PES UNIVERSITY ONLINE

DATA ANALYTICS

Unit 4: Brief review of other classifiers: SVM, ANN and Data Driven Approaches

Jyothi R.

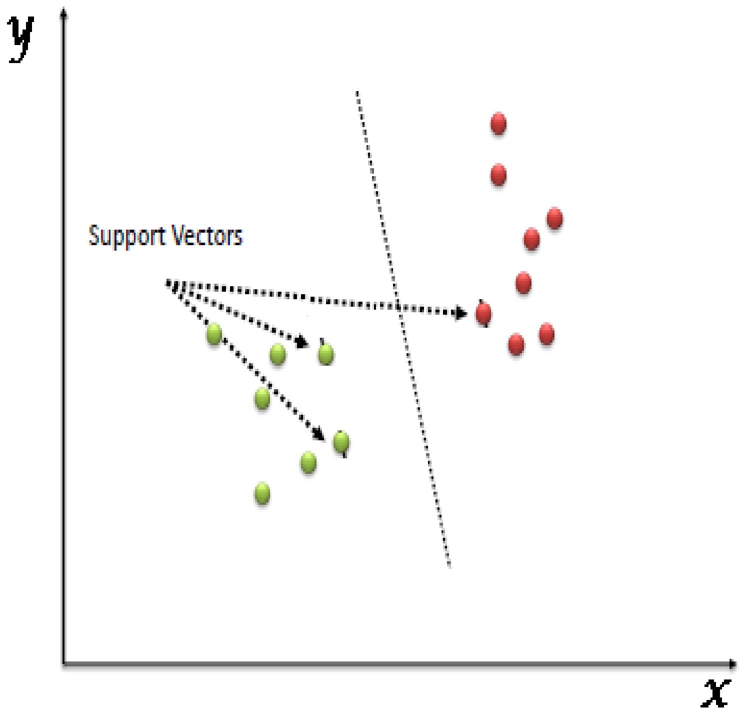
Department of Computer Science
and
Engineering

Introduction

- Support vector machine(SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges.
- In the SVM algorithm, we plot each data item as a point in n-dimensional space with the value of each feature being the value of a particular coordinate.
- Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

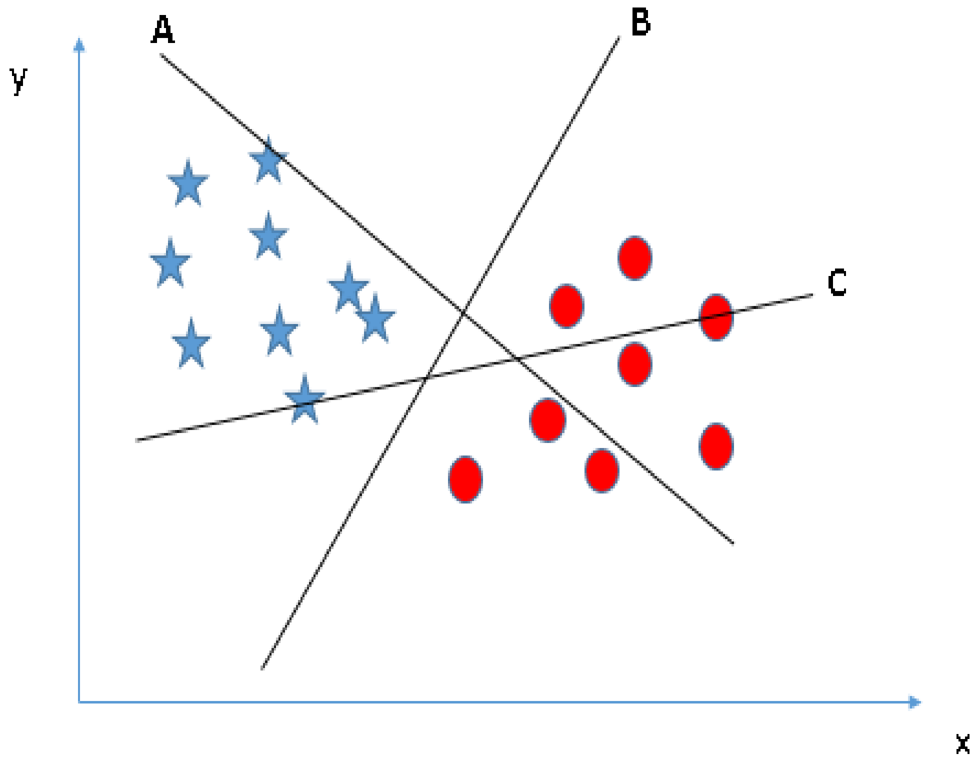
Introduction

- Support Vectors are simply the co-ordinates of individual observation.
- The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).



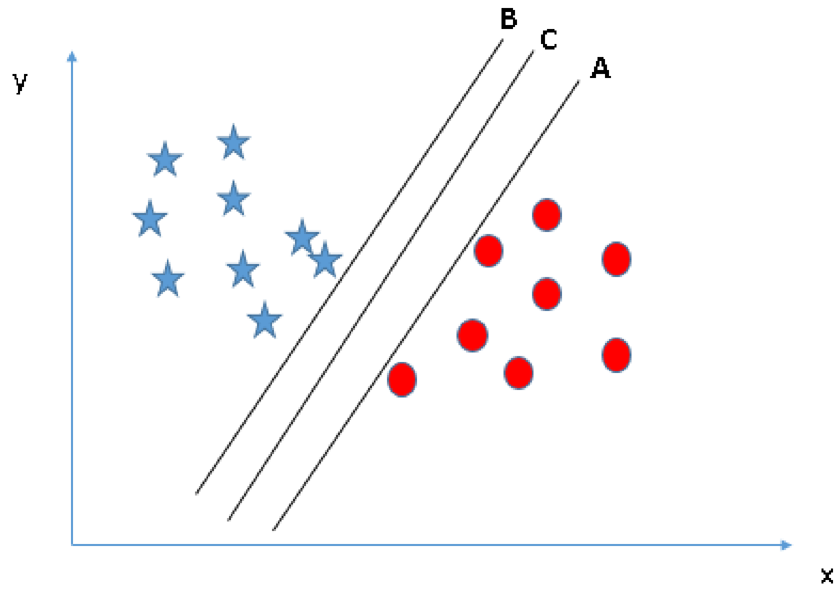
How does it work?

