upGrad ENTERPRISE

# Machine Learning

24 July, 2024

upGrad ENTERPRISE

Course : Machine Learning

Lecture On : Regressions

Instructor : Prabodh Ranjan

# In Today's session, we will cover:

01 What is sklearn

02 Fundamentals :

(1) Maths

(2) Machine Learning

03 Machine Learning : Regression using sklearn

# What is ‘scikit-learn’ or ‘sklearn’ ?

Scikit-learn (often referred to as sklearn) is one of the most popular Python libraries for machine learning.

Scikit-learn is an open-source Python library that provides a range of supervised and unsupervised learning algorithms.

The name "scikit" refers to "SciPy Toolkit" – and this library builds on the capabilities of other foundational Python libraries like NumPy and SciPy

It provides tools for data mining and data analysis, and supports both supervised and unsupervised learning, as well as various utilities for model fitting, data preprocessing, model evaluation, and more

**Scikit Learn**

# Why is scikit-learn beneficial?

**Versatility**: scikit-learn offers tools for various tasks in the machine learning workflow, from data preprocessing to model evaluation.

**Ease of Use**: With its consistent API design, once you understand how to work with one type of model in scikit-learn, you can easily apply that knowledge to other models.

**Documentation**: The library is well-documented, with many examples and tutorials available online. This means you can often find guidance or solutions to problems you might encounter.

**Community Support**: Being open-source and popular means there's a large community of users and contributors. This ensures continuous improvements and updates to the library, as well as help when you run into issues.

**Performance**: While scikit-learn is written in Python, it's built on top of libraries like NumPy that are implemented in C. This means that the heavy computations are optimized and run at C-speed, giving a good balance of ease of use and performance.

# How can scikit-learn help in machine learning?

- **Algorithms**: It provides a wide range of algorithms, from classic statistical models to cutting-edge machine learning techniques.
- **Preprocessing**: Data rarely comes in the perfect format for direct use in machine learning models. scikit-learn offers tools for normalization, scaling, encoding categorical variables, and more.
- **Model Selection**: With utilities to split datasets, perform cross-validation, and tune hyperparameters, scikit-learn makes it easier to find the best model for your data.
- **Evaluation**: After training a model, it's crucial to understand how well it's performing. scikit-learn provides tools to evaluate models using various metrics, depending on the type of problem (classification, regression, clustering, etc.).
- **Pipeline**: This is a feature that allows you to create a sequence of data processing steps and modeling, ensuring that all steps are executed in the correct order.

- NOTE: While scikit-learn is fantastic for traditional machine learning models, it's not designed for deep learning. For deep learning tasks, libraries like TensorFlow and PyTorch are more suitable.

Applications of sklearn

## Supervised Learning

- Linear Regression
- Logistic Regression
- Decision Trees and Random Forests
- Support Vector Machine
- Gradient Boosting classifier
- K Neighbors Classifier

## Unsupervised Learning

- K-means Clustering
- Principal Component Analysis (PCA)
- DBSCAN for Clustering

## Visualization

- Cross-validation
- Confusion Matrix and Classification Report

# Supervised Learning

- Subcategory of machine learning.
- Algorithms learn from labeled training data.
- Provided with input data (features) and corresponding desired outputs (labels).
- Aims to identify patterns and relationships between features and labels.
- Adjusts internal parameters to minimize the difference between predicted outputs and actual labels
- Uses optimization techniques like gradient descent for parameter adjustment.
- Makes predictions on new, unseen data based on learned patterns.
- Widely used in applications such as image classification, spam detection, sentiment analysis, etc.
- Requires labeled training data for accurate predictions.

upGrad ENTERPRISE

# Supervised Learning

Some common methods or functions in sklearn that are used for supervised learning include:

## Model selection and evaluation:

A. **train_test_split**: Splits the dataset into training and testing subsets.

B. **cross_val_score**: Performs cross-validation to evaluate model performance.

C. **GridSearchCV**: Performs hyperparameter tuning using a grid search approach.

## Preprocessing and feature engineering:

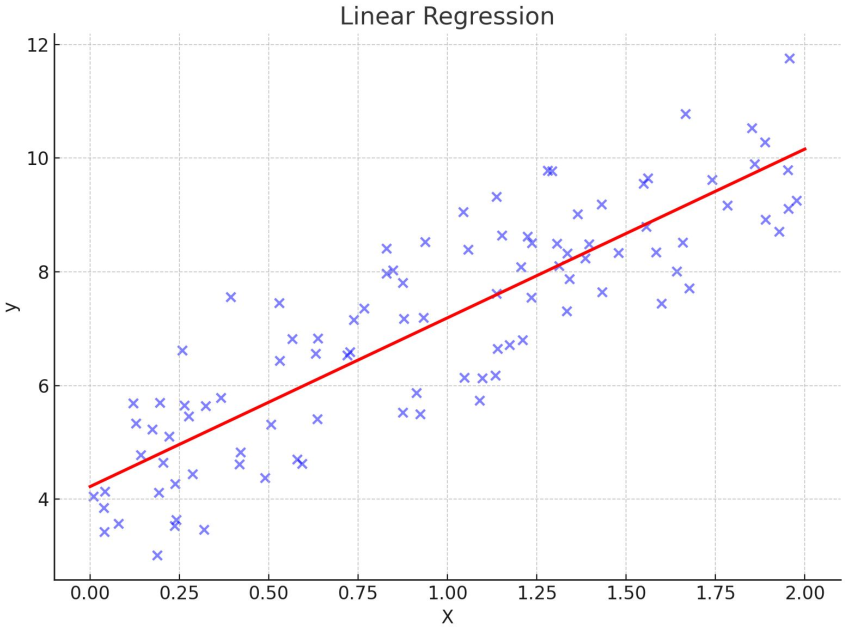
A. **StandardScaler**: Standardizes features by removing the mean and scaling to unit variance.

