Data Mining & Knowledge Discovery

Classification - 05

Md. Biplob Hosen

Lecturer, IIT-JU

Email: biplob.hosen@juniv.edu

Logistic Regression

- Logistic Regression is a statistical method used for binary classification, which involves predicting one of two possible outcomes or classes.
- Despite its name, logistic regression is a classification algorithm rather than a regression algorithm used for predicting continuous values.
- The primary goal of logistic regression is to model the probability that an instance belongs to a certain class based on one or more input features.
- It's particularly useful when dealing with problems where the response variable is categorical (e.g., yes/no, true/false, 0/1) and you want to understand the relationship between the predictor variables and the likelihood of an event occurring.

Continue…

- **Sigmoid Function (Logistic Function):** Logistic regression uses the sigmoid function to transform the output of a linear equation into a value between 0 and 1. The sigmoid function is defined as:

$$S(z) = \frac{1}{1 + e^{-z}}$$

  - where z is the linear combination of the input features and their respective weights.
- **Linear Combination:** The linear combination of input features and their associated weights is calculated as:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

Here, $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients (weights) associated with the features $x_0, x_1, \ldots, x_n$, where x_0 is usually set to 1 (to account for the intercept).

- **Probability Prediction:** The sigmoid function takes the linear combination z as input and maps it to a value between 0 and 1. This value represents the probability that the given instance belongs to the positive class (class 1). The probability of belonging to the negative class (class 0) is simply 1−P (positive class)

Continue…

- **Decision Boundary:** To make a classification decision, you can choose a threshold value (often 0.5) for the predicted probability. If the predicted probability is above the threshold, the instance is classified as the positive class; otherwise, it's classified as the negative class.
- **Training:** The logistic regression model is trained using a method called maximum likelihood estimation.
- The goal is to find the coefficients $\beta_0, \beta_1, \ldots, \beta_n$ that maximize the likelihood of the observed outcomes given the inputs.

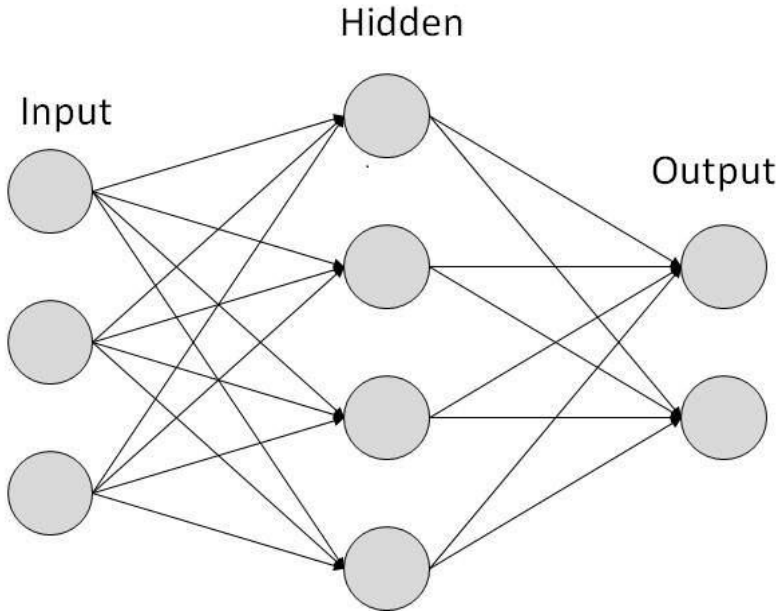
- Logistic regression can be extended to handle multi-class classification using techniques like **softmax regression**.

Artificial Neural Network (ANN)

- An Artificial Neural Network (ANN) is a computational model inspired by the structure and functioning of the human brain. It is a machine learning algorithm that is particularly well-suited for tasks such as pattern recognition, classification, regression, and more complex tasks involving large amounts of data.
- An ANN consists of interconnected nodes, or "neurons," organized into layers. Each neuron performs a simple computation, and the connections between neurons are characterized by weights. The architecture of an ANN typically includes the following components:
- **Input Layer:** The input layer receives the raw data or features from the dataset. Each neuron in this layer corresponds to a feature in the input data.
- **Hidden Layers:** Hidden layers are intermediate layers between the input and output layers. They process and transform the data using weighted connections and activation functions. The number of hidden layers and the number of neurons in each hidden layer are design choices that can affect the network's performance.

