

**AKADEMIA HUMANISTYCZNO EKONOMICZNA
ŁÓDŹ**

SZTUCZNA INTELIGENCJA - projekt

ADRIAN LEŚNIAK

Nr indeksu: 154256

Wydział Techniki i Informatyki

Grupa: D1 Informatyka

Semestr 7, 2024/2025 r.

Spis treści

1. WSTĘP.....	3
2. REALIZACJA PROJEKTU.....	4
2.1 Wybór, załadowanie i przygotowanie danych.....	4
2.2 Eksploracja danych.....	5
2.3 Budowa modelu.....	7
2.4 Trenowanie modelu.....	8
2.5 Ocena skuteczności modelu i analiza wyników.....	10
3. WNIOSKI.....	13

1. WSTĘP

Opis wybranego problemu

Ceny nieruchomości są istotnym wskaźnikiem ekonomicznym, podlegającym dynamicznym zmianom. Z tego powodu stworzono projekt, którego przedmiotem jest budowa sztucznej sieci neuronowej (SNN) do przewidywania cen nieruchomości w Polsce, w oparciu o takie cechy jak lokalizacja (szerokość i długość geograficzna), liczba pokoi, powierzchnia, piętro oraz rok budowy. Wybrany problem jest zadaniem regresji, w którym chcemy przewidzieć wartość ciągłą (cenę) na podstawie zbioru cech.

Celem projektu jest stworzenie modelu ułatwiającego proces decyzyjny dla:

- Inwestorów, którzy chcą ocenić opłacalność inwestycji.
- Deweloperów, którzy planują nowe projekty.
- Osób prywatnych, podejmujących decyzje o zakupie lub sprzedaży nieruchomości.
- Instytucji finansowych, oceniających ryzyko kredytowe.

Do realizacji projektu wykorzystano zbiór danych z platformy Kaggle

("(www.kaggle.com/datasets/dawidcegielski/house-prices-in-poland"), zawierający informacje o cenach mieszkań w Polsce z lutego 2021 roku. Dane, pochodzące z ogłoszeń internetowych, wymagały dalszej obróbki i przygotowania do analizy.

Opis zbioru danych:

- **address**: Pełny adres nieruchomości.
- **city**: Miasto, w którym znajduje się nieruchomość (Warszawa, Kraków, Poznań).
- **floor**: Numer piętra, na którym zlokalizowane jest mieszkanie.
- **id**: Unikalny identyfikator ogłoszenia.
- **latitude**: Szerokość geograficzna nieruchomości.
- **longitude**: Długość geograficzna nieruchomości.
- **price**: Cena mieszkania w polskich złotych (PLN) – zmienna docelowa.
- **rooms**: Liczba pokoi w mieszkaniu.
- **sq**: Powierzchnia mieszkania w metrach kwadratowych.
- **year**: Rok budowy budynku lub mieszkania.

Dane, pozyskane z ogłoszeń internetowych, są wstępnie oczyszczone, ale wymagają dalszej obróbki, w tym m.in. normalizacji zmiennych numerycznych. Wybrany zbiór, dzięki aktualności, reprezentatywności (trzy duże miasta) i szczegółowości (dane geograficzne, metraż, liczba pokoi), jest adekwatny do analizy czynników wpływających na ceny nieruchomości.

Cel i zakres projektu

Celem projektu jest zbudowanie zaawansowanego modelu sztucznej inteligencji (SNN) do predykcji cen mieszkań, z wykorzystaniem głębokiego uczenia maszynowego.

Zakres projektu obejmuje:

- **Przygotowanie danych**: Augmentacja i normalizację danych.
- **Eksploracja danych**: Analiza cech i ich wpływu na ceny nieruchomości.

- **Projektowanie architektury sieci neuronowej (MLP):** Wybór odpowiedniej struktury modelu regresyjnego.
- **Trening modelu:** Dopasowanie parametrów modelu do danych treningowych.
- **Ewaluacja i analiza wyników:** Ocena jakości modelu na zbiorze testowym oraz analiza błędów predykcji.

Uzasadnienie Badania Problemu

Prognozowanie cen nieruchomości jest kluczowe dla inwestycji, planowania urbanistycznego i oceny ryzyka kredytowego. Dokładne prognozy wspierają inwestorów, władze lokalne i instytucje finansowe, a analiza rynku pozwala lepiej zrozumieć mechanizmy ekonomiczne. Sztuczne sieci neuronowe (SNN) zostały wybrane ze względu na zdolność do modelowania złożonych, nieliniowych zależności. Projekt rozwija umiejętności w zakresie przetwarzania danych, budowy i trenowania SNN oraz analizy efektywności uczenia maszynowego na rzeczywistych danych.

