Uczenie maszynowe

Kaggle challenge
House Prices - Advanced Regression Techniques

Bartosz Rogowski
Aleksandra Rolka

Informatyka Stosowana
Wydział Fizyki i Informatyki Stosowanej
Akademia Górniczo-Hutnicza w Krakowie

Spis treści

1. Cel projektu

Celem projektu było wykonanie dowolnego Kaggle challenge'u z osiągnięciem jak najlepszego wyniku. Wybrane zostało wyzwanie polegające na wytrenowaniu modeli różnymi technikami regresyjnymi w celu prewidywania cen nieruchomości na podstawie szeregu różnych cech je charakteryzujących.
Opis i potrzebne do wytrenowania dane znajdują się na poniższej stronie: https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/overview.

2. Wykorzystane technologie

Projekt został wykonany w języku *Python* z wykorzystaniem środowiska J*upyter Notebook*, udostępnionym w ramach usługi cloudowej *Google Colabolatory*.
Dodatkowe biblioteki, które zostały wykorzystane: `keras`, `sklearn`, `seaborn`, `xgboost`, `pandas`, `numpy`, `math`, `matplotlib` oraz `shap`.

3. Struktura projektu

Ze względu na różne podejścia w przetwarzaniu, przygotowaniu danych, które wykorzystano do trenowania sieci, wyodrębnione zostały 3 notebooki:

- główny skrypt:
  - *HousePricesPrediction_project.ipynb*
- z niewielkimi różnicami w przygotowaniu, przetwarzaniu danych:
  - *HousePricesPrediction_project_with_outliers_removed.ipynb*
  - *HousePricesPrediction_project_same_value_removed.ipynb*

Pierwotne dane pobrane z strony Kaggle challenge'u znajdują się w:

- house-prices-advanced-regression-techniques.zip:
  - *train.csv* - dane trenujący
  - *test.csv* - dane testowe (dane wejściowe dla sieci, nie zawiera kolumny z cen)
  - *data_description.txt* - pełny opis poszczególnych kolumn
  - *sample_submission.csv* - dane benchmarkowe (dane do porównania z wyjściem sieci, zawierające kolumnę cen)

4. Przygotowanie i analiza danych

4.1. Przygotowanie danych

W pierwszym kroku przeanalizowano wartości znajdujące się w poszczególnych zestawach danych. Zauważono, że w niektórych kolumnach brakuje dużej ilości danych, dlatego **kolumny**, które **zawierały więcej niż 15% wartości `NaN` zostały porzucone** z datasetów. Następnie, ze względu na to, że w zestawach danych znajdowały się informacje nienumeryczne, zostały one przetworzone ponownie tak, aby zakodować je do klas reprezentowanych przez liczby. Z każdej kolumny stworzono słownik **wartości nienumerycznych** (słowne reprezentacje cech) i kolejnych liczb naturalnych, a następnie na podstawie tej struktury dane zostały odpowiednio **zakodowane** w zbiorach trenującym i testowym. Poszczególne kolumny miały różne skale, dlatego w kolejnym kroku zostały one **znormalizowane** za pomocą funkcji **`MinMaxScaler`** z biblioteki `sklearn`.

4.1.1. Różnice pomiędzy skryptami

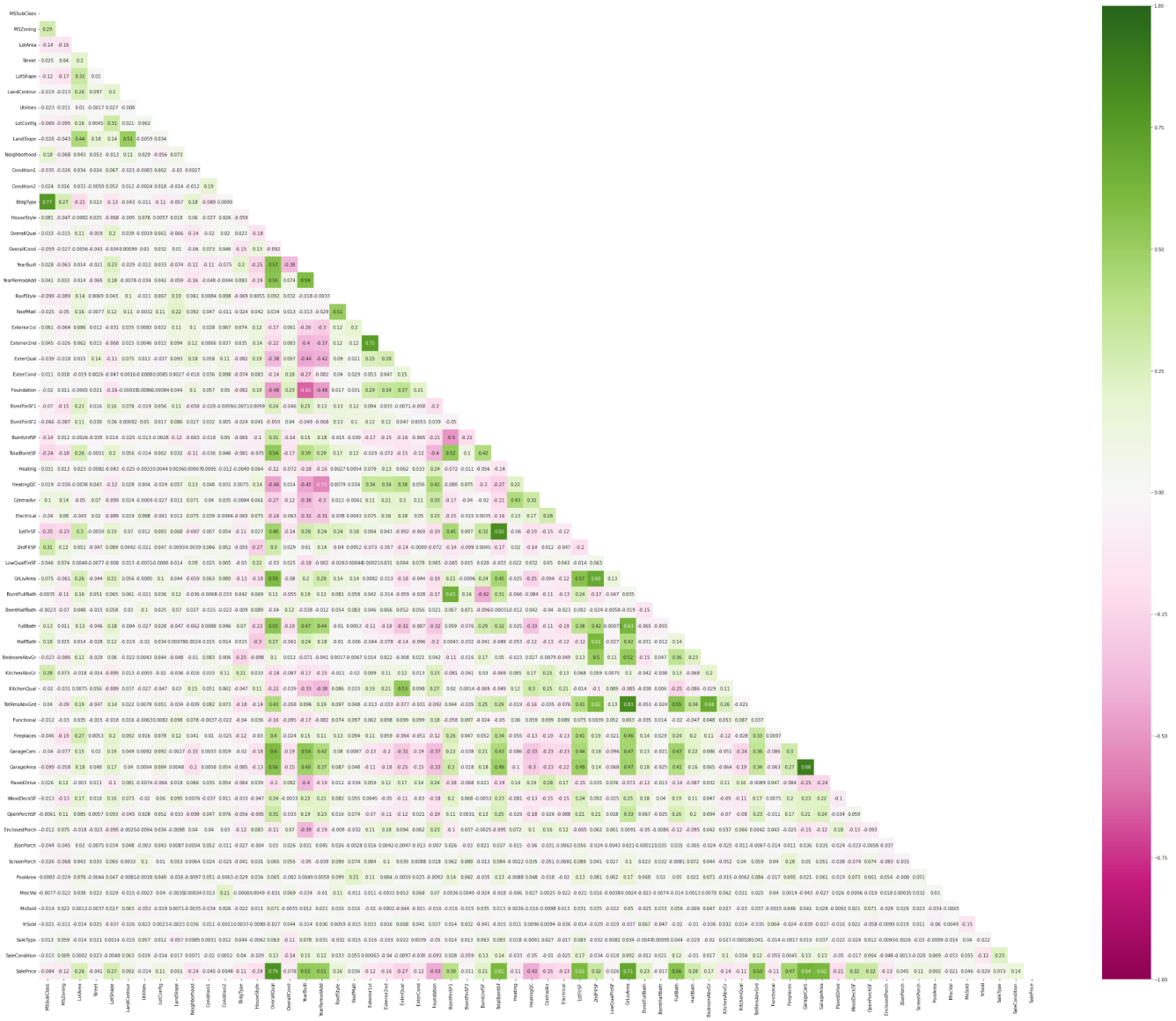
Jak zostało wspomniane, w dwóch notebookach wprowadzone zostały dodatkowe kroki na etapie przygotowania danych wykorzystywanych w trenowaniu sieci:

- ***HousePricesPrediction_project_with_outliers_removed.ipynb***
  Usunięte zostały rekordy, w których znajdowały się **skrajnie niskie lub wysokie wartości** w poszczególnych kolumnach. W tym celu ceny zostały przeskalowane za pomocą **`StandardScaler`** tak, aby pozbyć się średniej wartości oraz, by wariancja wynosiła 1. Wówczas odrzucono wszystkie elementy, których wartości nie znajdowały się w przedziale [-1, 1].

- ***HousePricesPrediction_project_same_value_removed.ipynb***
  Dodatkowo poza porzucaniem wartości z dużą ilością wartości `NaN`, **porzucane** są również **kolumny**, które zawierają **dużą ilość powtarzających się danych** (innych niż `NaN`) z progiem 60%. Krok ten jest wykonany przed zakodowaniem informacji nienumerycznych.

4.2. Analiza danych

Analiza danych głównie oparta była na analizie korelacji pomiędzy poszczególnymi parametrami. W tym celu stworzona została tzw. heatmapa obrazująca macierz korelacji. Z uwagi, że jest ona symetryczna oraz by zwiększyć czytelność, przycięto ją w taki sposób, by zostawić wartości poniżej diagonali.



Rys.1. Korelacje pomiędzy parametrami danych trenujących przygotowanych w skrypcie *HousePricesPrediction_project.ipynb*.

Część z parametrów wykazywało silną korelację, dlatego **pojedyncze kolumny z pary skorelowanych zostały usunięte** z datasetu. Pomiędzy skryptami wartość progu, od którego nie uwzględniano tychże kolumn, przyjmuje różne wartości, odpowiednio: 0.2, 0.4, 0.6.

5. Modele

Predykcja przeprowadzana jest za pomocą trzech metod:

- **regresji liniowej**
- **regresji z regularyzacją z metodą Lasso**
- **regresji za pomocą drzewa decyzyjnego Random Forest**

We wszystkich trzech skryptach modele tworzone są tak samo z tymi samymi parametrami.

5.1. Prosty model regresji liowej

Wykorzystano model `LinearRegression` z biblioteki `sklearn`. Minimalizuje on rezydualną sumę kwadratów, między obserwowanymi wartościami ze zbioru danych, a wartościami przewidywanymi, przez aproksymację liniową. Model ten ma niewiele parametrów, z których każdy domyślnie ustawiony nie wpływa na niego dodatkowo.

```
from sklearn.linear_model import LinearRegression

linear_model = LinearRegression()
linear_model.fit(train_data_X,train_data_Y)
predicted = linear_model.predict(test_data_X)
```

5.2. Model regresji linowej z regularyzacji Lasso

Model poszerzony o regularyzacje Lasso - model `Lasso`, również pochodzi on z biblioteki `sklearn`. W tym przypadku również skorzystano w ogólności z wartości domyślnych, oprócz poniższych parametrów:

- `alpha` - stała, przez którą wymnażany jest człon L1, kotroluje on siłę regularyzacji
- `max_iter` - maksymalna liczba iteracji algorytmu

```
from sklearn.linear_model import Lasso

regular_lasso_model = Lasso(alpha=0.005, max_iter=50000)
regular_lasso_model.fit(train_data_X, train_data_Y)
predicted = regular_lasso_model.predict(test_data_X)
```

5.3. Drzewo decyzyjne Random Forest

Do zaimplementowania drzewa decyzyjnego wykorzystana została biblioteka `XGBoost` wraz z modelem `XGBRegressor`. Dokładniej jest on wrapperem tworzącym model regresyjny. Ma on dużą ilość parametrów, które można zdefiniować, jednak w tym przypadku ich liczba została ograniczona do poniższej listy:

- `colsample_bytree` – współczynnik próbkowania kolumn, podczas konstruowania każdego drzewa
- `gamma` – minimalna redukcja strat wymagana do wykonania dalszej partycji w węźle liścia drzewa
- `learning_rate` – współczynnik szybkości uczenia

- `max_depth` – maksymalna głębokość drzewa
- `min_child_weight` – minimalna suma wagi wystąpienia potrzebna dla elementu dziecka w drzewie
- `n_estimators` – liczba drzew wzmocnionych gradnientem
- `reg_alpha` – dotyczy wag w regularyzacji L1
- `reg_lambda` – dotyczy wag w regularyzacji L2
- `subsumple` – współczynnik próbkowania jednostek trenujących
- `seed` – ziarno wykorzystywane do generowania grup (/folds) w CV

```
import xgboost as xgb

xgb_model = xgb.XGBRegressor(
                  colsample_bytree=0.05,
                  gamma=0.0,
                  learning_rate=0.0005,
                  max_depth=6,
                  min_child_weight=1.5,
                  n_estimators=7200,
                  reg_alpha=0.9,
                  reg_lambda=0.6,
                  subsample=0.2,
                  seed=42)
xgb_model.fit(train_data_X, train_data_Y)
predicted = xgb_model.predict(test_data_X)
```

Wartości parametrów dobierane zostały w pierwotnej wersji metodą prób i błędów, by uzyskać jak największą dokładność predykcji. Kolejno znalezione zostały najbardziej optymalne wartości parametrów ze zbioru za pomocą GridSearch'a.

