Module 1: FUNDAMENTALS OF PROGRAMMING

a) **Python for Data**
   i) Python, Anaconda and relevant packages installations Science Introduction
   ii) Why learn Python?
   iii) Keywords and identifiers
   iv) comments, indentation and statements
   v) Variables and data types in Python
   vi) Standard Input and Output
   vii) Operators
   viii) Control flow: if else
   ix) Control flow: while loop
   x) Control flow: for loop
   xi) Control flow: break and continue
b) **Python for Data Science: Data Structures**
   i) Lists
   ii) Tuple
   iii) Sets
   iv) Dictionary
   v) String
c) **Python for Data Science: Functions**
   i) Introduction
   ii) Types of Functions
   iii) Function Arguments
   iv) Recursive Functions
   v) Lamda Function
   vi) Modules
   vii) Packages
   viii) File Handling
   ix) Exception Handling
   x) Debugging Python
d) **Python for Data Science: Numpy**
   i) Numpy Introduction
   ii) Muerical Operations on Numpy
e) **Python for Data Science: Matplotlib**
   i) Getting started with Matplotlib
f) **Python for Data Science: Pandas**
   i) Getting started with Pandas
   ii) Data Frame Basics
   iii) Key operations on Data Frames
g) **Python for Data Science: Computational Complexity**
   i) Space and Time Complexity: Searching for a Number in a List
   ii) Binary Search
   iii) Find elements common in two lists
   iv) Find elements common in two lists using Hashtable/DIct
h) **SQL**
   i) Introduction to Database
   ii) Why SQL?
   iii) Execution of an SQL Statement
   iv) IMDB Dataset

v) Installing MySQL
vi) Load IMDB Dataset
vii) Use, Describe, Show Tables
viii) Select
ix) Limit, Offset
x) Orderby
xi) Distinct
xii) Where, Comparison Operators, NULL
xiii) Logical Operators
xiv) Aggregate Functions: Count, MIN, MAX, AVG, SUM
xv) Group By
xvi) Having
xvii) Order of Keywords
xviii) Join and Natural Joins
xix) Inner, Left, Right, Outer Joins
xx) DML: Insert
xxi) DML: Update, Delete
xxii) DML: Create Table
xxiii) DDL: Alter: Add, Modify, Drop
xxiv) DDL: Drop Table, Truncate, Delete
xxv) Data Control Language: Grant, Revoke
xxvi) Learning Resources

i) Module 1: Live sessions

Module 2: DATA SCIENCE: EXPLORATORY DATA ANALYSIS AND DATA VISUALIZATION

a) Plotting for exploratory data analysis (EDA)
b) Linear Algebra
c) Probability and Statistics
d) Interview Questions on Probability and statistics
e) Dimensionality reduction and Visualization:
f) PCA (principal component analysis)
g) (t-SNE) T-distributed Stochastic Neighbourhood Embedding
h) Interview Questions on Dimensionality Reduction
i) Module 2: Live Sessions

Module 3: FOUNDATION OF NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING

a) Real world problem: Predict rating given product reviews on Amazon

i) Dataset Overview: Amazon Fine Food Reviews(EDA)
ii) Data Cleaning: Deduplication
iii) Why convert text to a vector?
iv) Bag of Words(BOW)
v) Text Pre-processing: Stemming, Stop Word Removal, Tokenization, Lemmatization
vi) Unigram, Bi-gram, n-gram
vii) tf-idf(Term Frequency-Inverse Document Frequency)
viii) Why use log in IDF
ix) Word2Vec

x) Avg-Word2Vec, tf-idf weighted Word2Vec
xi) Bag of Words (Code Sample)
xii) Text Pre-processing (Code Sample)
xiii) Avg-Word2Vec, tf-idf weighted Word2Vec(Code Sample)

