

# APPLIED AI COURSE SYLLABUS

## Module 1 : Fundamentals of Programming

### Chapter 1 : How to utilise Applied AI Course?

- 1.1 How to learn from Applied AI Course?
- 1.2 How does the Job Guarantee Program works?

### Chapter 2 : Python for Data Science: Introduction

- 2.1 Python, Anaconda and relevant packages installations
- 2.2 Why learn Python?
- 2.3 Keywords and Identifiers
- 2.4 Comments, Indentation, and Statements
- 2.5 Variables and Datatypes in Python
- 2.6 Standard Input and Output
- 2.7 Operators
- 2.8 Control flow: If...else
- 2.9 Control flow: while loop
- 2.10 Control flow: for loop
- 2.11 Control flow: break and continue

### Chapter 3 : Python for Data Science: Data Structures

- 3.1 Lists
- 3.2 Tuples part 1
- 3.3 Tuples part 2
- 3.4 Sets
- 3.5 Dictionary
- 3.6 Strings

### Chapter 4 : Python for Data Science: Functions

- 4.1 Introduction
- 4.2 Types of Functions
- 4.3 Function Arguments
- 4.4 Recursive Functions
- 4.5 Lambda Functions
- 4.6 Modules
- 4.7 Packages
- 4.8 File Handling
- 4.9 Exception Handling
- 4.10 Debugging Python
- 4.11 **Optional Assignment 1**

### Chapter 5 : Python for Data Science: Functions

- 5.1 Introduction to NumPy.

## 5.2 Numerical operations.

### **Chapter 6 : Python for Data Science: Matplotlib**

#### 6.1 Introduction to Matplotlib

### **Chapter 7 : Python for Data Science: Pandas**

#### 7.1 Getting started with pandas

#### 7.2 Data Frame Basics

#### 7.3 Key Operations on Data Frames.

#### 7.4 **Optional Assignment 2**

### **Chapter 8 : Computational Complexity: an Introduction**

#### 8.1 Space and Time Complexity: Find the largest number in a list

#### 8.2 Binary search

#### 8.3 Find elements common in two lists.

#### 8.4 Find elements common in two lists using a Hashtable/Dict

### **Chapter 9 : SQL**

#### 9.1 Introduction to databases.

#### 9.2 Why SQL?

#### 9.3 Execution of an SQL statement.

#### 9.4 IMDB Dataset

#### 9.5 Installing MySQL

#### 9.6 Load IMDB data.

#### 9.7 Use, Describe, Show table.

#### 9.8 Select.

#### 9.9 Limit, Offset.

#### 9.10 Order By.

#### 9.11 Distinct.

#### 9.12 Where, Comparison Operators, NULL.

#### 9.13 Logic Operators.

#### 9.14 Aggregate Functions: COUNT, MIN, MAX, AVG, SUM.

#### 9.15 Group By.

#### 9.16 Having.

#### 9.17 Order of Keywords.

#### 9.18 Join and Natural Join.

#### 9.19 Inner, Left, Right, and Outer Joins.

#### 9.20 Sub Queries/Nested Queries/Inner Queries.

#### 9.21 DML: INSERT

#### 9.22 DML: UPDATE, DELETE

#### 9.23 DML: CREATE, TABLE

#### 9.24 DDL: ALTER, ADD, MODIFY, DROP

#### 9.25 DDL: DROP TABLE, TRUNCATE, DELETE

#### 9.26 Data Control Language: GRANT, REVOKE

- 9.27 Learning Resources.
- 9.28 **Assignment-22: SQL Assignment on IMDB data**

## **Module 2: Data Science: Exploratory Data Analysis and Data Visualization**

### **Chapter 1 : Plotting for exploratory data analysis (EDA)**

- 10.1 Introduction to Iris dataset and 2D scatter-plot
- 10.2 3D Scatter-plot.
- 10.3 Pair plots.
- 10.4 Limitations of Pair plots
- 10.5 Histogram and introduction to PDF(Probability Density Function)
- 10.6 Univariate analysis using PDF.
- 10.7 CDF(Cumulative distribution function)
- 10.8 Variance, Standard Deviation
- 10.9 Median
- 10.10 Percentiles and Quantiles
- 10.11 IQR(InterQuartile Range), MAD(Median Absolute Deviation).
- 10.12 Box-plot with whiskers
- 10.13 Violin plots.
- 10.14 Summarizing plots, Univariate, Bivariate, and Multivariate analysis.
- 10.15 Multivariate probability density, contour plot.
- 10.16 **Assignment 1: Perform EDA on Haberman dataset.**

### **Chapter 2 : Linear Algebra**

- 11.1 Why learn it?
- 11.2 Introduction to Vectors (2-D, 3-D, n-D), row vectors and column vector
- 11.3 Dot product and the angle between 2 vectors.
- 11.4 Projection and unit vector
- 11.5 Equation of a line (2-D), plane(3-D) and hyperplane (n-D)
- 11.6 Distance of a point from a plane/hyperplane, half-spaces.
- 11.7 Equation of a circle (2-D), sphere (3-D) and hypersphere (n-D)
- 11.8 Equation of an ellipse (2-D), ellipsoid (3-D) and hyperellipsoid (n-D)
- 11.9 Square, Rectangle.
- 11.10 Hypercube, Hyper cuboid.
- 11.11 Revision Questions

### **Chapter 3 : Probability and Statistics**

- 12.1 Introduction to Probability and Statistics.
- 12.2 Population & Sample.
- 12.3 Gaussian/Normal Distribution and its PDF(Probability Density Function).

