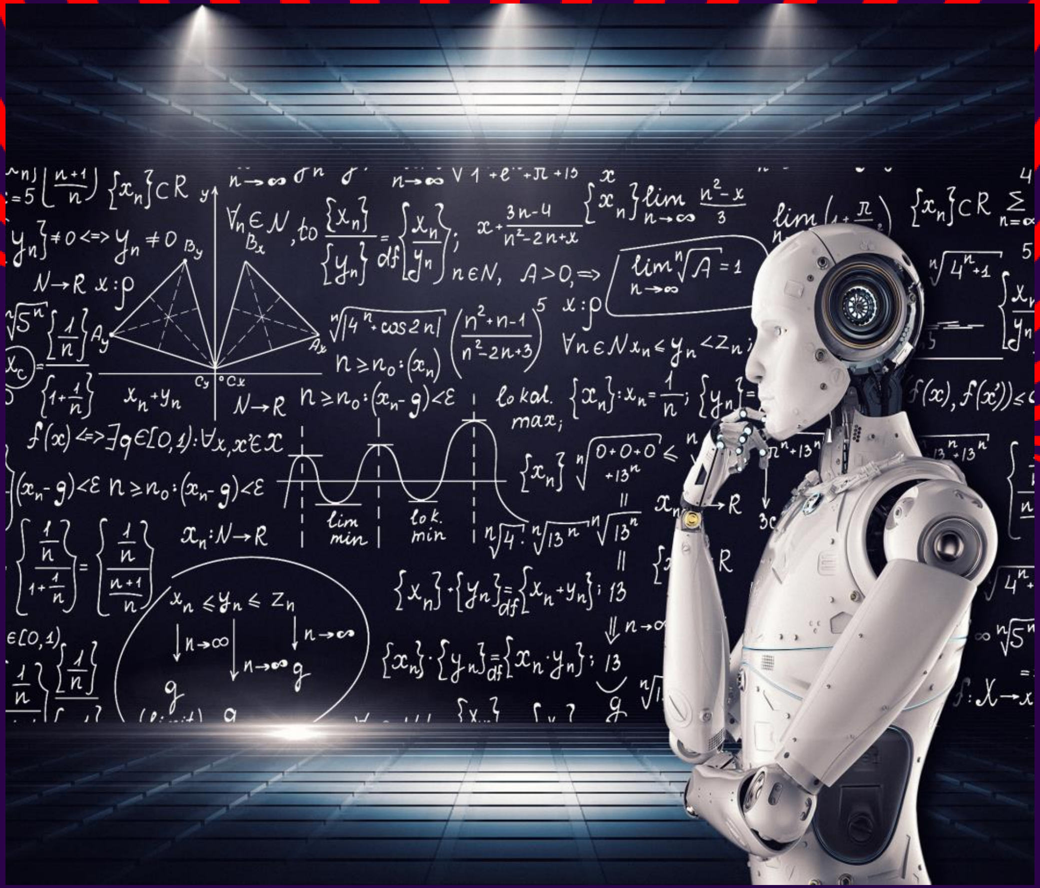
MACHINE LEARNING

Trainer :

Kaleem u Allah

DEFINITION:

Machine learning (ML) is a subfield of artificial intelligence (AI) that allows computers to learn without being explicitly programmed. Instead, they learn from data, identify patterns, and make predictions on new data. Imagine teaching a friend a game by showing examples – machine learning works similarly.



TYPES OF MACHINE LEARNING

Supervised

- Works under supervision
- Teacher teaches
- Label Data Required
- Prediction
- Outcome

Unsupervised

- No supervision
- No teacher
- Self Learning
- No labeling of data
- Find patterns by itself