b) Classification and Regression Models: K-Nearest Neighbours

i) How Classification works?
ii) Data Matrix Notation
iii) Classification vs Regression(Examples)
iv) K-Nearest Neighbours Geometric Intuition with toy example
v) Failure cases of KNN
vi) Distance measure: Euclidean(L2) , Manhattan (L1), Minkowski, Hamming
vii) Cosine distance and Cosine Similarity
viii) How to measure the effectiveness of K-NN
ix) Test/Evaluation time and space complexity
x) KNN Limitation
xi) Describe Surface for KNN as K Changes
xii) Overfitting and Underfitting
xiii) Need for Cross Validation
xiv) Visualize train, validation and test datasets
xv) How to determine overfitting and underfitting
xvi) Time based splitting
xvii) KNN for regression
xviii) Weighted KNN
xix) Voronoi Diagram
xx) Binary Search Tree
xxi) Find nearest neighbour using kd-tree
xxii) Limitation of kd-tree
xxiii) Extensions
xxiv) Hashing vs LSH
xxv) LSH for cosine similarity
xxvi) LSH for Euclidean distance
xxvii) Probabilistic class label
xxviii) Code Sample: Decision Boundary
xxix) Code Sample: Cross Validation
xxx) Revision Questions'

c) Interview Questions on K-NN(K Nearest Neighbour)

i) Question and Answers

d) Classification algorithms in various situations

i) Introduction
ii) Imbalanced vs Balanced dataset
iii) Multiclass Classification
iv) K-NN, given a distance or similarity matrix
v) Train and Test set Difference
vi) Impact of Outliers
vii) Local outliers factor (Simple Solution: Mean Distance to Knn)
viii) K Distance
ix) Reachability Distance (A, B)
x) Local reachability –density(A)
xi) Local Outlier factor(A)
xii) Impact of scale and column standardization
xiii) Interpretability

xiv) Feature importance and Forward Feature Selection
xv) Handling Categorical and Numerical Features
xvi) Handling missing values by imputation
xvii) Curse of dimensionality
xviii) Bias-Variance Tradeoff
xix) Intuitive understanding of bias-variance
xx) Revision Questions
xxi) Best and Worst case algorithm

e) Performance measurement of models

i) Accuracy
ii) Confusion matrix, TPR, FPR, FNR, TNR
iii) Precision, Recall and F1 Score
iv) Receiver Operating Characteristics Curve (ROC) curve and AUC
v) Log-Loss
vi) R-Squared/ Coefficient of Determination
vii) Median Absolute Deviation (MAD)
viii) Distribution of Errors
ix) Revision Questions

f) Interview Questions on Performance Measurement Models

i) Questions and Answer

g) Naive Bayes

i) Conditional Probability
ii) Independent vs Mutually Exclusive Events
iii) Bayes Theorem with Examples
iv) Exercise Problems on Bayes Theorem
v) Naïve Bayes Algorithm
vi) Toy example: Train and Test Stages
vii) Naïve Bayes on text data
viii) Laplace/ Additive Smoothing
ix) Log-Probabilities for Numerical Studies
x) Bias and Variance Tradeoff
xi) Feature importance and interpretability
xii) Imbalanced Data
xiii) Outliers
xiv) Missing Values
xv) Handling Numerical Feature (Guassian NB)
xvi) Multiclass Classification
xvii) Similarity or Distance Matrix
xviii) Large Dimensionality
xix) Best and Worst Case
xx) Code Example
xxi) Revision Question

h) Logistic Regression

i) Geometric Intuition of Logistic Regression
ii) Sigmoid Function: Squashing
iii) Mathematical Formulation of Objective Function
iv) Weight Vector
v) L2 Regularization: Overfitting and Underfitting
vi) L1 regularization and sparsity
vii) Probabilistic Interpretation: Guassian Naïve Bayes
viii) Loss minimization Interpretation

ix) Hyper parameter Search: Grid search and random search
x) Column Standardization
xi) Feature importance and Model Interpretability
xii) Collinearity of Features
xiii) Train and Run Time space and time complexity
xiv) Real world cases
xv) Non Linear separable data and feature engineering
xvi) Code sample: Logistic Regression, GridSearchCV, RandomSearchCV
xvii) Extension to Logistic Regression: Generalized Linear Models(GLM)

i) Linear Regression
i) Geometric Intuition of Linear Regression
ii) Mathematical Formulation
iii) Real world cases
iv) Code sample for Linear Regression

j) Solving Optimization Problems
i) Differentiation
ii) Online Differentiation Tools
iii) Maxima and Minima
iv) Vector Calculus: Grad
v) Gradient Descent: Geometric Intuition
vi) Learning Rate
vii) Gradient descent for linear regression
viii) SGD algorithm
ix) Constrained Optimization & PCA
x) Logistic regression formulation revisited
xi) Why L1 regularization created sparsity
xii) Revision questions

k) Interview Questions on Logistic Regression and Linear Regression
i) Question and Answers

