Efektivita marketingových akcí

Bc. Petr Boháč

2025-03-02

Abstrakt

Tato semestrální práce si klade za cíl úspěšně vypracovat dataminingový projekt na téma **efektivita marketingových akcí**. Projekt je zaměřen na analýzu historických dat z marketingových kampaní a jejich vyhodnocení. Cílem projektu je zjistit, jak různé typy produktů reagují na marketing, jaký vliv má výše investice do marketingu na úspěšnost kampaně a zda je možné predikovat úspěšnost kampaně na základě historických dat. Projekt bude realizován v programovacím jazyce R a výsledky budou prezentovány v podobě R Markdown dokumentu. Struktura práce se drží jednotnou DM metodologií CRSIP-DM používanou při řešení data miningových projektů.

I. Business Understanding

Základem každého úspěšného dataminingového projektu je pochopení jeho pointy, respektive proč se do projektu pouštím a co od něj očekávám.

Náš pomyslný zákazník je marketingová agentura, která se pravděpodobně zabývá marketingovými kampaněmi pro širokou škálu produktů. Po několika úspěšných kampaních se rozhodla, že by bylo dobré analyzovat data z těchto kampaní, což by mohlo přinést nové poznatky, které by potenciálně mohly celý proces zefektivnit. Pro zákazníka jsou nejzajímavější následující informace:

- jak různé typy produktů reagují na marketing
- jak velký vliv má výše investice do marketingu na úspěšnost kampaně
- možná predikce úspěšnosti kampaně na základě historických dat

Definice úspešnosti

V této fázi je také dobré si ujasnit, co je vlastně úspěšnost tohoto projektu z technické data miningové stránky. Hlavím cílem projeku je jednoznačně dozvědět se z historických data něco nového, o čem jsme doposud nevěděli, ale bez stanovení nějakého cíle se těžko určuje úspěšnost.

Jelikož agentura dělá marketing k různým typům produktů, bylo by dobré si udělat představu o tom, které kategorie reagují na marketing dobře a které ne, což by mohlo vést k vyřazení produktů, které jednoduše zákazníky tak nepřitahují. Tudíž nějaký způsob **seřazení produktů podle úspěšnosti kampaně** by byl ideální.

Další důležitou metrikou je výše investice do marketingu. Zde bychom se měli zaměřit na to, jaký vliv má výše investice na úspěšnost kampaně. Je jasné, že čím více peněz do marketingu investujeme, tím větší úspěch můžeme očekávat. **Ale jak moc?** A je to lineární závislost? Nebo je to spíše exponenciální? Nebo je to od určité částky kontraproduktivní? To jsou otázky, které bychom si měli položit a pokusit se na ně v rámci tohoto projektu najít odpovědi.

Zlatým grálem celého projektu je predikce úspěšnosti kampaně na základě historických dat. Zde bychom se měli zaměřit na to, jaké faktory ovlivňují úspěšnost kampaně a jak je možné tyto faktory využít k predikci úspěšnosti kampaně. Úspěšným výstupem by tedy mohl být nějaký **regresní model**, který by měl být schopen predikovat úspěšnost kampaně na základě historických dat. Tento model by mohl být použit k tomu,

abychom byli schopni předpovědět úspěšnost kampaně ještě před jejím spuštěním, což by mohlo vést k úspoře peněz a času.

Projektový plán

Kvalitně dopředu naplánovaný projekt je základním kamenem úspěšného projektu. V rámci technického plánu bychom měli mít jasno v tom, jaké kroky nás čekají a jakým způsobem budeme projekt realizovat. Co se týče výběru programovacího jazyka pro proces analýzy, nabízí se dva jasní kandidáti - Python a R. Oba jazyky jsou extensivně používané v oblasti datové analytiky a oba mají své výhody a nevýhody. Python je jazyk, který je velmi populární a má širokou škálu knihoven, které nám mohou pomoci při realizaci projektu. Na druhou stranu R je jazyk, který je více zaměřen na statistiku a analýzu dat. Jeho specializace nabízí uživatelům intuitivnější přístup k řešení úloh statistické analýzy, což by mohlo být pro náš projekt výhodou.

Programovací jazyk pro tento projekt byl hned ze začátku zvolen zadavatelem projektu, panem doktorem Lamrem, a to **R**. Jak již bylo zmíněno výše, R je specificky zaměřen na tento typ projektů, což by mělo usnadnit a urychlit celý proces.

