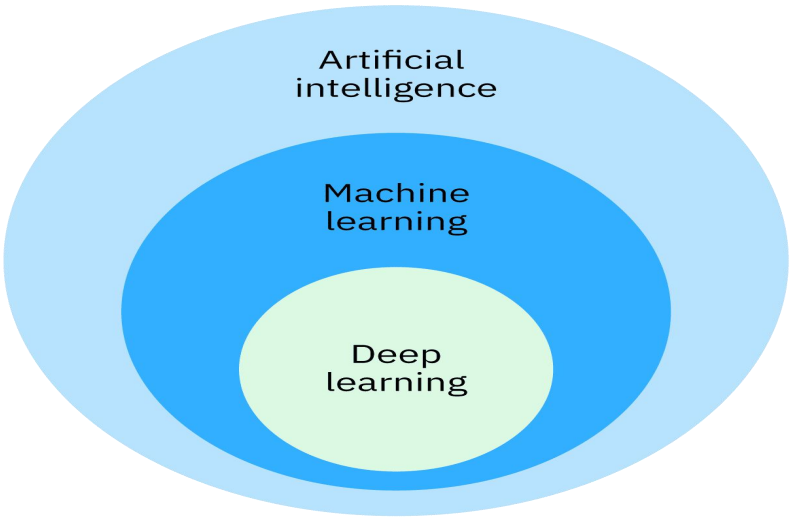
the three levels of predictions that AI can make:

- Narrow
- Broad
- GeneraL

Artificial intelligence (AI) refers to the ability of a machine to learn patterns and make predictions.In its simplest form, artificial intelligence is a field that combines computer science and robust datasets to enable problem-solving.

Augmented intelligence has a modest goal of helping humans with tasks that are not practical to do. For example, "reading" 1000 pages in an hour. In contrast, artificial intelligence has a lofty goal of mimicking human thinking and processes.

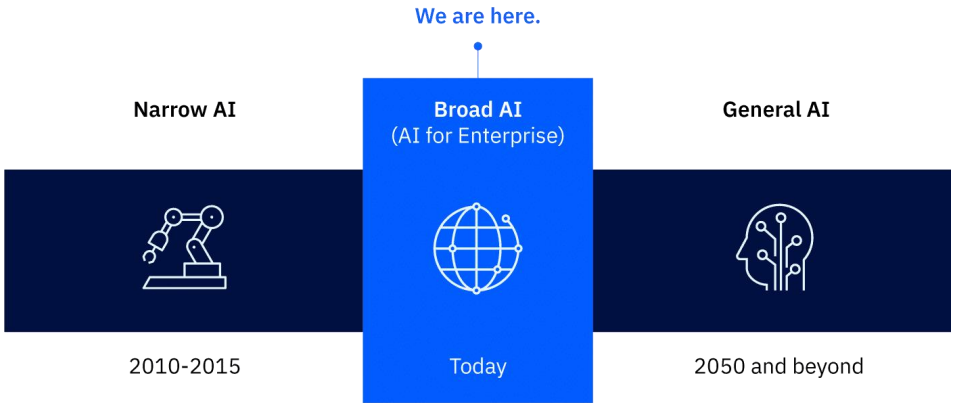


Ai = label

Ml = characteristics (giving more, then no err)

Dp = eliminate the need for feature extractions

Based on data **analysis**, Ai can make **predictions**.



Narrow AI

- Narrow AI is focused on addressing a single task such as predicting your next purchase or planning your day.

Broad AI

- Broad AI is a midpoint between Narrow and General AI.
- Rather than being limited to a single task, Broad AI systems are more versatile and can handle a wider range of related tasks.

General AI

General AI refers to machines that can perform any intellectual task that a human can

ERA OF TABULATION

Dark Data

information can be difficult to extract from a very large amount of data. Because it's hard to see without help, scientists call this **dark data**. It's information without a structure: just a huge, unsorted mess of facts.

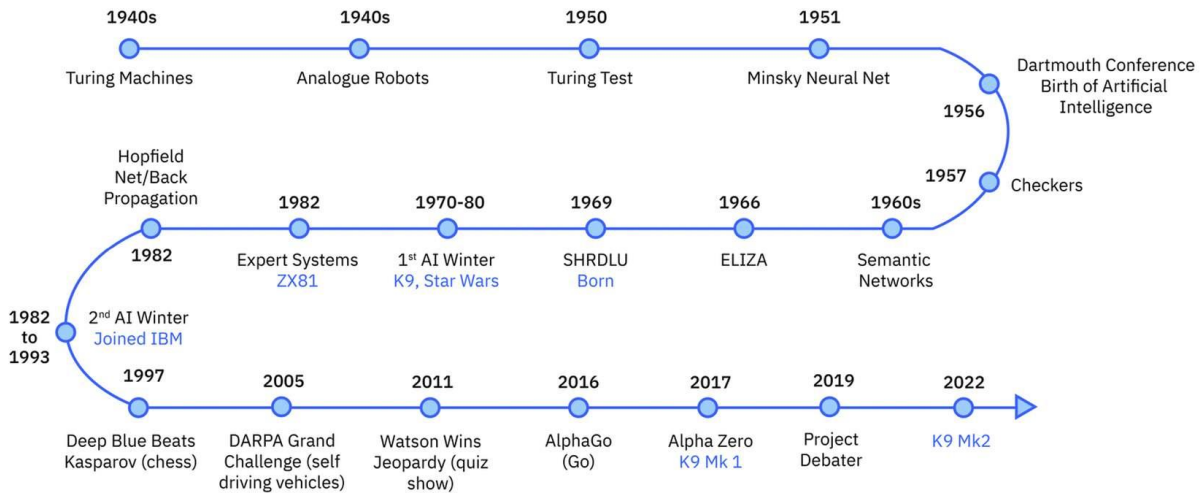
To sort out the these data:
tax collectors for **Emperor Qin Shihuang** used the **abacus**

Charles Babbage and **Ada Lovelace** designed (but never finished) what they called a "**difference engine**" designed to handle **complex calculations using logarithms and trigonometry.**

Researchers call these centuries the **Era of Tabulation**, a time when machines helped humans sort data into structures to reveal its secrets.

