Describe fundamental principles of machine learning on Azure

What is Machine Learning

Why even consider Machine Learning?

Let's say your company has an ecommerce website
You sell products online to customers.

Over time you want to take the historical data collected based on the purchases made on the site and try to forecast the following

1. If you were to introduce a new product , what is the expected revenue you forsee for the product.

2. If you introduce a new business feature in your application , would it increase the chances of a customer buying a product.

There are several ways to forecast - We could build reports based on previous data and maybe see if we can come to a conclusion based on the above requirements. We can use reporting tools to provide such forecasting capabilities.

Or we could build an application. We would need to add logic via code to the application that would give us the desired results based on the above 2 requirements.

But another way is to build a Machine Learning model that would give us the desired results.

What is Machine Learning model - This is an object that has been trained beforehand on a data set. The model can then give you an output based on input values.

The machine learning object is nothing but a file or probably a set of files. It would have the desired logic that would take in input data and spur out output data.

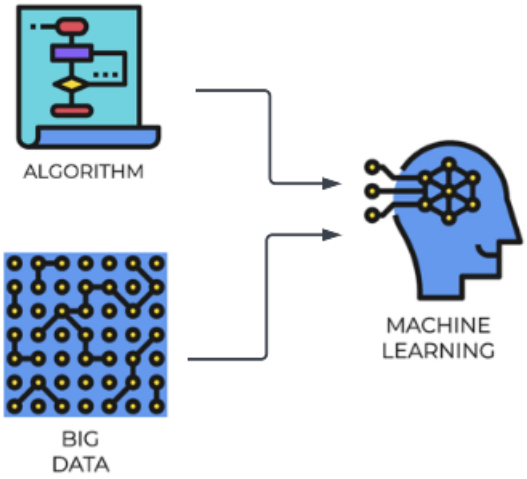
When should you consider machine learning - If the problem or requirement is simple. If you can create a simple script to solve the requirement, then you don't need to use Machine learning.

Machine Learning is ideal to use when you want to carry out predictions based on incoming data, look at trends, make informed decisions based on a large set of data values.

The Machine Learning Process

Now there are many pre-built models that are available online. For example there are models that can be used to detect objects in images. There are models that can be used to covert text to speech.

But you might want to build a model for your own very specific business requirement.



To build a machine learning model, you need to use a machine learning algorithm and an input data set.

What is the data set?

So let's say based on our previous requirement, we want to know what would be the predicted revenue when introducing a new product to the product catalogue for our ecommerce web site.

So first we need to train our machine learning model with existing data. We can have data in the form of rows and columns. This is data that already exists based on the existing customer base and revenue.

This data set could contain information about the revenue of products in different categories over time. It would also learn when products were added to the catalogue in the past, how did they fare over time.

Hence we are feeding it data that we already have. And training it on how to predict future values.

Then we have the machine learning algorithm which is nothing but a set of instructions. These instructions will go through the input data , look at patterns in the data and then develop logic that can be used to output values based on the future input.

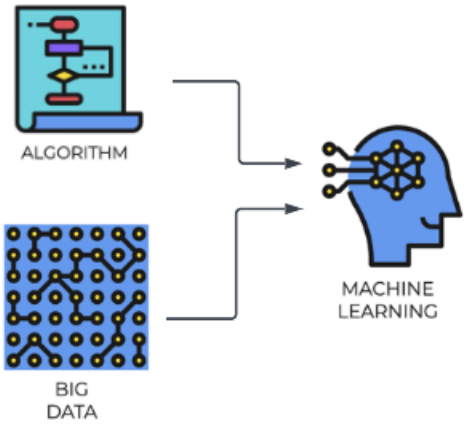
Now the Machine Learning process is an interative process. Once we build the model, we would want to fine tune the model at later points in time.

Also if the future input data changes , we need to train the model again so that it would always give the right output values.

Your data set

The data set

Building your data set to train your Machine Learning model is the important step and takes the most time.



First and foremost you need to decide where your data is going to come in from.

This boils down to your requirement and the problem you are trying to solve.

