**Data Science & Machine Learning**

**(Full semester course)**

**Instructor :** [**Dr. Kushal Shah**](https://www.bekushal.com/kushal)

[**YouTube Channel**](https://www.youtube.com/c/evolutionaryintelligence)

[**ML Blog**](https://medium.com/@atmabodha)

| **Week** | **Topics (tentative plan)** | **YouTube Links** | **Reference** | **Codes and Datasets** |
| --- | --- | --- | --- | --- |
| 1 | Data Science & Machine Learning Overview | <https://youtu.be/DwqbxP5BDFo>  <https://youtu.be/UnjyNYco7uQ> |  | [NLTK Code](https://drive.google.com/file/d/18z9J5RPnuOxNzQ4eEd2cGRLxasvoUdhH/view?usp=sharing) [Jupyter Notebook]  [Fictometer Paper @ ACL 2019](https://www.aclweb.org/anthology/W19-3409)  Machine Learning Projects:  <https://data-flair.training/blogs/advanced-python-project-detecting-fake-news/>  ML Datasets:  <http://archive.ics.uci.edu/ml/index.php> |
| Random Variables and Probability Distributions | <https://youtu.be/rGYqfhmdEmo> |  |  |
| Expectation, Moments and CLT | <https://youtu.be/3-A2siEN8w8> | <https://www.youtube.com/watch?v=JNm3M9cqWyc>  <http://www.math.uchicago.edu/~may/VIGRE/VIGRE2011/REUPapers/Krokhmal.pdf> |  |
| 2 | Bayes’ Theorem | <https://youtu.be/q0p6VWj8N4I> | Conditional probability:  <https://www.mathsisfun.com/data/probability-events-conditional.html>  Practice problems:  <https://www.statisticshowto.com/bayes-theorem-problems/> |  |
| Naïve Bayes, Gaussian Naive Bayes, Bayes’ Optimal Classifier | <https://youtu.be/DV3F7AZOAE0> | Naive Bayes : <https://www.youtube.com/watch?v=O2L2Uv9pdDA>  Gaussian Naive Bayes : <https://www.youtube.com/watch?v=H3EjCKtlVog>  Bayesian Networks: <https://towardsdatascience.com/introduction-to-bayesian-networks-81031eeed94e> | Spam filter using Naive Bayes:  <https://www.kdnuggets.com/2020/07/spam-filter-python-naive-bayes-scratch.html>  <https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/>  <https://towardsdatascience.com/sentiment-analysis-of-tweets-using-multinomial-naive-bayes-1009ed24276b> |
| Supervised Learning : Regression vs Classification, Curse of Dimensionality | <https://youtu.be/NYJZDRRUO84> | <https://in.springboard.com/blog/regression-vs-classification-in-machine-learning/> | K-Nearest Neighbour (KNN) Algorithm:  <https://scikit-learn.org/stable/modules/neighbors.html>  KNN from scratch:  <https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/> |
| 3 | Bayesian Parameter Estimation | <https://youtu.be/8P7tdwFF0is> |  |  |
| Maximum Likelihood Estimation | <https://youtu.be/WIPUh9yWM4c> |  |  |
| Maximum A Posteriori Estimation | <https://youtu.be/XKkfIU8ugjM> | MLE vs MAP : <https://towardsdatascience.com/mle-vs-map-a989f423ae5c> |  |
| 4 | Supervised Learning : Gradient Descent & Regularisation | <https://youtu.be/IUmFzIU-Cp4> | <https://towardsdatascience.com/gradient-descent-the-learning-rate-and-the-importance-of-feature-scaling-6c0b416596e1>  Bias vs Variance : <https://www.youtube.com/watch?v=EuBBz3bI-aA>  Convex function:  <https://en.wikipedia.org/wiki/Convex_function> |  |
| Linear Regression : Normal Equation & Regularisation | <https://youtu.be/BGiAcoU4yWk> | <https://machinelearningmastery.com/linear-regression-for-machine-learning/>  Normal Equation and Matrix Calculation:  <https://eli.thegreenplace.net/2015/the-normal-equation-and-matrix-calculus/>  Overfitting vs Underfitting:  <https://towardsdatascience.com/overfitting-vs-underfitting-a-complete-example-d05dd7e19765>  <https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html> | <https://towardsdatascience.com/linear-regression-using-python-b136c91bf0a2>  <https://stackabuse.com/multiple-linear-regression-with-python/>  With Regularization:  <https://scikit-learn.org/stable/modules/linear_model.html>  <https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/> |
|
| 5 | Logistic Regression : Loss Function, Convexity, Gradient Descent | <https://youtu.be/sxaYUg6zc7I> | Applications of Logistic Regression:  <https://activewizards.com/blog/5-real-world-examples-of-logistic-regression-application>  Non-Convexity of LSE for LR : <https://towardsdatascience.com/why-not-mse-as-a-loss-function-for-logistic-regression-589816b5e03c>  Decision boundaries:  <https://www.youtube.com/watch?v=F_VG4LNjZZw>  <https://web.stanford.edu/~jurafsky/slp3/5.pdf> | <https://heartbeat.fritz.ai/logistic-regression-in-python-using-scikit-learn-d34e882eebb1>  Logistic Regression:  <https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc>  <https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf>  Logistic Regression (Handwritten Digit Recognition):  <https://realpython.com/logistic-regression-python/#logistic-regression-in-python-handwriting-recognition>  Logistic Regression (University admission):  <https://towardsdatascience.com/building-a-logistic-regression-in-python-301d27367c24>  Logistic Regression (Consumer Purchase):  <https://www.geeksforgeeks.org/ml-logistic-regression-using-python/>  One-Hot Encoding for Logistic Regression:  <https://www.analyticsvidhya.com/blog/2020/03/one-hot-encoding-vs-label-encoding-using-scikit-learn/> |
