Hands-On GenAI: LLMs, RAGs, and Agentic Systems for Beginners

Day 5

Adya Bhat and Tejas Venugopalan

High Level API Design

Intro to ML

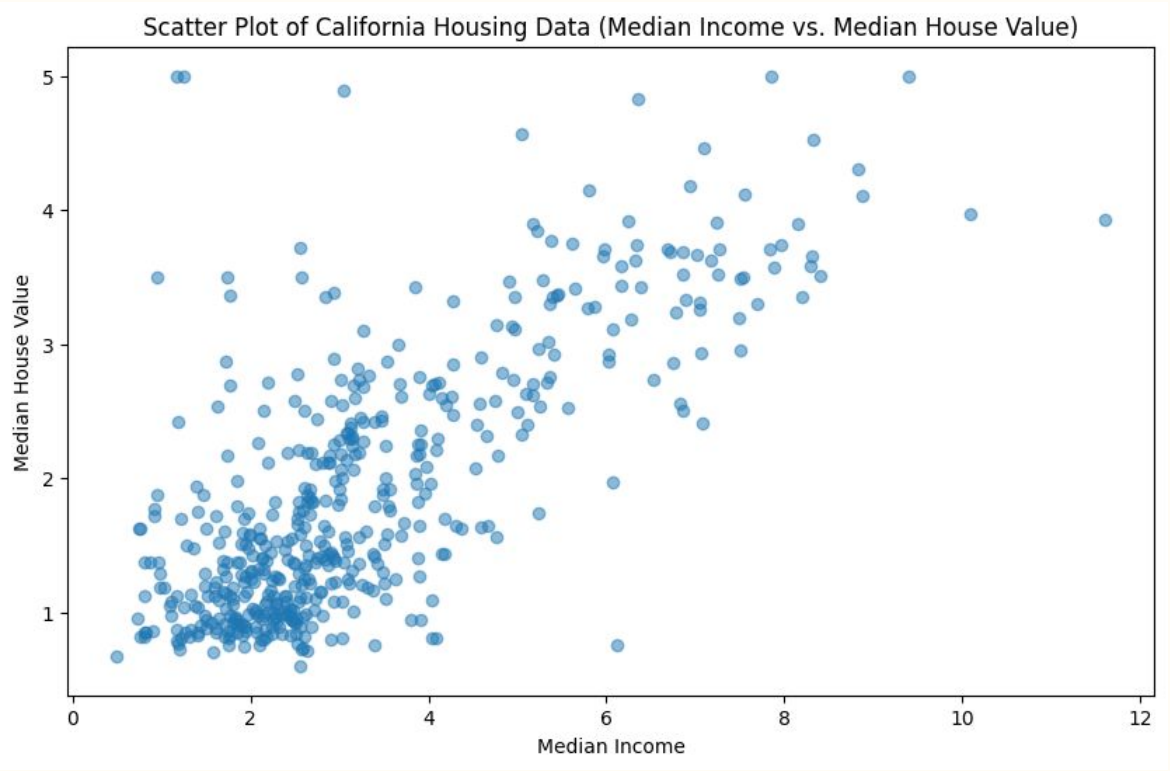
What does data look like in real life?

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseVal
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422
...	...	...	...	...	...	...	...	...	...
20635	1.5603	25.0	5.045455	1.133333	845.0	2.560606	39.48	-121.09	0.781
20636	2.5568	18.0	6.114035	1.315789	356.0	3.122807	39.49	-121.21	0.771
20637	1.7000	17.0	5.205543	1.120092	1007.0	2.325635	39.43	-121.22	0.923
20638	1.8672	18.0	5.329513	1.171920	741.0	2.123209	39.43	-121.32	0.847
20639	2.3886	16.0	5.254717	1.162264	1387.0	2.616981	39.37	-121.24	0.894

20640 rows × 9 columns

Eg. California housing prices data, from sklearn.datasets

What would you do here?