- **Identify the right hyper-plane (Scenario-1):** Here, we have three hyper-planes (A, B and C).
- Now, identify the right hyper-plane to classify star and circle.
- We need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”.
- In this scenario, hyper-plane “B” has excellently performed this job.



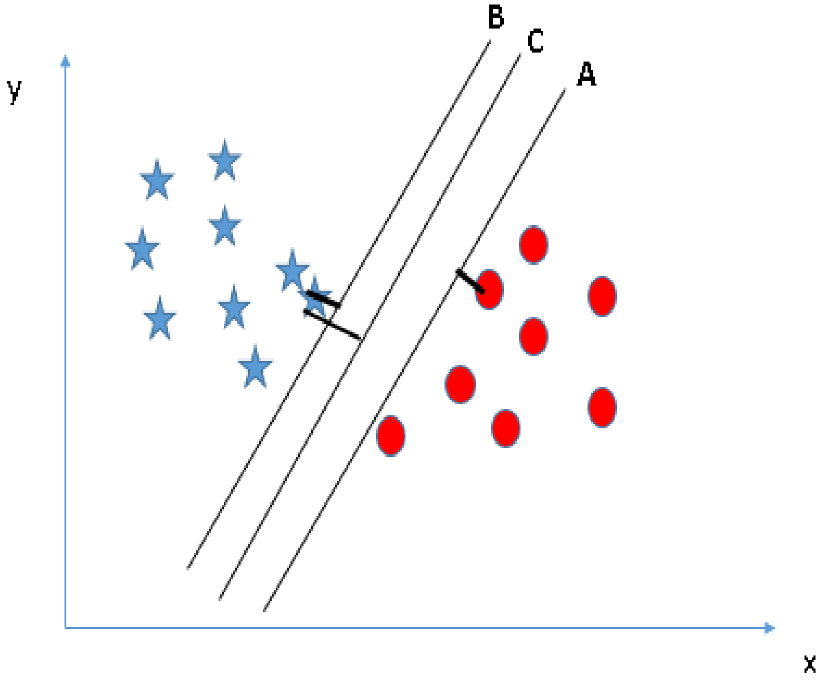
How does it work?

- **Identify the right hyper-plane (Scenario-2):** Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?
- Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as Margin



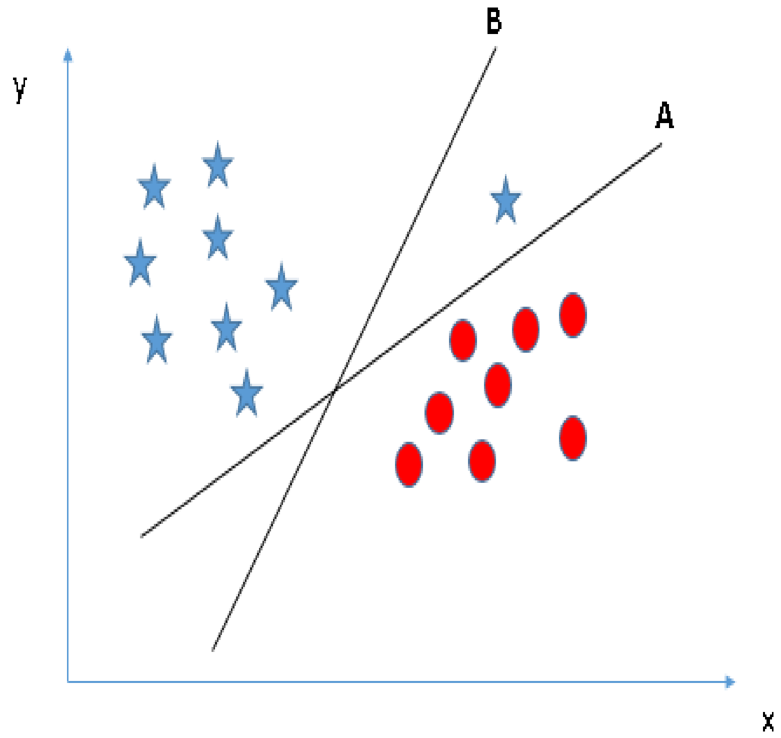
How does it work?

- **Identify the right hyper-plane (Scenario-2):** Above, you can see that the margin for hyper-plane C is high as compared to both A and B.
- Hence, we name the right hyper-plane as C.
- Another lightning reason for selecting the hyper-plane with higher margin is robustness.
- If we select a hyper-plane having low margin then there is high chance of miss-classification.



How does it work?

