P P Savani University
School of Engineering

Institute of Computer Science and Application

SSCA3021: Data Science

Module 5: Data Science Application & BOKEH

By Misha Patel

Data Science Application

Data Science is the interdisciplinary field that combines statistics, machine learning, programming, and domain knowledge to analyze large datasets and extract valuable insights. Its applications are vast and are transforming industries by enabling data-driven decision-making.

Major Applications of Data Science

1. **Healthcare**
   - **Medical Image Analysis:** Identifying tumors, fractures, or organ abnormalities using deep learning.
   - **Drug Discovery & Genomics:** Predicting protein structures and genetic variations.
   - **Disease Prediction & Prevention:** Using patient history and wearable devices.
   - **Example:** IBM Watson Health assists in cancer diagnosis.

2. **Business & Marketing**
   - **Customer Segmentation:** Grouping customers based on behavior, purchase history, or demographics.
   - **Recommendation Systems:** Used by Amazon, Netflix, and Spotify.
   - **Churn Prediction:** Identifying customers likely to leave a service.
   - **Sentiment Analysis:** Understanding public opinion from social media.

Major Applications of Data Science

3. **Finance & Banking**
    - **Fraud Detection:** Real-time anomaly detection in transactions.
    - **Credit Scoring & Risk Management:** Assessing customer creditworthiness.
    - **Algorithmic Trading:** Predictive models for high-frequency stock trading.
    - **Customer Service:** AI-driven chatbots for banking support.

4. **Retail & E-Commerce**
    - **Inventory Management:** Predicting demand to avoid stockouts and overstock.
    - **Dynamic Pricing:** Adjusting product prices in real time.
    - **Personalized Marketing:** Based on browsing and purchase history.
    - **Example:** Walmart uses data science for supply chain optimization.

Major Applications of Data Science

5. **Transportation & Logistics**
   - **Route Optimization:** Uber and Ola use predictive modeling for routes.
   - **Predictive Maintenance:** Airlines forecast equipment failures.
   - **Self-Driving Cars:** Autonomous vehicles rely on deep learning.
   - **Example:** DHL uses AI for package tracking and route planning.

6. **Social Media & Entertainment**
   - **Content Recommendation:** YouTube suggests videos based on watch history.
   - **Targeted Advertising:** Facebook and Instagram ads use user behavior.
   - **Fake News Detection:** Identifying misinformation using NLP.
   - **Engagement Analysis:** Measuring likes, shares, and comments.

Major Applications of Data Science

7. **Education**
   - **Adaptive Learning Systems:** Platforms personalize learning paths.
   - **Student Performance Prediction:** Identifying students at risk.
   - **Automated Grading:** AI systems grade assignments and quizzes.
   - **Example:** EdX analyzes learner engagement.

8. **Government & Public Policy**
   - **Smart Cities:** Traffic control, waste management, and urban planning.
   - **Crime Prediction & Policing:** Predictive models analyze hotspots.
   - **E-Governance:** Improving citizen services through digital data.
   - **Public Health:** Forecasting disease outbreaks like COVID-19.

Data Science Real-Life Case studies

1. Predicting Equipment Failures in Airlines (GE Aviation)

Scenario: Aircraft engines require frequent monitoring, and unexpected failures cause delays, high costs, and safety risks.

Data-Driven Solution:

- Used **IoT sensors** in jet engines to collect real-time flight data (temperature, vibration, pressure).
- Applied **predictive analytics & machine learning** to forecast maintenance needs.

Applications:

- Optimized maintenance schedules (predictive maintenance).
- Reduced flight delays and operational costs.
- Increased passenger safety.

2. Personalized Content Recommendations (Netflix)

Scenario: Netflix has millions of users worldwide with diverse preferences. To keep users engaged, content recommendations must be precise.

Data-Driven Solution:

- Analyzed user behavior: watch history, viewing duration, device type, and ratings.
- Used **collaborative filtering & deep learning** models to recommend personalized shows/movies.

Applications:

- Increased user retention and reduced churn by 10%.
- Enhanced customer experience with tailored recommendations.
- Helped in **data-driven content creation** (like *House of Cards*).