Then the next step is to prepare the data. Once you have located your data sources, you need to select only the columns that would be of interest in training the Machine Learning model.

House prices over time

Location	Number of rooms	Agent	Area (sq ft)	Price (USD)
LocationA	5	AgentA	10,000	500,000
LocationB	4	AgentB	12,000	550,000

Here for the Machine Learning model , when training it with input data, you might want to leave out the Agent name because it might have little relevance in determining the house price.

Feature - This is a measurable property within the data set.

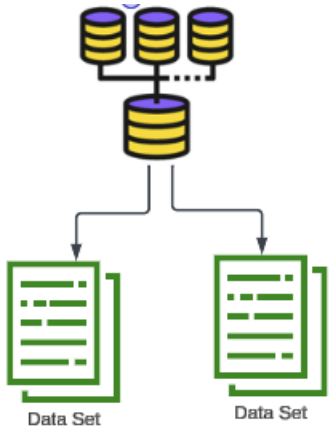
Label - This is used to signify to the Machine Learning model that for a set of input values this was the output.

Also when it comes to your input data you would need to inspect and transform your data so that it is clean and makes sense when training your machine learning model.

Clean data

Missing data

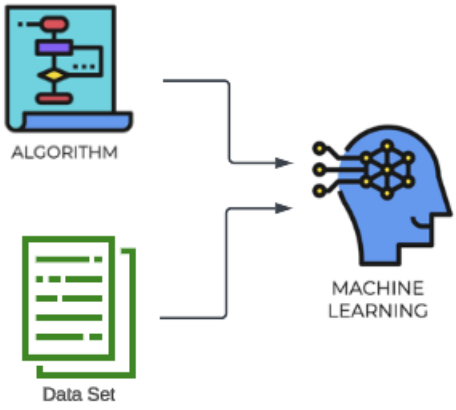
Feature Engineering - Here you can generate new features based on existing one's.



70%

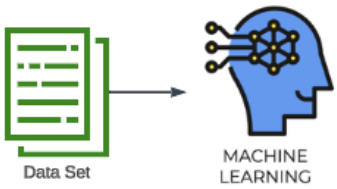
30%

For your input data set you would want to split your data set for the purposes of training and validation.



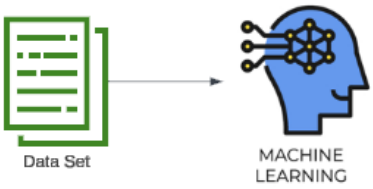
70% Training data

You first use the training data to train the machine learning model.



30% Test data

You then take the 30% test data to test the Machine Learning model. For the 30% test data, you already know the output, hence the Machine Learning model would be fed all of the feature values. And we would ask it to predict the label. And then a comparision is made with the predicted value and the actual label value to see how accurate the Machine Learning model was.