B. **MinMaxScaler**: Scales features to a specified range (e.g., 0 to 1).

C. **OneHotEncoder**: Converts categorical variables into a binary sparse matrix.

## Supervised learning algorithms:

- **LinearRegression**: Implements linear regression for continuous target variables.
- **LogisticRegression**: Performs logistic regression for binary classification tasks.
- **DecisionTreeClassifier**: Builds a decision tree for classification.
- **RandomForestClassifier**: Creates an ensemble of decision trees for classification.
- **SupportVectorMachine**: Implements Support Vector Machine (SVM) for classification tasks.
- **GradientBoostingClassifier**: Implements gradient boosting for classification.
- **KNeighborsClassifier**: Performs k-nearest neighbors classification.



# Supervised Learning
# - Linear Regression

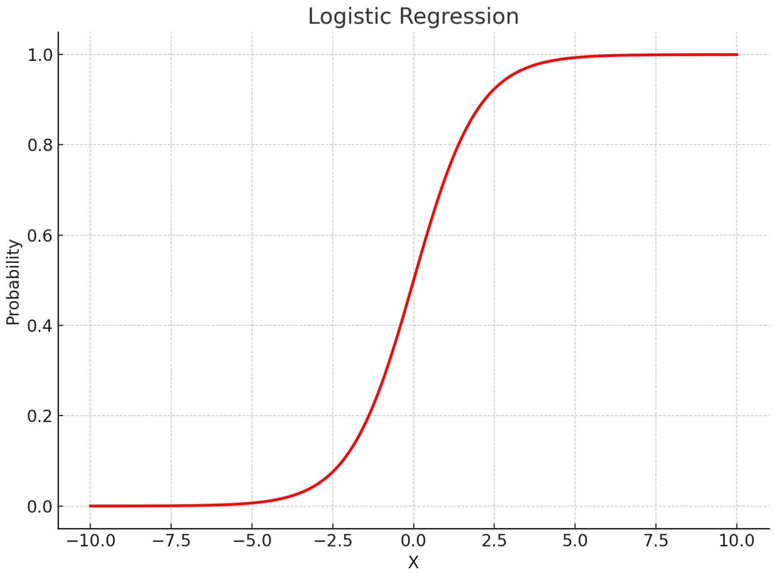
## When to Use:

- Use linear regression when you have a continuous target variable and you want to predict its value based on one or more independent variables.

## Explanation:

- Linear regression tries to find the best linear relationship (a straight line) between the dependent (target) and independent (features) variables.

# Supervised Learning
# - Logistic Regression



## When to Use:

- Use logistic regression when you have a binary classification problem, i.e., when the target variable has two classes/categories.

## Explanation:

- Logistic regression predicts the probability that a given instance belongs to a particular category. It outputs values between 0 and 1..

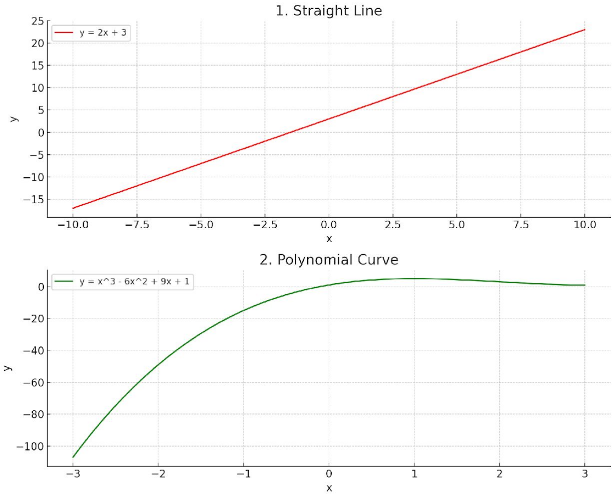
# Maths – Fundamentals Underfitting and over-fitting

- **Underfitting** occurs when a model is too simple to capture the underlying pattern of the data. This usually happens when the model has too few parameters (e.g., a linear model trying to fit non-linear data). The model performs poorly on both the training data and unseen data.

- **Overfitting** happens when a model is too complex relative to the amount and noisiness of the training data. The model learns the noise in the training data as if it were a real pattern, leading to poor performance on new, unseen data despite potentially excellent performance on the training data.

# Maths – Fundamentals

## Straight Line and curves

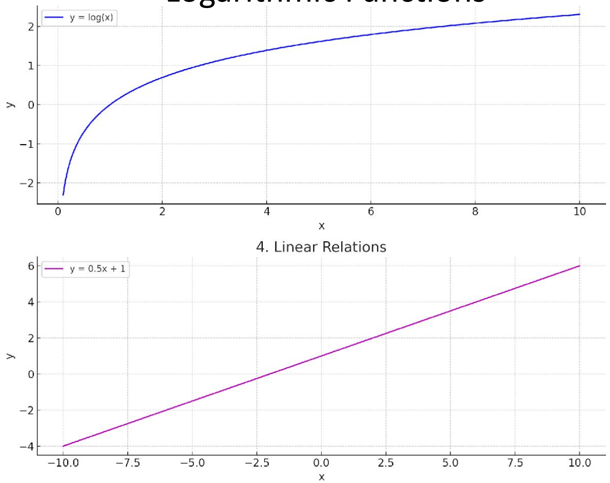


Straight Line : It represents a linear relationship where y changes at a constant rate with x.

Polynomial Curve: Polynomial curves represent more complex relationships than linear ones, allowing for the modeling of curves in data.

# Maths – Fundamentals

Logarithmic Functions

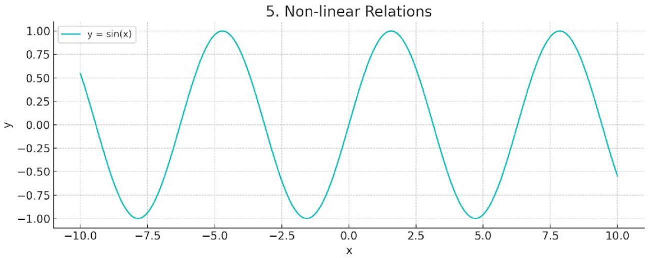


Logarithmic functions: They are useful for modeling phenomena that grow rapidly at first and then slow down as they progress.