3. Smart City Traffic Optimization (City of Los Angeles)

Scenario: Traffic congestion in Los Angeles caused delays, fuel wastage, and pollution.
Data-Driven Solution:

- Deployed **real-time traffic sensors, cameras, and GPS data**.
- Applied **AI-based traffic signal optimization algorithms**.

Applications:

- Reduced traffic jams by **12% in major intersections**.
- Improved public transportation efficiency.
- Contributed to **sustainability goals** with lower emissions.

4. Fraud Detection in Banking (PayPal)

Scenario: Millions of online transactions daily made PayPal vulnerable to fraud.

Data-Driven Solution:

- Collected transaction metadata (IP address, time, frequency, device info).
- Applied anomaly detection & machine learning classification models.

Applications:

- Detected fraudulent transactions in real time.
- Saved billions in losses annually.
- Built trust among customers and improved security.

BOKEH

Bokeh is a powerful Python library used for creating **interactive** and **visually rich data visualizations** directly in **web browsers.**
It is well-suited for building **dashboards** and sharing data insights and allows for integration with web technologies like **HTML and JavaScript**.

Key Features

- Bokeh enables high-performance, interactive charts and plots, including line plots, scatter plots, bar charts, and more.
- **Plots** and **dashboards** can be displayed in **Jupyter notebooks** or **HTML files** or can be **embedded in web applications** using frameworks such as **Django or Flask**.
- Supports **exporting** visuals as **PNG or PDF**.
- Provides **interactive tools (zoom, pan, hover, select)** directly in the browser.
- **Integrates** seamlessly with popular Python libraries like **Pandas** and **NumPy** for **enhanced data analysis workflows.**

Advanced Interactive Features

Feature	Description
Custom Toolbars	Select, arrange and position plot tools
Linked Interactions	Sync axes, ranges, and data selections
Interactive Legends	Hide/mute plot elements by clicking the legend.
Widgets	Add sliders, buttons, dropdowns for controls
Callbacks	Python & JS for dynamic interactivity
Hover/Tooltips	Display extra info on mouse-over
Selection Tools	Enable box, lasso, tap selection on plots
Streaming Data	Update plots with live data feeds
Dashboard Layout	Multi-plot widgets with coordinated actions

Installation of BOKEH

Bokeh can be installed using pip:

```
pip install bokeh
```

Or using conda (recommended with Anaconda):

```
conda install bokeh
```

Bokeh Interfaces

bokeh.plotting: High-level interface for quickly creating basic plots with minimal code.

bokeh.models: Low-level interface for more advanced customizations and flexibility.

```
from bokeh.plotting import figure, output_file, show
p = figure(title="Sample Plot", x_axis_label='X-Axis',
y_axis_label='Y-Axis')

p.line(x, y, legend_label='Line', line_width=2)
p.circle(x, y, size=10, color="navy", alpha=0.5)
output_file("output.html")
show(p)
```

Common Visualizations

- Line Plot
- Scatter Plot
- Box Plot
- Histogram
- Bar Plot
- Pie/Donut Chart
- Grid Plot (Multiple plots on one canvas)

Advanced Usage

- Integrate with streaming data for real-time dashboards.
- Link with web apps (Flask/Django).
- Custom JavaScript can be added for advanced cases.

Basic Structure of a Bokeh Plot

Bokeh plots generally require:

- A figure object to define the plotting area.
- Data for visualization, often provided in a Pandas DataFrame.
- Glyphs (shapes like circles, lines, etc.) to represent data points.

Python–Data Visualization Using Bokeh

- **figure():** Create a plot figure.
- **Glyphs (like line, circle):** Add visual elements.
- **HoverTool:** Add interactivity.
- **DataFrame integration:** Use data directly from a Pandas DataFrame.
- Bokeh's flexibility and interactivity make it a great choice for visualizing data, especially in Jupyter Notebooks or web applications. Let me know if you'd like more specific examples or a walkthrough on any aspect!

