IPASJ International Journal of Computer Science (IIJCS)
Web Site: http://www.ipasj.org/IIJCS/IIJCS.htm
A Publisher for Research Motivation ....
Email:editoriijcs@ipasj.org
Volume 5, Issue 8, August 2017
ISSN 2321-5992

Identification Of Iris Flower Species Using Machine Learning

Shashidhar T Halakatti[1], **Shambulinga T Halakatti**[2]

[1]Department. of Computer Science Engineering, Rural Engineering College ,Hulkoti – 582205
Dist : Gadag State : Karnataka Country : India

[2]Department of Electronics & Communication Engineering BVVS Polytechnic, Bagalkot – 587101
Dist : Bagalkot State : Karnataka Country: India

ABSTRACT

In Machine Learning, we are using semi-automated extraction of knowledge of data for identifying IRIS flower species. Classification is a supervised learning in which the response is categorical that is its values are in finite unordered set. To simply the problem of classification, scikit learn tools has been used. This paper focuses on IRIS flower classification using Machine Learning with scikit tools. Here the problem concerns the identification of IRIS flower species on the basis of flowers attribute measurements. Classification of IRIS data set would be discovering patterns from examining petal and sepal size of the IRIS flower and how the prediction was made from analyzing the pattern to from the class of IRIS flower. In this paper we train the machine learning model with data and when unseen data is discovered the predictive model predicts the species using what it has been learnt from the trained data.

Keywords: Classification, Logistic Regression, K Nearest Neighbour, Machine Learning.

1. INTRODUCTION

The Machine Learning is the subfield of computer science, according to Arthur Samuel in 1959 told “computers are having the ability to learn without being explicitly programmed”. Evolved from the study of pattern recognition and computational learning theory in artificial intelligence machine learning explores the study and construction of algorithms that can learn from and make predictions on data such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions, through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicitly algorithms with good performance is difficult or unfeasible; example applications include email filtering, detection of network intruders, learning to rank and computer vision.

Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data. It is a research field at the intersection of statistics, artificial intelligence and computer science and is also known as predictive analytics or statistical learning. There are two main categories of Machine learning. They are Supervised and Unsupervised learning and here in this, the paper focuses on supervised learning. Supervised learning is a task of inferring a function from labeled training data. The training data consists of set of training examples. In supervised learning, each example is a pair of an input object and desired output value. A supervised learning algorithm analyze the training data and produces an inferred function, which can be used for mapping new examples. Supervised learning problems can be further grouped into regression and classification problems. Classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”. Regression problem is when the output variable is a real value, such as “dollars” or “weight”.

In this paper a novel method for Identification of Iris flower species is presented. It works in two phases, namely training and testing. During training the training dataset are loaded into Machine Learning Model and Labels are assigned. Further the predictive model, predicts to which species the Iris flower belongs to. Hence, the expected Iris species is labeled.

This paper focuses on IRIS flower classification using Machine Learning with scikit tools. The problem statement concerns the identification of IRIS flower species on the basic of flower attribute measurements. Classification of IRIS data set would be discovering patterns from examining petal and sepal size of the IRIS flower and how the prediction was made from analyzing the pattern to form the class of IRIS flower. In this paper we train the Machine Learning Model with data and when unseen data is discovered the predictive model predicts the species using what it has learn from trained data.

2. RELATED WORK

Many methods have been presented for Identification of Iris Flower Species. Every method employs different strategy. Review of some prominent solutions is presented.

The methodology for Iris Flower Species System is described [1]. In this work, IRIS flower classification using Neural Network. The problem concerns the identification of IRIS flower species on the basis of flower attribute measurements. Classification of IRIS data set would be discovering patterns from examining petal and sepal size of the IRIS flower and how the prediction was made from analyzing the pattern to form the class of IRIS flower. By using this pattern and classification, in future upcoming years the unknown data can be predicted more precisely. Artificial neural networks have been successfully applied to problems in pattern classification, function approximations, optimization, and associative memories. In this work, Multilayer feed-forward networks are trained using back propagation learning algorithm.

The model for Iris Flower Species System is described [2]. Existing iris flower dataset is preloaded in MATLAB and is used for clustering into three different species. The dataset is clustered using the k-means algorithm and neural network clustering tool in MATLAB. Neural network clustering tool is mainly used for clustering large data set without any supervision. It is also used for pattern recognition, feature extraction, vector quantization, image segmentation, function approximation, and data mining. Results/Findings: The results include the clustered iris dataset into three species without any supervision.