Linear relations: This refer to a direct relationship between two variables where if one variable changes, the other variable changes at a constant rate. This relationship is typically represented by a straight line in a graph when plotted

# Maths – Fundamentals

## Straight Line and curves



Non-linear relations refer to relationships between two variables where the change in one variable does not correspond to a constant change in the other variable.

This means that the rate of change of the dependent variable with respect to the independent variable is not constant, leading to graphs that are not straight lines.

Non-linear relationships can manifest in various forms, including curves (such as parabolas, hyperbolas), oscillations (like sine waves), exponential growth or decay, and more complex patterns.

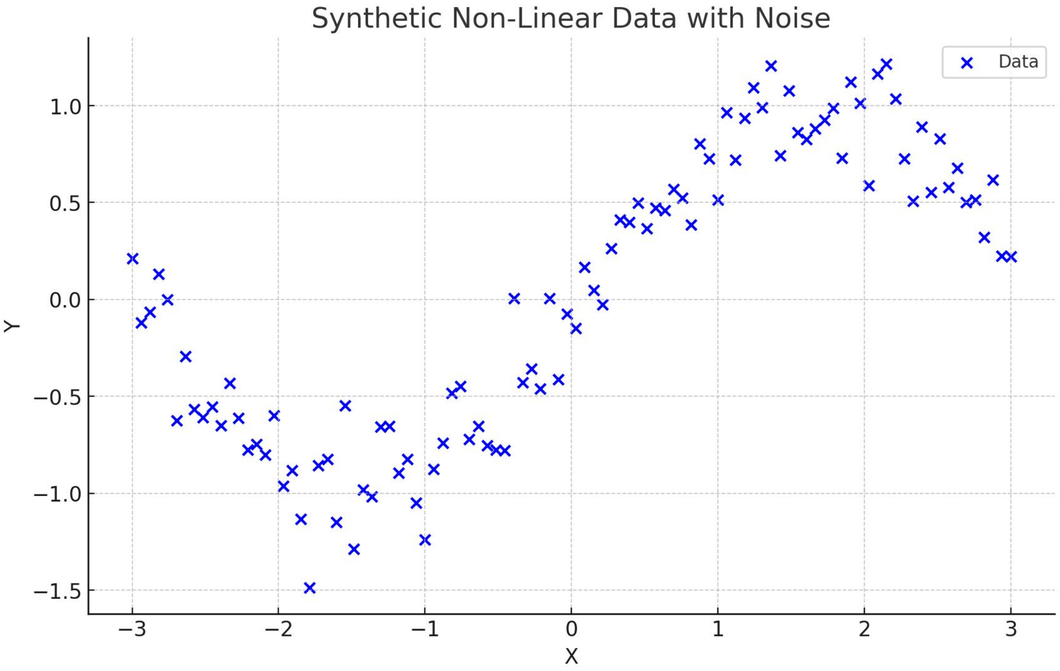
Key characteristics of non-linear relationships include:

Variable Rate of Change: The rate at which y changes in response to changes in x is not constant and can vary across the domain of x.

Curved Graphs: When plotted on a coordinate plane, non-linear equations typically result in curved lines that can take various shapes, such as U-shapes, S-shapes, waves, and more.

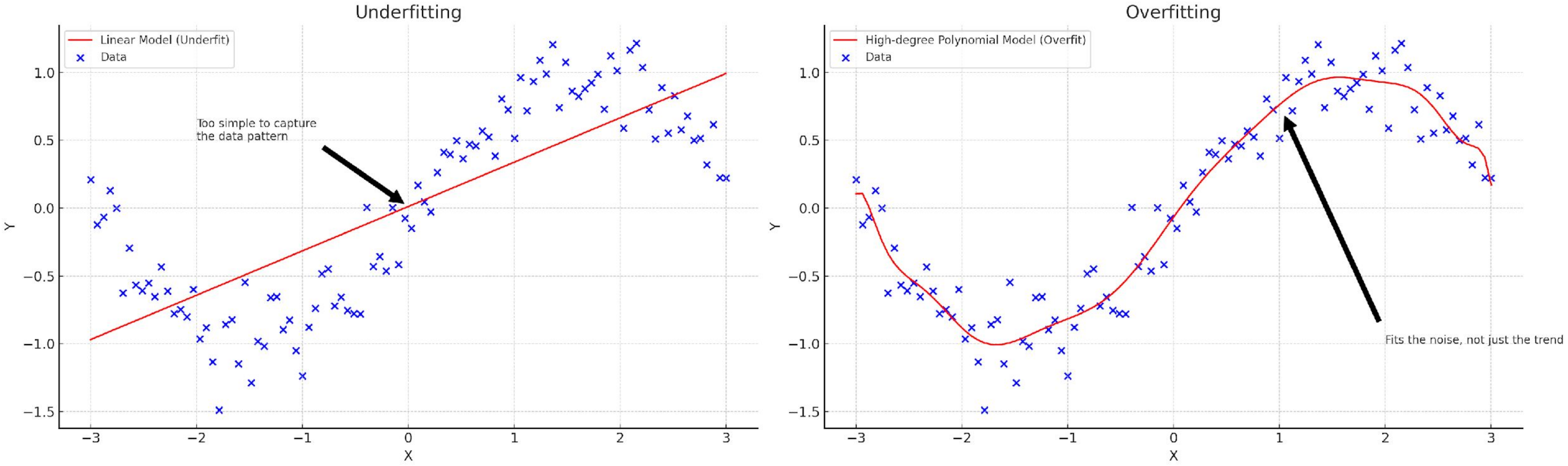
Complexity in Modeling: Non-linear relationships can model more complex phenomena than linear relationships, capturing intricate patterns in data that linear models cannot.

# Underfitting and over-fitting



Maths – Fundamentals

# Underfitting and over-fitting



# ML – Fundamentals

## Arithmetic Mean

Definition: The arithmetic mean is the most commonly used type of average. It is calculated by summing a set of values and then dividing by the count of values.

Practicality: The arithmetic mean is used when you want a central or typical value for a set of numbers. It is sensitive to outliers and can be skewed by values that are significantly higher or lower than the rest of the data set.

## Geometric Mean

Definition: The geometric mean is a type of average that is calculated by multiplying all the values together and then taking the nth root (where n is the count of values).

