A Gentle Introduction to Machine Learning

Presented by Yasin Ceran

November 29th, 2020

What is Machine Learning?

- What do you picture when you hear "Machine Learning"?
  - A robot: a dependable butler or a deadly Terminator?
  - Self learning super computers?

Machine Learning

- Machine Learning is the science (and art) of programming computers so they can learn from data.
- It is the field of study that gives computers the ability to learn without being explicitly programmed. —Arthur Samuel, 1959

What is Machine Learning?

- What do you picture when you hear "Machine Learning"?
  - A robot: a dependable butler or a deadly Terminator?
  - Self learning super computers?

Machine Learning

- Machine Learning is the science (and art) of programming computers so they can learn from data.
- It is the field of study that gives computers the ability to learn without being explicitly programmed. —Arthur Samuel, 1959

What is Machine Learning?

- What do you picture when you hear "Machine Learning"?
  - A robot: a dependable butler or a deadly Terminator?
  - Self learning super computers?

Machine Learning

- Machine Learning is the science (and art) of programming computers so they can learn from data.
- It is the field of study that gives computers the ability to learn without being explicitly programmed. —Arthur Samuel, 1959

What is Machine Learning?

- What do you picture when you hear "Machine Learning"?
  - A robot: a dependable butler or a deadly Terminator?
  - Self learning super computers?

Machine Learning

- Machine Learning is the science (and art) of programming computers so they can learn from data.
- It is the field of study that gives computers the ability to learn without being explicitly programmed. —Arthur Samuel, 1959

What is Machine Learning?

- What do you picture when you hear "Machine Learning"?
  - A robot: a dependable butler or a deadly Terminator?
  - Self learning super computers?

Machine Learning

- Machine Learning is the science (and art) of programming computers so they can learn from data.
- It is the field of study that gives computers the ability to learn without being explicitly programmed. —Arthur Samuel, 1959

What is Machine Learning?

- What do you picture when you hear "Machine Learning"?
  - A robot: a dependable butler or a deadly Terminator?
  - Self learning super computers?

Machine Learning

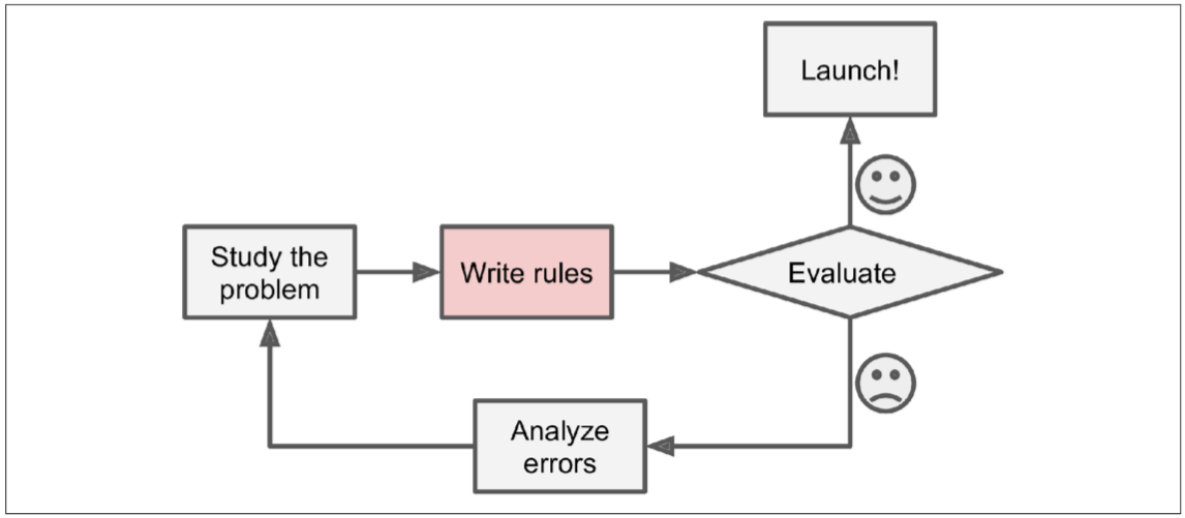
- Machine Learning is the science (and art) of programming computers so they can learn from data.
- It is the field of study that gives computers the ability to learn without being explicitly programmed. —Arthur Samuel, 1959

Why Machine Learning?

- Machine Learning is about extracting knowledge from data.
- Why machine learning?
  - Rule based systems are difficult to generalize
  - Rule based systems require a good understanding of the system
  - Scale of data
  - Unexpected findings

An Example on Why Machine Learning (1).

