# Legal Data Analysis

## Course Overview

**This course will introduce students to basic and common data analysis methods and tools, as applied in particular to legal data and legal knowledge.**

This is a course for beginners: no pre-requisite is required, as we will start from the ground up. Students will only need a computer and an internet connection. Students who already know how to code in Python need not apply.

The course is aimed at all students interested in data analytics, since the skills learned on the basis of legal datasets are transferable to *any* kind of data. The lessons will elaborate on why mastering these skills are critical for their future career in the digital economy.

The course should be of particular interest for students who want to work in tech start-ups (especially legal techs), law firms, academia, or in devising public policy. Students will also learn how data can be used in a variety of business applications.

At the end of the course, students should be able to:

1. Identify and constitute a dataset;
2. Develop measures, metrics, and categories to explore and explain the data; and
3. Identify technological “needs” and devise a proof of concept.

The course is designed as a gradual introduction to Python and relevant methods of legal data analysis, within a 16- or 24-hour schedule. The first 12/16 hours are meant to be a sufficient introduction for the purpose of the Final Presentation; the further hours are reserved for more specialised methods and uses.

Each course will be built around three elements:

* Some opening considerations of a theoretical nature;
* Practical teaching, based on python scripts that students will run on their computer. Students will be invited to fiddle with the pre-inputted code and discover by themselves its limits; and
* A number of exercises based on the material covered.

Given the intensive schedule, the course can accommodate only a limited number of students.

## Suggested Reading

* Data Science For Lawyers (available at <https://www.datascienceforlawyers.org/>);
* The Programming Historian (available at <https://programminghistorian.org/>);
* Al Sweigart, *Automate the Boring Stuff with Python: Practical Programming for Total Beginners* (No Starch Press 2015);
* Folgert Karsdorp, Mike Kestemont, Allen Riddell, *Humanities Data Analysis: Case Studies with Python* (Princeton University Press 2021);
* *AT&T Archives: The UNIX Operating System*, 1982, video available on [Youtube](https://www.youtube.com/watch?v=tc4ROCJYbm0&t=681s&ab_channel=AT%26TTechChannel); and
* Damien Charlotin & Wolfgang Alschner, ‘Data Mining, Text Analytics, and Investor-State Arbitration’ *forthcoming in* Pietro Ortolani et al. (eds.), *International Arbitration and Technology* (Wolters Kluwer 2022).
* UC Davis Data Lab, *Adventures in Data Science*, available [here](https://ucdavisdatalab.github.io/adventures_in_data_science/introduction-to-version-control.html).
* Paul Ford, ‘What is Code’ (*Businessweek*, 11 June 2015).

# Course Detail

For each course, students will be provided with material, available on a Colab Notebook.

## Introduction to Python

### Course 1 – Basics

**Theory :** We’ll start by introducing programming languages in general, and what distinguishes them. Students will learn what makes Python a good language for most tasks, but also how to find more resources to go further.

**Practice :** After introducing the basic commands and uses of Colab and the main vocabulary, we’ll review: (i) Basic Operations; (ii) Variables; (iii) Functions; (iv) Syntax (if/else, loops); and (v) List Comprehension.

### Course 2 – Files and Data

**Theory :** The challenges facing lawyers – and professionals in general – arising from the automation of their tasks and the increasing use of machines and robots.

**Practice :** Introduction to the following concepts: (i) Manipulating Files; (ii) Encoding; (iii) Data Storage; and (iv) Regexes.

### Course 3 – Dataframes

**Theory :** We’ll discuss version control methods and tools such as Git and Github.

**Practice :** The practical part of the course will introduce students to pandas and dataframes in general. Students will be invited to perform their first data analyses over a legal dataset, mapping the numbers of CADA requests over time and space. We’ll also check if the number and type of decisions is altered as local and presidential elections loom.

