APPLIED ROOTS

Diploma in AI and ML

6309798882/
6309798883

pgduoh@appliedroots.com

www.appliedroots.com

Semester-1

Essentials of AI (7 credits)

Python

Python for Data Science Introduction

- Python, Anaconda and relevant packages installations
- Why learn Python?
- Keywords and identifiers
- comments, indentation and statements
- Variables and data types in Python
- Standard Input and Output
- Operators
- Control flow: if else
- Control flow: while loop
- Control flow: for loop
- Control flow: break and continue

Python for Data Science: Data Structures

2.1 - Lists
2.2 - Tuples part 1
2.3 - Tuples part-2
2.4 - Sets
1.5 - Dictionary
1.6 - Strings

SQL

SQL

1.1 - Introduction to Databases

1.2 - Why SQL?

1.3 - Execution of an SQL statement.

1.4 - IMDB dataset

1.5 - Installing MySQL

1.6 - Load IMDB data.

1.7 - USE, DESCRIBE, SHOW TABLES

1.8 - SELECT

1.9 - LIMIT, OFFSET

1.10 - ORDER BY

1.11 - DISTINCT

1.12 - WHERE, Comparison operators, NULL

1.13 - Logical Operators

1.14 - Aggregate Functions: COUNT, MIN, MAX, AVG, SUM

1.15 - GROUP BY

1.16 - HAVING

1.17 - Order of keywords.

1.18 - Join and Natural Join

1.19 - Inner, Left, Right and Outer joins.

1.20 - Sub Queries/Nested Queries/Inner Queries

1.21 - DML:INSERT

1.22 - DML:UPDATE , DELETE

1.23 - DDL:CREATE TABLE

1.24 - DDL:ALTER: ADD, MODIFY, DROP

1.25 - DDL:DROP TABLE, TRUNCATE, DELETE

1.26 - Data Control Language: GRANT, REVOKE

1.27 - Learning resources

Linear Algebra

✥ Linear Algebra

1.1 - Why learn it ?

1.2 - Introduction to Vectors(2-D, 3-D, n-D) , Row Vector and Column Vector

1.3 - Dot Product and Angle between 2 Vectors

1.4 - Projection and Unit Vector

1.5 - Equation of a line (2-D), Plane(3-D) and Hyperplane (n-D), Plane Passing through origin, Normal to a Plane

1.6 - Distance of a point from a Plane/Hyperplane, Half-Spaces

1.7 - Equation of a Circle (2-D), Sphere (3-D) and Hypersphere (n-D)

1.8 - Equation of an Ellipse (2-D), Ellipsoid (3-D) and Hyperellipsoid (n-D)

1.9 - Square ,Rectangle

1.10 - HyperCube,Hyper Cuboid

Basics of Probability

✥ Probability

1.1 - Why study Probability?

1.2 - Introduction

1.3 - Axioms of Probability, Properties and Examples

1.4 - Conditional Probability & Examples

1.5 - Multiplication theorem

1.6 - Independent events

1.7 - Law of total Probability

1.8 - Bayes Theorem

1.9 - Random variables: an introduction

1.10 - PMF, CDF and PDF of random variables

1.11 - Expectation

1.12 - Probability Distributions: Bernoulli and Binomial

Subject 2

Data Analysis and Visualization (7 credits)

Plotting

Plotting for exploratory data analysis (EDA)

1.1 - Introduction to IRIS dataset and 2D scatter plot

1.2 - 3D scatter plot

1.3 - Pair plots

1.4 - Limitations of Pair Plots

1.5 - Histogram and Introduction to PDF(Probability Density Function)

1.6 - Univariate Analysis using PDF

1.7 - CDF(Cumulative Distribution Function)

1.8 - Mean, Variance and Standard Deviation

1.9 - Median

1.10 - Percentiles and Quantiles

1.11 - IQR(Inter Quartile Range) and MAD(Median Absolute Deviation)

1.12 Box-plot with Whiskers

1.13 Violin Plots

1.14 Summarizing Plots, Univariate, Bivariate and Multivariate analysis

1.15 Multivariate Probability Density, Contour Plot

1.16 Assignment-1: Data Visualization with Haberman Dataset

Statistics for Data Analysis

2.1 - Population and Sample

2.2 - Gaussian/Normal Distribution and its PDF(Probability Density Function)

2.3 - CDF(Cumulative Distribution function) of Gaussian/Normal distribution

Dimensionality reduction

✣ Dimensionality reduction and Visualization:

1.1 - What is Dimensionality reduction?