- 12.4 CDF(Cumulative Density Function) of Gaussian/Normal Distribution
- 12.5 Symmetric distribution, Skewness, and Kurtosis
- 12.6 Standard normal variate (z) and standardization.
- 12.7 Kernel density estimation.
- 12.8 Sampling distribution & Central Limit Theorem.
- 12.9 Q-Q Plot: Is a given random variable Gaussian distributed?
- 12.10 How distributions are used?
- 12.11 Chebyshev's inequality
- 12.12 Discrete and Continuous Uniform distributions.
- 12.13 How to randomly sample data points. [Uniform Distribution]
- 12.14 Bernoulli and Binomial distribution
- 12.15 Log-normal
- 12.16 Power law distribution
- 12.17 Box-Cox transform.
- 12.18 Application of Non-Gaussian Distributions?
- 12.19 Co-variance
- 12.20 Pearson Correlation Coefficient
- 12.21 Spearman Rank Correlation Coefficient
- 12.22 Correlation vs Causation
- 12.23 How to use Correlations?
- 12.24 Confidence Intervals(C.I) Introduction
- 12.25 Confidence Interval vs Point estimate.
- 12.26 Computing confidence-interval has given the underlying distribution.
- 12.27 C.I for the mean of a normal random variable.
- 12.28 Confidence Interval using bootstrapping.
- 12.29 Hypothesis Testing methodology, Null-hypothesis, p-value
- 12.30 Hypothesis testing intuition with coin toss example
- 12.31 Resampling and permutation test.
- 12.32 K-S Test for the similarity of two distributions.
- 12.33 Code Snippet K-S Test.
- 12.34 Hypothesis Testing: another example.
- 12.35 Resampling and permutation test: another example.
- 12.36 How to use Hypothesis testing?
- 12.37 Proportional Sampling.
- 12.38 Revision Questions.

#### **Chapter 4 : Interview Questions on Probability and Statistics**

- 13.1 Question & Answers

#### **Chapter 5 : Dimensionality reduction and Visualization:**

- 14.1 What is dimensionality reduction?
- 14.2 Row vector, and Column vector.
- 14.3 How to represent a dataset?
- 14.4 How to represent a dataset as a Matrix.
- 14.5 Data preprocessing: Feature Normalization

- 14.6 Mean of a data matrix.
- 14.7 Data preprocessing: Column Standardization
- 14.8 Co-variance of a Data Matrix.
- 14.9 MNIST dataset (784 dimensional)
- 14.10 Code to load MNIST data set.

#### **Chapter 6 : Principal Component Analysis.**

- 15.1 Why learn it.
- 15.2 Geometric intuition.
- 15.3 Mathematical objective function.
- 15.4 Alternative formulation of PCA: distance minimization
- 15.5 Eigenvalues and eigenvectors.
- 15.6 PCA for dimensionality reduction and visualization.
- 15.7 Visualize MNIST dataset.
- 15.8 Limitations of PCA
- 15.9 Code example.
- 15.10 PCA for dimensionality reduction (not-visualization)

#### **Chapter 7 : T-distributed stochastic neighborhood embedding (t-SNE)**

- 16.1 What is t-SNE?
- 16.2 Neighborhood of a point, Embedding.
- 16.3 Geometric intuition.
- 16.4 Crowding problem.
- 16.5 How to apply t-SNE and interpret its output (distill.pub)
- 16.6 t-SNE on MNIST.
- 16.7 Code example.
- 16.8 Revision Questions.

#### **Chapter 8 : Interview Questions on Dimensionality Reduction**

- 17.1 Question & Answers

### **Module 3 : Foundations of Natural Language Processing and Machine Learning**

#### **Chapter 1 : Real world problem: Predict rating given product reviews on Amazon.**

- 18.1 Dataset overview: Amazon Fine Food reviews(EDA)
- 18.2 Data Cleaning: Deduplication.
- 18.3 Why convert text to a vector?
- 18.4 Bag of Words (BoW)
- 18.5 Text Preprocessing: Stemming, Stop-word removal, Tokenization, Lemmatization
- 18.6 uni-gram, bi-gram, n-grams.
- 18.7 tf-idf (term frequency- inverse document frequency)
- 18.8 Why use the log in IDF?

