Twitter Thread by Santiago

Santiago
@svpino

Beginners guide to start with machine learning.

What you need:

1. A programming language
2. A place to write and run code
3. A way to deal with data
4. A way to visualize results
5. A cool algorithm
6. An introductory course
7. A project to solve

Here are my recommendations.

Most people recommend Python to start.

I do too.

You can do machine learning with many different languages, but today, Python is the best option.

My advice: Get comfortable writing code before looking into machine learning.

You don't have to be a great developer to start.

But if you are, it helps.

Everything you know about building software is helpful to build machine learning systems.

Good developers have the odds in their favor.

"How much Python do I need to know before starting?"

If you ask, you aren't ready yet.

Ideal scenario: You should be comfortable building software. Most people get here within a year.

Let's move on to the 2nd item from the list: A place to write and run your code.

Get familiar with Google Colab.

- It's free.
- Requires no setup to start.
- It's available from anywhere.

Alternative: Kaggle.

Why not your favorite IDE running on your computer?

That also works, but I'd recommend you get familiar with notebooks from day 1 (@code supports Jupyter notebooks!)

If/when you need a GPU, Google Colab has them for free.

3rd item: You need a way to load and manipulate data.

Pandas is Python's most popular library to do this.

You can go through this tutorial in about 4 hours: https://t.co/RMJJSAVPwT.

Pandas

Solve short hands-on challenges to perfect your data manipulation skills.

Lessons

Tutorial Exercise

1 **Creating, Reading and Writing**
You can't work with data if you can't read it. Get started here.

2 **Indexing, Selecting & Assigning**
Pro data scientists do this dozens of times a day. You can, too!

3 **Summary Functions and Maps**
Extract insights from your data.

4 **Grouping and Sorting**
Scale up your level of insight. The more complex the dataset, the more this matters

5 **Data Types and Missing Values**
Deal with the most common progress-blocking problems

6 **Renaming and Combining**
Data comes in from many sources. Help it all make sense together

Why the big deal with a library to handle data?

Most of the work is just that.

This sounds boring, but I promise it isn't. I can also tell you that it's one of the places where you'll get to show off your creativity.

Visualizing results is number 4 in the list.

Many people skip this step. That's a mistake.

A couple of popular options for you:

• Matplotlib
• Seaborn

Communication is one of the most powerful traits you could build. These libraries will help you do that.

Another 4-hour tutorial: https://t.co/V0ya8kLeCW

This will give you everything you need to start with Seaborn.

Data Visualization

Make great data visualizations. A great way to see the power of coding!

Lessons	Tutorial	Exercise
1 **Hello, Seaborn** Your first introduction to coding for data visualization		
2 **Line Charts** Visualize trends over time		
3 **Bar Charts and Heatmaps** Use color or length to compare categories in a dataset		
4 **Scatter Plots** Leverage the coordinate plane to explore relationships between variables		
5 **Distributions** Create histograms and density plots		
6 **Choosing Plot Types and Custom Styles** Customize your charts and make them look snazzy		
7 **Final Project** Practice for real-world application		
8 **Creating Your Own Notebooks** How to put your new skills to use for your next personal or work project		

At this point, you should be ready to start with specific machine learning content.

Many people go right away and start a course.

Here is a different way: Learn about one algorithm that could solve a problem for you.

This will motivate you to dive deeper.

As a developer, you already have experience learning new things.

- You find a problem.
- You look for a solution.
- You learn about it.
- You implement it.

I want you to try the same here.

Algorithm recommendation: Learn about Decision Trees to start.

Listen up: you don't need to go and become an expert on Decision Trees.

At this point:

- You don't need to worry about the math.
- You don't need to understand the full theory.

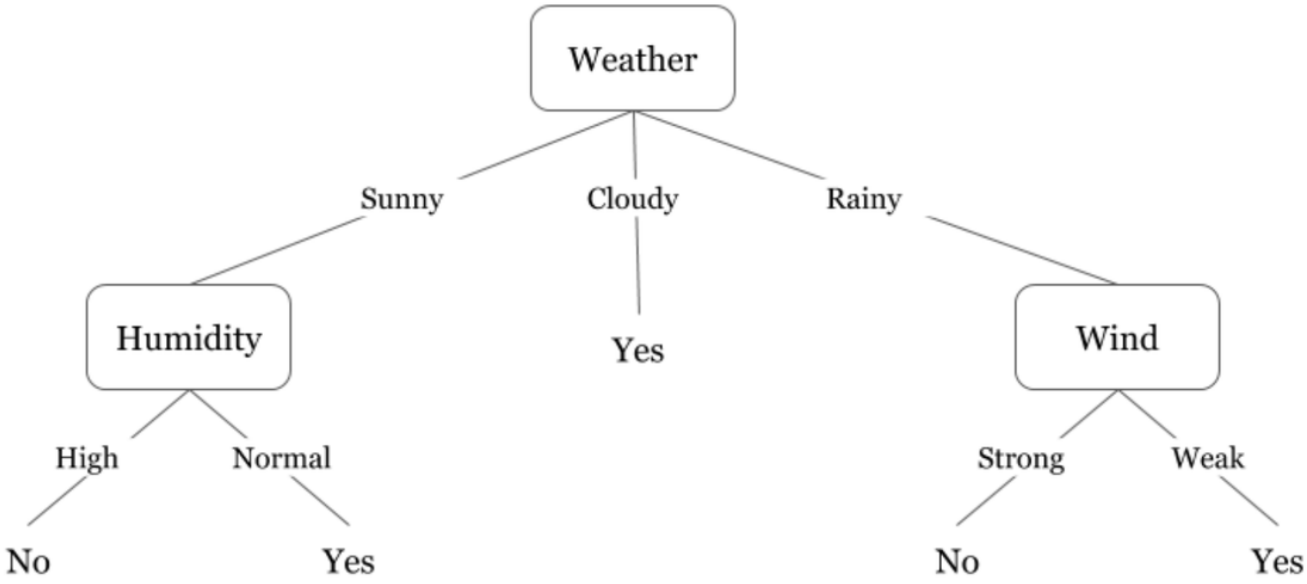
All of that can come later.