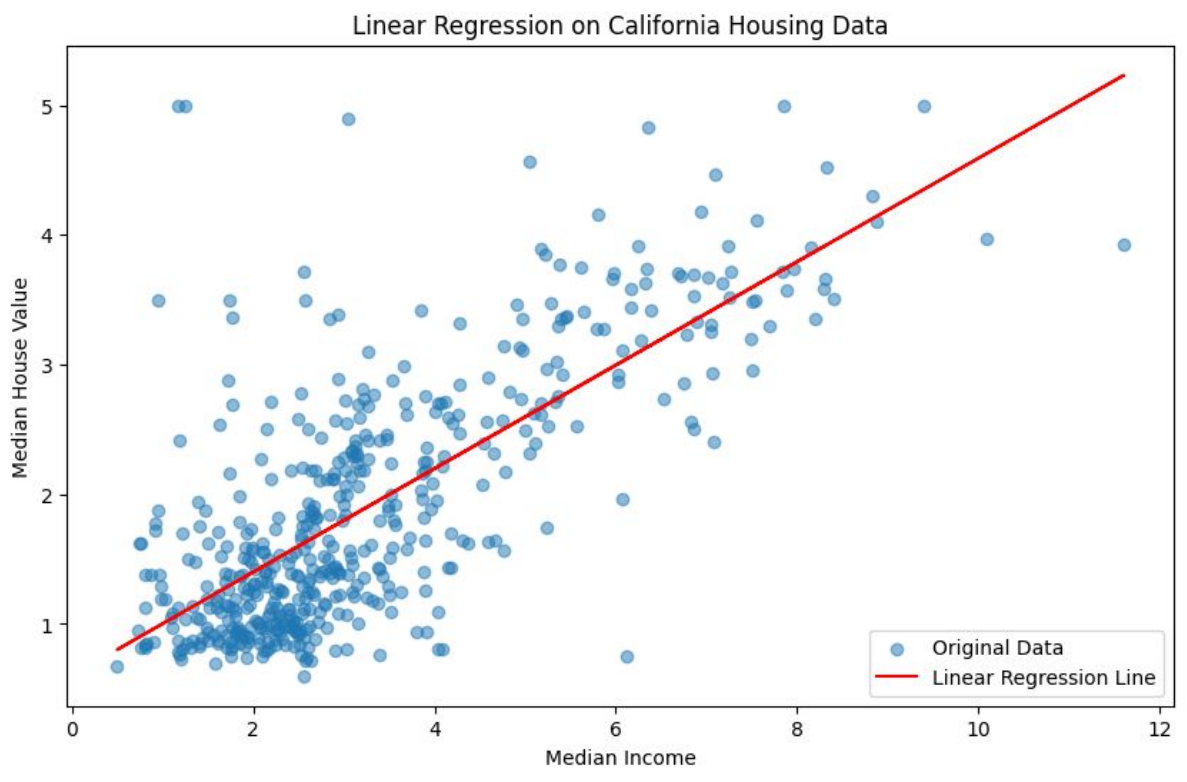


I want to predict housing prices, with respect to corresponding median household income. How can I do it?

