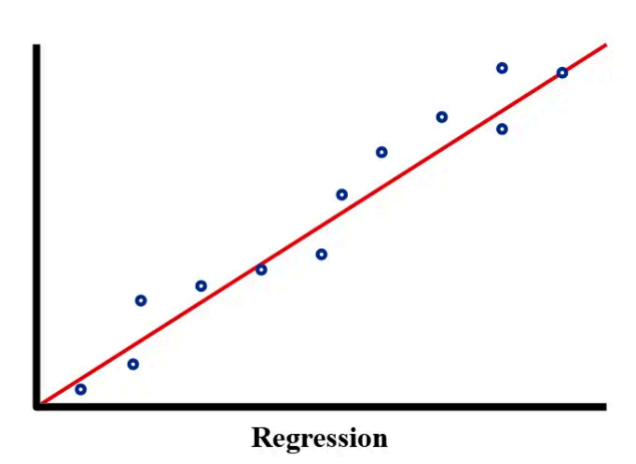
Supervised Machine Learning

Supervised Learning

Regression

In regression, the label is a **continuous numerical value.**

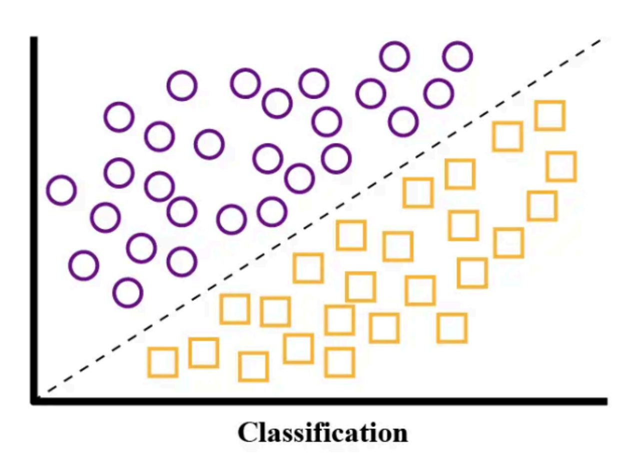
We approximate or predict a target value (like housing prices, stock prices, etc.).



Classification

In classification, the label is chosen from a **finite** set of **classes.**

We assign a category or class to an input (like spam detection, image recognition, etc.).



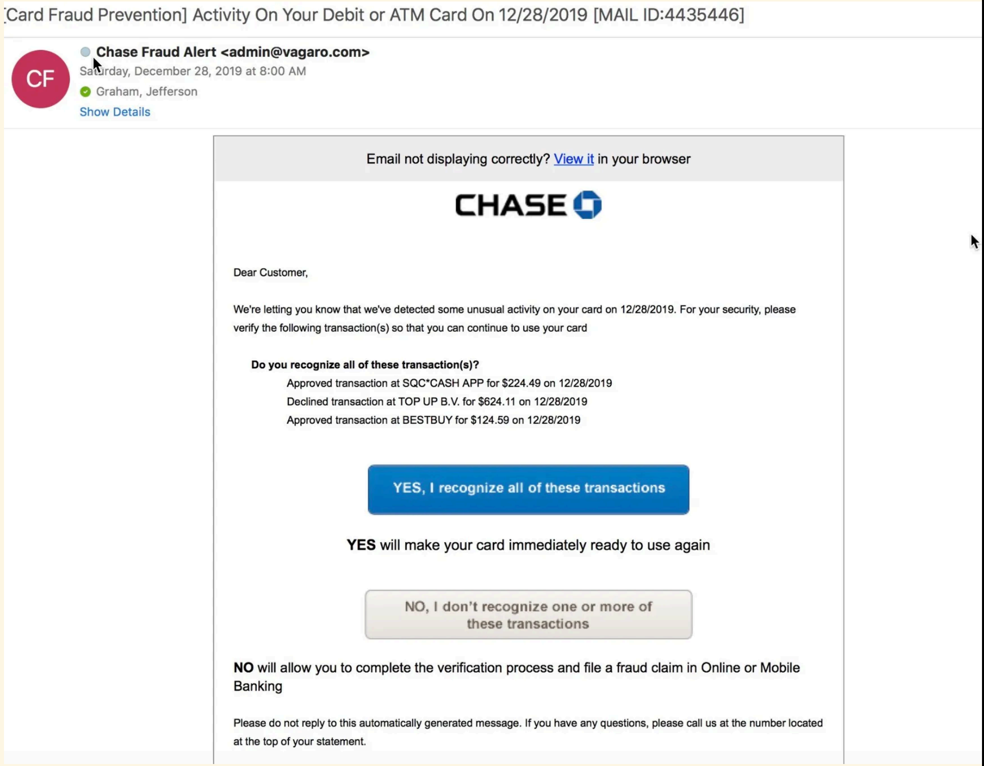
Supervised Learning Examples

Credit Card Fraud Detection

- **Features:** Vendor, location, time, distance from last transaction
- **Labels:** Chargebacks on previous transactions

Question: What kind of supervised learning?

