Canseza Avağ Erdurak

Table of contents

Introduction

This progress journal covers Canseza Avağ Erdurak's work during their term at BDA 503 Fall 2022.

Each section is an assignment or an individual work.

1 Assignment 1

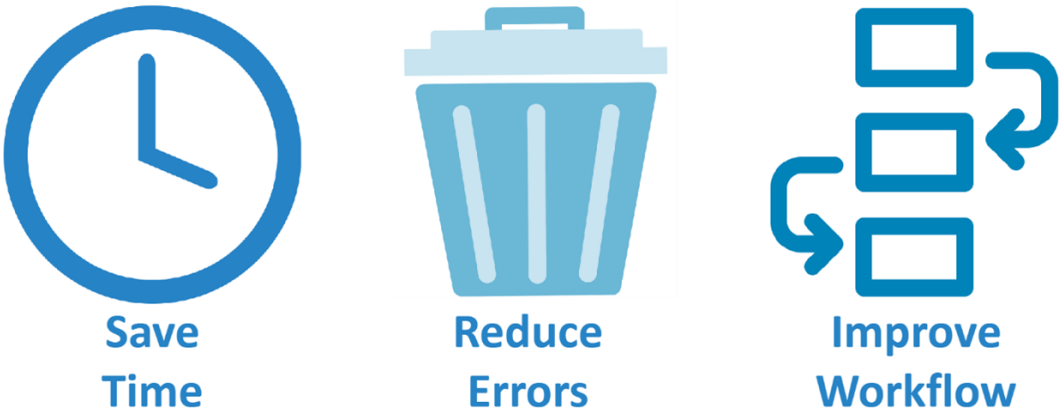
Canseza Avağ Erdurak
2022-10-11

I am Canseza Avağ Erdurak. I graduated from Management Information Systems, Boğaziçi University in 2011. Since then, I always dreamed about having a master's degree. Being a MEFIAN made my dream come through. I am super excited to be a part of this programme. I was less interested in coding staff when I was an undergradute. I'm ended up like coding is not that boring after I started to work as a BI Developer. Being a BI Developer restricted me in a way that I could not involve in building a structure of the reports I am working on. I decided to choose a path that I could get my hands dirty in both BI and DWH jobs. I work as a BI Engineer in Pegasus Airlines since July 2018. It started to feel like I should step in the field of AI to make my job better and to be more satisfied at work. As a BI Engineer, I only provide data and support for architectural structure in AI Projects. However, I want to initiate and lead AI projects on my own. I believe that I am passionate, intellectually capable, and prepared to set out on this exhilarating and challenging path. I look forward to learn exploratory data analysis, machine learning and automation in R.

Here is the link of my linkedin profile : https://www.linkedin.com/in/cansezaavag/

2 RStudio Global 2022 Conference Talks : Save an Ocean of Time

The speaker, Danielle Dempsey works for CMAR that has over 250 oceanographic sensors deployed around the coast of Nova Scotia, Canada. Together, these generate around 4 million rows of data every year. Her challenge was to compile all of data stored in Excel files in different formats into a nice, tidy format that they could post online for any interested stakeholder to download and use in their own analysis.



She mentions about how skeptical her manager was at first, since there's an existing process that's already "good enough". However, she was decided not to proceed old-fashioned methods like copy+paste. She was not confident in data wrangling and package development skills. She spent time in coding to save time. She used Tidyverse packages mostly in her project. It took 9 months rather than 2 years to complete the task. If she didn't change the way how the work is done, she couln't get the results below. Skills that you learn working on one coding project are often transferable to other projects. She has used packages and skills she developed for further projects. She says how easier it is to train new people getting them on board. Errors can

pop up anywhere in data since humans are involved. Writing code makes to find and reduce errors in your dataset. It makes get clients more confident in data and make analysis with that data more reliable. Automate your data wrangling make your workflow more traceable and reproducible especially compared to copying and pasting. At the end, she mentions about how her manager changed her mind over streamlining a workflow. Her manager made him a hand-made present shaped as Super "R" so that she thought "R can do anything that we put our minds to." It think this metaphor she puts made me chose Danielle's talk for the assignment : "Making a salad left in the fridge. Most of the time, it tastes ok but eventually you will get something that doesn't taste quite right. Because you have no record of what went into that salad it's really hard to tell what exactly is making it taste funny, and it's also hard to avoid making the same mistake in the future. You can't take the dressing back out of the salad. In contrast, writing code is like writing and following a recipe. You can follow it step by step to get a delicious salad every time. As a bonus, you can give that recipe to a friend so they can make their own salad or help you make dinner." This conference talk made me impressed and motivated about my philosophy at work. At first, I go over process of how clients complete an existing job. Then, I start writing SQL codes to automate workflow. When clients are totally on board with me, they bring on new projects more reluctantly to work for. It has also changed the way how they make use of DWH/BI skills on processes and help them make more informative decisions about work.

