

# AI

* is about building machines that can perform tasks that a human would typically perform.

# Machine learning

* is the scientific study of algorithms and statistical models to perform a task using inference(conclusion) instead of instructions.
* It focuses on using data to train ML models so the models can make predictions.

# Deep learning

* represents a significant leap forward in the capabilities for AI and ML.
* The theory behind deep learning was created from how the human brain works.
* It uses layers of neurons to build networks that solve problems.

# Generative AI

* is a type of AI that can create new content and ideas, including conversations, stories, images, videos, and music. Generative AI is powered by very large ML models, commonly called foundation models (FMs), that are pre-trained on vast amounts of data.

# Module 2

## Machine Learning:

### Machine Learning Examples:

* Spam
* Recommendations
* Credit card fraud

### Types Of machine Learning:

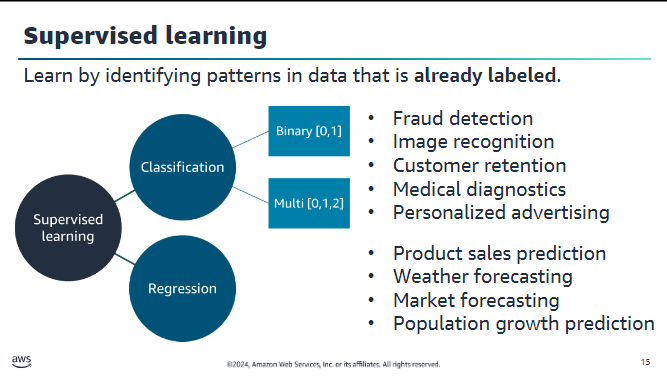
* Supervised
* Unsupervised
* reinforcement learning

#### 1)Supervised Learning:

* **Definition:** A type of machine learning where a model is trained on labeled data, learning by example with the help of a "teacher" who provides the correct answers.
* **Training Process:** The model uses training data to learn patterns and relationships between inputs and outputs, allowing it to make predictions on new, unseen data.

**Types of Supervised Learning Problems:**

1. **Classification:**
   * **Binary Classification:** Involves categorizing observations into one of two categories (e.g., fraudulent vs. not fraudulent transactions).
   * **Multiclass Classification:** Involves categorizing observations into one of three or more categories (e.g., predicting the reason for a customer call among various customer support departments).
2. **Regression:** Not detailed in the text but typically involves predicting continuous values.



#### 2) unsupervised Learning

* **Definition:** Machine learning where the model uncovers and creates labels from the data itself without pre-provided labels.
* **Purpose:** Detects emerging properties and constructs patterns from the dataset.

**Common Subcategory:**

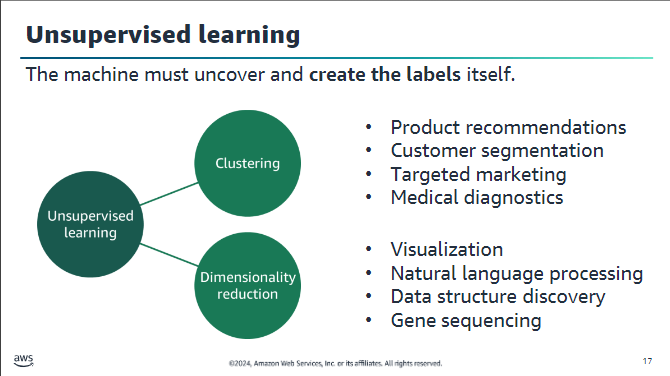
* **Clustering:** Groups data into clusters based on similar features to understand specific attributes.

**Example:**

* **Marketing Organization:** Analyzing customer purchasing habits to classify companies into small, medium, or large, helping tailor marketing strategies.

**Advantage:**

* **Pattern Discovery:** Reveals patterns in the data that were previously unknown, such as identifying distinct customer types.



##### Natural Language Processing (NLP)

**Definition:** A machine learning area focused on understanding and processing human language, both spoken and written.

**Applications:**

* **Voice Assistants:** Example: Amazon Alexa uses NLP to answer questions.
* **Chat/Call Centre Bots:** Automate tasks like checking bank balances or ordering food.
* **Translation Tools:** Convert text between languages or translate menus in real time.
* **Voice-to-Text Translations:** Convert spoken words into text for automatic subtitles.
* **Sentiment Analysis:** Analyse sentiments in reviews to provide audience ratings for products, music, and movies.

#### 3) Reinforcement Learning

**Definition:** A type of machine learning where an agent learns through trial and error by interacting with an environment, continuously improving its model based on feedback from previous iterations.

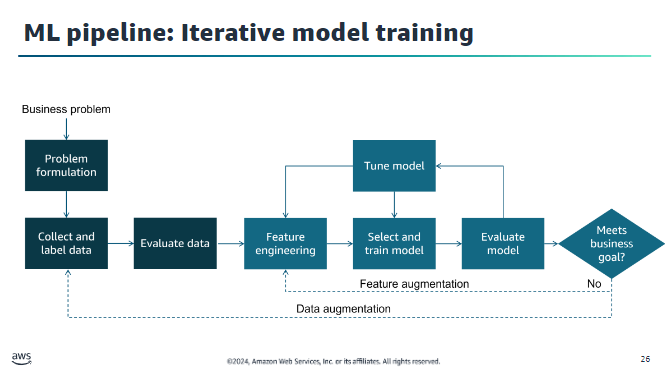
**Key Concepts:**

* **Agent:** The entity that learns and makes decisions (e.g., AWS DeepRacer car).
* **Environment:** The setting where the agent operates and learns (e.g., virtual racetrack).
* **Actions:** The moves the agent makes in the environment (e.g., throttle and steering inputs).
* **Rewards/Penalties:** Feedback given to the agent based on its actions to reinforce or discourage certain behaviours.

**Example:**

* **AWS DeepRacer:**
  + **Agent:** Virtual car.
  + **Environment:** Virtual racetrack.
  + **Goal:** Complete the track as quickly as possible without deviating.
  + **Learning Process:** Uses rewards to incentivize the car to learn the optimal driving behavior through continuous interaction and feedback.

**Use Case:** Reinforcement learning is beneficial when the reward of an intended outcome is known, but the path to achieving it requires discovery through trial and error.



### ML Pipeline: Iterative Model Training

This diagram illustrates the iterative process involved in training a machine learning model to meet business goals. Here are the key steps:

1. **Problem Formulation:**
   * Define the business problem to be solved with machine learning.
2. **Collect and Label Data:**
   * Gather and annotate the data required for training the model.
3. **Evaluate Data:**
   * Assess the quality and relevance of the collected data.
4. **Feature Engineering:**
   * Process and transform raw data into features that better represent the underlying problem to the predictive models.
   * **Feature Augmentation:** If needed, enhance the features to improve model performance.
5. **Select and Train Model:**
   * Choose an appropriate model and train it using the prepared data.
6. **Evaluate Model:**
   * Assess the performance of the trained model to ensure it meets predefined metrics.
7. **Tune Model:**
   * Adjust model parameters and configurations to improve performance.
8. **Meets Business Goal?**
   * **Yes:** If the model meets the business goal, the process ends.
   * **No:** If the model does not meet the business goal, iterate back through the pipeline, possibly performing data augmentation to enhance the training data and repeating the process until the desired performance is achieved.

**Iterative Process:**

* The process is iterative, involving repeated tuning of the model and re-evaluation until the model meets the business goal.

This iterative approach ensures continuous improvement and optimization of the machine learning model to align with business objectives.

