International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

Supervised Learning Classification Algorithms Comparison

Aditya Singh Rathore

B.Tech, J.K. Lakshmipat University

Abstract: *Under supervised machine learning, classification tasks are one of the most important tasks as a part of data analysis. It gives a lot of actionable insights to data scientists after using different machine learning algorithms. For study purpose the Titanic dataset has been chosen. Through this paper, an effort has been made to evaluate different classification models' performance on the dataset using the scikit-learn library. The classifiers chosen are Logistic Regression, K-Nearest Neighbor, Decision Tree, Random Forest, Support Vector Machine and Naïve Bayes classifier. In the end evaluation parameters like confusion matrix, precision, recall, f1-score and accuracy have been discussed.*

Key Words: Numpy, Pandas, classification, knn, logistic regression, decision tree, random forest, Naïve Bayes, SVM, sklearn.

1. INTRODUCTION:

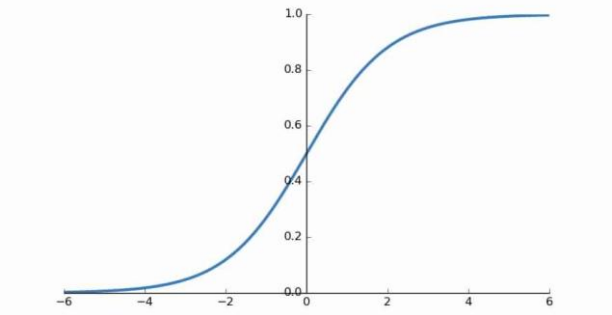
Machine learning has been getting a lot of attention for the past few years because it holds the key to solve the problems of the future. In the 21st Century, we are living in a world where data is everywhere and as we go by of our day we are generating data, be it text messaging or simply walking down the street and different antennas picking GPS signals. The majority of practical machine learning uses supervised learning. In supervised learning, the machine learning models are first trained over dataset with associated correct labels. The objective of training the models first is to create a very good mapping between the independent and dependent variables, so that when that model is run on a new dataset, it can predict the output variable. Classification is a type of problem under supervised learning. A classification problem is when the output is a category and when given one or more inputs a classification model will try to predict the value based on what relation it developed between the input and output. Below are the classifiers chosen for evaluation:

1.1 Logistic Regression:

This type of technique proves to be more appropriate when the dependent variable/output is in the form of binary output, i.e. 1 or 0. This technique is mainly used when one wants to see the relationship between the dependent variable and one or more independent variable provided that the dataset is linear. The output of this technique gives out a binary output which is a better type of output than absolute values as it elucidates the output. In order to limit our output between 0 and 1 we will be using Sigmoid Function.

Below is the mathematical representation of Sigmoid Function:

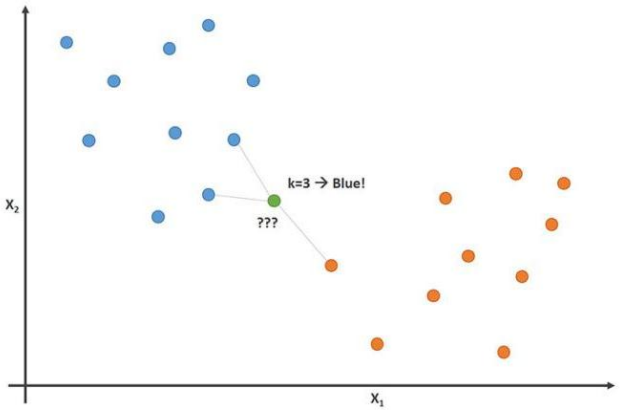
f(x)= 1/(1+e^-x)



1.2 K- Nearest Neighbor:

This type of technique is one of the simplest classification algorithm. It first stores the entire data in its training phase as an array of data points. . Whenever a prediction is required for a new data it searches through the entire training dataset for a number of most similar instances and the data with most similar instance is returned as prediction. This number is defined by an integer 'k'. The k denotes the number of neighbours which are the determining class of the test data. For example, if k=3, then the labels of the 3 closest instances are checked and the most common label is assigned to the test data. The distance measured from test data to the closest instance is the least distance measure. This can be achieved using various measuring tools such as Euclidean Distance. The Euclidean distance between point p and q is calculated as below:

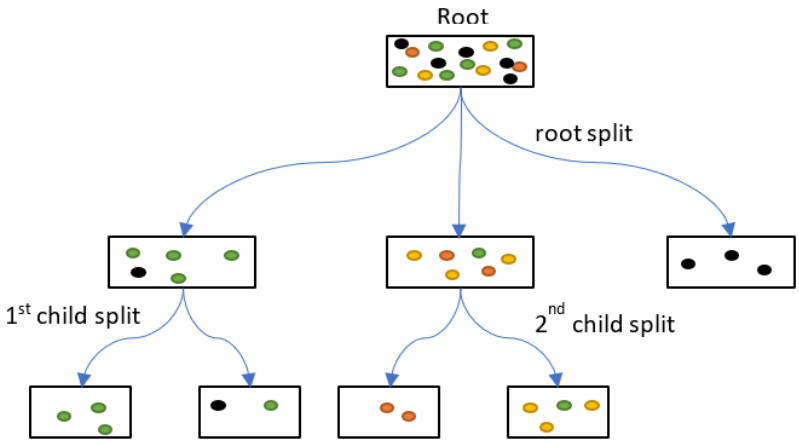
$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$



1.3 Decision Tree:

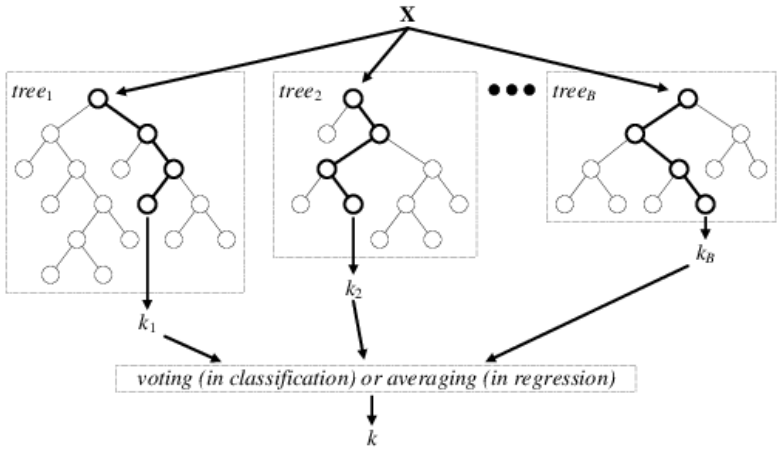
A decision tree is a graphical representation of all possible solutions to a decision based on certain conditions. It consists of a Root/Parent node which is the base node or the first node of a tree. Then comes the splitting which is basically dividing the Root/Child nodes into different parts on a certain condition. These parts are called the Branch/ Sub Tree. To remove unwanted branches from the tree we make use of Pruning. The last node is called the leaf node when the data cannot be segregated to further levels. We will use CART (Classification and Regression Tree) algorithm to our dataset. Below are some terminologies used by CART algorithm in order to select the root node :

a) Gini Index: It is the measure of impurity used to build a decision tree
b) Information Gain: It is the decrease in entropy after a dataset is split on the basis of a criteria.
c) Reduction in Variance: The split with lower variance is selected as a criteria.
d) Chi Square: It is the measure of statistical significance between the differences between sub-nodes and parent node.



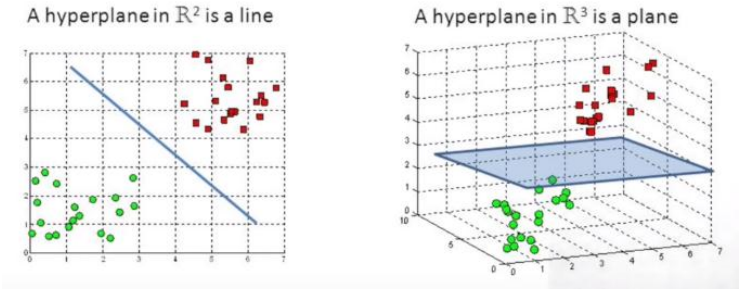
1.4 Random Forest:

This classifier is one of the most used algorithms today because it can be used for classification as well as regression. The main problem with decision trees is that it can over fit the data. One solution to prevent this is to create multiple decision trees on random subsamples of the data. Now when a new data point is to be predicted then that will be computed by all decision trees and a majority vote will be calculated. This vote will determine the class of the data. This in turns gives higher accuracy when compared to decision tree alone. More decision trees directly correlates with more accuracy.



1.5 Support Vector Machine (SVM):

This classifier uses a hyperplane/decision boundary that separates the classes. This hyperplane is positioned in such a way that it maximises the margin i.e. the space between the decision boundary and the points in each of the classes that are closest to the decision boundary. These points are called Support Vectors. A hyperplane is a decision surface that splits the space into 2 parts i.e. it is a binary classifier. This hyperplane in R^n space is an n-1 dimensional subspace.



SVMs can easily be used for linear classification. However, in order to use SVM for non-linear classification once can use a kernel trick with which the data is transformed in another dimension to determine the appropriate position of hyperplane.