Glyphs Using Bokeh

- Glyphs are nothing but visual shapes that are drawn to represent the data, such as circles, squares, lines, rectangles, etc.
- Display a variety of visual shapes whose attributes can be associated with data columns from ColumnDataSources.
- These are used for scatter plots and to mark individual data points.
    - asterisk()
    - circle()
    - circle_cross()
    - diamond()
    - square()

Circle Glyphs

- **circle():** The circle() method adds a circle glyph to the figure and needs x and y coordinates of its center. Additionally, it can be configured with the help of parameters such as fill_color, line-color, line_width etc.
- **circle_cross():** The circle_cross() method adds a circle glyph with a '+' cross through the center.
- **circle_x():** The circle_x() method adds a circle with an 'X' cross through the center.

Data Science Real-Life Case studies

1. Predicting Equipment Failures in Airlines (GE Aviation)

Scenario: Aircraft engines require frequent monitoring, and unexpected failures cause delays, high costs, and safety risks.

Data-Driven Solution:

- Used **IoT sensors** in jet engines to collect real-time flight data (temperature, vibration, pressure).
- Applied **predictive analytics & machine learning** to forecast maintenance needs.

Applications:

- Optimized maintenance schedules (predictive maintenance).
- Reduced flight delays and operational costs.
- Increased passenger safety.

2. Personalized Content Recommendations (Netflix)

Scenario: Netflix has millions of users worldwide with diverse preferences. To keep users engaged, content recommendations must be precise.

Data-Driven Solution:

- Analyzed user behavior: watch history, viewing duration, device type, and ratings.
- Used **collaborative filtering & deep learning** models to recommend personalized shows/movies.

Applications:

- Increased user retention and reduced churn by 10%.
- Enhanced customer experience with tailored recommendations.
- Helped in **data-driven content creation** (like *House of Cards*).

3. Smart City Traffic Optimization (City of Los Angeles)

Scenario: Traffic congestion in Los Angeles caused delays, fuel wastage, and pollution.
Data-Driven Solution:

- Deployed **real-time traffic sensors, cameras, and GPS data**.
- Applied **AI-based traffic signal optimization algorithms**.

Applications:

- Reduced traffic jams by **12% in major intersections**.
- Improved public transportation efficiency.
- Contributed to **sustainability goals** with lower emissions.

4. Fraud Detection in Banking (PayPal)

Scenario: Millions of online transactions daily made PayPal vulnerable to fraud.

Data-Driven Solution:

- Collected transaction metadata (IP address, time, frequency, device info).
- Applied anomaly detection & machine learning classification models.

Applications:

- Detected fraudulent transactions in real time.
- Saved billions in losses annually.
- Built trust among customers and improved security.

Other Case studies submitted by you…

Link:
https://drive.google.com/drive/folders/10q2ChotvrPDWOVQ_FAf7AyjayuVy44hGnJojjvjIgHZoEbYjpCS2IR51v-6iejYyhqPuNpA4?usp=sharing

Random Forest Algorithm

Random Forest is a widely-used machine learning algorithm, which combines the output of multiple decision trees to reach a single result.

Assumptions of Random Forest

To effectively use Random Forest, it is important to understand the underlying assumptions of the algorithm:

- **Independence of Trees:** The decision trees in the forest should be independent of each other. This is achieved through bootstrap sampling and feature randomness.
- **Sufficient Data:** Random Forest requires a large amount of data to build diverse trees and achieve optimal performance.
- **Balanced Trees:** The algorithm assumes that the individual trees are grown sufficiently deep to capture the underlying patterns in the data.
- **Noisy Data Handling:** Random Forest can handle noisy data, but it assumes that the noise is randomly distributed and not systematic.

What is Random Forest?

Random forest, a popular machine learning algorithm, merges the outputs of numerous decision trees to produce a single outcome.

Its popularity stems from its user-friendliness and versatility, making it suitable for both classification and regression tasks.