ERA OF PROGRAMMING

During the turmoil of World War II **ERA OF PROGRAMMING** emerged. Scientists began building electronic computers, like the Electronic Numerical Integrator and Computer (ENIAC) at the University of Pennsylvania, that could run more than one program. Programmed computer helps astronauts to safely come back to earths.but this era is on crisis.



Data

Data is raw information.

Data can be organized into the following three types.

1) **Structured data** is typically categorized as *quantitative* data and is highly organized. Structured data is information that can be organized in rows and columns. Eg.spreadsheet,hotel and reservation data

2) **Unstructured data**, also known as **dark data**, is typically categorized as *qualitative* data. It cannot be processed and analyzed by conventional data tools and methods. Eg. Medical records,song lyrics,images,text

3)**Semi-structured data** is the "bridge" between structured and unstructured data. It doesn't have a predefined data model. It *combines* features of both structured data and unstructured data. It's more complex than structured data, yet easier to store than

unstructured data. Semi-structured data uses **metadata** to identify specific data characteristics and scale data into records and preset fields. Data that is not completely raw and contains elements such as tags and organizational metadata is known as semi-structured data Eg.video on a social media site, tweets

Experts estimate that about 80% of all the data in today's world is unstructured.unstructured data is rapidly increasing. Recent projections(opens in a new tab) indicate that 95% of businesses prioritize unstructured data management.

Machine Learning

Machine learning is a type of AI with advantages over programmable computers. Machine learning can predict and learn.

Machine learning find answers in unstructured data more quickly than a programmable computer because Machine learning can search every combination of factors very quickly.

Machine learning has analyse dark data more efficiently than programmable computer.

There are two other ways to contrast classical and machine learning systems.

- **Deterministic**

For a **deterministic** system, there must be an enormous, gigantic database of possibilities from which the machine can make its choice.This is basically binary thinking: on or off, yes or no.

- **probabilistic**.

Machine learning is **probabilistic**. It never says "YES" or "NO". Machine learning is analog rather than binary

AI excels at pinpointing patterns.

Capability of AI

endless memory capacity,pattern identifiaction,structuring capability,ml

Capability of Human

abstraction,dreaming,generalization,compassion

Common sense draws generalizations mixed with compassion and abstractions.

Three common Methods of ML

machine learning solves problems in three ways:

- supervised learning

All supervised learning algorithms need **labeled data**. Labeled data is data that is grouped into samples that are tagged with one or more labels

- unsupervised learning

In **unsupervised learning**, a person feeds a machine a large amount of information, asks a question, and then the machine is left to figure out how to answer the question by itself.The algorithm is given unlabeled data.It is helpful when don't know how to classify the data.

- reinforcement learning.

Reinforcement learning is a machine learning model similar to supervised learning, but the algorithm isn't trained using sample data. This model learns as it goes by using trial and error.As a machine learns through trial and error, it tries a **prediction**, then compares it with data in its **corpus**.

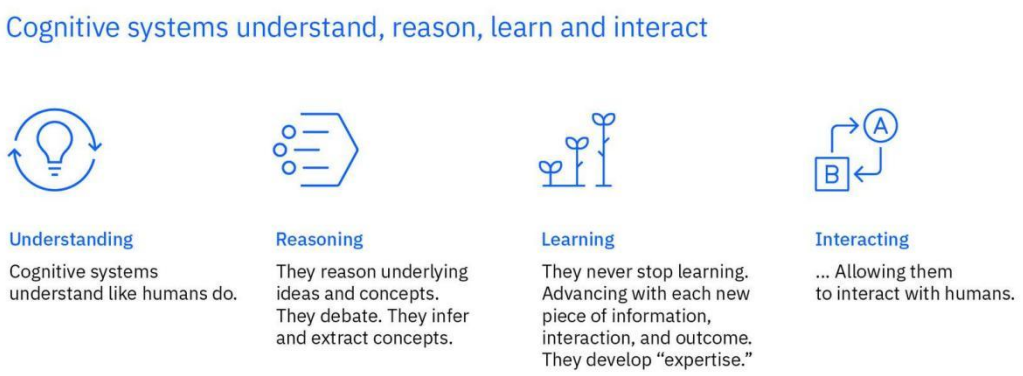
- Each time the comparison is *positive*, the machine receives positive numerical feedback, or a **reward**.
- Each time the comparison is *negative*, the machine receives negative numerical feedback, or a **penalty**.

Natural Language Processing and Computer Vision

Four steps in debate

- Learn and understand the topic,
- Build a position,
- Organize your proof,
- Respond to your opponent.

collection of learned material is called **corpus**.



Human language is incredibly complex, so machines require systems that research scientists call natural language processing (NLP) to understand human language.