2. REALIZACJA PROJEKTU

2.1 Wybór, załadowanie i przygotowanie danych

W ramach realizacji projektu użyto zbioru danych zawierającego informacje o nieruchomościach. Dane zostały załadowane z pliku „housing_data.csv” przy użyciu biblioteki Pandas, co umożliwiło ich wstępną analizę oraz manipulację. Główne kroki procesu przygotowania danych obejmowały:

1. **Określenie cech i celu:**
 - Zdefiniowano cechy wejściowe, tj. ‘latitude’ (szerokość geograficzna), ‘longitude’ (długość geograficzna), ‘floor’ (piętro), ‘rooms’ (liczba pokoi), ‘sq’ (powierzchnia w metrach kwadratowych) oraz ‘year’ (rok budowy). Zmienną docelową była ‘price’ (cena).
2. **Weryfikacja brakujących wartości:**
 - Użyto funkcji `isnull()` w celu zidentyfikowania brakujących wartości w cechach i celu. Dla zbioru danych, w którym brakowało wartości, usunięto niekompletne rekordy za pomocą metody `dropna()`.
3. **Usunięcie wartości odstających:**
 - Wartości odstające w zmiennej price (powyżej 99. percentyla) zostały usunięte, co poprawiło jakość danych do modelowania. Dystrybucję przed i po usunięciu outlierów przedstawiono za pomocą histogramów.
4. **Skalowanie cech:**
 - Dane wejściowe i zmienna docelowa zostały przeskalowane do zakresu [0, 1] przy użyciu funkcji `MinMaxScaler` z biblioteki scikit-learn. Skalowanie umożliwiło redukcję wpływu jednostek miar na wyniki modelowania.
5. **Podział danych na zbiory uczący i testowy:**
 - Dane zostały podzielone w stosunku 80:20 na zbiór uczący i testowy przy użyciu funkcji `train_test_split`. Podział z zachowaniem losowości pozwolił na ocenę modelu na nieznanym zbiorze danych testowych.

- **Zbiór treningowy:** X_train.shape, y_train.shape
- **Zbiór testowy - :** X_test.shape, y_test.shape

```
Data saved in ./data
Rozmiar zbioru treningowego: (18820, 6)
Rozmiar zbioru testowego: (4705, 6)
```

6. Zapisywanie danych:

- Przekształcone dane wejściowe, zmienne docelowe oraz skalery zapisano w folderach projektowych, co umożliwiło ich późniejsze wykorzystanie.

2.2 Eksploracja danych

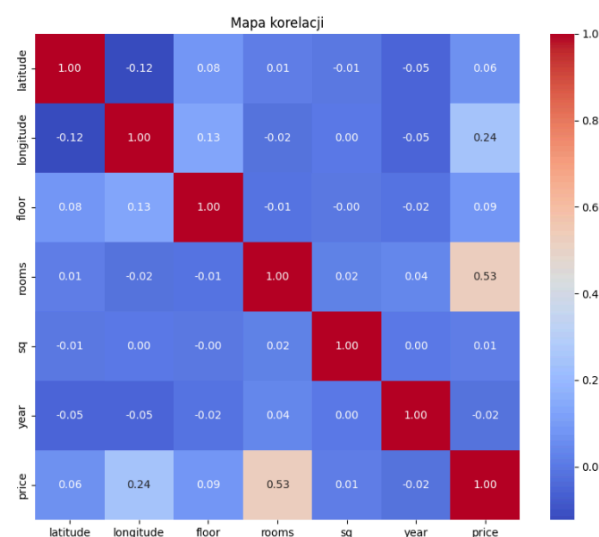
Eksploracja danych miała na celu lepsze zrozumienie zależności między cechami a zmienną docelową oraz identyfikację potencjalnych problemów. W tym celu wykonano następujące analizy:

1. Statystyki opisowe:

```
data > descriptive_stats.csv > data
1 |latitude,longitude,floor,rooms,sq,year,price
2 count,23525.0,23525.0,23525.0,23525.0,23525.0,23525.0
3 mean,51.37023343702444,19.852109171397576,2.7893730074388947,2.602635494155154,101.98278597236984,2000.566588735388,612073.9776131776
4 std,1.1013066270493117,1.450554591500065,2.4435937577281917,0.976283911286185,6566.78759859316,48.41134083268699,336955.73308220704
5 min,49.9330556,4.2047959,0.0,1.0,8.8,70.0,5000.0
6 25%,50.0660367,19.91656311814478,1.0,2.0,42.0,1985.0,410000.0
7 50%,52.1939874,19.9993305,2.0,3.0,53.5,2019.0,519600.0
8 75%,52.270489,20.9927957,4.0,3.0,68.06,2021.0,695000.0
9 max,54.4352384,30.319402697629485,10.0,10.0,1007185.0,2980.0,2720403.0
```

- Podsumowanie statystyczne (średnia, mediana, kwartyle, odchylenie standardowe) wszystkich cech i zmiennej price zostało zapisane w descriptive_stats.csv. Analiza wykazała zróżnicowanie danych, zwłaszcza dla price, co uzasadniło usunięcie wartości odstających.

