

-
- machine-learning
 - Documentation Listings
 - * deep learning
 - * model deployment
 - * operation research
 - * reinforcement learning
 - * ad
 - * search
 - * time series
 - * projects
 - * ab tests
 - * model selection
 - * dim reduct
 - * recsys
 - * trees
 - * clustering
 - * keras
 - * text classification
 - * regularization
 - * networkx
 - * association rule
 - * big data
 - * data science is software
 - * ga
 - * unbalanced
 - * clustering old
 - * linear regression
 - Python Programming

machine-learning



This is a continuously updated repository that documents personal journey on learning data science, machine learning related topics.

Goal: Introduce machine learning contents in Jupyter Notebook format. The content aims to strike a good balance between mathematical notations, educational implementation from scratch

using Python's scientific stack including numpy, numba, scipy, pandas, matplotlib, pyspark etc. and open-source library usage such as scikit-learn, fasttext, huggingface, onnx, xgboost, lightgbm, pytorch, keras, tensorflow, gensim, h2o, ortools, ray tune etc.

Documentation Listings

deep learning

Curated notes on deep learning.

- Softmax Regression from scratch. [\[nbviewer\]](#)[\[html\]](#)
- Softmax Regression - Tensorflow hello world. [\[nbviewer\]](#)[\[html\]](#)
- Multi-layers Neural Network - Tensorflow. [\[nbviewer\]](#)[\[html\]](#)
- Convolutional Neural Network (CNN) - Tensorflow. [\[nbviewer\]](#)[\[html\]](#)
- Recurrent Neural Network (RNN).
 - Vanilla RNN - Tensorflow. [\[nbviewer\]](#)[\[html\]](#)
 - Long Short Term Memory (LSTM) - Tensorflow. [\[nbviewer\]](#)[\[html\]](#)
 - RNN, LSTM - PyTorch hello world. [\[nbviewer\]](#)[\[html\]](#)
- Word2vec (skipgram + negative sampling) using Gensim. [\[nbviewer\]](#)[\[html\]](#)
- Sequence to Sequence Neural Network (Seq2Seq).
 - Seq2Seq for German to English Machine Translation - PyTorch. Includes quick intro to torchtext [\[nbviewer\]](#)[\[html\]](#)
 - Seq2Seq with Attention for German to English Machine Translation - PyTorch. [\[nbviewer\]](#)[\[html\]](#)
- Subword Tokenization.
 - Byte Pair Encoding (BPE) from scratch and quick walkthrough of sentencepiece. [\[nbviewer\]](#)[\[html\]](#)
- Fasttext.
 - Multi-Label Text Classification with Fasttext and Huggingface Tokenizers. [\[nbviewer\]](#)[\[html\]](#)
 - Product Quantization for Model Compression. [\[nbviewer\]](#)[\[html\]](#)
 - Approximate Nearest Neighborhood Search with Navigable Small World. [\[nbviewer\]](#)[\[html\]](#)
- Graph Neural Network (GNN).
 - Quick Introduction to Graph Neural Network Node Classification Task (DGL, GraphSAGE). [\[nbviewer\]](#)[\[html\]](#)

-
- Transformer.
 - Transformer, Attention is All you Need - PyTorch, Huggingface Datasets. [nbviewer][html]
 - Machine Translation with Huggingface Transformers mT5. [nbviewer][html]
 - Fine Tuning Pre-trained Encoder on Question Answer Task. [nbviewer][html]
 - Training Bi-Encoder Models with Contrastive Learning Notes. [nbviewer][html]
 - Sentence Transformer: Training Bi-Encoder via Contrastive Loss. [nbviewer][html]
 - Introduction to CLIP (Contrastive Language-Image Pre-training), LiT, ViT [nbviewer][html]
 - Self Supervised (SIMCLR) versus Supervised Contrastive Learning. [nbviewer][html]
 - Multilingual Sentence Embedding with LLM and PEFT LoRA (PyTorch Lightning) [nbviewer][html]
 - Tabular
 - Deep Learning for Tabular Data - PyTorch. [nbviewer][html]
 - Deep Learning - Learning to Rank 101 (RankNet, ListNet). [nbviewer][html]
 - BERT CTR. [nbviewer][html]

model deployment

- FastAPI & Azure Kubernetes Cluster. End to end example of training a model and hosting it as a service. [folder]
- Quick Intro to Gradient Boosted Tree Inferencing. [nbviewer][html]
- Speeding Up Transformers Inferencing. [folder]
- Working with AWS (Amazon Web Services). [folder]

operation research

- Operation Research Quick Intro Via Ortools. [nbviewer][html]

reinforcement learning

- Introduction to Multi-armed Bandits. [nbviewer][html]

ad

Notes related to advertising domain.

