1. **Python for ML/AI**
    - 1.1. Why Python?
    - 1.2. Setup
        - 1.2.1. Install Python.
        - 1.2.2. Installing packages: numpy, pandas, scipy, matplotlib, seaborn, sklearn)
        - 1.2.3. iPython setup.
    - 1.3. Introduction
        - 1.3.1. Keywords and Identifiers
        - 1.3.2. Statements, Indentation and Comments
        - 1.3.3. Variables and Datatypes
        - 1.3.4. Input and Output
        - 1.3.5. Operators
    - 1.4. Flow Control
        - 1.4.1. If...else
        - 1.4.2. while loop
        - 1.4.3. for loop
        - 1.4.4. break and continue
    - 1.5. Data Structures
        - 1.5.1. Lists
        - 1.5.2. Tuples
        - 1.5.3. Dictionary
        - 1.5.4. Strings
        - 1.5.5. Sets
    - 1.6. Functions
        - 1.6.1. Introduction
        - 1.6.2. Types of functions
        - 1.6.3. Function Arguments
        - 1.6.4. Recursive Functions
        - 1.6.5. Lambda Functions
        - 1.6.6. Modules
        - 1.6.7. Packages
    - 1.7. File Handling
    - 1.8. Exception Handling
    - 1.9. Debugging Python
    - 1.10. NumPy
        - 1.10.1. Introduction to NumPy.
        - 1.10.2. Numerical operations.
    - 1.11. Matplotlib
    - 1.12. Pandas
        - 1.12.1. Getting started with pandas
        - 1.12.2. Data Frame Basics
        - 1.12.3. Key Operations on Data Frames.

1.13. Computational Complexity: an Introduction
    1.13.1. Space and Time Complexity: Find largest number in a list
    1.13.2. Binary search
    1.13.3. Find elements common in two lists.
    1.13.4. Find elements common in two lists using a Hashtable/Dict
    1.13.5. Further reading about Computational Complexity [Please add a section with these links for reference]
        1.13.5.1. https://medium.com/omarelgabrys-blog/the-big-scary-o-notation-ce9352d827ce
        1.13.5.2. https://rob-bell.net/2009/06/a-beginners-guide-to-big-o-notation/
        1.13.5.3. https://medium.freecodecamp.org/time-is-complex-but-priceless-f0abd015063c

2. Plotting for exploratory data analysis (EDA)

2.1. Iris dataset
    2.1.1. Data-point, vector, observation
    2.1.2. Dataset
    2.1.3. Input variables/features/dimensions/independent variable
    2.1.4. Output Variable/Class Label/ Response Label/ dependent variable
    2.1.5. Objective: Classification.
2.2. Scatter-plot: 2D, 3D.
2.3. Pair plots.
2.4. PDF, CDF, Univariate analysis.
    2.4.1. Histogram and PDF
    2.4.2. Univariate analysis using PDFs.
    2.4.3. Cumulative distribution function (CDF)
2.5. Mean , Variance, Std-dev
2.6. Median, Percentiles, Quantiles, IQR, MAD and Outliers.
2.7. Box-plot with whiskers
2.8. Violin plots.
2.9. Summarizing plots.
2.10. Univariate, Bivariate and Multivariate analysis.
2.11. Multivariate probability density, contour plot.
2.12. Exercise: Perform EDA on Haberman dataset.

3. Probability and Statistics

3.1. Introduction to Probability and Stats
    3.1.1. Why learn it?
    3.1.2. P(X=x1) , Dice and coin example
    3.1.3. Random variables: discrete and continuous.
    3.1.4. Outliers (or) extreme points.
    3.1.5. Population & Sample.
3.2. Gaussian/Normal Distribution
    3.2.1. Examples: Heights and weights.
    3.2.2. Why learn about distributions.