To deal with the "messiness" of unstructured information, computers begin with one sentence at a time. This is called **sentence segmentation**. Computers then break the information into small chunks of information, called **tokens**, that can be individually classified. Once the tokens in text have been sorted into a structure based on what they mean, NLP can work with them.

Activity 1: Entities

An **entity** is a noun representing a person, place, or thing.

Activity 2: Relationships

A **relationship** is a group of two or more entities that have a strong connection to one another.

Activity 3: Concepts

A **concept** is something implied in a sentence but not actually stated.

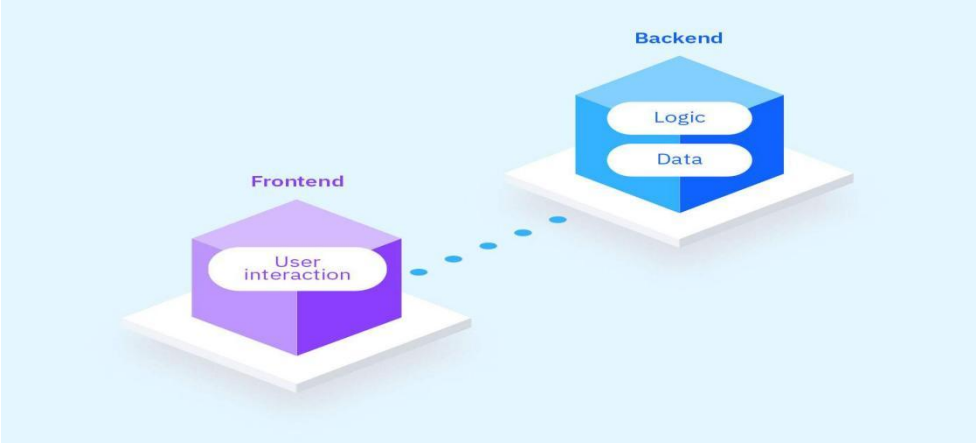
Emotion detection and sentimental analysis are not same thing.

The **Emotion detection** identifies distinct human emotion types.

The **Sentiment analysis** is a measure of the strength of an emotion.

AI system that is working with human language handle the classification problem by **It learns from many instances.**

Chat Bots



Classifiers can map many different ways of asking a question to a very small set of answers. Questions the chatbots can't answer are sent to human customer service representatives.

Chatbot backend includes three parts:

1)Intent:
An **intent** is a purpose: the reason why a user is contacting the chatbot.
A kind of action

2)Entities:
An **entity** is a noun: a person, place, or thing. This concept was covered earlier in this course.

3)Dialog:
A **dialog** is a flowchart—an IF / THEN tree structure that illustrates how a machine will respond to user intents. A

dialog is what the machine replies after a human asks a question

Chatbot software condenses each moment of the conversation into a **node.** A node contains a statement by the chatbot and a long, expandable list of possible replies.

Convolutional Neural Network (CNN)

It is a way to analyze only small parts of an image at a time. This process, called a **convolutional neural network**, or **CNN**, makes it possible for visual recognition systems to identify things in an image, as in facial recognition.

Generative Adversarial network (GAN)

by pitting two convolutional neural networks (CNNs) against each other in a "contest" called a **generative adversarial network**, or **GAN**. In effect, the CNNs battle each other until one of them becomes pretty good at creating art.

In simple form, the GAN have two submodel, generator and discriminator. The generator generates the fake images from the fake input. This inputs are send to the discriminator. The discriminator checks these images and say whether this fake or original. Whenever the discriminator finds that image is fake then the generator have to update with his fake images. On other hand the

generator fools the discriminator then discriminator have to update to identify the fake and original images.
For more information read the page below.

A GAN is an example of unsupervised learning, it effectively supervises itself, and it consists of two sub-models. So, we have a generator sub-model, and we have a discriminator sub-model.

Now, the generator's job is to create fake input or fake samples. And the discriminator's job is to take a given sample and figure out if it is a fake sample or if it's a real sample from the domain. And therein lies the adversarial nature of this. We have a generator creating fake samples and sending them to a discriminator. The discriminator is taking a look at a given sample and figuring out, "Is this a fake sample from the generator? Or is this a real sample from the domain set?"

Now, this sort of scenario is often applied in image generation. There are images all over the internet of generators that have been used to create fake 3D models, fake faces, fake cats, and so forth.So, this really works by the generator iterating through a number of different cycles of creating samples, updating its model and so forth until it can create a sample that is so convincing that it can fool a discriminator and also fool us humans as well.

So, let's take an example of how this works with, let's say, a flower. So, we are going to train a generator to create really convincing fake flowers, and the way that we start by doing this is we need to, first of all, train our discriminator model to recognize what a picture of a flower looks like. So, our domain is lots of pictures of flowers, and we will be feeding this into the discriminator model and telling it to look at all of the attributes that make up those flower images. Take a look at the colors, the shading, the shapes, and so forth. And when our discriminator gets good at recognizing real flowers, then we'll feed in some shapes that are not flowers at all and make sure that it can discriminate those as being not-flowers.

Now, this whole time our generator here was frozen, it wasn't doing anything. But when our discriminator gets good enough at recognizing things from our domain, then we apply our generator to start creating fake versions of those things. So, a generator is going to take a random input vector, and it is going to use that to create its own fake flower.

Now, this fake flower image is sent to the discriminator, and now the discriminator has a decision to make: is that image of a flower the real thing from the domain, or is it a fake from the generator?