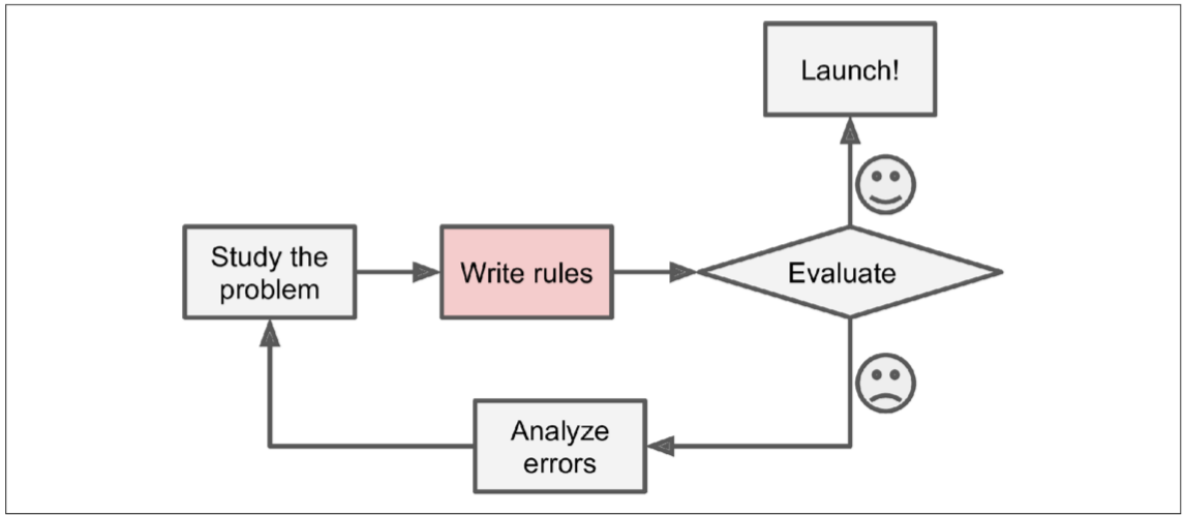
- Consider how you would write a spam filter using traditional programming techniques



1. Consider what spam typically looks like
2. You would write a detection algorithm for each of the patterns that you noticed
3. You would test your program and repeat steps 1 and 2 until it was good enough to launch.

An Example on Why Machine Learning (1).

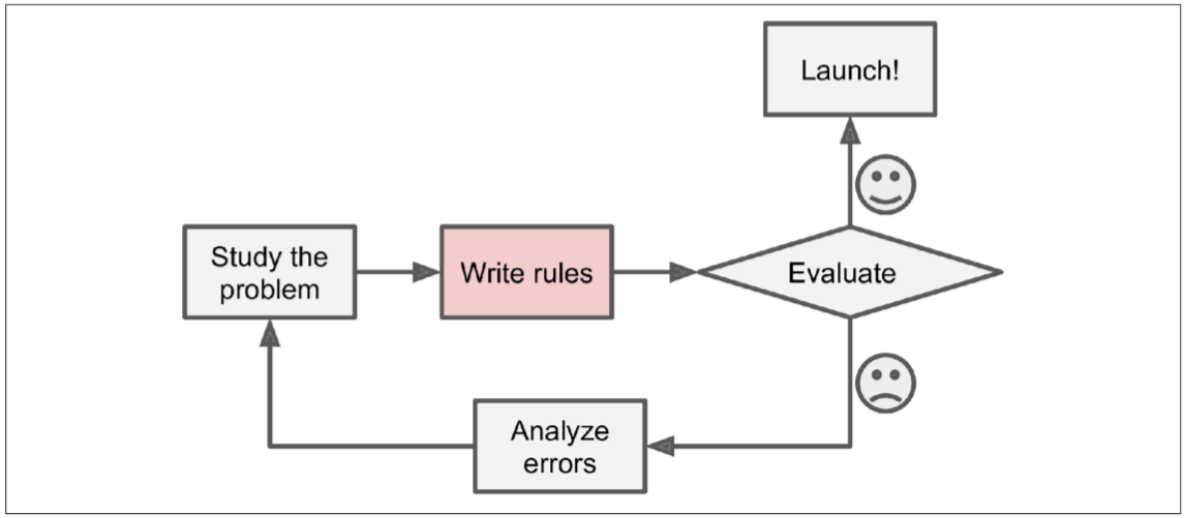
- Consider how you would write a spam filter using traditional programming techniques



1. Consider what spam typically looks like
2. You would write a detection algorithm for each of the patterns that you noticed
3. You would test your program and repeat steps 1 and 2 until it was good enough to launch.

