Mnohorozměrná analýza dat

Povaha vybraných dat k zpracování:

Tento dataset obsahuje údaje o různých charakteristikách městských částí Bostonu. Boston dataset obsahuje **506 řádků a 14 sloupců**. Každý řádek reprezentuje jednu městskou část (nebo obec) v oblasti Bostonu, a každý sloupec obsahuje jednu z následujících proměnných popisujících různé charakteristiky dané oblasti. Níže uvádím popis jednotlivých proměnných v datasetu:

1. **crim**: Míra kriminality na osobu podle města.
2. **zn**: Procento rezidenčních pozemků určených pro pozemky větší než 25 000 čtverečních stop.
3. **indus**: Procento nekomerčních podnikatelských ploch na město.
4. **chas**: Dummy proměnná týkající se řeky Charles (1, pokud daná městská část sousedí s řekou; 0 jinak).
5. **nox**: Koncentrace oxidů dusíku (v částech na 10 milionů).
6. **rm**: Průměrný počet pokojů na bytovou jednotku.
7. **age**: Procento obydlených domů postavených před rokem 1940.
8. **dis**: Vážený průměr vzdáleností k pěti hlavním zaměstnaneckým centrům v Bostonu.
9. **rad**: Index přístupnosti k radiálním dálnicím.
10. **tax**: Daňová sazba na plně ohodnocený majetek na $10,000.
11. **ptratio**: Poměr žáků na učitele podle města.
12. **black**: Výpočet založený na podílu černošského obyvatelstva v daném městě, podle vzorce $1000 \cdot (B_! - 0.63)^"$, kde $B_!$ je podíl černošské populace.
13. **lstat**: Procento obyvatelstva s nižším společenským statusem.
14. **medv**: Mediánová hodnota domů vlastněných obyvateli v tisících dolarech.

Řešení:

Prvním krokem při práci s daty bylo prozkoumání základní struktury datasetu. Použila jsem příkaz summary(Boston) v **RStudiu**, který zobrazil základní statistické charakteristiky všech proměnných v datasetu **Boston**.

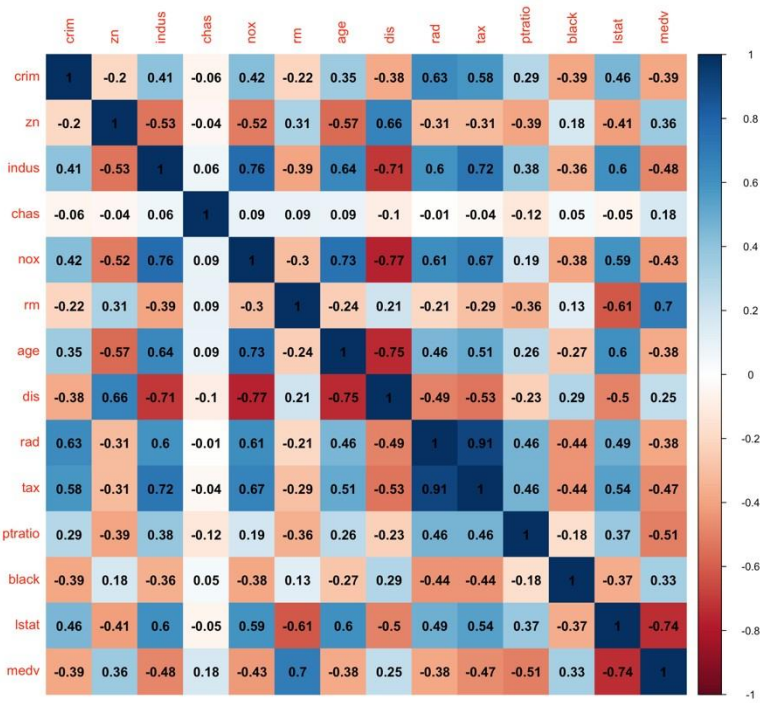
> summary(Boston)