Semi-supervised

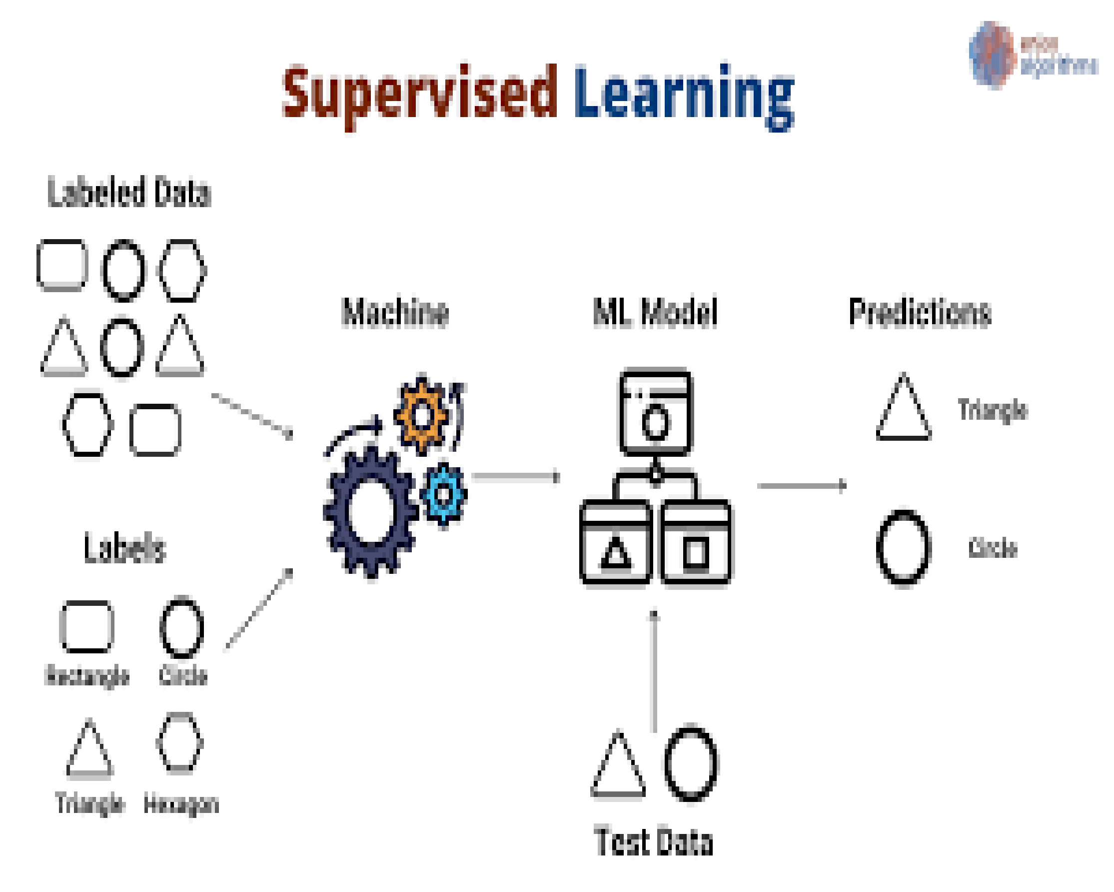
- Mixture of label and unlabeled data
- Can improve performance when labeled data is limited or expensive to obtain.

Reinforcement

- Hit and Trial Learning
- Learn from mistakes
- Prediction based on Reward and punishment
- Depends on feedback

SUPERVISED LEARNING

Supervised machine learning is a type of machine learning where algorithms are trained using labeled data. This labeled data consists of inputs and their corresponding desired outputs, like a teacher showing a student example and giving them the answers. The algorithm learns the relationship between the inputs and outputs, allowing it to make predictions for new, unseen data.