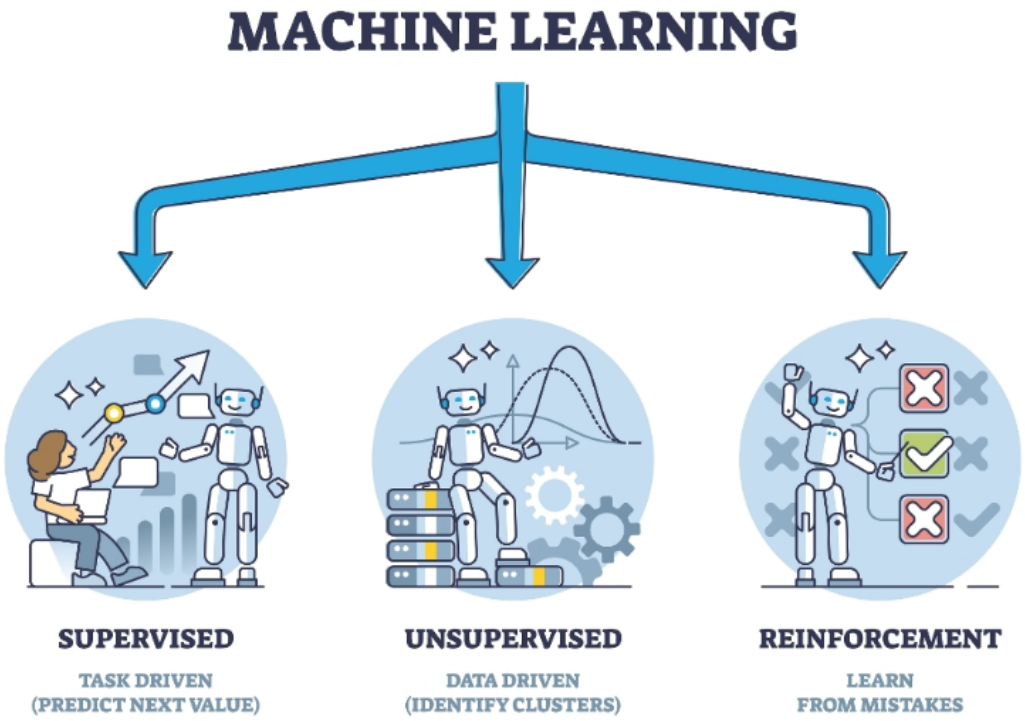
Live data set

Once you have your machine learning model in place, you can then feed in live data and get predictions made accordingly.

Location	Number of rooms	Area (sq ft)
LocationA	5	10,500

The Machine Learning Algorithms

Machine Learning techniques



In supervised learning, for the training data set you feed in the features along with the labels. Here the machine learning model is supervised on the training based on the label data.

In unsupervised learning, we don't know what the output will be , so we cannot provide labels in training data. Here we just want the model to split our data into different groupings or clusters.

Based on the different techniques , we have different Machine Learning algorithms in place.

Classification Algorithm

This is simply used to classify your data. For example, will the customer purchase the new product - Simple Yes or No answer.

We also have multi-classification algorithms that can classify into different categories. Predict the best three categories of products that would be sold during a particular time period.

Regresion algorithms

This is used to predict values of new data points based on historical data. For example, calculate the forecasted revenue of a new product over the coming week based on historical data.

Anomaly detection algorithms

This is used to detect if data deviates from the norm. For example this can be used to detect fradulent credit card purchases.

Clustering algorithms

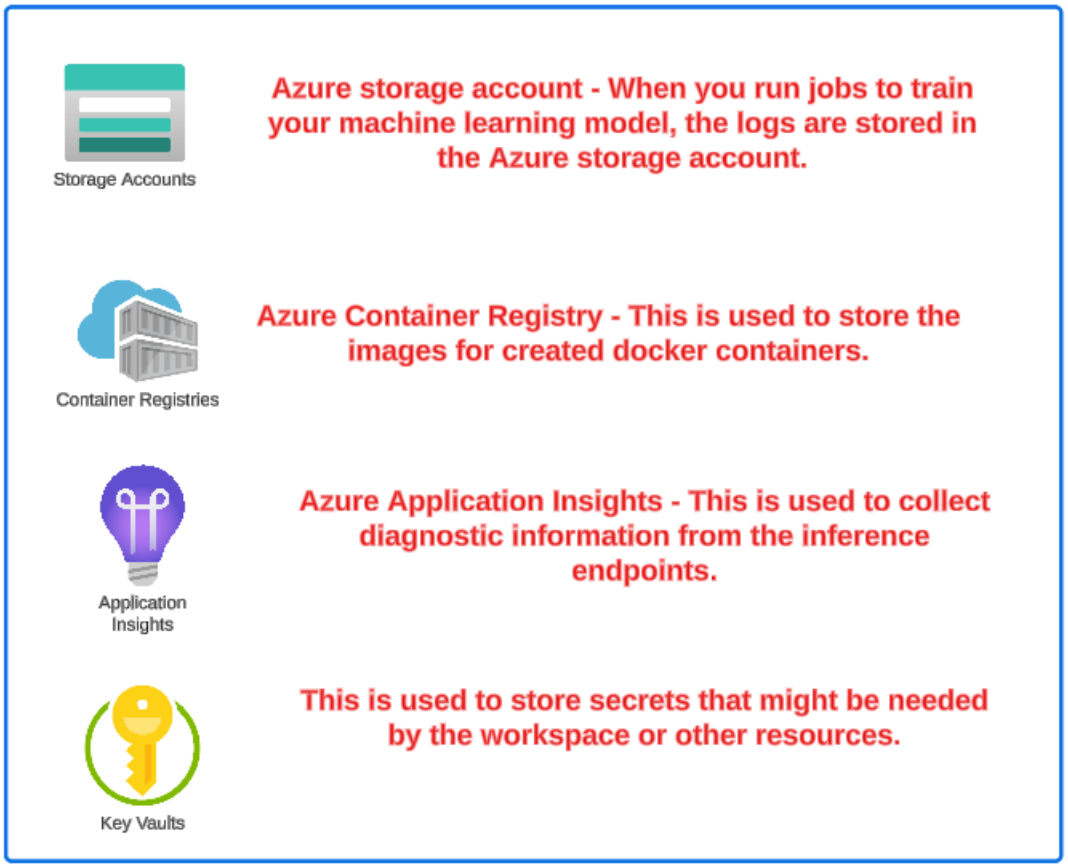
This can be used to seggregate data points into different categories or clusters.

