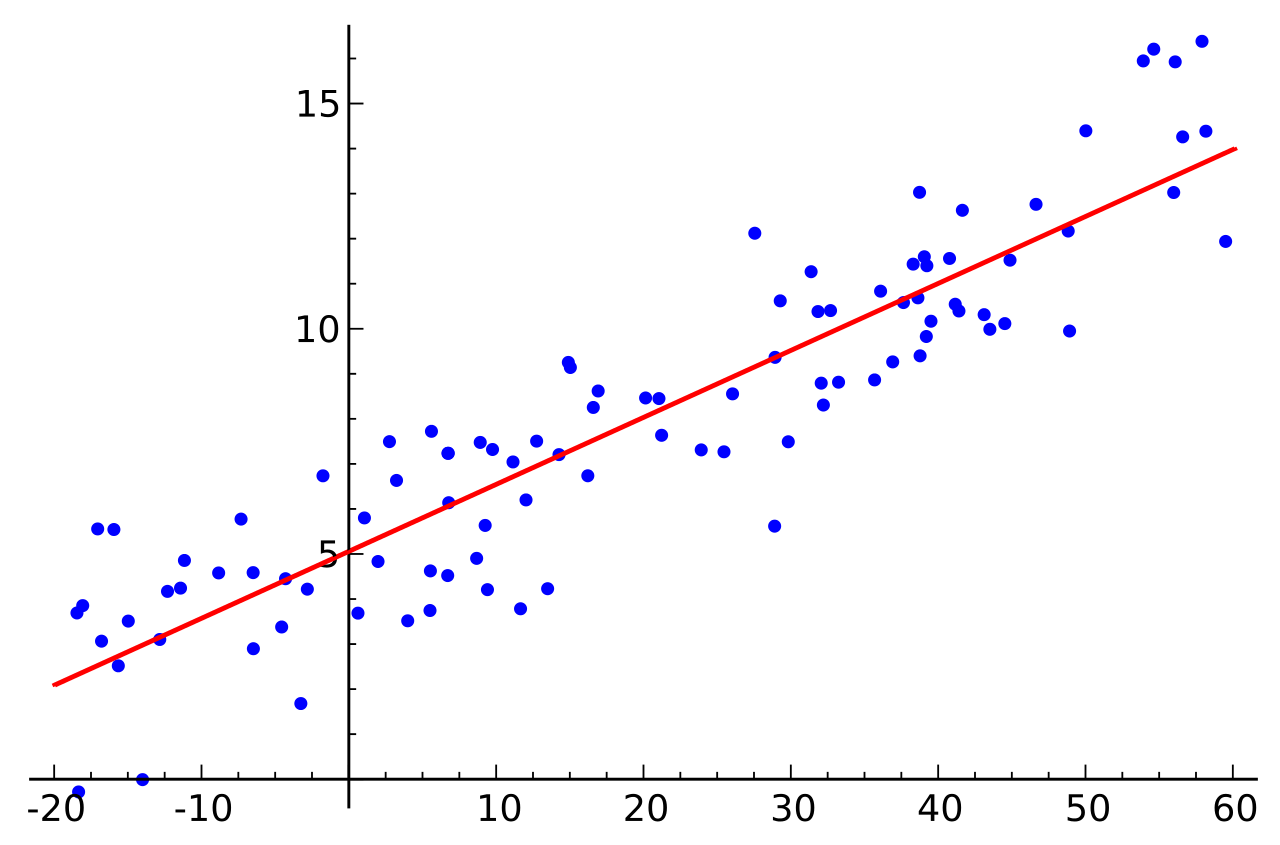
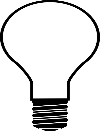
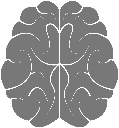
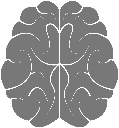
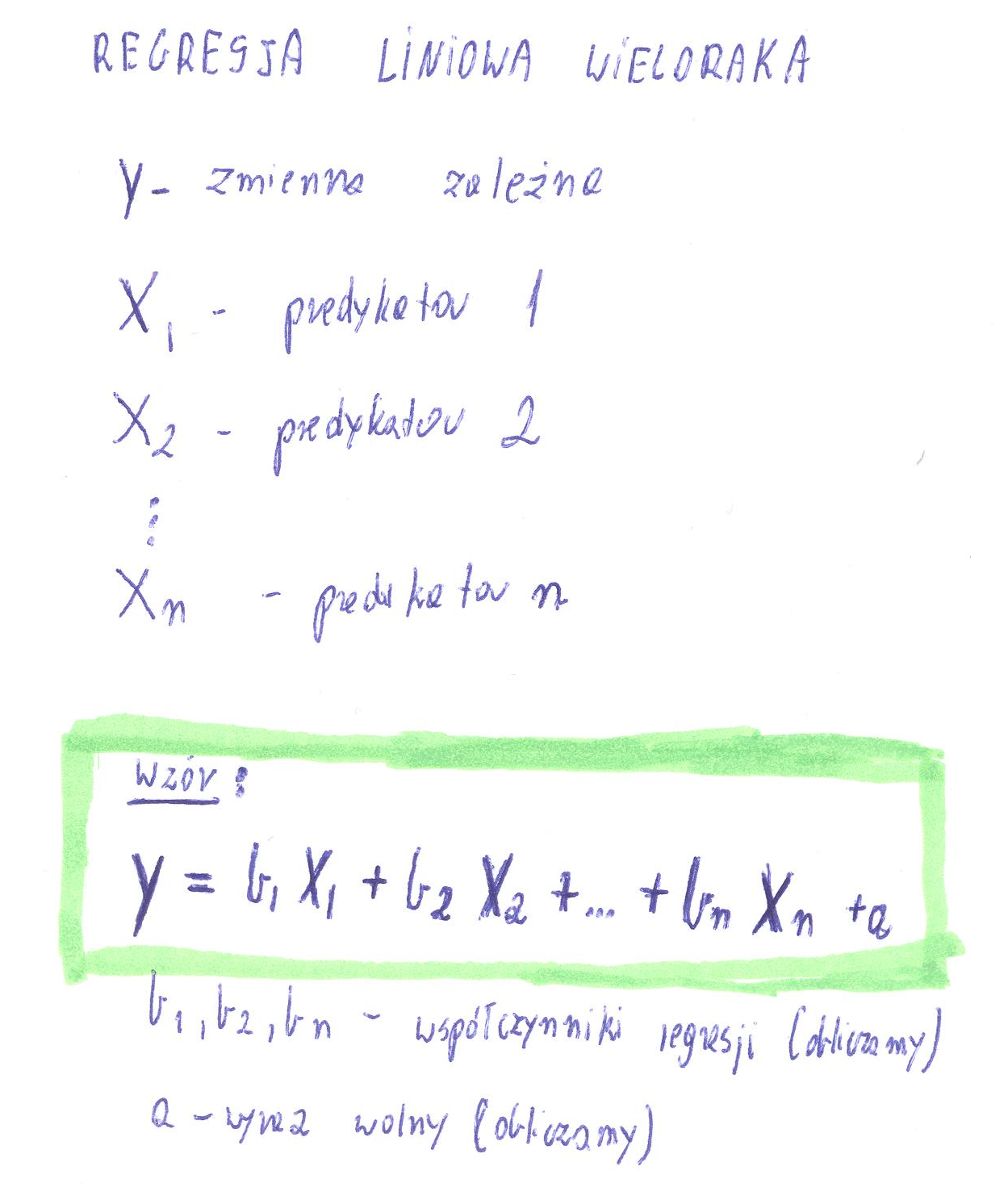
**Lab 4b – Analityka predyktywna – tworzenie modeli liniowych z kilkoma predykatorami – regresja liniowa wieloraka**