### Course 4 – Basic NLP

**Theory :** History and theory of Natural Language Processing (NLP), and its applications in day-to-day applications.

**Practice :** Introduction to NLP tools and methods. Exercises will focus on a dataset of contracts, as students will be asked to identify verbs, subjects, and different kind of provisions on that basis.

## Collect and Scrap Data

### Course 5 – Formatted Data

**Theory :** We’ll start by discussing xml, html, and formatted data in general: its uses and logic in the context of sharing and analysing data. This will segue into a discussion of the Open Data movement, including in the context of French law.

**Practice :** Students will be introduced to the xPath syntax, and try their hand on a number of exercises based on XML docs released by the Conseil d’Etat’s Open Data Initiative.

### Course 6 – Scraping

**Theory :** Shortconsiderations as to the ethics of scraping, and possible alternatives. We’ll also discuss the distinction between static and dynamic websites.

**Practice :** Starting withrequests, we’ll scrap the (static) website of the Constitutional Council. Switching to Selenium, we’ll scrap judgments from the Conseil d’Etat’s (dynamic) Ariane database.

Time allowing,we’ll introduce APIs, with examples (e.g., Google Doc; JudiLibre).

## Organise, Clean, and Exploit a Dataset

### Course 7 – Clean and Exploit a Dataset

**Theory :** Some considerations as to the turn to empirical study in legal practice & scholarship.

**Practice :** We’ll go back to pandas and perform more sophisticated operations over a dataset of decisions from the Cour de cassation. In this context, we’ll introduce basic plotting and statistical tools to investigate this jurisprudence.

### Course 8 – Advanced NLP

**Theory :** Short discussion as to the use of NLP and other methods to boost litigation and arbitration skills.

**Practice :** We’ll address similarity measures, vector-models, topic models. We’ll apply these tools to the information collected from *LégiFrance*, and identify the evolution of French law over the years.

## Advanced Topics

### Course 9 – Networks

**Theory :** Introductionto network science and related methods and measures, and relevant classic legal scholarship. We’ll also discuss uses of network analysis beyond case citation networks, such as cliques detection, recommendation algorithms, probability designs, contract/boilerplate analysis, visualisations, etc.

**Practice :** From a dataset of international court decisions (European Court of Human Rights or other international court or tribunal), we’ll use regexes methods to create a dataset of nodes and edges.

We’ll then study the resulting case citation network to identify sub-areas of jurisprudence, tally a relevant “study list” of important precedents, and compare this approach with classic textbook.

### Course 10 – Machine Learning

**Theory :** What’s behind the (overbroad) term “Artificial Intelligence”, how to think about associated risk (“existential” or more mundane), what’s the issue with so-called “algorithmic bias”, etc. Given time we’ll discuss ethics, the singularity, etc.

**Practice :** Introduction to supervised and unsupervised Machine Learning, with examples. Will discuss basic prediction models, random forests, fitting, etc.

Based on the dataset of Cour de cassation judgments, we’ll feed a basic machine learning model with text data as well as outcome metadata (rejet du pourvoi/cassation), and see if we can create a model that scores better than chance (spoiler: we will).

Finally, we’ll see how to review this model and evaluate it in terms of accuracy, robustness, etc.

### Course 11 – Varia (Text Generation, APIs, etc.)

This last section will be used to review the entire course, deal with any code or script that had to be passed over, and answer any lingering questions.

We’ll spend most of the time, however, discussing generative AI, and Large Language Models (LLMs) in particular: their training, limitations, and how they’ll change the legal profession. We’ll also touch on prompt engineering.

### Course 12 – Presentations

Final presentations by the students.

# Evaluation

* A presentation along the following lines:
  + Identify a dataset and a research question;
  + Collect and organise the data; and
  + Answer the research question.
* Students will be provided with examples and suggestions, but are invited to design their own analysis in any (legal) field that is of interest to them.
* The Python code used to perform the analysis will be taken into account for grading; elegance, efficiency, etc., are irrelevant (as long as it runs, it runs), but all students will be expected to be able to explain the logic behind coding choices and the input/output processes.