Now, the answer is revealed to both the generator and the discriminator. The flower was fake and based upon that, the generator and discriminator will change their behavior. This is a zero-sum game, there's always a winner and a loser. The winner gets to remain blissfully unchanged. Their model doesn't change at all, whereas the loser has to update their model.

So if the discriminator successfully spotted that this flower was a fake image, the lead discriminator remains unchanged. But the generator will need to change its model to generate better fakes.Whereas if

the reverse is true and the generator is creating something that fools the discriminator, the discriminator model will need to be updated itself in order to better be able to tell where we have a fake sample coming in, so it's fooled less easily.

Machine Learning and Deep Learning

Supervised learning requires that an AI system ingest structured data. Unsupervised learning and reinforcement learning require a system to develop its own structure either by analyzing large amounts of data (unsupervised learning) or by trial-and-error (reinforcement learning).

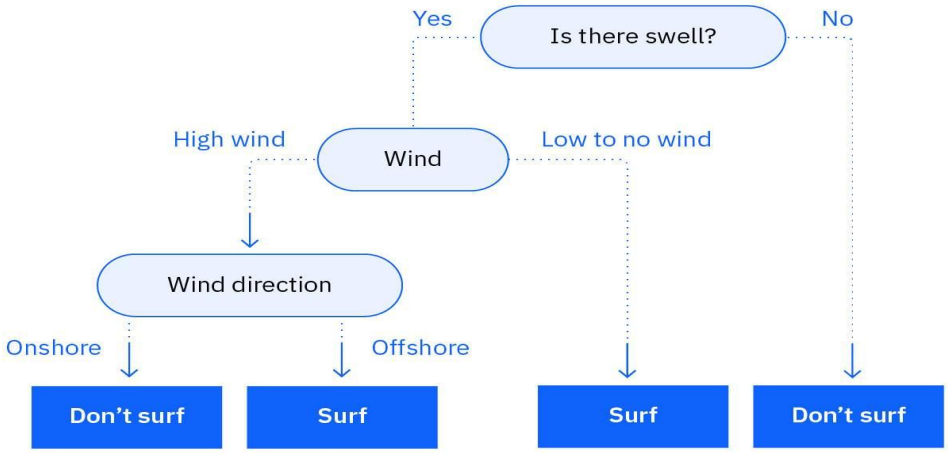
two different technologies by which machine learning takes place: **classical machine learning** and members of a group of technologies called the **deep learning ecosystem**.

Classical Machine Learning

Here are three typical algorithms that are used in classical computing:

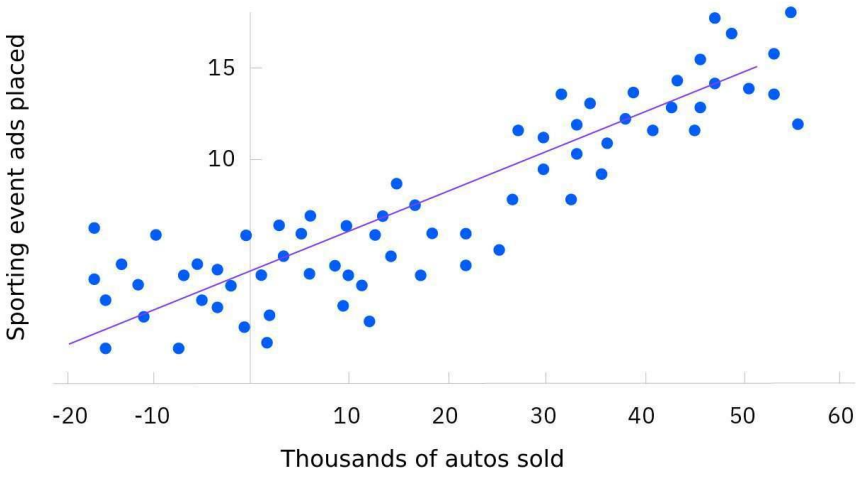
- **Decision tree**

A **decision tree** is a supervised learning algorithm. It operates like a flowchart. You can think of a flowchart as an upside-down decision tree. The flowchart has a **root node** (where the flowchart begins), branches that connect to **internal nodes**, and more branches that connect to **leaf nodes**.



- **Linear regression**

It relates to data that might be graphed as a straight line.