#### Overfitting and Underfitting in Model Training

**Overfitting:**

* **Definition:** Occurs when a model performs well on training data but poorly on evaluation data.
* **Cause:** The model memorizes the training data and fails to generalize to unseen examples.
* **Consequence:** Poor performance on new, unseen data.

**Underfitting:**

* **Definition:** Occurs when a model performs poorly on training data.
* **Cause:** The model cannot capture the relationship between input examples (X) and target values (Y).
* **Consequence:** Poor performance on both training and evaluation data.

#### Importance of Understanding Model Fit

* **Identifying Root Causes:** Helps diagnose whether poor model accuracy is due to overfitting or underfitting.
* **Taking Corrective Steps:** Guides actions to improve model performance by analyzing prediction errors on training and evaluation data.

### Python Libraries for Data Science and Machine Learning

**pandas:**

* **Purpose:** Open-source library for data handling and analysis.
* **Key Feature:** Represents data in a table format similar to a spreadsheet, known as a DataFrame.

**Matplotlib:**

* **Purpose:** Library for creating static, animated, and interactive visualizations in Python.
* **Usage:** Used to generate plots of data.

**Seaborn:**

* **Purpose:** Data visualization library built on Matplotlib.
* **Key Feature:** Provides a high-level interface for drawing informative statistical graphics.

**NumPy:**

* **Purpose:** Fundamental package for scientific computing in Python.
* **Features:**
  + N-dimensional array objects.
  + Useful mathematical functions (e.g., linear algebra, Fourier transform, random number capabilities).

**scikit-learn:**

* **Purpose:** Open-source machine learning library supporting both supervised and unsupervised learning.
* **Features:**
  + Model fitting, data preprocessing, model selection and evaluation.
  + Built on NumPy, SciPy, and Matplotlib.
* **Usage:** Good for exploring machine learning, although in the course it’s used to borrow a few functions.

### Challenges in Machine Learning

**Data Challenges:**

* **Quality and Consistency:** Much data is poor-quality and inconsistent.
* **Representation:** Ensuring data accurately represents the problem (e.g., having enough examples of credit card fraud).
* **Quantity:** More data generally improves model performance.
* **Model Fit:** Avoiding overfitting and underfitting.

**User Challenges:**

* **Experience:** Need for data-science expertise.
* **Staffing:** Cost-effectiveness of hiring a team of data scientists.
* **Management Support:** Ensuring management backs the use of ML.

**Business Challenges:**

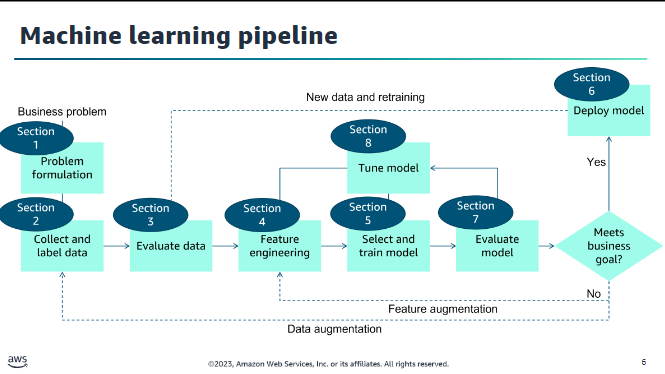
* **Complexity:** Difficulty in formulating complex problems into ML problems.
* **Explainability:** Ensuring the model can be understood and accepted by the business.
* **Cost:** Considerations around the cost of building, updating, and operating an ML solution.

**Technology Challenges:**

* **Data Access and Security:** Ensuring access to required data and meeting regulatory requirements.
* **Tools and Frameworks:** Selecting appropriate tools and frameworks.
* **Integration:** Ensuring the ML solution integrates well with other systems.

# Module 3

## Machine Learning pipeline



Section 1 covers problem formulation and the datasets that you will use throughout this module.

Section 2 explains how to obtain and secure data for your machine learning activities.

Section 3 shows you the tools and techniques for understanding your data.

Section 4 deals with preprocessing your data so that it is ready to train a model.

Section 5 is about selecting and training an appropriate machine learning model.

Section 6 shows you how to deploy a module so that you can predict possible outcomes.

Section 7 examines the process of evaluating the performance of a machine learning model.

Section 8 walks you through tuning the model.

The machine learning pipeline is an iterative process. When you work on a real-world problem, you might iterate many times before you find a solution that meets the business needs.

## Key Takeaways

Business problems must be converted into an ML problem. Questions to ask include –

•Have we asked why enough times to get a solid business problem statement and know why it is

important?

•Can you measure the outcome or impact if your solution is implemented?

•Most business problems fall into one of two categories –

•Classification (binary or multi): Does the target belong to a class?

•Regression: Can you predict a numerical value?

## Section 2 - Collecting and securing data:

### Data Sources:

**Private Data**: Data from your or your customers' systems, like log files and databases.

**Commercial Data**: Data from commercial entities (e.g., Reuters, Change Healthcare) available via subscription, providing curated information such as news, healthcare transactions, and business records.

**Open-Source Data**: Publicly available datasets for research and teaching from sources like AWS, Kaggle, and government organizations.

### Observations:

Supervised machine learning problems need much data—also called observations—for which you know the target answer or prediction. This kind of data is called labeled data. Each observation in your data is made up of two elements: the targetand the features. The target is the answer that you want to predict.

### Extracting, Transforming, Loading Data:

* **Extract** –Pull the data from the sources to a single location.
* **Transform** –During extraction, the data might need to be modified, matching records might need to be combined, or other transformations might be necessary.
* **Load** –Finally, the data is loaded into a repository, such as Amazon S3 or Amazon Athena.

### AWS Glue:

AWS Glue is a fully managed ETL service that simplifies and automates data categorization, cleaning, enrichment, and transfer between data stores. Key components include:

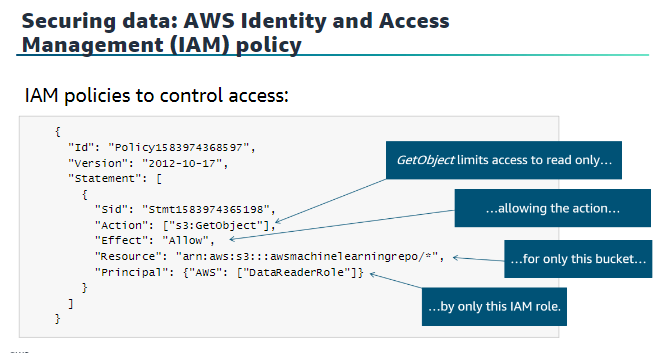
* **Data Catalog**: Central metadata repository.
* **ETL Engine**: Auto-generates Python or Scala code.
* **Flexible Scheduler**: Manages dependencies, monitoring, and retries.

AWS Glue is serverless and supports semi structured data with dynamic frames, which offer schema flexibility and advanced transformations.

It integrates with Apache Spark and provides both a console and API for data discovery, transformation, and querying.

Additionally, it includes machine learning capabilities for tasks like identifying duplicate records.

### IAM:



The AWS Identity and Access Management (IAM) service controls access to resources. Make sure that you correctly secure your data within AWS to avoid data breaches.

**Securing data: AWS CloudTrail for audit**

AWS CloudTrail tracks user activity and application programming

interface (API) usage.