An Example on Why Machine Learning (1).

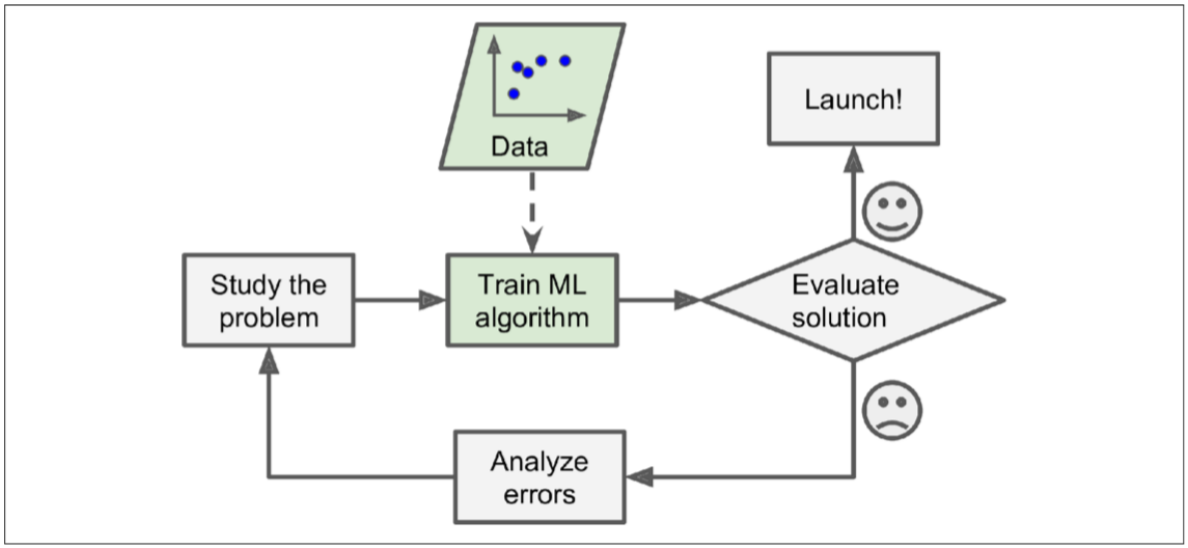
- Consider how you would write a spam filter using traditional programming techniques



1. Consider what spam typically looks like
2. You would write a detection algorithm for each of the patterns that you noticed
3. You would test your program and repeat steps 1 and 2 until it was good enough to launch.

An Example on Why Machine Learning (2).

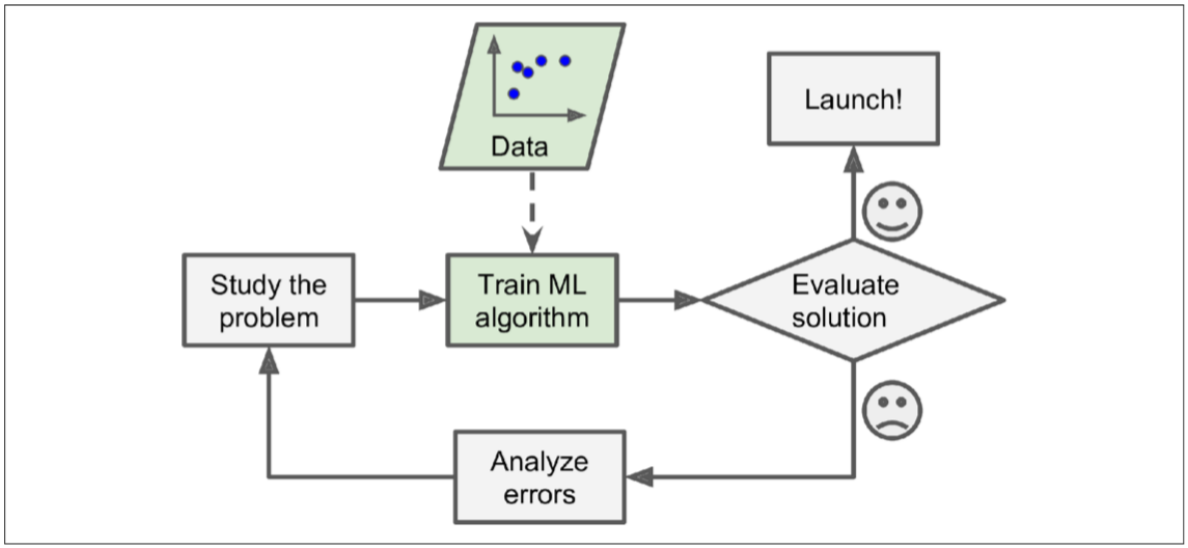
- A spam filter based on Machine Learning techniques automatically learns which words and phrases are good predictors of spam



- Detect unusually frequent patterns of words in the spam examples compared to the ham examples

An Example on Why Machine Learning (2).