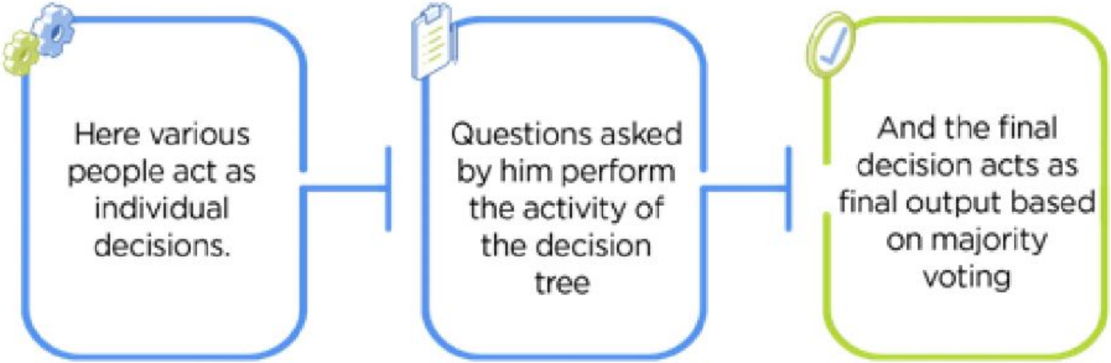
One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables, as in the case of regression, and categorical variables, as in the case of classification.

Random Forest Applications

- **Customer churn prediction:** Businesses can use random forests to predict which customers are likely to churn (cancel their service) so that they can take steps to retain them. For example, a telecom company might use a random forest model to identify customers who are using their phone less frequently or who have a history of late payments.
- **Fraud detection:** Random forests can identify fraudulent transactions in real-time. For instance, a bank might employ a random forest model to spot transactions made from unusual locations or involving unusually large amounts of money.
- **Stock price prediction:** It can predict future stock prices. However, it is important to note that stock price prediction is a very difficult task, and no model is ever going to be perfectly accurate.
- **Medical diagnosis:** These can help doctors diagnose diseases. For example, a doctor might use a random forest model to help them diagnose a patient with cancer.
- **Image recognition:** It can recognize objects in images. For example, a self-driving car might use a random forest model to identify pedestrians and other vehicles on the road.

Real-Life Analogy of Random Forest

- Let’s dive into a real-life analogy to understand this concept further. A student named X wants to choose a course after his 10+2, and he can't decide which course fits his skill set. So he decides to consult various people like his cousins, teachers, parents, degree students, and working people.
- He asks them varied questions like why he should choose, job opportunities with that course, course fee, etc. Finally, after consulting various people about the course, he decides to take the course suggested by most people.

