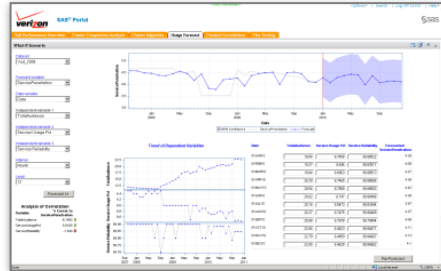


PROYECTO TÍPICO

En el ámbito empresarial los proyectos de Data Science suelen tener 3 componentes o fases

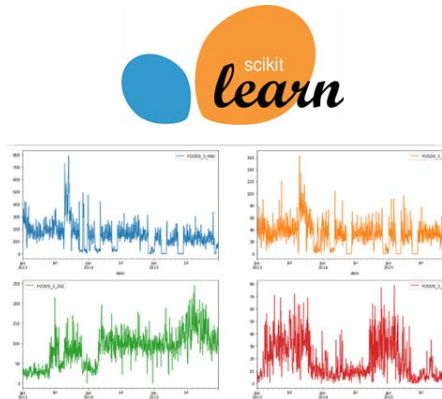
1

BUSINESS ANALYTICS



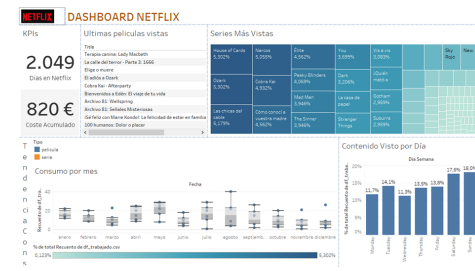
2

MACHINE LEARNING



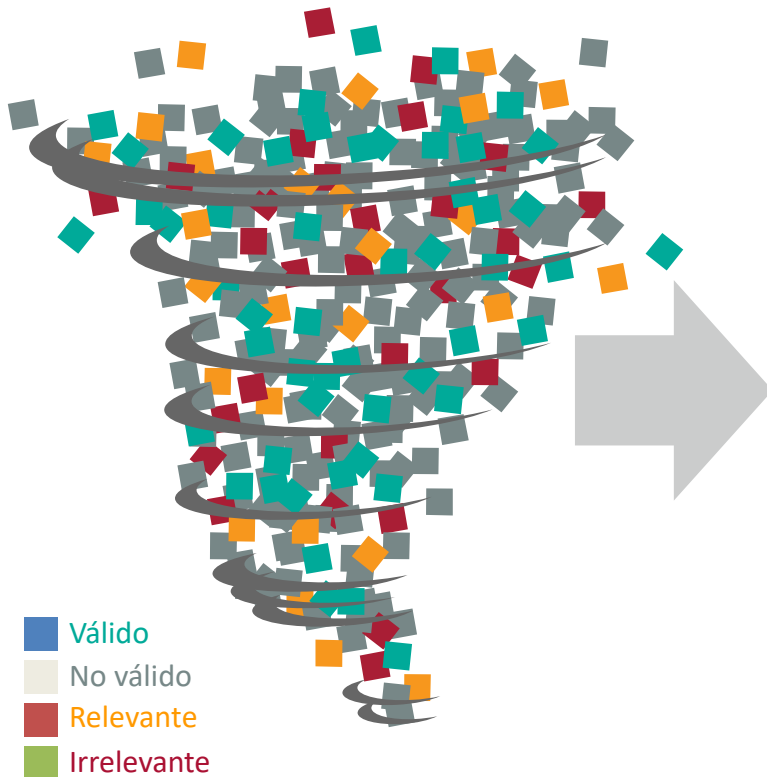
3

PRODUCTIVIZACIÓN

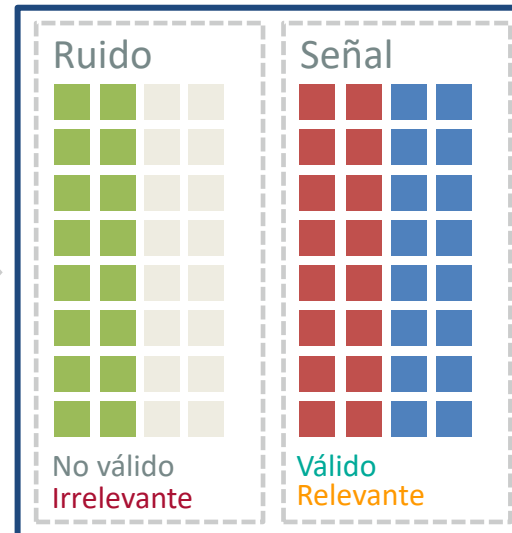


¿QUÉ HACE LA FASE DE BUSINESS ANALYTICS?