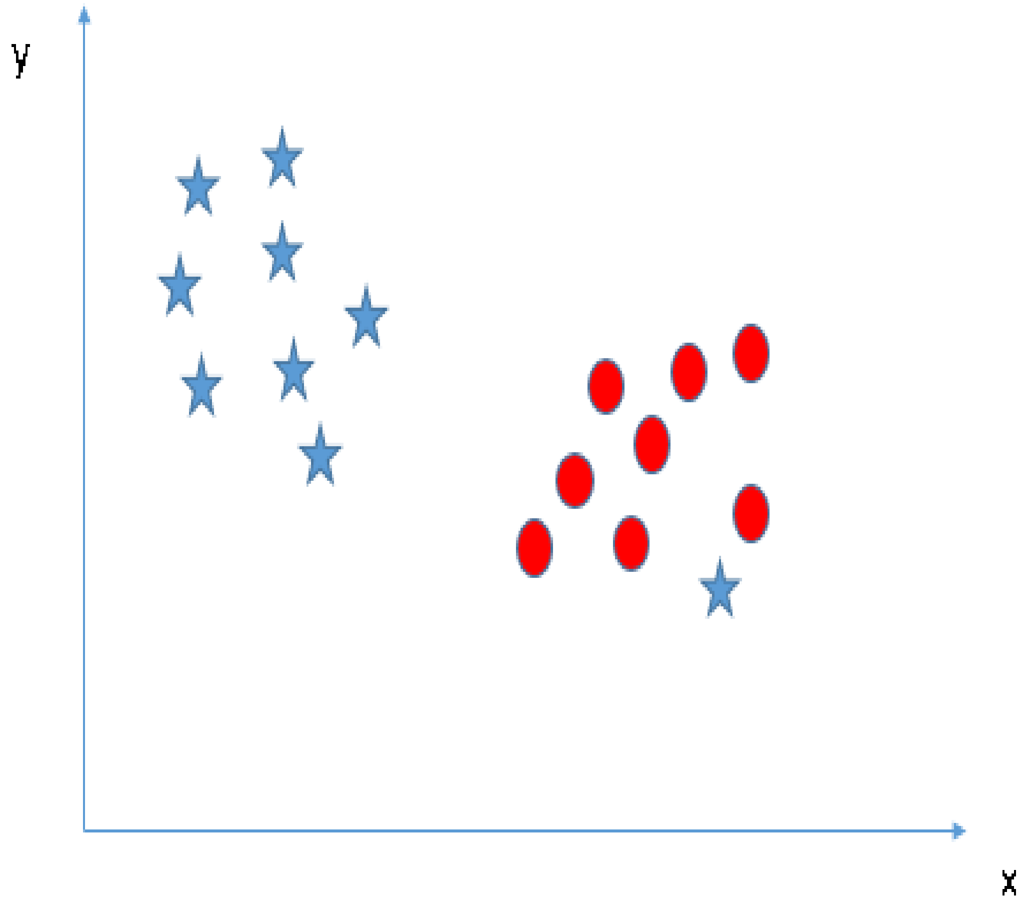
- **Identify the right hyper-plane (Scenario-3):**
- Use the rules as discussed in previous section to identify the right hyper-plane.
- Some of you may have selected the hyper-plane B as it has higher margin compared to A.
- But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin.
- Here, hyper-plane B has a classification error and A has classified all correctly.



Therefore, the right hyper-plane is A.

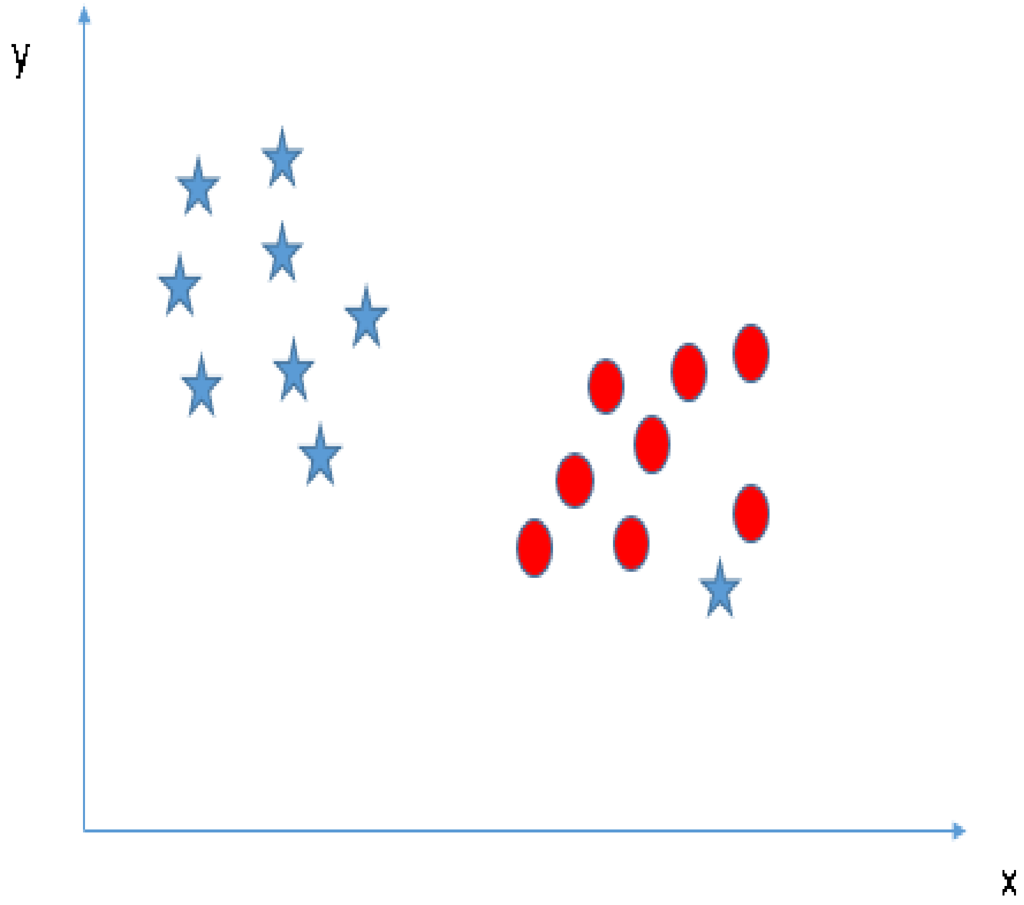
How does it work?

- **Can we classify two classes (Scenario-4)?:**
- It is unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier.



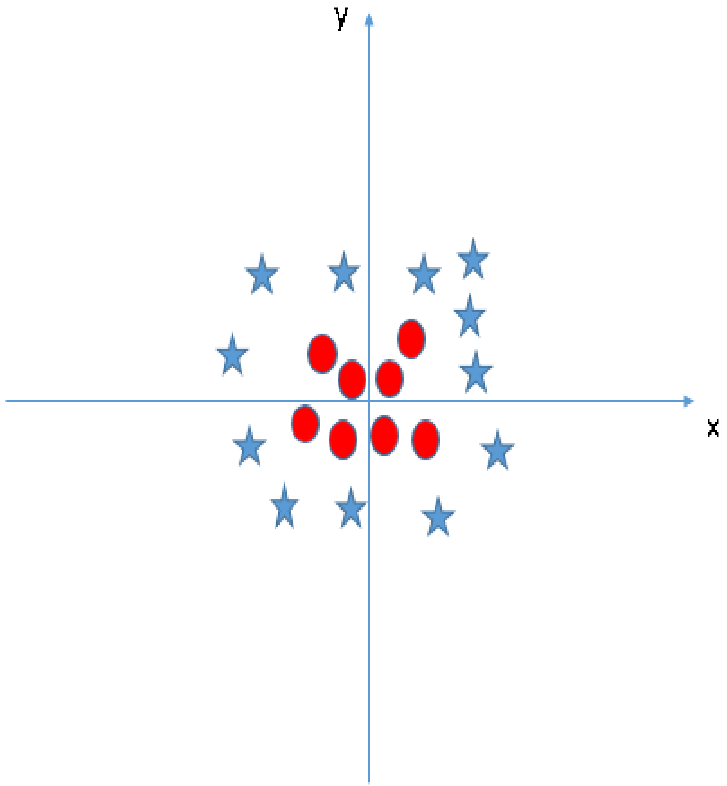
How does it work?

