**1. Introduction**

In today's fast-paced, rapidly growing, and highly competitive business world, human capital is perhaps one of the most valuable assets a company might have. Workers directly influence productivity, culture, and overall success of a company. Even with that, numerous businesses find it challenging to retain their workers. When a worker leaves, not only is workflow dynamics disrupted, but it also results in substantial costs associated with replacing them, training, and bringing them onboard. Due to this, predicting employee turnover that is, the likelihood an employee will depart from the company has emerged as a leading priority for firms of all sizes.

Previously, companies have relied on manual methods like exit interviews and surveys to establish why employees leave. Though these can provide some information, they are too late to prevent turnover and are limited in scope. A silver lining to this issue is that the advent of data science and machine learning offers a new way to solve this issue. By analyzing past data about employees, we can search for patterns to forecast whether an existing employee will be prone to leave. Therefore, organizations can actively try to improve the happiness of employees, their engagement, and retention.

This project revolves around using machine learning models to research a dataset which contains information regarding employees. The dataset contains fields such as the level of education, gender, age, city, number of years in current field, and whether or not the employee was ever benched (i.e., not placed on a project for a stretch of time). All of that information helps us build a more solid concept of what could potentially make someone leave.

The general goal of this project is to create accurate and reliable predictive models from this employee data. These models will enable us to estimate the value of a target variable called LeaveOrNot. This column in the data indicates whether an employee has left the company (value 1) or not (value 0). We want to create models that can analyze all the other features of an employee and predict this outcome as accurately as possible.

By properly predicting employee attrition, firms can cut costs, ease turnover, and create a healthful working environment. Moreover, the identification of reasons underlying employee attrition helps management in decision-making for workplace policies, employee engagement, and career advancement opportunities. With this, evidence-based decisions may not only prove to be advantageous to individual employees but can also aid towards organizational long-term success.

Over the course of this report, I will walk you through each phase of the machine learning process, from understanding datasets to model creation and comparison. By the final page, we will have a good sense of how well machine learning can help predict employee attrition and how this can be applied in the real world to improve HR decision-making.

**Dataset Overview**

The data used in this project is a data set of employee data with various attributes relating to his or her professional and demographic background. These attributes give details regarding the background of the employee, work experience, and potential factors influencing his or her intention to stay at or leave the company. The next is a detailed description of each attribute

**Key Features:**

**Education:**

**Description:** It represents the employee's highest level of education. It can be a categorical variable with levels such as 'High School', 'Bachelor's', 'Master's', or 'Ph.D.'.

**Importance:** Educational level usually corresponds to one's career orientation, decision-making, and expectations in an organization. Employees holding higher levels of education can have different job expectations or career orientation than employees holding lower levels of education.

**City:**

**Description:** The geographic location where the employee works or lives. It can be represented as a categorical variable with values such as 'City A', 'City B', etc.

**Importance**: City may be a factor in employee turnover. Employees working in cities with greater living expenses, for instance, may have varying levels of job satisfaction or career advancement opportunities, which could affect their probability of staying or leaving.

**Gender:**

**Description:** Refers to the gender of the employee, usually as 'Female' or 'Male', although in certain information it may also have 'Other' or 'Non-binary'.

**Importance:** Occasionally gender can be tied to employee history in the workforce, though the latter is controversial and sensitive as a topic. It's helpful to record whether there are gendered trends among those leaving employment.

**Age:**

**Description:** Employee age.

**Importance**: Age can affect an employee's choice to quit. Younger workers might have greater turnover, looking for additional opportunities or advancement, while older employees might be looking for stability or retirement options. Age is a significant driver to predict career satisfaction and probability of leaving.

**ExperienceInCurrentDomain:**

**Description:** The number of years the employee has worked in their current job or field.

**Importance**: More experienced staff might be less likely to leave, as they are invested in the business and possess domain expertise. However, high-experience workers can leave when they perceive that they are being undervalued and are not challenged anymore.

**EverBenched:**

**Description:** Refers to whether the employee ever has been on the bench (i.e., not working on any project).

**Importance:** Being benched is likely to be among the major reasons why an employee quits a firm. Benched workers are most likely to feel neglected, under-valued, or demotivated and therefore more likely to quit the firm.

**Target Variable:**

**LeaveOrNot:**

**Description:** This is our target variable to predict. It indicates whether a worker left the company (1) or stayed (0).

