Comparative Study on Classification Algorithms in Machine Learning

Shikha Goel

Machine Learning Intern

AI Tech Systems

www.ai-techsystems.com

Delhi, India

shikhagoel555@gmail.com

***Abstract* - Machine Learning is an application of Artificial Intelligence and is revolutionizing the way companies do business. At its core, it's an algorithm or model that learns patterns in big data and then predicts similar patterns in new data. In layman's terms, it's the theory that machines should be able to learn and adapt through experience to produce reliable, repeatable decisions and results. Selecting the right algorithm is a key part of any machine learning project, and because there are dozens to choose from, understanding their strengths and weaknesses in various business applications is essential. This report discusses five of the most common machine learning algorithms and some of their potential use cases.**

***Keywords* - Machine Learning, Artificial Intelligence, Decision Tree, Algorithm, Random Forest, Boosted Tree, Support Vector Machine, Neural Network**

I. INTRODUCTION

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people. The principles of machine learning are based on how humans learn and acquire knowledge and arrive at some decision. Machine learning is a generic set of algorithms. They are not dictated by specific set of problem. These are some of the general set of techniques available to us. The solution to specific problems lies in the data. Machine learning algorithms allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs. In this report, we'll discuss about the most common algorithmic approaches in machine learning, including the decision tree learning, random forest, boosted tree learning, support vector machines and neural network. We'll explore which programming languages are most used in machine learning, providing with some of the positive and negative attributes of each.

II. DECISION TREES

In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A decision tree is a tree-like graph with nodes representing the place where we pick an attribute and ask a question; edges represent the answers to the question; and the leaves represent the actual output or class label. They are used in non-linear decision making with simple linear decision surface.

A. Working of Decision Trees

Let's illustrate the working with help of an example of a decision tree based off a few splits of data from kyphosis dataset. It shows the number of patients who have kyphosis condition. So, it shows the presence and absence of kyphosis condition in patients.

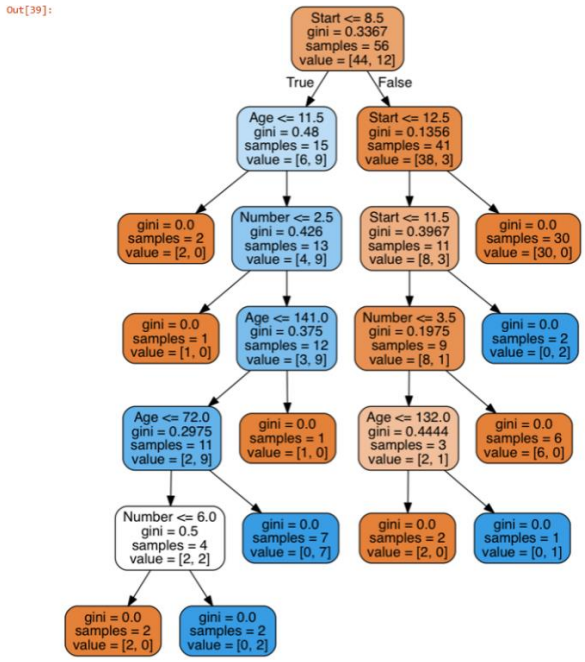


Fig 1: Decision Tree based on kyphosis data set.

Here we can see this Decision Tree is splitting the patients based on Age, Number and Start to predict the presence or absence of kyphosis condition.
Now an important question emerges.
How do we decide which features to split on?
In order to pick which feature to split on, we need a way of measuring how good the split is. This is where information gain and entropy come in. We would like to choose questions that give a lot of information about the tree's prediction. For example, if there is a single yes/no question that accurately predicts the outputs 99% of the time, then that question allows us to "gain" a lot of information about our data. In order to measure how much information we gain, we introduce entropy. The entropy is a measure of uncertainty associated with our data. We can intuitively think that if a data set had only one label, then we have a low entropy. So, we would like to split our data in a way that minimizes the entropy. The better the splits, the better our prediction will be. The equation for entropy is:

$$H = -\sum p(x) \log p(x)$$

Here, p(x) is the percent of the group that belongs to a given class and H is the entropy. Decision tree is the most powerful and popular tool for classification and prediction.

B. Positive Attributes

- Easy to use and understand.
- Can handle both categorical and numerical data.
- Resistant to outliers. Hence, require little data preprocessing.
- New features can be easily added.
- Can be used to build larger classifiers by using ensemble methods.

C. Negative Attributes

- Prone to overfitting.
- Require measurement to know how well they are doing.
- Need to be careful with parameter tuning.
- Can create biased learned trees if some classes dominate.