Na základě tohoto přehledu jsem zjistila, že některé proměnné, jako **crim** nebo **zn** mají široký rozsah hodnot. Také proměnná **chas** je binární. Z tohoto důvodu jsem se rozhodla data standardizovat, aby se odstranily rozdíly v měřítkách a zlepšila se výkonnost budoucích regresních modelů.

```
> summary(Boston)
      crim                zn              indus            chas              nox               rm             age              dis        
 Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000   Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
 1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000   1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
 Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000   Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
 Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917   Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
 3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000   3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
 Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000   Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
      rad              tax           ptratio          black            lstat            medv      
 Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32   Min.   : 1.73   Min.   : 5.00  
 1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38   1st Qu.: 6.95   1st Qu.:17.02  
 Median : 5.000   Median :330.0   Median :19.05   Median :391.44   Median :11.36   Median :21.20  
 Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67   Mean   :12.65   Mean   :22.53  
 3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23   3rd Qu.:16.95   3rd Qu.:25.00  
 Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90   Max.   :37.97   Max.   :50.00  
```

Po standardizaci dat a vytvoření korelační matice máte k dispozici informace o vzorcích korelací mezi různými proměnnými v datasetu Boston. Standardizace zajišťuje, že všechny proměnné mají stejný průměr (0) a stejnou směrodatnou odchylku (1), což znamená, že korelace mezi proměnnými není ovlivněna jejich měřítkem.

Interpretace Korelační Matice



Obrázek 1. Korelační Matice

1. **Výrazně negativní korelace:**
   - **dis** (vzdálenost k pracovnímu místu) **a indus** (průmyslová koncentrace) = -0.71
   - **dis** (vzdálenost k pracovnímu místu) **a nox** (znečištění) = -0.77
   - **dis** (vzdálenost k pracovnímu místu) **a age** (věk nemovitostí) = -0.75
   - **medv** (mediánová cena domů) a **lstat** (obyvatelstvo s nižším společenským statusem) = -0.74

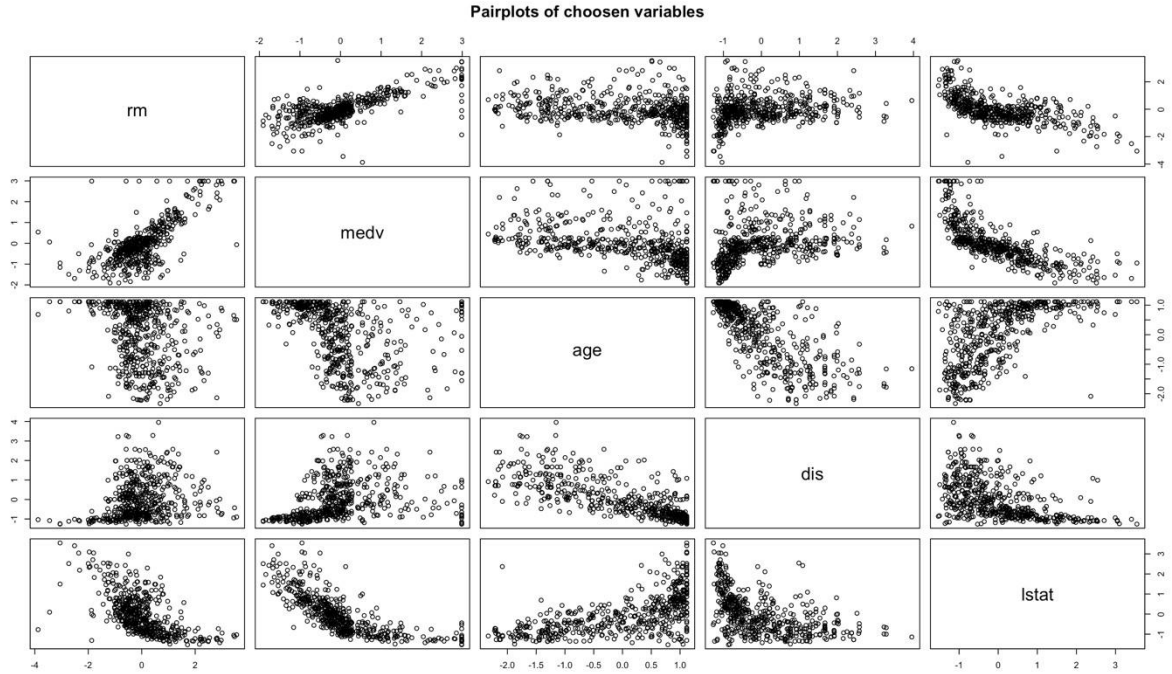
To naznačuje, že oblasti s vyšší průmyslovou koncentrací a znečištěním mají často menší vzdálenost k pracovnímu místu. Starší nemovitosti mají obvykle nižší ceny domů a větší vzdálenost k pracovnímu místu.

