**# Overview on Machine Learning and Tensorflow Framework -**

* <https://www.tensorflow.org/tutorials>
* <https://www.youtube.com/watch?v=njKP3FqW3Sk>
* <https://www.youtube.com/watch?v=tPYj3fFJGjk&t>
* <https://www.youtube.com/watch?v=7O60HOZRLng>
* <https://www.youtube.com/watch?v=uu1SV41IZ4Y>
* <https://youtube.com/playlist?list=PLeo1K3hjS3uvCeTYTeyfe0-rN5r8zn9rw>
  + (Basic intro to ML,Regression,Classification and feature engineering)

**# Pytorch -**

* <https://www.fast.ai/>

**# PyTorch vs Tensorflow -**

* <https://towardsdatascience.com/top-5-deep-learning-frameworks-to-watch-in-2021-and-why-tensorflow-98d8d6667351>
* [https://realpython.com/pytorch-vs-tensorflow/#conclusion](https://realpython.com/pytorch-vs-tensorflow/" \l "conclusion)
* <https://viso.ai/deep-learning/pytorch-vs-tensorflow/>
* <https://youtube.com/watch?v=NuJB-RjhMH4&feature=share>

**# DataSets -**

* <https://www.kaggle.com/datasets>
* <https://archive.ics.uci.edu/ml/datasets>
* <https://www.mygreatlearning.com/blog/dataset-in-machine-learning/>
* <https://serokell.io/blog/best-machine-learning-datasets>
* <https://www.dataquest.io/blog/free-datasets-for-projects/>

**# Statistics for Data Science -**

* <https://www.youtube.com/watch?v=Vfo5le26IhY>

**# Code Samples -**

* <https://keras.io/examples/>

**# Important Parameters -**

* Measure accuracy of Regression models -

<https://www.youtube.com/watch?v=mHuUD1QvANE>

* Loss function -

<https://www.youtube.com/watch?v=QBbC3Cjsnjg>

<https://www.youtube.com/watch?v=NJpABYQB9PI>

* Mean & Standard Deviation - <https://www.youtube.com/watch?v=SzZ6GpcfoQY>
* Choose no of Hidden Layers - <https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw>
* Epoch, Batch Size, Iterations - <https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9>
* Ideal no of epochs - <https://www.geeksforgeeks.org/choose-optimal-number-of-epochs-to-train-a-neural-network-in-keras/>
* Choose the ML Algorithm - <https://www.youtube.com/watch?v=38SUUaMX5Rg>
* Box Plot - [https://miro.medium.com/max/1400/1\*2c21SkzJMf3frPXPAR\_gZA.png](https://miro.medium.com/max/1400/1*2c21SkzJMf3frPXPAR_gZA.png)
* Deal Categorical Variables - <https://www.youtube.com/watch?v=WXHLLO4FnZs>
* https://towardsdatascience.com/master-the-most-hated-task-in-ds-ml-3b9779276d7c
* Scaling & Normalization - <https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02>
* Recipe for training neural networks - <http://karpathy.github.io/2019/04/25/recipe/>
* Linear and Non Linear Data - <https://www.youtube.com/watch?v=rfveCzGqvUA>
* K-Bins-Discretization - <https://www.youtube.com/watch?v=_I_aNLZHxng>
* K-fold cross validation - <https://www.youtube.com/watch?v=TIgfjmp-4BA>
* Momentum, RMSProp and Adam Optimizers - <https://www.youtube.com/watch?v=M2xkmc2oHUc>
* Gradient Descent and BackPropagation - <https://www.youtube.com/watch?v=odlgtjXduVg>
* Neural Network Hyperparameter tuning - <https://www.analyticsvidhya.com/blog/2018/11/neural-networks-hyperparameter-tuning-regularization-deeplearning/>
* Hyperparameter Tuning Techniques (GridSearch, RandomSearch, Hyperopt, topt, optuna) -