## Section 3 - Evaluating your data

### Pandas:

* Reformats data into tabular representation (DataFrame)
* Converts common formats like comma-separated values (CSV), JavaScript Object Notation (JSON), Excel, Pickle, and others
* Use descriptive statistics to learn about the dataset

• Create visualizations with pandas to examine the dataset in more detail

### Descriptive statistics:

are essential for understanding and preparing data for machine learning models. They can be categorized into:

1. **Overall Statistics**: Provide the number of rows (instances) and columns (features or attributes) in a dataset. This helps identify issues like high dimensionality, which can negatively impact model performance.
2. **Attribute Statistics**: Focus on numeric attributes, providing insights into the data's shape through metrics like mean, standard deviation, variance, minimum, and maximum values.
3. **Multivariate Statistics**: Examine relationships between multiple variables, such as correlations. Identifying high correlations between attributes is crucial because it can lead to poor model performance, such as failure in model convergence.

Mean and median are two different measures that describe the extent that your data is clustered around some value or position.

Mean can be a useful method for understanding your data when the data is symmetrical.

### Plotting Techniques:

#### Histogram:

Histogram is often a good visualization technique for seeing the overall behaviour of a particular feature. With a histogram, you can answer questions, such as:

Is the feature data normally distributed?

How many peaks are in the data?

Does that particular feature have any skewness?

#### Scatter Plot:

When you have more than two numerical variables in a feature dataset, sometimes you want to look at their relationship. A scatter plot is a good way to spot any special relationships among those variables.

### Correlation Matrix:

If the correlation is zero, it means that no linear relationship exists, but it does not mean that no relationship exists. It's only an indication that the two variables have no linear relationship.

## Section 4

### Feature Engineering

1. feature selection
2. feature extraction or creation

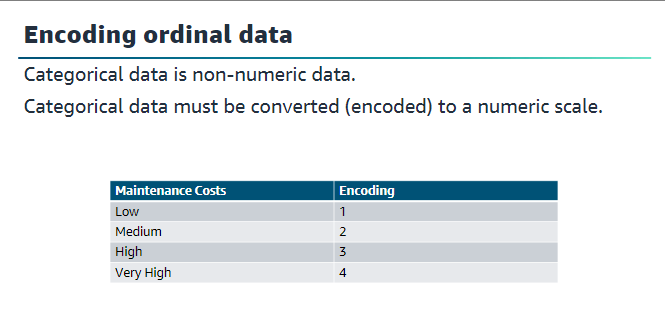
**1.feature selection:**

Feature selection is applied to prevent either redundancy or irrelevance in the existing features, or to get a limited number of features to prevent overfitting

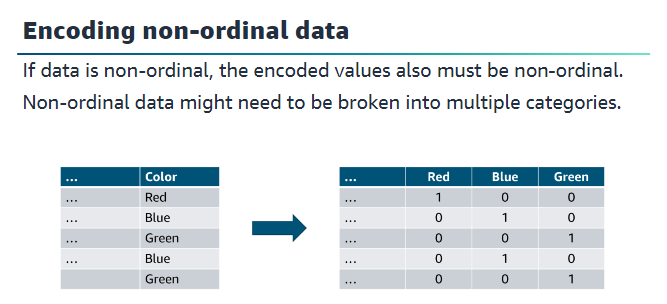
**2.Feature Extraction:**

Feature extraction is about building up valuable information from raw data by reformatting, combining, and transforming primary features into new ones

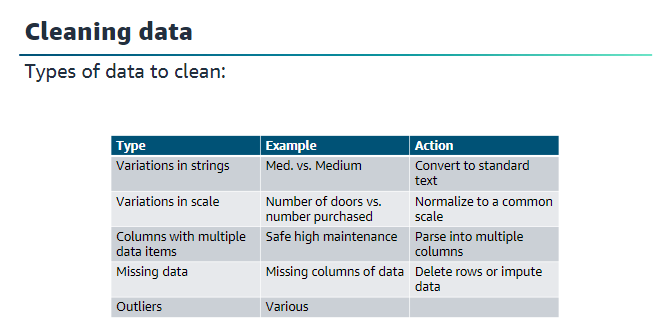
### Data Cleaning:



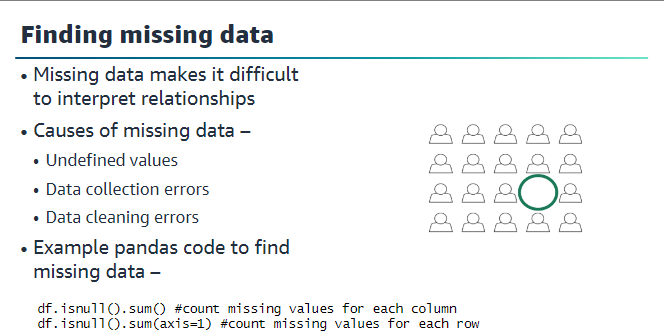
Tools such as Scikit-learns and pandas can be used to encode your categorical data

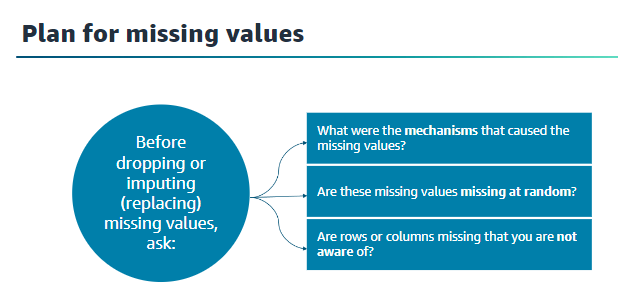


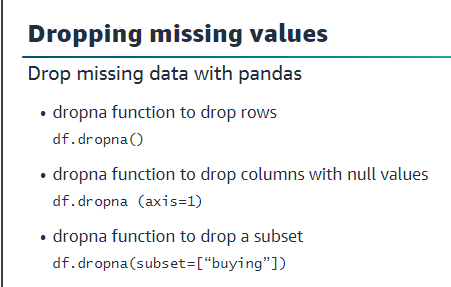
#### Types of data to clean:

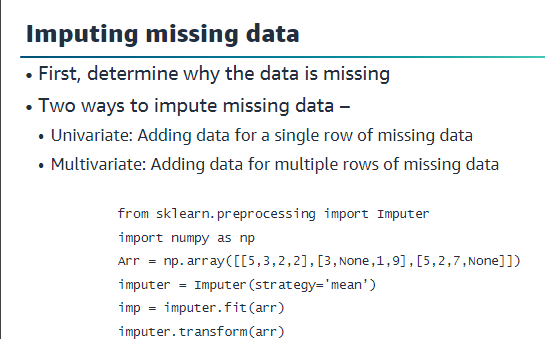


### Finding Missing Data:









### Outliers

•

Outliers can –

•Provide a broader picture of the data

•Make accurate predictions difficult Indicate the need for more columns

•Types of outliers –

•Univariate: Abnormal values for a single variable

•Multivariate: Abnormal values for a combination of two or more variables

Outliers are points in your dataset that lie at an abnormal distance from other values. They are not always something that you want to clean up, because they can add richness to your dataset. However, they can also make it harder to make accurate predictions. The outliers affect accuracy because they skew values away from the other more normal

values that are related to that feature.

### Finding outliers

•Box plots show variation and distance from the mean

•Scatter plots can also show outliers.

### Feature selection

is crucial for training effective machine learning models and involves three main methods:

1. **Filter Methods**: Use statistical measures like Pearson’s correlation, LDA, ANOVA, and **Chi-square** to evaluate feature relevance. They are fast, less computationally intensive, and provide general feature sets not tailored to specific models.
2. **Wrapper Methods**: Involve training models on different feature subsets to evaluate their performance. Methods include forward selection (adding features) and backward selection (removing features). Although computationally intensive, they typically find the best-performing feature set for a specific model.
3. **Embedded Methods**: Integrate feature selection within the model training process. Examples include decision trees, LASSO, and RIDGE regression, which have built-in mechanisms for selecting features and reducing overfitting.

