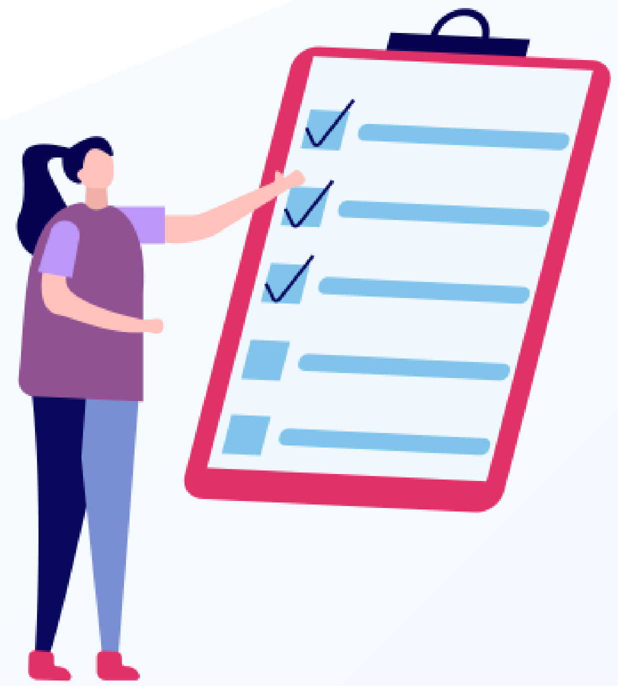
Machine Learning

Ensemble Learning

Learning Objectives

By the end of this lesson, you will be able to:

- Master ensemble modeling
- Examine the different ensemble modeling techniques, like bagging, boosting and stacking
- Evaluate performance enhancement and fine-tuning
- Reduce errors with ensembles
- Develop data flow and normalizations
- Evaluate different machine learning frameworks, like TensorFlow and Keras



Business Scenario

ABC is a machine learning-based company that is developing a program that will look for solutions on Google to address any question asked by students. The company wants to keep the application running smoothly and improve it based on collected usage data.

To initiate these tasks, the company needs to organize its collected data and normalize it, as well as manage errors.

Therefore, ABC's data scientists decided to use TensorFlow and Keras. They also decided to explore various ensemble modeling techniques, such as bagging, boosting, and stacking.

Ensemble Learning

Ensemble Learning

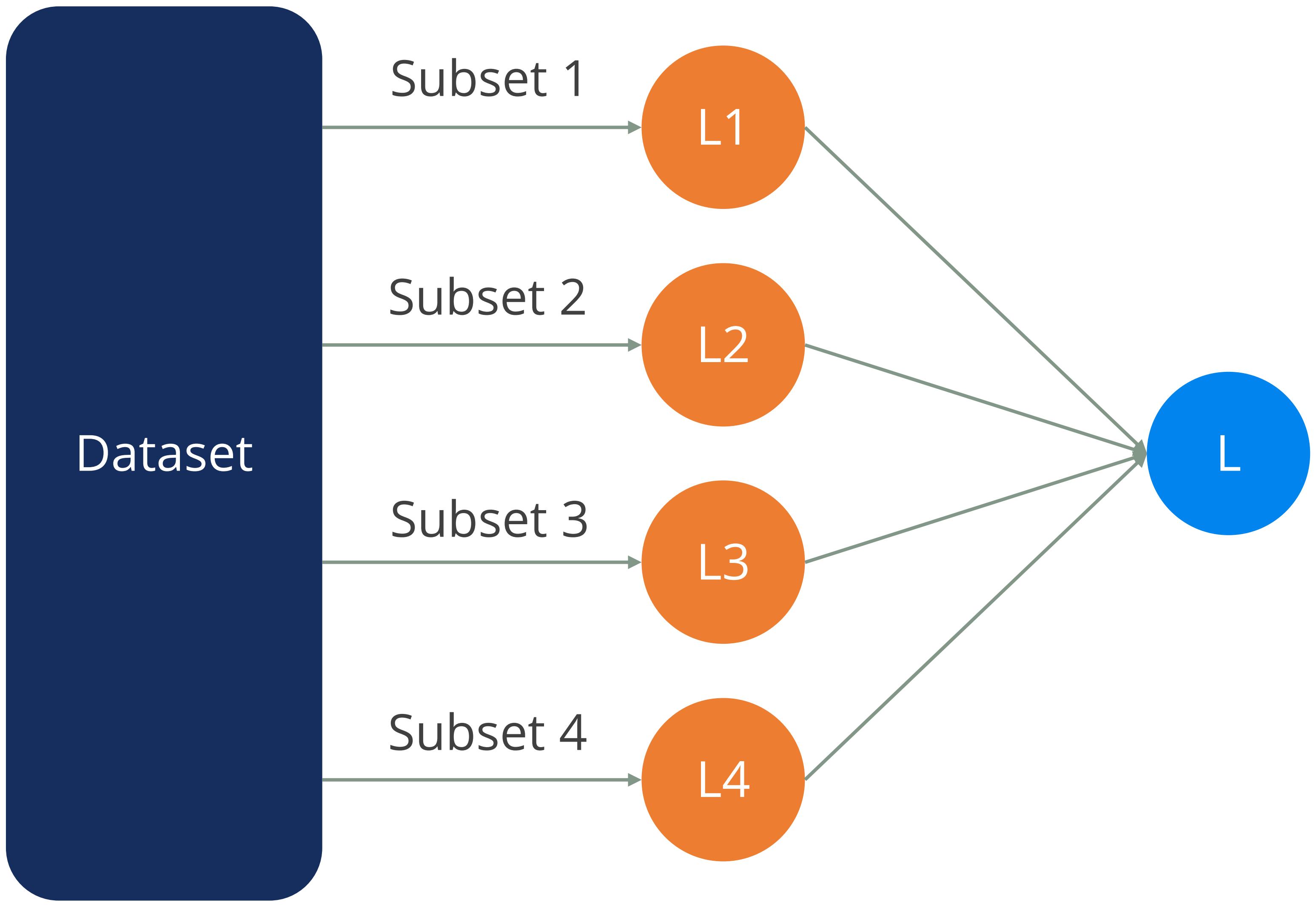
Ensemble learning is a machine learning technique that combines several base models to produce one optimal predictive model.

It aims to achieve better performance with the ensemble of models than with any individual model.

It is the approach which involves fitting two or more models to the same data and combining their predictions.