Continue…

- **Output Layer:** The output layer produces the final prediction or output of the network. The number of neurons in this layer depends on the nature of the task. For binary classification, there is usually one neuron with a sigmoid activation function, while for multi-class classification, the number of output neurons matches the number of classes, often using a softmax activation function.
- **Connections and Weights:** Each neuron in a layer is connected to every neuron in the previous and subsequent layers. Each connection has an associated weight, which is learned during the training process. The weights determine the strength of the influence that one neuron's output has on another neuron's input.
- **Activation Functions:** Activation functions introduce non-linearity to the network, enabling it to model complex relationships in the data. Common activation functions include the sigmoid, ReLU (Rectified Linear Unit), tanh (hyperbolic tangent), and softmax functions.



Continue…

The process of training an ANN involves the following steps:

- **Initialization:** Initialize the weights and biases of the network randomly or using specific techniques to ensure proper convergence.
- **Forward Propagation:** Calculate the weighted sum of inputs for each neuron, apply the activation function, and pass the outputs to the next layer.
- **Loss Function:** Define a loss function that measures the difference between the network's predictions and the actual target values.
- **Backpropagation:** Calculate the gradient of the loss with respect to the weights using techniques like the chain rule. Update the weights in the opposite direction of the gradient to minimize the loss.
- **Optimization:** Use optimization algorithms (e.g., gradient descent) to adjust the weights iteratively, aiming to minimize the loss function.
- **Validation and Testing:** Evaluate the trained network on a validation dataset to monitor its performance and prevent overfitting. Test the final model on unseen data to assess its generalization ability.
- ANNs have shown remarkable success in various domains, including image and speech recognition, natural language processing, recommendation systems, and more.

Deep Learning

- Deep Learning is a subfield of machine learning that focuses on using artificial neural networks with multiple layers (hence the term "deep") to model and solve complex problems. Deep Learning has revolutionized various fields, including computer vision, natural language processing, speech recognition, and even game playing.
- Key characteristics of Deep Learning include:
- **Deep Neural Networks:** Deep Learning models typically involve neural networks with many hidden layers. These networks are capable of automatically learning hierarchical features from the data, allowing them to capture intricate patterns and representations.
- **Representation Learning:** Deep Learning aims to learn feature representations directly from the raw data, reducing the need for manual feature engineering. This is achieved through multiple layers of abstraction, where each layer learns increasingly complex and abstract features.
- **End-to-End Learning:** Deep Learning models often learn to perform a complete task from start to finish, without relying on intermediate steps or explicit rules. For example, in image classification, a deep neural network can learn to directly map raw pixel values to class labels.

Continue…

- **Scalability:** Deep Learning models can handle large and complex datasets. The availability of powerful hardware (such as GPUs and TPUs) and efficient training algorithms enables the training of deep networks on massive amounts of data.
- **Complex Architectures:** Deep Learning encompasses a variety of architectures beyond traditional feedforward neural networks, including Convolutional Neural Networks (CNNs) for image analysis, Recurrent Neural Networks (RNNs) for sequential data, and Transformers for natural language processing.
- **Unsupervised Learning:** Deep Learning includes techniques for unsupervised learning, where networks can learn meaningful representations from unlabeled data. Autoencoders and Generative Adversarial Networks (GANs) are examples of unsupervised learning techniques used in Deep Learning.
- **Transfer Learning:** Pretrained deep models can be used as a starting point for new tasks, allowing the transfer of learned knowledge from one domain to another. This has proven effective in scenarios with limited labeled data.
- **Challenges and Advances:** Deep Learning is associated with challenges such as overfitting, vanishing gradients, and the need for large amounts of data. Advances in architectures (e.g., residual networks), regularization techniques (e.g., dropout), and optimization algorithms (e.g., Adam) have contributed to addressing these challenges.

