**Kacper Chmura  
Szymon Borowicz  
Jakub Frydlewicz**

Ames of City – analiza cen sprzedaży mieszkań

**Zadanie Projektowe**

Opiekun pracy:  
dr inż. Andrzej Paszkiewicz.

Rzeszów, 2023

Spis treści

[2. Badany zbiór danych 4](#_Toc135168328)

[3. Opis danych 6](#_Toc135168329)

[4. Wstęp 9](#_Toc135168330)

[5. Rozwiązanie problemu. 10](#_Toc135168331)

[5.1. Wykorzystane biblioteki 10](#_Toc135168332)

[5.2. Czyszczenie danych. 10](#_Toc135168333)

[5.3. Analiza za pomocą korelacji. 13](#_Toc135168334)

[6. Transformacja danych 15](#_Toc135168335)

[6.1. Usuwanie wartości odstających. 15](#_Toc135168336)

[6.2. OneHotEncoding oraz rozdzielanie danych. 18](#_Toc135168337)

[6.3. Walidacja krzyżowa w celu sprawdzenia możliwości modeli uczenia maszynowego. 19](#_Toc135168338)

[6.4. Dlaczego Walidacja krzyżowa? 19](#_Toc135168339)

[6.5. Wyniki walidacji krzyżowej 20](#_Toc135168340)

[7. Algorytmy uczenia maszynowego w predykcji cen domów 21](#_Toc135168341)

[7.1. Testowanie modeli przewidujących ceny mieszkań. 21](#_Toc135168342)

[7.2. Metryki oceny modelu. 22](#_Toc135168343)

[7.3. Ridge Regression 22](#_Toc135168344)

[7.3.1. Parametry 23](#_Toc135168345)

[7.3.2. Wyniki 24](#_Toc135168346)

[7.4. KNN Regression 24](#_Toc135168347)

[7.4.1. Parametry 25](#_Toc135168348)

[7.5. DecisionTreeRegressor 26](#_Toc135168349)

[7.5.1. Parametry 26](#_Toc135168350)

[7.5.2. Wyniki 27](#_Toc135168351)

[7.6. RandomForestRegressor 27](#_Toc135168352)

[7.6.1. Parametry 27](#_Toc135168353)

[7.6.2. Wyniki 28](#_Toc135168354)

[7.7. AdaptiveBoost 28](#_Toc135168355)

[7.7.1. Parametry 28](#_Toc135168356)

[7.7.2. Wyniki 29](#_Toc135168357)

[7.8. Gradient Boost 30](#_Toc135168358)

[7.8.1. Parametry 30](#_Toc135168359)

[7.8.2. Wyniki 31](#_Toc135168360)

[8. Wizualizacja z wykorzystaniem modeli uczenia maszynowego 31](#_Toc135168361)

[9. Analiza opłacalności poszczególnych parametrów. 34](#_Toc135168362)

[9.1. Dachy 34](#_Toc135168363)

[9.2. Typ forniru budowlanego 36](#_Toc135168364)

[9.3. Rodzaj fundamentu 36](#_Toc135168365)

[9.4. Ogrzewanie 37](#_Toc135168366)

[9.5. Kominek 38](#_Toc135168367)

[10. Podsumowanie 38](#_Toc135168368)

# Badany zbiór danych

Źródło danych do projektu zostało pobrane ze strony Kaggle.com.

(<https://www.kaggle.com/datasets/prevek18/ames-housing-dataset>).

Zbiór posiada 79 parametrów, które opisują prawie każdy aspekt domów mieszkalnych   
w mieście Ames – stan Iowa, a znajduje się w nim 2930 rekordów.

Poniżej znajduje się opis każdego z parametrów danych zawartych w pliku csv:

* SalePrice: cena sprzedaży nieruchomości w dolarach. Jest to zmienna docelowa, którą próbujesz przewidzieć.
* MSSubClass: Klasa budynku
* MSZoning: Ogólna klasyfikacja zagospodarowania przestrzennego
* LotFrontage: Liniowe stopy ulicy połączonej z nieruchomością
* LotArea: Wielkość działki w stopach kwadratowych
* Street: Rodzaj dostępu do drogi
* Alley: Typ dostępu do alei
* LotShape: Ogólny kształt nieruchomości
* LandContour: Płaskość nieruchomości
* Utilities: Rodzaj dostępnych mediów
* LotConfig: Konfiguracja działki
* LandSlope: Nachylenie nieruchomości
* Neighborhood: Fizyczne lokalizacje w granicach miasta Ames
* Condition1: Bliskość głównej drogi lub linii kolejowej
* Condition2: Bliskość głównej drogi lub linii kolejowej (jeśli druga jest obecna)
* BldgType: Typ mieszkania
* HouseStyle: Styl mieszkania
* OverallQual: Ogólna jakość materiału i wykończenia
* OverallCond: Ogólna ocena stanu technicznego
* YearBuilt: Oryginalna data budowy
* YearRemodAdd: Data przebudowy
* RoofStyle: Typ dachu
* RoofMatl: Materiał na dach
* Exterior1st: Zewnętrzne pokrycie domu
* Exterior2nd: Zewnętrzne pokrycie domu (jeżeli więcej niż jeden materiał)
* MasVnrType: Typ forniru murowanego
* MasVnrArea: Powierzchnia forniru murowanego w stopach kwadratowych
* ExterQual: Jakość materiału zewnętrznego
* ExterCond: Obecny stan materiału na zewnątrz
* Foundation: Rodzaj fundamentu
* BsmtQual: Wysokość piwnicy
* BsmtCond: Ogólny stan piwnicy
* BsmtExposure: Ściany piwnicy na poziomie Walkout lub ogrodu
* BsmtFinType1: Jakość wykończonej powierzchni piwnicy
* BsmtFinSF1: Typ 1 wykończonych stóp kwadratowych
* BsmtFinType2: Jakość drugiej wykończonej powierzchni (jeśli występuje)
* BsmtFinSF2: Typ 2 wykończonych stóp kwadratowych
* BsmtUnfSF: Niewykończone stopy kwadratowe powierzchni piwnicy
* TotalBsmtSF: Całkowite stopy kwadratowe powierzchni piwnicy
* Heating: Typ ogrzewania
* HeatingQC: Jakość i stan ogrzewania
* CentralAir: Centralna klimatyzacja
* Electrical: System elektryczny
* 1stFlrSF: Stopy kwadratowe pierwszego piętra
* 2ndFlrSF: Stopy kwadratowe drugiego piętra
* LowQualFinSF: Niska jakość wykończonych stóp kwadratowych (wszystkie piętra)
* GrLivArea: Powierzchnia mieszkalna nad poziomem gruntu stopy kwadratowe
* BsmtFullBath: Pełne łazienki w piwnicy
* BsmtHalfBath: Pół łazienki w piwnicy
* FullBath: Pełne łazienki powyżej poziomu
* HalfBath: Half Baths powyżej poziomu
* Bedroom: Liczba sypialni powyżej poziomu piwnicy
* Kitchen: Liczba kuchni
* KitchenQual: Jakość kuchni
* TotRmsAbvGrd: Całkowita liczba pokoi powyżej poziomu gruntu (nie obejmuje łazienek)
* Functional: Ocena funkcjonalności domu
* Fireplaces: Liczba kominków
* FireplaceQu: Jakość kominka
* GarageType: Lokalizacja garażu
* GarageYrBlt: Rok wybudowania garażu
* GarageFinish: Wykończenie wnętrza garażu
* GarageCars: Wielkość garażu w pojemności samochodów
* GarageArea: Wielkość garażu w stopach kwadratowych
* GarageQual: Jakość garażu
* GarageCond: Stan garażu
* PavedDrive: Wybrukowany podjazd
* WoodDeckSF: Powierzchnia drewnianych pokładów w stopach kwadratowych
* OpenPorchSF: Powierzchnia otwartego ganku w stopach kwadratowych
* EnclosedPorch: Powierzchnia zamkniętego ganku w stopach kwadratowych
* 3SsnPorch: Powierzchnia ganku z trzema sezonami w stopach kwadratowych
* ScreenPorch: Powierzchnia ganku z ekranem w stopach kwadratowych
* PoolArea: Powierzchnia basenu w stopach kwadratowych
* PoolQC: Jakość basenu
* Fence: Jakość ogrodzenia
* MiscFeature: Różnorodna cecha nie ujęta w innych kategoriach
* MoSold: Miesiąc sprzedaży
* YrSold: Rok Sprzedany
* SaleType: Typ sprzedaży.
* SaleCondition: Stan sprzedaży

# Opis danych

1. mssubclass: Określa rodzaj nieruchomości objętej sprzedażą.

* 20 jednokondygnacyjne z lat 1946 i nowsze - wszystkie style
* 30 jednokondygnacyjne z lat 1945 i wcześniejsze
* 40 jednokondygnacyjne z wykończonym strychem - wszystkie lata
* 45 jednoipółkondygnacyjne - nieukończone - wszystkie lata
* 50 jednoipółkondygnacyjne - ukończone - wszystkie lata
* 60 dwukondygnacyjne z lat 1946 i nowsze
* 70 dwukondygnacyjne z lat 1945 i wcześniejsze
* 75 dwupółkondygnacyjne - wszystkie lata
* 80 podzielone lub wielopoziomowe
* 85 split foyer
* 90 dwuodcinkowe - wszystkie style i lata
* 120 jednokondygnacyjne w osiedlu planowanym - z lat 1946 i nowsze
* 150 jednoipółkondygnacyjne w osiedlu planowanym - wszystkie lata
* 160 dwukondygnacyjne w osiedlu planowanym - z lat 1946 i nowsze
* 180 osiedle planowane - wielopoziomowe - zawiera split level/foyer
* 190 konwersja na dwie rodziny - wszystkie style i lata

1. Neighborhood: Fizyczne lokalizacje w granicach miasta Ames

* Blmngtn Bloomington Heights
* Blueste Bluestem
* BrDale Briardale
* BrkSide Brookside
* ClearCr Clear Creek
* CollgCr College Creek
* Crawfor Crawford
* Edwards Edwards
* Gilbert Gilbert
* IDOTRR Iowa DOT and Rail Road
* MeadowV Meadow Village
* Mitchel Mitchell
* Names North Ames
* NoRidge Northridge
* NPkVill Northpark Villa
* NridgHt Northridge Heights
* NWAmes Northwest Ames
* OldTown Old Town
* SWISU South & West of Iowa State University
* Sawyer Sawyer
* SawyerW Sawyer West
* Somerst Somerset
* StoneBr Stone Brook
* Timber Timberland
* Veenker Veenker

1. overallqual: Ocena ogólna materiałów i wykończenia domu

* 10 Bardzo doskonałe
* 9 Doskonałe
* 8 Bardzo dobre
* 7 Dobre
* 6 Ponad przeciętne
* 5 Przeciętne
* 4 Poniżej przeciętne
* 3 Słabe
* 2 Złe
* Bardzo złe

1. OverallCond: Ocenia ogólny stan domu

* 10 Bardzo doskonałe
* 9 Doskonałe
* 8 Bardzo dobre
* 7 Dobre
* 6 Ponad przeciętne
* 5 Przeciętne
* 4 Poniżej przeciętne
* 3 Słabe
* 2 Złe
* 1 Bardzo złe

1. roofmatl: Materiał pokrycia dachowego

* ClyTile Glina lub dachówka
* CompShg Standardowa (kompozytowa) gontowa
* Membran Membrana
* Metal Metalowy
* Roll Rola
* Tar&Grv Żwir i smoła
* WdShake Drewniane strzechy
* WdShngl Drewniane gonty

1. masvnrtype: Rodzaj okładziny murarskiej

* BrkCmn Zwykła cegła
* BrkFace Cegła elewacyjna
* CBlock Blok betonowy
* None Brak okładziny
* Stone Kamień

1. heating: Typ ogrzewania

* Floor Piece podłogowe
* GasA Gazowy piec z wymuszonym obiegiem powietrza
* GasW Gazowy piec na gorącą wodę lub parę
* Grav Grawitacyjny piec
* OthW Ogrzewanie gorącą wodą lub parą innym niż gaz
* Wall Piece ścienne

# Wstęp

W projekcie skupimy się na cenach domów w mieście Ames, a konkretnie na tym jak poszczególne parametry wpłyną na ceny domów. Dane do projektu zostały pobrane ze znanej platformy - stosowanej przez analityków "kaggle.com". Zawierają one dogłębny opis każdego domu w postaci różnego rodzaju parametrów. Naszym celem będzie wyselekcjonowanie kilku najważniejszych, które mogłyby posłużyć deweloperowi bądź innej firmie zajmującej się remontem nieruchomości zwiększyć cenę domu. Przed analizą konieczna będzie odpowiednia obróbka danych obejmująca usunięcie braków danych,   
a także zamienienie części z nich na format liczbowy do czego wykorzystamy OneHotEncoding. Projekt zostanie oparty głównie na modelach uczenia maszynowego, lecz niewielką część poświęcimy wskaźnikom korelacji. W jego trakcie zastosujemy różnego rodzaju modele od liniowych przez drzewa decyzyjne na lasach kończąc. Do wyboru odpowiednich modeli posłuży nam walidacja krzyżowa, która w najbardziej miarodajny sposób określi, które modele są warte dalszego rozwijania. Każdy model zostanie odpowiednio dostrojony poprzez zmianę hyperparametów, w których doborze posłuży nam funkcja GridSearchCV, która testuje każdą możliwą kombinację zadanych parametrów oraz zwraca model z zastosowaną najlepsza kombinacją parametrów. Po odpowiednim dostrojeniu za pomocą metryk takich jak MeanAbsoluteError, RootMeanSquaredError oraz R2Score zostanie wybrany najlepszy model, który posłuży do dalszej analizy parametrów wpływających na cenę domu. Za pomocą wybranego modelu dokonana zostanie analiza poszczególnych cech, które w znaczący sposób wpływają na cenę domu oraz są stosunkowo proste do zastosowania przez firmę budowlaną bądź remontową. Częścią projektu będą również wizualizacje, których część będzie wykorzystywała model uczenia maszynowego, konkretnie zwizualizuje wpływ poprawy parametru odpowiedzialnego za stan mieszkania na średnią cenę mieszkań w dzielnicy. Na koniec zostanie sformułowane podsumowanie zawierające wskazówki dla firm, które cechy najbardziej wpływają na cenę domu.