3.2.3. Mu, sigma: Parameters
3.2.4. PDF (iris dataset)
3.2.5. CDF
3.2.6. 1-std-dev, 2-std-dev, 3-std-dev range.
3.2.7. Symmetric distribution, Skewness and Kurtosis
3.2.8. Standard normal variate (z) and standardization.
3.2.9. Kernel density estimation.
3.2.10. Sampling distribution & Central Limit theorem.
3.2.11. Q-Q Plot: Is a given random variable Gaussian distributed?

3.3. Uniform Distribution and random number generators
3.3.1. Discrete and Continuous Uniform distributions.
3.3.2. How to randomly sample data points. [UniformDisb.ipynb]

3.4. Bernoulli and Binomial distribution

3.5. Log-normal and power law distribution:
3.5.1. Log-normal: CDF, PDF, Examples.
3.5.2. Power-law & Pareto distributions: PDF, examples
3.5.3. Converting power law distributions to normal: Box-Cox/Power transform.

3.6. Correlation
3.6.1. Co-variance
3.6.2. Pearson Correlation Coefficient
3.6.3. Spearman Rank Correlation Coefficient
3.6.4. Correlation vs Causation

3.7. Confidence Intervals
3.7.1. Confidence Interval vs Point estimate.
3.7.2. Computing confidence-interval given a distribution.
3.7.3. For mean of a random variable
3.7.3.1. Known Standard-deviation: using CLT
3.7.3.2. Unknown Standard-deviation: using t-distribution
3.7.4. Confidence Interval using empirical bootstrap [BootstrapCI.ipynb]

3.8. Hypothesis testing
3.8.1. Hypothesis Testing methodology, Null-hypothesis, test-statistic, p-value.
3.8.2. Resampling and permutation test.
3.8.3. K-S Test for similarity of two distributions.
3.8.4. Code Snippet [KSTest.ipynb]

4. Linear Algebra

4.1. Why learn it ?

4.2. Fundamentals
4.2.1. Point/Vector (2-D, 3-D, n-D)
4.2.2. Dot product and angle between 2 vectors.
4.2.3. Projection, unit vector
4.2.4. Equation of a line (2-D), plane(3-D) and hyperplane (n-D)
4.2.5. Distance of a point from a plane/hyperplane, half-spaces
4.2.6. Equation of a circle (2-D), sphere (3-D) and hypersphere (n-D)

4.2.7. Equation of an ellipse (2-D), ellipsoid (3-D) and hyperellipsoid (n-D)
4.2.8. Square, Rectangle, Hyper-cube and Hyper-cuboid..

5. Dimensionality reduction and Visualization:

5.1. What is dimensionality reduction?
5.2. Data representation and pre-processing
5.2.1. Row vector, Column vector: Iris dataset example.
5.2.2. Represent a dataset: D= {x_i, y_i}
5.2.3. Represent a dataset as a Matrix.
5.2.4. Data preprocessing: Column Normalization
5.2.5. Mean of a data matrix.
5.2.6. Data preprocessing: Column Standardization
5.2.7. Co-variance of a Data Matrix.
5.3. MNIST dataset (784 dimensional)
5.3.1. Explanation of the dataset.
5.3.2. Code to load this dataset.
5.4. Principal Component Analysis.
5.4.1. Why learn it.
5.4.2. Geometric intuition.
5.4.3. Mathematical objective function.
5.4.4. Alternative formulation of PCA: distance minimization
5.4.5. Eigenvalues and eigenvectors.
5.4.6. PCA for dimensionality reduction and visualization.
5.4.7. Visualize MNIST dataset.
5.4.8. Limitations of PCA
5.4.9. Code example.
5.4.10. PCA for dimensionality reduction (not-visualization)
5.5. T-distributed stochastic neighborhood embedding (t-SNE)
5.5.1. What is t-SNE?
5.5.2. Neighborhood of a point, Embedding.
5.5.3. Geometric intuition.
5.5.4. Crowding problem.
5.5.5. How to apply t-SNE and interpret its output (distill.pub)
5.5.6. t-SNE on MNIST.
5.5.7. Code example.