Continue…

Applications of Deep Learning:

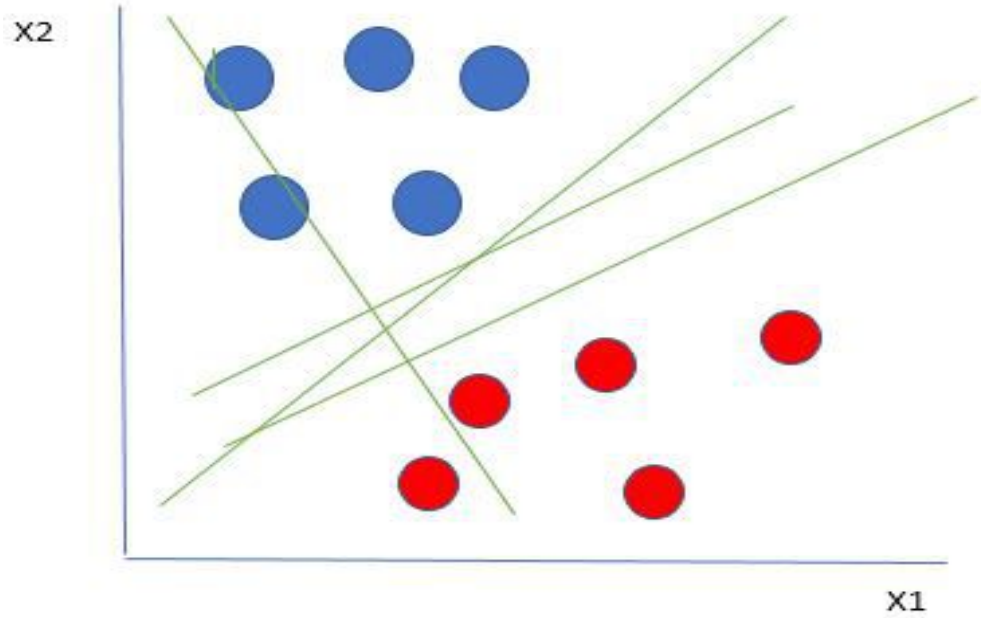
- **Image and Video Analysis:** Object detection, image segmentation, facial recognition, style transfer, and autonomous driving.
- **Natural Language Processing:** Machine translation, sentiment analysis, text generation, chatbots, and language modeling.
- **Speech and Audio Recognition:** Speech-to-text conversion, speaker identification, and music generation.
- **Healthcare:** Medical image analysis, disease detection, drug discovery, and personalized medicine.
- **Finance:** Stock price prediction, fraud detection, and algorithmic trading.
- **Gaming:** AI agents for playing games like chess, Go, and video games.

Support Vector Machine (SVM)

- Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression.
- The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space.
- The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible.
- The dimension of the hyperplane depends upon the number of features.
- If the number of input features is two, then the hyperplane is just a line.
- If the number of input features is three, then the hyperplane becomes a 2-D plane.
- It becomes difficult to imagine when the number of features exceeds three.

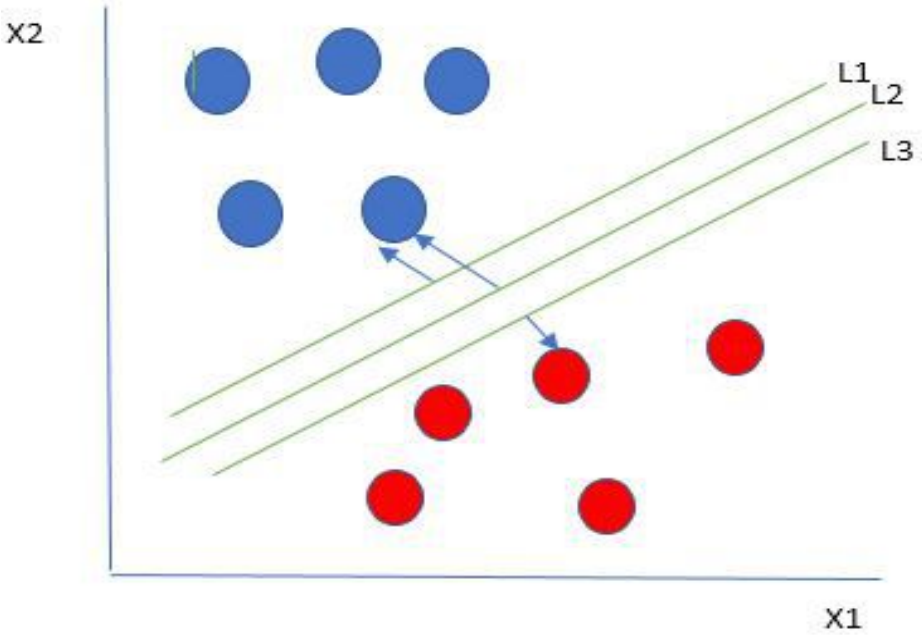
Continue…

- From the figure above it's very clear that there are multiple lines (our hyperplane here is a line because we are considering only two input features x_1, x_2) that segregate our data points or do a classification between red and blue circles. So how do we choose the best line or in general the best hyperplane that segregates our data points?



