INTRODUCTION TO MACHINE LEARNING

Dr. Kamakshaiah Musunuru

About the Author

Dr. M. Kamakshaiah is a open source software evangelist, enterprise solutions architect and academic of data science and analytics. His teaching interests are IT practices in business, MIS, Data Science, Business Analytics including functional analytics related to marketing, finance, HRM, quality and operations. He also teaches theoretical concepts like multivariate analytics, numerical simulations & optimization, machine learning & AI using few programming languages like R, Python, Java. Internet of Things and big data analytics (Hadoop) are his all time favorites for teaching.

He has taught abroad for two years and credits two country visits. He has developed few free and open source software solutions meant for corporate practitioners, academics, scholars and students engaging in data science and analytics. All his software applications are available from his Github portal https://github.com/Kamakshaiah.

Foreword

I started writing this book for one simple reason that I got to teach machine learning for business analytics students. Actually I started preserving every little code, chunk by chunk, as and when I struck with this subject. Though I thought it as notes at first but the stuff that I gathered became rather more significant to compile it as a book.

I turned into data analyst due to my being accidentally exposed to open source software when I was in Ethiopia. I got to use GNU/Linux to keep my personal computer away from virus infection at the university. I crashed Windows a couple of times when I was working in office at the University, eventually I found GNU/Linux as a wonderful solution to address virus issues. My first experience of GNU/Linux, perhaps, is *Jaunty Jackalope*, as I suppose. This must be Ubuntu 9.04, an LTS version, if I am not wrong. However, my involvement in GNU/Linux became a serious affair though *Karmic Koala*. [1] I still remember few of my colleagues used to visit my home for OS installation in those days of stay at Ethiopia.

My first encounter with analytics software is R. R is *lingua franca* of statistics. I taught "business statistics" several times during

[1] *Karmic Koala* is Ubuntu 9.10. Visit https://wiki.ubuntu.com/Releases for more information on Ubuntu releases.

my stint as academic. Once, I notice this little yet uppercase R letter over there while I was browsing for statistical software in Ubuntu *package manager*. R changed not only my understanding of Statistics but the very way of learning the same. I addicted to R so much so that for every pretty little calculations I used to refer R manuals. I must be the first academic to introduce R in south Indian business management curriculum, yet remained unknown to others. Today, R is one of the best tools for statistical analysis and practice of quantitative techniques.

I came across Python just as in same way as I discovered R in Ubuntu. Python was one of the default programming languages for many things in Ubuntu. At first I did not know the power of Python for everything I learned it was only by self study. All my learning is from online resources. However, I could produce thousands of students majored in data analysts through my formal classes. All that I learned and taught is only by my self study, I mean, through very informal personal learning.

I am writing all this not to show myself as a valiant learner, but to demonstrate how can a novice and a naive enthusiast like me can learn programming tools like R and Python. Today, I am offering a couple of courses to teach Hadoop and IoT, that is all by passion. So, I would like to give confidence to the reader that you don't need any formal learning in computer science or IT to learn data science and adopt it as a profession. However, you need tons and tons of patience and passion.

This book is meant for beginners of machine learning and practice. It has 5 chapters each section represents a unique concept of data analytics. This book may be useful for both practitioners and academics to acquire knowledge of machine learning accompanied with Python practice. The first chapter *introduction*, deals with very short information related to basics of machine learning. Chapter I has information related to installation of Python in Linux, Windows and few other OSes. Chapter 2 deals with *Supervised learning* and this chapter has information re-

lated to various supervised machine learning algorithms such as ... Chapter 3, *Unsuptervised learning* deals with algorithms such as ... Chapter 4, *Deep lerning*, deals with different types of techniques related to ...

As far as coding is concerned; those code sections where there exist left-bar such as the below

```
statement 1
statement 2
statement 3
```

represents a ***script***. A script is a plain code file in which there exists program statements. There are other code sections where each statement is preceded by python prompt (>>>). These code sections are meant for testing. These sections are useful either to evaluate a Python statement or provide evidence for logic.

All the code chunks used in this book are provided though my github portal with a project name *PfDSaA*. Feel free to visit the site https://github.com/Kamakshaiah/PfDSaA and download required *.py* files for practice.

One last note is that this book is meant for both data scientists and analysts. The learner need to have basics of Python. However, little logical thinking together with passion and patience are required. The book is written in such a way that even a person having no knowledge on Python can pick up and become maverick at the end of reading. This book covers from very basic details ranging from installation to creating packages. I am not covering very advanced concepts such as software and applications development due to one reason to keep this book reasonable for both beginners and advanced users.

Happy reading $\cdots$

Author
Dr. M. Kamakshaiah

Contents

Chapter 1

Introduction

Machine learning (ML) is a field of inquiry devoted to understanding and building methods that "learn", that is, methods that leverage data to improve performance on some set of tasks. [1] It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers, but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. [2] Some implementations of machine learning use data and neural networks in a way that

mimics the working of a biological brain. [3] In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning programs can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step. [4]

The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm(s) it uses to determine correct answers. For example, to train a system for the task of digital character recognition, the MNIST dataset of handwritten digits has often been used.

1.0.1 History and relationships to other fields

The term machine learning was coined in 1959 by *Arthur Samuel*, an IBM employee and pioneer in the field of computer gaming and artificial intelligence. [5] [6] Also the synonym self-teaching computers were used in this time period. [7]

By the early 1960s an experimental "learning machine" with punched tape memory, called *Cybertron*, had been developed by *Raytheon Company* to analyze sonar signals, electrocardiograms and speech patterns using rudimentary reinforcement learning.

It was repetitively trained by a human operator/teacher to recognize patterns and equipped with a "goof" button to cause it to re-evaluate incorrect decisions. A representative book on research into machine learning during the 1960s was *Nilsson's* book on Learning Machines, dealing mostly with machine learning for pattern classification. Interest related to pattern recognition continued into the 1970s, as described by Duda and Hart in 1973. [8] In 1981 a report was given on using teaching strategies so that a neural network learns to recognize 40 characters (26 letters, 10 digits, and 4 special symbols) from a computer terminal.

Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." [9] This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper "Computing Machinery and Intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?".

Modern day machine learning has two objectives, one is to classify data based on models which have been developed, the other purpose is to make predictions for future outcomes based on these models. A hypothetical algorithm specific to classifying data may use computer vision of moles coupled with supervised learning in order to train it to classify the cancerous moles. A machine learning algorithm for stock trading may inform the trader of future potential predictions.

1.0.2 Artificial intelligence

As a scientific endeavor, machine learning grew out of the quest for artificial intelligence. In the early days of AI as an academic

discipline, some researchers were interested in having machines learn from data. They attempted to approach the problem with various symbolic methods, as well as what was then termed "neural networks"; these were mostly *perceptrons* and other models that were later found to be reinventions of the generalized linear models of statistics.[10] Probabilistic reasoning was also employed, especially in automated medical diagnosis. [11]

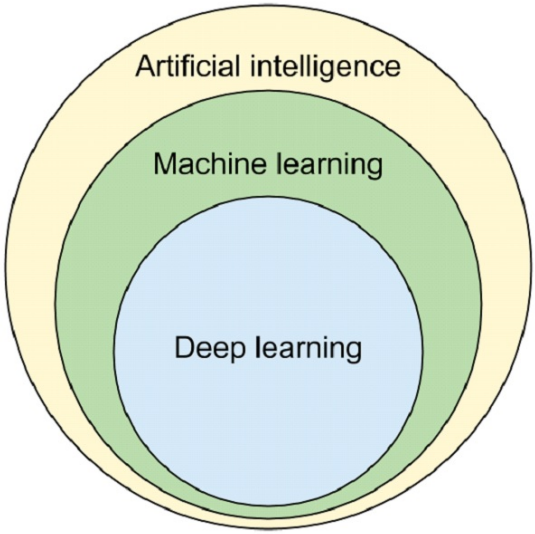

Figure 1.1: AI vs ML

However, an increasing emphasis on the logical, knowledge-based approach caused a rift between AI and machine learning. Probabilistic systems were plagued by theoretical and practical problems of data acquisition and representation.? By 1980, expert systems had come to dominate AI, and statistics was out of favor. [12] Work on symbolic/knowledge-based learning did continue within AI, leading to inductive logic programming, but the more statistical line of research was now outside the field

of AI proper, in pattern recognition and information retrieval. Neural networks research had been abandoned by AI and computer science around the same time. This line, too, was continued outside the AI/CS field, as "connectionism", by researchers from other disciplines including *Hopfield, Rumelhart* and *Hinton*. Their main success came in the mid-1980s with the reinvention of *backpropagation*.

Machine learning (ML), reorganized as a separate field, started to flourish in the 1990s. The field changed its goal from achieving artificial intelligence to tackling solvable problems of a practical nature. It shifted focus away from the symbolic approaches it had inherited from AI, and toward methods and models borrowed from statistics, fuzzy logic, and probability theory.

The difference between ML and AI is frequently misunderstood. ML learns and predicts based on passive observations, whereas AI implies an agent interacting with the environment to learn and take actions that maximize its chance of successfully achieving its goals. [13]

As of 2020, many sources continue to assert that ML remains a subfield of AI. Others have the view that not all ML is part of AI, but only an "intelligent subset" of ML should be considered AI.

1.0.3 Data mining

Machine learning and data mining often employ the same methods and overlap significantly, but while machine learning focuses on prediction, based on known properties learned from the training data, data mining focuses on the discovery of (previously) unknown properties in the data (this is the analysis step of knowledge discovery in databases). Data mining uses many machine learning methods, but with different goals; on the other hand, machine learning also employs data mining methods as "unsupervised learning" or as a *preprocessing* step to improve learner

accuracy. Much of the confusion between these two research communities comes from the basic assumptions they work with. *In machine learning, performance is usually evaluated with respect to the ability to reproduce known knowledge, while in knowledge discovery and data mining (KDD) the key task is the discovery of previously unknown knowledge.* Evaluated with respect to known knowledge, an uninformed (unsupervised) method will easily be outperformed by other supervised methods, while in a typical KDD task, supervised methods cannot be used due to the unavailability of training data.

1.0.4 Optimization

Machine learning also has intimate ties to optimization. Many learning problems are formulated as minimization of some loss function on a training set of examples. Loss functions express the discrepancy between the predictions of the model being trained and the actual problem instances. For example, in classification, one wants to assign a label to instances, and models are trained to correctly predict the preassigned labels of a set of examples.

1.0.5 Generalization

The difference between optimization and machine learning arises from the goal of generalization. While optimization algorithms can minimize the loss on a training set, machine learning is concerned with minimizing the loss on unseen samples. Characterizing the generalization of various learning algorithms is an active topic of current research, especially for deep learning algorithms.

1.0.6 Statistics

Machine learning and statistics are closely related fields in terms of methods, but distinct in their principal goal. Statistics draws population inferences from a sample, while machine learning finds generalizable predictive patterns. [14] According to *Michael*

I. Jordan, the ideas of machine learning, from methodological principles to theoretical tools, have had a long pre-history in statistics.[15] He also suggested the term data science as a placeholder to call the overall field. *Leo Breiman* distinguished two statistical modeling paradigms. Data model and algorithmic model, wherein "algorithmic model" means more or less the machine learning algorithms like Random forest. Some statisticians have adopted methods from machine learning, leading to a combined field that they call statistical learning.[16]

1.1 Categories of Machine Learning

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

1. *Supervised learning*: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

2. *Unsupervised learning*: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

3. *Reinforcement learning*: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.

1.1.1 Supervised learning

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs.[17] The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs.[18] An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.[19]

Types of supervised-learning algorithms include active learning, classification and regression.[20] Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. As an example, for a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email.

Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

1.1.2 Unsupervised learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function. [21] Though unsupervised learning encompasses other domains involving summarizing and explaining data features.

Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.

1.1.3 Semi-supervised learning

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Some of the training examples are missing training labels, yet many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy. In weakly supervised learning, the training labels are noisy, limited, or imprecise; however, these labels are often cheaper to obtain, resulting in larger

effective training sets.

1.1.4 Reinforcement learning

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov decision process (MDP). Many reinforcement learning algorithms use dynamic programming techniques.[22] Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

1.2 Machine learning process

Machine learning is the process of imparting intelligence to machines seems daunting and impossible. But it is actually really easy. It can be broken down into 7 major steps.

1.2.1 Collecting Data

Machines initially learn from the data that you give them. It is of the utmost importance to collect reliable data so that machine learning model can find the correct patterns. The quality of the data that was fed in to the machine will determine how accurate the model is. Incorrect or outdated data, gives rise to wrong outcomes or predictions which are not relevant.

It is always better to procure data from a reliable source, as it

will directly affect the outcome of the model. Good data is relevant, contains very few missing and repeated values, and has a good representation of the various subcategories/classes present. Given the problem that needs to be solved, then investigation to obtain data need to be done. The quality and quantity of information is very important since it will directly impact how well or badly the model will work. At times data may be available from existing database or it needs to be created from scratch. If it is a small project you can create a spreadsheet that will later be easily exported as a CSV file. It is also common to use the web scraping technique to automatically collect information from various sources such as APIs.

1.2.2 Preparing the Data

Once data is collected visualizing data becomes a priority. Correlations between the different characteristics need to be assessed. It will be necessary to make a selection of characteristics since these characteristics will impact the execution times and the results. PCA is useful for reducing dimensions if necessary. Additionally, balance the amount of data for each class so that it is significant as the learning may be biased towards a type of response and when your model tries to generalize knowledge it will fail. You must also separate the data into two groups: one for *training* and the other for *model evaluation* which can be divided approximately in a ratio of 80/20 but it can vary depending on the case and the volume of data. At this stage, data needs to be processed through normalization, eliminating duplicates, and making error corrections.

1.2.3 Choose the model

A machine learning model determines the output you get after running a machine learning algorithm on the collected data. It is important to choose a model which is relevant to the task at hand. Over the years, scientists and engineers developed various

models suited for different tasks like speech recognition, image recognition, prediction, etc. Apart from this, model needs to be chosen based on type of data i.e. categorical & non-categorical, numerical & non-numerical etc.

There are several models available for machine learning practice. Mostly they are algorithms available for different methods such as classification, prediction, linear regression, clustering, K-Nearest Neighbor, Deep Learning, Neural Networks, and more. There are various models to be used depending on the data that is going to be processed such as images, sound, text, and numerical values. In the following table, there are few models and their applications that can be applied in projects.

Model	Applications
Logistic Regression	Price prediction
Fully connected networks	Classification
Convolutional Neural Networks	Image processing
Recurrent Neural Networks	Voice recognition
Random Forest	Fraud Detection
Reinforcement Learning	Learning by trial and error
Generative Models	Image creation
K-means	Segmentation
k-Nearest Neighbors	Recommendation systems
Bayesian Classifiers	Spam and noise filtering

Table 1.1: ML Algorithms and applications

1.2.4 Training the Model

Training is the most important step in machine learning. In training, data will be passed to the machine learning model to find patterns and make predictions. It results in the model learning from the data so that it can accomplish the task set. Over time, with training, the model gets better at predicting.

Data need to be trained smoothly to see an incremental improve-

ment in the prediction rate. Remember to initialize the weights of your model randomly. The weights are the values that multiply or affect the relationships between the inputs and outputs. These weights are automatically adjusted by the selected algorithm while getting trained.

1.2.5 Evaluating the Model

After training the model, the model need to be tested for the performance. This is done by testing the performance of the model on previously unseen data. The unseen data used is the testing set that was split earlier. If testing was done on the same data which is used for training, that will not get an accurate measure, as the model is already used to the data, and finds the same patterns in it, as it previously did. This will give you disproportionately high accuracy. When used on testing data, it is possible to get accurate measure of how the model will perform and its speed. If the accuracy is less than or equal to 50%, that model will not be useful since it would be like tossing a coin to make decisions. If the precision is 90% or more, it means the model is a best one to rely upon.

1.2.6 Parameter Tuning

Once the model was created and evaluated, it should be verfied that whether the accuracy can be improved or not. This is done by tuning the parameters present in your model. Parameters are the variables in the model that the programmer generally decides. The accuracy can be maximum at a particular value of the parameter. Parameter tuning refers to finding these values.

At times a model can be overfitting or underfitting if the evaluation did not obtain good predictions and precision is not the minimum desired. It is possible to increase the number of training times as in *epochs*. Another important parameter is the one known as the “learning rate”, which is usually a value that mul-

tiplies the gradient to gradually bring it closer to the global or local minimum to minimize the cost of the function.

Increasing your values by 0.1 units from 0.001 is not the same as this can significantly affect the model execution time. Indicate the maximum error allowed for the model. At times it can take a few minutes to hours, and even days, to train teh machine. These parameters are often called *Hyperparameters*. This "tuning" is still more of an art than a science and will improve as you experiment. There are usually many parameters to adjust and when combined they can trigger all the options. Each algorithm has its own parameters to adjust. To name a few more, in Artificial Neural Networks (ANNs) architecture the number of hidden layers need to be decided and gradually to be tested with how many neurons each layer can accommodate. This will be a work of great effort and patience to give good results.

1.2.7 Making Predictions

In the end, the model will be used on the data to make predictions accurately. Usually, predictions are not a matter of concern, but accuracies will be calculated based on *confusion matrix*. A confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa. The name stems from the fact that whether the system is confusing two classes (i.e. commonly mislabeling one as another).