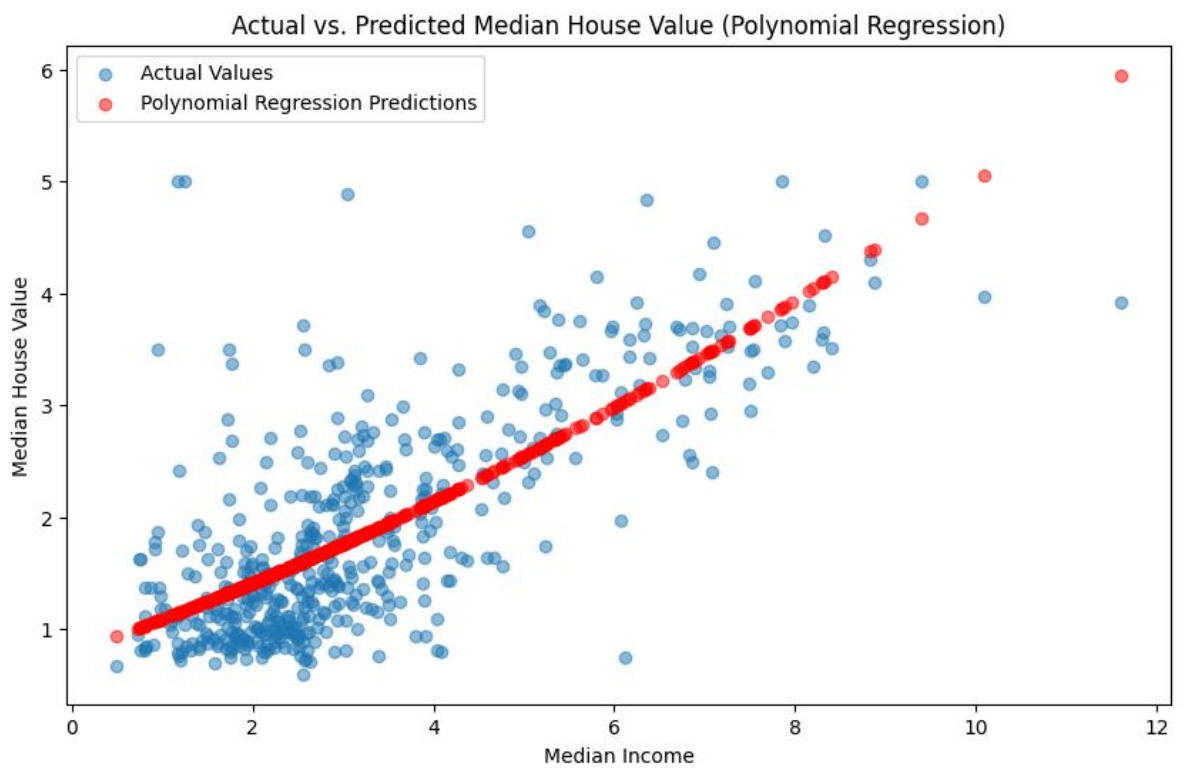
Here:

- **X**: Median Income
- **Y**: Median House Value

You can plot a simple linear regression model



You can plot a polynomial regression model



Intro to Machine Learning

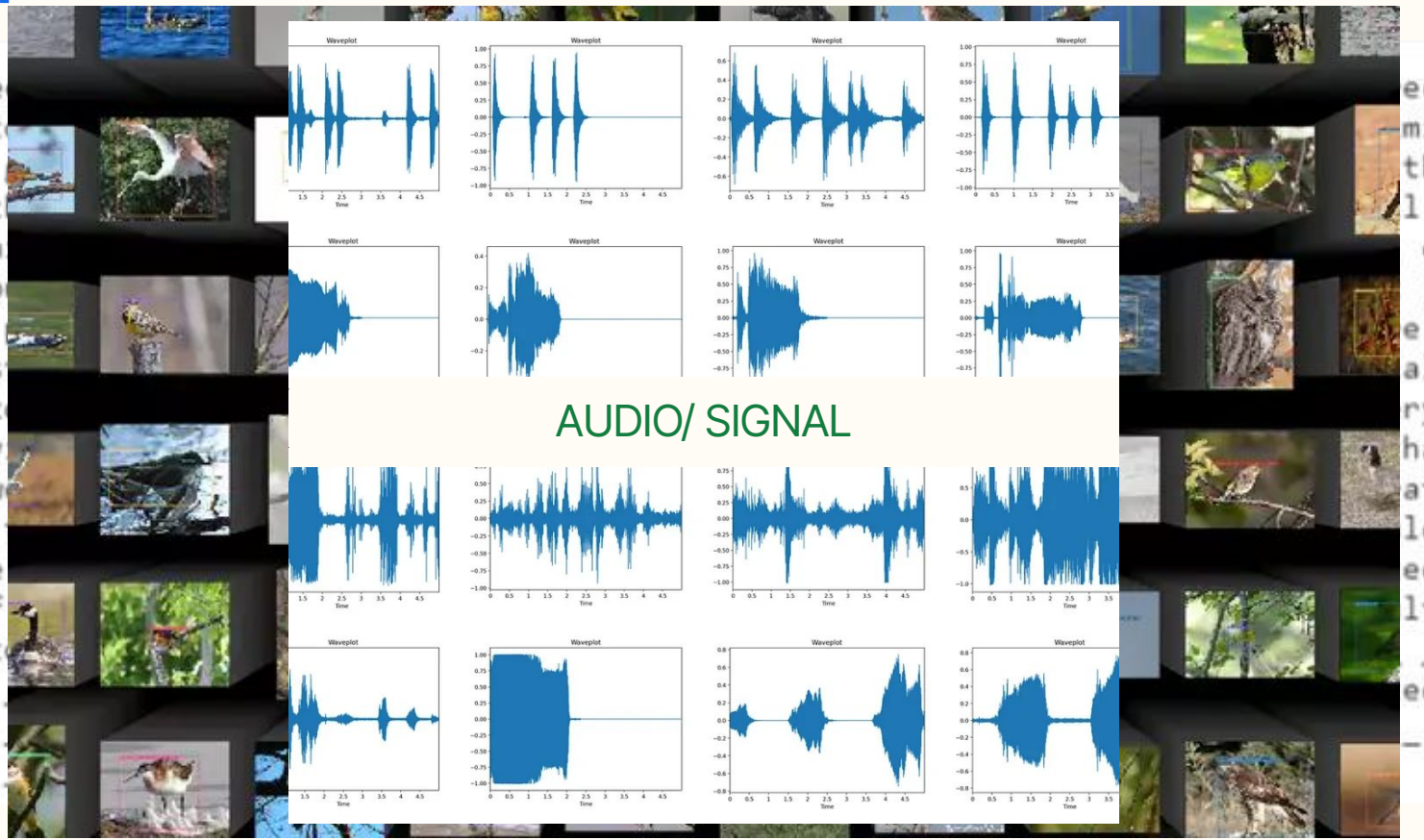
- what is machine learning? essentially- pattern recognition in data + understanding structure + using this in a task
- steps in an ML experiment:
  - **data** acquiring, analysis, conversion to appropriate format
  - deciding the **task** for the model
  - formulating **model architecture** and building: number of layers, kind of layers, input- output sizes of layers, different kind of connections between layers
  - **training** the model: tweaking model parameters over iterations by running through data
  - **evaluation** of the model on the task: using a metric depending on the task: accuracy, etc