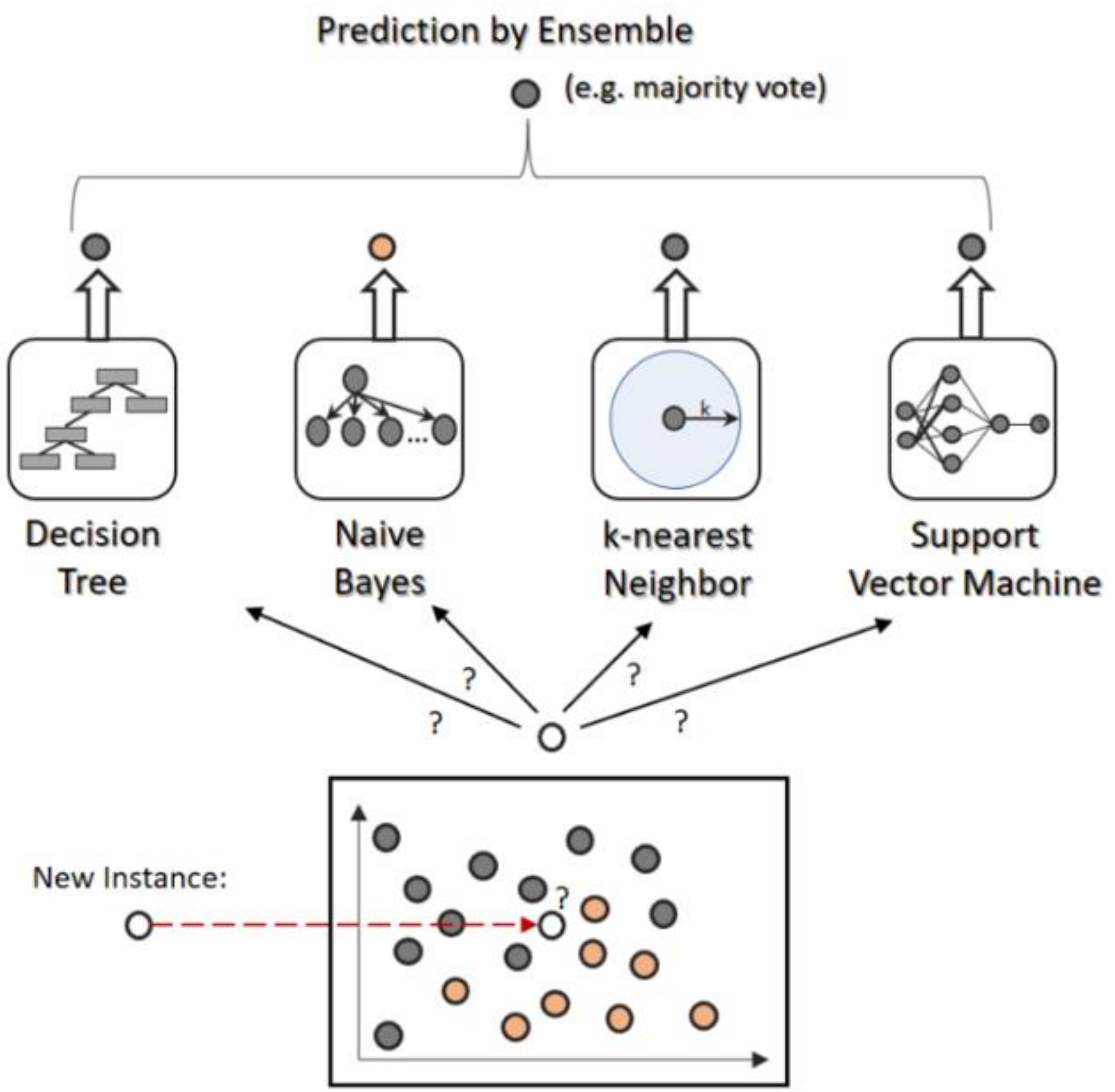
Ensemble Learning

The following diagram shows one dataset split into four subsets where each subset will be processed through learning algorithms (learners) L1 to L4.



Ensemble Learning

The below image shows the working of different algorithms together



Ensemble Learning

The term **learner** implies a learning algorithm.

L represents the learner algorithm that has combined all the learnings from the previous base models.

The predictive capability of L is more robust when compared to the base learners L1 to L4.

The accuracy of a model is greatly increased through ensemble learning, which combines the predictive power of multiple models.

Example of Ensemble Learning

Use case: problems in detecting melanoma in healthcare

Training a single classifier model to detect the presence of melanoma by feeding the patient's images is often not viable.

Solution: train multiple models with various patient images

Challenge: one cannot infer the presence of melanoma every time by feeding data through each of these trained models

In such scenarios, ensemble learning can combine the predictive power of all the trained models and boost the performance of the model.

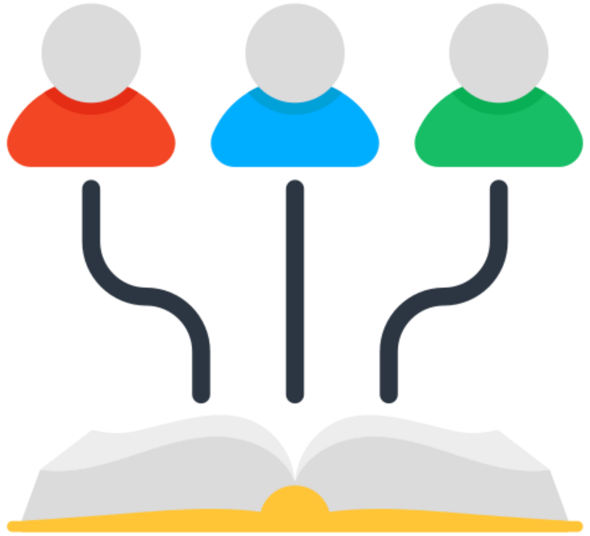
Categories in Ensemble Learning

Categories in Ensemble Learning

Ensemble learning is broadly classified into two categories:

Sequential Ensemble Technique

In this, the learners (models) are generated and sequenced one after the other.



Different learners learn sequentially from one another, that is, from earlier learners.

The later learners analyze data and calibrate based on the errors from the early learners.

Dependence between the base learners is used to reduce error and improve overall model performance.

The performance is improved by assigning larger weights to earlier learners that made mistakes.

Parallel Ensemble Technique

In this, the base learners are generated parallelly.

Examples

When referred for a job to various companies by multiple peers in parallel, the chances of a favorable outcome are higher.

When there are recommendations from multiple friends to watch a movie, you are more likely to watch that movie.

The key design principle is to use the independence between the base learners to derive the outcome.

Ensemble Methods

Most ensemble learning methods use a single base learning algorithm in a sequential or parallel form.

When learners are of the same type, they are called homogeneous learners.

Example

Random forest (parallel) and AdaBoost (sequential)

Ensemble Methods

When learners are different types of learning algorithms, they are known as heterogeneous learners.

Example

The voting classifier in scikit-learn leverages different learning algorithms in the same ensemble learning design.

Ensemble Learning Category

The ensemble learning category can be decided based on the data variables and the base learners’ dependence.

Parallel ensembles leverage the independence of base learners, whereas sequential ensembles leverage their dependence.

Sequential ensembles train a new base estimator such that the mistakes of the previously trained base estimator are minimized.

Sequential Ensemble Technique

Sequential Ensemble Technique

The sequential ensemble method generates base learners consecutively.

It uses the dependency between base learners, that is, every other piece of data in the base learner has some dependency on previous data.

Sequential Ensemble Technique

Larger weights are assigned to the mistakes made by the earlier learners to improve overall performance.

It typically uses weak learners as base estimators, as weak learners are early learners with high errors.

The key objective of this technique is to reduce errors due to data biases while learning from it.

Sequential Ensemble Technique

The steps involved in the sequential ensemble technique depend on the data construct and application needs.

01

Choosing a base model

02

Splitting the data set by random sampling

03

Updating the dataset with revised weights, based on the output of each learner

04

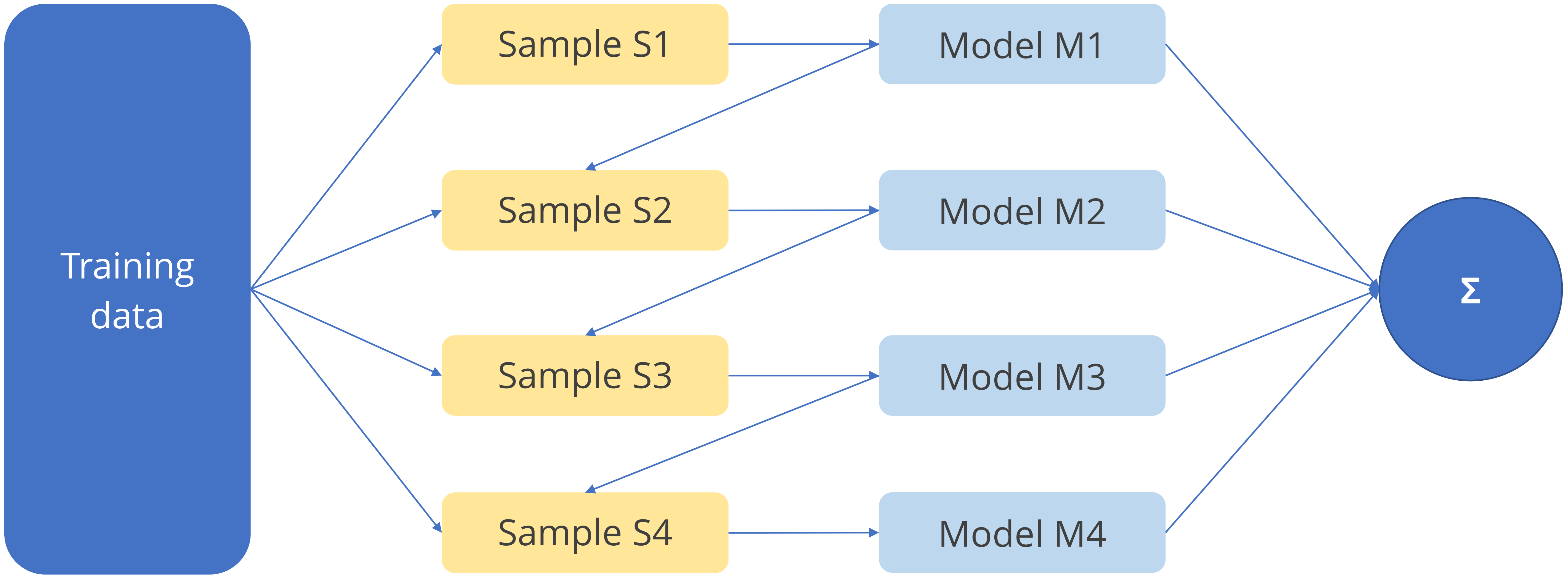
Building the learners sequentially

05

Combining all the learners with a weighted averaging strategy

Sequential Ensemble Technique

The following figure shows how the sequential ensemble technique works:



If all four base models are of the same type, it is homogeneous. If they are different, it is heterogeneous.