1.3 Train, validate & test data

In machine learning, a common task is the study and construction of algorithms that can learn from and make predictions on

data.[23] Such algorithms function by making data-driven predictions or decisions, through building a mathematical model from input data. These input data used to build the model are usually divided in multiple data sets. In particular, three data sets are commonly used in different stages of the creation of the model: *training, validation* and *test* sets.

The model is initially fit on a training data set, which is a set of examples used to fit the parameters (e.g. weights of connections between neurons in artificial neural networks) of the model.[24] The model (e.g. a naive Bayes classifier) is trained on the training data set using a supervised learning method, for example using optimization methods such as gradient descent or stochastic gradient descent. In practice, the training data set often consists of pairs of an input vector (or scalar) and the corresponding output vector (or scalar), where the answer key is commonly denoted as the target (or label). The current model is run with the training data set and produces a result, which is then compared with the target, for each input vector in the training data set. Based on the result of the comparison and the specific learning algorithm being used, the parameters of the model are adjusted. The model fitting can include both variable selection and parameter estimation.

Successively, the fitted model is used to predict the responses for the observations in a second data set called the validation data set. The validation data set provides an unbiased evaluation of a model fit on the training data set while tuning the model's hyperparameters (e.g. the number of hidden units layers and layer widths in a neural network). Validation datasets can be used for regularization by early stopping (stopping training when the error on the validation data set increases, as this is a sign of over-fitting to the training data set). This simple procedure is complicated in practice by the fact that the validation dataset's error may fluctuate during training, producing multiple local minima. This complication has led to the creation of many *ad hoc* rules for deciding when over-fitting has truly begun.

Finally, the test data set is a data set used to provide an unbiased evaluation of a final model fit on the training data set. If the data in the test data set has never been used in training (for example in cross-validation), the test data set is also called a holdout data set. The term "validation set" is sometimes used instead of "test set" in some literature (e.g., if the original data set was partitioned into only two subsets, the test set might be referred to as the validation set). Deciding the sizes and strategies for data set division in training, test and validation sets is very dependent on the problem and data available.

1.3.1 Training data set

A training data set is a data set of examples used during the learning process and is used to fit the parameters (e.g., weights) of, for example, a classifier.[25] For classification tasks, a supervised learning algorithm looks at the training data set to determine, or learn, the optimal combinations of variables that will generate a good predictive model.[26] The goal is to produce a trained (fitted) model that generalizes well to new, unknown data. The fitted model is evaluated using "new" examples from the held-out datasets (validation and test datasets) to estimate the model s accuracy in classifying new data. To reduce the risk of issues such as over-fitting, the examples in the validation and test datasets should not be used to train the model. Most approaches that search through training data for empirical relationships tend to overfit the data, meaning that they can identify and exploit apparent relationships in the training data that do not hold in general.

1.3.2 Validation data set

A validation data set is a data-set of examples used to tune the hyperparameters (i.e. the architecture) of a classifier. It is sometimes also called the development set or the "dev set". An example of a hyperparameter for artificial neural networks

includes the number of hidden units in each layer. [27] It, as well as the testing set (as mentioned below), should follow the same probability distribution as the training data set.

In order to avoid overfitting, when any classification parameter needs to be adjusted, it is necessary to have a validation data set in addition to the training and test datasets. For example, if the most suitable classifier for the problem is sought, the training data set is used to train the different candidate classifiers, the validation data set is used to compare their performances and decide which one to take and, finally, the test data set is used to obtain the performance characteristics such as *accuracy, sensitivity, specificity, F-measure*, and so on. The validation data set functions as a hybrid: it is training data used for testing, but neither as part of the low-level training nor as part of the final testing.

1.3.3 Test data set

A test data set is a data set that is independent of the training data set, but that follows the same probability distribution as the training data set. If a model fit to the training data set also fits the test data set well, minimal overfitting has taken place (see figure below). A better fitting of the training data set as opposed to the test data set usually points to over-fitting.

A test set is therefore a set of examples used only to assess the performance (i.e. generalization) of a fully specified classifier. To do this, the final model is used to predict classifications of examples in the test set. Those predictions are compared to the examples' true classifications to assess the model's accuracy.

In a scenario where both validation and test datasets are used, the test data set is typically used to assess the final model that is selected during the validation process. In the case where the original data set is partitioned into two subsets (training and test datasets), the test data set might assess the model only once

(e.g., in the holdout method). Note that some sources advise against such a method. However, when using a method such as cross-validation, two partitions can be sufficient and effective since results are averaged after repeated rounds of model training and testing to help reduce bias and variability.

1.3.4 Cross validation

Cross-validation, sometimes called rotation estimation or out-of-sample testing, is any of various similar model validation techniques for assessing how the results of a statistical analysis will generalize to an independent data set. Cross-validation is a resampling method that uses different portions of the data to test and train a model on different iterations. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of known data on which training is run (training dataset), and a dataset of unknown data (or first seen data) against which the model is tested (called the validation dataset or testing set). The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias and to give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem).

One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). To reduce variability, in most methods multiple rounds of cross-validation are performed using different partitions, and the validation results are combined (e.g. averaged) over the rounds to give an estimate of the model's predictive performance. In summary, cross-validation combines (averages) measures of fitness in prediction to derive a more accurate estimate of model prediction

performance.

Method

Assume a model with one or more unknown parameters, and a data set to which the model can be fit (the training data set). The fitting process optimizes the model parameters to make the model fit the training data as well as possible. If an independent sample of validation data is taken from the same population as the training data, it will generally turn out that the model does not fit the validation data as well as it fits the training data. The size of this difference is likely to be large especially when the size of the training data set is small, or when the number of parameters in the model is large. Cross-validation is a way to estimate the size of this effect.