6. Real world problem: Predict sentiment polarity given product reviews on Amazon.

6.1. Exploratory Data Analysis.
6.1.1. Dataset overview: Amazon Fine Food reviews
6.1.2. Data Cleaning: Deduplication.
6.2. Featurizations: convert text to numeric vectors.
6.2.1. Why convert text to a vector?
6.2.2. Bag of Words (BoW)
6.2.3. Text Preprocessing: Stemming, Stop-word removal, Tokenization, Lemmatization.

6.2.4. uni-gram, bi-gram, n-grams.
6.2.5. tf-idf (term frequency- inverse document frequency)
[7.2.5 a] [New Video] Why use log in IDF?
6.2.6. Word2Vec.
6.2.7. Avg-Word2Vec, tf-idf weighted Word2Vec
6.3. Code samples
6.3.1. Bag of Words.
6.3.2. Text Preprocessing
6.3.3. Bi-Grams and n-grams.
6.3.4. TF-IDF
6.3.5. Word2Vec
6.3.6. Avg-Word2Vec and TFIDF-Word2Vec
6.4. Exercise: t-SNE visualization of Amazon reviews with polarity based color-coding

7. Classification and Regression Models: K-Nearest Neighbors

7.1. Foundations
7.1.1. How “Classification” works?
7.1.2. Data matrix notation.
7.1.3. Classification vs Regression (examples)
7.2. K-Nearest Neighbors
7.2.1. Geometric intuition with a toy example.
7.2.2. Failure cases.
7.2.3. Distance measures: Euclidean(L2) , Manhattan(L1), Minkowski, Hamming
7.2.4. Cosine Distance & Cosine Similarity
7.2.5. How to measure the effectiveness of k-NN?
7.2.6. Simple implementation:
7.2.6.1. Test/Evaluation time and space complexity.
7.2.6.2. Limitations.
7.2.7. Determining the right “k”
7.2.7.1. Decision surface for K-NN as K changes.
7.2.7.2. Overfitting and Underfitting.
7.2.7.3. Need for Cross validation.
7.2.7.4. K-fold cross validation.
[NEW]8.2.7.4 a Visualizing train, validation and test datasets
7.2.7.5. How to determine overfitting and underfitting?
7.2.7.6. Time based splitting
7.2.8. k-NN for regression.
7.2.9. Weighted k-NN
7.2.10. Voronoi diagram.
7.2.11. kd-tree based k-NN:
7.2.11.1. Binary search tree
7.2.11.2. How to build a kd-tree.

7.2.11.3. Find nearest neighbors using kd-tree
7.2.11.4. Limitations.
7.2.11.5. Extensions.
7.2.12. Locality sensitive Hashing (LSH)
7.2.12.1. Hashing vs LSH.
7.2.12.2. LSH for cosine similarity
7.2.12.3. LSH for euclidean distance.
7.2.13. Probabilistic class label
7.2.14. Code Samples for K-NN
7.2.14.1. Decision boundary. [./knn/knn.ipynb and knn folder]
7.2.14.2. Cross Validation.[./knn/kfold.ipynb and knn folder]
7.2.15. Exercise: Apply k-NN on Amazon reviews dataset.

8. Classification algorithms in various situations:
8.1. Introduction
8.2. Imbalanced vs balanced dataset.
8.3. Multi-class classification.
8.4. k-NN, given a distance or similarity matrix
8.5. Train and test set differences.
8.6. Impact of Outliers
8.7. Local Outlier Factor.
8.7.1. Simple solution: mean dist to k-NN.
8.7.2. k-distance (A), N(A)
8.7.3. reachability-distance(A, B)
8.7.4. Local-reachability-density(A)
8.7.5. LOF(A)
8.8. Impact of Scale & Column standardization.
8.9. Interpretability
8.10. Feature importance & Forward Feature Selection
8.11. Handling categorical and numerical features.
8.12. Handling missing values by imputation.
8.13. Curse of dimensionality.
8.14. Bias-Variance tradeoff.
9.14a Intuitive understanding of bias-variance.
8.15. Best and worst cases for an algorithm.
8.16. Performance measurement of models:
8.17. Accuracy
8.18. Confusion matrix, TPR, FPR, FNR, TNR
8.19. Precision & recall, F1-score.
8.20. Receiver Operating Characteristic Curve (ROC) curve and AUC.
8.21. Log-loss.
8.22. R-Squared/ Coefficient of determination.
8.23. Median absolute deviation (MAD)
8.24. Distribution of errors.