In our example

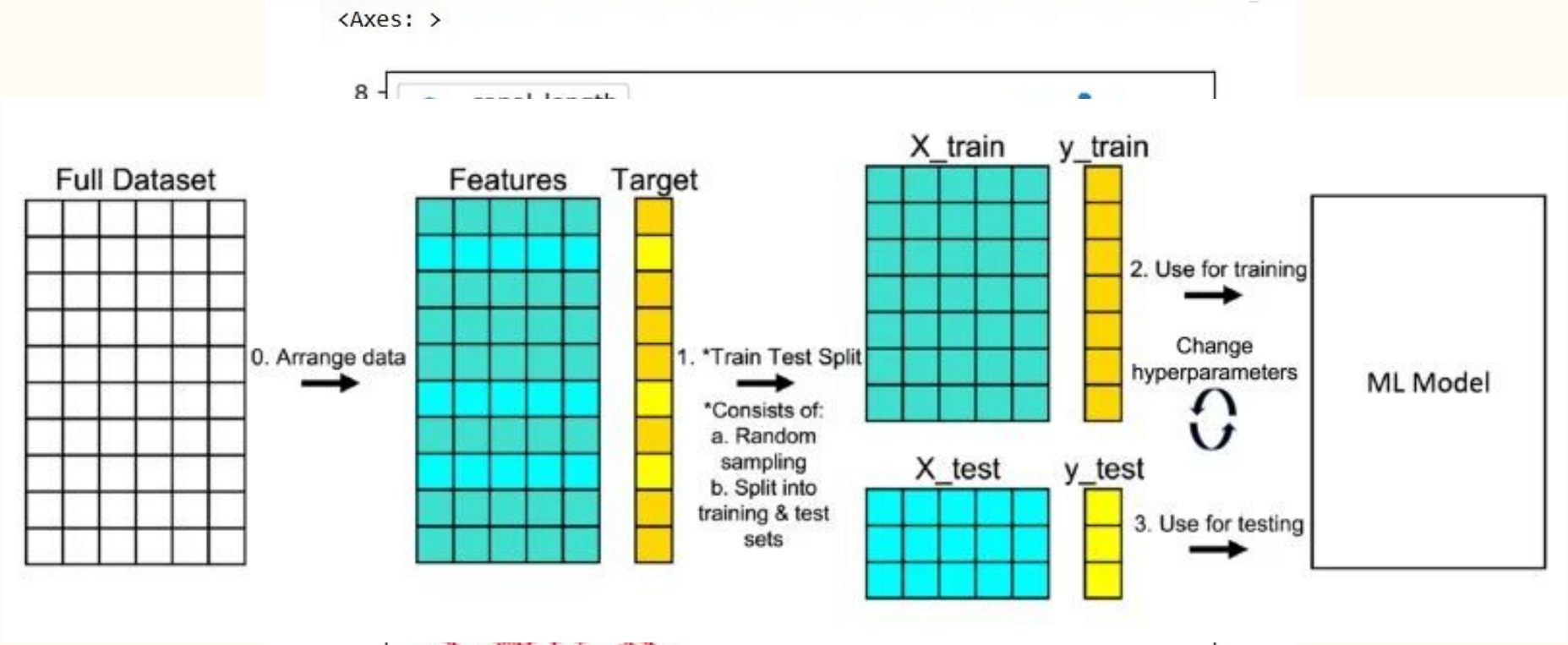
- **data:** tabular data of housing prices and housing details in California
- **task:** predict the housing prices, given the information on Median Household Income, House Age, Average Rooms, Average Bedrooms, Block Group Population, Average Number of Members in a Household, Block Group Latitude, Block Group Longitude
- **model:** linear regression, polynomial regression ⇒ essentially is finding a **function that fits most data points well**
- **training/ fitting:** neural network models are 'trained' on the data, non-neural models are 'fitted' on the data
- **evaluation:** of the model on the task (error between predicted value and actual value)

household ⇒ a group of people who reside together in a home
block ⇒ smallest geographical unit for which the U.S. Census Bureau publishes sample data