1. Regression
2. Binary Classification
3. Multiclass Classification



[Card Fraud Prevention] Activity On Your Debit or ATM Card On 12/28/2019 [MAIL ID:4435446]

CF

Chase Fraud Alert <admin@vagaro.com>
Saturday, December 28, 2019 at 8:00 AM
Graham, Jefferson
Show Details

Email not displaying correctly? View it in your browser

CHASE

Dear Customer,

We're letting you know that we've detected some unusual activity on your card on 12/28/2019. For your security, please verify the following transaction(s) so that you can continue to use your card

Do you recognize all of these transaction(s)?
Approved transaction at SQC*CASH APP for $224.49 on 12/28/2019
Declined transaction at TOP UP B.V. for $624.11 on 12/28/2019
Approved transaction at BESTBUY for $124.59 on 12/28/2019

YES, I recognize all of these transactions

YES will make your card immediately ready to use again

NO, I don't recognize one or more of these transactions

NO will allow you to complete the verification process and file a fraud claim in Online or Mobile Banking

Please do not reply to this automatically generated message. If you have any questions, please call us at the number located at the top of your statement.

Supervised Learning Examples

Classification

- **Email Spam Filters**
  - Features: Words, sender, links
  - Label: Spam or Not Spam
- **Face ID / Fingerprint Unlock**
  - Features: Facial/fingerprint data
  - Label: You or Not You
- **Letter Recognition**
  - Features: Pixel values of images
  - Label: A-Z, 0-9, etc.

Regression

- **Weather Forecasting**
  - Features: Pressure, humidity, wind
  - Label: Temperature/rainfall amount
- **Uber/Lyft Pricing**
  - Features: Distance, time, demand
  - Label: Trip cost
- **YouTube View Count Predictions**
  - Features: Title, thumbnail, creator stats
  - Label: Expected views

Supervised Regression

What is Linear Regression?

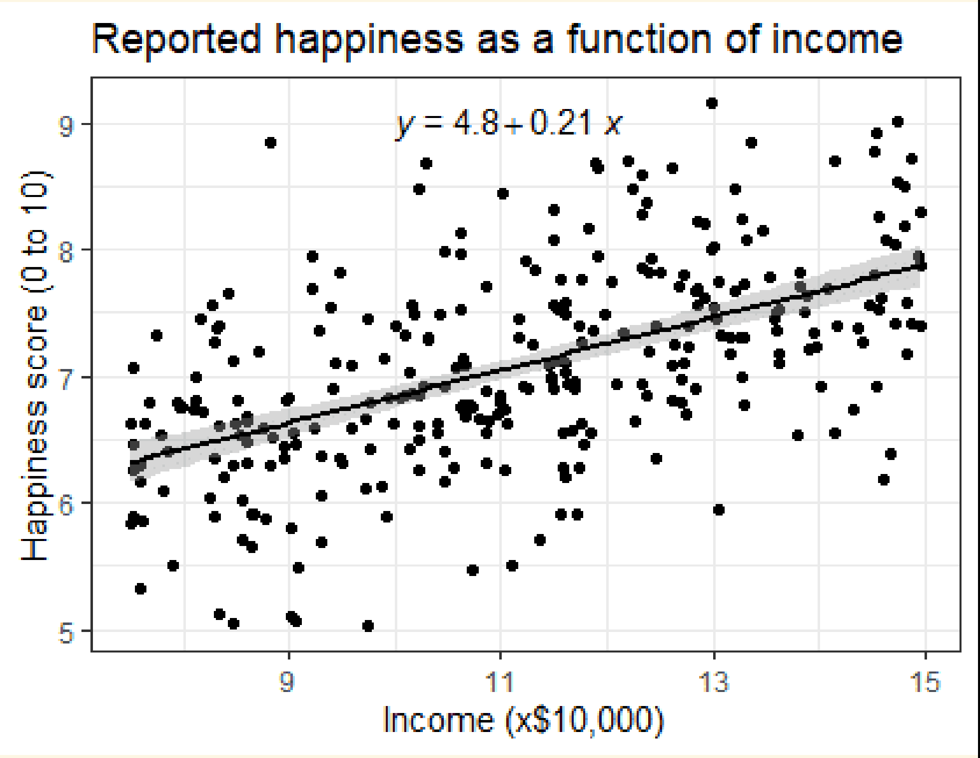
Linear regression is a **fundamental algorithm** in machine learning and can be thought of as simple supervised learning.