Filter methods are often used as a preprocessing step for wrapper methods to handle larger problems more efficiently.

## Section 5

#### Training:

##### Hold-out

* The hold-out method involves splitting your data into training, validation, and testing sets.
* Typically, the training data, which includes both features and labels, is used to train the model.
* The model's performance is then evaluated and tuned using the validation dataset.
* Finally, the test dataset, which contains only features, is used to predict labels and assess the model's performance in a production-like scenario.
* Common splits for the data are 80% for training, 10% for validation, and 10% for testing, or 70% for training, 15% for validation, and 15% for testing if the dataset is large.

##### K-fold validation

* K-fold cross-validation is a technique used for small datasets to utilize as much data as possible and obtain good metrics for model selection.
* The data is divided into K segments, and each segment is used as a validation set while the remaining segments are used for training.
* This process is repeated K times, ensuring all data is used for training and validation. The results are averaged to evaluate model performance.
* For example, in 5-fold cross-validation, the data is split into five chunks, and each chunk is used as a test set while the others are used for training, repeated five times. This helps in comparing the performance of different models.

##### K-means

* It attempts to find discrete groupings within data, where members of a group are as similar as possible.
* The means in k-Means is the averaging of the data. This averaging helps to find the center of the grouping.

#### Key Takeaways:

* Split data into training and testing sets
* Optionally, split into three sets, including validation set
* Can use k-fold cross validation to use all the non-test data for validation
* Can use two key algorithms for supervised learning –
* XGBoost
* Linear learner
* Use k-means for unsupervised learning
* Use Amazon SageMaker training jobs

## Section-6

Some key takeaways from this section of the module include:

* You can deploy your trained model by using Amazon SageMaker to handle API calls from applications, or to perform predictions by using a batch transformation.
* The goal of your model is to generate predictions to answer the business problem. Be sure that your model can generate good results before you deploy it to production.
* Use Single-model endpoints for simple use cases and use multi-model endpoint support to save resources when you have multiple models to deploy.

## Section -7

 Hold-out Method:

* Splits data into training, validation, and testing sets.
* Training data is used to train the model.
* Validation data is used for tuning the model.
* Test data evaluates the model’s performance.
* Common splits: 80/10/10 or 70/15/15 for training/validation/testing.

 K-Fold Cross-Validation:

* Divides data into K segments.
* Each segment is used as a validation set while the rest are used for training.
* Repeated K times, ensuring all data is used for training and validation.
* Results are averaged for model performance evaluation.
* Useful for comparing different models.

####  Model Selection Based on Sensitivity and Specificity:

* Model A: Sensitivity 60%, Specificity 64%.
* Model B: Sensitivity 84%, Specificity 18%.
* Choose Model B to identify as many cats as possible.
* Choose Model A to identify animals that are not cats accurately.
* For diagnosing heart disease, consider the trade-offs carefully.

####  Threshold Adjustment:

* Models return probabilities; thresholds determine classification.
* Changing the threshold affects the sensitivity and specificity.
* Visualize this with ROC curves.

####  ROC and AUC-ROC:

* ROC graph plots sensitivity against false-positive rate.
* High sensitivity and low false-positive rate are ideal.
* AUC-ROC provides a single metric to compare models.

####  Other Classification Metrics:

* **Accuracy**: (TP + TN) / (TP + TN + FP + FN), limited by class imbalance.
* **Precision**: TP / (TP + FP), important when false positives are costly.
* **F1 Score**: Combines precision and sensitivity, useful for class imbalance.

####  Regression Metrics:

* Mean squared error (MSE): Measures average squared differences between predicted and actual values.
* R-squared: Indicates proportion of variance explained by the model.

####  ML Tuning Process:

* Train model, evaluate on test data, calculate metrics.
* Use metrics to inform model tuning.
* Possible adjustments include feature selection, data changes, and parameter tuning.

## Section-8

#### Hyperparameter Tuning Overview

##### **Hyperparameters**:

* + **Model Hyperparameters**: Define the model architecture, such as the number of layers, activation functions, filter size, pooling, stride, and padding.
  + **Optimizer Hyperparameters**: Determine how the model learns patterns, including optimizers like gradient descent, Adam, and methods for initializing weights.
  + **Data Hyperparameters**: Relate to data attributes and augmentation techniques, useful for small or homogeneous datasets.

##### **Manual Tuning**:

* + Traditionally done manually based on intuition and experience.
  + Involves training the model, scoring on validation data, and repeating until satisfactory results are achieved.
  + Often not thorough or efficient.

##### **Automated Hyperparameter Tuning with Amazon SageMaker**:

* + Runs multiple training jobs to find the best model version based on specified hyperparameter ranges.
  + Uses Gaussian Process regression and Bayesian optimization.
  + Compatible with SageMaker built-in algorithms, prebuilt deep learning frameworks, and custom algorithms.
  + Example: Maximizing AUC for a binary classification problem on a fraud dataset.

##### **Tuning Best Practices**:

* + Focus on key hyperparameters rather than adjusting all.
  + Limit the range of values to what is most effective.
  + Prefer running one training job at a time for better results.
  + Ensure the correct objective metric is reported in distributed training jobs.
  + Convert log-scaled hyperparameters to linear-scaled if possible for better optimization.

##### **General Notes**:

* + Tuning doesn’t always improve the model.
  + It is an iterative process and should be part of the scientific method in ML development.
  + Impractical to explore all possible combinations in complex systems like deep learning neural networks.

# Module 4

## Section -1 : Forecasting

Forecasting in machine learning is essential due to its capability to predict future outcomes based on historical data, particularly when these data involve a time component.

Time series data can be categorized into two types: univariate (one variable) and multivariate (more than one variable). These datasets often exhibit patterns such as:

* **Trend:** Values increase, decrease, or remain stable over time.
* **Seasonal:** Patterns repeat based on seasons within a year.
* **Cyclical:** Patterns repeat in a non-seasonal manner.
* **Irregular:** Changes appear random or without a discernible pattern.

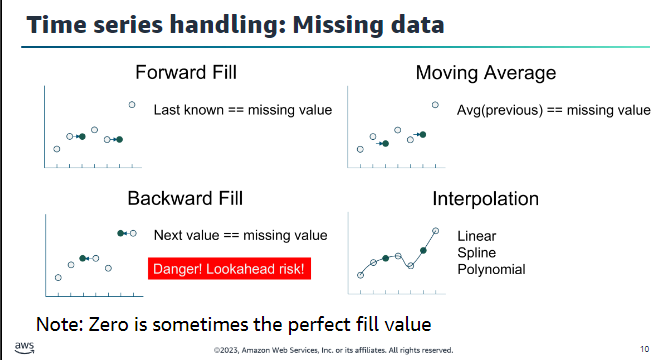
Common use cases for forecasting include:

* **Marketing:** Sales forecasting and demand projections.
* **Inventory Management:** Anticipating inventory levels and delivery times.
* **Energy Consumption:** Predicting when and where energy is needed.
* **Weather Forecasting:** Used by governments and for commercial agricultural applications.

## Section:2 Time Series Handling

* **Time Series Data**:
  + Captured in chronological sequence over a defined period.
  + Adds value to machine learning models by deriving meaning from changes over time.
  + Often correlated, indicating dependencies between data points.
* **Related data** informs the time series data for example, price or Promotions
* **Metadata** might also be needed to explain predictions—for example, brand name or category
* **Handling Data Dependencies**:
  + Regression assumes independence of data points, requiring methods to handle dependencies.
  + Methods increase prediction validity.
* **Augmenting Forecasting Models**:
  + Related data: e.g., product information like item ID or sales price.
  + Metadata: e.g., brand names for retail datasets or genres for music/videos.
* **Timestamp Challenges**:
  + Incomplete and varying formats (e.g., yyyy-mm-dd HH:MM, yyyy-mm).
  + Differences between UTC and local time.
  + Misleading timestamps (e.g., arrival vs. completion times).
  + Adjusting target timescale if data granularity is mismatched.
  + Daylight savings time complications.