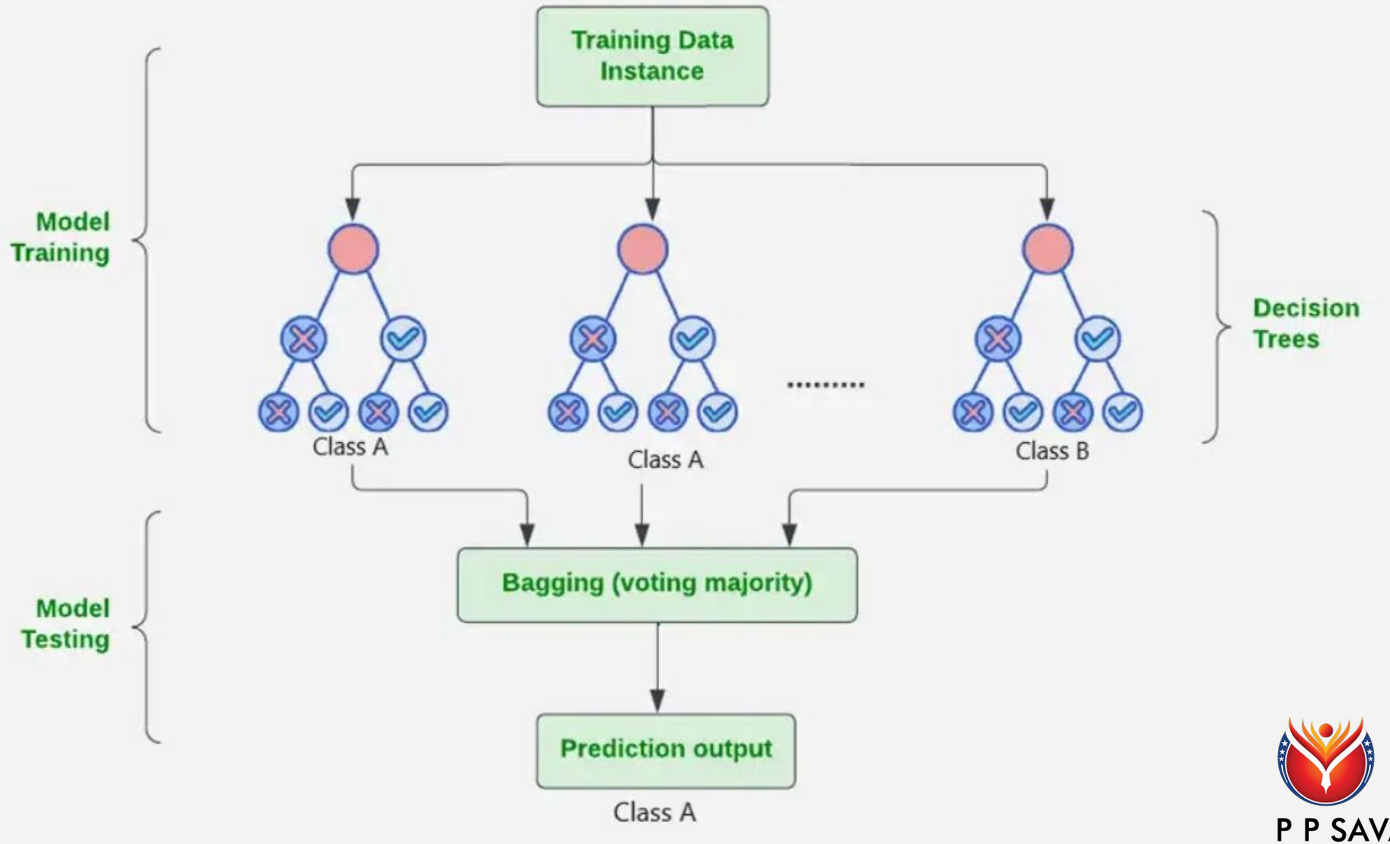


Here various people act as individual decisions.

Questions asked by him perform the activity of the decision tree

And the final decision acts as final output based on majority voting

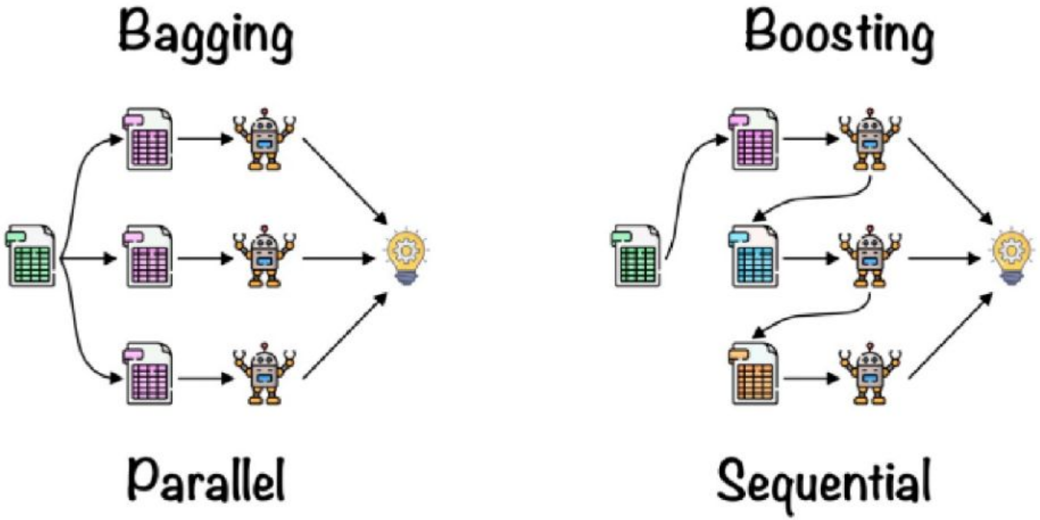
Random Forest Algorithm in Machine Learning



Working of Random Forest Algorithm

Before understanding the working of the random forest algorithm in machine learning, we must look into the ensemble learning technique. **Ensemble** simply means combining multiple models. Thus a collection of models is used to make predictions rather than an individual model.