<https://www.youtube.com/watch?v=355u2bDqB7c>

<https://www.youtube.com/watch?v=ZrIcll1cMJs>

* Encoding Categorical Data -

<https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/>

* Dummy Variable Trap -

<https://www.youtube.com/watch?v=5Q69P5r2u2A>

* Curse of Dimensionality -

<https://www.youtube.com/watch?v=_4DaqzLyT08>

* Gini Index and Entropy -

<https://www.youtube.com/watch?v=-W0DnxQK1Eo>

* Ensemble Learning (Bagging & Boosting) -

<https://www.youtube.com/watch?v=m-S9Hojj1as>

* Multicollinearity -

<https://www.youtube.com/watch?v=ekuD8JUdL6M>

<https://towardsdatascience.com/targeting-multicollinearity-with-python-3bd3b4088d0b>

* Eigenvalues and EigenVectors in PCA -

<https://towardsdatascience.com/eigenvalues-and-eigenvectors-378e851bf372>

**# Other good blogs/videos -**

* AI - <https://deepmind.com/blog>
* Binary Classification - <https://www.youtube.com/watch?v=PM6uvCLyeXM>
* Regression Algorithms -

<https://towardsdatascience.com/7-of-the-most-commonly-used-regression-algorithms-and-how-to-choose-the-right-one-fc3c8890f9e3>

<https://towardsdatascience.com/whats-the-difference-between-linear-regression-lasso-ridge-and-elasticnet-8f997c60cf29>

* Car Auction Regression problem - <https://medium.com/@chenlemuge/a-houswifes-journey-toward-data-scientist-d1354288e9ac>
* Machine Learning - <https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471>
* CNN - <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

<https://cs231n.github.io/neural-networks-case-study/>

<https://www.youtube.com/watch?v=Y1qxI-Df4Lk&t=2s>

<https://stackoverflow.com/questions/36243536/what-is-the-number-of-filter-in-cnn>

<https://www.youtube.com/watch?v=aircAruvnKk>

* Data Preprocessing - <https://towardsdatascience.com/data-pre-processing-techniques-you-should-know-8954662716d6>

<https://www.youtube.com/watch?v=V0u6bxQOUJ8>

* Electricity Demand - <https://www.sciencedirect.com/science/article/pii/S0378778819319139>
* Helper Functions - <https://raw.githubusercontent.com/mrdbourke/tensorflow-deep-learning/main/extras/helper_functions.py>
* Sliding Window Vectorization (Time Series) - <https://towardsdatascience.com/fast-and-robust-sliding-window-vectorization-with-numpy-3ad950ed62f5>
* Multiple Useful blogs on Machine Learning -

<https://machinelearningmastery.com/blog/>

* MultiLabel Classification -

<https://www.youtube.com/watch?v=hraKTseOuJA>

* Support Vector Machine for Classification - <https://www.youtube.com/watch?v=H9yACitf-KM>
* AR and ML Model understanding using Excel - <https://www.youtube.com/watch?v=2kmBRH0caBA>
* TimeSeries models -

<https://neptune.ai/blog/select-model-for-time-series-prediction-task>

* Clustering Algorithms -

<https://neptune.ai/blog/clustering-algorithms>

<https://machinelearningmastery.com/clustering-algorithms-with-python/>

* Gaussian Process Explanation -

<https://arxiv.org/pdf/2009.10862.pdf>

* Deploy ML Models using PyWebIO and Flask -

<https://www.youtube.com/watch?v=2wjFQXNYLMI>

<https://pywebio.readthedocs.io/en/latest/>

* A/B Testing -

<https://www.youtube.com/watch?v=6BERouEM_bA>

* AUC ROC Curve: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

<https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/>

* Numpy:

<https://www.youtube.com/watch?v=QUT1VHiLmmI>

* Incremental learning: River library:  
  <https://riverml.xyz/latest/api/overview/>  
  <https://towardsdatascience.com/river-the-best-python-library-for-online-machine-learning-56bf6f71a403>
* Coding Mistakes -

