Comparing 5 Different Classification Algorithms

Decision Tree, Boosted Trees, Random Forest, Support Vector Machines and Neural Networks.

Rajath Singh
Machine Learning Intern
AI Technologies and Systems
ai-techsystems.com
rajathsinghcool@gmail.com

***Abstract* – This project deals with 5 different Classification algorithms and it is tested on the Wine dataset from UCI Machine Learning Repository. The dataset is of a multivariate characteristic with around 178 number of instances and with 13 variables and has no missing data. We will use Machine Learning by applying all five algorithms Decision Tree, Boosted Trees, Random Forest, Support Vector Machines and Neural Networks and testing out which performs better. The evaluation metric will be based on the Accuracy of the algorithm which will decide the performance of the algorithm.**

Keywords *– Classification, Decision Tree, Boosted Tree, Random Forest, SVM, Neural Network, Wine data, machine learning.*

I. INTRODUCTION

Wine is an alcoholic drink made from fermented grapes. Yeast consumes the sugar in the grapes and converts it to ethanol, carbon dioxide, and heat. Different varieties of grapes and strains of yeasts produce different styles of wine. These variations result from the complex interactions between the biochemical development of the grape, the reactions involved in fermentation, the terroir, and the production process. Many countries enact legal appellations intended to define styles and qualities of wine. These typically restrict the geographical origin and permitted varieties of grapes, as well as other aspects of wine production. Wines not made from grapes include rice wine and fruit wines such as plum, cherry, pomegranate, currant and elderberry.

II. DATA PREPROCESSING

The dataset comprises of 178 rows and 13 columns. It is used mainly for multiple class classification problem. With this dataset we get a clear understanding that which class out of the three given classes has a good concentration and quality of wine. The dataset comprises of the following items ‘data’, ‘feature names’, ‘targets’, ‘target names’. Where data consists of all the recorded data and targets are the desired target for the specified data item.

These are the following columns or attributes –

1) Alcohol
2) Malic acid
3) Ash
4) Alkalinity of ash
5) Magnesium
6) Total phenols
7) Flavonoids
8) Non-flavonoid phenols
9) Proanthocyanins
10) Color intensity
11) Hue
12) OD280/OD315 of diluted wines
13) Proline

III. EVALUATION METRIC

The evaluation of the algorithms depends on the following factors –

- Classification Accuracy
- Confusion Matrix
- Classification Report

Classification Accuracy

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

It works well only if there are equal number of samples belonging to each class.

For example, consider that there are 98% samples of class A and 2% samples of class B in our training set. Then our model can easily get **98% training accuracy** by simply predicting every training sample belonging to class A.

When the same model is tested on a test set with 60% samples of class A and 40% samples of class B, then the **test accuracy would drop down to 60%.** Classification Accuracy is great, but gives us the false sense of achieving high accuracy.

The real problem arises, when the cost of misclassification of the minor class samples are very high. If we deal with a rare but fatal disease, the cost of failing to diagnose the disease of a sick person is much higher than the cost of sending a healthy person to more tests.

Confusion Matrix

Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.

Let's assume we have a binary classification problem. We have some samples belonging to two classes : YES or NO. Also, we have our own classifier which predicts a class for a given input sample. On testing our model on 165 samples we get the following result.

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

There are 4 important terms –

- **True Positives** : The cases in which we predicted YES and the actual output was also YES.
- **True Negatives** : The cases in which we predicted NO and the actual output was NO.
- **False Positives** : The cases in which we predicted YES and the actual output was NO.
- **False Negatives** : The cases in which we predicted NO and the actual output was YES.

Accuracy for the matrix can be calculated by taking average of the values lying across the **"main diagonal".**

$$\text{Formula 1: Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Classification Report

The classification report visualizer displays the precision, recall, F1, and support scores for the model.

The classification report shows a representation of the main classification metrics on a per-class basis. This gives a deeper intuition of the classifier behavior over global accuracy which can mask functional weaknesses in one class of a multiclass problem.

The metrics are defined in terms of true and false positives, and true and false negatives. Positive and negative in this case are generic names for the classes of a binary classification problem.

F1 Score – F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).

Precision – It is the number of correct positive results divided by the number of positive results predicted by the classifier.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Recall – It is the number of correct positive results divided by the number of ***all*** relevant samples (all samples that should have been identified as positive).

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

IV. ALGORITHMS

We have used 5 algorithms and we need to figure out the best performing algorithm on our dataset according to the evaluation metric. The data was split into training set and test set and fed to the specific algorithms and after fitting the classifier the accuracy, confusion matrix and F1 score are displayed and the one with the highest scores performs well on the Wine dataset.

A. Decision Trees

A **decision tree** is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map non-linear relationships quite well. They are adaptable at solving any kind of problem at hand (classification or regression). Decision Tree algorithms are referred to as **CART (Classification and Regression Trees)**.

For the research purpose we load the dataset from Scikit learn and split the data into training and test sets and call the Decision Tree Classifier and fit the classifier to the two sets and then predict the model on the training set and calculate the accuracy, confusion matrix and see the classification report.

