#_Toc198701895)

[2.1. Funkcjonalności aplikacji 3](#_Toc198701896)

[2.2. Wykorzystane technologie 4](#_Toc198701897)

[2.3. Projekt GUI 6](#_Toc198701898)

[Główny panel wczytywania i przygotowania danych 6](#_Toc198701899)

[Zakładka statystyk 7](#_Toc198701900)

[Zakładka wizualizacji 8](#_Toc198701901)

[Panel uczenia maszynowego 9](#_Toc198701902)

[Zakładka przetwarzania danych 10](#_Toc198701903)

[3. Wygląd i użytkowanie aplikacji 11](#_Toc198701904)

[3.1. Wymagania do uruchomienia aplikacji 11](#_Toc198701905)

[Wymagania sprzętowe 11](#_Toc198701906)

[Wymagania programowe 11](#_Toc198701907)

[Instalacja 11](#_Toc198701908)

[Wymagania dotyczące danych wejściowych 11](#_Toc198701909)

[3.2. Obsługa aplikacji 12](#_Toc198701910)

[Wczytanie i przygotowanie danych 12](#_Toc198701911)

[Ekstrakcja podtablic 12](#_Toc198701912)

[Analiza statystyczna 12](#_Toc198701913)

[Wizualizacja danych 12](#_Toc198701914)

[Machine Learning 13](#_Toc198701915)

[Edycja danych 13](#_Toc198701916)

[4. Eksperymenty na danych 14](#_Toc198701917)

[4.1. Wykorzystane zbiory danych 14](#_Toc198701918)

[4.3. Przebieg eksperymentów i wyniki 18](#_Toc198701919)

[Eksperyment – pierwszy zbiór danych 18](#_Toc198701920)

[Eksperyment – drugi zbiór danych 19](#_Toc198701921)

[4.4. Analiza uzyskanych wyników i wnioski 21](#_Toc198701922)

[Pierwszy zbiór danych 21](#_Toc198701923)

[Drugi zbiór danych 22](#_Toc198701924)

[5. Literatura 23](#_Toc198701925)

# Temat i cel projektu

HomeValue-Analytics to aplikacja desktopowa stworzona w języku Python, służąca do kompleksowej analizy danych tabelarycznych ze szczególnym uwzględnieniem danych dotyczących rynku nieruchomości. Aplikacja wykorzystuje nowoczesne biblioteki do analizy danych (pandas), wizualizacji (plotly) oraz uczenia maszynowego (scikit-learn), udostępniając je poprzez intuicyjny interfejs użytkownika stworzony przy pomocy frameworka Streamlit i pyWebView, na komputerach stacjonarnych.

Celem projektu jest stworzenie narzędzia analitycznego umożliwiającego użytkownikom bez zaawansowanej wiedzy programistycznej przeprowadzanie złożonych analiz danych poprzez:

* Wczytywanie i wstępne przetwarzanie danych (czyszczenie, transformacja)
* Generowanie podstawowych statystyk opisowych
* Tworzenie interaktywnych wizualizacji danych
* Przeprowadzanie analizy metodami uczenia maszynowego

Aplikacja pozwala na szybką analizę zbiorów danych w formacie CSV, automatyzując procesy przygotowania danych, ich wizualizacji oraz modelowania predykcyjnego, co znacząco przyspiesza proces analizy danych i wyciągania z nich wartościowych wniosków.

# Techniczne aspekty projektu

## Funkcjonalności aplikacji

Aplikacja zawiera wiele podstawowych czynności do manipulacji i wyświetlania danych w przejrzysty i czytelny sposób. Poniżej zostały wymienione kluczowe funkcjonalności:

1. Operacje na danych wejściowych

* Import danych z plików CSV
* Podgląd i edycja danych w interaktywnej tabeli
* Wykrywanie typów kolumn (numeryczne, kategoryczne, daty)

1. Analiza statystyczna

* Miary położenia: średnia, mediana, moda, min, max

1. Manipulacja danymi

* Ekstrakcja podtablic przez:
  + Wybór kolumn z listy
  + Definiowanie zakresów wierszy (np. 1:10, 1,5,9)
  + Edycja wartości bezpośrednio w tabeli
  + Usuwanie duplikatów
  + Usuwanie wierszy z brakującymi wartościami
* Standaryzacja danych numerycznych (średnia=0, odch.std=1)
* Kodowanie zmiennych kategorycznych (binarnie, one-hot encoding, ordynalnie) w zależności od wartości kategorii

