The supervised learning workshop

Chapter 1 : Fundamentals of Supervised Learning Algorithms

Introduction

In the context of our project we worked on the first chapter of the book "The Supervised Learning Workshop"

In this chapter we will address the fundamentals of supervised learning algorithms through 2 main parts :

- Problem presentation
- Solutions available for Supervised Learning

There will be no part 3 which concerns the demonstration because there is no case of implementation in this first chapter of the book.

Part 1 : Problem presentation

Problem : What is supervised learning ?

Definitions and fundamentals of supervised learning

Definition Supervised Learning

- Supervised learning = type of machine learning where an algorithm learns to map input data to output data by being trained on labeled examples
- Input data = features or predictors
- This book "Supervised Learning Workshop" explains that supervised learning is a powerful technique that has many practical applications, such as image recognition, speech recognition, natural language processing, and predictive modeling
- The algorithm is given a dataset consisting of input-output pairs, and it tries to learn the relationship between the input and output by finding a function that maps the input to the output. It is trained by adjusting its parameters to minimize the difference between its predicted outputs and the true outputs. This technique has become **increasingly popular and important** due to the large **amounts of data** being generated in various fields

The fundamentals of Supervised Learning

- **Labeled data :** In supervised learning, the dataset used to train the algorithm must be labeled, meaning that each input data point is associated with a corresponding output label. The labeled data is used to train the algorithm to learn the relationship between the input and output

- **Feature engineering :** The process of selecting and transforming the input features is a crucial step in supervised learning. Feature engineering is essential for supervised learning to improve performance.

- **Model selection :** Choosing the appropriate model for the task at hand is another important consideration in supervised learning. Exists several algorithms available = linear regression, logistic regression, decision trees, random forests, and neural networks, each with its own strengths and weaknesses

- **Training and evaluation :** Once a model has been selected, it is trained on the labeled data using an appropriate optimization algorithm. The performance of the model is then evaluated on a separate dataset, known as the validation or test set, to determine how well it can generalize to new, unseen data

- **Hyperparameter tuning :** Parameters that control behavior of algorithm (E.g. leaning rate or number of hidden layers in neural network). They have can have a great impact on performance of the model

Fundamentals of Supervised Learning =

selecting & transforming input features

choosing appropriate model

training & evaluating model on labeled data

tuning hyperparameters

Comparison between Supervised - Unsupervised and highlights of the benefits

Supervised learning = Learning a mapping function from labeled training data. The input data is accompanied by the correct output, and the aim of the algorithm is to learn the relationship between the input and output data. The algorithm then uses this relationship to make predictions on new, unseen data. Supervised learning is ***useful in*** applications where there is a clear target variable (E.g. predicting the price of a house based on its features)

Unsupervised learning = Finding patterns or structure in unlabeled data. Unlike supervised learning, unsupervised learning does not use a labeled dataset to learn from, but instead the algorithm attempts to identify structure in the data based on some criteria, such as clustering or dimensionality reduction. Unsupervised learning is ***useful in*** applications where the goal is identifying patterns in data (E.g. identifying customers with similar behavior)

Examples

Supervised learning : To build a model in order to predict the price of a house based on its features (E.g. number of bedrooms, square footage, location) = we collect dataset of labeled examples with the features of a house and corresponding price.

We would use this labeled data to train a supervised learning algorithm, which would learn the relationship between the features and the price, and use that relationship to make predictions on new, unseen data.

Unsupervised learning example : We are given a dataset of customer purchase histories at a grocery store. We do not have any labels indicating which customers belong to which groups, but we want to identify groups of customers with similar purchasing behavior. We could use an unsupervised learning algorithm, such as k-means clustering, to identify groups of customers based on the patterns in their purchase history.

The algorithm will allow us to identify segments of customers with distinct preferences and behaviors

In summary

Supervised learning is used when there is a clear target variable and the goal is to learn a mapping function from labeled data to make predictions on new data, while unsupervised learning is used when the goal is to identify patterns or structure in unlabeled data