9. Naive Bayes
    9.1. Conditional probability.
    9.2. Independent vs Mutually exclusive events.
    9.3. Bayes Theorem with examples.
    9.4. Exercise problems on Bayes Theorem.
    9.5. Naive Bayes algorithm.
    9.6. Toy example: Train and test stages.
    9.7. Naive Bayes on Text data.
    9.8. Laplace/Additive Smoothing.
    9.9. Log-probabilities for numerical stability.
    9.10. Cases:
        9.10.1. Bias and Variance tradeoff.
        9.10.2. Feature importance and interpretability.
        9.10.3. Imbalanced data.
        9.10.4. Outliers.
        9.10.5. Missing values.
        9.10.6. Handling Numerical features (Gaussian NB)
        9.10.7. Multiclass classification.
        9.10.8. Similarity or Distance matrix.
        9.10.9. Large dimensionality.
        9.10.10. Best and worst cases.
    9.11. Code example
    9.12. Exercise: Apply Naive Bayes to Amazon reviews.
10. Logistic Regression:
    10.1. Geometric intuition.
    10.2. Sigmoid function & Squashing
    10.3. Optimization problem.
    10.4. Weight vector.
    10.5. L2 Regularization: Overfitting and Underfitting.
    10.6. L1 regularization and sparsity.
    10.7. Probabilistic Interpretation: GaussianNaiveBayes
    10.8. Loss minimization interpretation
    10.9. Hyperparameter search: Grid Search and Random Search
    10.10. Column Standardization.
    10.11. Feature importance and model interpretability.
    10.12. Collinearity of features.
    10.13. Train & Run time space and time complexity.
    10.14. Real world cases.
    10.15. Non-linearly separable data & feature engineering.
    10.16. Code sample: Logistic regression, GridSearchCV, RandomSearchCV
    10.17. Exercise: Apply Logistic regression to Amazon reviews dataset.
    10.18. Extensions to Logistic Regression: Generalized linear models (GLM)

11. Linear Regression and Optimization.
    - 11.1. Geometric intuition.
    - 11.2. Mathematical formulation.
    - 11.3. Cases.
    - 11.4. Code sample.
    - 11.5. Solving optimization problems
        - 11.5.1. Differentiation.
        - 13.5.1_a Online differentiation tools
        - 11.5.2. Maxima and Minima
        - 11.5.3. Vector calculus: Grad
        - 11.5.4. Gradient descent: geometric intuition.
        - 11.5.5. Learning rate.
        - 11.5.6. Gradient descent for linear regression.
        - 11.5.7. SGD algorithm.
        - 11.5.8. Constrained optimization & PCA
        - 11.5.9. Logistic regression formulation revisited.
        - 11.5.10. Why L1 regularization creates sparsity?
        - 11.5.11. Exercise: Implement SGD for linear regression.

12. Support Vector Machines (SVM)
    - 12.1. Geometric intuition.
    - 12.2. Mathematical derivation.
    - 12.3. Loss minimization: Hinge Loss.
    - 12.4. Dual form of SVM formulation.
    - 12.5. Kernel trick.
    - 12.6. Polynomial kernel.
    - 12.7. RBF-Kernel.
    - 12.8. Domain specific Kernels.
    - 12.9. Train and run time complexities.
    - 12.10. nu-SVM: control errors and support vectors.
    - 12.11. SVM Regression.
    - 12.12. Cases.
    - 12.13. Code Sample.
    - 12.14. Exercise: Apply SVM to Amazon reviews dataset.