The presentation will (hopefully) take place in the offices of a law firm, legal department, or legal tech start-up.

# Annexes

## Annex 1: Install Python on your Computer

Follow this link:

<https://www.python.org/downloads/>

Click Download

Then click on the Installer, and **make sure that the option “Add Python to PATH” is ticked** (it is not by default). In “Customize Installation”, click on “Install for All Users”.

Graphical user interface, text, application

Description automatically generated

(For further info, Mac users can also follow the instructions [here](https://ehmatthes.github.io/pcc/chapter_01/osx_setup.html#:~:text=Python%20is%20probably%20already%20installed,%2C%20and%20then%20press%20Enter.)&text=If%20you%20have%20Python%203.4,by%20using%20the%20installed%20version.); Windows users [here](https://www.digitalocean.com/community/tutorials/install-python-windows-10).)

Check that all is fine by opening “cmd” or “Command Prompt” (on Windows) or the “Terminal” (on Mac). Type “python”, and press Enter. If you see no error, you are good.[[1]](#footnote-2) Then type (exactly) …

print(“Hello World !”)

… and check that it renders as follows:

Text

Description automatically generated

## Annex 2 : Install PyCharm

PyCharm can be found [here](https://www.jetbrains.com/pycharm/download/#section=windows). Be sure to install the “Community” (free) version.

Graphical user interface, text, application

Description automatically generated

Click Download (right button), then follow the installation process.

## Annex 3 : Install a Source Code Editor [Optional]

This is optional, though could prove handy as we study structured data in XML files.

Windows users may want to install NotePad++ (<https://notepad-plus-plus.org/>), which is a better version of the basic notepad, and allows you to see and investigate coding scripts in more details.

Mac OS users can refer to this article. <https://setapp.com/how-to/alternative-to-notepad++-for-mac>. From the suggested alternatives, Sublime Text and Atom seem the most relevant.

## Annex 4 : Access the Course

You will need:

1. To create a GitHub account [here](https://github.com/join).
2. Add your GitHub account name (or “handle”) next to your name on [this shared file](https://docs.google.com/spreadsheets/d/1lMzAzdP7LBBkAIR-zdEIjX8XSa9TuH39jHBQsiTpyCU/edit?usp=sharing).
3. Wait for the invitation to join the GitHub Repository holding the course archive.

The course’s repository is [here](https://github.com/smartlawhub/Legal-Data-Analysis/tree/master) (can be accessed only after invitation).

## Annex 5 : Bring it All Together

Once you have been added to the course repository:

* Start PyCharm;
* In the “Projects” window, click on “Get from VCS/Version Control”; from this, either:
  + Input your GitHub details, and you should be able to see the course’s repo. If you don’t see the course’s repo, check that you accepted my invitation to the repository; or