### Handling Missing Data:

* + Common in real-world forecasting, e.g., out-of-stock situations in retail.
  + Methods to address missing values:
    - Forward fill: Last known value.
    - Moving average: Average of last known values.
    - Backward fill: Next known value (avoid lookahead).
    - Interpolation: Equation-based calculation.
    - Zero fill: Often used in retail to represent no orders.
* 

### Downsampling:

* + Converting to a less finely grained time (e.g., hourly to daily).
  + Deciding on value combination methods (e.g., summing sales quantities or averaging temperatures)

### Upsampling:

* + Difficult without additional data or domain knowledge.
  + Matching time series frequencies or dealing with irregular time series.
  + Using formulas to create finer-grained data (e.g., daily to hourly sales).

### Smoothing Data:

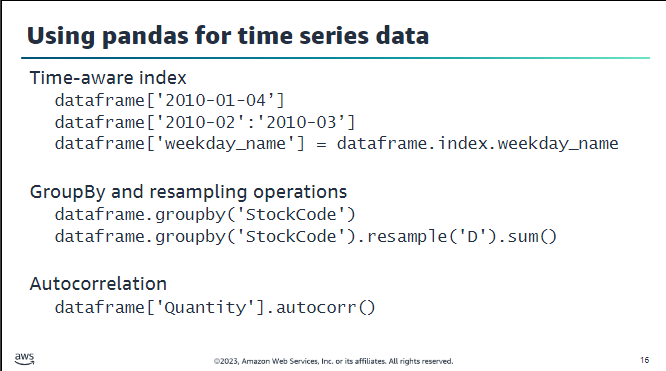
* + Removes outliers and anomalies.
  + Reasons for smoothing:
    - Data preparation: Removing errors.
    - Visualization: Reducing noise.
  + Impact: Cleaner data, model compatibility, potential production improvements.
  + Consider the model’s need for noisy data and the feasibility of smoothing in production.

### Seasonality:

* + Repeating observations with stable frequency (e.g., quarterly or yearly sales spikes).
  + Multiple types of seasonality in one dataset.
  + Incorporating local holidays into forecasts.

### Stationarity, Trends, and Autocorrelation:

* + **Stationarity**: Stability of the system; influences the predictability of future behavior.
  + **Trends**: Trends can dominate values and affect correlation estimates.
  + **Autocorrelation**: Linear relationship between time points; special problem for time series data.
    - Can overstate model accuracy.
    - Some algorithms can correct for autocorrelation.
  + Be cautious with correlations; they don’t imply causation.



## Section-3 : Amazon forecast

 **Supported Domains**:

* **Retail**: Product demand.
* **Inventory Planning**: Raw materials requirements.
* **Amazon EC2 Capacity**: Capacity demand for Amazon EC2.
* **Workforce**: Workload projections.
* **Web Traffic**: Projected website traffic.
* **Metrics**: Projecting metrics like revenue, sales, or cash flow.
* **Custom**: Projections for unique domains.

 **Retail Forecasting Example**:

* **Time Series Data**: Timestamp, item ID, quantity sold.
* **Metadata**: Item category, item color.
* **Related Data**: In-stock data, promotion data (timestamp, item ID, price).

 **Web Traffic Forecast Example**:

* **Time Series Data**: Webpage ID, page views per month, timestamp.
* **Related Data**: Page category, geographic identifier.
* **Metadata**: Region, sales promotion information.

 **Amazon Forecast Algorithms**:

* **Algorithms**: ARIMA, DeepAR+, ETS, NPTS, Prophet.
* **AutoML**: Tries all algorithms to determine the best fit.
* **Reference**: AWS documentation for detailed algorithm selection.

 **Evaluating Forecasts: Back Testing**:

* **Data Split**: Training and test datasets.
* **Back Test Windows**: Multiple windows to train and test the model.
* **BackTestWindowOffset**: Parameter to adjust data splitting.
* **Accuracy Measurement**: Essential for model validation.

 **Evaluation Metrics**:

* **Weighted Quantile Loss (wQuantileLoss)**:
  + **Quantiles**: 10%, 50%, and 90%.
  + **Error Calculation**: Average error for each quantile.
  + **Use Case Example**: Forecasting product demand with varying levels of inventory risk.
* **Root Mean Square Error (RMSE)**:
  + **Calculation**: Squares the differences between actual and forecasted values.
  + **Suitability**: Good for data with consistent error sizes.
  + **Lower RMSE**: Indicates more reliable forecasts.

 **Model Accuracy Example**:

* **Scenario**: Predicting demand for AnyCompany brand shoes.
* **Forecast**: 1,000 pairs per month.
* **Quantiles**:
  + **P10**: 10% of the time, fewer than 880 pairs ordered.
  + **P50**: 50% of the time, fewer than 1,050 pairs ordered.
  + **P90**: 90% of the time, fewer than 1,200 pairs ordered.
* **Decision Making**: Based on risk assessment of inventory levels.

# Module:5

## Computer Vision overview

Computer vision is the automated extraction of information from

digital images

Some of the primary use cases for computer vision include these examples.

**Public safety and home security**

Computer vision with image and facial recognition can help to quickly identify unlawful entries or persons of

interest. This process can result in safer communities and a more effective way of deterring crimes.

**Authentication and enhanced computer**

-human interaction Enhanced human

-computer interaction can improve customer satisfaction. Examples include products that are based on customer sentiment analysis in retail outlets or faster banking services with quick authentication that

is based on customer identity and preferences.

**Content management and analysis**

Millions of images are added every day to media and social channels. The use of computer vision technologies—such as metadata extraction and image classification—can improve efficiency and revenue opportunities.

**Autonomous driving**

By using computer-vision technologies, auto manufacturers can provide improved and safer self-driving car navigation, which can help realize autonomous driving and make it a reliable transportation option.

**Medical imaging**

Medical image analysis with computer vision can improve the accuracy and speed of a patient's medical

diagnosis, which can result in better treatment outcomes and life expectancy.

Manufacturing process control Well-trained computer vision that is incorporated into robotics can improve quality assurance and operational

efficiencies in manufacturing applications. This process can result in more reliable and cost-effective products.

### Computer vision use cases

* Content recognition
* Video analysis
* Instance tracking
* Action recognition
* Motion estimation

## Section:2 :Amazon Rekognition

AmazonRekognition enables you to perform the following types of analysis:

**Searchable image and video libraries–**

Amazon Rekognition makes images and stored videos searchable so

that you can discover the objects and scenes that appear in them.

**Face-based user verification–**

Amazon Rekognition enables your applications to confirm user identities by comparing their live image with a reference image.

**Sentiment and demographic analysis–**

Amazon Rekognition interprets emotional expressions, such as

happy, sad, or surprise. It can also interpret demographic information from facial images, such as gender.

