

Module 2

Setting up Deep Learning Environment





Road map!

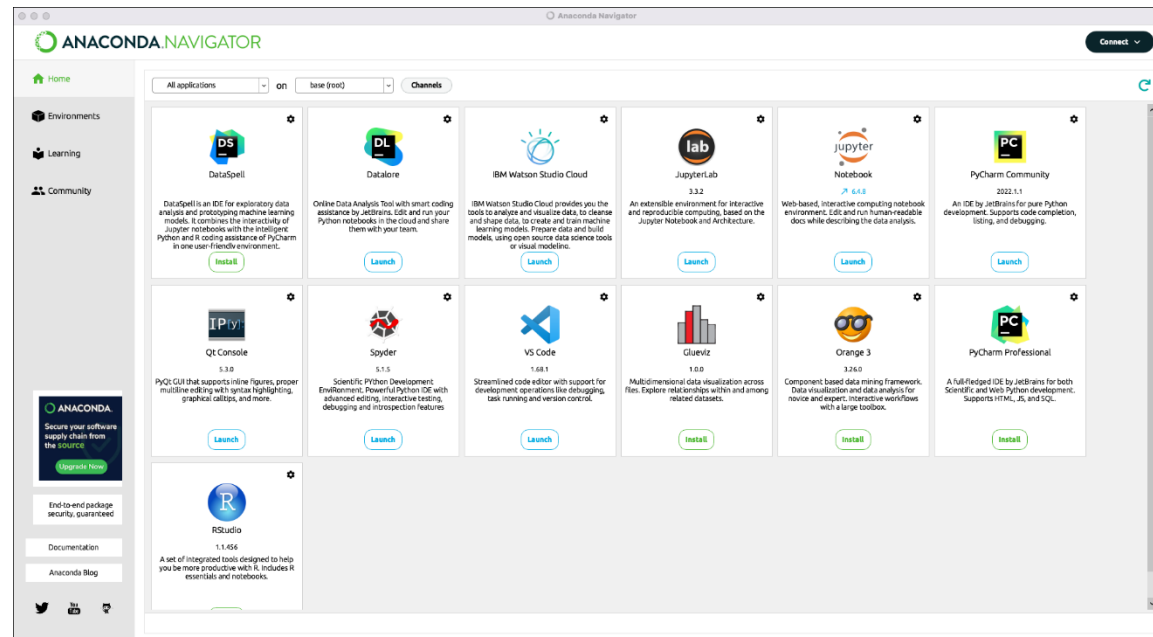
- Module 1- Introduction to Deep Learning
- **Module 2- Setting up Deep Learning Environment**
- Module 3- Machine Learning review (ML fundamentals + models)
- Module 4- Deep Neural Networks (NN and DNN)
- Module 5- Deep Computer Vision (CNN, FCN)
- Module 6- Deep Sequence Modeling (RNN, LSTM)
- Module 7- Transformers (Attention is all you need!)
- Module 8- Deep Generative Modeling (AE, VAE, GAN)
- Module 9- Deep Reinforcement Learning (DQN, PG)

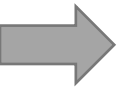


Install through



- Anaconda is a **distribution** of the Python and R programming languages for scientific computing, that aims to simplify package management with conda **environments**.
- Anaconda offers the easiest way to perform data science and machine learning on a single machine.
- Install Anaconda @ <https://www.anaconda.com/products/distribution>