1.2 - Row Vector and Column Vector

1.3 - How to represent a data set?

1.4 - How to represent a dataset as a Matrix.

1.5 - Data Preprocessing: Feature Normalisation

1.6 - Mean of a data matrix

1.7 - Data Preprocessing: Column Standardization

1.8 - Co-variance of a Data Matrix

1.9 - MNIST dataset (784 dimensional)

1.10 - Code to Load MNIST Data Set

✣ PCA(principal component analysis)

2.1 - Why learn PCA?

2.2 - Geometric intuition of PCA

2.3 - Mathematical objective function of PCA

2.4 - Alternative formulation of PCA: Distance minimization

2.5 - Eigen values and Eigen vectors (PCA): Dimensionality reduction

2.6 - PCA for Dimensionality Reduction and Visualization

2.7 - Visualize MNIST dataset

2.8 - Limitations of PCA

2.9 - PCA Code example

2.10 - PCA for dimensionality reduction (not-visualization)

Visualising high dimensional data

✣ (t-SNE)T-distributed Stochastic Neighbourhood Embedding

3.1 - What is t-SNE?

3.2 - Neighborhood of a point, Embedding

3.3 - Geometric intuition of t-SNE

3.4 - Crowding Problem

3.5 - How to apply t-SNE and interpret its output

3.6 - t-SNE on MNIST

3.7 - Code example of t-SNE

Subject 3

Machine Learning (7 credits)

Calculus and Numerical Optimization

Solving Optimization Problems

1.1 - Differentiation

1.2 - Online differentiation tools

1.3 - Maxima and Minima

1.4 - Vector calculus: Grad

1.5 - Gradient descent: geometric intuition

1.6 - Learning rate

1.7 - Gradient descent for linear regression

1.8 - SGD algorithm

1.9 - Constrained Optimization & PCA

1.10 - Logistic regression formulation revisited

1.11 Why L1 regularization creates sparsity?

Classification

Classification and Regression Models: K-Nearest Neighbors

1.4 - How "Classification" works?

1.5 - Data matrix notation.

1.6 - Classification vs Regression (examples)

1.7 - K-Nearest Neighbors Geometric intuition with a toy example.

1.8 - Failure cases of K-NN

1.9 - Distance measures: Euclidean(L2) , Manhattan(L1), Minkowski, 1.10 - Hamming

1.11 - Cosine Distance & Cosine Similarity

1.12 - How to measure the effectiveness of k-NN?

1.13 - Test/Evaluation time and space complexity.

1.14 - k-NN Limitations.

1.15 - Decision surface for K-NN as K changes.

1.16 - Overfitting and Underfitting.

1.17 - Need for Cross validation.

1.18 - K-fold cross validation.

1.19 - Visualizing train, validation and test datasets

1.20 - How to determine overfitting and underfitting?

1.21 - Time based splitting

1.22 - k-NN for regression.

1.23 - Weighted k-NN

1.24 - Voronoi diagram.

1.25 - Binary search tree

1.26 - How to build a kd-tree.

1.27 - Find nearest neighbors using kd-tree

1.28 - Limitations of kd-tree

1.29 - Extensions.

1.30 - Hashing vs LSH.

1.31 - LSH for cosine similarity

1.32 - LSH for euclidean distance.

1.33 - Probabilistic class label

1.34 - Code Sample: Decision boundary.

1.35 - Code Samples: Cross-Validation

❖ Classification algorithms in various situations:

2.1 - Introduction

2.2 - Imbalanced vs balanced dataset.

2.3 - Multi-class classification.

2.4 - k-NN, given a distance or similarity matrix

2.5 - Train and test set differences.

2.6 - Impact of Outliers

2.7 - Local Outlier Factor(Simple solution: mean distance to k-NN).

2.8 - k-distance (A), N(A)

2.9 - reachability-distance(A, B)

2.10 - Local-reachability-density(A)

2.11 - Local Outlier Factor(A)

2.12 - Impact of Scale & Column standardization.

2.13 - Interpretability

2.14 - Feature importance & Forward Feature Selection

2.15 - Handling categorical and numerical features.

2.16 - Handling missing values by imputation.

2.17 - Curse of dimensionality.