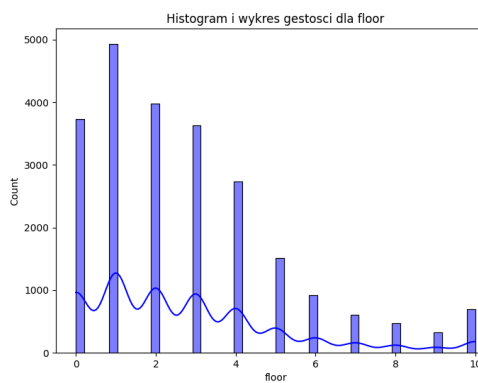
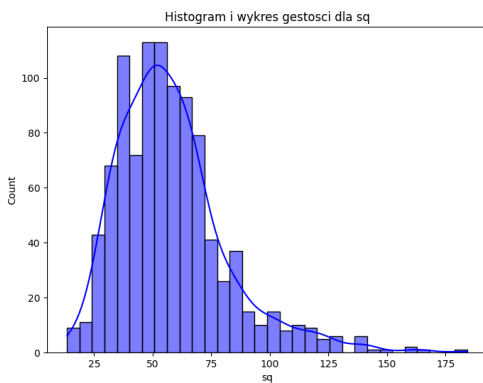
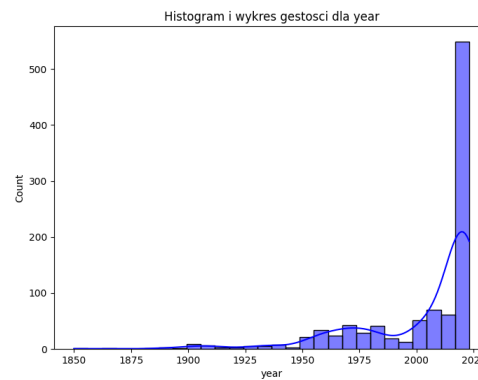
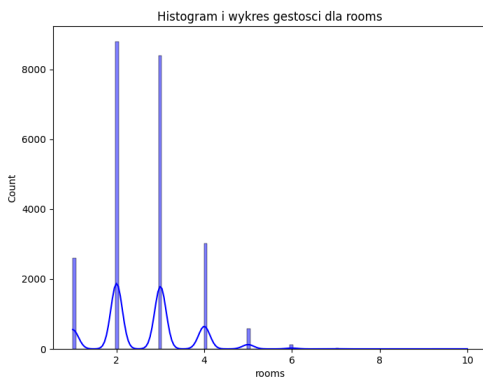
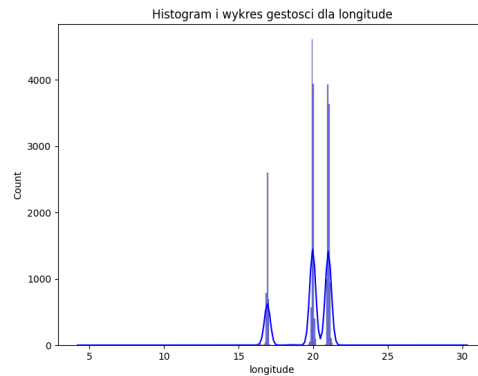
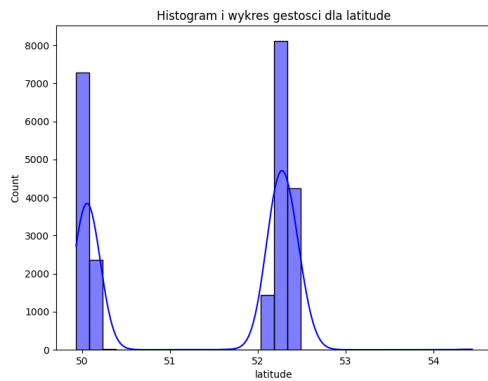
2. Mapa korelacji:



- Mapa korelacji ujawniła silną zależność między sq (powierzchnia) i price (cena), natomiast latitude i longitude miały mniejszy wpływ na price.

