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

**Przedmiot:**

**Hurtownie danych**

**Dokumentacja projektu:**

***HomeValue-Analysis***

**Wykonał: Mateusz Bocak**

**Prowadzący: mgr inż. Adam Szczur**

**Rzeszów 2025**

Spis treści

[Temat i cel projektu 3](#_Toc195205109)

[Techniczne aspekty projektu 3](#_Toc195205110)

[Funkcjonalności aplikacji 3](#_Toc195205111)

[Wykorzystane technologie 4](#_Toc195205112)

[Projekt GUI 5](#_Toc195205113)

[Główny panel wczytywania i przygotowania danych 5](#_Toc195205114)

[Zakładka wizualizacji 5](#_Toc195205115)

[Panel uczenia maszynowego 6](#_Toc195205116)

[Użytkowanie aplikacji 7](#_Toc195205117)

[Wymagania do uruchomienia aplikacji 7](#_Toc195205118)

[Wymagania sprzętowe 7](#_Toc195205119)

[Wymagania programowe 7](#_Toc195205120)

[Instalacja 7](#_Toc195205121)

[Wymagania dotyczące danych wejściowych 7](#_Toc195205122)

[Obsługa aplikacji 8](#_Toc195205123)

[Wczytanie i przygotowanie danych 8](#_Toc195205124)

[Ekstrakcja podtablic 8](#_Toc195205125)

[Analiza statystyczna 8](#_Toc195205126)

[Wizualizacja danych 8](#_Toc195205127)

[Machine Learning 9](#_Toc195205128)

[Edycja danych 9](#_Toc195205129)

[Eksperymenty na danych 10](#_Toc195205130)

[Wykorzystane zbiory danych 10](#_Toc195205131)

[Przebieg eksperymentu i wyniki 10](#_Toc195205132)

[Analiza wyników i wnioski 11](#_Toc195205133)

[Literatura 12](#_Toc195205134)

# 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
* Visual Studio Code
* Git/GitHub

**Interfejs**

* Streamlit - framework do tworzenia webowych aplikacji analitycznych

**Biblioteki analityczne**

* pandas - manipulacja danymi
* NumPy - operacje numeryczne
* scikit-learn - uczenie maszynowe (RandomForestClassifier, preprocessing)

**Wizualizacja**

* Plotly Express - interaktywne wykresy
* Matplotlib/Seaborn - wykresy statyczne

## Projekt GUI

### Główny panel wczytywania i przygotowania danych

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

* 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 wizualizacji

Obraz zawierający zrzut ekranu, tekst, Oprogramowanie multimedialne, Oprogramowanie graficzne

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

* Dropdown z wyborem typu wykresu
* Opcje konfiguracji wykresu rozrzutu:
  + Wybór zmiennych na osiach X i Y
  + Opcja kolorowania punktów według kategorii
* Wygenerowany interaktywny wykres z linią trendu
* Informacje o korelacji między zmiennymi

### Panel uczenia maszynowego

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.  
Obraz zawierający zrzut ekranu, Oprogramowanie multimedialne, Oprogramowanie graficzne, tekst

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

* 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

# 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

Wykorzystano zbiór danych dotyczący rynku nieruchomości w Polsce, podzielony na pomniejsze CSV za dany rok/miesiąc:

* Liczba rekordów: 1001-8000 w jednym pliku
* Liczba cech: 7
* Cechy numeryczne: cena, powierzchnia, rok budowy
* Cechy kategoryczne: typ nieruchomości, miasto, stan własności
* Okres: 2023 rok

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

## Przebieg eksperymentu i wyniki

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: 10
* Maksymalna głębokość: 10
* Podział danych: 0.4 test, 0.6 analiza

1. Wyniki klasyfikacji:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Obraz zawierający tekst, zrzut ekranu, Oprogramowanie multimedialne, oprogramowanie

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.Obraz zawierający tekst, zrzut ekranu, menu

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

## Analiza wyników i wnioski

Największy wpływ na klasyfikację mają:

* **centreDistance** (odległość od centrum),
* **hasElevator** (posiadanie windy),
* **buildYear** (rok budowy),
* **latitude** (szerokość geograficzna),

Cechy związane z lokalizacją, co sugeruje, że nieruchomości w różnych miastach różnią się głównie pod względem geograficznym. Cechy takie jak  *ownership, condition* czy *floorCount* mają marginalne znaczenie.

1. **Słabe punkty modelu**:
   * **Tylko 2 miasta (Kraków, Warszawa) klasyfikowane poprawnie** (F1-score: 0.58 i 0.71), pozostałe mają zerowe metryki.
   * **Accuracy = 49%** – zawyżone przez dominujące klasy (Warszawa i Kraków: 49% danych).
   * **Macro avg F1 = 0.086** – potwierdza katastrofalną jakość dla mniejszych miast.
2. **Główne problemy**:
   * **Niezbalansowane dane**: Warszawa i Kraków dominują, małe miasta (np. Częstochowa: 1 próbka) są ignorowane.
   * **Niedopasowane parametry**: 10 drzew i maks. głębokość 10 to za mało dla wieloklasowej klasyfikacji.
3. **Rekomendacje**:
   * **Balansuj dane**: Oversampling dla mniejszych klas lub zbierz więcej próbek.
   * **Ulepsz model**: Zwiększ liczbę drzew (np. 50-100), testuj głębsze drzewa

**Podsumowanie**: Model działa jak „detektor Warszawy i Krakowa”. Jest to wynikiem niezbalansowanych danych w datasecie. Aby klasyfikować wszystkie miasta, kluczowe jest rozwiązanie problemu balansu danych i optymalizacja parametrów.

# Literatura

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