**PROJECT TITLE:**

**CUSTOMER CHURN PREDICTION**

**PROBLEM DEFINITION:**

* The project involves using IBM Cognos to predict customer churn and identify factors influencing customer retention. The goal is to help businesses reduce customer attrition by understanding the patterns and reasons behind customers leaving. This project includes defining analysis objectives, collecting customer data, designing relevant visualizations in IBM Cognos, and building a predictive model.

**PRE-PROCESSING:**

**STEPS:**

1. REVIEWING THE DATASET
2. EXPLORATORY DATA ANALYSIS
3. BUILDING THE MODEL

**REVIEWING THE DATASET:**

The dataset we will use is the Customer churn prediction dataset of 2020. It is all about measuring why customers are leaving the business or stating whether customers will change telecommunication providers or not is what churning is. The dataset contains 4250 samples. Each sample has 19 input features and 1 boolean target variable, which indicates the class of the sample.

**EXPLORATORY DATA ANALYSIS:**

Churn Analysis describes the company’s customer loss rate. Churn means Attrition in simple words, which occurs in two forms customer attrition and employee attrition. When the attrition is high, the company’s growth graph starts coming down, and the company suffers a high loss time during the attrition. Churn can be minimized by analyzing the company’s work environment, product growth, market conditions, dealer connections, etc. If churn increases by only one point, then it directly affects the business in a negative perspective. High Churn rates compound very fast that can have a massive loss to the company, so it is important to calculate churn regularly.

**BUILDING THE MODEL:**

Customer churn is the phenomenon in which a client stops doing business with an entity. Users can stop using a company’s product or service for a variety of reasons, such as affordability, dissatisfaction with the offering, and bad customer service.

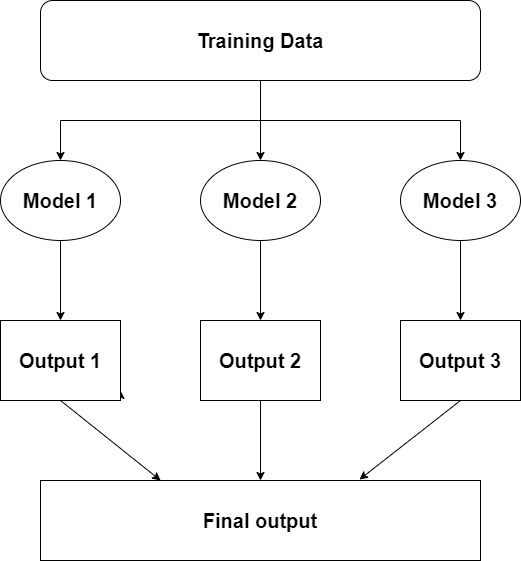
More often than not, customers who churn from one company will start doing business with their competitor. For instance, if you aren’t happy with your current mobile service provider due to slow Internet speed, you are likely to switch to an alternative.

**ALGORITHM:**

1. ENSEMBLE LEARNING
2. SUPERVISED LEARNING

**ENSEMBLE LEARNING:**

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**.**

****

**SUPERVISED LEARNING:**

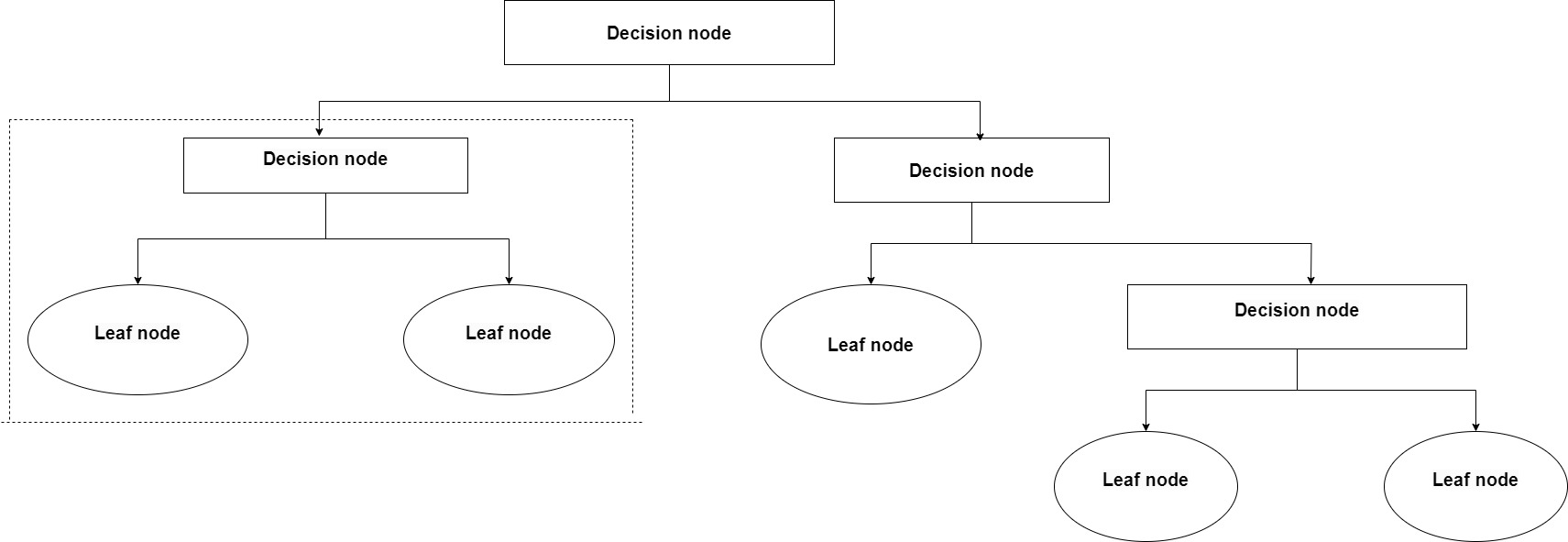
Supervised learning, also known as supervised machine learning, is a subcategory of [machine learning](https://www.ibm.com/topics/machine-learning) and [artificial intelligence](https://www.ibm.com/topics/artificial-intelligence). It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately, which occurs as part of the cross validation process. Supervised learning helps organizations solve for a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox.