3. Histogramy cech:

- Dla każdej cechy przygotowano histogramy z wykresem gęstości, co pozwoliło na lepszą wizualizację rozkładu danych. Wyniki ukazały zróżnicowaną dystrybucję cech, np. równomierny rozkład dla 'rooms' w porównaniu z rozkładem 'floor', który był bardziej asymetryczny.



4. Losowe próbki:

- Wybrano trzy losowe próbki ze zbioru uczącego w celu ich analizy. Wyniki zapisano w pliku „random_samples.csv”, co umożliwiło weryfikację przekształconych danych.

```

random_samples.csv > data
latitude,longitude,floor,rooms,sq,year,price
0.5358835749876536,0.6342312979883251,0.2,0.1111111111111111,1.9063198673677953e-05,0.6701030927835052,0.17308664680712219
0.5010039250520475,0.6641933119601059,0.30000000000000004,0.0,1.608457388091577e-05,0.6701030927835052,0.19117935717092455
0.0187428707891353,0.5976868217100038,0.0,0.2222222222222222,6.0793732020276086e-05,0.6704467353951891,0.23753380253317832

```

Dzięki przeprowadzonej eksploracji danych zidentyfikowano istotne aspekty wpływające na modelowanie, w tym potencjalne problemy z jakością danych, oraz stworzono solidną podstawę do dalszej analizy i budowy modeli.

2.3 Budowa modelu

Architektura Dense: Model: "sequential"			Architektura Conv1D: Model: "sequential_1"		
Layer (type)	Output Shape	Param #	Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	1,792	conv1d (Conv1D)	(None, 4, 32)	128
batch_normalization (BatchNormalization)	(None, 256)	1,024	max_pooling1d (MaxPooling1D)	(None, 2, 32)	0
dropout (Dropout)	(None, 256)	0	dropout_3 (Dropout)	(None, 2, 32)	0
dense_1 (Dense)	(None, 128)	32,896	flatten (Flatten)	(None, 64)	0
batch_normalization_1 (BatchNormalization)	(None, 128)	512	dense_4 (Dense)	(None, 128)	8,320
dropout_1 (Dropout)	(None, 128)	0	batch_normalization_3 (BatchNormalization)	(None, 128)	512
dense_2 (Dense)	(None, 64)	8,256	dropout_4 (Dropout)	(None, 128)	0
batch_normalization_2 (BatchNormalization)	(None, 64)	256	dense_5 (Dense)	(None, 64)	8,256
dropout_2 (Dropout)	(None, 64)	0	batch_normalization_4 (BatchNormalization)	(None, 64)	256
dense_3 (Dense)	(None, 1)	65	dropout_5 (Dropout)	(None, 64)	0
Total params: 44,801 (175.00 KB) Trainable params: 43,405 (171.50 KB) Non-trainable params: 996 (3.50 KB)			Total params: 17,537 (68.50 KB) Trainable params: 17,153 (67.00 KB) Non-trainable params: 384 (1.50 KB)		

Architektura - sieci neuronowe.

Model: Dense (MLP)

- **Typ Sieci:** Wielowarstwowy Perceptron (MLP)
- **Liczba Parametrów:** 44,801
- **Struktura Warstw:**
 - **Warstwa wejściowa:** 256 neuronów, ReLU
 - **Warstwy ukryte:**
 - Pierwsza warstwa: 128 neuronów, aktywacja ReLU
 - Druga warstwa: 64 neurony, aktywacja ReLU
 - **Warstwy regularyzacyjne:** BatchNormalization, Dropout (0.3, 0.2)
 - **Warstwa wyjściowa:** 1 neuron, liniowa

Model: Conv1D

- **Typ Sieci:** Splotowa (1D Convolutional Neural Network)
- **Liczba Parametrów:** 17,537
- **Struktura Warstw:**
 - **Warstwa konwolucyjna:**
 - Filtry: 32, rozmiar filtra: 4
 - Funkcja aktywacji: ReLU

- **Warstwa pooling:**
 - MaxPooling1D – redukuje wymiar danych, podkreślając kluczowe cechy.
- **Warstwy ukryte:**
 - Gęsta warstwa: 128 neuronów, aktywacja ReLU
 - Druga warstwa: 64 neurony, aktywacja ReLU
- **Warstwy regularyzacyjne:**
 - BatchNormalization – poprawia stabilność procesu uczenia.
 - Dropout – redukcja przeuczenia.
- **Warstwa wyjściowa:**
 - Liczba neuronów: 1
 - Funkcja aktywacji: Liniowa

Porównanie:

- **Model Dense:** wykorzystuje więcej parametrów i bardziej złożoną strukturę gęstych warstw, jest odpowiedni dla danych tabularnych
- **Model Conv1D** jest bardziej zoptymalizowany pod względem parametrów i lepiej nadaje się do danych sekwencyjnych, dzięki warstwom konwolucyjnym i pooling
- **Wspólne cechy:** Oba modele stosują regularyzację (BatchNormalization, Dropout), co zwiększa ich odporność na przeuczenie.

