Podstawy modelu regresji wielomianowej

Wstęp

Regresja wielomianowa jest potężnym narzędziem w statystyce i uczeniu maszynowym, pozwalającym modelować związki nieliniowe między zmiennymi niezależnymi a zmienną zależną. Podczas gdy regresja liniowa jest ograniczona do modelowania zależności liniowych, regresja wielomianowa rozszerza ten zakres, umożliwiając dopasowanie bardziej złożonych wzorców danych.

Teoria

Aby zgłębić teorię regresji wielomianowej, warto rozpocząć od zrozumienia, jak regresja wielomianowa rozszerza pojęcie regresji liniowej i jak jest stosowana do modelowania zależności nieliniowych między zmiennymi. Regresja wielomianowa jest formą regresji, która pozwala na użycie wielomianu - wyrażeń matematycznych zawierających zmienne podniesione do różnych potęg - do modelowania związku między zmienną niezależną a zmienną zależną.

Aby zgłębić teorię regresji wielomianowej, warto rozpocząć od zrozumienia, jak regresja wielomianowa rozszerza pojęcie regresji liniowej i jak jest stosowana do modelowania zależności nieliniowych między zmiennymi. Regresja wielomianowa jest formą regresji, która pozwala na użycie wielomianu - wyrażeń matematycznych zawierających zmienne podniesione do różnych potęg - do modelowania związku między zmienną niezależną a zmienną zależną.

Matematyczne Podstawy

Model regresji wielomianowej stopnia n dla jednej zmiennej niezależnej ma postać:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_n x^n + \varepsilon$$

gdzie:
- y jest zmienną zależną,
- x jest zmienną niezależną,
- $\beta_0, \beta_1, ..., \beta_n$ są parametrami modelu, które należy oszacować,
- n jest stopniem wielomianu,
- ε jest błędem, reprezentującym różnicę między wartościami przewidywanymi a rzeczywistymi.

Proces Dopasowania Modelu

Dopasowanie modelu regresji wielomianowej do danych polega na znalezieniu wartości parametrów $\beta_0, \beta_1, ..., \beta_n$, które minimalizują różnice między przewidywanymi a obserwowanymi wartościami. Typowo stosuje się metodę najmniejszych kwadratów, która polega na minimalizacji sumy kwadratów różnic (błędów) między wartościami przewidywanymi przez model a wartościami rzeczywistymi.

Złożoność Modelu vs. Nadmiarowe Dopasowanie

Podczas gdy dodawanie kolejnych stopni do wielomianu może poprawić dopasowanie modelu do danych treningowych, zwiększa to ryzyko nadmiernego dopasowania. Nadmierne dopasowanie występuje, gdy model jest zbyt skomplikowany, co skutkuje jego dobrą wydajnością na danych treningowych, ale słabą generalizacją na nowych, nieobserwowanych danych.

Wybór Stopnia Wielomianu

Wybór odpowiedniego stopnia wielomianu jest kluczowy dla skuteczności modelu. Zbyt niski stopień może nie uchwycić wszystkich zależności nieliniowych (niedopasowanie), podczas gdy zbyt wysoki stopień może prowadzić do nadmiernego dopasowania. Często stosowaną metodą jest przeprowadzenie analizy eksploracyjnej danych, testowanie modeli o różnych stopniach i ocenianie ich wydajności za pomocą walidacji krzyżowej lub innych technik oceny modelu.

Analiza i implementacja kodu

Analiza i implementacja kodu przedstawia szczegółowe wykorzystanie biblioteki scikit-learn do budowy modelu regresji liniowej. Poniżej przedstawiono kroki realizacji wraz z kodem i omówieniem kluczowych funkcji.

Import bibliotek

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Importujemy biblioteki *numpy* dla operacji matematycznych, *matplotlib.pyplot* do tworzenia wykresów oraz *pandas* do manipulacji danymi i ich wczytywania.

Wczytanie danych

```
dane = pd.read_csv('mieszkania3_1.csv')
X = dane.iloc[:, 0:1].values
y = dane.iloc[:, -1].values
```

Dane wczytujemy za pomocą *pandas.read_csv*, a następnie selekcjonujemy kolumny: *X* dla cech (tutaj powierzchnia mieszkań) i *y* dla wartości docelowych (ceny mieszkań).

Trening modelu regresji liniowej

```
from sklearn.linear_model import LinearRegression
model_reg_lin = LinearRegression()
model_reg_lin.fit(X, y)
```

Tworzymy instancję *LinearRegression* z *sklearn.linear_model* i używamy metody *.fit()* do trenowania modelu na danych treningowych.

