# Data Science Professional Survey

## Introduction

Being a keen data scientist, and wanting to maximise my employability, I needed to know what skills and abilities are commonplace in order to be a competitive applicant. This was an exploratory analysis of a dataset survey provided by Kaggle on the data science field. The purpose of the analysis is therefore to gain a basic, and visual, idea of what technologies are most common within data science; for example, what the most common data science libraries are. My aim was not to perform an overly scientific, or rigorous analysis, but more as quick introduction.

The particular questions of interest were: age distribution; years of experience; level of education; and the most common machine learning techniques.

## Methods

This analysis used the data collected by Kaggle in 2020 where respondents were queried about their data science career. The professional data science survey has been ongoing annually since 2017, and occurs for 3.5 weeks in the month of October.

The analysis utilised the Jupyter Notebook IDE, alongside the standard Python data science library stack. A more novel module was PyPlot Express which was used for generating visually pleasing and interactive plots. This will be expanded upon in a later section.

# Results

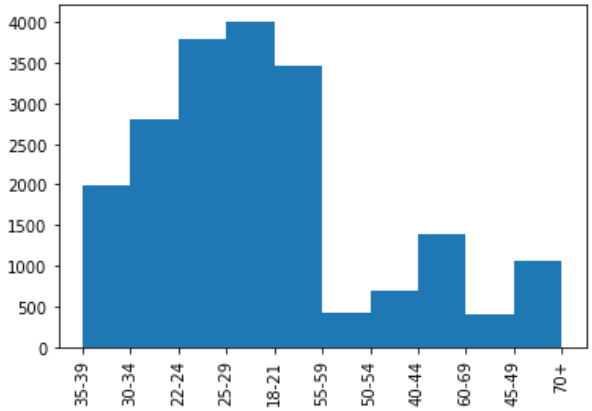


*Figure 1. This plot displays the age distribution of the respondents to the 2020 Kaggle Data Science Survey. Plotted using Matplotlib.*

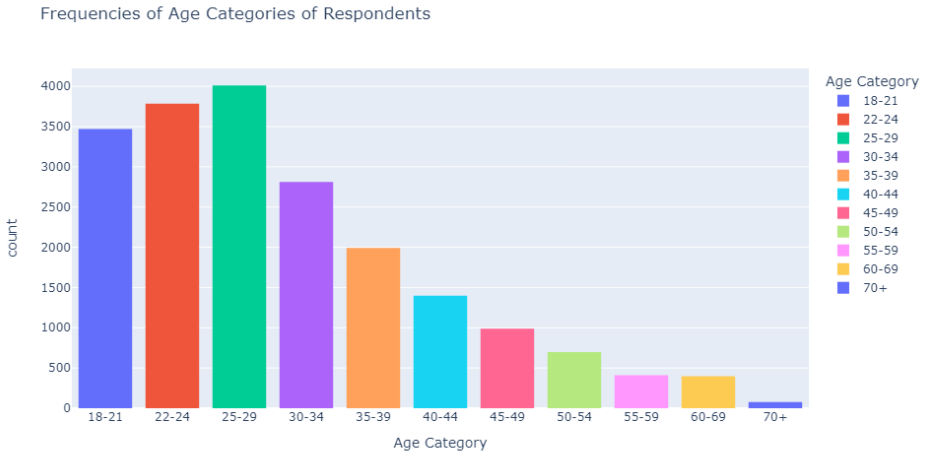


*Figure 2. The age distribution of respondents to the 2020 Kaggle Data Science Survey. Plotted using PyPlot Express.*

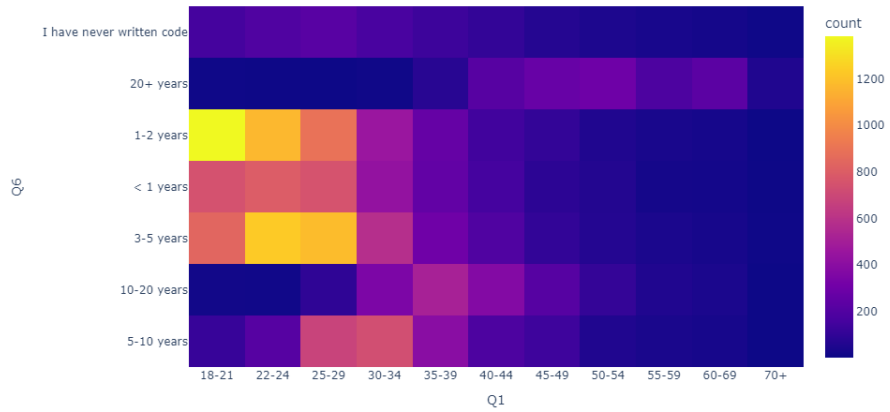


*Figure 3. A heatmap displaying the age of respondents, and their experience level from those who responded to the 2020 Kaggle Data Science Profession Survey. Plotted using PyPlot Express.*