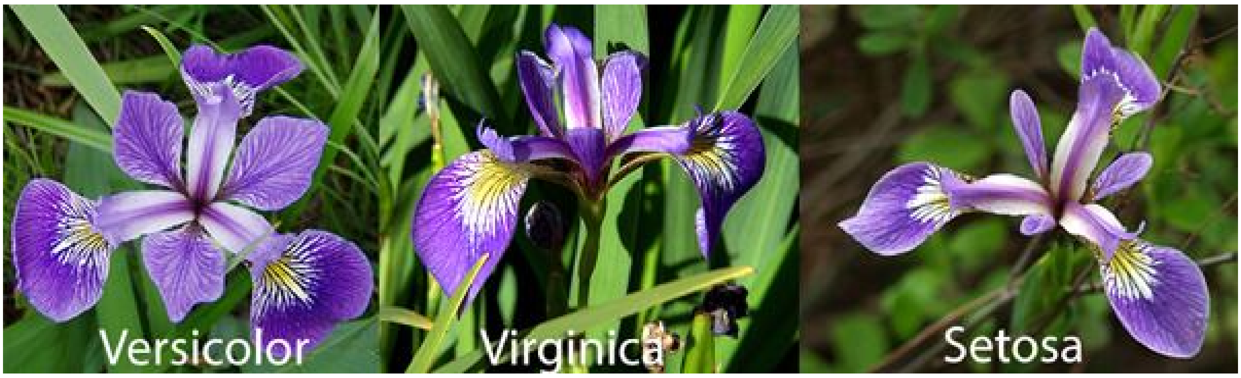
The model for Iris Flower Species System is described [3]. The proposed method is applied on Iris data sets and classifies the dataset into four classes. In this case, the network could select the good features and extract a small but adequate set of rules for the classification task. For Class one data set we obtained zero misclassification on test sets and for all other data sets the results obtained are comparable to the results reported in the literature.

3. MOTIVATIONAL WORK

3.1 Motivation for the Work

It is observed from the literature survey that the existing algorithms face several difficulties like the computational power is increases when run Deep Learning on latest computation, requires a large amount of data, is extremely computationally expensive to train, they do not have explanatory power that is they may extract the best signals to accurately classify and cluster data, but cannot get how they reached a certain conclusion. Neural Networks cannot be retrained that is it is impossible to add data later. To address these problems the current work is taken up to develop a new technique for Identification of Iris Flower Species using Machine Learning.

3.2 Iris Flower Species



The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper. The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis. It is sometimes called Anderson’s Iris data set because Edgar Anderson collected the data to quantify the morphologic variation of Iris Flower of three related species. Two of the three species were collected in Gaspe Peninsula all from the same pasture, and picked on the same day and measured at the same time by the same person with same apparatus.

The data set consists of 50 samples from each of three species of Iris that is 1) Iris Setosa 2) Iris Virginica 3) Iris Versicolor. Four features were measured from each sample. They are 1) Sepal Length 2) Sepal Width 3) Petal Length 4) Petal Width. All these four parameters are measured in Centimeters. Based on the combination of these four features, the species among three can be predicted.

3.3 Summary Statistics

	Min	Max	Mean	SD	Class Correlation
sepal length:	4.3	7.9	5.84	0.83	0.7826
sepal width:	2.0	4.4	3.05	0.43	-0.4194
petal length:	1.0	6.9	3.76	1.76	0.9490 (high!)
petal width:	0.1	2.5	1.20	0.76	0.9565 (high!)

3.4 Problem Definition

To design and implement the Identification of Iris Flower species using machine learning using Python and the tool Scikit-Learn.

3.5 Work Carried Out

- Data collection: Various datasets of Iris Flower are collected. There are totally 150 datasets belonging to three different species of Iris Flower that is Setosa, Versicolor and Virginca.
- Literature survey: Studied various papers related to proposed work.
- Algorithms developed

1. A K-Nearest Neighbor Algorithm to predict the species of Iris Flower.
2. A Logistic Regression Algorithm to predict the species of Iris Flower.

4. Block Diagram of the Proposed Work

The proposed method comprises of sub-phases that is Loading and Modeling as schematic diagram of the proposed model is given in figure 1.