For now: How can you use Decision Trees? How are they helpful?

A couple of recommendations to get into Decision Trees:

- A tutorial with a lot of code: https://t.co/xz1yUaDxF6

- A video that builds a Decision Tree from scratch: https://t.co/tKtUpO1K3l



It's time for a machine learning introductory course.

(If you looked into Decision Trees already, great! This course will be easy.)

Starting from scratch, in 3 hours, you can go through this: https://t.co/qQXBcdvnsj.

Intro to Machine Learning

Learn the core ideas in machine learning, and build your first models.

Lessons

		Tutorial	Exercise
1	**How Models Work** The first step if you're new to machine learning		
2	**Basic Data Exploration** Load and understand your data		
3	**Your First Machine Learning Model** Building your first model. Hurray!		
4	**Model Validation** Measure the performance of your model ? so you can test and compare alternatives		
5	**Underfitting and Overfitting** Fine-tune your model for better performance.		
6	**Random Forests** Using a more sophisticated machine learning algorithm.		
7	**Machine Learning Competitions** Enter the world of machine learning competitions to keep improving and see your progress		

Bonus Lessons

		Tutorial	Exercise
☆	**Intro to AutoML** Learn how to use automated machine learning (AutoML) to accelerate your work.		

This course puts together everything we just discussed.

It even takes you through a simple problem and helps you solve it!

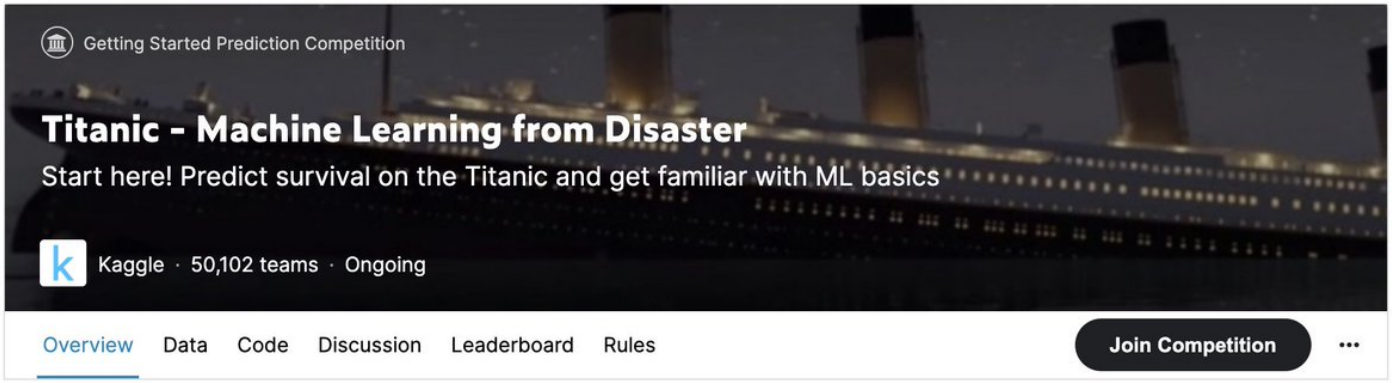
Good news: The course focuses on building and doesn't worry too much about math or theory.

(These are important, but not now.)

Final item from the list: You need a project.

One of the best problems to start: "Titanic - Machine Learning from Disaster."

You can find it here: https://t.co/eQzuGeePe2.



Optionally, you can take a look at this tutorial on how to solve the Titanic challenge:

https://t.co/DTA0B3GncE

A step-by-step guide that will help you get your first problem done!

Logging into Kaggle for the first time can be daunting. Our competitions often have large cash prizes, public leaderboards, and involve complex data. Nevertheless, we really think all data scientists can rapidly learn from machine learning competitions and meaningfully contribute to our community. To give you a clear understanding of how our platform works and a mental model of the type of learning you could do on Kaggle, we've created a Getting Started tutorial for the Titanic competition. It walks you through the initial steps required to get your first decent submission on the leaderboard. By the end of the tutorial, you'll also have a solid understanding of how to use Kaggle's online coding environment, where you'll have trained your own machine learning model.

So if this is your first time entering a Kaggle competition, regardless of whether you:

- have experience with handling large datasets,
- haven't done much coding,
- are newer to data science, or
- are relatively experienced (but are just unfamiliar with Kaggle's platform),

you're in the right place!

Let's recap:

1. You need experience with Python
2. Learn Google Colab
3. Pandas for data
4. Seaborn for visualizations
5. Decision Trees is a good start
6. Finish "Intro to Machine Learning."
7. Solve the Titanic challenge