Practicality: The geometric mean is used for sets of positive numbers and is particularly useful for rates of change or percentages, like growth rates, as it tends to dampen the effect of very large or small values.

# ML – Fundamentals

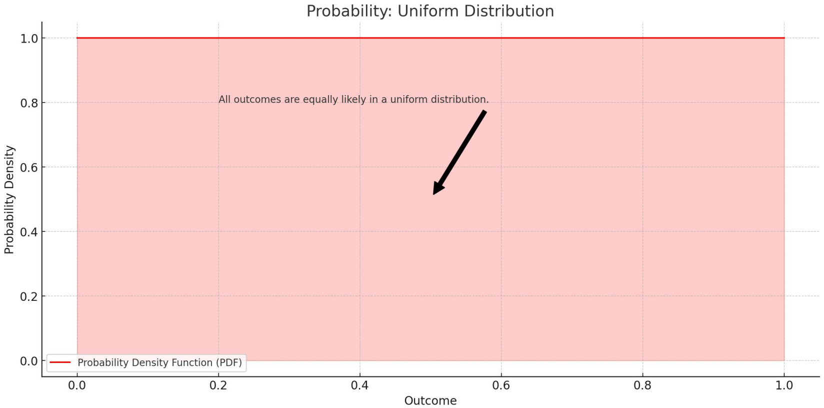
**Harmonic Mean**

Definition: The harmonic mean is a type of average which is calculated by dividing the count of values by the sum of the reciprocals of the values.

Practicality: The harmonic mean is useful when the values in a dataset are rates or ratios. In the context of the F1 score, which is the harmonic mean of precision and recall, it provides a single measure that balances these two metrics, punishing extreme values more than the arithmetic mean would.

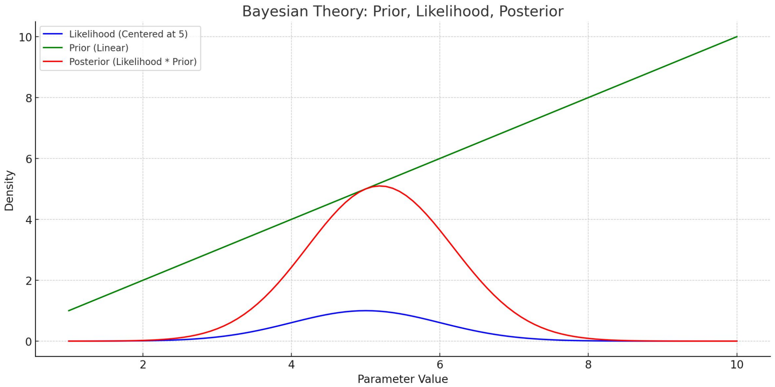
# Maths – Fundamentals

## Probability and Bayesian Theory



**Probability: Uniform Distribution**

This plot shows a uniform probability distribution, where all outcomes are equally likely. The probability density function (PDF) is constant across all outcomes, indicating that no single outcome is more likely than another within this range. The shaded area under the curve represents the probability of outcomes within a specific range, illustrating how probability is distributed across different outcomes.



**Bayesian Theory: Prior, Likelihood, Posterior**

This plot illustrates the fundamental components of Bayesian theory:

- **Prior**: Represents our initial belief about the parameter before observing any data, shown as a linear increase in density with the parameter value.
- **Likelihood**: Represents how probable the observed data is for different parameter values, assuming a Gaussian distribution centered at 5 for simplicity.
- **Posterior**: Combines the prior belief and the evidence from the data (likelihood), shown as the product of the prior and likelihood. The posterior distribution reflects our updated belief about the parameter after considering the data

# Exploring Beyond Simple Equations
# The Limitations of Basic Tools

Complexity of Real-World Data: Real-world data is often too complex to be accurately modeled with simple equations.

While certain tools like Microsoft Excel can handle straight lines, logarithmic, sine functions, and polynomial curves, they fall short when data exhibits non-linear patterns, high dimensionality, or when relationships between variables are not straightforward.

Inadequate for Advanced Analysis: Basic tools lack the sophisticated algorithms required for tasks like image recognition, natural language processing, and predicting trends based on large datasets.

These tasks require analyzing vast amounts of data in ways simple equations cannot manage effectively.

# The Imperative for Machine Learning Overcoming the Challenges

Achieving Accuracy and Efficiency: Despite the higher initial costs associated with tagging data, requiring powerful computational resources, and the steep learning curve, machine learning algorithms can achieve levels of accuracy and efficiency that justify the investment.

They can automate tasks that would be impractical or impossible to perform manually or with simpler analytical methods.

Innovation and Competitive Advantage: The use of machine learning is not just about handling data or automating tasks; it's a strategic investment.

Businesses and technologies that leverage machine learning gain a competitive edge through innovation, better customer insights, and the ability to solve complex problems.

# The Imperative for Machine Learning Overcoming the Challenges

Handling Complexity and Volume: Machine learning algorithms are designed to navigate through and make sense of the complexity and volume of data that today's world generates. They can identify patterns and insights that are not apparent through traditional methods.

Adaptability and Learning: Unlike static equations, machine learning models can improve and adapt over time as they are exposed to more data. This learning capability is crucial for applications where conditions and requirements evolve.

Automating Decision-Making: Machine learning enables the automation of decision-making processes in real-time, which is essential for applications such as autonomous driving, fraud detection, and personalized recommendations.

# The Imperative for Machine Learning Overcoming the Challenges

While the path to implementing machine learning involves challenges, including time, budget, and computational resources, the capabilities it unlocks are indispensable for modern applications.

*Machine learning is not just another tool*

It's a transformative technology that is reshaping industries, enhancing decision-making, and opening new frontiers in innovation and efficiency.

# Machine learning  - Use cases

Forecasting market trends, weather patterns, or equipment failures before they happen.

Automate decision-making

Envision systems that make intelligent decisions, from diagnosing medical conditions to optimizing traffic flow in real-time.

Make sense of vast amounts of data?

Consider the power of extracting valuable insights from the data deluge in every domain, be it social media analytics, genomics, or customer behavior.