1. Wizualizacja danych

* Wykres rozrzutu z linią trendu
* Histogram rozkładu zmiennej
* Wykres słupkowy
* Wykres kołowy
* Mapa ciepła korelacji
* Wykres częstości kategorii

1. Machine Learning

* Klasyfikacja z użyciem Random Forest:
  + **Parametry**: liczba drzew, głębokość, min. próbki
  + **Walidacja**: train-test split
  + **Metryki**: macierz pomyłek, dokładność, raport klasyfikacji
  + Analiza ważności cech

1. Interfejs użytkownika

* Podział na zakładki funkcjonalne
* Interaktywne formularze konfiguracji
* Wizualizacja wyników w czasie rzeczywistym
* Obsługa błędów i komunikaty statusu

## Wykorzystane technologie

**Język i środowisko**

* Python 3.11 – główny język programowania używany w projekcie. Wersja 3.11 oferuje zwiększoną wydajność oraz nowe funkcjonalności wspierające rozwój aplikacji analitycznych i uczenia maszynowego
* Visual Studio Code – edytor kodu zapewniający wsparcie dla Pythona, debugowanie, integrację z Gitem i liczne rozszerzenia ułatwiające programowanie
* Git/GitHub – system kontroli wersji i platforma do współdzielenia kodu. Umożliwiają śledzenie zmian w projekcie, pracę zespołową oraz publikację kodu w repozytorium zdalnym

**Interfejs**

* Streamlit - framework do tworzenia webowych aplikacji webowych dla analiz danych i modeli ML. Umożliwia szybkie przekształcenie skryptu Pythona w interaktywną aplikację dostępną w przeglądarce

**Biblioteki analityczne**

* pandas - manipulacja i analiza danych. Umożliwia wczytywanie, filtrowanie, grupowanie oraz przekształcanie danych w wygodny i czytelny sposób dzięki strukturze DataFrame
* NumPy – wsparcie dla operacji numerycznych i pracy na dużych tablicach wielowymiarowych. Wykorzystywany jako podstawa do wielu operacji matematycznych w analizie danych
* scikit-learn – biblioteka do uczenia maszynowego, wykorzystywana m.in. do klasyfikacji danych. W projekcie zastosowano klasyfikator RandomForestClassifier oraz moduły preprocessing do przygotowania danych wejściowych

**Wizualizacja**

* Plotly Express – umożliwia tworzenie interaktywnych wykresów, które można dynamicznie eksplorować w aplikacji Streamlit
* Matplotlib/Seaborn – biblioteka do tworzenia statycznych wykresów. Seaborn łatwia budowanie atrakcyjnych wizualnie wykresów opartych na danych statystycznych. Matplotlib zapewnia pełną kontrolę nad elementami graficznymi

## Projekt GUI

### Główny panel wczytywania i przygotowania danych

A screenshot of a computer

AI-generated content may be incorrect.

* Uploader plików CSV
* Opcje czyszczenia danych (usuwanie duplikatów i brakujących wartości)
* Wybór transformacji danych (standaryzacja lub kodowanie binarne)
* Podgląd wczytanych danych w formie edytowalnej tabeli
* Informacje o wykonanych operacjach

### Zakładka statystyk

A screenshot of a computer

AI-generated content may be incorrect.

* Statystyki opisowe – analiza kolumn numerycznych porzez obliczenie podstawowych statystyk
* Analiza kategoryczna
* Analiza korelacji – narzędzia do badania związków między zmiennymi z wyborem korelacji (Pearsona, Kendalla i Spearmana)

### Zakładka wizualizacji

A screenshot of a computer

AI-generated content may be incorrect.

* Wizualizacja danch – oferuje różne typy wykresów do graficznej prezentacji danych (histogram, wykres słupkowy, mapa ciepła korelacji, wykres kołowy, wykres części kategorii, wykres porównawczy kategorii)
* Opcje konfiguracji wykresu rozrzutu:
  + Wybór zmiennych na osiach X i Y
  + Opcja kolorowania punktów według kategorii
* Dodatkowe wykresy (Violin Plot, Pair Plot)

### Panel uczenia maszynowego

A screenshot of a computer

AI-generated content may be incorrect.  
A screenshot of a computer

AI-generated content may be incorrect.

