EXPERT INSIGHT

Data Analysis and Machine Learning with Kaggle

How to win competitions on Kaggle and build a successful career in data science

Konrad Banachewicz
Luca Massaron

Packt

Data Analysis and Machine Learning with Kaggle

This is an Early Access product. Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the content and extracts of this book may evolve as it is being developed to ensure it is up-to-date.

The information contained in this book is sold without warranty, either express or implied. Neither the author nor Packt Publishing or its dealers and distributors will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Early Access Publication: Data Analysis and Machine Learning with Kaggle

Early Access Production Reference: B17574

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham

B3 2PB, UK

ISBN: 978-1-80181-747-9

www.packt.com

Table of Contents

Data Analysis and Machine Learning with Kaggle: How to win competitions on Kaggle and build a successful career in data science

Welcome to Packt Early Access. We're giving you an exclusive preview of this book before it goes on sale. It can take many months to write a book, but our authors have cutting-edge information to share with you today. Early Access gives you an insight into the latest developments by making chapter drafts available. The chapters may be a little rough around the edges right now, but our authors will update them over time. You'll be notified when a new version is ready. This title is in development, with more chapters still to be written, which means you have the opportunity to have your say about the content. We want to publish books that provide useful information to you and other customers, so we'll send questionnaires out to you regularly. All feedback is helpful, so please be open about your thoughts and opinions. Our editors will work their magic on the text of the book, so we'd like your input on the technical elements and your experience as a reader. We'll also provide frequent updates on how our authors have changed their chapters based on your feedback. You can dip in and out of this book or follow along from start to finish; Early Access is designed to be flexible. We hope you enjoy getting to know more about the process of writing a Packt book. Join the exploration of new topics by contributing your ideas and see them come to life in print.

Introducing data science competitions

Competitive programming has a long story, starting in the 1970s with the first editions of the **ICPC**, the “**International Collegiate Programming Contest**”. In the original ICPC, small teams from universities and companies participated in a competition that required solving a series of problems using a computer program (at the beginning participants coded in FORTRAN). In order to achieve a good final rank, teams had to display good skills in team working, problem solving and programming.

The experience of participating in the heat of such a competition and the opportunity to have a spotlight for recruiting companies provided the students enough motivation and it made the competition popular for many years. Among ICPC finalists, a few ones have become renowned. Among these, there is Adam D'Angelo, the former CTO of Facebook and founder of Quora, Nikolai Durov, the co-founder of Telegram Messenger, and Matei Zaharia, the creator of Apache Spark. Together with many others professionals, they all share the same experience: having taken part to an ICPC edition.

After ICPC, programming competitions flourished, especially after 2000, when remote participation become more feasible, allowing international competitions more easily and at a lower cost. The format is similar and simply the same for most of these competitions: there is a series of problems and you have to code a solution to solve them. The winners can then take a prize, but also make themselves noticed by recruiting companies or simply become famous and popular among their peers.

In this chapter, we will explore how competitive programming evolved into data science competitions, why the Kaggle platform is the most popular site for such competitions and how it works.

The rise of data science competition platforms

Commonly, problems in competitive programming range from combinatory to number theory, graph theory, algorithmic game theory, computational geometry, string analysis, and data structures. Recently also problems relative to artificial intelligence have successfully emerged, in particular after the launch of the KDD Cup, a contest in knowledge discovery and data mining, held by the **Association for Computing Machinery's** (**ACM**) **Special Interest Group** (**SIG**) on Knowledge Discovery and Data Mining, during its annual conference.

The first KDD cup, held in 1997, involved a problem on direct marketing for lift curve optimization and it started a long series of competitions (you can find the archives containing datasets, instructions, and winners at: https://www.kdd.org/kdd-cup) that continues up to nowadays (here is the latest available at the time of writing: https://www.kdd.org/kdd2020/kdd-cup). KDD cups proved quite effective in establishing best practices with many published papers describing solutions and techniques and competition dataset sharing that has been useful for many practitioners for experimentation, education and benchmarking.

The experience of competitive programming and KDD cups together gave rise to data science competition platforms, platforms where companies can host data science challenges that are somehow hard to solve and that could benefit from a crowdsourcing approach. In fact, given the fact that there is no golden approach for all the problems in data science, many problems require a time-consuming approach of the kind of try-all-that you-can-try.

In fact, no algorithm on the long run can beat all the others on all the problems, but each machine learning algorithm performs if and only if its space of hypothesis comprises the solution. Yet you cannot know that beforehand, hence you have to try and test to be assured that you are doing the right thing. You can consult the no free lunch theorem for a theoretical explanation of this practical truth, here is a complete article from Analytics India Magazine on the topic: https://analyticsindiamag.com/what-are-the-no-free-lunch-theorems-in-data-science/.

Crowdsourcing proves ideal in such conditions when you need to test in an extensively manner algorithms and data transformations in order to find the best possible combinations but you lack man and computer power for that. That's why for instance, governments and companies resort to competitions in order to advance in certain fields. On the government side, we can quote DARPA and its many competitions on self-driving cars, robotic operations, machine translation, speaker identification, fingerprint recognition, information retrieval, OCR, automatic target recognition, and many others. On the business side, we can quote a company such as Netflix, which entrusted a competition in order to improve its algorithm to predict user movie selection.

The Netflix competition was based on the idea of improving existing collaborative filtering, whose purpose is simply to predict the potential rating of a user for film, solely based on the previous ratings that the user gave on other films without knowing in specific who the user is or what the films are. Since no user description or movie title or description were available (all replaced by identity codes), the competition required to develop smart ways to use the available past ratings. The grand prize of USD $1,000,000 was to be assigned only if the solution could improve the existing Netflix algorithm, Cinematch, above a certain threshold. The competition run from 2006 to 2009 and in the end saw a team made up by the fusion of many previously teams in competitions (a team from Commendo Research & Consulting GmbH, Andreas Töscher and

Michael Jahrer, quite renown also in Kaggle competitions, two researchers from AT&T Labs and two others from Yahoo!). In the end, winning the competition required so much computation power and the ensembling of different solutions, that teams were forced to merge in order to keep pace. Such situation, reflected also on the actual usage of the solution by Netflix, which preferred not to implement it, but to simply take the most interesting insight from it in order to improve its existing Cinematch algorithm (you can read more about it on this Wired article: https://www.wired.com/2012/04/netflix-prize-costs/). What mattered more in the end of the Netflix competition has not been the solution per se, which has been quickly superseded by the change in business of Netflix from DVDs to online movies. The real benefit for both participants (who gained a huge reputation in collaborative filtering) and the company (who could transfer its improved recommendation knowledge to its new business) were the insights that has been gained from participating in the competition (and we can advance that knowledge will be also the leitmotiv of most of this book).

Kaggle competition platform

Other companies than Netflix indeed benefitted from data science competitions. The list is indeed long but we can quote a few examples where the company holding the competition reported a clear benefit from it. For instance, we can quote the insurance company AllState that could improve its actuarial models built by their experts thanks to a competition involving hundreds of data scientists (https://www.kaggle.com/c/ClaimPredictionChallenge). As another well documented example, we can also mention about General Electric that could improve by 40% the industry standard on predicting arrival times of airline flights thanks to an analogue competition (https://www.kaggle.com/c/flight). Both these competitions were held on the Kaggle competition platform.

The Kaggle competition platform has until now held hundreds of competitions and these two are just a couple of examples of companies that successfully used its competitions to boost their own models and analytics efforts. Let's take a step back from specific competitions for a moment and let's talk about the Kaggle company, which is the common thread of all this book.

Kaggle took its first steps in February 2010 thanks to the idea of Anthony Goldbloom, an Australian trained economist (he has a degree in Economics and Econometrics from Melbourne University). After working at Australia's Department of Treasury and in the Research department at the Reserve Bank of Australia, Goldbloom worked in London as an intern at The Economist, the international weekly newspaper on current affairs, international business, politics, and technology. At The Economist he had occasion to write an article about big data that inspired his idea of building a competition platform that could crowdsource the best analytical experts in solving interesting machine learning problems (https://www.smh.com.au/technology/from-bondi-to-the-big-bucks-the-28yearold-whos-making-data-science-a-sport-20111104-1myq1.html). Since the crowdsourcing dynamics had a relevant part in the business idea for this platform, he derived the name Kaggle, which recalls by rhyme the term "gaggle" i.e. a flock of geese (the goose is also the symbol of the platform).

After moving to the Silicon Valley in the USA, his Kaggle start-up received $11.25 million in Series A funding from a round led by Khosla Ventures and Index Ventures, two quite renown venture capital firms. First competitions rolled out, the community grew and some of the initial competitors came to become quite prominent, such as Jeremy Howards, the Australian data scientist and entrepreneur, who, after winning a couple of competitions on Kaggle, become the President and Chief Scientist of the company. Jeremy Howard left his position as President in December 2013 and thereafter he started a new start-up, fast.ai (www.fast.ai), offering machine learning courses and a deep learning library for coders.

At the times there were other prominent Kagglers (the name to indicate frequent participants to competitions held by Kaggle) such as Jeremy Achin and Thomas de Godoy. After reaching the top 20 global rankings on the platform, they promptly decided to retire and to found their own company, DataRobot. They soon after started hiring their best employers among the participants in the Kaggle competitions in order to instill the best machine learning knowledge and practice into the software they were developing. Today DataRobot is an undoubted leader in autoML (automatic machine learning).

The Kaggle competitions claimed more and more attention from a larger audience and even Geoffrey Hinton, the Godfather of deep learning participated (and won) in a Kaggle competition hosted by Merck in 2012 (https://www.kaggle.com/c/MerckActivity/overview/winners). Kaggle has also been the platform where Francois Chollet launched his deep learning package Keras during the Otto Group Product Classification Challenge (https://www.kaggle.com/c/otto-group-product-classification-challenge/discussion/13632) and Tianqi Chen launched XGBoost, a speedier and more accurate version of the gradient boosting machines, in the Higgs Boson Machine Learning Challenge (https://www.kaggle.com/c/higgs-boson/discussion/10335).

Competition after competition the community revolving around Kaggle grew to touch one million in 2017, the same year as, during her keynote at Google Next, Fei-Fei Li, Chief Scientist at Google, announced that Google Alphabet was going to acquire Kaggle. Since then Kaggle has become part of Google.

Today the Kaggle community is still active and growing. It has offered to many of his participants opportunities to create their own company, to launch machine learning software and packages, to get interviews on magazines (https://www.wired.com/story/solve-these-tough-data-problems-and-watch-job-offers-roll-in/), to arrange a course on Coursera (https://www.coursera.org/learn/competitive-data-science), to write machine learning books (https://twitter.com/antgoldbloom/status/745662719588589568), to find their dream job and, most important, or just to learn more about skills and technicalities about data science.

Other competition platforms

Though this book focused on competitions on Kaggle, we cannot forget that many data competitions are held on private platforms or on other competitions platforms. In truth, most of the information you will find in this book will hold also for all the other competitions, since they all basically operate under similar principals and the benefits for the participants are more or less the same as Kaggle's.

Since many other competition platforms are localized in specific countries or are specialized in certain kinds of competitions, for completeness we will briefly introduce some of them, at least those we have some experience and knowledge of.

DrivenData (https://www.drivendata.org/competitions/) is crowdsourcing competition platform devoted to social challenges (see https://www.drivendata.co/blog/intro-to-machine-learning-social-impact/). The company itself is a social enterprise whose aim is to bring data science solutions, thanks to data scientists building algorithms for social good, to organizations tackling the world's biggest challenges. For instance you an read in this article, https://www.engadget.com/facebook-ai-hate-speech-covid-19-160037191.html, how Facebook has choosen DrivenData for its competition on building models against hate speech and misinformation.

Numerai (https://numer.ai/) is an AI-powered, crowd-sourced hedge fund based in San Francisco which hosts a weekly tournament in which you can submit your predictions on hedge fund obfuscated data and earn your prizes in the company's crypto currency, Numeraire.

CrowdAnalytix (https://www.crowdanalytix.com/community) a bit less active now, this platform used to host quite a few challenging competitions a short ago, as you can read from this blog post: https://towardsdatascience.com/how-i-won-top-five-in-a-deep-learning-competition-753c788cade1. Also the community blog is quite interesting for having an idea of what challenges you can find on this platform: https://www.crowdanalytix.com/jq/communityBlog/listBlog.html.

Signate (https://signate.jp/competitions) is a Japanese data science competition platform. It is quite rich in contests and it offers a ranking system similar to Kaggle's one (https://signate.jp/users/rankings).

Zindi (https://zindi.africa/competitions) is a data science competition platform from Africa. It hosts competitions focused on solving Africa's most pressing social, economic and environmental problems.

Alibaba Cloud (https://www.alibabacloud.com/campaign/tianchi-competitions) is a Chinese cloud computer and AI provider who has launched the Tianchi Academic competitions, partnering with academic conferences such as SIGKDD, IJCAI-PRICAI and CVPR and featuring challenges such as image-based 3D shape retrieval, 3D object reconstruction, or instance segmentation.