**Unsafe content detection–**

Amazon Rekognition can detect inappropriate content in images and in stored videos

**Text detection–**

Amazon Rekognition Text in Image enables you to recognize and extract text content from images

### Facial Detection and Recognition with Amazon Rekognition

#### Facial Detection

Amazon Rekognition uses a model tuned for detecting faces and facial features, which provides the following details:

1. **Bounding Box**: Coordinates of the box surrounding the face.
2. **Confidence**: The confidence level that the bounding box contains a face.
3. **Facial Landmarks**: Array of facial landmarks (e.g., left eye, right eye, mouth) with x and y coordinates.
4. **Facial Attributes**: Attributes like beard presence, gender, and sunglasses, with confidence scores.
5. **Quality**: Brightness and sharpness of the face.
6. **Pose**: Rotation of the face within the image.
7. **Emotions**: Set of detected emotions with confidence levels.

#### Facial Recognition

Amazon Rekognition can compare two images to determine if they contain the same person, requiring a source and a target image. The results include:

1. **Face Match**:
   * Bounding box and confidence score.
   * Similarity score.
   * Facial landmark locations.
2. **Source Face Information**:
   * Bounding box and confidence.
   * Facial landmarks.
3. **Unmatched Faces**:
   * Bounding box and confidence.
   * Facial landmarks.

#### Searching for Known Faces

To search for known faces, you must train the model with a collection of target faces:

1. **Train the Model**: Provide a collection of images.
2. **Store Metadata**: Store facial metadata associated with each image.
3. **SearchFacesByImage Operation**: Searches for faces from the collection, returning:
   * Bounding boxes.
   * Confidence scores.
   * ExternalImageId (links to source images).

#### Guidelines for Responsible Use

1. **Confidence Scores**: Use appropriate confidence scores for your use case.
2. **Image-Based Detection**: Gender and emotion are inferred from images, not identity or actual emotional state.
3. **Privacy and Rights**: Ensure the use of facial recognition does not violate individual rights or privacy.
4. **Human Analysis**: Avoid autonomous decisions in critical scenarios; use technology to narrow potential matches for human analysis.
5. **Disclosure and Consent**: Clearly disclose the use of facial recognition technology and obtain consent.

Amazon Rekognition captures bounding boxes, facial attributes, emotions, quality, and landmarks with associated confidence scores, inferring gender and emotions from the image. Responsible use includes respecting privacy and ensuring human oversight in critical applications.

### Working with Stored and Streaming Videos Using Amazon Rekognition Stored Videos

**Process Overview:**

1. **Start Detection**: Initiate detection for various elements such as people, faces, labels, celebrities, text, and inappropriate content.
2. **Monitor SQS Queue**: Monitor the Amazon Simple Queue Service (Amazon SQS) queue for completion status notifications.
3. **Retrieve Results**: Use corresponding Get operations to fetch detection results.

**Stored Videos Detection Steps:**

* **Storage**: Store videos in an S3 bucket.
* **Detection Types and Operations**:
  + **People**: StartPersonTracking / GetPersonTracking
  + **Faces**: StartFaceDetection / GetFaceDetection
  + **Labels**: StartLabelDetection / GetLabelDetection
  + **Celebrities**: StartCelebrityRecognition / GetCelebrityRecognition
  + **Explicit Content**: StartContentModeration / GetContentModeration

**Notification and Result Retrieval:**

* Amazon Rekognition publishes completion status to an Amazon Simple Notification Service (Amazon SNS) topic.
* Route SNS messages to an SQS queue for durability.
* Monitor the SQS queue for notifications.
* Use Get operations to retrieve results, which include labels and timestamps indicating where they were detected in the video.

#### Streaming Videos

**Application Process:**

1. **Stream Video**: Send video to Amazon Kinesis Video Streams.
2. **Connect Stream Processor**: Use Amazon Rekognition Video stream processor to analyze the video stream.
3. **Read Analysis Results**: Read analysis results from the Amazon Kinesis data stream.

**Steps for Streaming Video Detection:**

* **Resources Required**:
  + **Kinesis Video Stream**: For sending video streams to Amazon Rekognition Video.
  + **Rekognition Video Stream Processor**: To manage video stream analysis.
  + **Kinesis Data Stream Consumer**: To read analysis results from the Kinesis data stream.

**Operational Details:**

* **Face Detection**: Create a collection for known faces, similar to the process for images.
* **Stream Processing**: Amazon Rekognition Video analyzes frames, sending JSON frame records to the Kinesis output stream.
* **Frame Records**: Include information about the fragment, frame position, recognized faces, and status information.

## Section 3

### Custom Model Training with Amazon Rekognition Overview

Amazon Rekognition's pre-built models may not detect specific objects relevant to your domain. To address this, you can train custom models using Amazon Rekognition Custom Labels. This process involves creating and using custom datasets to train and evaluate models tailored to your needs.

#### Benefits of Amazon Rekognition Custom Labels

* **Simplified Data Labeling**: Provides a UI for labeling images, including defining bounding boxes.
* **Automated Machine Learning**: Automates data loading, algorithm selection, model training, and performance metrics generation.
* **Simplified Model Evaluation and Feedback**: Allows side-by-side comparison of model predictions versus labels, detailed performance metrics, and iterative retraining with new data.

#### Custom Labeling Process

1. **Collect Images**: Gather images specific to your domain, typically a few hundred, in JPEG or PNG format.
2. **Create Training Dataset**: Upload and label images using the console or Amazon SageMaker Ground Truth. Ensure at least two labels with a minimum of 10 images per label.
3. **Create Test Dataset**: Separate a portion of labeled images for testing model performance.
4. **Train the Model**: Use training datasets to train the model. Automated processes handle algorithm selection and training.
5. **Evaluate Model**: Assess model performance on the test dataset, adding more images to improve accuracy if needed.
6. **Use the Model**: Deploy the model to analyze images via an API operation.

#### Training Steps

1. **Collect Images**:
   * Gather a few hundred domain-specific images.
   * Ensure images vary in lighting, background, and resolution to match production conditions.
2. **Create Training Dataset**:
   * Label images directly using the console or Amazon SageMaker Ground Truth.
   * Ensure each image has at least one label indicating the object, scene, or concept.
   * For object detection, label images with bounding boxes around objects.
3. **Create Test Dataset**:
   * Create a dataset for evaluating the model, ensuring it represents the diversity of production images.
4. **Train the Model**:
   * Use the labeled training dataset to train the custom model.
5. **Evaluate**:
   * Evaluate model performance using the test dataset.
   * Review predictions versus labels and improve by adding more labeled images if necessary.
6. **Use Model**:
   * Deploy the custom model for image analysis, utilizing feedback to continuously improve performance.

#### Image and Label Guidelines

* **Image-Level Labels**: Apply labels to the entire image for scene or concept detection (e.g., labeling a beach scene).
* **Object-Level Labels**: Apply bounding boxes around specific objects within an image for object detection (e.g., labeling different Amazon Echo devices).

#### Practical Considerations

* Training a custom model typically requires fewer images than starting from scratch due to the foundational training on tens of millions of images in Amazon Rekognition.
* Using different models for different domains (e.g., machine parts versus plant health) improves accuracy and relevance.
* Ensuring labeled images reflect production conditions (lighting, background, resolution) enhances model performance.

#### Step 3: Create Test Dataset

* **Purpose**: Validate and evaluate the model’s performance.
* **Process**:
  + **Create Your Own Test Dataset**: Manually select images for testing.
  + **Automatic Split**: Use Amazon Rekognition Custom Labels to split the training dataset into an 80/20 split (80% for training, 20% for testing).

#### Step 4: Train the Model

* **Process**:
  + Define training and test datasets.
  + Amazon Rekognition Custom Labels automatically loads and inspects data, selects ML algorithms, trains the model, and provides performance metrics.
* **Cost**: Based on the amount of time the model takes to train. Larger datasets take longer.