13. Decision Trees
    - 13.1. Geometric Intuition: Axis parallel hyperplanes.
    - 13.2. Sample Decision tree.
    - 13.3. Building a decision Tree:
        - 13.3.1. Entropy
        - 15.3.1.a Intuition behind entropy
        - 13.3.2. Information Gain
        - 13.3.3. Gini Impurity.

13.3.4. Constructing a DT.
13.3.5. Splitting numerical features.
15.3.5a Feature standardization.
13.3.6. Categorical features with many possible values.
13.4. Overfitting and Underfitting.
13.5. Train and Run time complexity.
13.6. Regression using Decision Trees.
13.7. Cases
13.8. Code Samples.
13.9. Exercise: Decision Trees on Amazon reviews dataset.

14. Ensemble Models:
14.1. What are ensembles?
14.2. Bootstrapped Aggregation (Bagging)
14.2.1. Intuition
14.2.2. Random Forest and their construction.
14.2.3. Bias-Variance tradeoff.
14.2.4. Train and Run-time Complexity.
14.2.5. Code Sample.
14.2.6. Extremely randomized trees.
14.2.7. Cases
14.3. Boosting:
14.3.1. Intuition
14.3.2. Residuals, Loss functions and gradients.
14.3.3. Gradient Boosting
14.3.4. Regularization by Shrinkage.
14.3.5. Train and Run time complexity.
14.3.6. XGBoost: Boosting + Randomization
14.3.7. AdaBoost: geometric intuition.
14.4. Stacking models.
14.5. Cascading classifiers.
14.6. Kaggle competitions vs Real world.
14.7. Exercise: Apply GBDT and RF to Amazon reviews dataset.

15. Featurizations and Feature engineering.
15.1. Introduction.
15.2. Time-series data.
15.2.1. Moving window.
15.2.2. Fourier decomposition.
15.2.3. Deep learning features: LSTM
15.3. Image data.
15.3.1. Image histogram.
15.3.2. Keypoints: SIFT.

15.3.3. Deep learning features: CNN
15.4. Relational data.
15.5. Graph data.
15.6. Feature Engineering.
15.6.1. Indicator variables.
15.6.2. Feature binning.
15.6.3. Interaction variables.
15.6.4. Mathematical transforms.
15.7. Model specific featurizations.
15.8. Feature orthogonality.
15.9. Domain specific featurizations.
15.10. Feature slicing.
15.11. Kaggle Winners solutions.

17a. Miscellaneous Topics
17a.1 Calibration of Models.
17a.1.1 Need for calibration.
17a.1.2 Calibration Plots.
17a.1.3 Platt's Calibration/Scaling.
17a.1.4 Isotonic Regression
17a.1.5 Code Samples
17a.1.6 Exercise: Calibration + Naive Bayes.
15.12. Modeling in the presence of outliers: RANSAC
15.13. Productionizing models.
15.14. Retraining models periodically as needed.
15.15. A/B testing.
15.16. VC Dimensions.

16. Unsupervised learning/Clustering: K-Means (2)
16.1. What is Clustering?
16.2. Unsupervised learning
16.3. Applications.
16.4. Metrics for Clustering.
16.5. K-Means
16.5.1. Geometric intuition, Centroids
16.5.2. Mathematical formulation: Objective function
16.5.3. K-Means Algorithm.
16.5.4. How to initialize: K-Means++
16.5.5. Failure cases/Limitations.
16.5.6. K-Medoids
16.5.7. Determining the right K.
16.5.8. Time and space complexity.

16.5.9. Code Samples
16.5.10. Exercise: Cluster Amazon reviews.

17. Hierarchical clustering
    17.1. Agglomerative & Divisive, Dendrograms
    17.2. Agglomerative Clustering.
    17.3. Proximity methods: Advantages and Limitations.
    17.4. Time and Space Complexity.
    17.5. Limitations of Hierarchical Clustering.
    17.6. Code sample.
    17.7. Exercise: Amazon food reviews.