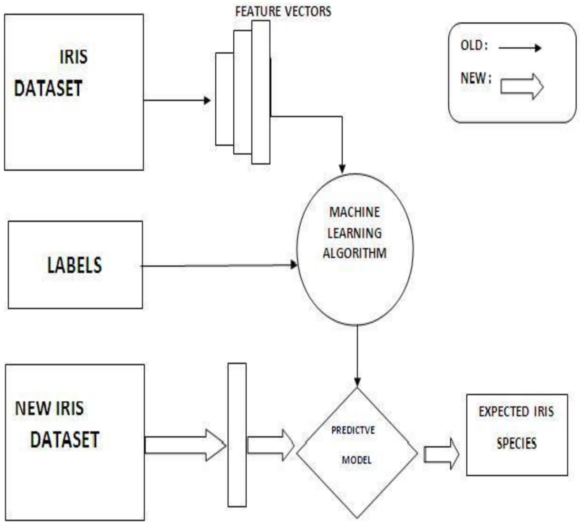


Figure1: Block Diagram of Machine Learning Model

The detailed description of each processing step is presented in the following sub sections.

4.1 Loading

Various datasets of Iris Flower are collected. There are totally 150 datasets belonging to three different species of Iris Flower that is Setosa, Versicolor and Virginca. The collected Iris Datasets are loaded into the Machine Learning Model. Scikit-learn comes with a few standard datasets, for instance the Iris Dataset for Classification. The load_iris function is imported from Scikit-learn. The load_iris function is run and save the return value in an object called “iris”. The iris object is of type “sklearn.datasets.base.bunch”, here bunch is a special object in scikit-learn to store datasets and attributes. The few attributes of iris object are data, feature names, target, target names etc. The iris.data is of type numpy.ndarray and is stored in a Feature Matrix say “X”. Here X is a two dimensional array, the two dimensions of it are number of observations and number of features. The iris.target is of type numpy.ndarray and is stored in a Response Vector say “y”. Here y is a one dimensional array, the one dimension of it is number of observations.
In Scikit-learn, each Row is an observation that is sample, example, instance, record and each Column is a feature that is predictor, attribute, independent variable, input, regressor, covariate.
Four key requirements for working with data in Scikit-learn,
Features and Response are separate objects.
Features and Response should be numeric.
Features and Response should be NumPy arrays.
Features and Response should have specific shapes.
The shapes of X and y here is (150 , 4) and (150,) respectively that is 150 observations and 4 features. The Target Names for Iris dataset are ['setosa','versicolor','virginica'] and the Feature Names for Iris dataset are ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)'].

4.2 Modeling

Scikit-learn has four step Modeling Pattern.

Step 1: Import the class which is needed from Scikit-learn.
In first case, we import KNeighborsClassifier from Sklearn Neighbors. Sklearn Neighbors provides functionality for supervised neighbors-based learning methods. The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these.
In second case, we import Logistic Regression from Sklearn Linear Model module. The module implements generalized linear models. It includes Ridge regression, Bayesian Regression, Lasso and Elastic Net estimators computed with Least Angle Regression and coordinate descent. It also implements Stochastic Gradient Descent related algorithms.

Step 2: Here we Instantiate the Estimator.
Scikit-learn refers its model as Estimator. A estimator is an object that fits a model based on some training data and is capable of inferring some properties on new data. It can be, for instance, a classifier or a regressor. Instantiation concerns the creation of an object that is Instantiate the object “Estimator”. Here in first case, Instantiate the Estimator means make instance of KNeighborsClassifier Class. The object here has various parameters that is KNeighborsClassifier(algorithm='auto',leaf_size=30, metric='minkowski', metric_params=None, n_jobs=1, n_neighbors=5, p=2, weights='uniform'). Here in first case, Instantiate the Estimator means make instance of LogisticRegression Class. The object here has various parameters that is LogisticRegression(C=1.0,class_weight=None,dual=False,fit_intercept=True,intercept_scaling=1,max_iter=100,multi_class='ovr', n_jobs=1, penalty='l2', random_state=None, solver='liblinear', tol=0.0001,verbose=0, warm_start=False)
Now, there are Objects that knows how to do K-Nearest Neighbor and Logistic Regression and waiting for user to give data. The name of the Estimator object can be anything, we can tend to choose the name that reflex the model it represents, “est” short of estimator or “clf ” short of classifier. The Tuning Parameter that is Hyper Parameter can be specified at this step. For example, n_neighbors is a tuning parameter. All the other parameters which are not specified here are set to their default values. By printing the Estimator object we can get all the parameters and its values.

Step 3: Fit the Model with Data
This is the model training step. Here the Model learns the relationship between the features X and response y. Here fit method is used on the object of type KNeighborsClassifier Class and LogisticRegression Class. The fit method takes two parameters that is the feature matrix X and response vector y. The model is underfitting or over fitting the training data. The model is underfitting the training data when the model performs poorly on the training data. This is because the model is unable to capture the relationship between the input examples (often called X) and the target values (often called Y). The model is overfitting your training data when you see that the model performs well on the training data but does not perform well on the evaluation data. This is because the model is memorizing the data it has seen and is unable to generalize to unseen examples.