Video link : https://www.rstudio.com/conference/2022/talks/save-ocean-of-time-streamline/

Talk materials' link : https://github.com/dempsey-CMAR/2022_rstudio_conf

3 3 R Posts Relevant to My Interests

3.1 Tidyverse

When I started to watch conference talk that I mentioned above, I realized I could do anything related to data wrangling I did so far in Pegasus by using R. So, I decided to elaborate on this article. Tidyverse has 8 core packages named **ggplot2, dplyr, tidyr, readr, purrr, tibble, stringr** and **forcats**.

1. **Data Visualization and Exploration**

- **ggplot2** is used to create data visualizations like bar charts, pie charts, histograms, scatterplots, error charts, etc.

2. **Data Wrangling and Transformation**

- **Dplyr** is known for data manipulation. It has five important functions namely mutate(), select(), filter(), summarise() and arrange(). These functions are used with group_by().
- **Tidyr** helps create clean data.
- **Stringr** has many functions for data cleaning and data preparation. All functions in this library starts with "str" and take a string vector as a first argument.
- **Forcats** handles issues like changes the orders of values in vectors, reordering the vectors, etc.

3. **Data Import and Management**

- **Readr** helps read rectangular data such as that with file formats tsv, csv, delim, fwf, etc. in a simple and speedy way.
- **Tibble** is a form of a data.frame which includes the useful parts of it and discards the parts that are not so important.

4. **Functional Programming**

- **Purrr** turns messed-up codes into simpler ones.

Article : https://www.geeksforgeeks.org/what-are-the-tidyverse-packages-in-r-language/

3.2 Uncover the R Applications

I watched many conference talks so that I could chose one for this assignment. These talks make me wonder why R Programming Language is used by top companies from various industries like banking, e-commerce, finance, etc.

Applications of R Programming :

- **Finance :** R helps financial institutions perform downside risk measurement, adjust risk performance and utilize visualizations like candlestick charts, density plots, drawdown plots, etc. Time-series statistical processes of R are used to model the movement of financial industries' stock-market and predict the prices of shares. R provides financial data mining capabilities through its packages like quantmod, pdfetch, TFX, pwt, etc. Rshiny helps extract data from online assets.
- **Banking :** R is most widely used for credit risk modeling and other forms of risk analytics. *Hadoop* is an ally of R in the fields like analysis of customer quality, customer segmentation, and retention.
- **Healthcare :** R helps perform pre-clinical trials and analyze the drug-safety data. R is also used for statistical modeling in the field of epidemiology, where data scientists analyze and predict the spread of diseases.
- **Social Media :** Some of the important statistical tools like sentiment analysis and other forms of social media data mining are used with R. Social media is used for potential customer segmentation and targeting them as new customers.
- **E-commerce :** E-commerce companies use R is for analyzing cross-selling products to their customers. Various statistical procedures like linear modeling are necessary to analyze the purchases made by the customers as well as in predicting product sales. Furthermore, companies use R for carrying out A/B testing analysis across the pages of their products.
- **Manufacturing :** Analyzing customer sentiment helps them optimize their product according to trending consumer interests and also to match their production volume to varying market demand. They also use R to minimize their production costs and maximize profits.

Article : https://data-flair.training/blogs/r-applications/

3.3 K-Fold Cross Validation in R (Step-by-Step)

In AI projects, a few models are run to figure out which one is the best for prediction.

K-fold cross validation is widely used method for model verification.

Following dataset is created.

```
#create data frame
df <- data.frame(y=c(6, 8, 12, 14, 14, 15, 17, 22, 24, 23),
                 x1=c(2, 5, 4, 3, 4, 6, 7, 5, 8, 9),
                 x2=c(14, 12, 12, 13, 7, 8, 7, 4, 6, 5))

#view data frame
df
##     y x1 x2
## 1   6  2 14
## 2   8  5 12
## 3  12  4 12
## 4  14  3 13
## 5  14  4  7
## 6  15  6  8
## 7  17  7  7
## 8  22  5  4
## 9  24  8  6
## 10 23  9  5
```

Multiple linear regression model is fit to the dataset. k-fold cross validation with k=5 is performed to evaluate the model performance.