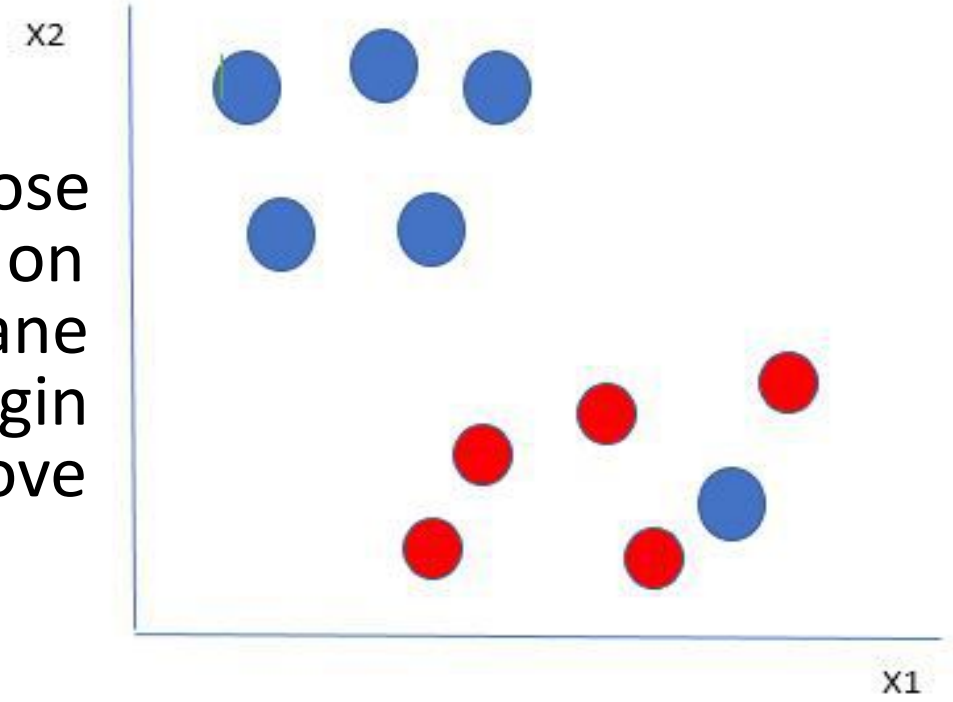
How does SVM work?

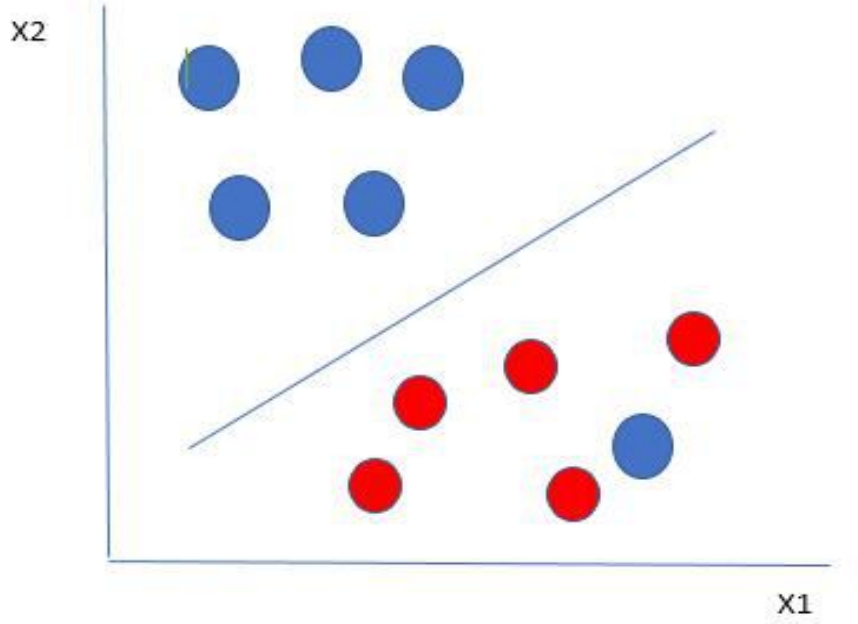
- One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the two classes.