Step 4: Predict the response for a new observation.
In this step, the response is predicted for a new observation. Here a new observation means “out-of-sample” data. Here, its inputing the measurements for unknown iris and asking the fitted model to predict the iris species based on what it has learnt in previous step. The predict method is used on the KNeighbors Classifier Class object and Logistic Regression Class object and pass the features of Unknown iris as a Python list. Actually, expects numpy array but it still works with list since numpy automatically converts it to an array of appropriate shape. The predict method returns a object of type numpy array with predicted response value. The model can predict the species for multiple observations at once.

5. IMPLEMENTATION OF ALGORITHMS

5.1 K-Nearest Neighbors Algorithm

The k-Nearest Neighbors algorithm (or kNN for short) is an easy algorithm to understand and to implement, and a powerful tool to have at your disposal. The implementation will be specific for classification problems and will be demonstrated using the Iris flowers classification problem.

5.1.1 What is k-Nearest Neighbors

The model for kNN is the entire training dataset. When a prediction is required for a unseen data instance, the kNN algorithm will search through the training dataset for the k-most similar instances. The prediction attribute of the most similar instances is summarized and returned as the prediction for the unseen instance. The similarity measure is dependent on the type of data. For real-valued data, the Euclidean distance can be used. Other types of data such as categorical or binary data, Hamming distance can be used. In the case of regression problems, the average of the predicted attribute may be returned. In the case of classification, the most prevalent class may be returned.

5.1.2 How does k-Nearest Neighbors Work

The kNN algorithm is belongs to the family of instance-based, competitive learning and lazy learning algorithms. Instance-based algorithms are those algorithms that model the problem using data instances (or rows) in order to make predictive decisions. The kNN algorithm is an extreme form of instance-based methods because all training observations are retained as part of the model.

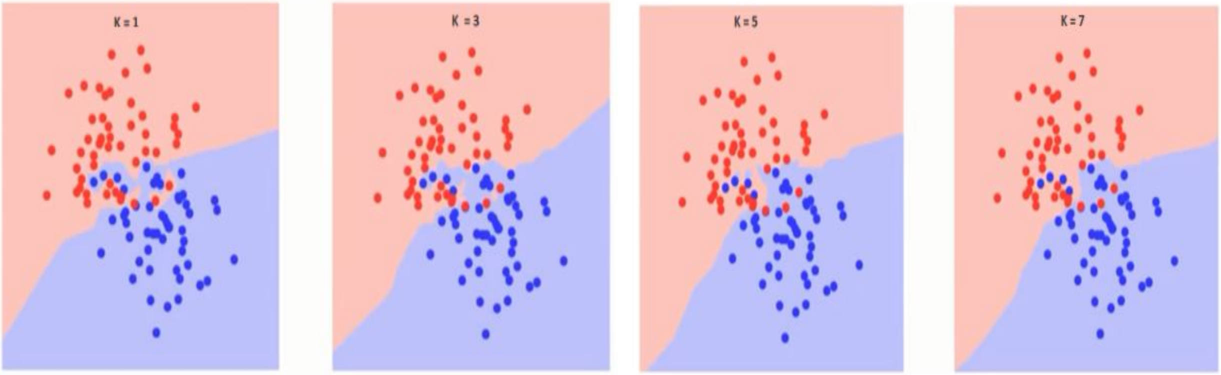
It is a competitive learning algorithm, because it internally uses competition between model elements (data instances) in order to make a predictive decision. The objective similarity measure between data instances causes each data instance to compete to “win” or be most similar to a given unseen data instance and contribute to a prediction.

Lazy learning refers to the fact that the algorithm does not build a model until the time that a prediction is required. It is lazy because it only does work at the last second. This has the benefit of only including data relevant to the unseen data, called a localized model. A disadvantage is that it can be computationally expensive to repeat the same or similar searches over larger training datasets.

Finally, kNN is powerful because it does not assume anything about the data, other than a distance measure can be calculated consistently between any two instances. As such, it is called non-parametric or non-linear as it does not assume a functional form.