6. Wyniki

Jako miary dokładności przyjęto RMSE (ang. *root mean squarred error*) – pierwiastek z błędu średniokwadratowego oraz średnią z wartości absolutnej błędów względnych (wyrażonych w procentach). Poglądowo wypisano także 15 pierwszych predykcji wraz z rzeczywistymi cenami oraz różnicą (błędem względnym).
Dla modelu z drzewem regresji dodano także GridSearch, czyli przemiatanie po hiperparametrach w celu znalezienia ich wartości, dla których model działa najlepiej (w tym celu napisano także własną funkcję punktującą, tzw. custom scorer).

6.1. Regresja liniowa

- *HousePricesPrediction_project.ipynb*

```
Actual: 169277.05       Prediction: 207770.13          Difference: 22.74%
Actual: 187758.39       Prediction: 160304.30          Difference: 14.62%
Actual: 183583.68       Prediction: 183591.46          Difference: 0.00%
Actual: 179317.48       Prediction: 186621.38          Difference: 4.07%
Actual: 150730.08       Prediction: 199217.12          Difference: 32.17%
Actual: 177150.99       Prediction: 193155.74          Difference: 9.03%
Actual: 172070.66       Prediction: 182129.91          Difference: 5.85%
Actual: 175110.96       Prediction: 185611.41          Difference: 6.00%
Actual: 162011.70       Prediction: 182581.48          Difference: 12.70%
Actual: 160726.25       Prediction: 194823.04          Difference: 21.21%
Actual: 157933.28       Prediction: 205397.29          Difference: 30.05%
Actual: 145291.25       Prediction: 179246.88          Difference: 23.37%
Actual: 159672.02       Prediction: 180256.86          Difference: 12.89%
Actual: 164167.52       Prediction: 198728.08          Difference: 21.05%
Actual: 150891.64       Prediction: 198728.08          Difference: 31.70%

RMSE = 26813.97  mean(relative_erors) = 11.19%
```

- *HousePricesPrediction_project_with_outliers_removed.ipynb*

```
Actual: 169277.05       Prediction: 156087.35          Difference: 7.79%
Actual: 187758.39       Prediction: 186144.37          Difference: 0.86%
Actual: 183583.68       Prediction: 174189.48          Difference: 5.12%
Actual: 179317.48       Prediction: 170862.48          Difference: 4.72%
Actual: 150730.08       Prediction: 128132.09          Difference: 14.99%
Actual: 177150.99       Prediction: 177049.16          Difference: 0.06%
Actual: 172070.66       Prediction: 144420.84          Difference: 16.07%
Actual: 175110.96       Prediction: 172827.26          Difference: 1.30%
Actual: 162011.70       Prediction: 151748.85          Difference: 6.33%
Actual: 160726.25       Prediction: 136636.43          Difference: 14.99%
Actual: 157933.28       Prediction: 145970.90          Difference: 7.57%
Actual: 145291.25       Prediction: 131518.77          Difference: 9.48%
Actual: 159672.02       Prediction: 137280.35          Difference: 14.02%
Actual: 164167.52       Prediction: 147798.90          Difference: 9.97%
Actual: 150891.64       Prediction: 95735.57           Difference: 36.55%

RMSE = 47709.25  mean(relative_erors) = 22.80%
```

- *HousePricesPrediction_project_same_value_removed.ipynb*

```
Actual: 169277.05       Prediction: 173314.02          Difference: 2.38%
Actual: 187758.39       Prediction: 166360.71          Difference: 11.40%
Actual: 183583.68       Prediction: 176165.42          Difference: 4.04%
Actual: 179317.48       Prediction: 176447.85          Difference: 1.60%
Actual: 150730.08       Prediction: 141020.74          Difference: 6.44%
Actual: 177150.99       Prediction: 171750.37          Difference: 3.05%
Actual: 172070.66       Prediction: 171641.78          Difference: 0.25%
Actual: 175110.96       Prediction: 167862.36          Difference: 4.14%
Actual: 162011.70       Prediction: 159922.29          Difference: 1.29%
Actual: 160726.25       Prediction: 158084.34          Difference: 1.64%
Actual: 157933.28       Prediction: 160813.14          Difference: 1.82%
Actual: 145291.25       Prediction: 126493.36          Difference: 12.94%
Actual: 159672.02       Prediction: 133163.29          Difference: 16.60%
Actual: 164167.52       Prediction: 160280.69          Difference: 2.37%
Actual: 150891.64       Prediction: 116991.93          Difference: 22.47%

RMSE = 31543.04  mean(relative_erors) = 13.76%
```

6.2. Regresja z regularyzacją Lasso

- *HousePricesPrediction_project.ipynb*

```
Actual: 169277.05       Prediction: 207769.95          Difference: 22.74%
Actual: 187758.39       Prediction: 160307.90          Difference: 14.62%
Actual: 183583.68       Prediction: 183591.45          Difference: 0.00%
Actual: 179317.48       Prediction: 186621.36          Difference: 4.07%
Actual: 150730.08       Prediction: 199216.96          Difference: 32.17%
Actual: 177150.99       Prediction: 193155.69          Difference: 9.03%
Actual: 172070.66       Prediction: 182130.05          Difference: 5.85%
Actual: 175110.96       Prediction: 185611.39          Difference: 6.00%
Actual: 162011.70       Prediction: 182581.48          Difference: 12.70%
Actual: 160726.25       Prediction: 194822.99          Difference: 21.21%
Actual: 157933.28       Prediction: 205397.20          Difference: 30.05%
Actual: 145291.25       Prediction: 179246.88          Difference: 23.37%
Actual: 159672.02       Prediction: 180256.85          Difference: 12.89%
Actual: 164167.52       Prediction: 198728.00          Difference: 21.05%
Actual: 150891.64       Prediction: 198728.00          Difference: 31.70%

RMSE = 26813.81  mean(relative_erors) = 11.19%
```