In linear regression, there exist real response values *y1, ..., yn*, and n p-dimensional vector covariates *x1, ..., xn*. The components of the vector *xi* are denoted *xi1, ..., xip*. If least squares is used to fit a function in the form of a hyperplane $\hat{y} = a + \beta^T x$ to the data $(xi, yi)1 \leq i \leq n$, then the fit can be assessed using the mean squared error (MSE). The MSE for given estimated parameter values a and β on the training set $(xi, yi)1 \leq i \leq n$ is defined as:

$$\begin{aligned}\text{MSE} &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - a - \boldsymbol{\beta}^T \mathbf{x}_i)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}(y_i - a - \beta_1 x_{i1} - \cdots - \beta_p x_{ip})^2\end{aligned}$$

If the model is correctly specified, it can be shown under mild assumptions that the expected value of the MSE for the training set is $(n?p?1)/(n + p + 1) < 1$ times the expected value of the MSE for the validation set (the expected value is taken over the distribution of training sets). Thus, a fitted model and computed MSE on the training set will result in an optimistically biased

assessment of how well the model will fit an independent data set. This biased estimate is called the in-sample estimate of the fit, whereas the cross-validation estimate is an out-of-sample estimate.

Since in linear regression it is possible to directly compute the factor $(n?p?1)/(n+p+1)$ by which the training MSE underestimates the validation MSE under the assumption that the model specification is valid, cross-validation can be used for checking whether the model has been overfitted, in which case the MSE in the validation set will substantially exceed its anticipated value. (Cross-validation in the context of linear regression is also useful in that it can be used to select an optimally regularized cost function.) In most other regression procedures (e.g. logistic regression), there is no simple formula to compute the expected out-of-sample fit. Cross-validation is, thus, a generally applicable way to predict the performance of a model on unavailable data using numerical computation in place of theoretical analysis.

1.3.5 Overfitting/Underfitting a Model

As we know, in machine learning the data is usually split into two subsets: *training data* and *testing data*, and fit our model on the train data, in order to make predictions on the test data. As a result, there are two issues while fitting a model. They are *overfitting* or *underfitting*. These issues must be tackled in order to maintain accuracy

Overfitting

Overfitting means that model is beign trained "too well". This usually happens when the model is too complex (i.e. too many features/variables compared to the number of observations). This model will be very accurate on the training data but will probably be very not accurate on untrained or new data. It is because this model is not generalized leading to bad results or prediction.

Basically, when this happens, the model learns or describes the "noise" in the training data instead of the actual relationships between variables in the data. This noise, obviously, is not part in of any new dataset, and cannot be applied to it.

Underfitting

In contrast to overfitting, when a model is underfitted, it means that the model does not fit the training data and therefore misses the trends in the data. It also means the model cannot be generalized to new data. This is usually the result of a very simple model. This happens due to lack of enough predictors/independent variables. It could also happen when, for example, we fit a linear model (like linear regression) to data that is not linear. It almost goes without saying that this model will have poor predictive ability (on training data and can t be generalized to other data). It is worth noting the underfitting is not as prevalent as overfitting. Nevertheless, avoiding both of those problems in data analysis yields good outcomes.

1.3.6 Python practice

Scikit-Learn library and specifically the `train_test_split` method. `train_test_split` method is used to split arrays or matrices into random train and test subsets. Following code snippet shows as to how packages can be imported. The another package is *numpy* which is used to simulate data sets required for our task.

```
>>> import numpy as np
>>> from sklearn.model_selection import train_test_split
>>> X, y = np.arange(10).reshape((5, 2)), range(5)
>>> X
array([[0, 1],
       [2, 3],
       [4, 5],
       [6, 7],
       [8, 9]])
>>> list(y)
```

```
[0, 1, 2, 3, 4]
```

In the above snippet there are two packages that were imported namely *numpy* and *sklearn*. The package *numpy* is useful for data simulations. The function `train_test_split` from the package *sklearn* is useful for train, test and split tasks. Below code snippet performs the required tasks.

```
>>> X_train, X_test, y_train, y_test = train_test_split(X
                                ↪ , y, test_size=0.33,
                                ↪ random_state=42)
>>> X_train
array([[4, 5],
       [0, 1],
       [6, 7]])
>>> X_test
array([[2, 3],
       [8, 9]])
>>> y_train
[2, 0, 3]
>>> y_test
[1, 4]
```

Let's try fitting data using Linear Regression and then test our data with cross validations.

```
from sklearn.linear_model import LinearRegression

lm = LinearRegression()
model = lm.fit(X_train, y_train)
predictions = lm.predict(X_test)
print(model.score(X_test, y_test))
print(predictions)
```

The output would be

```
>>> model.score(X_test, y_test)
1.0

>>> predictions
array([1., 4.])
```

Compare `predictions` with `y_test` values. Perfect predictions. Is not? There are methods to know about intercept and coefficients. That is not required for the aim of this text here is not about linear regression. As far as cross validation is required, it is not necessary, because it is a perfect fit as we know (score is 1). However, it can be done as shown below for practice

```
>>> scores = cross_val_score(model, X, y, cv=2)
>>> print("Cross-validated scores:", scores)
Cross-validated scores: [1. 1.]
>>> predictions = cross_val_predict(model, X, y, cv=2)
>>> for i in predictions:
...     print(i)
...
3.885780586188048e-16
1.0000000000000002
2.0000000000000004
3.0
4.0
```

Perfect predictions.

Exercises

1. Learn data simulations using Python. There are number of ways to make data sets quickly using Python programming. Appendix 1 shows few methods to perform data simulations. Create few data variables such as numeric and non-numeric and try to create data sets using *pandas* library.
2. Take a sample data set with certain valid varibles

Chapter 2

Supervised Learning

2.1 Linear Regression

2.2 Logistic Regression

2.3 Decision Trees

2.4 Na ve Bayes Algorithm

2.5 K Nearest Neighbour (KNN)

2.6 Random Forest

2.7 Rule based learning

2.7.1 Apriori Algorithm

Notes

[1]Mitchell, Tom (1997). Machine Learning. New York: McGraw Hill. ISBN 0-07-042807-7.

[2]Friedman, Jerome H. (1998). "Data Mining and Statistics: What's the connection?". Computing Science and Statistics. 29 (1): 3 9.

[3]Zhou, Victor (2019-12-20). "Machine Learning for Beginners: An Introduction to Neural Networks". Medium. Retrieved 2021-08-15.