* Wybór zmiennych do klasyfikacji
* Konfigurację parametrów modelu Random Forest
* Ustawienia walidacji (podział train/test, cross-validation)
* Wyniki trenowania:
  + Macierz pomyłek
  + Wykres ważności cech
  + Metryki skuteczności modelu

### Zakładka przetwarzania danych

A screenshot of a computer

AI-generated content may be incorrect.

* Kodowanie kolumn symbolicznych (metody One-Hot, Target)
* Zamiana wartości
* Tabela danych z podglądem danych po wykonanym przetwarzaniu danych

# Wygląd i użytkowanie aplikacji

## Wymagania do uruchomienia aplikacji

### Wymagania sprzętowe

* Procesor: minimum 2 rdzenie, zalecane 4 rdzenie
* RAM: minimum 4GB, zalecane 8GB
* Dysk: minimum 500MB wolnego miejsca
* System operacyjny: Windows 10/11, macOS 10.14+, lub Linux z GUI

### Wymagania programowe

* Python 3.11 lub nowszy
* Biblioteki Python:
  + streamlit>=1.29.0
  + pandas>=2.1.0
  + numpy>=1.24.0
  + scikit-learn>=1.3.0
  + plotly>=5.18.0
  + matplotlib>=3.7.0
  + seaborn>=0.13.0
  + i inne wymienione w pliku requirements.txt

### Instalacja

1. Zainstaluj Python 3.11 ze strony python.org
2. Sklonuj repozytorium: *git clone*[*https://github.com/JustFiesta/HomeValue-Analytics.git*](https://github.com/JustFiesta/HomeValue-Analytics.git)
3. Wejdź do folderu z repozytorium i terminal
4. Uruchom setup odpowiedni dla systemu operacyjnego: *setup.bat* / *setup.sh* w terminalu

### Wymagania dotyczące danych wejściowych

* Format pliku: CSV
* Kodowanie: UTF-8
* Separator: przecinek (,)
* Pierwsza linia: nagłówki kolumn
* Minimalna liczba wierszy: 10

## Obsługa aplikacji

### Wczytanie i przygotowanie danych

1. Uruchom aplikację i wczytaj plik CSV używając przycisku "Wybierz plik CSV"
2. W sekcji "Przygotowanie danych" możesz:
   * Usunąć powtarzające się wiersze
   * Usunąć wiersze z brakującymi wartościami
   * Wybrać transformację danych:
     + Brak transformacji
     + Standaryzacja danych numerycznych
     + Kodowanie binarne kolumn kategorycznych

### Ekstrakcja podtablic

1. Wybierz kolumny z listy rozwijanej
2. Wprowadź zakresy wierszy używając składni:
   * 1:10 - wiersze od 1 do 10
   * 1,5,9 - konkretne wiersze
   * -10: - ostatnie 10 wierszy
3. Kliknij "Wyodrębnij podtablicę" aby zastosować zmiany
4. Użyj "Resetuj" aby wrócić do oryginalnych danych

### Analiza statystyczna

1. Przejdź do zakładki "Statystyki"
2. Zobacz podstawowe statystyki dla zmiennych:
   * Minimum, maksimum
   * Średnia, mediana
   * Odchylenie standardowe
   * Kwartyle
3. Przeanalizuj macierz korelacji między zmiennymi

### Wizualizacja danych

1. Przejdź do zakładki "Wizualizacje"
2. Wybierz typ wykresu:
   * Wykres rozrzutu
   * Histogram
   * Wykres słupkowy
   * Mapa ciepła korelacji
   * Wykres kołowy
   * Wykres częstości kategorii
3. Skonfiguruj parametry wykresu
4. Analizuj interaktywne wizualizacje

