

Module 2 – Part I

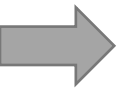
Setting up Deep Forecasting Environment (Python)



➔ Road map!

- Module 1- Demystifying Timeseries Data and Modeling
- **Module 2- Setting up Deep Forecasting Environment (Python)**
- Module 3- Exponential Smoothing
- Module 4- ARIMA models
- Module 5- Machine Learning for Time series Forecasting
- Module 6- Deep Neural Networks
- Module 7- Deep Sequence Modeling (RNN, LSTM)
- Module 8- Prophet and Neural Prophet





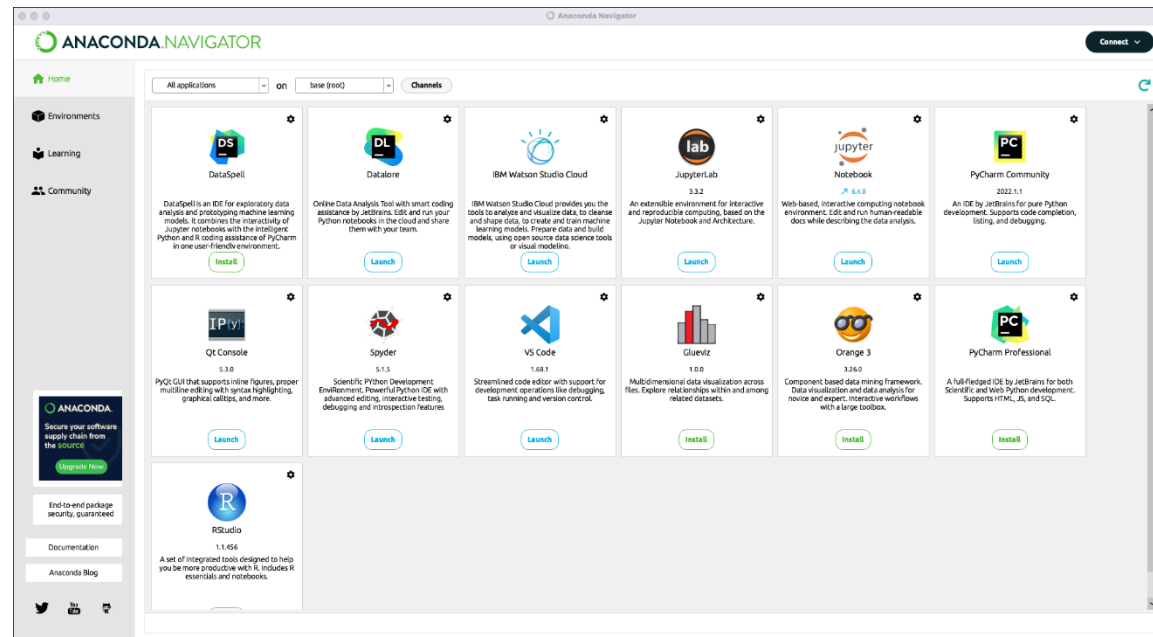
Install

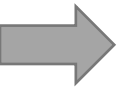


through



- 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/>

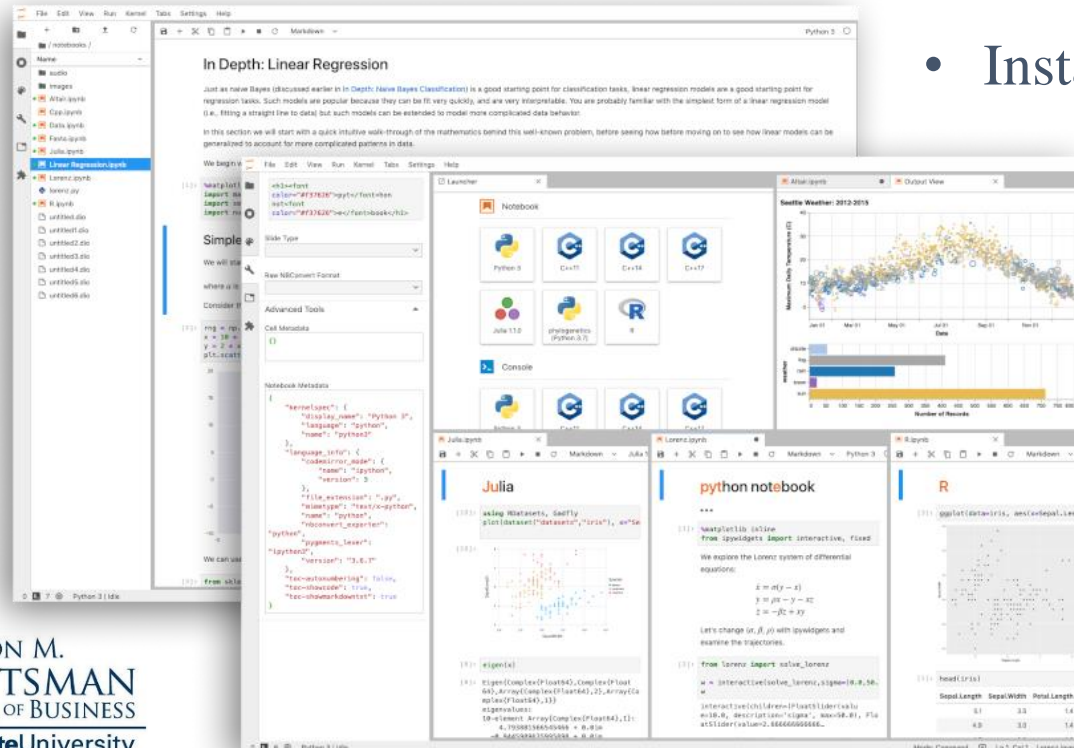




JupyterLab



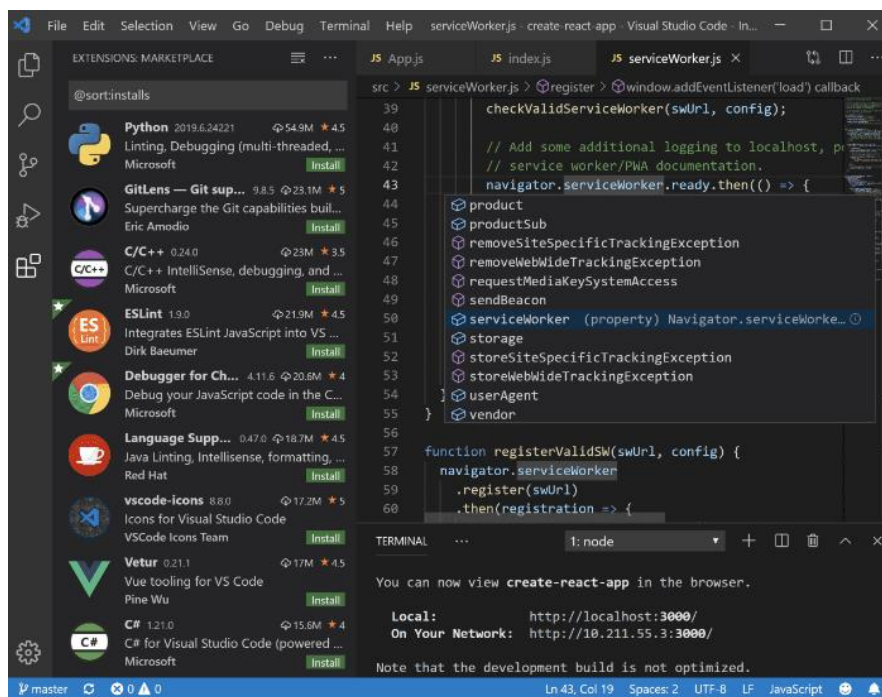
- JupyterLab is the latest **web-based interactive development environment** for notebooks, code, and data
- Jupyter's name is a reference to the three core programming languages supported by Jupyter, which are **Julia**, **Python** and **R**



