Sample size calculator

<https://www.surveysystem.com/sscalc.htm>

Deep Learning book

(Part 1 will give you a great intro to the math you will need)

<http://www.deeplearningbook.org/>

Types of error

|  |  |
| --- | --- |
| TRUE POSITIVE | FALSE NEGATIVE (TYPE II) |
| FALSE POSITIVE (TYPE I) | TRUE NEGATIVE |

Jupyter Notebook Markup Formatting

<https://help.github.com/en/articles/basic-writing-and-formatting-syntax>

Numpy Documentation

<https://www.numpy.org/>

SciPy Documentation

<https://www.scipy.org/>

<https://docs.scipy.org/doc/scipy/reference/>

<https://scipy-lectures.org/>

Examples of signal processing with SciPy

<https://www.kaggle.com/xhlulu/exploring-signal-processing-with-scipy>

<https://docs.scipy.org/doc/scipy/reference/tutorial/signal.html>

Pandas

<https://pandas.pydata.org/>

<http://pandas.pydata.org/pandas-docs/stable/user_guide/index.html>

Scikit-Learn

<https://scikit-learn.org/stable/>

scikit-learn Tutorials¶

<https://scikit-learn.org/stable/tutorial/>

Sci-kit Learn Supervised Learning Algorithms

<https://scikit-learn.org/stable/supervised_learning.html>

Sci-kit Learn Unsupervised Learning Algorithms

[https://scikit-learn.org/stable/supervised\_learning.html](https://scikit-learn.org/stable/unsupervised_learning.html)

The Linear Separability Problem: Some Testing Methods

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.6481&rep=rep1&type=pdf>

How to choose the right estimator (model)

<https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html>

Performance Metrics (model evaluation)

<https://scikit-learn.org/stable/modules/model_evaluation.html>

Regression Sample Code with Scikit-learn

<https://www.kaggle.com/bsivavenu/house-price-calculation-methods-for-beginners>

<https://www.kaggle.com/sangwookchn/regression-techniques-using-scikit-learn>

Classification Sample Code with Scikit-learn

<https://www.kaggle.com/jeffd23/10-classifier-showdown-in-scikit-learn>

<https://www.kaggle.com/gautham11/building-a-scikit-learn-classification-pipeline>

NLP Sample Code with Scikit-learn

<https://www.kaggle.com/agaleana/predicting-south-park-dialogues>

<https://github.com/abdulfatir/twitter-sentiment-analysis>

XGBoost, CatBoost and LightGBM

<https://xgboost.readthedocs.io/en/latest/python/index.html>

<https://catboost.ai/docs/concepts/python-quickstart.html>

<https://lightgbm.readthedocs.io/en/latest/Python-Intro.html>

Principal Component Analysis (PCA)

<https://towardsdatascience.com/a-step-by-step-explanation-of-principal-component-analysis-b836fb9c97e2>

<https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/>

Natural Language Processing with Python

<http://www.nltk.org/book/>

Time Series

<https://towardsdatascience.com/an-end-to-end-project-on-time-series-analysis-and-forecasting-with-python-4835e6bf050b>

<https://medium.com/datadriveninvestor/time-series-analysis-with-python-f5ab388b865a>

Multivariate: [https://www.analyticsvidhya.com/blog/2018/09/multivariate-time-series-guide-forecasting-modeling-python-codes](https://www.analyticsvidhya.com/blog/2018/09/multivariate-time-series-guide-forecasting-modeling-python-codes/?utm_source=DataCamp.com&utm_medium=Community&utm_campaign=News)

<http://barnesanalytics.com/analyzing-multivariate-time-series-using-arimax-in-python-with-statsmodels>

Multivariate with deep learning: <https://www.kaggle.com/lokeshkumarn/timeseries-multivariate>

**Data Visualization**

[https://python-graph-gallery.com](https://python-graph-gallery.com/)

Matplotlib Pyplot tutorial

<https://matplotlib.org/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py>

Seaborn tutorial

<http://seaborn.pydata.org/tutorial.html>

Bokeh tutorials

<https://bokeh.pydata.org/en/latest/docs/user_guide.html>

<https://www.kaggle.com/kanncaa1/visualization-bokeh-tutorial-part-1>

<https://www.kaggle.com/kanncaa1/interactive-bokeh-tutorial-part-2>

Plot.ly user guides

<https://plot.ly/python/>

<https://plot.ly/python/ipython-notebook-tutorial/>

Data Visualization Samples

<https://www.kaggle.com/biphili/seaborn-matplotlib-plot-to-visualize-iris-data>

<https://www.kaggle.com/chandraroy/plotting-with-pandas-matplotlib-and-seaborn>

Public Datasets

<https://scikit-learn.org/stable/datasets/index.html>

<https://www.kaggle.com/>

<https://toolbox.google.com/datasetsearch>

<https://www.data.gov/>

<https://data.gov.in/>

Hyperparameter\_optimization

<https://en.m.wikipedia.org/wiki/Hyperparameter_optimization>

<http://hyperopt.github.io/hyperopt-sklearn/>

<https://epistasislab.github.io/tpot/>

Feature engineering libraries

<https://feature-engine.readthedocs.io/en/latest/>

<https://www.featuretools.com/>

**End to end DS/ML Project:**

(sample code: <https://github.com/ageron/handson-ml2/blob/master/02_end_to_end_machine_learning_project.ipynb>)