5.1.3 How do we choose the factor K

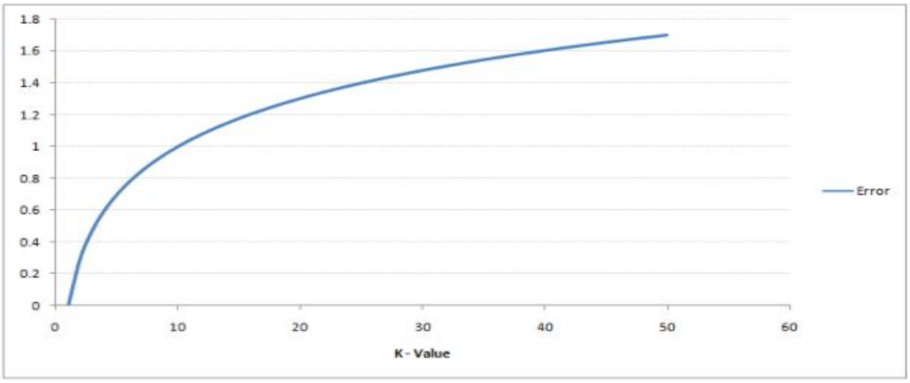
First let us try to understand what exactly does K influence in the algorithm. If we see the last example, given that all the 6 training observation remain constant, with a given K value we can make boundaries of each class. These boundaries will segregate RC from GS. The same way, let's try to see the effect of value "K" on the class boundaries. Following are the different boundaries separating the two classes with different values of K.



If you watch carefully, you can see that the boundary becomes smoother with increasing value of K. With K increasing to infinity it finally becomes all blue or all red depending on the total majority.

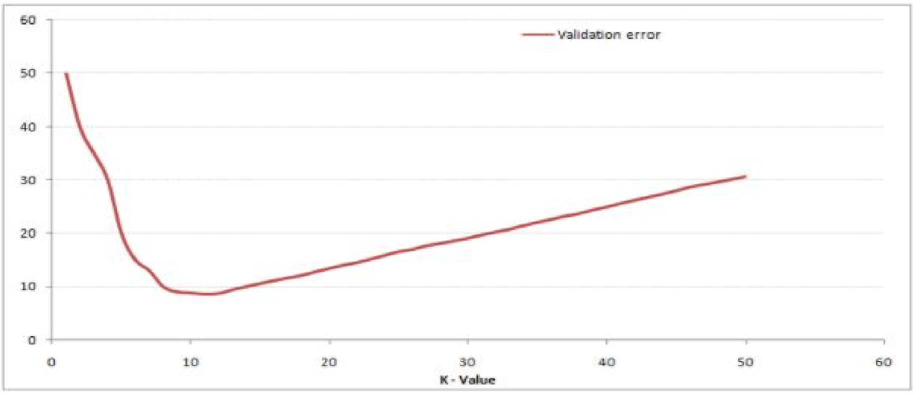
The training error rate and the validation error rate are two parameters we need to access on different K-value.

Following is the curve for the training error rate with varying value of K :



As you can see, the error rate at K=1 is always zero for the training sample. This is because the closest point to any training data point is itself. Hence the prediction is always accurate with K=1. If validation error curve would have been similar, our choice of K would have been 1.

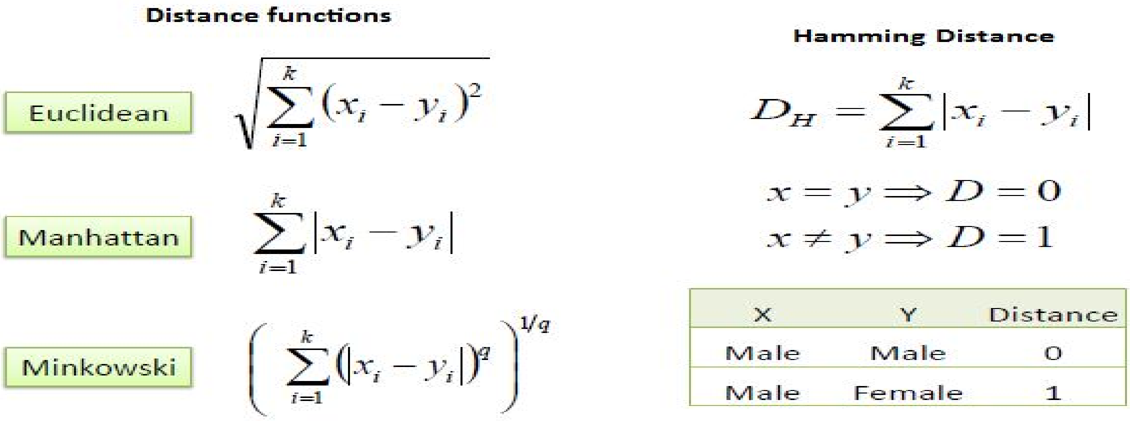
Following is the validation error curve with varying value of K:



This makes the story more clear. At K=1, we were overfitting the boundaries. Hence, error rate initially decreases and reaches a minima. After the minima point, it then increase with increasing K. To get the optimal value of K, you can segregate the training and validation from the initial dataset. Now plot the validation error curve to get the optimal value of K. This value of K should be used for all predictions.