Module 4: MACHINE LEARNING-II (SUPERVISED MACHINE LEARNING)

a) Support Vector Machines (SVM)
i) Geometric Intuition
ii) Mathematical Derivation
iii) Why we take values+1 and -1 for SVM
iv) Loss function (Hinge Loss) based interpretation
v) Dual form of SVM formulation
vi) Kernel trick
vii) Polynomial Kernel
viii) RBF-Kernel
ix) Domain specific kernels
x) Train and run time complexities
xi) Nu-SVM: control errors and support vector
xii) SVM Regression
xiii) Cases
xiv) Code Samples
xv) Revision Questions

b) Interview Questions on Support Vector Machine
i) Question and Answer

c) Decision Trees

i) Geometric Intuition of Decision Tree: Axis Parallel Hyperplanes
ii) Sample Decision Tree
iii) Building a Decision Tree: Entropy
iv) KL Divergence
v) Building a Decision Tree: Information Gain
vi) Building a Decision Tree: Constructing a DT
vii) Building a Decision Tree: Splitting Numerical Features
viii) Feature Standardization
ix) Building a Decision Tree: Categorical feature with many possible values
x) Overfitting and underfitting
xi) Train and Run time complexity
xii) Regression using Decision Tree
xiii) Cases
xiv) Code Samples
xv) Revision Questions

d) Interview Questions on decision Trees

i) Question and Answers

e) Ensemble Models

i) What are ensembles
ii) Bootstrapped Aggregation (Bagging) Intuition
iii) Random forest and their construction
iv) Bias-variance trade-off
v) Bagging train and run time complexity
vi) Bagging: Code samples
vii) Extremely randomized trees
viii) Random forest: cases
ix) Boosting Intuition
x) Residual, Loss functions and gradients
xi) Gradient Boosting
xii) Regularization by shrinking
xiii) Train and run time complexity
xiv) XGBoost: Boosting + Randomization
xv) AdaBoost: Geometric Intuition
xvi) Stacking models
xvii) Kaggle competitions vs Real world
xviii) Revision Questions

f) Module 4: Live Sessions

Module 5: FEATURE ENGINEERING, PRODUCTIONIZATION AND DEPLOYMENT OF ML MODELS

a) Featurization and Feature engineering.

i) Introduction
ii) Moving window for time series data
iii) Fourier decomposition
iv) Deep learning features: LSTM
v) Image histogram
vi) Keypoints : SIFT
vii) Deep learning features: CNN
viii) Relational data
ix) Graph data
x) Indicator Variables
xi) Feature Binning
xii) Interaction variables
xiii) Mathematical Transforms
xiv) Model specific featurization
xv) Feature Orthogonality
xvi) Domain specific featurization
xvii) Feature slicing
xviii) Kaggle winner solution

b) Miscellaneous Topics

i) Calibration of Models: Need for calibration
ii) Calibration plots
iii) Platt's calibration/scaling
iv) Isotonic Regression
v) Code Samples
vi) Modeling in the presence of outliers: RANSAC
vii) Retraining models periodically
viii) A/B Testing
ix) Data Science Lifecycle
x) Productionization and Deployment of ML models
xi) Productionization and Deployment + Spark
xii) Hands on Live session: Deploy an ML models using Flask on AWS
xiii) Building web apps for ML/AL using StreamLit-I
xiv) Building web apps for ML/AL using StreamLit-II
xv) ML model productionization using Heroku
xvi) VC Dimension

c) Module 5: Live Sessions

Module 6: MACHINE LEARNING REAL-WORLD CASE STUDIES

a) Case Study 1: Quora question Pair Similarity Problem

i) How to optimally learn from case studies in the course?
ii) Business/Real world problem: Problem Definition
iii) Business Objectives and Constraints
iv) Mapping to ML Problem: Data Overview
v) Mapping to ML Problem: ML Problem and performance metric
vi) Mapping to ML Problem: Train-Test Split