1.6 Naïve Bayes:

This classifier is used primarily for text classification which generally involves training on high dimensional datasets. This classifier is relatively faster than other algorithms in making predictions. It is called "naïve" because it makes an assumption that a feature in the dataset is independent of the occurrence of other features i.e. it is conditionally independent. This classifier is based on the Bayes Theorem which states that the probability of an event Y given X is equal to the probability of the event X given Y multiplied by the probability of X upon probability of Y.

$P(X|Y) = (P(Y|X) \cdot P(X))/P(Y)$

2. METHODOLOGY:

2.1 Dataset:

The data has been split into two groups:

a) training set (train.csv)
b) test set (test.csv)

The training set is used to build and train the machine learning models. The test set is used to see how well the model performed on unseen data. We will use the trained models to predict whether or not the passengers in the test set survived the sinking of the Titanic.

Below is the original structure of the dataset:

```
In [7]: train.head()
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Data Dictionary:

- Survived: 0 = No, 1 = Yes
- pclass: Ticket class 1 = 1st, 2 = 2nd, 3 = 3rd
- sibsp: # of siblings / spouses aboard the Titanic
- parch: # of parents / children aboard the Titanic
- ticket: Ticket number
- cabin: Cabin number
- embarked: Port of Embarkation C = Cherbourg, Q = Queenstown, S = Southampton

After pre-processing of the data and feature engineering, below is the dataset we would be finally be working with:

```
In [65]: train.head()
```

Out[65]:

	Survived	Pclass	Age	SibSp	Parch	Fare	male	Q	S
1	1	1	38.0	1	0	71.2833	0	0	0
3	1	1	35.0	1	0	53.1000	0	0	1
6	0	1	54.0	0	0	51.8625	1	0	1
10	1	3	4.0	1	1	16.7000	0	0	1
11	1	1	58.0	0	0	26.5500	0	0	1

2.2 Evaluation Parameters:

2.2.1 Confusion Matrix:

On a classification problem a confusion matrix gives out a summary of predicted results. The basic idea behind it is to count the number of times the instances of class A are classified as class B.

Below is the sample confusion matrix:

	Pridicted Yes	Pridicted No
Actual Yes	True Positive	False Negative
Actual No	False Positive	True Negative

2.2.2 Precision:

It is the ratio of total number of true positives divided by total sum of true positives and false positives. In summary it describes how many of the returned hits were true positive i.e. how many of the found were correct hits.

Precision = True Positives / (True Positives + False Positives)

2.2.3 Recall:

It is the ratio of the total number of true positives divided by total sum of the true positives and the false negatives. It is also called sensitivity. In summary it describes how many of the true positives were recalled (found), i.e. how many of the correct hits were also found.

Recall = True Positives / (True Positives + False Negatives)

2.2.4 F1- Score:

It is simply the weighted average of precision and recall wherein the best value for a model is 1 and the worst value is 0. Below is the formula to calculate the same:

F1-score = 2 * (precision * recall) / (precision + recall)

2.3 Experimental Results:

Below are the experimental results of different Classifiers:

Classifier	**Class**	**Classification Report**				**Confusion Matrix**
		Precision	Recall	f1-score	Accuracy	
Logistic Regression	0	0.81	0.86	0.84	78.90%	[[120 19]
	1	0.75	0.67	0.7		[28 56]]
KNN	0	0.73	0.83	0.77	69.90%	[[115 24]
	1	0.63	0.49	0.55		[43 41]]
Decision Tree	0	0.84	0.78	0.81	77.17%	[[108 31]
	1	0.67	0.76	0.72		[20 64]]
Random Forest	0	0.87	0.88	0.88	84.30%	[[123 16]
	1	0.8	0.77	0.79		[19 65]]
SVM	0	0.73	0.76	0.74	67.20%	[[106 33]
	1	0.57	0.52	0.55		[40 44]]
Naïve Bayes	0	0.83	0.83	0.83	78.47%	[[115 24]
	1	0.71	0.71	0.71		[24 60]]

3. CONCLUSION:

In this paper we experimented with 6 types of classifiers on the Titanic dataset. In order to evaluate the above the train dataset was split into train & test dataset, since there is no true output for test dataset. From the above the random forest seems to be most effective with highest precision and recall values and has the least number of false positive and false negatives.

4. REFERENCES:

[1].P-N. Tan, M. Steinbach, V. Kumar, "Introduction to Data Mining," Addison-Wesley Publishing, 2006.

[2]. Simm, J., Magrans de Abril, I., & Sugiyama, M., Tree-based ensemble multi-task learning method for classification and regression, 2014.

[3] Takeuchi, I. & Sugiyama, M, Target neighbor consistent feature weighting for nearest neighbor classification.

[4] M. Kantardzic, "Data Mining: Concepts, Models, Methods, and Algorithms," John Wiley & Sons Publishing, 2003.