2.18 - Bias-Variance tradeoff.

2.19 - Intuitive understanding of bias-variance.

❖ Performance measurement of models:

3.1 - Accuracy

3.2 - Confusion matrix, TPR, FPR, FNR, TNR

3.3 - Precision & recall, F1-score.

3.4 - Receiver Operating Characteristic Curve (ROC) curve and AUC.

3.5 Log-loss.

3.6 R-Squared/ Coefficient of determination.

3.7 Median absolute deviation (MAD)

3.8 Distribution of errors.

❖ Naive Bayes

4.1 - Conditional probability.

4.2 - Independent vs Mutually exclusive events.

4.3 - Bayes Theorem with examples.

4.4 - Exercise problems on Bayes Theorem

4.5 - Naive Bayes algorithm.

4.6 - Toy example: Train and test stages.

4.7 - Naive Bayes on Text data.

4.8 - Laplace/Additive Smoothing.

4.9 - Log-probabilities for numerical stability.

4.10 - Bias and Variance tradeoff.

4.11 - Feature importance and interpretability.

4.12 - Imbalanced data

4.13 - Outliers.

4.14 - Missing values.

4.15 - Handling Numerical features (Gaussian NB)

4.16 - Multiclass classification.

4.17 - Similarity or Distance matrix.

4.18 - Large dimensionality.

4.19 - Best and worst cases.

4.20 - Code example

❖ Logistic Regression

5.1 - Geometric intuition of logistic regression

5.2 - Sigmoid function: Squashing

5.3 - Mathematical formulation of objective function.

5.4 - Weight Vector.

5.5 L2 Regularization: Overfitting and Underfitting.

5.6 L1 regularization and sparsity.

5.7 Probabilistic Interpretation: Gaussian Naive Bayes

5.8 Loss minimization interpretation

5.9 Hyperparameter search: Grid Search and Random Search

5.10 Column Standardization.

5.11 Feature importance and model interpretability.

5.12 Collinearity of features.

5.13 Train & Run time space and time complexity.

5.14 Real world cases.

5.15 Non-linearly separable data & feature engineering.

5.16 - Code sample: Logistic regression, GridSearchCV, RandomSearchCV

5.17 - Extensions to Logistic Regression: Generalized linear models (GLM)

❖ Support Vector Machines (SVM)

6.1 - Geometric intuition.

6.2 - Mathematical derivation.

6.3 - why we take values +1 and -1 for support vector planes

6.4 - Loss function(Hinge Loss) based interpretation.

6.5 Dual form of SVM formulation.

6.6 Kernel trick.

6.7 Polynomial kernel.

6.8 RBF-Kernel.

6.9 Domain specific Kernels.

6.10 Train and run time complexities.

6.11 nu-SVM: control errors and support vectors.

6.12 SVM Regression.

6.13 Cases.

6.14 Code Sample.

❖ Decision Trees

7.1 - Geometric Intuition of decision tree: Axis parallel hyperplanes.

7.2 - Sample Decision tree.

7.3 - Building a decision Tree: Entropy(Intuition behind entropy)

7.4 - Building a decision Tree: Information Gain

7.5 Building a decision Tree: Gini Impurity.

7.6 Building a decision Tree: Constructing a DT.

7.7 Building a decision Tree: Splitting numerical features.

7.8 Feature standardization.

7.9 Categorical features with many possible values.

7.10 Overfitting and Underfitting.

7.11 Train and Run time complexity.

7.12 Regression using Decision Trees.

7.13 - Cases

7.14 - Code Samples

❖ Ensemble Models

8.1 - What are ensembles?

8.2 - Bootstrapped Aggregation (Bagging) Intuition.

8.3 - Random Forest and their construction.

8.4 - Bias-Variance tradeoff.

8.5 Train and Run-time Complexity.

8.6 Bagging: code Sample.

8.7 Extremely randomized trees.

8.8 Random Forest: Cases.

8.9 Boosting Intuition

8.10 Residuals, Loss functions, and gradients.

8.11 Gradient Boosting

8.12 Regularization by Shrinkage.

8.13 Train and Run time complexity.

8.14 XGBoost: Boosting + Randomization

8.15 AdaBoost: geometric intuition.

8.16 Stacking models.

8.17 Cascading classifiers.

8.18 Kaggle competitions vs Real world

Regression and Clustering algorithms

❖ Linear Regression

1.1 - Geometric intuition of Linear Regression.

