Chapter

Analytical Statistics Techniques of Classification and Regression in Machine Learning

Pramod Kumar, Sameer Ambekar, Manish Kumar and Subarna Roy

Abstract

This chapter aims to introduce the common methods and practices of statistical machine learning techniques. It contains the development of algorithms, applications of algorithms and also the ways by which they learn from the observed data by building models. In turn, these models can be used to predict. Although one assumes that machine learning and statistics are not quite related to each other, it is evident that machine learning and statistics go hand in hand. We observe how the methods used in statistics such as linear regression and classification are made use of in machine learning. We also take a look at the implementation techniques of classification and regression techniques. Although machine learning provides standard libraries to implement tons of algorithms, we take a look on how to tune the algorithms and what parameters of the algorithm or the features of the algorithm affect the performance of the algorithm based on the statistical methods.

Keywords: machine learning, statistics, classification, regression, algorithms

1. Introduction

Stating that statistical methods are useful in machine learning is analogous to saying that wood working methods are helpful for a carpenter. Statistics is the foundation of machine learning. However not all machine learning methods have been said to have derived from statistics. To begin with let us take a look at what statistics and machine learning means.

Statistics is extensively used in areas of science and finance and in the industry. Statistics is known to be mathematical science and not just mathematics. It is said to have been originated in seventeenth century. It consists of data collection, organizing the data, analyzing the data, interpretation and presentation of data. Statistical methods are being used since a long time in various fields to understand the data efficiently and to gain an in-depth analysis of the data [1].

On the other hand, machine learning is a branch of computer science which uses statistical abilities to learn from a particular dataset [2]. It was invented in the year 1959. It learns using algorithm and then has the ability to predict based on what it has been fed with. Machine learning gives out detailed information than statistics [3].

Most of the techniques of machine learning derive their behavior from statistics. However not many are familiar with this since both of them have their own jargons. For instance learning in statistics is called as fitting, supervised learning from machine learning is called as regression. Machine learning is a subfield of computer science and artificial intelligence. Machine learning is said to be a subdivision of computer science and artificial intelligence. It does use fewer assumptions than statistics. Machine learning unlike statistics deals with large amount of data and it also requires minimum human effort since most of its computation is done by the machine or the computer itself. Machine learning unlike statistics has a strong predicting power than statistics. Depending on the type of data machine learning can be categorized into supervised machine learning, unsupervised machine learning and reinforcement learning [4].

There seems to be analogy between machine learning and statistics. The following picture from textbook shows how statistics and machine learning visualize a model. **Table 1** shows how terms of statistics have been coined in machine learning.

To understand how machine learning and statistics come out with the results let's look at **Figure 1**. In statistical modeling on the left half of the image, linear regression with two variables is fitting the best plane with fewer errors. In machine learning the right half of the image to fit the model in the best possible way the independent variables have been converted into the square of error terms. That is machine learning strives to get a better fit than the statistical model. In doing so, machine learning minimizes the errors and increases the prediction rates.

Statistics methods are not just useful in training the machine learning model but they are helpful in many other stages of machine learning such as:

Machine learning	Statistics
Network, graphs	Model
Weights	Parameters
Learning	Fitting
Generalization	Tool set performance
Supervised learning	Regression/classification
Unsupervised learning	Density estimation, clustering

Table 1.
Machine learning jargons and corresponding statistics jargons.

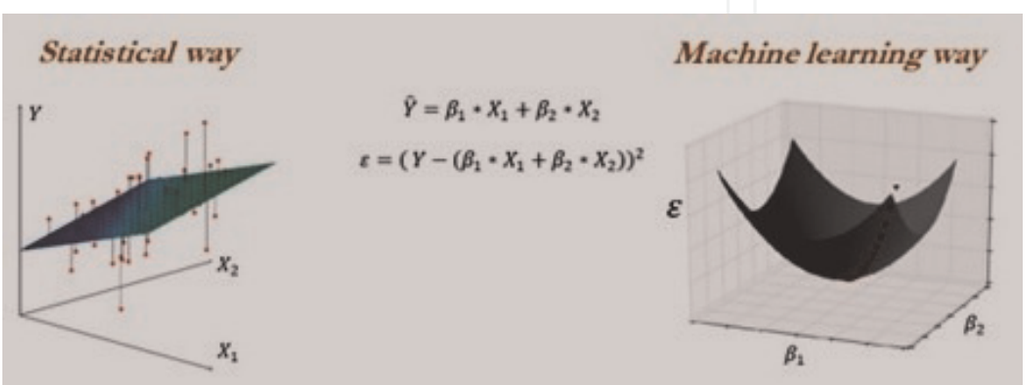

Figure 1.
Statistical and machine learning method.

- Data preparation—where statistics is used for data preprocessing which is later sent to the model. For instance when there are missing values in the dataset, we compute statistical mean or statistical median and fill it in the empty spaces of the dataset. It is recommended that machine learning model should never be fed with a dataset which has empty cells in it. It also used in preprocessing stage to scale the data by which the values are scaled to a particular range by which the mathematical computation becomes easy during the training of machine learning.