# Rozwiązanie problemu.

## Wykorzystane biblioteki

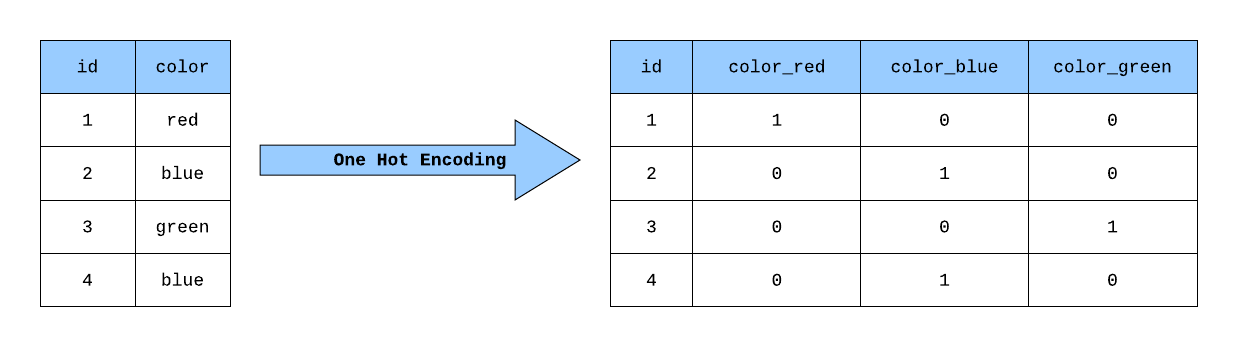
* Json – umożliwia kodowanie i dekodowanie danych w formacie JSON.
* Numpy - zapewnia wsparcie dla operacji numerycznych i matematycznych,
* Pandas – posiada narzędzia do manipulacji oraz analizy danych,
* Matplotlib, plotly, seaborn – biblioteki umożliwiające tworzenie wizualizacji danych,
* Sklearn – dostarcza narzędzia do uczenia maszynowego

## Czyszczenie danych.

Zbiór danych był dosyć dobrze przygotowany do dalszej analizy i jedynym problemem dotyczącym czyszczenia danych okazało się zniwelowanie

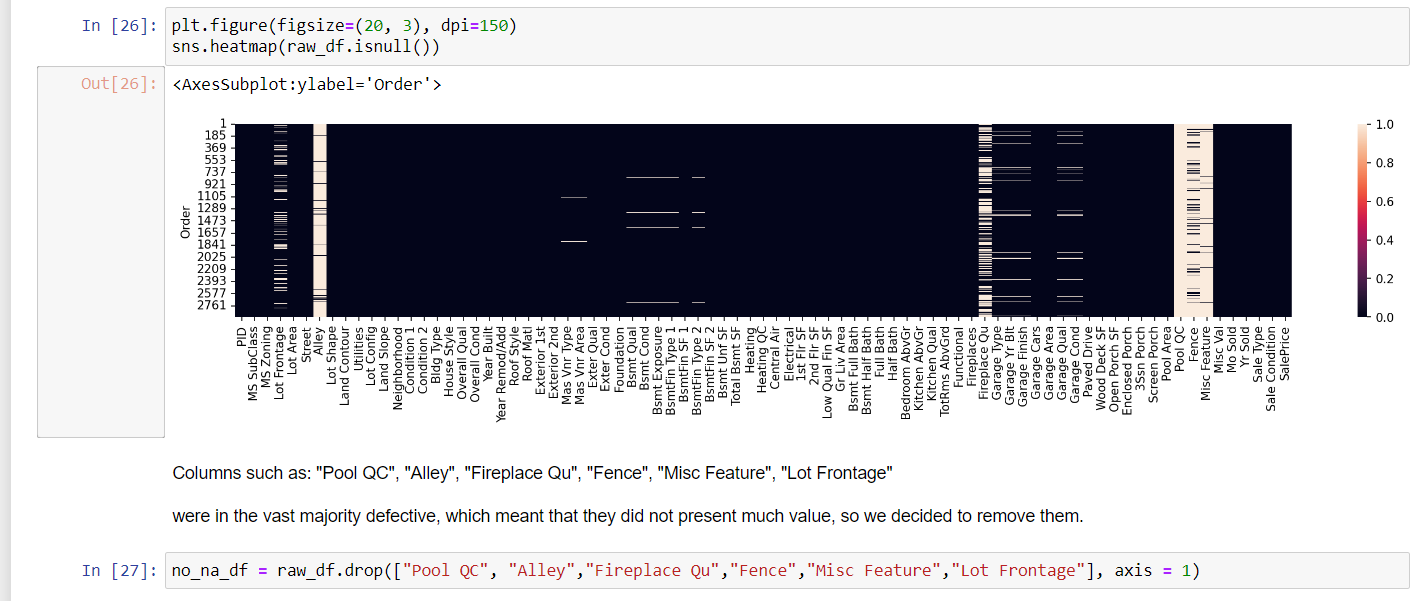
wartości wybrakowanych (NA). Wartym odnotowanie jest fakt, że dane tekstowe ze zbioru danych zostaną zamienione na liczbowe za pomocą OneHotEncoding.

Aby modele uczenia maszynowego działały prawidłowo, wprowadzone do nich dane powinny być w odpowiednim formacie liczbowym tj. int, float, etc. Dlatego też zmienne zapisane w naszym zbiorze danych jako łańcuchy znaków (string) zostały zmienione na liczbowe za pomocą funkcji z pakietu pandas o nazwie get\_dummies, wykorzystującą metodę OneHotEncoding, która służy do przekształcenia kolumn z kategoriami na zestaw binarnych kolumn, gdzie każda kolumna reprezentuje pojedynczą kategorię, a wartości 1 wskazują przynależność do danej kategorii. Cały proces został przedstawiony na poniższym przykładzie:



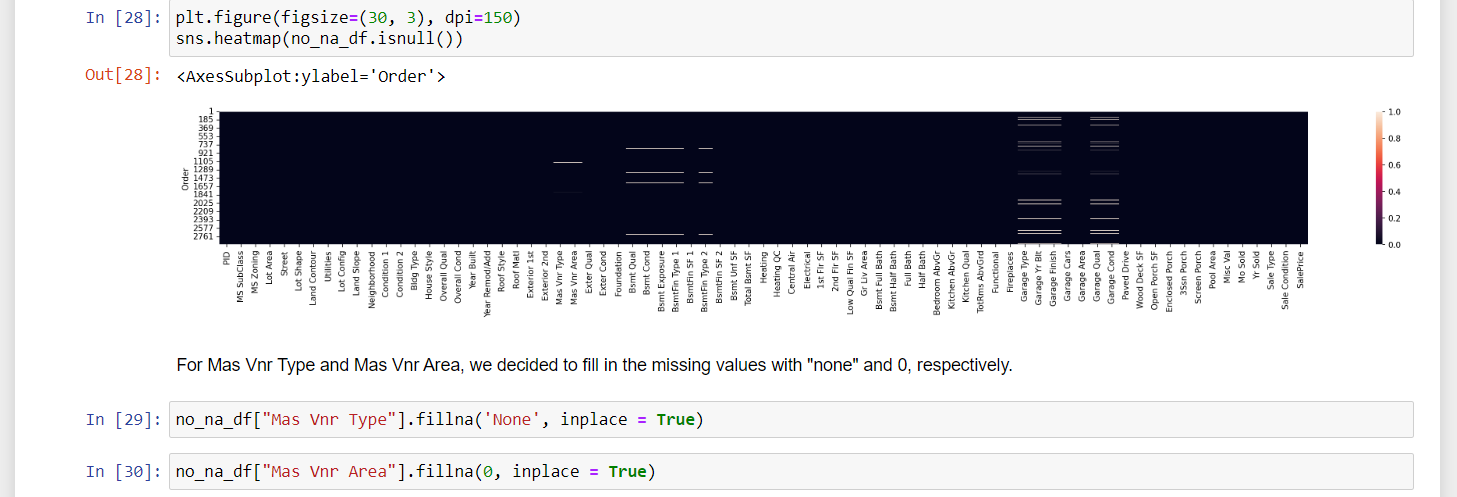
Rys. 4.1 Przykład działania OneHotEncoding

Zbiór danych, był w niektórych miejscach wybrakowany, tzn. część rekordów posiadała wartości *NaN*. Był to jeden z powodów, dla których dane musiały zostać wyczyszczone. Cały proces został przedstawiony poniżej.



Rys. 4.2 Wybrakowane dane

Kolumny takie jak „POOL QC”, „Alley”, „Fireplace Qu”, „Fence”, “Misc Feature”, “Lot Frontage” posiadały niemal w pełni wybrakowane dane. Dlatego też postanowiliśmy je usunąć, ponieważ nie niosły ze sobą żadnej wartości.



Rys. 4.3 Wybrakowane dane

Brakujące wartości w kolumnie „Mas Vnr Type” zostały zastąpione wartością NaN, ponieważ takie oznaczenie zostało domyślnie przyjęte w opisie danych. „Mas Vnr Area” została wypełniona zerami, gdyż w przypadku braku wartości, algorytmy uczenia maszynowego nie są w stanie poprawnie działać.

 Rys. 4.4 Wybrakowane dane

Braki w takich kolumnach jak “Bsmt Qual”, „Bsmt Cond”, itd. zostały uzupełnione wartością „Na”, ponieważ taka konwencja została przyjęta w opisie danych.

Konwencja z opisu danych:

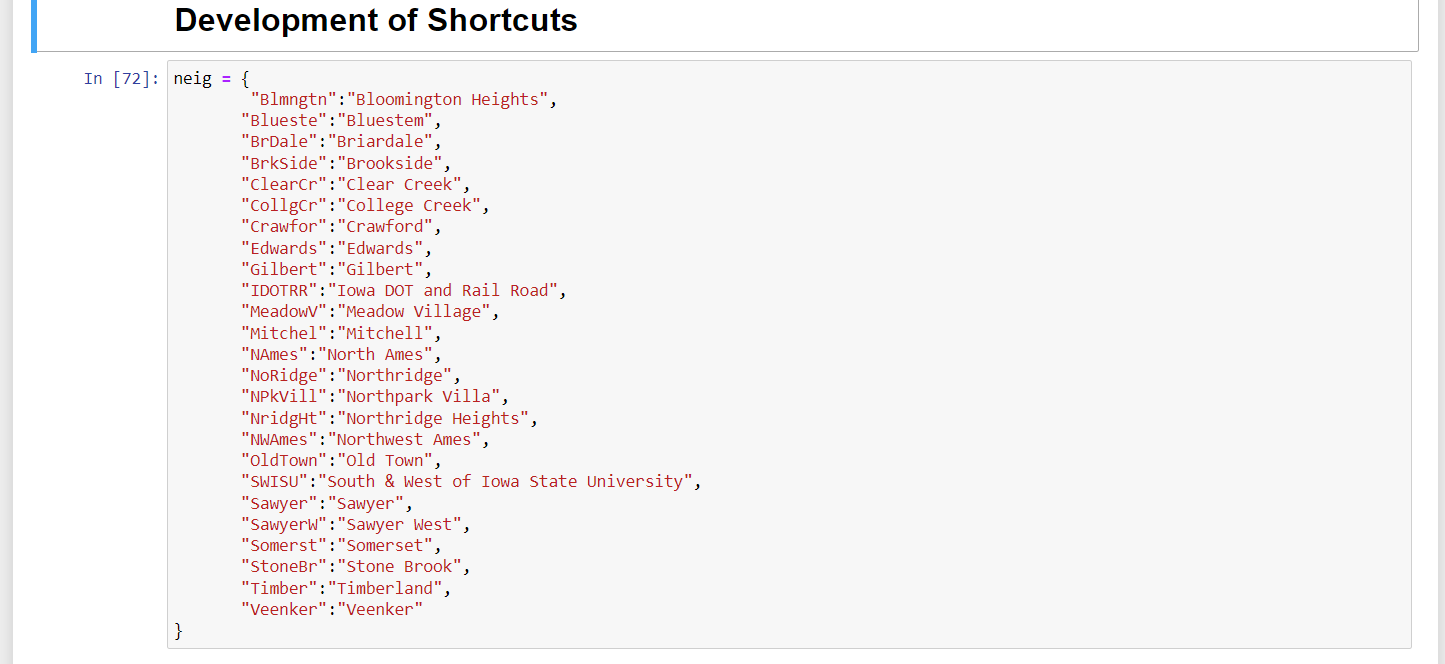
* Ex - Excellent
* Gd - Good
* TA - Typical
* Fa - Fair
* Po – Poor

NA.

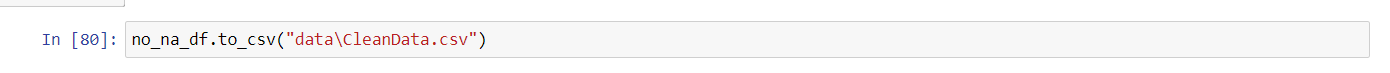
Według tej konwencji, konkretne skróty takie jak „Ex”, „Gd”, itd. opisują taki parametr jak na przykład stan jakości piwnicy, z tym, że w zbiorze danych wszystkie wartości poza „NA” są zapisane w formacie łańcuchu tekstu, natomiast NA jest zapisane   
w formacie *NAN*. Dlatego też dla ujednolicenia zapisu danych, przyjęte zostało, że wartości NAN zostają zastąpione na „Na”.