1.2 - Mathematical formulation.

1.3 - Real world Cases.

1.4 - Code sample for Linear Regression

❖ Unsupervised learning/Clustering

2.1 - What is Clustering?

2.2 - Unsupervised learning

2.3 - Applications.

2.4 - Metrics for Clustering.

2.5 - K-Means: Geometric intuition, Centroids.

2.6 - K-Means: Mathematical formulation: Objective function

2.7 - K-Means Algorithm.

2.8 - How to initialize: K-Means++

2.9 - Failure cases/Limitations.

2.10 - K-Medoids

2.11 - Determining the right K.

2.12 - Code Samples.

2.13 - Time and Space complexity.

❖ Hierarchical clustering Technique

3.1 - Agglomerative & Divisive, Dendrograms

3.2 - Agglomerative Clustering.

3.3 - Proximity methods: Advantages and Limitations.

3.4 - Time and Space Complexity.

3.5 - Limitations of Hierarchical Clustering.

3.6 - Code sample.

❖ DBSCAN (Density based clustering)

4.1 - Density based clustering

4.2 - MinPts and Eps: Density

4.3 - Core, Border and Noise points.

4.4 - Density edge and Density connected points.

4.5 - DBSCAN Algorithm.

4.6 - Hyper Parameters: MinPts and Eps.

4.7 - Advantages and Limitations of DBSCAN.

3.1 - Time and Space Complexity

3.2 - Code samples

Projects which we are covering

Diagnosis using medical records

Question Similarity

Stack overflow Tag Predictor

facebook friend recommendation using graph mining

Demand Prediction

Malware Detection

Semester - 2

Subject 1

Advanced Machine Learning (7 credits)

Recommender Systems

Recommender Systems and Matrix Factorization

- 1.1 - Problem formulation: Movie reviews.
- 1.2 - Content based vs Collaborative Filtering.
- 1.3 - Similarity based Algorithms.
- 1.4 - Matrix Factorization: PCA, SVD.
- 1.5 - Matrix Factorization: NMF.
- 1.6 - Matrix Factorization for Collaborative filtering
- 1.7 - Matrix Factorization for feature engineering.
- 1.8 - Clustering as MF.
- 1.9 - Hyperparameter tuning.
- 1.10 - Matrix Factorization for recommender systems: Netflix Prize Solution.
- 1.11 - Cold Start problem.
- 1.12 - Word Vectors as MF.
- 1.13 - Eigen-Faces.
- 1.14 - Code example.

Neural Networks

Deep Learning: Neural Networks.

1.1 - History of Neural networks and Deep Learning.

1.2 - How Biological Neurons work?

1.3 - Growth of biological neural networks.

1.4 - Diagrammatic representation: Logistic Regression and Perceptron

MLPs

❖ Deep Multi-layer perceptrons

1.1 - Deep Multi-layer perceptrons: 1980s to 2010s

1.2 - Multi-Layered Perceptron (MLP).

1.3 - Notation.

1.4 - Training a single-neuron model.

1.5 - Training an MLP: Chain rule

1.6 - Training an MLP: Memoization

1.7 - Backpropagation algorithm.

1.8 - Activation functions.

1.9 - Vanishing Gradient problem.

1.10 - Bias-Variance tradeoff.

1.11 - Decision surfaces: Playground

1.12 - Auto Encoders.

1.13 - Word2Vec: CBOW.

1.14 - Word2Vec: Skip-gram

1.15 - Word2Vec: Algorithmic Optimizations

Advanced Optimization methods

1.1 - Optimizers: Hill-descent analogy in 2D

1.2 - Optimizers: Hill descent in 3D and contours.

1.3 - SGD recap.

1.4 - Batch SGD with Momentum.

1.5 - Nesterov Accelerated Gradient (NAG)

1.6 - Optimizers: AdaGrad

1.7 - Optimizers: Adadelta and RMSProp

1.8 - Adam

1.9 - Which algorithm to choose when?

Subject 2

Deep Learning

CNNs

Deep Learning: Convolutional Neural Nets.

1.1 - Biological inspiration: Visual Cortex

1.2 - Convolution: Edge Detection on images.

1.3 - Convolution: Padding and strides

1.4 - Convolution over RGB images.

1.5 - Convolutional layer.

1.6 - Max-pooling.