**Importance:** The general goal of this project is to predict worker turnover, which has significant implications for HR management. Understanding why workers are leaving can allow HR departments to act pre-emptively in reducing turnover, improving retention, and increasing job satisfaction.

**Data Preprocessing**

The raw data must be cleaned and transformed into model-understandable format before we can train machine learning models. This is called data preprocessing. It ensures the dataset is stable, complete, and structured for analysis. Below is a description of the main preprocessing steps that have been used in this project:

**1. Handling Missing Values**

Sometimes datasets hold missing data—e.g., an employee's age or education level may never have been recorded. Missing pieces here can negatively impact the accuracy of our models.

* Carefully checked for missing values in each column.
* Where columns were numerical like Age or Experience, were substituted with the average or median value.
* For categorical columns like Gender or City, replaced missing entries using the most occurring category.

**2. Encoding: Converting Categorical Data into Numbers**

Some columns in the data, like City and Gender, contain text values. But machine learning processes work with numbers, not with words—so categorical variables had to be converted to numerical numbers.

For columns like Gender (with only two categories), we used Label Encoding. This is giving each category a number (for example, Male = 0, Female = 1).

For columns with over two categories, like City, One-Hot Encoding was used, which creates a separate column for each category.

By doing this, made sure the models could interpret and process all columns correctly.

**3. Feature Scaling**

The dataset has values which are in a different range or scale. For example, Age can be anything between 20 and 60, but Experience can range between 0 to 10. These scales will affect how the models can be learned.

An approach called Standard Scaling that made all the numeric features' scale have an average of 0 and standard deviation of 1.

This guarantees that every numerical feature contributes equally to the model's learning process.

Scaling guarantees improved performance and stability of certain algorithms, especially those involving distance or gradient computation (e.g., Logistic Regression).

**4. Selecting the Most Significant Features**

Not all the features are created equal when it comes to predicting whether an employee will leave. Some may be of little or even detrimental effect.

Correlations between each feature and the target were checked (LeaveOrNot).

Tools like feature importance scores from models like Random Forest or XGBoost were used to see which features were most significant.

Deleting unnecessary features can make the model faster, easier, and more understandable.

**5. Preparing the Final Dataset**

Once the dataset was converted and cleaned, it was divided into:

**Features (X):** All the input variables that are used to predict things (Age, Gender, City).

**Target (y):** The variable that is to be predicted — in this case, if the employee quit (1) or not (0).

This division is very important when we are going to train and test the model.

**6. Separation of Data into Training and Testing Sets**

To verify how well models perform, dataset was divided into two:

**Training Set (80%):** Used to train the model by exposing it to instances of employee traits and outcomes.

**Testing Set (20%):** Used to test the performance of the model on unseen data.

train\_test\_split function from Scikit-learn was used to accomplish this. Separating the data this way ensures that model is not just memorizing the training set but can actually generalize and make predictions on real cases.

After preprocessing, a clean, consistent, and well-structured dataset was accomplished:

* All missing values had been handled.
* All features were numeric and scaled properly.
* Irrelevant or extreme values had been handled.

The dataset had been split into training and testing datasets for model development.

**Exploratory Data Analysis (EDA)**

Exploratory Data Analysis (EDA) is a crucial step in every data science project. It helps us understand the structure, patterns, relationships, and any issues with the dataset before we build machine learning models.

Think of EDA as a way of getting to know our data—how it looks, what's normal, what's abnormal, and what can affect the result we're trying to predict. Here is how we performed it in this project:

**1. Summary Statistics**

We began with descriptive statistics in order to have a broad idea of each feature:

**Mean (average):** Helps us understand the central value (e.g., average age of an employee).

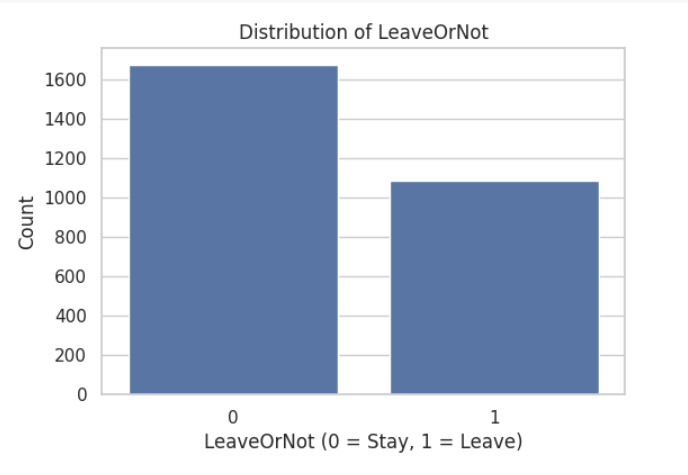
**Median:** The middle value, giving a clearer picture than the mean in the presence of outliers.