18. DBSCAN (Density based clustering)
    18.1. Density based clustering
    18.2. MinPts and Eps: Density
    18.3. Core, Border and Noise points.
    18.4. Density edge and Density connected points.
    18.5. DBSCAN Algorithm.
    18.6. Hyper Parameters: MinPts and Eps.
    18.7. Advantages and Limitations of DBSCAN.
    18.8. Time and Space Complexity.
    18.9. Code samples.
    18.10. Exercise: Amazon Food reviews.

19. Recommender Systems and Matrix Factorization. (3)
    19.1. Problem formulation: Movie reviews.
    19.2. Content based vs Collaborative Filtering.
    19.3. Similarity based Algorithms.
    19.4. Matrix Factorization:
        19.4.1. PCA, SVD
        19.4.2. NMF
        19.4.3. MF for Collaborative filtering
        19.4.4. MF for feature engineering.
        19.4.5. Clustering as MF
    19.5. Hyperparameter tuning.
    19.6. Matrix Factorization for recommender systems: Netflix Prize Solution
    19.7. Cold Start problem.
    19.8. Word Vectors using MF.
    19.9. Eigen-Faces.
    19.10. Code example.
    19.11. Exercise: Word Vectors using Truncated SVD.

20. Neural Networks.

20.1. History of Neural networks and Deep Learning.
20.2. Diagrammatic representation: Logistic Regression and Perceptron
20.3. Multi-Layered Perceptron (MLP).
20.4. Training an MLP.
20.5. Backpropagation.
20.6. Weight initialization.
20.7. Vanishing Gradient problem.
20.8. Bias-Variance tradeoff.
20.9. Decision surfaces: Playground
Refer: playground.tensorflow.org
20.10. Tensorflow and Keras.
20.10.1. Introduction to tensorflow
20.10.2. Introduction to Keras.
20.10.3. Computational Graph.
20.10.4. Building and MLP from scratch.
20.10.5. MLP in Keras.
20.10.6. GPU vs CPU.
20.10.7. GPUs for Deep Learning.
20.11. Assignment: MNIST using MLP.

21. Deep Multi-layer perceptrons

21.1. Dropout layers & Regularization.
21.2. Rectified Linear Units (ReLU).
21.3. Batch Normalization.
21.4. Optimizers:
21.4.1. Local and Global Optima.
21.4.2. ADAM
21.4.3. RMSProp
21.4.4. AdaGrad
21.5. Gradient Checking.
21.6. Initialization of models.
21.7. Softmax and Cross-entropy for multi-class classification.
21.8. Code Sample: MNIST
21.9. Auto Encoders.
21.10. Word2Vec.

22. Convolutional Neural Nets.

22.1. MLPs on Image data
22.2. Convolution operator.
22.3. Edge Detection on images.
22.4. Convolutional layer.
22.5. Max Pooling & Padding.
22.6. Imagenet dataset.
22.7. AlexNet.
22.8. Residual Network.

22.9. Inception Network.
22.10. Transfer Learning: Reusing existing models.
22.10.1. How to reuse state of the art models for your problem.
22.10.2. Data Augmentation.
22.10.3. Code example: Cats vs Dogs.

23. Long Short-term memory (LSTMs)

23.1. Recurrent Neural Network.
23.2. Backpropagation for RNNs.
23.3. Memory units.
23.4. LSTM.
23.5. GRUs.
23.6. Bidirectional RNN.
23.7. Code example: Predict a stock price using LSTM.
23.8. Sequence to Sequence Models.