5.2 Mathematical Formula

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor. It should also be noted that all three distance measures are only valid for continuous variables. In the instance of categorical variables the Hamming distance must be used. It also brings up the issue of standardization of the numerical variables between 0 and 1 when there is a mixture of numerical and categorical variables in the dataset.



Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

Hamming Distance

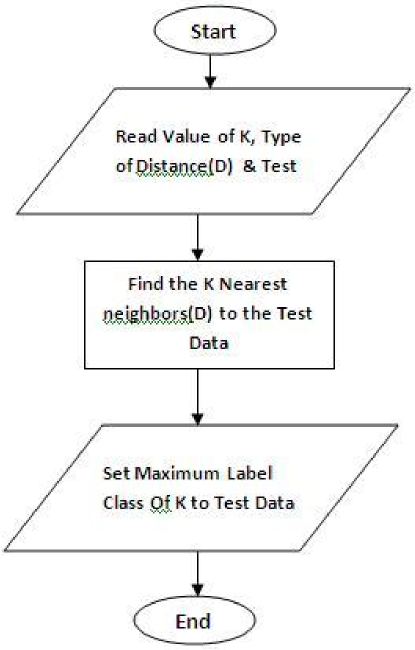
$$D_H = \sum_{i=1}^{k}|x_i - y_i|$$

$$x = y \Rightarrow D = 0$$
$$x \neq y \Rightarrow D = 1$$

X	Y	Distance
Male	Male	0
Male	Female	1

Choosing the optimal value for K is best done by first inspecting the data. In general, a large K value is more precise as it reduces the overall noise but there is no guarantee. Cross-validation is another way to retrospectively determine a good K value by using an independent dataset to validate the K value. Historically, the optimal K for most datasets has been between 3-10. That produces much better results than 1NN.

5.2.1 Flow chart of k-Nearest Neighbors


Start
Read Value of K, Type of Distance(D) & Test
Find the K Nearest neighbors(D) to the Test Data
Set Maximum Label Class Of K to Test Data
End

6.LOGISTIC REGRESSION ALGORITHM

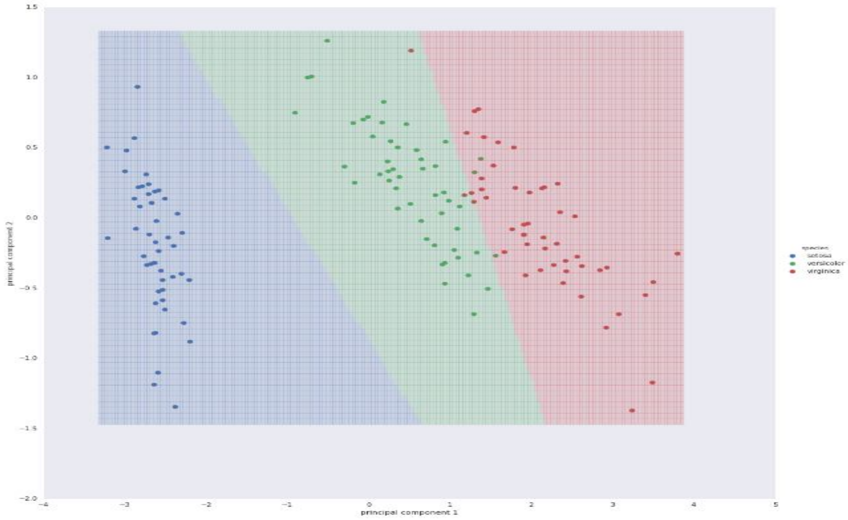
Logistic Regression is a type of regression that predicts the probability of occurrence of an event by fitting data to a logit function (logistic function). Like many forms of regression analysis, it makes use of several predictor variables that may be either numerical or categorical. For instance, the probability that a person has a heart attack within a specified time period might be predicted from knowledge of the person's age, sex and body mass index. This regression is quite used in several scenarios such as prediction of customer's propensity to purchase a product or cease a subscription in marketing applications and many others.

6.1 What is Logistic Regression?

Logistic Regression, also known as Logit Regression or Logit Model, is a mathematical model used in statistics to estimate (guess) the probability of an event occurring having been given some previous data. Logistic Regression works with binary data, where either the event happens (1) or the event does not happen (0). So given some feature x it tries to find out whether some event y happens or not. So y can either be 0 or 1. In the case where the event happens, y is given the value 1. If the event does not happen, then y is given the value of 0. For example, if y represents whether a sports teams wins a match, then y will be 1 if they win the match or y will be 0 if they do not. This is known as Binomial Logistic Regression. There is also another form of Logistic Regression which uses multiple values for the variable y. This form of Logistic Regression is known as Multinomial Logistic Regression.