The 3 metrics, RMSE, R-squared, and MAE are used to decide which model is the best.

- **RMSE :** the average difference between the predictions made by the model and the actual observations.
- **Rsquared :** the correlation between the predictions made by the model and the actual observations.
- **MAE :** the average absolute difference between the predictions made by the model and the actual observations.

```
library(caret)
## Zorunlu paket yükleniyor: ggplot2
## Zorunlu paket yükleniyor: lattice

#specify the cross-validation method
ctrl <- trainControl(method = "cv", number = 5)

#fit a regression model and use k-fold CV to evaluate performance
model <- train(y ~ x1 + x2, data = df, method = "lm", trControl = ctrl)
```

```
#view summary of k-fold CV
print(model)
## Linear Regression
##
## 10 samples
##  2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 8, 8, 8, 8, 8
## Resampling results:
##
##   RMSE      Rsquared  MAE
##   3.371476  1         3.054156
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

The final model formula is :

$$\mathbf{y = 21.2672 + 0.7803*(x_1) - 1.12538(x_2)}$$

```
#view final model
model$finalModel
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Coefficients:
## (Intercept)           x1           x2
##     21.2672       0.7803      -1.1253
```

The following code displays the model predictions made for each fold :

```
#view predictions for each fold
model$resample
##       RMSE Rsquared      MAE Resample
## 1 2.440045        1 2.011223    Fold1
## 2 4.026464        1 3.980386    Fold2
## 3 1.741127        1 1.679767    Fold3
## 4 5.530702        1 4.768203    Fold4
## 5 3.119044        1 2.831199    Fold5
```

Article : https://www.statology.org/k-fold-cross-validation-in-r/

4 In-class Exercise 1

Canseza Avağ Erdurak
2022-10-21

First of all, I installed packages namely **dplyr**, **nycflights13** as well as **stringr**.

I used **sub** function in **stringr** package to create a driven column, aircraft_type (ac_type).

```
library(dplyr)
library(nycflights13)
library(stringr)

options(dplyr.summarise.inform = FALSE)
```

I take a high-level look at planes data which has 3,222 rows and 9 columns. I also have information about what column names are, which data type they have and how their content is in a short notice.

```
planes %>% glimpse()
```

```
Rows: 3,322
Columns: 9
$ tailnum      <chr> "N10156", "N102UW", "N103US", "N104UW", "N10575", "N105UW~
$ year         <int> 2004, 1998, 1999, 1999, 2002, 1999, 1999, 1999, 1999, 199~
$ type         <chr> "Fixed wing multi engine", "Fixed wing multi engine", "Fi~
$ manufacturer <chr> "EMBRAER", "AIRBUS INDUSTRIE", "AIRBUS INDUSTRIE", "AIRBU~
$ model        <chr> "EMB-145XR", "A320-214", "A320-214", "A320-214", "EMB-145~
$ engines      <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
$ seats        <int> 55, 182, 182, 182, 55, 182, 182, 182, 182, 182, 55, 55, 5~
$ speed        <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
$ engine       <chr> "Turbo-fan", "Turbo-fan", "Turbo-fan", "Turbo-fan", "Turb~
```

4.1 Create a Dataset

I created a dataset according to manufacturer column. I filtered planes which are manufactured by Airbus, Airbus Industrie and Boeing. I recoded AIRBUS INDUSTRIE as AIRBUS to have a clean dataset.

```
ab_df <- planes %>%
          select(tailnum:engine) %>%
          filter(manufacturer %in% c('AIRBUS', 'AIRBUS INDUSTRIE', 'BOEING')) %>%
          mutate(manufacturer = recode(manufacturer, 'AIRBUS INDUSTRIE' = 'AIRBUS'))
```

4.2 Top 5 Year by Manufacturers

Here is shown top 5 plane count, average number of seats and engines by manufacturer and year. Top 5 years by manufacturers are mostly overlaps. 5th elements in both Boeing and Airbus have the highest change compared to last years respectively.