- 18.9 Word2Vec.
- 18.10 Avg-Word2Vec, tf-idf weighted Word2Vec
- 18.11 Bag of Words(code sample)
- 18.12 Text Preprocessing(code sample)
- 18.13 Bi-Grams and n-grams(code sample)
- 18.14 TF-IDF(code sample)
- 18.15 Word2Vec(code sample)
- 18.16 Avg-Word2Vec and TFIDF-Word2Vec(Code Sample)
- 18.17 **Assignment 2: Apply t-SNE.**

## **Chapter 2 : Classification and Regression Models: K-Nearest Neighbors**

- 19.1 How "Classification" works?
- 19.2 Data matrix notation.
- 19.3 Classification vs Regression (examples)
- 19.4 K-Nearest Neighbors Geometric intuition with a toy example.
- 19.5 Failure cases of K-NN
- 19.6 Distance measures: Euclidean(L2) , Manhattan(L1), Minkowski, Hamming
- 19.7 Cosine Distance & Cosine Similarity
- 19.8 How to measure the effectiveness of k-NN?
- 19.9 Test/Evaluation time and space complexity.
- 19.10 k-NN Limitations.
- 19.11 Decision surface for K-NN as K changes.
- 19.12 Overfitting and Underfitting.
- 19.13 Need for Cross validation.
- 19.14 K-fold cross validation.
- 19.15 Visualizing train, validation and test datasets
- 19.16 How to determine overfitting and underfitting?
- 19.17 Time based splitting
- 19.18 k-NN for regression.
- 19.19 Weighted k-NN
- 19.20 Voronoi diagram.
- 19.21 Binary search tree
- 19.22 How to build a kd-tree.
- 19.23 Find nearest neighbors using kd-tree
- 19.24 Limitations of kd-tree
- 19.25 Extensions.
- 19.26 Hashing vs LSH.
- 19.27 LSH for cosine similarity
- 19.28 LSH for euclidean distance.
- 19.29 Probabilistic class label
- 19.30 Code Sample: Decision boundary.
- 19.31 Code Samples: Cross-Validation
- 19.32 Revision Questions

### **Chapter 3 : Interview Questions on k-NN**

#### **20.1 Question & Answers**

### **Chapter 4 : Classification algorithms in various situations:**

- 21.1 Introduction
- 21.2 Imbalanced vs balanced dataset.
- 21.3 Multi-class classification.
- 21.4 k-NN, given a distance or similarity matrix
- 21.5 Train and test set differences.
- 21.6 Impact of Outliers
- 21.7 Local Outlier Factor(Simple solution: mean distance to k-NN).
- 21.8 k-distance (A), N(A)
- 21.9 reachability-distance(A, B)
- 21.10 Local-reachability-density(A)
- 21.11 Local Outlier Factor(A)
- 21.12 Impact of Scale & Column standardization.
- 21.13 Interpretability
- 21.14 Feature importance & Forward Feature Selection
- 21.15 Handling categorical and numerical features.
- 21.16 Handling missing values by imputation.
- 21.17 Curse of dimensionality.
- 21.18 Bias-Variance tradeoff.
- 21.19 Intuitive understanding of bias-variance.
- 21.20 Revision Questions.
- 21.21 Best and worst case of an algorithm.

### **Chapter 5 : Performance measurement of models:**

- 22.1 Accuracy
- 22.2 Confusion matrix, TPR, FPR, FNR, TNR
- 22.3 Precision & recall, F1-score.
- 22.4 Receiver Operating Characteristic Curve (ROC) curve and AUC.
- 22.5 Log-loss.
- 22.6 R-Squared/ Coefficient of determination.
- 22.7 Median absolute deviation (MAD)
- 22.8 Distribution of errors.
- 22.9 **Assignment-3: Apply k-Nearest Neighbor**
- 22.10 Revision Questions

### **Chapter 6 : Interview Questions on Performance Measurement models.**

#### **23.1 Question & Answers**

### **Chapter 7 : Naive Bayes**

- 24.1 Conditional probability.
- 24.2 Independent vs Mutually exclusive events.
- 24.3 Bayes Theorem with examples.

- 24.4 Exercise problems on Bayes Theorem
- 24.5 Naive Bayes algorithm.
- 24.6 Toy example: Train and test stages.
- 24.7 Naive Bayes on Text data.
- 27.8 Laplace/Additive Smoothing.
- 24.9 Log-probabilities for numerical stability.
- 24.10 Bias and Variance tradeoff.
- 24.11 Feature importance and interpretability.
- 24.12 Imbalanced data
- 24.13 Outliers.
- 24.14 Missing values.
- 24.15 Handling Numerical features (Gaussian NB)
- 24.16 Multiclass classification.
- 24.17 Similarity or Distance matrix.
- 24.18 Large dimensionality.
- 24.19 Best and worst cases.
- 24.20 Code example
- 24.21 **Assignment-4: Apply Naive Bayes**
- 24.22 Exercise: Apply Naive Bayes to Amazon reviews.