Sequential Ensemble Technique

As can be seen in the figure:

The training data is split into four samples.

Each sample is trained onto a different base model.

The learnings from model M1 flow into model M2 along with sample S2.

M2 learns based on the outcomes of M1 and calibrates the weights and biases accordingly.

Sequential Ensemble Technique

This weight improvement from the weak learners is leveraged to improve the overall model performance.

Σ

The summation sign indicates the function that adjusts the model to boost its overall performance.

Boosting is a sequential ensemble technique.

Sequential Ensemble Technique

Sequential ensemble technique is used when:

Boundaries of a data set must be refined to improve outlier detection.

Multiple dependent factors are found to impact the model.

Example

Electricity consumption forecasts

Sequential Ensemble Technique

New events cause cascading effects that could impact the prediction when computational resources are not limited.

Example

Financial forecasts based on sequential events

Classification based on one's logic is used to perform subclassification based on other features and weight addition, such as surface classifications.

Parallel Ensemble Technique

Parallel Ensemble Technique

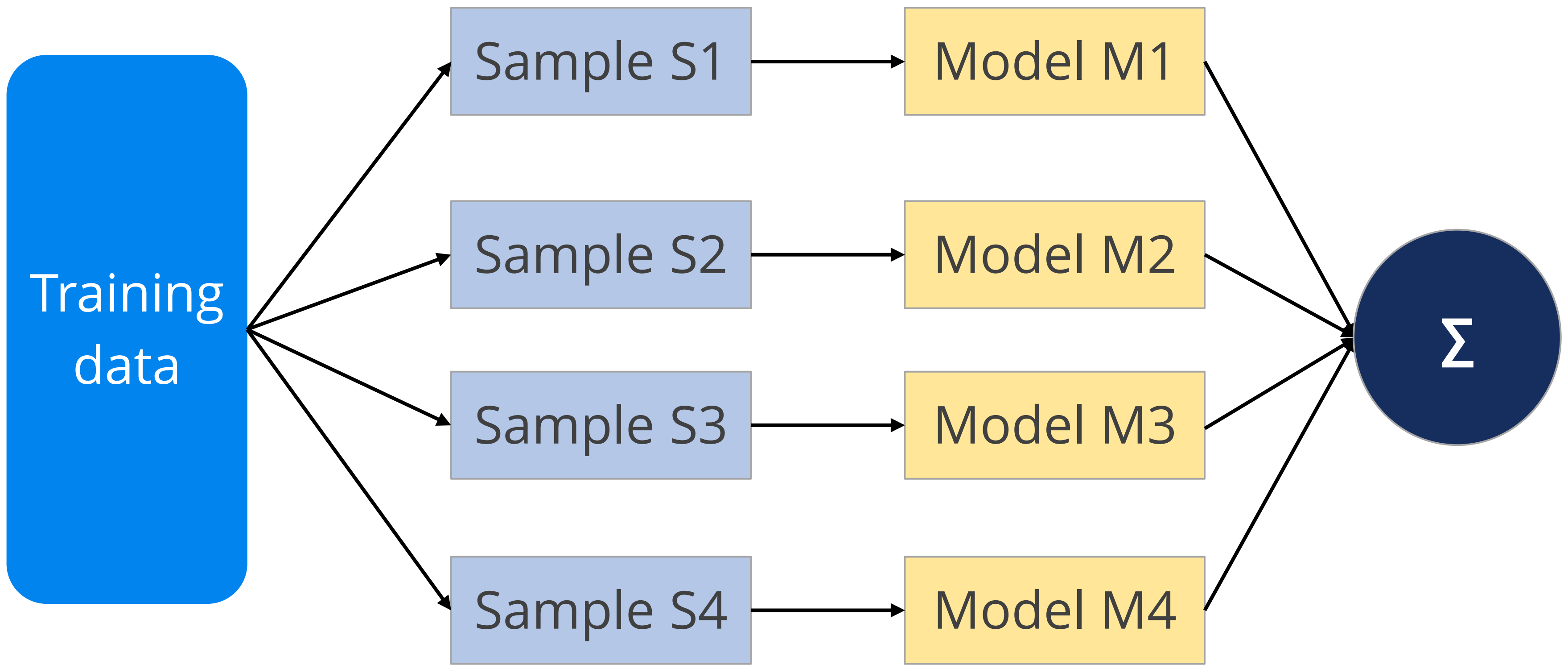
In this, the base learners are generated in parallel.

All data in the base learner is generated independently.

The key design principle is to use this independence to reduce errors due to the variance in the data while learning from it.

Parallel Ensemble Technique

Consider the following image:



Training data is split into four samples, S1 to S4.

Every sample is trained on a different base model.

There are four base models, M1 to M4.

Base estimator models are typically strong learners.

Parallel Ensemble Technique

Unlike the sequential ensemble technique, the data in the base learner doesn't have any dependency between each model.

Σ

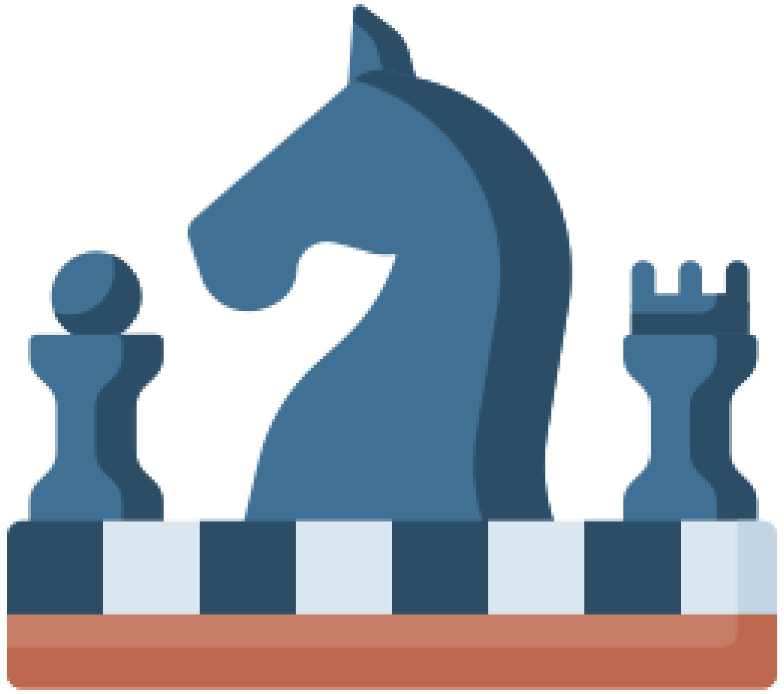
The summation sign indicates the aggregated model with improved performance, which has been learned from all independent base models, M1 to M4.

This independence of base learners significantly reduces the error due to the application of averages.

Bagging is a form of parallel ensemble learning.

Application of Parallel Ensemble Technique

Gaming: Every move of a chess or poker model competing with a player has to predict the next move.



This can be achieved by parallel ensemble computing to derive real time predictions.