Analytics Vidhya (https://datahack.analyticsvidhya.com/) the largest Indian community for data science, offers a platform for data science hackatons.

CodaLab (https://codalab.lri.fr/) is instead a French-based data science competition platform, created as a joint venture between Microsoft and Stanford University in 2013. They feature a similar

Kernel feature (here called Worksheets: https://worksheets.codalab.org/) for knowledge sharing and reproducible modeling as Kaggle.

Other minor platforms are **CrowdAI** (https://www.crowdai.org/) from École Polytechnique Fédérale de Lausanne in Switzerland, **InnoCentive** (https://www.innocentive.com/), **Grand-Challenge** (https://grand-challenge.org/) for biomedical imaging, **DataFountain** (https://www.datafountain.cn/business?lang=en-US), **OpenML** (https://www.openml.org/) and the list could go on. You can always find a list of many on-running major competitions on the Russian community **Open Data Science** (https://ods.ai/competitions) and thus discover even new competition platforms from time to time.

The alternatives and opportunities besides Kaggle are quite a lot. The interesting aspect of such an abundance of opportunities is that you can find more easily a competition that could interests you more because of its specialization and data. Also, expect less competitive pressure on these challenges since they are less known and advertized. Also, expect less sharing among participants since no other competition platform up to now has reached the same richness of sharing and networking tools as Kaggle has.

Stages of a competition

A competition on Kaggle is arranged through different steps. By having a glance at each of them, you can get a better understanding at how a data science competition works and what to expect from it.

When a competition is launched, there are usually some posts on social media (for instance on Kaggle Twitter profile: https://twitter.com/kaggle) that announce it and a new tab will appear between active competitions at the page https://www.kaggle.com/competitions. If you click on the competition's tab, you'll be taken to the competition page.

Immediately, at a glance you can at least check if the competition will have prizes (and if it awards points and medals, a secondary consequence of participating in a competition), how many teams are at the moment involved and how much is still left for you to work on a solution.

There you can explore the overview menu first which will provide information to you about the topic of the competition, its evaluation metric (your models will be evaluated against that metric), the time line of the competition, the prizes and the legal or competition requirements. Usually the time line is a bit overlook, but it should be one of the first things you have to check, in fact it doesn't tell you simply when the competition starts and ends, but it will provide you with the rules acceptance deadline, which is usually from seven day to two weeks before the competitions closes. The rule acceptance deadline marks the time limit you can join the competition, by accepting its rules, and the team merger deadline: you can arrange to combine your team with other competitors' one just before that deadline, after that it won't be possible. In addition, the rules menu is quite often overlooked (with people just jumping to data), but it is important to check on them because they can tell you about the requirements of the competition. Among the key information you can get from the rules there is the eligibility for a prize and a few other important details such as if you can use external data to improve your score, how many submissions (tests of your solution) a day you get, how many final solutions you can choose, and so on. Finally, you can have a look at the data, though you can download it only after accepting the rules of the competition.

Once you have accepted the rules, you can download any data or directly start working on Kaggle Kernel, an only notebook, re-using code that others have made available to other participants or creating your own code from scratch. If you decide to download the data, also consider that you have a Kaggle API that can help you to run downloads and submission in an almost automated way. You can find more details about the API at https://www.kaggle.com/docs/api and you can get the code from Github from https://github.com/Kaggle/kaggle-api. By the way, if you closely check on Kaggle Github repo, you can also find all the docker images they use for their online notebooks the Kaggle Kernels. At this point, as you develop your solution, it is not a bad idea to contact other competitors through the discussion forum: there you can ask and answer questions. Often you will also find useful hints at specific problems with the data or even useful ideas there to improve your solution.

Once your solution is ready, you can submit it to the Kaggle evaluation engine, accordingly to the specification of the competition (some competitions will accept a csv file as a solution, other will require you to code and produce results in a Kaggle Kernel). During all the competition, you can keep submitting solutions.

Every time you submit a solution, the leaderboard will provide you soon after, depending on the computations necessary for the evaluation, with a score and a position among the competitors. That position is only indicative anyway, because it reflects the performance of your model on a part of the test set, called as the public one since the performance on it is made public during the competition for everyone to know. Only when the competition closes and the contestants have decided on what among their model have to be scored, it is reveled their score on another part of the test set, called the private one. This new leaderboard, the private leaderboard, constitutes the final, effective rankings of the competition.

When a competition is closed, the Kaggle team will take a certain time to check that everything is correct and that all contestants have respected the rules of the competition. After a while (and sometimes after some changes) the private leaderboard will become definitive, the winners will be declared, and many among participants, at their own will, will unveil their strategies, their solutions and their code to others on the competition discussion forum.

Types of competitions and examples

Kaggle competitions are categorized based on “competitions categories” and each category has different implications in terms of how to compete and what to expect from each of them. Type of data, difficulty of the problem, awarded prizes and competition dynamics are quite diverse inside the categories, therefore it is important to understand beforehand what each implies.

Here are the official categories that you can use to filter out the different competitions:

- Featured
- Masters
- Annuals
- Research
- Recruitment
- Getting started
- Playground
- Analytics
- InClass

“**Featured**” are the most common type of competitions, featuring a business related problem from a sponsor company and a prize for the top performers in the competition. The winners will grant a non-exclusive license of their work to the sponsor company and they will have to prepared a detailed report of their solution and sometimes even participate in meetings with the sponsor company.

There are examples of Featured competitions every time you access to Kaggle. At the moment, many of them are problems related to the application of deep learning methods to unstructured data like text, images, videos or sound but in the past tabular data competitions, that is competitions based on problems related to structured data that can be found in a database. Now such competitions are really less required because a crowdsourced solution would not often be such an advance in respect of what a good team of data scientists or even an autoML software could reach. First using random forests, then using gradient boosting methods with clever feature engineering, in the past tabular data solution derived from Kaggle could really largely improve a solution. Nowadays, given the spread of better software and good practices, the obtainable increased results from competitions could be indeed marginal. In the unstructured data world, instead, a good deep learning solution could still do the difference since, for instance, pretrained networks such as BERT brought about double-digit increases in previous standards of many well known with text.

“**Masters**” are less usual now, but they are private, invite-only competitions. The purpose was to create competitions only among experts (generally competitors ranked as Masters or Grandmasters, based on Kaggle medal rankings), based on their rankings on Kaggle.

“**Annuals**” are competitions that always appear on a certain period of the year. Among the Annuals we have the Santa Claus competitions (usually based on an algorithmic optimization problem) and the March Machine Learning Competition, run every year since 2014 during the US College Basketball Tournaments.

“**Research**” competitions imply a research or science purpose instead of a business one, sometimes for serving the public good. That’s why these competitions do not always offer prizes. In addition, these competitions sometimes require the winning participants to release their solution as open-source.

Google has released a few Research competitions in the past such as the Google Landmark Recognition 2020 (https://www.kaggle.com/c/landmark-recognition-2020) - Label famous (and not-so-famous) landmarks in images

Sponsors that want to test potential job candidates for their ability hold “**Recruitment**” competitions. These competitions are limited to teams of one and offer to best placed competitors an interview with the sponsor as a prize. The competitors have to upload their curriculum vitae at the end of the competition if they want to be considered for being contacted.

Examples of recruiting competitions have been:

- The Facebook Recruiting Competition (https://www.kaggle.com/c/FacebookRecruiting but actually Facebook held a few ones of these kind)
- The Yelp Recruiting Competition (https://www.kaggle.com/c/yelp-recruiting)

“**Getting started**” competitions do not offer any prize but a friendly and easy problem for beginners to get accustomed by Kaggle principles and dynamics. Usually they are semi-permanent competitions whose leaderboard is refreshed from time to time. If you are looking for a tutorial in machine learning, these competitions are the right place where to start.

Famous on-running Getting Started competitions are:

- **Digit Recognizer** (https://www.kaggle.com/c/digit-recognizer)
- **Titanic**: Machine Learning from Disaster (https://www.kaggle.com/c/titanic) - Predict survival on the Titanic
- **Housing Prices**: Advanced Regression Techniques (https://www.kaggle.com/c/house-prices-advanced-regression-techniques)

“**Playground**” competitions are a little bit more difficult than the Getting Started ones, but they are also meant for having competitors learn and test their abilities without the pressure of a fully fledged Featured competition (though sometimes also in Playground competitions the heat of the competition may turn quite high). The usual prizes of such competitions are just swag or little money.

One famous Playground competition has been the original “Dogs vs Cats” competition (https://www.kaggle.com/c/dogs-vs-cats) - Create an algorithm to distinguish dogs from cats

A mention should be spent for Analytics competitions, where the evaluation is qualitative and participants are required to provide ideas, drafts of solutions, PowerPoint slides, charts and so on and the **InClass** which are competitions held by academic institutions.

Cross-sectional to this taxonomy of Kaggle competitions, you also have to consider that competitions may have different formats. The usual format is the so-called “Simple format” where you provide a solution and it is evaluated as we previously described. More sophisticated, the two-stage competition splits the contested into two parts and the final dataset is released only after the first part has completed and only to the participants of the first phase. The two-stage competition format has emerged in order to limit the chance that some competitor may cheat and infringe the rules since the evaluation is done on a completely untried test set that it is available for a short time. For the same reason, also the Code competitions have recently appeared, where all submissions are made from a Kaggle Notebook, and any direct upload of submissions is disabled.

Submission and leaderboard dynamics

Apparently, the way Kaggle works seems simple: the test set is hidden to participants; you fit your model, if your model is the best in predicting the test set, then you score high and you possibly win. Unfortunately, such description renders the inner workings of Kaggle competitions in a too simplistic way and it doesn't take into account that there are dynamics regarding the direct and indirect interactions of competitors between themselves and the nuances of the problem you are facing and of its training and test set.

A more comprehensive description of how Kaggle works is actually given by Professor David Donoho, professor of statistics at Stanford University (https://web.stanford.edu/dept/statistics/cgi-bin/donoho/), in his writing "50 Years of Data Science". The paper has first appeared in the Journal of Computational and Graphical Statistics and then subsequently posted on the MIT Computer Science and Artificial Intelligence Laboratory (see http://courses.csail.mit.edu/18.337/2015/docs/50YearsDataScience.pdf). Professor Donoho does not refer to Kaggle specifically, but to all data science competition platforms generically. Quoting computational linguist Mark Liberman, he refers to data science competitions and platforms as part of a **Common Task Framework** (**CTF**) paradigm that has been silently and steadily progressing data science in many fields during the last decades. He states that a CTF can work incredibly well at improving the solution of a problem in data science from an empirical point of view. He quotes as successful examples the NetFlix competition and many DARPA competitions that have reshaped the best in class solutions for problems in many fields.

A CTF is composed of "ingredients" and a "secret sauce" (see paragraph 6 of Donoho, David. "50 years of data science." Journal of Computational and Graphical Statistics 26.4 (2017): 745-766.). The ingredients are simply:

1. A public available dataset and a related prediction task
2. A set of competitors who share the common task of producing the best prediction for the task

3. A system for scoring the predictions by the participants in a fair and objective way, without providing too specific hints (or limiting them at most) about the solution

The system works the best if the task is well defined and the data is of good quality. On the long run, the performance of solutions improves by small gains until they reach an asymptote. The process can be speeded up by allowing a certain amount of sharing among participants (as it happens on Kaggle by means of discussions and sharing Kernel notebooks and extra data by Datasets). It can be noticed that the only competitive pressure, in spite of any degree of sharing among participants doesn't stop the improvement in the solution, it just makes it slower.

This is because the secret sauce in the CTF paradigm is the competition itself, that, in the framework of a practical problem whose empirical performance has to be improved, always leads to the emergence of new benchmarks, new data and modeling solutions, and in general to an improved application of machine learning to the problem objected of the competition. A competition can therefore provide a new way to solve a prediction problem, new ways of feature engineering, new algorithmic or modeling solutions. For instance, deep learning has not simply emerged from academic research, but it has first gained a great boost because of successful competitions that declared its efficacy (we have already mentioned for instance the Merck competition won by Geoffrey Hinton's team https://www.kaggle.com/c/MerckActivity/overview/winners).

Coupled with the open software movement, which allows everyone to access to powerful analytical tools (such as Scikit-learn or TensorFlow or PyTorch), the CTF paradigm brings even better results because all competitors are on the same line at the start. On the other end, the reliance of a solution to a competition on specialized or improved hardware can limit the achievable results because it can prevent competitors without access to such resources to properly participate and contribute directly to the solution or indirectly by exercising competitive pressure on the other participants. Understandably, that the reason why Kaggle started also offering cloud services for free to participants to its competitions (the Kernels): it can flatten some differences in hardware intense competitions (like most deep learning ones are) and increase the overall competitive pressure.

There are, anyway, occurrences that can go wrong and instead led to a suboptimal result in a competition:

1. Leakage from the data
2. Probing from the Leaderboard (the scoring system)
3. Overfitting and consequent shake-up
4. Private Sharing

You have leakage from data when part of the solution is to be retraced in the data itself. For instance, certain variables are posterior to the target variable (so they reveal something of it) or the ordering of the training and test examples or some identifier is evocative of the solution. Such solution leakage, sometimes named by the competitors as “golden features” (because getting a hint of such nuances in the data problem could turn into gold prizes for the participants) invariably leads to a not reusable solution. It also implies a suboptimal result for the sponsor (who at least should have learned something about leaking features that should affect a solution to his/her problem).