Continue…

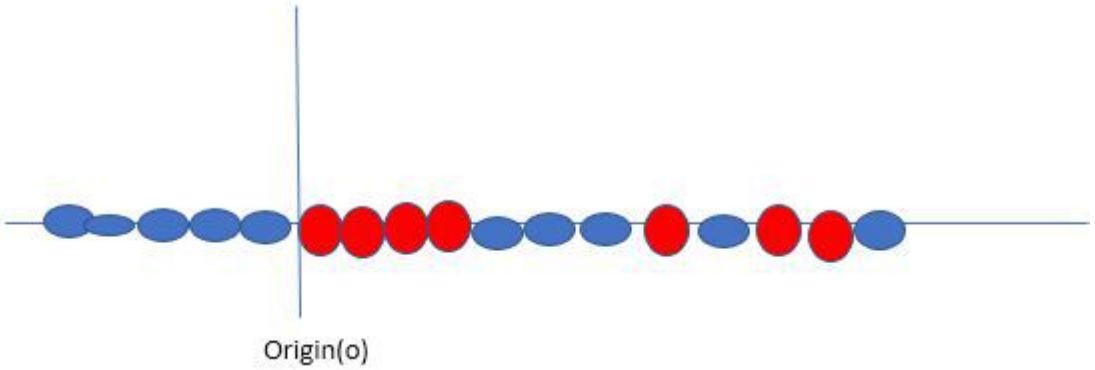
- So we choose the hyperplane whose distance from it to the nearest data point on each side is maximized. If such a hyperplane exists it is known as the maximum-margin hyperplane/hard margin. So from the above figure, we choose L2.
- Let’s consider another scenario.
- Here we have one blue ball in the boundary of the red ball. So how does SVM classify the data? It’s simple! The blue ball in the boundary of red ones is an outlier of blue balls. The SVM algorithm has the characteristics to ignore the outlier and finds the best hyperplane that maximizes the margin. SVM is robust to outliers.