[4]Ethem Alpaydin (2020). Introduction to Machine Learning (Fourth ed.). MIT. pp. xix, 1 3, 13 18. ISBN 978-0262043793.

[5]Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of Research and Development. 3 (3): 210 229.

[6]R. Kohavi and F. Provost, "Glossary of terms," Machine Learning, vol. 30, no. 2 3, pp. 271 274, 1998.

[7]Gerovitch, Slava (9 April 2015). "How the Computer Got Its Revenge on the Soviet Union". Nautilus. Retrieved 19 September 2021.

[8]Duda, R., Hart P. Pattern Recognition and Scene Analysis, Wiley Interscience, 1973

[9]Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2. ISBN 978-0-07-042807-2.

[10]Sarle, Warren (1994). "Neural Networks and statistical models". CiteSeerX 10.1.1.27.699.

[11]Russell, Stuart; Norvig, Peter (2003) [1995]. Artificial Intelligence: A Modern Approach (2nd ed.). Prentice Hall. ISBN 978-0137903955.

[12]Langley, Pat (2011). "The changing science of machine learning". Machine Learning. 82 (3): 275 279. doi:10.1007/s10994-011-5242-y

[13]Alpaydin, Ethem (2010). Introduction to Machine Learning. MIT Press. p. 9. ISBN 978-0-262-01243-0.

[14]Bzdok, Danilo; Altman, Naomi; Krzywinski, Martin (2018). "Statistics versus Machine Learning". Nature Methods. 15 (4): 233 234.

[15]Michael I. Jordan (2014-09-10). "statistics and machine learning". reddit. Retrieved 2014-10-01.

[16]Gareth James; Daniela Witten; Trevor Hastie; Robert Tibshirani (2013). An Introduction to Statistical Learning. Springer. p. vii.

[17]Russell, Stuart J.; Norvig, Peter (2010). Artificial Intelligence: A Modern Approach (Third ed.). Prentice Hall. ISBN 9780136042594.

[18]Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2012). Foundations of Machine Learning. The MIT Press. ISBN 9780262018258.

[19]Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2. ISBN

978-0-07-042807-2.

[20]Alpaydin, Ethem (2010). Introduction to Machine Learning. MIT Press. p. 9. ISBN 978-0-262-01243-0.

[21]Jordan, Michael I.; Bishop, Christopher M. (2004). "Neural Networks". In Allen B. Tucker (ed.). Computer Science Handbook, Second Edition (Section VII: Intelligent Systems). Boca Raton, Florida: Chapman Hall/CRC Press LLC. ISBN 978-1-58488-360-9.

[22]van Otterlo, M.; Wiering, M. (2012). Reinforcement learning and markov decision processes. Reinforcement Learning. Adaptation, Learning, and Optimization. Vol. 12. pp. 3 42. doi:10.1007/978-3-642-27645-3$_1$.$ISBN978 - 3 - 642 - 27644 - 6$.

[23]Ron Kohavi; Foster Provost (1998). "Glossary of terms". Machine Learning. 30: 271 274.

[24]James, Gareth (2013). An Introduction to Statistical Learning: with Applications in R. Springer. p. 176.

[25]Ripley, B.D. (1996) Pattern Recognition and Neural Networks, Cambridge: Cambridge University Press, p. 354

[26]Larose, D. T.; Larose, C. D. (2014). Discovering knowledge in data : an introduction to data mining. Hoboken: Wiley.

[27]Ripley, B.D. (1996) Pattern Recognition and Neural Networks, Cambridge: Cambridge University Press, p. 354

Chapter 3

Unsupervised Learning

3.1 Clustering

3.1.1 K-Means Clustering

3.2 Anomaly Detection

3.3 Expectation Maximization (EM) algorithm

3.4 Reinforcement Learning

Notes

[1]Mitchell, Tom (1997). Machine Learning. New York: McGraw Hill. ISBN 0-07-042807-7.

[2]Friedman, Jerome H. (1998). "Data Mining and Statistics: What's the connection?". Comput-

ing Science and Statistics. 29 (1): 3 9.

[3]Zhou, Victor (2019-12-20). "Machine Learning for Beginners: An Introduction to Neural Networks". Medium. Retrieved 2021-08-15.

[4]Ethem Alpaydin (2020). Introduction to Machine Learning (Fourth ed.). MIT. pp. xix, 1 3, 13 18. ISBN 978-0262043793.

[5]Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of Research and Development. 3 (3): 210 229.

[6]R. Kohavi and F. Provost, "Glossary of terms," Machine Learning, vol. 30, no. 2 3, pp. 271 274, 1998.

[7]Gerovitch, Slava (9 April 2015). "How the Computer Got Its Revenge on the Soviet Union". Nautilus. Retrieved 19 September 2021.

[8]Duda, R., Hart P. Pattern Recognition and Scene Analysis, Wiley Interscience, 1973

[9]Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2. ISBN 978-0-07-042807-2.

[10]Sarle, Warren (1994). "Neural Networks and statistical models". CiteSeerX 10.1.1.27.699.

[11]Russell, Stuart; Norvig, Peter (2003) [1995]. Artificial Intelligence: A Modern Approach (2nd ed.). Prentice Hall. ISBN 978-0137903955.

[12]Langley, Pat (2011). "The changing science of machine learning". Machine Learning. 82 (3): 275 279. doi:10.1007/s10994-011-5242-y

[13]Alpaydin, Ethem (2010). Introduction to Machine Learning. MIT Press. p. 9. ISBN 978-0-262-01243-0.

[14]Bzdok, Danilo; Altman, Naomi; Krzywinski, Martin (2018). "Statistics versus Machine Learning". Nature Methods. 15 (4): 233 234.

[15]Michael I. Jordan (2014-09-10). "statistics and machine learning". reddit. Retrieved 2014-10-01.

[16]Gareth James; Daniela Witten; Trevor Hastie; Robert Tibshirani (2013). An Introduction to Statistical Learning. Springer. p. vii.

