

# Deploying AI

- During **deployment** data comes from the world and actions are carried out in the world.
- What technologies should we use?

- During **deployment** data comes from the world and actions are carried out in the world.
- What technologies should we use?
- Questions to consider:
  - ▶ Are the stakes high or low?  
(Can mistakes have severe consequences?)

- During **deployment** data comes from the world and actions are carried out in the world.
- What technologies should we use?
- Questions to consider:
  - ▶ Are the stakes high or low?  
(Can mistakes have severe consequences?)
  - ▶ Is there abundant homogeneous data to learn from?

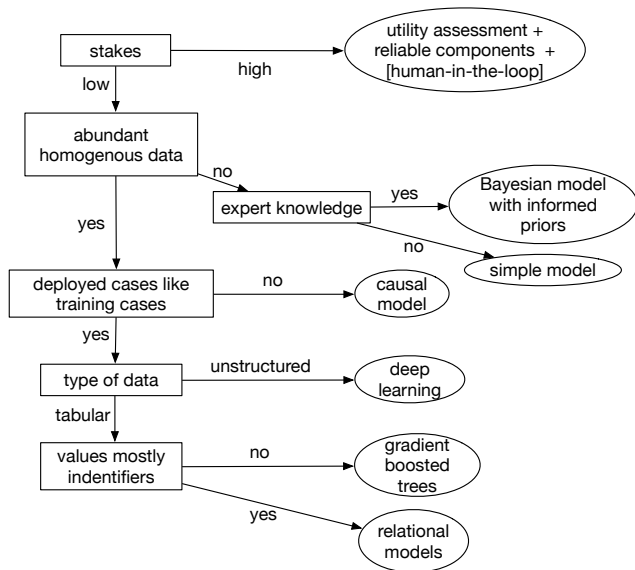
- During **deployment** data comes from the world and actions are carried out in the world.
- What technologies should we use?
- Questions to consider:
  - ▶ Are the stakes high or low?  
(Can mistakes have severe consequences?)
  - ▶ Is there abundant homogeneous data to learn from?
  - ▶ Is there expert (prior) knowledge?

- During **deployment** data comes from the world and actions are carried out in the world.
- What technologies should we use?
- Questions to consider:
  - ▶ Are the stakes high or low?  
(Can mistakes have severe consequences?)
  - ▶ Is there abundant homogeneous data to learn from?
  - ▶ Is there expert (prior) knowledge?
  - ▶ Are deployed cases like the training cases?

- During **deployment** data comes from the world and actions are carried out in the world.
- What technologies should we use?
- Questions to consider:
  - ▶ Are the stakes high or low?  
(Can mistakes have severe consequences?)
  - ▶ Is there abundant homogeneous data to learn from?
  - ▶ Is there expert (prior) knowledge?
  - ▶ Are deployed cases like the training cases?
  - ▶ Is the data tabular or more unstructured (e.g., images, sounds)?

- During **deployment** data comes from the world and actions are carried out in the world.
- What technologies should we use?
- Questions to consider:
  - ▶ Are the stakes high or low?  
(Can mistakes have severe consequences?)
  - ▶ Is there abundant homogeneous data to learn from?
  - ▶ Is there expert (prior) knowledge?
  - ▶ Are deployed cases like the training cases?
  - ▶ Is the data tabular or more unstructured (e.g., images, sounds)?
  - ▶ If tabular, are most values identifiers (e.g, transaction/ product numbers) or discrete/real values or other?

# Deploying AI





## Deploying AI (cont.)

- When the stakes are low, there is abundant homogenous data and the deployed cases are expected to be like the training cases, pure **machine learning** can be used.

## Deploying AI (cont.)

- When the stakes are low, there is abundant homogenous data and the deployed cases are expected to be like the training cases, pure **machine learning** can be used.
- **Deep learning** has proved to be the choice for unstructured and perceptual data where there are not pre-defined features, such as images, speech, text, and protein sequences.

## Deploying AI (cont.)

- When the stakes are low, there is abundant homogenous data and the deployed cases are expected to be like the training cases, pure **machine learning** can be used.
- **Deep learning** has proved to be the choice for unstructured and perceptual data where there are not pre-defined features, such as images, speech, text, and protein sequences.
- For tabular data where the values in the tables can be used to construct features, **gradient-boosted trees** work well.

## Deploying AI (cont.)

- When the stakes are low, there is abundant homogenous data and the deployed cases are expected to be like the training cases, pure **machine learning** can be used.
- **Deep learning** has proved to be the choice for unstructured and perceptual data where there are not pre-defined features, such as images, speech, text, and protein sequences.
- For tabular data where the values in the tables can be used to construct features, **gradient-boosted trees** work well.
- **Relational models** are used for tabular data where most values are identifiers.

## Deploying AI (cont.)

- When the stakes are low, there is abundant homogenous data and the deployed cases are expected to be like the training cases, pure **machine learning** can be used.
- **Deep learning** has proved to be the choice for unstructured and perceptual data where there are not pre-defined features, such as images, speech, text, and protein sequences.
- For tabular data where the values in the tables can be used to construct features, **gradient-boosted trees** work well.
- **Relational models** are used for tabular data where most values are identifiers.
- If the assumption that deployment is like training is inappropriate, **causal models** can be used to adjust to the deployed situation.

## Deploying AI (cont.)

- If there is not much data, but there is expert knowledge, a causal model with informed priors (e.g., using a **Dirichlet distribution**) can combine expertise and data.

## Deploying AI (cont.)

- If there is not much data, but there is expert knowledge, a causal model with informed priors (e.g., using a **Dirichlet distribution**) can combine expertise and data.
- If there is little data and no expertise, a simple model such as a **decision tree** or a **linear model** is typically the best that can be done. Simplicity depends on the amount of data.

## Deploying AI (cont.)

- If there is not much data, but there is expert knowledge, a causal model with informed priors (e.g., using a **Dirichlet distribution**) can combine expertise and data.
- If there is little data and no expertise, a simple model such as a **decision tree** or a **linear model** is typically the best that can be done. Simplicity depends on the amount of data.
- When the stakes are high:



- If there is not much data, but there is expert knowledge, a causal model with informed priors (e.g., using a **Dirichlet distribution**) can combine expertise and data.
- If there is little data and no expertise, a simple model such as a **decision tree** or a **linear model** is typically the best that can be done. Simplicity depends on the amount of data.
- When the stakes are high:
  - ▶ Carry out a complex cost–benefit analysis, based on the **utility** of all affected stakeholders.
  - ▶ Ensure that failures and possibly bad outcomes can be explained and debugged
  - ▶ Use a combination of techniques, where each component is well tested and reliable.