

# Index

## A

### Activity selection algorithm

- appointment times, 29, 31
- greedy-choice property, 32
- induction, 29, 30
- operational time, 31
- optimal substructure
  - property, 32
- patients, 28
- proof by contradiction, 33
- properties, 32

### Artificial intelligence (AI)

- algorithmic bias, 239–241
- BMI calculation
  - approaches, 4
  - big data definition, 10
  - data science operations, 10
  - decision-making process, 6
  - deep learning, 8
  - definition, 3
  - features, 4
  - formal-ish definition, 7
  - machine learning, 7, 9
  - neural networks, 6
  - neurons layers, 8
  - parameters, 4
  - radiologist, 6

- scenario, 5
- supervised algorithms, 9
- training and testing data, 5
- unsupervised algorithms, 9
- chest x-ray classification,
  - 230, 243
- concepts, 229
- considerations, 238
  - conception, 11
  - height measurements, 11
  - pneumonia detection, 242
  - privacy-patient data/
    - biases, 13
  - training and testing
    - dataset, 12
  - transfer learning, 12
  - vertebral fracture, 11
- debugging, 231–238
- fundamental concepts, 15–17
- healthcare information
  - paradox, 2, 3
- Kaggle competitions code, 230
- meaning, 1
- optimize accuracy, 244
- patient privacy, 238
- potential architecture, 243
- snake-oil claims, 242–244

## INDEX

### Artificial neural networks (ANNs)

- backpropagation, 114
- biological neuron, 111
- classification/regression
  - tasks, 115
- components, 115
- convolutions
  - AlexNet, 119
  - convolutional operation
    - on, 117
  - filters, 116
  - ImageNet, 119
  - max pooling, 118
  - meaning, 116
  - transfer learning, 119
- hyperparameters, 115
- input layer, 113
- intermediate representation, 110
- learning rate, 114
- multilayer perceptron, 113
- neural networks, 110
- neurons, 110
- perceptrons, 112, 114
- RCNNs, LSTMs/RNNs, GANs,
  - 120, 121
- softmax, 113
- weights/biases, 111

## B

- Bagging, 101
- Big data/data science, 10, 11
- Brute-force approach, 21

## C

### Callback method

- EarlyStopping, 196, 197
- get\_callbacks method, 193, 194
- TensorBoard visualization,
  - 195, 196

### Central processing units (CPU), 18

### Clustering

- dimensionality reduction, 104
- k-Means clustering, 104–108
  - convergence, 104, 105
  - dendrogram, 107
  - elbow plot, 106
  - prior knowledge, 105, 106
  - silhouette score, 107
  - squared errors (WSS), 106
  - variations, 107

### machine learning, 103

- principal components analysis,
  - 108–110

### Computational thinking

- algorithmic thinking, 17
- algorithms/class, 25
  - activity selection algorithm,
    - 28–33, 35
  - approximation
    - algorithms, 38
  - Big O notation, 36
  - comparison check, 35
  - data structures, 38
  - Fibonacci numbers, 37
  - function header/
    - parameters, 35

- graph algorithms, 37
- random access memory, 34
- repetitive operations, 34
- sorting algorithms, 36
- sort operation, 36
- stable matching, 26–28
- strings, 38
- time step operation, 34
- brute-force approach, 21
- calculation, 19
- complexity classes, 24
- computer memory, 18
- conceptual perspective, 18
- greedy algorithm, 22
- input/output, 19
- looser constraints/shortcuts, 24
- MRI scans, 19
- NP-Hard problems, 24
- numerical operations, 18
- output device, 18
- possible combination, 21
- potential problems, 23
- recursion, 22
- traditional algorithmic
  - methods, 20
- training data, 19, 20
- Convolutional neural network (CNN)
  - backpropagation process, 191
  - binary cross entropy formula, 191
  - callback (*see* Callback method)
  - components, 185
  - convolutional part
    - activation function, 186
    - Conv2D method, 185
    - filters, 185
    - max pooling, 187, 189
    - parameters, 186
    - RGB color images, 186
    - sequential model, 186
  - data generators/augmenting
    - images
      - arguments, 177, 178
      - create\_generators
        - method, 178, 180
      - dataframes, 172
      - ImageDataGenerator,
        - 174, 175
      - load images from, 172
      - neural network libraries, 172
      - parameters, 177
      - pneumonia and normal
        - cases, 180
      - test generators, 173
      - train\_generator.next(), 181
    - dense part, 189
    - DICOM images, 226
    - evaluation (SmallNet *vs.* VGG16)
      - classification report, 219
      - confusion matrix, 219
      - get\_stats method, 219
      - load\_model method, 220
      - roc\_curve method, 219
      - RocCurveDisplay
        - method, 220
      - scikit-learn metrics, 219
      - SmallNet classification
        - report, 222
      - statistics, 218