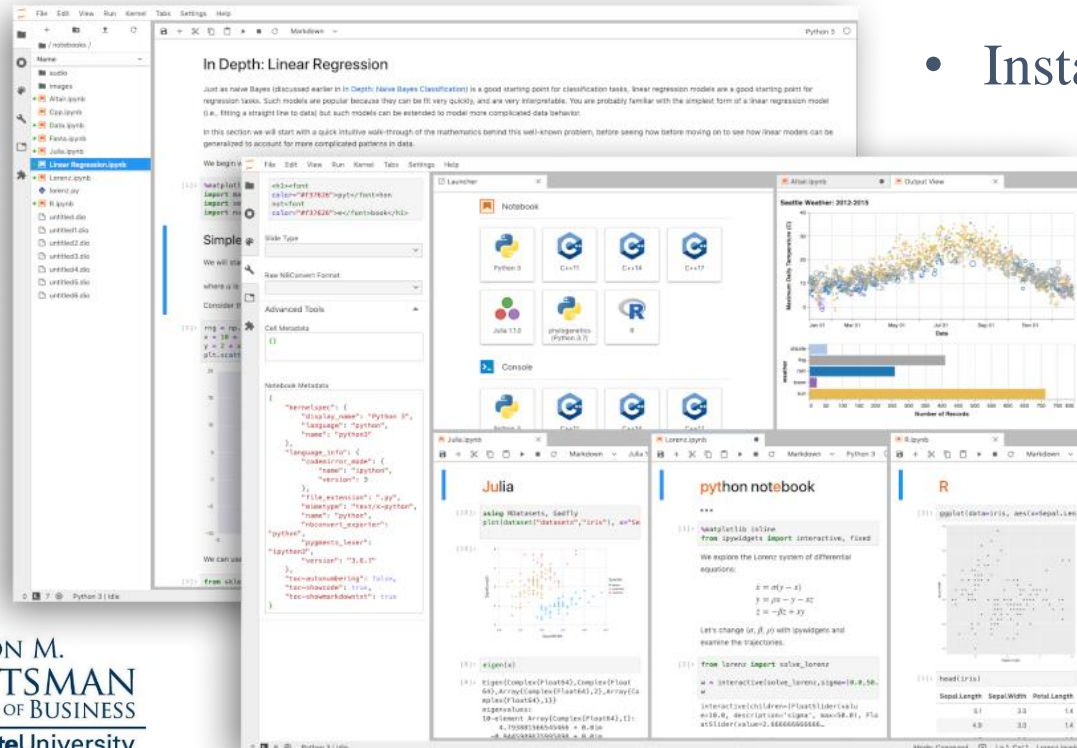




JupyterLab



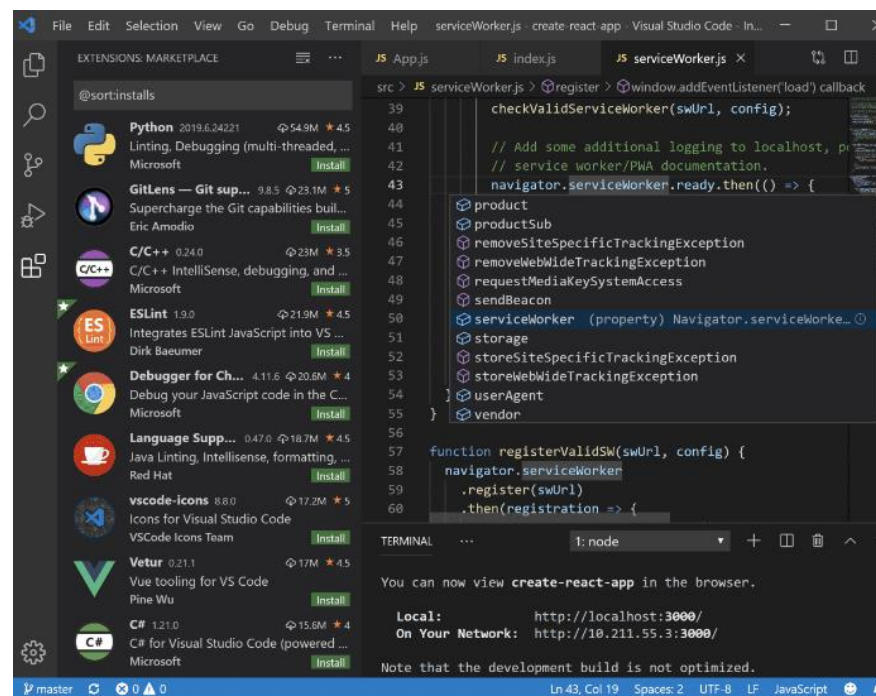
- JupyterLab is the latest **web-based interactive development environment** for notebooks, code, and data
- Jupyter's name is a reference to the three core programming languages supported by Jupyter, which are **Julia**, **Python** and **R**



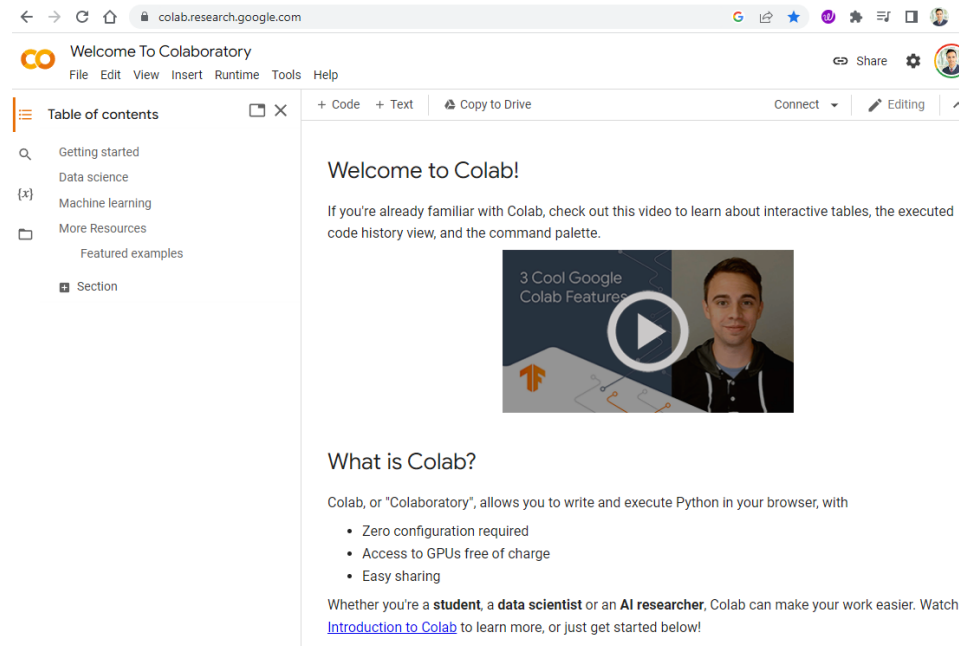
- Install JupyterLab @ <http://jupyter.org/install>

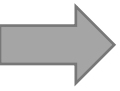
VS Code

- VS Code is one of the most popular source code editors
- Features include support for **debugging**, syntax highlighting, intelligent **code completion**, code refactoring, and **embedded Git**.
- Install VS code @ <https://code.visualstudio.com/>



- Colab is a free hosted **Jupyter notebook-style environment** that runs entirely in the **cloud** and requires no setup to use. It also provides access to **machine learning libraries** and computing resources including **GPU**.
- Colab allows anybody to write and execute arbitrary **python code** through the **browser**, and is especially well suited to machine learning, data analysis and education. <https://colab.research.google.com/>



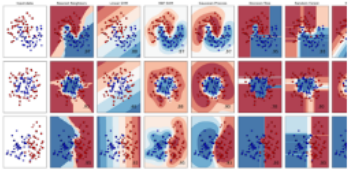


Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...



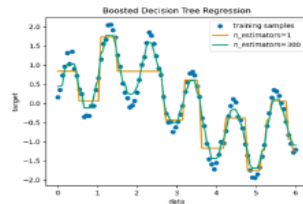
Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...



Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, and more...



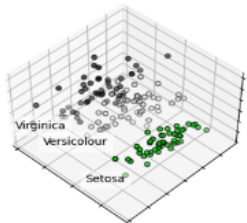
Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: PCA, feature selection, non-negative matrix factorization, and more...



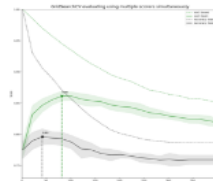
Examples

Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning

Algorithms: grid search, cross validation, metrics, and more...



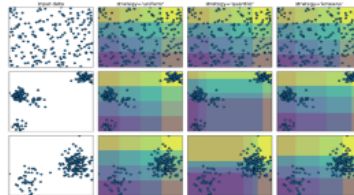
Examples

Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.

Algorithms: preprocessing, feature extraction, and more...



Examples

- Scikit-learn is an **open-sourced Python** library and includes a variety of unsupervised and supervised learning techniques.
- It is based on technologies and libraries like Matplotlib, Pandas and NumPy and helps simplify the coding task.
- Install Scikit-learn @ <https://scikit-learn.org/stable/install.html>



- PyCaret is an **open-source**, **low-code** machine learning library in Python that automates machine learning workflows.
- PyCaret is essentially a **Python wrapper** around several machine learning libraries and frameworks
- Install PyCaret @ <https://pycaret.gitbook.io/docs/get-started/installation>

```
# load dataset
import pandas as pd
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')

# init setup
from pycaret.classification import *
s = setup(train, target= 'target')

# model training and selection
best = compare_models()

# analyze best model
evaluate_model(best)

# predict on new data
predictions = predict_model(best, data =test )

# save best pipeline
save_model(best, 'my_best_pipeline')
```