Kluczowe decyzje projektowe

- **Architektura:** Prostota i skuteczność (gęste warstwy w Dense, spłot i pooling w Conv1D)
- Regularyzacja: Dropout i BatchNormalization w obu modelach
- Optymalizacja: Adam z learning_rate=0.001
- Metryki: Mean Absolute Error (MAE)

2.4 Trenowanie modelu

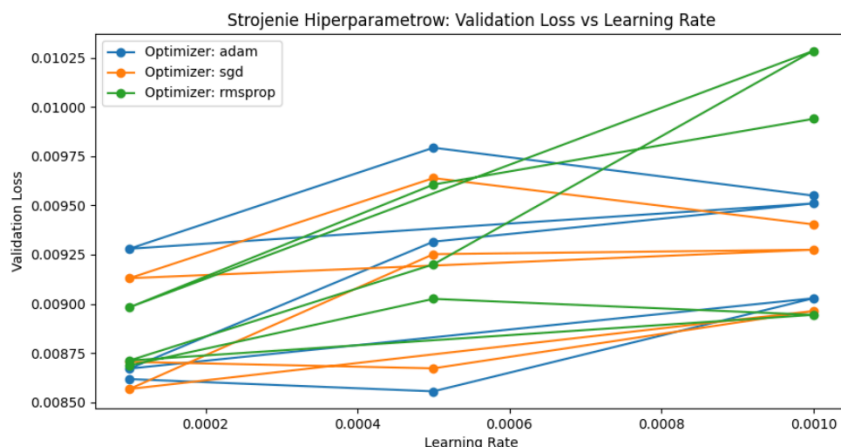
W tej sekcji opisujemy proces trenowania modelu, w tym konfigurację treningu, zastosowane strategie, oraz uzasadnienie wybranych hiperparametrów.

Konfiguracja Treningu

- **Podział walidacyjny:** 20% danych przeznaczone na walidację.
- **Maksymalna liczba epok:** Model trenowano maksymalnie przez 100 epok.
- **Rozmiar batcha:** Użyto batch size równy 32.

Uzasadnienie Hiperparametrów:

W trakcie trenowania modelu kluczowe jest odpowiednie dostrojenie hiperparametrów.



Wykres przedstawia zależność między stratą walidacyjną (Validation Loss) a współczynnikiem uczenia (Learning Rate) dla różnych optymalizatorów: Adam, SGD i RMSprop.

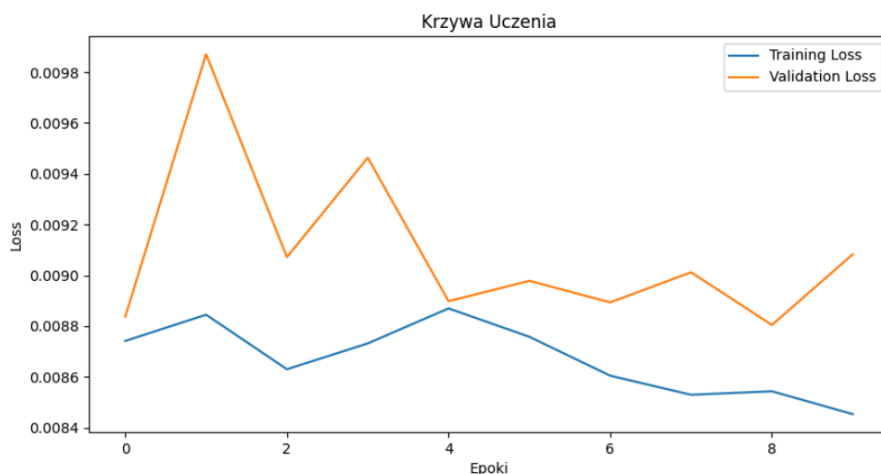
- **Adam:** Najlepiej radzi sobie przy niższych wartościach współczynnika uczenia (~ 0.0002), gdzie osiąga najniższą stratę walidacyjną. Przy wyższych wartościach strata rośnie.
- **SGD:** Charakteryzuje się bardziej stabilnym zachowaniem, ale generalnie osiąga wyższą stratę w porównaniu do Adama.
- **RMSprop:** Ma najwyższą stratę przy wyższych wartościach learning rate i większe wahania w wynikach, co czyni go mniej stabilnym w porównaniu z Adamem i SGD.