It models the relationship between a dependent variable y and one or more independent variables X by fitting a linear equation to the observed data.

For a simple model, we write it:

$$y = w_0 + w_1 x$$

- y: Dependent variable
- x: Independent variable
- w: A vector Coefficients



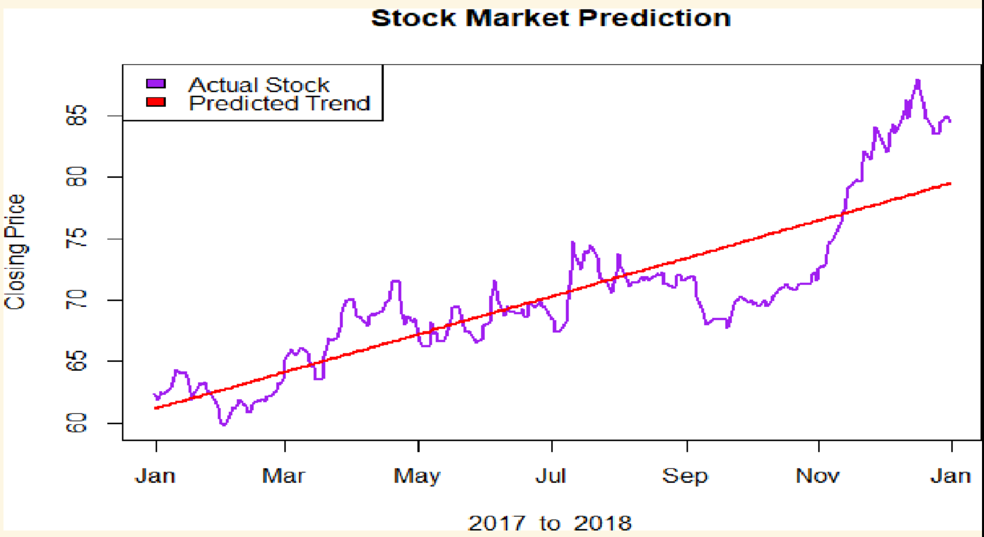
Linear Regression Example

Stock Market Prediction

- **Features:** Stock price from Feb 1st to March 1st
- **Labels:** Stock price on March 7th.

In the example above, the linear regression model would learn the relationship between the stock prices over time and use that to predict future prices.

- y: Closing Price
- x: Date
- w: The trained coefficients



Polynomial Regression Equation

The polynomial regression equation is:

We represent polynomial features as:

Real-World Example

Outer Wall Thickness of an extruded vinyl profile is measured and recorded manually using a cut profile and a pair of calipers once every 12 hours.

- If this wall is **too thin**, the profile will create failure points.
- If this wall is **too thick**, the profile will be too heavy and expensive.
- Customer is **losing $400K/mo** in overage.

Multiplication Rule

The probability of something happening is denoted as:

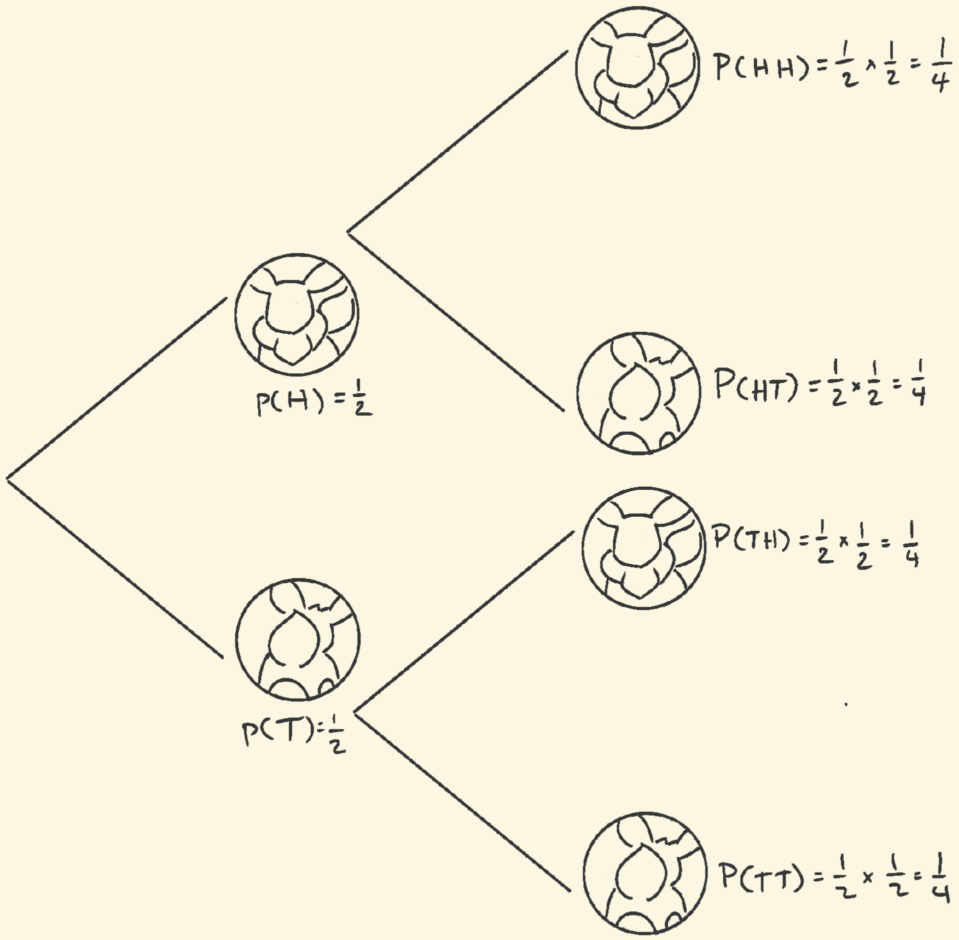
$$P(\text{event})$$

Two independent events have a joint probability:

$$P(A \cap B) = P(A) \times P(B)$$

For example, if the probability of flipping a coin on heads once is 1/2 than the probability of flipping it twice and getting heads both times is:

$$P(\text{heads}) \times P(\text{heads}) = 1/2 \times 1/2 = 1/4$$



Conditional Probability

The probability of the positive class y given some non-independent event x is denoted as:

Bayes' theorem relates the probability of the positive class to the likelihood and prior probability:

Example Calculation

Suppose:

- $P(\text{COVID}) = 0.05$ (5% of the population is infected)
- $P(\text{symptoms}|\text{COVID}) = 0.90$ (90% of infected people show symptoms)
- $P(\text{symptoms}) = 0.20$ (20% of the population shows symptoms)

Then:

$$P(\text{COVID}|\text{symptoms}) = \frac{0.90 \times 0.05}{0.20} = \frac{0.045}{0.20} = 0.225$$

Interpretation:

*Given that you have symptoms, there is a **22.5% probability that you have COVID-19**.*

Logistic Regression

Logistic Regression

Logistic regression is used for **binary classification** problems (e.g., is this email spam or not?).

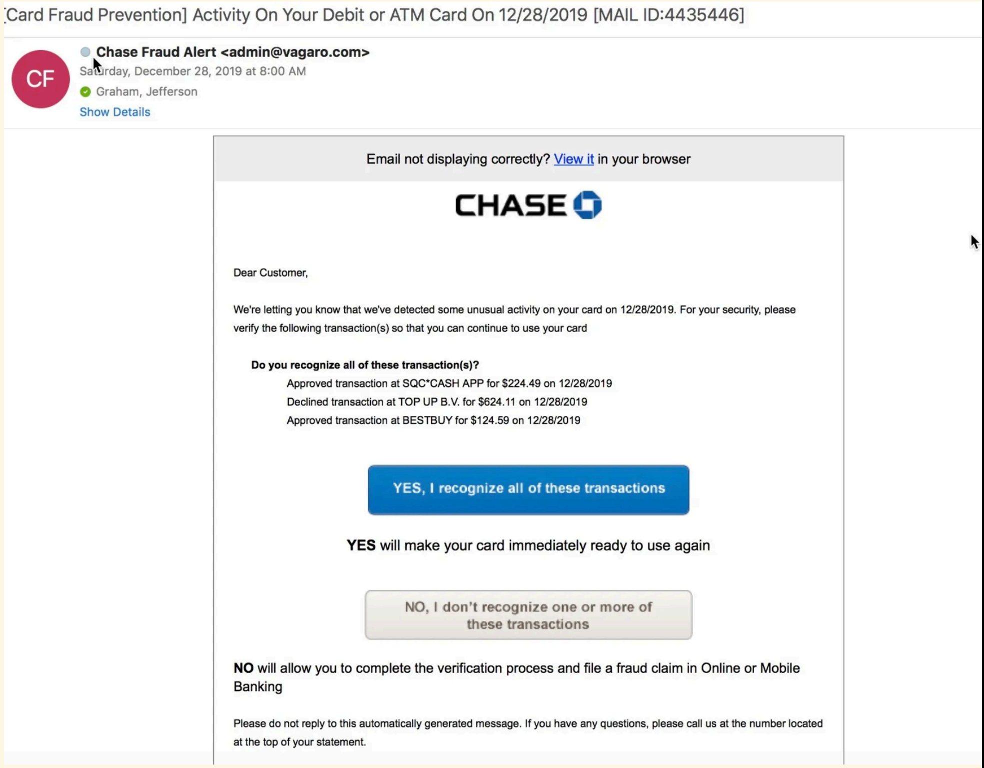
It models the probability that a given input x belongs to a particular category (often “positive” vs. “negative”).