The accuracy obtained by the model is 93.33%

The confusion matrix is as follows –

Actual Values of the Target variables	**Predicted Values of Target Variables**		
	Class 0	*Class 1*	*Class 2*
Class 0	15	1	0
Class 1	1	19	1
Class 2	0	0	8

The Classification report is as follows –

	precision	*recall*	*f1-score*
Class 0	0.94	0.94	0.94
Class 1	0.95	0.90	0.93
Class 2	0.89	1.00	0.94

B. Boosted Tree

XGBoost stands for "Extreme Gradient Boosting", where the term "Gradient Boosting" originates from the paper *Greedy Function Approximation: A Gradient Boosting Machine*, by Friedman.

This algorithm goes by lots of different names such as gradient boosting, multiple additive regression trees, stochastic gradient boosting or gradient boosting machines.

Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added sequentially until no further improvements can be made. A popular example is the AdaBoost algorithm that weights data points that are hard to predict.

Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

XGBoost dominates structured or tabular datasets on classification and regression predictive modeling problems.

The accuracy obtained by the model is 95.56%

The confusion matrix is as follows –

Actual Values of the Target variables	**Predicted Values of Target Variables**		
	Class 0	*Class 1*	*Class 2*
Class 0	16	0	0
Class 1	1	19	1
Class 2	0	0	8

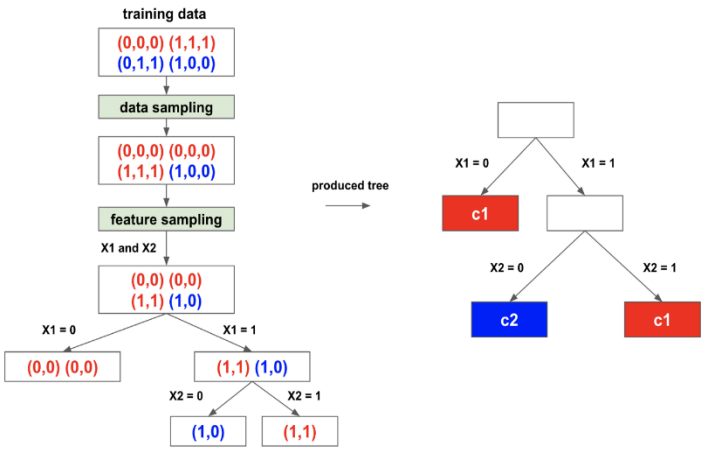
The Classification report is as follows –

	precision	*recall*	*f1-score*
Class 0	0.94	1.00	0.97
Class 1	1.00	0.90	0.95
Class 2	0.89	1.00	0.94

C. Random Forest

Random forests, also known as random decision forests, are a popular ensemble method that can be used to build predictive models for both classification and regression problems. Ensemble methods use multiple learning models to gain better predictive results — in the case of a random forest, the model creates an entire forest of random uncorrelated decision trees to arrive at the best possible answer.

The figure below demonstrates how to build a random forest tree.



The accuracy obtained by the model is 93.33%

The confusion matrix is as follows –

Actual Values of the Target variables	Predicted Values of Target Variables		
	Class 0	*Class 1*	*Class 2*
Class 0	15	1	0
Class 1	1	19	1
Class 2	0	0	8

The Classification report is as follows –

	precision	*recall*	*f1-score*
Class 0	0.94	0.94	0.94
Class 1	0.95	0.90	0.93
Class 2	0.89	1.00	0.94

D. .Support Vector Machine

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

The accuracy obtained by the model is 51.11%

The confusion matrix is as follows –

Actual Values of the Target variables	Predicted Values of Target Variables		
	Class 0	*Class 1*	*Class 2*
Class 0	1	15	0
Class 1	0	21	0
Class 2	0	7	1

The Classification report is as follows –

	precision	*recall*	*f1-score*
Class 0	1.00	0.06	0.12
Class 1	0.49	1.00	0.66
Class 2	1.00	0.12	0.22

E. Neural Networks

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems.

A neural network works similarly to the human brain's neural network. A "neuron" in a neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis.

Hidden layers fine-tune the input weightings until the neural network's margin of error is minimal. It is hypothesized that hidden layers extrapolate salient features in the input data that have predictive power regarding the outputs. This describes feature extraction, which accomplishes a utility similar to statistical techniques such as principal component analysis.

The accuracy obtained by the model on training set is – 30.075%

The accuracy obtained by the model on test set is – 0.177%

V. RESULT

We observe that out of all the algorithms the XGBoost algorithm is performing well on the dataset. The obtained accuracy of XGboost is 95.56% which is much higher than Neural Networks. The following dataset has very less values and columns to work on with and that is the main reason why a Neural Network couldn't perform well on the given dataset.

Therefore, XGBoost is the classifier that outperformed every other algorithm.

REFERENCES

[1] Scikit Learn Library
[2] Towards Data Science
[3] Investopedia
[4] Jonathon Bechtel Blog on Wine Data