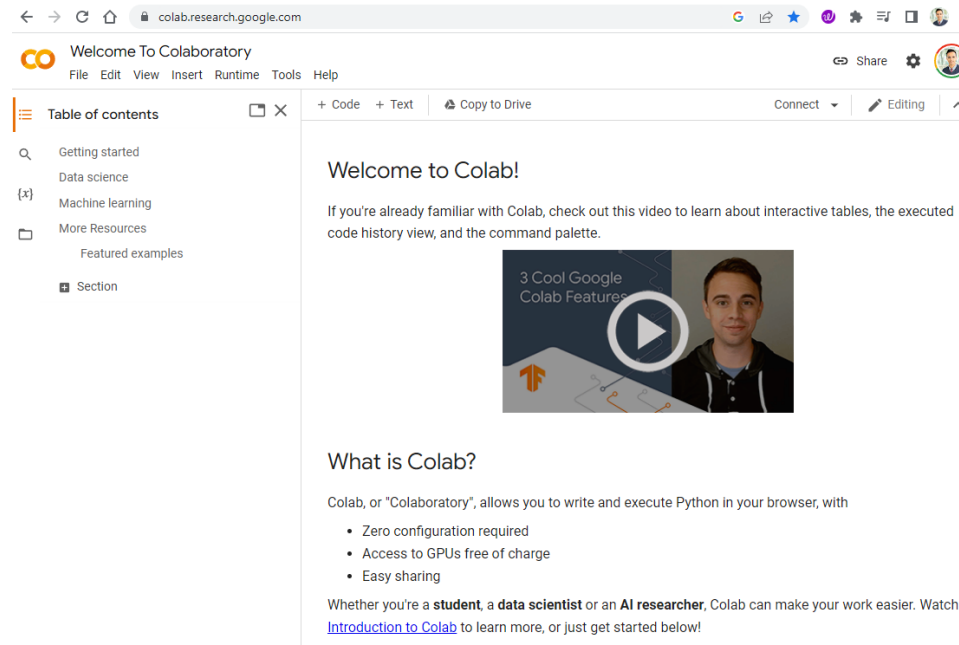
- Install JupyterLab @ <http://jupyter.org/install>

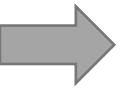
VS Code

- 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/>



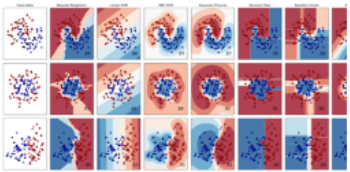


Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

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



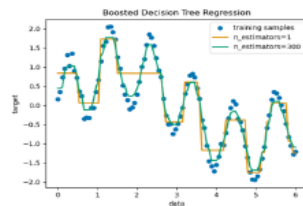
Examples

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...



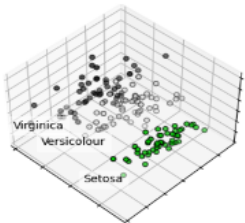
Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

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



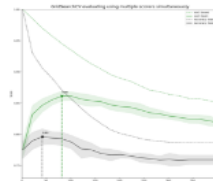
Examples

Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning

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



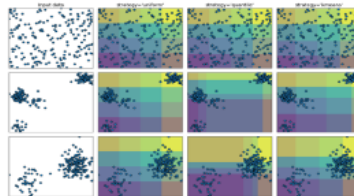
Examples

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>

```
from pycaret.time_series import *  
  
exp = TSForecastingExperiment()  
exp.setup(data = df, target='Passengers' , fh = 12, coverage=0.95)  
  
# comparing all models  
best = exp.compare_models()
```




- List of **Timeseries models** available in PyCaret
- This is the [Full] version.
- `pip install pycaret[full]`

```
from pycaret.time_series import *  
  
exp = TSForecastingExperiment()  
exp.setup(data = df, target='Passengers')  
  
# comparing all models  
best = exp.compare_models()
```

ID	Description
naive	Naive Forecaster
grand_means	Grand Means Forecaster
snaive	Seasonal Naive Forecaster
polytrend	Polynomial Trend Forecaster
arima	ARIMA (Autoregressive Integrated Moving Average)
auto_arima	Auto ARIMA
exp_smooth	Exponential Smoothing
ets	ETS (Error, Trend, Seasonality)
theta	Theta Forecaster
stlf	STLF (Seasonal and Trend decomposition using Loess)
croston	Croston's method for intermittent demand forecasting
lr_cds_dt	Linear Regression with Conditional Deseasonalizing and Detrending
en_cds_dt	Elastic Net with Conditional Deseasonalizing and Detrending
ridge_cds_dt	Ridge Regression with Conditional Deseasonalizing and Detrending
lasso_cds_dt	Lasso Regression with Conditional Deseasonalizing and Detrending
llar_cds_dt	Lasso Least Angular Regressor with Cond. Deseasonalizing and Detrending
br_cds_dt	Bayesian Ridge with Cond. Deseasonalizing and Detrending
huber_cds_dt	Huber Regressor with Cond. Deseasonalizing and Detrending
omp_cds_dt	Orthogonal Matching Pursuit with Cond. Deseasonalizing and Detrending
knn_cds_dt	K Neighbors Regressor with Cond. Deseasonalizing and Detrending
dt_cds_dt	Decision Tree Regressor with Cond. Deseasonalizing and Detrending
rf_cds_dt	Random Forest Regressor with Cond. Deseasonalizing and Detrending
et_cds_dt	Extra Trees Regressor with Cond. Deseasonalizing and Detrending
gbr_cds_dt	Gradient Boosting Regressor with Cond. Deseasonalizing and Detrending
ada_cds_dt	AdaBoost Regressor with Cond. Deseasonalizing and Detrending
xgboost_cds_dt	XGBoost Regressor with Cond. Deseasonalizing and Detrending
lightgbm_cds_dt	LightGBM Regressor with Cond. Deseasonalizing and Detrending
catboost_cds_dt	CatBoost Regressor with Cond. Deseasonalizing and Detrending