Credit Card Fraud Detection

Features: Vendor, location, time, distance from last transaction

Labels: Chargebacks on previous transactions

Modeled Relationship: The probability of a transaction being fraudulent.



Odds and Log-Odds

Odds

The ratio of the probability of an event to the probability of not the event.

$$\text{Odds} = \frac{P(y=1|x)}{1-P(y=1|x)}$$

Log-Odds (Logit)

The natural logarithm of the odds.

$$\text{Log-Odds} = \log\left(\frac{P(y=1|x)}{1-P(y=1|x)}\right)$$

Logistic regression is linear in the **log-odds** space:

$$\log\left(\frac{P(y=1|x)}{1-P(y=1|x)}\right) = w_0 + w_1 x_1 + \cdots + w_p x_p$$

Interpretation of Coefficients

Each coefficient w_i represents how the **log-odds** of the positive class changes with respect to a one-unit change in x_i.

Coefficient	Interpretation
$w_i > 0$	Increasing x_i increases the log-odds (increasing probability)
$w_i < 0$	Increasing x_i decreases the log-odds (decreasing probability)

Measuring Performance

Accuracy

Layman friendly. Can be misleading for imbalanced datasets.

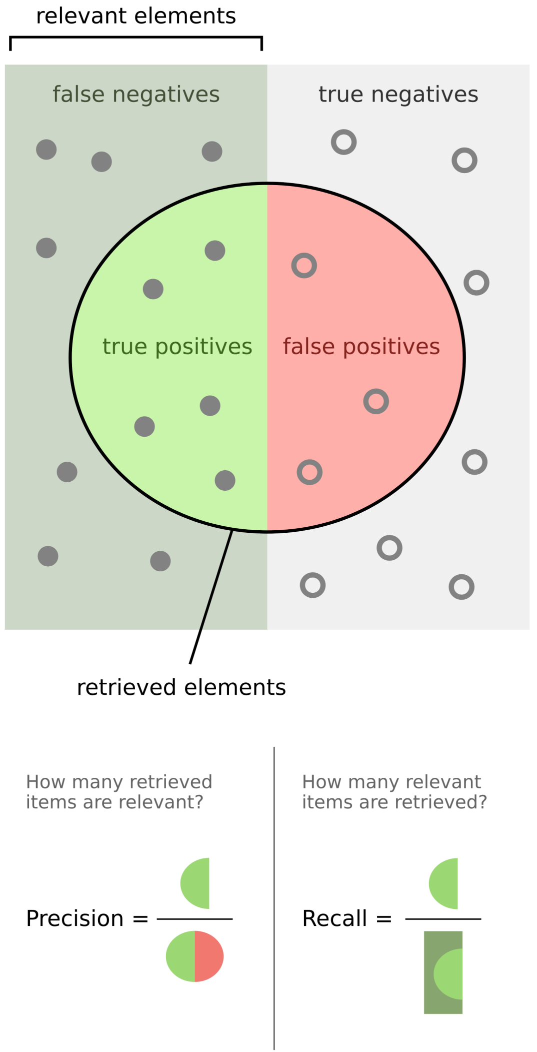
$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

Precision and Recall

Less intuitive but more informative for imbalanced datasets.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$



Training a Supervised Learning Model

(4) Model Evaluation

- **Predict**: Use the model to make predictions on the test features.
- **Evaluate**: Compare predicted labels to the test labels with an evaluation metric.