Lab - Azure Machine Learning - Creating a workspace

The Azure Machine Learning workspace

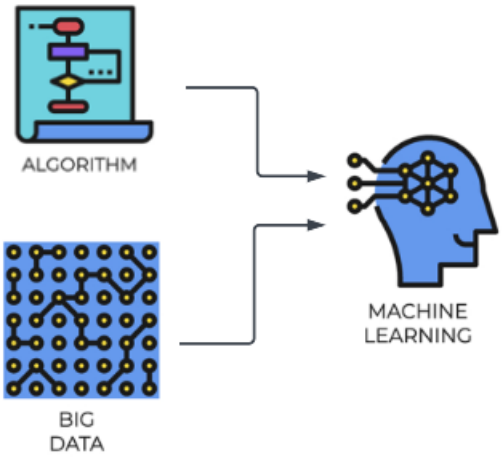
This is an encapsulation of your machine learning artifacts that includes your data sets, your machine learning models etc.

When you deploy an Azure Machine Learning workspace , there are some Azure resources that get deployed along with it.

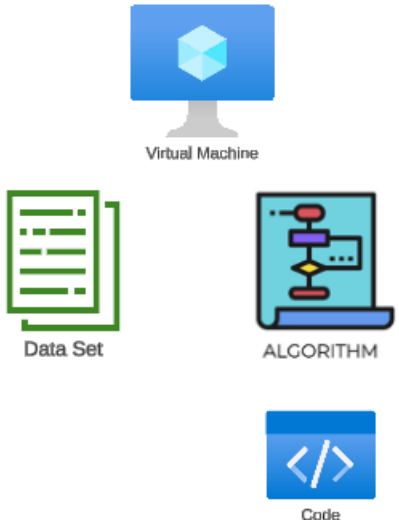


Lab - Binary Classification Model - Compute Instance

Compute instance to run our pipeline



When you want to start training your machine learning model with training data and the machine learning algorithm, you need to have compute power in place.



Here remember that the algorithm is a set of instructions like code. These instructions need to go through the large data sets to look at patterns in data. Hence we need a compute machine to run the instructions and build our own machine learning model.

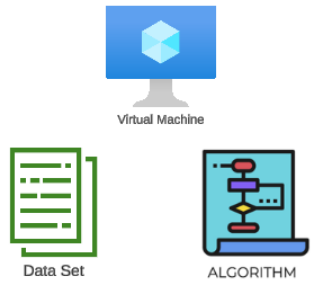
We can build a compute instance to run as part of our machine learning workspace to run the pipeline to build our machine learning model.