Benefits of adressing this problem

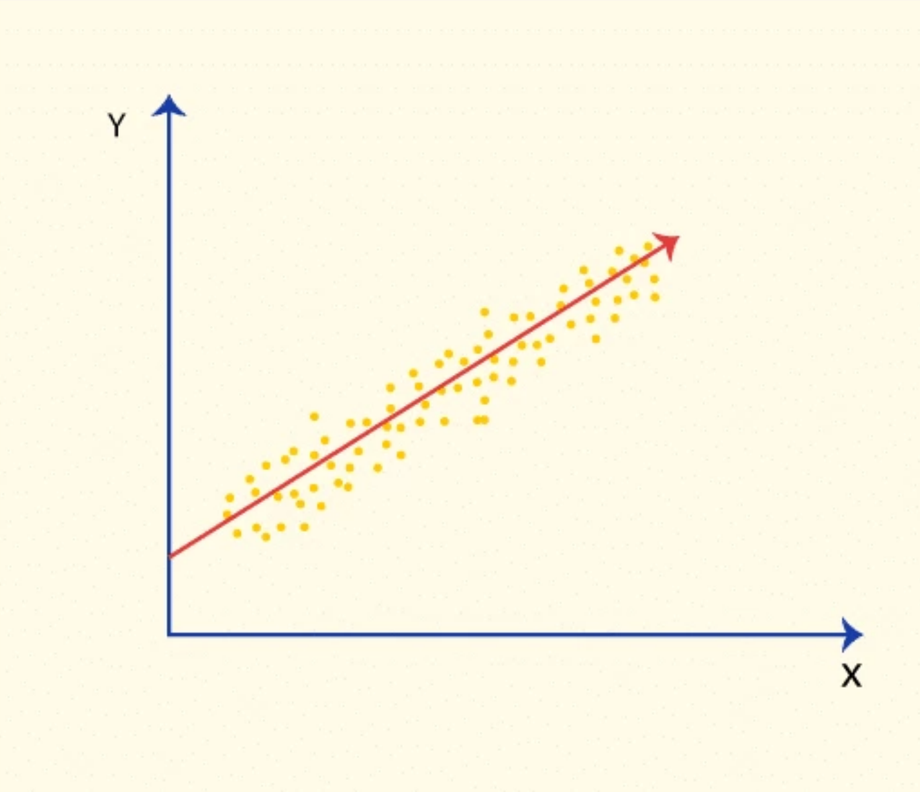
- Improved accuracy and efficiency
- Real-world applications
- Advancing the field of machine learning
- Economic benefits

To sum up, knowing the principles of supervised learning is advantageous for programmers, researchers, and society at large, resulting in better algorithm accuracy and efficiency, practical applications, developments in machine learning, and financial gains