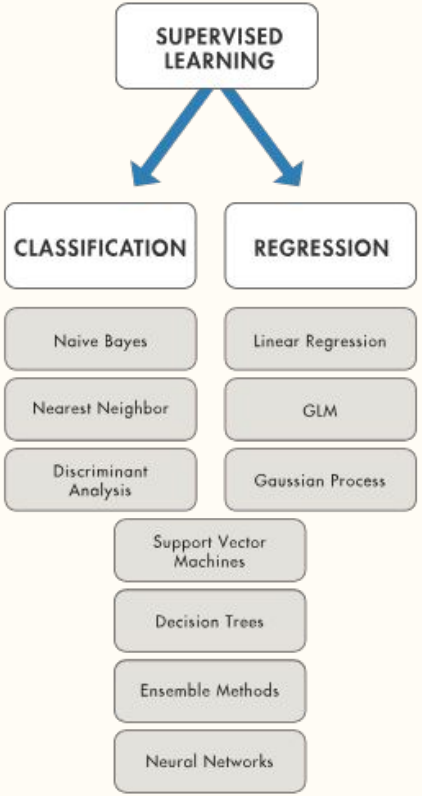
Data-

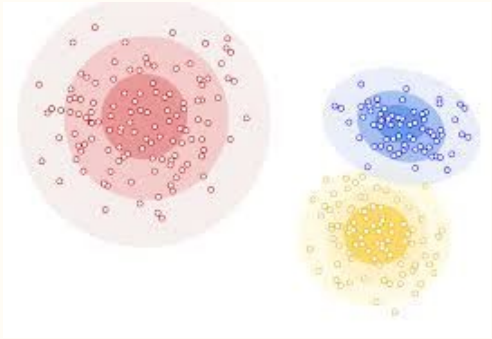


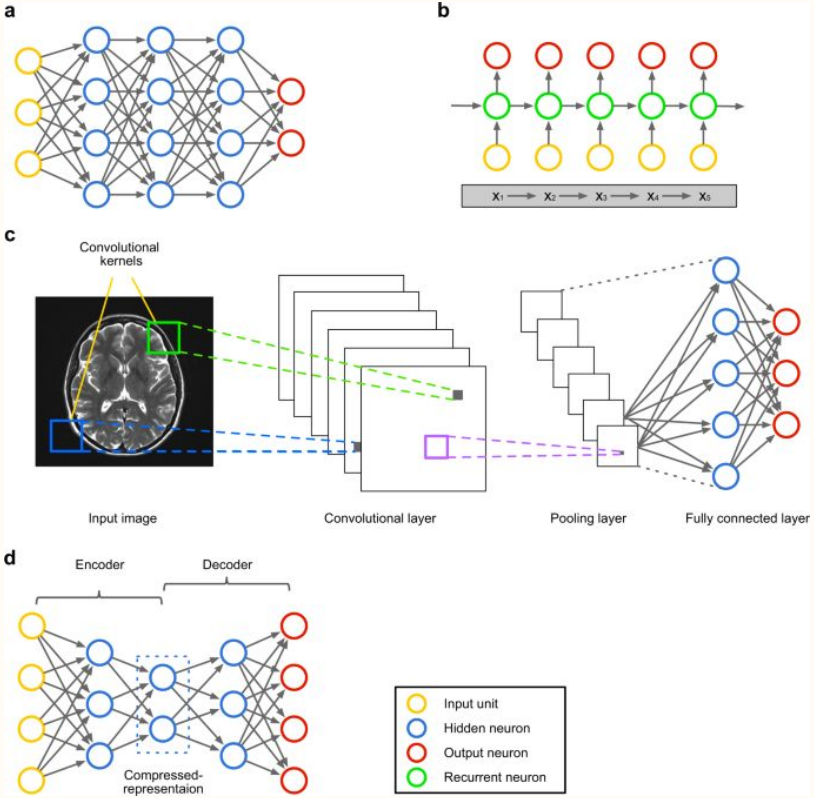
Data Analysis, Data Split



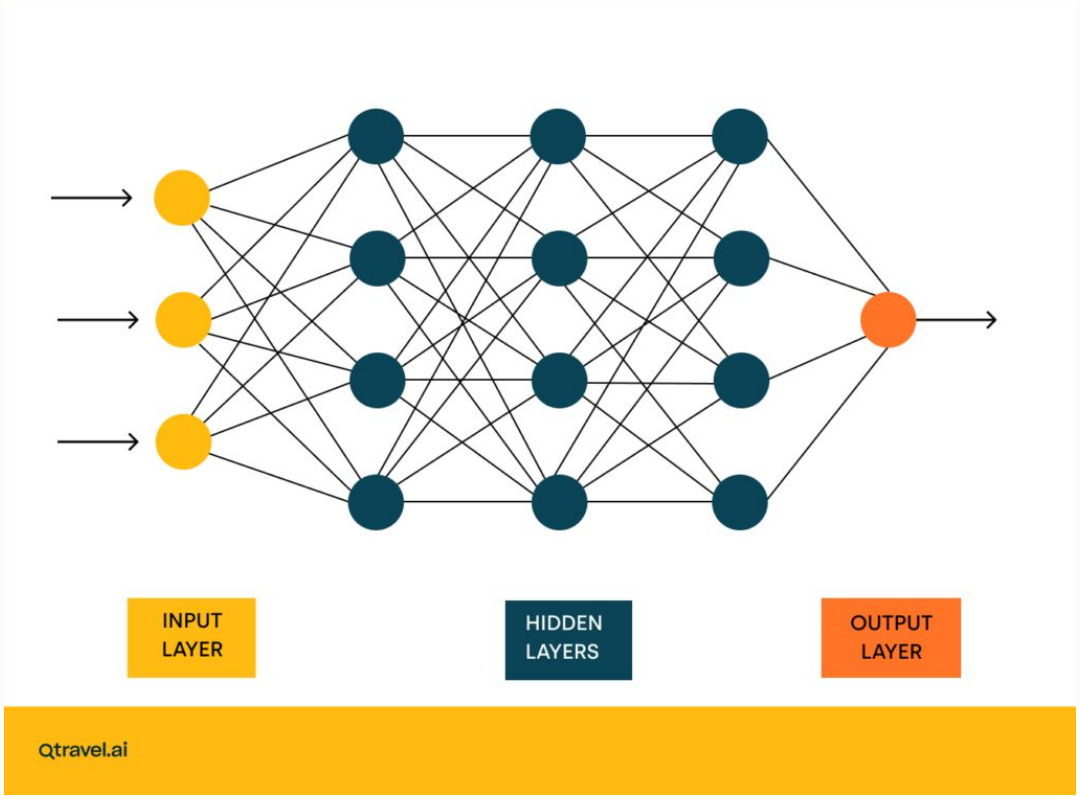
Models


SUPERVISED LEARNING
CLASSIFICATION
REGRESSION
Naive Bayes
Linear Regression
Nearest Neighbor
GLM
Discriminant Analysis
Gaussian Process
Support Vector Machines
Decision Trees
Ensemble Methods
Neural Networks




a
b
c
d
Convolutional kernels
Input image
Convolutional layer
Pooling layer
Fully connected layer
Encoder
Decoder
Compressed-representaion
Input unit
Hidden neuron
Output neuron
Recurrent neuron