You have support for different target environments

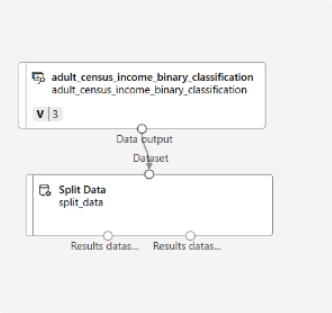
Training targets	Automated machine learning	Machine learning pipelines	Azure Machine Learning designer
Local computer	Yes		
Azure Machine Learning compute cluster	Yes	Yes	Yes
Azure Machine Learning serverless compute	Yes	Yes	Yes
Azure Machine Learning compute instance	Yes (through SDK)	Yes	Yes
Azure Machine Learning Kubernetes		Yes	Yes
Remote VM	Yes	Yes	
Apache Spark pools (preview)	Yes (SDK local mode only)	Yes	
Azure Databricks	Yes (SDK local mode only)	Yes	
Azure Data Lake Analytics		Yes	
Azure HDInsight		Yes	
Azure Batch		Yes	

Reference - https://learn.microsoft.com/en-us/azure/machine-learning/concept-compute-target?view=azureml-api-2

Lab - Binary Classification Model - Running the pipeline



Let's run our machine learning pipeline once we have the compute instance in place. Here we are not creating a machine learning model as of yet. We just want to see how a pipeline run works.



For us ,we are using a no-code experience via the ML designer to create the pipeline.

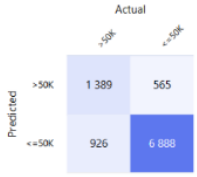
But in the end , the Machine Learning service on Azure will run code in the background based on the components in the piepline.

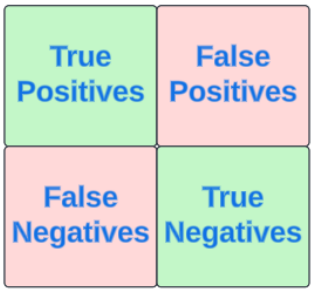
Lab - Building a Classification Machine Learning Pipeline – Results

Two-class Logistic regression

Confusion Matrix

This table gives us a summary of the results when it comes to what the model was supposed to predict and the actual prediction.





Accuracy - This tells how often the classifier is right in predicting results.

Accuracy	0.847
Precision	0.711
Recall	0.6
F1 Score	0.651
AUC	0.901

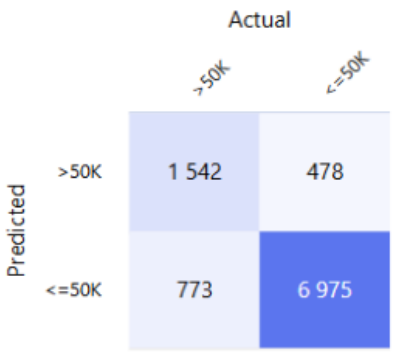
$$\frac{\text{True Positives + True Negatives}}{\text{True Positives + True Negatives + False Positives + False Negatives}}$$

Precision - This tells to what extend does the model accurately predict results.

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

Two-class boosted decision tree

Accuracy	0.872
Precision	0.763
Recall	0.666
F1 Score	0.711
AUC	0.927



Lab - Building a Classification Machine Learning Pipeline – Deployment

Now we have built our machine learning model. Let's say that we are happy with the accuracy and precision of the model.

We now want users to be able to submit requests to the model, specify input values and get a result back when it comes to the income.

Now we could embed the machine learning model within an application. So users could submit requests via the application, specify the input data and then get the results from the machine learning model.

One thing we need to understand is that the machine learning model needs to run on compute infrastructure. This is because the model needs to take in the input data , perform the required processing and give the output.

We also need to have an endpoint in place for the model. This is so that we can call the endpoint and hence make a call to the model.

From a successful run job, we will now create a real-time inference pipeline.

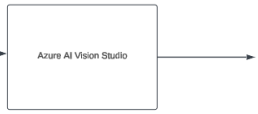
Once we have our real time inference endpoint in place, we can deploy this to a compute setup. Then we can call the endpoint.

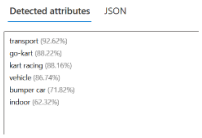
We wil create a new compute environment in Azure ML, which is a Kubernetes compute setup. The Machine Learning model would be then be hosted on this setup.

