Regresja liniowa

Semestr letni 2021/22

Prosta regresja liniowa

Używam zbioru danych `Bangalore` dostępnego na stronie `https://www.kaggle.com/datasets/ruchi798/housing-prices-`

```
dane <- read.csv("Bangalore.csv")
names(dane)
dim(dane)
??dane
head(dane)
```

Dopasowanie (uczenie) modelu liniowego wykonuje się przy pomocy funkcji `lm()`. Postać modelu określa się przy pomocy **formuły** (czyli obiektu klasy `formula`). Modelowi

$$Y = \beta_0 + \beta_1 X + \epsilon$$

odpowiada formuła `Y ~ X`. Poniższe instrukcje są równoważne i oznaczają model

$$medv = \beta_0 + \beta_1 \cdot lstat + \epsilon.$$

```
fit_simple <- lm(Price ~ Area, data = dane)
summary(fit_simple)
```

Analiza wyników sugeruje, że zmienna **`Area`** ma istotny wpływ na cenę mieszkań (Price), ponieważ p-wartość (< 0.05) dla współczynnika **`Area`** jest bardzo mała. Współczynnik determinacji (R-squared) wynosi 0.1581, co oznacza, że tylko około 15.81% zmienności w cenie mieszkań jest wyjaśnione przez zmienną **`Area`** w tym modelu regresji liniowej.

Natomiast poniższa ma działanie szersze

```
attach(dane)
fit_simple <- lm(Price ~ Area)
detach(dane)
```

Wynikiem w każdym przypadku jest obiekt klasy `lm`, który jest też listą

```
fit_simple
class(fit_simple)
is.list(fit_simple)
names(fit_simple)
```

Składowe obiektu modelu liniowego są dostępne przez indeksowanie typu listowego lub przez odpowiednie funkcje/metody akcesorowe (co jest metodą zalecaną), np.

```
fit_simple$coefficients
coef(fit_simple)
```

Dodatkowe informacje można uzyskać przy pomocy funkcji `summary()`

```
?summary.lm
summary(fit_simple)
```

Funkcja `summary()` zwraca listę (składowa `sigma` to RSE)

```
summaryList <- summary(fit_simple)
summaryList$sigma
summaryList$r.squared
summaryList$fstatistic
```

Podsumowanie wyników regresji liniowej dostarcza istotnych informacji na temat dopasowania modelu do danych. Wartość odchylenia standardowego reszt (sigma) wynosząca około 12,947,121 jednostek waluty wskazuje na rozproszenie reszt od linii regresji, co sugeruje, że model nie jest doskonale dopasowany do danych. Współczynnik determinacji (R-squared) o wartości około 0.1581 oznacza, że zmienne niezależne w modelu wyjaśniają około 15.81% zmienności zmiennej zależnej, co sugeruje, że model nie tłumaczy znacznej części zmienności. Natomiast istotność regresji jako całości potwierdza statystyka F o wartości około 1165.29, co sugeruje, że przynajmniej jedna zmienna niezależna ma istotny wpływ na zmienną zależną.

Aby zoptymalizować działanie regresji, podjęto decyzję o wyłączeniu kolumn zawierających wiersze, w których występuje liczba 9. Wiersze te zostały uznane za zawierające błędne lub brakujące dane. Pomimo tego, w ramach procesu oczyszczania danych zachowano kolumny i wiersze, które zawierają istotne i potrzebne informacje.

```
dane_bez_9 <- dane[!apply(dane, 1, function(row) any(grepl("9", as.character(row)))), ]

fit_simple <- lm(Price ~ Area, data = dane_bez_9)
summary(fit_simple)
```

Ten model regresji liniowej sugeruje, że powierzchnia (Area) ma istotny wpływ na cenę (Price) mieszkań w zbiorze danych "dane_bez_9". Współczynnik determinacji wynoszący około 82,37% wskazuje, że model dobrze dopasowuje się do danych.

Przedziały ufności dla współczynników regresji oblicza funkcja `confint()`

```
confint(fit_simple)
```

- **(Intercept)**: Przedział ufności dla intercepta wynosi od około -12 994 229,19 do -11 653 781,77. Oznacza to, że możemy być pewni z 95% pewnością, że prawdziwa wartość intercepta dla populacji znajduje się w tym zakresie.
- **Area**: Przedział ufności dla współczynnika zmiennej Area wynosi od około 14 167,85 do 14 960,69. To oznacza, że możemy być pewni z 95% pewnością, że prawdziwa wartość współczynnika dla zmiennej Area dla populacji znajduje się w tym zakresie.