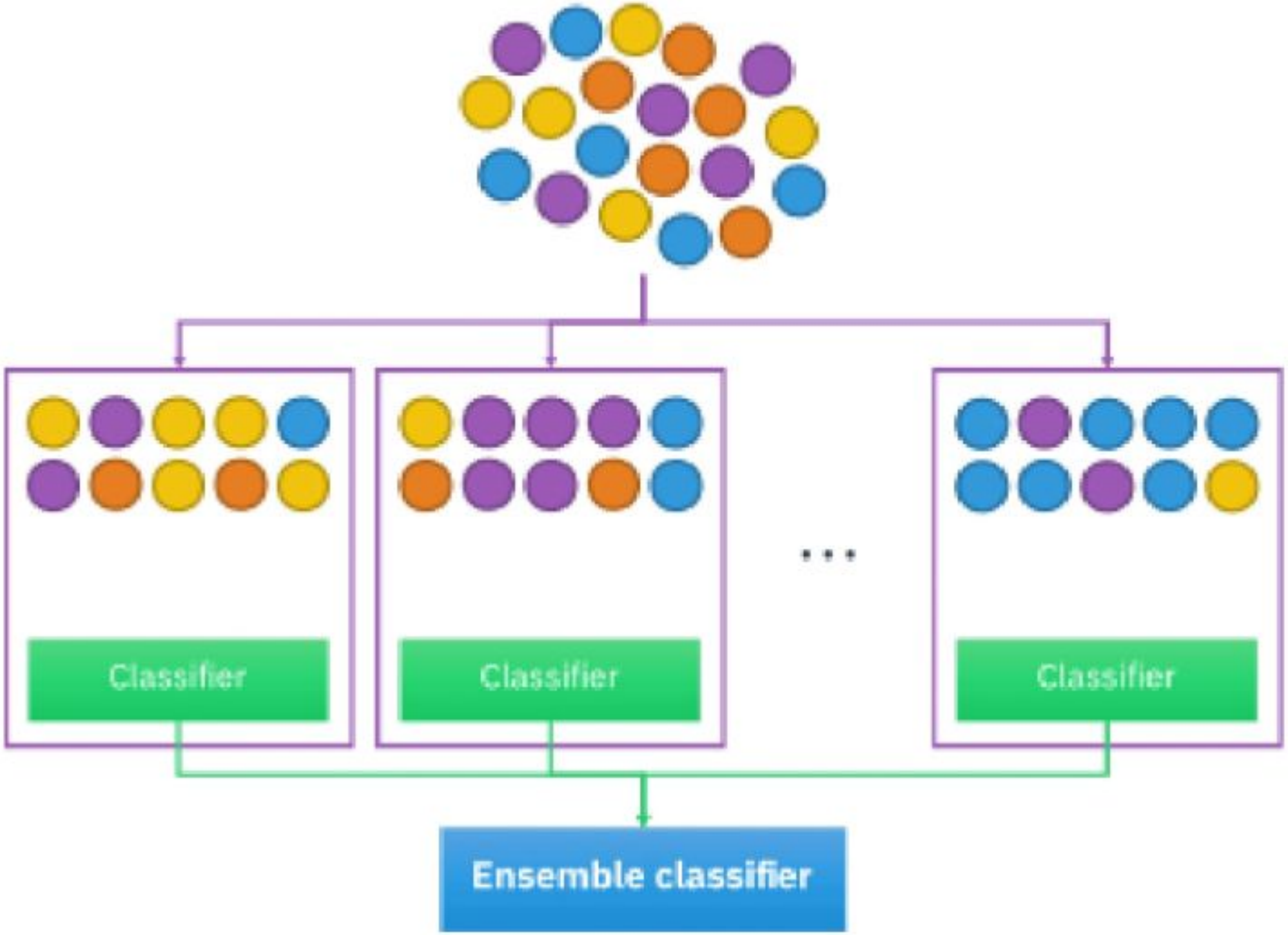
Ensemble uses two types of methods:



Bagging

Bagging, also known as Bootstrap Aggregation, serves as the ensemble technique in the Random Forest algorithm. Here are the steps involved in Bagging:

- **Selection of Subset:** Bagging starts by choosing a random sample, or subset, from the entire dataset.
- **Bootstrap Sampling:** Each model is then created from these samples, called Bootstrap Samples, which are taken from the original data with replacement. This process is known as row sampling.
- **Bootstrapping:** The step of row sampling with replacement is referred to as bootstrapping.
- Independent Model Training: Each model is trained independently on its corresponding Bootstrap Sample. This training process generates results for each model.
- **Majority Voting:** The final output is determined by combining the results of all models through majority voting. The most commonly predicted outcome among the models is selected.
- **Aggregation:** This step, which involves combining all the results and generating the final output based on majority voting, is known as aggregation.



Original Data

Bootstrapping

Aggregating

Bagging

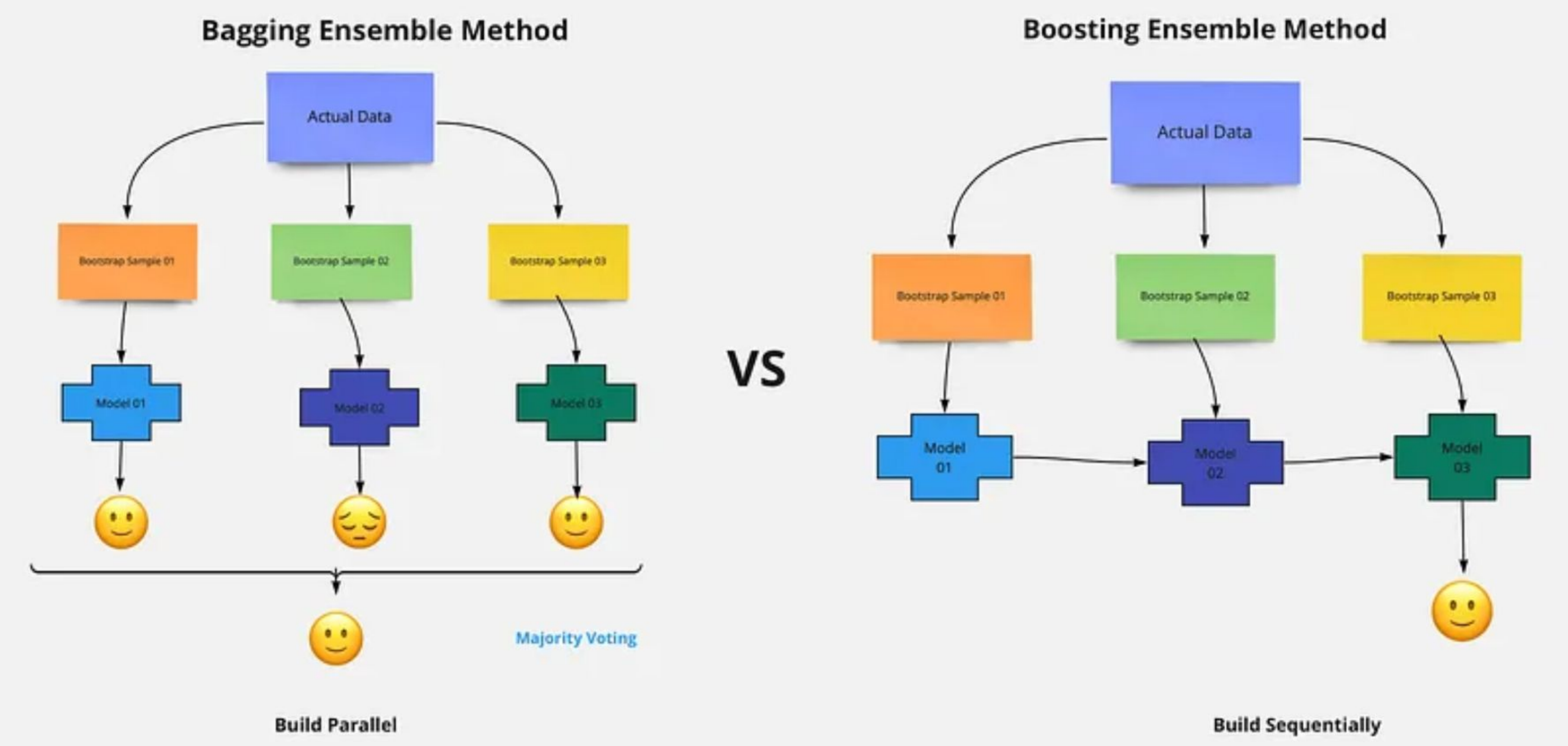
Boosting

Boosting is one of the techniques that use the concept of ensemble learning. A boosting algorithm **combines multiple simple models** (also known as weak learners or base estimators) to generate the final output. It is done by building a model by using weak models in series.

There are several boosting algorithms; **AdaBoost** was the first really successful boosting algorithm that was developed for the purpose of binary classification.

Steps Involved in Random Forest Algorithm

- **Step 1:** In this model, a subset of data points and a subset of features is selected for constructing each decision tree. Simply put, n random records and m features are taken from the data set having k number of records.
- **Step 2:** Individual decision trees are constructed for each sample.
- **Step 3:** Each decision tree will generate an output.
- **Step 4:** Final output is considered based on Majority Voting or Averaging for Classification and regression, respectively.



Bagging Ensemble Method

Build Parallel

VS

Boosting Ensemble Method

Build Sequentially

Recent trends in various data collection and analysis techniques

Advanced trends in data collection and analysis for data science in 2025 involve rapid innovation across platforms, automation, and artificial intelligence.

Advanced Data Collection Trends

- **Real-Time Data Collection:** Use of streaming technologies like Apache Kafka and Flink enables organizations to gather data instantly from sensors, IoT devices, mobile apps, and web activity for immediate analysis.