Part 2 : Solutions available for Supervised Learning

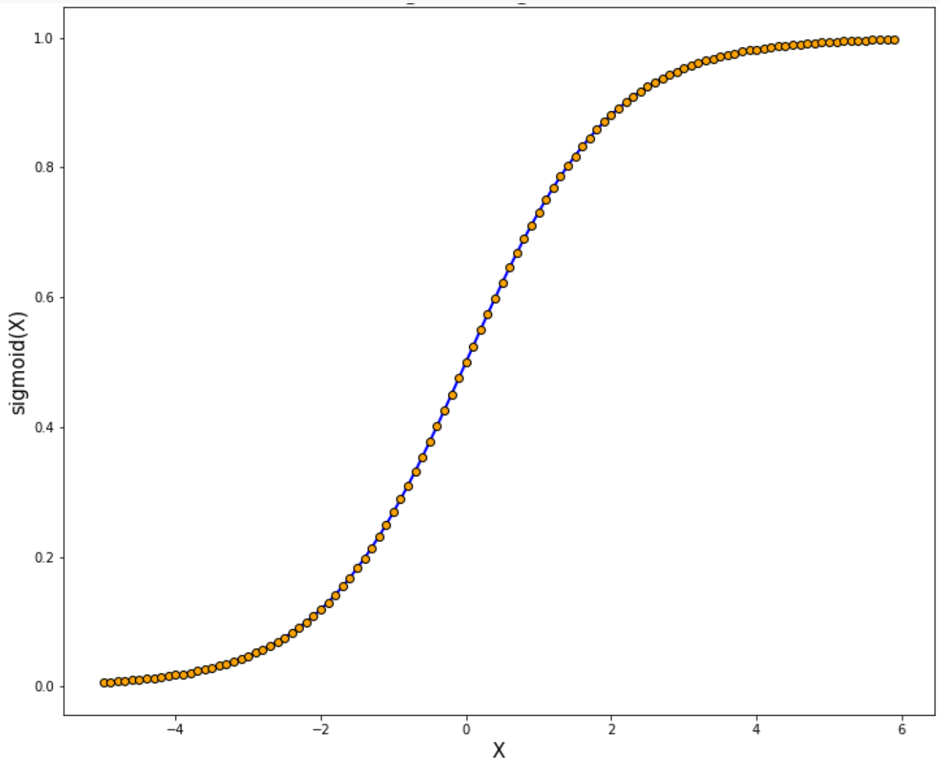
Classification & Regression Models

Linear regression



Linear regression is a supervised learning algorithm that is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable. This type of analysis estimates the coefficients of the linear equation, which involves one or more independent variables that best predict the value of the dependent variable. Linear regression fits a straight line (figure on the left) or surface that minimizes the discrepancies between predicted and actual output values.

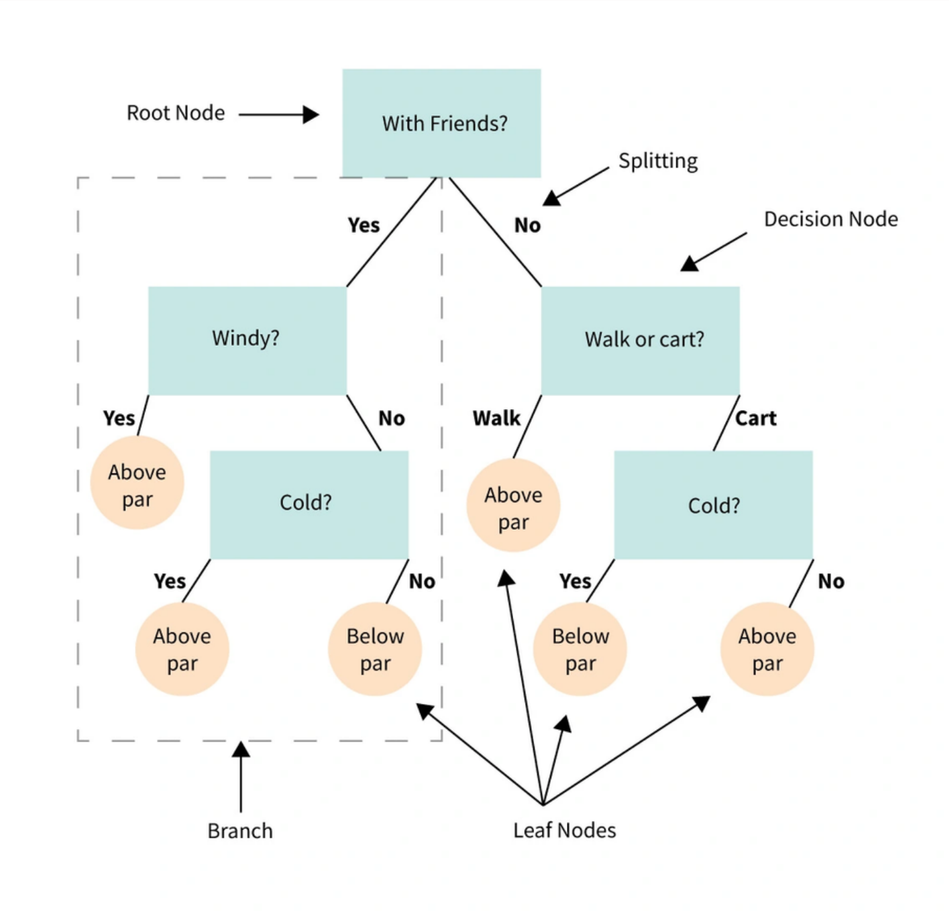
Logistic regression



Similar to linear regression, logistic regression is also used to estimate the relationship between a dependent variable and one or more independent variables, but the difference is that logistic regression is used to make predictions about a categorical variable versus a continuous variable. A categorical variable can be true or false, yes or no, 1 or 0, etc. The unit of measure also differs from linear regression in that it produces a probability, whereas the logit function turns the S-curve into a straight line (as shown in the figure). Although both models are used in regression analysis to make predictions about future outcomes, linear regression is generally easier to understand and does not require a sample as large as logistic regression, which needs an adequate sample to represent the values of all response categories. Without a larger, representative sample, the model may not have sufficient statistical power to detect a significant effect.

Decision trees

A decision tree is a type of supervised machine learning used to categorize or make predictions based on the answers given to a set of previous questions.