- **Can we classify two classes (Scenario-4)?:**
- As we already mentioned, one star at other end is like an outlier for star class.
- The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.



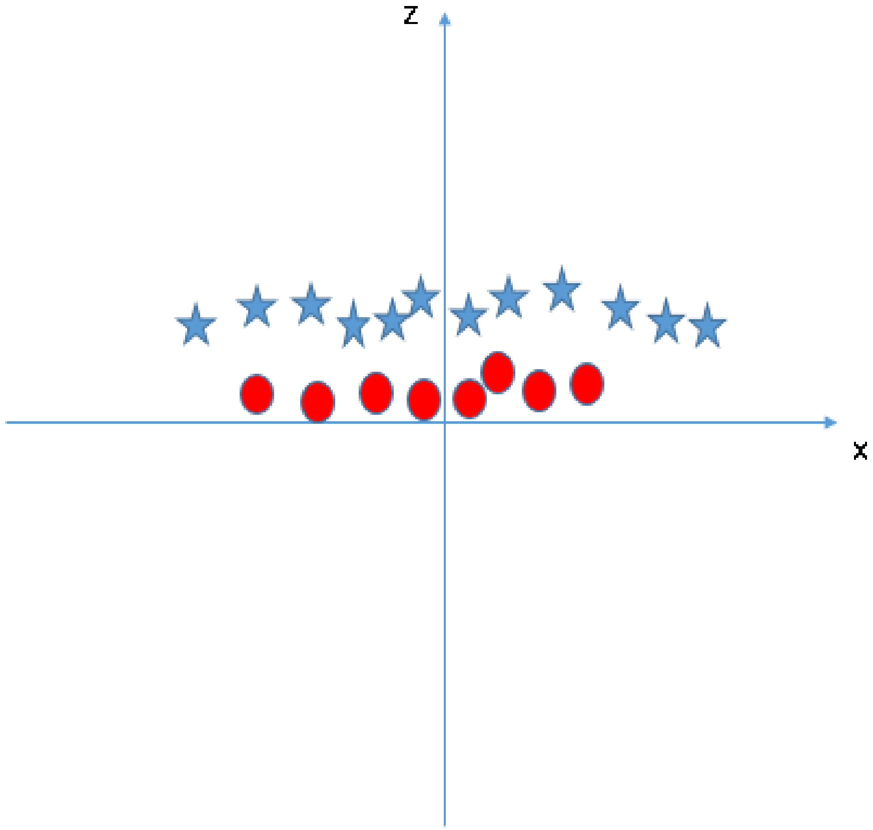
How does it work?

- **Find the hyper-plane to segregate to classes (Scenario-5):**
- In the scenario below, we can’t have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.
- SVM can solve this problem, easily! It solves this problem by introducing additional feature.
- Here, we will add a new feature z=x^2+y^2.
- Now, let’s plot the data points on axis x and z:



How does it work?

- **Find the hyper-plane to segregate to classes (Scenario-5):**
- In above plot, points to consider are:
- All values for z would be positive always because z is the squared sum of both x and y
- In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z.

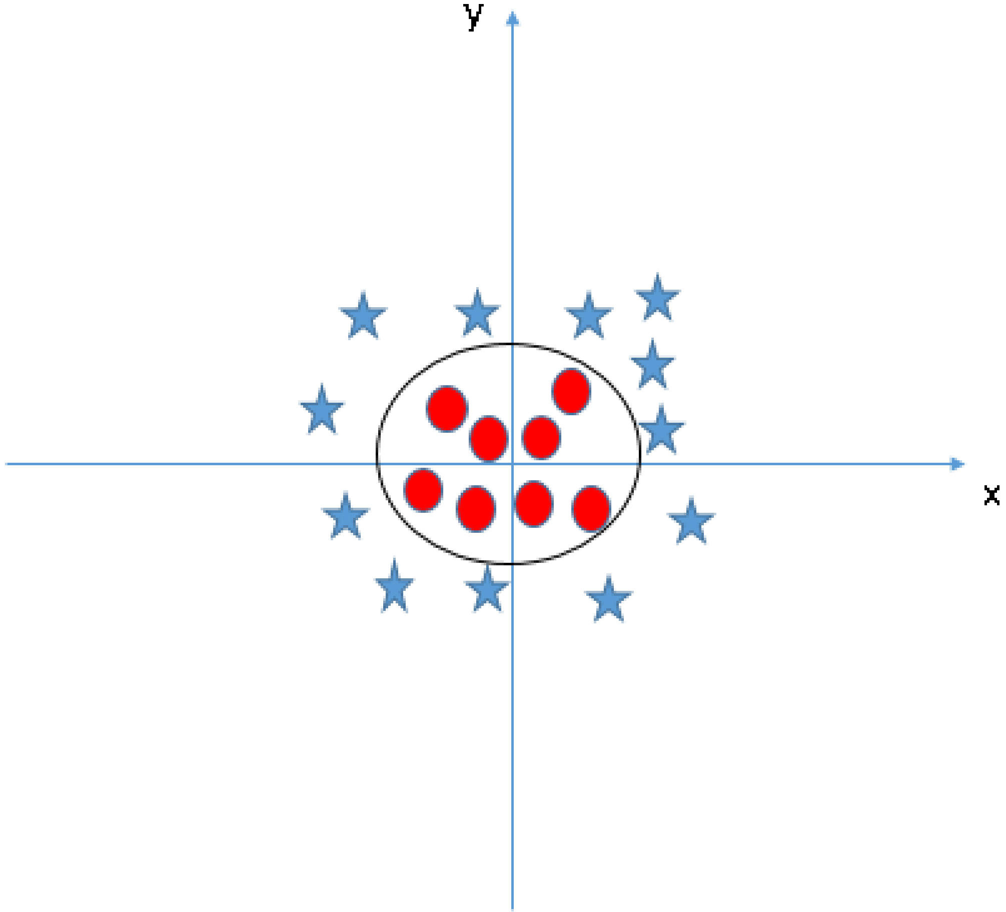


How does it work?