Another problem is the possibility to probe a solution from the leaderboard. In such situation, you can snoop the solution by repeated submission trials on the leaderboard. Again, in this case the solution is completely unusable in different circumstances. A clear example of such happened in the competition “Don’t Overfit II” (https://www.kaggle.com/c/dont-overfit-ii). In this competition, the winning participant, Zachary Mayers, submitted every individual variable as a single submission, gaining information on the possible weight of each variable that allowed him to estimate the correct coefficients for his model (you can read Zach's detailed solution here: https://www.kaggle.com/c/dont-overfit-ii/discussion/91766). Generally, time series problems (or other problems where there are systematic shifts in the test data) may be seriously affected by probing, since they can help competitors to successfully define some kind of 'post-processing' (like multiplying their predictions by a constant) that is most suitable for scoring high on the specific test set.

Another form of leaderboard snooping happens when participants tend to rely more on the feedback from the public leaderboard than their own tests. Sometimes such turns into a complete failure of the competition, with a wild shake-up, that is a complete and unpredictable reshuffling of the competitors' positions on the final leaderboard. The winning solutions, in such a case, may (but not always) turn out to be not so optimal for the problem or even sometimes just dictated by chance. This has led to precise analysis by competitors if the training set is much different from the test set they have to guess. Such analysis, called "Adversarial testing" can provide insight if to rely on the leaderboard and if there are features that are too different from the training and test set to be better avoided completely (For an example, you can have a look at this Kernel by Bojan Tunguz: https://www.kaggle.com/tunguz/adversarial-ieee). Another form of defense against leaderboard overfitting is choosing safe strategies to avoid submitting solutions too based on the leaderboard results. For instance, since two solutions are allowed to be chosen by each participant for the evaluation at the end of the competition, a good strategy is to submit the best one performing on the leaderboard and the best performing one based on one's own cross-validation tests.

In order to avoid problems with leaderboard probing and overfitting, Kaggle has recently introduced different innovations based on Code Competitions, where the evaluation is split into two distinct stages, that is you have a test set for the public leaderboard (the leaderboard you follow during the competition) and a completely hold out test set for the final private leaderboard. In this way, participants are actually blind to the actual data their solutions will be evaluated against and they should be forced to consider more their own tests and a more general solution against a solution specific to the test set.

Finally, another possible distortion of a competition is due to private sharing (sharing ideas and solutions in a closed circle of participants) and other illicit moves such as playing by multiple accounts or playing in multiple teams and stealing ideas from each in favor of another team. All such actions create an asymmetry in information between participants that can be favorable to a few and detrimental to the most. Again, the resulting solution may be affected because sharing has been imperfect during the competition and fewer teams have been able to exercise full competitive pressure. Moreover, if such situations appear evident to participants (for instance see https://www.kaggle.com/c/ashrae-energy-prediction/discussion/122503), it can lead to distrust and less involvement in the competition or in following competitions.

Computational resources

Some competitions do pose limitations in order to render available to production feasible solutions, for instance the Bosh Production Line Performance competition - https://www.kaggle.com/c/bosch-production-line-performance - had strict limits on execution time, model file output and memory limit for your solution. Also Kernel based competitions, when requiring both training and inference to be executed on Kernels, do not pose a problem for the resources you have to use because Kaggle will provide with all the resources you need (and this is also intended as a way to put all participants on the same line for a better competition result).

Problems for you arise when you have kernel competitions just limited to inference time and therefore you can train your models on your own machine and the only limit is then based at test time on the number and complexity of models you produce. Since at moment most competitions require deep learning solutions, you have to consider that you surely need specialized hardware such as GPUs in order to achieve some interesting result in a competition. Anyway, also if you participate in some of the now rare tabular competitions, you'll soon realize that you need a strong machine with quite a number of processors and memory in order to easily apply feature engineering to data, run experiments and build models quickly.

Standards do change rapidly, therefore it is difficult to mention a standard hardware that you should have in order to compete at least on the same league with others. We can anyway take a hint at such standard by looking at what other competitors are using, as their own machine or as a machine on the cloud.

For instance, recently HP has launched a program where it awarded a HP Z4 or Z8 to a few selected Kaggle participants in exchange with visibility for its brand. For instance, a Z8 machine has 56 cores, 3TB of memory, 48TB of storage (a good share by solid storage hard drives) and a NVIDIA RTX as GPU. We understand that such could be a bit out of reach for many as well as even also renting a similar machine for a short time on a cloud instance such as Google's GCP or Amazon's AWS is out of discussion for the consequent expenses for even a moderate usage.

Our suggestion, unless your ambition is to climb to the top rankings of Kaggle participants is therefore to go with the machines provided free by Kaggle, the Kaggle Notebooks (also previously known as the Kaggle Kernels).

Kaggle Notebooks are a versioned computational environment, based on Docker containers running in cloud machines, which allow you to write and execute both scripts and notebooks in R and Python languages. The Kaggle Notebooks are integrated into the Kaggle environment (you can make submissions from them and keep track what submission refers to what Notebook), they come with most data science packages pre-installed, and they allow some customization (you can download files and install further packages). The basic Kaggle Notebook is just CPU based, but you can have versions boosted by a NVIDIA Tesla P100 or a TPU v3-8 (TPUs are hardware accelerators specialized in deep learning tasks). Though bounded by a usage number and time quota limit, Kaggle Notebooks provide the computational workhorse to build your baseline solutions on Kaggle competitions:

- A CPU Notebook owns 4 CPU cores and 16 GB of memory, you can run 10 Notebooks of this kind at a time but you don't have any time quote for them
- A GPU features 2 CPU cores and 13 GB of memory, you can run 2 Notebooks of this kind at a time and you have a 30 hours weekly quota for such kind of Notebook
- A TPU features 4 CPU cores and 16 GB of memory, you can run 2 Notebooks of this kind at a time you have a 30 hours weekly quota for such kind of Notebook

All Notebooks can run for 9 hours maximum, and have 20 GB disk saving allowance to store your models and results plus an additional scratchpad disk that can exceed 20 GBs for temporary usage during script running.

In certain cases, the GPU enhanced machine provided by Kaggle kernels may not be enough. For instance, the recent Deepfake Detection Challenge (https://www.kaggle.com/c/deepfake-detection-challenge) required to process data consisting of about around 500 GB of videos. That is especially because of the time limit of weekly usage, that at the time of this writing is about 30 hours a week and because of the fact that you cannot have more than two machines with GPU running at the same time (10 machines at a time is the limit for the CPU only instances). Even if you can double your machine time by changing your code to leverage the usage of TPUs instead of GPUs (and you can find some guidance for achieving that easily here: https://www.kaggle.com/docs/tpu), that may still prove not enough for fast experimentation on a data heavy competitions such as the Deepfake Detection Challenge. That's the reason in the chapter devoted to Kaggle Kernel we are going to provide you with many tips and tricks for successful coping with such limitations with decent results without having to buy a heavy performing machine. We are also going to show you how to integrate Kaggle Kernels with **Google Cloud Services** (**GCP**) or simply how to move away all your work on another cloud based solution.

Teaming and networking

However, the computational power plays its part, the human expertise and ability only can make the real difference in a Kaggle competition. Sometimes a competition to be handled successfully requires the collaborative efforts of a team of contestants. Apart from recruitment competitions, where the sponsor may require individual participants for a better evaluation of their abilities, usually there is no limit to forming teams during competitions. Usually the teams can be at maximum made of five contestants. Teaming has its own advantages because it can multiply the efforts on finding a better solution, since a team can spend more time all together on the problem and different skills can be of great help: not all data scientists have the same skills or the same level of skill in the different models and data manipulations.

Anyway, teaming is not all positive. Coordinating different individuals and different efforts toward a common goal may prove not so easy and some suboptimal situations may arise. The usual problem with teams is when part of the participants is not involved or simply idle, but the worst is surely when, a more rare occurrence, someone infringes the rules of the competition (to the detrimental of everyone since the all team could be disqualified) or even spies on the team in order to advantage other ones.

In spite of any negative side, teaming in Kaggle competition is a great opportunity to know better other data scientists, to collaborate for a purpose and to achieve more, since Kaggle rules do reward teams in respect of lonely competitors. Teaming together is not the only possibility of networking in Kaggle, though it is certainly the more profitable and interesting for the participants. You can actually network with others by discussions on the forums, by sharing datasets and notebooks during competitions. All these opportunities on the platform can help you to know other data scientist and to be recognized by them.

There are also quite many occasions to network with other Kagglers outside of the Kaggle platform itself. First of all, there are a few Slack channels that can be helpful. For instance, KaggleNoobs (see: https://www.kaggle.com/getting-started/20577) is a channel, opened up 5 years ago, that feature many discussions about Kaggle competitions and they have a supportive community that can help you if you have some specific problem with code and models. There are quite a few other channels around devoted to exchanging opinions about Kaggle competitions and data science related topics. Some channels are organized on a regional or national basis. For instance, the Japanese channel Kaggler-ja (http://kaggler-ja-wiki.herokuapp.com/) or the Russian community, created six years ago, Open Data Science Network (https://ods.ai/) which later opened also to non-Russian speaking participants. The Open Data Science Network (mostly simply known as ODS) doesn't simply offer a Slack channel but also courses on how to win competitions, events, and reporting on active competitions around on all known data science platforms (see https://ods.ai/competitions).

Apart from Slack channels, also quite a lot of local meetups themed about Kaggle in general or about specific competitions have sprout

out, some for short time, others for longer. A meetup on Kaggle competition, usually built around a presentation from a competitor who wants to share her or his experience and suggestions, is the best situation to meet other Kagglers in person, to exchange opinions and to build alliances for participating together in data science contests. In this league, a mention apart is for Kaggle Days (https://kaggledays.com/), built by Maria Parysz and Paweł Jankiewicz, a renowned Kaggle competitor. The Kaggle Days organization arranged a few events in major locations around the World (https://kaggledays.com/about-us/) with the aim of bringing together a conference of Kaggle experts (which had to come to an abrupt stop due to the COVID-19 pandemic) and it created a network of local meetups in different countries which are still quite active (https://kaggledays.com/meetups/).

Performance tiers and rankings

Apart from monetary prizes, Kaggle offers many other more immaterial awards (apart from some material ones such as cups, t-shorts, hoodies and stickers). The point is that Kagglers, the participants in Kaggle competitions, do spend really a lot of time and efforts when in competition (not to count in the specialty skills they put on that in truth are quite rare in the general population). The monetary prizes usually cover the efforts of the few top ones, if not of the only top one, leaving the rest with an astonishing amount of hours just voluntary spent for no return. On the long, being in competition with no tangible result may lead to disaffection and disinterest, thus lowering the competitive intensity. Hence, Kaggle has found at least a way to reward competitors with an honor system based on medals and points. The idea is that the more medals and the more points, the more relevant are ones skills, opening for opportunities in job search or any other relevant activity based on reputation.

First, there is a general leaderboard, that combines all the leaderboards of the single competitions. In this general leaderboard (https://www.kaggle.com/rankings), one is ranked based on the position on each single competition she or he took, which awards some points that all summed together provide the ranking in the general leaderboard. At first glance, the formula for the scoring of the points in a competition may look a bit complex:

$$\left[\frac{100000}{\sqrt{N_{\text{teammates}}}}\right]\left[\text{Rank}^{-0.75}\right]\left[\log_{10}(1+\log_{10}(N_{\text{teams}}))\right]\left[e^{-t/500}\right]$$

Nevertheless, in reality it is simply based on a few ingredients: the rank in a competition, your team size, the popularity of the competition and how much the competition is old.

Intuitively, ranking high in popular competitions brings many points. Less intuitively, the size of your team matters in a non-linear way. That’s due to the inverse square root part of the formula since the part of points you have to give up grows with the number of people involved but it is still quite favorable if your team is relatively small (2, max 3 people) due the advantage in wits and computational power brought about by larger collaborative teams.

Another point to keep in mind is that point decay with time. The decaying is not linear, but as a rule of thumb just think that after a year very little is left of the points you gained. Therefore, glory on the general leaderboard of Kaggle cannot last long and it is ephemeral unless you keep on participating on competitions coming up with similar results as before. As a consolation, on your profile you’ll always keep the highest rank you ever reach, as a memento of your great combined results at a certain time.

More last longing is the medal system that covers all the four aspects of competing in Kaggle. You will be prized with medals in competitions, notebooks, discussion and datasets based on your results. In competitions, medals are awarded based on your position on the leaderboard. In other areas such as discussion, notebook and datasets medals are awarded based on the upvotes of other competitors (which actually led sometimes to some suboptimal situation since upvotes are a less objective metric and also depends on popularity). The more medals you get, the higher ranks of Kaggle mastery you can enter. The ranks are classified in Novice, Contributor, Expert, Master, and Grandmaster. The page https://www.kaggle.com/ progression explains everything about how to get medals and how many and of what kind are needed to access the different ranks.