1.7 - CNN Training: Optimization

1.8 - Example CNN: LeNet [1998]

1.9 - ImageNet dataset

1.10 - Data Augmentation.

1.11 - Convolution Layers in Keras

1.12 - AlexNet

1.13 - VGGNet

1.14 - Residual Network.

1.15 - Inception Network.

1.16 - What is Transfer Learning?

1.17 - Code example: Cats vs Dogs.

RNNs

✣ Deep Learning: Long Short-Term Memory (LSTMS)

1.1 - Why RNNs?

1.2 - Recurrent Neural Network.

1.3 - Training RNNs: Backprop.

1.4 - Types of RNNs.

1.5 - Need for LSTM/GRU.

1.6 - LSTM.

1.7 - GRUs.

1.8 - Deep RNN.

1.9 - Bidirectional RNN.

1.10 - Code example : IMDB Sentiment classification

Transformers

✣ Deep Learning: Transformers and BERT

1.1 - Transformers and BERT

TensorFlow

✣ Deep Learning: Tensorflow and Keras.

1.1 - Tensorflow and Keras Overview.

1.2 GPU vs CPU for Deep Learning.

1.3 Google Collaboratory.

1.4 Install TensorFlow.

1.5 Online documentation and tutorials.

1.6 Softmax Classifier on MNIST dataset.

1.7 MLP: Initialization

1.8 Model 1: Sigmoid activation.

1.9 Model 2: ReLU activation.

1.10 Model 3: Batch Normalization.

1.11 Model 4 : Dropout.

1.12 - MNIST classification in Keras.

1.13 - Hyperparameter tuning in Keras.

Subject 3

Thesis Projects (8 credits)

Industry or Research focused Thesis - Machine Learning

NLP

Quora question Pair Similarity Problem

1.1 - Business/Real world problem : Problem Definition

1.2 - Business objectives and constraints.

1.3 - Mapping to an ML problem: Data overview

1.4 - Mapping to an ML problem: ML problem and performance metric.

1.5 - Mapping to an ML problem: Train-test split

1.6 - EDA: Basic Statistics.

1.7 - EDA: Basic Feature Extraction.

1.8 - EDA: Text Preprocessing.

1.9 - EDA: Advanced Feature Extraction.

1.10 - EDA: Feature analysis.

1.11 - EDA: Data Visualization: T-SNE.

1.12 - EDA: TF-IDF weighted word-vector featurization.

1.13 - ML Models: Loading data.

1.14 - ML Models: Random Model.

1.15 - ML Models: Logistic Regression & Linear SVM

1.16 - ML Models: XGBoost

Personalized Cancer Diagnosis

2.1 - Business/Real world problem overview

2.2 - Business objectives and constraints.

2.3 - ML problem formulation: Data

2.4 - ML problem formulation: Mapping real world to ML problem.

2.5 - ML problem formulation: Train, CV and Test data construction.

2.6 - Exploratory Data Analysis: Reading data & preprocessing

2.7 - Exploratory Data Analysis: Distribution of Class-labels.

2.8 - Exploratory Data Analysis: "Random" Model.

2.9 - Univariate Analysis: Gene feature.

2.10 - Univariate Analysis: Variation Feature.

2.11 - Univariate Analysis: Text feature.

2.12 - Machine Learning Models: Data Preparation

2.13 - Baseline Model: Naive Bayes

2.14 - K-Nearest Neighbors Classification.

2.15 - Logistic Regression with class balancing

2.16 - Logistic Regression without class balancing

2.17 - Linear-SVM.

2.18 - Random-Forest with one-hot encoded features

2.19 - Random-Forest with response-coded features

2.20 - Stacking Classifier

2.21 - Majority Voting classifier.

❖ Stackoverflow Tag Predictor

3.1 - Business/Real world problem.

3.2 - Business objectives and constraints.

3.3 - Mapping to an ML problem: Data Overview

3.4 - Mapping to an ML problem: ML problem formulation.

3.5 - Mapping to an ML problem: Performance metrics.

3.6 - Hamming loss.

3.7 - EDA: Data Loading.

3.8 - EDA: Analysis of tags.

3.9 - EDA: Data Preprocessing.

3.10 - Data Modeling: Multi label Classification.

3.11 - Data Preparation.

3.12 - Train-Test Split.

3.13 - Featurization.

3.14 - Logistic Regression: One Vs Rest.