- Model evaluation—no model is perfect in predicting when it is built for the first time. Simply building the model is not enough. It is vital to check how well is it performing and if not then by how much is it closer to being accurate enough. Hence, we evaluate the model by statistical methods, which tell by how much the result is accurate and a lot many things about the end result obtained. We make use of metrics such as confusion matrix, Kolmogorov Smirnov chart, AUC—ROC, root mean squared error and many metrics to enhance our model.

- Model selection—we make use of many algorithms to train the algorithm and there is a chance of selecting only one which gives out accurate results when compared to others. The process of selecting the right solution for this is called model selection. Two of the statistical methods can be used to select the appropriate model such as statistical hypothesis test and estimation statistics [5].

- Data selection—some datasets carry a lot of features with them. Of many features, it may happen so that only some contribute significantly in estimation of the result. Considering all the features becomes computationally expensive and as well as time consuming. By making use of statistics concepts we can eliminate the features which do not contribute significantly in producing the result. That is it helps in finding out the dependent variables or features for any result. But it is important to note that this method requires careful and skilled approach. Without which it may lead to wrong results.

In this chapter we take a look at how statistical methods such as, regression and classification are used in machine learning with their own merits and demerits.

2. Regression

Regression is a statistical measure used in finance, investing and many other areas which aims to determine relationship between the dependent variables and 'n' number of independent variables. Regression consists of two types:

Linear regression—where one independent variable is used to explain or predict the outcome of the dependent variable.

Multiple regression—where two or more independent variables are used to explain or predict the outcome of the dependent variable.

In statistical modeling, regression analysis consists of set of statistical methods to estimate how the variables are related to each other.

Linear and logistic are the types of regression which are used in predictive modeling [6].

Linear assumes that the relationship between the variables are linear that is they are linearly dependent. The input variables consist of variables X_1, X_2, ..., X_n (where n is a natural number).

Linear models were developed long time ago but till date they are able to produce significant results. That is even in the modern computer's era they are well off. They are widely used because they are not complex in nature. In prediction, they can even out perform complex nonlinear models.

There are 'n' number of regressions that can be performed. We look at the most widely used five types of regression techniques. They are:

- Linear regression
- Logistic regression
- Polynomial regression
- Stepwise regression
- Ridge regression

Any regression method would involve the following:

- The unknown variables is denoted by beta
- The dependent variables also known as output variable
- The independent variables also known as input variables

It is denoted in the form of function as:

$$Y \approx f(X, \beta) \tag{1}$$

2.1 Linear regression

It is the most widely used regression type by far. Linear regression establishes a relationship between the input variables (independent variables) and the output variable (dependent variable).

$$\text{That is } Y = X_1 + X_2 + \ldots + X_n$$

It assumes that the output variable is a combination of the input variables. A linear regression line is represented by Y = a + bX, where X is the explanatory variable and Y is the dependent variable. The slope of the line is 'b', and 'a' is the intercept (the value of y when x = 0).

A line regression is represented by the equation:

$$Y = a + bX$$

where X indicates independent variables and 'Y' is the dependent variable [7]. This equation when plotted on a graph is a line as shown below in **Figure 2**.

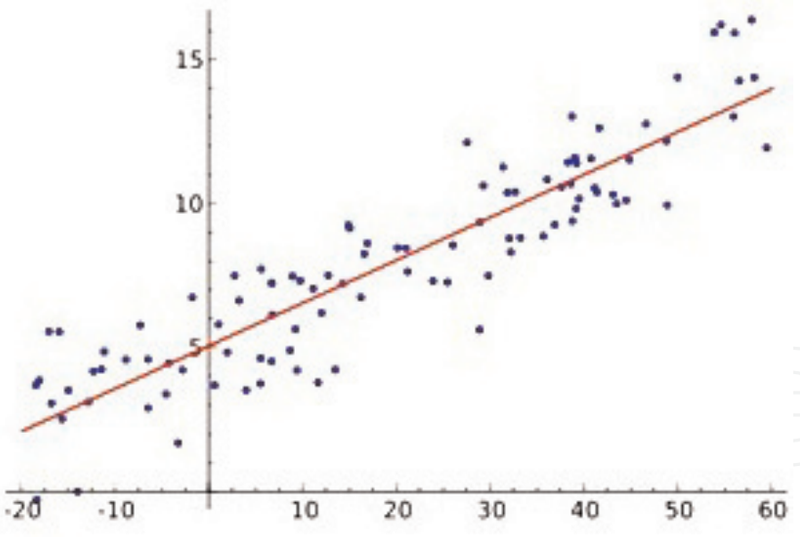

Figure 2.
Linear regression on a dataset.

However, linear regression makes the following assumptions:

- That there is a linear relationship
- There exists multivariate normality
- There exists no multi collinearity or little multicollinearity among the variables
- There exists no auto-correlation between the variables
- No presence of homoscedasticity

It is fast and easy to model and it is usually used when the relationship to be modeled is not complex. It is easy to understand. However linear regression is sensitive to outliners.

Note: In all of the usages stated in this chapter, we have assumed the following:

The dataset has been divided into training set (denoted by X) and test set (denoted by y_test)

The regression object "reg" has been created and exists.

We have used the following libraries:

Scipy and Numoy for numerical calculations

Pandas for dataset handling

Scikit-learn to implement the algorithm, to split the dataset and various other purposes.