6.2 How does logistic Regression work?

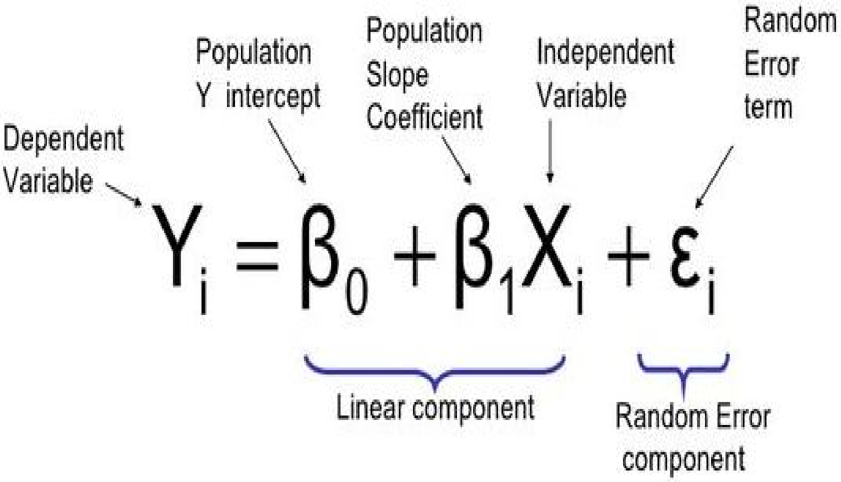
Logistic Regression uses the logistic function to find a model that fits with the data points. The function gives a 'S' shaped curve to model the data. The curve is restricted between 0 and 1, so it is easy to apply when y is binary. Logistic Regression can then model events better than linear regression, as it shows the probability for y being 1 for a given x value. Logistic Regression is used in statistics and machine learning to predict values of an input from previous test data.



Iris plot - logistic regression

A mesh when drawn over the plot shows the three classes of the logistic regression. Supervised learning consists in learning the link between two datasets: the observed data X and an external variable y that we are trying to predict, usually called “target” or “labels”. Most often, y is a 1D array of length n_samples. All supervised estimators in scikit-learn implement a fit(X, y) method to fit the model and a predict(X) method that, given unlabeled observations X, returns the predicted labels y.

7.MATHEMATICAL MODEL



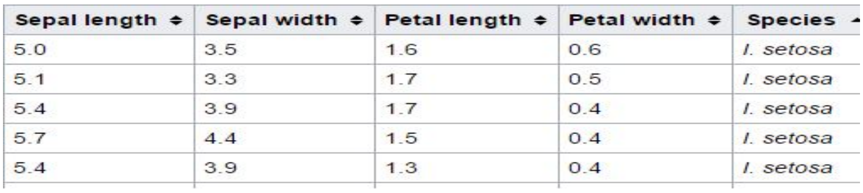
$$Y_i = \underbrace{\beta_0 + \beta_1 X_i}_{\text{Linear component}} + \underbrace{\varepsilon_i}_{\text{Random Error component}}$$

8.EXPERIMENTAL RESULTS AND DISCUSSION

Here are samples of datasets.

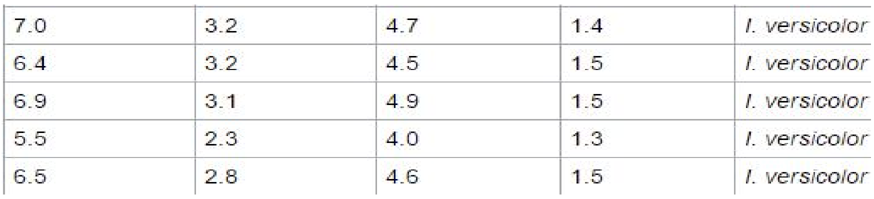
8.1 Illustration of Sample Iris Dataset

8.1.1 Sample datasets of Iris Setosa



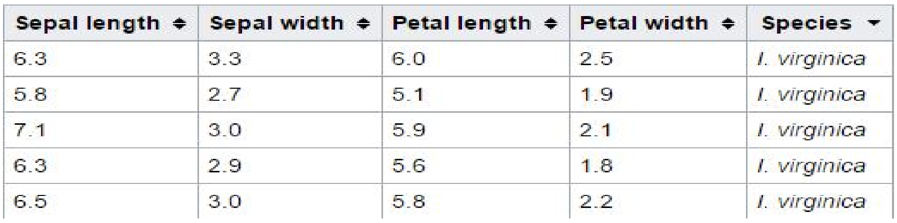
Sepal length	Sepal width	Petal length	Petal width	Species
5.0	3.5	1.6	0.6	*I. setosa*
5.1	3.3	1.7	0.5	*I. setosa*
5.4	3.9	1.7	0.4	*I. setosa*
5.7	4.4	1.5	0.4	*I. setosa*
5.4	3.9	1.3	0.4	*I. setosa*