```
ab_df %>%
  group_by(manufacturer, year) %>%
  summarise(count = n(), avg_seats = mean(seats), avg_engines = mean(engines)) %>%
  mutate(yoy_count = (count-lag(count))/lag(count)*100) %>%
  relocate(yoy_count, .after = count) %>%
  top_n(5, count) %>%
  arrange(manufacturer, desc(count)) %>%
  print(n = Inf)
```

```
# A tibble: 10 x 6
# Groups:   manufacturer [2]
   manufacturer  year count yoy_count avg_seats avg_engines
   <chr>        <int> <int>     <dbl>     <dbl>       <dbl>
 1 AIRBUS        2001    82      2.5       186            2
 2 AIRBUS        2000    80     35.6       182.           2
 3 AIRBUS        2002    60    -26.8       175.           2
 4 AIRBUS        1999    59     15.7       184.           2
 5 AIRBUS        1998    51    325         185.           2
 6 BOEING        2001   142      5.97      172.           2
 7 BOEING        2000   134      8.06      176.           2
 8 BOEING        1999   124     20.4       180.           2
 9 BOEING        1998   103    119.        182.           2
10 BOEING        2004    77    157.        148.           2
```

4.3 Aircraft Type Analysis

A new column is created named "ac_type" derived from planes' model. I work in Pegasus currently. As far as I know there is no ac_type starts with "MD" in the fleet. So, I excluded them. As the seat capacity increases, plane count generally decreases independently of manufacturer.

```
ab_df %>%
  mutate(ac_type = sub("\\-.*", "", model)) %>%
  filter(ac_type != "MD") %>%
  group_by(manufacturer, ac_type) %>%
  summarise(count = n(), avg_seats = mean(seats), med_seats = median(seats), avg_engines =
  arrange(manufacturer, desc(count))
```

```
# A tibble: 12 x 6
# Groups:   manufacturer [2]
   manufacturer ac_type count avg_seats med_seats avg_engines
   <chr>        <chr>   <int>     <dbl>     <dbl>       <dbl>
 1 AIRBUS       A320      415      193.       200        2
 2 AIRBUS       A319      208      164.       179        2
 3 AIRBUS       A321       94      297.       379        2
 4 AIRBUS       A330       18      377.       377        2.11
 5 AIRBUS       A340        1      375        375        4
 6 BOEING       737      1037      153.       149        2
 7 BOEING       757       308      186.       178        2
 8 BOEING       767       170      315.       330        2
 9 BOEING       717        88      100        100        2
10 BOEING       777        12      400        400        2
11 BOEING       787         4      260        260        2
12 BOEING       747         1      450        450        4
```

4.4 Seat Count Group Analysis

I created 8 bins sized 50. Fleet consists of planes which have seats between 150 and 200 mostly.

```
ab_df %>%
  group_by(seat_group = cut(seats, c(0,50,100,150,200,250,300,350,400,450,500), include.lo
  summarise(count = n())
```

```
# A tibble: 8 x 2
  seat_group count
  <fct>      <int>
1 (50,100]     102
2 (100,150]    934
3 (150,200]   1035
4 (200,250]     13
5 (250,300]     85
6 (300,350]    114
7 (350,400]     82
8 (400,450]      1
```

4.5 Conclusion

Consequently, I analysed data based on year, aircraft type and seat counts.

1. Top 5 plane count over years doesn't change much according to manufacturer. Airbus and Boeing have gone hand-in-hand in manufacture-wise over years.
2. There is a negative relationship between seat capacity and plane count. As the seat capacity increases, plane count decreases and engine count increases. Since maintenance costs are high in aircraft industry, companies may not want to prefer planes that have 4 engines. Therefore, production amount in high-engine sized planes are lower.
3. Fleet mostly consists of planes which have a seat capacity between 150 and 200. This analysis is aligned with the 2nd one.

Enjoy reading ! :)

5 ShinyApp Assignment

Canseza Avağ Erdurak
2022-11-27

First of all, I installed packages.

```
library(dplyr)
library(readxl)
library(tidyr)
library(ggplot2)
library(reshape)
library(tidyverse)
library(scales)
library(openxlsx)

options(dplyr.summarise.inform = FALSE)
```

I fetched raw data and stored in **students** variable.

```
students<-read_excel("foreign_students_by_nationality_2021_2022.xlsx")
```

I high-level look at raw data which has 7 columns and 10.461 rows.