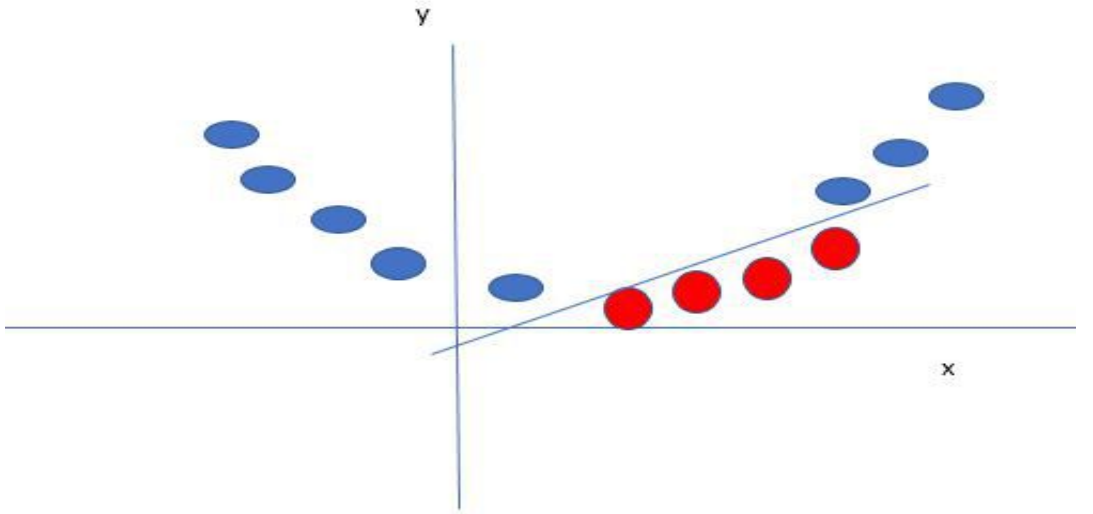




Continue…

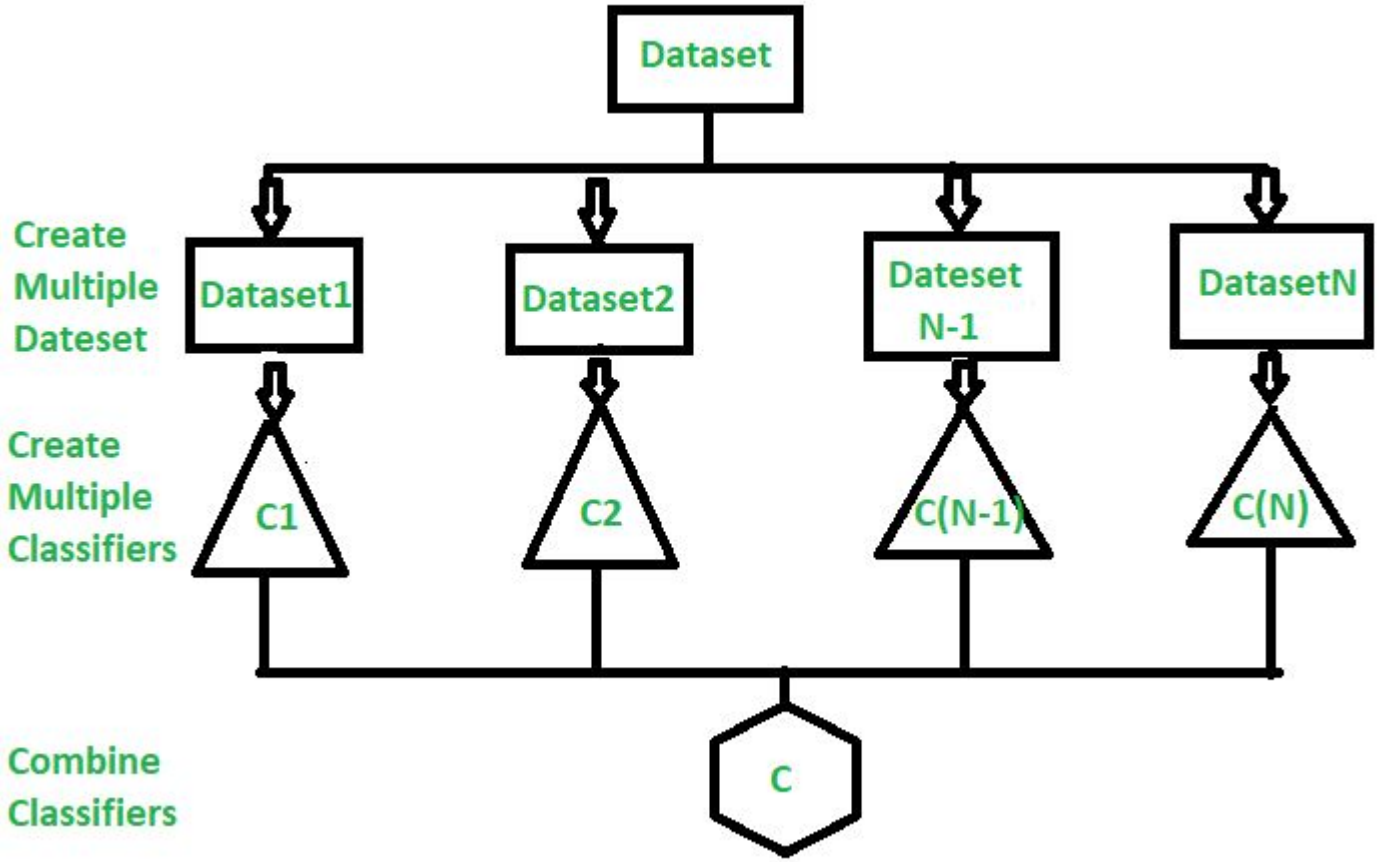
- Till now, we were talking about linearly separable data(the group of blue balls and red balls are separable by a straight line/linear line). What to do if data are not linearly separable?
- Say, our data is shown in the figure above. SVM solves this by creating a new variable using a kernel. We call a point xi on the line and we create a new variable yi as a function of distance from origin o.so if we plot this we get something like as shown below.
- In this case, the new variable y is created as a function of distance from the origin. A non-linear function that creates a new variable is referred to as a kernel.