**Ikona pomyśl. Zatrzymaj się na chwilę na tym etapie oznaczonej tym symbolem i spróbuj zauważyć pewne prawidłowości. Opisz w kilku zdaniach wnioski.**

**Ikona mózg. Krótka teoria wprowadzająca do danego zagadnienia.**

Regresja liniowa wieloraka

* Zakłada, że istnieje zależność pomiędzy zmienną zależną a predykatorem jest zależnością liniową.
* Zmienne wprowadzane do modelu: zmienna zależna i predyktor muszą mieć postać ilorazową, nie mogą być to zmiennymi nominalnymi czy porządkowymi
* Podobnie jak w korelacji liniowej: wzrostowi jednej zmiennej (predyktorowi) towarzyszy wzrost lub spadek na drugiej zmiennej.
* Celem budowy modelu regresji jest wyliczenie współczynników regresji (b) tak aby model jak najlepiej przewidywał wartość zmiennej zależnej
* Do oceny błędu przewidywania służą różne współczynniki: SSE, RMSE, MAE – wspomniane na pierwszych zajęciach. Dąży się do tego aby model uzyskał jak najmniejszy błąd.
* Wybieramy predykatory do modelu które są istotne statystycznie. P-value <0,05

# Tworzenie modeli predykcyjnych na bazie regresji liniowej z jednym predykatorem – przykład uproszczony

1. **Kontekst biznesowy.** Opisujemy sytuację dla której stworzymy modele predykcyjne.
2. **Identyfikujemy dane**. Sprawdzamy jak wyglądają dane i co znaczą poszczególne zmienne.
3. **Statystyka opisowa.** Opisujemy podstawowe parametry, średnia, mediana, kwartyle, predykatory, zmienna zależna itd.
4. **Braki danych.** Gdy mamy braki danych musimy zrobić jedną z następujących rzeczy:
   1. Eliminujemy rekord - w przypadku gdy mamy dużo obserwacji(~ powyżej 100)
   2. Uzupełniamy uśrednioną wartością z kolumny dot. zmiennej- mniejsza dokładność
   3. Uzupełniamy uśrednioną wartością z wykorzystaniem algorytmu knn - zwykle większa dokładność od pozostałych (polecany).
5. **Wartości odstające.** Usuwamy wartości odstające i wartości ekstremalne. IRQ=Q3-Q1.

1.5\* IRQ – wartość odstająca

3 \* IRQ – wartość ekstremalna

* 1. Metoda I. Przekroczenie wartości poniżej V1=Q1 - 1.5\*IQR lub powyżej V2=Q3 + 1.5\*IQR
  2. Metoda II. Boxplot- wykres pudełkowy

1. **Korelacje.** Sprawdzamy związkowość między potencjalnymi predykatorami, a zmienną zależną wybierając najwyżej skorelowany predykator.
   1. **Wykresy rozrzutu**
   2. **Współczynniki korelacji**
2. **Dzielimy dane.** Sortujemy wcześniej losowo dobraną kolejność obserwacji funkcją i dzielimy dane na uczące(około 70% obserwacji) i testujące model (około 30% obserwacji).
3. **Budowa i wizualizacja modeli.** Dodatek Toolpack Analysis). Następnie przewidujemy wartości na danych testujących. Rysujemy modele.
4. **Testowanie modeli.** Testujemy model na danych testujących. Metody, SSE, MAE, Root Mean Squared Error (RMSE). Mniejszy współczynnik=lepiej.

# Tworzenie modeli predykcyjnych na bazie regresji liniowej z kilkoma predykatorami (regresja wieloraka) – przykład rozszerzony

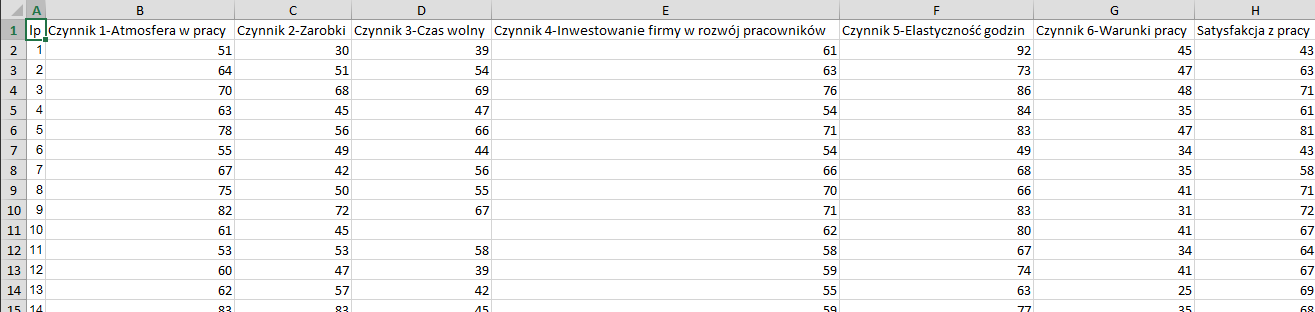
1. **Kontekst biznesowy.** Opisujemy sytuację dla której stworzymy modele predykcyjne.