<https://towardsdatascience.com/top-10-coding-mistakes-made-by-data-scientists-bb5bc82faaee>

* SQL Like Queries with Pandas -

<https://medium.com/jbennetcodes/how-to-rewrite-your-sql-queries-in-pandas-and-more-149d341fc53e>

**# Sample Datasets Picked up for Analysis -**

* <https://www.kaggle.com/uciml/pima-indians-diabetes-database>
* <https://archive.ics.uci.edu/ml/datasets/Real+estate+valuation+data+set>
* <https://www.kaggle.com/srinuti/residential-power-usage-3years-data-timeseries>
* <https://www.kaggle.com/iamsouravbanerjee/animal-image-dataset-90-different-animals>
* <https://www.kaggle.com/doaaalsenani/usa-cers-dataset>
* <https://www.kaggle.com/tunguz/used-car-auction-prices>
* <https://www.kaggle.com/pepepython/spotify-huge-database-daily-charts-over-3-years>
* <https://www.kaggle.com/crailtap/taxi-trajectory/code>
* <https://www.kaggle.com/c/bosch-production-line-performance/data>
* <https://archive.ics.uci.edu/ml/datasets/Bank+Marketing>
* <https://archive.ics.uci.edu/ml/datasets/Demand+Forecasting+for+a+store>
* <https://www.kaggle.com/datasets/shivashi11/food-demand-prediction>
* <https://www.kaggle.com/competitions/store-sales-time-series-forecasting>
* <https://www.kaggle.com/datasets/andrewmvd/car-plate-detection>
* <https://towardsdatascience.com/sales-forecasting-with-price-promotion-effects-b5d70207b128>
* <https://towardsdatascience.com/implementing-a-profitable-promotional-strategy-for-starbucks-with-machine-learning-part-1-2f25ec9ae00c>

<https://towardsdatascience.com/implementing-a-profitable-promotional-strategy-for-starbucks-with-machine-learning-part-2-8dd82b21577c>

<https://github.com/joshxinjie/Data_Scientist_Nanodegree/tree/master/capstone>

**# Few Github Projects -**

* Tensor Flow - <https://github.com/mrdbourke/tensorflow-deep-learning>
* ML - <https://github.com/GokuMohandas/MadeWithML/tree/main/notebooks>
* List of 500+ AI projects with code - <https://github.com/ashishpatel26/500-AI-Machine-learning-Deep-learning-Computer-vision-NLP-Projects-with-code>
* TensorFlow - <https://github.com/ageron/tf2_course>
* Car Auction - <https://github.com/Tselmeg-C/Udacity-Data_Scientist_Nanodegree_Project_One-Car_Auction_Dataset_Analysis>
* TimeSeries -

<https://github.com/ritvikmath/Time-Series-Analysis>

<https://github.com/krishnaik06/Live-Time-Series>

* GANs
  + InterfaceGAN - <https://github.com/genforce/interfacegan>
  + PSGAN - <https://github.com/wtjiang98/PSGAN>
* Pandas - [https://github.com/KeithGalli/pandas/blob/master/Pandas%20Data%20Science%20Tutorial.ipynb](https://github.com/KeithGalli/pandas/blob/master/Pandas Data Science Tutorial.ipynb)

**# Data Preprocessing -**

* <https://www.youtube.com/watch?v=V0u6bxQOUJ8>
* Difference between Normalization and Standardization: <https://towardsdatascience.com/normalization-vs-standardization-cb8fe15082eb>
* One Hot Encoding process: <https://www.educative.io/blog/one-hot-encoding>

**# Time Series -**

* <https://towardsdatascience.com/introduction-to-time-series-forecasting-part-2-arima-models-9f47bf0f476b>
* Different Algorithms - <https://www.advancinganalytics.co.uk/blog/2021/06/22/10-incredibly-useful-time-series-forecasting-algorithms>
* [https://www.analyticsvidhya.com/blog/2020/11/stock-market-price-trend-prediction-using-time-series-forecasting](https://www.analyticsvidhya.com/blog/2020/11/stock-market-price-trend-prediction-using-time-series-forecasting/)
* ARIMA -