## **Chapter 8 : Logistic Regression:**

- 25.1 Geometric intuition of logistic regression
- 25.2 Sigmoid function: Squashing
- 25.3 Mathematical formulation of objective function.
- 25.4 Weight Vector.
- 25.5 L2 Regularization: Overfitting and Underfitting.
- 25.6 L1 regularization and sparsity.
- 25.7 Probabilistic Interpretation: Gaussian Naive Bayes
- 25.8 Loss minimization interpretation
- 25.9 Hyperparameter search: Grid Search and Random Search
- 25.10 Column Standardization.
- 25.11 Feature importance and model interpretability.
- 25.12 Collinearity of features.
- 25.13 Train & Run time space and time complexity.
- 25.14 Real world cases.
- 25.15 Non-linearly separable data & feature engineering.
- 25.16 Code sample: Logistic regression, GridSearchCV, RandomSearchCV
- 25.17 **Assignment-5: Apply Logistic Regression.**
- 25.18 Extensions to Logistic Regression: Generalized linear models (GLM)

## **Chapter 9 : Linear Regression.**

- 26.1 Geometric intuition of Linear Regression.
- 26.2 Mathematical formulation.
- 26.3 Real world Cases.
- 26.4 Code sample for Linear Regression



## **Chapter 10 : Solving optimization problems**

- 27.1 Differentiation.
- 27.2 Online differentiation tools
- 27.3 Maxima and Minima
- 27.4 Vector calculus: Grad
- 27.5 Gradient descent: geometric intuition.
- 27.6 Learning rate.
- 27.7 Gradient descent for linear regression.
- 27.8 SGD algorithm
- 27.9 Constrained optimization & PCA
- 27.10 Logistic regression formulation revisited.
- 27.11 Why L1 regularization creates sparsity?
- 27.12 **Assignment-6: Implement SGD for linear regression**
- 27.13 Revision Questions.

## **Chapter 11 : Interview questions on Logistic Regression and Linear Regression**

- 28.1 Question & Answers

## **Module 4 : Machine Learning- II (Supervised Learning Models)**

### **Chapter 1 : Support Vector Machines (SVM)**

- 29.1 Geometric intuition.
- 29.2 Mathematical derivation.
- 29.3 why we take values +1 and -1 for support vector planes
- 29.4 Loss function(Hinge Loss) based interpretation.
- 29.5 Dual form of SVM formulation.
- 29.6 Kernel trick.
- 29.7 Polynomial kernel.
- 29.8 RBF-Kernel.
- 29.9 Domain specific Kernels.
- 29.10 Train and run time complexities.
- 29.11 nu-SVM: control errors and support vectors.
- 29.12 SVM Regression.
- 29.13 Cases.
- 29.14 Code Sample.
- 29.15 **Assignment-7: Apply SVM**
- 29.16 Revision Questions.

### **Chapter 2 : Interview Questions on Support Vector Machine**

- 30.1 Questions & Answers

### **Chapter 3 : Decision Trees**

- 31.1 Geometric Intuition of decision tree: Axis parallel hyperplanes.
- 31.2 Sample Decision tree.
- 31.3 Building a decision Tree: Entropy(Intuition behind entropy)
- 31.4 Building a decision Tree: Information Gain
- 31.5 Building a decision Tree: Gini Impurity.
- 31.6 Building a decision Tree: Constructing a DT.
- 31.7 Building a decision Tree: Splitting numerical features.
- 31.8 Feature standardization.
- 31.9 Categorical features with many possible values.
- 31.10 Overfitting and Underfitting.
- 31.11 Train and Run time complexity.
- 31.12 Regression using Decision Trees.
- 31.13 Cases
- 31.14 Code Samples.
- 31.15 **Assignment-8: Apply Decision Trees.**
- 31.16 Revision questions

#### **Chapter 4 : Interview Questions on Decision Trees.**

- 32.1 Question & Answers

#### **Chapter 5 : Ensemble Models:**

- 33.1 What are ensembles?
- 33.2 Bootstrapped Aggregation (Bagging) Intuition.
- 33.3 Random Forest and their construction.
- 33.4 Bias-Variance tradeoff.
- 33.5 Train and Run-time Complexity.
- 33.6 Bagging: code Sample.
- 33.7 Extremely randomized trees.
- 33.8 Random Forest: Cases.
- 33.9 Boosting Intuition
- 33.10 Residuals, Loss functions, and gradients.
- 33.11 Gradient Boosting
- 33.12 Regularization by Shrinkage.
- 33.13 Train and Run time complexity.
- 33.14 XGBoost: Boosting + Randomization
- 33.15 AdaBoost: geometric intuition.
- 33.16 Stacking models.
- 33.17 Cascading classifiers.
- 33.18 Kaggle competitions vs Real world.
- 33.19 **Assignment-9: Apply Random Forests & GBDT**
- 33.20 Revision Questions.

## **Module 5 : Feature Engineering, Productionisation and deployment of ML Models**

### **Chapter 1 : Featurizations and Feature engineering.**

- 34.1 Introduction.
- 34.2 Moving window for Time-series data.
- 34.3 Fourier decomposition.
- 34.4 Deep learning features: LSTM
- 34.5 Image histogram.
- 34.6 Key points: SIFT.
- 34.7 Deep learning features: CNN
- 34.8 Relational data.
- 34.9 Graph data.
- 34.10 Indicator variables.
- 34.11 Feature binning.
- 34.12 Interaction variables.
- 34.13 Mathematical transforms.
- 34.14 Model specific featurizations.
- 34.15 Feature orthogonality.
- 34.16 Domain specific featurizations.
- 34.17 Feature slicing.
- 34.18 Kaggle Winner's solutions.