vii) EDA: Basic Statistics
viii) EDA: Basic Feature Extraction
ix) EDA: Text Preprocessing
x) EDA: Advanced Feature Extraction
xi) EDA: Feature Analysis
xii) EDA: Data Visulaization: t-SNE
xiii) EDA: TF-IDF weighted Word2Vec Featurization
xiv) ML Models: Loading Data
xv) ML Models: Random Model
xvi) ML Models: Logistic regression and Linear SVM
xvii) ML Models: XGBoost

b) Case Study 2: Personalized Cancer Diagnosis

i) Business/Real world problem: Overview
ii) Business Objectives and Constraints
iii) ML Problem Formulation: Data
iv) ML Problem Formulation: Mapping real world to ML Problem
v) ML Problem Formulation: Train, CV and Test data construction
vi) EDA: Read Data & pre-processing
vii) EDA: Distribution of class labels
viii) EDA: Random Model
ix) Univariate Analysis: Gene Feature
x) Univariate Analysis: Variation Feature
xi) Univariate Analysis: Text Feature
xii) Machine Learning: Data Preparation
xiii) Baseline Model: Naïve Bayes
xiv) K-Nearest Neighbor Classification
xv) Logistic Regression and Class Balancing
xvi) Linear SVM
xvii) Random Forest with one-hot encoded features
xviii) Random forest with response coded feature
xix) Stacking classifier
xx) Majority Voting Classifier

c) Case Study 3: Facebook Friend Recommendation using Graph Mining

i) Problem Definition
ii) Overview of graphs: Node/Vertex/, Edge/Link, Directed Edge, Path
iii) Data Format and Limitations
iv) Mapping to a supervised classification problem
v) Business constraints and Metrics
vi) EDA: Basic stats
vii) EDA: Follower and Following Stats
viii) EDA: Binary Classification Task
ix) EDA: Trains-Test Split
x) Feature Engineering on Graphs: Jaccard and Cosine Similarities
xi) Page Rank
xii) Shortest Path
xiii) Connected Components
xiv) Adar Index
xv) Kartz Centrality
xvi) HITS Score
xvii) SVD
xviii) Weight Features

xix)Modeling

d) Case study 4:Taxi demand prediction in New York City

i) Business/Real World Problem Overview
ii) Objective and Constraints
iii) Mapping to ML problem: Data
iv) Mapping to ML problem: Dask dataframes
v) Mapping to ML problem: Fields/Features
vi) Mapping to ML problem: Time Series Forecasting /Regression
vii) Mapping to ML problem: Performance Metrics
viii) Data Cleaning: Latitude and Longitude Data
ix) Data Cleaning: Trip Duration
x) Data Cleaning: Speed
xi) Data Cleaning: Distance
xii) Data Cleaning: Fare
xiii) Data Cleaning: Remove all outliers/erroneous points
xiv)Data Preparation: Clustering/Segmentation
xv) Data Preparation: Time Binning
xvi)Data Preparation: Smoothing Time Series Data
xvii) Data Preparation: Time Series and Fourier Transforms
xviii) Ratios and Previous Time bin values
xix)Simple Moving Average
xx) Weighted Moving Average
xxi)Exponential Weighted Moving Average
xxii) Results
xxiii) Regression Models: Training-Test Split and Features
xxiv) Linear Regression
xxv) Radom Forest Regression
xxvi) XGBoost Regression
xxvii) Model Comparison

e) Case study 5: Stackoverflow tag predictor

i) Business/Real World Problem Overview
ii) Objective and Constraints
iii) Mapping to ML problem: Data Overview
iv) Mapping to ML problem: ML Problem Formulation
v) Mapping to ML problem: Performance Metrics
vi) Hamming Loss
vii) EDA: Data Loading
viii) EDA: Analysis of Tags
ix) EDA: Data Pre-processing
x) Data Modelling: Multi Label Classification
xi) Data Preparation
xii) Train-Test Split
xiii) Featurization
xiv)Logistic Regression: One VS Rest
xv) Sampling Data and tags + Weighted Models
xvi)Logistic Regression Revisited
xvii) Why not use Advanced Techniques

f) Case Study 6: Microsoft Malware Detection

i) Business/Real World Problem Overview
ii) Objective and Constraints
iii) Mapping to ML problem: Data Overview