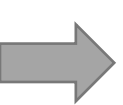
ID	Name
'lr'	Logistic Regression
'knn'	K Nearest Neighbour
'nb'	Naives Bayes
'dt'	Decision Tree Classifier
'svm'	SVM - Linear Kernel
'rbfsvm'	SVM - Radial Kernel
'gpc'	Gaussian Process Classifier
'mlp'	Multi Level Perceptron
'ridge'	Ridge Classifier
'rf'	Random Forest Classifier
'qda'	Quadratic Discriminant Analysis
'ada'	Ada Boost Classifier
'gbc'	Gradient Boosting Classifier
'lda'	Linear Discriminant Analysis
'et'	Extra Trees Classifier
'xgboost'	Extreme Gradient Boosting
'lightgbm'	Light Gradient Boosting
'catboost'	CatBoost Classifier



K Keras

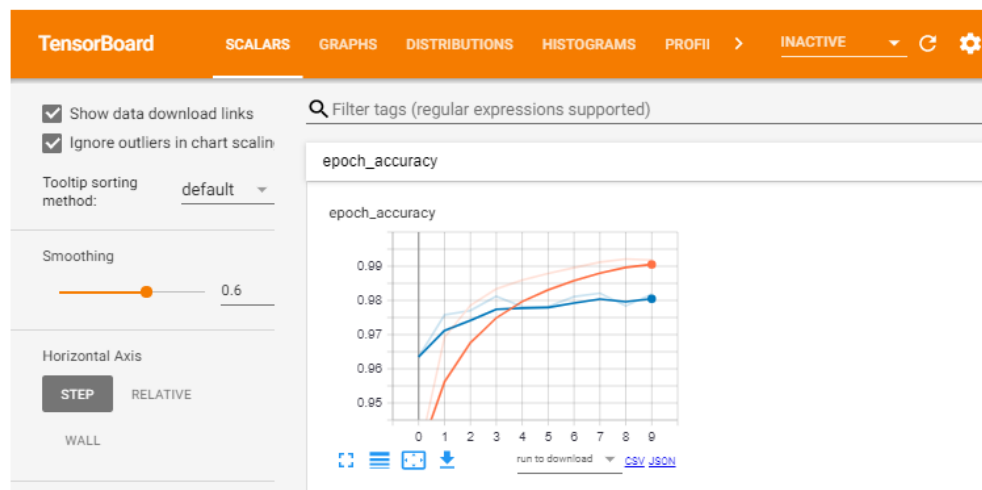
- Keras is a **high-level**, open-source **neural network** library written in Python. It was developed to make it easier for researchers and developers to build and experiment with deep learning models.
- The Keras API became the official high-level API for TensorFlow 2.0 in **2019**.
<https://keras.io/>

```
model = keras.Sequential(  
    [  
        keras.Input(shape=input_shape),  
        layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),  
        layers.MaxPooling2D(pool_size=(2, 2)),  
        layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),  
        layers.MaxPooling2D(pool_size=(2, 2)),  
        layers.Flatten(),  
        layers.Dropout(0.5),  
        layers.Dense(num_classes, activation="softmax"),  
    ]  
)  
  
model.summary()
```



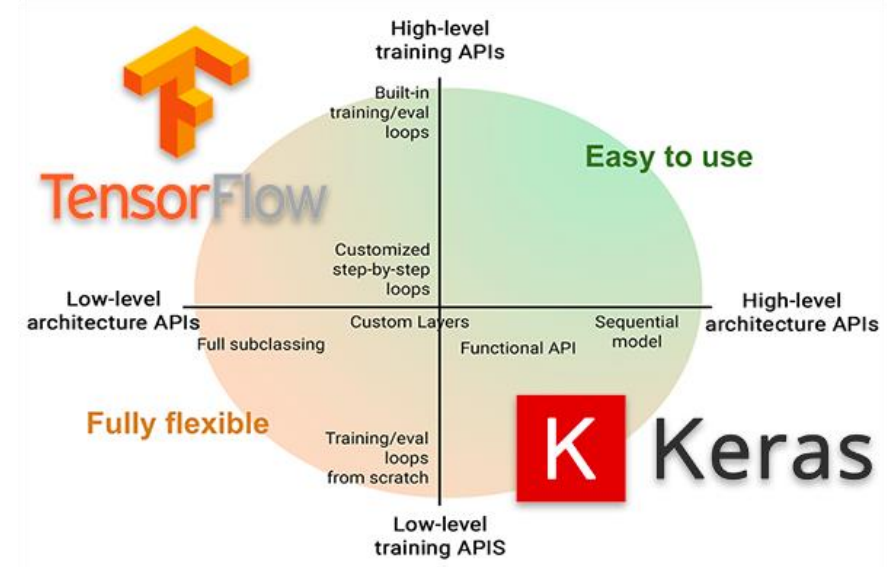
TensorFlow

- TensorFlow is a Google-maintained open-source end-to-end platform for prototyping and assessing machine learning models, **primarily neural networks**.
- TensorFlow also offers **TensorBoard**, a visualization tool for comparing and tracking our learned models.
- It can scale from a single CPU, to a GPU or cluster of GPUs all the way up to a multi-node TPU infrastructure.
- Build-in Google Colab. For local installation visit <https://www.tensorflow.org/install>

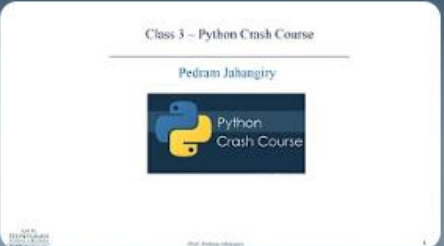


➔ K Keras Vs TF TensorFlow

- **Level of abstraction:** Keras is a **higher-level** library that provides a more intuitive interface for building and training models, while TensorFlow is a **lower-level** library that provides more flexibility but requires the user to specify more details of the model.
- Keras is a **standalone** library, while TensorFlow includes both a low-level library for numerical computations and a high-level library for building and training machine learning models.
- Keras is a user-friendly interface to TensorFlow



Available YouTube playlists



Python Crash Course


Pedram Jahangiry

Public

9 videos 1,612 views Updated today

[Play all](#) [Shuffle](#)

Codes are available on my GitHub account:
<https://github.com/PJalgotrader/platforms-and-tools>



Programming tips


Pedram Jahangiry

Public

13 videos 194 views Updated 3 days ago

[Play all](#) [Shuffle](#)

Codes are available on my GitHub account:
<https://github.com/PJalgotrader/platforms-and-tools>



Google Colab


Pedram Jahangiry

Public

3 videos 73 views Updated 6 days ago

[Play all](#) [Shuffle](#)

Codes are available on my GitHub account:
<https://github.com/PJalgotrader/platforms-and-tools>



PyCaret (Automated machine learning Python package)

Pedram Jahangiry

Public

3 videos 26 views Updated 3 days ago

[Play all](#) [Shuffle](#)

All you need from PyCaret Python library to automate your ML workflow.
Codes are available on my GitHub account:
<https://github.com/PJalgotrader/platforms-and-tools>

→ Platforms and Packages

Listed below are some Python packages and platforms that will be used in the deep learning and deep forecasting courses.