### **Chapter 2 : Miscellaneous Topics**

- 35.1 Calibration of Models: Need for calibration.
- 35.2 Calibration Plots.
- 35.3 Platt's Calibration/Scaling.
- 35.4 Isotonic Regression.
- 35.5 Code Samples.
- 35.6 Modeling in the presence of outliers: RANSAC.
- 35.7 Productionizing models.
- 35.8 Retraining models periodically.
- 35.9 A/B testing.
- 35.10 Data Science Life Cycle.
- 35.11 Production and Deployment of Machine Learning Models.
- 35.12 VC Dimension.

## **Module 6 : Machine Learning Real-World Case Studies**

### **Chapter 1 : Case study 1 : Quora Question pair similarity problem**

- 36.1 Business/Real world problem : Problem Definition
- 36.2 Business objectives and constraints.
- 36.3 Mapping to an ML problem: Data overview
- 36.4 Mapping to an ML problem: ML problem and performance metric.
- 36.5 Mapping to an ML problem: Train-test split
- 36.6 EDA: Basic Statistics.
- 36.7 EDA: Basic Feature Extraction.
- 36.8 EDA: Text Preprocessing.
- 36.9 EDA: Advanced Feature Extraction.
- 36.10 EDA: Feature analysis.
- 36.11 EDA: Data Visualization: T-SNE.
- 36.12 EDA: TF-IDF weighted word-vector featurization.
- 36.13 ML Models: Loading data.
- 36.14 ML Models: Random Model.
- 36.15 ML Models: Logistic Regression & Linear SVM
- 36.16 ML Models: XGBoost
- 36.17 **Assignment - Quora Question pair similarity.**

### **Chapter 2 : Case study 2 : Personalized Cancer Diagnosis**

- 37.1 Business/Real world problem overview
- 37.2 Business objectives and constraints.
- 37.3 ML problem formulation: Data
- 37.4 ML problem formulation: Mapping real world to ML problem.
- 37.5 ML problem formulation: Train, CV and Test data construction.
- 37.6 Exploratory Data Analysis: Reading data & preprocessing
- 37.7 Exploratory Data Analysis: Distribution of Class-labels.
- 37.8 Exploratory Data Analysis: "Random" Model.
- 37.9 Univariate Analysis: Gene feature.
- 37.10 Univariate Analysis: Variation Feature.
- 37.11 Univariate Analysis: Text feature.
- 37.12 Machine Learning Models: Data Preparation
- 37.13 Baseline Model: Naive Bayes
- 37.14 K-Nearest Neighbors Classification.
- 37.15 Logistic Regression with class balancing
- 37.16 Logistic Regression without class balancing
- 37.17 Linear-SVM.
- 37.18 Random-Forest with one-hot encoded features
- 37.19 Random-Forest with response-coded features
- 37.20 Stacking Classifier
- 37.21 Majority Voting classifier.
- 37.22 **Assignment - Personalized Cancer Diagnosis**

### **Chapter 3 : Case Study 3 : Facebook Friend Recommendation using Graph mining.**

- 38.1 Problem Definition.
- 38.2 Overview of graphs: Node/Vertex, edge/link, directed edge, path.
- 38.3 Data Format & Limitations.
- 38.4 Mapping to a supervised classification problem.
- 38.5 Business Constraints & Metrics.
- 38.6 EDA: Basic Stats.
- 38.7 EDA: Follower and following stats.
- 38.8 EDA: Binary Classification Tasks.
- 38.9 EDA: Train and test split.
- 38.10 Feature engineering on graphs: Jaccard & Cosine similarities.
- 38.11 PageRank.
- 38.12 Shortest Path.
- 38.13 Connected-Components.
- 38.14 Adar index.
- 38.15 Katz Centrality.
- 38.16 HITS Score.
- 38.17 SVD.
- 38.18 Weight Features.
- 38.19 Modeling.

### **Chapter 4 : Case study 4:Taxi demand prediction in New York City.**

- 39.1 Business/Real world problem overview.
- 39.2 Objectives and Constraints
- 39.3 Mapping to ML problem: Data
- 39.4 Mapping to ML problem: dask dataframes
- 39.5 Mapping to ML problem: Fields/Features.
- 39.6 Mapping to ML problem: Time series forecasting/Regression.
- 39.7 Mapping to ML problem: Performance metrics.
- 39.8 Data Cleaning: Latitude and Longitude data
- 39.9 Data Cleaning: Trip Duration.
- 39.10 Data Cleaning: Speed.
- 39.11 Data Cleaning: Distance.
- 39.12 Data Cleaning: Fare.
- 39.13 Data Cleaning: Remove all outliers/erroneous points.
- 39.14 Data Preparation: Clustering/Segmentation
- 39.15 Data Preparation: Time binning
- 39.16 Data Preparation: Smoothing time-series data.
- 39.17 Data Preparation: Smoothing time-series data cont..
- 39.18 Data Preparation: Time series and Fourier transforms.
- 39.19 Ratios and previous-time-bin values.
- 39.20 Simple moving average.
- 39.21 Weighted Moving average.
- 39.22 Exponential weighted moving average.