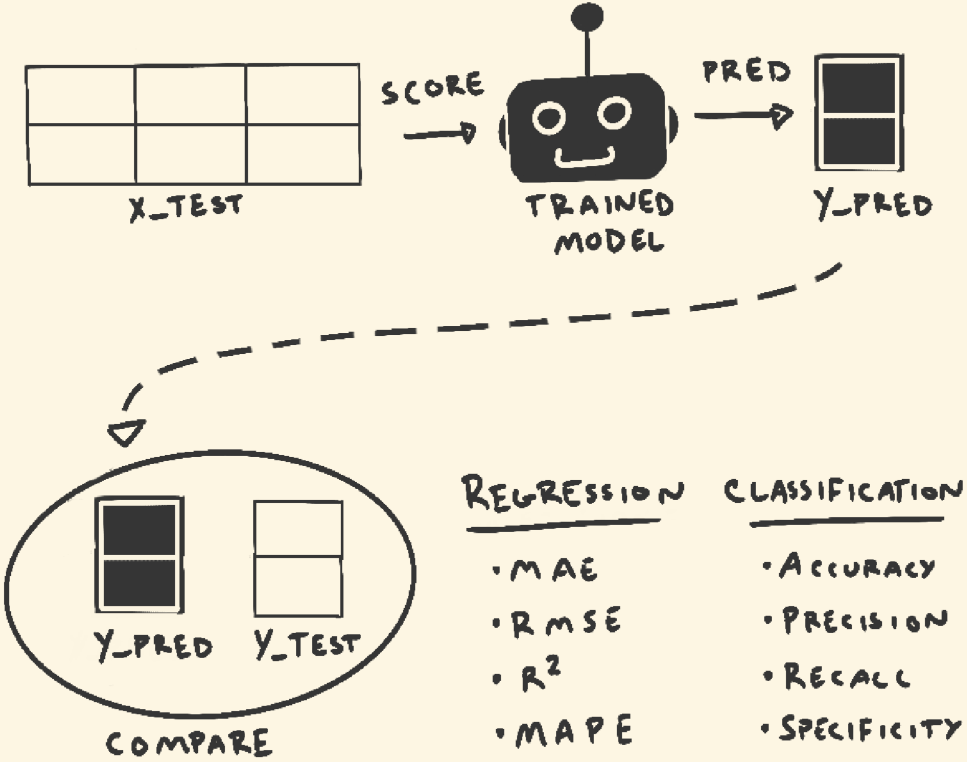
Regression Evaluation

```
from sklearn.metrics import mean_squared_error
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'Regression Mean Squared Error: {mse}')
```

Classification Evaluation

```
from sklearn.metrics import accuracy_score
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Classification Accuracy: {accuracy}')
```

Different evaluation metrics are used for regression and classification problems.



Support Vector Machines (SVM)

SVM in Practice

```python
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y,

# IMPORTANT: Scale features for SVM!
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```python
# Train SVM with RBF kernel
svm = SVC(kernel='rbf', C=1.0, gamma='scale')
svm.fit(X_train_scaled, y_train)

# Evaluate
y_pred = svm.predict(X_test_scaled)
tp, tn, fp, fn = calculate_confusion_matrix(y_test, y_pred
accuracy = (tp + tn) / (tp + tn + fp + fn)
print(f"Accuracy: {accuracy:.1%}")
```