EXAMPLES

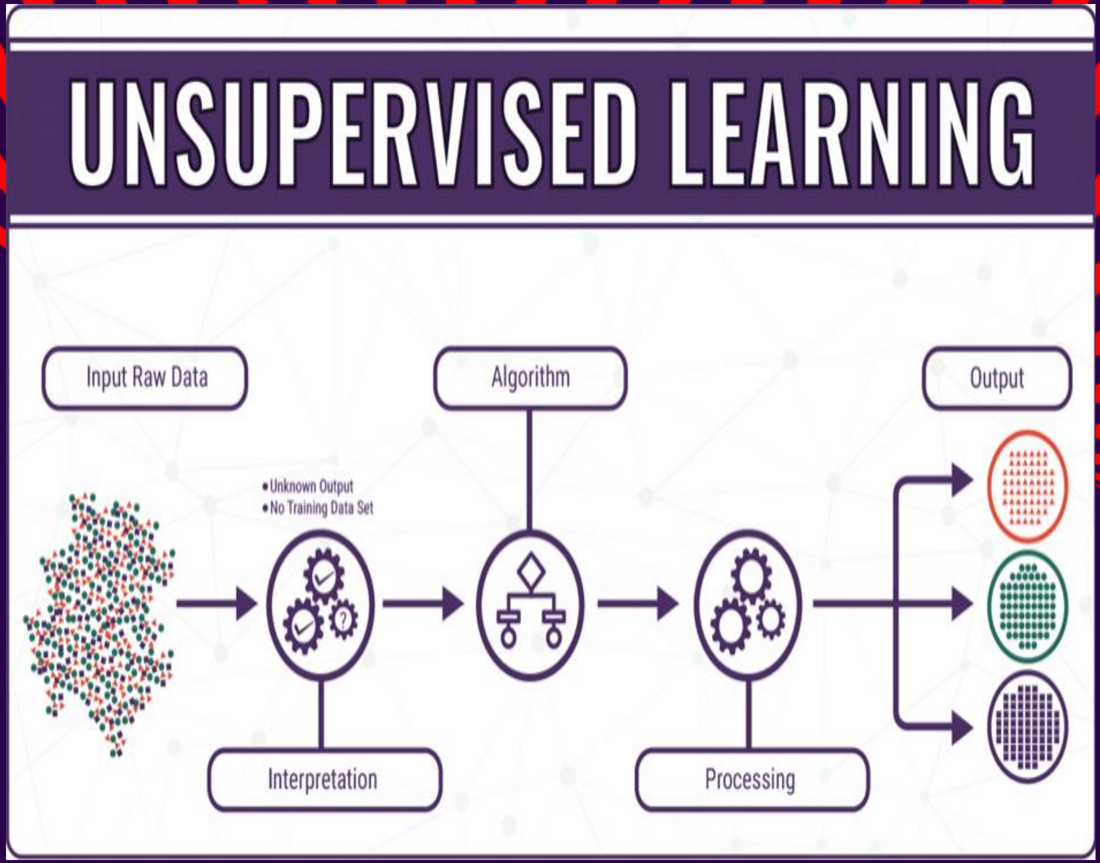
- Stock Market prediction
- Medical Diagnosis
- Loan Approval Prediction
- Self Driving cars
- Spam Email Detection
- Image classification
- House price prediction
- Sentimental Analysis
- Machine Translations

ALGORITHMS

- Linear Regression
- Logistic Regression
- Decision Trees
- K-Nearest Neighbors (KNN)
- Support Vector Machine (SVM)
- Random Forest
- Naïve Bayes
- Gradient Boosting Machines (GBM)

UNSUPERVISED LEARNING

Unsupervised machine learning is a type of machine learning where the algorithm learns patterns and structures from unlabeled data. Unlike supervised learning, there are no predefined labels associated with the input data. Instead, the algorithm tries to find inherent structures or relationships within the data on its own. Unsupervised learning is particularly useful for exploring and understanding the underlying structure of data, identifying hidden patterns, and segmenting data into meaningful groups.