1. DATOS



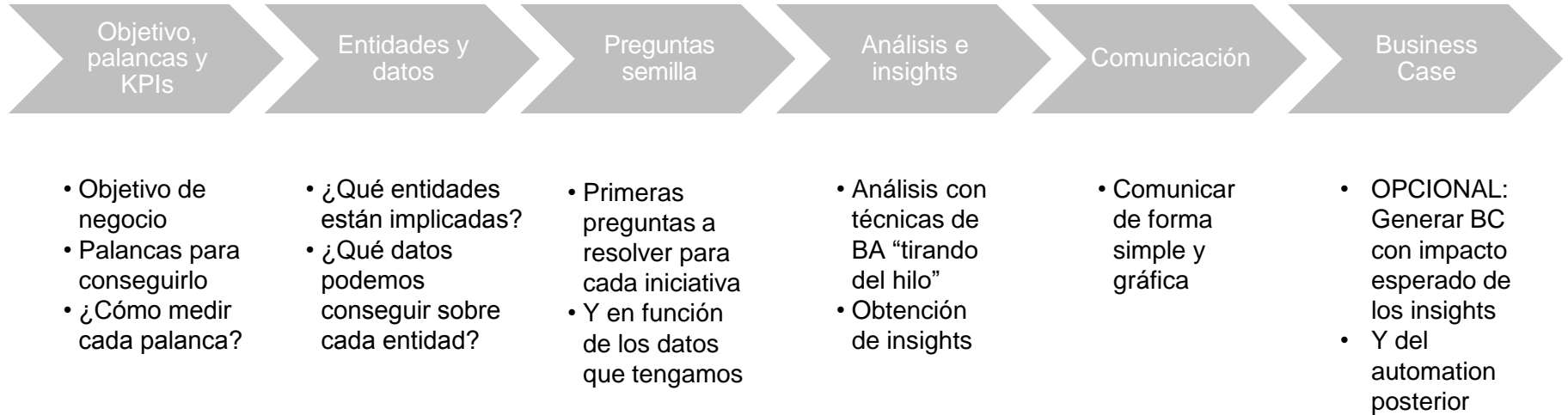
2. ANÁLISIS



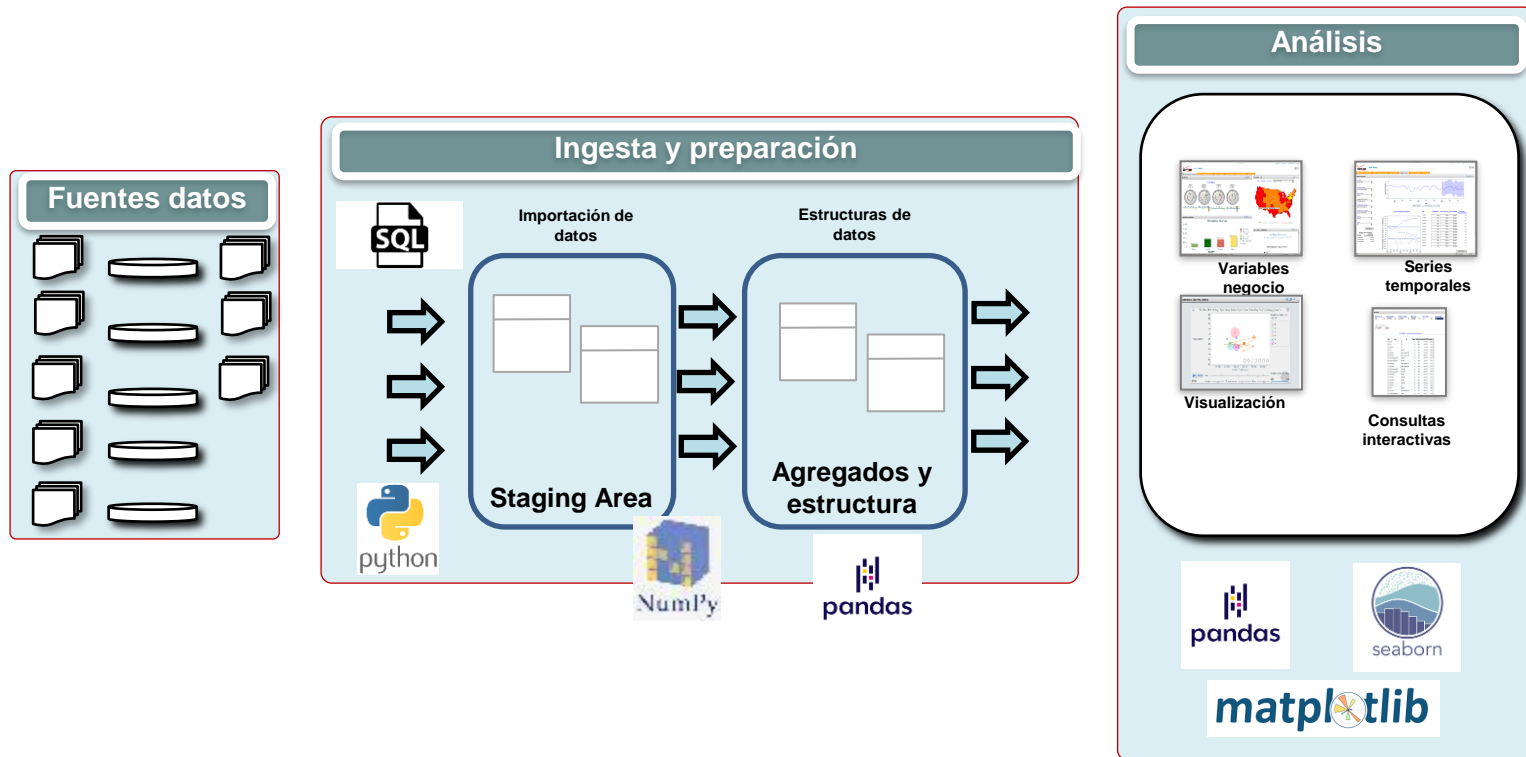
3. INSIGHTS



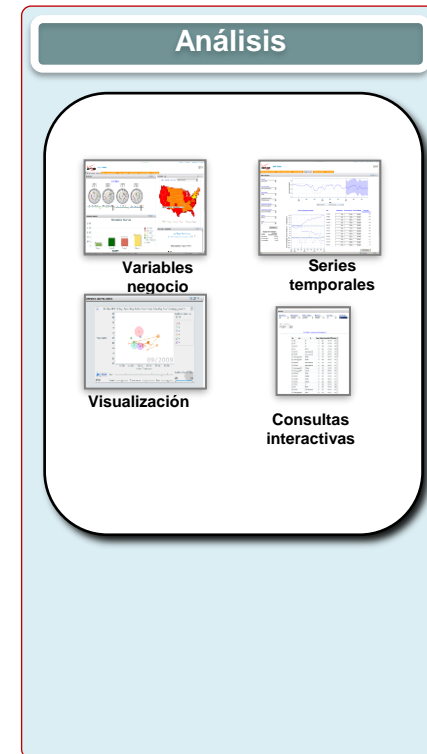