Usage of linear regression in python:

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
#Declare the linear regression function
reg=linear_model.LinearRegression()
#call the method
reg.fit(height,weight)
#to check slope and intercept
```

```
m=reg.coef_[0]
b=reg.intercept_
print("slope=",m, "intercept=",b)
# check the accuracy on the training set
reg.score(X, y)
```

2.2 Logistic regression

Logistic regression is used when the dependent variable is binary (True/False) in nature. Similarly the value of y ranges from 0 to 1 (**Figure 3**) and it is represented by the equation:

$$\text{Odds} = p/(1-p) = \text{probability that event will occur/probability that the event will not occur}$$

$$\ln(\text{odds}) = \ln(p/(1-p)) \tag{2}$$

$$\text{logit}(p) = \ln(p/(1-p)) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 \ldots + b_kX_k$$

Logistic regression is used in classification problems. For example to classify emails as spam or not and to predict whether the tumor is malignant or not. It is not mandatory that the input variables have linear relationship to the output variable [8]. The reason being that it makes us of nonlinear log transformation to the predicted odds. It is advised to make use of only the variables which are powerful predictors to increase the algorithms performance.

However, it is important to note the following while making use of logistic regression:

Doesn't handle large number of categorical features.

The non-linear features should be transformed before using them.

Usage of logistic regression in python:

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
# instantiate a logistic regression model, and fit with X and y
reg = LogisticRegression()
reg = model.fit(X, y)
# check the accuracy on the training set
reg.score(X, y)
```

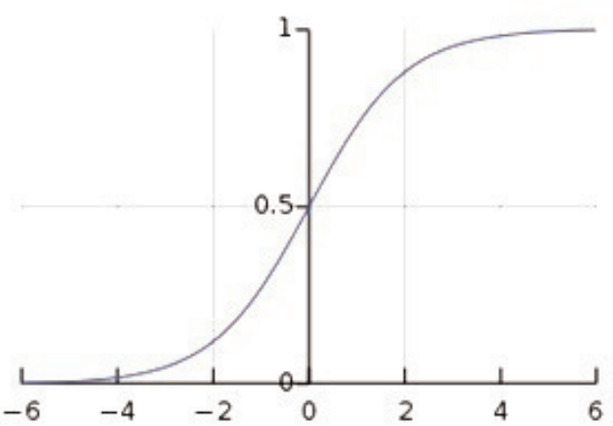

Figure 3.
Standard logistic function.

2.3 Polynomial regression

It is a type of regression where the independent variable power is greater than 1. Example:

$$Y = a + b(X_2 + X_3 + ...X_n). \tag{3}$$

The plotted graph is usually a curve in nature as shown in **Figure 4**.

If the degree of the equation is 2 then it is called quadratic. If 3 then it is called cubic and if it is 4 it is called quartic. Polynomial regressions are fit with the method of least squares. Since the least squares minimizes the variance of the unbiased estimators of all the coefficients which are done under the conditions of Gauss-Markov theorem. Although we may get tempted to fit a higher degree polynomial so that we could get a low error, it may cause over-fitting [9].

Some guidelines which are to be followed are:

The model is more accurate when it fed with large number of observations.

Not a good thing to extrapolate beyond the limits of the observed values.

Values for the predictor shouldn't be large else they will cause overflow with higher degree.

Usage of polynomial regression in python:

```
from sklearn.preprocessing import PolynomialFeatures
import numpy as np
#makes use of a pre-processor called degree for the function
reg = PolynomialFeatures(degree=2)
reg.fit_transform(X)
reg.score(X, y)
```

2.4 Step-wise regression

This type of regression is used when we have multiple independent variables. To select the variables which are independent an automatic process is used. If used in the right way it puts more power and presents us ton of information. It can be used when the number of variables is too many. However if it is used haphazardly it may affect the models performance.

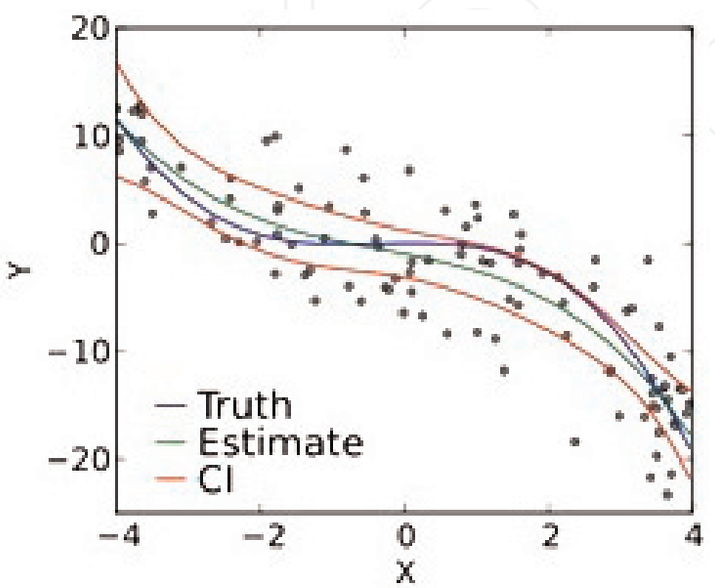

Figure 4.
Plotted graph is looks as curve in nature.