 Rys. 4.5 Niewybrakowane dane

Dzięki zastosowanym wcześniej zabiegom, zbiór danych nie posiada już żadnych wybrakowanych wartości.

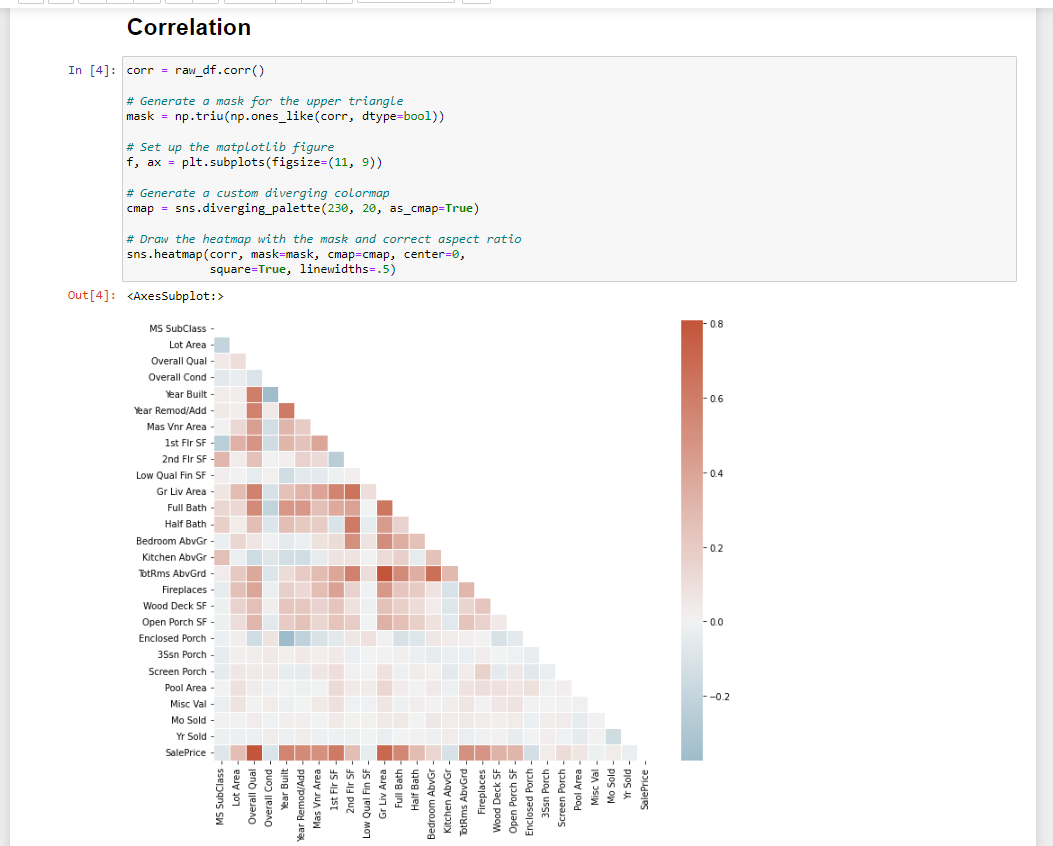
 Rys. 4.6 Pełne nazwy dzielnic

Skrótowe nazwy dzielnic zostały zastąpione pełnymi nazwami, które bardziej intuicyjnie prezentują się podczas wizualizacji, aniżeli nazwy skrótowe.

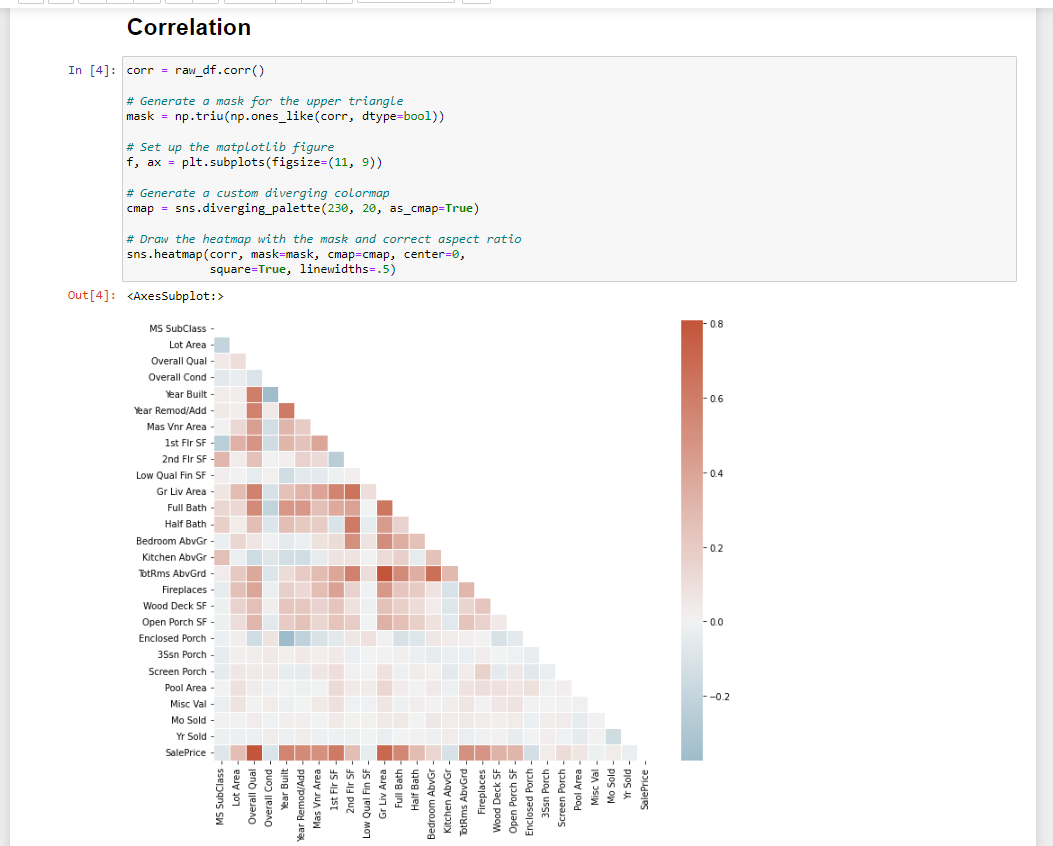


Rys. 4.7 Podpis zapisanych danych

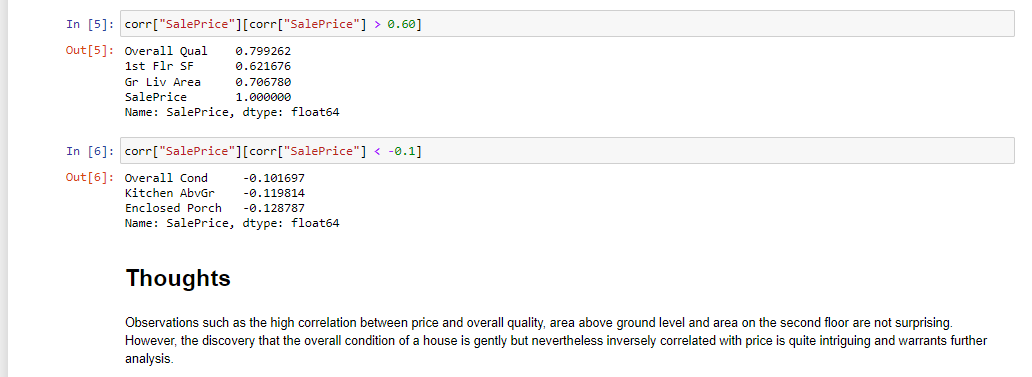
## Analiza za pomocą korelacji.

 Rys. 4.8 Wykres korelacji pomiędzy zmiennymi

Wykres „heatmap” wygenerowany za pomocą bibliotek seaborn oraz numPy przedstawia wskaźniki korelacji pomiędzy konkretnymi parametrami, w tym najbardziej istotnymi między parametrami, a ceną.



Rys. 4.9 Korelacje pomiędzy zmiennymi a ceną

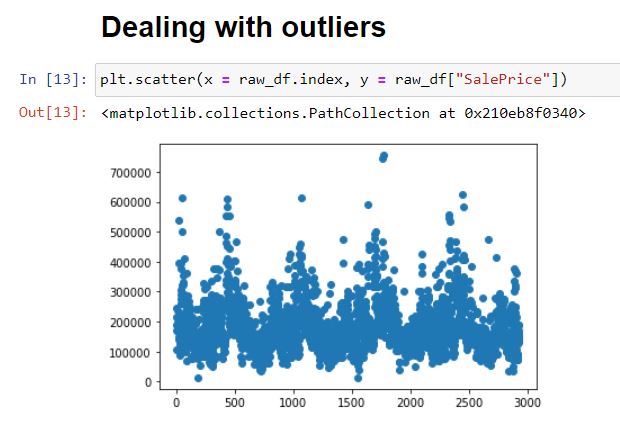
Rys. 4.10 Wybranie najsilniej skorelowanych zmiennych

Poza tak oczywistymi obserwacjami takimi jak wysoka zależność pomiędzy ceną,   
a ogólną jakością domu („Overall Qual”), powierzchnią pierwszego piętra („1st Fir SF”) oraz powierzchnią do życia („Gr Liv Area”), można zaobserwować wzorce takie jak odwrotna korelacja ceny do stanu domu („Overall Cond”) wynosząca ok.   
-0,1. Wywnioskować można również, że kuchnia na piętrze oraz zamknięta weranda nie są pożądane na rynku nieruchomości, ponieważ mają negatywny wpływ na cenę.

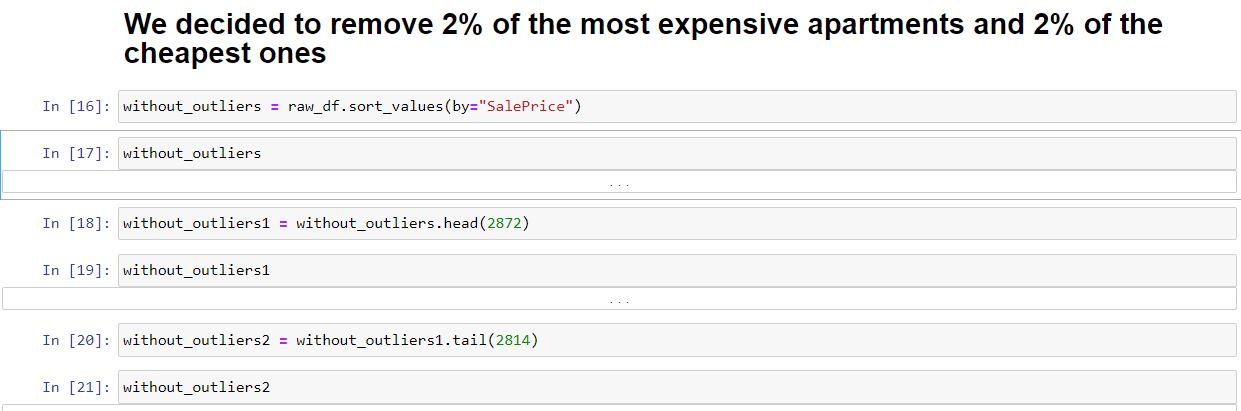
# Transformacja danych

## Usuwanie wartości odstających.

Zdecydowaliśmy się dla poprawienia skuteczności naszej analizy wyrzucić skrajne wartości z naszego zbioru danych. Konkretnie 2% najdroższych mieszkań oraz głównie dla zachowania balansu 2% najtańszych. Zbiór przedstawiający ceny mieszkań z naszego zbioru danych przedstawia wykres poniżej. Możemy na nim zaobserwować, że co najmniej kilka/kilkadziesiąt wartości zdecydowanie odstaje od pozostały. Z tego powodu postanowiliśmy nie uwzględniać ich w naszej analizie.