iv) Mapping to ML problem: ML Problem Formulation
v) Mapping to ML problem: Train and Test Splitting
vi) EDA: Class Distribution
vii) EDA: Feature Extraction from byte files
viii) EDA: Multivariate analysis of features from byte files
ix) EDA: Train-Test Class Distribution
x) ML models: using byte files only: Random Model
xi) K-NN
xii) Logistic Regression
xiii) Random Forest and XGBoost
xiv) ASM Files: Feature Extraction and Multiprocessing
xv) File Size Feature
xvi) Univariate Analysis
xvii) T-SNE Analysis
xviii) ML Models on ASM File Features
xix) Models on all feature: t-SNE
xx) Models on all Features: Random Forest and XGBoost

g) Module 6: Live Sessions

Module 7: DATA MINING (UNSUPERVISED LEARNING) AND RECOMMENDATION SYSTEMS + REAL WORLD CASE STUDIES

a) Unsupervised learning/Clustering
   i) What is clustering?
   ii) Unsupervised Learning
   iii) Applications
   iv) Metrics of Clustering
   v) K-Means: Geometric Intuition, Centroids
   vi) K-Means: Mathematical Formulation: Objective Function
   vii) K-Means Algorithm
   viii) How to initialize: K Means++
   ix) Failure Cases/Limitations
   x) K-Medoids
   xi) Determining the right K

b) Hierarchical clustering Technique
   i) Agglomerative & Divisive, Dendrograms
   ii) Agglomerative Clustering
   iii) Proximity Methods: Advantages and Limitations
   iv) Time and Space Complexity
   v) Limitation of Hierarchical Clustering
   vi) Code Samples

c) DBSCAN (Density based clustering) Technique
   i) Density Based Clustering
   ii) MinPts and EPS: Density
   iii) Core, Border, and Noise Points
   iv) Density Edge and Density Connected Points
   v) DBSCAN Algorithm
   vi) Hyper Parameters: MinPts and EPsA
   vii) Advantages and Limitation of DBSCAN
   viii) Time and Space Complexity
   ix) Code Samples
   x) Revision Questions

d) Recommender Systems and Matrix Factorization
   i) Problem Formulation: Movies Reviews
   ii) Content based vs Collaborative Filtering
   iii) Similarity based Algorithms
   iv) Matrix Factorization : PCA, SVD
   v) Matrix Faxtorization: NMF
   vi) Matrix Factorization and Collaborative Filtering
   vii) Matrix Factorization and Feature Engineering
   viii) Clustering as MF
   ix) Hyper parameter Tuning
   x) Matrix Factorization for recommendation system: Netflix Prize Solution
   xi) Cold Start Problem
   xii) World Vector as MF
   xiii) Eigen Faces

e) Interview Questions on Recommender Systems and Matrix Factorization.
   i) QA

f) Case Study 8: Amazon fashion discovery engine (Content Based recommendation)

i) Problem Statement: Recommend similar apparel products in e-commerce using product descriptions and Images
ii) Plan of action
iii) Amazon product advertising API
iv) Data Folders and Paths
v) Overview of the data and Terminology
vi) Data Cleaning and Understanding: Missing data in various features
vii) Understanding duplicate rows
viii) Remove duplicates: Part 1
ix) Remove duplicates: Part 2
x) Text Pre-Processing: Tokenization and Stop-word removal
xi) Stemming
xii) Text based product similarity: Converting text to an n-D vector: Bag of Words
xiii) Code for Bag of Words based product similarity
xiv) TF-IDF: Feature Text based on word importance
xv) Code for TF-IDF based product similarity
xvi) Code for IDF based product similarity
xvii) Text Semantics based product similarity: Word2Vec(Feature Engineering based on semantic similarity)
xviii) Code for Average Word2Vec product similarity
xix) Code for IDF weighted Word2Vec Product Similarity
xx) Weighted similarity using brand and color
xxi) Code for weighted similarity
xxii) Building a real world solution
xxiii) Deep learning based visual product similarity: ConvNets: How to fearturize an images: edges, shapes, parts
xxiv) Using Keras + Tensorflow to extract feature
xxv) Visual similarity based product similarity
xxvi) Measuring goodness of our solution: A/B Testing
xxvii) Assignment 24-Apparel Recommendation

g) Case Study 9: Netflix Movie Recommendation System (Collaborative based recommendation)