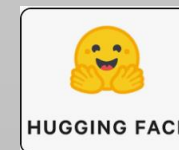
General Python libraries



Machine Learning libraries



Deep Learning libraries





Setting up Deep Learning Environment



Personal Workstation



Cloud Platforms



Google Colaboratory

Pros

- Full control over hardware and software
- Work offline
- Fixed cost

Cons

- Scalability
- Maintenance (both hardware and software)

- Powerful computing resources
- Scalability
- Ease of use
- Cost-effective: Pay-as-you-go
- Collaboration

- Expensive for large-scale experiments
- Dependency on the provider
- Limited control
- Internet connection
- Security

- Powerful computing resources (GPU, TPU)
- Ease of use
- Collaboration
- No need to set up a local environment

- Time limit
- Hardware limitation
- Data storage
- Limited control
- Internet connection
- Security

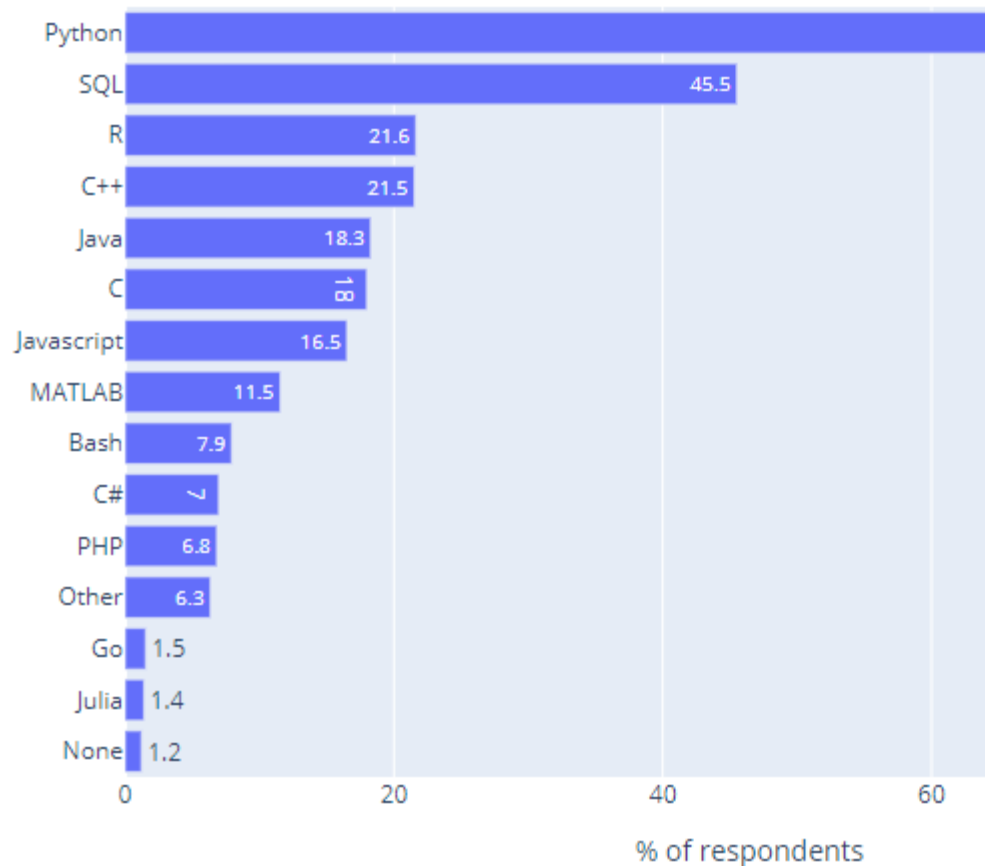
➔ Kaggle Survey 2022



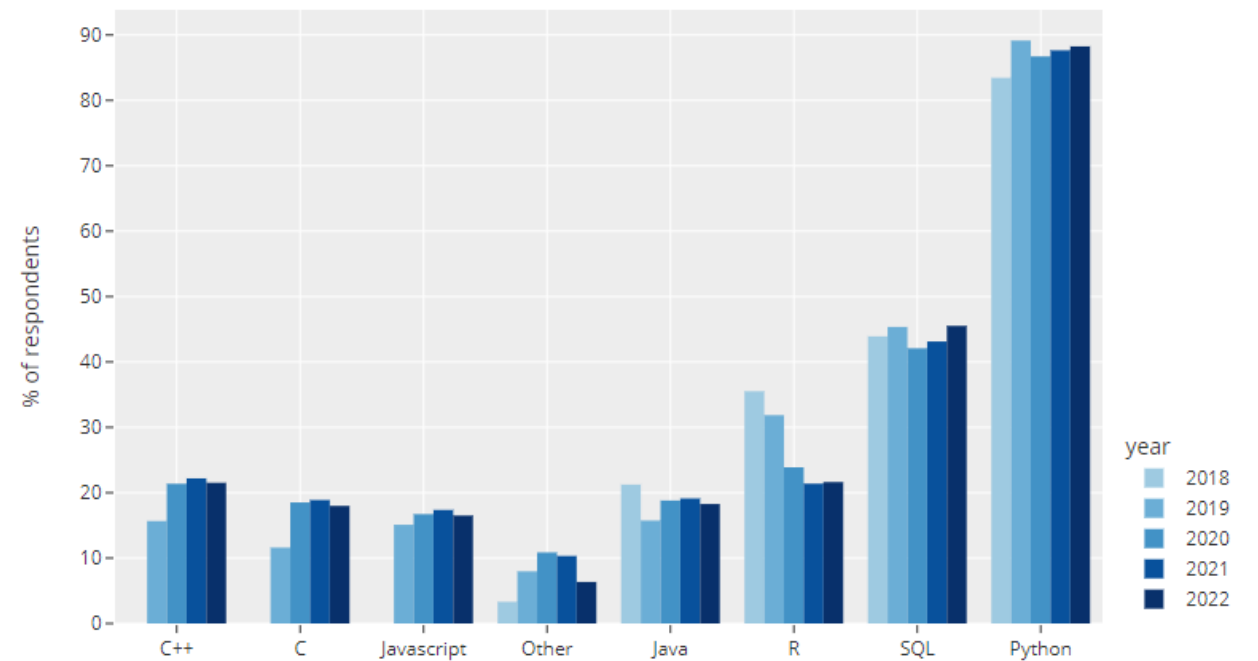
- Kaggle runs a yearly survey among machine learning and data science professionals worldwide.
- This survey is one of our **most reliable** sources about the **state of the industry!!!**