**Minimum and Maximum values:** Help to identify outliers.

**Standard Deviation:** It tells us about how dispersed the data is.

These figures give a quick but good overview of the data set and allow us to identify any abnormalities, such as very high or low values that may affect our models.

# Fig 1: Plot distribution of target variable



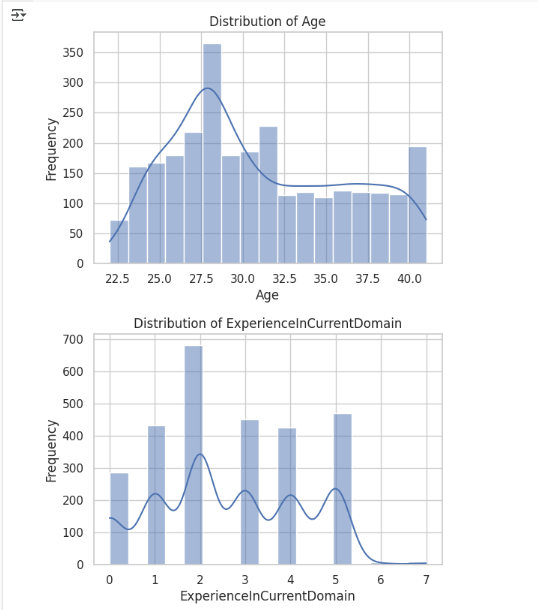
Interpretation: This graph provides a visual summary of employee attrition, and more precisely, the count of workers who chose to depart from the company versus the count of workers who chose to stay. The plotting variable is known as LeaveOrNot, which has two possible values:

**0:** Indicates that the worker stayed with the company.

**1:** Indicates that the worker departed from the company.

* On the x-axis, it is easy to identify these two groups. On the y-axis, it is easy to recognize the count of employees within each group as simply counts.
* The chart clearly allows us to note that: '0' (Stayed) bar is clearly higher than the '1' bar.
* We can understand from this that in the data set, there are more employees staying with the organization than the one who leaves.
* Overall, this chart tells us that the majority of the employees stayed, but there is still a large chunk who left, and that provides a good foundation to build a predictive model on employee attrition.

Fig 2:



Interpretation:

**a. Age Distribution**

This chart indicates the distribution of employee ages within the dataset. The x-axis presents the age of employees, which goes from approximately 22 to 40 years. The y-axis represents frequency, or the number of employees in each age category.

From the chart, we can see:

The most common age is 27 to 30 years, with a peak at 28 years, where frequency is highest. There is a consistent decline in frequency as age exceeds 30, indicating fewer older employees are represented in the data. The curve drawn over the bars (a KDE curve) provides the smoothed sense of the distribution, emphasizing that the majority of employees are in their late 20s to early 30s.

The distribution is right-skewed, suggesting a younger employee population, with decreasing numbers of employees in the older age brackets (35+). This information is useful as age is usually a deciding factor in most employee decisions, such as staying or leaving an organization.

**b. Distribution of Experience in Current Domain**

This chart shows how many years of experience employees have in their current work domain. The x-axis shows the number of years of experience from 0 to 7 years. The y-axis shows the number of employees that fall under each experience level.