- In the SVM classifier, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane.
- No, the SVM algorithm has a technique called the kernel trick. The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem.

How does it work?

- It is mostly useful in non-linear separation problem.
- Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.
- When we look at the hyper-plane in original input space it looks like a circle:



How to implement SVM in Python and R?

- In Python, scikit-learn is a widely used library for implementing machine learning algorithms.
- SVM is also available in the scikit-learn library and we follow the same structure for using it(Import library, object creation, fitting model and prediction).

- Now, let us have a look at a real-life problem statement and dataset to understand how to apply SVM for classification

Problem Statement

- Dream Housing Finance company deals in all home loans.

- They have a presence across all urban, semi-urban and rural areas.
- A customer first applies for a home loan, after that the company validates the customer's eligibility for a loan.

- Company wants to automate the loan eligibility process (real-time) based on customer details provided while filling an online application form.

- These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others.
- 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.

Support Vector Machine(SVM) code in Python

- Use the coding window below to predict the loan eligibility on the test set. Try changing the hyper parameters for the linear SVM to improve the accuracy.

Support Vector Machine(SVM) code in R

- The e1071 package in R is used to create Support Vector Machines with ease.
- It has helper functions as well as code for the Naive Bayes Classifier.
- The creation of a support vector machine in R and Python follow similar approaches, let’s take a look now at the following code:

Pros and Cons associated with SVM

- Pros:
- It works really well with a clear margin of separation
- It is effective in high dimensional spaces.
- It is effective in cases where the 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.

Pros and Cons associated with SVM

- Cons:
- 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 included in the related SVC method of Python scikit-learn library.

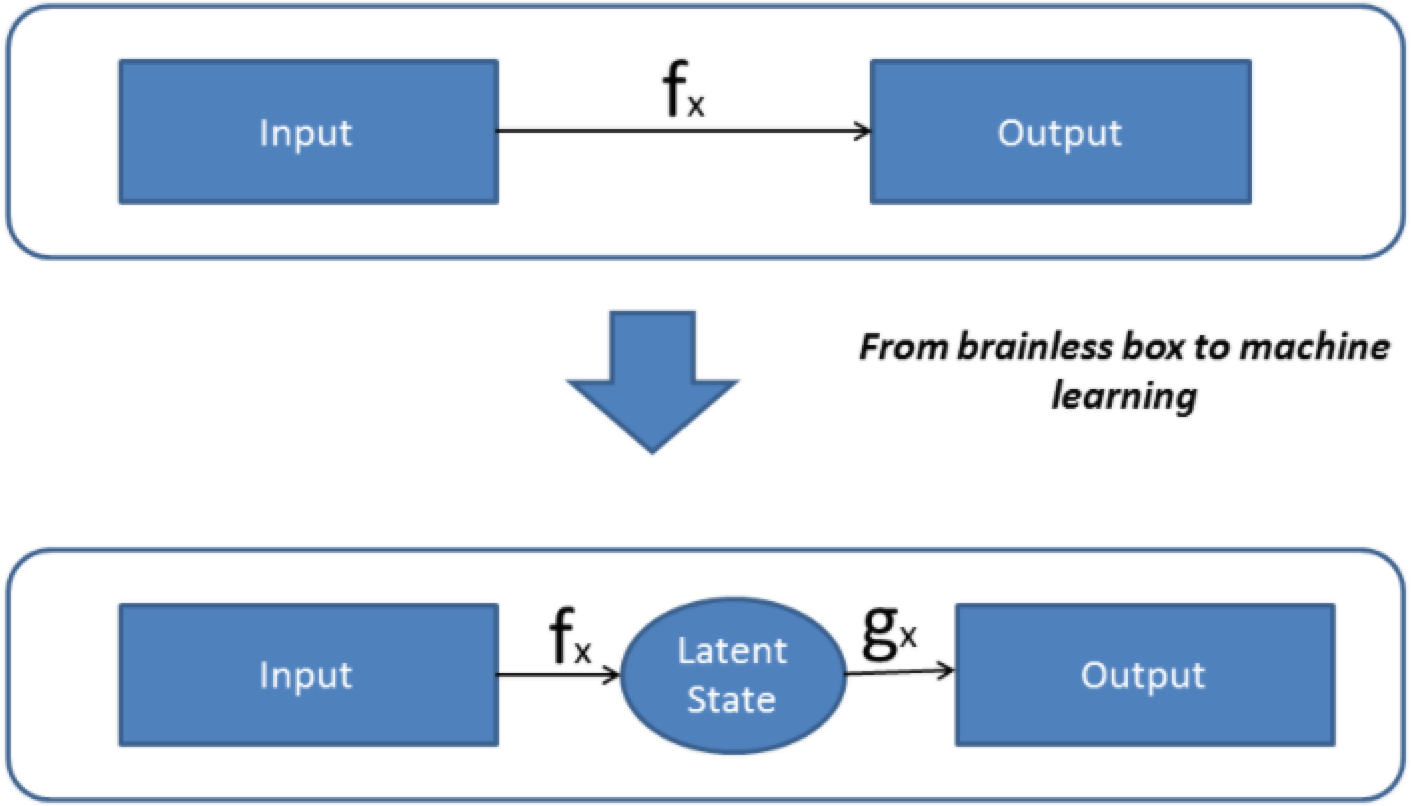
Introduction to Artificial Neural Network

- A simple machine is a set of algorithm, which converts input(s) to output(s)
- In this scenario, the same input will always lead to the same output.
- Human brain, on the other hand, has a unique characteristic of creating transient states through neurons in between the sensory organs and the brain (decision taking unit).
- Hence, the probabilistic interim state brings out a factor of randomness, which brings out what we call “Creativity”.