EXAMPLES

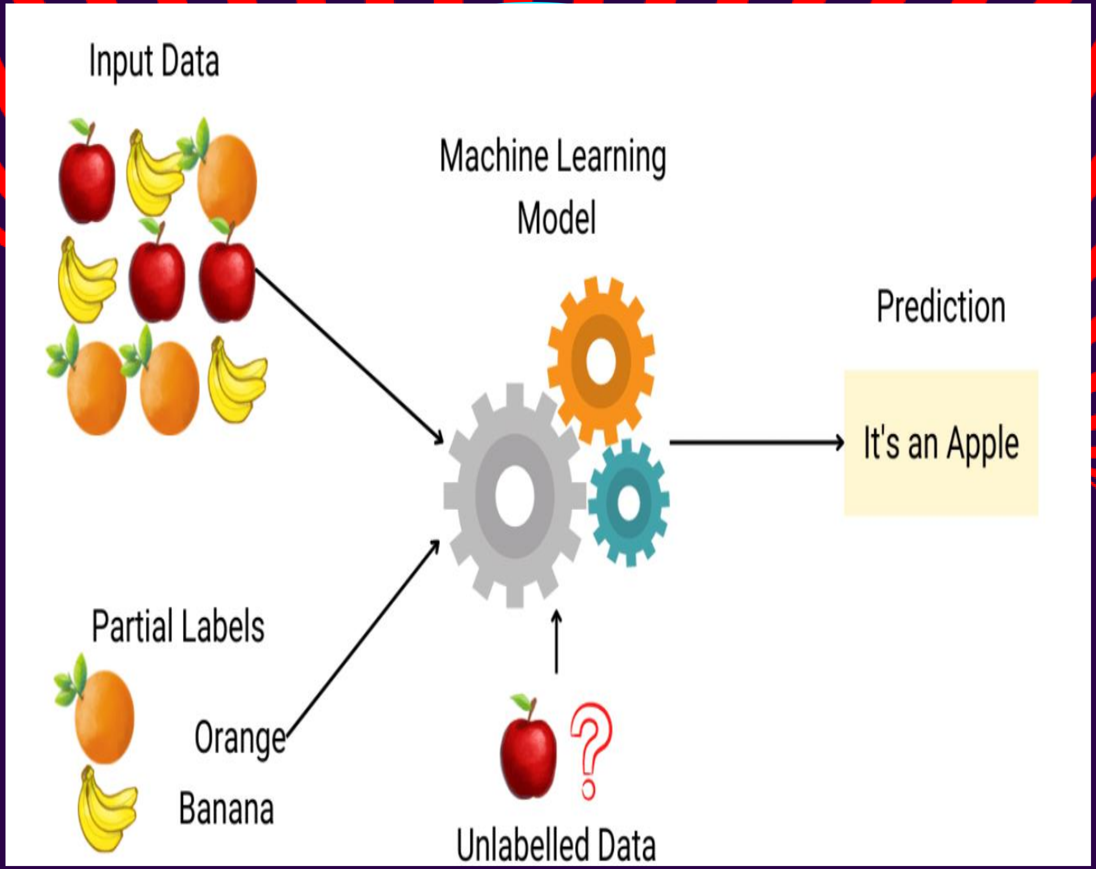
- Clustering
- Dimensionality Reduction
- Anomaly Detection
- Feature Learning
- Topic Modeling
- Customer segmentation
- Social Network Analysis
- Density Estimation

ALGORITHMS

- K-Mean Clustering
- Hierarchical Clustering
- Principal Component Analysis (PCA)
- Autoencoders
- Anomaly Detection Algorithm
- DBSCAN

SEMI SUPERVISED LEARNING

Semi-supervised machine learning is a learning paradigm that falls between supervised and unsupervised learning. In semi-supervised learning, the algorithm is trained on a dataset that contains both labeled and unlabeled data. The availability of labeled data is typically limited or expensive, while unlabeled data is more abundant. The primary goal of semi-supervised learning is to leverage both labeled and unlabeled data to improve the performance of the model.