Co se týče formy výstupu, nabízí se několik možností podle toho, komu budou výsledky projektu předkládány. V případě, že by se měl výsledek projektu prezentovat v místnosti plné technologicky nedotčených lidí (primárně management), bylo by dobré mít výstup v podobě prezentace, která by byla zaměřena na business a na to, co projekt přinesl. Ovšem vzhledem k tomu, že výsledky budou prezentovány před skupinou lidí, kteří se v oblasti datové analytiky pohybují, byl zvolen formát **R Markdown**, který je dobrým kompromisem mezi prezentací a technickým reportem. Tento formát umožňuje kombinovat text, kód a grafy do jednoho dokumentu, což je ideální pro prezentaci výsledků projektu.

II. Data Understanding

Data mining jako proces je postavený na předpokladu, že máme k dispozici dostatečné množství dat - obecně platí že čím více, tím lépe. S rostoucím množstvím dat je obtížnější si udělat představu o tom, co vlastně data obsahují a jaké informace nám mohou poskytnout. Proto je důležité si data nejprve důkladně prozkoumat a zjistit, s čím vlastně pracujeme.

Pro účely tohoto projektu nám byl poskytnut dataset, který obsahuje historická data z marketingových kampaní. Tento dataset nám obecně říká, na jaké typy produktů byly kampaně zaměřeny, kolik peněz bylo do jednotlivých produktů investováno a jak se kampaně promítly do zisků.

Co máme k dispozici za data?

Dataset nám byl poskytnut ve formátu **CSV** (z angl. "Comma-Separated Values"), což je standardní formát pro ukládání dat v tabulkové podobě. Tento formát je velmi populární a je podporován většinou programovacích jazyků, což usnadňuje práci s daty. Hodnoty jsou odděleny čárkami, desetinná místa jsou oddělena tečkami a celý soubor je očistěn od nadbytečných whitespace znaků - prakticky perfekně zformátované data.

Ke čtení dat z CSV souboru máme ve standardní knihovně R (dále jen *stdlib*) k dispozici několik funkcí, které nám umožňují načíst data do paměti a začít s nimi pracovat. Hlavní funkcí, která provádí vlastní parsování je funkce `read.table()`, která slouží pro čtení obecných *tabular* dat. Ostatní níže uvedené funkce jsou pouze šikovné wrapper funkce.

- `read.csv()`
  - data oddělena čárkami
  - desetinná místa oddělena tečkami
- `read.csv2()`
  - data oddělena středníky
  - desetinná místa oddělena čárkami
- `read.delim()`
  - data oddělena tabulátory (`\t`)

    - desetinná místa oddělena tečkami
- `read.delim2()`
    - data oddělena tabulátory
    - desetinná místa oddělena čárkami

```
# Načtení dat z CSV souboru do proměnné df
df <- read.csv("data/GOODS1n.csv")
```

V dalším kroku by bylo dobré si udělat high-level overview našich dat, která jsme načetli do proměnné `df`. Pro tento účel nám poslouží funkce `str()`, která je pro náš účel naprosto ideální. Zavoláním této jedné funkce získáme základní informace o struktuře dat, jako jsou názvy sloupců, datové typy a počet řádků.

```
# Základní informace o struktuře dat
str(df)
```

```
## 'data.frame':    200 obs. of  5 variables:
##  $ Class    : chr  "Confection" "Drink" "Luxury" "Confection" ...
##  $ Cost     : num  24 79.3 82 74.2 90.1 ...
##  $ Promotion: int  1467 1745 1426 1098 1968 1486 1248 1364 1585 1835 ...
##  $ Before   : int  114957 123378 135246 231389 235648 148885 123760 251072 287043 240805 ...
##  $ After    : int  122762 137097 141172 244456 261940 156232 128441 268134 310857 272863 ...
```

```
# Všechny možné kategorie produktů
unique(df$Class)
```

```
## [1] "Confection" "Drink"      "Luxury"     "Meat"
```

Jak z výstupu vidíme, načtený dataframe má rozměry 200 x 5, což znamená, že obsahuje **200 řádků**, ve kterých jsou data rozdělena do 5 sloupců. Sloupce, respektive proměnné, které máme k dispozici, jsou tedy následující:

- `Class`
    - Kategorie produktu (`Confection`, `Drink`, `Luxury`, `Meat`)
- `Cost`
    - Cena produktu
- `Promotion`
    - Výše investice do marketingu pro daný produkt
- `Before`
    - Zisk **před** marketingovou kampaní
- `After`
    - Zisk **po** marketingové kampani

Kvalitativní ověření dat

Na první pohled se zdá, že data jsou v pořádku a že neobsahují žádné chybějící hodnoty. Nicméně je dobré si udělat základní statistiku pro jednotlivé sloupce, abychom měli jistotu, že data jsou v pořádku a že neobsahují žádné extrémní hodnoty, které by mohly ovlivnit výsledky analýzy. Pro tento účel nám poslouží funkce `summary()` (informační hodnotou velice podobná `Data Audit` uzlu v SPSS Modeleru), která nám poskytne základní deskriptivní statistické informace o jednotlivých sloupcích, jako jsou průměr, medián, minimum a maximum.

```
# Základní statistika pro jednotlivé sloupce
summary(df)
```

```
##     Class                Cost          Promotion        Before
##  Length:200         Min.   :  5.08   Min.   :1004   Min.   :100751
##  Class :character   1st Qu.: 30.95   1st Qu.:1208   1st Qu.:149175
##  Mode  :character   Median : 53.68   Median :1470   Median :203421
##                     Mean   : 54.91   Mean   :1485   Mean   :201183
```

```
##                    3rd Qu.: 79.36   3rd Qu.:1745   3rd Qu.:251121
##                    Max.   :104.98   Max.   :1986   Max.   :299340
##      After
##  Min.   :104393
##  1st Qu.:159918
##  Median :215303
##  Mean   :214671
##  3rd Qu.:270884
##  Max.   :346375
```

I když náš dataset doposud vypadá v pořádku, ničemu neublíží, když si uděláme ještě pár dalších kontrol. Zmínili jsme například možnou existenci chybějících hodnot, které by mohly ovlivnit výsledky analýzy. Pro tento účel nám poslouží funkce `is.na()`, která nám vrátí TRUE pro každou chybějící hodnotu a FALSE pro každou hodnotu, která není chybějící. Funkci `colSums()` pak použijeme k tomu, abychom zjistili, kolik chybějících hodnot máme v jednotlivých sloupcích.

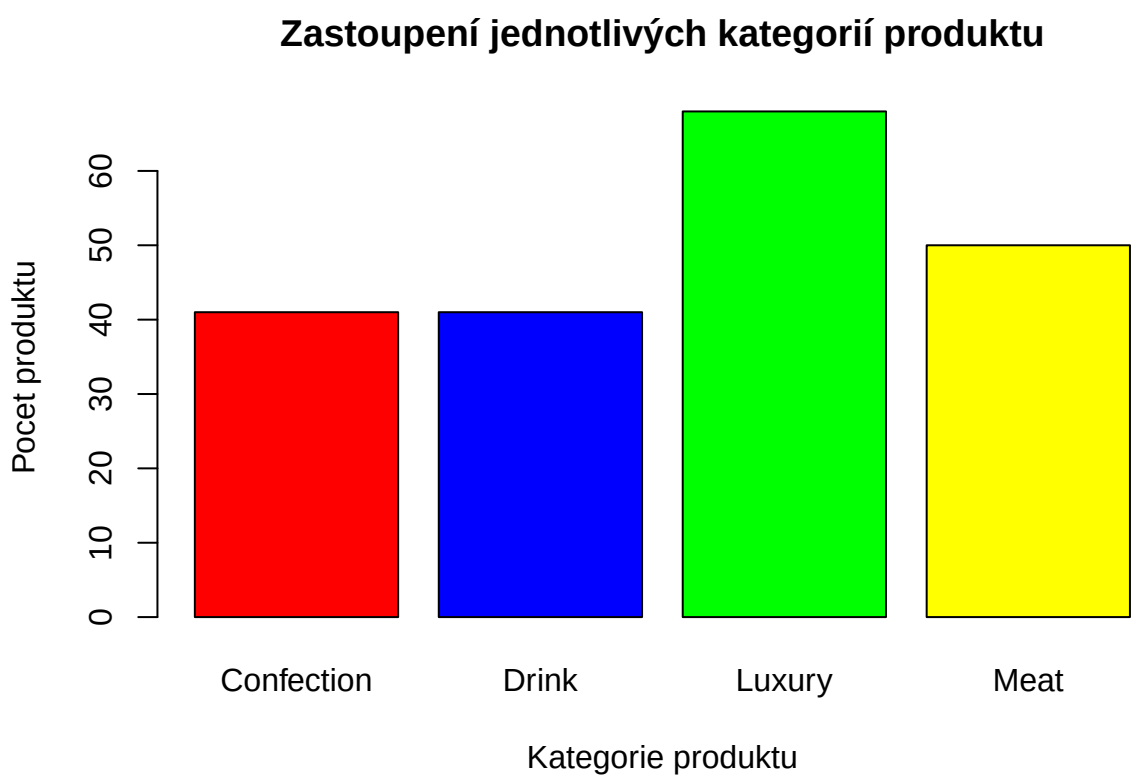
```r
# Kontrola chybějících hodnot
colSums(is.na(df))
```

```
##     Class      Cost Promotion    Before     After
##         0         0         0         0         0
```