Think about augmenting human efforts with AI, such as enhancing diagnostic accuracy in healthcare or improving efficiency in manufacturing.

Make computers learn from experience

Ponder over creating systems that improve over time, learning from past actions, mistakes, and successes, much like humans do..

# Key ML Terminology

**Labels**

- Think of a label as the answer we're trying to predict.
- It can be anything, like the future price of a product, the type of animal in a photo, or what an audio clip is saying.

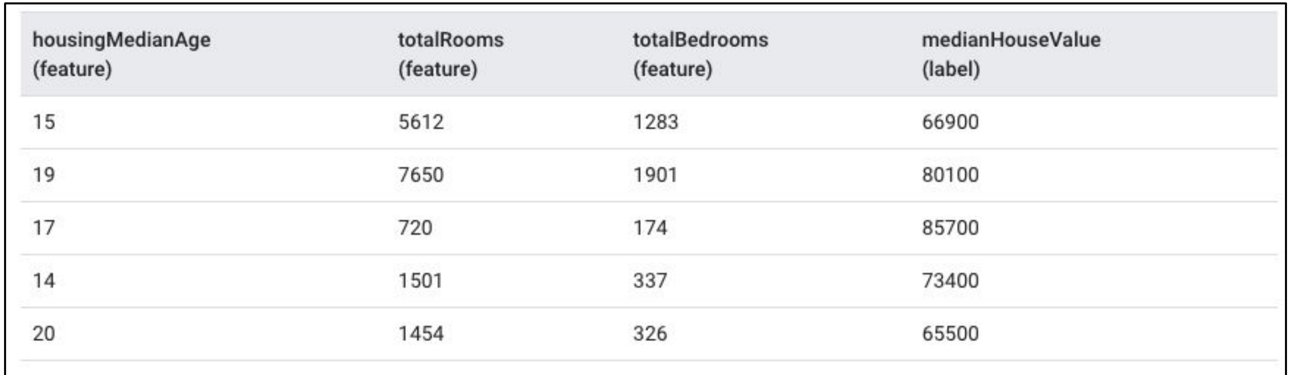
**Features**

- A feature is like a clue or a piece of information that helps us make a prediction.
- Features can be simple, like just one piece of data, or complex, involving millions of pieces of data.
- In a spam email detector, features might include the words used in the email, who sent it, and when it was sent.

**Examples**

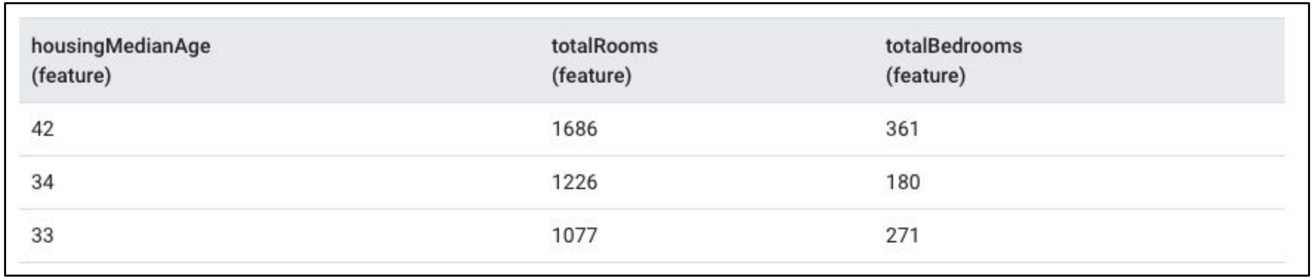
- We can have labeled examples, which have both the features and the label, like an email marked as spam or not spam.
- Unlabeled examples have features but no label. They are like questions without answers

# Key ML Terminology



| housingMedianAge<br>(feature) | totalRooms<br>(feature) | totalBedrooms<br>(feature) | medianHouseValue<br>(label) |
|---|---|---|---|
| 15 | 5612 | 1283 | 66900 |
| 19 | 7650 | 1901 | 80100 |
| 17 | 720 | 174 | 85700 |
| 14 | 1501 | 337 | 73400 |
| 20 | 1454 | 326 | 65500 |

Shown above is 5 labeled examples from a data set containing information about housing prices in California



| housingMedianAge<br>(feature) | totalRooms<br>(feature) | totalBedrooms<br>(feature) |
|---|---|---|
| 42 | 1686 | 361 |
| 34 | 1226 | 180 |
| 33 | 1077 | 271 |

Shown above is an  unlabeled example that contains features but no labels

# Key ML Terminology

## Models

- A model is a system that learns from examples to make predictions.
- Training a model involves showing it labeled examples to learn from.
- Once trained, a model can make predictions on new, unlabeled examples.

## Regression vs. Classification

- Regression models predict a continuous amount, like guessing the price of a house.
- Classification models choose between distinct categories, like deciding if an email is spam or not, or identifying the type of animal in a picture.
- This simplified explanation is designed to introduce these concepts in a way that's easy to understand, avoiding too much technical detail.