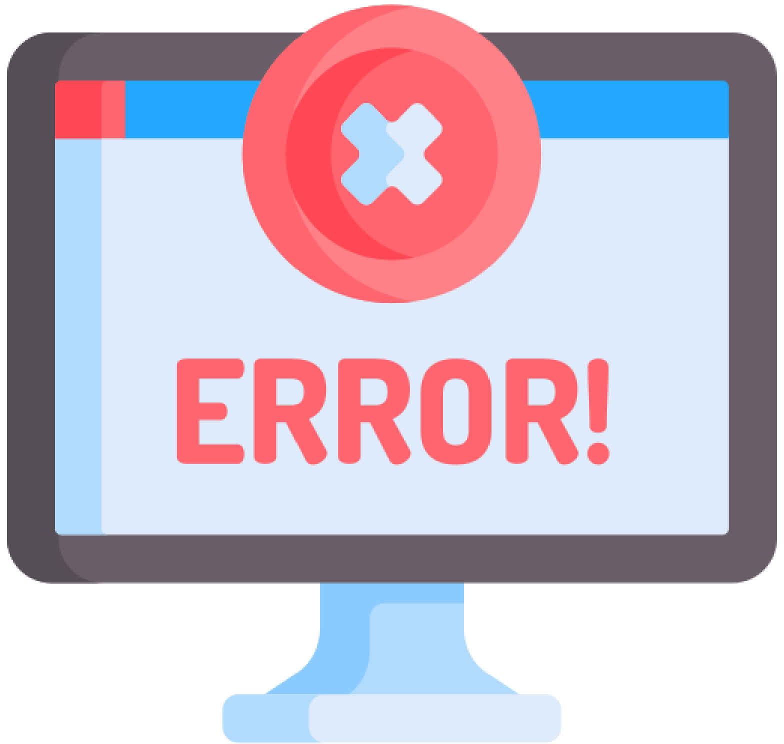
Application of Parallel Ensemble Technique

Healthcare: prediction of diabetes and chronic kidney diseases based on various classifier outcomes

The data is sampled randomly and processed through SVM and KNN independently, and the outcomes are aggregated.

Application of Parallel Ensemble Technique

Social media: High variance of user sentiments is causing variance errors.



To minimize such errors, parallel ensembles are employed.

Types of Ensemble Methods

Types of Ensemble Methods

There are mainly three types of ensemble methods:

Bagging

Boosting

Stacking

Bagging

It is an ensemble method in which each weak model is trained individually and then systematically integrated.

This term is an abbreviation for bootstrap aggregating. It helps:

It is mainly applied in classification and regression problems.

Bagging

It typically combines homogeneous weak models.

It can be classified into the following types:

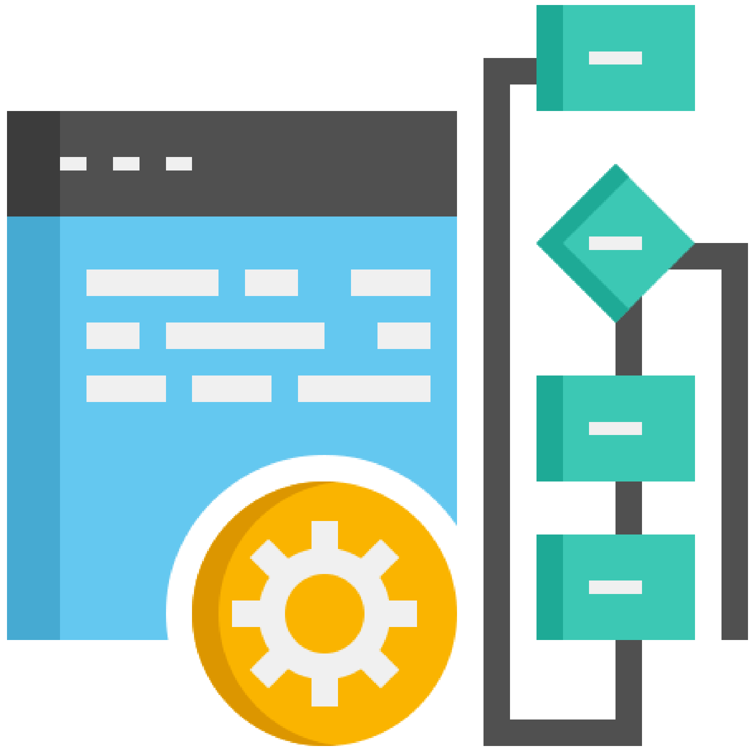
Bootstrapping: It is a sampling technique where samples are derived from the whole population through the replacement procedure.

Aggregation: It incorporates all possible outcomes of the prediction and randomizes the outcome.

Without aggregation, predictions will not be accurate because all outcomes are not taken into consideration.

Boosting

This is a technique where the algorithm learns from the previous predictor mistakes and makes better predictions.



It combines several weak base learners to form one strong learner. It typically combines homogeneous weak models.

It takes several forms such as gradient boosting, adaptive boosting, and XGBoost.

It is a sequential ensemble method that makes weak learners learn from the next learner in the sequence to create better predictive models.

Stacking

This technique is often referred to as stacked generalization. It typically combines heterogeneous weak models.

It allows a trained model to ensemble several other similar learning models' predictions.

It is implemented in regression, density estimations, and classifications.

It is used to measure the error rate in bagging.

Bagging

Bagging

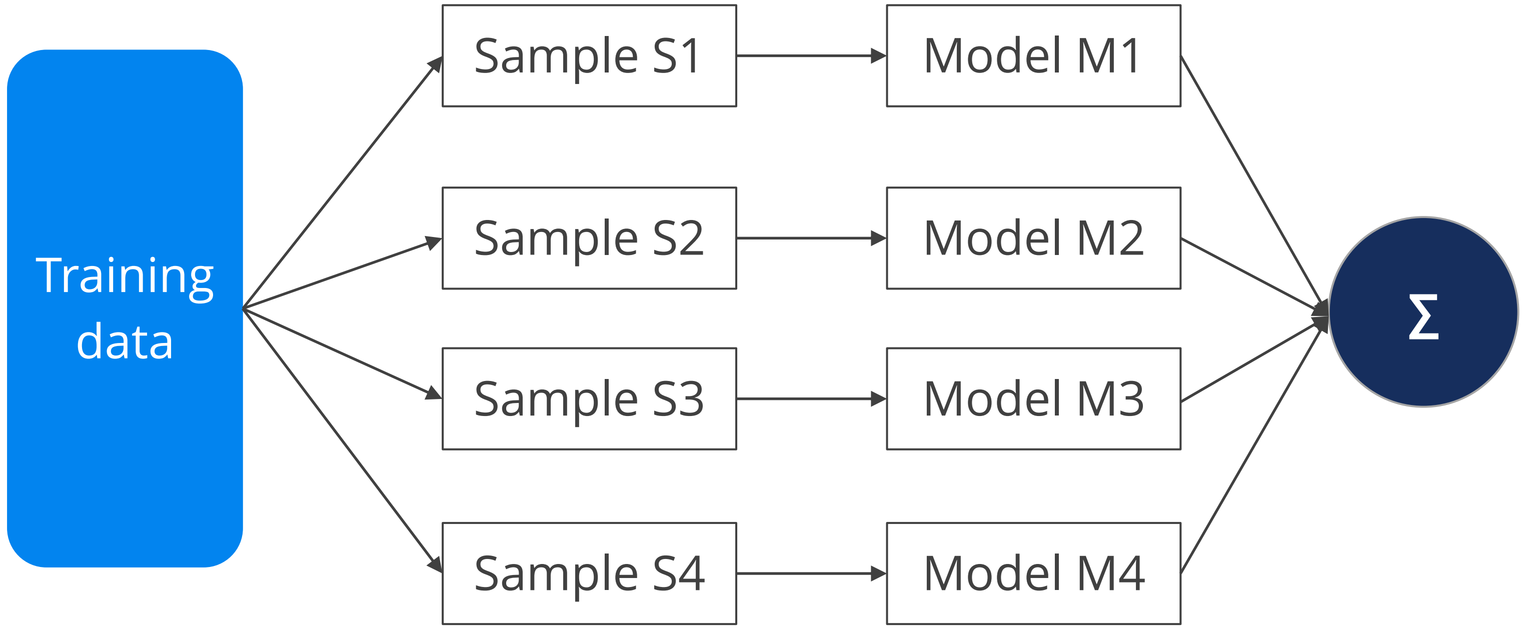
It is an ensemble technique that is frequently used in classification and regression issues.

The primary focus of bagging is to:

It uses bootstrapping and aggregation techniques to derive data set samples through the replacement procedure and incorporates all possible prediction outcomes to randomize these.