### Machine Learning

1. Przejdź do zakładki "Machine Learning"
2. Wybierz zmienne:
   * Cechy numeryczne
   * Cechy kategoryczne
   * Zmienną do przewidywania
3. Skonfiguruj model Random Forest:
   * Liczba drzew
   * Głębokość drzew
   * Parametry podziału
4. Ustaw parametry walidacji:
   * Rozmiar zbioru testowego
   * Liczba foldów CV
5. Trenuj model i analizuj wyniki:
   * Metryki skuteczności
   * Macierz pomyłek
   * Ważność cech

### Edycja danych

1. Kliknij w komórkę w podglądzie danych, aby ją edytować
2. Wprowadź nową wartość
3. Zmiany są automatycznie zapisywane w pamięci
4. Format liczb jest dostosowany do typu danych:
   * Liczby całkowite bez separatorów
   * Liczby zmiennoprzecinkowe z 2 miejscami po przecinku

# Eksperymenty na danych

## Wykorzystane zbiory danych

**Zbiór pierwszy:**

Wykorzystano zbiór danych dotyczący rynku nieruchomości w Polsce, podzielony na 11 pomniejszych plików CSV za dany miesiąc. Dataset został utworzony przez Krzysztofa Jamroza i udostępniony na platformie Kaggle.

Dane zostały zebrane prawdopodobnie metodą web scrapingu z polskich serwerów ogłoszeniowych dotyczących nieruchomości. Zawierają informacje o cenach mieszkań w Polsce wraz z różnymi cechami tych nieruchomości.

Dataset dotyczy rynku nieruchomości w Polsce – cen mieszkań i czynników wpływających na ich wartość. Dane można wykorzystać do:

* analizy predykcyjnej - tworzenie modeli prognozujących ceny mieszkań na podstawie różnych cech (np. lokalizacja, powierzchnia, liczba pokoi)
* analizy regresji – badanie zależności między ceną a innymi zmiennymi
* analizy eksploracyjnej - wizualizacja rozkładów cen w różnych miastach/regionach Polski
* segmentacji rynku - grupowanie podobnych mieszkań z wykorzystaniem algorytmów klastrujących
* wykrywania anomalii - dentyfikacja nietypowo wycenionych mieszkań

Techniki analizy mogą obejmować:

* statystyki opisowe
* wizualizacje (wykresy rozrzutu, histogramy, mapy cieplne korelacji)
* modele regresji (liniowa, drzewa decyzyjne, lasy losowe)
* techniki uczenia nienadzorowanego (K-means, hierarchiczna analiza skupień)
* redukcja wymiarowości (PCA, t-SNE) dla lepszej wizualizacji danych wielowymiarowych

Podział plików w zbiorze danych:

* apartments\_pl\_YYYY\_MM.csv - miesięczny przegląd ofert sprzedaży
* apartments\_rent\_pl\_YYYY\_MM.csv - miesięczny przegląd ofert wynajmu

Projekt używa plików o nazewnictwie „apartments\_pl\_YYYY\_MM.csv”, dotyczących miesięcznego przeglądu ofert sprzedaży.

* Liczba rekordów: (w pojedyńczym pliku)
  + apartments\_pl\_2023\_08: 18905
  + apartments\_pl\_2023\_09: 16997
  + apartments\_pl\_2023\_10: 16690
  + apartments\_pl\_2023\_11: 16302
  + apartments\_pl\_2023\_12: 16483
  + apartments\_pl\_2024\_01: 15521
  + apartments\_pl\_2024\_02: 16361
  + apartments\_pl\_2024\_03: 17318
  + apartments\_pl\_2024\_04: 19259
  + apartments\_pl\_2024\_05: 20231
  + apartments\_pl\_2024\_06: 21501
* Liczba cech: 27
* Występują brakujące wartości cech
* Cechy numeryczne:
  + powierzchnia
  + liczba pokoi
  + liczba pięter w mieszkaniu
  + liczba pięter w budynku
  + rok budowy
  + szerokość geograficzna
  + długość geograficzna
  + dystans do centrum
  + liczba punktów użyteczności publicznej w promieniu 500m od apartamentu
  + dystans do szkoły
  + dystans do klinik
  + dystans do punktu poczty
  + dystans do przedszkola
  + dystans do restauracji
  + dystans do uczelni
  + dystans do aptek
  + cena
* Cechy kategoryczne:
  + miasto (15 wartości: Warszawa, Kraków, Wrocław, Łódź, Szczecin, Lublin, Radom, Bydgoszcz, Poznań, Katowice, Gdańsk, Białystok, Rzeszów, Gdynia, Częstochowa)
  + typ nieruchomości (3 wartości: blok mieszkalny, budynek apartamentowy, kamienica)
  + stan własności (2 wartości: spółdzielnia, kondominium)
  + materiał budowlany (2 wartości: cegła, płyta betonowa)
  + stan (2 wartości: premium, niski)
  + miejsce parkingowe (2 wartości: tak, nie)
  + balkon (2 wartości: tak, nie)
  + winda (2 wartości: tak, nie)
  + ochrona (2 wartości: tak, nie)
  + pomieszczenie do przechowywania (2 wartości: tak, nie)
* Okres: 08.2023-06.2024