Przykład:

Właściciel firmy X chce wiedzieć jak jego pracownicy są zadowoleni ze swojej pracy. Zrobił ankietę w której pytania dotyczyły wybranych czynników. Ostatnie pytanie dotyczyło jak bardzo są zadowoleni z pracy. Wybierz co najmniej dwie zmienne, które najbardziej przekładają się na wynik dotyczący satysfakcji, zrób i przetestuj obydwa modele na danych i zasugeruj odpowiedni wybór czynnika klientowi przedstawiając wynik analizy. Po podaniu wartości z wybranego czynnika model powinien przewidzieć jego satysfakcję z pracy.

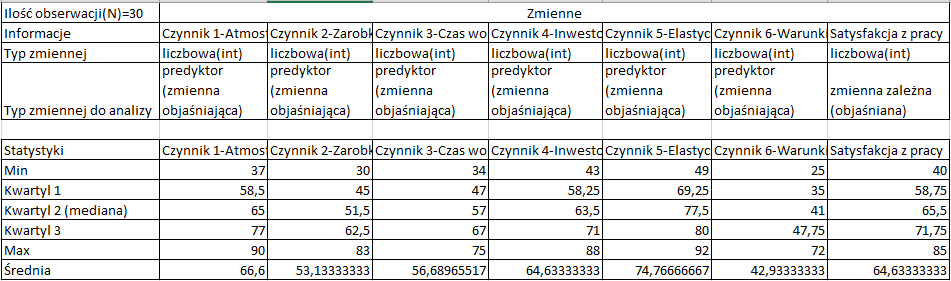
1. **Identyfikujemy dane.** Sprawdzamy jak wyglądają dane i co znaczą poszczególne zmienne.

Przykład:



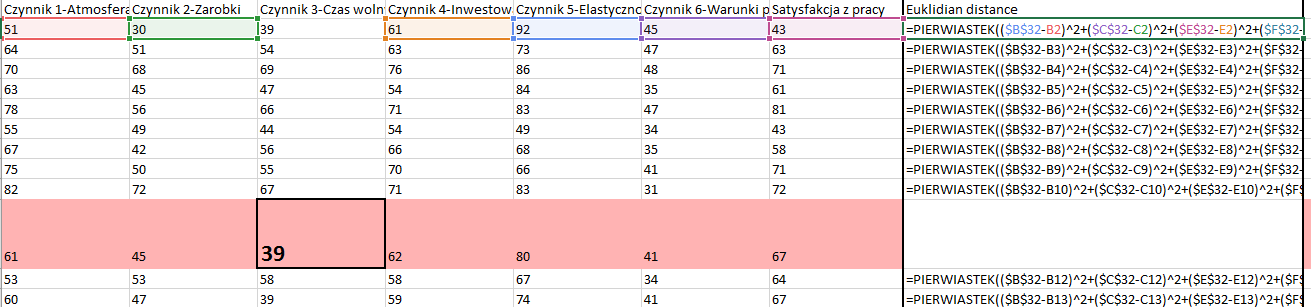
1. **Statystyka opisowa.** Opisujemy podstawowe parametry, średnia, mediana, kwartyle, predykatory, zmienna zależna itd.

Przykład:



1. **Braki danych.** Gdy mamy braki danych musimy zrobić jedną z następujących rzeczy:
   1. **Eliminujemy rekord** - w przypadku gdy mamy dużo obserwacji(~ pow 100)
   2. **Uzupełniamy uśrednioną wartością z kolumny dot. zmiennej** - mniejsza dokładność
   3. **Uzupełniamy uśrednioną wartością z wykorzystaniem algorytmu knn** (odległość Euklidesa) - większa dokładność od pozostałych (polecany). Temat do zapoznania: odległość Euklidesa





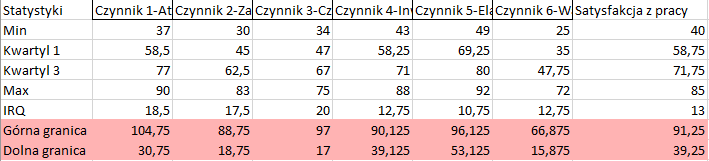
1. **Wartości odstające.** Usuwamy wartości odstające i wartości ekstrmalne. IRQ=Q3-Q1.

**1.5\* IRQ – wartość odstająca**

**3 \* IRQ – wartość ekstremalna**

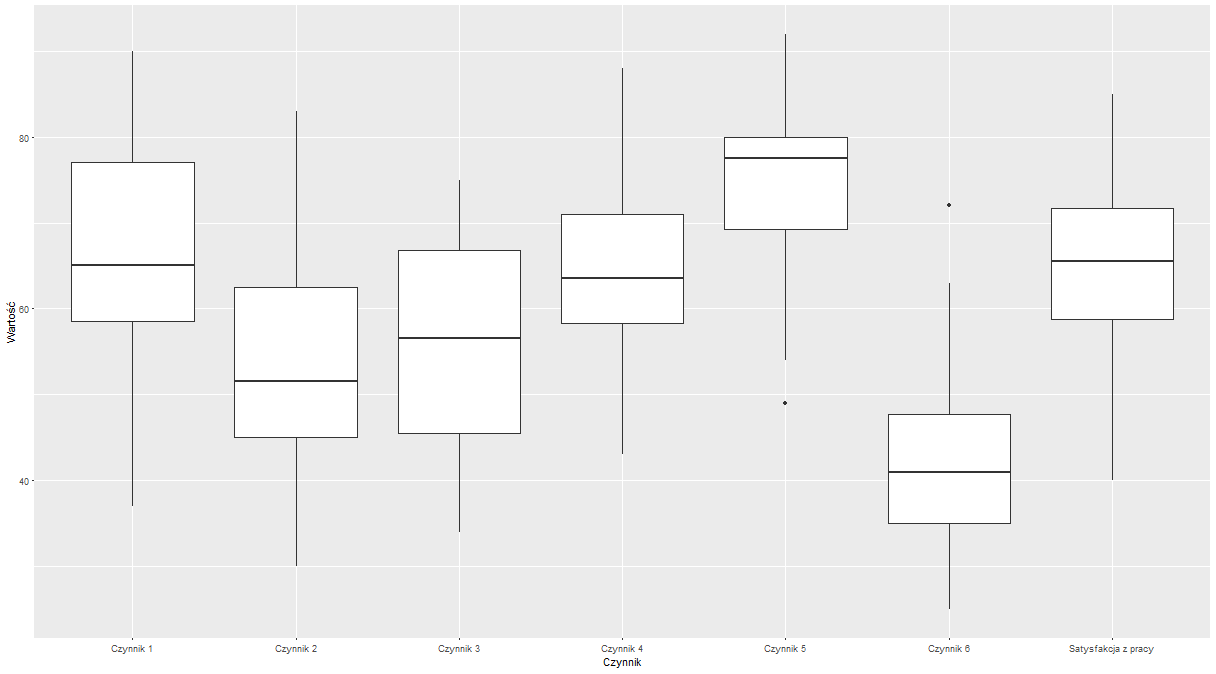
* 1. Metoda I. Przekroczenie wartości poniżej V1=Q1 - 1.5\*IQR lub powyżej V2=Q3 + 1.5\*IQR