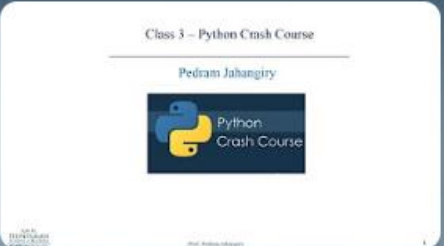


K Keras

- 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()
```

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

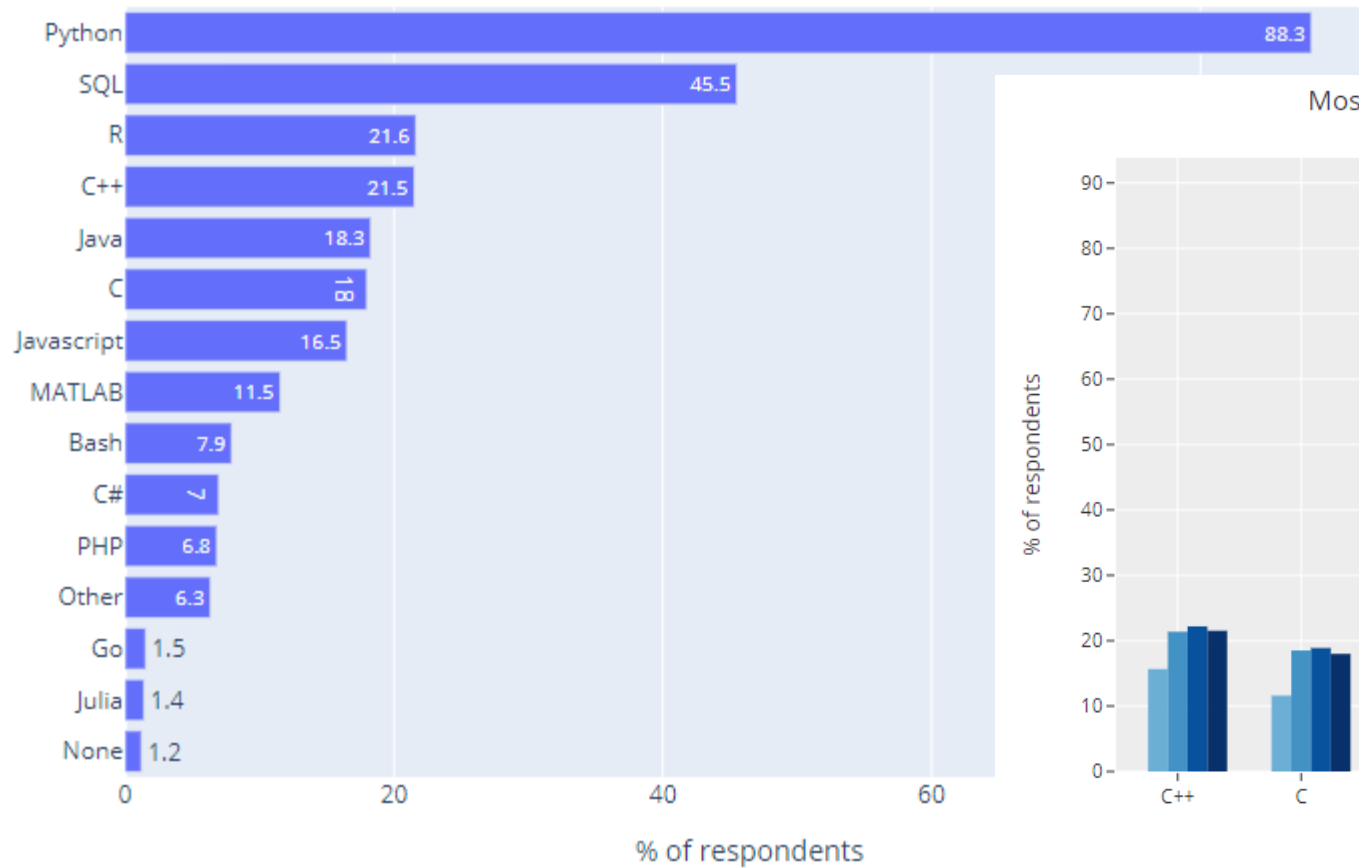
➔ Kaggle Survey 2022



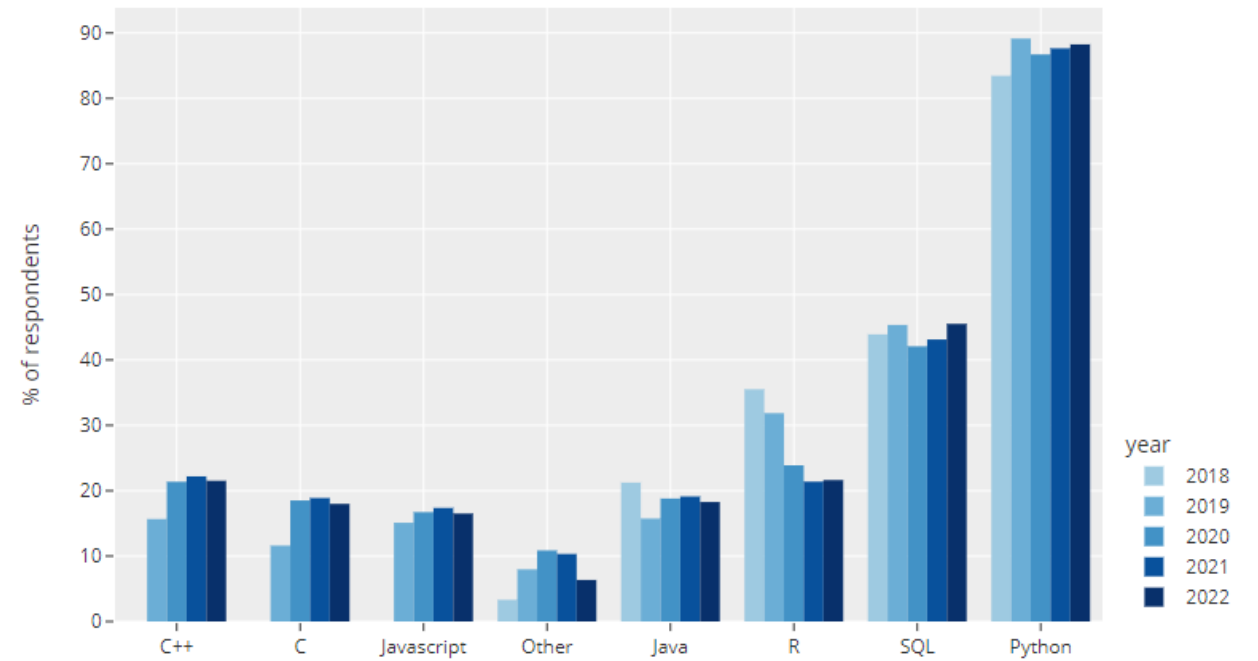
- Kaggle runs a yearly survey among machine learning and data science professionals worldwide.
- This survey is one of our **most reliable** sources about the **state of the industry!!!**