Bagging

Consider the following figure where the technique:



Extracts n subsets from the data set seen here as four subsets, S1 to S4

Uses subsets to train n base learners of the same type (M1 to M4)

Feeds each n learner with a test sample to make a prediction

Determines the number of subsets and items per subset by the nature of the machine learning problem

Bagging

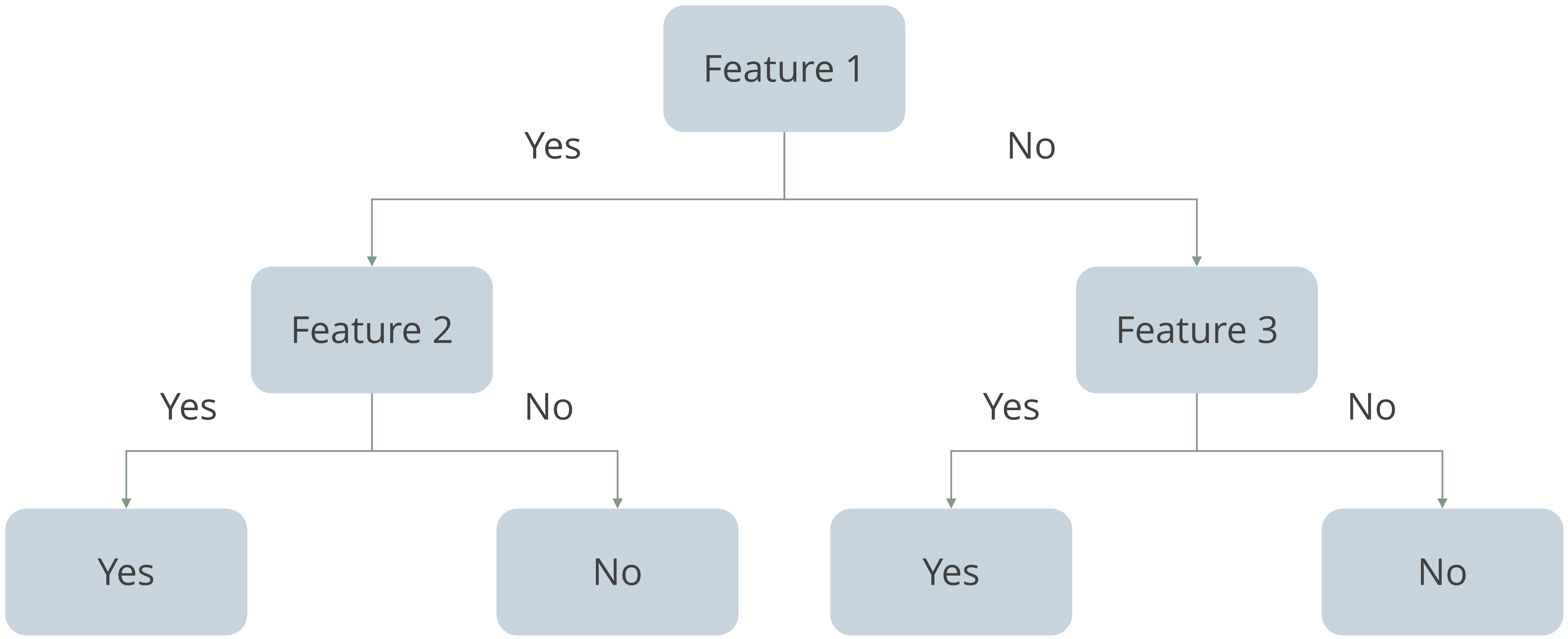
The detailed steps are as follows:

Step 1 Shape the data into bootstrapped samples and 'out of bag' samples

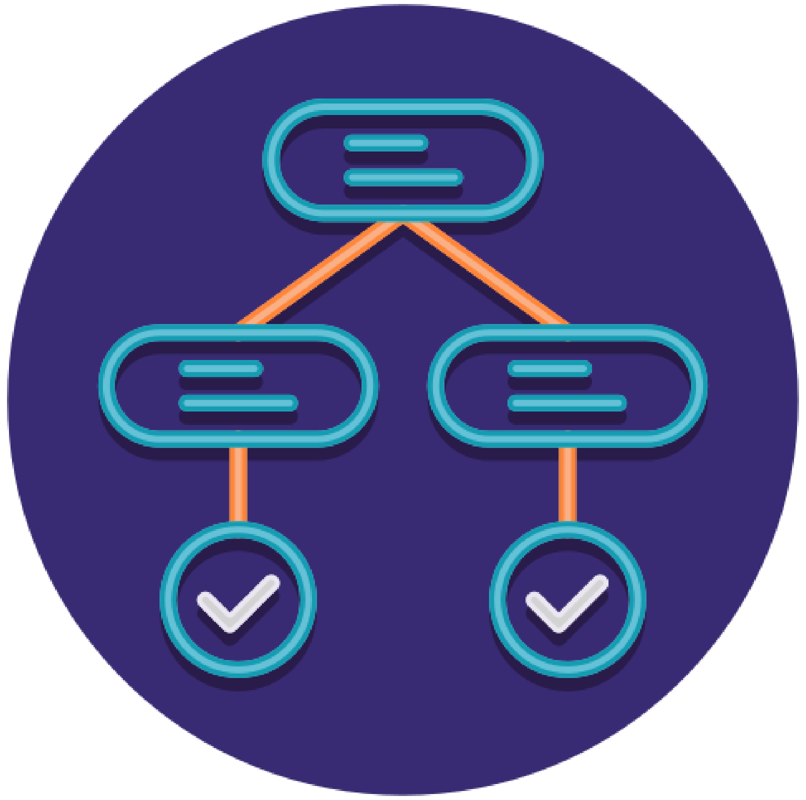
- In contrast to bootstrapped data, which may contain duplicates from the original data, the actual data is first randomly sampled.
- Out of bag samples are the original data set's remaining samples that weren't included in the bootstrapped samples.

Bagging

Step 2 Train the decision trees (learners) based on the samples



Bagging



Step 3	Improvise the design with random forest
Step 4	Verify the coverage and diversity
Step 5	Aggregate the outcomes

Bagging

To summarize the process of bagging:

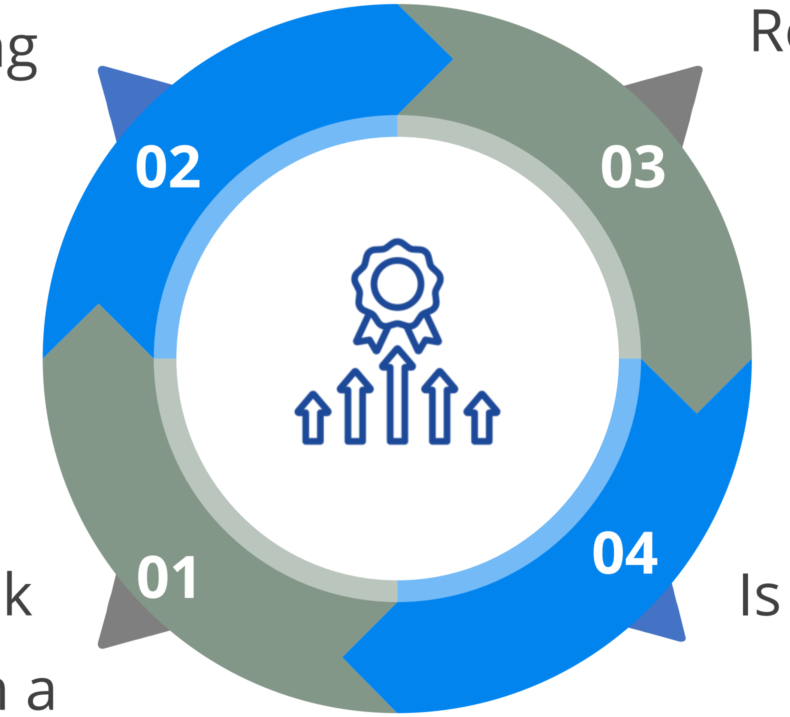
Randomly sample the data set to reduce the variance

Create bootstrap samples that are trained over independent learning models

Aggregate the outcomes of each model to derive an outcome: prediction or classification

Advantages of Bagging

Bagging has the following advantages:



01 Aggregates many weak learners to outperform a single learner

02 Reduces overfitting

03 Removes high variances

04 Is good for parallel processing

Disadvantages of Bagging

Disadvantages of the bagging technique are:

- High bias due to weak learners
- Loss of model interpretability
- Computationally heavy

Assisted Practices

Let’s understand the topic below using Jupyter Notebook.