Z výsupu je očividné, že data neobsahují žádné chybějící hodnoty, což nám o to více usnadní třetí fázi projektu (Data Preparation).

Proměnné, které nabývají pouze několika hodnot, jsou kvalitativní (kategorické) a proměnné, které nabývají libovolných hodnot, jsou kvantitativní (numerické). V našem případě máme k dispozici jednu kvalitativní proměnnou `Class`, která obsahuje 4 různé kategorie produktů. Kategorické proměnné můžou trpět tzv. *nevyvážeností*, což znamená, že některé kategorie mohou být zastoupeny více než jiné, což by mohlo ovlivnit výsledky analýzy.

```r
barplot(
  table(df$Class),
  main = "Zastoupení jednotlivých kategorií produktů",
  xlab = "Kategorie produktu",
  ylab = "Počet produktů",
  col = c("red", "blue", "green", "yellow"),
  names.arg = c("Confection", "Drink", "Luxury", "Meat")
)
```



Ze sloupcového grafu je vidět, že kategorie `Luxury` je zastoupena nejvíce s celkovým počtem produktů 70. Tata distribuce by mohla být problémová v případě trénování klasifikačního modelu a pravděpodobně by v další fázi projektu vyžadovala umělé vyvážení. Jelikož ale náš cílový model bude regresního charakteru, tak by to neměl být problém.

Co ale je pro regresní modely relevantní, je rozložení numerických proměnných. Pro tento účel nám poslouží funkce `hist()`, která nám zobrazí histogram pro jednotlivé sloupce. Histogram je grafické znázornění rozložení dat, které nám umožňuje vidět, jak jsou data rozložena a zda obsahují nějaké extrémní hodnoty.