- Quick introduction to generalized second price auction. [nbviewer][html]

search

Information Retrieval, some examples are demonstrated using ElasticSearch.

- Introduction to BM25 (Best Match). [\[nbviewer\]](#)[\[html\]](#)

time series

Forecasting methods for timeseries-based data.

- Getting started with time series analysis with Exponential Smoothing (Holt-Winters). [\[nbviewer\]](#)[\[html\]](#)
- Framing time series problem as supervised-learning. [\[nbviewer\]](#)[\[html\]](#)
- First Foray Into Discrete/Fast Fourier Transformation. [\[nbviewer\]](#)[\[html\]](#)

projects

End to end project including data preprocessing, model building.

- Kaggle: Rossman Store Sales Predicting daily store sales. Also introduces deep learning for tabular data. [\[folder\]](#)
- Kaggle: Quora Insincere Questions Classification Predicting insincere questions. [\[folder\]](#)

ab tests

A/B testing, a.k.a experimental design. Includes: Quick review of necessary statistic concepts. Methods and workflow/thought-process for conducting the test and caveats to look out for.

- Frequentist A/B testing (includes a quick review of concepts such as p-value, confidence interval). [\[nbviewer\]](#)[\[html\]](#)
- Quantile Regression and its application in A/B testing.
 - Quick Introduction to Quantile Regression. [\[nbviewer\]](#)[\[html\]](#)
 - Quantile Regression's application in A/B testing. [\[nbviewer\]](#)[\[html\]](#)
- Casual Inference
 - Propensity Score Matching. [\[nbviewer\]](#)[\[html\]](#)
 - Inverse Propensity Weighting. [\[nbviewer\]](#)[\[html\]](#)
 - Quick introduction to difference in difference. [\[nbviewer\]](#)[\[html\]](#)

model selection

Methods for selecting, improving, evaluating models/algorithms.

- K-fold cross validation, grid/random search from scratch. [\[nbviewer\]\[html\]](#)
- AUC (Area under the ROC curve and precision/recall curve) from scratch (includes the process of building a custom scikit-learn transformer). [\[nbviewer\]\[html\]](#)
- Evaluation metrics for imbalanced dataset. [\[nbviewer\]\[html\]](#)
- Detecting collinearity amongst features (Variance Inflation Factor for numeric features and Cramer's V statistics for categorical features), also introduces Linear Regression from a Maximum Likelihood perspective and the R-squared evaluation metric. [\[nbviewer\]\[html\]](#)
- Curated tips and tricks for technical and soft skills. [\[nbviewer\]\[html\]](#)
- Partial Dependence Plot (PDP), model-agnostic approach for directional feature influence. [\[nbviewer\]\[html\]](#)
- Kullback-Leibler (KL) Divergence. [\[nbviewer\]\[html\]](#)
- Probability Calibration for classification models with Platt Scaling, Histogram Binning, Isotonic Regression. [\[nbviewer\]\[html\]](#)
- Probability Calibration for deep learning classification models with Temperature Scaling. [\[nbviewer\]\[html\]](#)
- HyperParameter Tuning with Ray Tune and Hyperband. [\[nbviewer\]\[html\]](#)

dim reduct

Dimensionality reduction methods.

- Principal Component Analysis (PCA) from scratch. [\[nbviewer\]\[html\]](#)
- Introduction to Singular Value Decomposition (SVD), also known as Latent Semantic Analysis/Indexing (LSA/LSI). [\[nbviewer\]\[html\]](#)

recsys

Recommendation system with a focus on matrix factorization methods. Starters into the field should go through the first notebook to understand the basics of matrix factorization methods.