Przedziały ufności są istotne, ponieważ pozwalają nam ocenić precyzję estymacji współczynników regresji. Im szerszy przedział ufności, tym mniej precyzyjna jest estymacja współczynnika. Dlatego ważne jest, aby przedziały ufności były możliwie jak najmniejsze, co oznacza większą pewność co do prawdziwej wartości parametru populacji.

Funkcja `predict()` oblicza przedziały ufności dla predykcji — zarówno dla przewidywania średniej wartości

```r
predict(fit_simple, newdata = data.frame(Area = c(5, 10, 15)), interval = "confidence")
```

jak i dla przewidywania przyszłej wartości

```r
predict(fit_simple, newdata = data.frame(Area = c(5, 10, 15)), interval = "prediction")
```

Dla każdej z wartości zmiennej niezależnej (5, 10 i 15 jednostek), funkcja **predict()** zwraca przewidywaną wartość zmiennej zależnej (fit) oraz przedział ufności (confidence interval), w którym z określonym prawdopodobieństwem (najczęściej 95%) znajduje się prawdziwa wartość przewidywanej zmiennej.

Wykresy prostej regresji liniowej

Prosta regresji na tle danych

```r
plot(dane_bez_9$Area, dane_bez_9$Price,
     xlab = "Powierzchnia", ylab = "Cena mieszkania",
     main = "Prosta regresja liniowa: Cena mieszkania od powierzchni")
abline(fit_simple, , col = "red")
```

Wykresy diagnostyczne

```r
par(mfrow = c(2, 2))
plot(fit_simple)
```

1. **Wykres rozrzutu reszt vs. wartości przewidywane**: Pozwala on sprawdzić, czy wariancja reszt jest stała wzdłuż zakresu wartości przewidywanych. W zasadzie powyższy wykres pokazuje losowe rozrzucenie punktów wokół osi poziomej, co sugeruje, że założenia regresji liniowej są spełnione.
2. **Wykres kwantylowy reszt**: Pozwala na ocenę, czy reszty są zgodne z rozkładem normalnym. Powyższy wykres pokazuje linię prosta, jedynie wartości boczne są lekko odchylone, co sugeruje, że reszty są rozkładem normalnym.
3. **Wykres wpływu (Leverage vs. Residuals)**: Pozwala na identyfikację obserwacji o dużym wpływie na dopasowanie modelu. Obserwacje skupiają się w lewym dolnym rogu, może to być spowodowane przez heteroskedastyczność (niestałość wariancji reszt wzdłuż zakresu wartości przewidywanych przez model), zmienną niezależną nieuwzględnioną w modelu czy też obserwacje odstające
4. **Wykres wpływu (Leverage vs. Cook's distance)**: Pokazuje, które obserwacje mają największy wpływ na współczynniki regresji. Obserwacje skupiają się w lewym dolnym rogu, może to być spowodowane przez heteroskedastyczność (niestałość wariancji reszt wzdłuż zakresu wartości przewidywanych przez model), zmienną niezależną nieuwzględnioną w modelu czy też obserwacje odstające

Identyfikacja obserwacji wpływowych (statystyka "dźwigni" [*leverage*])

```r
plot(hatvalues(fit_simple),
     xlab = "Indeks obserwacji", ylab = "Wartość hat",
     main = "Identyfikacja obserwacji wpływowych")
which.max(hatvalues(fit_simple))
```

Na wykresie zauważyć można wartości odstające, jest ich niewiele. Reszta z obserwacji skupia się wśród jednej poziomej linii.

Regresja wielokrotna

Model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

reprezentowany jest przez formułę `Y ~ X1 + X2 + X3`, np.

```
fit_la <- lm(Price ~ Area + Bedrooms, data = dane_bez_9)
summary(fit_la)
```

Ogólnie rzecz biorąc, wyniki sugerują, że zarówno powierzchnia mieszkania (Area), jak i liczba sypialni (Bedrooms) mają istotny wpływ na cenę mieszkania (Price), przy czym większa powierzchnia i liczba sypialni zazwyczaj prowadzą do wyższej ceny mieszkania.

Jeśli chcemy wykonać regresję pewnej zmiennej względem wszystkich pozostałych stosuje się składnię (parametr `data` jest tu wymagany)

```
fit_all <- lm(Price ~ ., data = dane_bez_9)
summary(fit_all)
```

Ogólnie rzecz biorąc, model **`fit_all`** wydaje się dobrze dopasowany do danych, a większość zmiennych objaśniających jest istotna statystycznie. Jednak z uwagi na dużą liczbę zmiennych w modelu, istnieje ryzyko nadmiernego dopasowania, co może prowadzić do utraty ogólności i trudności w generalizacji wyników na nowe dane.