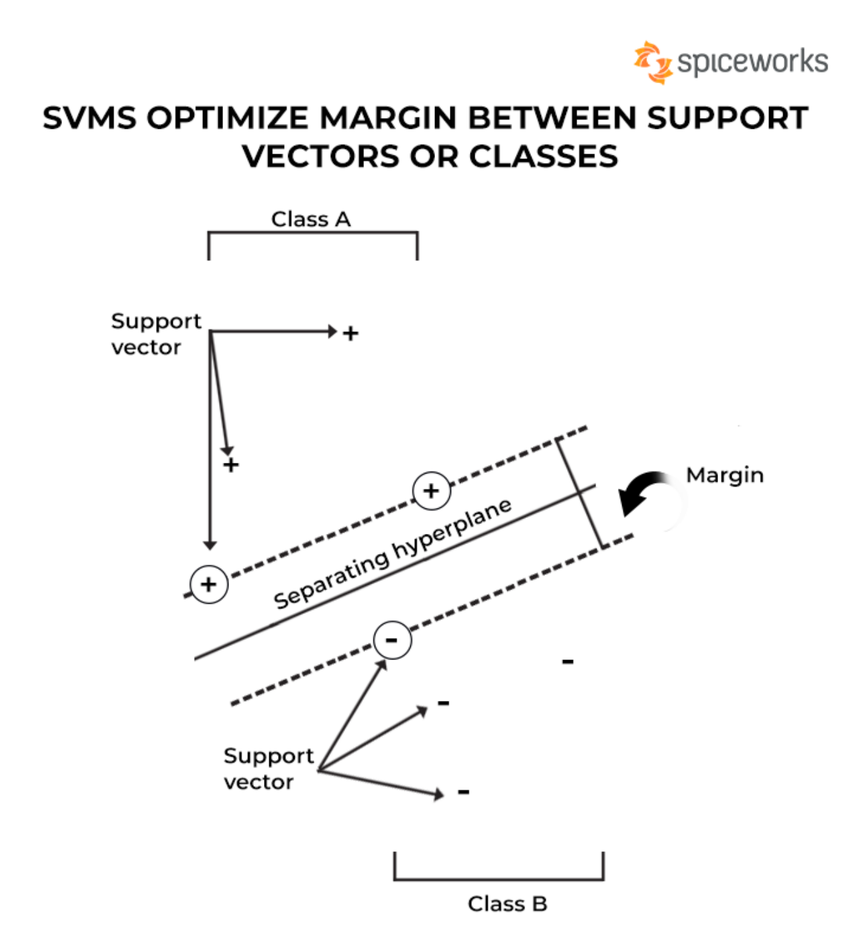
Source: Master's in Data Science

A decision tree looks like a tree. The base of the tree is the root node. The root node is followed by a series of decision nodes that describe the decisions to be made. The decision nodes are followed by leaf nodes that represent the consequences of those decisions. Each decision node represents a question or separation point, and the leaf nodes that flow from a decision node represent possible answers. Leaf nodes grow from decision nodes in the same way that a leaf grows from a tree branch. For this reason, we call each subsection of a decision tree a “branch”.

The Golfer example

Support vector machines (SVMs)