Źródło: <https://www.kaggle.com/datasets/krzysztofjamroz/apartment-prices-in-poland>

**Zbiór drugi:**

Wykorzystano zbiór danych dotyczący rynku nieruchomości w Melbourne. Zbiór danych zawiera dwa pliki CSV: „MEBLOURNE\_HOUSE\_PRICES\_LESS” oraz „Melbourne\_housing\_FULL”.

Dataset został utworzony przez Anthony’ego Pino i udostępniony na platformie Kaggle.

Dane zostały zebrane prawdopodobnie z australijskich serwisów sprzedaży nieruchomości lub oficjalnych rejestrów transakcji nieruchomości w Melbourne. Zbiór danych zawiera szczegółowe informacje o domach sprzedanych na rynku nieruchomości w Melbourne.

Dataset koncentruje się na rynku mieszkaniowym w Melbourne w Australii. Dane mogą być wykorzystane do:

* modelowania predykcyjnego – prognozowanie cen nieruchomości na podstawie cech takich jak lokalizacja, powierzchnia, liczba pokoi
* analizy geograficznej – badanie wpływu lokalizacji na ceny nieruchomości
* analizy trendów czasowych – analiza zmiany cen w czasie
* segmentacji rynku – identyfikacja różnych segmentów rynku nieruchomości
* analizy wpływu czynników – badanie, które cechy mają największy wpływ na cenę nieruchomości

Techniki analizy mogą obejmować:

* analiza korelacji między zmiennymi
* wizualizacje geolokalizacyjne (mapy cieplne cen w różnych obszarach)
* modele uczenia maszynowego (regresja, drzewa decyzyjne, sieci neuronowe)
* analiza szeregów czasowych
* analiza ważności zmienych
* walidacja krzyżowa modeli predykcyjnych

Projekt używa pliku „MEBLOURNE\_HOUSE\_PRICES\_LESS”.

* Liczba rekordów: 63023
* Liczba cech: 13
* Nie występują brakujące wartości
* Cechy numeryczne:
  + ilość pokoi
  + cena
  + kod pocztowy
  + liczba nieruchomości znajdujących się na terenie dzielnicy
  + odległość od CBD
* Cechy kategoryczne:
  + data sprzedaży (112 wartości: 24/03/2018, 28/10/2017, 17/03/2018 itp.)
  + dzielnica (380 wartości: Reservoir, Bentleigh East, Mount Waverley, Richmond itp.)
  + adres (57754 wartości: 14 Northcote St, 5 Charles St, 14 Moray St, 52 Station St itp.)
  + typ nieruchomości (3 wartości: h- dom, domek letniskowy, willa, bliźniak, szeregowiec; u - jednostka, dupleks; t - dom szeregowy)
  + metoda sprzedaży (9 wartości: S - nieruchomość sprzedana; SP - nieruchomość sprzedana wcześniej; PI - nieruchomość przekazana; PN - sprzedana wcześniej, nieujawniona; SN - sprzedana, nieujawniona; NB - brak oferty; VB - oferta sprzedawcy; W - wycofana przed aukcją; SA - sprzedana po aukcji; SS - sprzedana po aukcji, nieujawniona cena; N/A - cena lub najwyższa oferta niedostępna
  + nazwa sprzedawcy (476 wartości: Barry, Jellis, Nelson itp.)
  + nazwa regionu (8 wartości: Southern Metropolitan, Northern Metropolitan, Western Metropolitan, Eastern Metropolitan, South-Eastern Metropolitan, Eastern Victoria, Northern Victoria, Western Victoria)
  + rada zarządzająca obszarem (34 wartości: Boroondara City Council, Darebin City Council, Banyule City Council itp.)
* Okres: od 2016 do 2018 rok (dotyczy całego okresu)