Programming Languages

Most Popular Programming Languages in 2022



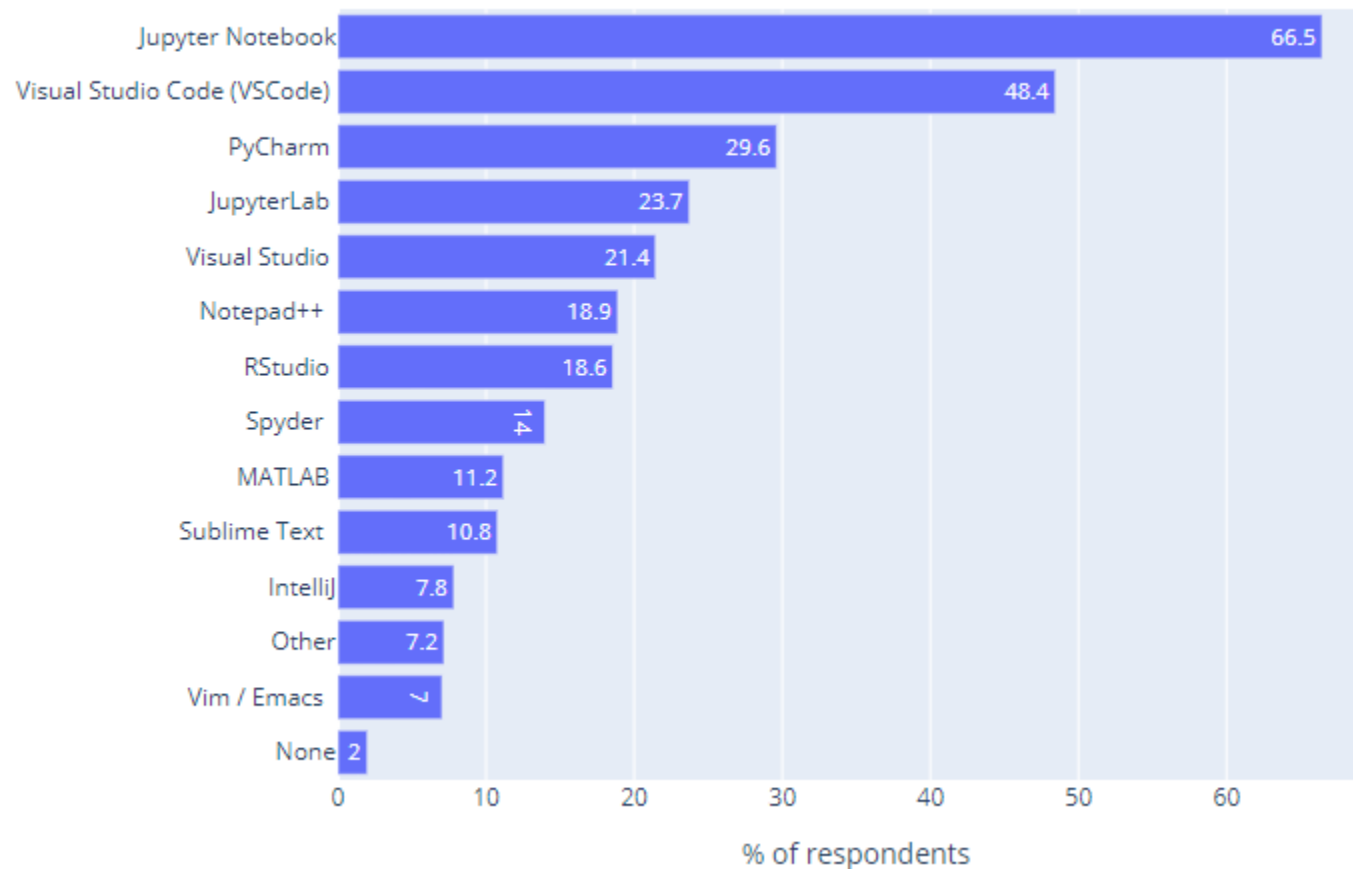
Most Popular Programming Languages 2018-2022