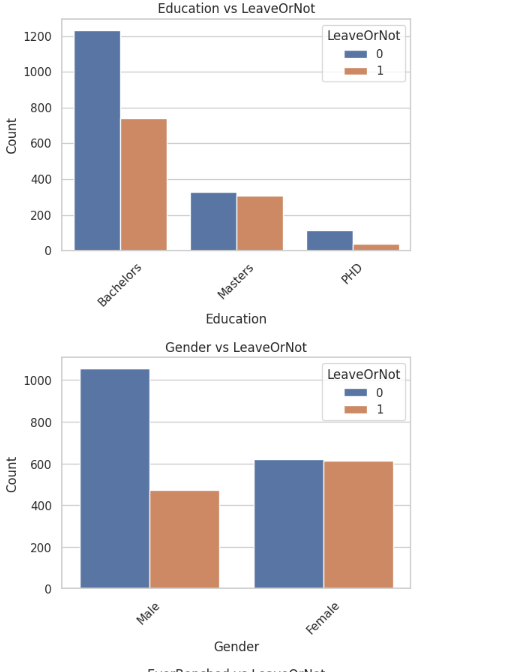
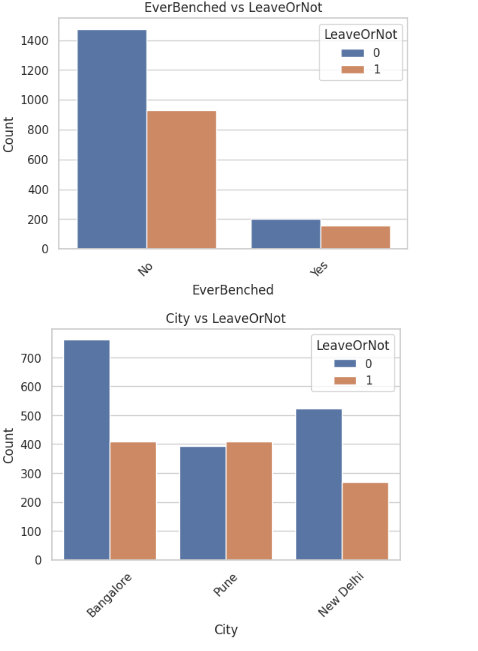
**Key findings:**

There are many employees in the 2 years of experience, which is the highest bar in the chart. Some employees also have 1, 3, 4, and 5 years of experience, showing a diverse distribution of early- to mid-level experience. There is a fall-off in employees with 6 or more years of experience.

The distribution is multi-peaked, which can mean there are several entry points or career plateaus in the company hierarchy. Overall, this chart reveals that the dataset is dominated by employees with low to moderate domain experience, which may influence their job stability and likelihood of leaving. Less experienced workers may be in the early stages of career exploration or job hopping.

Collectively, these charts provide a baseline understanding of the employee profile — young individuals with comparatively fewer years of domain-related experience — which are both relevant factors when exploring trends related to employee attrition.

Fig 3

**a. EverBenched vs. LeaveOrNot**

This bar graph shows the relationship between whether or not an employee has ever been benched (i.e., taken out of active projects briefly) and whether they tend to leave the company or not. The "No" column shows a significantly greater number of employees staying than leaving. Of those employees who were benched at some point in time, the proportion of employees leaving is much more similar to that of those who remained, but still lower. This would mean benching would be associated with an increased risk of attrition, possibly for job dissatisfaction or perceived job insecurity.

**b. City vs. LeaveOrNot**

The chart displays the distribution of employee retention and attrition in three cities: Bangalore, Pune, and New Delhi. The largest overall employee number is in Bangalore, with a huge difference between the employees retained and those lost. Pune has very close numbers for both categories, while New Delhi has a larger retention than attrition but with a larger difference than Pune. This could imply that geographical location is a factor in employee turnover to a certain degree, possibly because of differences in job markets, cost of living, or working culture among cities.

