Chapter 4 SYSTEM DESIGN



**4.3 SciKit-learn**

Scikit-learn is a [free software](https://en.wikipedia.org/wiki/Free_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programming language. It features various [classification,](https://en.wikipedia.org/wiki/Statistical_classification) [regression](https://en.wikipedia.org/wiki/Regression_analysis) and [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) algorithms including [support vector](https://en.wikipedia.org/wiki/Support_vector_machine) [machines,](https://en.wikipedia.org/wiki/Support_vector_machine) [random forests,](https://en.wikipedia.org/wiki/Random_forests) [gradient boosting,](https://en.wikipedia.org/wiki/Gradient_boosting) [k-means](https://en.wikipedia.org/wiki/K-means_clustering) and [DBSCAN,](https://en.wikipedia.org/wiki/DBSCAN) and is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy) and [SciPy.](https://en.wikipedia.org/wiki/SciPy)

The scikit-learn project started as scikits-.learn, a [Google Summer of Code](https://en.wikipedia.org/wiki/Google_Summer_of_Code) project by [David Cournapeau.](https://en.wikipedia.org/wiki/David_Cournapeau) Its name stems from the notion that it is a "SciKit" (SciPy Toolkit), a separately-developed and distributed third-party extension to SciPy. The original codebase was later rewritten by other developers. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel, all from [INRIA](https://en.wikipedia.org/wiki/INRIA) took leadership of the project and made the first public release on February the 1st 2010. Of the various scikits, scikit-learn as well as [scikit-image](https://en.wikipedia.org/wiki/Scikit-image) were described as "well-maintained and popular" in November 2012.

**4.4 Numpy and Pandas**

NumPy is a library for the [Python programming language,](https://en.wikipedia.org/wiki/Python_(programming_language)) adding support for large, multi-dimensional [arrays](https://en.wikipedia.org/wiki/Array_data_structure) and [matrices,](https://en.wikipedia.org/wiki/Matrix_(math)) along with a large collection of [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) [mathematical](https://en.wikipedia.org/wiki/Mathematics) [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) to operate on these arrays. NumPy enriches the programming language Python with powerful data structures, implementing multi-dimensional arrays and matrices. These data structures guarantee efficient calculations with matrices and arrays. The implementation is even aiming at huge matrices and arrays, better known under the heading of "big data".

The ancestor of NumPy, Numeric, was originally created by [Jim Hugunin](https://en.wikipedia.org/wiki/Jim_Hugunin) with contributions from several other developers. In 2005, [Travis Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant) created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is [open-source software](https://en.wikipedia.org/wiki/Open-source_software) and has many contributors. By using them in Python programming, they can be used with two simple commands:

*>>> import numpy*

The core functionality of NumPy is its "ndarray", for n-dimensional array, data structure. These arrays are [strided](https://en.wikipedia.org/wiki/Stride_of_an_array) views on memory. In contrast to Python's built-in list data structure (which, despite the name, is a [dynamic array)](https://en.wikipedia.org/wiki/Dynamic_array) these arrays are homogeneously typed: all elements of a single array must be of the same type.



|  |  |  |
| --- | --- | --- |
| Department of CSE, RNSIT | 2018-2019 | 13 |

Chapter 4 SYSTEM DESIGN



Pandas is a [software library](https://en.wikipedia.org/wiki/Software_library) written for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)) for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and [time series.](https://en.wikipedia.org/wiki/Time_series) It is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [three-clause BSD license.](https://en.wikipedia.org/wiki/3-clause_BSD_license) The name is derived from the term ["panel data",](https://en.wikipedia.org/wiki/Panel_data) an [econometrics](https://en.wikipedia.org/wiki/Econometrics) term for data sets that include observations over multiple time periods for the same individuals. Library features are:

* DataFrame object for data manipulation with integrated indexing.
* Tools for reading writing data between in-memory data structures different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of data sets.
* Label-based slicing, fancy indexing, and sub setting of large data sets.
* Data structure column insertion and deletion.
* Group by engine allowing split-apply-combine operations on data sets.
* Data set merging and joining.
* Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
* Time series-functionality: Date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.