3.15 - Sampling data and tags + Weighted Models.

3.16 - Logistic Regression revisited.

3.17 - Why not use advanced techniques?

❖ Microsoft Malware Detection

4.1 - Business/Real world problem: Problem Definition.

4.2 - Business/Real world problem: objectives and constraints.

4.3 - Machine Learning Problem Mapping: Data Overview.

4.4 - Machine Learning Problem Mapping: ML Problem.

4.5 - Machine Learning Problem Mapping: Train and test splitting.

4.6 - Exploratory Data Analysis: Class Distribution.

4.7 - Exploratory Data Analysis: Feature extraction from byte files.

4.8 - Exploratory Data Analysis: Multivariate analysis of features from byte files.

4.9 - Exploratory Data Analysis: Train-Test class distribution.

4.10 - ML models- using byte files only: Random Model.

4.11 - k-NN.

4.12 - Logistic regression.

4.13 - Random Forest and Xgboost.

4.14 - ASM Files: Feature extraction and Multiprocessing.

4.15 - File-size feature.

4.16 - Univariate Analysis.

4.17 - t-SNE analysis.

4.18 - ML models on ASM file features.

4.19 - Models on all features: t-SNE.

4.20 - Models on all features: Random Forest and Xgboost

Graph Based

✣ Facebook Friend Recommendation using Graph mining.

1.1 - Problem Definition.
1.2 - Overview of graphs: Node/Vertex, edge/link, directed edge, path.
1.3 - Data Format & Limitations.
1.4 - Mapping to a supervised classification problem.
1.5 - Business Constraints & Metrics.
1.6 - EDA: Basic Stats.
1.7 - EDA: Follower and following stats.
1.8 - EDA: Binary Classification Tasks.
1.9 - EDA: Train and test split.
1.10 - Feature engineering on graphs: Jaccard & Cosine similarities.
1.11 - PageRank.
1.12 - Shortest Path.
1.13 - Connected-Components.
1.14 - Adar index.
1.15 - Kartz Centrality.
1.16 - HITS Score.
1.17 - SVD.
1.18 - Weight Features.
1.19 - Modeling.

Time Series

✣ Taxi demand prediction in New York City

1.1 - Business/Real world problem overview.
1.2 - Objectives and Constraints
1.3 - Mapping to ML problem: Data
1.4 - Mapping to ML problem: dask dataframes
1.5 - Mapping to ML problem: Fields/Features.
1.6 - Mapping to ML problem: Time series forecasting/Regression.
1.7 - Mapping to ML problem: Performance metrics.

1.8 - Data Cleaning: Latitude and Longitude data
1.9 - Data Cleaning: Trip Duration.
1.10 - Data Cleaning: Speed.
1.11 - Data Cleaning: Distance.
1.12 - Data Cleaning: Fare.
1.13 - Data Cleaning: Remove all outliers/erroneous points.
1.14 - Data Preparation: Clustering/Segmentation
1.15 Data Preparation: Time binning
1.16 Data Preparation: Smoothing time-series data.
1.17 Data Preparation: Smoothing time-series data cont..
1.18 Data Preparation: Time series and Fourier transforms.
1.19 Ratios and previous-time-bin values.
1.20 Simple moving average.
1.21 Weighted Moving average.
1.22 Exponential weighted moving average.
1.23 Results.
1.24 Regression models: Train-Test split & Features
1.25 Linear regression.
1.26 Random Forest regression.
1.27 Xgboost Regression.
1.28 Model comparison.

Recommender Systems

Amazon Fashion Discovery Engine

1.1 - Problem Statement: Recommend similar apparel products in e-commerce using product descriptions and images.
1.2 - Plan of action.
1.3 - Amazon Product Advertising API.
1.4 - Data Folders and Paths.
1.5 - Overview of the data and terminology.

1.6 - Data Cleaning and Understanding: Missing data in various features.

1.7 - Understand Duplicate rows.

1.8 - Remove duplicates: Part 1

1.9 - Remove duplicates: Part 2

1.10 - Text- Preprocessing: Tokenization and stop-word removal.

1.11 - Stemming

1.12 - Text-based product similarity: Converting text to an n-D vector: Bag Of Words

1.13 - Code for bag of words based product similarity

1.14 - TF-IDF: featuring text based on word-importance.

1.15 - Code for TF-IDF based product similarity.

1.16 - Code for IDF based product similarity.