Wizualizacja modelu regresji liniowej

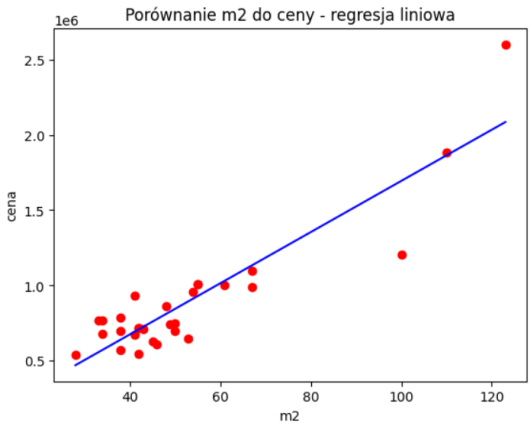
```
plt.scatter(X, y, color = 'red')
plt.plot(X, model_reg_lin.predict(X), color = 'blue')
plt.title('Porównanie m2 do ceny - regresja liniowa')
plt.xlabel('m2')
plt.ylabel('cena')
plt.show()
```

`plt.scatter` wywołanie tej funkcji tworzy wykres rozrzutu. Na wykresie rozrzutu punkty danych są reprezentowane przez kropki w przestrzeni dwuwymiarowej, gdzie jedna oś odpowiada jednemu zestawowi danych, a druga oś - drugiemu.

`plt.plot` tworzy wykres liniowy.

`model_reg_lin.predict(X)` wywołuje metodę `predict` na obiekcie modelu regresji liniowej. Wynikiem jest zbiór wartości przewidywanych dla danych `X`.

Te przewidywane wartości są następnie rysowane na wykresie jako linia, gdzie `color = 'blue'` określa kolor tej linii, w tym przypadku niebieski.Wizualizacja wyników - dane treningowe



Trening modelu regresji wielomianowej

```
from sklearn.preprocessing import PolynomialFeatures
```

Importuje klasę *PolynomialFeatures* z biblioteki *scikit-learn*, która służy do generowania nowej macierzy cech zawierającej wszystkie kombinacje wielomianowe cech z oryginalnego zestawu danych do określonego stopnia. Dzięki temu możliwe jest aplikowanie linearnych modeli uczenia maszynowego do rozwiązywania zadań wymagających odwzorowań nieliniowych.

```
model_pol = PolynomialFeatures(degree = 4)
```

Tworzy instancję *PolynomialFeatures*, ustawiając stopień wielomianu na 4. Oznacza to, że cechy będą kombinowane w wielomiany do czwartego stopnia (włącznie), co pozwala na modelowanie bardziej skomplikowanych zależności nieliniowych między zmiennymi niezależnymi a zmienną zależną.

```
X_pol = model_pol.fit_transform(X)
```

Używa metody *fit_transform* na obiekcie P*olynomialFeatures* utworzonym wcześniej, aby przekonwertować oryginalne cechy *X* na ich wielomianową formę i przypisać je do zmiennej *X_wie*. Dzięki temu, zwiększa się wymiarowość przestrzeni cech, co umożliwia lepsze oddanie nieliniowych zależności.

```
model_wie = LinearRegression()
```

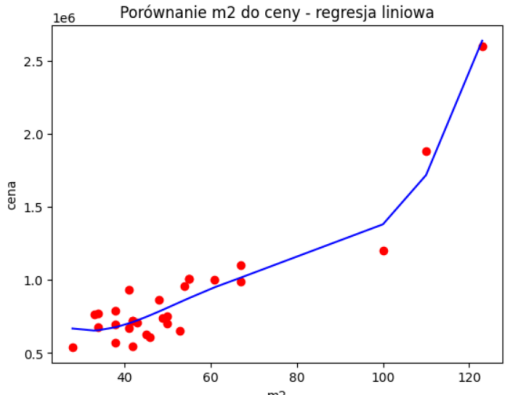
Ponownie wykorzystuje nazwę zmiennej *model_wi*e do utworzenia instancji modelu regresji liniowej. Jest to możliwe dzięki temu, że regresja wielomianowa w istocie wykorzystuje model liniowy do pracy z cechami przetransformowanymi do postaci wielomianowej.

```
model_wie.fit(X_pol, y)
```

Metoda *fit* jest używana do trenowania modelu regresji liniowej na zbiorze cech *X_wie*, które są teraz w formie wielomianowej, oraz na zmiennej zależnej *y*. Model uczy się najlepiej dopasować danych wielomianowych do przewidywanych wartości, co pozwala na osiągnięcie dopasowania nieliniowego za pomocą modelu liniowego.

Wizualizacja modelu regresji wielomianowej

```
plt.scatter(X, y, color = 'red')
plt.plot(X, model_wie.predict(model_pol.fit_transform(X)), color = 'blue')
plt.title('Porównanie m2 do ceny - regresja liniowa')
plt.xlabel('m2')
plt.ylabel('cena')
plt.show()
```



Druga linia kodu prezentuje sposób na narysowanie wykresu, który pokazuje wyniki predykcji modelu regresji wielomianowej. Omówimy tylko ją:

`plt.plot`: Ta funkcja z pakietu `matplotlib.pyplot` tworzy wykres liniowy. Jej głównym celem jest połączenie punktów danych w ciągły wykres, w tym przypadku linie reprezentujące przewidziane wartości modelu.