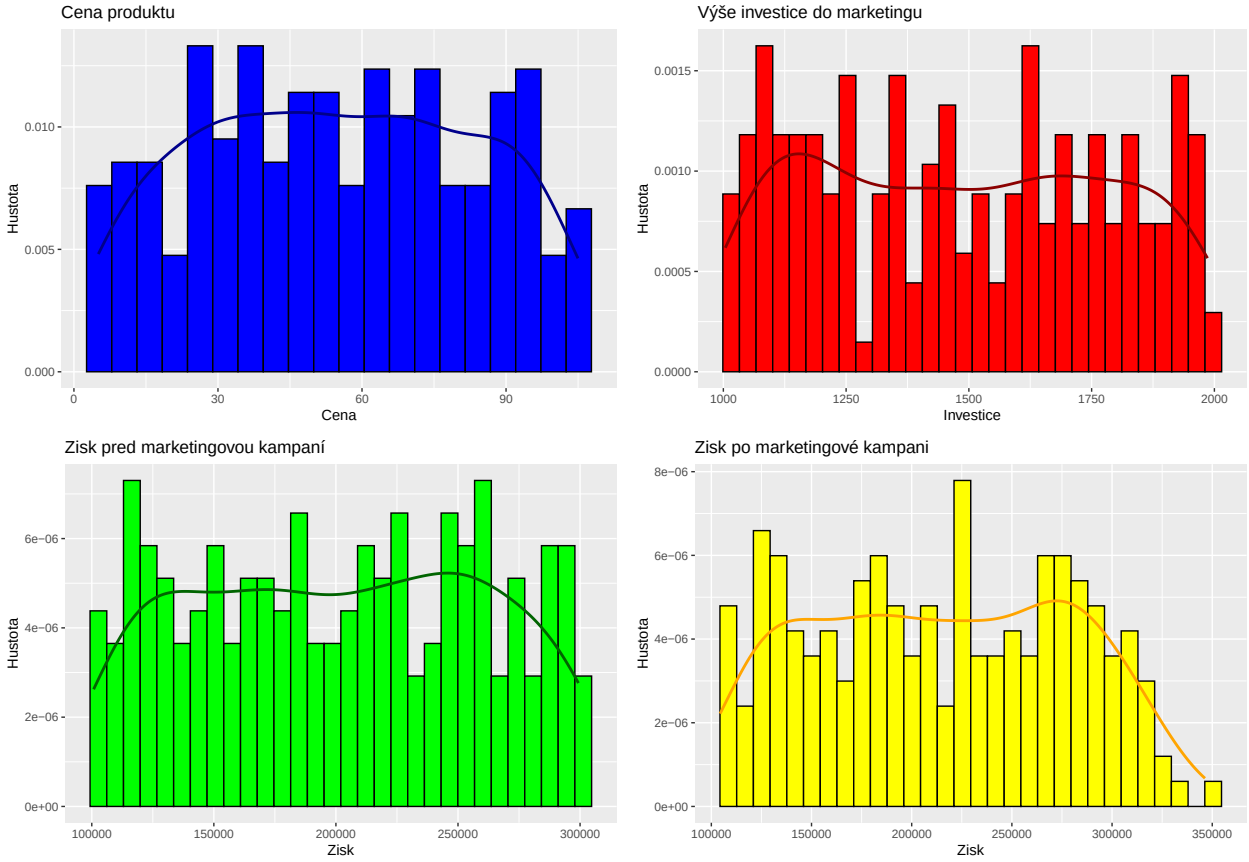
```
# Cena produktu
ggplot(df, aes(x = Cost)) +
  geom_histogram(
    aes(y = after_stat(density)),
    fill = "blue",
    color = "black",
    bins = 20
  ) +
  geom_density(color = "darkblue", linewidth = 1) +
  labs(title = "Cena produktu", x = "Cena", y = "Hustota")

# Výše investice do marketingu
ggplot(df, aes(x = Promotion)) +
  geom_histogram(
    aes(y = after_stat(density)),
    fill = "red",
    color = "black",
    bins = 30
```

```
  ) +
  geom_density(color = "darkred", linewidth = 1) +
  labs(title = "Výše investice do marketingu", x = "Investice", y = "Hustota")

# Zisk před marketingovou kampaní
ggplot(df, aes(x = Before)) +
  geom_histogram(
    aes(y = after_stat(density)),
    fill = "green",
    color = "black",
    bins = 30
  ) +
  geom_density(color = "darkgreen", linewidth = 1) +
  labs(title = "Zisk před marketingovou kampaní", x = "Zisk", y = "Hustota")

# Zisk po marketingové kampani
ggplot(df, aes(x = After)) +
  geom_histogram(
    aes(y = after_stat(density)),
    fill = "yellow",
    color = "black",
    bins = 30
  ) +
  geom_density(color = "orange", linewidth = 1) +
  labs(title = "Zisk po marketingové kampani", x = "Zisk", y = "Hustota")
```



Z výstupu je vidět, že rozložení jednotlivých proměnných je v pořádku a **neobsahuje žádné extrémní hodnoty** (tzv. *outliers*).

Hlubší průzkum dat

Ke konci této fáze, která je zaměřena na porozumění datům, by bylo také dobré prozkoumat, jesli data mezi sebou nemají nějaké zřejmé vztahy. Je dost možné, že žádné převratné vztahny mezi daty neodhalíme, ale jak již bylo řečeno, v této fázi máme porozumět datům a **pokusit** se odhalit potenciální vzory, které by mohly být užitečné pro další fáze projektu.