Przykład:



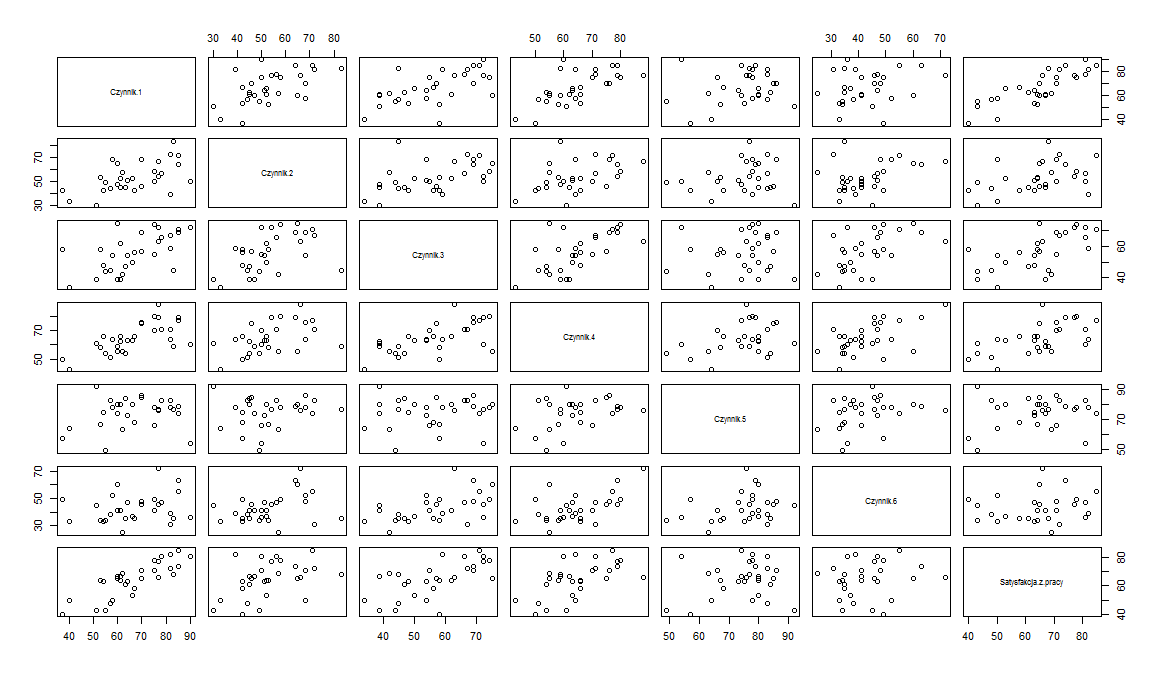
* 1. Metoda II. Boxplot- wykres pudełkowy

Przykład:



1. **Korelacje.** Sprawdzamy związkowość między potencjalnymi predykatorami, a zmienną zależną. W Excel -wykres rozrzutu oraz funkcja *wsp.korelacji()- mierząca współczynnik korelacji Pearsona.*

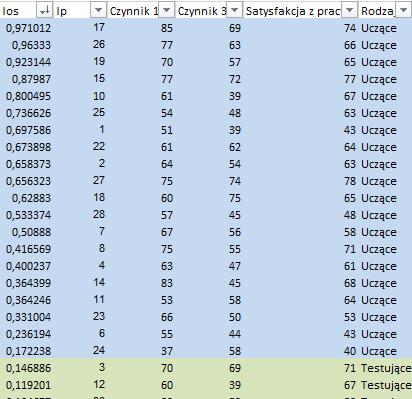
Przykład:



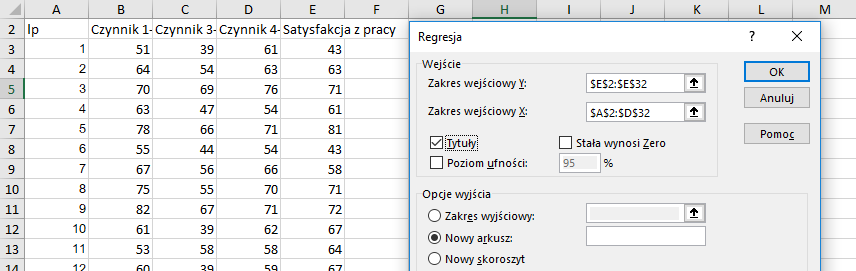


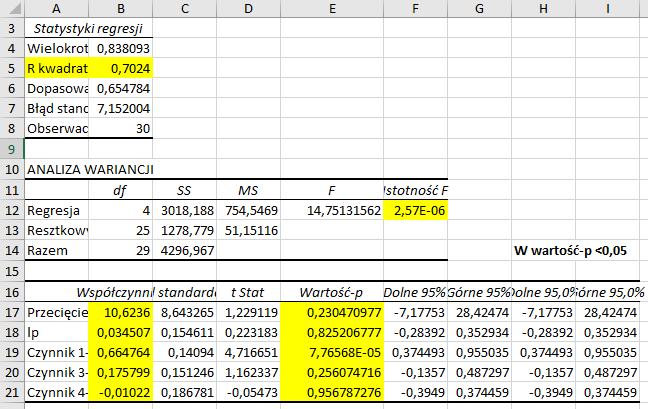
1. **Dzielimy dane.** Sortujemy wcześniej losowo dobraną kolejność obserwacji funkcją los() i dzielimy dane na uczące(około 70% obserwacji) i testujące model (około 30% obserwacji).