<https://www.youtube.com/watch?v=Aw77aMLj9uM>

<https://www.youtube.com/watch?v=e8Yw4alG16Q>

<https://www.youtube.com/watch?v=8FCDpFhd1zk>

* XgBoost - <https://www.youtube.com/watch?v=4rikgkt4IcU>
* VAR -

<https://www.youtube.com/watch?v=4jv1NGlAc_0>

<https://www.youtube.com/watch?v=RBD8lqrG6l8>

* Theory -

<https://www.youtube.com/watch?v=Prpu_U5tKkE>

<https://www.youtube.com/watch?v=s3XH7fTHMb4>

**# GAN -**

* <https://learnopencv.com/introduction-to-generative-adversarial-networks/>
* <https://developers.google.com/machine-learning/gan/discriminator>
* <https://medium.com/@jain.yasha/gan-latent-space-1b32cd34cfda>
* <https://www.youtube.com/watch?v=RRTuumxm3CE&list=PLdxQ7SoCLQAMGgQAIAcyRevM8VvygTpCu>
* PGGAN - <https://machinelearningmastery.com/introduction-to-progressive-growing-generative-adversarial-networks/>
* StyleGAN - <https://medium.com/analytics-vidhya/what-is-stylegan-an-overview-of-the-key-concepts-of-stylegan-3c1031775fb>

<https://machinelearningmastery.com/introduction-to-style-generative-adversarial-network-stylegan/>

* Code -

<https://www.youtube.com/watch?v=OljTVUVzPpM>

<https://machinelearningmastery.com/how-to-interpolate-and-perform-vector-arithmetic-with-faces-using-a-generative-adversarial-network/>

* Latent Space - <https://towardsdatascience.com/understanding-latent-space-in-machine-learning-de5a7c687d8d>

**# GeoSpatial**

* Article on route optimization -

<https://towardsdatascience.com/improving-operations-with-route-optimization-4b8a3701ca39>

* Google CVRPTW library for Vehicle Routing -

<https://developers.google.com/optimization/routing/cvrptw_resources>

* Vehicle Routing Problem Library (VRPy):

<https://vrpy.readthedocs.io/en/latest/>

* Distance Matrix API :

[https://developers.google.com/maps/documentation/distance-matrix/overview#maps\_http\_distancematrix\_latlng-py](https://developers.google.com/maps/documentation/distance-matrix/overview" \l "maps_http_distancematrix_latlng-py)

* VRP-RL:

<https://github.com/OptMLGroup/VRP-RL>

* VRP (Google OR Tool):  
  [https://developers.google.com/optimization/routing/vrp#evaluator](https://developers.google.com/optimization/routing/vrp" \l "evaluator)

**# Object Detection**

* <https://www.youtube.com/watch?v=yqkISICHH-U>

# **Amazon SageMaker**

* <https://www.youtube.com/watch?v=LkR3GNDB0HI&list=PLZoTAELRMXVONh5mHrXowH6-dgyWoC_Ew>

**# Deploy ML Models -**

* <https://www.youtube.com/watch?v=-UYyyeYJAoQ>

**# BE Development**

* Host Data Science Application using FastAPI

<https://towardsdatascience.com/build-and-host-fast-data-science-applications-using-fastapi-823be8a1d6a0>

* Deploy FastAPI apps over HTTPs with Traefik Proxy - <https://www.youtube.com/watch?v=7N5O62FjGDc>
* MicroServices -

<https://microservices.io/>

<https://dearsikandarkhan.medium.com/microservices-architecture-for-e-commerce-f8b49270e72f>