- 39.23 Results.
- 39.24 Regression models: Train-Test split & Features
- 39.25 Linear regression.
- 39.26 Random Forest regression.
- 39.27 Xgboost Regression.
- 39.28 Model comparison.
- 39.29 **Assignment- Taxi Demand Prediction**

## **Chapter 5 : Case Study 5 : Stackoverflow Tag Predictor**

- 40.1 Business/Real world problem.
- 40.2 Business objectives and constraints.
- 40.3 Mapping to an ML problem: Data Overview
- 40.4 Mapping to an ML problem: ML problem formulation.
- 40.5 Mapping to an ML problem: Performance metrics.
- 40.6 Hamming loss.
- 40.7 EDA: Data Loading.
- 40.8 EDA: Analysis of tags.
- 40.9 EDA: Data Preprocessing.
- 40.10 Data Modeling: Multi label Classification.
- 40.11 Data Preparation.
- 40.12 Train-Test Split.
- 40.13 Featurization.
- 40.14 Logistic Regression: One Vs Rest.
- 40.15 Sampling data and tags + Weighted Models.
- 40.16 Logistic Regression revisited.
- 40.17 Why not use advanced techniques?
- 40.18 **Assignment - Stack Overflow tagging**

## **Chapter 6 : Case Study 6 : Microsoft Malware Detection**

- 41.1 Business/Real world problem: Problem Definition.
- 41.2 Business/Real world problem: objectives and constraints.
- 41.3 Machine Learning Problem Mapping: Data Overview.
- 41.4 Machine Learning Problem Mapping: ML Problem.
- 41.5 Machine Learning Problem Mapping: Train and test splitting.
- 41.6 Exploratory Data Analysis: Class Distribution.
- 41.7 Exploratory Data Analysis: Feature extraction from byte files.
- 41.8 Exploratory Data Analysis: Multivariate analysis of features from byte files.
- 41.9 Exploratory Data Analysis: Train-Test class distribution.
- 41.10 ML models- using byte files only: Random Model.
- 41.11 k-NN.
- 41.12 Logistic regression.
- 41.13 Random Forest and Xgboost.
- 41.14 ASM Files: Feature extraction and Multiprocessing.
- 41.15 File-size feature.

- 41.16 Univariate Analysis.
- 41.17 t-SNE analysis.
- 41.18 ML models on ASM file features.
- 41.19 Models on all features: t-SNE.
- 41.20 Models on all features: Random Forest and Xgboost.
- 41.21 Assignments.

## **Chapter 7 : Case study 7 : AD-Click Prediction**

- 42.1 Live sessions on Ad-Click Prediction
- 42.2 Live sessions on Ad-Click Prediction(contd) and Performance Metrics

# **Module 7 : Data Mining(Unsupervised Learning) and Recommender Systems + Real -world Case Studies**

## **Chapter 1 : Unsupervised learning/Clustering**

- 43.1 What is Clustering?
- 43.2 Unsupervised learning
- 43.3 Applications.
- 43.4 Metrics for Clustering.
- 43.5 K-Means: Geometric intuition, Centroids.
- 43.6 K-Means: Mathematical formulation: Objective function
- 43.7 K-Means Algorithm.
- 43.8 How to initialize: K-Means++
- 43.9 Failure cases/Limitations.
- 43.10 K-Medoids
- 43.11 Determining the right K.
- 43.12 Code Samples.
- 43.13 Time and Space complexity.
- 43.14 **Assignment-10: Apply K-means, Agglomerative, DBSCAN clustering algorithms.**

## **Chapter 2 : Hierarchical clustering Technique**

- 44.1 Agglomerative & Divisive, Dendrograms
- 44.2 Agglomerative Clustering.
- 44.3 Proximity methods: Advantages and Limitations.
- 44.4 Time and Space Complexity.
- 44.5 Limitations of Hierarchical Clustering.
- 44.6 Code sample.
- 43.7 **Assignment-10: Apply K-means, Agglomerative, DBSCAN clustering algorithms.**

### **Chapter 3 : DBSCAN (Density based clustering)**

- 45.1 Density based clustering
- 45.2 MinPts and Eps: Density
- 45.3 Core, Border and Noise points.
- 45.4 Density edge and Density connected points.
- 45.5 DBSCAN Algorithm.
- 45.6 Hyper Parameters: MinPts and Eps.
- 45.7 Advantages and Limitations of DBSCAN.
- 45.8 Time and Space Complexity.
- 45.9 Code samples.
- 45.10 **Assignment-10: Apply K-means, Agglomerative, DBSCAN clustering algorithms.**
- 45.11 Revision Questions

### **Chapter 4 : Recommender Systems and Matrix Factorization.**

- 46.1 Problem formulation: Movie reviews.
- 46.2 Content based vs Collaborative Filtering.
- 46.3 Similarity based Algorithms.
- 46.4 Matrix Factorization: PCA, SVD.
- 46.5 Matrix Factorization: NMF.
- 46.6 Matrix Factorization for Collaborative filtering
- 46.7 Matrix Factorization for feature engineering.
- 46.8 Clustering as MF.
- 46.9 Hyperparameter tuning.
- 46.10 Matrix Factorization for recommender systems: Netflix Prize Solution.
- 46.11 Cold Start problem.
- 46.12 Word Vectors as MF.
- 46.13 Eigen-Faces.
- 46.14 Code example.
- 46.15 Assignment-11: Apply Truncated SVD.
- 46.16 Revision Questions.