Introduction to Artificial Neural Network

- In ANN (Artificial neural network) or rather all machine learning algorithm, we build some kind of transient states, which allows the machine to learn in a more sophisticated manner.
- The objective here is to bring out the framework of ANN algorithm in parallel to the functionality of human brain.
- A single perceptron (or neuron) can be imagined as a Logistic Regression. Artificial Neural Network, or ANN, is a group of multiple perceptron's/ neurons at each layer.
- ANN is also known as a Feed-Forward Neural network because inputs are processed only in the forward direction:

Introduction to Artificial Neural Network



From brainless box to machine learning

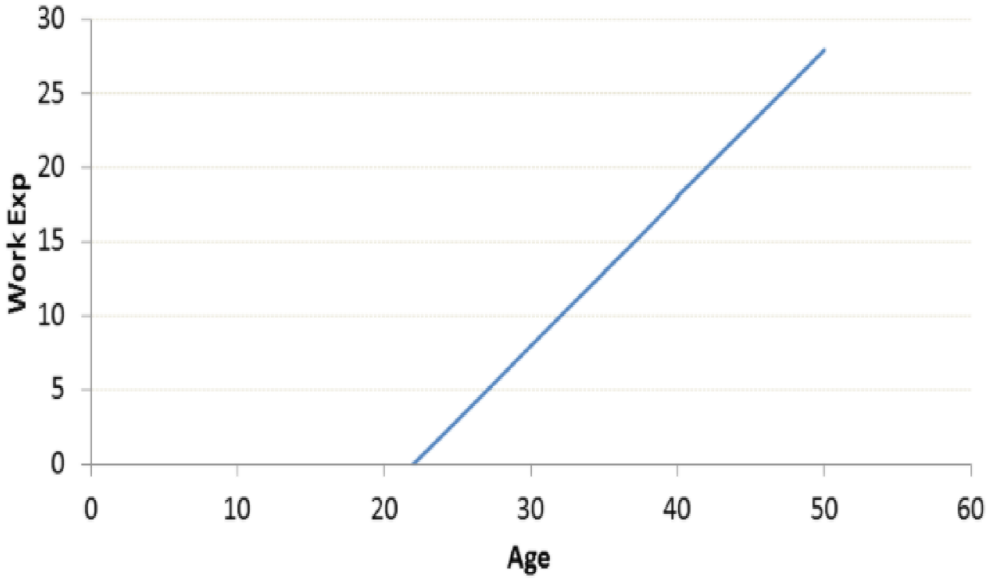
How does a simple predictive algorithm work?

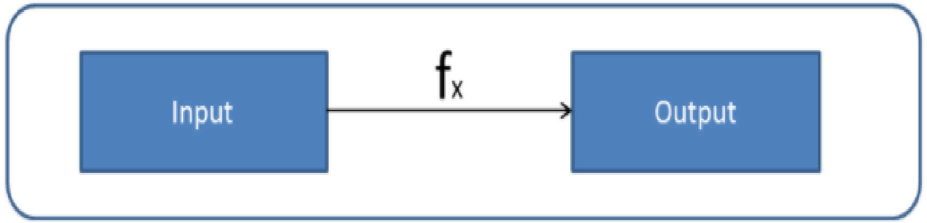
- A simple predictive algorithm tries to mimic the relationship between the

  Input and the output variables.

- The function derived in such routines is a direct linear or non-linear function between input and output variables.

- For instance, if we try to predict the total work experience of a person using his age, following is the kind of relationship we will observe:



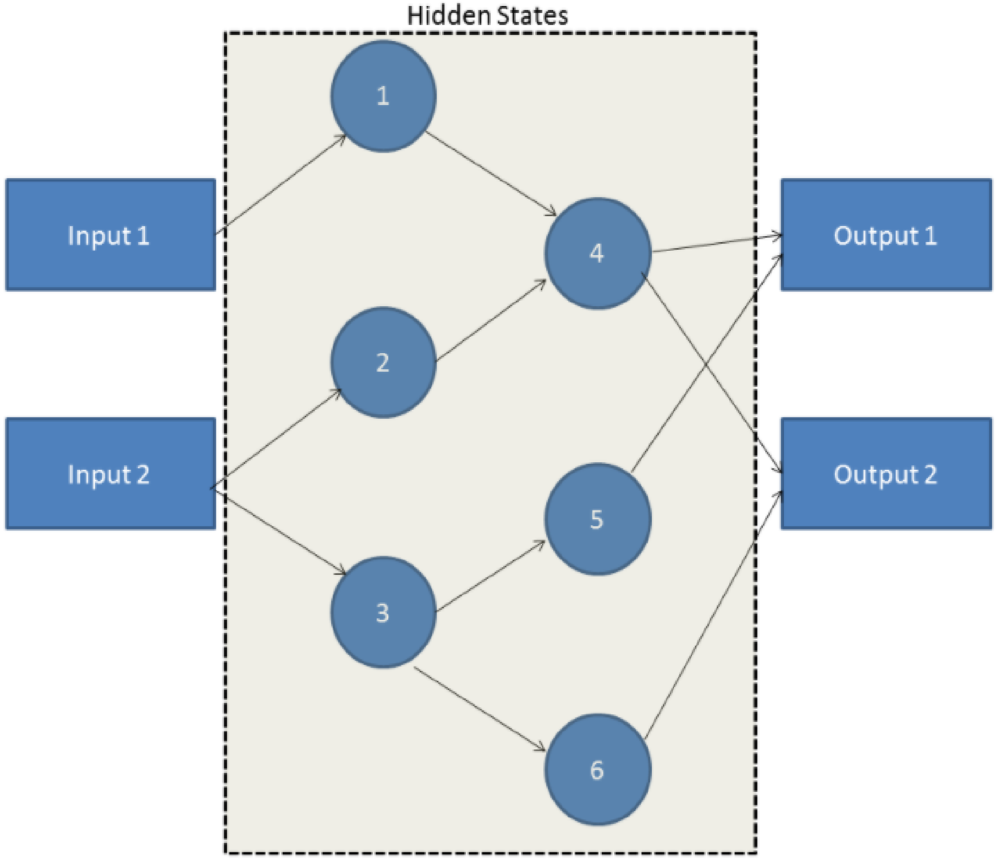


How does a simple predictive algorithm work?