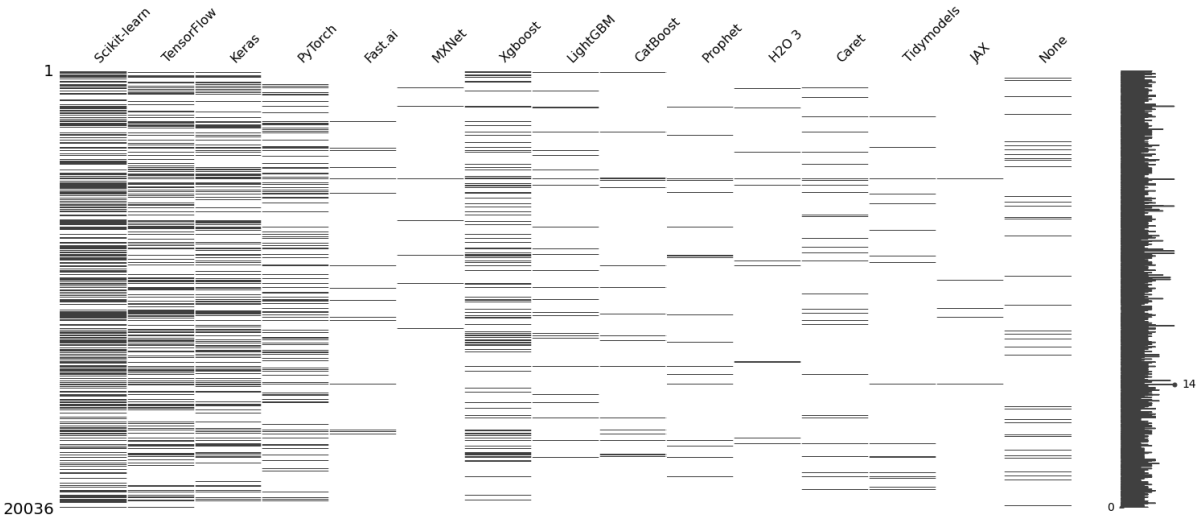


*Figure 4. A plot of missingness for the machine learning techniques of interest, from the 2020 Kaggle Data Science Survey. Plotted using the Python library Missingno.*

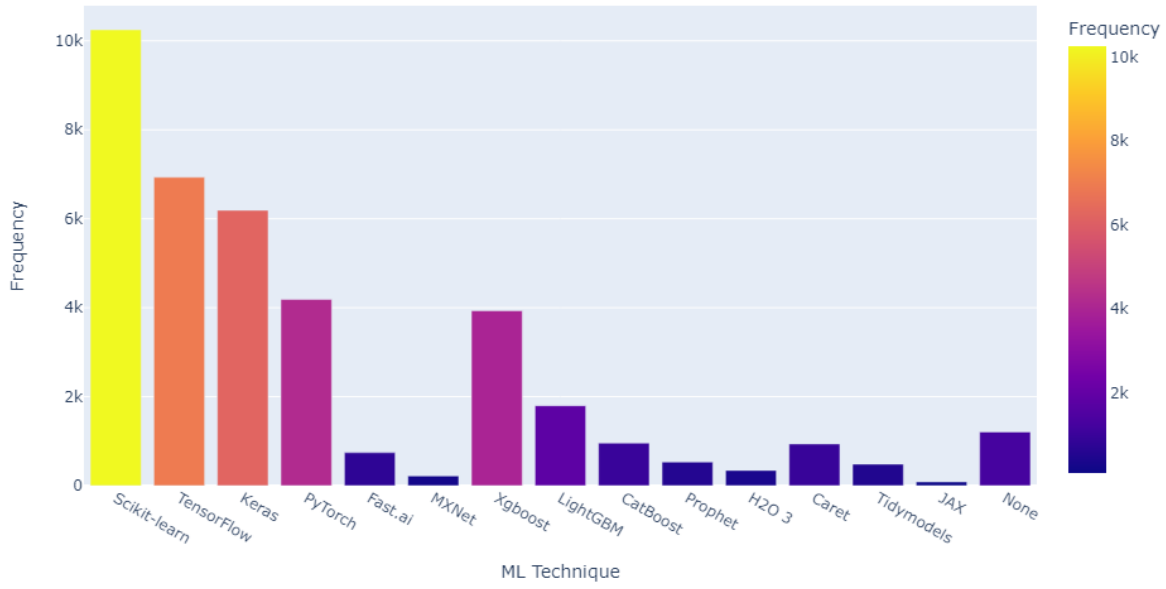


*Figure 5. The frequency of use of different machine learning techniques. Data from the 2020 Kaggle Data Science Survey. Plotted using PyPlot Express.*