### **Chapter 5 : Interview Questions on Recommender Systems and Matrix Factorization.**

- 47.1 Question & Answers.

### **Chapter 6 : Case Study 8 : Amazon Fashion Discovery Engine**

- 48.1 Problem Statement: Recommend similar apparel products in e-commerce using product descriptions and images.
- 48.2 Plan of action.
- 48.3 Amazon Product Advertising API.
- 48.4 Data Folders and Paths.
- 48.5 Overview of the data and terminology.
- 48.6 Data Cleaning and Understanding: Missing data in various features.
- 48.7 Understand Duplicate rows.
- 48.8 Remove duplicates: Part 1
- 48.9 Remove duplicates: Part 2



- 48.10 Text- Preprocessing: Tokenization and stop-word removal.
- 48.11 Stemming
- 48.12 Text-based product similarity: Converting text to an n-D vector: Bag Of Words.
- 48.13 Code for bag of words based product similarity
- 48.14 TF-IDF: featuring text based on word-importance.
- 48.15 Code for TF-IDF based product similarity.
- 48.16 Code for IDF based product similarity.
- 48.17 Text semantics based product similarity: Word2Vec(Featurizing text based on semantics similarity).
- 48.18 Code for average Word2Vec product similarity.
- 48.19 TF-IDF Weighted Word2Vec
- 48.20 Code for IDF weighted Word2Vec product similarity.
- 48.21 Weighted similarity using brand and color.
- 48.22 Code for weighted similarity.
- 48.23 Building a real-world solution.
- 48.24 Deep learning based visual product similarity: ConvNets: How to featurize an image: Edges, Shapes, and Parts.
- 48.25 Using Keras + Tensorflow to extract features.
- 48.26 Visual similarity based product similarity
- 48.27 Measuring goodness of our solution: A/B Testing.
- 48.28 **Assignment: Apparel Recommendation**

## **Chapter 7 : Case Study 9 : Netflix Movie Recommendation System**

- 49.1 Business/Real world problem: Problem Definition.
- 49.2 Objectives and constraints.
- 49.3 Mapping to an ML Problem: Data Overview.
- 49.4 Mapping to an ML Problem: ML Problem formulation.
- 49.5 Exploratory Data Analysis: Data Preprocessing.
- 49.6 Exploratory Data Analysis: Temporal Train-Test split.
- 49.7 Exploratory Data Analysis: Preliminary data analysis.
- 49.8 Exploratory Data Analysis: Sparse matrix representation.
- 49.9 Exploratory Data Analysis: Average rating for various slices.
- 49.10 Exploratory Data Analysis: Cold start problem.
- 49.12 Computing similarity Matrices: User-User similarity matrix.
- 49.13 Computing similarity Matrices: Movie-Movie similarity matrix
- 49.14 Computing similarity Matrices: Does movie-movie similarity work?
- 49.15 ML models: Surprise Library.
- 49.16 Overview of the modeling strategy.
- 49.17 Data Sampling.
- 49.18 Google drive with intermediate files.
- 49.19 Featurization for regression.
- 49.20 Data Transformation for surprise.
- 49.21 Xgboost with 13 features.

- 49.22 Surprize Baseline model.
- 49.23 Xgboost + 13 features + Surprize baseline model.
- 49.24 Surprize KNN predictors.
- 49.25 Matrix factorization models using surprise.
- 49.26 SVD++ with implicit feedback.
- 49.27 Final models with all features and predictors.
- 49.28 Comparison between various models.
- 49.29 Assignment: Netflix Prize

## **Module 8 : Neural Networks, Computer Vision and Deep Learning**

### **Chapter 1 : Deep Learning: Neural Networks.**

- 50.1 History of Neural networks and Deep Learning.
- 50.2 How Biological Neurons work?
- 50.3 Growth of biological neural networks.
- 50.4 Diagrammatic representation: Logistic Regression and Perceptron
- 50.5 Multi-Layered Perceptron (MLP).
- 50.6 Notation.
- 50.7 Training a single-neuron model.
- 50.8 Training an MLP: Chain rule
- 50.9 Training an MLP: Memoization
- 50.10 Backpropagation algorithm.
- 50.11 Activation functions.
- 50.12 Vanishing Gradient problem.
- 50.13 Bias-Variance tradeoff.
- 50.14 Decision surfaces: Playground

### **Chapter 2 : Deep Learning: Deep Multi-layer perceptrons**

- 51.1 Deep Multi-layer perceptrons: 1980s to 2010s
- 51.2 Dropout layers & Regularization.
- 51.3 Rectified Linear Units (ReLU).
- 51.4 Weight initialization.
- 51.5 Batch Normalization.
- 51.6 Optimizers: Hill-descent analogy in 2D
- 51.7 Optimizers: Hill descent in 3D and contours.
- 51.8 SGD recap.
- 51.9 Batch SGD with Momentum.
- 51.10 Nesterov Accelerated Gradient (NAG)
- 51.11 Optimizers: AdaGrad
- 51.12 Optimizers: Adadelta and RMSProp

- 51.13 Adam
- 51.14 Which algorithm to choose when?
- 51.15 Gradient Checking and Clipping.
- 51.16 Softmax and cross-entropy for multi-class classification.
- 51.17 How to train a Deep MLP?
- 51.18 Auto Encoders.
- 51.19 Word2Vec: CBOW.
- 51.20 Word2Vec: Skip-gram
- 51.21 Word2Vec: Algorithmic Optimizations.