**4.5 Matplotlib**

Matplotlib is a [plotting](https://en.wikipedia.org/wiki/Plotter) [library](https://en.wikipedia.org/wiki/Library_(computer_science)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programming language and its numerical mathematics extension [NumPy.](https://en.wikipedia.org/wiki/NumPy) It provides an [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [API](https://en.wikipedia.org/wiki/API) for embedding plots into applications using general-purpose [GUI toolkits](https://en.wikipedia.org/wiki/GUI_toolkit) like [Tkinter,](https://en.wikipedia.org/wiki/Tkinter) [wxPython,](https://en.wikipedia.org/wiki/WxPython) [Qt,](https://en.wikipedia.org/wiki/Qt_(software)) or GTK+.There is also a [procedural](https://en.wikipedia.org/wiki/Procedural_programming) "pylab" interface based on a [state machine](https://en.wikipedia.org/wiki/State_machine) (like [OpenGL),](https://en.wikipedia.org/wiki/OpenGL) designed to closely resemble that of [MATLAB,](https://en.wikipedia.org/wiki/MATLAB) though its use is discouraged. [SciPy](https://en.wikipedia.org/wiki/SciPy) makes use of matplotlib.

Matplotlib was originally written by [John D. Hunter,](https://en.wikipedia.org/wiki/John_D._Hunter) has an active development community, and is distributed under a [BSD-style license.](https://en.wikipedia.org/wiki/BSD_licenses) Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012, and further joined by Thomas Caswell.

As of 23 June 2017, matplotlib 2.0.x supports Python versions 2.7 through 3.6. Matplotlib 1.2 is the first version of matplotlib to support Python 3.x. Matplotlib 1.4 is the last version of matplotlib to support Python 2.6.



|  |  |  |
| --- | --- | --- |
| Department of CSE, RNSIT | 2018-2019 | 14 |

Chapter 4 SYSTEM DESIGN



**4.6 Training the Machine to Predict Result**

Data modelling is the process of estimating the underlying structure of a given dataset, with the goal of finding useful patterns (correlations, clusters, eigenvectors, etc.) and/or predicting properties of previously unseen instances (classification, regression, anomaly detection, etc.). A key part of this estimation process is continually evaluating how good a given model is. Depending on the task at hand, you will need to choose an appropriate accuracy/error measure (e.g. log-loss for classification, sum-of-squared-errors for regression, etc.) and an evaluation strategy (training-testing split, sequential vs. randomized cross-validation, etc.). Iterative learning algorithms often directly utilize resulting errors to tweak the model (e.g. back propagation for neural networks), so understanding these measures are very important even for just applying standard algorithms.

Standard implementations of Machine Learning algorithms are widely available through libraries/packages/APIs (e.g. scikit-learn, TensorFlow etc.), but applying them effectively involves choosing a suitable model (decision tree, nearest neighbour, neural net, support vector machine, ensemble of multiple models, etc.), a learning procedure to fit the data (linear regression, gradient descent, genetic algorithms, bagging, boosting, and other model-specific methods), as well as understanding how hyper parameters affect learning. You also need to be aware of the relative advantages and disadvantages of different approaches, and the numerous gotchas that can trip you (bias and variance, over fitting and under fitting, missing data, data leakage, etc.).