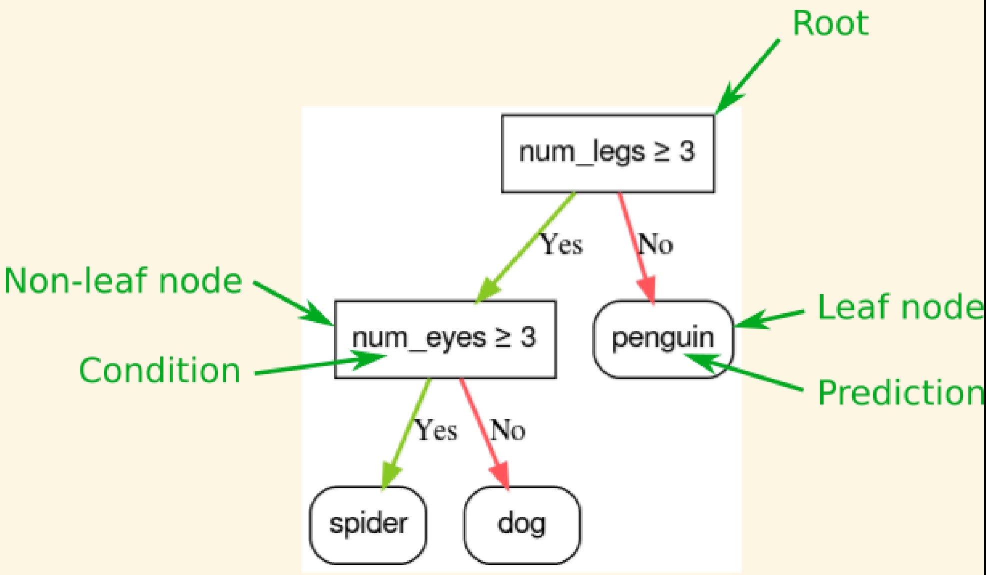
Note: SVMs are sensitive to feature scales! Always **normalize** your data.

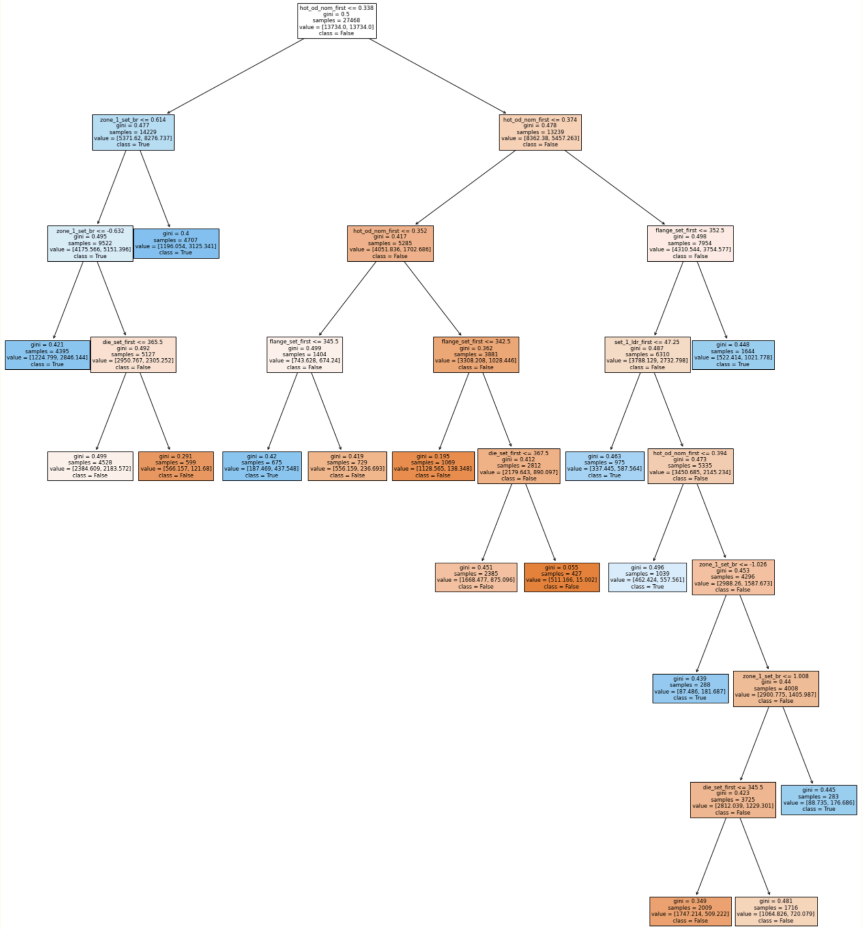
Decision Trees

What is a Decision Tree?

A decision tree is a flowchart-like structure used for classification and regression tasks.