Please keep in mind that such ranks and honors are always relative and that they do change in time. A few years ago, in fact the scoring system and the ranks were quite different. Most probably in the future, the ranks will change again in order to keep the higher ones rarer and thus more valuable.

Criticism and opportunities

Kaggle has drawn quite a few criticisms since its start and if participating to data science competitions is still quite debated today by many claiming different negative or positive opinions.

On the side of negative criticism:

- Kaggle provides a false perception of what machine learning really is since it is just focused on leaderboard dynamics
- Kaggle is just a game of hyper-parameter optimization and ensembling many models just for scraping a little more accuracy (while in reality overfitting the test set)
- Kaggle is filled with in-experienced enthusiasts who are ready to try anything under the sky in order to get a score and a spotlight in the hope to be spotted by recruiters

- As a further consequence, competition solutions are too complicated and often too specific of a test set to be implemented

Many perceive Kaggle, as many other data science competition platform, far from what data science is in reality. The point they raise are that business problems are not given from nowhere and you seldom already have a well-prepared dataset to start with since you usually built it along the way based on refining the business specifications and understanding of the problem at hand. Moreover, they emphasize that production is neither considered, since a winning solution cannot be constrained by resource limits or considerations about technical debt (though this is not always true for all competitions).

We cannot but not notice how all such criticism is related in the end about both the fact that Kaggle is a crowdsourcing experience with a purpose (the CTF paradigm) and how Kaggle ranking standings do relate in the data science world in comparison with data science education and work experience. One persistent myth that ailments criticism is in fact that Kaggle competitions may help getting you a job or a better job in data science or that performing in Kaggle competitions may put you on another plane in respect of data scientists that do not participate at all.

Our stance on such a myth is that it is misleading belief that Kaggle rankings do have an automatic value beyond the Kaggle community. For instance, in a job search, Kaggle can provide you with some very useful competencies on modeling data and problems and effective model testing. It can also expose you to many techniques and different data/business problems (even beyond your actual experience and comfort zone), but it cannot supplement you with everything you need to successfully place yourself as a data scientist in a company.

You can use Kaggle for learning and for differentiating yourself from other candidates in a job search; however, how this will be considered will considerably vary from company to company. Anyway, what will learn on Kaggle will invariably prove useful throughout all of your career and will provide you a hedge when you'll have to solve complex and unusual problems with data modeling because by participating in Kaggle competitions you build up strong competencies in modeling and validating. You also network with other data scientists and that can get you a reference for a job more easily and provide you with another way to handle difficult problems beyond your skills because you will have access to others' competencies and opinions.

Hence, our opinion is that Kaggle can more indirectly help you in your career as a data scientist and that it can do that in different ways. Of course, sometimes Kaggle will help you directly being contacted as a job candidate based on your competitions' successes, but more often Kaggle will be helpful by providing you with the intellectual and experience skills you need to succeed first as a candidate then as a practitioner. In fact, after playing with data and models on Kaggle for a while, you'll have had the chance to see enough different datasets problems and ways to deal with them under the pressure of time, that when faced with similar problems in real settings you'll get quite skilled in finding solutions quickly and effectively.

Actually, this latter opportunity of a skill upgrade is why we got motivated writing this book in the first place and what this book is actually about. In fact, you won't exactly find here a guide just about how to win or score high on Kaggle competitions (there are also online resources that can enlighten you on that, actually) but you'll absolutely will find a guide about how to compete better on Kaggle and how to get back the maximum from your competition experience.

Use Kaggle and other competition platforms in a smart way. Kaggle is not a passepartout, being first on a competition won't assure you a highly paid job or glory beyond the Kaggle community. However, consistently participating in competitions is instead a card to be played smartly to show interest and passion in your data science job search and to improve some specific skills that can differentiate you as a data scientist and not make you obsolete in front of autoML solutions.

If you are going to follow us along this book, we will show you how.

Organizing Data with Datasets

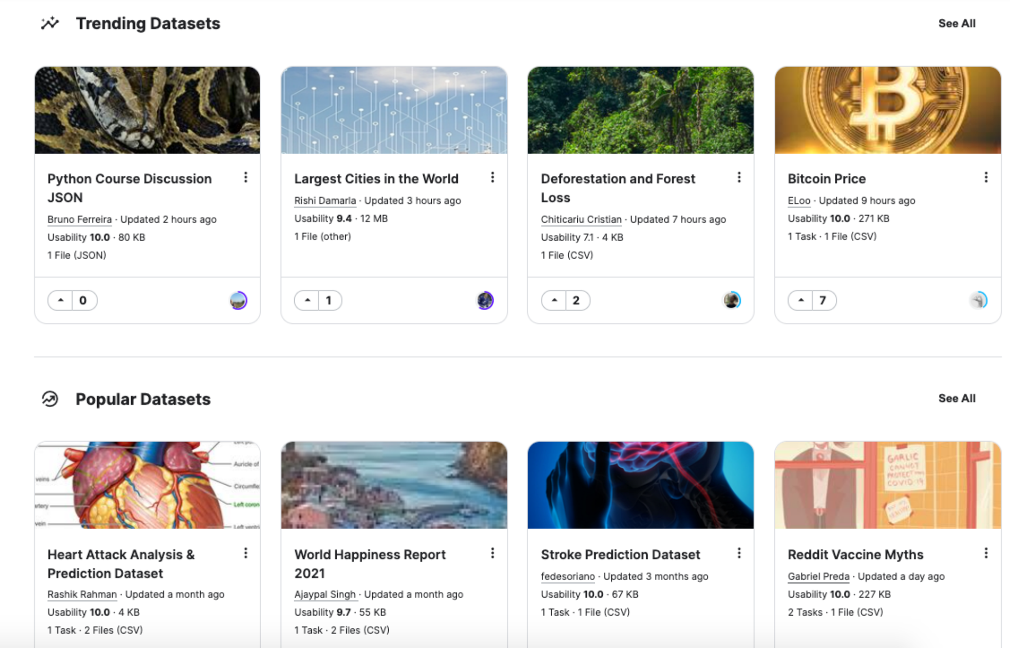
In his story “The Adventure of the Copper Breeches”, Arthur Conan Doyle has Sherlock Holmes shout “*Data! Data! Data! I cannot make bricks without clay*”—and this mindset, which served the most famous detective in literature so well, should be adopted by every data scientist. For that reason, we begin the more technical part of this book with a chapter dedicated to data: specifically, in the Kaggle context, leveraging the power of Kaggle Datasets functionality for our purposes.

Setting up a dataset

In principle, any data you can use (subject to limitations—see the legal caveats section below), you can upload to Kaggle. The specific limits at the time of writing this book are: 20 gigabytes per dataset and 100 gb total quota. Keep in mind that the size limit per single dataset is calculated uncompressed—uploading compressed versions speeds up the transfer but does not help against the limits. You can check the most recent documentation of the datasets at this link:

https://www.kaggle.com/docs/datasets

Kaggle promotes itself as a “home of open data science” and the impressive collection of datasets available from the site certainly lends some credence to that claim: before uploading the data for your project into a dataset, make sure to check the existing content—for several popular applications, there is a chance it has already been stored there:



Trending Datasets

Python Course Discussion JSON
Bruno Ferreira · Updated 2 hours ago
Usability 10.0 · 80 KB
1 File (JSON)

0

Largest Cities in the World
Rishi Damarla · Updated 3 hours ago
Usability 9.4 · 12 MB
1 File (other)

1

Deforestation and Forest Loss
Chiticariu Cristian · Updated 7 hours ago
Usability 7.1 · 4 KB
1 File (CSV)

2

Bitcoin Price
ELoo · Updated 9 hours ago
Usability 10.0 · 271 KB
1 Task · 1 File (CSV)

7

Popular Datasets

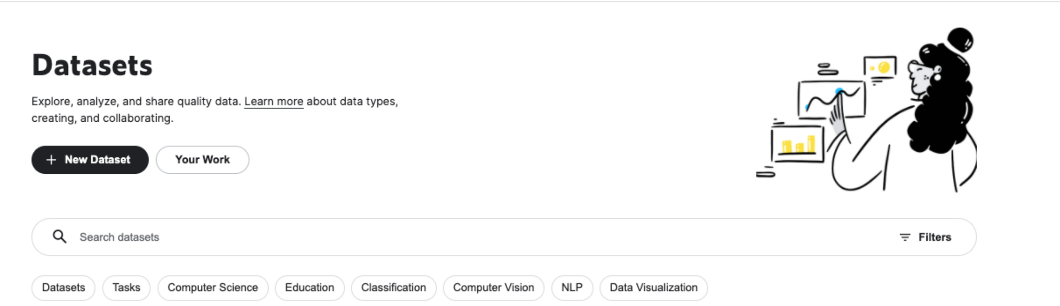
Heart Attack Analysis & Prediction Dataset
Rashik Rahman · Updated a month ago
Usability 10.0 · 4 KB
1 Task · 2 Files (CSV)

World Happiness Report 2021
Ajaypal Singh · Updated a month ago
Usability 9.7 · 55 KB
1 Task · 2 Files (CSV)

Stroke Prediction Dataset
fedesoriano · Updated 3 months ago
Usability 10.0 · 67 KB
1 Task · 1 File (CSV)

Reddit Vaccine Myths
Gabriel Preda · Updated a day ago
Usability 10.0 · 227 KB
2 Tasks · 1 File (CSV)

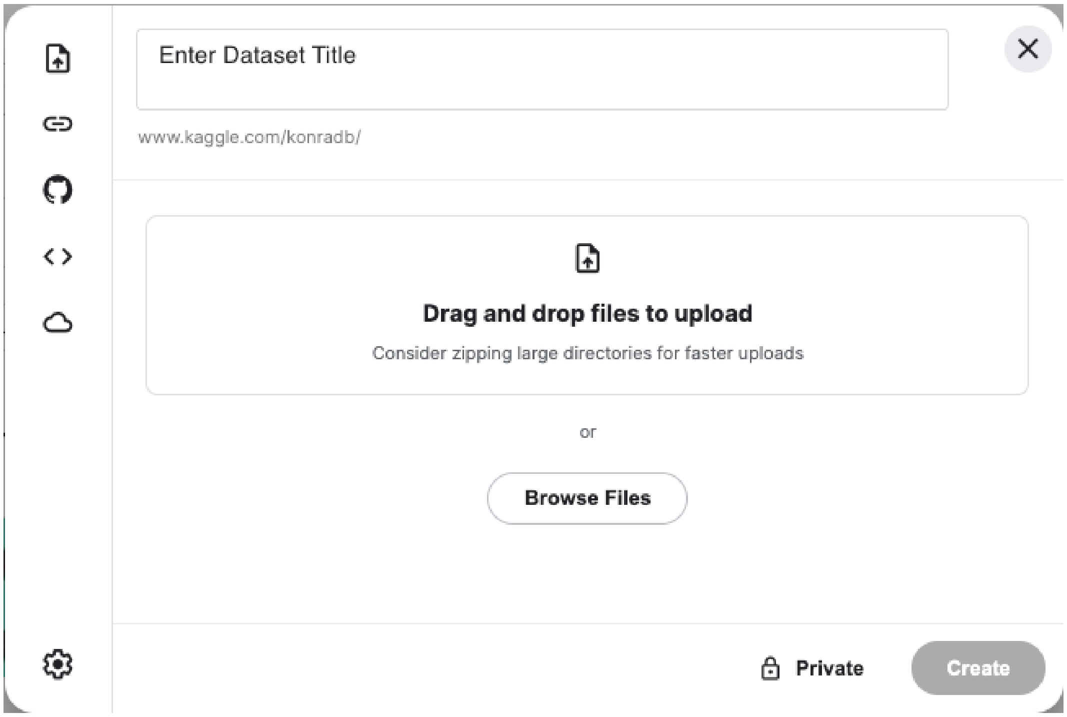
For the sake of this introduction, let us assume the kind of data you will be using in your project is not already there—so you need to create a new one. When you head to the menu with three lines on the left-hand side and click on **Data** you will be redirected to the dataset page:



Datasets

Explore, analyze, and share quality data. Learn more about data types, creating, and collaborating.

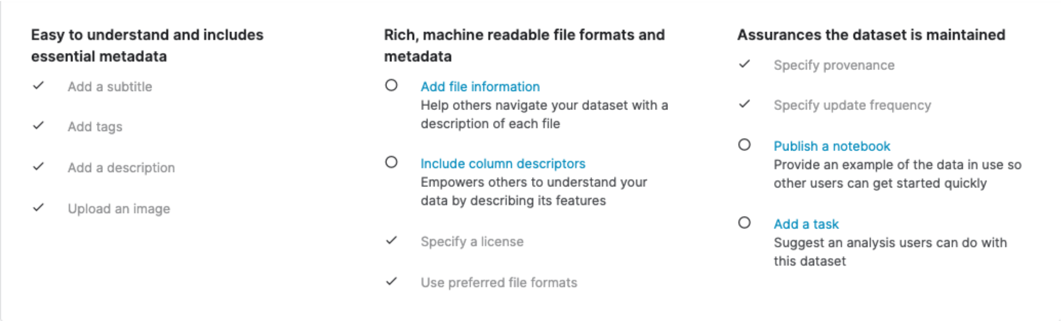
When you click on **New Dataset** you will be prompted for the basics: uploading the actual data and giving it a title:



Consider zipping large directories for faster uploads

Keep in mind that Kaggle is a popular platform, so numerous people upload their data there—including private (not publicly visible) ones —so try to think of a non-generic title.