Przykład:

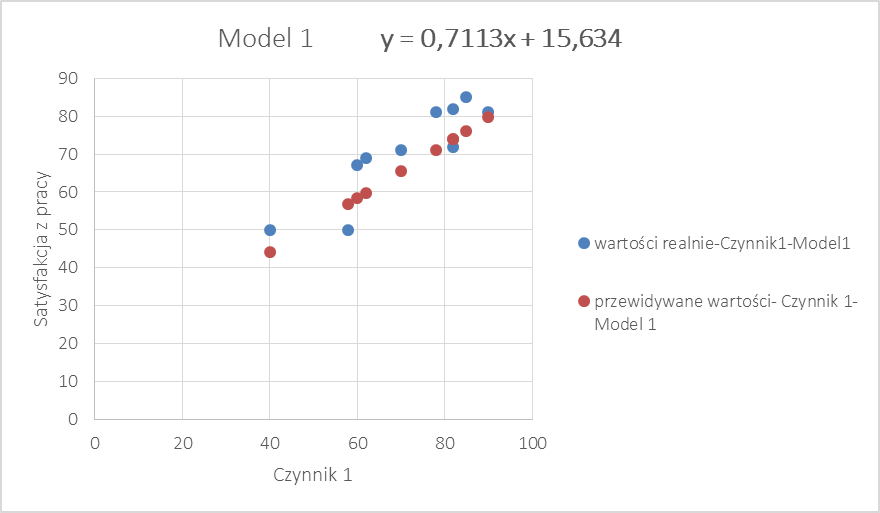


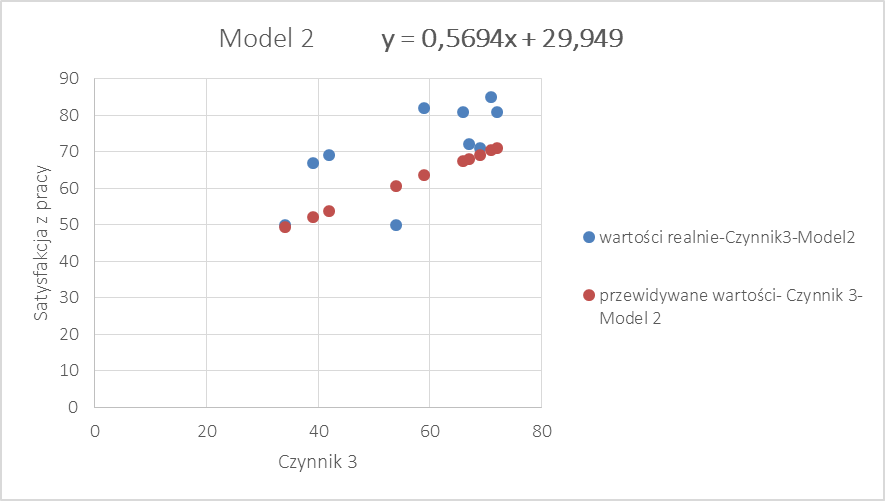
1. **Budowa i wizualizacja modeli.** Zakładka Dane/Dodatek Toolpack Analysis (Analiza danych)/Regresja. Wybieramy zmienne z najsilniejszą korelacją(jako X) jako Y(Zmienna zależna)

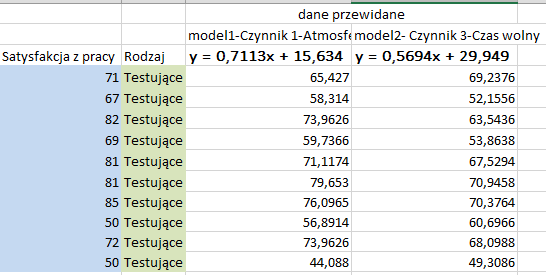




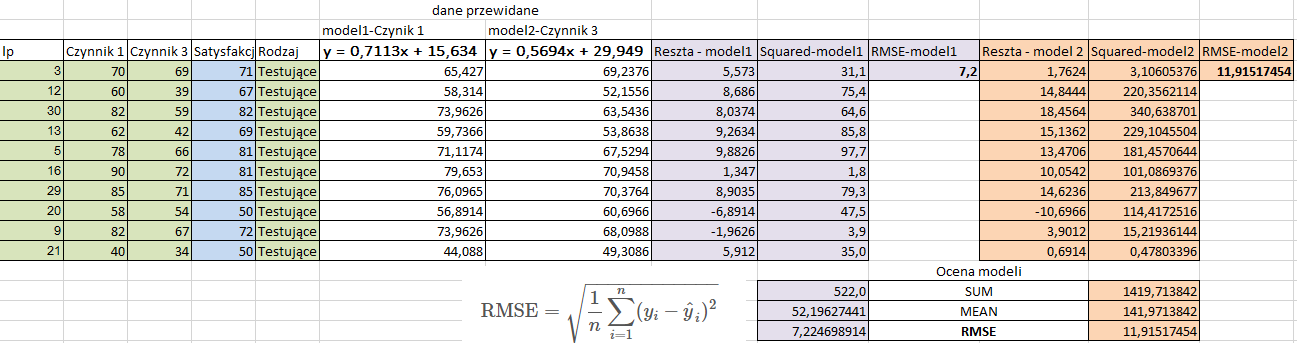
1. Następnie przewidujemy wartości na danych testujących nowo powstałymi modelami.