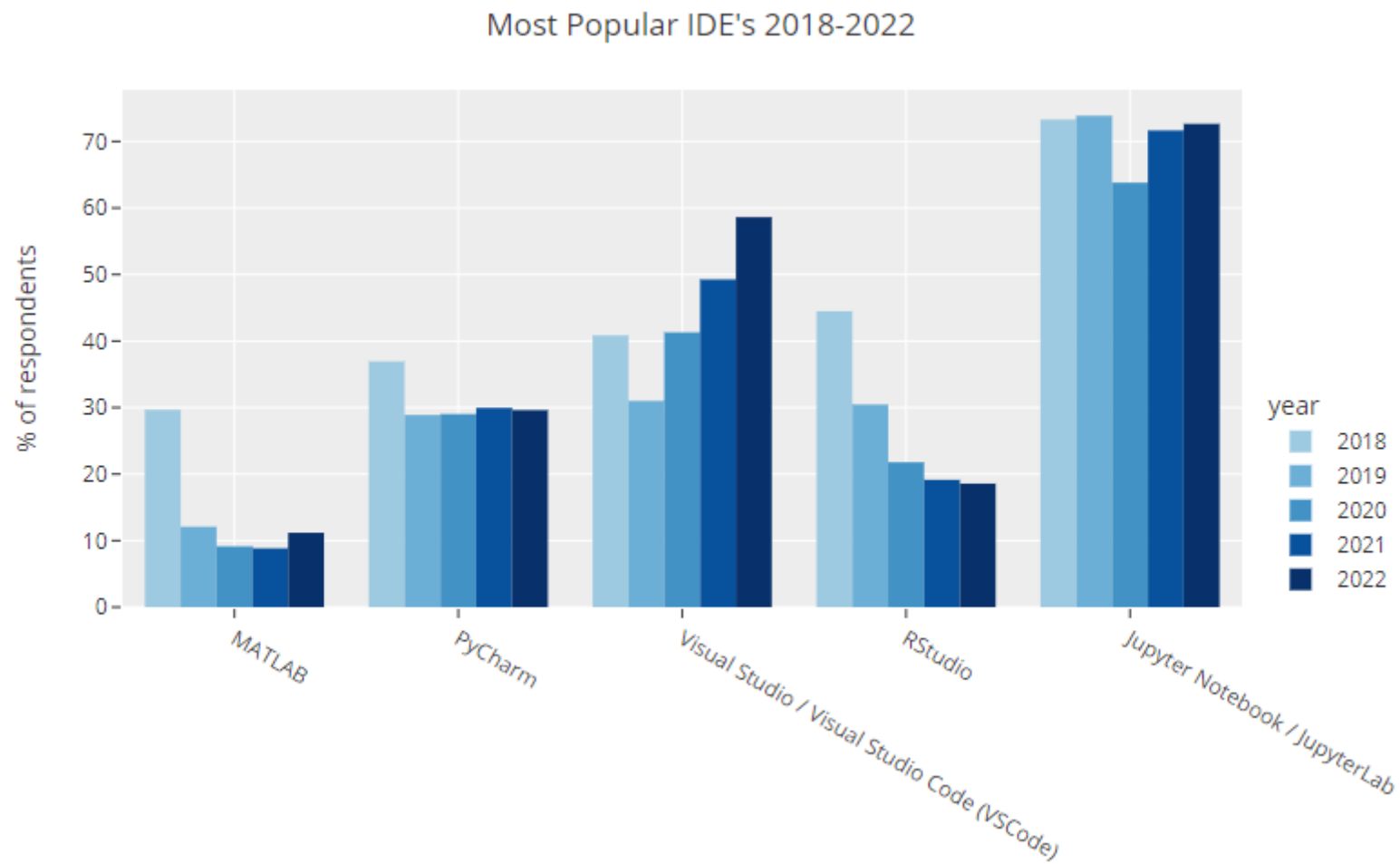
Popular IDE's

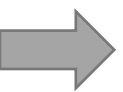
Most Popular IDE's in 2022





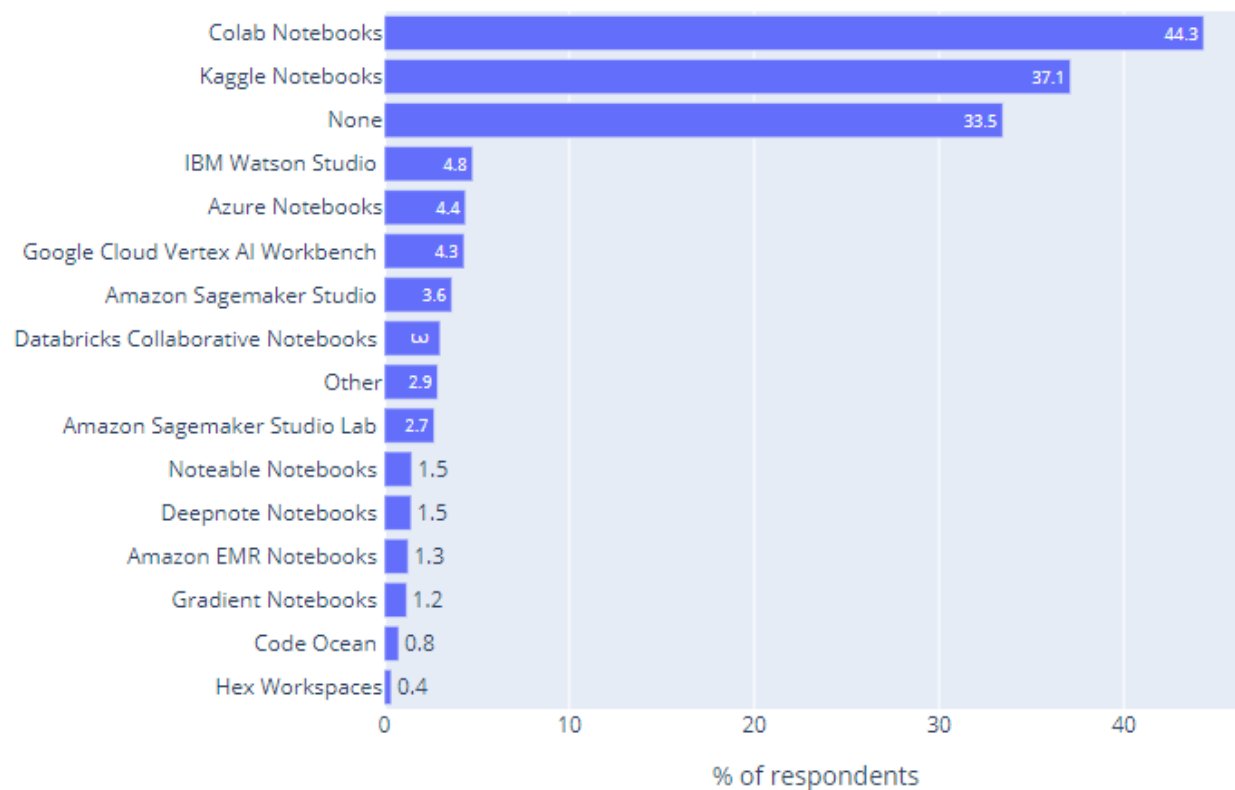
Popular IDE's



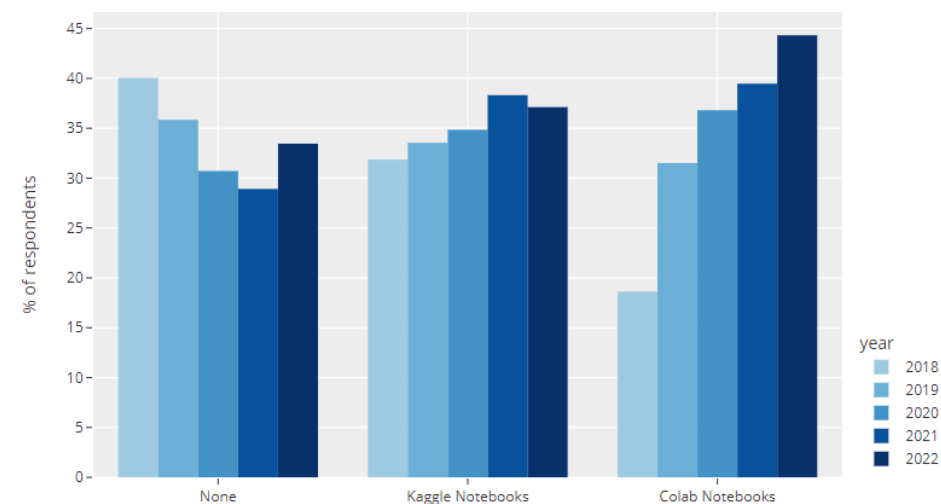


Popular notebooks