- Alternating Least Squares with Weighted Regularization (ALS-WR) from scratch. [\[nbviewer\]\[html\]](#)
- ALS-WR for implicit feedback data from scratch & Mean Average Precision at k (mapk) and Normalized Cumulative Discounted Gain (ndcg) evaluation. [\[nbviewer\]\[html\]](#)
- Bayesian Personalized Ranking (BPR) from scratch & AUC evaluation. [\[nbviewer\]\[html\]](#)
- WARP (Weighted Approximate-Rank Pairwise) Loss using lightfm. [\[nbviewer\]\[html\]](#)

-
- Factorization Machine from scratch. [\[nbviewer\]\[html\]](#)
 - Content-Based Recommenders:
 - (Text) Content-Based Recommenders. Introducing Approximate Nearest Neighborhood (ANN) - Locality Sensitive Hashing (LSH) for cosine distance from scratch. [\[nbviewer\]\[html\]](#)
 - Approximate Nearest Neighborhood (ANN):
 - Benchmarking ANN implementations (nmslib). [\[nbviewer\]\[html\]](#)
 - Calibrated Recommendation for reducing bias/increasing diversity in recommendation. [\[nbviewer\]\[html\]](#)
 - Maximum Inner Product for Speeding Up Generating Recommendations. [\[nbviewer\]\[html\]](#)

trees

Tree-based models for both regression and classification tasks.

- Decision Tree from scratch. [\[nbviewer\]\[html\]](#)
- Random Forest from scratch and Extra Trees. [\[nbviewer\]\[html\]](#)
- Gradient Boosting Machine (GBM) from scratch. [\[nbviewer\]\[html\]](#)
- Xgboost API walkthrough (includes hyperparameter tuning via scikit-learn like API). [\[nbviewer\]\[html\]](#)
- LightGBM API walkthrough and a discussion about categorical features in tree-based models. [\[nbviewer\]\[html\]](#)
- Monotonic Constraint with Boosted Tree. [\[nbviewer\]\[html\]](#)

clustering

TF-IDF and Topic Modeling are techniques specifically used for text analytics.

- TF-IDF (text frequency - inverse document frequency) from scratch. [\[nbviewer\]\[html\]](#)
- K-means, K-means++ from scratch; Elbow method for choosing K. [\[nbviewer\]\[html\]](#)
- Gaussian Mixture Model from scratch; AIC and BIC for choosing the number of Gaussians. [\[nbviewer\]\[html\]](#)
- Topic Modeling with gensim's Latent Dirichlet Allocation(LDA). [\[nbviewer\]\[html\]](#)

keras

For those interested there's also a keras cheatsheet that may come in handy.

-
- Multi-layers Neural Network (keras basics). [\[nbviewer\]\[html\]](#)
 - Multi-layers Neural Network hyperparameter tuning via scikit-learn like API. [\[nbviewer\]\[html\]](#)
 - Convolutional Neural Network (CNN)
 - Image classification basics. [\[nbviewer\]\[html\]](#)
 - Introduction to Residual Networks (ResNets) and Class Activation Maps (CAM). [\[nbviewer\]\[html\]](#)
 - Recurrent Neural Network (RNN) - language modeling basics. [\[nbviewer\]\[html\]](#)
 - Text Classification
 - Word2vec for Text Classification. [\[nbviewer\]\[html\]](#)
 - Leveraging Pre-trained Word Embedding for Text Classification. [\[nbviewer\]\[html\]](#)
 - Sentencepiece Subword tokenization for Text Classification. [\[nbviewer\]\[html\]](#)

text classification

Deep learning techniques for text classification are categorized in its own section.

- Building intuition with spam classification using scikit-learn (scikit-learn hello world). [\[nbviewer\]\[html\]](#)
- Bernoulli and Multinomial Naive Bayes from scratch. [\[nbviewer\]\[html\]](#)
- Logistic Regression (stochastic gradient descent) from scratch. [\[nbviewer\]\[html\]](#)
- Chi-square feature selection from scratch. [\[nbviewer\]\[html\]](#)

regularization

Building intuition on Ridge and Lasso regularization using scikit-learn.

- View [\[nbviewer\]\[html\]](#)

networkx

Graph library other than [networkx](#) are also discussed.