Models



Tasks

Regression

Classification

Clustering

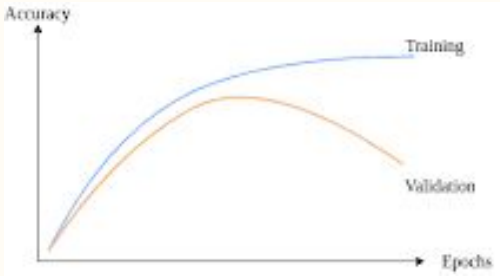
Association Rule Learning

Dimensionality Reduction

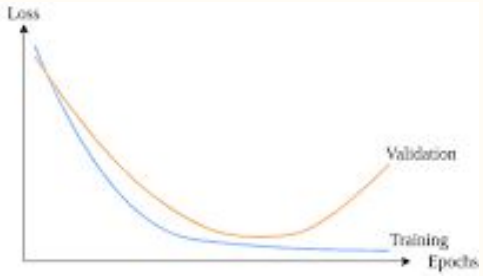
Generative Modelling

Training

```
Epoch: 070, Train: 0.8250, Test: 0.9000,Time(s) 11.7551
Epoch: 071, Train: 0.8250, Test: 0.8000,Time(s) 11.7546
Epoch: 072, Train: 0.9250, Test: 0.8000,Time(s) 11.7539
Epoch: 073, Train: 0.8000, Test: 0.8500,Time(s) 11.7541
Epoch: 074, Train: 0.8750, Test: 0.9000,Time(s) 11.7536
Epoch: 075, Train: 0.9000, Test: 0.8000,Time(s) 11.7535
Epoch: 076, Train: 0.8750, Test: 0.7500,Time(s) 11.7521
Epoch: 077, Train: 0.9000, Test: 0.8500,Time(s) 11.7505
Epoch: 078, Train: 0.9000, Test: 0.8000,Time(s) 11.7482
Epoch: 079, Train: 0.8750, Test: 0.8000,Time(s) 11.7456
Epoch: 080, Train: 0.7500, Test: 0.7500,Time(s) 11.7439
Epoch: 081, Train: 0.9000, Test: 0.8000,Time(s) 11.7423
Epoch: 082, Train: 0.9000, Test: 0.8000,Time(s) 11.7400
Epoch: 083, Train: 0.8500, Test: 0.8000,Time(s) 11.7383
Epoch: 084, Train: 0.8750, Test: 0.8500,Time(s) 11.7361
Epoch: 085, Train: 0.6250, Test: 0.7500,Time(s) 11.7338
Epoch: 086, Train: 0.6000, Test: 0.7000,Time(s) 11.7323
Epoch: 087, Train: 0.7250, Test: 0.7500,Time(s) 11.7304
Epoch: 088, Train: 0.6000, Test: 0.6500,Time(s) 11.7289
Epoch: 089, Train: 0.6750, Test: 0.7500,Time(s) 11.7274
Epoch: 090, Train: 0.8000, Test: 0.8000,Time(s) 11.7259
Epoch: 091, Train: 0.8500, Test: 0.6500,Time(s) 11.7245
Epoch: 092, Train: 0.7750, Test: 0.7000,Time(s) 11.7227
Epoch: 093, Train: 0.8000, Test: 0.8000,Time(s) 11.7209
Epoch: 094, Train: 0.8250, Test: 0.8500,Time(s) 11.7192
Epoch: 095, Train: 0.7500, Test: 0.8000,Time(s) 11.7179
Epoch: 096, Train: 0.8250, Test: 0.8500,Time(s) 11.7171
Epoch: 097, Train: 0.7750, Test: 0.8500,Time(s) 11.7156
Epoch: 098, Train: 0.8500, Test: 0.8500,Time(s) 11.7144
Epoch: 099, Train: 0.8750, Test: 0.8500,Time(s) 11.7143
```



Accuracy vs. epochs



Loss vs. epochs

Evaluation

- metrics
  - classification: accuracy
  - regression: mean squared error

Sample pipeline for image processing

- data- import, analysis, conversion to appropriate size
- task- predicting class of image (identifying animal from classes: dog, cat)
- model- **pretrained** image processing neural networks/ **building your own model**
- training- several epochs
- evaluation- accuracy of classification of data

LLMs

Intro to LLMs

- A Large Language Model (LLM) is an AI system trained to understand and generate human-like language.
- They work with text — reading, writing, summarizing, answering.
- LLMs are trained to predict the next word in a sentence, which leads to intelligent-seeking behavior.
- Like autocorrect on steroids — but smarter, because it learns meaning, context, and structure.

Predicting the next Token

LLMs are trained to predict the next word in a sequence. Over time, this simple objective helps them learn:

- Meaning
- Context
- Structure

A token is basically a small unit of text that the model can understand

The “Large” refers to:
Size of the model (parameters)
Size of training data

T5

- A Large Language Model (LLM) is an AI system trained to understand and generate human-like language.
- They work with text — reading, writing, summarizing, answering.
- LLMs are trained to predict the next word in a sentence, which leads to intelligent-seeking behavior.
- Like autocorrect on steroids — but smarter, because it learns meaning, context, and structure.

Sources

1. https://inria.github.io/scikit-learn-mooc/python_scripts/datasets_california_housing.html
2. Internet sources for images

Thank You!