## Conclusions

### Age distribution

As indicated in figures 1, and 2, the most common age category within the data science field is 18-29. Beyond that, there is a marked progressive decrease in the frequency of the age categories, as the data is heavily positively skewed.

I want to discuss the contrast in the quality of the plots in figures 1 and 2; displaying the same information, albeit the age categories are out of order in figure 1. Figure 1 was plotted using with Matplotlib, whereas figure 2 was plotted using PyPlot Express. Figure 2 looks much more visually pleasing, and is perfect for use in a presentation because it is able to display the information clearly. After learning Pyplot Express, I have been pleasantly surprised with the package and its versatility. I will continue using this plotting package extensively in the future, particularly for presentations.

However, I will still continue using Matplotlib, albeit more for diagnostic and basic exploration of the data.

## Age-Experience

As displayed in figure 3, the heatmap indicates a high density of respondents between the ages of 18-29, with 1-5 years of experience writing code. This is particularly interesting because 1-2 years of coding experience is the most common within my age bracket (18-21). I believe that in order to stand out in such a competitive industry, I need to continue working extensively on my personal projects; this will allow me to gain greater experience with coding, as well as the intricacies of data science.

## Machine learning techniques

Figure 4 displays an attempt to quickly visualise the most common machine learning techniques using the Missingno Python library. Missingno was intended to visualise missing data, however, due to the nature of the way the data is input into the dataset, it can be repurposed to quickly visualise the most common ML techniques. This library was so easy to use, and is so useful for quickly visualising missingness, that I will be sure to use it on a regular basis on my data analysis tasks in the future.

Figure 5 displays the same information as figure 4, except it was plotted using PyPlot Express. Although this is much more visually appealing than figure 4, it require greater data processing, and was a much more finnicky experience than Missingno. However, the final output is well worth the effort.

Scikit-Learn, TensorFlow, and Keras, are the three most popular ML techniques. I am fairly familiar with these packages as part of my experience working on my own machine learning techniques. For example, my dissertation project on classifying depressed patients based on audio recordings of their speech, required extensive use of these three packages.

Figure 5 also indicates that PyTorch is also very popular. I need to perform some research into the package, and understand its advantages over other libraries. I believe this would be a useful library to have experience with, as it is clearly very popular in the industry.

## What have I learnt from this experience?

One of the things I have learnt from this experience was a new library package, known as PyPlot Express. For a while I have been displeased with Matplotlib for data visualisation for anything other than basic visualisation. Although Matplotlib is exceptionally easy to use, the graphics are not visually appealing which decreases its utility for use in presentations. Therefore, I wanted to try PyPlot Express as an alternative. Although it quickly became apparent that PyPlot is more finicky than Matplotlib, and required more processing of the data into an acceptable format, the final graphs were well worth the effort; not only do they have a pleasing modern aesthetic, but they are interactive when you hover the cursor above a point in the graph. I am therefore very glad I have learned this new plotting package, and will use it extensively in the future.

I have learnt an additional Python library known as *missingno*. A key feature of this library is it ability to readily create a barchart for each column within the dataset to indicate how much data is missing. This is achieved by using the *matrix( dataframe )* command. Within seconds I can have a rough idea of how much data is missing and decide on an appropriate plan to deal with the missing data. Since learning about the library, I have used it extensively on a regular basis in many other projects.

As a challenge to myself, I performed minimal pre-processing of the original data using Excel. This was done on purpose to ensure I develop the skills required for future data science endeavours involving large and complex datasets. However, what this experience has highlighted to me, is that if you have a pre-defined set of questions for the analysis, it can be faster and more efficient to pre-process the data as much as possible in Excel before loading it into the IDE of choice; it can limit future analysis because certain information was excluded, but in the case of having a set question, it can significantly speed up the process of analysis as there isn't a need to tailor an algorithm to process the data within the IDE.