Most popular hosted notebooks in 2022 in 2022



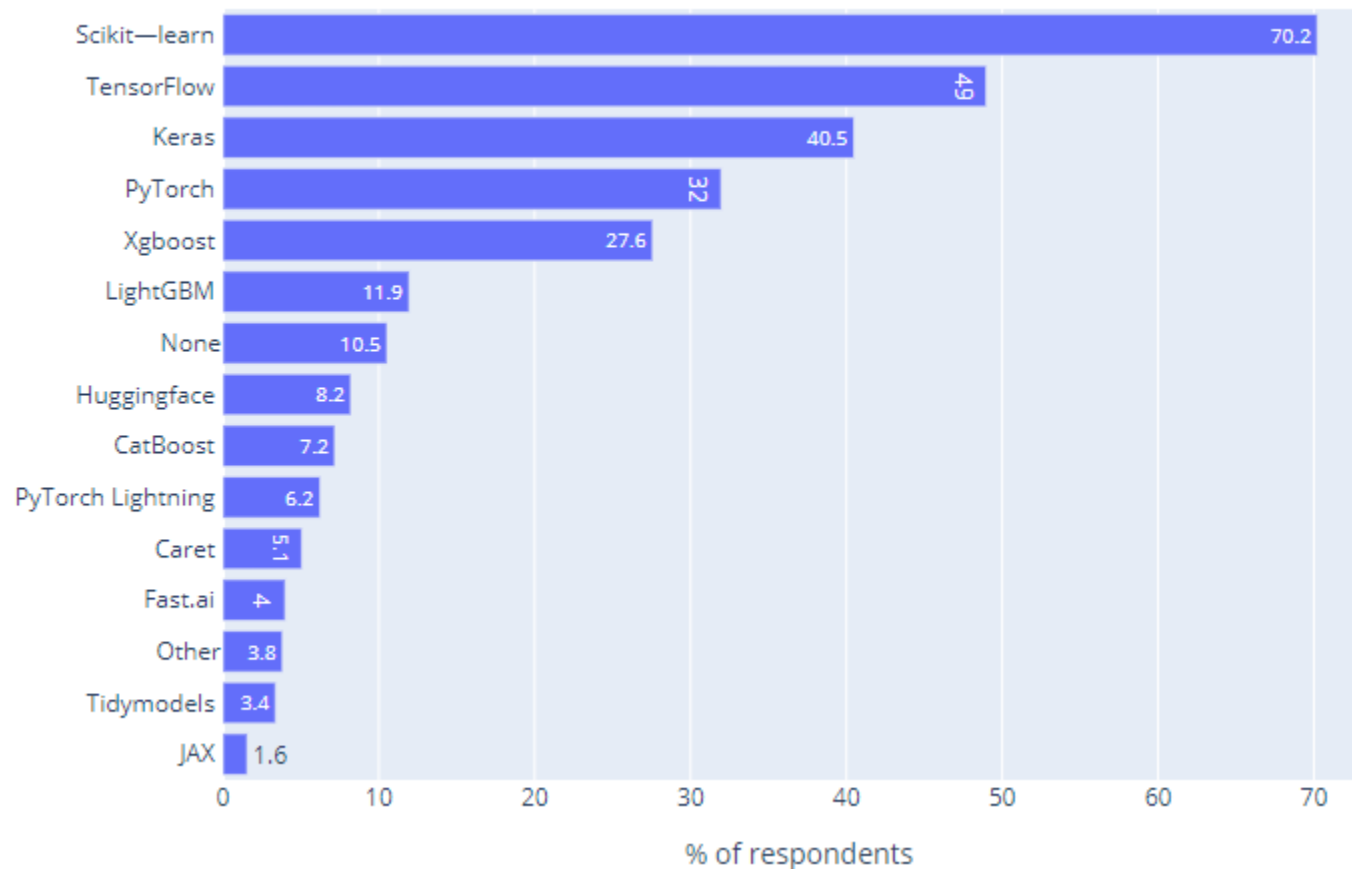
Most popular hosted notebooks products 2018-2022





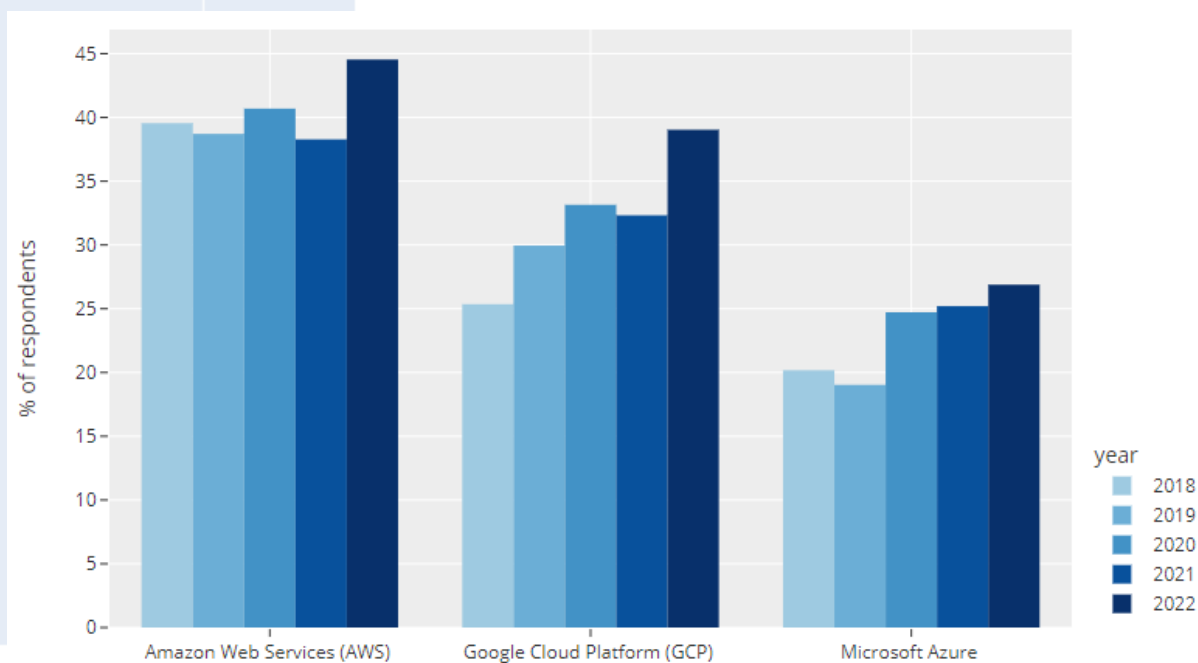
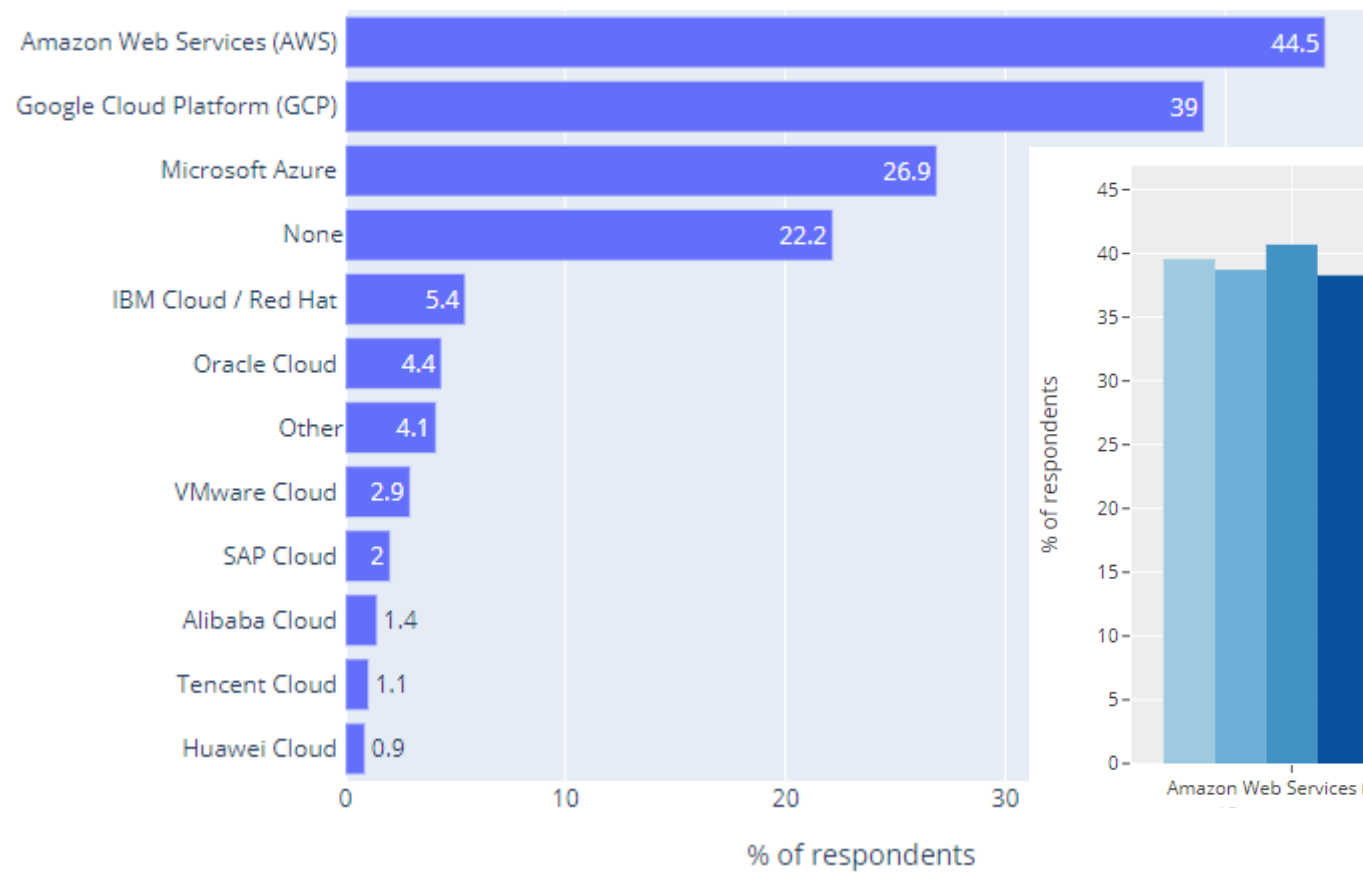
Machine Learning Frameworks