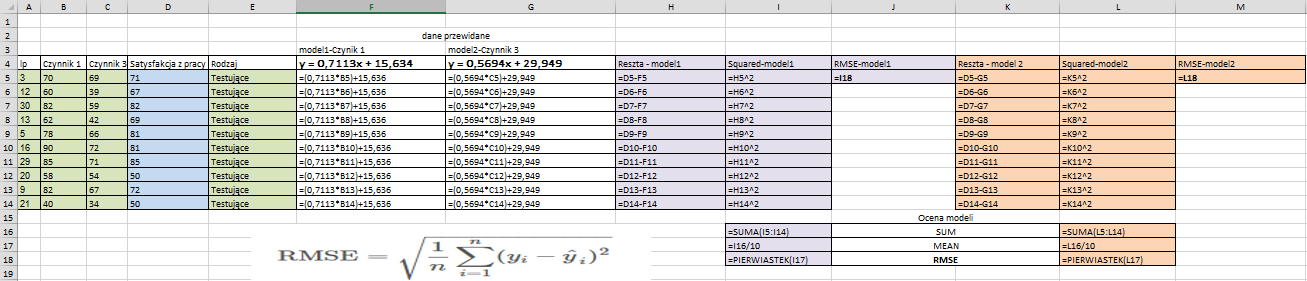






1. **Testowanie modeli**. Testujemy model na danych testujących. Metody, SSE, MAE, Root Mean Squared Error (RMSE). Mniejszy współczynnik=lepiej.





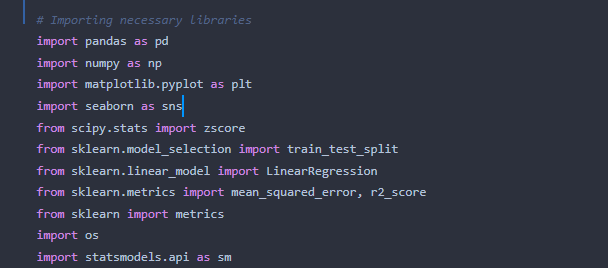
# Zadania

1. Dane dot. ceny mieszkań i ich charakterystyk
2. Załaduj dane odpowiednimi nagłówkami

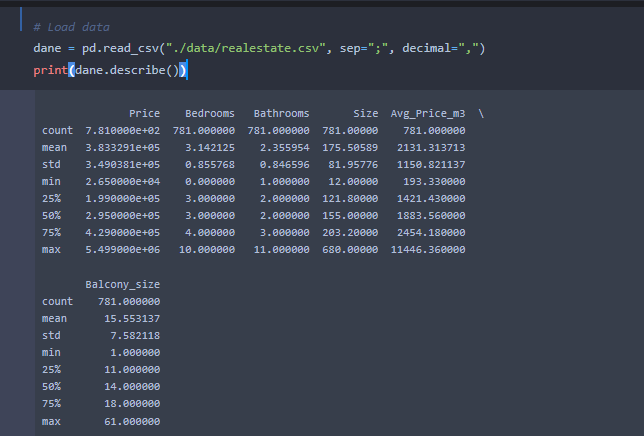
7. Attribute Information:

* Location - char
* Price - int
* Bedrooms- int
* Bathrooms- int
* Size - num
* Avg\_Price\_m3- num
* Balcony\_size – int

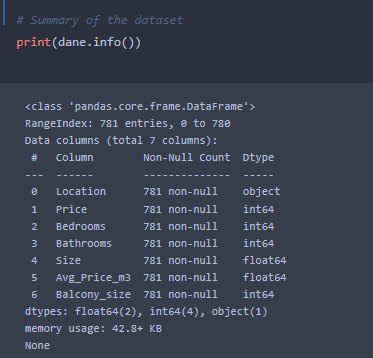
1. Wczytaj biblioteki



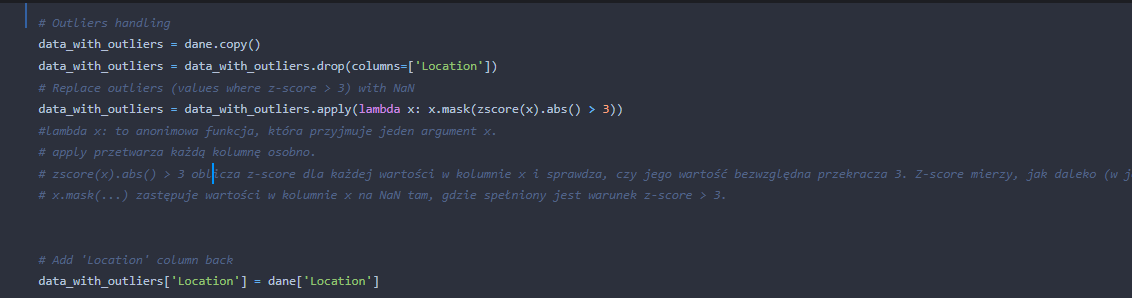
1. Wczytaj dane (realestate.csv) i Oblicz podstawowe statystyki

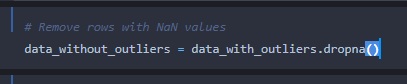


1. sprawdź strukturę wczytanego pliku

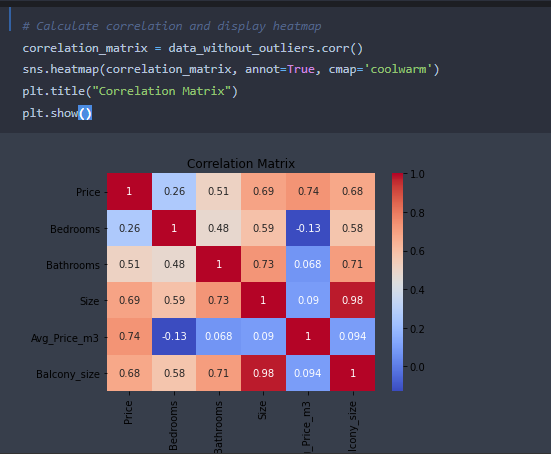


1. usuwanie wartości odstających

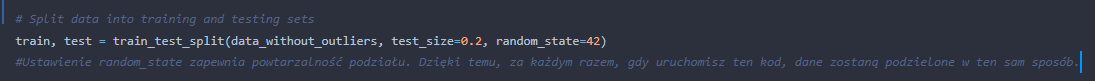


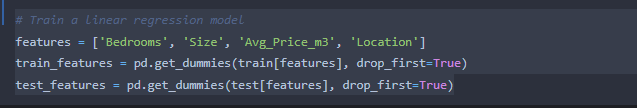


1. Sprawdź korelacje



1. Posortuj i podziel na dane budujące i testowe





Wyobraźmy sobie, że mamy zmienną kategoryczną Location z trzema kategoriami:

* City
* Suburb
* Rural

Kiedy używamy pd.get\_dummies bez drop\_first=True, dla każdej kategorii tworzymy oddzielną kolumnę wskaźnikową (dummy variable). Efekt kodowania wyglądałby tak:

| **Location** | **Location\_City** | **Location\_Suburb** | **Location\_Rural** |
| --- | --- | --- | --- |
| City | 1 | 0 | 0 |
| Suburb | 0 | 1 | 0 |
| Rural | 0 | 0 | 1 |

Tutaj każda kolumna Location\_City, Location\_Suburb i Location\_Rural przyjmuje wartość 1 lub 0, w zależności od lokalizacji. Jednak taka reprezentacja może powodować pułapkę współliniowości (dummy variable trap) w regresji liniowej. Jest to sytuacja, w której jedna kolumna można dokładnie przewidzieć na podstawie pozostałych kolumn, co prowadzi do nadmiaru informacji i problemów z estymacją parametrów modelu.

W powyższym przykładzie, jeśli znamy wartości dwóch kolumn (np. Location\_City i Location\_Suburb), to możemy jednoznacznie określić wartość trzeciej kolumny (Location\_Rural). Na przykład:

* Jeśli Location\_City = 0 i Location\_Suburb = 0, to Location\_Rural musi być 1.

**Rozwiązanie z drop\_first=True**

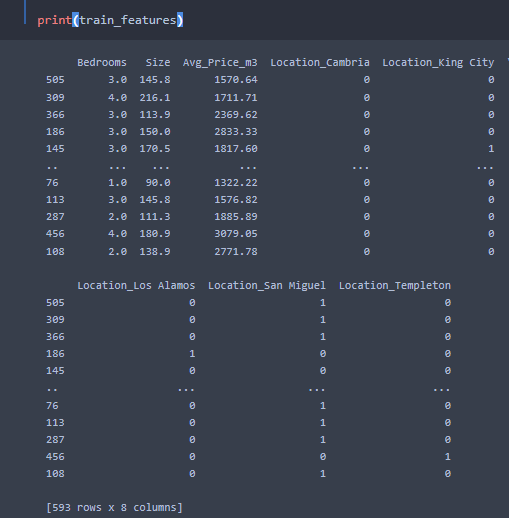
Aby uniknąć tego problemu, używamy drop\_first=True, co skutkuje usunięciem jednej kolumny. Kodując Location z drop\_first=True, otrzymujemy:

| **Location** | **Location\_Suburb** | **Location\_Rural** |
| --- | --- | --- |
| City | 0 | 0 |
| Suburb | 1 | 0 |
| Rural | 0 | 1 |

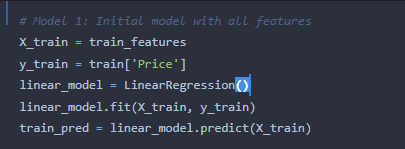
Kolumna Location\_City została usunięta. Teraz możemy odczytać każdą lokalizację:

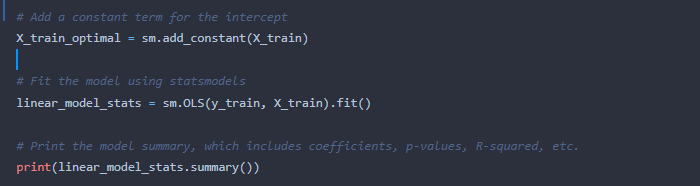
* Jeśli obie kolumny (Location\_Suburb i Location\_Rural) mają wartość 0, oznacza to City.
* Jeśli Location\_Suburb = 1, to jest to Suburb.
* Jeśli Location\_Rural = 1, to jest to Rural.

W ten sposób unikamy współliniowości, ponieważ każda lokalizacja jest unikalnie określona przez wartości w pozostałych kolumnach, a nasz model ma jednoznaczne, niezależne zmienne wskaźnikowe.

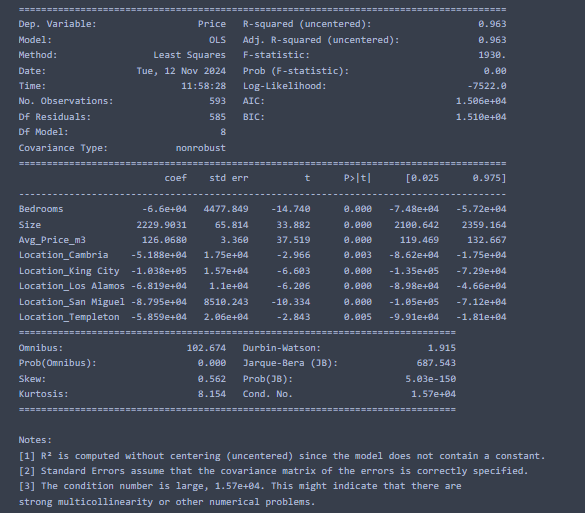


1. Stwórz model przewidujący zmienną - Price: estimated relative performance from the original article (integer)

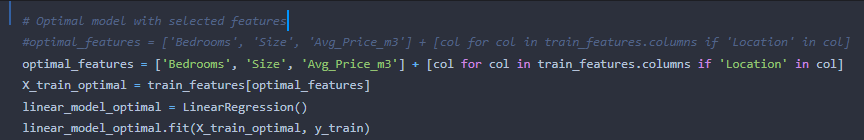




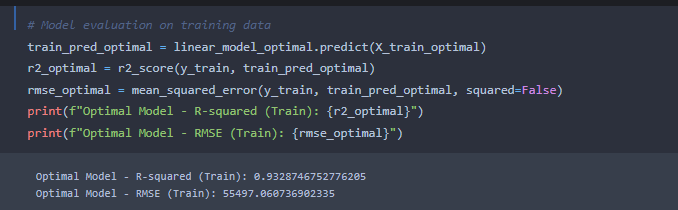
1. Oceń które z zmiennych są istotne i wybierz do optymalnego modelu