i) Business/Real world Problem: Problem Definition
ii) Objective and Constraints
iii) Mapping to an ML problem: Data Overview
iv) Mapping to an ML problem: ML problem formulation
v) EDA: Data Processing
vi) EDA: Temporal Train-Test Split
vii) EDA: Preliminary Data Analysis
viii) EDA: Sparse Matrix Representation
ix) EDA: Average Rating for various slices
x) EDA: Cold start problem
xi) Computing Similarity Matrices: User-User Similarity Matrix
xii) Computing Similarity Matrices: Movie-Movie Similarity Matrix
xiii) Computing Similarity Matrices: Does Movie-Movie similarity works?
xiv) ML Models: Surprise Library
xv) Overview of Modelling Strategy
xvi) Data Sampling
xvii) Google drive with intermediate files
xviii) Featurizations for regression
xix) Data transformation for surprise

xx) Xgboost with 13 features
xxi) Surprise Baseline Model
xxii) Xgboost +13 features+ Surprise Baseline Model
xxiii) Surprise KNN Predictors
xxiv) Matrix factorixzation models using Surprise
xxv) SVD++ with implicit feedback
xxvi) Final models with all features and predictors
xxvii) Comparison between various models
xxviii) Assignment 18- Netflix prize

h) High Level + End-End Design of a Music Recommendation system
   i) Live Sessions
i) Module 7: Live Sessions

Module 8: NEURAL NETWORK, COMPUTER VISION AND DEEP LEARNING

a) Deep Learning: Neural Networks.

i) History of Neural Networks and Deep Learning
ii) How Biological Neurons work?
iii) Growth of Biological Neural Network
iv) Diagrammatic Representation: Logistic Regression and Perceptron
v) Multi-Layered Perceptron
vi) Notation
vii) Training a single neuron model
viii) Training an MLP: Chain Rule
ix) Training an MLP: Memorization
x) Backpropagation
xi) Activation Function
xii) Vanishing Gradient Problem
xiii) Bias variance trade-off
xiv) Decision Surface: Playground
xv) Interview Questions

b) Deep Learning: Deep Multi-layer perceptron

i) Deep Multilayers Perceptron: 19080 to 2010
ii) Dropouts layers and Regularization
iii) Rectified Linear Units(RELU)
iv) Weight Initialization
v) Batch Normalization
vi) Optimizers: Hill-Descent analogy in 2D
vii) Optimizers: Hill-Descent analogy in 3D and contours
viii) SGD Recap
ix) Batch SGD with momentum
x) Nesterov Accelerated Gradient (NAG)
xi) Optimizers: AdaGrad
xii) Optimizers: Adadelta and RMSProp
xiii) Adam
xiv) Which algorithm to choose when?
xv) Gradient checking and clipping
xvi) Softmax and cross –entropy for multi-class classification
xvii) How to train a Deep MPL
xviii) Auto Encoders

xix)Word2Vec: CBOW
xx) Word2Vec: Skip-gram
xxi)Word2Vec: Algorithm Optimizations

c) Deep Learning: Tensorflow and Keras.
i) Tensorflow and Keras Overview
ii) GPU vs CPU for Deep Learning
iii) Google Colaboratory
iv) Install Tensor Flow
v) Online documentation and tutorials
vi) Softmax classifier on MNIST dataset
vii) MLP: Initialization
viii) Model 1: Sigmoid Activation
ix) Model 2: ReLU Activation
x) Model 3: Batch Normalization
xi) Model 4: Dropout
xii) MNIST Classification in Keras
xiii) Hyperparameter tuning in Keras

d) Deep Learning: Convolutional Neural Nets.
i) Biological Inspiration: Visual Cortex
ii) Convolution: Edge Detection on Images
iii) Convolution: Padding and Strides
iv) Convolution over RGB images
v) Convolution layer
vi) Max Pooling
vii) CNN Training Optimization
viii) Receptive Fields and Effective Receptive Fields
ix) Example CNN: LeNet[1998]
x) ImageNet Dataset
xi) Data Augmentation
xii) AlexNet
xiii) VGGNet
xiv)Residual Network
xv) Inception Network
xvi)What is Transfer Learning
xvii) Code Example: Cats vs Dogs
xviii) Code Example: MNIST Dataset
xix)[Interview Questions]: How to build a face recognition system?