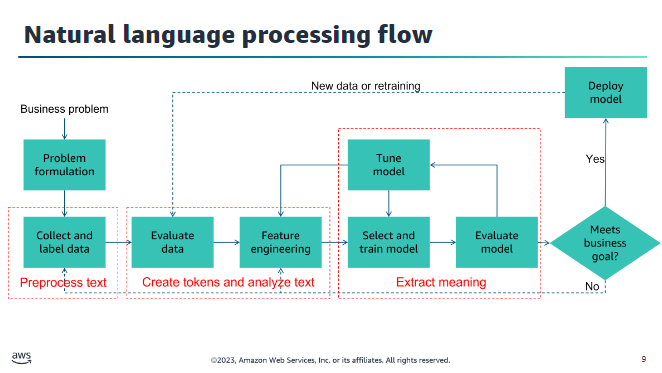
#### Step 5: Evaluate the Model

* **Metrics**:
  + **Precision**: Proportion of positive results correctly classified.
  + **Recall**: Fraction of test set labels correctly classified.
  + **Overall Model Performance**: F1 Score combines precision and recall, indicating overall performance.
* **Confusion Matrix**:
  + **True Positive (TP)**: Correctly predicts the presence of the label.
  + **False Positive (FP)**: Incorrectly predicts the presence of the label.
  + **False Negative (FN)**: Fails to predict the label when it is present.
  + **True Negative (TN)**: Correctly predicts the absence of the label.
* **Improvement Strategies**:
  + **Reducing False Positives**: Increase confidence threshold, add additional classes, or refine training labels.
  + **Reducing False Negatives**: Lower confidence threshold, use better examples, or split labels into more specific classes.

#### Step 6: Use the Model

* **Deployment**:
  + Start the model from the console or using code.
  + Perform inference using the AWS CLI or SDK.
* **API Call**:
  + Specify the model’s Amazon Resource Name (ARN) and input image.
  + Returns an array of custom labels, each with:
    - Label name.
    - Bounding box coordinates.
    - Confidence score.
  + Use the MinConfidence parameter to filter labels by confidence score. The default threshold can be adjusted based on model training results.

# Module -6



### Natural Language Processing (NLP)

#### What is NLP?

* **Definition**: NLP develops computational algorithms to automatically analyze and represent human language.
* **Purpose**: To process large sets of words, phrases, and sentences by evaluating the structure of language.
* **Historical Context**: NLP systems predate machine learning (ML) and include early examples like speech-to-text on old cell phones and screen readers. Many modern NLP systems now use ML.

#### Hierarchical Structure of Language

* **Words**: The lowest layer of the language hierarchy.
* **Phrases**: Groups of words that form a higher level.
* **Sentences**: Combinations of phrases that convey ideas.

#### NLP Challenges

1. **Lack of Precision**: Language is not precise; words can have different meanings based on context.
2. **Context-Based Meaning**: The meaning of words depends on surrounding words.
3. **Complex Dependencies**: Multiple meanings and dependencies within language.
4. **Lack of Structure**: Language varies greatly, making it difficult to standardize.

#### Key NLP Challenges

* **Discovering Text Structure**: Breaking text into meaningful units like words, phrases, and sentences.
* **Labeling Data**: Applying labels to represent parts of speech according to different grammatical rules.
* **Representing Context**: Capturing the meaning of words based on their context, a complex task due to numerous contexts.
* **Applying Grammar**: Dealing with the infinite variations in human language usage.

#### Use Cases for NLP

1. **Search Applications**: Improving search engines like Google and Bing.
2. **Human-Machine Interfaces**: Enhancing interactions with systems like Alexa.
3. **Market and Social Research**: Performing sentiment analysis for marketing or political campaigns.
4. **Chatbots**: Mimicking human speech in various applications to enhance user interactions.

#### Preprocessing Text

1. **Common Preprocessing Steps**:
   * **Remove Stop Words**: Eliminate common words that are not essential for analysis.
   * **Normalize Text**: Convert similar words into a common form (e.g., run, running, ran).
   * **Standardize Text**: Remove unrecognized words (e.g., acronyms, slang).
2. **Other Preprocessing Steps**:
   * **Encoding**: Convert text into a specific format.
   * **Spelling and Grammar Checks**: Correct errors to improve analysis quality.
3. **Tools**:
   * **Libraries**: Multiple libraries, such as NLTK for Python, assist in text preprocessing.

#### Sample Preprocessing

* **Text Conversion**: Transform text into data by removing unnecessary words.
* **Normalization**: Use stemming and lemmatization to standardize word forms.
* **Standardization**: Remove non-dictionary words.

#### Creating Tokens and Feature Engineering

1. **Tokenization**:
   * **Function**: Use tokens to convert words into items in a DataFrame.
   * **Example Code**:

python

Copy code

from nltk.tokenize import word\_tokenize

text = "this is some sample text."

print(word\_tokenize(text))

# Output: ['this', 'is', 'some', 'sample', 'text', '.']

1. **Feature Engineering**:
   * **Bag of Words**: Captures word frequency in a document.
   * **Term Frequency-Inverse Document Frequency (TF-IDF)**: Calculates a weight for words based on frequency and document appearance.

#### Example NLP Model: Bag of Words

* **Concept**: Creates a vector for each sentence/phrase based on word frequency.
* **Application**: Used in document classification and sentiment analysis.

#### Text Analysis Categories

1. **Classifying Text**:
   * **Process**: Extract features from text, apply ML algorithms, and classify text using models like those in NLTK.
2. **Discovering Similarities**:
   * **Applications**: Auto-correct, spell check, grammar check.
   * **Algorithm**: Edit distance (Levenshtein distance).
3. **Deriving Relationships**:
   * **Process**: Use coreference resolution to find relationships between words or phrases.

#### Capture Context

* **Challenges**: Understanding the context of words is complex.
* **Solution**: Tagging words with appropriate parts of speech and using token functions to help with tagging.
* **Example**: Differentiating meanings of the word "tablet" based on context.

#### Derive Meaning by Entity Extraction

1. **Named Entity Recognition (NER)**:
   * **Components**:
     + Identify noun phrases using dependency charts and part-of-speech tagging.
     + Classify phrases using algorithms like Word2Vec.
     + Disambiguate entities using a knowledge graph.
2. **Example**: Extract entities like "Titanic" and "North Atlantic" from text and derive meaning using a knowledge graph.

## Amazon Services for NLP

#### Amazon Transcribe

Amazon Transcribe is a fully managed service that uses advanced machine learning to convert speech in audio files into text. It supports recognizing recorded voices, converting streaming audio to text, customizing vocabularies, integrating with applications using WebSockets, and building real-time subtitles for multiple languages.

**Key Features**:

* **Speech Recognition**: Recognizes specific voices in an audio file.
* **Customized Vocabulary**: Allows for specialized terms.
* **Integration**: Uses WebSockets for two-way communication.
* **Real-Time Subtitles**: Supports real-time transcription for live feeds.

**Common Use Cases**:

1. **Medical Transcription**: Captures spoken notes of medical professionals.
2. **Video Subtitles**: Automatically generates subtitles from video.
3. **Streaming Content Labeling**: Labels media content for further analysis.
4. **Call Center Monitoring**: Records and analyzes customer service or sales calls for insights.

#### Amazon Polly

Amazon Polly is a managed service that converts text into lifelike speech. It supports multiple languages and offers various lifelike voices, generating voice from plain text or SSML (Speech Synthesis Markup Language) format and creating output in multiple audio formats.

**Key Features**:

* **Text to Speech**: Converts plain text or SSML to speech.
* **Multiple Audio Formats**: Outputs in MP3, Vorbis, and PCM formats.
* **Pay-for-Use Policy**: Cost-effective pricing using AWS infrastructure.
* **Regulated Workloads**: Eligible for HIPAA and PCI DSS regulated workloads.