## INDEX

- Convolutional neural
  - network (CNN) (*cont.*)
  - test generator, 219
  - VGG16 classification
    - report, 221
- external image, 223–227
- fit\_model method, 225
- fit/smallnet method
  - architecture output, 200
  - max pooling layers, 201
  - methods, 198
  - model.fit, 201
  - model layers, 197
  - plot\_model method, 199
  - TensorBoard, 203
  - training model, 201, 202
- Grad-CAM (*see* Grad-CAM technique)
- gradcam issues, 225
- loss function, 191
- metrics, 193
- optimizer model, 192
- perceptrons, 190
- project setup
  - colab setup, 164
  - download data, 165–167
  - relevant tasks, 164
- pydicom.dcmread(), 227
- radiologist’s opinion, 227
- ReLU activation function
  - graphs, 190, 191
- sigmoid function, 190
- SmallNet
  - architecture, 184
  - code structure, 182
  - general steps, 182
  - methods, 183
  - ReduceLROnPlateau
    - method, 184
  - sequential model, 183
- splitting data, 168–172
  - distribution, 171
  - generate\_dataframe
    - method, 170, 171
  - list comprehension, 170
  - NORMAL and PNEUMONIA
    - folder, 168
  - normal *vs.* pneumonia, 172
  - pd.DataFrame method, 171
  - source code, 168, 169
- training pipeline, 226
- VGG16 (transfer learning)
  - Adam optimizer, 208
  - dropout layer, 207
  - fit\_model method, 208, 209
  - Flatten() method, 207
  - ImageNet
    - classification, 204
  - make\_smallnet() method,
    - 205, 206
  - model\_fname method, 209
  - perceptron dense layer, 207
  - plot history, 211
  - pneumonia *vs.* normal, 204
  - SmallNet model history, 212
  - TensorBoard, 211
  - trainable property, 206
  - training history, 213

training progress, 210  
transfer learning, 204

## D

Data processing/cleaning  
categorical data, 135  
categorical data/one-hot  
encoding, 138–140  
classification\_report  
function, 150  
cleaning data, 136–138  
code classifier, 152  
Colab notebook, 128  
confusion matrix, 148  
counter function, 142  
cross\_validate function, 144  
data/isolating columns,  
130–133  
decision tree classifier, 144–146  
delimiter, 131  
disposition, 128  
dummy-variable version, 140  
evaluation, 147–151  
fig.add\_subplot method, 135  
graphviz library, 152  
grid searching, 146, 147  
libraries, 129–131  
NEISS.txt file, 130  
pipeline, 140–143  
visualization, 133–136, 151  
Debugging  
callback metrics, 234

case scenario, 234  
code search results, 237  
considerations  
patient privacy, 238  
documentation, 235, 236  
error message, 231  
guidelines, 232, 233  
make\_network method, 232  
stack overflow queries, 234

Decision trees  
bootstrapped aggregation, 101  
classification and regression  
trees (CART), 100, 101  
data processing/cleaning,  
150–154  
ensemble methods, 101–103  
Gini impurity coefficient, 100  
node/branches, 99  
structure, 98, 99  
Deep learning (DL), 8, 9, 11  
ANNs (*see* Artificial neural  
networks (ANNs))  
concepts, 229  
Dimensionality reduction, 104  
Dynamic programming  
approach, 38