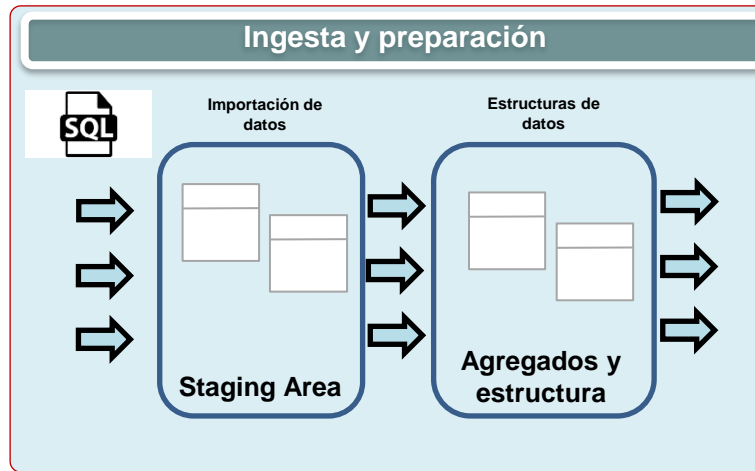
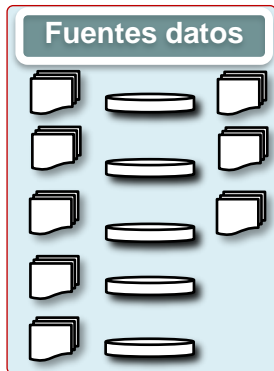
METODOLOGÍA DE LA FASE DE BUSINESS ANALYTICS