8.1.2 Sample datasets of Iris Versicolor



7.0	3.2	4.7	1.4	*I. versicolor*
6.4	3.2	4.5	1.5	*I. versicolor*
6.9	3.1	4.9	1.5	*I. versicolor*
5.5	2.3	4.0	1.3	*I. versicolor*
6.5	2.8	4.6	1.5	*I. versicolor*

8.1.3 Sample datasets of Iris Virginica



Sepal length	Sepal width	Petal length	Petal width	Species
6.3	3.3	6.0	2.5	*I. virginica*
5.8	2.7	5.1	1.9	*I. virginica*
7.1	3.0	5.9	2.1	*I. virginica*
6.3	2.9	5.6	1.8	*I. virginica*
6.5	3.0	5.8	2.2	*I. virginica*

9.PERFORMANCE OF SYSTEM WITH VARIOUS IMAGES OF DATASET

Identification species for testing samples are given Table 8.1.1 and Table 8.1.2 and Table 8.1.3

Table 9.1 Identification of species of various sample data with parameters.

ALGORITHM APPLIED	PARAMETERS	SAMPLE DATA	PREDICTED IRIS SPECIES
K-NN	N_NEIGHBORS = 1	(3, 5, 4, 2)	ARRAY ([2])
K-NN	N_NEIGHBORS = 5	(3, 5, 4, 2)	ARRAY ([1])

Table 9.2 Identification of species of various sample data

ALGORITHM APPLIED	SAMPLE DATA	PREDICTED IRIS SPECIES
Logistic Regression	(5, 3, 1, 0)	ARRAY ([0])
Logistic Regression	(6, 2, 4, 1)	ARRAY ([1])

Identification accuracy for testing samples are given Table 9.3.

Table 9.3 Identification of species of various sample data

ALGORITHM APPLIED	ACCURACY
K-Nearest Neighbor (N_NEIGHBORS =5)	96.666667%
K-Nearest Neighbor (N_NEIGHBORS =1)	100.00%
Logistic Regression	96.00%
Logistic Regression (TRAIN AND SPLIT METHOD)	95.00%
K-Nearest Neighbor(TRAIN AND SPLIT METHOD AND N_NEIGHBORS=1)	95.00%
K-Nearest Neighbor(TRAIN AND SPLIT METHOD AND N_NEIGHBORS=5)	96.666667%

7.CONCLUSION

The primary goal of supervised learning is to build a model that “generalizes”. Here in this project we make predictions on unseen data which is the data not used to train the model hence the machine learning model built should accurately predicts the species of future flowers rather than accurately predicting the label of already trained data.

References

[1] Diptam Dutta, Argha Roy, Kaustav Choudhury, “Training Aritificial Neural Network Using Particle Swarm Optimization Algorithm”, International Journal on Computer Science And Engineering(IJCSE), Volume 3, Issue 3, March 2013.

[2] Poojitha V, Shilpi Jain, “A Collecation of IRIS Flower Using Neural Network CLusterimg tool in MATLAB”, International Journal on Computer Science And Engineering(IJCSE).

[3] Vaishali Arya, R K Rathy, “An Efficient Neura-Fuzzy Approach For Classification of Dataset”, International Conference on Reliability, Optimization and Information Technology, Feb 2014.

AUTHORS

Shashidhar T Halakatti received the MSc(Information Technology) 2006 from KSOU, Mysore and M.Tech(Computer Cognition Technology) 2009 from University of Mysore. Since from 2009 working as an Assistant Professor for the Department of Computer Science at Rural Engineering College, Hulkoti. Area of Specialization's: Computer Vision, Image Processing Machine Learning, Pattern Recognition.

Shambulinga T.Halakatti received the M.Tech(CS&E) degree in Computer Science and Engineering 2009 from Vishweshwaraya Technological University, Belgaum and working as a Lecturer in the Department of Electronics & Communication Engineering from 1991 in B.V.V.S.Polytechnic(Autonomous), Bagalkot and as Head of Instrumentation Technology from 2007.

Area of Specialization's: Digital communication, MEMs, Neural Networks, Machine Learning.