Regresja z jedną zmienną usuniętą

```
fit_no_age <- lm(Price ~ . - Bedrooms, data = dane_bez_9)
summary(fit_no_age)
```

- **Wysoka wartość R-squared** (0.9098) sugeruje, że model dobrze tłumaczy zmienność cen nieruchomości.
- **Istotność lokalizacji**: Lokalizacja nieruchomości jest jednym z najważniejszych czynników wpływających na jej cenę.
- **Istotność cech**: Niektóre cechy (np. ogrody, backup mocy) mają znaczący wpływ na cenę, co sugeruje, że takie udogodnienia mogą być ważne dla potencjalnych kupujących.

Alternatywnie można skorzystać z funkcji `update()`

```
fit_no_age2 <- update(fit_all, ~ . - Bedrooms)
summary(fit_no_age2)
```

Zbiór ufności dla dwóch współczynników można obliczyć korzystając np. z funkcji `ellipse()` z pakietu `ellipse`.

```
library(ellipse)
plot(ellipse(fit_la, which = -1), type = "l")
la_coefs <- coef(fit_la)
points(la_coefs[2], la_coefs[3])
```

Elipsa jest wydłużona w kierunku `Area`, co wskazuje na większą zmienność szacunków tego współczynnika w porównaniu z `Bedrooms`. Punkt rzeczywistych współczynników znajduje się w środku elipsy co oznacza, że są one dobrze zdefiniowane i mieszczą się w granicach ufności.

Interakcje między zmiennymi

Obecność składnika $X_1 \cdot X_2$ zaznacza się w formule przez człon `X1 : X2`. Składnia `X1 * X2` jest skrótem do `X1 + X2 + X1:X2`. Np.

```
summary(lm(Price ~ Area * Bedrooms, data = dane_bez_9))
```

Nieliniowe transformacje predyktorów

Model z kwadratową zależnością od `lstat`, czyli

$$medv = \beta_0 + \beta_1 \cdot lstat + \beta_2 \cdot lstat^2 + \epsilon$$

dopasowywany jest następująco (funkcja `I()` jest konieczna ze względu na specjalne znaczenie operatora `^` w formułach)

```
fit_l2 <- lm(Price ~ Area + I(Area^2), data = dane_bez_9)
summary(fit_l2)
```

Dopasowanie modeli `fit_simple` i `fit_l2` można porównać porównując RSE i R^2. Funkcja `anova()` wykonuje test statystyczny, w którym hipotezą zerową jest jednakowe dopasowanie.

```
anova(fit_simple, fit_l2)
```

Regresja wielomianowa wyższego stopnia może wykorzystywać funkcję `poly()`

```
fit_l5 <- lm(Price ~ poly(Area, 5), data = dane_bez_9)
summary(fit_l5)
```

Logarytmiczna transformacja predyktora

```
summary(lm(Price ~ log(Area), data = dane_bez_9))
```

- **Kwadratowa zależność**: Model kwadratowy (**`fit_l2`**) poprawia dopasowanie (ma wyższe R-squared i niższe RSE) w porównaniu do modelu liniowego (**`fit_simple`**), oznacza to, że istnieje nieliniowa zależność między **`Area`** a **`Price`**.
- **Regresja wielomianowa**: Wyższe stopnie wielomianu (np. piąty stopień) mogą lepiej dopasować dane, ale mogą też prowadzić do nadmiernego dopasowania (overfitting). Należy uważać na interpretację wyników.
- **Logarytmiczna transformacja**: Może być użyteczna, gdy zmienność ceny zmniejsza się z rosnącą powierzchnią, co może lepiej pasować do rzeczywistych danych.

Predyktory jakościowe

Zbiór `Bangalore` zawiera zmienne jakościowe (czynniki)

```
summary(dane_bez_9)
```

Dla czynników generowane są automatycznie zmienne zastępcze, np.

```r
sales_all_ia_fit <- lm(Price ~ . + Area:Gymnasium, data = dane_bez_9)
summary(sales_all_ia_fit)
```

Funkcja `contrasts()` pokazuje kodowanie używane przez `R` dla zmiennych zastępczych.

```r
dane_bez_9$MaintenanceStaff <- as.factor(dane_bez_9$MaintenanceStaff)
contrasts(dane_bez_9$MaintenanceStaff)
```

- **Zmienne jakościowe**: Możliwość uwzględnienia zmiennych jakościowych w modelu regresji pozwala na bardziej kompleksową analizę wpływu różnych czynników na zmienną zależną.
- **Interakcje**: Analizowanie interakcji między zmiennymi jakościowymi a ilościowymi (np. **`Area:Gymnasium`**) może ujawnić dodatkowe informacje o złożonych relacjach w danych.
- **Kodowanie zmiennych**: Zrozumienie, jak zmienne jakościowe są kodowane jako zmienne zastępcze, jest kluczowe dla prawidłowej interpretacji wyników modelu regresji.