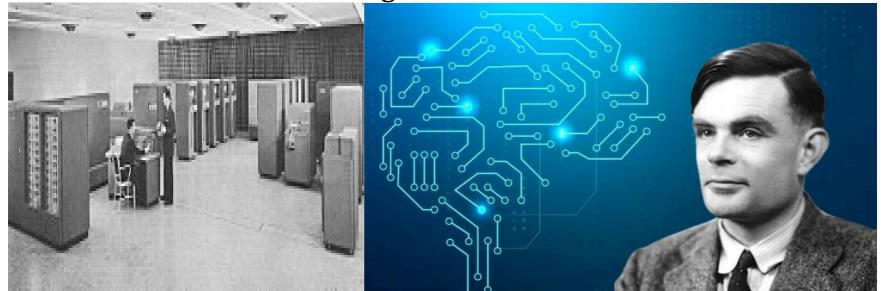
# Artificial Intelligence

Going back to 1930



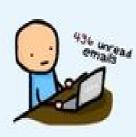


## 10 YEARS AGO





# NOW





## Going back to recent years



Geoffrey Hinton (Godfather of AI)



**CHESS AI** 









### Should we worried?

Tom was the first guy losing his job because of Artificial intelligence



#### **Artificial Intelligence**

The theory and development of computer systems able to perform tasks normally requiring human intelligence

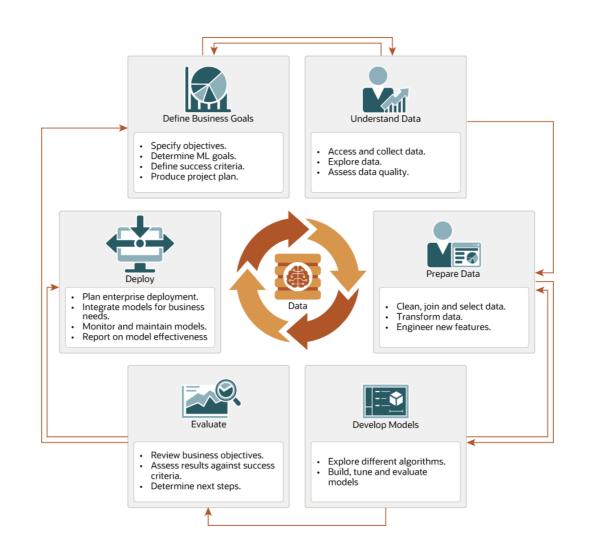
#### **Machine Learning**

Gives computers "the ability to learn without being explicitly programmed"

#### **Deep Learning**

Machine learning algorithms
with brain-like logical
structure of algorithms
called artificial neural
networks

LEVITY

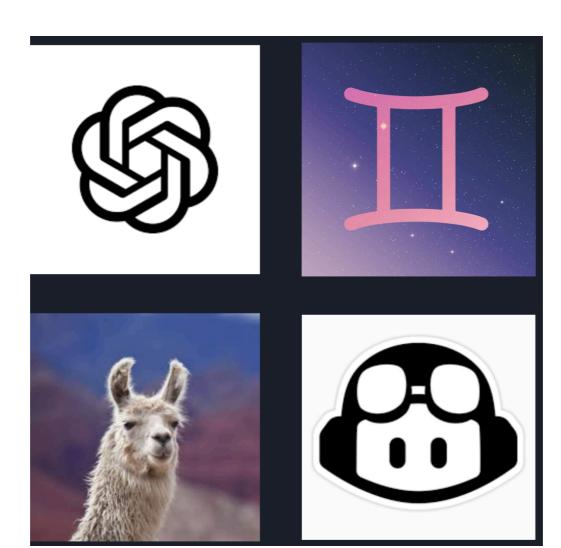


## AI Categories

- Salar Natural Language Processing
  - Text Recognition
  - Text Generation
- **©** Computer Vision
  - Image Segmentation
  - Image Classification
  - Object Detection
- Audio Recognition
  - Speech Recognition
  - Audio Classification

## Large Language Model

AI That Understands Human Language that Trained on Very Large Data





https://huggingface.co/

# Simple Code Example

```
!pip install datasets
!pip install accelerate -U
!pip install pip install transformers[torch]

# import all functions
from datasets import load_dataset
from transformers import AutoModelForSequenceClassification, AutoTokenizer, TrainingArguments, Trainer, DataCollatorWithPadding
import numpy as np

from google.colab import drive
drive.mount('/content/drive')
```

```
url = 'https://docs.google.com/spreadsheets/d/100vf3ZXFFKXTQXinkE58YnvC4njhvsk3fnBwrHRW5N8/export?format=csv'
ds = load_dataset('csv', data_files=url)
ds = ds["train"].train_test_split(test_size=0.2)
```

ds

model name = "distilbert-base-uncased"

model = AutoModelForSequenceClassification.from\_pretrained(model\_name)

tokenizer = AutoTokenizer.from\_pretrained(model\_name)

```
def tokenize(examples):
   outputs = tokenizer(examples['text'], truncation=True, padding=True)
   return outputs
```

tokenized\_ds = ds.map(tokenize, batched=True)

```
path = F"/content/drive/My Drive/distilbert-dana-mini"
training_args = TrainingArguments(num_train_epochs=1,
                                  output_dir=path,
                                  push_to_hub=False,
                                  per_device_train_batch_size=32,
                                  per_device_eval_batch_size=32,
                                  learning rate=5e-5,
                                  evaluation strategy="epoch")
trainer = Trainer(model=model, tokenizer=model tokenizer,
                  data_collator=data_collator,
                  args=training_args,
                  train_dataset=tokenized_ds["train"],
                  eval_dataset=tokenized_ds["test"])
trainer.train()
```

trainer.save\_model()

```
pipe_kwargs = {
   "top_k": None,
   "batch_size": 16
path = F"/content/drive/My Drive/distilbert-dana-review"
text = "Aplikasi Terbaik sepanjang masa"
reward_pipe = pipeline("sentiment-analysis", path, device=-1)
reward_output = reward_pipe(text, **pipe_kwargs)
print(reward_output)
```