- 7.8_Bagging

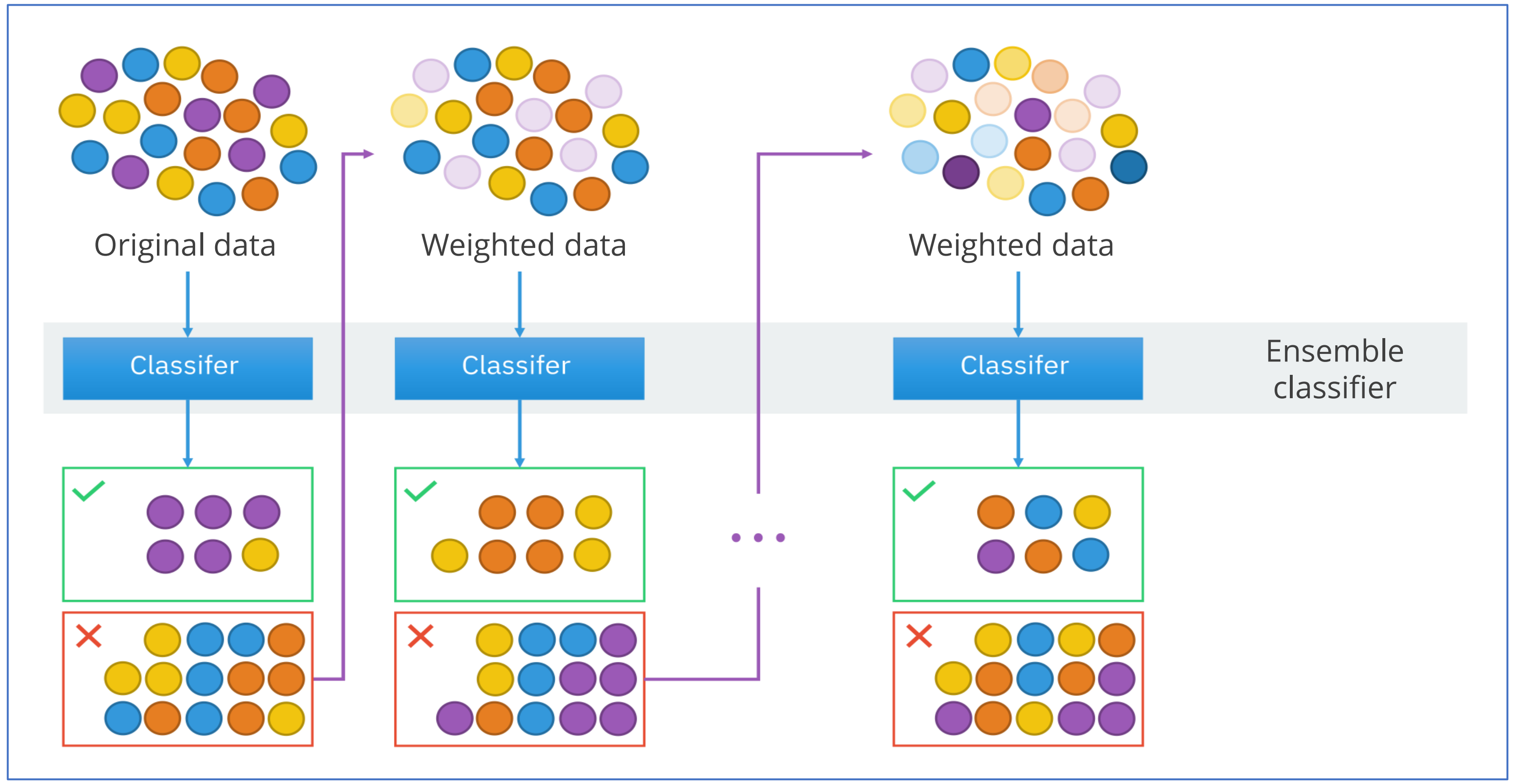
Note: Please download the pdf files for each topic mentioned above from the Reference Material section.

ASSISTED PRACTICE

Boosting

Boosting

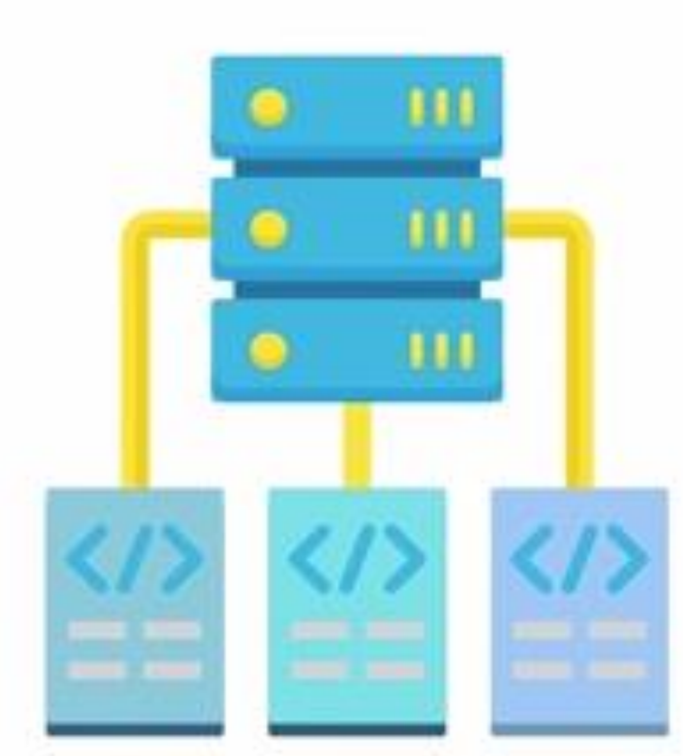
Boosting is a type of sequential ensemble learning.



In the above image, the data is calibrated to form weighted data based on the classifier, whether early or weak learners.

Boosting

Boosting makes use of sequential homogeneous ensemble learning.



The design architecture helps improve accuracies for applications with multiple iterations amongst sequential learners.

Boosting

The process involves iteratively learning from weak classifiers to build an overall strong classifier.

After processing the data through the weak classifiers, the data weights are readjusted. This is called reweighting.

Misclassified input data gain a higher weight, while correctly classified examples lose weight.

The future weak students pay more attention to the cases that earlier weak students incorrectly classified.

Boosting

Various boosting algorithms differ mainly in the way they weigh the training data points and hypotheses.

Some recent algorithms are:

- BrownBoost
- XGBoost
- AdaBoost
- LPBoost
- LogitBoost

AdaBoost is a popular boosting algorithm that adapts to the weak learners.

Boosting

Many boosting algorithms fit into the AdaBoost framework, which shows that boosting performs gradient descent in a function space using a convex cost function.

Assisted Practices

Let’s understand the topic below using Jupyter Notebook.

- 7.10_Boosting

Note: Please download the pdf files for each topic mentioned above from the Reference Material section.