III. 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. 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.

A. Working of Random forest

The idea behind a Random Forest is pretty simple. We repeatedly select data from the data set (with replacement) and build a Decision Tree with each new sample. It is important to note that since we are sampling with replacement, many data points will be repeated, and many won't be included as well. This is important to keep in mind when we talk about measuring error of a Random Forest. Another important feature of the Random Forest is that each node of the Decision Tree is limited to only considering splits on random subsets of the features.

Suppose John wants to visit a place for a holiday. So, he asks his friends to recommend the best place to visit. His friends recommended him different places to visit. He considered all the recommendations and calculated the votes. He will choose the place which will get the highest number of votes.

In this case, the recommended place is the Target Prediction which is considered by many friends. Each friend is the tree and combining all the friends will form a forest. This forest is the random forest. This

algorithm helped him to find the best place to visit for a holiday.

In the case of classification with Random Forests, we use each tree in our forest to get a prediction, then the label with the most votes becomes the predicted class for that data point.

If we wanted to know how well our Random Forest performed, we could use a standard cross validation method of splitting the data into a training and testing set, then comparing the predictions to the actual values.

B. Positive Attributes

- It is powerful and accurate.
- The predictive performance can compete with the best supervised learning algorithms.
- They provide a reliable feature importance estimate. They offer efficient estimates of the test error without incurring the cost of repeated model training associated with cross-validation

C. Negative Attributes

- No interpretability.
- Overfitting can easily occur.
- Need to choose the number of trees.
- An ensemble model is inherently less interpretable than an individual decision tree.
- Training a large number of trees can have high computational cost.
- Predictions are slower, which may create challenges for applications.

IV. BOOSTED TREES

Over the past few years, this technique has emerged as one of the most powerful methods for predictive data mining. Some implementations of these powerful algorithms allow them to be used for regression as well as classification problems, with continuous and/or categorical predictors. Boosting means that each tree is dependent on prior trees. The algorithm learns by fitting the residual of the trees that preceded it. Thus, boosting in a decision tree ensemble tends to improve accuracy with some small risk of less coverage.

A. Working of Boosted Trees

Gradient Boosting is basically about "boosting" many weak predictive models into a strong one, in the form of ensemble of weak models. Here, a weak predict model can be any model that works just a little better than random guess.

To build the strong model, we need to find a good way to "combine" weak models.

1. Train a weak model m using data samples drawn according to some weight distribution
2. Increase the weight of samples that are misclassified by model m, and decrease the weight of samples that are classified correctly by model m
3. Train next weak model using samples drawn according to the updated weight distribution

In this way, the algorithm always trains models using data samples that are "difficult" to learn in previous rounds, which results an ensemble of models that are good at learning different "parts" of training data.
As a variant of multiple decision trees, gradient boosting selects binary questions that will improve prediction accuracy for each new tree. Decision trees are therefore grown sequentially, as each tree is created using information derived from the previous decision tree.

The way this works is that mistakes incurred on the training data are recorded and then applied to the next round of training data. At each iteration, weights are added to the training data based on the results of the previous iteration. Higher weighting is applied to instances that were incorrectly predicted from the training data and instances that were correctly predicted receive less weight. The training and test data are then compared and errors are again logged in order to inform weighting at each subsequent round. Earlier iterations that do not perform well, and that perhaps misclassified data, can thus be improved upon via further iterations.

This process is repeated until there is a low level of error. The final result is then obtained from a weighted average of the total predictions derived from each model.

B. Positive Attributes

- It is a robust out of the box classifier (regressor) that can perform on a dataset on which minimal effort has been spent on cleaning.

- It can learn complex non-linear decision boundaries via boosting.

C. Negative Attributes

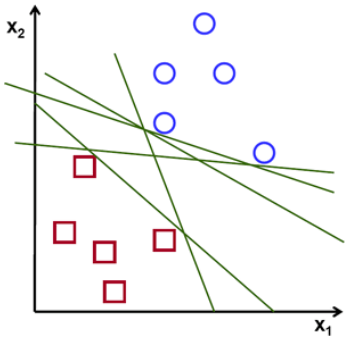
- Harder to tune other models, because you have so many hyperparameters and you can easily overfit.
- Lack of interpretability compared to linear classifiers. All you get are "variable importance" stats, but you do not have a straightforward way to study how variables interact and contribute to the final prediction.
- Not very speedy to train or score.