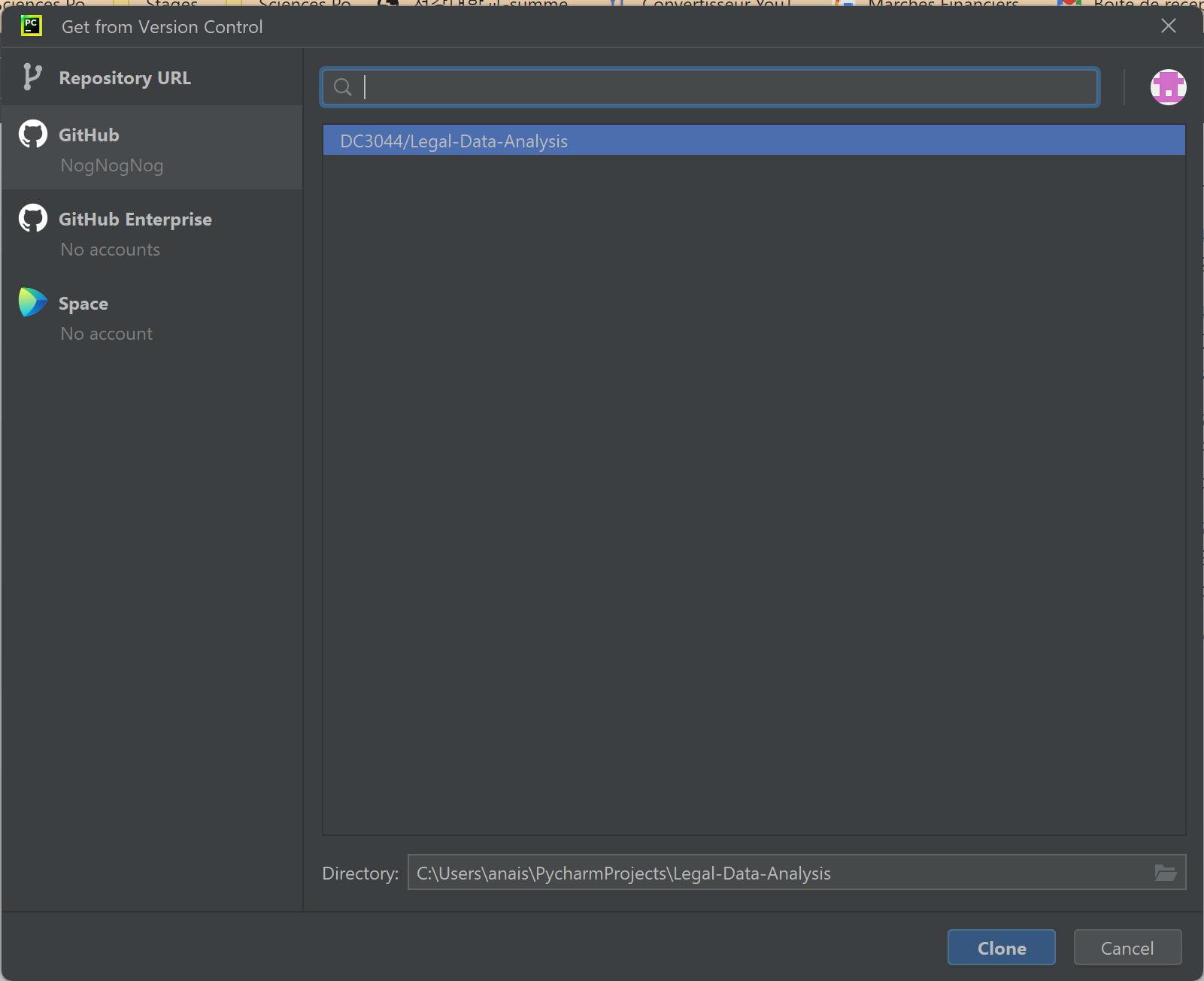


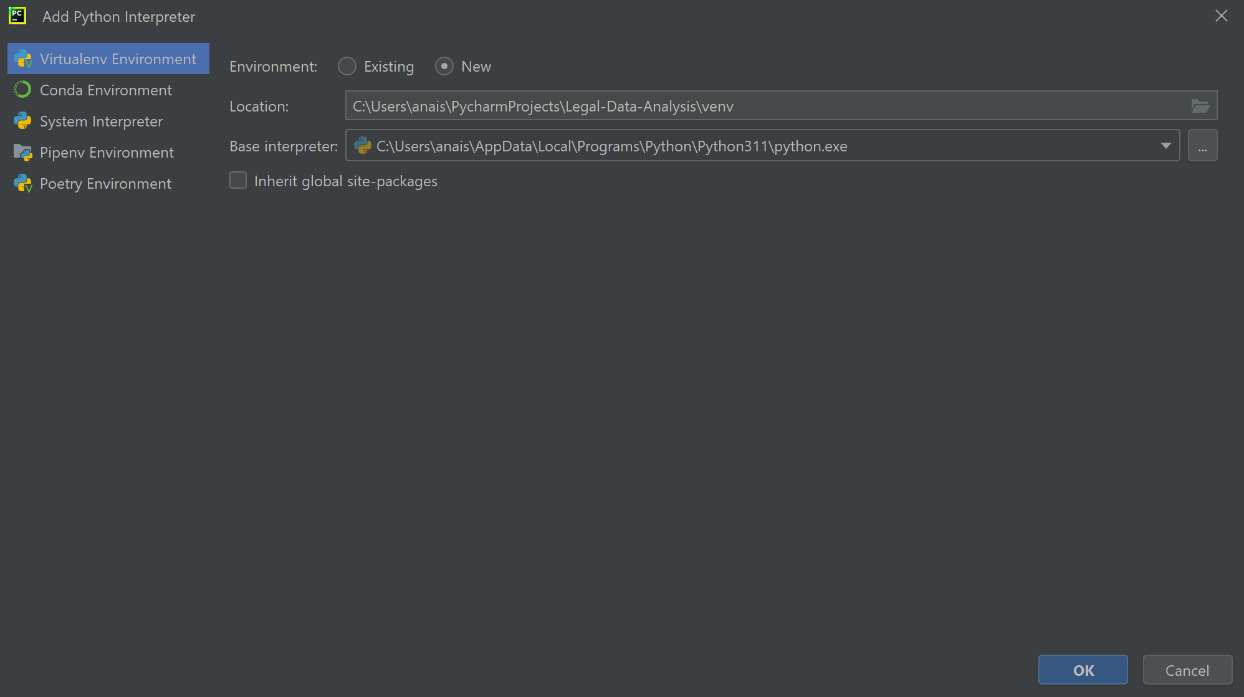
Figure : What it looks like when you have inputted your GitHub login

* + Input the course’s repository [URL](https://github.com/DC3044/Legal-Data-Analysis), and click on “Clone” (you will first need to install Git, but there is a button for this).
* Go to File > Settings, and in the search box type “Interpreter”:
  + There should be two results, named “Python Interpreter” and “Console Interpreter”.