ASSISTED PRACTICE

Stacking

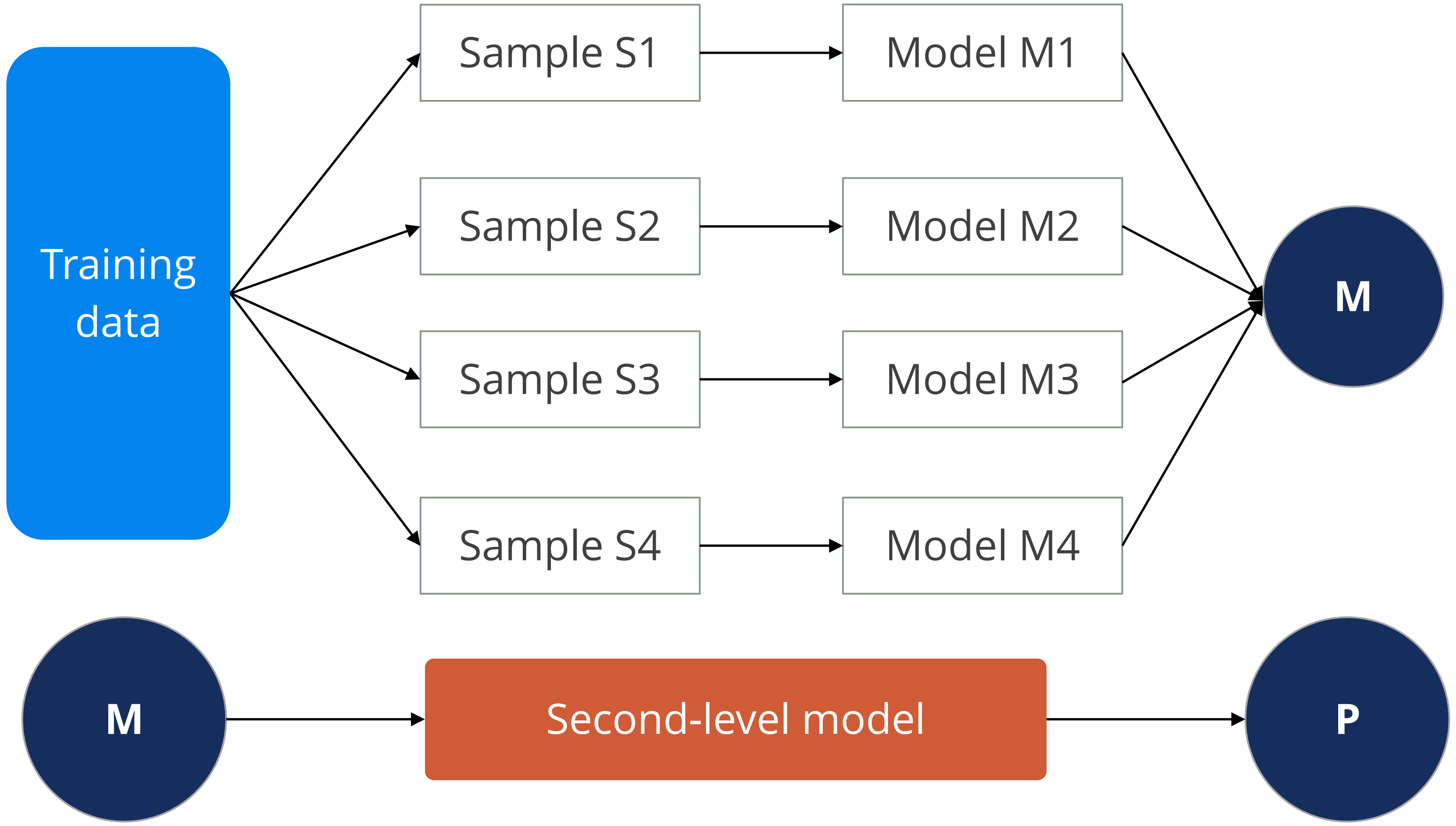
Stacking

In stacking, an ensemble machine learning technique, trained models make improved predictions by stacking several similar learning ensemble predictions.

It then runs these through another machine learning model called a meta-learner.

Stacking

The following visual of stacking uses heterogeneous models, where the base learners learn in parallel.



After individual training, a second-level model called a metamodel is trained.

The metamodel predicts based on the outcomes of different weak base learners.

Stacking Variants: Voting Ensembles

This uses various learners to train the models and aggregates the outcome based on a common logic.

On regression problems, it predicts the mean or median of the predictions from ensemble members.

For classification problems, it predicts the label with the most votes, called hard voting.

Otherwise, it predicts the label with the largest sum probability, called soft voting.

Unlike the stacking technique, voting does not add any weights to the models. All models are assumed to have the same skill level on average.

Stacking Variants: Weighted Average Ensemble

This method uses a diverse model collection of contributing members.

It weighs each model on a performance basis in a training data set.

An improved approach weighs each member based on their performance in a holdout data set.

Improvement involves tuning the coefficient weights for each model using an optimization algorithm and the model's performance in a holdout data set.

These continued weighted average improvements resemble a primitive stacking model with a linear model trained to combine the predictions.

Stacking Variants: Blending Ensemble

Blending is an explicit stacked generalization model with a specific configuration.

Stacking lacks a generally accepted configuration which is challenging for beginners.

Any model can be used as the base model and metamodel, and any resampling method can be used to prepare the training data set for the metamodel.

Stacking Variants: Blending Ensemble

Blending makes two prescriptions:

- Uses a holdout validation data set to prepare the out-of-sample predictions to train the metamodel
- Uses a linear model as the metamodel

Assisted Practices

Let’s understand the topics below using Jupyter Notebook.

- 7.12_Stacking
- 7.13_Reducing Errors with Ensembles
- 7.14_Applying Averaging and Max Voting

Note: Please download the pdf files for each topics mentioned above from the Reference Material section.

TensorFlow

TensorFlow

TensorFlow is an open-source machine learning framework developed by Google.

It has the following:

TensorFlow is an all-around effective and adaptable machine learning library that is suitable for a wide range of tasks and helps researchers build innovative models.

TensorFlow

In TensorFlow, the data is not stored as integers, floats, or strings. Instead, the values are encapsulated in an object called a tensor.

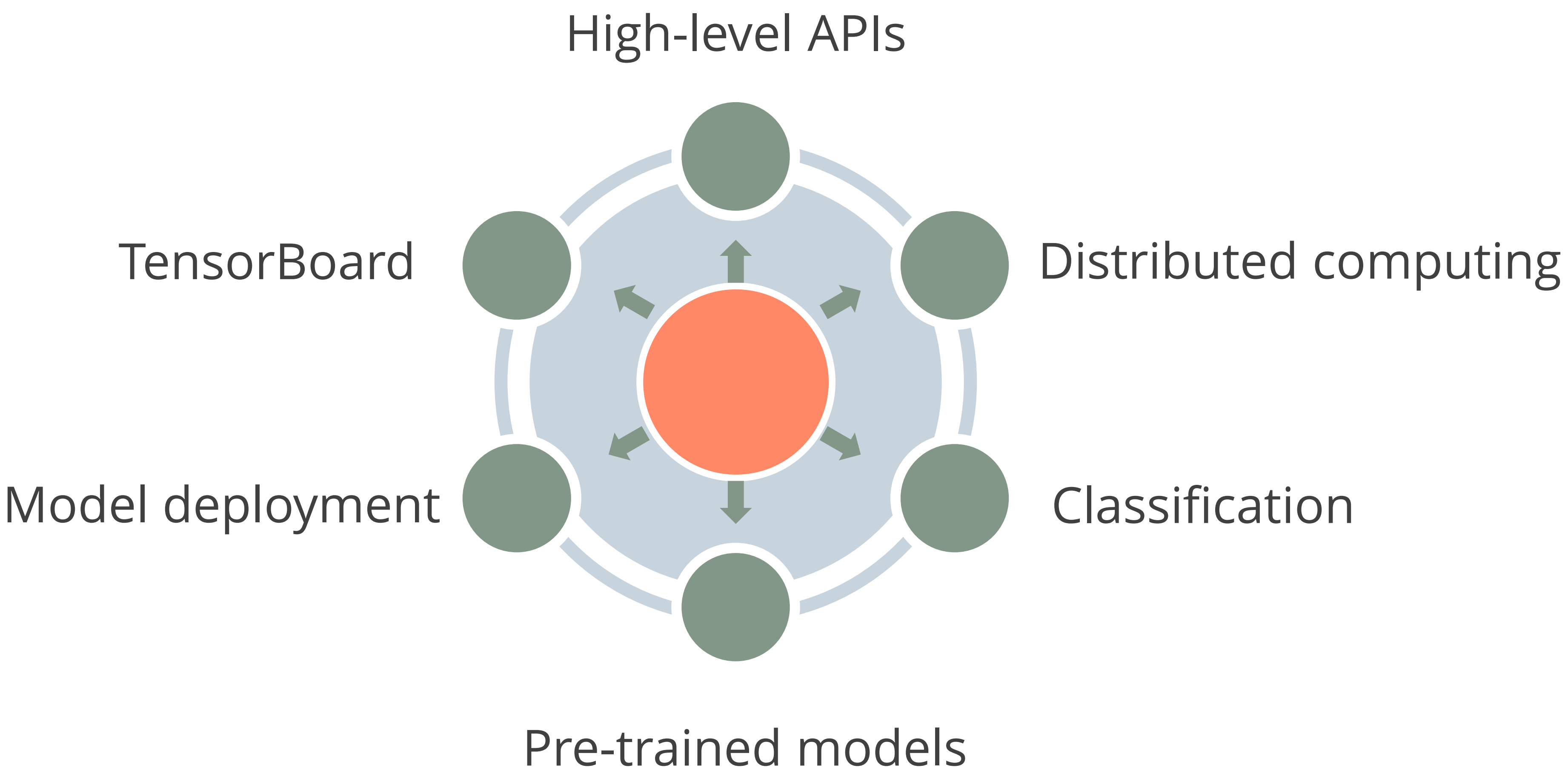
Example of a tensor

If a Python list is passed to TensorFlow, it will convert that into a tensor.

TensorFlow

It is a powerful tool for artificial intelligence as it helps create large-scale neural networks with complex layers.

Application of TensorFlow is vast and includes:



Case Studies: Healthcare

In the healthcare industry, TensorFlow is used to increase the speed and accuracy of an MRI.

Google developed DermAssist using TensorFlow, which allows one to take pictures of the skin and identify health complications.

Sinovation Ventures uses TensorFlow to identify and classify eye diseases from optical coherence tomography (OCT) scans.

Case Studies: Social Media

TensorFlow has been implemented in the following applications.

Twitter implemented TensorFlow to rank tweets with users' preferences.

VSCO, a photo-sharing app, uses TensorFlow to suggest filters for photos.