- **Edge Computing:** Data is processed close to its source (on devices, sensors, etc.), reducing latency and bandwidth use for applications in autonomous vehicles, smart factories, and healthcare monitoring.
- **Internet of Things (IoT):** The explosion of IoT devices results in massive volumes of live data, which is used across industries for monitoring, optimization, and prediction.
- **Decentralized Data Architectures (Data Mesh):** Data ownership and management are distributed across teams, making large, complex datasets easier to access and analyze without central bottlenecks.
- **Cloud-Native Data Services:** Scalable, flexible cloud platforms (Kubernetes, serverless computing) streamline data storage, collection, and processing while enabling seamless collaboration.

Advanced Data Analysis Trends

- **Augmented Analytics:** Incorporates AI and machine learning directly into analytics workflows, automating tasks like data preparation, insight generation, and visual explanation, making advanced analysis accessible to non-experts.
- **Automated Machine Learning (AutoML):** Platforms automate much of the model training, selection, and validation, enabling rapid and robust predictive analytics without the need for specialized knowledge.
- **Natural Language Processing (NLP):** Enables machines to analyze text and voice data, summarize documents, and extract insights from unstructured sources like social media and customer feedback.
- **AI-Powered Stream Processing:** Real-time ML models that adapt continuously, allowing immediate insight generation, automated decision-making, and fraud detection as new data arrives.
- **Explainable AI (XAI):** Tools and frameworks designed to clarify why AI or machine learning models make certain predictions, addressing bias, fairness, and model transparency.
- **Data Democratization & No-Code/Low-Code Analytics:** Self-service BI and analytics platforms empower anyone to generate insights, regardless of programming skill, boosting cross-departmental innovation.
- **Enhanced Data Governance & Privacy:** Stricter frameworks for the ethical use, security, and confidentiality of data, especially with global standards like GDPR, become increasingly critical.

Emerging Tools and Technologies

- **Streaming Data Platforms:** Apache Kafka, Flink, and Google Dataflow for real-time analytics.
- **Cloud Data Warehouses:** Snowflake and BigQuery for scalable storage, streaming access, and easy sharing.
- **AutoML Providers:** Google AutoML and Microsoft Azure ML for automated, beginner-accessible analytics.
- **Self-Service BI:** Tableau and Power BI for democratized data analysis and visualization.