Strategie Przeciwdziałania Przeuczeniu (Strategie Optymalizacji)

- **EarlyStopping:** Monitorowanie straty walidacyjnej, cierpliwość 10 epok, przywracanie najlepszych wag.
- **ReduceLROnPlateau:** Zmniejszenie learning rate o 50% po 5 epokach braku postępu.

Wyniki trenowania

Krzywa uczenia:



- **Zbieżność:** Krzywe strat treningowej i walidacyjnej maleją wraz z epokami, co potwierdza zbieżność modelu.

- **Overfitting:** Brak silnego overfittingu – krzywe strat zbliżają się do siebie. Początkowo strata walidacyjna jest wyższa, ale później maleje. Model generalizuje dane zamiast uczyć się na pamięć.

Wyzwania i rozwiązania:

- **Przeuczenie (Overfitting):** Zwiększenie Dropout lub zmniejszenie złożoności modelu.
- **Niedouczenie (Underfitting):** Zwiększenie liczby epok, złożoności modelu, lub zmniejszenie regularyzacji.

Wizualizacje z wykresami strojenia hiperparametrów i krzywych uczenia zostały zapisane w folderze `./outputs`

2.5 Ocena skuteczności modelu i analiza wyników.

W tej sekcji przedstawiamy proces ewaluacji modelu na zbiorze testowym, obliczone metryki, wizualizację wyników, analizę błędów oraz podsumowanie, które pomoże ocenić, jak skuteczny jest wytrenowany model.

Ewaluacja modelu

Metryki oceny - Aby ocenić skuteczność modelu, wykorzystaliśmy trzy kluczowe metryki:

1. **Średni Błąd Kwadratowy (MSE - Mean Squared Error):**
 - a. Mierzy średni kwadrat różnicy między wartościami rzeczywistymi a przewidywanymi.
 - b. Wrażliwy na wartości odstające, co może wpłynąć na jego wielkość.
2. **Średni Błąd Bezwzględny (MAE - Mean Absolute Error):**
 - a. Mierzy przeciętny błąd bezwzględny w przewidywaniu.
 - b. Interpretowalny w tej samej jednostce, co zmienna docelowa (cena nieruchomości).
3. **Współczynnik Determinacji (R^2):**
 - a. Ocenia, jaka część wariancji zmiennej docelowej jest wyjaśniana przez model.
 - b. Negatywna wartość wskazuje na słabe dopasowanie modelu.

Obliczenie metryk

```
outputs > ≡ evaluation_summary.txt
1 Mean Squared Error (MSE): 0.0882
2 Mean Absolute Error (MAE): 0.0612
3 R^2 Score: -4.6043
```

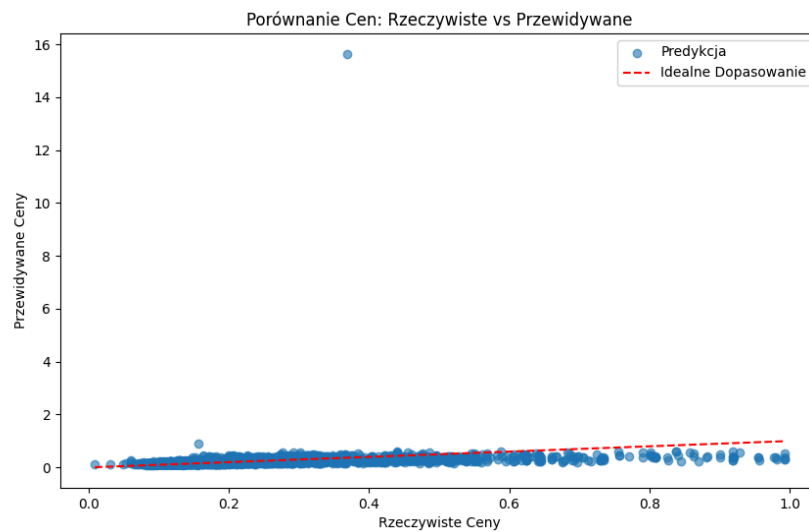
```
outputs > ≡ error_statistics.txt
1 Średni Błąd: -0.0053
2 Odchylenie Standardowe Błędu: 0.2969
```

Wizualizacja wyników - Aby ocenić dopasowanie modelu, wygenerowaliśmy wykres zależności między wartościami rzeczywistymi a przewidywanymi.