| Logistic Regression : Multiclass Classification | <https://youtu.be/E6MNiNsESl4> | Naive Bayes vs. Logistic Regression:  <http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf> | <https://towardsdatascience.com/multiclass-classification-algorithm-from-scratch-with-a-project-in-python-step-by-step-guide-485a83c79992>  <https://towardsdatascience.com/logistic-regression-using-python-sklearn-numpy-mnist-handwriting-recognition-matplotlib-a6b31e2b166a>  <https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/> |
| 6 | Support Vector Machines with Linear Kernel | <https://youtu.be/KvIdyz0KdCU> | SVM Applications:  <https://medium.com/@rinu.gour123/8-unique-real-life-applications-of-svm-8a96ca43313>  SVM vs. Logistic Regression : <https://medium.com/axum-labs/logistic-regression-vs-support-vector-machines-svm-c335610a3d16>  SVM vs. Linear Discriminant Analysis:  <https://stats.stackexchange.com/questions/243932/what-is-the-difference-between-svm-and-lda>  <https://www.youtube.com/watch?v=_PwhiWxHK8o>  <https://towardsdatascience.com/understanding-support-vector-machine-part-1-lagrange-multipliers-5c24a52ffc5e>  Hinge Loss Formulation of SVM:  <https://www.youtube.com/watch?v=9dPjKuTYkx4>  Lagrange Multiplier Intuition:  <https://medium.com/@andrew.chamberlain/a-simple-explanation-of-why-lagrange-multipliers-works-253e2cdcbf74>  Lagrange Multiplier (practice problems):  <https://tutorial.math.lamar.edu/problems/calciii/lagrangemultipliers.aspx> | <https://scikit-learn.org/stable/modules/svm.html>  <https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/> |
| Support Vector Machines : Slack Variables and Nonlinear Kernels | <https://youtu.be/kPuHAK4RCfM> | Intuitive understanding of RBF Kernel for SVM:  <http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exercises/ex8/ex8.html> |  |
| 7 | Decision Trees | <https://youtu.be/OlreMWwTJ_8> | Decision Tree Algorithm (using Gini Index instead of Information Gain):  <https://www.youtube.com/watch?v=7VeUPuFGJHk>  Pruning of Decision Trees:  <https://www.cs.cmu.edu/~bhiksha/courses/10-601/decisiontrees/>  Decision Tree for Regression:  <https://www.youtube.com/watch?v=g9c66TUylZ4>  <https://towardsdatascience.com/understanding-decision-trees-for-classification-python-9663d683c952> | <https://www.geeksforgeeks.org/decision-tree-implementation-python/>  <https://scikit-learn.org/stable/modules/tree.html>  <https://www.datacamp.com/community/tutorials/decision-tree-classification-python> |
| Random Forest + Comparison of Trees, SVM & LogisticRegression | <https://youtu.be/pEFBH_371sU> | <https://www.youtube.com/watch?v=J4Wdy0Wc_xQ>  Comparison of various ML algorithms:  <https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222> | <https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76>  From scratch:  <https://machinelearningmastery.com/implement-random-forest-scratch-python/> |
| 8 | Artificial Neural Networks : Introduction | <https://youtu.be/6nkylSKqaAc> | Universal Approximation Theorem:  <http://neuralnetworksanddeeplearning.com/chap4.html>  Difference between Brain and ANN:  <https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7>  <https://medium.com/digital-catapult/are-artificial-neural-networks-like-the-human-brain-and-does-it-matter-3add0f029273>  Perceptron Learning Algorithm:  <https://towardsdatascience.com/perceptron-learning-algorithm-d5db0deab975>  Weight initialisation in ANN:  <https://www.geeksforgeeks.org/weight-initialization-techniques-for-deep-neural-networks/>  <https://datascience-enthusiast.com/DL/Improving-DeepNeural-Networks-Initialization.html>  Regularization for ANN:  <https://towardsdatascience.com/how-to-improve-a-neural-network-with-regularization-8a18ecda9fe3>  Dropout:  <https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9>  Embedding layer vs Dense layer:  <https://medium.com/logivan/neural-network-embedding-and-dense-layers-whats-the-difference-fa177c6d0304> | House Price Prediction using ANN:  <https://medium.com/@robertjohn_15390/simple-housing-price-prediction-using-neural-networks-with-tensorflow-8b486d3db3ca>  Image recognition using ANN:  <https://nextjournal.com/gkoehler/digit-recognition-with-keras>  Multi-Class classification:  <https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/>  Dropout:  <https://machinelearningmastery.com/how-to-reduce-overfitting-with-dropout-regularization-in-keras/> |
| Artificial Neural Networks : Backpropagation | <https://youtu.be/ntnwjWEpnkk> | <http://neuralnetworksanddeeplearning.com/chap2.html>  For a visual intuitive understanding:  <https://www.youtube.com/watch?v=IHZwWFHWa-w&ab_channel=3Blue1Brown>  Stochastic Gradient Descent:  <https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/>  Stochastic vs Batch Gradient Descent:  <https://medium.com/@divakar_239/stochastic-vs-batch-gradient-descent-8820568eada1>  Weight initialisation:  <https://www.deeplearning.ai/ai-notes/initialization/> | From scratch:  <https://machinelearningmastery.com/implement-backpropagation-algorithm-scratch-python/> |