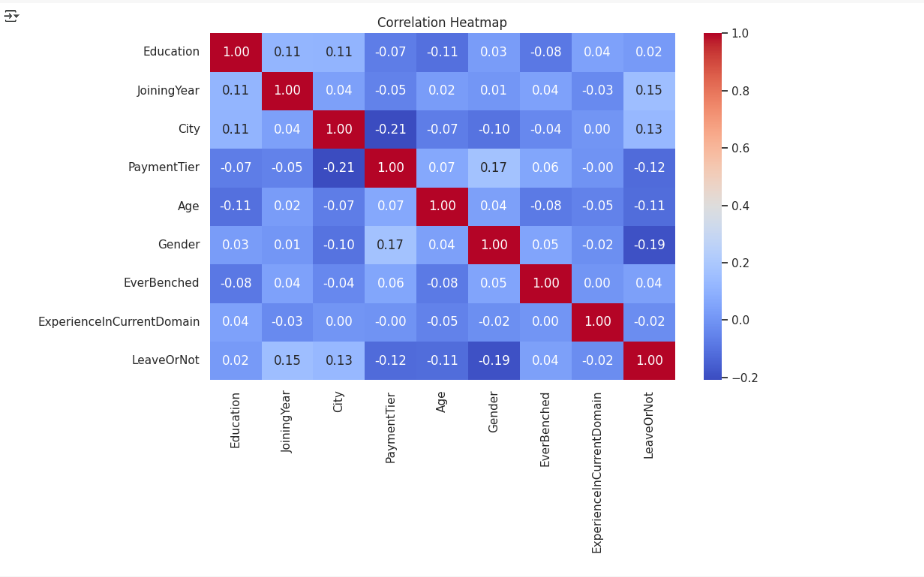
**c. Education vs. LeaveOrNot**

This graph analyzes the relationship between employee turnover and education level. Employees with Bachelor's degrees comprise the majority of the dataset and have the highest number of both staying and leaving. The category who possess Master's degrees have relatively balanced proportions, while PhD holders have the lowest number in total and a much lower attrition. This may suggest that more advanced degrees like a PhD are associated with better retention, perhaps from more senior roles or job satisfaction, whereas those holding a Bachelor's degree might have more career mobility or constricted development, so that they leave.

**d. Gender vs. LeaveOrNot**

This bar graph contrasts gender with employee turnover. For male workers, significantly more stayed than left, whereas that for women is nearly equal. This pattern may be explained by variations in job roles, support networks, or other external factors affecting women's retention more significantly, e.g., work-life balance or organizational culture. These results can guide organizations to design gender-specific retention strategies.

Fig 4:



The correlation heatmap gives a graphical representation of the correlation between various numerical and encoded categorical features of the dataset. The correlation is between -1 and 1, with numbers close to 1 indicating strong positive correlation, numbers close to -1 indicating strong negative correlation, and numbers close to 0 indicating little or no linear relationship between variables.

**Key Observations:**

Low Overall Correlation with Target (LeaveOrNot): None of the features have a high correlation with the target variable LeaveOrNot.

**Gender (-0.19 correlation with LeaveOrNot):**

A moderately negative correlation suggests that gender could be an issue with attrition, and one gender is slightly more likely to depart than the other. The correlation is not strong enough, however, to make anything conclusive without further analysis.

**Joining Year (0.15 correlation with LeaveOrNot):**

Moderate positive correlation indicates that the date an employee started working for the company may have an influence on whether or not they are going to leave, perhaps due to changes in organizational policies and expectations from the workforce over time.

**City (0.13 correlation with LeaveOrNot):**

A correlation by location, where employees in certain cities might have more job mobility or other opportunities, which would influence their stay or leave decision.

**Payment Tier (-0.12 correlation with LeaveOrNot):**

There is a weak negative correlation, suggesting that staff in lower-paid brackets are maybe more likely to leave, perhaps due to dissatisfaction with pay or career advancement opportunities.

**Age (-0.11 correlation with LeaveOrNot):**

Older employees might be that little bit more settled or committed, and this leads to a slightly reduced level of attrition.

**Education, EverBenched, and ExperienceInCurrentDomain:**

All have extremely close-to-zero correlation with the target variable, and thus no direct linear effect on attrition in themselves.

Some traits have low correlations with each other, which are:

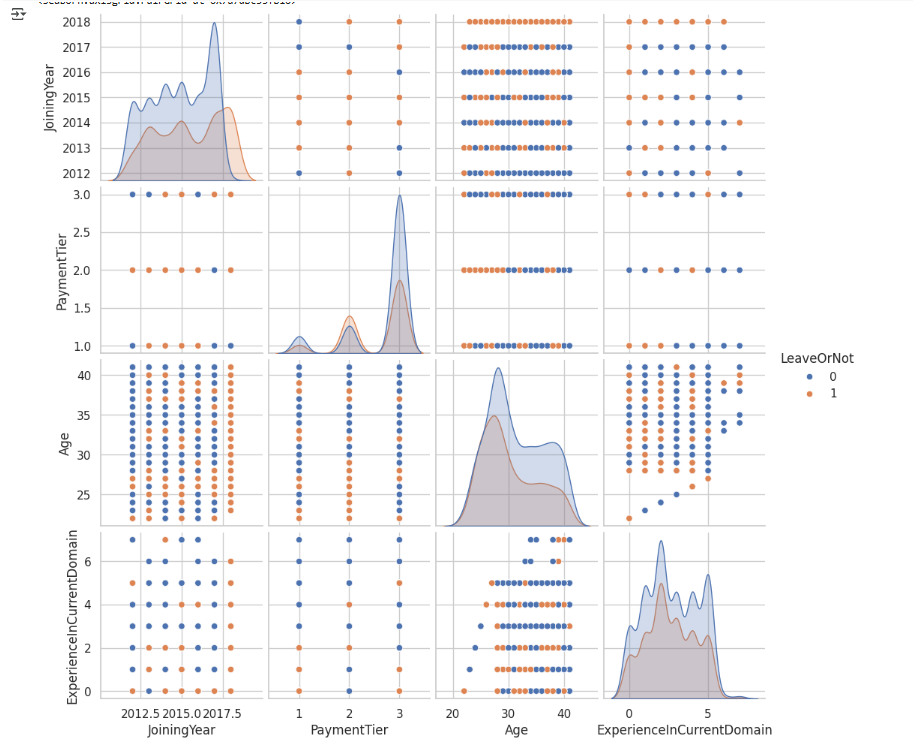
**PaymentTier and Gender (0.17)**

**Education and City (0.11)**

These can indicate structural in the data set, such as the genders being spread out over pay levels.

This heatmap is a first diagnostic tool to understand the dataset. While it shows broad trends, it does not expose non-linear interactions or variable interactions. Hence, more advanced tools like feature importance analysis and SHAP values can expose more about how each variable contributes to model predictions.

Fig 5:



This is a pair plot used to visually explore relationships between different employee features in a dataset, distinguishing between those who left the company (LeaveOrNot = 1) and those who remained (LeaveOrNot = 0). Both groups are color-coded: the one who stays are blue and who leaves are orange.

**Variables Examined**

JoiningYear, PaymentTier, Age, ExperienceInCurrentDomain, LeaveOrNot

**Diagonal Elements**

Every diagonal cell shows a distribution (density plot) of one variable divided by whether the employees left or not:

**JoiningYear**: More employees joined in 2016 and 2017. Those remaining are more likely to have joined recently.

**PaymentTier:** The majority of employees are Tier 1. Those in higher tiers (2 or 3) are slightly more likely to leave.

**Age:** Most employees are aged between 25 and 35. Density of stayees is higher in that group.

**ExperienceInCurrentDomain:** Most employees have 1–5 years of experience. More experience weakly corresponds to staying.

**Off-Diagonal Elements**

These show scatter plots between two variables, color-coded by leave or not. Some observations worth noting: There's a high concentration of those who stayed among newer hires (2015–2017), especially at younger ages and lower levels of experience. Higher Pay is more likely to have left (more orange dots in Tier 2 and 3). Age is strongly correlated with experience — older employees have more domain experience, and most of them stayed.

**Overall Insight is:**

Younger employees with low payment levels and fewer years of experience in the domain have higher chances of staying. On the other hand, older, more experienced, and higher-paid employees have a relatively higher chance of leaving.

**Model Selection**

In order to figure out which algorithm is most effective at predicting whether or not an employee remains at the company, we compared and contrasted three machine learning models. Every model has its own strengths, and by attempting a number of various approaches, we have a better idea of which algorithm is most suitable for our data and type of problem.

**1. Logistic Regression**

Logistic Regression is a simple, popular, and widely used algorithm for binary classification problems—where the output can be one of two classes, in our case: Leave (1) or Stay (0). It approximates the relationship between the features (such as age, experience, etc.) and the probability of the target class through a sigmoid function.

**2. Random Forest Classifier**

Random Forest is an ensemble method that builds multiple decision trees and combines their predictions to improve accuracy and avoid overfitting. Each tree is fitted to a random subset of data and features, which improves robustness.

**3. XGBoost Classifier**

XGBoost (Extreme Gradient Boosting) is an extremely efficient gradient boosting approach that is well-known for its performance, speed, and scalability. It builds trees sequentially, with each subsequent tree trying to correct the errors of the previous one, and includes regularization techniques to avoid overfitting.

XGBoost is a top performer in most practical applications and machine learning competitions because of its excellent trade-off between flexibility, efficiency, and accuracy.

**Why Multiple Models?**

* By attempting a range of models—from simple to complex—we can:
* Successfully benchmark performance.
* Observe how model complexity relates to predictive power.

Make an informed decision taking into account both accuracy and pragmatic considerations like interpretability, training time