Ensemble Methods

- Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Basic idea is to learn a set of classifiers (experts) and to allow them to vote.



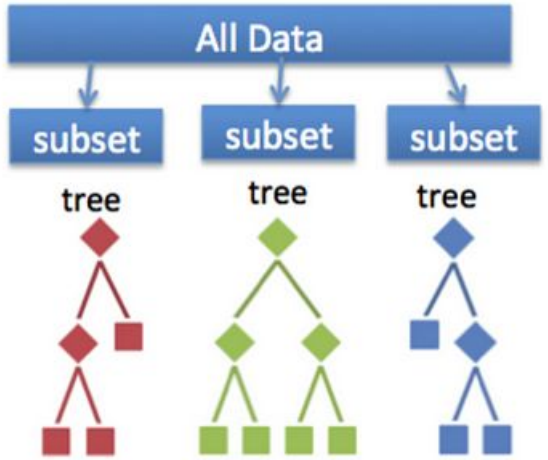
Continue…

Example of Ensemble Classifier:

- **Bagging:** It involves training multiple instances of the same base model on different subsets of the training data. Each subset is obtained through random sampling with replacement (bootstrapping). The final prediction is typically the average (for regression) or majority vote (for classification) of the predictions from individual models. Random Forest is a well-known bagging-based ensemble method that uses decision trees as base models.

Implementation steps of Random Forest:

1) Multiple subsets are created from the original data set, selecting observations with replacement.
2) A subset of features is selected randomly and whichever feature gives the best split is used to split the node iteratively.
3) The tree is grown to the largest.
4) Repeat the above steps and prediction is given based on the aggregation of predictions from n number of trees.



Class Imbalance Problem

What is the Class Imbalance Problem?

- It is the problem in machine learning where the total number of a class of data (positive) is far less than the total number of another class of data (negative). This problem is extremely common in practice and can be observed in various disciplines including fraud detection, anomaly detection, medical diagnosis, oil spillage detection, facial recognition, etc.

Why is it a problem?

- Most machine learning algorithms and works best when the number of instances of each classes are roughly equal. When the number of instances of one class far exceeds the other, problems arise. This is best illustrated below with an example.
- Given a dataset of transaction data, we would like to find out which are fraudulent and which are genuine ones. Now, it is highly cost to the e-commerce company if a fraudulent transaction goes through as this impacts our customers trust in us, and costs us money. So we want to catch as many fraudulent transactions as possible.
- If there is a dataset consisting of 10000 genuine and 10 fraudulent transactions, the classifier will tend to classify fraudulent transactions as genuine transactions. The reason can be easily explained by the numbers. Suppose the machine learning algorithm has two possibly outputs as follows:

Continue…

1) Model 1 classified 7 out of 10 fraudulent transactions as genuine transactions and 10 out of 10000 genuine transactions as fraudulent transactions.
2) Model 2 classified 2 out of 10 fraudulent transactions as genuine transactions and 100 out of 10000 genuine transactions as fraudulent transactions.

- If the classifier's performance is determined by the number of mistakes, then clearly Model 1 is better as it makes only a total of 17 mistakes while Model 2 made 102 mistakes. However, as we want to minimize the number of fraudulent transactions happening, we should pick Model 2 instead which only made 2 mistakes classifying the fraudulent transactions. Of course, this could come at the expense of more genuine transactions being classified as fraudulent transactions, but will be a cost we can bear for now. Anyhow, a general machine learning algorithm will just pick Model 1 than Model 2, which is a problem. In practice, this means we will let a lot of fraudulent transactions go through although we could have stopped them by using Model 2. This translates to unhappy customers and money lost for the company.

Continue…

How to mitigate this problem?

1) **Cost function based approaches:** The intuition behind cost function based approaches is that if we think one false negative is worse than one false positive, we will count that one false negative as, e.g., 100 false negatives instead. For example, if 1 false negative is as costly as 100 false positives, then the machine learning algorithm will try to make fewer false negatives compared to false positives (since it is cheaper).
2) **Sampling based approaches:** This can be roughly classified into three categories:
    1) Oversampling, by adding more of the minority class so it has more effect on the machine learning algorithm.
    2) Undersampling, by removing some of the majority class so it has less effect on the machine learning algorithm.
    3) Hybrid, a mix of oversampling and undersampling.

Thank You!