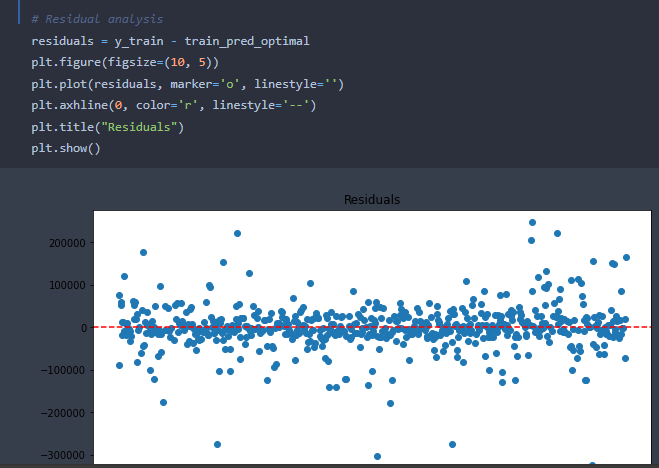
1. Stwórz model optymalny ( w razie potrzeby (gdyby pvalue (P>|t|) przekraczałaby wartość 0.05)

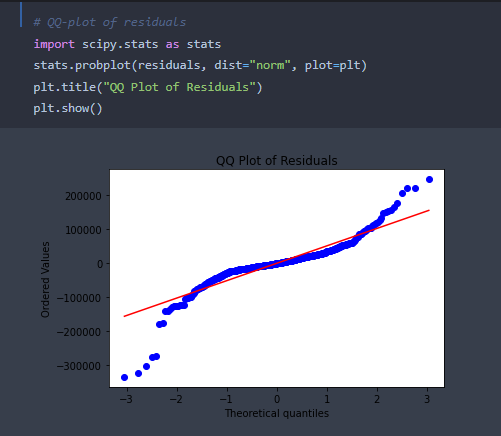


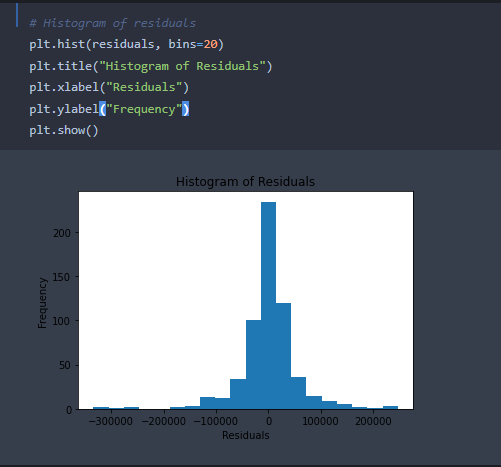
1. Oceń model



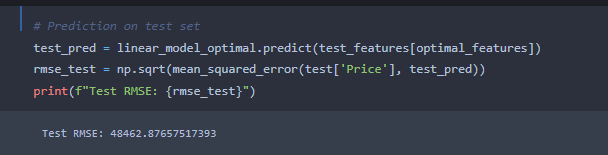
1. Ocena wartości reszt



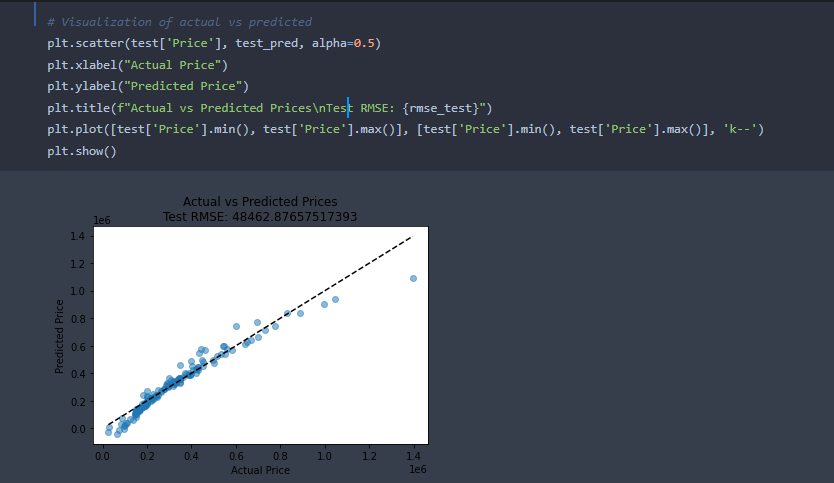




1. Przewidywania na danych testowych



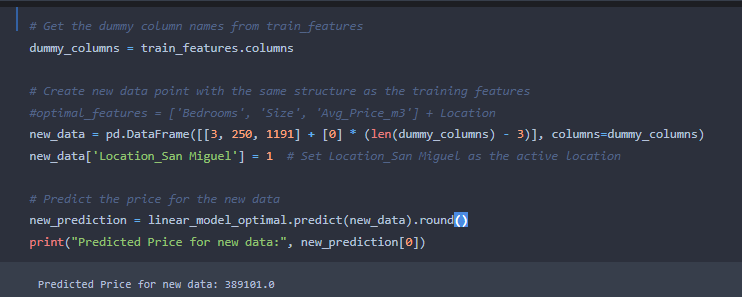
1. Wizualizacja



plt.plot(...) rysuje linię przerywaną ('k--') na przekątnej wykresu. Jest to linia idealnej prognozy, gdzie y\_true jest równe y\_pred, czyli wszystkie punkty leżą na tej linii, gdy model przewiduje ceny doskonale.

Użycie [test['Price'].min(), test['Price'].max()] dla obu osi X i Y tworzy linię od minimalnej do maksymalnej wartości rzeczywistych cen.

1. Testowanie modelu na 1 wierszu



Zadanie 2

Przygotuj ocenę modelu i przetestuj w podobnych krokach dwa modele przewidujące y1 i y2 – porównaj wyniki

<https://archive.ics.uci.edu/ml/datasets/Energy+efficiency>

Specifically:

* X1 Relative Compactness
* X2 Surface Area
* X3 Wall Area
* X4 Roof Area
* X5 Overall Height
* X6 Orientation
* X7 Glazing Area
* X8 Glazing Area Distribution
* y1 Heating Load
* y2 Cooling Load