2. **Pozitivní korelace:**
   - **tax** (daňová sazba) **a rad** (index přístupnosti k radiálním dálnicím) = 0.91
   - **tax** (daňová sazba) **a indus** (průmyslová koncentrace) = 0.72
   - **rm** (průměrný počet pokojů) **a medv** (mediánová cena domů) = 0.7

To znamená, že oblasti s vyššími daňovými sazbami mohou mít lepší přístup k radiálním dálnicím a vyšší průmyslovou koncentraci. Vyšší průměrný počet pokojů je také spojen s vyššími cenami domů.

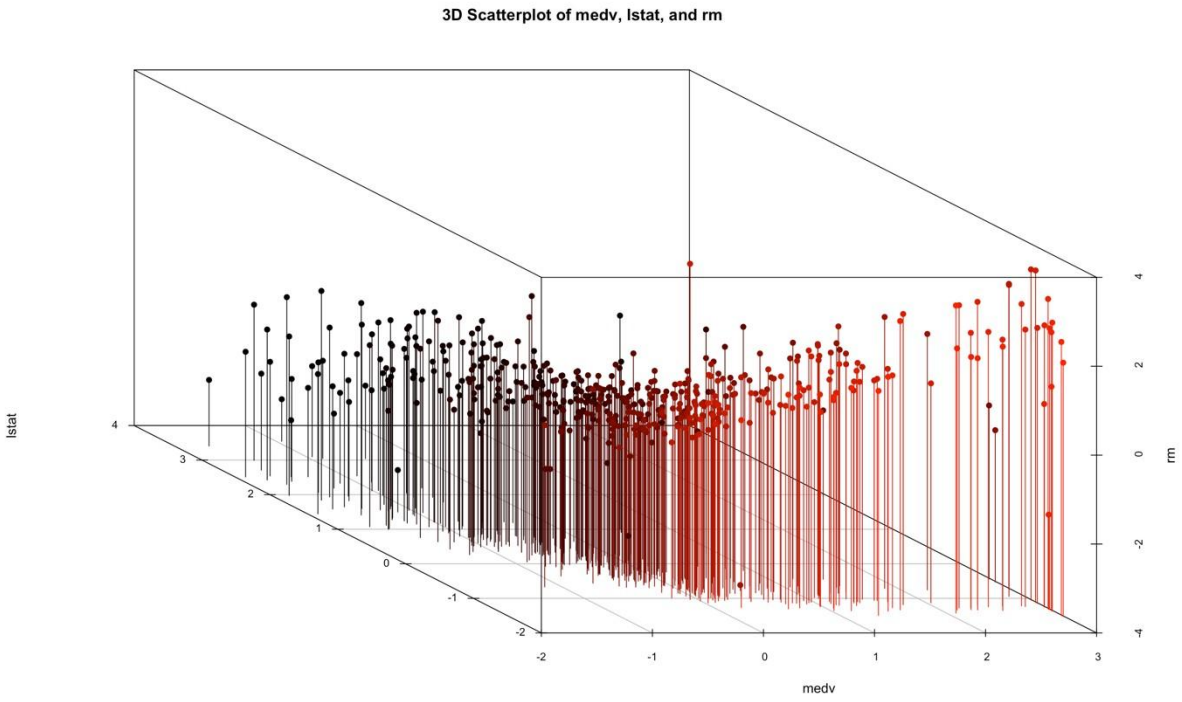
Vztah mezi počtem pokojů a cenou domů: na základě párových grafů, které jsme vytvořili, je zřejmé, že mediánová cena domů má lineární vzestupnou tendenci s rostoucím počtem pokojů. Tento vztah naznačuje, že domy s větším počtem pokojů mají tendenci být dražší.

Vztah mezi sociálním statusem (lstat) a cenou domů (medv): grafy ukazují, že obyvatelé s nižším sociálním statusem (měřeným proměnnou lstat) se častěji nacházejí v nemovitostech s nižší mediánovou cenou (medv).



Obrázek 2. Párové grafy

Také na závěr pro vizualizaci jsem vytvořila 3D scatterplot, ukazující závislost mezi: *medv*, *lstat* a *rm.* Na tomto obrázku je líp vidět jak souvisí proměnná medv s počtem pokojů a sociálním statusem.