Jednou věcí, kterou můžeme zkusit, je vytvořit nad daty tzv. *korelační matici*. Korelační matice je tabulka, která nám ukazuje, jak jsou jednotlivé proměnné mezi sebou korelovány. Korelace je míra toho, jak jsou dvě proměnné mezi sebou spojeny a může nabývat hodnot od -1 do 1. Pokud je korelace blízká 1, znamená to, že obě proměnné jsou silně pozitivně korelovány. Pokud je korelace blízká -1, znamená to, že obě proměnné jsou silně negativně korelovány. Pokud je korelace blízká 0, znamená to, že obě proměnné nejsou mezi sebou korelovány. Pro výpočet korelační matice použijeme funkci `cor()`, která nám vrátí korelační matici pro všechny numerické sloupce v datovém rámci. Jelikož máme k dispozici pouze 4 numerické sloupce, tak matice bude mít rozměry 4 x 4.

```
# Korelační matice
cor(df[, c("Cost", "Promotion", "Before", "After")])
```

```
##                  Cost    Promotion      Before        After
## Cost       1.00000000 -0.032464896 -0.02322929 -0.020879598
## Promotion -0.03246490  1.000000000 -0.06568177 -0.008242122
## Before    -0.02322929 -0.065681769  1.00000000  0.993937445
## After     -0.02087960 -0.008242122  0.99393744  1.000000000
```

Většina hodnot v korelační matici je blízká 0, což znamená, že mezi jednotlivými proměnnými není žádná silná korelace. Nicméně je zde jedna zajímavá věc - a to je silná pozitivní korelace mezi proměnnými `Before` a `After`, což je logické, jelikož zisk po marketingové kampani by měl být vyšší než zisk před marketingovou kampaní a naopak.

Korelační matice nám toho tedy moc neřekla, ale stálo to za pokus. Mohli bychom zkusit ještě nějaké pokročilejší metody, ale náš originální dataset stejně neobsahuje moc zajímavých dat, které bychom mohli prozkoumat. V další fázi projektu se zaměříme na přípravu dat pro další fáze projektu, což by nám mohlo otevřít nové možnosti pro hlubší analýzu.

III. Data Preparation

Třetí fáze CRISP-DM projektů obecně zabírá nejvíce času z celého projektu. Obecně se říká, že **80% času strávíme přípravou dat a 20% času analýzou dat**. Tento poměr je samozřejmě orientační a může se lišit projekt od projektu, ale je dobré mít na paměti, že příprava dat je velmi důležitá a že by se jí mělo věnovat dostatek času.

V reálných projektech se většinou setkáváme s daty, která nejsou v ideálním stavu a potřebují nejdříve trochu lásky. Může jít o různé problémy, jako jsou chybějící hodnoty, duplicitní hodnoty, extrémní hodnoty, špatné formátování dat a podobně. V našem případě jsme ale měli štěstí a dataset byl prakticky v perfektním stavu, což tuto fázi značně urychlí.