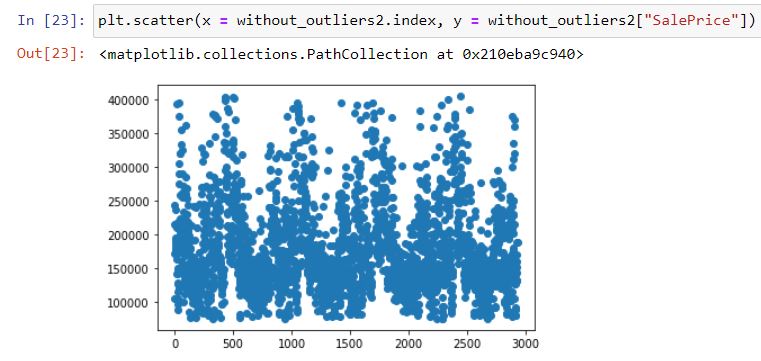


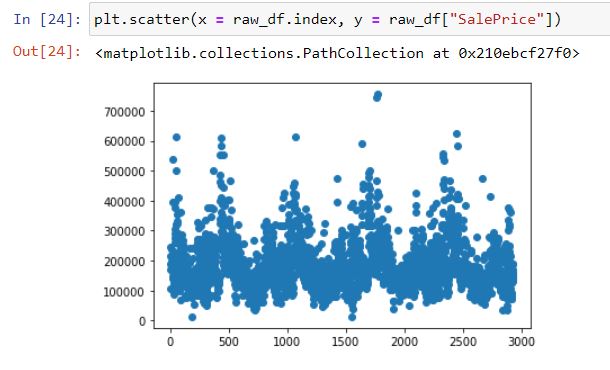
Rys. 5.1 Ceny wszystkich domów



Rys. 5.2 Usuwanie wartości odstających

Różnice między zbiorami możemy zaobserwować na poniższych wykresach:

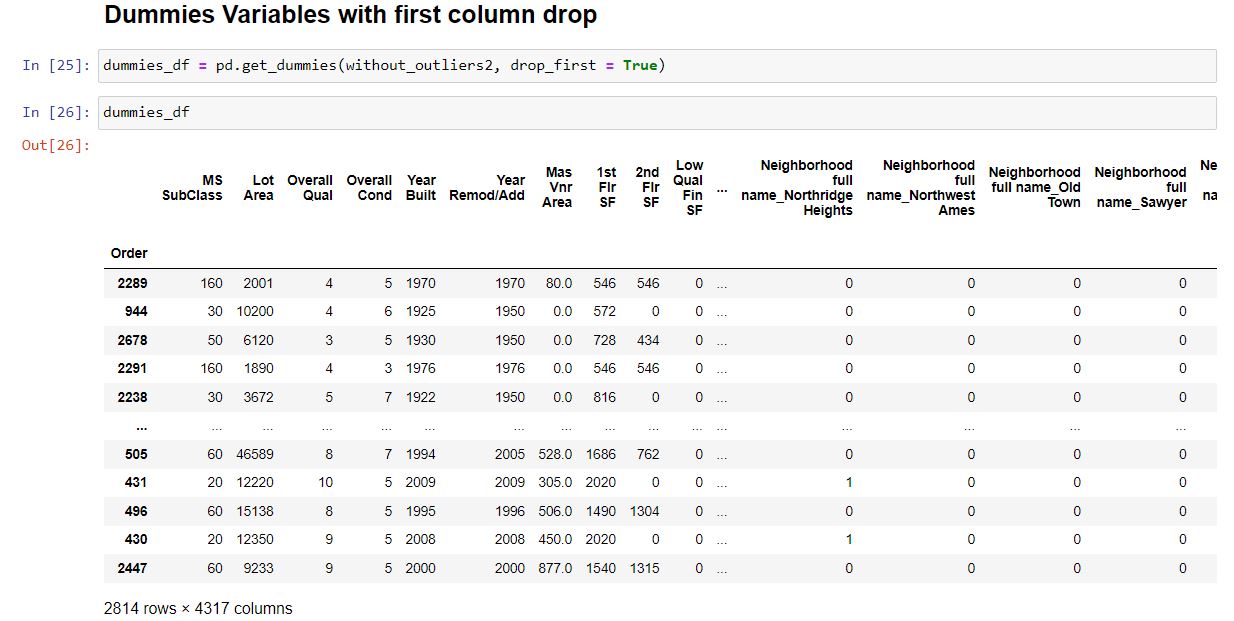
 Rys. 5.3 Dane bez wartości odstających



Rys. 5.4 Dane z wartościami odstającymi

## OneHotEncoding oraz rozdzielanie danych.

Do poprawnego funkcjonowania modele uczenia maszynowego wymagają danych zapisanych w formie liczbowym (int, float, etc.) dlatego też zmienne zapisane w naszym zbiorze danych jako napisy(string) zostały zamienione na liczbowe za pomocą funkcji   
z pakietu pandas o nazwie get\_dummies.

 Rys. 5.5 Dane po transformacji OneHotEncoding

Dane po odpowiedniej zmianie typu muszą zostać rozdzielone na odpowiednie zbiory. W pierwsze kolejności zostają one przypisane do zmiennych y oraz X, gdzie y jest etykietą (Ceną mieszkania) tzn. wartością, którą ma nasz model przewidywać. Natomiast   
X jest wszystkim innym czyli zmiennymi na podstawie, których nasz przyszły model ma przewidzieć cenę mieszkania. Kolejnym krokiem jest rozdzielenie powstałych w ten sposób zbiorów na treningowe oraz testowe. Zbiory treningowe X\_train oraz y\_train posłużą do uczenia modeli uczenia maszynowego, natomiast zbiory testowe X\_test oraz y\_test do sprawdzania wyników naszego modelu. Cały podział odbywa się za pomocą funkcji train\_test\_split z pakietu sklearn. Funkcja przyjmuje rozdzielone zmienne X,y oraz parametry takie jak test\_size odpowiadajacy za rozmiar zbioru testowego (w naszym przypadku 20%) oraz random\_state jest swego rodzajem ziarnem służącym do wyboru, które wiesze mają trafić do zbioru testowego, a które do uczącego. Random\_state przyjmuje liczbę całkowitą jako wartość ziarna według, której konkretne wiersze trafiają do danego podzbioru test lub train.

## Walidacja krzyżowa w celu sprawdzenia możliwości modeli uczenia maszynowego.

Za pomocą walidacji krzyżowej przetesowaliśmy takie modele jak Support Vector Machine Regressor, Linear Regression, KNN Regressor, Decision Tree Regresor oraz Random Forest Regressor. Walidacja krzyżowa daje bardziej miarodajne wyniki niż zwykłe testowanie modeli, ponieważ jako zbiór treningowy wykorzystuje wszystkie dane.   
W naszym przypadku wykorzystaliśmy podział cv = 10, co oznacza że każdy zbiór został podzielony na 10 części. Następnie dany model uczył się na 9 częściach oraz sprawdzany na tej ostatniej nie użytej do nauki, wynik uzyskany został zapisany, a cały proces został powtórzony jeszcze 9 razy za każdym razem ze zmianą części użytej do sprawdzenia. Po 10 takich iteracjach została wyciągnięta średnia jak wynik całego procesu.

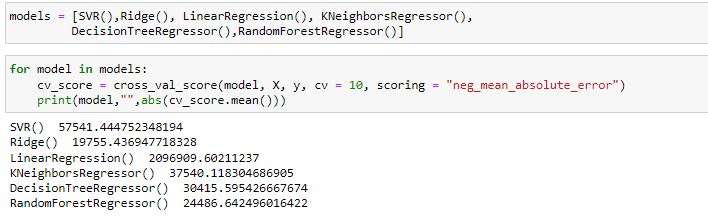
## Dlaczego Walidacja krzyżowa?

Walidacja krzyżowa (cross-validation) jest lepsza od zwykłego testowania modeli z kilku powodów:

* Lepsze wykorzystanie danych: Walidacja krzyżowa pozwala na bardziej efektywne wykorzystanie dostępnych danych. Zamiast podzielić zestaw danych na jedną próbkę testową i treningową, walidacja krzyżowa dzieli je na kilka podzbiorów, z których jeden jest używany jako zbiór testowy, a pozostałe jako zbiory treningowe. Dzięki temu każda próbka danych jest wykorzystywana zarówno do treningu, jak i do testowania modelu, co prowadzi do bardziej wiarygodnych wyników.
* Wiarygodniejsza ocena modelu: Walidacja krzyżowa zapewnia bardziej wiarygodną ocenę jakości modelu. Podczas zwykłego testowania, wynik może być uzależniony od przypadkowego podziału danych na zbiory testowe i treningowe. W przypadku walidacji krzyżowej, wynik jest uśredniany z kilku prób, gdzie każda próba używa innego podziału danych. Pozwala to na bardziej obiektywne oszacowanie skuteczności modelu.
* Lepsze wykrywanie przeuczenia (overfittingu): Walidacja krzyżowa jest skuteczniejsza w wykrywaniu przeuczenia modelu. Przeuczenie występuje, gdy model jest zbyt mocno dopasowany do danych treningowych i słabo generalizuje na nowe dane. Przy użyciu walidacji krzyżowej można wykryć ten problem poprzez porównanie wyników na różnych podzbiorach danych. Jeśli model jest przeuczony, to wyniki na zbiorze testowym będą gorsze niż na zbiorze treningowym.
* Lepszy podział danych treningowych i testowych: Walidacja krzyżowa pozwala na lepszy podział danych treningowych i testowych. W zwykłym testowaniu często zdarza się, że niektóre ważne wzorce mogą być pomijane przez przypadek, gdy zostaną one umieszczone w zbiorze testowym. Walidacja krzyżowa pozwala na uwzględnienie wszystkich próbek danych zarówno w zbiorze treningowym, jak   
  i testowym w różnych kombinacjach, co pozwala na bardziej kompleksową ocenę modelu.
* Lepsza reprezentatywność wyników: Walidacja krzyżowa pozwala na uzyskanie bardziej reprezentatywnych wyników. Dzięki uśrednianiu wyników z wielu prób, możemy uzyskać bardziej stabilne i wiarygodne oszacowanie skuteczności modelu. Wyniki z jednej próby mogą być podatne na losowe fluktuacje, ale uśrednione wyniki z wielu prób są bardziej niezawodne.

## Wyniki walidacji krzyżowej

Wartym odnotowania jest fakt że modele użyte przez na w teście walidacji krzyżowej był testowane bez jakichkolwiek zmian hyperparametrów. Walidacja krzyżowa pokazała, które modele najbardziej nadają się do dalszego rozwój. Są to Ridge, KNeighborsRegressor, DecisionTreeRegressor oraz RandomForestRegressor, odrzuciliśmy dwa modele liniowe takie jak SVR oraz LinearRegression. Nie oznacza to jedna że modele liniowe nie sprawdziły się w naszym teście, ponieważ doskonale wypadł model będącym usprawnioną regresją liniową czyli Ridge Regression.

 Rys. 4.6 Wyniki modeli podczas testu walidacją krzyżową

# Algorytmy uczenia maszynowego w predykcji cen domów

## Testowanie modeli przewidujących ceny mieszkań.

W celu porównania modeli uczenia maszynowego stworzona została funkcja "ml", przyjmująca argumenty takie jak:

* df - ramka danych potrzebnych do nauki modelu
* model - instancja modelu który zostanie użyty
* grid - zmienna która decyduje czy użyć GridSearchCV sprawdzającej każdą kombinacje zadanych parametrów w kolejnym argumencie grid\_params
* grid\_params domyślnie przyjmująca zero służy głównie do przechowywania słownika potrzebnego funkcji GridSearchCV do znalezienia odpowiedniej kombinacji parametrów modelu.

Funkcja na początku dzieli lokalnie zbiór danych na podzbiory za pomocą train\_test\_split, następnie sprawdza czy użytkownik chce użyć GridSearchCV czy też nie za pomocą zmiennej Grid, która z kolei przyjmuje wartości True lub False. W dalszej części model uczy się na zbiorach treningowych, później przywiduje wynik na podstawie zbioru testowego. Żeby w późniejszej części porównać go z prawidłowymi wynikami za pomocą takich metryk jak MAE, RMSE oraz R2 Score. Wyniki są również wyświetlane na wykresie, który demonstruje jak bardzo predykcje modelu odbiegają od prawidłowych wartości. Na sam koniec jeśli użytkownik wybrał, że chce skorzystać z GridSearchCV wyświetlona zostaje najlepsza kombinacja parametrów. W dalszej części zaprezentowane zostanie działanie funkcji w praktyce.

 Rys. 6.1 Funkcja testująca model

## Metryki oceny modelu.

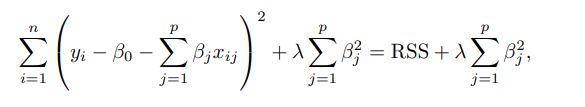
1. MAE (Mean Absolute Error) i RMSE (Root Mean Squared Error) to popularne metryki używane w problemach regresji do oceny jakości predykcji modelu.MAE mierzy średnią wartość bezwzględną błędów predykcji modelu, czyli średnią różnicę pomiędzy wartością rzeczywistą a wartością przewidywaną. MAE jest łatwy do interpretacji i nie wrażliwy na wartości odstające.
2. RMSE, podobnie jak MAE, mierzy błędy predykcji, ale jest to średnia kwadratowa różnica pomiędzy wartością rzeczywistą a wartością przewidywaną. Ponieważ RMSE bierze pod uwagę kwadrat błędów, to większe błędy są bardziej karane niż w przypadku MAE. RMSE jest często stosowany, gdy chcemy zwrócić uwagę na duże błędy predykcji.
3. R2 Score (lub R-squared) to metryka, która mierzy dopasowanie modelu do danych. R2 Score jest liczbą z zakresu od 0 do 1, gdzie wartość 1 oznacza idealne dopasowanie modelu do danych, a wartość 0 oznacza, że model nie wyjaśnia zmienności danych   
   w ogóle. R2 Score wyraża proporcję zmienności danych, która jest wyjaśniona przez model. W praktyce, R2 Score jest używany jako miara wydajności modelu, która pokazuje, jak dobrze model przewiduje zmienność danych. Wartości R2 Score powyżej 0,7 zazwyczaj uważane są za dobre dopasowanie modelu.