**4.6.1 Supervised Training**

In supervised training, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights which control the network. This process occurs over and over as the weights are continually tweaked. The set of data which enables the training is called the "training set." During the training of a network the same set of data is processed many times as the connection weights are ever refined. An artificial neural network configures itself with the general statistical trends of the data. Later, it continues to "learn" about other aspects of the data which may be spurious from a general viewpoint. When the system has been correctly trained, and no further learning is needed, the weights can, if desired, be "frozen."



|  |  |  |
| --- | --- | --- |
| Department of CSE, RNSIT | 2018-2019 | 15 |

Chapter 4 SYSTEM DESIGN



Networks also don't converge if there is not enough data to enable complete learning. Ideally, there should be enough data so that part of the data can be held back as a test. Many layered networks with multiple nodes are capable of memorizing data. To monitor the network to determine if the system is simply memorizing its data in some no significant way, supervised training needs to hold back a set of data to be used to test the system after it has undergone its training.

The designer then has to review the input and outputs, the number of layers, the number of elements per layer, the connections between the layers, the summation, transfer, and training functions, and even the initial weights themselves. Those changes required to create a successful network constitute a process wherein the "art" of neural networking occurs. Another part of the designer's creativity governs the rules of training. There are many laws (algorithms) used to implement the adaptive feedback required to adjust the weights during training. The most common technique is backward-error propagation, more commonly known as back-propagation.

**4.6.2 Unsupervised (Adaptive) Training**

The other type of training is called unsupervised training. In unsupervised training, the network is provided with inputs but not with desired outputs. The system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaption. At present time, the vast bulk of neural network work is in systems with supervised learning. Supervised learning is achieving results.

Learning from the unlabelled data to differentiating the given input data. All clustering algorithms comes under unsupervised learning algorithms. [Cluster analysis,](https://in.mathworks.com/discovery/cluster-analysis.html) which is used for exploratory data analysis to find hidden patterns or grouping in data. The clusters are modelled using a measure of similarity which is defined upon metrics such as Euclidean or probabilistic distance.

**4.7 Breast Cancer Wisconsin Dataset**

The [datasets](https://en.wikipedia.org/wiki/Data_set) are used for [machine-learning](https://en.wikipedia.org/wiki/Machine-learning) research and have been cited in [peer-reviewed](https://en.wikipedia.org/wiki/Peer_review) academic journals. Datasets are an integral part of the field of machine learning. Major advances in this field can result from advances in learning [algorithms](https://en.wikipedia.org/wiki/Algorithm) (such as [deep](https://en.wikipedia.org/wiki/Deep_learning) [learning),](https://en.wikipedia.org/wiki/Deep_learning) computer hardware, and, less-intuitively, the availability of high-quality training datasets. High-quality labeled training datasets for [supervised](https://en.wikipedia.org/wiki/Supervised_learning) and [semi-supervised](https://en.wikipedia.org/wiki/Semi-supervised_learning) machine