It recursively splits the dataset into subsets based on feature values, forming a tree of decisions.






```sql
WHERE
  (
    (hot_od_nom_first < 0.339) AND (zone_1_set_br >
  ) -- class: True (proba: 72.32%) based on 4,707 sa
  OR (
    (hot_od_nom_first < 0.339) AND (zone_1_set_br <
  ) -- class: True (proba: 69.91%) based on 4,395 sa
  OR (
    (hot_od_nom_first > 0.338) AND (hot_od_nom_first
  ) -- class: True (proba: 66.17%) based on 1,644 sa
  OR (
    (hot_od_nom_first > 0.338) AND (hot_od_nom_first
    AND (flange_set_first < 352.501) AND (set_1_ldr_
    AND (hot_od_nom_first < 0.395)
  ) -- class: True (proba: 54.66%) based on 1,039 sa
  OR (
    (hot_od_nom_first > 0.338) AND (hot_od_nom_first
    AND (flange_set_first < 352.501) AND (set_1_ldr_
  ) -- class: True (proba: 63.52%) based on 975 samp
  OR (
    (hot_od_nom_first > 0.338) AND (hot_od_nom_first
    AND (hot_od_nom_first < 0.353) AND (flange_set_
  ) -- class: True (proba: 70.01%) based on 675 samp
  OR (
    (hot_od_nom_first > 0.338) AND (hot_od_nom_first
    AND (flange_set_first < 352.501) AND (set_1_ldr_
    AND (hot_od_nom_first > 0.394) AND (zone_1_set_b
  ) -- class: True (proba: 67.5%) based on 288 sampl
```

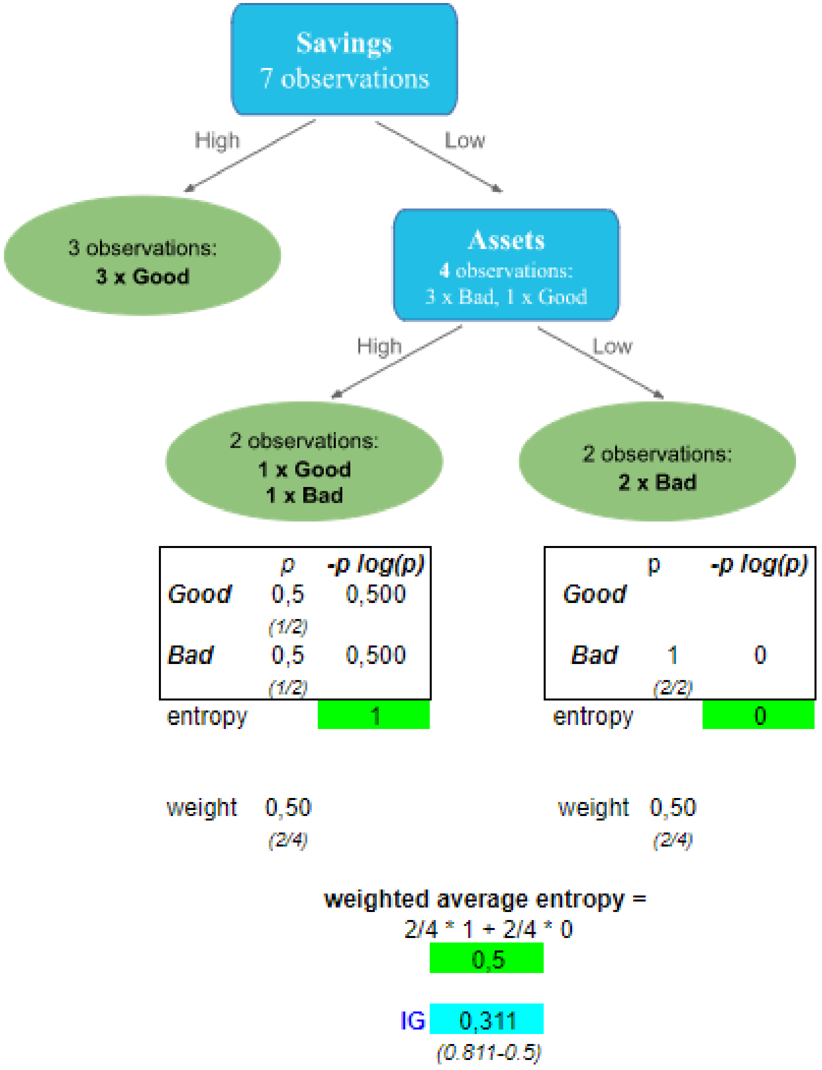
Splitting Criteria - Information Gain

- Measures the reduction in entropy after a split.
- Information Gain for a split S:

$$IG(S) = H(\text{parent}) - \sum_j \frac{|S_j|}{|S|} H(S_j)$$

where $H(\text{parent})$ is the entropy of the parent node, S_j are the subsets formed by the split, and $|S_j|$ is the number of instances in subset S_j.

> **Intuition:** How much less surprised am I after the split?



Exercise: Predicting Admissions v2

bigd103.link/decision-trees