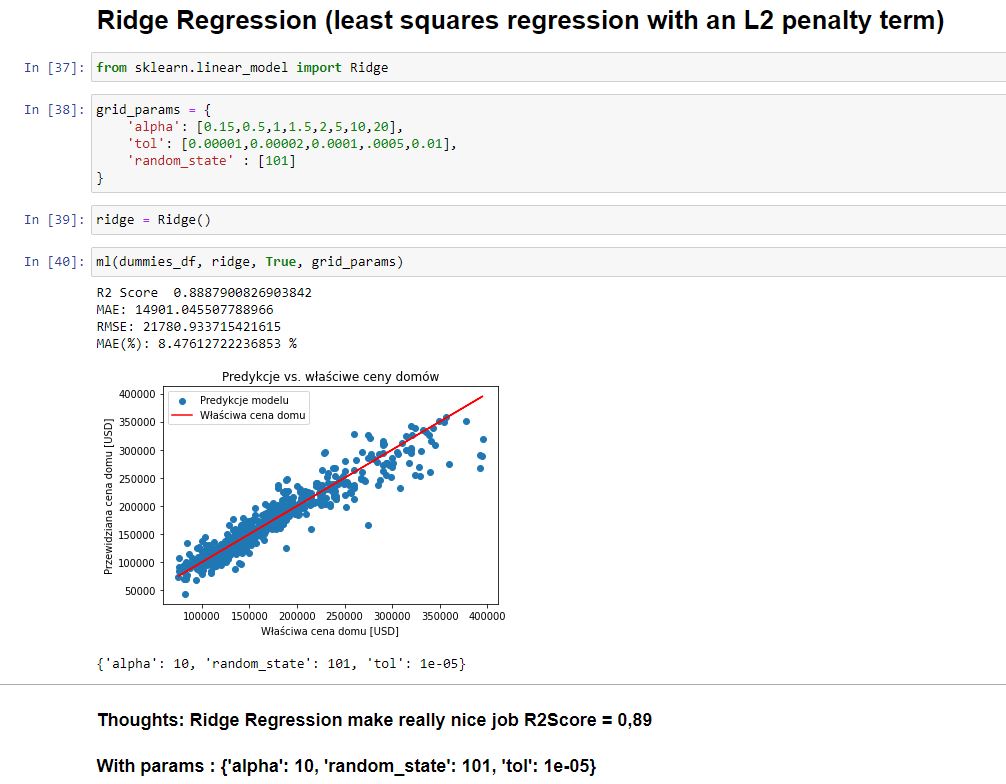
## Ridge Regression

Ridge Regression jest rozszerzeniem regresji liniowej wprowadzającym dodatkową regularyzajce L2. Dzięki regularyzacji L2, Ridge Regression daje elastyczność   
w dostosowywaniu modelu do danych treningowych i jednocześnie pomaga w unikaniu zbytniego dopasowania.

W tradycyjnej regresji liniowej dąży się do minimalizacji sumy kwadratów różnic między przewidywanymi wartościami a rzeczywistymi wartościami. Ridge Regression dodaje do tego celu dodatkowy składnik, który jest proporcjonalny do kwadratu normy L2 współczynników regresji. Kluczowym parametrem dla regresji ridge jest parametr alfa wyznaczający to jak wysoka będzie kara za złożoność. Im większa wartość alpha, tym większa kara za złożoność, co prowadzi do bardziej ograniczonych wartości wag. To ograniczenie wartości wag ma na celu zmniejszenie wpływu poszczególnych cech na model, zapobieganie zbytniemu dopasowaniu i poprawę ogólnej zdolności do generalizacji na nowych danych.

Dodanie regularyzacji L2 pomaga w redukcji przeuczenia (overfittingu) i poprawie stabilności modelu, szczególnie gdy mamy do czynienia z dużą liczbą cech lub kiedy cechy są silnie skorelowane. Regularyzacja L2 działa poprzez równoczesne minimalizowanie błędu dopasowania danych treningowych oraz ograniczanie wartości wag, co pozwala na uzyskanie bardziej wyważonego modelu.

 Rys. 5.2 Wzór na regularyzację L2

 Rys. 5.3 Kod użyty do stworzenia modelu Ridge

### Parametry

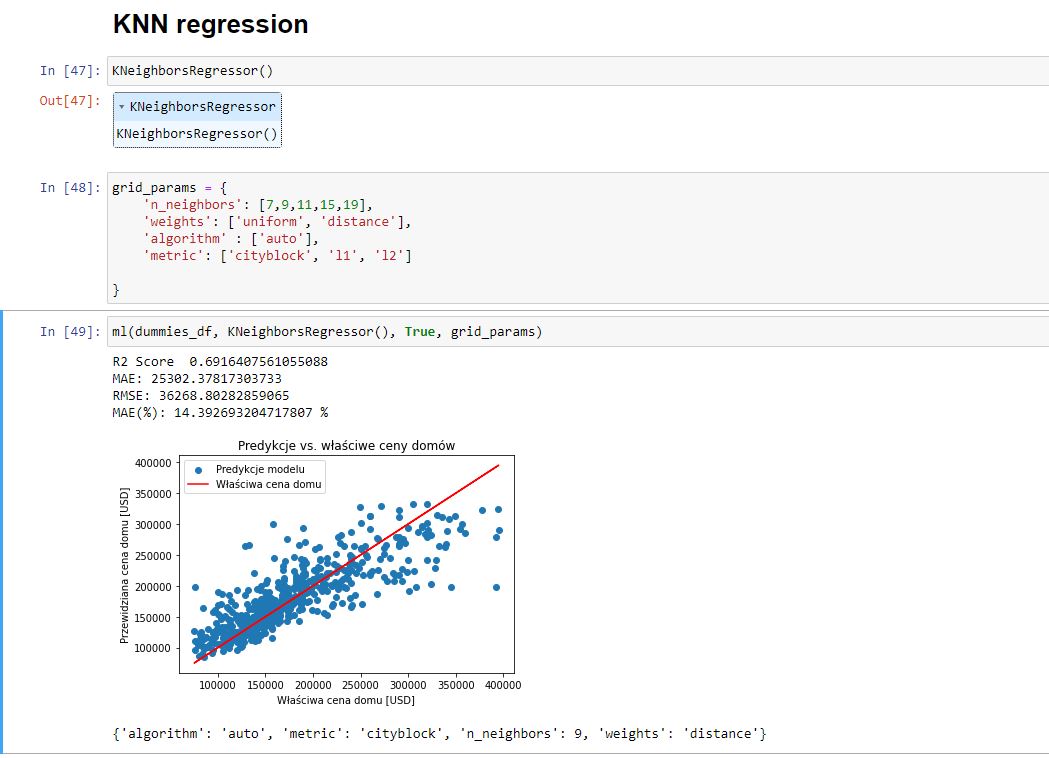
* Parametr "tol" w modelu Ridge Regression jest wartością progową, która kontroluje warunek zbieżności algorytmu optymalizacji podczas uczenia modelu. Im mniejsza wartość tol, tym bardziej precyzyjne jest kryterium zbieżności, ale wymaga to więcej iteracji. Warto dostosować tol, aby znaleźć odpowiedni kompromis między czasem uczenia a precyzją wyników.
* Parametr "alfa" w Ridge Regression kontroluje siłę regularyzacji, czyli wpływ kary za złożoność modelu. Większa wartość alfa prowadzi do większej kary i bardziej ograniczonych wag, co pomaga w zapobieganiu zbytniemu dopasowaniu. Optymalne dobranie wartości alfa jest ważne dla zachowania równowagi między dopasowaniem a regularyzacją.

### Wyniki

Wybrane zostały następujące parametry: 'alpha': 10, 'random\_state': 101, 'tol': 1e-05. Wysoki poziom parametru alfa, może świadczyć o sporej liczbie zmiennych nie wnoszących zbyt wiele do predykcji ceny mieszkania. Dla przypomnienia zbiór posiada ponad cztery tysiące zmiennych.

Wynik jaki uzyskał model Ridge okazał się zaskakująco dobry ponieważ R2 Score wyniósł niespełna 0.89 co przełożyło się między innymi na średni błąd poniżej 8.5%. Różnica pomiędzy modelami Linear Regression, a Ridge również może świadczyć, że ograniczenie złożoności modelu pozytywnie wpływa na wynik. Co potwierdza hipotezę, że spora część zmiennych ze zbioru danych jest niezbyt użyteczna. Wartym odnotowania jest fakt, że model dobrze radzi sobie z przewidywaniem ceny mieszkań z dowolnej półki cenowej co prezentuje wykres Predykcje vs. właściwe ceny domów.

## KNN Regression

Rys. 5.4 Kod użyty do stworzenia modelu KNN Regression

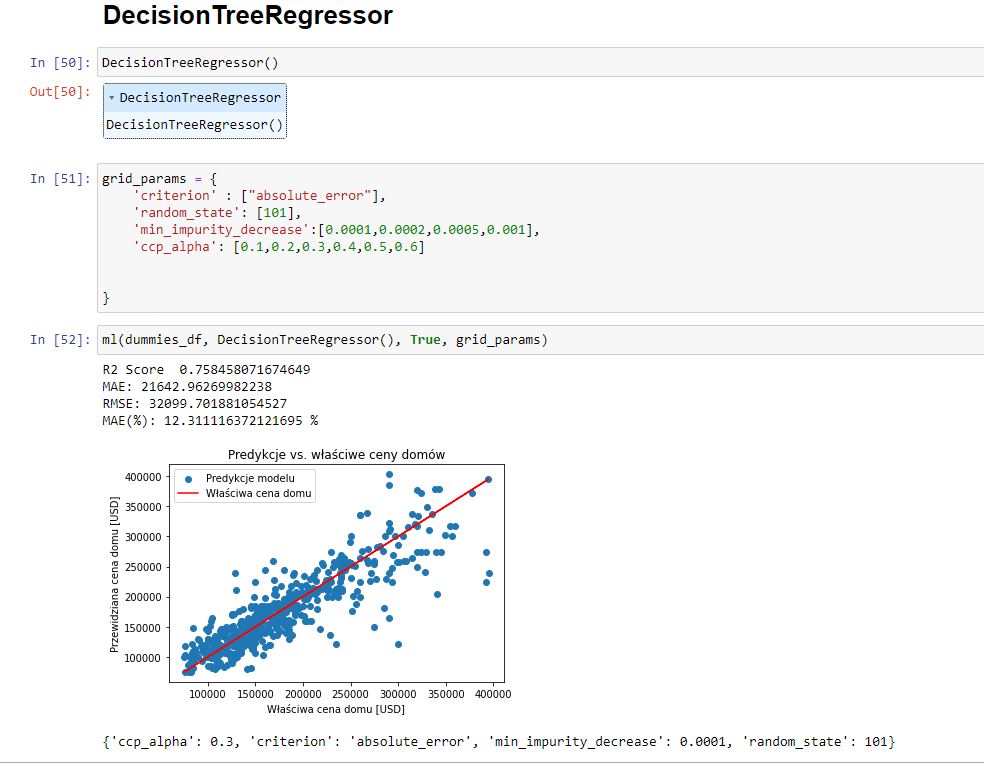
### Parametry

* Parametr "n\_neighbors" w KNN Regressor służy do określenia liczby sąsiadów branych pod uwagę podczas prognozowania wartości docelowych dla nowych obserwacji. Jest to jeden z kluczowych parametrów w algorytmie k-najbliższych sąsiadów (KNN), gdzie obiekt jest przewidywany na podstawie podobieństwa do najbliższych sąsiadów w zbiorze treningowym. Wybór optymalnej wartości n\_neighbors może mieć wpływ na wydajność i dokładność modelu KNN, zbyt mała wartość może prowadzić do nadmiernego dopasowania, podczas gdy zbyt duża może prowadzić do niedostatecznego dopasowania.
* Parametr 'weights' w KNN Regressor służy do określenia sposobu wagowania sąsiadów podczas prognozowania wartości docelowych dla nowych obserwacji. Istnieją dwie opcje: 'uniform' i 'distance'. Dla 'uniform', wszyscy sąsiedzi mają taką samą wagę, podczas gdy dla 'distance', wagi są odwrotnie proporcjonalne do odległości od nowej obserwacji. Wybór odpowiedniego sposobu wagowania zależy od charakteru danych i problemu, gdzie 'uniform' może być stosowane, gdy wszystkie punkty są równie ważne, a 'distance' może dawać lepsze wyniki, jeśli bliżsi sąsiedzi mają większy wpływ na prognozowanie.
* Parametr 'metric' w KNN Regressor służy do określenia metryki używanej do obliczania odległości między obserwacjami w algorytmie k-najbliższych sąsiadów (KNN). Domyślnie wartość tego parametru to 'euclidean', co oznacza obliczanie odległości Euklidesowej. Jednak można również wybrać inne metryki, takie jak 'manhattan' (odległość Manhattan), 'minkowski' (ogólna metryka Minkowskiego) czy 'chebyshev' (odległość Czebyszewa), w zależności od charakterystyki danych   
  i potrzeb problemu. Wybór odpowiedniej metryki ma istotny wpływ na dokładność i efektywność modelu KNN, ponieważ różne metryki mogą lepiej odzwierciedlać specyfikę danych i strukturę przestrzeni.
  + 1. Wyniki

Wybrane zostały następujące parametry: 'algorithm': 'auto', 'metric': 'cityblock', 'n\_neighbors': 9, 'weights': 'distance'.

Wynik modelu KNN nie wypadł tak dobrze jak w przypadku Ridge, metryki takie jak R2 Score oraz średni błąd są znacząco gorsze niż w przypadku wcześniej opisanego modelu liniowego. Poziom błędu na poziomie przewyższającym 14% jest zdecydowanie za wysoki, aby model ten mógł być użyty w celu analizy danych dotyczących ceny mieszkań.