We make use of the following scores to help us find out the independent variables which contribute to the output variable significantly—R-squared, Adj. R-squared, F-statistic, Prob (F-statistic), Log-Likelihood, AIC, BIC and many more.

It can be performed by any of the following ways:

- Forward selection—where we start by adding the variables to the set and check how affects the scores.

- Backward selection—we start by taking all the variables to the set and start eliminating them one by one by looking at the score after each elimination.

- Bidirectional selection—a combination of both the methods mentioned above.

The greatest limitation of using step-wise regression is that the each instance or sample must have at least five attributes. Below which it has been observed that the algorithm doesn't perform well [10].

Code to implement Backward Elimination algorithm:

Assume that the dataset consists of 5 columns and 30 rows, which are present in the variable 'X' and let the expected results contain in the variable 'y'. Let 'X_opt' contain the independent variables which are used to determine the value of 'y'.

We are making use of a package called statsmodels, which is used to estimate the model and to perform statistical tests.

```
#import stats models package
import statsmodels.formula.api as sm
#since it is a polynomial add a column of 1s to the left
X = np.append (arr = np.ones([30,1]).astype(int), values = X, axis = 1)
#Let X-opt contain the independent variables only and Let y contain the output variable
X_opt = X[:,[0,1,2,3,4,5]]
#assign y to endog and X_opt to exog
regressor_OLS = sm.OLS(endog = y, exog = X_opt).fit()
regressor_OLS.summary()
```

The above code outputs the summary and based on it the variable which should be eliminated should be decided. Once decided remove the variable from 'X-opt'.

It is used to handle high dimensionality of the dataset.

2.5 Ridge regression

It can be used to analyze the data in detail. It is a technique which is used to get rid of multi collinearly. That is the independent values may be highly correlated. It adds a degree of bias due to which it reduces the standard errors.

The multi collinearity of the data can be inspected by correlation matrix. Higher the values, more the multi collinearity. It can also be used when number of predictor variables in the dataset exceeds the number of instances or observations [11].

The equation for linear regression is

$$Y = A + bX \tag{4}$$

This equation also contains error. That is it can be expressed as

$$Y = A + bX + (\text{error})$$

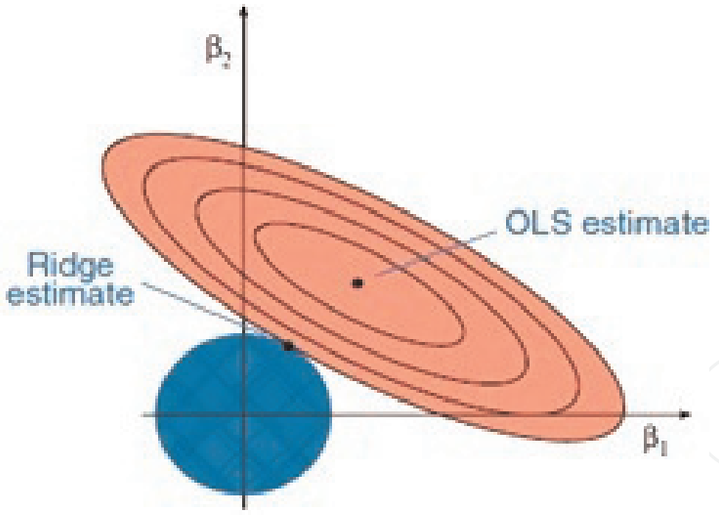

Figure 5.
Ridge and OLS.

Error with mean zero and known variance.

Ridge regression is known to shrink the size by imposing penalty on the size. It is also used to control the variance.

In (**Figure 5**) how ridge regression looks geometrically.

Usage of ridge regression in python:

```
from sklearn import linear_model
reg = linear_model.Ridge (alpha = .5)
reg.fit ([[0, 0], [0, 0], [1, 1]], [0, .1, 1])
Ridge(alpha=0.5, copy_X=True, fit_intercept=True, max_iter=None,
normalize=False, random_state=None, solver='auto', tol=0.001)
#to return the co-efficient and intercept
reg.coef_
reg.intercept_
```

2.6 Lasso regression

Least absolute shrinkage and selection operator is also known as LASSO. Lasso is a linear regression that makes use of shrinkage. It does so by shrinking the data values toward the mean or a central point. This is used when there are high levels of multi collinearity [12].

It is similar to ridge regression and in addition it can reduce the variability and improves the accuracy of linear regression models.

It is used for prostate cancer data analysis and other cancer data analysis.

Important points about LASSO regression:

- It helps in feature extraction by shrinking the co-efficient to zero.
- It makes use of L1 regularization.
- In the data if the predictors are have high correlation, the algorithm selects only one of the predictors discards the rest.