**Supervised learning algorithms:**

Various algorithms and computations techniques are used in supervised machine learning processes. Below are brief explanations of some of the most commonly used learning methods, typically calculated through use of programs like R or Python:

**Decision Tree:**

Decision trees are a nonparametric supervised learning method used for classification and regression. The goal is to build a model that predicts the value of a target variable by learning simple decision rules derived from the properties of the data. A tree can be viewed as a piecewise constant approximation.For example, in the following example, a decision tree learns from data to approximate a sine wave using a series of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the better the model.

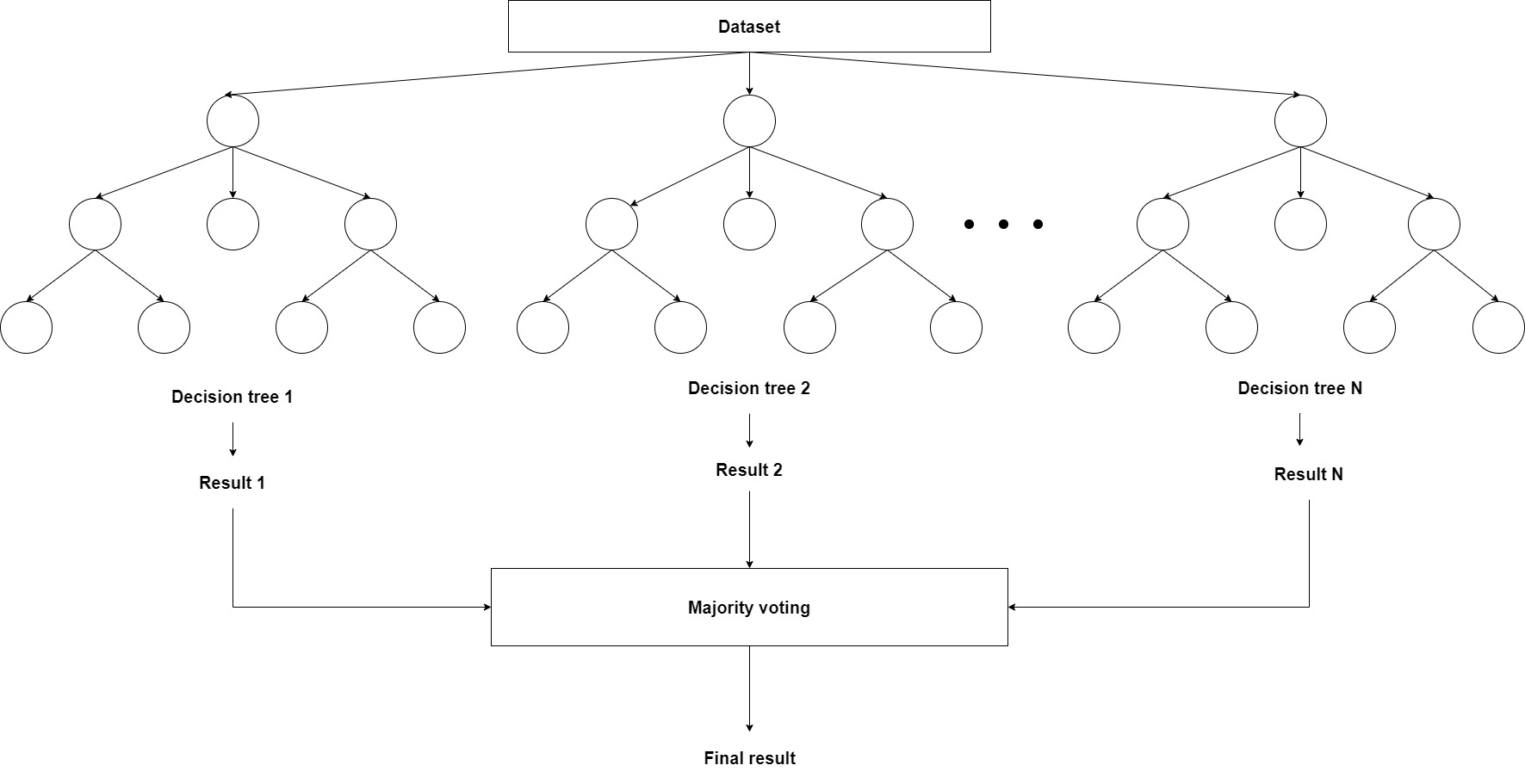


**Random Forest:**

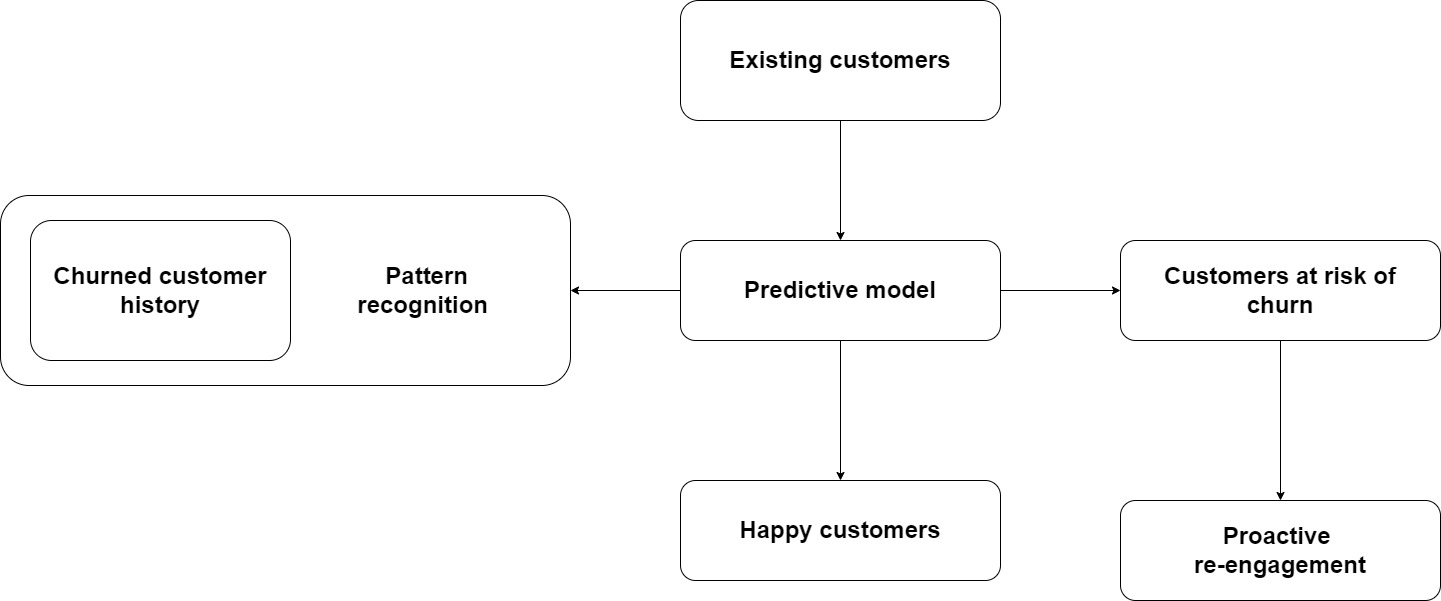
Random forest is a machine learning technique to solve regression and classification problems. It uses ensemble learning, a technique that combines many classifiers to provide solutions to complex problems.

A random forest algorithm consists of many decision trees. The “forest” created by the random forest algorithm is trained by bagging or bootstrap aggregation. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms. A (random forest) algorithm determines an outcome based on the predictions of a decision tree. Predict by averaging outputs from different trees. Increasing the number of trees improves the accuracy of the results.

Random forest removes the limitations of decision tree algorithms. Reduce data set overfitting and increase accuracy. Generate predictions without requiring a lot of configuration in your package



**PROJECT WORKFLOW:**

****