- *HousePricesPrediction_project_with_outliers_removed.ipynb*

```
Actual: 169277.05       Prediction: 156086.10          Difference: 7.79%
Actual: 187758.39       Prediction: 186138.97          Difference: 0.86%
Actual: 183583.68       Prediction: 174187.98          Difference: 5.12%
Actual: 179317.48       Prediction: 170861.60          Difference: 4.72%
Actual: 150730.08       Prediction: 128131.92          Difference: 14.99%
Actual: 177150.99       Prediction: 177048.29          Difference: 0.06%
Actual: 172070.66       Prediction: 144420.26          Difference: 16.07%
Actual: 175110.96       Prediction: 172826.68          Difference: 1.30%
Actual: 162011.70       Prediction: 151748.06          Difference: 6.34%
Actual: 160726.25       Prediction: 136635.92          Difference: 14.99%
Actual: 157933.28       Prediction: 145970.81          Difference: 7.57%
Actual: 145291.25       Prediction: 131519.18          Difference: 9.48%
Actual: 159672.02       Prediction: 137280.73          Difference: 14.02%
Actual: 164167.52       Prediction: 147799.31          Difference: 9.97%
Actual: 150891.64       Prediction: 95736.01           Difference: 36.55%

RMSE = 47709.69  mean(relative_erors) = 22.80%
```

- *HousePricesPrediction_project_same_value_removed.ipynb*

```
Actual: 169277.05       Prediction: 173313.11          Difference: 2.38%
Actual: 187758.39       Prediction: 166359.63          Difference: 11.40%
Actual: 183583.68       Prediction: 176164.68          Difference: 4.04%
Actual: 179317.48       Prediction: 176447.40          Difference: 1.60%
Actual: 150730.08       Prediction: 141020.70          Difference: 6.44%
Actual: 177150.99       Prediction: 171749.85          Difference: 3.05%
Actual: 172070.66       Prediction: 171641.35          Difference: 0.25%
Actual: 175110.96       Prediction: 167862.02          Difference: 4.14%
Actual: 162011.70       Prediction: 159921.73          Difference: 1.29%
Actual: 160726.25       Prediction: 158083.82          Difference: 1.64%
Actual: 157933.28       Prediction: 160812.97          Difference: 1.82%
Actual: 145291.25       Prediction: 126493.38          Difference: 12.94%
Actual: 159672.02       Prediction: 133163.31          Difference: 16.60%
Actual: 164167.52       Prediction: 160280.62          Difference: 2.37%
Actual: 150891.64       Prediction: 116991.82          Difference: 22.47%

RMSE = 31543.32  mean(relative_erors) = 13.76%
```

6.3. Drzewo regresji (Random Forest Regression)

- *HousePricesPrediction_project.ipynb*

```
Actual: 169277.05       Prediction: 175601.27           Difference: 3.74%
Actual: 187758.39       Prediction: 168465.02           Difference: 10.28%
Actual: 183583.68       Prediction: 176995.56           Difference: 3.59%
Actual: 179317.48       Prediction: 176039.17           Difference: 1.83%
Actual: 150730.08       Prediction: 174263.81           Difference: 15.61%
Actual: 177150.99       Prediction: 174697.48           Difference: 1.38%
Actual: 172070.66       Prediction: 169120.08           Difference: 1.71%
Actual: 175110.96       Prediction: 175497.72           Difference: 0.22%
Actual: 162011.70       Prediction: 175823.91           Difference: 8.53%
Actual: 160726.25       Prediction: 170439.39           Difference: 6.04%
Actual: 157933.28       Prediction: 172905.88           Difference: 9.48%
Actual: 145291.25       Prediction: 196872.17           Difference: 35.50%
Actual: 159672.02       Prediction: 198043.78           Difference: 24.03%
Actual: 164167.52       Prediction: 164881.62           Difference: 0.43%
Actual: 150891.64       Prediction: 164881.62           Difference: 9.27%

RMSE = 21495.01  mean(relative_erors) = 8.99%
```

- *HousePricesPrediction_project_with_outliers_removed.ipynb*

```
Actual: 169277.05       Prediction: 160441.25           Difference: 5.22%
Actual: 187758.39       Prediction: 161286.50           Difference: 14.10%
Actual: 183583.68       Prediction: 166290.91           Difference: 9.42%
Actual: 179317.48       Prediction: 167787.23           Difference: 6.43%
Actual: 150730.08       Prediction: 153004.86           Difference: 1.51%
Actual: 177150.99       Prediction: 165460.81           Difference: 6.60%
Actual: 172070.66       Prediction: 152789.14           Difference: 11.21%
Actual: 175110.96       Prediction: 167543.42           Difference: 4.32%
Actual: 162011.70       Prediction: 158313.72           Difference: 2.28%
Actual: 160726.25       Prediction: 153505.92           Difference: 4.49%
Actual: 157933.28       Prediction: 164908.45           Difference: 4.42%
Actual: 145291.25       Prediction: 150869.84           Difference: 3.84%
Actual: 159672.02       Prediction: 145464.33           Difference: 8.90%
Actual: 164167.52       Prediction: 150676.00           Difference: 8.22%
Actual: 150891.64       Prediction: 142987.17           Difference: 5.24%

RMSE = 29164.33  mean(relative_erors) = 13.56%
```