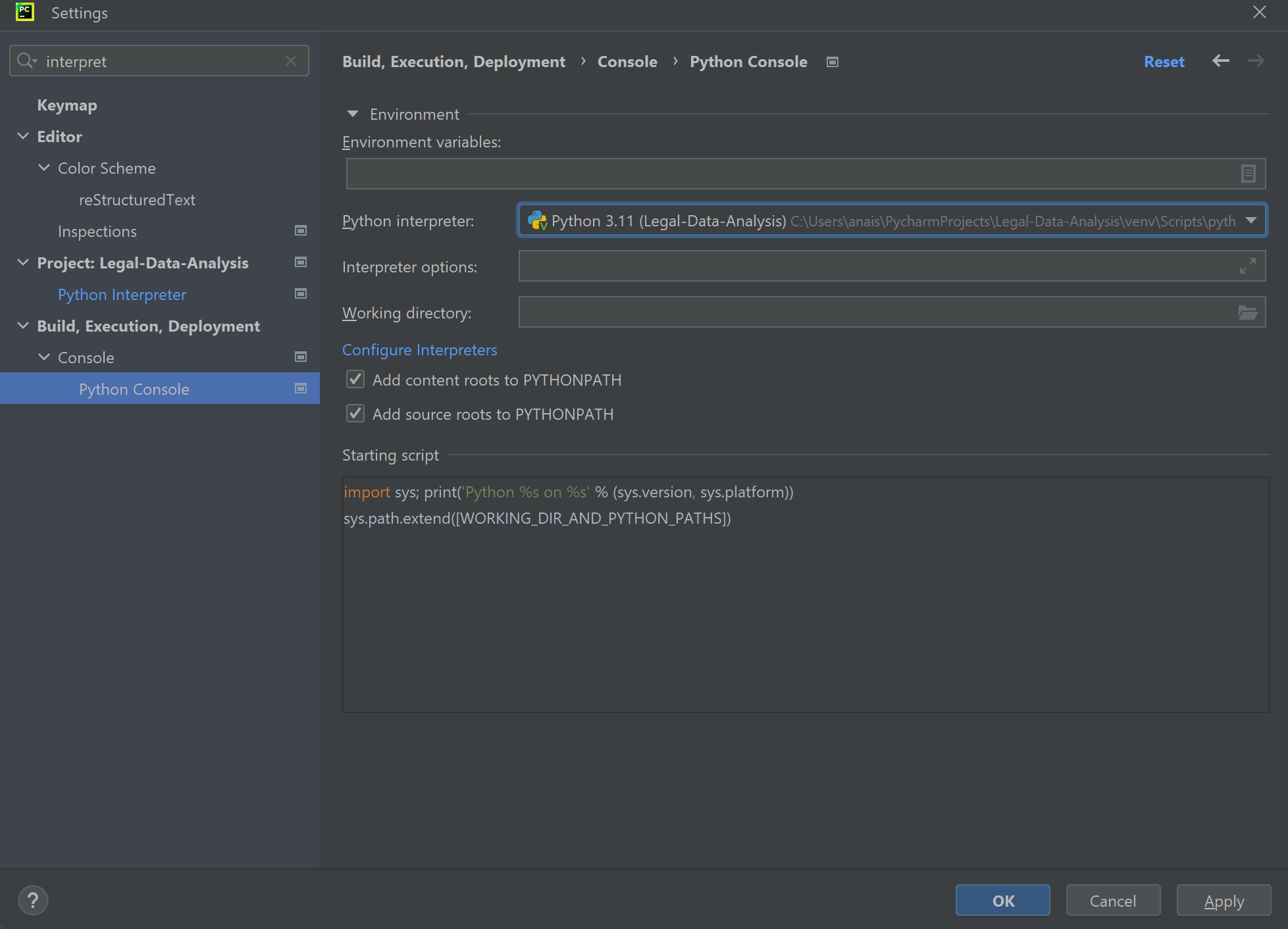
A screenshot of a computer

Description automatically generated with medium confidence

* + - Go on “Python Interpreter”:
      * If there is nothing in the dropdown list, click on “Add Interpreter”, top right.
      * You should be able to see a new window, with a list of interpreters on the left. Select “VirtualEnv Environment”, and click “Ok”.



* + - Go on “Python Console”, and on the line “Python Interpreter” you should be able to see the interpreter you just created.
      * Click on “Apply”.
      * Make sure **both** interpreters refer to the same path name that ends with “python.exe”.



* + Wait a few minutes, or even restart PyCharm, to be able to use the Console (see below).

Display the “Project” pane on the left-hand side. The written course can be found in the **Lessons** folder, with the corresponding code in the **Scripts** Folder. For ease of use, follow this process:

* Open both the .md and .py files with the relevant lessons;
* Click on “Preview Only” for the .md file (top right of the pane, third button);

Graphical user interface, text, application

Description automatically generated

Figure : Show Preview Only, on top right; Project pane on top left

* From the top menu, select Window > Editor Tab > Split Vertically or Split Right, until you can see both the .md and .py files;
* Close the Project pane on the left-hand side; and
* Click on PyCharm’s “Python Console” pane, which by default is at the bottom. Dock it on the “Right-Top” of the IDE, by clicking on the Wheel > Move To.

Final result should look like this:

Graphical user interface, text, application, email

Description automatically generated

Figure : Final Result, you are ready to Code

## Annex 6 : Install iPython for greater ease

One last step:

* In PyCharm, go on the “Python Package” pane (by default in the bottom).

Graphical user interface, text

Description automatically generated

* Look for “ipython”;
* Click on “install”;
* If needed, reload the console to now have the iPython display.

1. Otherwise, there might be an issue with your environment variables, etc., which we’ll fix together. [↑](#footnote-ref-2)