STACK TECNOLÓGICO PARA LA FASE DE BUSINESS ANALYTICS



STACK TECNOLÓGICO PARA LA FASE DE BUSINESS ANALYTICS

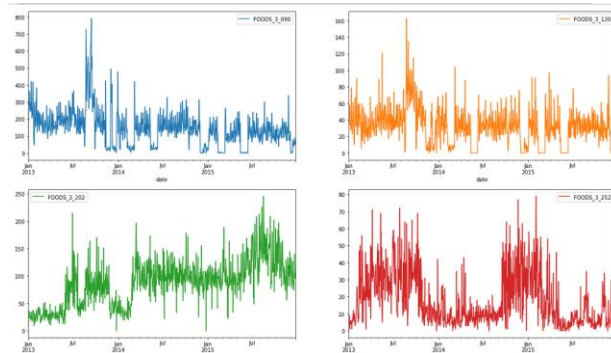


¿QUÉ HACE LA FASE DE MACHINE LEARNING?

BBDD 3 años
de histórico



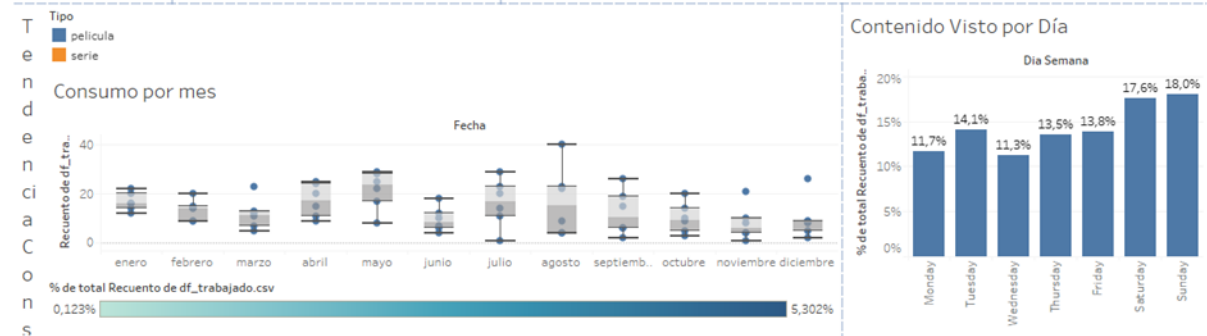
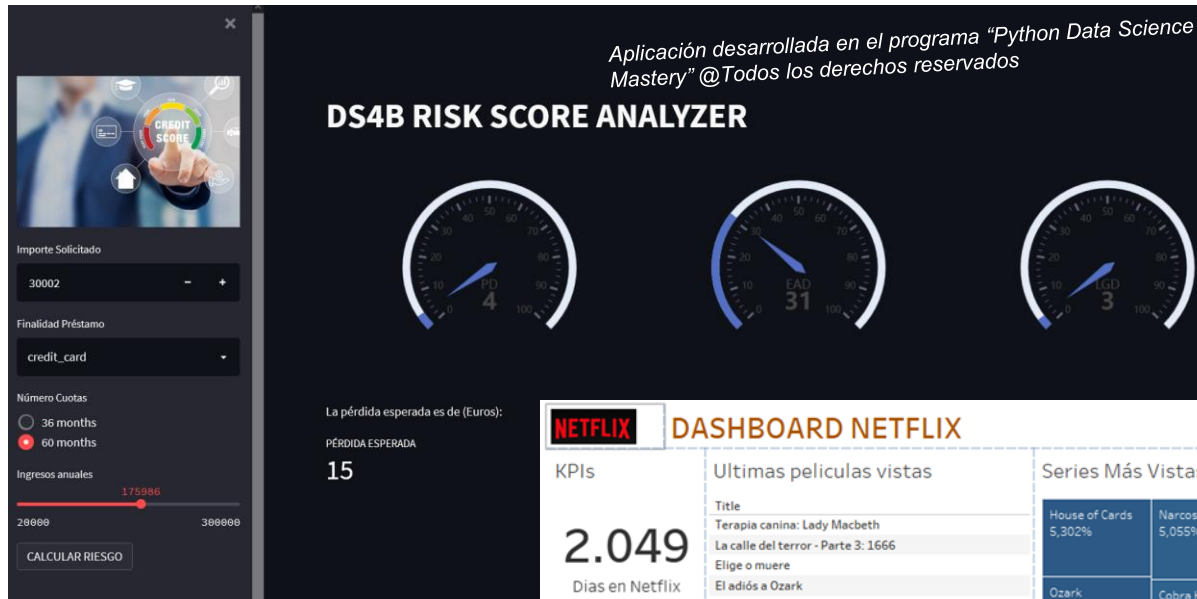
Modelos ML última generación