Źródło: [https://www.kaggle.com/datasets/anthonypino/melbourne-housing-market](https://www.kaggle.com/datasets/anthonypino/melbourne-housing-market?resource=download)

## Przebieg eksperymentów i wyniki

### Eksperyment – pierwszy zbiór danych

Użyty plik do eksperymentu: „apartments\_pl\_2023\_08”.

Przeprowadzono eksperyment klasyfikacji miast na podstawie pozostałych cech nieruchomości:

1. Przygotowanie danych:

* Usunięcie rekordów z brakującymi wartościami
* Standaryzacja cech numerycznych

1. Parametry modelu Random Forest:

* Liczba drzew: 20
* Maksymalna głębokość: 10
* Podział danych: 0.5 test, 0.5 analiza
* Liczba foldów walidacji krzyżowej: 5
* Leave-One-Out: tak

1. Wyniki klasyfikacji:

A screenshot of a graph

AI-generated content may be incorrect.

A screen shot of a video game

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

### Eksperyment – drugi zbiór danych

Użyty plik do eksperymentu: „MELBOURNE\_HOUSE\_PRICES\_LESS”

Przeprowadzono eksperyment klasyfikacji nazwy regionu na podstawie pozostałych cech nieruchomości:

1. Przygotowanie danych:

* Usunięcie rekordów z brakującymi wartościami
* Standaryzacja cech numerycznych

1. Parametry modelu Random Forest:

* Liczba drzew: 20
* Maksymalna głębokość: 10
* Podział danych: 0.5 test, 0.5 analiza
* Liczba foldów walidacji krzyżowej: 2
* Leave-One-Out: nie
* Usunięcie atrybutu Adress

1. Wyniki klasyfikacji:

A screenshot of a computer

AI-generated content may be incorrect.

A black screen with a black background

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

## Analiza uzyskanych wyników i wnioski

### Pierwszy zbiór danych

Model osiąga bardzo dobre wyniki ogólne:

* Średnia dokładność CV: 0.960 (+0.011)
* Średnia dokładność LOO: 0.963
* Dokładność na zbiorze testowym: 0.970
* Ogólna dokładność: 0.9697

Analiza macierzy pomyłek:

1. Silne pozytywne wyniki dla większości miast – liczby na przekątnej macierzy wskazują na poprawne klasyfikacje
2. Najlepiej klasyfikowane miasta to Kraków (165 próbek), Warszawa (276) i Wrocław (77)
3. Pomyłki klasyfikacji:
   1. Częstochowa często mylona z innymi miastami (brak prawidłowych klasyfikacji)
   2. Pewne pomyłki między Łodzią a Lublinem (73/1 i 42/0)
   3. Radom bywa mylony z Poznaniem (19/0)

Znaczenie cech w modelu:

1. longitude (długość geograficzna) – zdecydowanie najważniejsza
2. price (cena) – druga pod względem ważności
3. clinicDistance (odległość od kliniki)
4. kindergartenDistance (odległość od przedszkola)
5. buildYear (rok budowy)

Cechy o najmniejszym znaczeniu:

1. hasSecurity\_yes (posiadanie ochrony)
2. type\_tenement (typ budynku)
3. condition\_premium (stan mieszkania – premium)

Wnioski:

* Bardzo wysoka skuteczność ogólna – model osiąga ponad 96% dokładności we wszystkich metrykach, co wskazuje na bardzo dobrą jakość klasyfikacji
* Lokalizacja jako kluczowy czynnik – długość geograficzna jest zdecydowanie najważniejszą cechą, co sugeruje, że położenie wschód-zachód ma kluczowe znaczenie dla rozróżniania miast w Polsce
* Problemy z małymi próbkami – miasta z małą liczbą przykładów (Częstochowa, Gdynia) mają znacznie gorsze wyniki klasyfikacji
* Ceny jako ważny czynnik – druga najważniejsza cecha to cena, co potwierdza zróżnicowanie rynków nieruchomości między miastami
* Infrastruktura jako ważny wyróżnik – odległość od placówek (kliniki, przedszkola, szkoły) są istotnymi cechami rozróżniającymi miasta
* Perspektywy ulepszenia – model mógłby skorzystać z większej liczby próbek dla niektórych miast (Częstochowa, Gdynia) oraz potencjalnie z dodatkowych cech uwzględniających specyfikę lokalnych rynków nieruchomości,