### **Chapter 3 : Deep Learning: Tensorflow and Keras.**

- 52.1 Tensorflow and Keras Overview.
- 52.2 GPU vs CPU for Deep Learning.
- 52.3 Google Collaboratory.
- 52.4 Install TensorFlow.
- 52.5 Online documentation and tutorials.
- 52.6 Softmax Classifier on MNIST dataset.
- 52.7 MLP: Initialization
- 52.8 Model 1: Sigmoid activation.
- 52.9 Model 2: ReLU activation.
- 52.10 Model 3: Batch Normalization.
- 52.11 Model 4 : Dropout.
- 52.12 MNIST classification in Keras.
- 52.13 Hyperparameter tuning in Keras.
- 52.14 **Assignment: Try different MLP architectures on MNIST dataset.**

### **Chapter 4 : Deep Learning: Convolutional Neural Nets.**

- 53.1 Biological inspiration: Visual Cortex
- 53.2 Convolution: Edge Detection on images.
- 53.3 Convolution: Padding and strides
- 53.4 Convolution over RGB images.
- 53.5 Convolutional layer.
- 53.6 Max-pooling.
- 53.7 CNN Training: Optimization
- 53.8 Example CNN: LeNet [1998]
- 53.9 ImageNet dataset
- 53.10 Data Augmentation.
- 53.11 Convolution Layers in Keras
- 53.12 AlexNet
- 53.13 VGGNet
- 53.14 Residual Network.
- 53.15 Inception Network.

- 53.16 What is Transfer Learning?
- 53.17 Code example: Cats vs Dogs.
- 53.18 Code Example: MNIST dataset.
- 53.19 Assignment: Try various CNN networks on MNIST dataset.

## **Chapter 5 : Deep Learning: Long Short-Term Memory (LSTMS)**

- 54.1 Why RNNs?
- 54.2 Recurrent Neural Network.
- 54.3 Training RNNs: Backprop.
- 54.4 Types of RNNs.
- 54.5 Need for LSTM/GRU.
- 54.6 LSTM.
- 54.7 GRUs.
- 54.8 Deep RNN.
- 54.9 Bidirectional RNN.
- 54.10 Code example : IMDB Sentiment classification
- 54.11 **Assignment: Amazon Fine Food reviews LSTM model.**

## **Chapter 5 : Interview Questions on Deep Learning.**

- 55.1 Question & Answers

## **Module 9: Deep Learning Real-World Case Studies**

### **Chapter 1 : Case Study 11: Human Activity Recognition.**

- 56.1 Human Activity Recognition: Problem Definition.
- 56.2 Dataset Understanding
- 56.3 Data Cleansing & Preprocessing.
- 56.4 EDA: Univariate analysis
- 56.5 EDA: Data Visualization using t-SNE.
- 56.6 Classical ML models.
- 56.7 Deep Learning model.
- 56.8 **Assignment : Build deeper LSTM models and hyper-param tune them.**

### **Chapter 2 : Case Study 10: Self-Driving Car**

- 57.1 Self-driving car: Problem definition.
- 57.2 Datasets
- 57.3 Data Understanding & Analysis: Files and folders.
- 57.4 Dash-cam images and steering angles.
- 57.5 Split the dataset: Train VS Test
- 57.6 EDA: Steering angles.
- 57.7 Mean Baseline model: Simple.

- 57.8 Deep learning model: Deep Learning for regression: CNN, CNN+RNN.
- 57.9 Batch load the dataset.
- 57.10 NVIDIA's end-to-end CNN model.
- 57.11 Train the model.
- 57.12 Test and visualize the output.
- 57.13 Extensions.
- 57.14 **Assignment**

**Chapter 3 : Case Study 12: Music Generation using Deep Learning.**

- 58.1 Real world problem.
- 58.2 Music Representation.
- 58.3 Char-RNN with abc-notation: char-RNN model.
- 58.4 Char-RNN with abc-notation: Data Preparation
- 58.5 Char-RNN with abc-notation: Many to many RNN, Time Distributed Dense layer.
- 58.6 Char-RNN with abc-notation: State full RNN.
- 58.7 Char-RNN with abc-notation: Model architecture, Model training.
- 58.8 Char-RNN with abc-notation: Music Generation
- 58.9 Char-RNN with abc-notation: Generate Tabla music
- 58.10 MIDI music generation.
- 58.11 Survey Blog.

**Chapter 4 : Interview Questions**

- 59.1 Revision Questions.
- 59.2 Questions.
- 59.3 External Resources for Interview Questions.