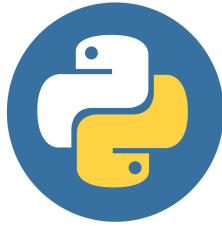
Predicciones nuevos datos



¿QUÉ HACE LA FASE DE PRODUCTIVIZACIÓN?

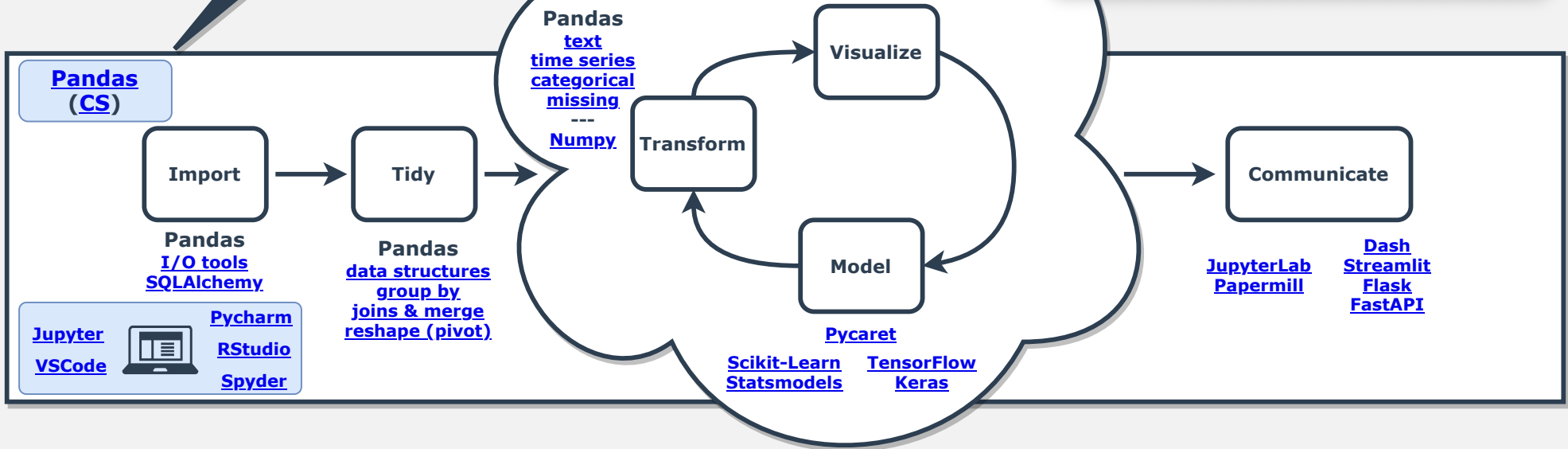
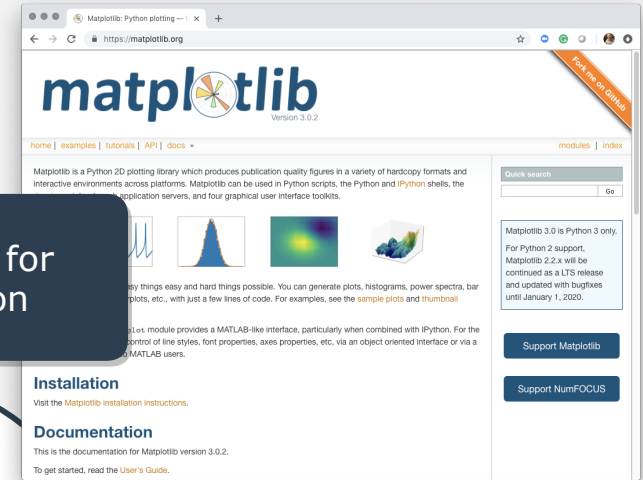