Most popular machine learning frameworks in 2022



→ Cloud computing platforms

Most popular cloud computing platforms in 2022





The modern machine learning landscape

- From 2016 to 2022, the entire machine learning and data science industry has been dominated by these **two approaches**:
 1. Deep learning
 2. Gradient boosted trees
- Most practitioners of deep learning use **Keras**, often in combination with its parent framework **TensorFlow**.
- This means you'll need to be familiar with **Scikit-learn, XGBoost, and Keras**

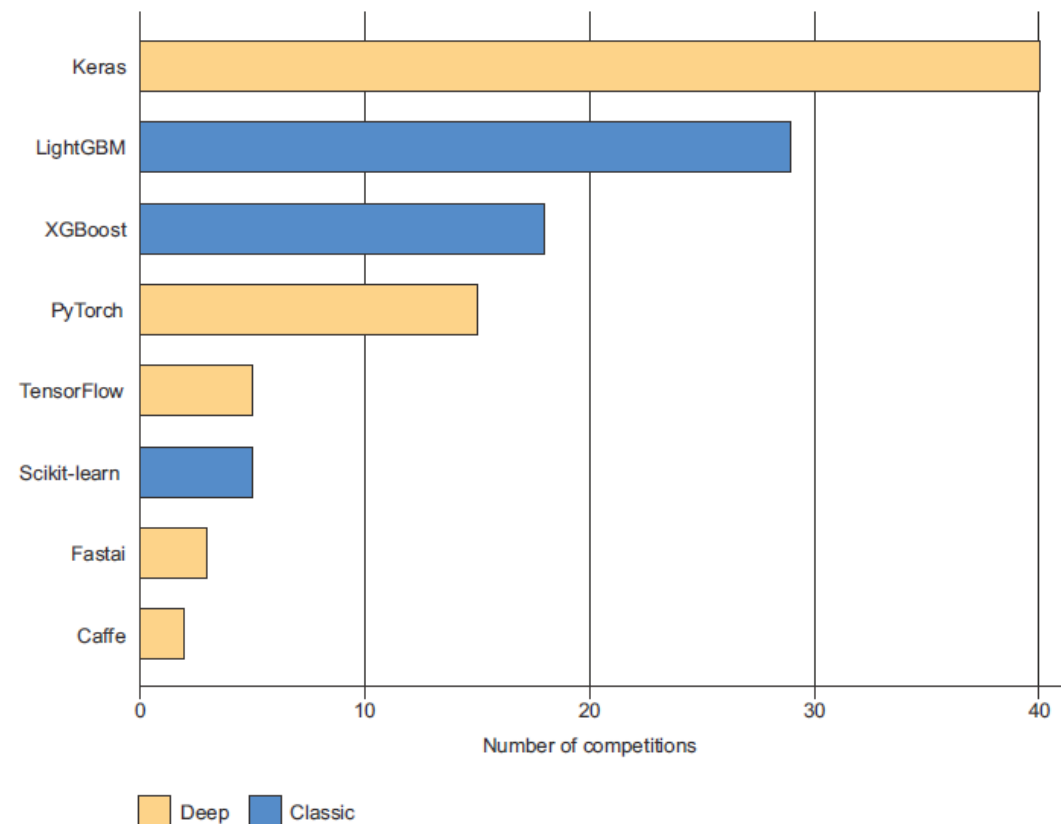


Figure 1.12 Machine learning tools used by top teams on Kaggle

➔ Road map!

- ✓ Module 1- Introduction to Deep Learning
- ✓ Module 2- Setting up Deep Learning Environment
- Module 3- Machine Learning review (ML fundamentals + models)
- Module 4- Deep Neural Networks (NN and DNN)
- Module 5- Deep Computer Vision (CNN, FCN)
- Module 6- Deep Sequence Modeling (RNN, LSTM)
- Module 7- Transformers (Attention is all you need!)
- Module 8- Deep Generative Modeling (AE, VAE, GAN)
- Module 9- Deep Reinforcement Learning (DQN, PG)