Derivace nových proměnných

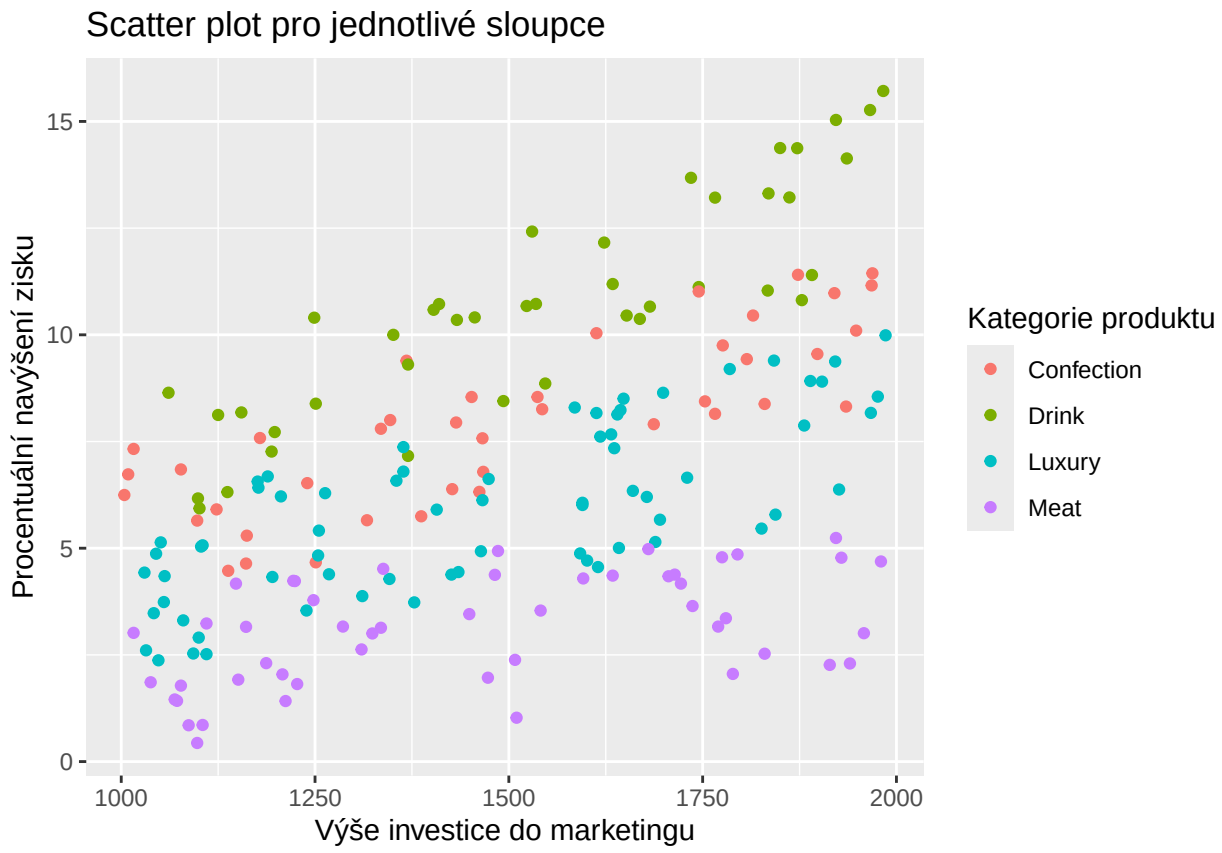
I tak bude ale potřeba si s daty trochu pohrát, než nad nimi budeme moci začít trénovat modely. Prvním krokem bude přidat do datového rámce nový sloupec, který bude zachycovat navýšení zisku po marketingové kampani. Nazveme tento sloupec `RevenueIncrease` a jeho hodnota bude vypočtena jako rozdíl mezi ziskem po marketingové kampani a ziskem před marketingovou kampaní v procentech, respektive následující vzorec:

$$\text{SalesIncrease} = \frac{\text{After} - \text{Before}}{\text{Before}} \times 100\%$$

```r
# Přidání nového sloupce do datového rámce
df$RevenueIncrease <- (df$After - df$Before) / df$Before * 100
```

Vybavme si poslední část předchozí fáze, kde jsme se snažili odhalit nějaké vzory v datech. V tu dobu jsme měli k dispozici pouze 4 numerické sloupce, které jsme prozkoumali. Nyní máme k dispozici 5 sloupců, což nám dává více možností pro analýzu dat. Například bychom mohli zkusit zjistit, jaký vliv má výše investice do marketingu na procentuální navýšení zisku. Pro tento účel použijeme funkci `plot()`, která nám zobrazí scatter plot pro sloupce `Promotion` a `RevenueIncrease`. Scatter plot je grafické znázornění dat, které nám umožňuje vidět, jak jsou data rozložena a zda obsahují nějaké extrémní hodnoty.

```r
# Scatter plot pro jednotlivé sloupce
ggplot(df) +
  geom_point(aes(
    x = Promotion,
    y = RevenueIncrease,
    color = Class
  )) +
  labs(
    title = "Scatter plot pro jednotlivé sloupce",
    x = "Výše investice do marketingu",
    y = "Procentuální navýšení zisku",
    color = "Kategorie produktu"
  )
```



Z předešlého grafu můžeme vyčíst hned několik věcí ohledně toho, jak různé kategorie produktů reagují na různé výše investice do marketingu. Je například patrné, že bez ohledu na to, kolik jsme investovali do marketingu pro produkty typu `Meat`, zisky se drží pod hranicí 5%. Ostatní kategorie se obecně drží jednoho trendu - čím více investujeme do marketingu, tím větší navýšení zisku můžeme očekávat. Nicméně nejzajímavější informací, kterou z grafu můžeme vidět, je to, že kategorie `Drink` reaguje na marketing **lépe, než všechny ostatní kategorie**.

Příznakování dat

Kategorie produktů jsou pro naše modely zásadní informací. Problémem ale je, že modely neumí pracovat s textovými hodnotami, které jsou v našem datasetu. Proto je potřeba převést tyto hodnoty na numerické hodnoty, které modely budou schopny zpracovat. Tento proces se nazývá *příznakování* (z angl. "feature engineering") a je velmi důležitý pro další fáze projektu.