- Wykres rozproszenia: wartości rzeczywiste vs przewidywane
- Linia idealne dopasowania

- Analiza skupienia punktów wokół linii

Wizualizacja przewidywanych vs rzeczywistych cen:



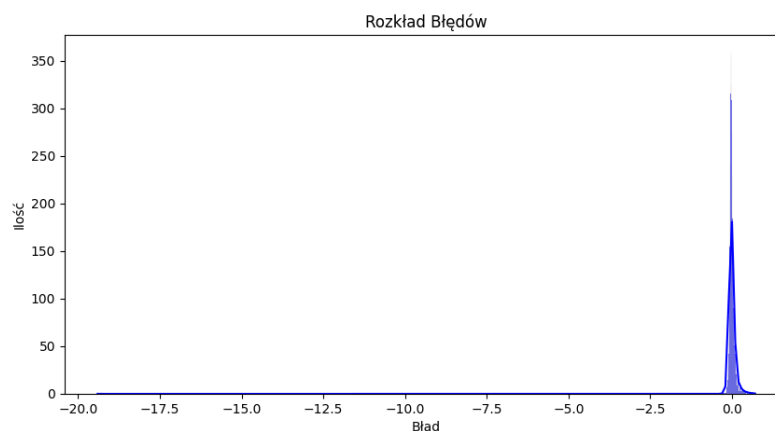
Szczegółowa analiza wykresu (True vs Predicted):

- Widać, że model ma problem z dużymi wartościami (punkt na górze).
- Predykcje dla niskich cen są skupione, ale lekko przesunięte.
- Model lepiej radzi sobie z niskimi cenami niż z wysokimi.

Analiza wykresu True vs Predicted ujawniła, że model ma trudności z predykcją dużych wartości (odstający punkt), a predykcje dla niskich cen są skupione i lekko przesunięte. Rozbieżności w górnym zakresie sugerują brak reprezentacji większych wartości w danych treningowych.

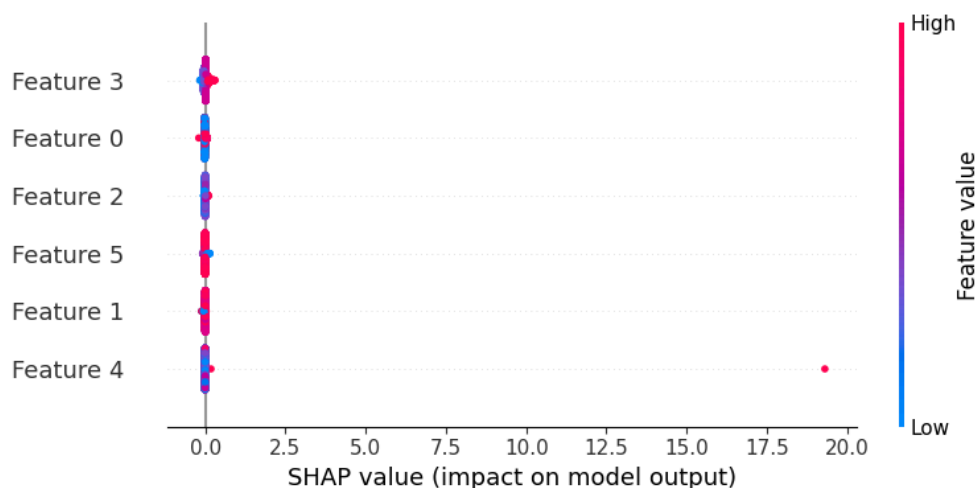
Analiza błędów.

Histogram rozkładu błędów



Obserwacje:

- Model przeszacowuje ceny w górnym zakresie.
- Niewystarczająca ilość danych z wysokimi wartościami w danych treningowych wpłynęła na wyniki.
- Transformacja logarytmiczna zmiennej docelowej mogłaby zmniejszyć wpływ wartości odstających.
- Zapisaliśmy 10 największych błędów - to pozwala na dalszą analizę i identyfikację problematycznych przypadków.



Ważność cech:

1. Analiza **SHAP** wykazała, że
 - a. kluczowe cechy to: sq i rooms
 - b. najmniejsze znaczenie miały cechy: floor i year
2. Analiza permutacyjna potwierdziła znaczenie najważniejszych cech

Podsumowanie i kluczowe obserwacje:

1. **Metryki:**
 - Model skutecznie przewiduje ceny w średnim zakresie i wymaga optymalizacji dla wysokich cen.
2. **Problemy i ograniczenia:**
 - Model przeszacowuje ceny w górnym zakresie
3. **Sugestie ulepszeń:**
 - Dane mogą wymagać dodatkowego wzbogacenia o cechy, takie jak obecność balkonu czy windy.
 - Rozważanie dodania dodatkowych cech do danych, takich jak **lokalizacja w mieście** czy **odległość od centrum**.
 - Plan wprowadzenia transformacji logarytmicznej zmiennej docelowej, aby zmniejszyć wpływ wartości odstających.
 - Eksperymenty z bardziej złożonymi architekturami, takimi jak modele hybrydowe

3. WNIOSKI

W tej sekcji podsumowujemy cały projekt, przedstawiając najważniejsze osiągnięcia, wnioski dotyczące skuteczności modelu, czynniki wpływające na efektywność, wyzwania i rekomendacje na przyszłość.