**Common Use Cases**:

1. **News Service Production**: Generates vocal content from written stories.
2. **Language Training Systems**: Creates systems for language learning.
3. **Navigation Systems**: Adds voice to geo-based applications.
4. **Animation Production**: Provides voices for animated characters.

#### Amazon Translate

Amazon Translate is a fully managed text translation service that uses advanced machine learning technologies to provide high-quality translation on demand. It is used to develop multilingual user experiences, translate documents, and analyze text in multiple languages.

**Key Features**:

* **Multilingual User Experiences**: Creates applications that support multiple languages.
* **Document Translation**: Translates documents into various languages.
* **Integration**: Works seamlessly with Amazon Comprehend, Amazon Transcribe, and Amazon Polly for enhanced functionality.

**Common Use Cases**:

1. **International Websites**: Globalizes websites quickly.
2. **Software Localization**: Reduces costs and time for localizing software.
3. **Multilingual Chatbots**: Creates chatbots that support multiple languages.
4. **International Media Management**: Lowers localization costs for media companies.

#### Amazon Comprehend

Amazon Comprehend uses natural language processing (NLP) to extract insights from documents by recognizing entities, key phrases, languages, sentiments, and other elements.

**Key Features**:

* **Entity Extraction**: Identifies key entities such as people and locations.
* **Language Identification**: Detects the language used in a document.
* **Sentiment Analysis**: Determines the sentiment expressed in a document.
* **Part of Speech Tagging**: Tags individual words with their part of speech.

**Common Use Cases**:

1. **Document Analysis**: Analyzes legal and medical documents.
2. **Fraud Detection**: Examines large datasets for patterns of illegal transactions.
3. **Mobile App Analysis**: Identifies usage patterns for app improvement.
4. **Content Management**: Tags and analyzes content for media companies.

#### Amazon Lex

Amazon Lex is a service for building conversational interfaces using voice and text. It enables developers to create chatbots that can interact through voice and text, ask questions, get answers, and complete tasks.

**Key Features**:

* **Conversational Interfaces**: Powers chatbots with the same engine as Amazon Alexa.
* **Scalability**: Automatically scales with AWS Lambda.
* **Conversation Logs**: Stores logs for analysis.

**Common Use Cases**:

1. **Inventory and Sales Interfaces**: Adds voice interfaces to inventory and sales applications.
2. **Interactive Assistants**: Develops sophisticated assistants for various industries.
3. **Customer Service Interfaces**: Creates human-like voice applications for customer service.
4. **Database Queries**: Combines with AWS database services for data analysis with a human-like language interface.

# Module 7

## Generative AI

* Generative AI is a type of artificial intelligence that can create new content and ideas, including conversations, stories, images, videos, and music.
* AI generators are powered by pretrained machine learning models called foundation models (FMs).
* These models are capable of producing content, so you don’t have to. This AI-generated content can be edited, so you can make the necessary modifications that meet your needs.

### Generative AI use cases

#### Enhance customer experiences:

Chatbots and virtual assistants can enhance customer service experiences by streamlining customer self-service and reducing operational costs. Generative AI can also help with summarizing and analyzing customer conversations and creating better personalized customer experiences.

#### Boost employee productivity:

Generative AI can help boost employee productivity by

accelerating application development, automating report generation, and improving content searches through a conversational interface.

#### Optimize processes:

Business operations can be optimized by using intelligent

document processing that extracts and summarizes data through generative AI-powered questions and answers. Generative AI can also help improve supply chain logistics and augment data by generating synthetic data to train ML models.

#### Enhance creativity and content creation:

AI-generated media can enhance creativity and content creation. Marketing content created using generative AI, such as blog

posts or social media updates, can save time and resources. Sales content, guidance, and enablement can also be generated to match a prospect’s profile and behavior, thereby improving response rates. Product developers can use generative AI to generate multiple design prototypes based on certain inputs and constraints, which speeds up the ideation phase.

## Foundation model:

AI model that is trained on broad data at scale, is designed for generality of output, and can be adapted to a wide range of distinctive tasks".

These tasks include text generation, data summarization, information extraction,

question and answer responses, and chatbot interactions.

### Comparison of traditional and foundation models

* Foundation models (FMs) differ from traditional machine learning models in that they are pretrained on large datasets and can be adapted for multiple tasks without the need for extensive labeled data or training from scratch.  
   FMs can be customized for domain-specific functions, utilizing significantly less data and compute resources.
* Examples of tasks FMs can perform include language processing, visual comprehension, and code generation.
* Notable examples of FMs are Amazon Titan for natural language processing and Stable Diffusion for visual comprehension.

## Large Language Models (LLMs):

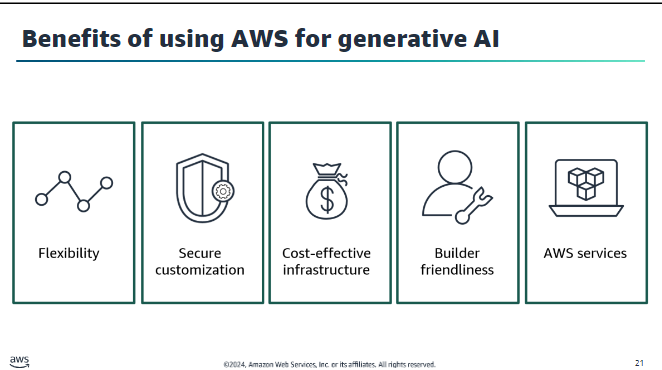
* **Training Data:**
  + Trained on trillions of words
* **Capabilities:**
  + Understand, learn, and generate text
  + Nearly indistinguishable from text produced by humans
* **Adaptability:**
  + Adaptable to various tasks such as:
    - Text generation
    - Summarization
    - Sentiment analysis
  + Can perform domain-specific functions
* **Advantages:**
  + Eliminates the need for gathering labeled data for each model
  + Can use the same pretrained LLM to adapt to various tasks
  + Customizable to add domain-specific knowledge, such as internal documents for a particular business

## Prompt engineering

* Prompt engineering is the process of designing and refining the prompts or input stimuli for a language model to generate specific types of output.
* Prompt engineering involves selecting appropriate keywords, providing context, and shaping the input in a way that encourages the model to produce the desired response. It is a vital technique to actively shape the behavior and output of foundation models.

### A prompt is composed of the following:

* Instructions: This is a task for the large language model to do. It provides a task description or instruction for how the model will perform.
* Context: This is external information to guide the model.
* Input data: This is the input for which you want a response.
* The model will then generate an output.
* Output: This is the output type or format.
* If the model does not give you the output that you want, you need to alter the prompt or provide the model with examples of the tasks in the prompt. This is called prompt engineering.



AWS offers several generative AI offerings: Amazon Bedrock, AWS Inferentia, AWS Trainium, and Amazon SageMaker JumpStart

## Amazon CodeWhisperer

Features and tools include the following:

* Integrates with your IDE
* Generates whole-line and full-function code suggestions in your IDE Speeds up development and code writing
* Makes suggestions based on the code comments on current and previous inputs using generative AI
* Detects securities vulnerabilities and aligns code to best practices

### Benefits of using Amazon CodeWhisperer

* Accelerate any coding tasks.
* Optimize for AWS.
* Use AI responsibly.
* Enhance application security.
* Generate functions.
* Write unit tests.

CodeWhisperer also has a security scanner that helps mitigate security vulnerabilities, thus safeguarding the integrity of the codebase.

There are two tiers of service for Amazon CodeWhisperer:

Individual Tier and

Professional Tier