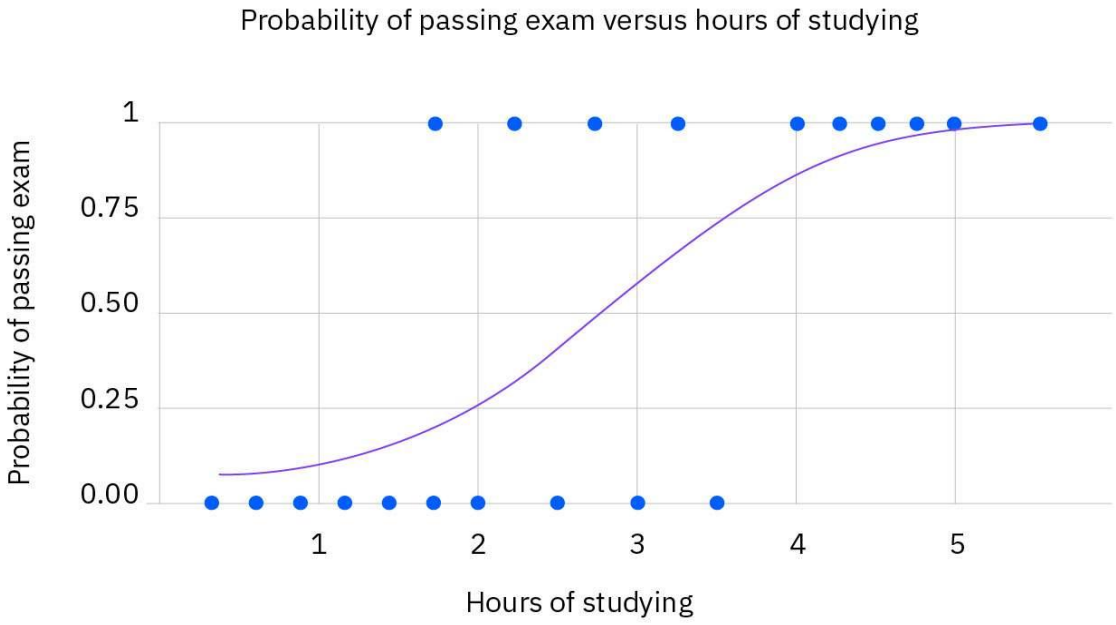
linear regression can learn all the variables, then calculate a reasonably accurate prediction. From above graph it predict how advertising will impact sales at some time and location in the future. In effect, linear regression resolves the mass of dots into a "most likely"

line that can be used for simple prediction.In simple A linear regression answers a question as "if this increases by x,how much will y increase?"

- **Logistic regression**

A **logistic** regression answers a question such as "If this increases by X, will the value of Y be closer to 0 or 1?"

In this situation, a graph can form what's called a **sigmoid function**, or an S-shaped curve, as shown in the accompanying example.



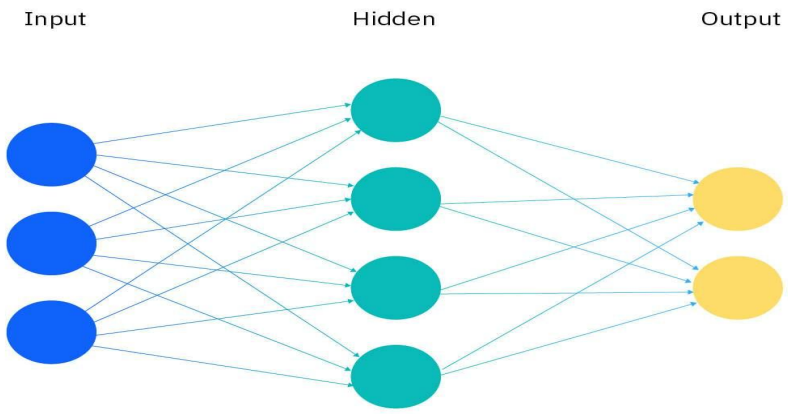
<u>Reasons to still use classical machine learning.</u>

- Work with structured data
- Lower expense to operate
- Easier to interpret

Deep learning ecosystem

In a neural network **(inspired from human brain cells)**, a building block, called a **perceptron**, A perceptron has an **input** layer, one or more **hidden layers**, and an **output** layer.

A signal enters the input layer and the hidden layers run algorithms on the signal. Then, the result is passed to the output layer.



The hidden layers contain **nodes**. Each node runs an algorithm and bits of additional code to test and adjust its result. When the value reaches a certain threshold, the node "fires". A node often uses a sigmoid function to determine whether or not to "fire".

How the neural networks learns

Once a neural network ingested, it already learned a certain amount of data, it stores the data in its "body of information", called its **corpus**. In order to learn, the neural network constantly tests new data or the results

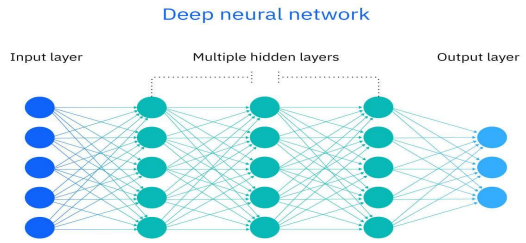
of its calculation against its corpus. If the network determines that the new data or results don't match the patterns it has already established, it modifies those patterns for a better fit. Sometimes, to improve a single match, the network tests hundreds or thousands of modifications very rapidly and makes adjustments. Then, the network tests to determine if the match is improving. So, step by step, the machine learns.

Why Many AI systems output a confidence value along with an prediction ?

Because the ML make a lot of calculation to make a guess to be right. It randomly makes a first guess and set as variable and test how accurate it with both old and new data. Then it make adjustments.

Deep Neural Network

The Advanced AI systems use many hidden layers whose algorithms pass the results of sophisticated calculations. This is a **deep neural network (DNN)**. DNN layers can be arranged in groups or elaborate blocks of groups for greater power. DNNs learn from each other's mistakes, without human intervention.



Generative AI

Generative AI is a type of artificial intelligence that creates new, original content that people have never seen before.

Generative AI models are a type of deep learning AI system that uses algorithms to generate content based on a submitted prompt

Overall Generative AI process

1) a person feeds the AI a large amount of data. This could be anything from images and sounds to text and numbers. It analyzes this data to identify patterns and relationships.
2) The AI neural network is trained on examples similar to desired output.Though this training, the network learns to recognize patterns and relationship in the data.
3) Using random seed value,the AI generates new output based on the learned patterns.