Describe features of computer vision workloads on Azure

Just a recap of what's happening in the background

Azure Vision Studio will submit the image to a machine learning model that has already been trained.





We are submitting our image to Azure Vision Studio. We are telling Azure Vision Studio to extract the common tags from the image.

You then get the relevant tags as the output

How would this work from an application

The Vision service will then give the desired output back, this might come in a format such as JSON.

From your application code, you will make an API call. Service on Azure allow you to make calls to the various functions available for a service.

When you make the call to the Azure AI service, you specify the different parameters in the call. And you can supply input if required like images.

Programing languages like .NET also have libraries in place that you can use to facilitate the calls to AI services on Azure.

Describe features of generative AI workloads on Azure

Generative AI

Generative Artificial Intelligence

We have seen how we can leverage AI-based services on Azure to identify objects, classify objects, detect faces, translate text etc.

In the backend you will have a trained machine learning model that could provide all of this functionality.

Generative AI is where the model can generate or produce output such as text, images, videos etc.

You can build machine learning models that can predict future values based on historical data. Or use the input to detect certain aspects such as object detection etc.

But Gen AI is in a different league altogether.

Here I am asking ChatGPT for advise, probably similar to asking a mentor on what I should do next.

It's courteous like a normal human being, telling us that we are moving in the right direction.

I am currently learning for the AZ-900 exam. My aim is to build my portfolio with cloud-based services. Can you suggest me based on the market trends , what should be next in my learning path.

Memory updated

That's a great step forward! The AZ-900 exam is an excellent foundation for understanding Azure's core concepts. Based on current market trends, here's a suggested learning path to complement your cloud knowledge and expand your portfolio:

And then its giving me a run down on the next steps.

Next Steps After AZ-900:

1. **Specialized Azure Certifications**:
   - **Azure Administrator (AZ-104)**: Dive into managing cloud resources, storage, virtual networks, and security.
   - **Azure Developer (AZ-204)**: Ideal for your .NET background, focusing on developing and deploying Azure applications.
   - **Azure Solutions Architect (AZ-305)**: For designing advanced cloud solutions.
2. **DevOps Practices**:
   - **Azure DevOps Engineer (AZ-400)**: Learn CI/CD pipelines, infrastructure as code, and monitoring, which are highly in demand.
3. **AI and Data Specializations**:
   - **AI Engineer (AI-102)**: If you're interested in building AI solutions.
   - **Data Engineer (DP-203)**: For working with big data and analytics pipelines.

Skills to Add to Your Portfolio:

1. **Infrastructure as Code (IaC)**:
   - Tools: **Terraform**, **Bicep**, and **Azure Resource Manager (ARM) Templates**.

This is all based on GenAI wherein it is generating text on the fly based on our input request.