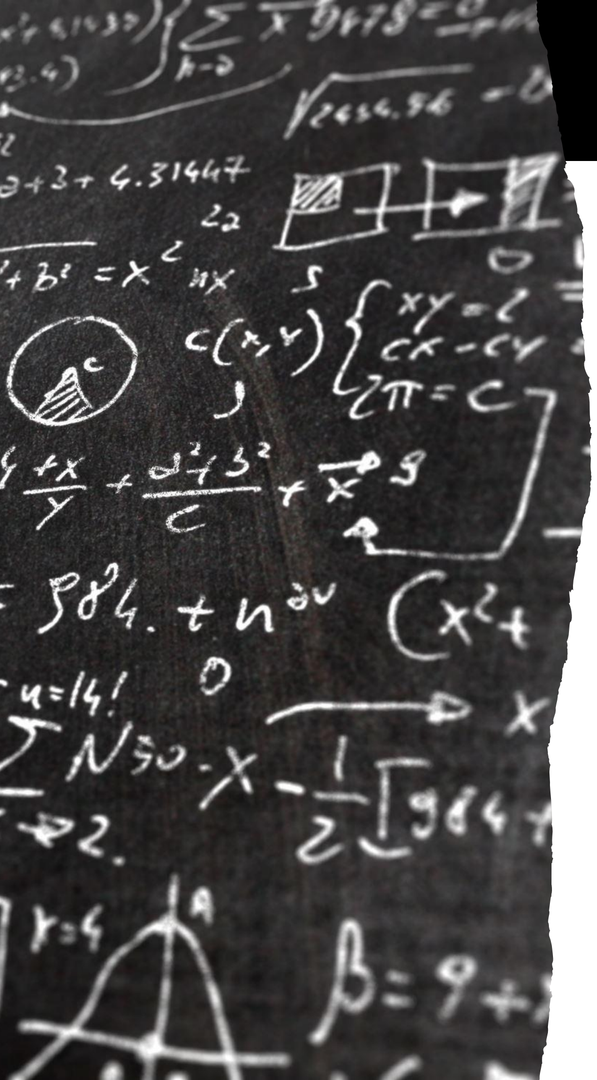
# Key ML Terminology

## Structured Data

- **Definition**: Structured data is highly organized and easily searchable because it is usually stored in well-defined formats like databases or spreadsheets.
- **Characteristics**:
    - **Format**: It follows a specific schema, meaning it is organized into fields and records. For example, a database table with columns for name, age, and address.
    - **Ease of Access**: Its standardized format makes it easy to search and query. For example, SQL (Structured Query Language) is often used to manage structured data.
    - **Examples**: Data in relational databases, Excel files, and CRM systems.
- **Applications**: Structured data is commonly used in traditional data analytics, where clear, defined data points are essential for computations and analysis

## Unstructured Data

- **Definition**: Unstructured data lacks a predefined format or structure, making it more complex and less straightforward to analyze and utilize.
- **Characteristics**:
    - **Format**: It comes in various formats and may include text, images, videos, and social media posts. There's no clear way to divide it into parts automatically.
    - **Complexity**: Requires more sophisticated methods for processing and analysis, like natural language processing (NLP) for text, or computer vision for images.
    - **Examples**: Emails, videos, customer reviews, social media posts, and audio recordings.
- **Applications**: Unstructured data is growing rapidly with the rise of social media and multimedia content. It's key in big data applications and advanced analytics like sentiment analysis and image recognition



# Types of Machine Learning

**Supervised Learning**

- In supervised learning, the algorithm is trained on a labeled dataset.
- This means that each example in the training dataset is paired with the correct output.
- The goal is to learn a mapping from inputs to outputs, which can be used to make predictions on new, unseen data.
- Common applications: regression, classification.

**Unsupervised Learning**

- In unsupervised learning, the algorithm is trained on a dataset without any labels.
- The goal is to discover inherent patterns, groupings, or structures in the data.
- It's more about exploring data and finding some structure within.
- Common applications: clustering, association rule learning

# Types of Machine Learning

| Criteria | Supervised Learning | Unsupervised Learning |
|---|---|---|
| **Data Type** | Labeled data (input-output pairs) | Unlabeled data (only input) |
| **Goal** | Predict the output for a new input. | Discover patterns and relationships in data. |
| **Example Task** | Classification or regression | Clustering or dimensionality reduction |
| **Training Process** | Learn to map input to output based on example pairs. | Explore the structure of data to find patterns or groupings. |
| **Evaluation** | Accuracy, precision, recall (comparison with true labels) | Cohesion, separation metrics (inherent properties of data) |
| **Outcome** | A model that predicts outputs for new inputs. | Insights into data, such as grouping similar items. |
| **Example** | Predicting house prices based on features like size, location (regression). | Grouping customers into segments based on purchasing behavior (clustering). |

# Types of Machine Learning

Let's take the example of a dataset containing information about various fruits.

- **Supervised Learning (Classification)**:
  - Data: Labeled data with fruit types (e.g., apple, banana) and their attributes (color, size, shape).
  - Goal: Train a model to classify fruits into their respective types.
  - Outcome: A model that can predict the type of a new fruit based on its attributes.

- **Unsupervised Learning (Clustering)**:
  - Data: Unlabeled data with only attributes of fruits (color, size, shape) and no fruit types.
  - Goal: Group fruits into clusters based on similarities in their attributes.
  - Outcome: Clusters of fruits, each cluster grouping similar types of fruits.

## Machine Learning in Practise

DATA COLLECTION

MODEL TRAINING

EVALUATION

DEPLOYMENT

# Machine Learning in Practise

### Data Collection

- **What It Is**: Gathering the information that your machine learning model will learn from.
- **Key Points**:
  - Can involve collecting new data or using existing datasets.
  - Data should be relevant to the problem you're trying to solve.
  - Quality and quantity of data are important – more high-quality data can lead to better model performance.
- **Example**: For a weather prediction model, this could mean gathering historical weather data like temperature, humidity, and rainfall

### Model Training

- **What It Is**: Teaching the machine learning model to make predictions or decisions based on the data.
- **Key Points**:
  - Involves feeding the data into the model so it can learn patterns and relationships.
  - The model iteratively adjusts its parameters to improve its predictions.
  - Supervised learning models require labeled data (input with corresponding output) for training.
- **Example**: Using the weather data to train a model to predict future weather conditions

# Machine Learning in Practise

## Evaluation

- **What It Is**: Testing the model to see how well it performs.
- **Key Points**:
  - Involves using a separate set of data (not used in training) to assess the model’s accuracy and effectiveness.
  - Helps in identifying any issues like overfitting (where the model performs well on training data but poorly on new data).
  - The goal is to ensure the model is reliable and works with new, unseen data.
- **Example**: Testing the weather prediction model with recent weather data to check its accuracy.

## Deployment

- **What It Is**: Putting the model into a real-world environment where it can provide practical value.
- **Key Points**:
  - Deployment means integrating the model into existing systems to make predictions on new data.
  - Requires careful planning to ensure the model remains reliable and efficient in a production environment.
  - Monitoring and maintenance are important after deployment to ensure the model continues to perform well.
- **Example**: Integrating the weather model into a weather forecasting app to provide daily predictions to users.

