“Appreciate existence of” 🡪 sufficient to know that it exists and roughly what it does, not necessary to know how or why it works

|  |  |  |
| --- | --- | --- |
| **Domain** | **Topics to be covered** | **Topics not to be covered** |
| Foundation  (2 weeks) | 1. Mathematics 2. Basic linear algebra 3. Basic probability and statistics (including Bayes’ Theorem) 4. Programming 5. Markdown 6. Python 3 and regex    * Google’s Python Class    * Official Python Tutorial 1 to 8, 9.8 to 9.10         (for officers without Python knowledge)   * + Do we want to give an exemplar regex cheat sheet?  1. Unix tools and scripting    * CS2043 lectures 1 to 8 2. Basic SQL    * How to use oracle’s XML tables    * Kaggle’s SQL tutorial?    * Use SQLite since it’s easier to play with? Or use some oracle DB? 3. Joel Spolsky’s article on Unicode 4. Avery’s comments on code style:  * Pls give important variables names that describe what they are * Pls break your code into logical chunks (ideally, reusable functions / classes) * Pls use a code formatter (any formatter)  1. Distance Measures 2. Fixed-length vectors    * Manhattan / Euclidean distances    * Hamming distance    * Cosine similarity, dot product 3. Variable-length strings    * Edit / Levenshtein distance (longest common *subsequence*)    * BLEU score    * Appreciate the existence of:  * Edit distance backtracking * METEOR * ROUGE * Damerau-Levenshtein distance  1. Variable-size sets    * Jaccard similarity (intersection over union) | 1. Advanced regex  * Backreferences * Positive / negative look-ahead * Positive / negative look-behind * Non-capturing groups * Efficiency * Character classes  1. How to install and set up Linux  * How to install GPU drivers * How to setup your network config  1. Less-common string algorithms  * Longest common *substring* * Sequence alignment / warping * Multiple string search (e.g. Aho Corasick) * Fuzzy string matching |
| Machine Learning  (1 month) | 1. Supervised Learning 2. Regression    * Linear, ridge, lasso    * Logistic 3. Classification    * KNN    * Support vector machine (linear SVM for binary classification)    * Decision trees 4. Ensemble    * Boosting (e.g. XGBoost, LightGBM)    * Stacking 5. Metrics    * Confusion matrix    * Recall, precision, F1 score    * AUC & ROC curve    * Micro and macro averaging    * How to choose a threshold 6. Optimization    * Loss function    * Bias / variance trade-off    * What is over-fitting and under-fitting    * Regularization    * Cross-validation, hyperparameter tuning 7. Unsupervised Learning 8. Clustering    * K-means    * EM GMM    * Density    * Hierarchical 9. Anomaly detection    * Parametric    * Non-parametric 10. Topic modelling     * LSA (or NMF; these are kind of based on factorization)     * LDA (based on generative models) 11. Frequent itemset mining 12. Reinforcement Learning & Genetic Algorithms 13. Know that they exist and appreciate their existence 14. Why they are different from supervised / unsupervised 15. And BTW we don’t use or need them (yet) 16. Neural Networks (still work in progress) 17. Backpropagation 18. Deep neural networks     * Feedforward     * CNN, pooling, cross-entropy, softmax     * RNN, LSTM     * Dropout 19. Tutorials Capstone Project 20. Pandas 21. Visualization (charts & graphs) 22. Titanic with walkthroughs     * Optional for those who have done it before     * Data exploration and data cleaning  * Dealing with missing values * Finding and fixing outliers   + Featurization * Categorical to one-hot encoding * Removing highly-correlated features to reduce dimensionality  1. Presentation to group covering process, outcome, and learning points | 1. “Advanced” regressions  * Isotonic * Heteroskedastic * Kernel * Quadratic / polynomial * Robust regressions (e.g. repeated median regression)  1. Non-linear and non-binary SVMs  * Kernels * Ranking * Ordinal * Multiclass  1. Curse of dimensionality  * VC dimension * Auto-encoders * Dimensionality reduction * PCA  1. Other ML topics  * Semi-supervised learning (e.g. LLDA) * Self-organising maps (clustering) * Association rules * MLE, maxent * GANs * Expectation maximisation * Working with unbalanced data * Embeddings * Recommendation systems  1. KL / JS divergence 2. Time-series data |
| Natural Language Processing  (1 month) | 1. Text Preprocessing (NLTK) 2. Tokenization and segmentation (and what is an n-gram) 3. Stemming and lemmatization 4. POS tagging 5. Bag of words / bag of n-grams 6. Word Embeddings (deeplearning.ai) 7. Word Mover’s Distance 8. Please appreciate the existence of:    * Sentence / document embeddings (e.g. doc2vec, Google’s USE)    * Cross-lingual embeddings (e.g. RCSLS, LASER) 9. Language Modelling (Stanford NLP) 10. HMMs 11. N-grams 12. OOV and simple smoothing methods 13. Interpolation and back-off 14. Text Classification and Sentiment Analysis (Stanford NLP) 15. Naïve Bayes classifier 16. Named Entity Recognition (Stanford NLP) 17. As a sequence labelling problem     * Greedy     * Beam     * Viterbi 18. No need to go into CRFs 19. Information Retrieval and Ranked IR (Stanford NLP) 20. Inverted index 21. TF-IDF 22. Appreciate the existence of:     * Locality sensitive hashing (e.g. minhash vectors)     * Variants of TF-IDF (SMART IR system, Okapi BM25)     * ANNOY or FAISS (for indexing vectors) 23. Neural Networks for NLP (optional) 24. Transformers & Muppet models 25. \*Capstone Project 26. Kaggle Avito (or new project?)     * MUST use word vectors     * MUST also use TF-IDF or edit / lev distance     * Images not necessary 27. Presentation to group as usual | 1. Linguistics  * Grammar and syntax parsing (CFGs, CYK) * Semantics * Polysemy / metonymy (word sense disambiguation) * Semiotics (e.g. emoji) * Morphology * Ontology  1. Machine translation  * IBM models * Back-translation * Syntax-based SMT * Corpus alignment / crawling / creation * Word vector alignment (e.g. Procrustes)  1. Hyperbolic / elliptic embeddings 2. Information extraction / relation extraction 3. Question answering 4. Summarization / simplification 5. How do CRFs work (used for NER tagging) 6. Byte pair encoding and subword tokenization (e.g. sentencepiece) 7. Spelling / grammar correction |
| Automatic Speech Recognition [optional for ATR]  (1 month) | 1. Fundamentals of Speech Recognition (Stanford CS224S Lecture 1) 2. Language Modelling (Edinburgh) 3. Lexicon and language model 4. Acoustic Modelling and Decoding 5. HMM, forward-backward, and Viterbi (Stanford CS224S Lecture 3) 6. Word error rate, training, and advanced decoding (Stanford CS224S Lecture 4)    * How to weight the AM and LM    * Viterbi beam decoding    * Multi-pass decoding e.g. N-best lists, lattices, word graphs, meshes / confusion networks    * Finite state methods 7. GMM acoustic modelling and feature extraction (Stanford CS224S Lecture 5) 8. Neural network acoustic models in speech recognition (Stanford CS224S Lecture 7) 9. End-to-End Neural Network Speech Recognition (Stanford CS224S Lecture 8) (optional) 10. Kaldi Tutorial 11. SAGE Tutorial (this will be imported into Confluence because it can’t be put on GitHub) |  |