- A spam filter based on Machine Learning techniques automatically learns which words and phrases are good predictors of spam



- Detect unusually frequent patterns of words in the spam examples compared to the ham examples

Machine Learning can help humans learn

- ML algorithms can be inspected to see what they have learned

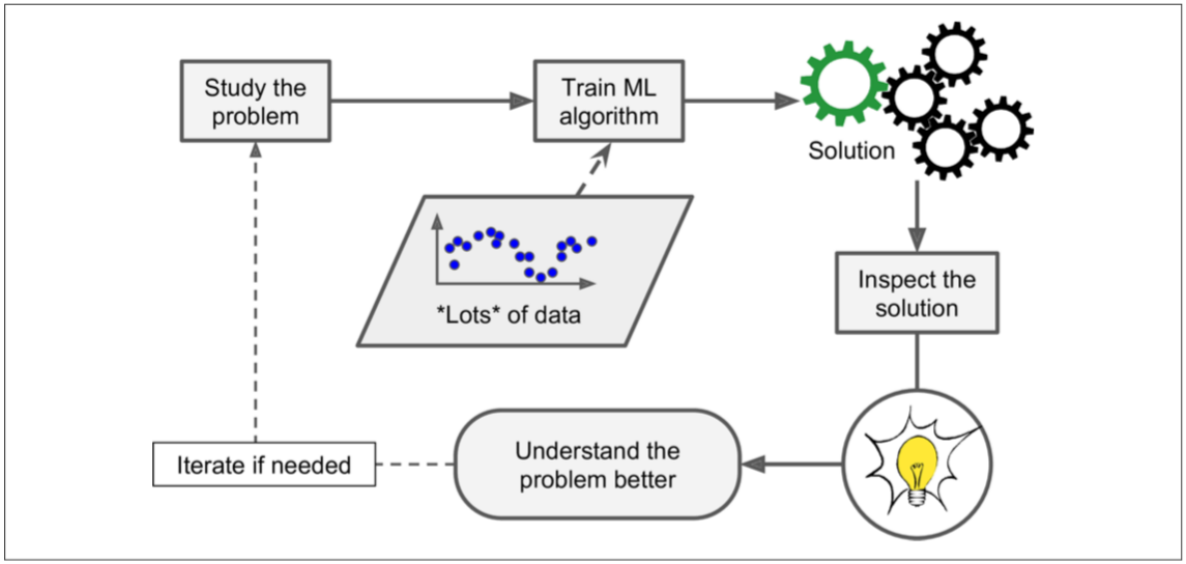