Types of Generative Ai models

- **Variational autoencoder (VAE)**

  The "encoder" network compresses the input data into a lower-dimensional representation and the "decoder" network reconstructs the original data from this compressed representation. This allows VAEs to capture

the underlying structure and patterns in the data, which can then generate new, similar data.

- **Generative adversarial network (GAN)**

In GANs, the generator creates new data, while the discriminator evaluates the quality of the generated data. The generator tries to create data that is realistic enough to fool the discriminator, while the discriminator learns to better distinguish between real and generated data. This competition leads to the generator creating increasingly realistic content

- **Autoregressive**

Autoregressive models generate new content by predicting the next element in a sequence based on the previous elements.

<u>Applications of generative AI</u>

1) Chatgpt

2) IBM wastson discovery – foundational technologies, such as large language models (LLMs),

3) Dall-E and Dall-E 2

4) Bard – Bard is founded on Google's Bidirectional Encoder Representations from Transformers (BERT) model.

Limitations Of generative AI = lack of originality, bias, incompleteness, computational resources.

Run AI Models with IBM Watson Studio

machine learning algorithms.

A **machine learning algorithm** is a set of program code. In a machine learning algorithm, that analysis often has a specific goal: to recognize patterns in data sets.

machine learning models

A **machine learning model** is a group of machine learning algorithms. Operating together, they detect patterns among their algorithms' output and use those patterns to make predictions

IBM Watson Studio features:

- A collaborative data science and machine learning environment
- Easy visualizations with drag-and-drop code
- An efficient workflow
- A built-in neural network modeler
- Open-source tools such as Jupyter Notebooks and RStudio
- IBM Watson Studio has a feature called AutoAI that prepares raw data for machine learning.

STEPS

1. After you log into IBM Cloud, you'll find the Dashboard. This Dashboard lets you access the
tools, services, resources and more in your IBM Cloud account. Select **Catalog** on the top
toolbar.

The IBM Cloud dashboard

With several bank IT workers looking on, you log into IBM Cloud. As it comes up, you
explain that you can access Watson Studio in just a few steps through an IBM Cloud
account. This acts as your doorway to amazing AI resources.

2. This is the catalog of IBM products and services. It is a best practice to create database storage
first. Type "object storage" in the **Search** field and select **Enter**

3. Select **Object Storage**.

4. For this simulation, you are using a free account, so you don't need to review a pricing plan.
Select the **Next arrow** to continue.

5. **Scroll down** to the **Configure your resource**section. Developers would say that you're going to

"provision" the object storage. A **Servicename**is automatically generated, but you should name it something meaningful to your project. Type "Cloud Object Storage-Risk_Fraud" in the **Service name**field and select **Enter**.

6. You don't need to fill out the other fields. In the lower right select **Create**.

7. Now that you've provisioned the Cloud Object storage, select the **sandwich icon** at the top left to display the navigation menu.

8.Select **Resource list**from the navigation menu.

9. This displays a list of the resources that you have created. Select **Storage**to see that the Cloud Object Storage_Risk_Fraud that you just created is listed.

10. Select **Catalog** on the top toolbar.

11. Notice on the left side under **Category** that there is a large collection of available tools. Select the **AI / Machine Learning** category. **An AI / machine learning service** Now you're through the IBM doorway. You've finished basic bookkeeping and you're

ready set up a project in IBM Watson Studio. Some of the IT folks haven't seen anything like this before, so more of them crowd around to watch you get ready.

12. Notice there are several AI and machine learning products listed. Find and select the **Watson Studio**block.

13. For this simulation, you are using a free account, so you don't need to review a pricing plan. Note that the location selected is **Dallas (us-south). Scroll down** to reveal the **Configure your resource** section, thenselect the **Next arrow** to continue.

14. Under the **Configure your resource** section, the **Service name** is automatically generated, in this case Watson Studio-dl. Type "Watson Studio-AI Fundamentals" in the **Service name** field and select **Enter**.

15. You don't need to fill out the other fields. In the lower right, assume that you've checked that you have read and agree to the license agreements, then select **Create**.

16. You've provisioned the IBM Watson Studio service! Notice the service name you picked displays

in the upper left. Now, you need to create a new project. Select **Launch in IBM Cloud Pak for Data**.

17. On the **Build and manage ML models**pop-up, **Provision Watson Machine Learning** is select by default. Select**Next. A new AI project** With your service set up, you'll create the AI project itself. This involves launching it, naming it, and associating it with the database that you'll use to test your machine learning models.

18. Make sure that **Dallas** displays in the **Select a region** field. Since **Dallas** was selected for Watson Studio, you must make sure that you select **Dallas** for all the other services. Select the **Next arrow** to continue.

19. Under the **Configure your resource** section, name your service. Type "Machine Learning Risk_Fraud" in the **Service name** field. Then select **Create**.

20. You are working on a new project, so select **New project**to create an empty project and then select **Next**.

21. Under the **Define details** section, you can name your project. Type "Risk_Fraud" in the **Name** field. Select the **Next arrow** to continue.

22. Notice under **Storage**that IBM Watson Studio already provisioned the Cloud Object Storage (COS) that you created earlier so you have a database. This COS can store unstructured data like images and text. Soon, you'll work with structured data in a comma-separated values (CSV) file. Select **Create**.

23. You've created the project! This is the IBM Watson Studio Projects dashboard. You can see your project name at the top. This is the **Overview** tab. Select the **Assets**tab.

24. It's time to upload your data set. Select **Drop data files here** to open your local drive and find the german_credit_data_biased_training.csv file that you'll be working with for the project.