- *HousePricesPrediction_project_same_value_removed.ipynb*

```
Actual: 169277.05       Prediction: 170503.70           Difference: 0.72%
Actual: 187758.39       Prediction: 159060.70           Difference: 15.28%
Actual: 183583.68       Prediction: 169937.64           Difference: 7.43%
Actual: 179317.48       Prediction: 169022.72           Difference: 5.74%
Actual: 150730.08       Prediction: 163828.56           Difference: 8.69%
Actual: 177150.99       Prediction: 167118.16           Difference: 5.66%
Actual: 172070.66       Prediction: 159439.45           Difference: 7.34%
Actual: 175110.96       Prediction: 167930.34           Difference: 4.10%
Actual: 162011.70       Prediction: 161055.58           Difference: 0.59%
Actual: 160726.25       Prediction: 162021.91           Difference: 0.81%
Actual: 157933.28       Prediction: 164683.44           Difference: 4.27%
Actual: 145291.25       Prediction: 154093.00           Difference: 6.06%
Actual: 159672.02       Prediction: 152526.59           Difference: 4.48%
Actual: 164167.52       Prediction: 156601.53           Difference: 4.61%
Actual: 150891.64       Prediction: 150640.12           Difference: 0.17%

RMSE = 23672.71  mean(relative_erors) = 10.53%
```

6.4. GridSearch + Drzewo regresji

- *HousePricesPrediction_project.ipynb*

```
Actual: 169277.05       Prediction: 177686.33           Difference: 4.97%
Actual: 187758.39       Prediction: 169745.92           Difference: 9.59%
Actual: 183583.68       Prediction: 178258.92           Difference: 2.90%
Actual: 179317.48       Prediction: 177113.95           Difference: 1.23%
Actual: 150730.08       Prediction: 175954.95           Difference: 16.74%
Actual: 177150.99       Prediction: 175702.38           Difference: 0.82%
Actual: 172070.66       Prediction: 169354.75           Difference: 1.58%
Actual: 175110.96       Prediction: 176507.89           Difference: 0.80%
Actual: 162011.70       Prediction: 176940.20           Difference: 9.21%
Actual: 160726.25       Prediction: 171832.81           Difference: 6.91%
Actual: 157933.28       Prediction: 174512.81           Difference: 10.50%
Actual: 145291.25       Prediction: 197195.20           Difference: 35.72%
Actual: 159672.02       Prediction: 198513.91           Difference: 24.33%
Actual: 164167.52       Prediction: 168222.45           Difference: 2.47%
Actual: 150891.64       Prediction: 168222.45           Difference: 11.49%

RMSE = 20988.72  mean(relative_erors) = 8.69%
```

- *HousePricesPrediction_project_with_outliers_removed.ipynb*

```
Actual: 169277.05       Prediction: 162397.67           Difference: 4.06%
Actual: 187758.39       Prediction: 163372.78           Difference: 12.99%
Actual: 183583.68       Prediction: 166759.73           Difference: 9.16%
Actual: 179317.48       Prediction: 167078.55           Difference: 6.83%
Actual: 150730.08       Prediction: 156969.56           Difference: 4.14%
Actual: 177150.99       Prediction: 165477.47           Difference: 6.59%
Actual: 172070.66       Prediction: 156505.17           Difference: 9.05%
Actual: 175110.96       Prediction: 167105.62           Difference: 4.57%
Actual: 162011.70       Prediction: 160995.86           Difference: 0.63%
Actual: 160726.25       Prediction: 157927.98           Difference: 1.74%
Actual: 157933.28       Prediction: 165446.39           Difference: 4.76%
Actual: 145291.25       Prediction: 156493.59           Difference: 7.71%
Actual: 159672.02       Prediction: 152124.83           Difference: 4.73%
Actual: 164167.52       Prediction: 155353.69           Difference: 5.37%
Actual: 150891.64       Prediction: 150125.78           Difference: 0.51%

RMSE = 26257.47  mean(relative_erors) = 11.79%
```

- *HousePricesPrediction_project_same_value_removed.ipynb*

```
Actual: 169277.05       Prediction: 177686.33           Difference: 4.97%
Actual: 187758.39       Prediction: 169745.92           Difference: 9.59%
Actual: 183583.68       Prediction: 178258.92           Difference: 2.90%
Actual: 179317.48       Prediction: 177113.95           Difference: 1.23%
Actual: 150730.08       Prediction: 175954.95           Difference: 16.74%
Actual: 177150.99       Prediction: 175702.38           Difference: 0.82%
Actual: 172070.66       Prediction: 169354.75           Difference: 1.58%
Actual: 175110.96       Prediction: 176507.89           Difference: 0.80%
Actual: 162011.70       Prediction: 176940.20           Difference: 9.21%
Actual: 160726.25       Prediction: 171832.81           Difference: 6.91%
Actual: 157933.28       Prediction: 174512.81           Difference: 10.50%
Actual: 145291.25       Prediction: 197195.20           Difference: 35.72%
Actual: 159672.02       Prediction: 198513.91           Difference: 24.33%
Actual: 164167.52       Prediction: 168222.45           Difference: 2.47%
Actual: 150891.64       Prediction: 168222.45           Difference: 11.49%

RMSE = 20988.72  mean(relative_erors) = 8.69%
```