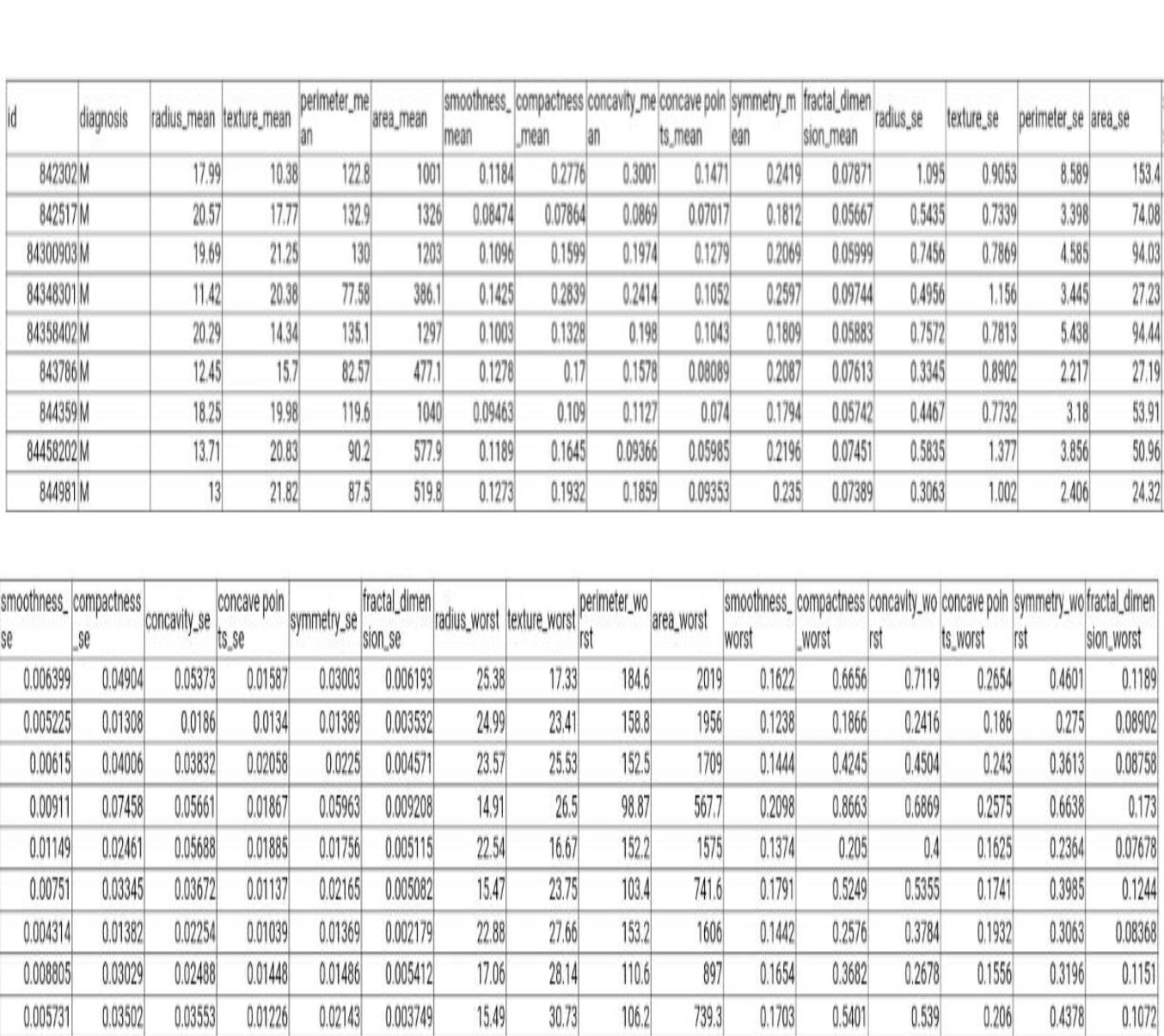


|  |  |  |
| --- | --- | --- |
| Department of CSE, RNSIT | 2018-2019 | 16 |

Chapter 4 SYSTEM DESIGN



learning algorithms are usually difficult and expensive to produce because of the large amount of time needed to label the data. Although they do not need to be labeled, high-quality datasets for [unsupervised](https://en.wikipedia.org/wiki/Unsupervised_learning) learning can also be difficult and costly to produce.



**Figure 4.1 Breast Cancer Wisconsin Dataset**

The data set consists of 570 cancer records where each record consists of 33 important features which are essential for the prediction of the stage of the breast cancer, whether it is malignant or benign. In the process of preparing a training data set and a testing data set, the greatest problem is how to find the most appropriate way to divide the data set into training data set and testing data set. In some cases, by using sampling theory and estimation theory, we can separate the whole data set into training data set and testing data set. However, sometimes, the method would be changed. The attributes and the property of the data set would be different in various machine learning objects. Thus, in this kind of situation, in order to



|  |  |  |
| --- | --- | --- |
| Department of CSE, RNSIT | 2018-2019 | 17 |

Chapter 4 SYSTEM DESIGN



achieve a better result of machine learning, the data set will be separated according to the property of attributes of the data set.