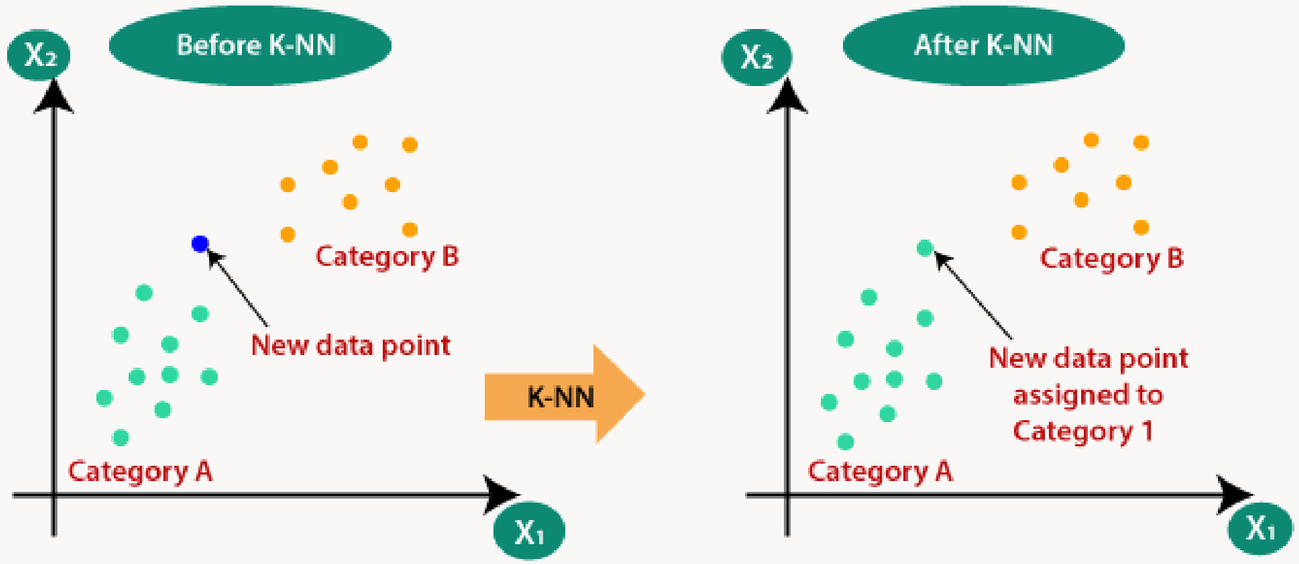
A Support Vector Machine (SVM) is a supervised machine learning algorithm that is commonly used for classification problems. But they are employed for regression purposes as well.



Source: Spiceworks

As shown in this figure on the left, the margin is the maximum width of the slice parallel to the hyperplane without an internal support vector. These hyperplanes are easier to define for linearly separable problems; however, for real-life problems or scenarios, the SVM algorithm attempts to maximize the margin between support vectors, resulting in incorrect classifications for smaller sections of data points. SVMs are potentially designed for binary classification problems. However, with the increase in computationally intensive multi-class problems, several binary classifiers are constructed and combined to formulate SVMs that can implement such multi-class classifications by binary means. SVMs are used in protein sorting processes, text categorization, facial recognition, autonomous cars, robotic systems, etc.

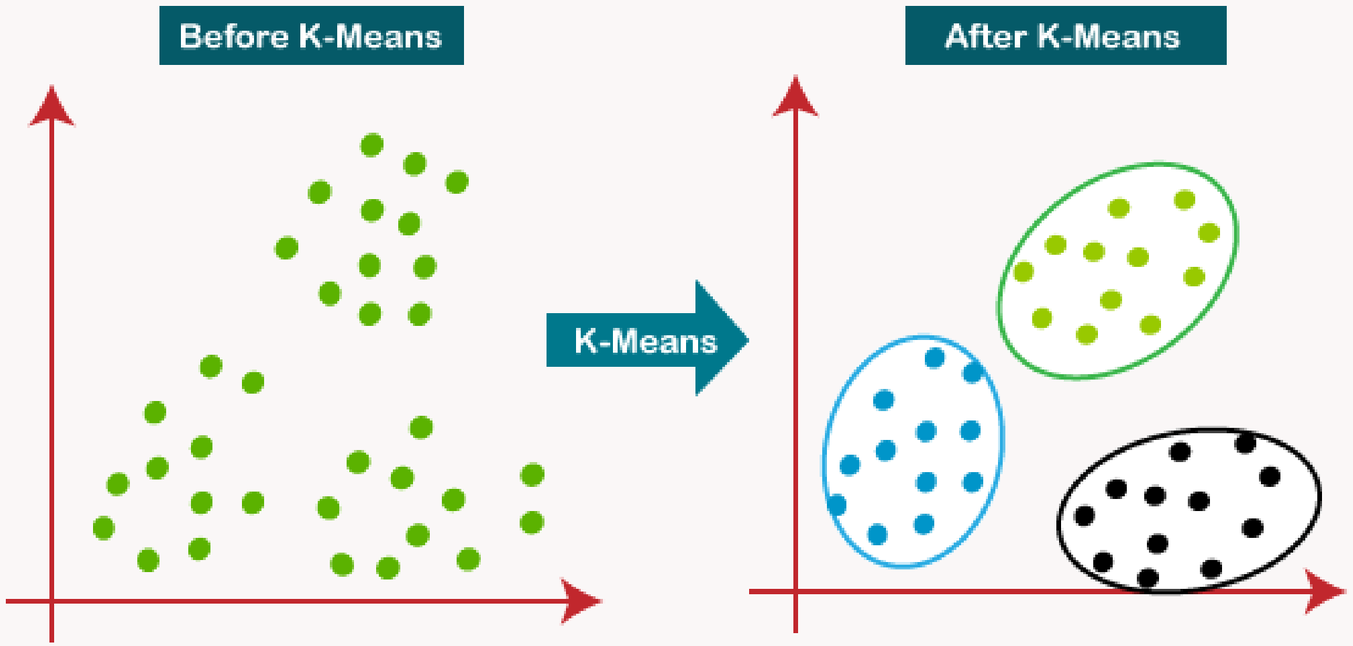
K-Nearest Neighbors (KNN)