# ML – Fundamentals

## Metrics for regression models

### MAE (Mean Absolute Error)

Definition: MAE measures the average magnitude of errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between the prediction and the actual observation where all individual differences have equal weight.

Practicality: MAE is a linear score which means that all individual differences are weighted equally in the average. It's particularly useful when you want to understand the magnitude of error without squaring.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}.$$

# ML – Fundamentals

## Metrics for regression models

### MSE (Mean Squared Error)

Definition: MSE measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value.

Practicality: MSE is more sensitive to outliers than MAE because it squares the prediction errors. This means that larger errors have a disproportionately large effect on MSE. It is useful when you are particularly concerned about large errors.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2.$$

# ML – Fundamentals

## Metrics for regression models

**RMSE (Root Mean Squared Error)**

Definition: RMSE is the square root of the mean square error. It measures the standard deviation of the residuals (prediction errors).

Practicality: RMSE is even more sensitive to outliers than MSE and gives a relatively high weight to large errors. This means the RMSE should be more useful when large errors are particularly undesirable.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}},$$

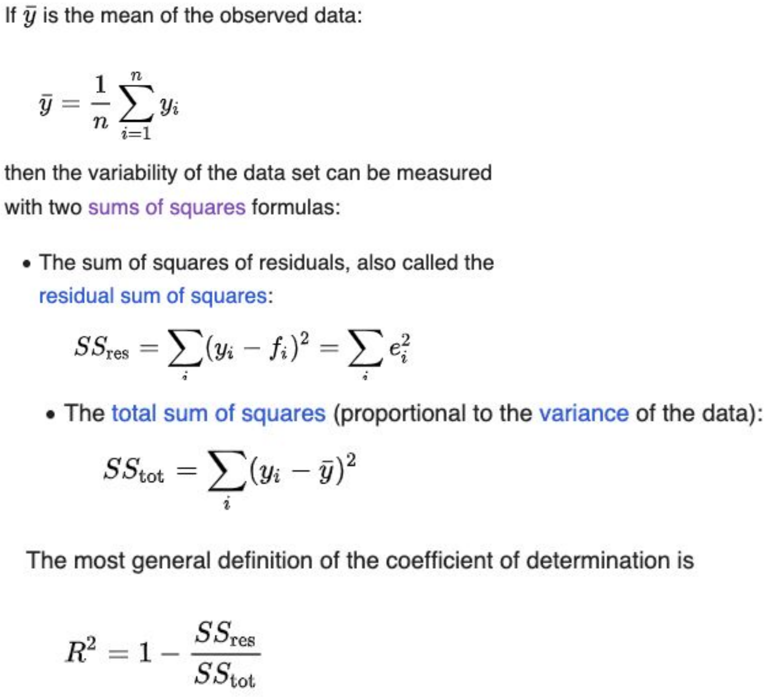
# ML – Fundamentals

## Metrics for regression models

**R-squared (Coefficient of Determination)**

Definition: R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

Practicality: R-squared represents the proportion of the variance for the dependent variable that's explained by the independent variables in the model. An R-squared of 1 indicates that the regression predictions perfectly fit the data. When comparing models, a higher R-squared is generally better.



If $\bar{y}$ is the mean of the observed data:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

then the variability of the data set can be measured with two sums of squares formulas:

- The sum of squares of residuals, also called the residual sum of squares:

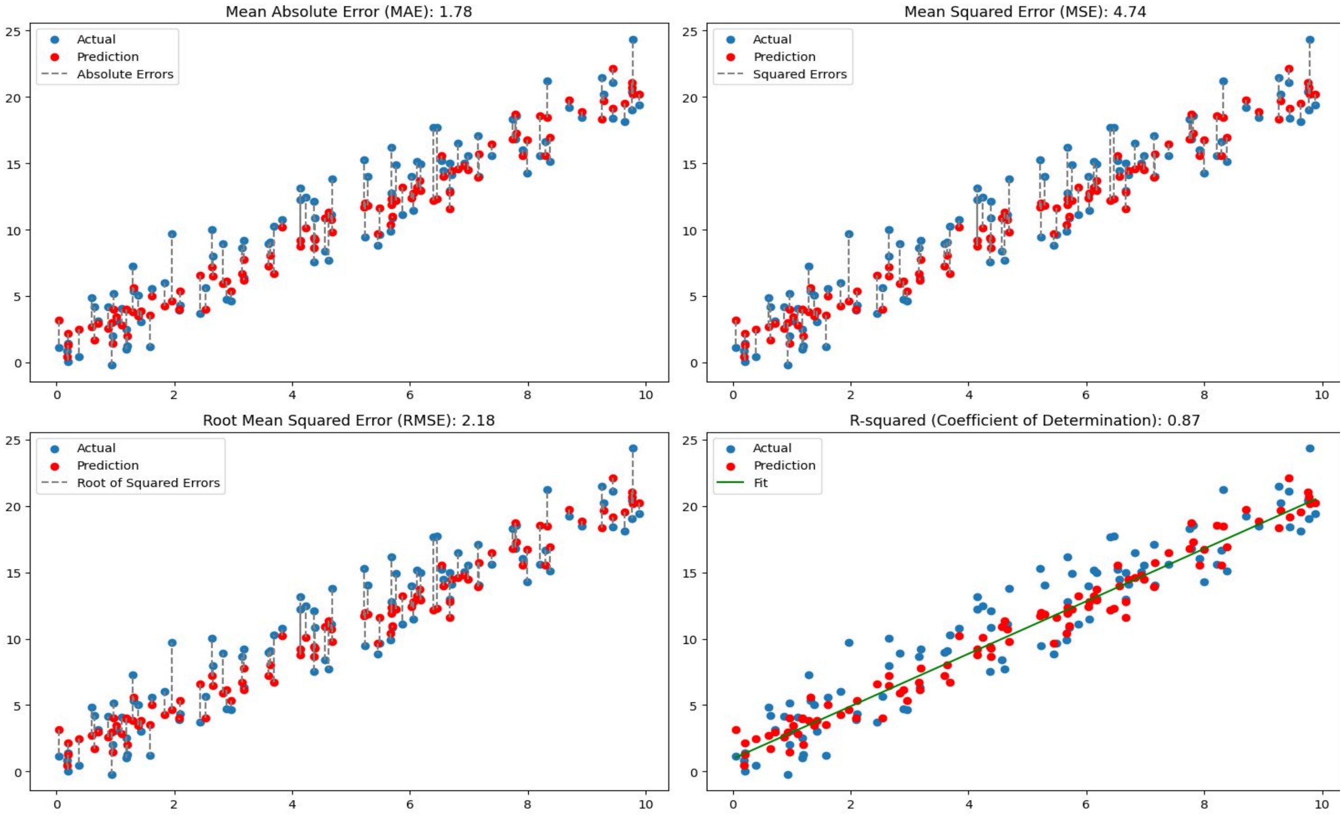
$$SS_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$$