## DecisionTreeRegressor

 Rys. 5.5 Kod użyty do stworzenia modelu DecisionTreeRegressor

### Parametry

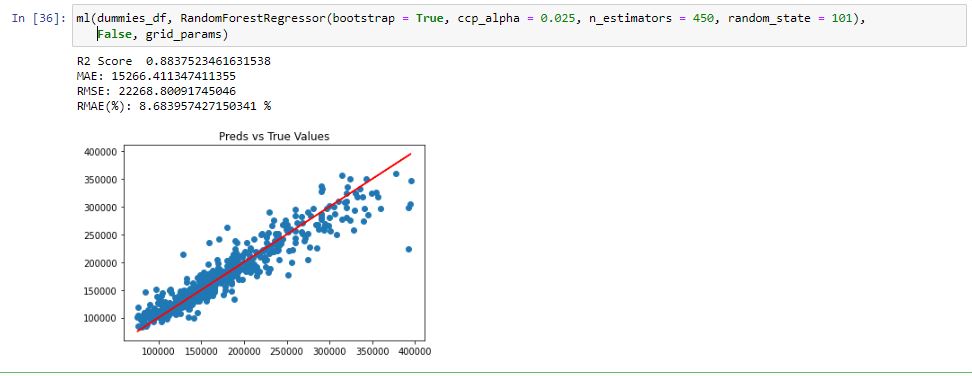
* Parametr 'min\_impurity\_decrease' w Decision Tree Regressor służy do kontrolowania minimalnego spadku impurity (czystości) w węźle, który jest wymagany do kontynuacji podziału. Wartość tego parametru określa minimalną różnicę impurity, która musi być osiągnięta w wyniku podziału węzła. Jeśli spadek impurity jest mniejszy niż wartość 'min\_impurity\_decrease', podział w tym węźle nie zostanie wykonany. Parametr ten jest używany do regularyzacji modelu   
  i zapobiegania nadmiernemu dopasowaniu. Wybór optymalnej wartości 'min\_impurity\_decrease' zależy od charakteru danych i pożądanego poziomu regularyzacji, gdzie większa wartość prowadzi do prostszych drzew i ogranicza podziały o niewielkim spadku impurity.
* Parametr ccp\_alpha (czyli Cost Complexity Pruning alpha) to parametr, który wpływa na proces przycinania drzewa decyzyjnego w algorytmie regresji drzewa decyzyjnego z ograniczeniami. Przycinanie drzewa decyzyjnego polega na usuwaniu nieistotnych węzłów z drzewa, aby zmniejszyć jego złożoność i zapobiec nadmiernemu dopasowaniu (overfitting).
* Parametr criterion to parametr, który określa kryterium podziału węzłów w drzewie decyzyjnym. Wyróżniamy mse ora friedman\_mse, który jest zmodyfikowaną wersją mse.

### Wyniki

Wybrane zostały następujące parametry: {'ccp\_alpha': 0.3, 'criterion': 'absolute\_error', 'min\_impurity\_decrease': 0.0001, 'random\_state': 101}.

Wyniki modelu pojedynczego Drzewa Decyzyjnego tak jak można się było spodziewać nie okazały się zbyt zadawalające, model ten uzyskał wyższy poziom błędów MAE oraz RMSE a co za tym idzie jego R2 Score rówanież był zdecydowanie wyższy od poprzedników. Nie mniej jednak spodziewaliśmy się podobnych wyników, rozpoczynając tą analize zdecydowaliśmy wytrenować oraz sprawdzić najlepsze do naszego zbioru danych hyperparametry dla drzewa decyzyjnego głównie dlatego że chcemy to drzewo decyzyjne wykorzystać w następnych modelach. Kolejne trzy modele będą modelami opartymi na drzewach decyzyjnych, a w jendym z nich bezpośrednio wykorzystamy drzewo ze zdefiniowanymi parametrami powyżej.

## RandomForestRegressor

 Rys. 5.6 Kod użyty do stworzenia modelu RandomForestRegressor

### Parametry

* bootstrap - parametr binarny (True/False), określający, czy w procesie budowy każdego drzewa decyzyjnego należy losować próbkę danych z powtórzeniami (zastąpieniami) czy bez powtórzeń (bez zastąpień). W przypadku, gdy bootstrap=True, drzewo będzie budowane na podstawie losowego podzbioru próbek, co zwiększa zróżnicowanie drzew i pomaga w uniknięciu przeuczenia.
* ccp\_alpha - parametr, który określa minimalną wielkość kosztu cięcia (ang. cost complexity pruning), który musi zostać osiągnięty, aby węzeł był przycinany. Im wyższa wartość ccp\_alpha, tym bardziej agresywnie dokonywane jest przycinanie drzewa, co może prowadzić do uproszczenia modelu i zmniejszenia jego skłonności do przeuczenia.
* n\_estimators - liczba drzew decyzyjnych w lesie losowym. Im wyższa wartość parametru, tym bardziej skomplikowany będzie model i tym większe prawdopodobieństwo uzyskania dokładniejszych wyników, ale zwiększy się też czas trenowania i predykcji. Z drugiej strony, zbyt mała wartość n\_estimators może prowadzić do niedouczenia modelu.

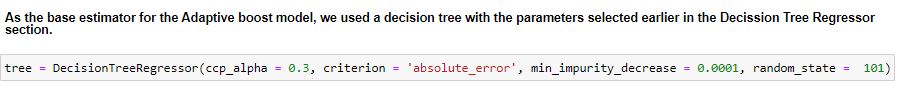
### Wyniki

Wybrane zostały następujące parametry:{'bootstrap': True, 'ccp\_alpha': 0.02, 'n\_estimators': 400, 'random\_state':101}

W przypadku RandomForest wyniki okazały się znacznie lepsze niż w przypadku pojedynczego drzewa. Powodem tego jest fakt, że dane na których operujemy są wysoce złozone nawet w pierwotnej formie posiadały ponad 70 kolumn, gdzie po zastosowaniu OneHotEncoding ta liczba wzrosła do 4000. Z tego powodu modele takie jak pojedyncze drzewo decyzyjne nie radzą sobie tak dobrze jak ich bardziej złożone odpowiedniki.

## AdaptiveBoost

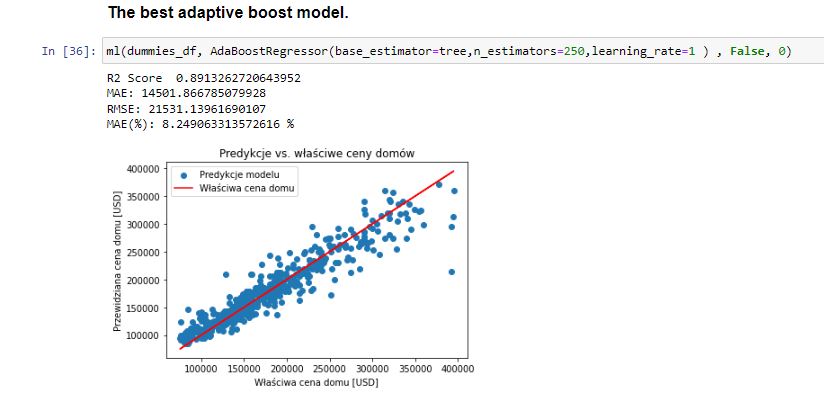
W tym modelu bezpośrednio użyjemy wcześniej zdefiniowanego drzewa decyzyjnego z poprzedniego rozdziału.

Rys. 5.7 Zdefiniowanie weaklearnera

### Parametry

W przypadku AdaBoost nie było potrzeby definiować takiej ilości hyperparametrów jak w przypadku poprzenich modeli, głównie dlatego że Adaptive Boost wykorzystuje wcześniej zdefiniowane (pod kątem hyperparametrów) drzewo decyzyjne. Dlatego też nie używaliśmy w tym przypadku GridSearchCV, a jedyne parametry jakie zdefiniowaliśmy, dotyczył:

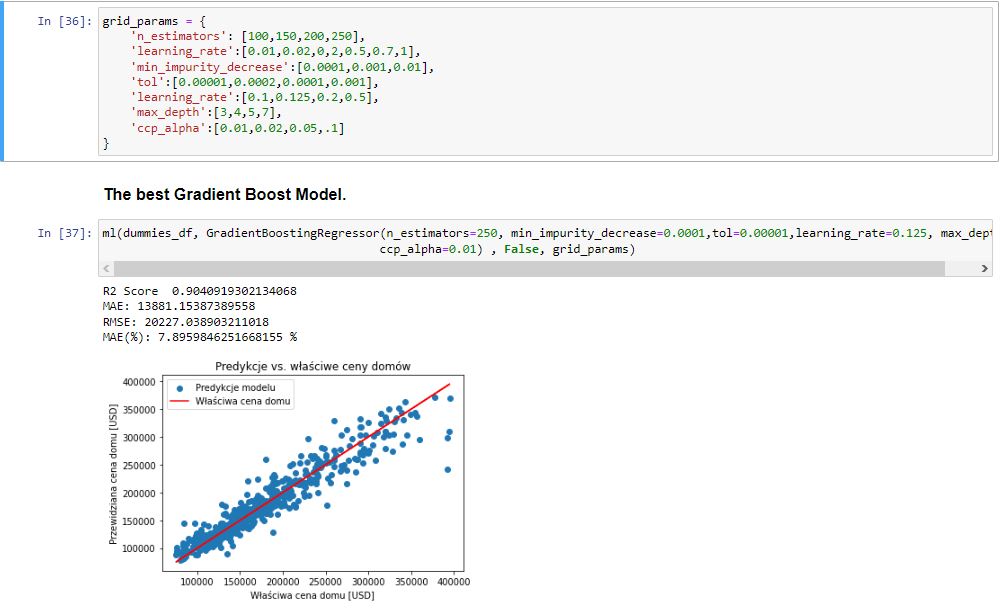
* n\_estimators - ilości drzew wykorzystywanych przez model.
* base\_estimators - definicja jakie ma być to drzewo, zdecydowaliśmy się na użycie drzewa zdefiniowanego wcześniej.
* learning\_Rate - współczynnik określający, jak bardzo każde nowe drzewo decyzyjne będzie próbowało poprawić błędy modelu w poprzednich iteracjach.

 Rys. 5.8 Kod użyty do stworzenia modelu AdaptiveBoost

### Wyniki

AdaptiveBoost osiągnał wynik zbliżony do Randomforest nieznacznie go przebijając, wartym jednak odnotowania jest fakt, że potrzebował na to kilkukortnie więcej czasu od swojego poprzednika na sam trening.

## Gradient Boost

 Rys. 5.9 Kod użyty do stworzenia modelu Gradient Boost

### Parametry

* n\_estimators: liczba drzew decyzyjnych, które zostaną zbudowane przez algorytm Gradient Boost. Im większa wartość parametru, tym bardziej złożony będzie model, ale jednocześnie wzrośnie też czas trenowania i predykcji.
* learning\_rate: współczynnik określający, jak bardzo każde nowe drzewo decyzyjne będzie próbowało poprawić błędy modelu w poprzednich iteracjach. Im wyższa wartość parametru, tym większe będą korekty i tym bardziej skomplikowany będzie model.
* min\_impurity\_decrease: minimalna zmiana impurity, która musi zostać osiągnięta, aby węzeł drzewa decyzyjnego był podzielony. Im wyższa wartość parametru, tym bardziej agresywnie dokonywane jest przycinanie drzewa, co może prowadzić do uproszczenia modelu i zmniejszenia jego skłonności do przeuczenia.
* tol: wartość progowa określająca minimalną wartość zmiany funkcji kosztu między kolejnymi iteracjami, która jest wymagana, aby algorytm zakończył iteracje. Im mniejsza wartość parametru, tym więcej iteracji jest potrzebnych, aby algorytm zakończył pracę.
* max\_depth: maksymalna głębokość drzewa decyzyjnego. Im większa wartość parametru, tym bardziej skomplikowany i bardziej dopasowany do danych będzie model, ale jednocześnie będzie bardziej skłonny do przeuczenia.
* ccp\_alpha: parametr określający minimalną wielkość kosztu cięcia, która musi zostać osiągnięta, aby węzeł drzewa decyzyjnego był przycinany. Im wyższa wartość parametru, tym bardziej agresywnie dokonywane jest przycinanie drzewa, co może prowadzić do uproszczenia modelu i zmniejszenia jego skłonności do przeuczenia.

### Wyniki

Wybrane zostały następujące parametry:

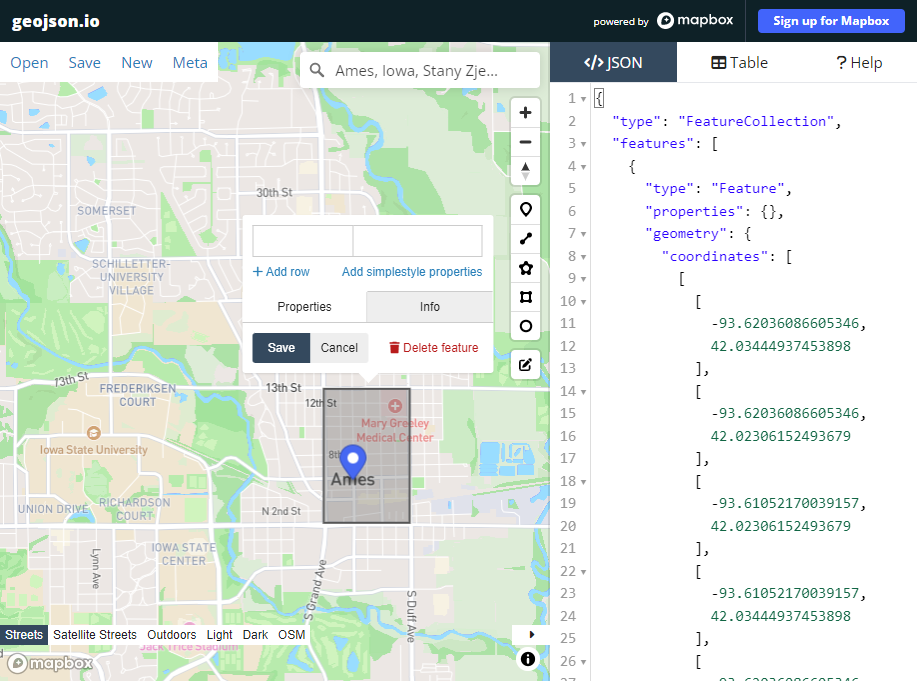
{n\_estimators=250, min\_impurity\_decrease=0.0001,tol=0.00001,learning\_rate=0.125, max\_depth=5, ccp\_alpha=0.01}

GradientBoost, tak jak dwa poprze wykorzystujący wiele drzew poradził sobie znakomicie z przewidywaniem ceny domów. Wynik, w którym średni błąd osiąga poziom poniżej 8 % jest zdecydowanie zadowalający. Można powiedzieć, że model ten jest bardzo podobny do RandomForest, ponieważ tak samo wykorzystuje wiele drzew decyzyjnych. Różnica jednak pomiędzy nimi jest diametralna. GradientBoost w przeciwieństwie do RandomForest nie traktuje każdego nowego drzewa jako osobny byt, kolejne drzewa w tym modelu "uczą" się na błędach poprzedników.