25. Drag and drop the **german_credit_data_biased_training.csv** file to **Drop data files or browse for files to upload** under **Data in this project**.

26. Notice your CSV file is right there for you. Select "**german_credit_data_biased_training.csv**".

27. A **Preview assets** page opens so you can now preview your data. Take a moment to check out
the data that displays. Select the **Next arrow** to continue.

Create and run AI models

1. Now that you've set up your data set in cloud storage, it is time to build the AI models. Select **New asset**to start creating your AI model. **AI models**
Creating artificial intelligence models used to be an incredibly complex and difficult task. But today you can show off to your colleagues how IBM Watson Studio does this for you—automatically!
2. On the left side, there is a **Tool type**list. Select the **Automated builders** tool. **AutoAI** is the only
Automated builder displayed. With AutoAI, you will be able to quickly set up and run AI models
using your data to train and test the models. Select **AutoAI**.
3. Creating and testing AI models is called experimenting. Now it's time to create your AutoAI
experiment. Type "Loan Risk" in the **Name**field to name your model and press **Enter**.
4. In the **Define configuration**section, you'll see a machine learning service isn't associated with your
project yet. Let's take care of that now. Select the **Associate a Machine Learning service instance**
link.
5. On the **Associate service** page,you'll see all the services you've created that can be associated with
your experiment. Right now, there's only one service displayed: **Machine Learning-Risk_Fraud**.

Select the checkbox next to the service and then select **Associate**.

6. Now that you've associated your machine learning service, select **Reload** to refresh the page.

7. You can see Machine Learning-Risk_Fraud displayed in the **Watson Machine Learning Service Instance** field now. You're ready to create your AutoAI experiment. Select **Create**.

8. It's time to add your data source. Select **Select from project**to find the data source you added previously.

Your experiment

You promised the bank you'd run several different machine learning algorithms competitively.

You'd train them all on the same partial set of **historic data**about people who took out loans and then paid them back or defaulted on them. Then you'd test how well they performed, based on what they'd learned, by feeding them identical sets of **new data**, without telling them who defaulted. Each algorithm would make predictions about everyone in the new data set, classifying people as **Risk** or **NoRisk**.

9. On the left, you'll see **Categories**listed. Select **Data asset**to reveal your data assets.

10. You can see your data asset. Select the check box next to: **german_credit_data_training.csv**.

11. On the right, information about the asset is displayed including: name, asset type, size, and when it was modified and created. Select **Select asset**.

12. Now it's time to configure the details for your experiment. The first question that displays asks if you want to create a time series forecast. For this experiment, you don't want to do that, so select **No**.

13. The next question that displays asks what you want to predict. This is asking what column from your data set you want the AI model to predict. Select **Select prediction columns** to open the drop-down list.

14. Next to **Risk** are the letters "STR". This indicates that the data type in the **Risk** column is a String. Strings are a sequence of letters, digits, punctuation, and so on. In this case, it's text data. Since you're trying to *predict which individuals are good risks for loans*, select **Risk.** 15. The **PREDICTION TYPE**that Watson AutoAI selected is Binary Classification. You're trying to determine whether an individual presents a **Risk** or **No Risk**, which is a classification with only two options, so leave the prediction type as **Binary Classification**. Select the **Next arrow** to continue.

16. By default, **Accuracy & run time**is selected in the **OPTIMIZED FOR** field. This means when Watson is evaluating which algorithmis best it will optimize by balancing accuracy and speed. Select

Experiment settingsto make further changes.

17. This is the **General**tab. It shows what youselected on the previous page. **Binary classification**is the prediction type, **No Risk** is the positive class, and **Accuracy**is the optimize metric. Select the **Next arrow** to continue.

Competitive algorithms

Some folks watching you had expected you to create mathematical algorithms that could analyze data and predict credit risk. It's time to astonish them. How? By using Watson Studios to line them up quickly.

18. Scroll down to view the **Algorithms to include** section, notice **GradientBoosting Classifier** is not selected. Select the check box next to **Gradient Boosting Classifier**.

19. Scroll down to view the **Algorithms to use** section. Watson AutoAI lets you test up to four algorithms. Select the **Next arrow** to continue.

20. By default, Watson will select the best two algorithms to test on the data. Select **Datasource**from the left menu.

21. Scroll down to view the **Training and holdout method** section.

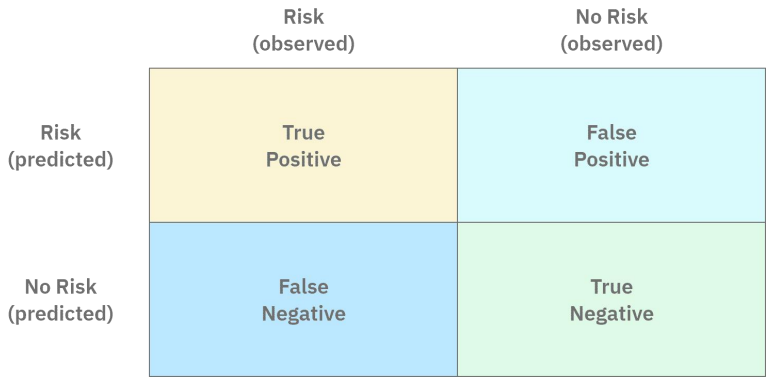
The **Trainingdatasplit** slider lets you choose how much of the data set to use for training and how much to use for testing. By default, 90% of the data set is used for training and