The KNN algorithm is a model that learns from labeled input data and is then able to propose an output with new unlabeled data. The algorithm ranks the new data based on the previous dataset to provide results. Its principle is described by this statement: “Tell me who your neighbors are, I will tell you who you are”.

KNN is very easy to implement since there will be no need to build a model, make many assumptions or adjust many parameters. However, KNN must keep in memory all the observations to be able to make a prediction. This is why the choice of the size of the training set is important, but also the number of neighbors and the method to calculate the distance. Trying several combinations, doing tuning or a test is sometimes necessary to limit errors. KNN can be used for data sets of any size

K-means clustering

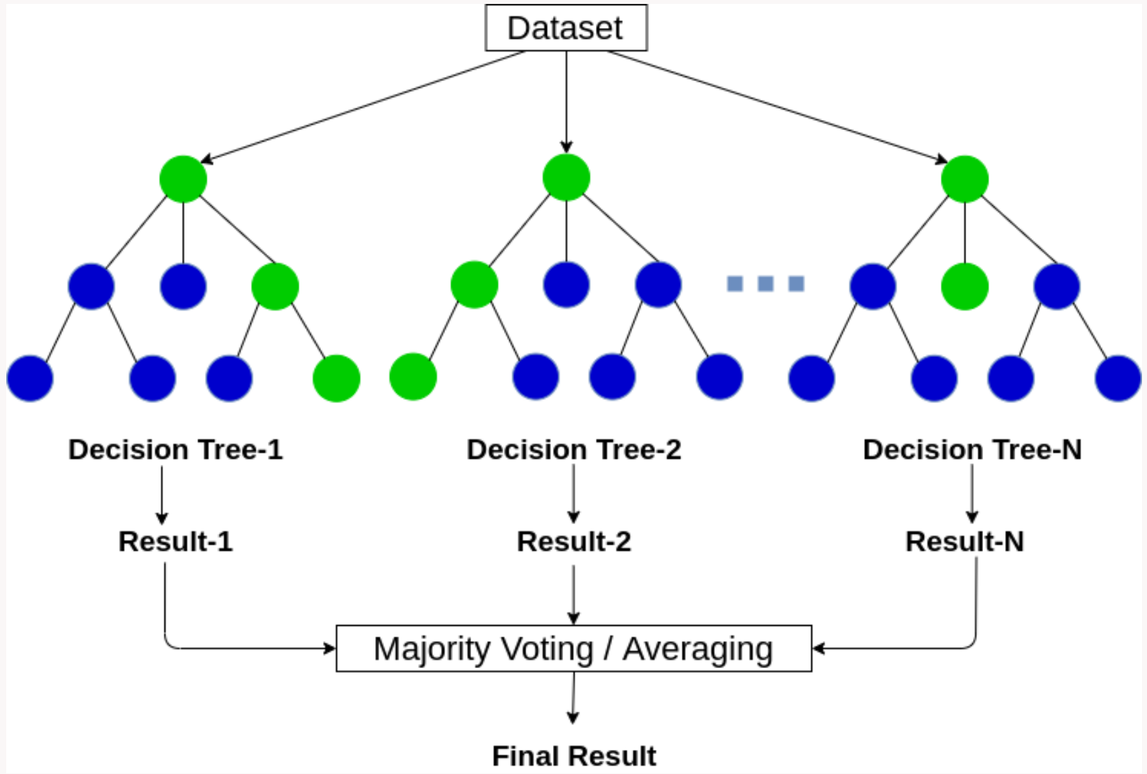


K-means clustering is a form of unsupervised learning, i.e. it does not require training data. K-means is an approach where each observation is assigned to a cluster based on the mean value of the cluster, it is a more deterministic approach than KNN for example since it tries to find clusters of data points that are close together in terms of distance instead of features. K-means is typically used for larger data set.