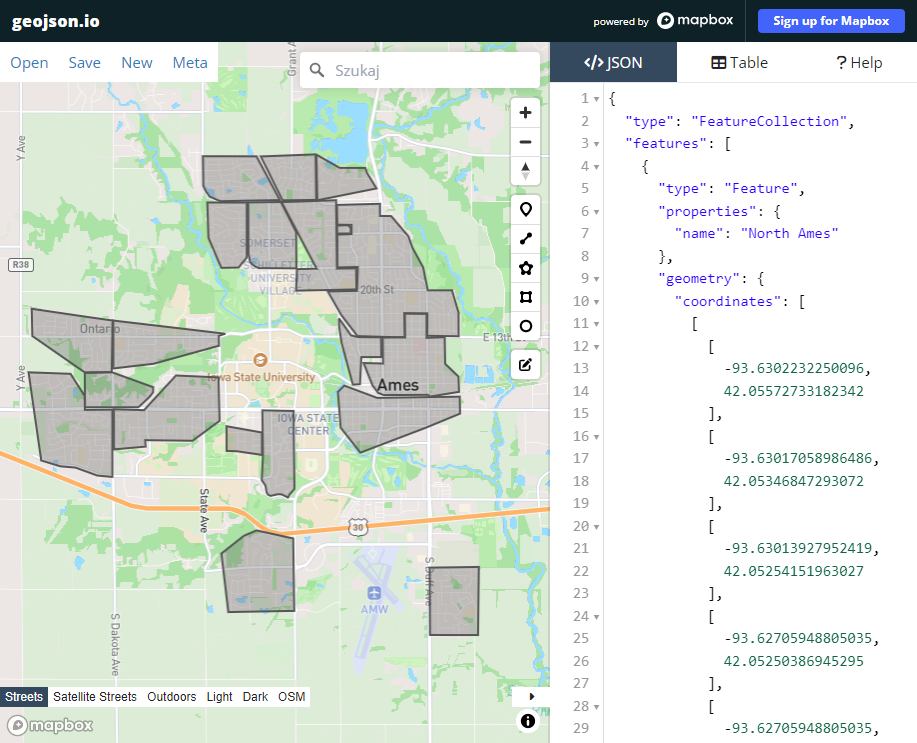
# Wizualizacja z wykorzystaniem modeli uczenia maszynowego

Korzystając z biblioteki plotly oraz JSON została stworzona interaktywna mapa przedstawiająca średnie ceny sprzedaży mieszkań (SalePrice) poszczególnych dzielnic   
w mieście Ames.

Do zrealizowania tego projektu potrzebne były szczegółowe koordynaty wierzchołków wszystkich dzielnic miasta. W sieci nie było takich plików, dlatego do projektu została stworzona własnoręczna mapka za pomocą narzędzia geojson.io, w którym to samodzielnie zaznaczało się obszar dzielnic, następnie dodawano nazwy i całość zapisywało się do pliku w formacie JSON. Następnie za pomocą biblioteki plotly złączony został plik z wierzchołkami i łączony z ramką danych w wyniku czego zarysowana została mapa z zaznaczonymi dzielnicami, których kolory w zależności od skali, zaznaczają średnią wartość cen z poszczególnych dzielnic.



Rys. 6.1 Procedura tworzenia pierwszego obramowania dzielnic

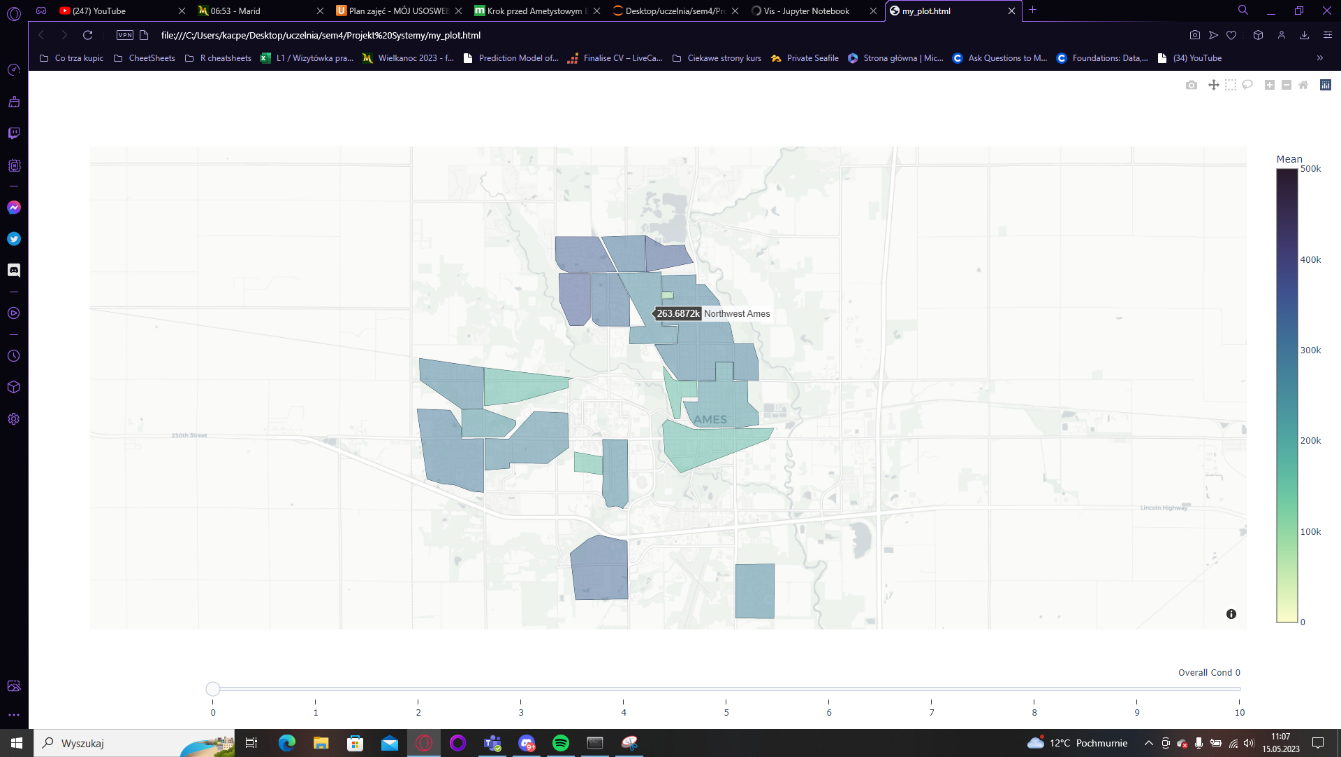
 Rys. 6.2 Zarysowanie wszystkich dzielnic na stronie geojson.io

Dodatkowo, na mapie dodawany jest suwak z wartościami od 0 do 10, który umożliwia zmianę wartości "Overall Cond". Dzięki temu, użytkownik może aktualizować wartości kolumny 'Overall Cond' w DataFrame, by następnie za pomocą algorytmu uczenia maszynowego pokazać jak zmiana tego parametru wpływa na cenę mieszkań - zmienia się kolor dzielnic. Po przesunięciu suwaka, funkcja "update\_z\_values()" jest wywoływana   
i aktualizuje wartości kolumny 'Overall Cond' w DataFrame zgodnie z wartością suwaka.

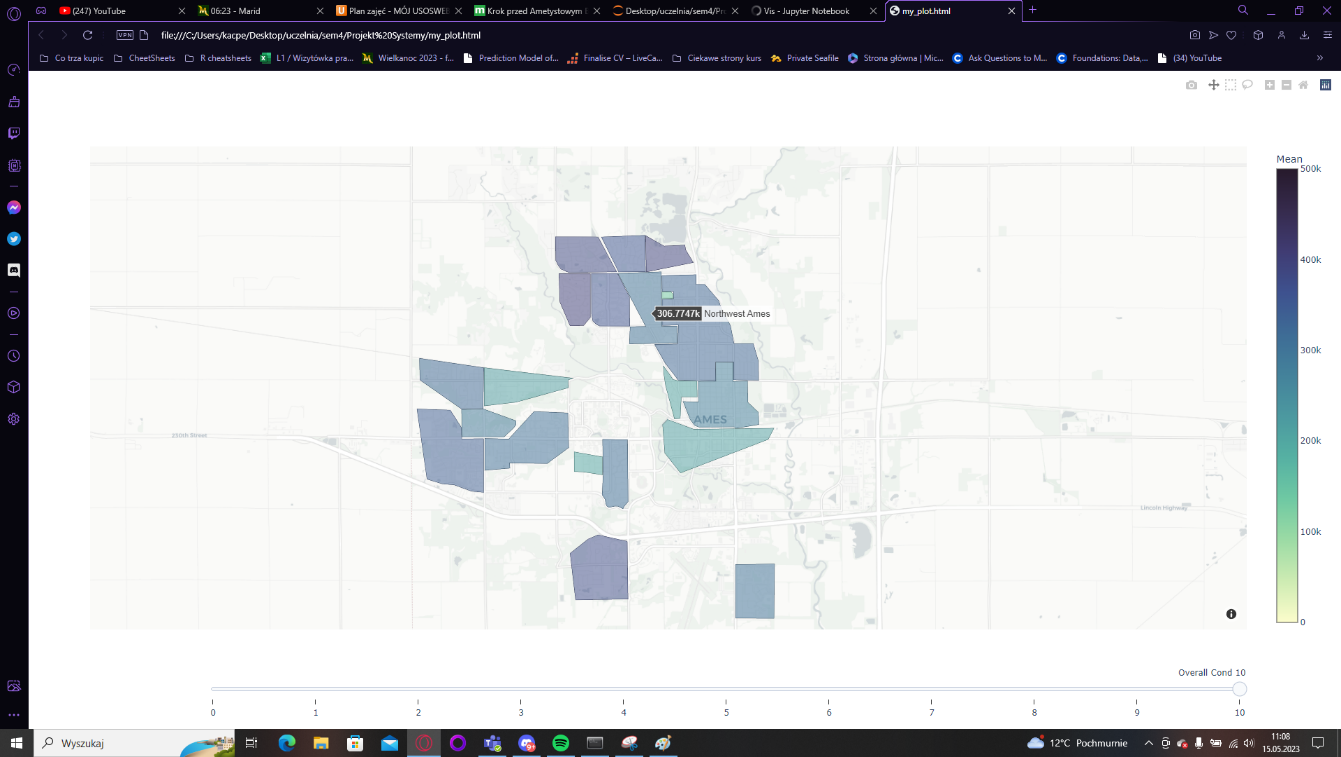
Następnie wartości "SalePrice" są przedstawione na mapie w formie chloropleth. Po najechaniu na poszczególną dzielnicę wyświetla się średnia cena mieszkań z tego regionu. Na jej podstawie można oszacować czy warto kupić dom pod wyremontowanie i późniejszą sprzedaż.



Rys. 6.3 Kod do funkcji tworzącej mapę interaktywną z suwakiem



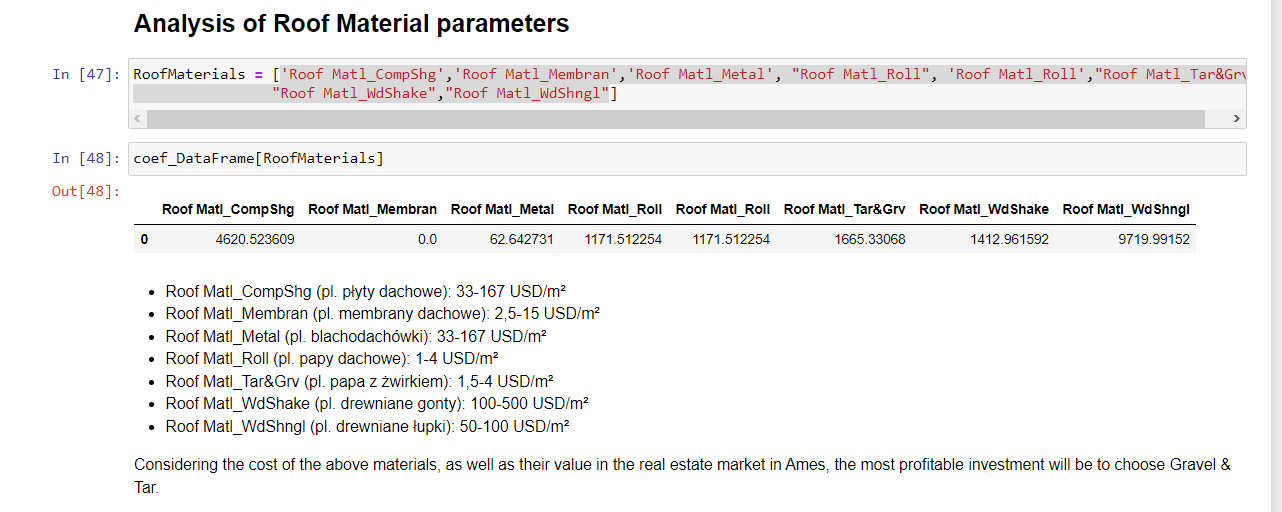
Rys. 6.4 Średnie ceny domów w dzielnicach, w których wszystkie domy posiadają niski stan kondycji