Obrázek 3. 3D-scatterplot mezi *medv*, *rm* a *lstat*

Implementace modelu:

Lineární regrese:

Jako první regresní metodu jsem zvolila lineární regresi. Pro implementaci modelu jsem použila už dříve standardizovaná data, která jsem rozdělila pomocí balíčku *'caret'* na dvě poloviny: ***TrainingSet, TestingSet***, kde pro trénování bylo použité **80%** dat a zbytek (**20%**) pro testování. Jako target proměnná byla zvolena *'medv'* (mediální cena nemovitosti) a ostátní jsou považovány jako features.

```
# Models implementation:
library(caret)
set.seed(123)
# Create a training index based on 80 % of the data using only the target variable (medv)
TrainingIndex <- createDataPartition(boston_standardized$medv, p = 0.8, list = FALSE)

# Split the boston_standardized dataset into the training set (80 %) and the testing set (20 %)
TrainingSet <- boston_standardized[TrainingIndex, ]  # 80 % of data for training
TestingSet <- boston_standardized[-TrainingIndex, ]  # Remaining 20 % of data for testing
```

Po rozdělení datasetu jsem nastavila funkci z balíčku pro vytvoření modelu lineární regresi. Nicméně kromě obyčejného modelu jsem také udělala i model s použitím **křížové cross-validace** (10-folds cross-validation).

```
# Linear regression model

Model_linr <- train(medv ~., data = TrainingSet,
                    method = "lm",
                    trControl = trainControl(method = "none"),
                    tuneGrid = expand.grid(intercept = FALSE)
                    )

Model_linr.cv <- train(medv ~., data = TrainingSet,
                       method = "lm",
                       trControl = trainControl(method = "cv", number = 10), # 10-fold cross-validation
                       tuneGrid = expand.grid(intercept = FALSE)
                       )

# Applying the model:

Model.training <- predict(Model_linr, TrainingSet)
Model.testing <- predict(Model_linr, TestingSet)
Model.training.cv <- predict(Model_linr.cv, TrainingSet)
```

Po aplikaci modelu na trénovací a testovací datasety, zjistila jsem následující výsledky:

> print(RMSE_perfomance_train)
[1] 0.512915

>
print(R_square_train)
[1] 0.7346007

> print(RMSE_perfomance_test) [1]
0.4988922

>
print(R_square_test)
[1] 0.761151

Hodnoty RMSE jsou poměrně nízké (kolem 0.51 pro tréninkovou a 0.50 pro testovací sadu). RMSE vyjadřuje průměrnou velikost chyb mezi předpovězenými a skutečnými hodnotami. Nižší hodnoty RMSE znamenají, že model vytváří relativně přesné předpovědi. Hodnoty RMSE pro tréninkovou i testovací sadu jsou si velmi podobné, což naznačuje, že model se dobře generalizuje a nedochází k jevům jako ‚overfitting' nebo ‚underfitting'.

Hodnota R^2 pro tréninkovou sadu je 0.7346 a pro testovací sadu je 0.7612. R^2 ukazuje, jak dobře model vysvětluje variabilitu v datech. Hodnoty v rozsahu kolem 0.73 a 0.76 naznačují, že model dobře vysvětluje zhruba 73 % až 76 % variability v datech, což je solidní výsledek.

> print(RMSE_perfomance_train_cv)
[1] 0.5129162

> print(R_square_train_cv)
[1] 0.7346007

Také u modelu s použitím křížové validace je vidět hodně podobné výsledky, což říká, že předešlé výsledky nebyly přehnaně optimistické. Křížová validace pomáhá ověřit, že výkonnost modelu je stabilní a není příliš závislá na konkrétním rozdělení dat.

Principal Component Regression (PCR):

Dalším model který jsem naimplementovala byl **PCR**, který kombinuje použití **PCA** (Principal Component Analysis) s lineární regresi. Použila jsem stejná standardizovaná data rozdělené na trénovací a testovací set obdobně jak u modelu s lineární regresí. Pro implementaci modelu je třeba rozhodnout kolik komponent bych měla použít, proto jsem po vytvoření modelu koukla na podrobnější informaci pomocí následujících příkazu:

```
# Principal Component Regression:

# Load necessary libraries
set.seed(123)
library(pls)

# PCR model without cross-validation
Model_pcr <- pcr(medv ~ ., data = TrainingSet, scale = TRUE, validation = "none")

# PCR model with cross-validation
Model_pcr.cv <- pcr(medv ~ ., data = TrainingSet, scale = TRUE, validation = "CV", segments = 10)

# Checking optimal number of ncomp
summary(Model_pcr.cv)
```