Code to implement in python:

```
from sklearn import linear_model
clf = linear_model.Lasso(alpha = 0.1)
```

```
clf.fit()
Lasso(alpha=0.1, copy_X=True, fit_intercept=True, max_iter=1000,
normalize=False, positive=False, precompute=False, random_state=None,
selection='cyclic', tol=0.0001, warm_start=False)
#to return the co-efficent and intercept
print(clf.coef_)
print(clf.intercept_)
```

3. Classification

A classification task is when the output is of the type "category" such as segregating data with respect to some property. In machine learning and statistics, classification consists of categorizing the new data to a particular category where it fits in on the basis of the data which has been used to train the model. Examples of tasks which make use of classification techniques are classifying emails as spam or not, detecting a disease on plants, predicting whether it will rain on some particular day, predicting the house prices based on the area it is located.

In terms of machine learning classification techniques fall under supervised learning [13].

The categories may be either:

- categorical (example: blood groups of humans—A, B, O)
- ordinal (example: high, medium or low)
- integer valued (example: occurrence of a letter in a sentence)
- Real valued

The algorithms which make use of this concept in machine learning and classify the new data are called as "Classifiers." Algorithms always return a probability score of belonging to the class of interest. That is considered an example where we are required to classify a gold ornament. Now when we input the image to the machine learning model the algorithms returns the probability value for each category, such as for if it is a ring the probability value may be higher than 0.8 if it not a necklace it may return less than 0.2, etc.

Higher the value more likely it is for it to belong to the particular group.

We make use of the following approach to build a machine learning classifier:

1. Pick a cut off probability above which we consider a record to belong to that class.
2. Estimate that a new observation belongs to a class.
3. If the obtained probability is above the cut off probability, assign the new observation to that class.

Classifiers are of two types: linear and nonlinear classifiers.

We now take a look at various classifiers are also statistical techniques:

1. Naive Bayes
2. stochastic gradient dissent (SGD)

3. K-nearest neighbors

4. decision trees

5. random forest

6. support vector machine

3.1 Naive Bayes

In machine learning, these classifiers belong to "probabilistic classifiers." This algorithm makes use of Bayes' theorem with strong independence assumptions between the features. Although Naive Bayes were introduced in the early 1950s, they are still being used today [14].

Given a problem instance to be classified, represented by a vector

$$X = (x_1, x_2, x_3, \ldots, x_n)$$

Which represent 'n' features.

$$P(Ck \mid x_1, x_2, \ldots, x_n)$$

We can observe that in the above formula that if the number of features is more or if a feature accommodates a large number of values, then it becomes infeasible. Therefore we rewrite the formula based on Bayes theorem as:

$$p(C_k|x) = p(C_k)p(x|C_k)/p(x) \tag{5}$$

Makes two "naïve" assumptions over attributes:

- All attributes are a priori equally important
- All attributes are statistically independent (value of one attribute is
- not related to a value of another attribute)

This classifier makes two assumptions:

- All attributes are equally important
- All attributes are not related to another attribute

There are three types of naive Bayes algorithms, which can be used: GaussianNB, BernoulliNB, and MultinomialNB.

Usage of naive Bayes in python:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.naive_bayes import GaussianNB
reg= GaussianNB()
reg.fit(X,y)
reg.predict(X_test)
reg.score()
```

3.2 Stochastic gradient dissent (SGD)

An example of linear classifier which implements regularized linear model (**Figure 6**) with stochastic gradient dissent. Stochastic gradient descent (often shortened to SGD), also known as incremental gradient descent, is an iterative method to optimize a differentiable objective function, a stochastic approximation of gradient descent optimization [15]. Although SGD has been a part of machine learning since ages it wasn't extensively used until recently.

In linear regression algorithm, we make us of least squares to fit the line. To ensure that the error is low we use gradient descent. Although gradient descent does the job it can't handle big tasks hence we use stochastic gradient classifier. SGD calculates the derivative of each training data and also calculates the update within no time.

The advantages of using SGD classifier are that they are efficient and they are easy to implement.

However it is sensitive to feature scaling.

Usage of SGD classifier:

```
from sklearn.linear_model import SGDClassifier
X = [[0., 0.], [1., 1.]]
y = [0, 1]
clf = SGDClassifier (loss = "hinge", penalty = "l2")
clf.fit(X, y)
#to predict the values
clf.fit(X_test)
```

3.3 K-nearest neighbors

Also known as k-NN is a method used to classify as well as for regression. The input consists of k number of closest training examples. It is also referred as lazy learning since the training phase doesn't require a lot of effort.

In k-NN an object's classification is solely dependent on the majority vote of the object's neighbors. That is the outcome is based on the presence of the neighbors. The object is assigned to the class most common among its k nearest neighbors. If the value of k is equal to 1 then it's assigned to its nearest neighbor. Simply put, the

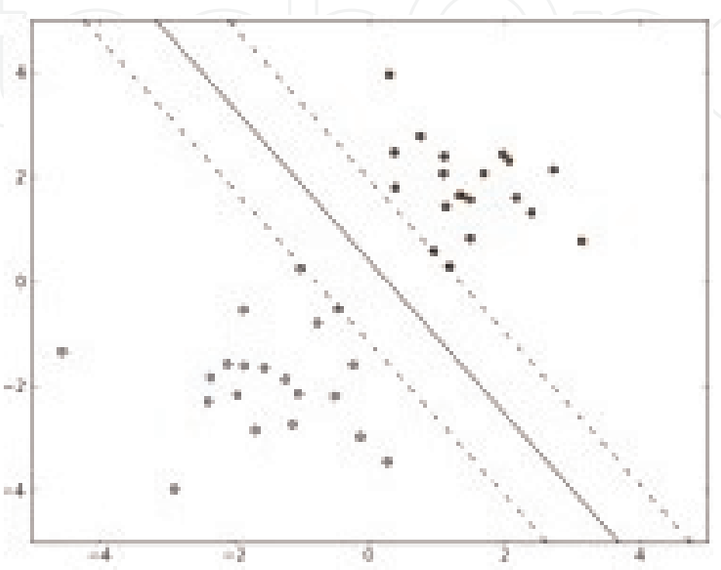

Figure 6.
Feature scaling classifier.

k-NN algorithm is entirely dependent on the neighbors of the object to be classified. Greater the influence of a neighbor, the object is assigned to it. It is termed as simplest machine learning algorithm among all the algorithms [16].

Let us consider an example where the green circle is the object which is to be classified as shown in **Figure 7**. Let us assume that there are two circles—the solid circle and the dotted circle.

As we know that there are two classes class 1 (blue squares) and class 2 (red squares). If we consider only the inner circle that is the solid circle then there are two objects of red circle existing which dominates the number of blue squares due to which the new object is classified to Class 1. But if we consider the dotted circle, the number of blue circle dominates since there are more number of blue squares due to which the object is classified to Class 2 [17].

However, the cost of learning process is zero.

The algorithm may suffer from curse of dimensionality since the number of dimensions greatly affects its performance. When the dataset is very large the computation becomes very complex since the algorithm takes time to look out for its neighbors. If there are many dimensions then the samples nearest neighbors can be far away. To avoid curse of dimensionality dimension reduction is usually performed before applying k-NN algorithm to the data.

Also the algorithm may not perform well with categorical data since it is difficult to find the distance between the categorical features.

Usage in python:

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier (n_neighbors=5)
classifier.fit(X_train, y_train)
```

3.4 Decision trees

Decision trees are considered to be most popular classification algorithms while classifying data. Decision trees are a type of supervised algorithm where the data is split based on certain parameters. The trees consist of decision nodes and leaves [18].

The decision tree consists of a root tree from where the tree generates and this root tree doesn't have any inputs. It is the point from which the tree originates. All the other nodes except the root node have exactly one incoming node. The other

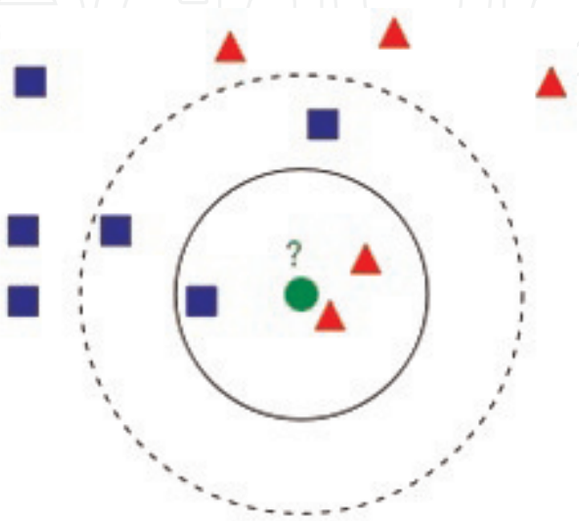

Figure 7.
K-Neighbors.

nodes except the root node are called leaves. Below is the example of a decision tree an illustration of how the decision tree looks like as shown in **Figure 8**.

"Is sex male" is the root node from where the tree originates. Depending on the condition the tree further bifurcates into subsequent leaf nodes. Few more conditions like "is Age >9.5?" are applied by which the depth of the node goes on increasing. As the number of leaf nodes increase the depth of the tree goes on increasing. The leaf can also hold a probability vector.

Decision tree algorithms implicitly construct a decision tree for any dataset.

The goal is to construct an optimal decision tree by minimalizing the generalization error. For any tree algorithm, it can be tuned by making changes to parameters such as "Depth of the tree," "Number of nodes," "Max features." However construction of a tree by the algorithm can get complex for large problems since the number of nodes increase as well as the depth of the tree increases.

Advantages of this tree are that they are simple to understand and can be easily interpreted. It also requires little data preparation. The tree can handle both numerical and categorical data unlike many other algorithms. It also easy to validate the decision tree model using statistical testes. However, disadvantages of the trees are that they can be complex in nature for some cases which won't generalize the data well. They are unstable in nature since if there are small variations in data they may change the structure of the tree completely.

Usage in python:

```
from sklearn.neighbors import tree
classifier = tree.DecisionTreeClassifier()
classifier.fit(X_train, y_train)
clf.predict(X_test)
```

3.5 Random forest

These are often referred as ensemble algorithms since these algorithms combine the use of two or more algorithms. They are improved version of bagged decision trees. They are used for classification, regression, etc.

