Applying the CRISP-DM Data Science Methodology to Sales Volume Forecasting and Budgeting Problems

Introduction

Matthias Hofmaier (11944050)

May 2023

Introduction

Reliable forecasts of a company's sales volume can be of massive advantage in budgeting and strategic planning. Traditional methods to forecast sales often rely on univariate moving average models. In the last decade, machine learning methods, which can incorporate information over various dimensions gained a lot of popularity. Those models allow the inclusion of variables from annual financial statements, including balance sheets and profit and loss statements, that potentially increase the prediction performance. But dealing with this data can be challenging. Especially for people who are from different domains and are not used to data science workflows. The Cross Industry Standard Process for Data Mining (CRISP-DM) can help to perform data science projects of this kind in a well-defined way. Therefore the goal of this project is to create a guideline project for students in economics that showcases how the CRISP-DM methodology can be applied to sales volume forecasting and budgeting problems. The project will be carried out on the example of comparing a univariate model and a multivariate machine learning model within the context of U.S. stock corporations in the S&P 500 from 2002 to 2022.

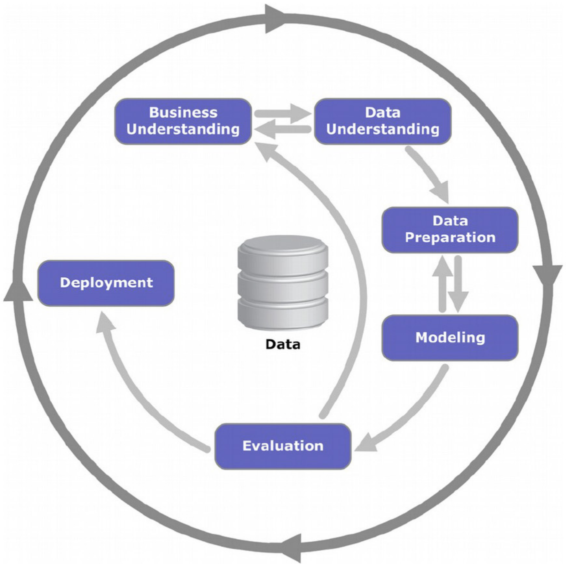

Figure 1: CRISP-DM Model

The CRISP-DM model consists of six stages that can be followed to successfully solve a data science problem.

In the first stage, Business Understanding, the needs of a business are identified and project goals and KPIs are defined. As this project is done within a university course, the goals and KPIs were already defined within a proposal and therefore this stage will be skipped. The dataset that will be used within this project stems from the Thomson Reuters Datastream database and includes the quarterly sales and variables from annual financial statements for S&P 500 stock corporations from 2002 to 2022. The financial statements consist of balance sheets with attributes describing the assets, liabilities, and equity as well as profit and loss statements which contain attributes describing the sales and expenses of a particular company. Datastream comes with an Excel interface but is not made to retrieve data in a clean way for multiple companies at once. Thus, an initial data transformation has to be performed to convert the data spreading over several Excel sheets to a few tables before being able to execute the Data Understanding stage of this project. This will be done within this notebook.

Imports

```
library(readxl)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(stringr)
```

Constants

```
RAW_DATA_PATH <- "../data/raw/datenabzug_sp500.xlsx"
OUTPUT_BASE_PATH <- "../data/processed"
SALES_OUTPUT_PATH <- paste(OUTPUT_BASE_PATH, "sales.csv", sep = "/")
BALANCE_SHEET_OUTPUT_PATH <- paste(OUTPUT_BASE_PATH, "balance_sheet.csv", sep = "/")
PROFIT_LOSS_OUTPUT_PATH <- paste(OUTPUT_BASE_PATH, "profit_loss.csv", sep = "/")
```

Sales

Load and transform quartely sales data

```
# apply function to all 4 quarters
df_sales <- map_dfr(1:4, load_transform_sales_quarter)
cat(
  paste(
    "Transformed interim sales data frame contains ",
    nrow(df_sales),
    " records for ",
    n_distinct(df_sales$company),
    " companies\nfrom year ",
    min(df_sales$year),
    " to ",
    max(df_sales$year),
    ".",
    sep = ""
  )
)
```

```
## Transformed interim sales data frame contains 34841 records for 500 companies
## from year 2002 to 2022.
```

```
head(df_sales)
```

```
## # A tibble: 6 x 4
##   company   interim_sales  year quarter
##   <chr>             <int> <int>   <int>
## 1 APPLE INC       1475000  2003       1
## 2 APPLE INC       1909000  2004       1
## 3 APPLE INC       3243000  2005       1
## 4 APPLE INC       4359000  2006       1
## 5 APPLE INC       5264000  2007       1
## 6 APPLE INC       7512000  2008       1
```

Save to CSV

```
write.csv(df_sales, SALES_OUTPUT_PATH)
```

Balance sheet

Function to add spaces to some company names

```
# add space to names of companies without space before the "-"
# without this, we will later have difficulties in correctly parsing the variable names
company_names_without_space <- c(
  "THERMO FISHER SCIENTIFIC",
  "ADOBE (NAS)",
  "CONSTELLATION BRANDS 'A'",
  "WALGREENS BOOTS ALLIANCE",
  "LYONDELLBASELL INDS.CL.A",
  "CITIZENS FINANCIAL GROUP",
  "MID-AMER.APT COMMUNITIES",
  "TERADYNE (XSC)",
  "UNITED AIRLINES HOLDINGS",
  "ALLIANT ENERGY (XSC)",
  "CBOE GLOBAL MARKETS(BTS)",
  "BIO-RAD LABORATORIES 'A'",
  "UNIVERSAL HEALTH SVS.'B'",
  "NEWELL BRANDS (XSC)"
)

company_names_without_space <-
  company_names_without_space[!duplicated(company_names_without_space)]

add_space_to_company_names <-
  function(name, c_names = company_names_without_space) {
    for (c_name in c_names) {
      if (str_detect(name, fixed(c_name))) {
        name = str_replace_all(name, fixed(c_name), paste0(c_name,  " "))
      }
    }
    return(name)
  }
```