Large Language Models

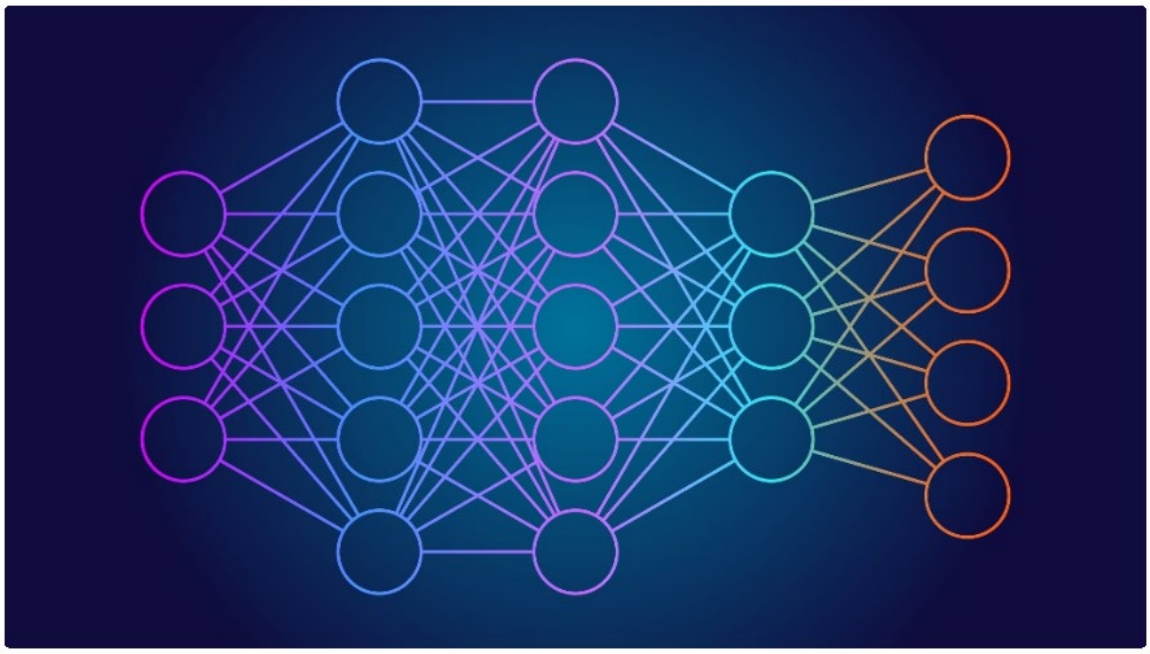
Large Language Models

This is a type of machine learning model which is specifically designed for tasks which involve natural language processing.

Here the model is trained on large amounts of data and are normally used to generate text.

We then have GPT - Generative pre-trained transformer.
This is a type of large-language model.

It is based on an artificial neural network.



This is a group of interconnected units called neurons.

It mimics the way our brain works. These neural networks are used to build artificial intelligence and make it behave similar to how we think and respond.

Its comprised of an input layer, hidden layers and an output layer. Each node or neuron in the network can use a simple or complex mathematical function to take the input and give the output. And the output of one neuron can then be given to the subsequent neuron.

ChatGPT is a generative artificial intelligence chatbot that was developed by a company known as OpenAI. Currently it is based on the GPT-4 large language model. The GPT , the transformer is what OpenAI developed to power ChatGPT.

Here the Chat part is just a friendly interface that allows us to interact with the Large Language model.

Now we have deployed simple machine learning models in our previous section. Can we build a large language model such as ChatGPT.

So the GPT models built by OpenAI has been trained on a large corpus of data - Think of the entire public data available on the Internet.

This large amount of data needs to be used in the training process. Then the outputs need to be validated. This is because , you take data from the Internet , the data can be wrong, have wrong intent or contain harmful content.

So you need to cater to all of this and retrain the model accordingly.

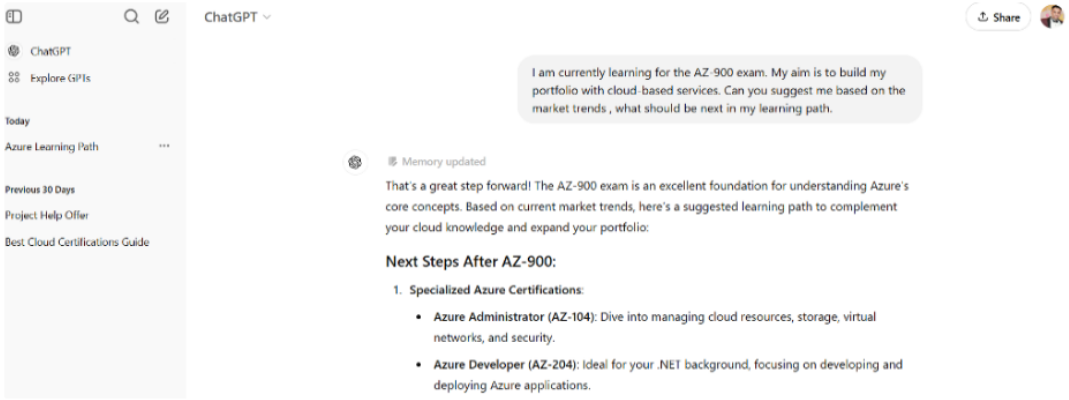
The initial version of GPT, GPT-1 was built on a parameter count of 117 million. And it was not disclosed to the public for fear for misuse.