[17]Russell, Stuart J.; Norvig, Peter (2010). Artificial Intelligence: A Modern Approach (Third ed.). Prentice Hall. ISBN 9780136042594.

[18]Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2012). Foundations of Machine Learning. The MIT Press. ISBN 9780262018258.

[19]Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2. ISBN 978-0-07-042807-2.

[20]Alpaydin, Ethem (2010). Introduction to Machine Learning. MIT Press. p. 9. ISBN 978-0-262-01243-0.

[21]Jordan, Michael I.; Bishop, Christopher M. (2004). "Neural Networks". In Allen B. Tucker (ed.).

Computer Science Handbook, Second Edition (Section VII: Intelligent Systems). Boca Raton, Florida: Chapman Hall/CRC Press LLC. ISBN 978-1-58488-360-9.

[22]van Otterlo, M.; Wiering, M. (2012). Reinforcement learning and markov decision processes. Reinforcement Learning. Adaptation, Learning, and Optimization. Vol. 12. pp. 3 42. doi:10.1007/978-3-642-27645-3$_1$.$ISBN978 - 3 - 642 - 27644 - 6$.

[23]Ron Kohavi; Foster Provost (1998). "Glossary of terms". Machine Learning. 30: 271 274.

[24]James, Gareth (2013). An Introduction to Statistical Learning: with Applications in R. Springer. p. 176.

[25]Ripley, B.D. (1996) Pattern Recognition and Neural Networks, Cambridge: Cambridge University Press, p. 354

[26]Larose, D. T.; Larose, C. D. (2014). Discovering knowledge in data : an introduction to data mining. Hoboken: Wiley.

[27]Ripley, B.D. (1996) Pattern Recognition and Neural Networks, Cambridge: Cambridge University Press, p. 354

Chapter 4

Introduction to Deep Learning

4.1 Concept

4.2 Artificial Neural Networks

4.2.1 Basic Structure of ANN

4.2.2 Types of ANN

4.2.3 Definition of ANN

4.2.4 Training ANN

Notes

[1]Mitchell, Tom (1997). Machine Learning. New York: McGraw Hill. ISBN 0-07-042807-7.

[2]Friedman, Jerome H. (1998). "Data Mining and Statistics: What's the connection?". Computing Science and Statistics. 29 (1): 3 9.

[3]Zhou, Victor (2019-12-20). "Machine Learning for Beginners: An Introduction to Neural Networks". Medium. Retrieved 2021-08-15.

[4]Ethem Alpaydin (2020). Introduction to Machine Learning (Fourth ed.). MIT. pp. xix, 1 3, 13 18. ISBN 978-0262043793.

[5]Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of Research and Development. 3 (3): 210 229.

[6]R. Kohavi and F. Provost, "Glossary of terms," Machine Learning, vol. 30, no. 2 3, pp. 271 274, 1998.

[7]Gerovitch, Slava (9 April 2015). "How the Computer Got Its Revenge on the Soviet Union". Nautilus. Retrieved 19 September 2021.

[8]Duda, R., Hart P. Pattern Recognition and Scene Analysis, Wiley Interscience, 1973

[9]Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2. ISBN 978-0-07-042807-2.

[10]Sarle, Warren (1994). "Neural Networks and statistical models". CiteSeerX 10.1.1.27.699.

[11]Russell, Stuart; Norvig, Peter (2003) [1995]. Artificial Intelligence: A Modern Approach (2nd ed.). Prentice Hall. ISBN 978-0137903955.

[12]Langley, Pat (2011). "The changing science of machine learning". Machine Learning. 82 (3): 275 279. doi:10.1007/s10994-011-5242-y

[13]Alpaydin, Ethem (2010). Introduction to Machine Learning. MIT Press. p. 9. ISBN 978-0-262-01243-0.

[14]Bzdok, Danilo; Altman, Naomi; Krzywinski, Martin (2018). "Statistics versus Machine Learning". Nature Methods. 15 (4): 233 234.

[15]Michael I. Jordan (2014-09-10). "statistics and machine learning". reddit. Retrieved 2014-10-01.

[16]Gareth James; Daniela Witten; Trevor Hastie; Robert Tibshirani (2013). An Introduction to Statistical Learning. Springer. p. vii.

[17]Russell, Stuart J.; Norvig, Peter (2010). Artificial Intelligence: A Modern Approach (Third ed.). Prentice Hall. ISBN 9780136042594.

[18]Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2012). Foundations of Machine Learning. The MIT Press. ISBN 9780262018258.

[19]Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2. ISBN 978-0-07-042807-2.

[20]Alpaydin, Ethem (2010). Introduction to Machine Learning. MIT Press. p. 9. ISBN 978-0-262-01243-0.

[21]Jordan, Michael I.; Bishop, Christopher M. (2004). "Neural Networks". In Allen B. Tucker (ed.). Computer Science Handbook, Second Edition (Section VII: Intelligent Systems). Boca Raton, Florida:

Chapman Hall/CRC Press LLC. ISBN 978-1-58488-360-9.

[22]van Otterlo, M.; Wiering, M. (2012). Reinforcement learning and markov decision processes. Reinforcement Learning. Adaptation, Learning, and Optimization. Vol. 12. pp. 3 42. doi:10.1007/978-3-642-27645-3$_1$.$ISBN 978-3-642-27644-6$.

[23]Ron Kohavi; Foster Provost (1998). "Glossary of terms". Machine Learning. 30: 271 274.

[24]James, Gareth (2013). An Introduction to Statistical Learning: with Applications in R. Springer. p. 176.

[25]Ripley, B.D. (1996) Pattern Recognition and Neural Networks, Cambridge: Cambridge University Press, p. 354

[26]Larose, D. T.; Larose, C. D. (2014). Discovering knowledge in data : an introduction to data mining. Hoboken: Wiley.