- The total sum of squares (proportional to the variance of the data):

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

The most general definition of the coefficient of determination is

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$


Mean Absolute Error (MAE): 1.78
Actual
Prediction
Absolute Errors
Mean Squared Error (MSE): 4.74
Actual
Prediction
Squared Errors
Root Mean Squared Error (RMSE): 2.18
Actual
Prediction
Root of Squared Errors
R-squared (Coefficient of Determination): 0.87
Actual
Prediction
Fit
0
5
10
15
20
25
0
2
4
6
8
10

# ML – Fundamentals

## Metrics for regression models

These metrics provide different ways to measure the error between the predicted values by a regression model and the actual values.

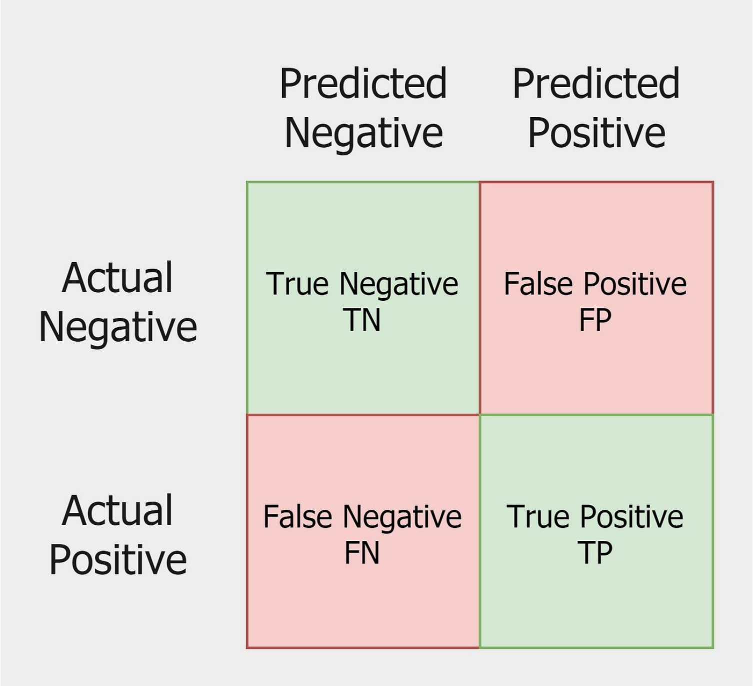
MAE gives a straightforward average error magnitude

MSE and RMSE emphasize larger errors.

And R-squared provides a measure of how well the model's predictions approximate the actual data.

When evaluating the performance of a regression model, it's common to look at several of these metrics together to get a comprehensive view of its accuracy and usefulness

## ML – Fundamentals for classification problems



| | Predicted Negative | Predicted Positive |
|---|---|---|
| Actual Negative | True Negative<br>TN | False Positive<br>FP |
| Actual Positive | False Negative<br>FN | True Positive<br>TP |

**1. (True Positive - TP):**

- This is where the actual condition was positive, and the model correctly predicted positive.
- Symbolized by a checkmark to indicate a correct prediction.
- Label this quadrant as "TP".

**2. (False Negative - FN):**

- Here, the actual condition was positive, but the model incorrectly predicted negative.
- Represented by an 'X' to indicate an incorrect prediction.
- Label this quadrant as "FN".

**3. (False Positive - FP):**

- The actual condition was negative, but the model incorrectly predicted positive.
- Also marked with an 'X' for incorrect prediction.
- Label this quadrant as "FP".

**4. (True Negative - TN):**

- The actual condition was negative, and the model correctly predicted negative.

# ML – Fundamentals for classification problems

## Metrics for classification models

**Precision**

Definition: Precision is the ratio of true positive predictions to the total positive predictions made. It is also known as the Positive Predictive Value (PPV).

Formula: Precision = TP / (TP + FP)

Practicality: Precision is a measure of a classifier's exactness. A high precision means that when the model predicts a positive result, it is likely to be correct. In practical terms, it's important when the cost of a false positive is high. For example, in spam detection, a high precision means that non-spam emails are less likely to be classified as spam.

**Recall**

Definition: Recall, also known as Sensitivity or the True Positive Rate (TPR), is the ratio of true positive predictions to the actual positive instances.

Formula: Recall = TP / (TP + FN)

Practicality: Recall is a measure of a classifier's completeness. A high recall means that the model is good at identifying all actual positives. It's important in cases where missing a positive is costly, such as in disease screening where failing to identify a sick patient could be dangerous.

# Metrics for classification models

### F1 Score

Definition: The F1 score is the harmonic mean of precision and recall. It is a single metric that combines both precision and recall into one number.

Formula: F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

Practicality: The F1 score is useful when you want to balance precision and recall. It's particularly important when the distribution of class instances is uneven (imbalanced classes). For instance, in fraud detection, where fraudulent transactions might be rare, the F1 score can provide insight into how well the model works on the minority class.

### Support

Definition: Support is the number of actual occurrences of the class in the specified dataset. For each class, it shows how many instances are available in the true labels.

Formula: Support for a class = The number of actual occurrences of the class in the dataset

Practicality: Support gives insight into the reliability of the metrics. Metrics calculated on a small support can be less reliable. It's also useful for assessing class imbalances in the dataset. In practice, knowing the support is essential for evaluating the significance of the precision and recall scores.