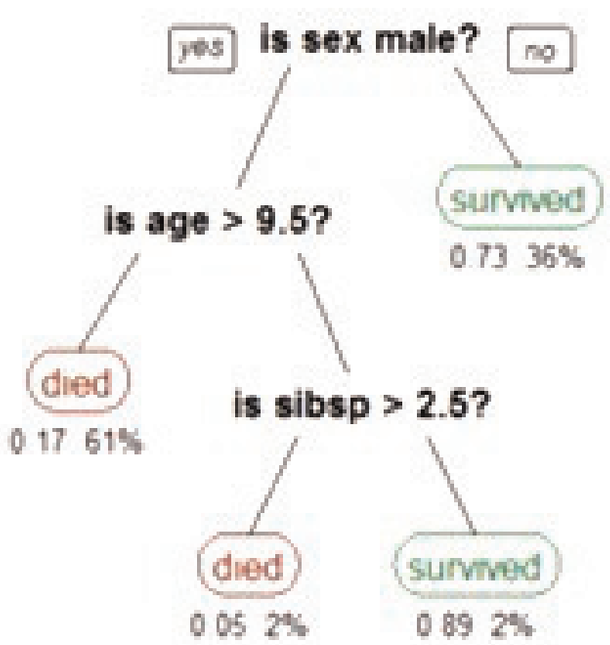

Figure 8.
Typical decision tree.

Random forest creates n number of decision trees from a subset of the data. On creating the trees it aggregates the votes from the different trees and then decides the final class of the sample object. Random forest is used in recommendation engines, image classification and feature selection [19].

The process consists of four steps:

1. It selects random samples from the dataset.

2. For every dataset construct a dataset and then predict from every decision tree.

3. For every predicted result perform vote.

4. Select the prediction which has the highest number of votes.

Random forest's default parameters often produce a good result in most of the cases. Additionally, one can make changes to achieve desired results. The parameters in Random Forest which can be used to tune the algorithm which can be used to give better and efficient results are:

1. Increasing the predictive power by increasing "n_estimators" by which the number of tress which will be built can be altered. "max_features" parameter can also be adjusted which is the number of features which are used to train the algorithm. Another parameter which can be adjusted is "min_sample_leaf" which is the number of leafs that are used to split the internal node.

2. To increase the model's speed, "n_jobs" parameter can be adjusted which is the number of processors it can use. To use as many as needed "−1" can be specified which signifies that there is no limit.

Due to large number of decision trees random forest is highly accurate. Since it takes the average of all the predictions which are computed the algorithm doesn't suffer from over fitting. Also it does handle missing values from the dataset. However, the algorithm is takes time to compute since it takes time to build trees and take the average of the predictions and so on.

One of the real time examples where random forest algorithm can be used is predicting a person's systolic blood pressure based on the person's height, age, weight, gender, etc.

Random forests require very little tuning when compared to other algorithms. The main disadvantage of random forest algorithm is that increased number of tress can make the process computationally expensive and lead to inaccurate results.

Usage in python:

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(max_depth=2, random_state=0)
clf.fit(X, y)
clf.predict(X_test)
```

3.6 Support vector machine

Support vector machines also known as SVMs or support vector networks fall under supervised learning. They are used for classification as well as regression purposes. Support vectors are the data points which lie close to the hyper plane. When the data is fed to the algorithm the algorithm builds a classifier which can be

used to assign new examples to one class or the other [20]. A SVM consists of points in space separated by a gap which is as wide as possible. When a new sample is encountered it maps it to the corresponding category.

Perhaps when the data is unlabeled it becomes difficult for the supervised SVM to perform and this is where unsupervised method of classifying is required.

A SVM constructs a hyper plane which can be used for classification, regression and many other purposes. A good separation can be achieved when the hyper plane has the largest distance to the nearest training point of a class.

In (**Figure 9**) H_1 line doesn't separate, while H_2 separates but the margin is very small whereas H_3 separates such as the distance between the margin and the nearest point is maximum when compared to H_1 and H_2.

SVMs can be used in a variety of applications such as:

They are used to categorize text, to classify images, handwritten images can be recognized, and they are also used in the field of biology.

SVMs can be used with the following kernels:

1. Polynomial kernel SVM

2. Linear kernel SVM

3. Gaussian kernel SVM

4. Gaussian radial basis function SVM (RBF)

The advantages of SVM are:

1. Effective in high dimensional data

2. It is memory efficient

3. It is versatile

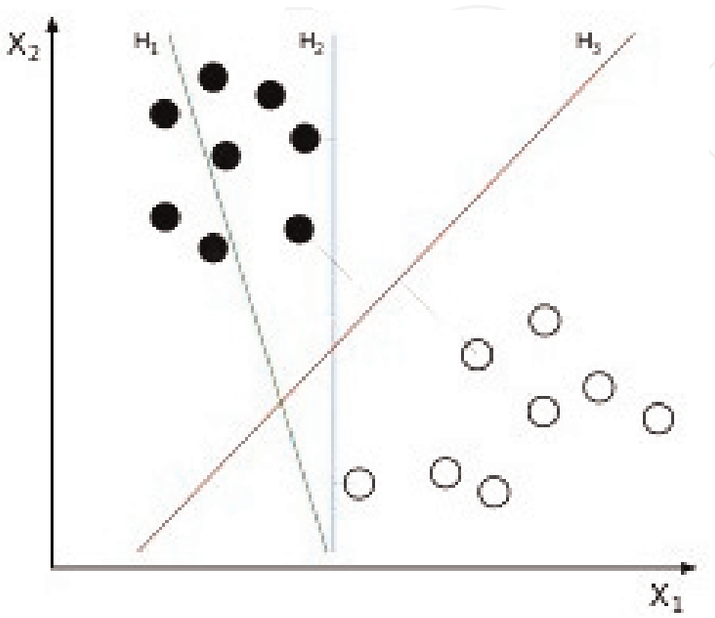

Figure 9.
Hyper plane construction and H_1, H_2 and H_3 line separation.

It may be difficult for SVM to classify at times due to which the decision boundary is not optimal. For example, when we want to plot the points randomly distributed on a number line.

It is almost impossible to separate them. So in such cases we transform the dataset by applying 2D or 3D transformations by using a polynomial function or any other appropriate function. By doing so it becomes easier to draw a hyper plane.

When the number of features is much greater than number of samples it doesn't perform well with the default parameters.

Usage of SVM in python:

```
from sklearn import svm
clf = svm.SVC()
clf.fit(X,y)
clf.predict(X_test)
```

4. Conclusion

It is evident from the above regression and classification techniques are strongly influenced by statistics. The methods have been derived from statistical methods which existed since a long time. Statistical methods also consist of building models which consists of parameters and then fitting it. However not all the methods which are being used derive their nature from statistics. Not all statistical methods are being used in machine learning. Extensive research in the field of statistical methods may give out new set methods which can be used in machine learning apart from the existing statistical methods which are being used today. It can also be stated that machine learning to some extent is a form of 'Applied Statistics.'

Author details

Pramod Kumar[1*], Sameer Ambekar[1†], Manish Kumar[2] and Subarna Roy[1]

1 Department of Health Research, Biomedical Informatics Centre, ICMR-National Institute of Traditional Medicine, Belagavi, Karnataka, India

2 Department of Electrical Engineering, College of Engineering, Bharti Vidyapeeth, Pune, Maharashtra, India

*Address all correspondence to: pramodbiotech@gmail.com

† Sameer Ambekar shares first authorship.

References

[1] Hawkins DM. On the investigation of alternative regressions by principal component analysis. Journal of the Royal Statistical Society Series. 1973;**22**: 275-286. https://www.jstor.org/stable/i316057

[2] Kourou K, Exarchos TP, Exarchos KP, Karamouzis MV, Fotiadis DI. Machine learning applications in cancer prognosis and prediction. Computational and Structural Biotechnology Journal. 2015;**13**:8-17. DOI: 10.1016/j.csbj.2014.11.005

[3] Machine Learning [Internet]. Available from: https://en.wikipedia.org/wiki/Machine_learning

[4] Trevor H, Robert T. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York: Springer; 2009. pp. 485-586. DOI: 10.1007/978-0-387-84858-7_14

[5] Aho K, Derryberry DW, Peterson T. Model selection for ecologists: The worldviews of AIC and BIC. Ecology. 2014;**95**(3):631-636. DOI: 10.1890/13-1452.1

[6] Freedman DA. Statistical Models: Theory and Practice. USA: Cambridge University Press; 2005. ISBN: 978-0-521-85483-2

[7] sklearn.linear_model.LinearRegression—scikit-learn 0.19.2 documentation [Internet]. Available from: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[8] Linear Regression—Wikipedia [Internet]. Available from: https://en.wikipedia.org/wiki/Linear_regression

[9] Shaw P et al. Gergonne's 1815 paper on the design and analysis of polynomial regression experiments. Historia Mathematica; 2006;**1**(4):431-439. DOI: 10.1016/0315-0860(74)90033-0

[10] Stepwise Regression—Wikipedia [Internet]. Available from: https://en.wikipedia.org/wiki/Stepwise_regression

[11] Tikhonov Regularization—Wikipedia [Internet]. Available from: https://en.wikipedia.org/wiki/Tikhonov_regularization

[12] sklearn.linear_model.LogisticRegression—scikit-learn 0.19.2 documentation [Internet]. Available from: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[13] Statistical Classification—Wikipedia [Internet]. Available from: https://en.wikipedia.org/wiki/Statistical_classification

[14] Naive Bayes Scikit-Learn 0.19.2 Documentation [Internet]. Available from: http://scikit-learn.org/stable/modules/naive_bayes.html

[15] Stochastic Gradient Descent—Scikit-Learn 0.19.2 Documentation [Internet]. Available from: http://scikit-learn.org/stable/modules/sgd.html

[16] k-Nearest Neighbors Algorithm—Wikipedia [Internet]. Available from: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[17] Kamiński B, Jakubczyk M, Szufel P. A framework for sensitivity analysis of decision trees. Central European Journal of Operations Research. 2017;**26**: 135-159. DOI: 10.1007/s10100-017-0479-6

[18] Lasso (Statistics)—Wikipedia [Internet]. Available from: https://en.wikipedia.org/wiki/Lasso_(statistics)

[19] sklearn.linear_model.Lasso—Scikit-Learn 0.19.2 Documentation [Internet]. Available from: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html

[20] Corinna C, Vapnik Vladimir N. Support-vector networks. Machine Learning. 1995;**20**(3):273-297. DOI: 10.1007/BF00994018