EXAMPLES

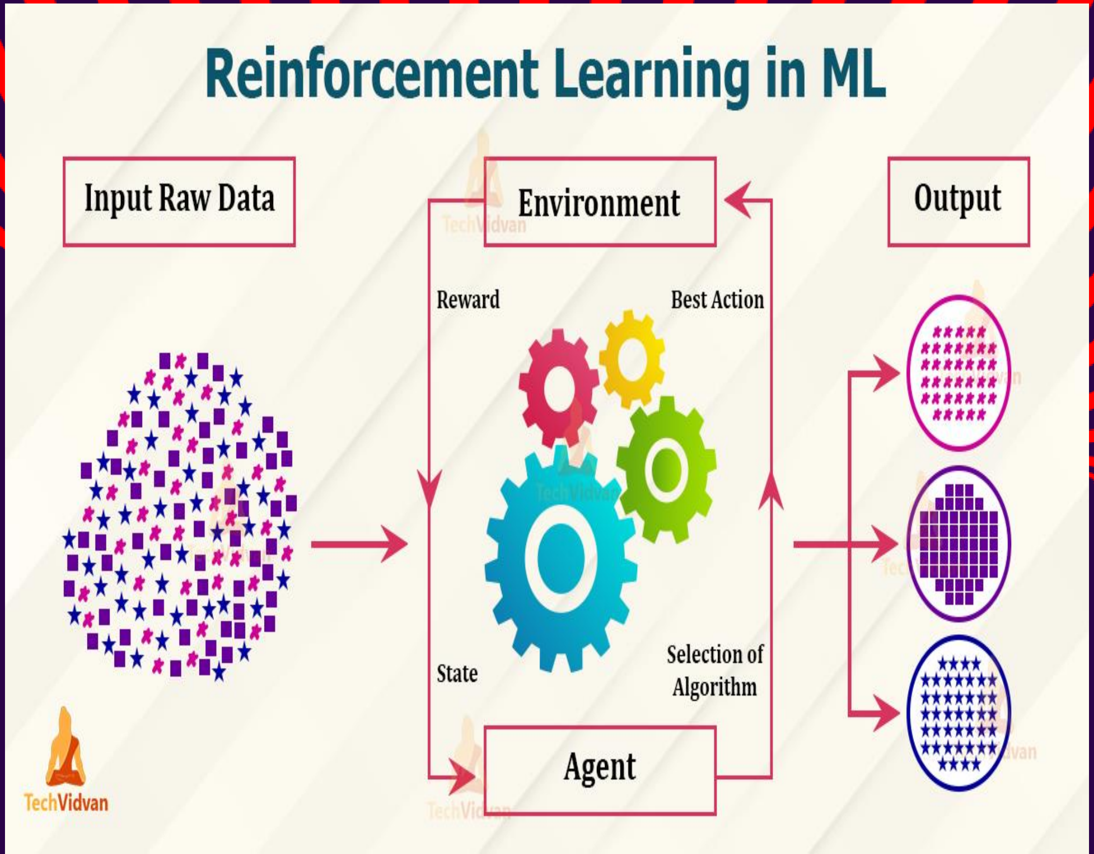
- Image Classification
- Sentimental Analysis
- Spam Filtering
- Document Classification
- Speech Recognition
- Video Analysis
- Drug Discovery
- Anomaly Detection

ALGORITHMS

- Self Training
- Label Propagation
- Co-Training
- Semi Supervised Support Vector Machines
- Generations Model

REINFORCEMENT LEARNING

Reinforcement Learning (RL) is a type of machine learning paradigm where an agent learns to make sequential decisions by interacting with an environment t achieve a certain goal or maximize cumulative rewards. In reinforcement learning, the agent learns through trial and error, receiving feedback in the form of rewards or penalties from the environment based on its actions.