- Relationships can easily be predicted using simple regression algorithms.
- But it becomes difficult to make predictions in case of complex non-linear relationships and significant covariate terms.
- In such cases, we need more sophisticated machine learning tools.
- To make such predictions, we have two options – either predict a complex non linear function or break this problem into multiple steps and solve for each step.
- The later can be achieved easily using an artificial neural network (ANN).

How does ANN work?

- It is truly said that the working of ANN takes its roots from the neural network residing in human brain.
- ANN operates on something referred to as Hidden State. These hidden states are similar to neurons. Each of these hidden state is a transient form which has a probabilistic behavior. A grid of such hidden state act as a bridge between the input and the output.



How does ANN work?

- Let’s try to understand what the diagram actually means.
- We have a vector of three inputs and we intend to find the probability that the output event will fall into class 1 or class 2.
- For this prediction we need to predict a series of hidden classes in between (the bridge). The vector of the three inputs in some combination predicts the probability of activation of hidden nodes from 1 – 4.
- The probabilistic combination of hidden state 1-4 are then used to predict the activation rate of hidden nodes 5-8. These hidden nodes 5-8 in turn are used to predict hidden nodes 9-12, which finally predicts the outcome.
- The intermediate latent states allows the algorithm to learn from every prediction.

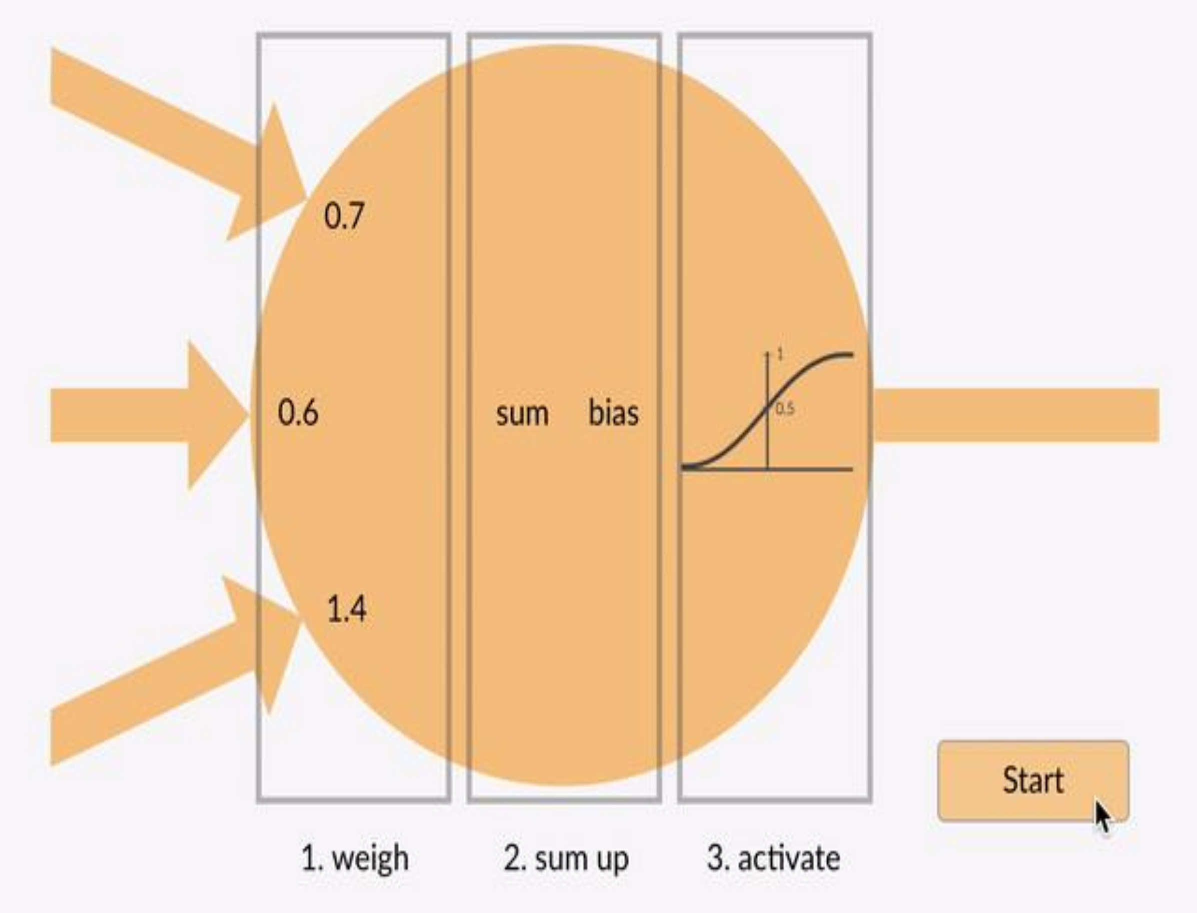
Advantages of Artificial Neural Network (ANN)

- Artificial Neural Network is capable of learning any nonlinear function.
- Hence, these networks are popularly known as Universal Function Approximators. ANNs have the capacity to learn weights that map any input to the output.

- One of the main reasons behind universal approximation is the activation function. Activation functions introduce nonlinear properties to the network.
- This helps the network learn any complex relationship between input and output.

Advantages of Artificial Neural Network (ANN)

- Here, the output at each neuron is the activation of a weighted sum of inputs.
- what happens if there is no activation function? The network only learns the linear function and can never learn complex relationships.
- An activation function is a powerhouse of ANN!