| 9 | Artificial Neural Networks : Activation Functions, Loss functions & Optimization Algorithms | <https://youtu.be/qctUEQn9Hj8> | Activation functions:  <https://towardsdatascience.com/comparison-of-activation-functions-for-deep-neural-networks-706ac4284c8a>  Optimization Algorithms:  <https://arxiv.org/pdf/1609.04747>  Loss Functions:  <https://towardsdatascience.com/understanding-different-loss-functions-for-neural-networks-dd1ed0274718>  Vanishing Gradient Problem:  <https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484>  SVM vs ANN analysis and datasets:  <https://web.archive.org/web/20120304030602/http://indiji.com/svm-vs-nn.html> |  |
| Solving ODEs using ANN | <https://youtu.be/-pMLqoMarEk> | <https://www.sciencedirect.com/science/article/pii/S0021999118307125> | <https://towardsdatascience.com/how-to-solve-an-ode-with-a-neural-network-917d11918932> |
| Deep Learning | <https://youtu.be/MTbBOu4M7_M> | MNIST Dataset Analysis:  <http://yann.lecun.com/exdb/mnist/> |  |
| 10 | Generative Vs. Discriminative Models | <https://youtu.be/WSxUOIGuGi8> |  | <https://machinelearningmastery.com/how-to-develop-a-generative-adversarial-network-for-an-mnist-handwritten-digits-from-scratch-in-keras/> |
| Probabilistic Graphical Models : Bayesian Network and Markov Network | <https://youtu.be/xJtyVQMV1A8> | <https://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html> | <https://www.youtube.com/watch?v=SkC8S3wuIfg> |
| Expectation Maximization | <https://youtu.be/PNrghbfK5r0> | <https://www.nature.com/articles/nbt1406> | <https://people.duke.edu/~ccc14/sta-663/EMAlgorithm.html> |
| 11 | K-Means clustering | <https://youtu.be/92-h3_1Yfz8> | K-Means vs KNN:  <https://becominghuman.ai/comprehending-k-means-and-knn-algorithms-c791be90883d>  Explanation and code :  <https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/>  VIsualising K-Means:  <https://www.naftaliharris.com/blog/visualizing-k-means-clustering/>  How to choose optimal K value:  <https://towardsdatascience.com/are-you-solving-ml-clustering-problems-using-k-means-68fb4efa5469> | <https://towardsdatascience.com/machine-learning-algorithms-part-9-k-means-example-in-python-f2ad05ed5203>  <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>  Image compression using K-Means:  <https://towardsdatascience.com/image-compression-using-k-means-clustering-aa0c91bb0eeb>  MNIST image classification using K-Means:  <https://medium.com/@joel_34096/k-means-clustering-for-image-classification-a648f28bdc47> |
| Gaussian Mixture Models | <https://youtu.be/he4lI0w1G-g> | <https://brilliant.org/wiki/gaussian-mixture-model/>  <https://scikit-learn.org/stable/modules/mixture.html> | <https://www.analyticsvidhya.com/blog/2019/10/gaussian-mixture-models-clustering/> |
| 12 | Principal Component Analysis (PCA) - I | <https://youtu.be/cERNIfg9TLM> | <https://builtin.com/data-science/step-step-explanation-principal-component-analysis>  <https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c>  PCA vs Linear Regression:  <https://shankarmsy.github.io/posts/pca-vs-lr.html> | <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>  <https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/>  <https://shankarmsy.github.io/posts/pca-sklearn.html> |
| Principal Component Analysis (PCA) - II | <https://youtu.be/ml1d1zC0jsg> | Kernel PCA : <http://fourier.eng.hmc.edu/e161/lectures/kernelPCA/node4.html> |  |
| 13 | Reinforcement Learning | <https://youtu.be/KEQhaBIZ9yk> | <https://www.youtube.com/watch?v=nZfaHIxDD5w&ab_channel=AlexanderAmini>  Tic Tac Toe using Reinforcement Learning:  <https://towardsdatascience.com/reinforcement-learning-implement-tictactoe-189582bea542>  <https://medium.com/vernacular-ai/reinforcement-learning-step-by-step-17cde7dbc56c> | <https://amunategui.github.io/reinforcement-learning/index.html>  <https://towardsdatascience.com/reinforcement-learning-implement-tictactoe-189582bea542> |
| Big data technologies & Information management in Big Data | <https://youtu.be/5dSmXcKsEBo> | <https://www.edureka.co/blog/top-big-data-technologies/>  MapReduce Algorithm:  <https://www.cs.amherst.edu/~ccmcgeoch/cs34/papers/p107-dean.pdf>  NPTEL course from IIT KGP:  <https://nptel.ac.in/courses/106/105/106105186/>  Comparison of Big Data Technologies:  <https://www.educba.com/hadoop-vs-spark/>  <https://phoenixnap.com/kb/hadoop-vs-spark>  <https://www.educba.com/kafka-vs-spark/>  <https://www.educba.com/apache-kafka-vs-flume/> |  |
| Emerging Issues in Data Science |  | <https://towardsdatascience.com/emerging-problems-in-data-science-and-machine-learning-36d37f6531a8>  Explainable AI:  <https://en.wikipedia.org/wiki/Explainable_artificial_intelligence>  AGI:  <https://en.wikipedia.org/wiki/Artificial_general_intelligence>  Few Shot Learning:  <https://medium.com/quick-code/understanding-few-shot-learning-in-machine-learning-bede251a0f67>  Micro Services:  <https://en.wikipedia.org/wiki/Microservices>  Functional Programming:  <https://en.wikipedia.org/wiki/Functional_programming> |  |