Data Science with Python Workflow



CS = Cheat Sheet

Click the links for Documentation



Important Resources

- Anaconda Distribution: <https://www.anaconda.com/download/>
- Python Documentation: <https://docs.python.org/>
- Python Standard Library: <https://docs.python.org/3/library>



Data Science with



Special Topics

Time Series Forecasting

- [sktime](#) - Scikit-Learn Extension for Time Series
- [statsmodels](#) - Time Series Analysis
- [GluonTS](#) - MXNet/Gluon Deep Learning for Time Series

Time Series Features

- [TSFresh](#) - Time Series Feature Engineering
- [tslearn](#) - Time Series Features
- [Pandas](#) Time Series
- [Arrow](#) - Human-Friendly Time

EDA

- [pandas-profiling](#), [SweetViz](#), [lux](#)

Web

- [beautifulsoup](#) - Extract data from HTML
- [requests-html](#) - HTML Parsing
- [scrapy](#) - Web crawling

MS Office & PDF

- [XlsxWriter](#) - Create Excel Workbooks
- [pyexcel](#) - Read/Write Excel
- [xlwings](#) - Call python from Excel
- [python-docx](#) - Word Documents
- [python-pptx](#) - PowerPoint Documents
- [pdfminer](#) - Text extraction from PDF
- [textextract](#) - Extract text from any document
- [PyPDF2](#) - Create PDF documents
- [gsppread](#) - Google Sheets

Text Analysis & NLP

- [NLTK](#) - Text Tokenization & Modeling
- [spaCy](#) - NLP using Cython for Speed
- [fuzzywuzzy](#) - Fuzzy String Matching

Recommendation Systems

- [Annoy](#) - Approximate Nearest Neighbors
- [LightFM](#) - Popular recommendation algo's.

Apps & APIs

- [FastAPI](#) - Web framework for building APIs in Python
- [Flask](#) - Web Development
- [Dash](#) & [Streamlit](#) - DS Web Frameworks

MLOps

- [Pycaret MLFlow Integration](#)
- [MLFlow](#) - Machine Learning Lifecycle, Tracking, Deployment
- [MetaFlow](#) - Scalable AWS Jobs for Data Scientists

Cloud

- [boto3](#) (AWS) - AWS Python SDK
- [Google Cloud](#) - GCP Python SDK
- [Azure](#) - Azure Python SDK

ETL & Automations

- [Airflow](#) - Workflow Scheduling & Monitoring
- [Luigi](#) - Batch Job Tool, Scheduling, Monitoring
- [Ansible](#) - Deployment Automation
- [JobLib](#) - Run python jobs

Machine Learning

- [Scikit-Learn](#) - ML in Python
- [H2O](#) - Scalable & AutoML
- [TPOT](#) - TPOT Automated ML Tool
- [PyCaret](#) - PyCaret Low Code ML
- [Dask ML](#) - Scalable ML with Dask
- ML Packages: [XGBoost](#), [LightGBM](#), [CatBoost](#)

Feature Engineering

- [Sklearn Data Transformations](#)
- [sklearn-pandas](#) - Sklearn Extension for Pandas
- [Featuretools](#) - Automated Feature Engineering
- [category_encoders](#) - Categorical Encoding
- [imbalanced-learn](#) - Resampling for Imbalanced
- [fancyimpute](#) - Extended imputation strategies

Deep Learning

- [TensorFlow](#) & [Keras](#)
- [PyTorch](#)
- [MXNet](#), [Gluon](#), & [GluonTS](#)
- [OpenAI Gym](#) - Reinforcement Learning

Image & Comp Vision

- [OpenCV](#) - Open Source Computer Vision
- [Scikit Image](#) - Image Processing
- [Pillow](#) - Python Imaging Library

Speed & Scale

- [datatable](#) - C++ Speed Up
- [Dask \(CS\)](#) - Parallel Pandas & Scikit Learn
- [RAPIDS \(CS\)](#) - GPU Accelerated Pandas
- [PySpark](#) - Spark Clusters
- [Optimus](#) - PySpark Extension for Humans

Coming from R?

- [R-to-Pandas Comparison](#)
- [siuba](#) & [plydata](#) - dplyr/tidyr ports
- [datatable](#) - data.table port
- [plotnine](#) - ggplot2 port