Voila! Your first dataset is ready. You can then head to the **Data** tab:



Easy to understand and includes essential metadata

- ✓ Add a subtitle
- ✓ Add tags
- ✓ Add a description
- ✓ Upload an image

Rich, machine readable file formats and metadata

- ○ Add file information
  Help others navigate your dataset with a description of each file
- ○ Include column descriptors
  Empowers others to understand your data by describing its features
- ✓ Specify a license
- ✓ Use preferred file formats

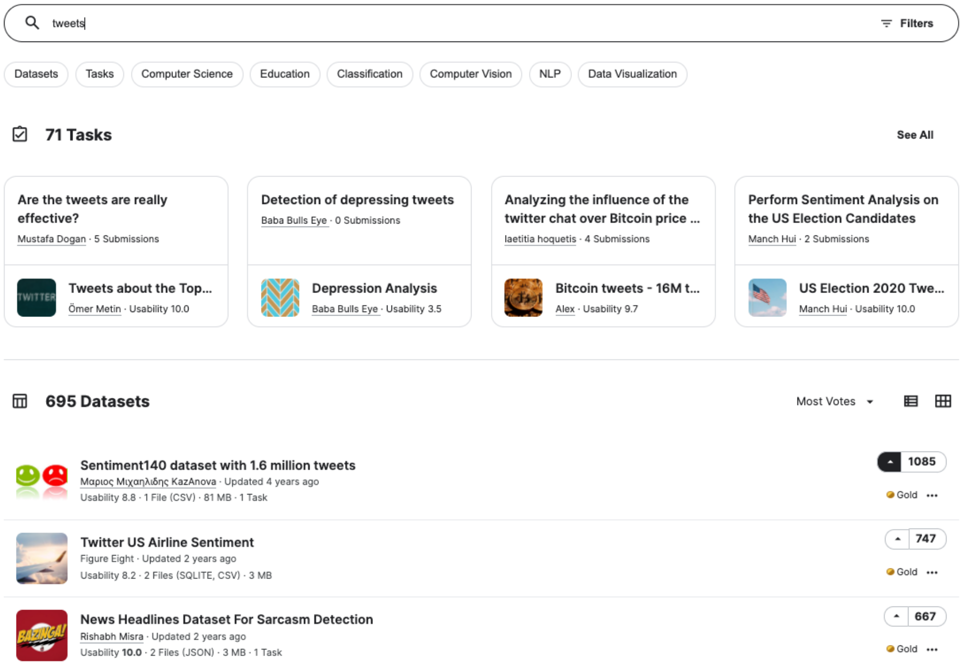
Assurances the dataset is maintained

- ✓ Specify provenance
- ✓ Specify update frequency
- ○ Publish a notebook
  Provide an example of the data in use so other users can get started quickly
- ○ Add a task
  Suggest an analysis users can do with this dataset

In principle you do not have to fill out all the fields—your newly created dataset is perfectly usable without them (and if it is a private one, you probably do not care—after all you know what is in it). However, the community etiquette would suggest filling the info for the ones you make public: the more you specify, the more usable the data will be to others (and measured by the usability score, displayed in the upper right corner).

Gathering the data

Apart from legal aspect (see the last section of this chapter), there is no real limit on the kind of content you can store in the datasets: tabular data, images, text—if you fit within the size requirements, you can store it. This includes data harvested from other sources: tweets by hashtag or topic are among the popular datasets at the time of writing:



71 Tasks

Are the tweets are really effective?
Mustafa Dogan · 5 Submissions
Tweets about the Top...
Ömer Metin · Usability 10.0

Detection of depressing tweets
Baba Bulls Eye · 0 Submissions
Depression Analysis
Baba Bulls Eye · Usability 3.5

Analyzing the influence of the twitter chat over Bitcoin price ...
laetitia hoquetis · 4 Submissions
Bitcoin tweets - 16M t...
Alex · Usability 9.7

Perform Sentiment Analysis on the US Election Candidates
Manch Hui · 2 Submissions
US Election 2020 Twe...
Manch Hui · Usability 10.0

695 Datasets

Sentiment140 dataset with 1.6 million tweets
Μαριος Μιχαηλιδης KazAnova · Updated 4 years ago
Usability 8.8 · 1 File (CSV) · 81 MB · 1 Task
1085
Gold

Twitter US Airline Sentiment
Figure Eight · Updated 2 years ago
Usability 8.2 · 2 Files (SQLITE, CSV) · 3 MB
747
Gold

News Headlines Dataset For Sarcasm Detection
Rishabh Misra · Updated 2 years ago
Usability 10.0 · 2 Files (JSON) · 3 MB · 1 Task
667
Gold

Discussion of the different frameworks for harvesting data from social media (Twitter, Reddit etc) is outside the scope of this book.

Using the Kaggle datasets outside of Kaggle

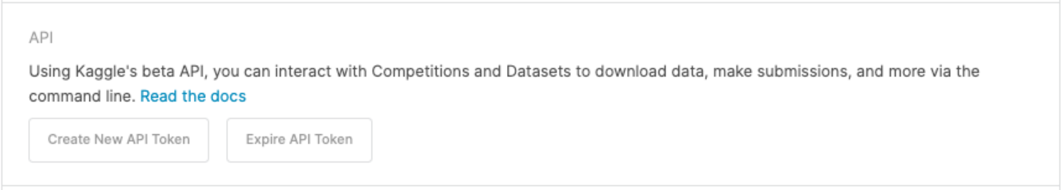
Kaggle kernels are free to use, but not without limits (more on that in *Chapter 4*)—and the first one you are likely to hit is the time limit of 8 hours. A popular alternative is to move to Google Colab a free Jupyter notebook environment that runs entirely in the cloud:

https://colab.research.google.com

But even once we move the computations there, we might still want to have access to the Kaggle datasets—so importing them into Colab is a rather handy feature.

The first thing we do—since you are reading this, we assume you already are registered on Kaggle—is head to the account page to generate the API token:

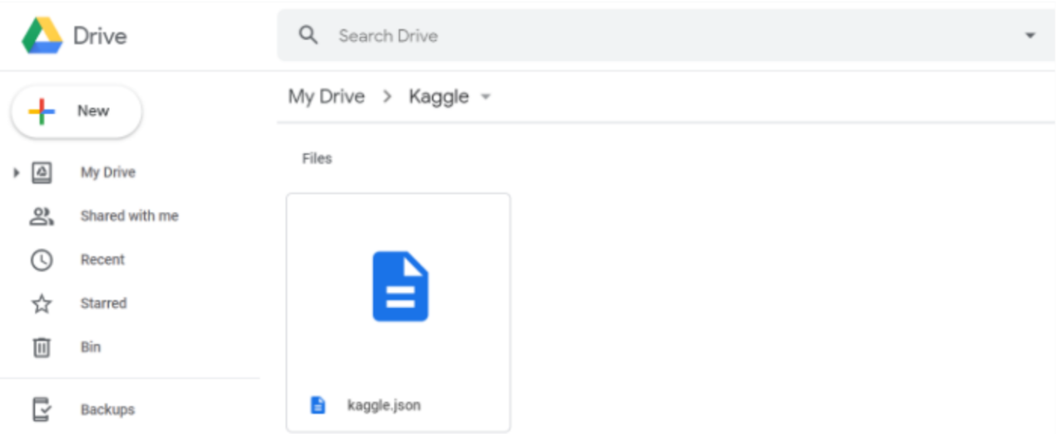
- Go to "your account" and click on **Create New API Token**
- A file named `kaggle.json` containing your username and token will be created



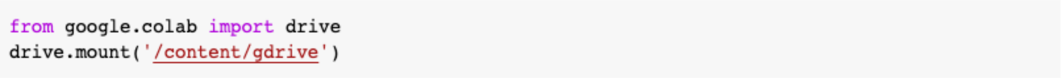
API

Using Kaggle's beta API, you can interact with Competitions and Datasets to download data, make submissions, and more via the command line. Read the docs

Next step is to create a folder named "`Kaggle`" in your drive and upload the `.json` there

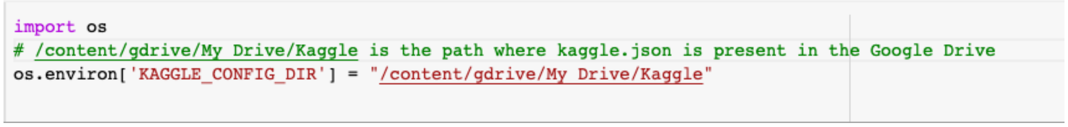


Once done, you need to create a new Colab notebook and mount your drive:

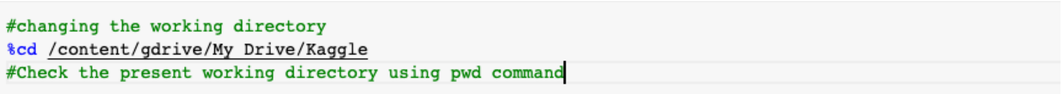


```
from google.colab import drive
drive.mount('/content/gdrive')
```

Get the authorization code from the URL prompt and provide in the empty box, then execute the following code to prove the path to the `.json` config:

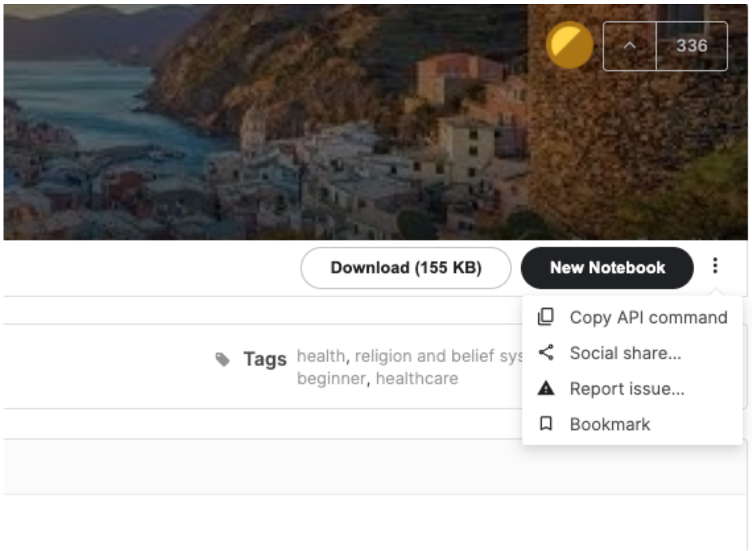


```
import os
# /content/gdrive/My Drive/Kaggle is the path where kaggle.json is present in the Google Drive
os.environ['KAGGLE_CONFIG_DIR'] = "/content/gdrive/My Drive/Kaggle"
```

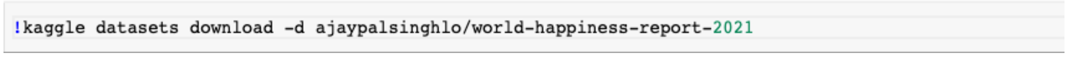


```
#changing the working directory
%cd /content/gdrive/My Drive/Kaggle
#Check the present working directory using pwd command
```

We can download the dataset now: begin by going to Kaggle and copying the API command:



Run the code:

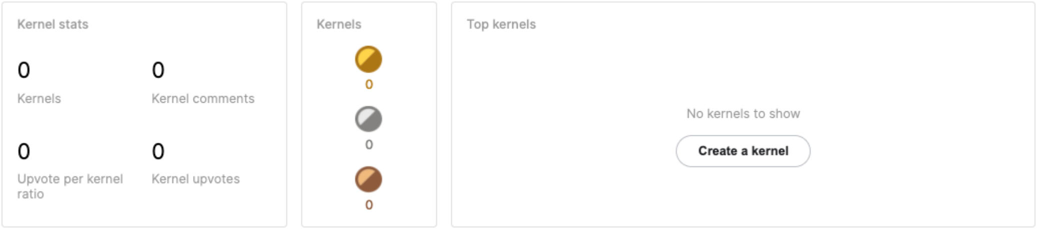


```
!kaggle datasets download -d ajaypalsinghlo/world-happiness-report-2021
```

The dataset will be downloaded as a `.zip` archive—unpack it and you are good to go.

Building around datasets

Once you have created a dataset, you probably want to use in your analysis. You can start a kernel using your dataset as a primary source: go to the **Activity** tab in the upper menu of your dataset page and scroll to this block:



Kernel stats

0 Kernels

0 Kernel comments

0 Upvote per kernel ratio

0 Kernel upvotes

Kernels

0

0

0

Top kernels

No kernels to show

Alternatively, you can start a conversation around the data by clicking on **Create a discussion**.