**Other Resources:**

Machine Learning by Dr. Andrew Ng:

<https://www.youtube.com/watch?v=PPLop4L2eGk&list=PLLssT5z_DsK-h9vYZkQkYNWcItqhlRJLN&ab_channel=ArtificialIntelligence-AllinOne>

Neural Networks and Deep Learning:

<http://neuralnetworksanddeeplearning.com>

Dr. Balaraman Ravindran, IIT Madras:

<https://www.cse.iitm.ac.in/~ravi/courses/Introduction%20to%20Machine%20Learning.html>

Dr. Balaji Srinivasan and Dr. Ganapathy Krishnamurthi, IIT Madras

<https://nptel.ac.in/courses/106/106/106106198/>

Dr. Parag Singla, IIT Delhi:

<http://www.cse.iitd.ac.in/~parags/teaching/col774/>

Dr. Sumeet Agarwal, IIT Delhi:

<http://web.iitd.ac.in/~sumeet/ell409.html>

**Python Programming**

<http://cl.indiana.edu/~md7/16/555/>

<https://www.tutorialspoint.com/python/index.htm>

<https://www.w3schools.com/python/>

**NLP Resources**

NLTK Tutorial:

<http://www.nltk.org/book/ch01.html>

Stanza POS Tagging:

<https://stanfordnlp.github.io/stanza/tokenize.html#start-with-pretokenized-text>

NLP Book by Jurafsky and Martin:

<https://web.stanford.edu/~jurafsky/slp3/>