The current version of GPT , GPT-4 is estimated to use around 1.7 trillion parameters.

You need a tremendous amount of infrastructure to train and host the model. It runs into million and millions of dollars.

You need special GPU's that can handle the processing of data. You need a large number of the GPU's and its takes days and days to train the model.

An initial look at using ChatGPT



We can start with the free version of ChatGPT.

What are some of the limitations of the free version

If you send it too many questions with a define time frame, it might bump you down to a lower GPT model which would not be trained on all current information.

For example , GPT 3.5 is trained with data until Sep 2021 and GPT 4 is trained with data till around Oct 2023.

With the paid subscription you will always get access to the GPT model. You also get additional features such as DALL-E which is an image generation tool. You also get access to data analysis features as well.

How does the pricing work with ChatGPT?

Model	Pricing	Pricing with Batch API*
gpt-4o	$2.50 / 1M input tokens	$1.25 / 1M input tokens
	$1.25 / 1M cached** input tokens	
	$10.00 / 1M output tokens	$5.00 / 1M output tokens
gpt-4o-2024-11-20	$2.50 / 1M input tokens	$1.25 / 1M input tokens
	$1.25 / 1M cached** input tokens	
	$10.00 / 1M output tokens	$5.00 / 1M output tokens
gpt-4o-2024-08-06	$2.50 / 1M input tokens	$1.25 / 1M input tokens
	$1.25 / 1M cached** input tokens	
	$10.00 / 1M output tokens	$5.00 / 1M output tokens
gpt-4o-audio-preview	Text $2.50 / 1M input tokens	
	$10.00 / 1M output tokens	

Your text data is broken into smaller units called tokens. These tokens are assigned numerical values and then fed into the large language model.

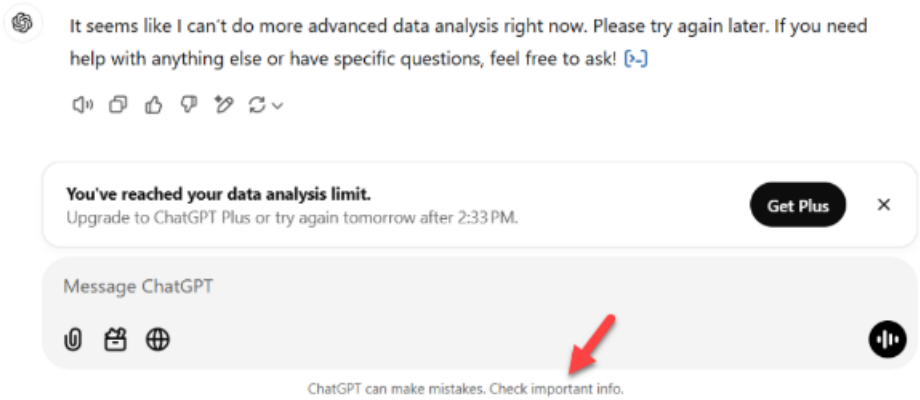
Tokens are considerd both for the input and the output. That gives rise to something known as prompt engineering. How can you effectively send prompts to ChatGPT. Because when conversing in your chat, when you make a subsequent request, the prior output data is also sent again to GPT to get the next output segment in your chat.

With GPT - What's amazing is that, all it does is generate the next word when it comes to generating output. After is generates a word, it looks at predicting the next word.

Alignment with Responsible AI

We have seen how powerful ChatGPT - Let's categorize this as Gen AI.

Nowadays a lot of individuals put a lot of dependency on the results generated by ChatGPT.



Next the GPT model was trained a lot of publicly available data. There is a lot of debate on whether it was right to take in all of the data.

What about data that its not trained on. If we ask it data about a recent event its not trained on, it should not propose a wrong output. This is also known as hallucination - This is wherein it tries to give us plausible answers that are incorrect.

With the data on the Internet, you can have data that biased in nature. Hence the model can also be biased. And this goes against the general principle of responsilble AI.

Also ChatGPT has been trained to reject certain prompts that could violate certain policies. But users have found workarounds wherein it can ask ChatGPT with prompts that can violate this policy.