Zjistila jsem, že pro použití metody by bylo optimální použít buď **4** nebo **5** komponent, jelikož je vidět, že hodnoty CV po dosáhnutí 4 a 5 komponenty se už tak moc nemění.

```
VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps  11 comps  12 comps
CV          0.9981   0.7920   0.7556   0.6168   0.5871   0.5719   0.5686   0.5723   0.5689   0.5718    0.5733    0.5671    0.5534
adjCV       0.9981   0.7916   0.7555   0.6159   0.5846   0.5709   0.5673   0.5712   0.5677   0.5705    0.5718    0.5639    0.5517
       13 comps
CV       0.5441
adjCV    0.5425
```

Z obrázku dál je také vidět, že po paté komponentě je vysvětleno **81.04%** variability dat. Toto posloužilo důvodem, proč jsem zvolila **5 komponent** jako optimální množství.

```
TRAINING: % variance explained
      1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
X       48.02    58.82    67.97    74.78    81.04    86.16    90.21    93.21    95.31     96.92     98.26     99.54    100.00
medv    37.80    43.76    62.57    67.24    68.66    69.30    69.31    69.87    69.88     70.64     71.74     72.59     73.46
```

Po dokončení implementace modelu jsem zjistila následující výsledky:

```
> print(RMSE_pcr_training) [1]
0.557398

> print(RMSE_pcr_testing) [1]
0.5247246

> print(R2_pcr_training)
[1] 0.6865652

> print(R2_pcr_testing)
[1] 0.7510312
```

Z výsledků je patrné, že PCR model má ve srovnání s klasickou lineární regresí o něco horší výkonnost na tréninkových datech. Například, **RMSE** pro tréninkovou sadu je v případě PCR vyšší (**0.557** vs**. 0.513**) a hodnota **R²** je nižší (**0.687** vs. **0.735**), což naznačuje, že model PCR vysvětluje o něco méně variability v tréninkových datech.

Na druhou stranu, pokud se podíváme na výkonnost na testovací sadě, jsou výsledky mezi PCR a lineární regresí velmi podobné. **RMSE** je téměř identické (**0.525** pro PCR vs. **0.499** pro lineární regresi) a **R²** je také podobné (**0.751** pro PCR vs. **0.761** pro lineární regresi).

To naznačuje, že ačkoli PCR model má o něco horší schopnost přizpůsobit se tréninkovým datům, je však dobrý v predikci na nových, neviděných datech. Pro ověření tohoto modelu jsem taky použila **křížovou validaci**, kde byly zjištěny následující výsledky:

```
> print(RMSE_pcr_training_cv)
[1] 0.557398

> print(R2_pcr_training_cv)
[1] 0.6865652
```

Výsledky křížové validace jsou konzistentní s původními výsledky tréninkové sady bez křížové validace. To naznačuje, že model PCR má stabilní výkonnost, a to nejen na konkrétním rozdělení dat, ale i při opakovaném testování na různých podmnožinách dat.

V praxi to znamená, že PCR je vhodný nástroj zejména v případech, kdy jsou vstupní proměnné korelované a hrozí **multikolinearita**. I když model nemusí být tak přesný na tréninkových datech, jeho robustnost a schopnost vyhnout se přeučení mu dávají výhodu v predikcích na nových datech.

Závěr:

Lineární regrese je výkonnější na tréninkových datech a může být dobrou volbou, pokud v datech není výrazná multikolinearita.

PCR model je méně výkonný na tréninkových datech, ale poskytuje velmi podobné výsledky na testovacích datech jako lineární regrese, což naznačuje, že je vhodný pro situace s vysokou korelací mezi vysvětlujícími proměnnými a snižuje riziko přeučení.

V závislosti na povaze dat lze doporučit použití lineární regrese, pokud nejsou problémem korelované proměnné, zatímco **PCR** je výhodný v situacích, kdy je multikolinearita problémem, a zároveň poskytuje robustní predikce na nových datech.

Source

Harrison, D. and Rubinfeld, D.L. (1978) Hedonic prices and the demand for clean air. *J. Environ. Economics and Management* **5**, 81–102.

Belsley D.A., Kuh, E. and Welsch, R.E. (1980) *Regression Diagnostics. Identifying Influential Data and Sources of Collinearity.* New York: Wiley.