Legal caveats

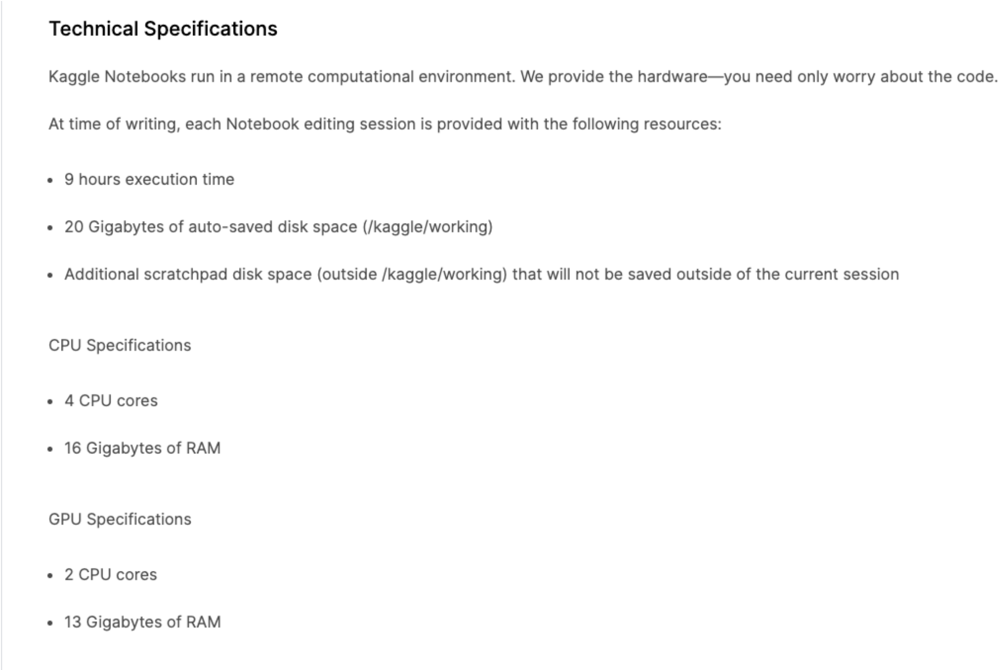
Just because you can put some data on Kaggle does not necessarily mean that you should—excellent example would be the “People of Tinder dataset”: in 2017, a developer used the Tinder API to scrape the website for semi-private profiles and uploaded the data on Kaggle. After the issue became known, Kaggle ended up taking the dataset down. You can read the full story here:

https://www.forbes.com/sites/janetwburns/2017/05/02/tinder-profiles-have-been-looted-again-this-time-for-teaching-ai-to-genderize-faces/?sh=1afb86b25454

In general, before you upload anything to Kaggle ask yourself two questions: is it legal (from a copyright standpoint—always check the licenses) and are there any risks associated with this dataset (privacy or otherwise).

Working and Learning with Kaggle Notebooks

Kaggle notebooks—*which until recently were called kernels, so please excuse me if I occasionally use those terms interchangeably*—are Jupyter notebooks in the browser that can run free of charge. This means you can execute your experiments from any device with an internet connection, although something bigger than a mobile phone is probably a good idea. The technical specification of the environment (as of the time of this writing) is given below:



Technical Specifications

Kaggle Notebooks run in a remote computational environment. We provide the hardware—you need only worry about the code.

At time of writing, each Notebook editing session is provided with the following resources:

- 9 hours execution time
- 20 Gigabytes of auto-saved disk space (/kaggle/working)
- Additional scratchpad disk space (outside /kaggle/working) that will not be saved outside of the current session

CPU Specifications

- 4 CPU cores
- 16 Gigabytes of RAM

GPU Specifications

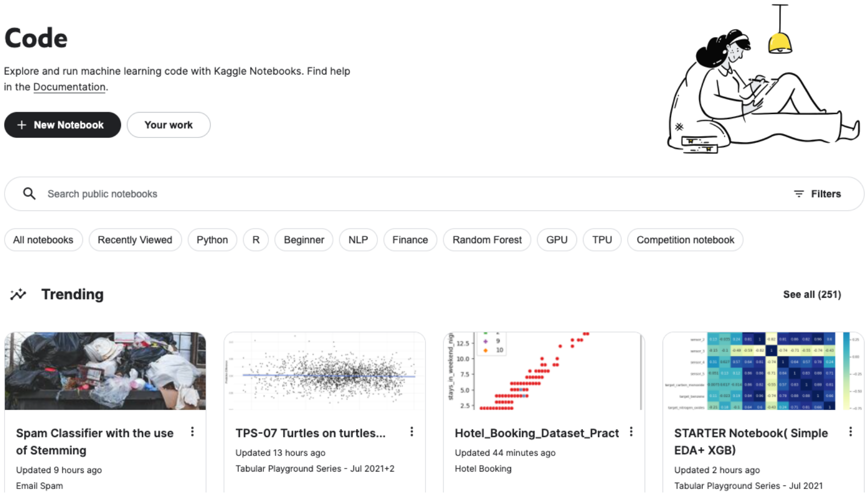
- 2 CPU cores
- 13 Gigabytes of RAM

Without further ado, let us jump into it. The first thing we do is figure out how to set up a notebook.

Setting up a kernel

There are two primary methods of creating a notebook: from the front page or from a dataset level.

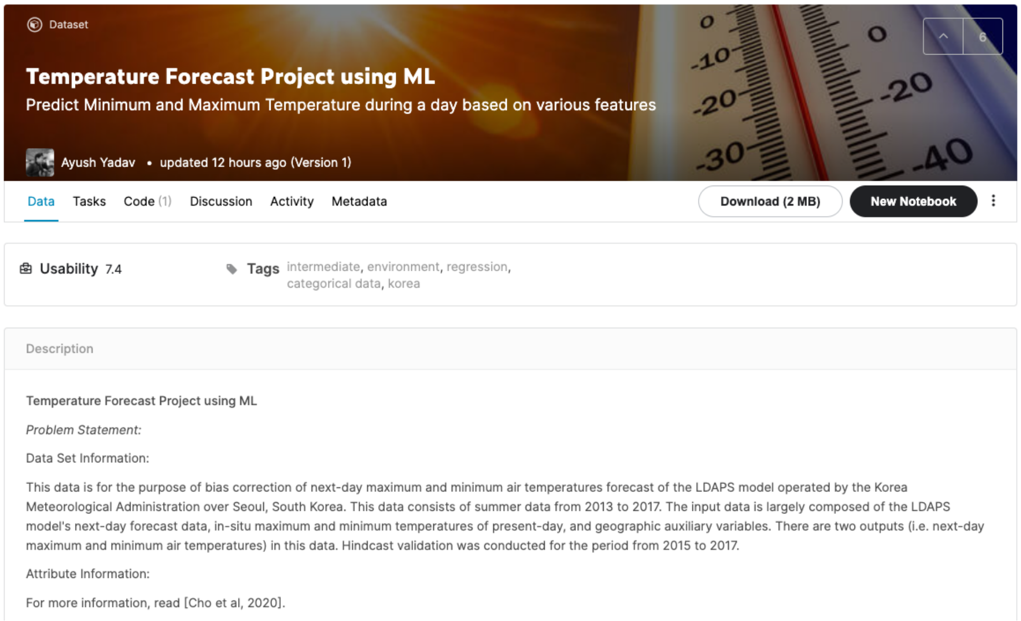
To proceed with the first method, go to the **Code** section of the menu on the left-hand side of the landing page at https://www.kaggle.com/ and press the **New Notebook** button. This is a preferred method if you are planning an experiment involving uploading your own dataset:



Code

Explore and run machine learning code with Kaggle Notebooks. Find help in the Documentation.

Alternatively, you can go to the page of the dataset you are interested in and click the **New Notebook** button there:



Temperature Forecast Project using ML

Predict Minimum and Maximum Temperature during a day based on various features

Ayush Yadav • updated 12 hours ago (Version 1)

Usability 7.4

Tags intermediate, environment, regression, categorical data, korea

Description

Temperature Forecast Project using ML

Problem Statement:

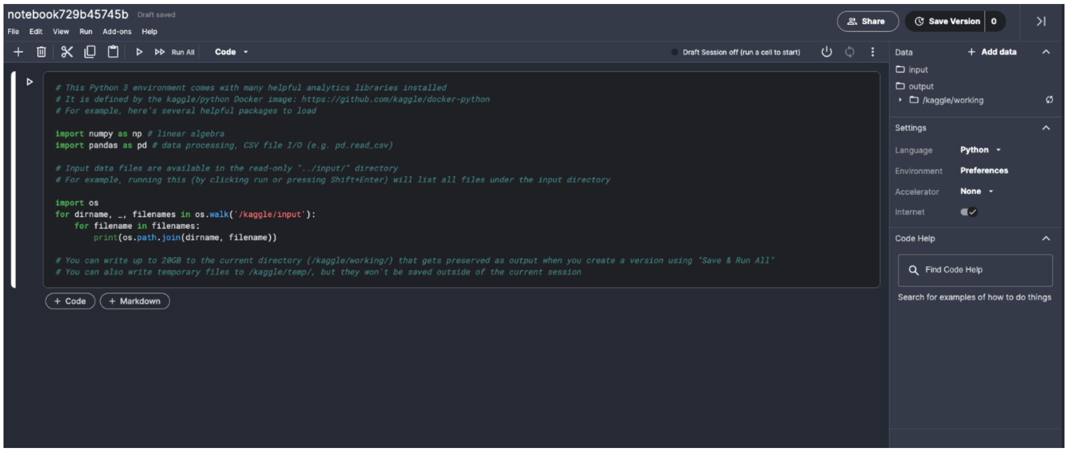
Data Set Information:

This data is for the purpose of bias correction of next-day maximum and minimum air temperatures forecast of the LDAPS model operated by the Korea Meteorological Administration over Seoul, South Korea. This data consists of summer data from 2013 to 2017. The input data is largely composed of the LDAPS model's next-day forecast data, in-situ maximum and minimum temperatures of present-day, and geographic auxiliary variables. There are two outputs (i.e. next-day maximum and minimum air temperatures) in this data. Hindcast validation was conducted for the period from 2015 to 2017.

Attribute Information:

For more information, read [Cho et al, 2020].

Whichever method you chose, after clicking **New Notebook** you will be taken to your notebook page:



```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

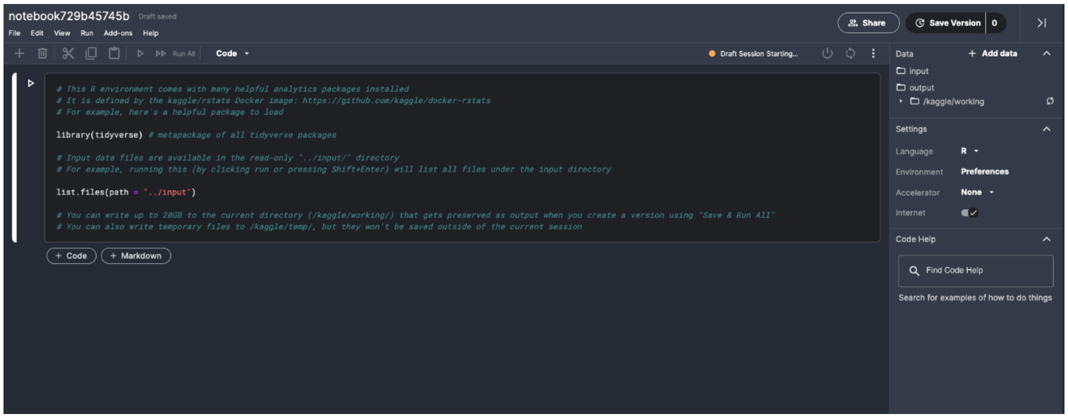
Language Python

Environment Preferences

Accelerator None

Internet ✓

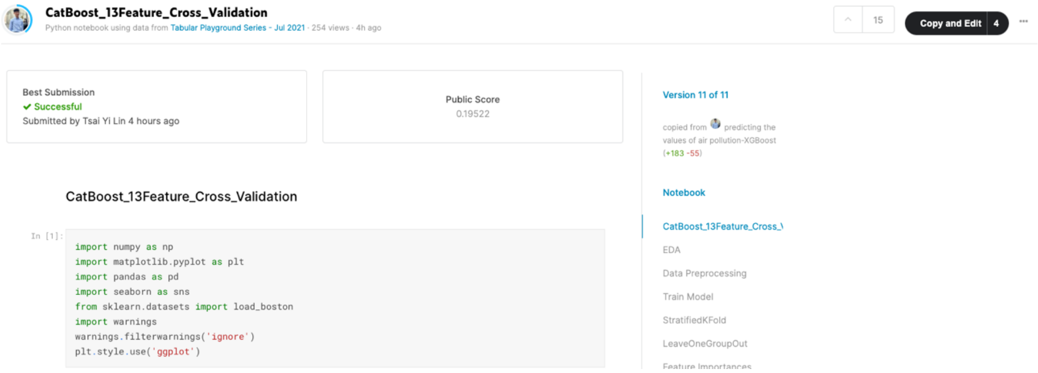
By default, a new notebook is initialized language set to **Python**—if you want to use R instead, click on the **Language** dropdown on the right-hand side and your notebook will switch to **R**:



```
# This R environment comes with many helpful analytics packages installed
# It is defined by the kaggle/rstats Docker image: https://github.com/kaggle/docker-rstats
# For example, here's a helpful package to load

library(tidyverse) # metapackage of all tidyverse packages

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

list.files(path = "../input")

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

Language R
Environment Preferences
Accelerator None
Internet

An important aspect of using notebooks: you can always take an existing one (created by somebody) and clone it to modify and adjust to your needs. This can be achieved by pressing the **Copy and Edit** button on the kernel page, although in Kaggle parlance, the process is referred to as **forking**:



CatBoost_13Feature_Cross_Validation

Python notebook using data from Tabular Playground Series - Jul 2021 · 254 views · 4h ago

Best Submission
✓ Successful
Submitted by Tsai Yi Lin 4 hours ago

Public Score
0.19522

copied from predicting the values of air pollution-XGBoost (+183 -55)

CatBoost_13Feature_Cross_Validation

```
In [1]:
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.datasets import load_boston
import warnings
warnings.filterwarnings('ignore')
plt.style.use('ggplot')
```

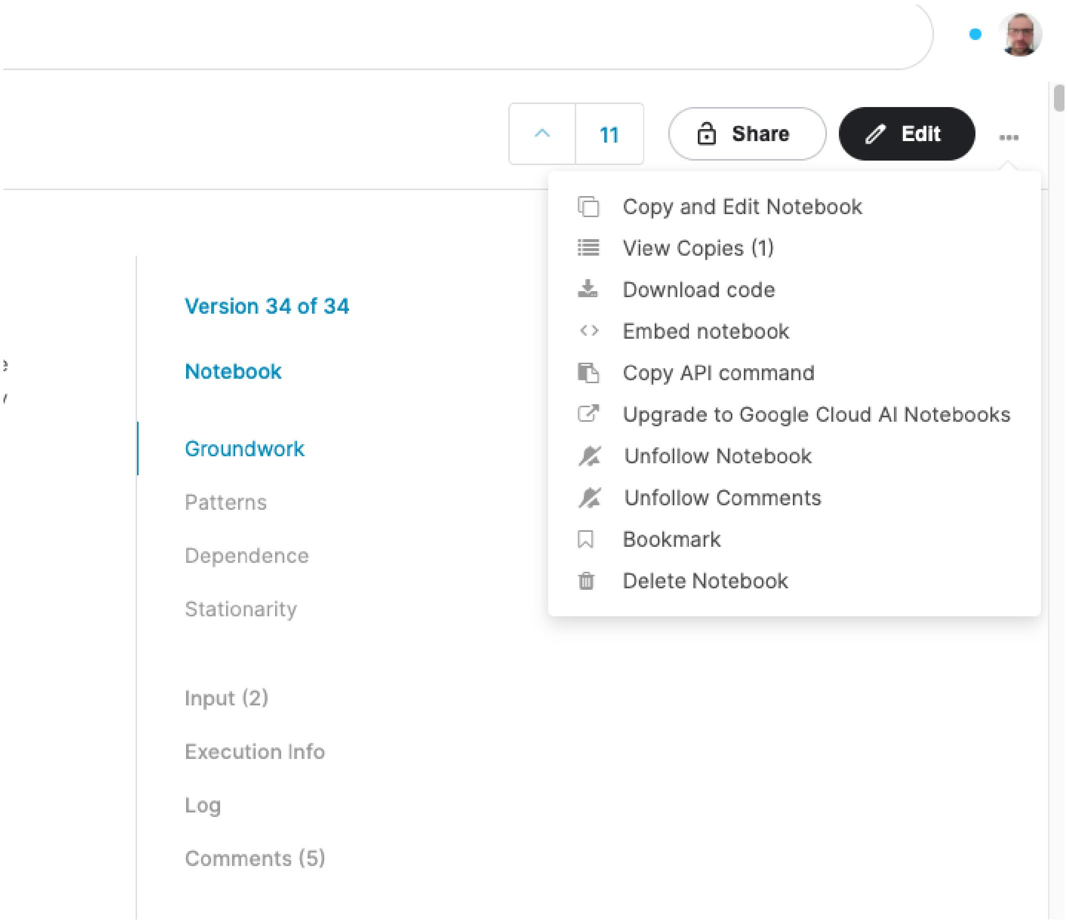
A note on etiquette: if you have participated in a Kaggle competition, you probably noticed that the leaderboard is flooded by forks of forks of well scoring notebooks. Nothing wrong with building on somebody else’s work—but if you do, remember to upvote the original author.

A notebook you create is private (i.e. only visible to you) by default. If you want to make it available to others, you can select

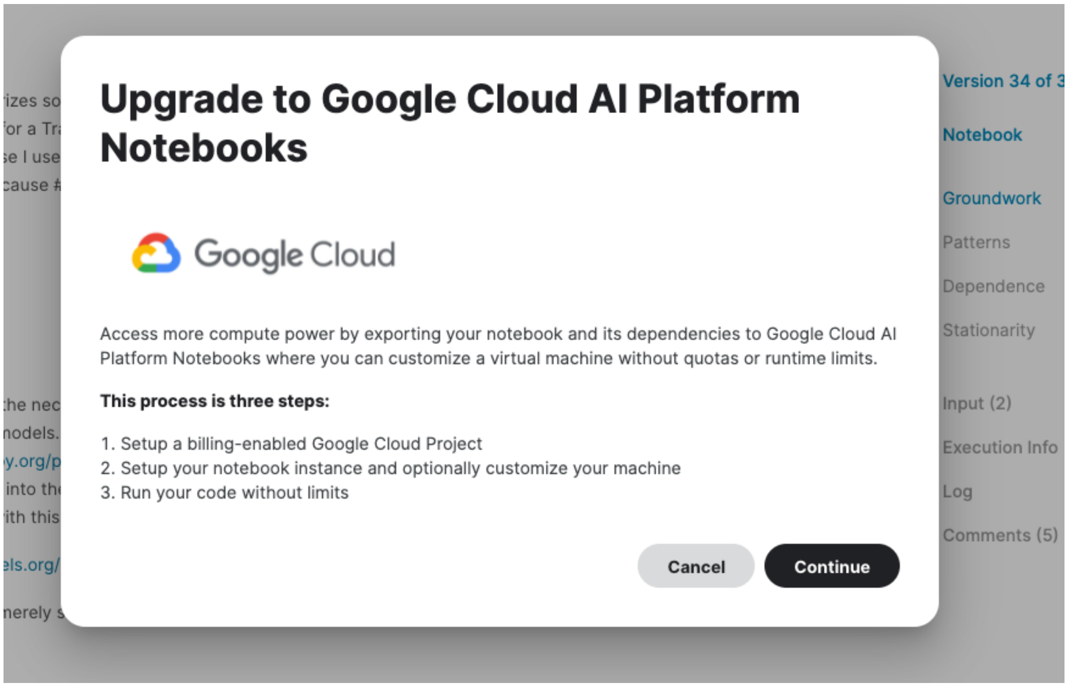
Upgrade to GCP

Sometimes the resources provided freely by Kaggle are not sufficient for the task you need, and you need to move to a beefier machine. You can setup the whole environment yourself—or you can stay within the framework of notebooks but swap the underlying machine. This is there Google Cloud AI Notebooks come in.

In order to migrate your notebook to the GCP environment, go to the sideline menu on the left-hand side and click on **Upgrade to Google Cloud AI Notebooks**:



You will be greeted by the prompt:



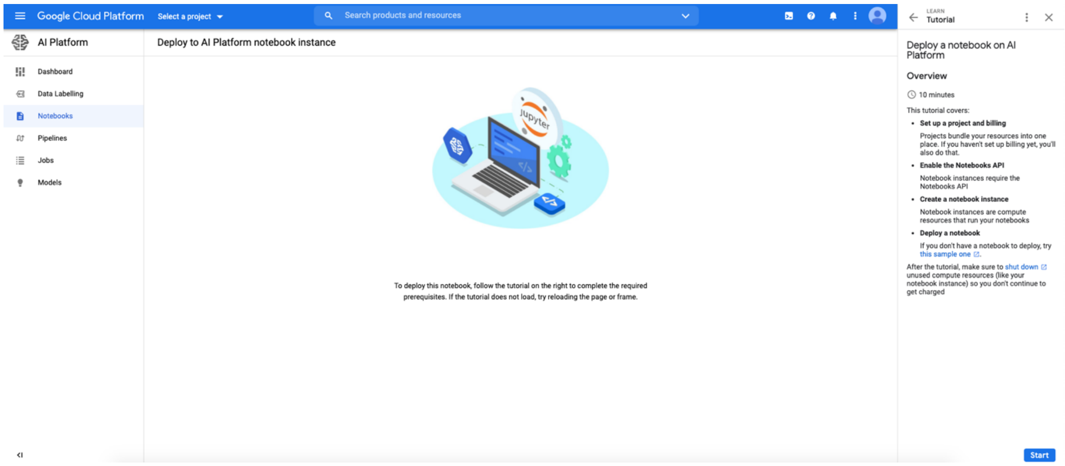
Upgrade to Google Cloud AI Platform Notebooks

Access more compute power by exporting your notebook and its dependencies to Google Cloud AI Platform Notebooks where you can customize a virtual machine without quotas or runtime limits.

This process is three steps:

1. Setup a billing-enabled Google Cloud Project
2. Setup your notebook instance and optionally customize your machine
3. Run your code without limits

After that, you will be redirected to the **Google Cloud Platform** console, where you need to configure your billing options—unlike Kaggle, GCP is not free. If it is your first time, you will need to complete a tutorial guiding you through the necessary steps:



Deploy to AI Platform notebook instance

To deploy this notebook, follow the tutorial on the right to complete the required prerequisites. If the tutorial does not load, try reloading the page or frame.

Deploy a notebook on AI Platform

Overview

10 minutes

This tutorial covers:

- **Set up a project and billing**
  Projects bundle your resources into one place. If you haven't set up billing yet, you'll also do that.
- **Enable the Notebooks API**
  Notebook instances require the Notebooks API
- **Create a notebook instance**
  Notebook instances are compute resources that run your notebooks
- **Deploy a notebook**
  If you don't have a notebook to deploy, try this sample one.

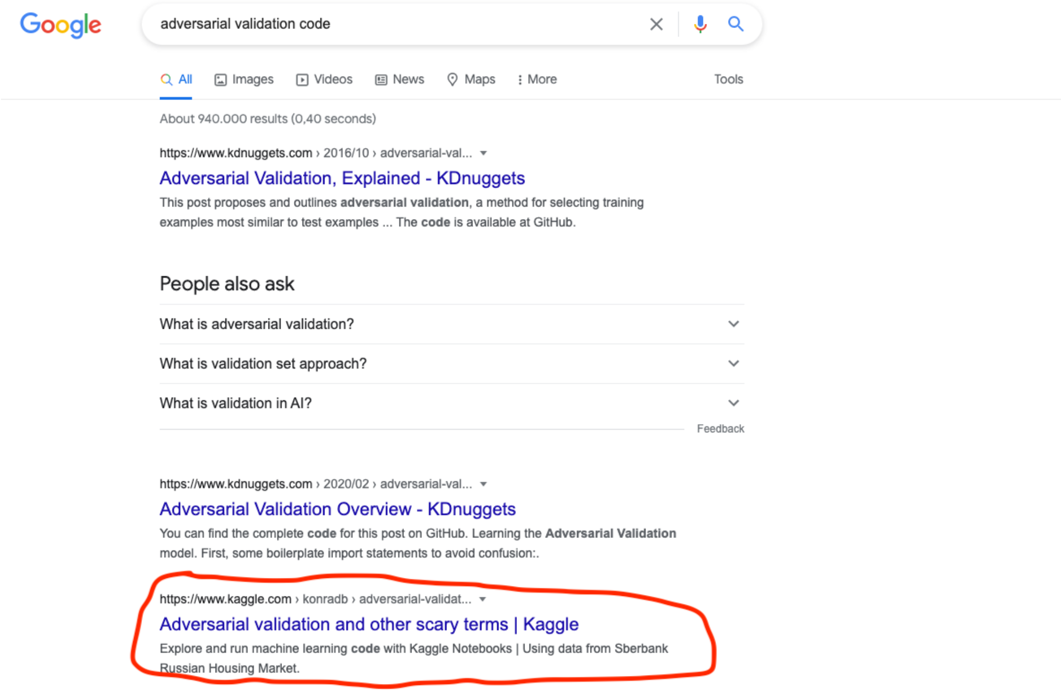
After the tutorial, make sure to shut down unused compute resources (like your notebook instance) so you don't continue to get charged

One step beyond

As mentioned earlier in this chapter, Kaggle notebooks are a fantastic tool for education and participating in competitions—but they also serve another extremely useful purpose, namely a component of a portfolio you can use to demonstrate your data science skills.