Programming Languages

Most Popular Programming Languages in 2022



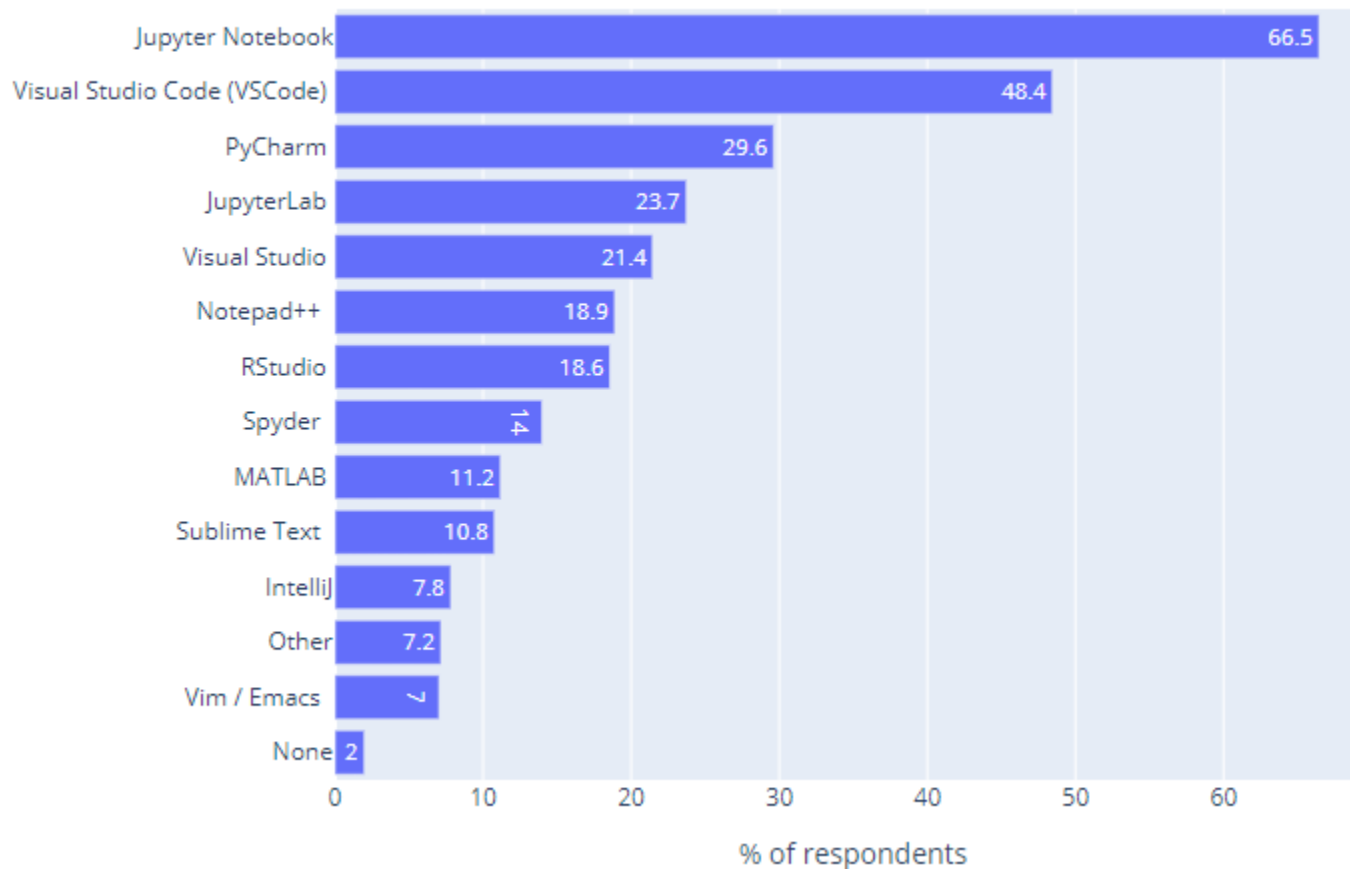
Most Popular Programming Languages 2018-2022





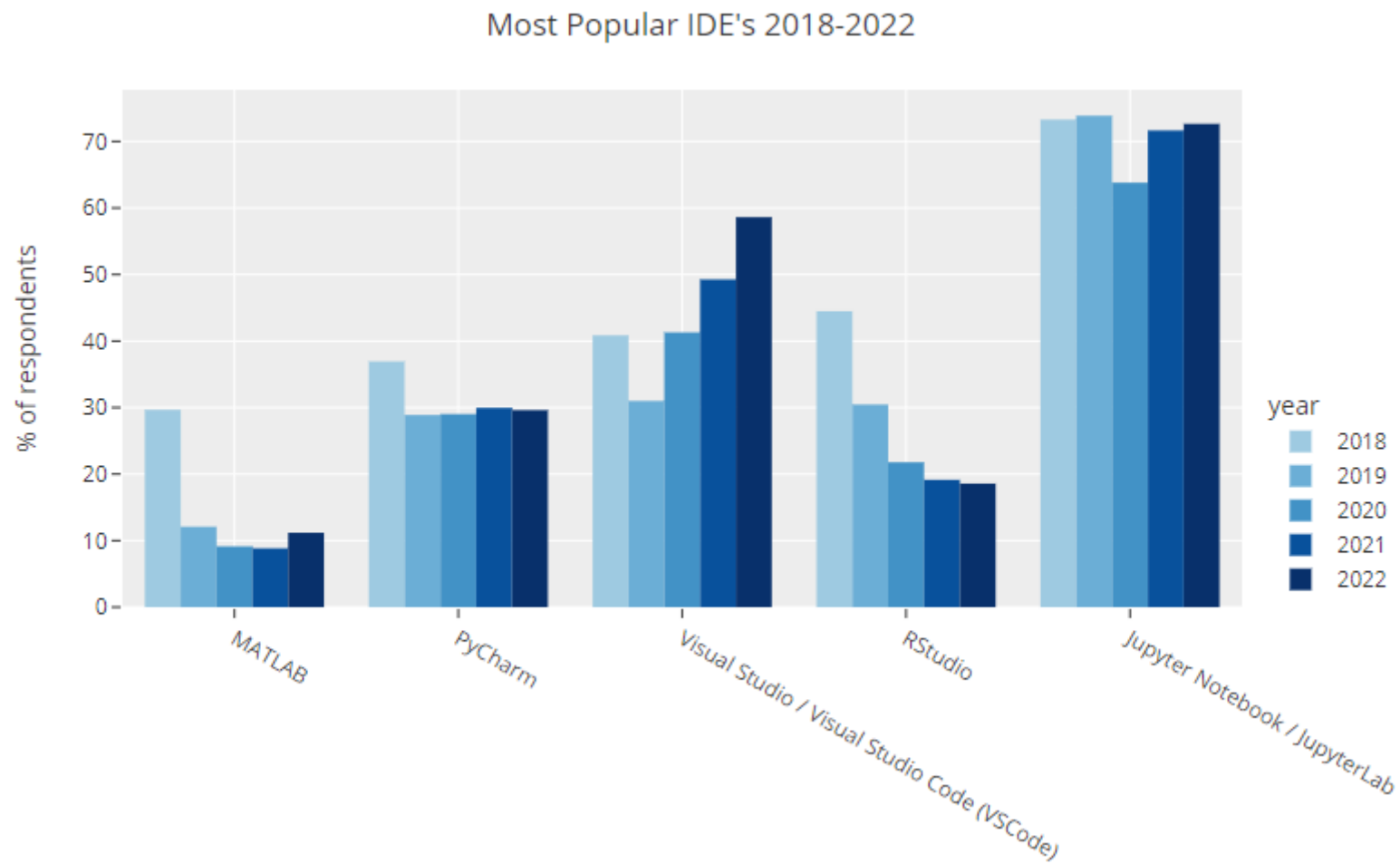
Popular IDE's

Most Popular IDE's in 2022












Popular IDE's





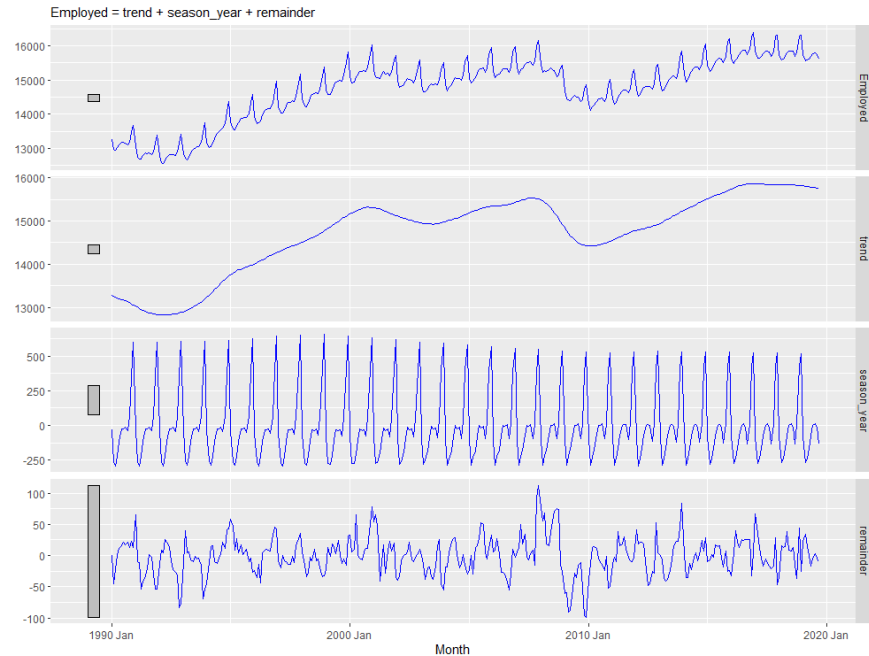
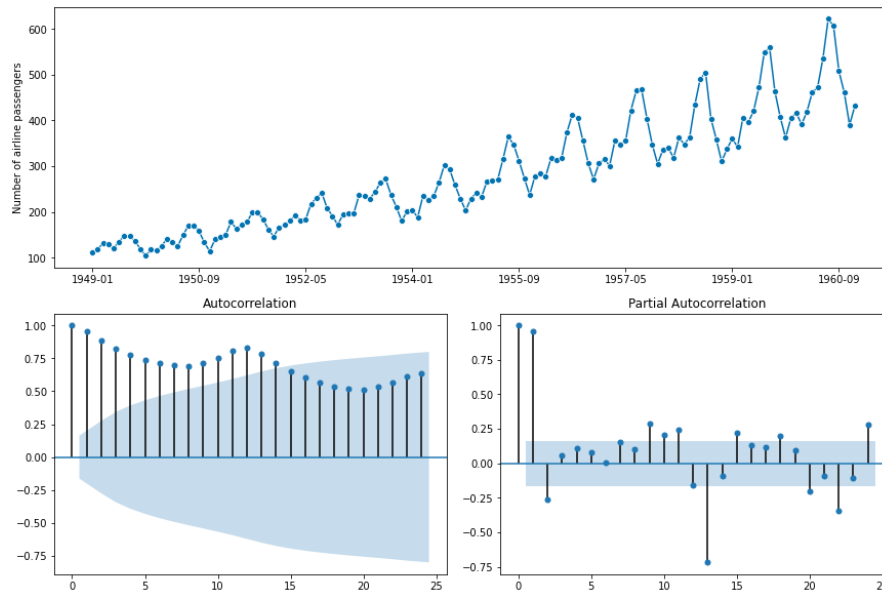
Python packages