Naive bayes

The naive Bayes classification method is a supervised machine learning algorithm that classifies a set of observations according to rules determined by the algorithm itself. This classification tool must first be trained on a training dataset that shows the expected class based on the inputs. During the learning phase, the algorithm develops its classification rules on this data set, to apply them a second time to the classification of a prediction data set. It is particularly useful for text classification issues. An example of the use of Naive Bayes is that of the spam filter. The Naive Bayes Classifier is very fast for classification, the probability calculations are not very expensive and classification is possible even with a small dataset.

Random forest



Random forest : composed of a set of independent decision trees. Each tree has a partial vision of the problem due to a double random draw. A random draw with replacement on the rows of your database = tree bagging. A random draw on the columns of your database = feature sampling. At the end, all these trees are assembled and the prediction made by the random forest for unknown data is then the average of all the trees. The random forest uses several simple estimators (of lower individual quality) to obtain the global vision of the problem. The only downside is that the random forest gives results that are not very explanatory.The major flaw of the decision tree is that its performance is highly dependent on the starting data sample.

Data Quality considerations

Whether you're working on supervised learning or unsupervised learning, the machine needs a dataset which will be analyzed and make their results based on that data.

Data is central to the learning process so it is extremely important to have good quality data

If not, many problems can occur → There are usually some issues regarding the quantity of available data, the quality or signal-to-noise ratio in the data, the correlation between the input and output, or some combination of all three factors.

Managing Missing Data

How to fix it?

- **Removal** → ignore it by removing the incomplete data (ex: dropna method) but the disadvantage of this method is that we lose a lot of important informations.

To help us identify the data with missing values instead of removing it (we can use the aggregate mrthod). We can get a lot of informations as which rows are missing information and whether the missing information is a problem unique to certain columns or is consistent throughout all columns of the dataset.
By identifying the issues you can determine the method/ the organization you will follow next and see if you can recover the missing data or if it won't help.

- **Imputation**→ You could also fill in the missing data with other plausible values depending on the context.

The imputation method develops reasonable guesses for missing data. It's most useful when the percentage of missing data is low.
There are many methods and combinations of values possible in order to fill in the missing data. The most important thing is the prediction accuracy.

The goal of these methods is to correct/ clean up the data in order to obtain the most accurate results possible.

Class imbalance

Class imbalance, which refers to the situation where the number of classes in a data set is greater than another, is a very common problem in supervised learning that can lead to missing important information. This is especially true if you want to perform sample classification, i.e. predict which class(es) a sample comes from. Lack of equality between class samples can reduce the accuracy of your model's predictions, because the algorithm will have to "guess" the probability of the results, i.e. it will predict an outcome based on insufficient data. As a result, the results obtained may not be as detailed as you would like or may be biased, which could have been avoided if the algorithm had access to sufficient data.

How can this problem be addressed?

As with missing data, one can delete samples from the over-represented class or on the contrary complete the under-represented class by randomly copying samples from the data. These 2 options are not really ideal as they lead to lack of precision and the inability to predict the labels of the unseen data (i.e. overfitting).

Low sample size

Using machine learning on small datasets can be problematic because, in general, the "power" of machine learning to recognize patterns is proportional to the size of the dataset; the smaller the dataset, the less powerful and accurate the machine learning algorithms are. And vice versa

There is not much one can do with a low sample/small dataset. However, few techniques exist (i.e.transfer learning) but it is beyond the scope of this book so we will not address it.

Evaluation and feedback

Supervised Learning is used in many industries. It is important to know at least the basis to have a general understanding of how things work. It helped us get used to looking for all the possible options to resolve a problem and see how the things we learned in math class (statistics) can be put into use in real situations to create a model that could simplify the process for us (automatisation of manual tasks).

There are many applications of supervised learning:

- automatic language processing
- voice recognition
- computer vision,
- bioinformatics...

We also use supervised learning for the detection of spam in emails, the management of chatbots and voicebots, as well as for robotics. This method also allows the development of embedded technologies dedicated to autonomous vehicles.

The quality of data really impacts the accuracy of the results. Not taking into account these issues can significantly reduce the predictive power of your model.

Conclusion

We saw in the first part of this chapter what is supervised learning and why is it important. We then compared it to unsupervised learning.

In a second part, we discussed a number of data quality issues such as missing data, class imbalance and low sample size. We also discussed the options available for managing these issues and emphasized the importance verifying these mitigations against model performance.