Intro

Things to keep in mind while doing a take-home challenge:

- Unless otherwise specified, you want to use R or Python. Comment the code as much as possible. They are also evaluating how clear is your code. Also, anyone should be able to understand the conclusions of your take-home even if they are not familiar with the language you used.
- Check the data. Never assume data is right. Always check data reliability and, if you find that some data doesn't make sense, clean it. This is also a big part of a data scientist job. There are companies that send take-homes that are only about identifying everything that is wrong with the data!
- Take-home challenges are usually fairly open ended. Play to your strengths: this could mean spending more time on visualization, machine learning, product ideas, or business insights depending on your skills.
- Don't make the solution over complicated. Focus on a few things and make sure the overall message is clear and consistent.
- Along the same lines: when you have to build a machine learning model, don't spend days optimizing its accuracy (this is not Kaggle, it is real world). Pick a model, explain why you picked that model and use parameters that make sense. You can then say what you would do if you had 1 more week to optimize it.
- Focus on the business impact that your work could have. How would the company benefit from your analysis? What would you suggest as a next step?
- If you find anything interesting in the data, by any means show it even if it is not related to the questions. If you find some info in the data that not even the hiring manager knows is there, you will pass the take-home for sure. After all, that's exactly why they will be hiring you: to discover things they don't know yet.
- A take-home challenge is rarely the place where over emphasizing your theoretical knowledge (unless specifically required in a question).
- Before extracting insights from a model, make sure your model predicts well. If your model doesn't predict well, its coefficients, splits, variable importance, etc. are totally irrelevant.
- The solutions provided in this book are in R. Obviously, Python would be perfectly fine too. Solutions are by no means exhaustive, and they simply show one (of the many) possible approaches.

Conversion Rate

Goal

Optimizing conversion rate is likely the most common work of a data scientist, and rightfully so.

The data revolution has a lot to do with the fact that now we are able to collect all sorts of data about people who buy something on our site as well as people who don't. This gives us a tremendous opportunity to understand what's working well (and potentially scale it even further) and what's not working well (and fix it).

The goal of this challenge is to build a model that predicts conversion rate and, based on the model, come up with ideas to improve revenue.

This challenge is significantly easier than all others in this collection. There are no dates, no tables to join, no feature engineering required, and the problem is really straightforward. Therefore, it is a great starting point to get familiar with data science take-home challenges.

You should not move to the other challenges until you fully understand this one.

Challenge Description

We have data about users who hit our site: whether they converted or not as well as some of their characteristics such as their country, the marketing channel, their age, whether they are repeat users and the number of pages visited during that session (as a proxy for site activity/time spent on site).

Your project is to:

- Predict conversion rate
- Come up with recommendations for the product team and the marketing team to improve conversion rate

Data

We have 1 table downloadable by clicking **here**.

```
The table is "conversion_data". It has information about signed-in users
during one session. Each row is a user session.
```

Columns:

- **country** : user country based on the IP address
- **age** : user age. Self-reported at sign-in step
- **new_user** : whether the user created the account during this session or had already an account and simply came back to the site
- **source** : marketing channel source
    - Ads: came to the site by clicking on an advertisement
    - Seo: came to the site by clicking on search results
    - Direct: came to the site by directly typing the URL on the browser
- **total_pages_visited**: number of total pages visited during the session. This is a proxy for time spent on site and engagement during the session.
- **converted**: this is our label. 1 means they converted within the session, 0 means they left without buying anything. The company goal is to increase conversion rate: # conversions / total sessions.

Example

```
Let's now check the characteristics of the user in the first row.
```

head(conversion_data, 1)

Field	Value	Description
country	UK	the user is based in the UK
age	25	the user is 25 yr old
new_user	1	she created her account during this session
source	Ads	she came to the site by clicking on an ad
total_pages_visited	1	she visited just 1 page during that session
converted	0	this user did not buy during this session. These are the users whose behavior we want to change!