RankBrain, a search engine released by Google, uses TensorFlow.

Case Studies: Education

TensorFlow is used to filter out toxic chat messages in classrooms in a virtual learning platform.

It is also used to accurately identify a student's current abilities and help students decide the most suitable future course of action depending on their capabilities.

Case Studies: Retail

Many e-commerce platforms use TensorFlow to get personalized recommendations for their customers.

Cosmetics companies use TensorFlow to create an augmented reality experience for customers to test various shades of make-up on their faces.

Assisted Practices

Let’s understand the topic below using Jupyter Notebook.

- 7.16_Hands-on with TensorFlow: Part A

Note: Please download the pdf files for each topic mentioned above from the Reference Material section.

ASSISTED PRACTICE

Keras

Keras

Keras is an open-source machine learning library that is written in Python.

Keras wraps around the functionalities of other libraries of machine learning and deep learning, including:

TensorFlow

Theano

CNTK

Keras

Keras is closely tied to the TensorFlow library and acts as an interface for it.

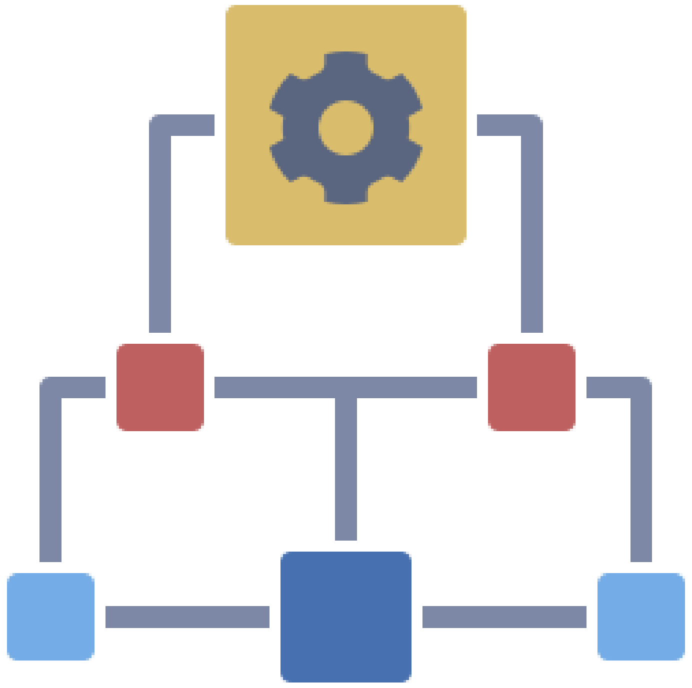
It also allows the user to define and train neural network models with only a few lines of code.

Keras

The following is a list of frameworks supported by Keras:

Theano

TensorFlow



PlaidML

MXNet

CNTK

Keras

Some of the characteristics of Keras are:

Time-Saving

Employs exceptional efficiency as a versatile library, enabling utilization across a wide range of machine learning tasks and accommodating multiple models

Flexible

Follows the principle of progressive disclosure of complexity

Powerful

Provides industry-strength performance and scalability

Additionally, it enables the user to define and train neural network models with very little code.

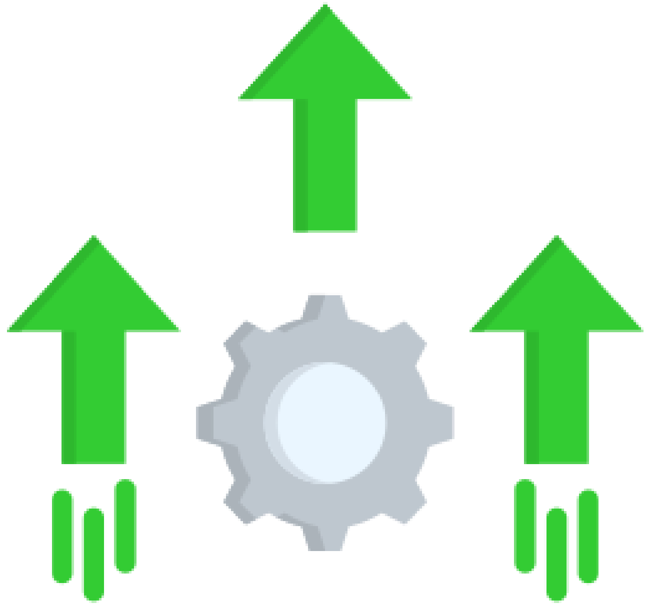
Need for Keras

Keras is a good choice for the following reasons:

- It is consistent and has simple APIs.
- It reduces the actions required to implement basic code and explains user errors clearly.
- Time taken to prototype with Keras is lesser than with comparable libraries.
- It provides a variety of deployment options depending on user needs.

Need for Keras

Many languages with a high level of inbuilt features are much slower, hence building custom features with them can be harder.

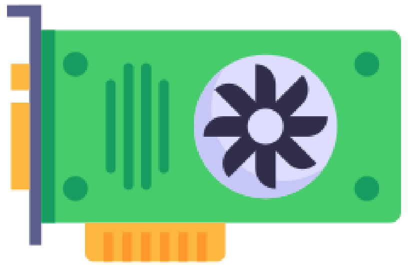


Since Keras runs on top of TensorFlow and is much faster, one can create customized workflows in a straightforward manner.

Many commercial and retail companies use Keras.

Features of Keras

The features of Keras are:



Runs on CPU and GPU

Supports neural network models

Is modular, which makes it expressive, flexible and expensive

Applications of Keras

The following are the applications of Keras:

- Create deep learning models that can be deployed on smartphones
- Train deep learning models
- Create and deploy the models very efficiently

Assisted Practices

Let’s understand the topic below using Jupyter Notebook.

- 7.18_Hands-on with TensorFlow: Part B

Note: Please download the pdf files for each topic mentioned above from the Reference Material section.

ASSISTED PRACTICE

Key Takeaways

- Ensemble learning improves models by combining the predictive power of multiple models.
- Ensemble techniques are broadly divided into two categories: sequential ensemble learning and parallel ensemble learning.
- Bagging, boosting and stacking are three types of ensemble methods.
- TensorFlow is an open-source platform that provides tools and resources for machine learning.
- Keras is an open-source machine learning library that provides a time-saving and flexible workflow for building and training neural network models.

Knowledge Check

Knowledge Check

1

What is the difference between sequential and parallel ensemble techniques?

A. In sequential ensemble, base learners are generated in parallel, and in parallel ensemble, learners are generated consecutively.

B. The sequential technique is applied when the base learners are generated in parallel, and the parallel is applied when the learners are generated consecutively.

C. The sequential technique uses dependence between the base learners to reduce error, whereas the parallel technique uses independence between the base learners to reduce error.

D. There is no difference between sequential and parallel ensemble techniques.

Knowledge Check 1

What is the difference between sequential and parallel ensemble techniques?

A. In sequential ensemble, base learners are generated in parallel, and in parallel ensemble, learners are generated consecutively.

B. The sequential technique is applied when the base learners are generated in parallel, and the parallel is applied when the learners are generated consecutively.

C. The sequential technique uses dependence between the base learners to reduce error, whereas the parallel technique uses independence between the base learners to reduce error.

D. There is no difference between sequential and parallel ensemble techniques.

The correct answer is **C**

Sequential technique uses dependence between the base learners to reduce error, whereas the parallel technique uses independence between the base learners to reduce errors.

Knowledge Check

2

What is the purpose of averaging and voting techniques?

A. To reduce errors in the model

B. To reduce the variance in the model

C. To increase the bias in the model

D. To increase the variance in the model

Knowledge Check

2

What is the purpose of averaging and voting techniques?

A. To reduce errors in the model

B. To reduce the variance in the model

C. To increase the bias in the model

D. To increase the variance in the model

The correct answer is **A**

Averaging and voting techniques are used to reduce errors in the model.

Knowledge Check

3

How does Keras follow the principle of progressive disclosure of complexity?

A. It has a simple workflow that is quick and easy.

B. It is flexible in its approach.

C. It provides industry-strength performance and scalability.

D. It reduces the load of developers to a large extent.

Knowledge Check 3

How does Keras follow the principle of progressive disclosure of complexity?

A. It has a simple workflow that is quick and easy.

B. It is flexible in its approach.

C. It provides industry-strength performance and scalability.

D. It reduces the load of developers to a large extent.

The correct answer is **A**

Keras follows the principle of progressive disclosure of complexity by having a simple workflow that is quick and easy.

Thank You!