e) Deep Learning: Long Short-term memory (LSTMs)
i) Why RNNs
ii) Recurrent Neural Network
iii) Training RNNs: Backprop
iv) Types of RNN
v) Need for LSTM/GRU
vi) LSTM
vii) GRUs
viii) Deep RNN
ix) Bidirectional RNN
x) Code example: IMDB Sentiment Classification

f) Deep Learning: Generative Adversarial Networks (GANs)
i) Live session

g) Encoder-Decoder Models

i) LiveSession

h) Attention Models in Deep Learning

i) LiveSession

i) Deep Learning: Transformers and BERT

i) LiveSession

j) Deep Learning: Image Segmentation

i) LiveSession

k) Deep Learning: Object Detection

i) Object Detection

ii) Object Detection YOLO V3

l) Deep Learning: GPT-1, 2 and GPT-3 Models

i) LiveSession

m) Interview Questions on Deep Learning

n) Module 8: Live Sessions

Module 9: DEEP LEARNING REAL-WORLD CASE STUDIES

a) Case Study 11: Human Activity Recognition

i) Human Activity Recognition Problem Definition

ii) Dataset Understanding

iii) Data cleaning and Pre-processing

iv) EDA: Univariate Analysis

v) EDA: Data Visualization using t-SNE

vi) Classical ML Models

vii) Deep Learning Models

b) Case Study 10: Self Driving Car

i) Problem Definition

ii) Datasets

iii) Data Understanding and Analysis: Files and Folders

iv) Dash cam images and steering angles

v) Split the dataset: Train vs Test

vi) EDA: Steering angles

vii) Mean baseline model: simple

viii) Deep Learning Model: Deep Learning for regression: CNN, CNN+RNN

ix) Batch load the dataset

x) NVIDIA's end to end CNN model

xi) Train the model

xii) Test and Visualize the output

xiii) Extensions

c) Case Study 12: Music Generation using Deep-Learning

i) Real world Problem

ii) Music Representation

iii) Char RNN with abc-notation: Char –RNN Model

iv) Char RNN with abc-notation: Data preparation

v) Char RNN with abc-notation: Many to Many RNN, Time Distributed –Dense Layer

vi) Char RNN with abc-notation: State full RNN

vii) Char RNN with abc-notation: Model Architecture, Model Training

viii) Char RNN with abc-notation: Music Generation

ix) Char RNN with abc-notation: Generate Tabla Music

x) MIDI music generation

xi) Survey Blog

d) Case Study 13: Semantic Search Engine for Q&A [Design + Code]
    i) High level design of the solution
    ii) Sentence vector and Docker Containerisation
    iii) Indexing using ElsaticSearch
    iv) Deployment using Flask API, Docker and Elastic Search
e) Case Study 14: Building a Smart Gym Assistant from scratch
    i) Live Session
f) Interview Questions
g) Module 9: Live Sessions

Module 10: MISC. TOPICS: SPARK FOR ML, BIG DATA, REINFORCEMENT LEARNING, ML DESIGN

a) Machine Learning High-Level Design
    i) Live Session
b) Sample Interview and Conceptual Questions [AUDIO]
    i) Live Session
c) Reinforcement Learning
    i) Applications of RL
    ii) Problem Formulation and Terminology
    iii) Bellman Equation
    iv) Dynamic Programming and RL
    v) Markov Decision Process(MDP)
    vi) Q-Function
    vii) Temporal Difference
    viii) Deep Q Learning-Intuition
    ix) QA
d) Module 10: Live Sessions
    i) An Overview of AI algorithms
    ii) Introduction to Big Data for ML and AI
    iii) Big Data and Cloud Storage for ML/AI applications
    iv) Spark for Data Science and ML [Architecture and Programming Model]
    v) Spark for Data Science and ML [SetUp+Codewalkthrough]-II
    vi) Spark for Data Science and ML [ML Lib and ML Pipelines]-II
    vii) How to build chatbots
    viii) Design and build a chatbot from scratch
    ix) How to build IOT + AI systems
    x) LIVE: How to crack ML Competitions
    xi) Live: Effective communication for DS and MLEs
    xii) Sample Interview and Comceptual Questions