### Drugi zbiór danych

Model osiąga dobre wyniki:

* Średnia dokładność CV: 0.910 (+0.004)
* Średnia dokładność LOO: Wyłączone (brak zasobów na przeprowadzenie testu klasyfikatora z LOO)
* Dokładność na zbiorze testowym: 0.906
* Ogólna dokładność: 0.9063

Analiza macierzy pomyłek:

1. Zróżnicowana skuteczność klasyfikacji – niektóre regiony klasyfikowane bardzo dobrze, inne znacznie gorzej
2. Najlepiej klasyfikowane regiony z wysokimi liczbami poprawnych klasyfikacji:
   1. Northen Metropolitan (5814/0)
   2. Southern Metropolitan (6240/0)
   3. Western Metropolitan (4671/0)
3. Kluczowe błędy klasyfikacji:
   1. Eastern Victoria błędnie klasyfikowana jako inne regiony (brak poprawnych klasyfikacji)
   2. Northern Victoria ma niski wskaźnik poprawnych klasyfikacji (24/173)
   3. Eastern Metropolitan mylona z Northern Metropolitan (604/0)
   4. South-Eastern Metropolitan mylona z Eastern Metropolitan (1471/0)
   5. Western Victoria błędnie klasyfikowana jako inne regiony (brak poprawnych klasyfikacji)

Znaczenie cech w modelu:

1. Postcode – zdecydowanie najważniejsza cecha
2. Suburb\_Mill Park – druga pod względem ważności
3. Suburb\_Mordialloc
4. Seller – kilka różnych sprzedawców ma znaczący wpływ

Wnioski:

* Ogólna skuteczność – model osiąga około 91% dokładności, co jest dobrym wynikiem (ale niższy niż w eksperymencie na pierwszym datasecie)
* Nierównomierna skuteczność – występuje duża dysproporcja w skuteczności klasyfikacji między regionami:
  + Duże regiony metropolitalne (Northern, Southern, Western Metropolitan) mają wysokie wskaźniki klasyfikacji
  + Mniejsze regiony (Eastern Victoria, Western Victoria) mają bardzo słabe wyniki
* Problem próbek – regiony z małą liczbą próbek (poniżej 200) mają znacznie gorsze wyniki klasyfikacji
* Lokalizacja jako kluczowy czynnik – kod pocztowy jest zdecydowanie najważniejszą cechą, co potwierdza, że lokalizacja geograficzna jest kluczowym elementem rozróżniającym regiony
* Wpływ sprzedawców – różni sprzedawcy mają różny wpływ na klasyfikację, co sugeruje, że mogą działać głównie w określonych regionach
* Wpływ dat – daty transakcji mają znaczenie dla klasyfikacji, co może wskazywać na sezonowe trendy lub zmieniające się w czasie wzorce sprzedaży w różnych regionach
* Obszary do poprawy:
  + Zwiększenie liczby próbek dla Eastern Victoria i Western Victoria
  + Rozważenie dodatkowych cech dla lepszego rozróżniania regionów o niższych wskaźnikach
  + Potencjalne połączenie podobnych regionów o małej liczbie próbek
  + Zbadanie przyczyn mylenia Eastern Metropolitan z Northern Metropolitan i innych częstych błędów

# Literatura

* 1. **Random Forest**
* Dokumentacja sklearn.ensemble.RandomForestClassifier
* [Random Forest Algorithm](https://scikit-learn.org/stable/modules/ensemble.html" \l "forest)
  1. **Preprocessing danych**
* sklearn.preprocessing
* [Feature Scaling](https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html)
  1. **Metryki oceny**
* [Classification Metrics](https://scikit-learn.org/stable/modules/model_evaluation.html" \l "classification-metrics)