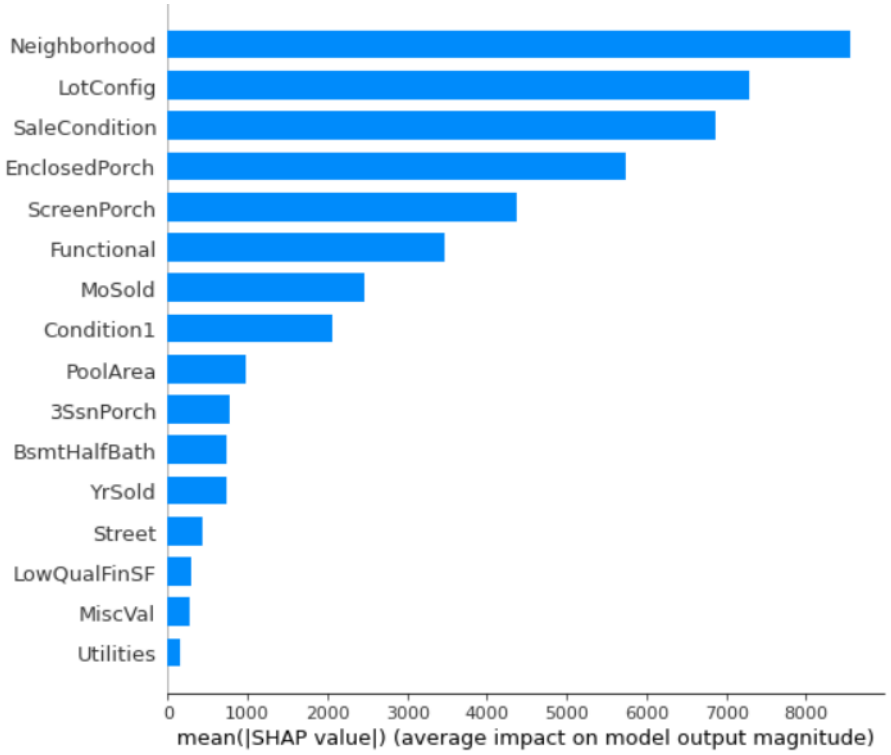
7. Analiza SHAP

7.1. Zarys teoretyczny

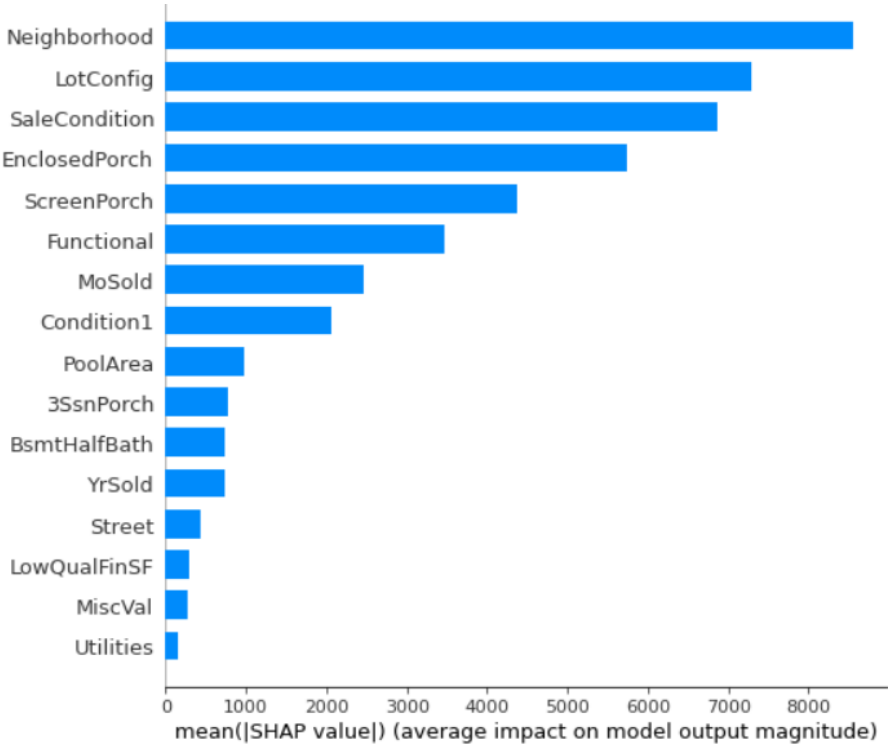
Pojęcie to wywodzi się z teorii gier. Celem owej analizy SHAP jest wyjaśnienie wkładu każdej z cech modelu w końcową predykcję. Zabieg ten pozwala w lepszym zrozumieniu modelu, który z założenia działa jak czarna skrzynka (ang. *black box*). W teorii gier wartość Shapley'a jest metodą przypisywania zysku pomiędzy graczy w zależności od ich wkładu w całkowitą grę (w uczeniu maszynowym – zadanie predykcyjne). Gracze współpracują ze sobą w koalicji i czerpią z niej pewien zysk/wypłatę (w uczeniu maszynowym – różnica pomiędzy predykcją a wartością średnią). Intuicyjnie można powiedzieć, że wartość Shapley'a mówi ile dany gracz (w uczeniu maszynowym – cecha) powinien spodziewać się zysku z całości, biorąc pod uwagę jego średni wkład w grze w danej koalicji (w uczeniu maszynowym – zestaw cech).