V. SUPPORT VECTOR MACHINES

Support vector machine is another simple algorithm that every machine learning expert should have in his/her arsenal. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables. For categorical variables a dummy variable is created with case values as either 0 or 1. To construct an optimal hyperplane, SVM employs an iterative training algorithm, which is used to minimize an error function.

A. Working of Support Vector Machines

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.



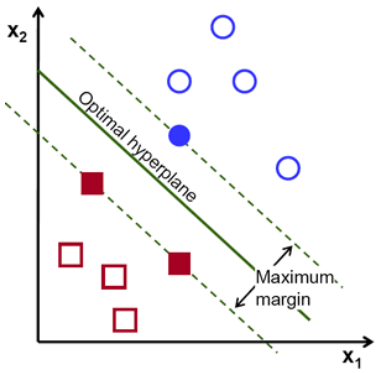


Fig 2: Possible hyperplanes

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. SVM is used for classifying. For example, Dream Housing Finance company deals in all home loans. They have a presence across all urban, semi-urban and rural areas. Company wants to automate the loan eligibility process based on customer detail provided while filling online application form. To automate this process, they have given a problem to identify the customers segments, those are eligible for loan amount so that they can specifically target these customers. These targeted customers can be identified using SVM.

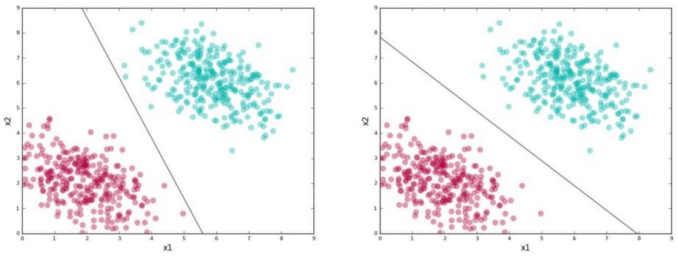


Fig 3: Classification of data set

The line here is our separating boundary (because it separates out the labels) or classifier (we use it classify points).

In case of complex data, turning parameters are used. Varying those we can achieve considerable non-linear classification line with more accuracy in reasonable amount of time. These are:

Kernel

The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra. This is where the kernel plays role.

Regularization

The Regularization parameter (often termed as C parameter in python's sklearn library) tells the SVM optimization how much you want to avoid misclassifying each training example. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger margin separating hyperplane, even if that hyperplane misclassifies more points.

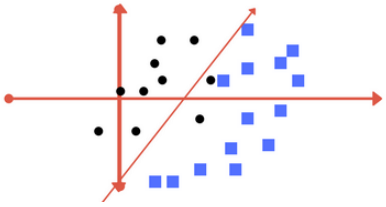


Fig 4: Low regularization value

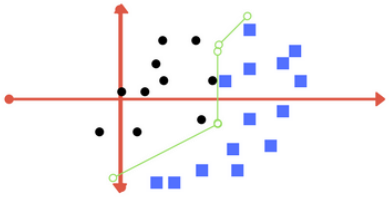


Fig 5: High regularization value

Gamma

The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. In other words, with low gamma, points far away from plausible separation line are considered in calculation for the separation line. Whereas high gamma means the points close to plausible line are considered in calculation.

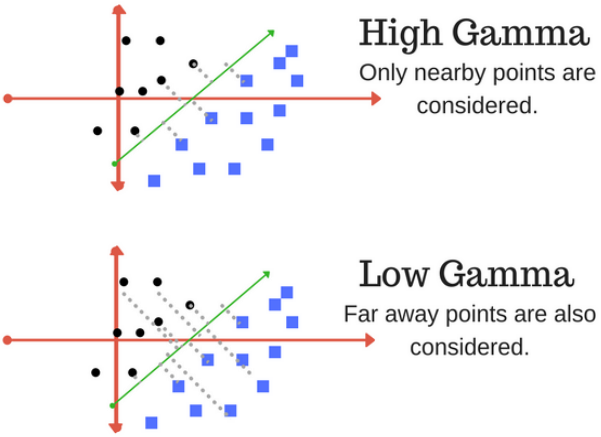


Fig 6: Gamma values

Margin

And finally, last but very important characteristic of SVM classifier. SVM to core tries to achieve a good margin. A margin is a separation of line to the closest class points.

A good margin is one where this separation is larger for both the classes. Images below gives to visual example of good and bad margin. A good margin allows the points to be in their respective classes without crossing to other class.