Rys. 6.5 Średnie ceny domów w dzielnicach, w których wszystkie domy posiadają wysoki stan kondycji

# Analiza opłacalności poszczególnych parametrów.

## Dachy

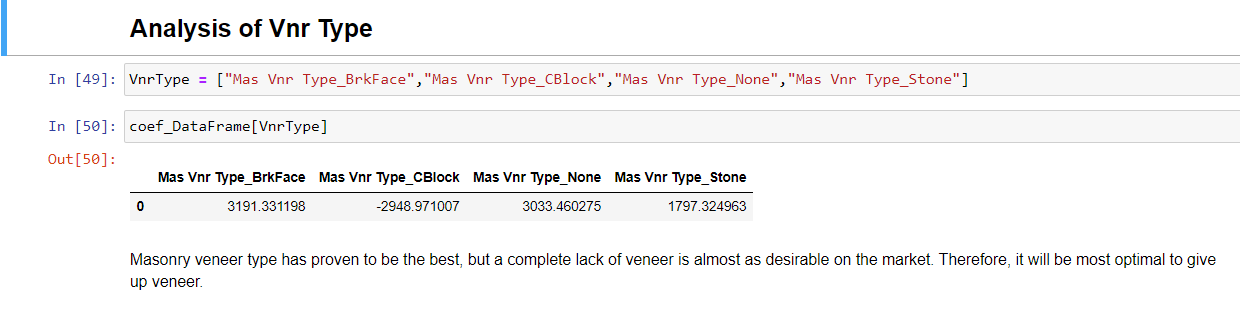


Rys. 6.3 Analiza wpływu rodzaju dachu na cenę mieszkania

* Wartość domu z dachem pokrytym płytami dachowymi wzrasta o 4620$ - koszt instalacji to 33-167$/m2.
* Wartość domu z dachem pokrytym membranami dachowymi pozostaje bez zmian - koszt instalacji to 2,5-15$/m2.
* Wartość domu zawierającego dach pokryty blachodachówką wzrasta o 62$ - koszt instalacji to 33-167$/m2.
* Wartość domu zawierającego dach pokryty papą dachową wzrasta o 1665$ - koszt instalacji to 33-167$/m2.
* Wartość domu zawierającego dach pokryty papą ze żwirkiem wzrasta o 1171$ - koszt instalacji to 1,5-4$/m2.
* Wartość domu zawierającego dach pokryty drewnianymi gontami wzrasta o 1412$ - koszt instalacji to 100-500$/m2.
* Wartość domu zawierającego dach pokryty drewnianymi łupkami wzrasta o 9719$ - koszt instalacji to 50-100$/m2.

Biorąc pod uwagę koszt powyższych materiałów, a także ich wartość na rynku nieruchomości w Ames, najbardziej opłacalną inwestycją będzie wybór papa ze żwirkiem (tar & grv). Wybór takiego typu dachu korzystnie wpływa na cenę domu przy niewielkim wkładzie finansowym.

## Typ forniru budowlanego

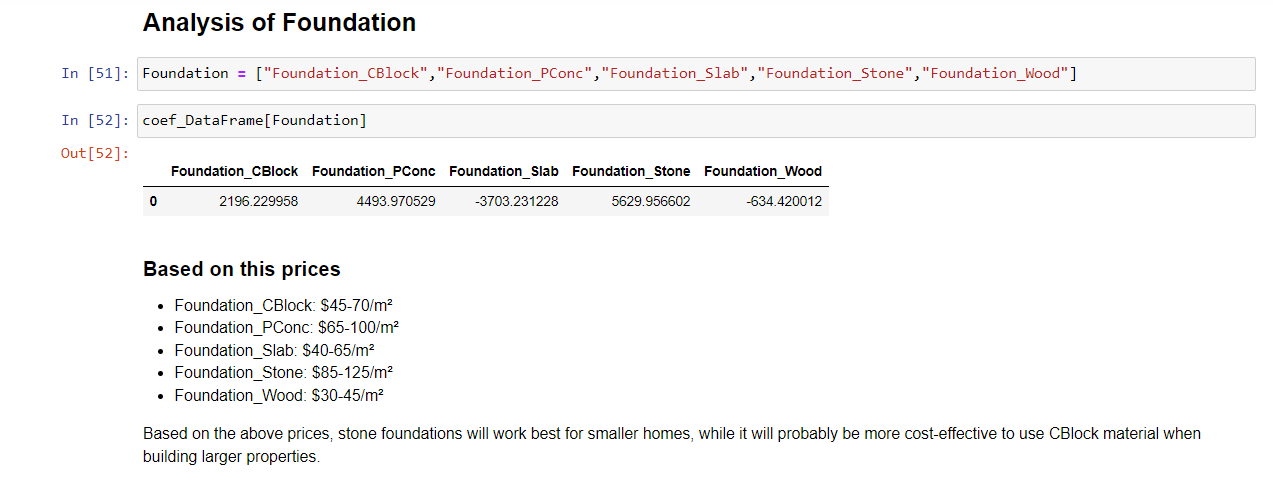


Rys. 6.3 Analiza wpływu forniru budowlanego na cenę mieszkania.

* Wartość domu zawierającego cegłę ciętą wzrasta o 3191$.
* Wartość domu zawierającego pustak żużlowy maleje o 2948$.
* Wartość domu niezawierającego forniru wzrasta o 3033$.
* Wartość domu zawierającego kamień jako fornir wzrasta o 1787$.

Fornir murowany okazał się najlepszy, ale całkowity brak forniru jest niemal tak samo pożądany na rynku. Dlatego najbardziej optymalna będzie rezygnacja z forniru.

## Rodzaj fundamentu

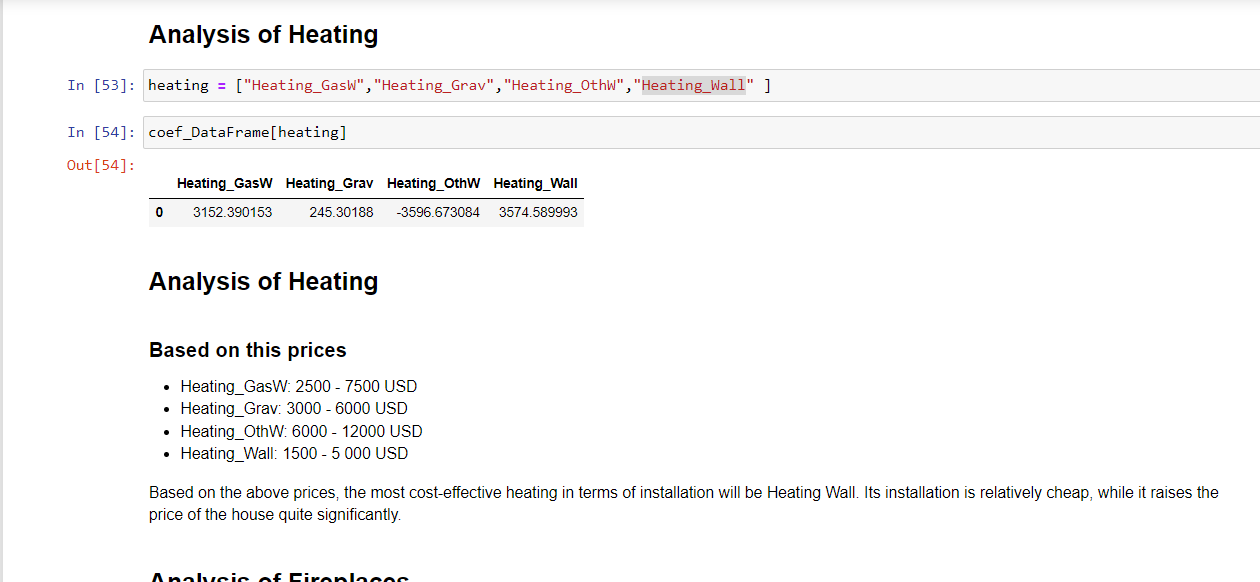


Rys. 6.3 Analiza wpływu rodzaju fundamentu na cenę mieszkania

* Wartość domu zawierającego pustak żużlowy wzrasta o 2196$ - cena za wykonanie wynosi 45-70$/m2.
* Wartość domu zawierającego wylany beton wzrasta o 4493$ - cena za wykonanie wynosi 65-100$/m2.
* Wartość domu zawierającego fundament płytowy maleje o 3703$ - cena za wykonanie wynosi 40-65$/m2.
* Wartość domu zawierającego kamień wzrasta o 5629$ - cena za wykonanie wynosi 85-125$/m2.
* Wartość domu zawierającego drewno maleje o 634$ - cena za wykonanie wynosi 30-45$/m2.

Biorąc pod uwagę powyższe ceny, fundamenty kamienne najlepiej sprawdzą się   
w przypadku mniejszych domów, podczas gdy prawdopodobnie bardziej opłacalne będzie zastosowanie pustaka żużlowego przy budowie większych nieruchomości.

## Ogrzewanie

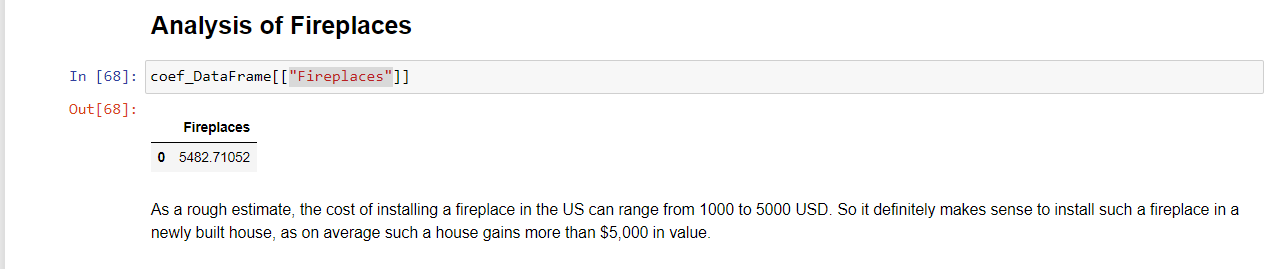


Rys. 6.3 Analiza wpływu rodzaju ogrzewania na cenę mieszkania

* Wartość domu zawierającego ogrzewanie na ciepłą wodę gazową lub parową wzrasta o 3152$ - cena za wykonanie wynosi 2500-7500$.
* Wartość domu zawierającego piec grawitacyjny wzrasta o 245$ - cena za wykonanie wynosi 3000-6000$.
* Wartość domu zawierającego ogrzewanie na ciepłą wodę lub parę wodną inna niż gaz maleje o 3596$ - cena za wykonanie wynosi 6000-12000$.
* Wartość domu zawierającego piec ścienny wzrasta o 3574$ - cena za wykonanie wynosi 1500-5000$.

W oparciu o powyższe ceny najbardziej opłacalnym pod względem instalacji ogrzewaniem będzie ogrzewanie ścienne. Jego montaż jest relatywnie tani, natomiast dość znacząco podnosi cenę domu. Wartym odnotowania jest fakt, iż można to zrobić tylko   
w przypadku budowy domów bądź przy generalnych remontach. Gdy nie jesteśmy w stanie pozwolić sobie na taką ingerencję warto pomyśleć o piecu ściennym.

## Kominek



Orientacyjnie koszt instalacji kominka w USA może wahać się od 1000 do 5000 USD. Zdecydowanie warto więc zainstalować taki kominek w nowo wybudowanym domu, ponieważ średnio taki dom zyskuje na wartości ponad 5000 dolarów.

# Podsumowanie

Analiza doprowadziła nas do kilku kluczowych wniosków w przypadku firmy zajmującej się budową domów. Najbardziej opłacalnym typem zadaszenia jest papa ze żwirkiem cechująca się zwiększeniem wartości domu przy najniższym koszcie. Kolejny analizowany parametr to fornir budowlany. Najbardziej opłacalny w tym przypadku okazał się brak jego zastosowania przy budowie lub ewentualnym remoncie. Jeżeli chodzi   
o fundamenty, to w przypadku małych domów najlepiej wybierać fundament kamienny,   
a w przypadku większych nieruchomości będzie to pustak żużlowy. Przy instalacji ogrzewania w nowo wybudowanym domu lub przy remoncie warto zainwestować   
w ogrzewanie ścienne, natomiast w innym przypadku najlepszym rozwiązaniem będzie zakup pieca ściennego. Wartym rozważenia jest również montaż kominka, zwłaszcza   
w nowym domach, ponieważ cena domu wzrasta znacznie bardziej od poniesionych kosztów jego instalacji.

W trakcie wyboru dzielnicy deweloper chcący wybudować bądź wyremontować dom może skorzystać z przygotowanej mapy, aby odpowiednio dobrać dzielnicę pod swoją inwestycję. W przypadku remontów, dzięki suwakowi w prosty i sugestywny sposób może określić jak poprawa kondycji domu pozytywnie wpłynie na cenę remontowanej przez niego nieruchomości. Wartym odnotowania jest fakt, że poprawa kondycji domu wpływa na cenę w inny sposób w każdej dzielnicy a rozpatrywane są w przypadku mapy jedynie wartości średnie.

Literatura

[1] Provost F., Fawcett T. (2019) Analiza danych w biznesie. Sztuka podejmowania skutecznych decyzji, Wydawnictwo Helion

[2] Witten D., James G., Hastie T., Tibshirani R. (2021) An Introduction to Statistical Learning with Applications in R, Wydawnictwo Springer

[3] <https://scikit-learn.org/stable/supervised_learning.html#supervised-learning>

[4] <https://towardsdatascience.com/hyperparameter-tuning-for-machine-learning-models-1b80d783b946>