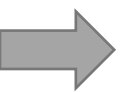
Package	Metrics	ML	DL	Focus
 statsmodels	✓✓✓			Statistics, Econometrics
 scikit-learn	✓	✓✓✓		General Machine Learning
 Keras			✓✓✓	General Deep Learning
 PYCARET	✓✓✓	✓✓✓		Auto ML, Rapid prototyping, Comparison
 Darts	✓✓✓	✓	✓✓	Advanced timeseries and forecasting
 GLUONS	✓	✓	✓✓	Large Scale probabilistic Models
 NIXTLA	✓✓✓	✓✓✓	✓✓✓	Workflows, SOTA methods



Module 2 – Part II

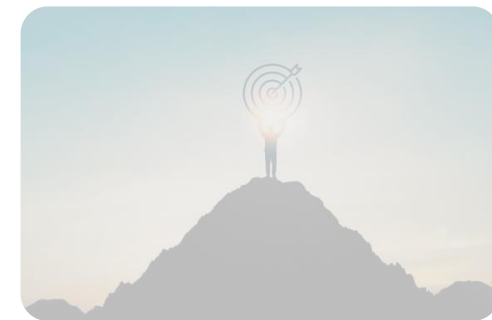
Setting up Deep Forecasting Environment (Time Series Basics in Python)





Road map!

- Module 1- Demystifying Timeseries Data and Modeling
- **Module 2- Setting up Deep Forecasting Environment (Python)**
- Module 3- Exponential Smoothing
- Module 4- ARIMA models
- Module 5- Machine Learning for Time series Forecasting
- Module 6- Deep Neural Networks
- Module 7- Deep Sequence Modeling (RNN, LSTM)
- Module 8- Prophet and Neural Prophet



➔ Road map!

- ✓ Module 1- Demystifying Timeseries Data and Modeling
- ✓ Module 2- Setting up Deep Forecasting Environment (Python)
- Module 3- Exponential Smoothing
- Module 4- ARIMA models
- Module 5- Machine Learning for Time series Forecasting
- Module 6- Deep Neural Networks
- Module 7- Deep Sequence Modeling (RNN, LSTM)
- Module 8- Prophet and Neural Prophet