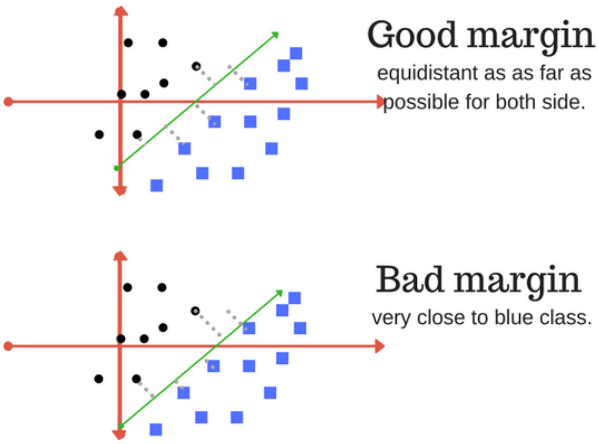


Fig 7: Margin values

B. Positive Attributes

- It works really well with clear margin of separation.
- It is effective in high dimensional spaces.
- It is effective in cases where number of dimensions is greater than the number of samples.

- It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

C. Negative Attributes

- It doesn't perform well, when we have large data set because the required training time is higher.
- It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping.
- SVM doesn't directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is related SVC method of Python scikit-learn library.

VI. NEURAL NETWORK

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated. Neural networks help us cluster and classify. You can think of them as a clustering and classification layer on top of the data you store and manage. They help to group unlabeled data according to similarities among the example inputs, and they classify data when they have a labeled dataset to train on.

A. Working of Neural Network

The whole idea of artificial neural network is based on the concept of the structure and functions of a human brain. Neural networks are composed of layers of computational units called neurons, with connections in different layers. These networks transform data until they can classify it as an output. Each neuron multiplies an initial value by some weight, sums results with other values coming into the same neuron, adjusts the resulting number by the neuron's bias, and then normalizes the output with an activation function.

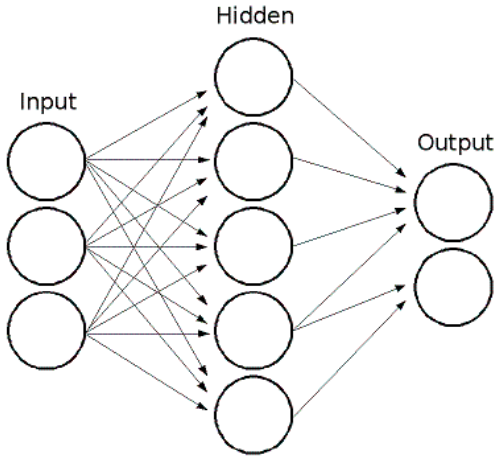


Fig 8: Working of Neural Network

For example, Target marketing involves market segmentation, where we divide the market into distinct groups of customers with different consumer behavior.

Neural networks are well-equipped to carry this out by segmenting customers according to basic characteristics including demographics, economic status, location, purchase patterns, and attitude towards a product. Unsupervised neural networks can be used to automatically group and segment customers based on the similarity of their characteristics, while supervised neural networks can be trained to learn the boundaries between customer segments based on a group of customers.

All in all, neural networks have made computer systems more useful by making them more human.

B. Positive Attributes

- Neural networks are flexible and can be used for both regression and classification problems. Any data which can be made numeric can be used in the model, as neural network is a mathematical model with approximation functions.
- Neural networks are good to model with nonlinear data with large number of inputs.
- Once trained, the predictions are pretty fast.
- Neural networks can be trained with any number of inputs and layers.
- Neural Networks work best with more data points.

C. *Negative Attributes*

- We cannot know how much each independent variable is influencing the dependent variables.
- It is computationally very expensive and time consuming to train with traditional CPUs.
- Neural networks depend a lot on training data. This leads to the problem of over-fitting and generalization. The mode relies more on the training data and may be tuned to the data.

VII. CONCLUSION

Machine Learning is a class of methods for automatically creating models from data. Machine learning algorithms are the engines of machine learning, meaning it is the algorithms that turn a data set into a model. In this report, we have discussed about the most common algorithmic approaches in machine learning, including the decision tree learning, random forest, boosted tree learning, support vector machines and neural network. Each machine learning algorithm or model has its own strengths and weaknesses. A problem that resides in the machine learning domain is the concept of understanding the details in the algorithms being used and its prediction accuracy. Some models are easier to interpret or understand but lack prediction power. Whereas other models may have really accurate predictions but lack interpretability. To answer which kind of algorithm works best depends on the kind of problem you're solving, the computing resources available, and the nature of the data.

REFERENCES

1. www.wikipedia.com
2. www.towardsdatascience.com
3. www.kaggle.com
4. www.statsoft.com
5. www.sciencedirect.com
6. www.geeksforgeeks.org