10% of the data set is used to test how the AI model is performing. Hover over the
underlined words to see more about each field in this section.
22. Below the Training data split is the **Select features to include** section. Here, you can select which
columns (or features) in the data set to use for training and testing. There are three pages of data.
Select the **forward arrow** to page through the columns in the data set.
23. All the features in the data set are available except **Risk**because that's what is being tested. By
default, all the features are selected. Select the check box next to **Telephone** to deselect it and not
use this feature in the experiment.
24. Select **Savesettings** to confirm all your choices.
25. Everything is set up! Select **Runexperiment**to start the experiment.
Launch time!
You're about to launch the experiment. Your models will ingest the historic training data,
then run the new data and make their predictions. Like people watching a horse race,
everyone leans in to see what happens.
26. The Relationship map shows by default. Select the **Swap view** link under **Progress map** to display
the Progress map.

27. Here, the Progress map shows the two models that are being tested. Take a moment to review.
Select the **Next arrow** to continue.
28. Watch as the AutoAI experiment runs. This will only take 45 seconds, but would take roughly 5
minutes when performed in the live environment.
29. Notice that each model has four pipelines (or algorithms)for a total of eight pipelines. The pipeline that'sproducing the most accurate predictions and is the fastest is shown with a star.
Select the pipeline with the star.30. This displays the **Pipeline details**. By default, the ROC curve is shown. Select **Confusionmatrix**
from the **Model viewer** on the left.
31. This displays the Confusion matrix.

Confusion Matrix

The confusion matrix is a visual representation of performance of the AI model. It looks at the **Observed** versus **Predicted** results.



True Positive: *The model predicts positive (Risk), and the result is positive (Risk).*

True Negative: *The model predicts negative (No Risk), and the result is negative (No Risk).*

False Positive: *The model predicts positive (Risk), but the result is negative (No Risk).*

False Negative: *The model predicts negative (No Risk), and the result is positive (Risk).*

Save AI models as Jupyter Notebooks

1. Once you have completed your experiment and determined the best model, your next step is to save
the model. Select **Save as. Models and notebooks**
You'll recall that the experiment ended by showing a confusion matrix. It's a mathematical
test that compares predictions from all of the algorithms to see which one worked best.
Basically, your job is done now. The IT folks who had gathered around begin to disperse—
except for a couple of people who linger so they can see how you'll save and store the
winning model.
Select **X** to close this window and continue. ⓘ
2. You have two options. You can save the model or you can save as a notebook. To use the model

again in Watson Studio you would select Model, but to interact with the model on a deeper level you
need to save it as a notebook. Select **Notebook.**

3. The name shown in the Name field in the **Define details** section will change to the name of the model
and the pipeline that Watson selected with "Notebook" at the end. Select **Create**.

4. Now, select the **View in project** link that displays in the notification box.

5. The **Pipeline notebook** displays, but it is currently view only. You need to be able to edit it. Select
Edit icon from the menu on the top right. This makes it possible to edit the notebook.

6. The notebook in an editable form displays, but it is not trusted. This means that the JavaScript shown
in the notebook will be disabled and not allowed to run. Since you want to be able to run the code,
you need to trust the notebook. Select the **Not Trusted** link in the upper right.

7. This displays the **Trust this notebook?** dialog box. Select **Trust**.

8. The display refreshed and the notebook now shows as trusted and all the JavaScript in the notebook
is enabled. Select the **Next arrow** to continue.

9. Now that your notebook is trusted, it's time to download the notebook. Select **File** from the top menu.

10. Now, select **Download as**. This displays the formats that you can download the notebook in.

11. Select **Notebook (.ipynb)** to download the notebook in Jupyter Notebook format.
12. This will open a new tab and automatically download the notebook in Jupyter Notebook format to your computer. Next, you'll look at what you can do with the downloaded notebook.

AI ETHICS QUICK SUMMARY

Fairness

- In AI, fairness is the equitable treatment of individuals or groups of individuals.
- Fairness is achieved when unwanted bias is mitigated.
- Protected attributes separate populations into groups.
- Groups that traditionally receive more favorable outcomes are called privileged groups.
- Groups that traditionally receive less or no favorable outcomes are called unprivileged groups.
- There isn't a defined set of protected attributes.
- Bias is a systematic error that, intentionally or not, might generate unfair decisions.

Robustness

- A robust AI system can effectively handle exceptional conditions, like abnormalities in input or malicious attacks, without causing unintentional harm.
- Adversarial attacks are intentionally carried out on AI systems to accomplish a malicious end goal by exploiting AI system vulnerabilities.
- Two types of adversarial attacks are poisoning and evasion.

Explainability

- AI systems are explainable when everyday people, who do not have any special training in AI, can understand how and why the system came to a particular prediction or recommendation.
- Interpretability is the degree to which an observer can understand the cause of a decision.
- Explainability looks at how the AI system arrived at the result.

Transparency

- Transparency is disclosing information related to the data used for building AI systems, design decisions made throughout the process, model creation, model evaluation, and model deployment.
- Governance ensures the process followed during the creation and deployment follows the internal policies.

Privacy

- Personal and sensitive personal information can be used to train models, as long as privacy techniques are applied to the data to preserve the privacy of individuals whose data is included.
- Many privacy techniques that can be applied to fortify AI against potential breaches of personal or sensitive data. Two that occur during model training are model anonymization and differential privacy. One that occurs after model training is data minimization.