* Getting Started with Python - RabbitMQ + Docker + Flask + Celery -

<https://www.youtube.com/watch?v=x98-JfEV7IA>

- Simple Serverless FASTAPI with AWS Lambda -

<https://www.youtube.com/watch?v=6fE31084Uks>

**# Amazon Services -**

* Sagemaker -

<https://towardsdatascience.com/aws-sagemaker-db5451e02a79>

<https://towardsdatascience.com/what-makes-aws-sagemaker-great-for-machine-learning-c8a42c208aa3>

<https://www.youtube.com/watch?v=LkR3GNDB0HI&list=PLZoTAELRMXVONh5mHrXowH6-dgyWoC_Ew>

* Personalize - <https://towardsdatascience.com/build-a-recommender-system-in-less-than-an-hour-using-amazon-personalize-68bee9931c60>

**# Explore Algorithms**

* Regression
  + Linear and Multiple Linear Regression (w & w/o SGD optimizer)
  + Lasso Regression
  + Ridge Regression
  + ElasticNet Regression
  + Polynomial Regression
  + RANSAC - Random Sample Consensus (Robust Regression)
  + Decision Tree
  + Random Forest
  + Gaussian Process Regression
  + Support Vector Regression
  + XgBoost Regression
  + Regression Artificial Neural Networks
* Classification
  + Logistic Regression
  + Decision Tree
  + Random Forest
  + MLPClassifier
  + ExtraTreesClassifier
  + AdaBoostClassifier
  + Naive Bayes
  + K-Nearest Neighbor
  + Gradient Boosting
  + Gaussian Process Classifier
  + QuadraticDiscriminantClassifier
  + SGDClassifier
  + Support Vector Classifier
  + XgBoost Classification
  + Convolutional Neural Networks
* TimeSeries
  + AR
  + MA
  + ARMA
  + ARIMA
  + SARIMA
  + SARIMAX
  + VAR
  + VARMA
  + VARMAX
  + XgBoost

# Recommendation System:

* Association Rule based(Apriori Algorithm):
  + <https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html>
  + <https://www.kaggle.com/code/evrenermis/association-rule-based-learning-explained/notebook>
  + https://towardsdatascience.com/the-frequently-bought-together-recommendation-system-b4ed076b24e5
* Collaborative Filtering
  + <https://www.kaggle.com/code/ayseymn/hybrid-recommender-system-netflix/notebook>
  + [https://medium.com/codex/hybrid-recommender-system-netflix-prize-dataset-e9f6b4a875aa#:~:text=There%20are%20many%20examples%20such,are%20based%20on%20recommendation%20systems](https://medium.com/codex/hybrid-recommender-system-netflix-prize-dataset-e9f6b4a875aa" \l ":~:text=There are many examples such,are based on recommendation systems)
* Content Filtering
  + <https://towardsdatascience.com/how-to-build-from-scratch-a-content-based-movie-recommender-with-natural-language-processing-25ad400eb243>

# Streamlit Links

\*\*\*Official Documentation

1.Get Started:<https://docs.streamlit.io/library/get-started>

\*\*\*Other Links

>>>Articles

1.[https://www.analyticsvidhya.com/blog/2020/12/deploying-machine-learning-](https://www.analyticsvidhya.com/blog/2020/12/deploying-machine-learning-models-using-streamlit-an-introductory-guide-to-model-deployment/) models-using-streamlit-an-introductory-guide-to-model-deployment/

2.[Streamlit Web App | Build Web Applications using Streamlit](https://www.analyticsvidhya.com/blog/2021/06/build-web-app-instantly-for-machine-learning-using-streamlit/)  (analyticsvidhya.com)

3.[Developing Data Web App Using Streamlit - Analytics Vidhya](https://www.analyticsvidhya.com/blog/2021/04/developing-data-web-streamlit-app/)

>>>Youtube Links

1.[How to Build a Streamlit App (Beginner level Streamlit tutorial) - Part 1](https://www.youtube.com/watch?v=-IM3531b1XU&list=PLM8lYG2MzHmTATqBUZCQW9w816ndU0xHc)