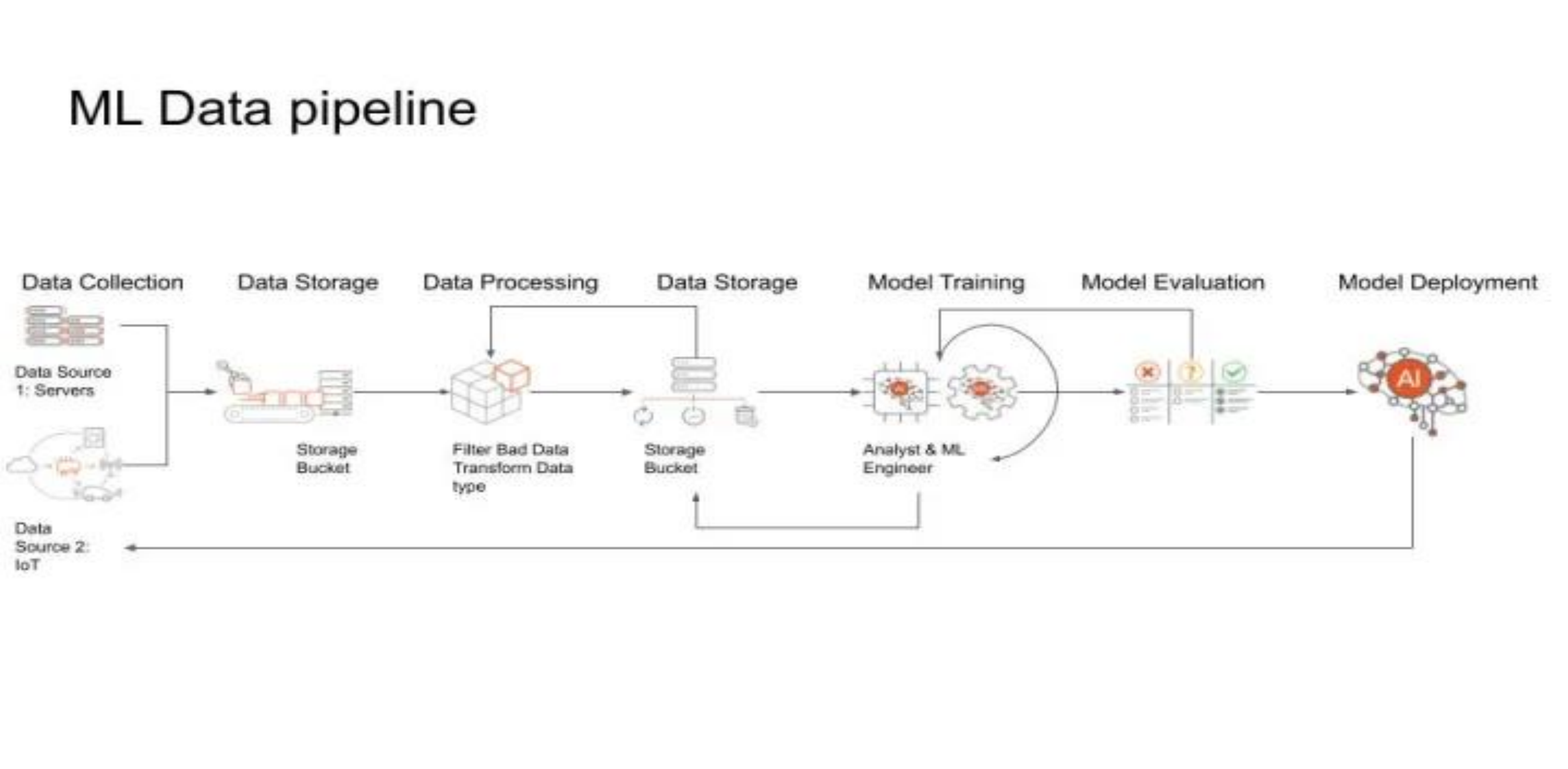
EXAMPLES

- Self-Driving Cars
- Playing Games
- Robot Control
- Recommendation System
- Inventory Management

ALGORITHMS

- Q-Learning
- Deep Q-Networks
- Policy Gradient Method
- Actor-Critic Method
- Deep Q-Networks (DQNs)

MACHINE LEARNING PIPELINE:



MACHINE LEARNING PIPELINE

Data Collection

This is the foundation of any machine learning project. It involves gathering the raw data you'll use to train your model. Data can come from various sources like databases, surveys, sensors, web scraping, or APIs.
Example: Imagine you want to build a model to predict house prices. You'd collect data on houses, including features like square footage, number of bedrooms, location, and historical sales data.

Data preprocessing

Raw data is rarely perfect. This step involves cleaning and preparing the data for analysis. You might address missing values, remove outliers, format the data consistently, or handle categorical variables.
Example: In your house price data, you might find houses with missing square footage entries. You'd need to decide how to handle these – fill them in with an estimate, remove those houses, or impute them using statistical methods.

Feature Crafting

This is where you transform your data into features the machine learning model can understand. Features are essentially the variables your model will use to make predictions. You might create new features based on existing ones to improve the model's performance.
Example: For the house price model, you could create a new feature combining square footage and several bedrooms to represent the total living space.

MACHINE LEARNING PIPELINE

Modeling

Here's where the magic happens! You choose and train a machine learning model on your prepared data. Different models are suited for different tasks (regression for predicting prices, classification for categorizing emails).
Example: You might choose a linear regression model for your house price prediction. The model will learn the relationship between the features (like square footage) and the target variable (house price).

Testing & Evaluation

You can't trust a model blindly. This step involves testing the model's performance on unseen data to assess its accuracy and generalizability. Metrics like mean squared error for regression models or F1-score for classification models help you gauge how well it performs.
Example: You'd split your house price data into training and testing sets. The model trains on the training data and is then evaluated on the testing data to see how well it predicts house prices for unseen examples.

Deployment

If your model performs well, it's time to deploy it! This means integrating it into a real-world application. The model can be used to make predictions on new data, like estimating the price of a new house on the market.
Example: Your house price model could be deployed on a website. Users could enter details of their house, and the model would predict a price based on the learned patterns from the training data.