- PyCon 2016: Practical Network Analysis Made Simple. Quickstart to networkx's API. Includes some basic graph plotting and algorithms. [\[nbviewer\]\[html\]](#)
- Short Walkthrough of PageRank. [\[nbviewer\]\[html\]](#)
- Influence Maximization from scratch. Includes discussion on Independent Cascade (IC), Sub-modular Optimization algorithms including Greedy and Lazy Greedy, a.k.a Cost Efficient Lazy Forward (CELF) [\[nbviewer\]\[html\]](#)

association rule

Also known as market-basket analysis.

- Apriori from scratch. [\[nbviewer\]\[html\]](#)
- Using R's arules package (apriori) on tabular data. [\[Rmarkdown\]](#)

big data

Exploring big data tools, such as Spark and H2O.ai. For those interested there's also a [pyspark rdd cheatsheet](#) and [pyspark dataframe cheatsheet](#) that may come in handy.

- Local Hadoop cluster installation on Mac. [\[markdown\]](#)
- PySpark installation on Mac. [\[markdown\]](#)
- Examples of manipulating with data (crimes data) and building a RandomForest model with PySpark MLlib. [\[nbviewer\]\[html\]](#)
- PCA with PySpark MLlib. [\[nbviewer\]\[html\]](#)
- Tuning Spark Partitions. [\[nbviewer\]\[html\]](#)
- H2O API walkthrough (using GBM as an example). [\[nbviewer\]\[html\]](#)
- Spark MLlib Binary Classification (using GBM as an example). [\[raw zeppelin notebook\]\[Zepl\]](#)

data science is software

Best practices for doing data science in Python.

- View [\[nbviewer\]\[html\]](#)

ga

Genetic Algorithm. Math-free explanation and code from scratch.

- Start from a simple optimization problem and extending it to traveling salesman problem (tsp).
- View [\[nbviewer\]\[html\]](#)

unbalanced

Choosing the optimal cutoff value for logistic regression using cost-sensitive mistakes (meaning when the cost of misclassification might differ between the two classes) when your dataset consists of unbalanced binary classes. e.g. Majority of the data points in the dataset have a positive outcome, while

few have negative, or vice versa. The notion can be extended to any other classification algorithm that can predict class's probability, this documentation just uses logistic regression for illustration purpose.

- Visualize two by two standard confusion matrix and ROC curve with costs using ggplot2.
- View [Rmarkdown]

clustering old

A collection of scattered old clustering documents in R.

- Toy sample code of the LDA algorithm (gibbs sampling) and the topicmodels library. [Rmarkdown]
- k-shingle, Minhash and Locality Sensitive Hashing for solving the problem of finding textually similar documents. [Rmarkdown]
- Introducing tf-idf (term frequency-inverse document frequency), a text mining technique. Also uses it to perform text clustering via hierarchical clustering. [Rmarkdown]
- Some useful evaluations when working with hierarchical clustering and K-means clustering (K-means++ is used here). Including Calinski-Harabasz index for determine the right K (cluster number) for clustering and bootstrap evaluation of the clustering result's stability. [Rmarkdown]

linear regression

- Training Linear Regression with gradient descent in R, briefly covers the interpretation and visualization of linear regression's summary output. [Rmarkdown]

Python Programming

- Extremely Quick Guide to Unicode. [markdown]
- Quick Example of Factory Design Pattern. [nbviewer][html]
- Parallel programming with Python (threading, multiprocessing, concurrent.futures, joblib). [nbviewer][html]
- Understanding iterables, iterator and generators. [nbviewer][html]
- Cohort analysis. Visualizing user retention by cohort with seaborn's heatmap and illustrating pandas's unstack. [nbviewer][html]
- Logging module. [nbviewer][html]
- Data structure, algorithms from scratch. [folder]
- Cython and Numba quickstart for high performance Python. [nbviewer][html]

-
- Optimizing Pandas (e.g. reduce memory usage using category type). [\[nbviewer\]\[html\]](#)
 - Unittest. [\[Python script\]](#)
 - Using built-in data structure and algorithm. [\[nbviewer\]\[html\]](#)
 - Tricks with strings and text. [\[nbviewer\]\[html\]](#)
 - Python's decorators (useful script for logging and timing function). [\[nbviewer\]\[html\]](#)
 - Pandas's pivot table. [\[nbviewer\]\[html\]](#)
 - Quick introduction to classmethod, staticmethod and property. [\[nbviewer\]\[html\]](#)