Konkrétní proces příznakování se bude skládat z následujících kroků:

1. Derivace nových binárních proměnných pro každou z kategorií produktů
   - Tímto způsobem vytvoříme 4 nové proměnné, které budou mít hodnotu 1, pokud je produkt v dané kategorii a 0, pokud není
2. Vynásobení každé z nových proměnných hodnotou `Promotion` pro daný produkt
   - tohle fr nevim proc delame
3. Odstranění nyní redundantních sloupců `Class` a `Promotion`
   - sloupec `Class` je redundantní, jelikož jsme ho převedli na 4 nové proměnné
   - sloupec `Promotion` je redundantní, jelikož jsme ho vynásobili každou z nových proměnných

```r
# Derivace nových proměnných pro každou z kategorií produktů
df$Class_Confection <- ifelse(df$Class == "Confection", 1, 0)
df$Class_Drink <- ifelse(df$Class == "Drink", 1, 0)
df$Class_Luxury <- ifelse(df$Class == "Luxury", 1, 0)
df$Class_Meat <- ifelse(df$Class == "Meat", 1, 0)

# Vynásobení každé z nových proměnných hodnotou Promotion pro daný produkt
df$Class_Confection <- df$Class_Confection * df$Promotion
df$Class_Drink <- df$Class_Drink * df$Promotion
df$Class_Luxury <- df$Class_Luxury * df$Promotion
df$Class_Meat <- df$Class_Meat * df$Promotion

# Odstranění redundantních sloupců
df$Class <- NULL
df$Promotion <- NULL

# Výsledek příznakování
head(df)
```

```
##    Cost Before  After RevenueIncrease Class_Confection Class_Drink Class_Luxury
## 1 23.99 114957 122762        6.789495              1467           0            0
## 2 79.29 123378 137097       11.119486                 0        1745            0
## 3 81.99 135246 141172        4.381645                 0           0         1426
## 4 74.18 231389 244456        5.647200              1098           0            0
## 5 90.09 235648 261940       11.157319              1968           0            0
## 6 69.85 148885 156232        4.934681                 0           0            0
##   Class_Meat
## 1          0
## 2          0
## 3          0
## 4          0
```

```
## 5          0
## 6       1486
```

Rozdělení dat na různé sady

V posledním kroku této fáze je potřeba rozdělit data na různé sady, které budeme používat pro trénování a testování modelů. Dělení datasetů tímto způsobem je velmi důležité, jelikož nám umožňuje testovat modely na datech, která nebyla použita pro trénování. Tímto způsobem můžeme zjistit, jak dobře modely fungují na nových datech a zda jsou schopny generalizovat na nová data. Pokud bychom modely testovali na stejných datech, na kterých byly trénovány, mohli bychom získat zkreslené výsledky, které by nám neřekly nic o tom, jak dobře modely fungují na nových datech. Obecně se doporučuje rozdělit data na 3 sady:

- **Trénovací sada** (z angl. "training set") - tato sada se používá pro trénování modelů
- **Testovací sada** (z angl. "test set") - tato sada se používá pro testování modelů
- **Validační sada** (z angl. "validation set") - tato sada se používá pro validaci modelů

V našem případě, kdy celý náš dataset obsahuje žalostných 200 řádků, se omezíme pouze na první dva typy sad.

Obecně se doporučuje rozdělit data na 70% pro trénovací sadu a 30% pro testovací sadu. V našem případě ale radši z pochopitelných důvodů použijeme poměr 9:1, což bude znamenat, že trénovací sada bude obsahovat 180 řádků a testovací sada bude obsahovat 20 řádků. Pro rozdělení dat použijeme funkci `sample()`, která nám vrátí náhodný vzorek z datového rámce.

```r
# Rozdělení dat na trénovací a testovací sadu
set.seed(123) # pro reprodukovatelnost

train_index <- sample(seq_len(nrow(df)), size = 0.9 * nrow(df))
train_data <- df[train_index, ]
test_data <- df[-train_index, ]

# Kontrola rozměrů datových rámců
dim(train_data)
```

```
## [1] 180   8
```

```r
dim(test_data)
```

```
## [1] 20  8
```

IV. Modeling

V. Evaluation

VI. Deployment