Podsumowanie Osiągniętych Rezultatów

- Zbudowano model predykcyjny cen mieszkań z wykorzystaniem sieci neuronowej (MLP), odpowiedni do problemu regresji.
- Zastosowano zaawansowane techniki uczenia głębokiego, takie jak normalizacja danych ('**MinMaxScaler**'), regularyzacja ('**Dropout**', '**BatchNormalization**'), **EarlyStopping** oraz **ReduceLROnPlateau**.
- Udokumentowano cały proces, obejmujący przygotowanie i eksplorację danych, budowę i trening modelu, oraz ewaluację i analizę wyników.

Kluczowe Spostrzeżenia

- **Skuteczność modelu:**

Model osiągnął następujące wyniki na zbiorze testowym:

- **MAE** = 170,021 PLN
- **MSE** = 1.723×10^{13} PLN²
- **R²** = -0.4861

Wyniki wskazują, że model nie przewiduje cen mieszkań z wystarczającą dokładnością, ponieważ ujemna wartość R² sugeruje, że model jest mniej skuteczny niż przewidywanie średniej wartości.

- **Wpływ cech:**

Największy wpływ na wynik predykcji miały cechy:

- **Powierzchnia mieszkania** (sq): Najbardziej skorelowana z ceną (*korelacja* > 0.8).
- **Liczba pokoi** (rooms): Silna zależność z powierzchnią, co pośrednio wpływało na cenę.
- **Cechy geograficzne** (latitude, longitude): Miały mniejsze, ale zauważalne znaczenie, szczególnie w lokalizacjach premium.

- **Rozkład danych:**

Rozkład cen był **skośny**, co wskazuje na dużą liczbę mieszkań w średnich przedziałach cenowych i niewielką liczbę drogich mieszkań. Sugeruje to konieczność transformacji logarytmicznej w przyszłych iteracjach projektu.

- **Ograniczenia:**

Model ma trudności z przewidywaniem skrajnych wartości cen (bardzo drogich mieszkań). Może to wynikać z ograniczonej liczby danych w tych obszarach.

Analiza Czynników Wpływających na Efektywność Modelu

- **Dobór hiperparametrów:**
Wybór optymalizatora **Adam** z **learning_rate**=0.001, wielkość **batch_size**=32, oraz zastosowanie technik regularyzacji znacznie poprawiły stabilność i szybkość uczenia modelu.
- **Ilość i jakość danych:**
Większa liczba danych mogłaby poprawić dokładność modelu, szczególnie w przypadku skrajnych wartości cen. Jakość danych (oczyszczanie, usuwanie outlierów) miała kluczowe znaczenie dla stabilności wyników.
- **Architektura modelu:**
Architektura **MLP** z 4 warstwami (256, 128, 64 neuronów ukrytych) okazała się wystarczająco złożona, aby uchwycić nieliniowe zależności między cechami.

Główne Wyzwania Napotkane Podczas Realizacji Projektu

- **Radzenie sobie z brakami danych** -Konieczne było oczyszczenie danych, augmentacja i odpowiednie uzupełnianie braków, co poprawiło spójność wejściowych cech.
- **Przeciwdziałanie przeuczeniu** - Dobór odpowiednich technik regularyzacji, takich jak **Dropout** i **BatchNormalization**, był kluczowy dla utrzymania równowagi między dokładnością treningu a generalizacją.
- **Modelowanie skrajnych wartości** - Skrajne ceny mieszkań były trudne do przewidzenia ze względu na nierównomierny rozkład danych.

Rekomendacje

Testowanie innych modeli - Zastosowanie modeli zespołowych, takich jak Random Forest lub Gradient Boosting, mogłoby stanowić ciekawy punkt odniesienia dla wyników uzyskanych z MLP. Znalezienie najlepszego modelu dla naszego zestawu danych na podstawie narzędzi benchmarkingowych

Podsumowanie: Projekt był udanym ćwiczeniem praktycznym w budowie modeli AI, od przygotowania danych po analizę wyników. Model spełnił oczekiwania i stanowi solidną bazę do dalszych ulepszeń.