```
students %>% glimpse()
```

```
Rows: 10,461
Columns: 7
$ `Üniversite Adı`  <chr> "ABDULLAH GÜL ÜNİVERSİTESİ", "ABDULLAH GÜL ÜNİVERSİT~
$ `Üniversite Türü` <chr> "DEVLET", "DEVLET", "DEVLET", "DEVLET", "DEVLET", "D~
$ `İl Adı`          <chr> "KAYSERİ", "KAYSERİ", "KAYSERİ", "KAYSERİ", "KAYSERİ~
$ Uyruk             <chr> "AFGANİSTAN İSLAM CUMHURİYETİ", "ALMANYA FEDERAL CUM~
$ E                 <chr> "1", "1", "0", "8", "4", "1", "1", "1", "1", "1", "1~
$ K                 <chr> "0", "0", "1", "2", "3", "0", "0", "1", "2", "0", "0~
```

```
$ T                 <chr> "1", "1", "1", "10", "7", "1", "1", "2", "3", "1", "~
```

I changed column names so that I could omit white spaces and Turkish characters.

```
new_cols <- c("university","university_type","country","nationality","male","female","tota

colnames(students) <- new_cols
```

I removed any incomplete cases.

```
students <- na.omit(students)
```

I selected columns from **university** to **female**.

```
students <- students %>%
  select('university':'female')
```

I unpivoted male and female column as **gender** column by using **melt** function.

```
students <- reshape2::melt(students, id = c("university", "university_type", "country", "n
```

I converted count column as numeric.

```
students$count <- as.numeric(students$count)
```

I saved file as **RDS**.

```
saveRDS(data, file = "apps/foreign_students_app/students.Rds")
```

I created another excel file named **foreign_students.xlsx** derived from raw data. I used this file in shiny app.

```
write.xlsx(students, "ForeignStudentAnalysis/foreign_students.xlsx")
```

5.1 Shinyapps.io

https://avagcanseza.shinyapps.io/foreign_students_app/

5.2 Command for local running

```
shiny::runGitHub(repo = "pjournal/mef06-avagcanseza",subdir="apps/foreign_students_app")
```

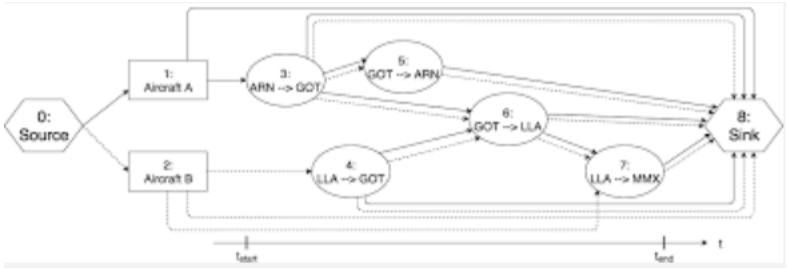
6 Operations Research

Canseza Avağ Erdurak
2022-12-26

Optimization is at the heart of business for many industries, especially for aviation. Tail assignment helps aviation companies fly more with a given number of aircrafts. When capacity utilization problem is solved, fuel-saving, reduction in operating costs and increase in on-time performance are achieved. They are all connected. The trick is to assign the best flight to an aircraft, while respecting the operational constraints.

An example would be organizing weekend plans with friends. You have to consider where / when to meet, whom to meet with, if you want to do some extra work before meeting like buying gifts, you have to leave home early, etc. You should organize your plan in a way that time and place can fit best. You should put them in order. While planning, you should also consider your friends' plans. So, scheduling is not an easy job to do even in personal life.

Aviation companies cope with this problem in their daily operations in order to build the most efficient schedule for sequencing the flights that each aircraft will perform.



6.1 Problem Description

Air France has a limited number of aircrafts. So, the Operations Research Team should assign the best-fitted flights for available aircrafts by maximizing the operational efficiency.

6.2 Model Building

The team modeled three criteria based on business expertise and statistical analysis of the past.

1. Fuel consumption estimation for each craft
2. Maintenance schedule estimation for individual parts of each aircraft
3. Cost estimation of each flight delay

A robust model is designed to help the operation run smoothly. At first, the model was working well with flights lasting from 6 to 12 hours. It was quite challenging to adapt the model with flights lasting 3 to 6 hours. Because shorter flights add much more complexity to the model by increasing the scheduled flights. That is where Gurobi Optimizer came in. The team used this tool to figure out an optimal or near-optimal solution to the tail assignment problem.

The team also developed a Decision Support Tool to help the internal users make decisions based on experience in case of any needs. The users are informed about the effects of their decisions on the running operations via this tool.

6.3 Benefits

Air France experienced the benefits below after building the tail assignment model :

- 1% decrease in fuel costs
- Saving on operating costs
- Reduction in delay propagations
- Optimizing business processes with the use of data

Here is the link to the case study.

Enjoy reading!