**4.8 Logistic Regression Algorithm**

Here to train machine for predicting the result the Logistic Regression Algorithm is used. It is a [non-parametric](https://en.wikipedia.org/wiki/Non-parametric_statistics) method used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression.](https://en.wikipedia.org/wiki/Regression_analysis)

Logistic regression can be binomial, ordinal or multinomial. Binomial or binary logistic regression deals with situations in which the observed outcome for a [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable) can have only two possible types, "0" and "1" (which may represent, for example, "dead" vs. "alive" or "win" vs. "loss"). [Multinomial logistic regression](https://en.wikipedia.org/wiki/Multinomial_logit) deals with situations where the outcome can have three or more possible types (e.g., "disease A" vs. "disease B" vs. "disease C") that are not ordered. [Ordinal logistic regression](https://en.wikipedia.org/wiki/Ordinal_logistic_regression) deals with dependent variables that are ordered.

In binary logistic regression, the outcome is usually coded as "0" or "1", as this leads to the most straightforward interpretation. If a particular observed outcome for the dependent variable is the noteworthy possible outcome (referred to as a "success" or a "case") it is usually coded as "1" and the contrary outcome (referred to as a "failure" or a "noncase") as "0". Binary logistic regression is used to predict the [odds](https://en.wikipedia.org/wiki/Odds) of being a case based on the values of the [independent variables](https://en.wikipedia.org/wiki/Independent_variable) (predictors). The odds are defined as the probability that a particular outcome is a case divided by the probability that it is a noncase.

Like other forms of [regression analysis,](https://en.wikipedia.org/wiki/Regression_analysis) logistic regression makes use of one or more predictor variables that may be either continuous or categorical. Unlike ordinary linear regression, however, logistic regression is used for predicting dependent variables that take [membership in one of a limited number of categories](https://en.wikipedia.org/wiki/Categorical_variable) (treating the dependent variable in the binomial case as the outcome of a [Bernoulli trial)](https://en.wikipedia.org/wiki/Bernoulli_trial) rather than a continuous outcome. Given this difference, the assumptions of linear regression are violated. In particular, the residuals cannot be normally distributed. In addition, linear regression may make nonsensical predictions for a binary dependent variable. What is needed is a way to convert a binary variable into a continuous one that can take on any real value (negative or positive). To do that, binomial logistic regression first calculates the [odds](https://en.wikipedia.org/wiki/Odds) of the event happening for different levels of each independent variable, and then takes its [logarithm](https://en.wikipedia.org/wiki/Logarithm) to create a continuous criterion as a transformed version of the dependent variable. The logarithm of the odds is the [logit](https://en.wikipedia.org/wiki/Logit) of the probability, the logit being defined as follows:



|  |  |  |
| --- | --- | --- |
| Department of CSE, RNSIT | 2018-2019 | 18 |

Chapter 4 SYSTEM DESIGN



logit p = ln (p/(1-p) for 0<p<1.

Although the dependent variable in logistic regression is Bernoulli, the logit is on an unrestricted scale. The logit function is the [link function](https://en.wikipedia.org/wiki/Link_function) in this kind of generalized linear model, i.e. we have

Logit E(Y)=a +bx

where Y is the Bernoulli-distributed response variable and x is the predictor variable.

The logit of the probability of success is then fitted to the predictors. The predicted value of the logit is converted back into predicted odds via the inverse of the natural logarithm, namely the [exponential function.](https://en.wikipedia.org/wiki/Exponential_function) Thus, although the observed dependent variable in binary logistic regression is a zero-or-one variable, the logistic regression estimates the odds, as a continuous variable, that the dependent variable is a success (a case). In some applications the odds are all that is needed. In others, a specific yes-or-no prediction is needed for whether the dependent variable is or is not a case; this categorical prediction can be based on the computed odds of a success, with predicted odds above some chosen cutoff value being translated into a prediction of a success.



|  |  |  |
| --- | --- | --- |
| Department of CSE, RNSIT | 2018-2019 | 19 |