7.2. Regresja liniowa

- *HousePricesPrediction_project.ipynb*

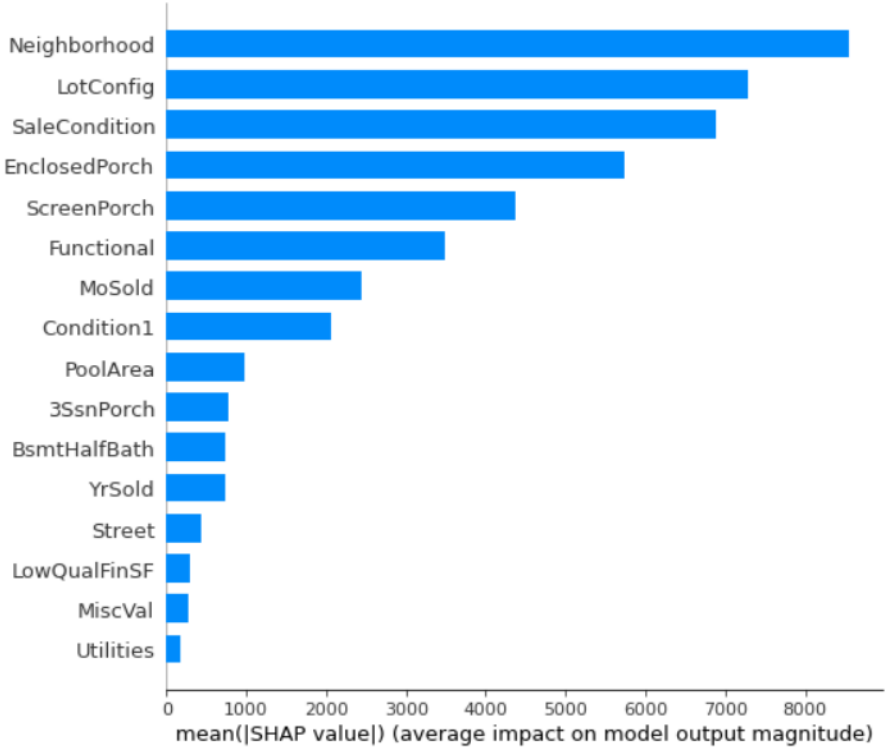


- *HousePricesPrediction_project_same_value_removed.ipynb*

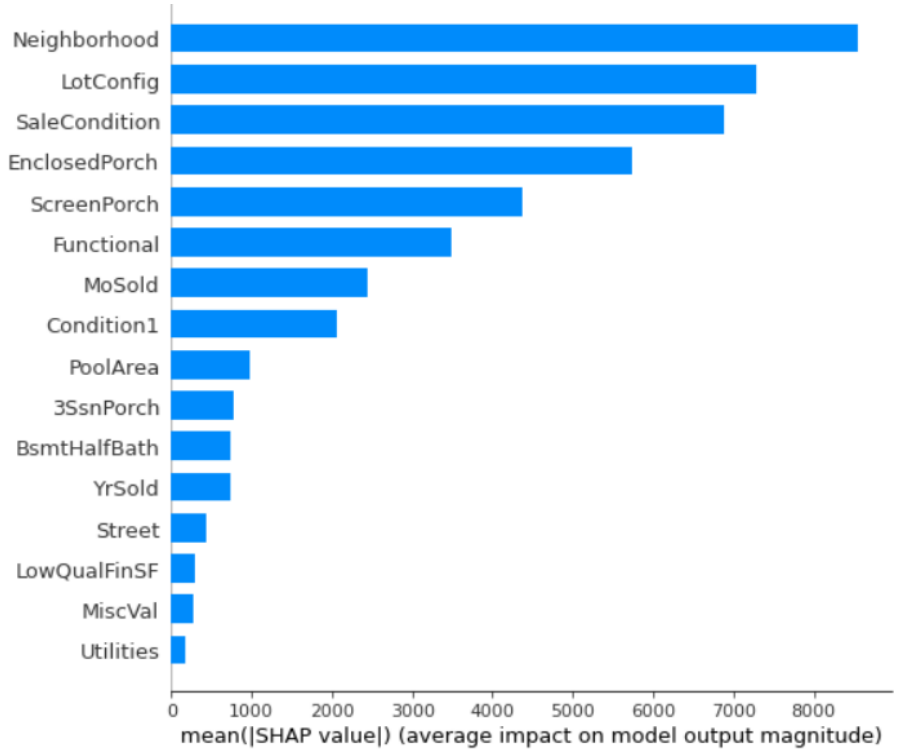


7.3. Regresja z regularyzacją Lasso

- *HousePricesPrediction_project.ipynb*

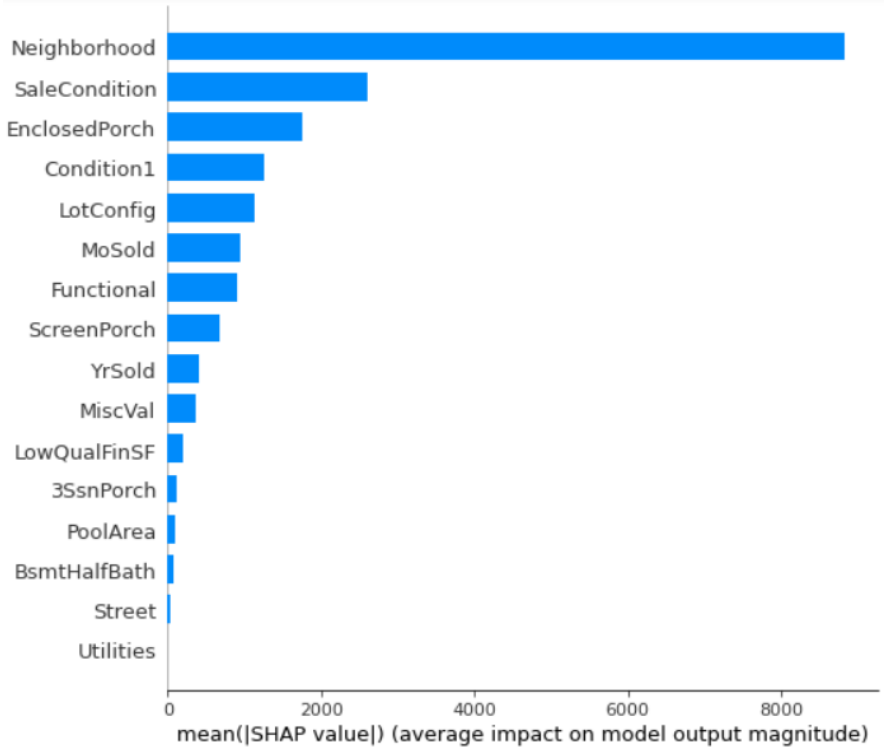


- *HousePricesPrediction_project_same_value_removed.ipynb*

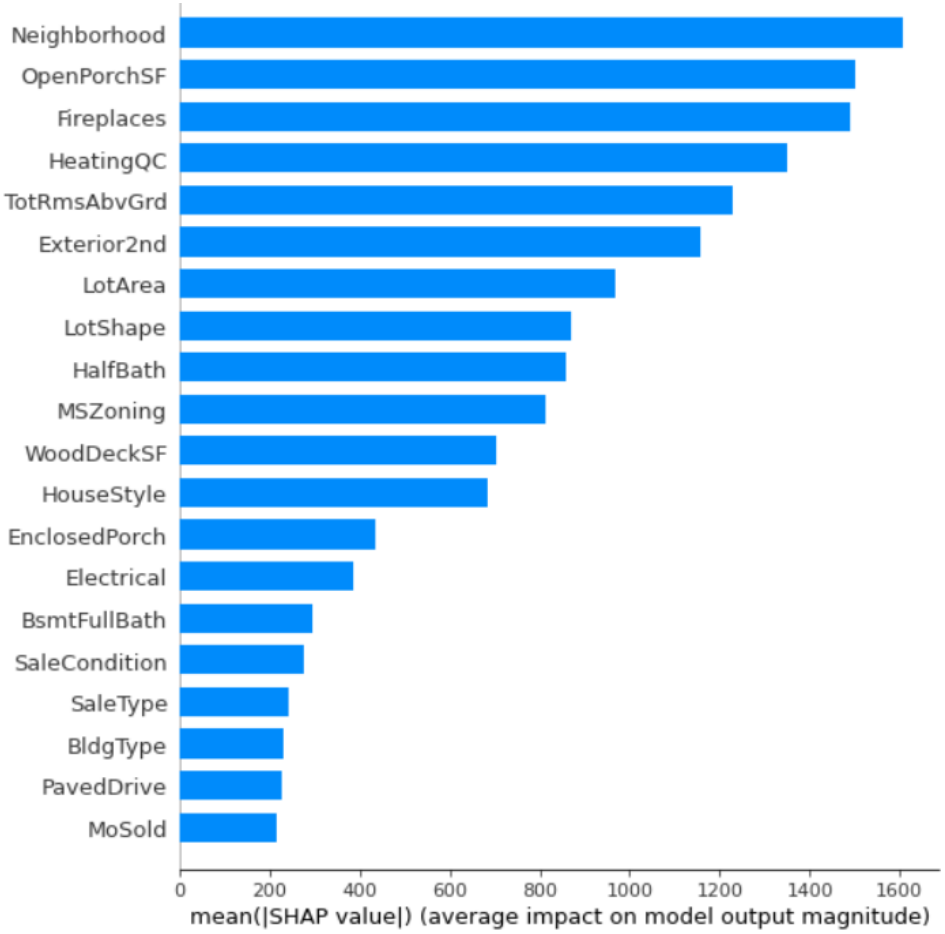


7.4. Drzewo regresji (Random Forest Regression)

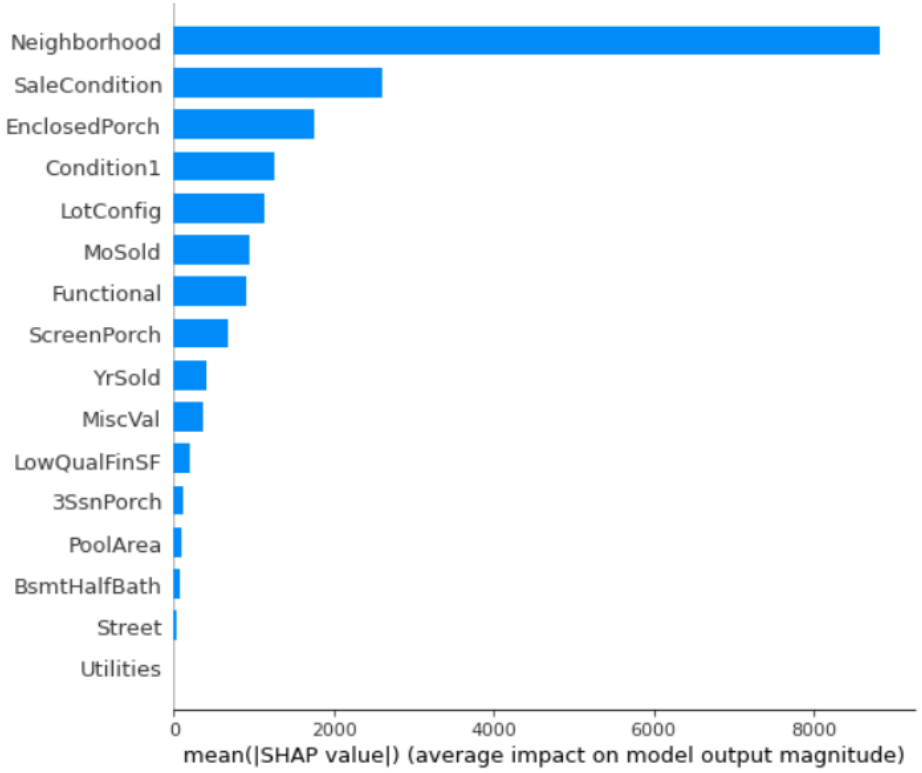
- *HousePricesPrediction_project.ipynb*



- *HousePricesPrediction_project_with_outliers_removed.ipynb*



- *HousePricesPrediction_project_same_value_removed.ipynb*



8. Podsumowanie

Predykcja cen nieruchomości jest możliwa z pomocą uczenia maszynowego i daje dobre rezultaty, jednak należy pamiętać, że tak duży zestaw cech jest bardzo trudny do prawidłowego przewidzenia. Ze wszystkich trzech podejść zdecydowanie najlepiej poradziło sobie to z wykorzystaniem drzewa decyzyjnego z parametrami uzyskanymi za pomocą metody GridSearch (czyli przemiatania po hiperparametrach) i to w każdym przypadku. Najlepsze podejście dało średnią absolutnych błędów względnych równą 8.69% (wynik jest zawyżany przez kilka skrajnie odstających wartości). Najlepsze wyniki uzyskano dla podstawowej wersji notebooka, tzn. tam, gdzie usunięto najwięcej kolumn zawierających wysokie korelacje.

Regularyzacja nie przyniosła widocznych poprawek, co może świadczyć o tym, że regresja liniowa sama w sobie radzi sobie "najlepiej jak potrafi". Usunięcie kolumn zawierających ponad 60% wystąpień tej samej wartości okazało się być lepszym podejściem niż odrzucenie skrajnych przypadków nieruchomości (takich o zbyt małych lub dużych wartościach). Jest to nieco zaskakująca obserwacja, jednak wyjaśnieniem na to może być występowanie tzw. outlierów w zbiorze testowym. Taka generalizacja w tym przypadku nie sprawdziła się.

Parametry modelu podane do przemiatania w GridSearchu można lepiej dopasować, jednak ze względu na złożoność obliczeniową tego procesu zależną od liczby wartości poszczególnych parametrów, listę parametrów ograniczono.