`X`: Reprezentuje dane osi X, które są wielkościami mieszkań wyrażonymi w metrach kwadratowych.

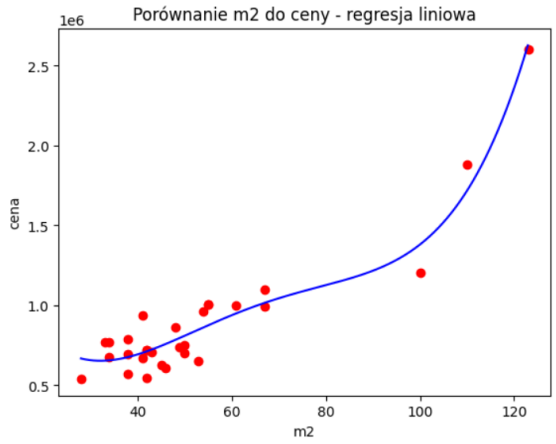
`model_wie.predict(...)`: Jest to wywołanie metody `predict` na modelu regresji wielomianowej, który oznaczony jest tutaj jako `model_wie`. Ta metoda jest używana do wygenerowania przewidywanych wartości `y` na podstawie danych wejściowych `X`.

`model_pol.fit_transform(X)`: Tutaj przekształcamy dane wejściowe `X` za pomocą metody `fit_transform` obiektu transformacji wielomianowej, który oznaczony jest jako `model_pol`. Ta metoda tworzy nowe cechy będące wielomianowymi kombinacjami oryginalnych cech (w tym przypadku stopni metrów kwadratowych), które są następnie używane przez model regresji wielomianowej do przewidywania.

Wizualizacja modelu regresji wielomianowej z dużą dokładnością

```
X_siatka = np.arange(min(X), max(X), 0.1)
X_siatka = X_siatka.reshape((len(X_siatka), 1))

plt.scatter(X, y, color = 'red')
plt.plot(X_siatka, model_wie.predict(model_pol.fit_transform(X_siatka)), color = 'blue')
plt.title('Porównanie m2 do ceny - regresja liniowa')
plt.xlabel('m2')
plt.ylabel('cena')
plt.show()
```



Ten fragment kodu przygotowuje dane dla celów wizualizacji lub predykcji modelu, transformując jednowymiarowy array (wektor) X, który zawiera wartości zmiennej niezależnej, na szczegółową, ciągłą siatkę wartości. Używany jest głównie w kontekście modelowania i wizualizacji predykcji dla modeli regresji. Oto dokładny opis krok po kroku:

```
X_siatka = np.arange(min(X), max(X), 0.1)
```

Tworzy array *X_siatka*, który jest ciągłym zakresem wartości od minimalnej wartości w *X* do maksymalnej, z krokiem co 0.1. Posłuży to do stworzenia gęstej siatki punktów na osi X, co jest przydatne do generowania płynnej linii w wizualizacji modelu. Krokiem w *numpy.arange(start, stop, step)* jest 0.1, co oznacza, że każda kolejna wartość jest o 0.1 większa od poprzedniej, aż do osiągnięcia wartości bliskiej maksymalnej wartości w *X*.

```
X_siatka = X_siatka.reshape((len(X_siatka), 1))
```

Ten krok przekształca siatka z jednowymiarowego array (wektora) w dwuwymiarową macierz o len(X_grid) wierszach i 1 kolumnie. Operacja reshape zmienia kształt array bez zmiany danych. Transformacja do formy dwuwymiarowej jest często wymagana przez algorytmy uczenia maszynowego z biblioteki scikit-learn, które oczekują danych wejściowych X w formie dwuwymiarowej (matrycy), nawet jeśli model przewiduje tylko na podstawie jednej zmiennej niezależnej.

```
plt.plot(X_siatka, model_wie.predict(model_pol.fit_transform(X_siatka)), color = 'blue')
```

W tym miejscu zamieniamy *X* na *X_siatka* dzięki czemu mamy dostęp do większej ilości punktów i wykres jest bardziej łagodny.

Podsumowanie

Regresja wielomianowa to rozszerzenie regresji liniowej, umożliwiające modelowanie zależności nieliniowych między zmiennymi. Poprzez wprowadzenie wyższych potęg zmiennej niezależnej, regresja wielomianowa jest w stanie uchwycić bardziej złożone wzorce w danych, co czyni ją potężnym narzędziem w przypadkach, gdzie zależność między

zmiennymi jest nieliniowa. Wybór odpowiedniego stopnia wielomianu jest kluczowy, aby uniknąć nadmiernego dopasowania i zachować zdolność generalizacji modelu na nowych danych. Pomimo swojej elastyczności, regresja wielomianowa wymaga ostrożnego stosowania i odpowiedniej walidacji, aby zapewnić skuteczne i rzetelne wyniki.