Load and transform balance sheet data

```
df_balance_sheet <- load_transform_balance_sheet()
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion to integer
## range
```

```
cat(
  paste(
    "Transformed balance sheet data frame contains",
    nrow(df_balance_sheet),
    " records",
    "with",
    ncol(df_balance_sheet),
    "variables\nfor",
    n_distinct(df_balance_sheet$company),
    " companies from year",
    min(df_balance_sheet$year),
    " to",
    max(df_balance_sheet$year),
    "."
  )
)
```

```
## Transformed balance sheet data frame contains 10563  records with 30 variables
## for 503  companies from year 2002  to 2022 .
```

```
head(df_balance_sheet)
```

```
## # A tibble: 6 x 30
##   company  year BORROW~1 EQUIT~2 NET C~3 NET D~4 ORDIN~5 PREFE~6 TOTAL~7 ASSET~8
##   <chr>   <int>    <int>   <int>   <int>   <int>   <int>   <int>   <int>   <int>
## 1 APPLE    2002        0  4.10e6  3.73e6 -4.02e6 1826000       0 2269000  6.23e6
## 2 APPLE    2003   304000  4.22e6  3.53e6 -4.26e6 1926000       0 2297000  6.76e6
## 3 APPLE    2004        0  5.08e6  4.38e6 -5.46e6 2514000       0 2562000  7.96e6
## 4 APPLE    2005        0  7.47e6  6.82e6 -8.26e6 3521000       0 3945000  1.14e7
## 5 APPLE    2006        0  9.98e6  8.04e6 -1.01e7 4355000       0 5629000  1.72e7
## 6 APPLE    2007        0  1.45e7  1.27e7 -1.54e7 5368000       0 9164000  2.53e7
## # ... with 20 more variables: `TOTAL ASSETS EMPLOYED` <int>,
## #   `TOTAL CAPITAL EMPLOYED` <int>, `TOTAL CASH & EQUIVALENT` <int>,
## #   `TOTAL CURRENT ASSETS` <int>, `TOTAL CURRENT LIABLITIES` <int>,
## #   `TOTAL DEBT` <int>, `TOTAL DEBTORS & EQUIVALENT` <int>,
## #   `TOTAL DEFERRED & FUTURE TAX` <int>, NET <int>, `TOTAL INTANGIBLES` <int>,
## #   `TOTAL INVESTMNTS (EX.ASSOC)` <int>, `TOTAL LOAN CAPITAL` <int>,
## #   `TOT. SHARE CAPITAL & RESERVES` <int>, `TOTAL STOCK AND W.I.P.` <int>, ...
```

Save to CSV

```
write.csv(df_balance_sheet, BALANCE_SHEET_OUTPUT_PATH)
```

Profit & Loss

Load and transform profit & loss data

```
df_profit_loss <- load_transform_profit_loss()
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
cat(
  paste(
    "Transformed profit loss data frame contains",
    nrow(df_profit_loss),
    " records",
    "with",
    ncol(df_profit_loss),
    "variables\nfor",
    n_distinct(df_profit_loss$company),
    " companies from year",
    min(df_profit_loss$year),
    " to",
    max(df_profit_loss$year),
    "."
  )
)
```

```
## Transformed profit loss data frame contains 10563  records with 42 variables
## for 503  companies from year 2002  to 2022 .
```

```
head(df_profit_loss)
```

```
## # A tibble: 6 x 42
##   company  year - AFTE~1 - A.W~2 - CAS~3 - COS~4 - DEP~5 - DIV~6 - EBI~7 - EAR~8
##   <chr>   <int>    <int>   <int>   <int>   <int>   <int>   <int>   <int>   <int>
## 1 APPLE    2002    65000      NA       0  4.02e6  118000       0  216000   65000
## 2 APPLE    2003    68000      NA       0  4.39e6  113000       0  213000   68000
## 3 APPLE    2004   276000      NA       0  5.87e6  150000       0  536000  276000
## 4 APPLE    2005  1335000   38000       0  9.71e6  179000       0 1994000 1335000
## 5 APPLE    2006  1989000   45000       0  1.35e7  225000       0 3043000 1989000
## 6 APPLE    2007  3496000   68000       0  1.55e7  317000       0 5325000 3496000
## # ... with 32 more variables: `- EARNED FOR ORDINARY-ADJ` <int>,
## #   `- EBIT` <int>, `- EXCEPTIONAL ITEMS` <int>,
## #   `- EXTRAORD. ITEMS AFTER TAX` <int>, `- GROSS PROFIT ON SALES` <int>,
## #   `- INTEREST CAPITALSED` <int>, `- INTEREST INCOME` <int>,
## #   `- INTEREST PAID` <int>, `- MINORITY INTERESTS` <int>,
## #   `- NET INTEREST CHARGES` <int>, `- OPERATING PROFIT` <int>,
## #   `- OPERATING PROFIT-ADJ` <int>, `- ORDINARY DIVIDENDS (GROSS)` <int>, ...
```

Save to CSV

```
write.csv(df_profit_loss, PROFIT_LOSS_OUTPUT_PATH)
```