1.17 - Text semantics based product similarity: Word2Vec(Featurizing text based on semantics similarity).

1.18 - Code for average Word2Vec product similarity.

1.19 - TF-IDF Weighted Word2Vec

1.20 - Code for IDF weighted Word2Vec product similarity.

1.21 - Weighted similarity using brand and color.

1.22 - Code for weighted similarity.

1.23 - Building a real-world solution.

1.24 - Deep learning based visual product similarity: ConvNets: How to featurize an - image: Edges, Shapes, and Parts.

1.25 - Using Keras + Tensorflow to extract features.

1.26 - Visual similarity based product similarity

1.27 - Measuring goodness of our solution: A/B Testing

Netflix Movie Recommendation System

2.1 - Business/Real world problem: Problem Definition. Objectives and constraints.

2.2 - Mapping to an ML Problem: Data Overview.

2.3 - Mapping to an ML Problem: ML Problem formulation.

2.4 - Exploratory Data Analysis: Data Preprocessing.

2.5 - Exploratory Data Analysis: Temporal Train-Test split.

2.6 - Exploratory Data Analysis: Preliminary data analysis.

2.7 - Exploratory Data Analysis: Sparse matrix representation.

2.8 - Exploratory Data Analysis: Average rating for various slices.

2.9 - Exploratory Data Analysis: Cold start problem.

2.10 - Computing similarity Matrices: User-User similarity matrix.

2.11 - Computing similarity Matrices: Movie-Movie similarity matrix

2.12 - Computing similarity Matrices: Does movie-movie similarity work?

2.13 - ML models: Suprise Library.

2.14 - Overview of the modeling strategy.

2.15 - Data Sampling.

2.16 - Google drive with intermediate files.

2.17 Featurization for regression.

2.18 - Data Transformation for surprise.

2.19 - Xgboost with 13 features.

2.20 - Surprize Baseline model.

2.21 - Xgboost + 13 features + Surprize baseline model.

2.22 - Surprize KNN predictors.

2.23 - Matrix factorization models using surprise.

2.24 - SVD++ with implicit feedback.

2.25 - Final models with all features and predictors.

2.26 - Comparison between various models.

Deep Learning - Time Series

Human Activity Recognition

2.1 - Human Activity Recognition: Problem Definition.

2.2 Dataset Understanding

2.3 - Data Cleansing & Preprocessing.

2.4 - EDA: Univariate analysis

2.5 - EDA: Data Visualization using t-SNE.

2.6 Classical ML models.

2.7 - Deep Learning model.

Convolutional Neural Networks

Self-Driving Car

1.1 - Self-driving car: Problem definition.

1.2 - Datasets

1.3 - Data Understanding & Analysis: Files and folders.

1.4 - Dash-cam images and steering angles.

1.5 - Split the dataset: Train VS Test

1.6 - EDA: Steering angles.

1.7 - Mean Baseline model: Simple.

1.8 - Deep learning model: Deep Learning for regression: CNN, CNN+RNN.

1.9 - Batch load the dataset.

1.10 - NVIDIA's end-to-end CNN model.

1.11 - Train the model.

1.12 - Test and visualize the output.

1.13 - Extensions.

Recurrent Neural Networks

Music Generation using Deep Learning

1.1 - Real world problem.

1.2 - Music Representation.

1.3 - Char-RNN with abc-notation: char-RNN model.

1.4 - Char-RNN with abc-notation: Data Preparation

1.5 - Char-RNN with abc-notation: Many to many RNN, Time Distributed Dense layer.

1.6 - Char-RNN with abc-notation: State full RNN.

1.7 - Char-RNN with abc-notation: Model architecture, Model training.

1.8 - Char-RNN with abc-notation: Music Generation

1.9 - Char-RNN with abc-notation: Generate Tabla music

1.10 - MIDI music generation.

1.11 - Survey Blog.

Projects which we are covering

Fashion Discovery Engine

Build An Autonomous Car

Movie Recommendetion System

Human Activity Recognition

Music Generation using Deep-Learning

Programming Tools, Languages and Libraries covered

python™

TensorFlow

SciPy

Keras

Pandas

matplotlib

NLP library NLTK

NumPy

scikit learn

Amazon EC2

jupyter

Thank You

good luck for your future endeavours

6309798883/
6309798882

pgduoh@appliedroots.com

www.appliedroots.com