## E, F

Elastic regression, 91  
Ensemble methods, 101–103  
Evaluation metrics, 121–124

**G, H**

- Gale-Shapley stable matching
  - algorithm, 26–28
- Grad-CAM technique, 163
  - Grad-CAM results, 216
  - model.predict, 215
  - prediction class, 216
  - SmallNet, 216
  - source code, 213
  - VGG16, 216
  - VizGradCam method, 213–215
- Greedy algorithm, 22
  - decision trees, 100, 101

**I, J**

- Instance learning algorithms, 93–98

**K**

- Kaggle hosts machine learning
  - competitions, 163
- k-fold cross validation, machine
  - learning, 124, 125
- K-nearest neighbors, machine
  - learning, 93–95

**L**

- LASSO, 90, 91
- Linear regression, 83–85
- Logistic regression, 85–89
- Long short-term
  - memory (LSTM), 120

**M**

- Machine learning (ML), 7, 9, 11, 79
  - algorithm fundamentals, 79–82
  - ANNs/DL, 110
  - clustering/dimensionality
    - reduction, 103–110
  - decision trees/tree-based
    - ensemble, 98–103
  - evaluation metrics, 121–124
  - instance learning, 93–98
  - k-fold cross validation, 124, 125
  - regression, 82–92
  - training process, 79, 80
- Maximum likelihood
  - estimation (MLE), 89
- Mean squared error (MSE), 122

**N, O**

- Natural language processing
  - algorithms (NLP), 8, 239

**P, Q**

- Principal components
  - analysis (PCA), 108–110
- Project setup
  - data generators/augmenting
    - images, test generators, 174
  - splitting data, training, validation,
    - and test data, 168
- PyCaret
  - blending, 159
  - classification methods, 152, 153

- compare\_models method, 156
- confusion matrix, 159
- decision stumps, 157
- description, 155
- exporting/loading model, 161
- GBM model, 158
- gradient boosting machine, 158
- interactive component, 154
- model selection, 152
- predict\_model function, 160
- setup function, 155
- top models variable, 157
- training process, 154, 156
- Python programming
  - bit information, 45–47
  - classes, 62
  - Colab notebook, 43
  - compiling, 42
  - data frame, 75
  - file input/for loops/iteration, 59–66
  - file output/dictionaries/list operations
    - dictionaries, 70, 73
    - equation, 66
    - opening and closing braces, 72
    - positive/negative root, 66, 67
    - read\_write\_file function, 68, 71
    - root\_finder function, 68, 70
    - source code, 72
  - function, 49–51
  - google account, 43
  - If statements, 54–58
  - input .csv file, 59, 60
  - interpolate, 53
  - for loop, 65
  - pandas, 74–77
  - prototyping tool, 44
  - CSV Reader object, 62
  - split method, 52
  - standard library, 45
  - text files, 41, 42
  - variables, 47–54
- R**
  - Random Access Memory (RAM), 34
  - Random forest, 102
  - Received operative characteristic (ROC), 87, 88
  - Recurrent neural networks (RNNs), 120
  - Recursion approach, 22
  - Region-based convolutional neural networks (RCNNs), 120
  - Regression techniques
    - hyperparameter, 91
    - LASSO/ridge/elastic Net, 89–92
    - linear regression, 83–85
    - logistic regression, 85–89
    - two/three dimensional data, 83
  - Ridge regression, 89–92
- S**
  - scikit-learn, 127

## INDEX

SmallNet (convolutional neural network), 183  
Supervised algorithms, 9  
Supervised learning algorithms, 81  
Support vector machines (SVMs)  
  line/polynomial tasks, 97  
  margin classifiers, 95, 96  
  optimal line/plane/hyperplane, 96, 97  
  real-world usage, 97  
  regression tasks, 97  
Synthetic Minority Oversampling Technique (SMOTE), 142

## T

Traveling salesman problem (TSP), 21, 23

## U, V, W

UNet and Siamese networks, 121  
Unsupervised algorithms, 9, 82

## X, Y, Z

XGBoost algorithm, 102