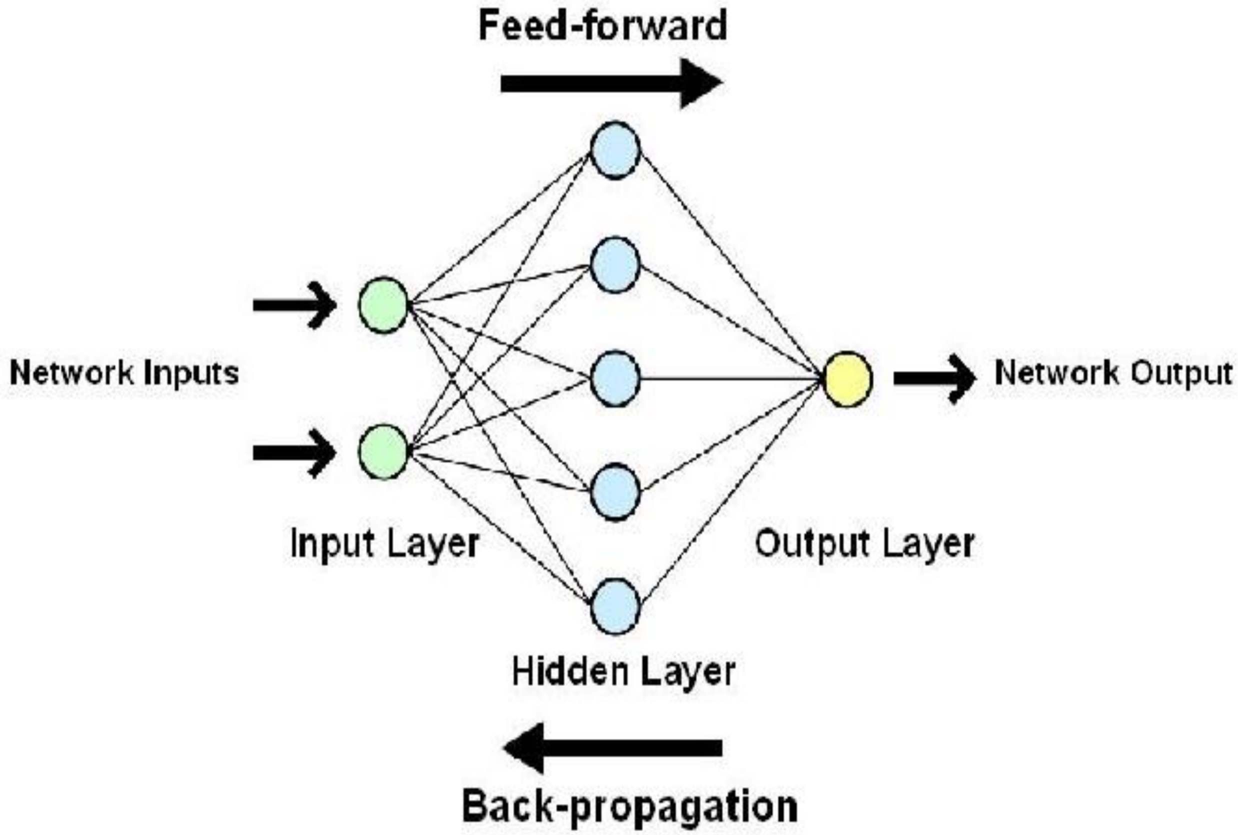


Challenges with Artificial Neural Network (ANN)

- While solving an image classification problem using ANN, the first step is to convert a 2-dimensional image into a 1-dimensional vector prior to training the model.
- This has two drawbacks:
- The number of trainable parameters increases drastically with an increase in the size of the image
- One common problem in all these neural networks is the Vanishing and Exploding Gradient.
- This problem is associated with the backpropagation algorithm.

Challenges with Artificial Neural Network (ANN)

- The weights of a neural network are updated through this backpropagation algorithm by finding the gradients:
- So, in the case of a very deep neural network (network with a large number of hidden layers), the gradient vanishes or explodes as it propagates backward which leads to vanishing and exploding gradient.
- ANN cannot capture sequential information in the input data which is required for dealing with sequence data



Data-Driven Approach

- If we want to challenge, to consider abandoning heuristics and best practices and take on a data-driven approach to algorithm selection.

- Rather than picking your favorite algorithm, try 10 or 20 algorithms.

- Double down on those that show signs of being better in performance, robustness, speed or whatever concerns interest you most.

- Rather than picking the common parameters, grid search tens, hundreds or thousands of combinations of parameters.

- Become the objective scientist, leave behind anecdotes and study the intersection of complex learning systems and data observations from your problem domain.

Data-Driven Approach in Action

- This is a powerful approach that requires less up-front knowledge, but a lot more back-end computation and experimentation.

- As such, it will be very likely be required to work with a smaller sample of your dataset so that you can get results quickly.
- We can have a test harness that we can have complete faith in.

- Note: how can you have complete trust in your test harness?

- You develop trust by selecting the test options in a data-driven manner that gives you objective confidence that your chosen configuration is reliable.

- The type of estimation method (split, boosting, k-fold cross validation, etc.) and it’s configuration (size of k, etc.).

Leverage Automation

- The data-driven approach is a problem of search.
- we can leverage automation.
- You can write re-usable scripts to search the for the most reliable test harness for our problem before we begin. No more ad hoc guessing.
- We can write a reusable script to try automatically 10, 20, 100 algorithms across a variety of libraries and implementations. No more favorite algorithms or libraries.
- The line between different algorithms is gone and a new parameter configuration is a new algorithm. we can write re-usable scripts to grid or random search each algorithm to truly sample its capability.
- Add feature engineering on the front so that each "view" on the data is a new problem for algorithms to be challenged against.
- Bolt-on ensembles at the end to combine some or all results (meta-algorithms).

Summary on Data-Driven Approach

- In this data-driven approach, we have looked at the common heuristic and best-practice approach to algorithm and algorithm parameter selection.
- We have considered that this approach leads to limitations in our thinking.
- We yearn for silver bullet general purpose best algorithms and best algorithm configurations, when no such things exist.
- There is no best general purpose machine learning algorithm.
- There are no best general purpose machine learning algorithm parameters.
- The transferability of capability for an algorithm from one problem to another is questionable.
- The solution is to become the scientist and to study algorithms on our problems.
- We must take a data-driven problem, to spot check algorithms, to grid search algorithm parameters and to quickly find methods that yield good results, reliably and fast.

References

Text Book:

“Business Analytics, The Science of Data-Driven Making”, U. Dinesh Kumar, Wiley 2017

“Recommender Systems, The text book, Charu C. Aggarwal, Springer 2016 Section 1.and Section 2.

Image Courtesy

https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

https://www.analyticsvidhya.com/blog/2014/10/introduction-neural-network-simplified/

THANK YOU

Jyothi R.

Assistant Professor,

Department of Computer Science

jyothir@pes.edu