There are many potential criteria to consider when building your data science portfolio (branding, audience reach, enabling a pitch to your potential employer etc.) but none of them matter if nobody can find them. Because Kaggle is part of Google, the notebooks are indexed by the most popular search engine in the world—so if someone is looking for a topic related to your code, it will show up in their search results.

Below I show a “personal” example: a few years ago, I wrote a notebook for a competition—the problem I wanted to tackle was adversarial validation (for those unfamiliar with the topic: a fairly easy way to see if your training and test sets have a similar distribution is to build a binary classifier trained to tell them apart). When writing this chapter, I tried to give it a try and lo and behold, it shows up high in the search results (notice the fact that I did not mention Kaggle or any personal details like name in my query):



adversarial validation code

About 940.000 results (0,40 seconds)

https://www.kdnuggets.com › 2016/10 › adversarial-val...

Adversarial Validation, Explained - KDnuggets

This post proposes and outlines **adversarial validation**, a method for selecting training examples most similar to test examples ... The **code** is available at GitHub.

People also ask

What is adversarial validation?

What is validation set approach?

What is validation in AI?

https://www.kdnuggets.com › 2020/02 › adversarial-val...

Adversarial Validation Overview - KDnuggets

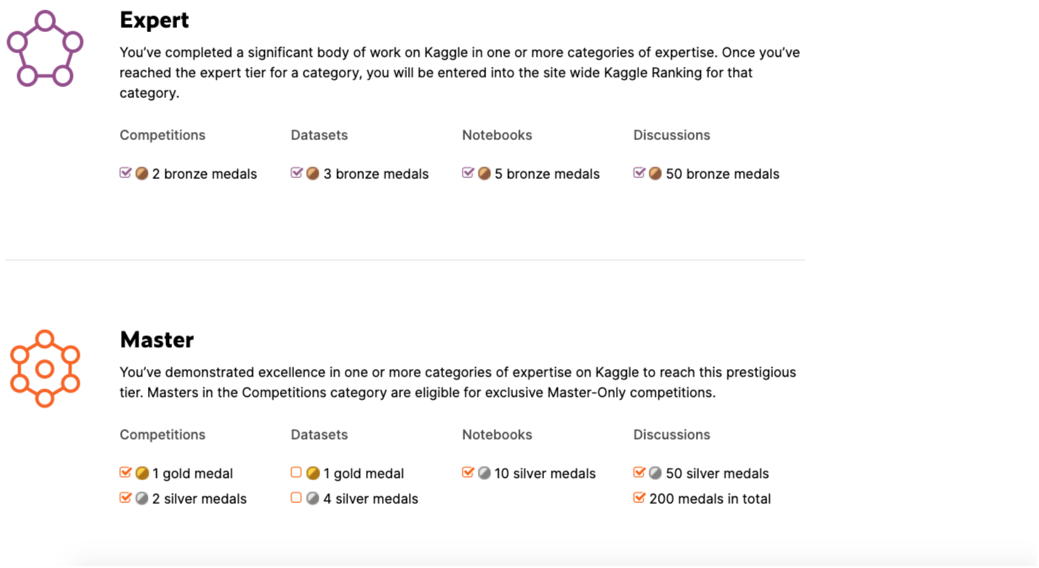
You can find the complete **code** for this post on GitHub. Learning the **Adversarial Validation** model. First, some boilerplate import statements to avoid confusion:.

https://www.kaggle.com › konradb › adversarial-validat...

Adversarial validation and other scary terms | Kaggle

Explore and run machine learning **code** with Kaggle Notebooks | Using data from Sberbank Russian Housing Market.

Moving on to other benefits of using notebooks to demonstrate your skillset: just like competitions, datasets and discussions, notebooks can be awarded votes/medals and thus position you in the progression system and ranking. You can stay away from the competitions track and become an expert/master/grandmaster purely by focusing on high quality code the community appreciates. The most up-to-date version of the progression requirements can be found at https://www.kaggle.com/progression, below we give a snapshot relevant to notebooks:



Expert

You've completed a significant body of work on Kaggle in one or more categories of expertise. Once you've reached the expert tier for a category, you will be entered into the site wide Kaggle Ranking for that category.

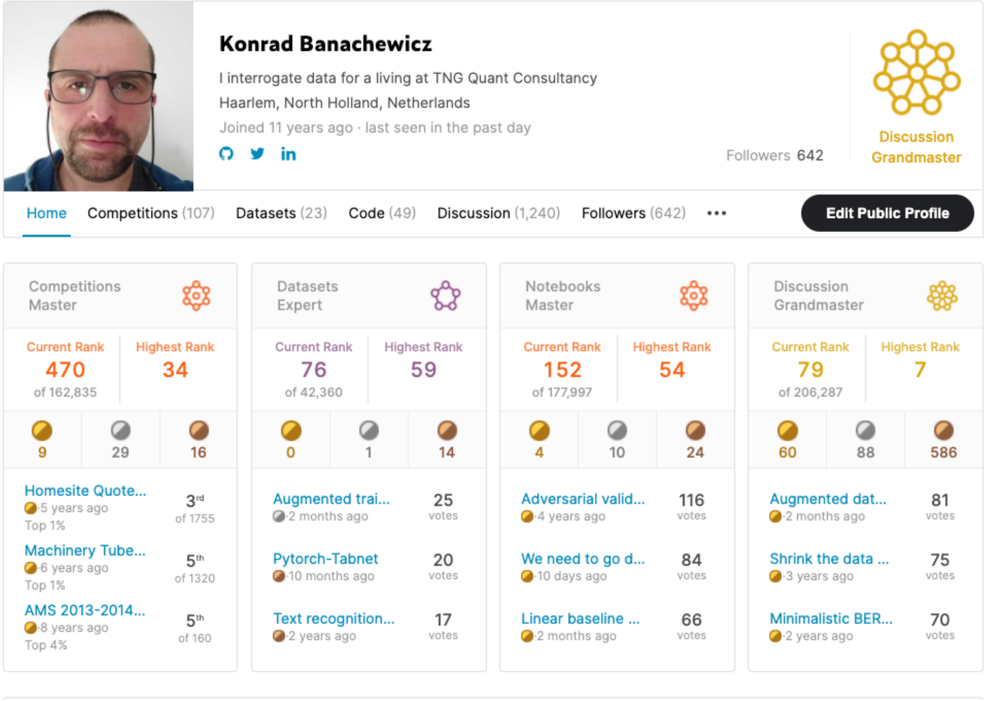
Competitions	Datasets	Notebooks	Discussions
☑ ● 2 bronze medals	☑ ● 3 bronze medals	☑ ● 5 bronze medals	☑ ● 50 bronze medals

Master

You've demonstrated excellence in one or more categories of expertise on Kaggle to reach this prestigious tier. Masters in the Competitions category are eligible for exclusive Master-Only competitions.

Competitions	Datasets	Notebooks	Discussions
☑ ● 1 gold medal	□ ● 1 gold medal	☑ ● 10 silver medals	☑ ● 50 silver medals
☑ ● 2 silver medals	□ ● 4 silver medals		☑ 200 medals in total

Your Kaggle profile comes with followers/following option and gives you a possibility to link other professional networks like LinkedIn or GitHub, so you can leverage the connection gained inside the community:



Konrad Banachewicz

I interrogate data for a living at TNG Quant Consultancy
Haarlem, North Holland, Netherlands
Joined 11 years ago · last seen in the past day

Followers 642

Discussion Grandmaster

Competitions Master	Datasets Expert	Notebooks Master	Discussion Grandmaster
Current Rank 470 of 162,835	Current Rank 76 of 42,360	Current Rank 152 of 177,997	Current Rank 79 of 206,287
Highest Rank 34	Highest Rank 59	Highest Rank 54	Highest Rank 7
9 · 29 · 16	0 · 1 · 14	4 · 10 · 24	60 · 88 · 586
Homesite Quote... · 5 years ago · Top 1% · 3rd of 1755	Augmented trai... · 2 months ago · 25 votes	Adversarial valid... · 4 years ago · 116 votes	Augmented dat... · 2 months ago · 81 votes
Machinery Tube... · 6 years ago · Top 1% · 5th of 1320	Pytorch-Tabnet · 10 months ago · 20 votes	We need to go d... · 10 days ago · 84 votes	Shrink the data ... · 3 years ago · 75 votes
AMS 2013-2014... · 8 years ago · Top 4% · 5th of 160	Text recognition... · 2 years ago · 17 votes	Linear baseline ... · 2 months ago · 66 votes	Minimalistic BER... · 2 years ago · 70 votes

In this day and age, it is easy to be skeptical about claims of "community building"—but in the case of Kaggle, it happens to actually be true. Their brand recognition in the data science universe is second to none, both among practitioners and among recruiters who actually do their homework. In practice, this means that a (decent enough) Kaggle profile can get you through the door already—which, as we all know, is frequently the hardest step.

Kaggle courses

A great many things about Kaggle are about acquiring knowledge be it the things you learn in a competition, datasets you manage to find in the ever-growing repository or demonstration of a hitherto unknown model class, there is always something new to find out. The newest addition to that collection are the courses gathered under the *Kaggle Learn* label: https://www.kaggle.com/learn. Those are micro-courses marketed by Kaggle as "the single fastest way to gain the skills you'll need to do independent data science projects", the core unifying theme being a crash course introduction across a variety of topics. Each course is divided into small chapters, followed by coding practice questions.

Below, we provide a brief summary of their content:

- **Python**: https://www.kaggle.com/learn/python You will learn the basics of functions, Boolean variables, loops, lists and dictionaries.
- **Intro to ML / Intermediate ML**: https://www.kaggle.com/learn/intro-to-machine-learning Those two courses are best viewed as a two-parter: the first one introduces different classes of models used in machine learning, followed by discussion of topics common to different models like under/overfitting or model validation. The second one goes deeper into feature engineering, dealing with missing values and handling categorical variables.
- **Pandas**: https://www.kaggle.com/learn/pandas: this course provides a crash introduction to one of the most fundamental tools used in modern data science. You first learn how to create / read / write data, moving on to data cleaning (indexing, selecting, combining, grouping etc).
- **Data visualization**: https://www.kaggle.com/learn/data-visualization Everybody knows a picture can be worth a thousand words – if you want to learn how to create such images summarizing your data science results, this course is for you. You will walk away know how to handle everything from line charts to heatmaps and scatterplots.
- **Feature engineering**: https://www.kaggle.com/learn/feature-engineering This short course demonstrates the basic ideas around encoding categorical data, general feature generation and selection.
- **Data cleaning**: https://www.kaggle.com/learn/data-cleaning Another short course, which helps address one of the most glaring omissions in the academic curriculum: making students realize how messy real-life data is.

- **Intro to SQL / Advanced SQL** https://www.kaggle.com/learn/intro-to-sql In this tandem of courses, you will learn to extract data using SQL. Starting with basic SELECT variations, you will go through GROUP BY, HAVING, all the way to JOINs/UNIONs and explore analytic functions and nested data.
- **Geospatial Analysis:** https://www.kaggle.com/learn/geospatial-analysis This course will teach you to create your first map using GeoPandas and introduce ways to create interactive and choropleth maps. Basics of proximity analysis are introduced as well.
- **Intro to Deep Learning** https://www.kaggle.com/learn/intro-to-deep-learning This course offers a crash introduction into what is arguably the most important methodology in modern deep learning. Using structured data, you will familiarize yourself with fundamental concepts like gradient descent, batch normalization and apply this knowledge to vintage problem of binary classification.
- **Computer Vision** https://www.kaggle.com/learn/computer-vision /**NLP** https://www.kaggle.com/learn/natural-language-processing are two crash courses introducing the most important domains where deep learning has been successfully applied, producing impressive state-of-the-art results. Crucial topics of transfer learning and data augmentation are introduced, giving you the tools to hit the ground running.
- **Game AI** https://www.kaggle.com/learn/intro-to-game-ai-and-reinforcement-learning This course is a great wrap-up of the tech-focused part of the curriculum introduced by Kaggle in the learning modules. You will write a game-playing agent, tinker with its performance and use the minimax algorithm.
- **Machine Learning Explainability** https://www.kaggle.com/learn/machine-learning-explainability Building models is fun, but in the real world not everybody is a data scientist, so you might find yourself in a position when you need to explain what you have done to others. This is where the mini course on model explainability comes in: you

will learn to assess how relevant your features are with three different methods: permutation importance, SHAP and partial dependence plots.

- **AI Ethics** https://www.kaggle.com/learn/intro-to-ai-ethics This last course is a very interesting addition to the proposition: it discusses the practical tools to guide the moral design of AI systems. You will learn how to identify the bias in AI models, examine the concept of AI fairness and find out how to increase transparency by communicating ML models information.

Apart from the original content created by Kaggle, there are multiple other learning opportunities available on the platform using kernels. A prominent example worth mentioning is the extremely popular fast.ai course: https://www.kaggle.com/general/63077

In this chapter, we have discussed Kaggle kernels: a multi-purpose, open coding environment, which can be used for education, experimentation as well promoting your data science project portfolio.