[27]Ripley, B.D. (1996) Pattern Recognition and Neural Networks, Cambridge: Cambridge University Press, p. 354

Chapter 5

Applications of Machine Learning

5.1 Sales and Marketing

5.2 Financial Services

5.3 Social Media Analysis

5.4 Fraud Detection

Notes

[1]Mitchell, Tom (1997). Machine Learning. New York: McGraw Hill. ISBN 0-07-042807-7.

[2]Friedman, Jerome H. (1998). "Data Mining and Statistics: What's the connection?". Computing Science and Statistics. 29 (1): 3 9.

[3]Zhou, Victor (2019-12-20). "Machine Learning for Beginners: An Introduction to Neural Networks". Medium. Retrieved 2021-08-15.

[4]Ethem Alpaydin (2020). Introduction to Machine Learning

(Fourth ed.). MIT. pp. xix, 1 3, 13 18. ISBN 978-0262043793.

[5]Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of Research and Development. 3 (3): 210 229.

[6]R. Kohavi and F. Provost, "Glossary of terms," Machine Learning, vol. 30, no. 2 3, pp. 271 274, 1998.

[7]Gerovitch, Slava (9 April 2015). "How the Computer Got Its Revenge on the Soviet Union". Nautilus. Retrieved 19 September 2021.

[8]Duda, R., Hart P. Pattern Recognition and Scene Analysis, Wiley Interscience, 1973

[9]Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2. ISBN 978-0-07-042807-2.

[10]Sarle, Warren (1994). "Neural Networks and statistical models". CiteSeerX 10.1.1.27.699.

[11]Russell, Stuart; Norvig, Peter (2003) [1995]. Artificial Intelligence: A Modern Approach (2nd ed.). Prentice Hall. ISBN 978-0137903955.

[12]Langley, Pat (2011). "The changing science of machine learning". Machine Learning. 82 (3): 275 279. doi:10.1007/s10994-011-5242-y

[13]Alpaydin, Ethem (2010). Introduction to Machine Learning. MIT Press. p. 9. ISBN 978-0-262-01243-0.

[14]Bzdok, Danilo; Altman, Naomi; Krzywinski, Martin (2018). "Statistics versus Machine Learning". Nature Methods. 15 (4): 233 234.

[15]Michael I. Jordan (2014-09-10). "statistics and machine learning". reddit. Retrieved 2014-10-01.

[16]Gareth James; Daniela Witten; Trevor Hastie; Robert Tibshirani (2013). An Introduction to Statistical Learning. Springer. p. vii.

[17]Russell, Stuart J.; Norvig, Peter (2010). Artificial Intelligence: A Modern Approach (Third ed.). Prentice Hall. ISBN 9780136042594.

[18]Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2012). Foundations of Machine Learning. The MIT Press. ISBN 9780262018258.

[19]Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2. ISBN 978-0-07-042807-2.

[20]Alpaydin, Ethem (2010). Introduction to Machine Learning. MIT Press. p. 9. ISBN 978-0-262-01243-0.

[21]Jordan, Michael I.; Bishop, Christopher M. (2004). "Neural Networks". In Allen B. Tucker (ed.). Computer Science Handbook, Second Edition (Section VII: Intelligent Systems). Boca Raton, Florida: Chapman Hall/CRC Press LLC. ISBN 978-1-58488-360-9.

[22]van Otterlo, M.; Wiering, M. (2012). Reinforcement learning and markov decision processes. Reinforcement Learning. Adaptation, Learning, and Optimization. Vol.

12. pp. 3 42. doi:10.1007/978-3-642-27645-3$_1$.$ISBN978-3-642-27644-6$.

[23]Ron Kohavi; Foster Provost (1998). "Glossary of terms". Machine Learning. 30: 271 274.

[24]James, Gareth (2013). An Introduction to Statistical Learning: with Applications in R. Springer. p. 176.

[25]Ripley, B.D. (1996) Pattern Recognition and Neural Networks, Cambridge: Cambridge University Press, p. 354

[26]Larose, D. T.; Larose, C. D. (2014). Discovering knowledge in data : an introduction to data mining. Hoboken: Wiley.

[27]Ripley, B.D. (1996) Pattern Recognition and Neural Networks, Cambridge: Cambridge University Press, p. 354

Chapter 6

Appendix-1

6.1 Data simulations

```
import random

def createFactor(cats=['male', 'female'], n=10):
    cats = cats*n
    out = random.sample(cats, n)
    return out

def createVector(start=1, end=10, n=10, m= 0, Type=None):
    if Type=='linear':
        out = list(range(n))
    elif Type=='random':
        out = [round(random.uniform(1, n)*m, 0) for i in
                                        ↪ range(n)]
    else:
        out = [random.randint(1, n)*m for i in range(n)]
    return out
```

Testing

```
>>> from datasimulations import createFactor,
                              ↪ createVector
```

```
>>> import random
>>> createFactor(['yes', 'no'], 10)
['no', 'yes', 'yes', 'yes', 'no', 'yes', 'no', 'yes', 'no
                              ↪ ', 'no']
>>> createVector(1, 10, m=100, Type='random')
[220.0, 621.0, 373.0, 344.0, 638.0, 361.0, 292.0, 499.0,
                              ↪ 308.0, 951.0]
```

Creating data sets using *pandas* package.

```
age = createVector(1, 10, m=10, Type='random')
age = [round(i, 0) for i in age]
education = gender = createFactor(['primary', 'secondary'
                              ↪ , 'higher'], 10)
gender = createFactor(['yes', 'no'], 10)
target = createVector(1, 2)
df = pd.DataFrame({'age': age, 'gender': gender, '
                              ↪ education':education, '
                              ↪ target': target})
>>> df
    age gender  education  target
0  73.0     no    primary       2
1  64.0     no    primary       2
2  98.0    yes     higher       1
3  58.0    yes  secondary       2
4  14.0     no    primary       1
5  13.0     no    primary       1
6  51.0     no     higher       2
7  30.0    yes    primary       2
8  30.0     no  secondary       1
9  86.0    yes  secondary       1

>>> df['age'].mean()
51.7
```