24. Case Study: Personalized Cancer Diagnosis.

24.1. Business/Real world problem
24.1.1. Overview.
24.1.2. Business objectives and constraints.
24.2. ML problem formulation
24.2.1. Data
24.2.2. Mapping real world to ML problem.
24.2.3. Train, CV and Test data construction.
24.3. Exploratory Data Analysis
24.3.1. Reading data & preprocessing
24.3.2. Distribution of Class-labels.
24.3.3. “Random” Model.
24.3.4. Univariate Analysis
24.3.4.1. Gene feature.
24.3.4.2. Variation Feature.
24.3.4.3. Text feature.
24.4. Machine Learning Models
24.4.1. Data preparation.
24.4.2. Baseline Model: Naive Bayes
24.4.3. K-Nearest Neighbors Classification.
24.4.4. Logistic Regression with class balancing
24.4.5. Logistic Regression without class balancing
24.4.6. Linear-SVM.
24.4.7. Random-Forest with one-hot encoded features
24.4.8. Random-Forest with response-coded features
24.4.9. Stacking Classifier
24.4.10. Majority Voting classifier.

24.5. Assignments.

25. Taxi demand prediction in New York City.

- 25.1. Business/Real world problem.
  - 25.1.1. Overview
  - 25.1.2. Objectives and Constraints
- 25.2. Mapping to ML problem
  - 25.2.1. Data
  - 25.2.2. dask dataframes
  - 25.2.3. Fields/Features.
  - 25.2.4. Time series forecasting/Regression.
  - 25.2.5. Performance metrics.
- 25.3. Data Cleaning
  - 25.3.1. Latitude and Longitude data
  - 25.3.2. Trip Duration.
  - 25.3.3. Speed.
  - 25.3.4. Distance.
  - 25.3.5. Fare.
  - 25.3.6. Remove all outliers/erroneous points.
- 25.4. Data Preparation
  - 25.4.1. Clustering/Segmentation
  - 25.4.2. Time binning
  - 25.4.3. Smoothing time-series data.
  - 25.4.4. Time series and Fourier transforms.
- 25.5. Baseline models
  - 25.5.1. Ratios and previous-time-bin values.
  - 25.5.2. Simple moving average.
  - 25.5.3. Weighted Moving average.
  - 25.5.4. Exponential weighted moving average.
  - 25.5.5. Results.
- 25.6. Regression models:
  - 25.6.1. Train-Test split & Features
  - 25.6.2. Linear regression.
  - 25.6.3. Random Forest regression.
  - 25.6.4. Xgboost Regression.
  - 25.6.5. Model comparison.
- 25.7. Assignment.

26. Case Study: Microsoft Malware Detection

- 26.1. Business/real world problem
  - 26.1.1. Problem definition
  - 26.1.2. Objectives and constraints.
- 26.2. Machine Learning problem mapping
  - 26.2.1. Data overview.
  - 26.2.2. ML problem.
  - 26.2.3. Train and test splitting.

- 26.3. Exploratory Data Analysis
  - 26.3.1. Class distribution.
  - 26.3.2. Feature extraction from byte files
  - 26.3.3. Multivariate analysis of features from byte files.
  - 26.3.4. Train-Test class distributions
- 26.4. ML models - using byte files only
  - 26.4.1. Random Model.
  - 26.4.2. k-NN
  - 26.4.3. Logistic regression
  - 26.4.4. Random Forest & Xgboost
- 26.5. ASM Files
  - 26.5.1. Feature extraction & Multi-threading.
  - 26.5.2. File-size feature
  - 26.5.3. Univariate analysis on ASM features.
  - 26.5.4. t-SNE analysis.
  - 26.5.5. ML models on ASM file features
- 26.6. Models on all features
  - 26.6.1. t-SNE
  - 26.6.2. RandomForest and Xgboost
- 26.7. Assignments.

27. Case studies/Projects:

- 27.1. Amazon fashion discovery engine.
- 27.2. Song Similarity engine.
- 27.3. Predict customer propensity to purchase using CRM data.
- 27.4. Suggest me a movie to watch: Netflix Prize.
- 27.5. Human Activity Recognition using mobile phone’s accelerometer and gyroscope data.
- 27.6. Which ad to show to which user: Ad Click prediction.