1. Understand problem
   * Define objective
   * How will the solution be used
   * What solutions do you have in place
   * How will you measure performance
   * Minimum performance required
2. Prepare environment
   * Lock library versions
   * Create environment
     1. <https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>
     2. <https://uoa-eresearch.github.io/eresearch-cookbook/recipe/2014/11/20/conda/>
3. Get data
   * Find data and document sources
   * Check for space
   * Check for terms and conditions
   * If applicable: get access
   * Create workspace
   * Get data
   * Deal with sensitive information (delete, protect, anonymize)
   * Sample test set
4. Explore data (gain insights)\*
   * Create copy for exploration
   * Understand data attributes and characteristics
   * For supervised: identify target
   * Visualize
   * Look for correlations
5. Prepare your data (create pipeline)

**Important note:** split your dataset in train and test first.

* + Feature engineering
    1. Missing Values
       1. Numerical - mean, median, mode
       2. Categorical - “**missing**”, mode
    2. Categorical (Label Encoding)
       1. Rare values ---> “rare”
       2. Categorical to numerical
    3. Outliers (*can be skipped for Tree-based algorithms*)
    4. Scaling (*can be skipped for Tree-based algorithms*)
    5. Normalization (*can be skipped for Tree-based algorithms)*
    6. Dimensionality reduction (*for viz, reduce overfitting, reduce compute*)
       1. LDA
       2. PCA
       3. tSNE
       4. IsoMap
  + Quantitative Feature selection (if applicable)

1. Try different ML algorithms *(*[*https://mlflow.org/*](https://mlflow.org/) *can be used to track models’ performance and hyper parameters)*
   * Try standard hyper-params first
   * Measure and compare performance
   * Short-list top 3 algo’s
2. Fine-tune you model
   * Cross-validation
   * **GridSearchCV**, **RandomSearchCV**, **Bayes Search** (Hyperopt), **Genetic Algorithms** (TPOT)
   * For final model: measure performance on test and don’t tweak it
3. Present solution
   * Document steps
   * Create a super cool presentation
     1. Focus on the big picture problem statement
   * Explain why your approach aligns to objectives
   * If applicable (list assumptions)
   * Use a lot of visualizations (with easy to remember statements)
4. Launch/Deploy
   * Create pipeline
     1. Transformers\*\* (to prepare new data for ingestion)
        1. Imputers for missing values
        2. Encoder for categorical to numerical
        3. Scalers for data scaling (can be skipped in tree-based algorithms)
        4. Normalize (can be skipped in tree-based algorithms)
        5. Dim reduction
     2. Model (to make predictions)
   * Get solution ready for production
   * Monitor
   * Re-train your models (sample code: <https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/>)

\* *Look for:*

*size of dataset --->*

*number of variables --->*

*types of data --->*

*Cardinality ---> if cardinality = number of samples, remove such variable*

*Rare values --->*

*distributions (normal) --->*

*Correlation / collinearity --->*

*Outliers --->*

*Scale --->*

*missing values --->*

*Relation vs target variable (for supervised learning)*

\*\* Training transformers ---> can remove observations for the sake of a better model

Test transformers ---> won’t remove any observations

**Recommender Systems**

**Sample code**

<https://github.com/PacktPublishing/Hands-On-Recommendation-Systems-with-Python>

<https://www.kaggle.com/ibtesama/getting-started-with-a-movie-recommendation-system>

What;s next?

1. Decide what you want to pursue
   1. Go after DS, MLE, AI Eng
2. Specialize
   1. Computer vision
   2. Finanze
   3. NLU
   4. Deep belief Networks (drug creation)
   5. etc

Other ML approaches:

* Deep Learning
* Evolutionary Algorithms
* Reinforcement Learning

-- Deep reinforcement learning

-- Boltzmann machines

-- [Transfer Learning](https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a)

**End to end Machine Learning**

Building a Reproducible ML Pipeline

<https://arxiv.org/ftp/arxiv/papers/1810/1810.04570.pdf>

Django

[https://www.codementor.io/jadianes/...earning-clustering-user-preferences-du107s5mk](https://www.codementor.io/jadianes/build-data-products-django-machine-learning-clustering-user-preferences-du107s5mk)

Flask

<https://hackernoon.com/deploy-a-machine-learning-model-using-flask-da580f84e60c>

Google Cloud

[https://codelabs.developers.google.com/codelabs/end-to-end-ml/index.html](https://codelabs.developers.google.com/codelabs/end-to-end-ml/index.html?index=..%2F..index#0)

<https://cloud.google.com/ml-engine/docs/scikit/using-pipelines>

MLFlow serving

<https://mlflow.org/docs/latest/models.html#deploy-mlflow-models>

Hadoop and Spark algorithms

<https://www.amazon.com/Data-Algorithms-Recipes-Scaling-Hadoop/dp/1491906189>

Cloudera Hadoop QuickStart VM

<https://www.cloudera.com/downloads/quickstart_vms/5-13.html>

VMware Workstation Player Download

<https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html>

VMware Fusion Download

<https://www.vmware.com/products/fusion/fusion-evaluation.html>

Apache Spark ML Library

<http://spark.apache.org/docs/2.0.0/api/python/pyspark.mllib.html>

**Recommended reading:**

***Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems***

by Aurélien Géron | Oct 15, 2019

***Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, 3rd Edition***

by Sebastian Raschka (Author), Vahid Mirjalili (Author) | December 12, 2019

***Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence (Addison-Wesley Data & Analytics Series)***

by Jon Krohn , Grant Beyleveld , et al. | Sep 28, 2019

***Python Crash Course, 2nd Edition: A Hands-On, Project-Based Introduction to Programming***

by Eric Matthes (Author) | May 3, 2019

***A Smarter Way to Learn Python: Learn it faster. Remember it longer.***

by Mark Myers (Author) | August 9, 2017

***Hands-On Recommendation Systems with Python***

By Rounak Banik (Author) | July 30, 2018

**Additional resources**

<https://machinelearningmastery.com/>