Image Processing

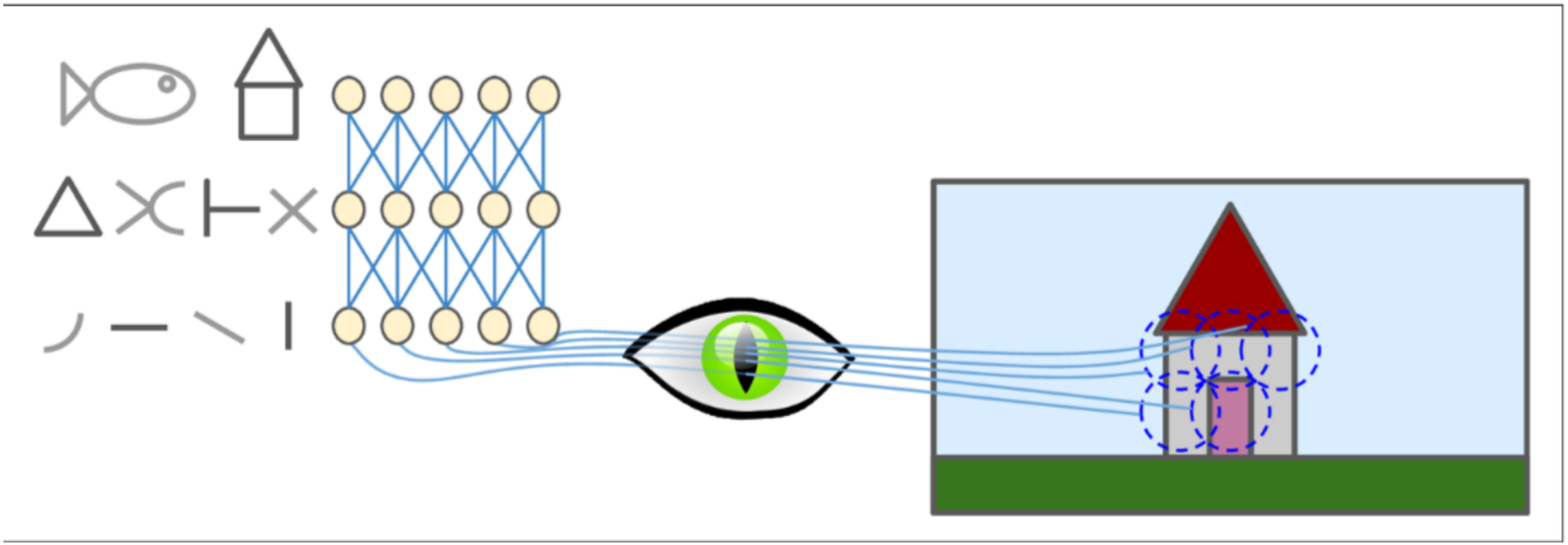


Image Processing

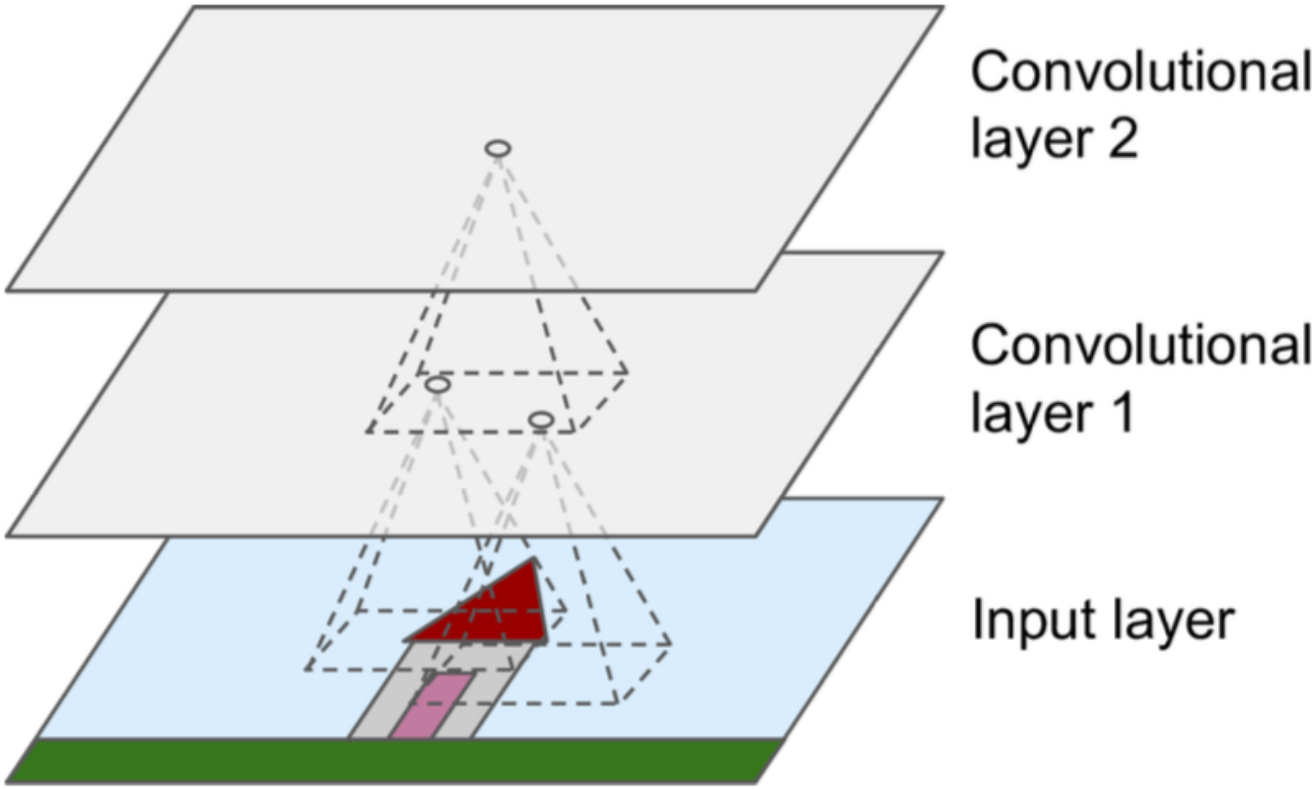


Image Processing

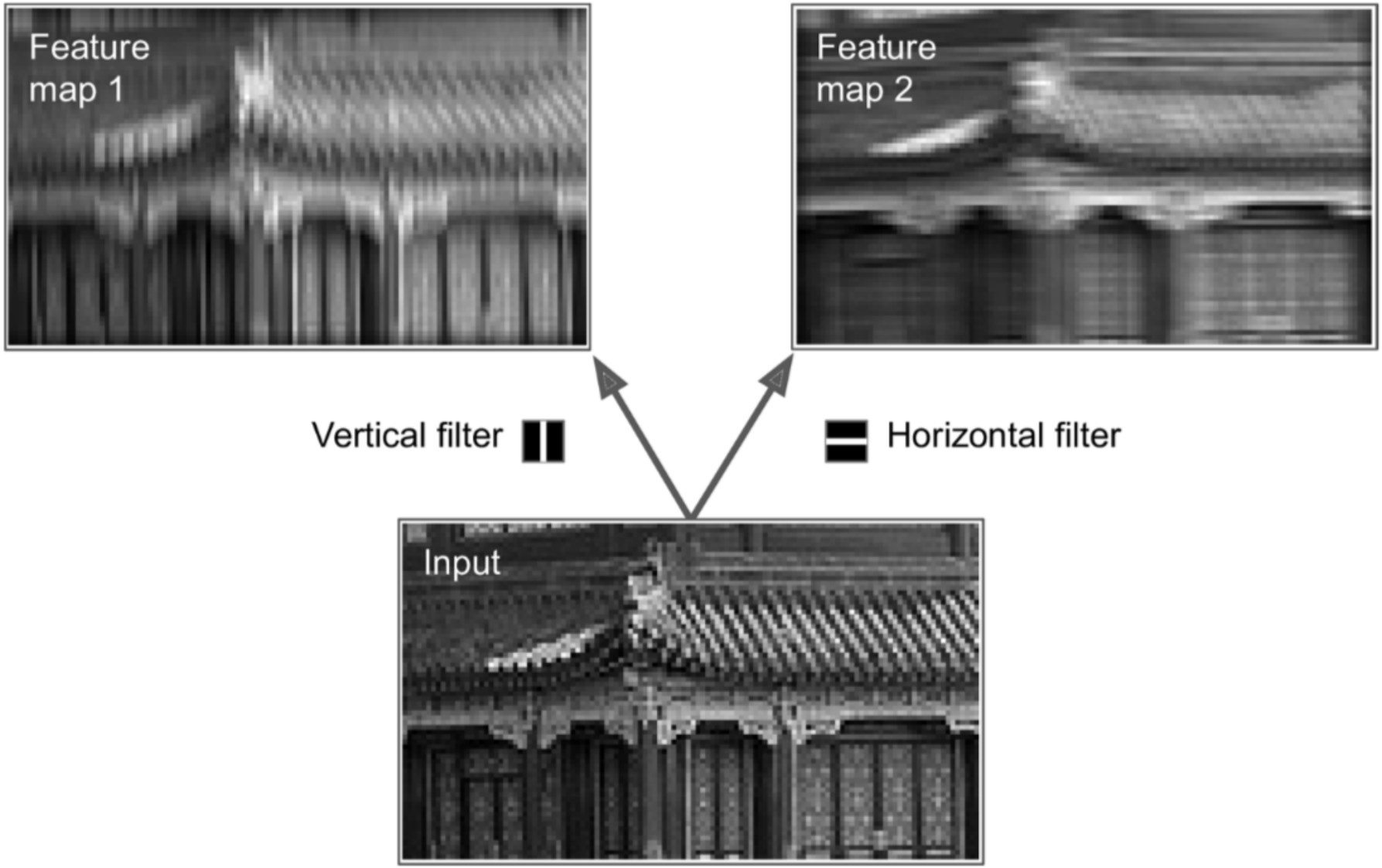
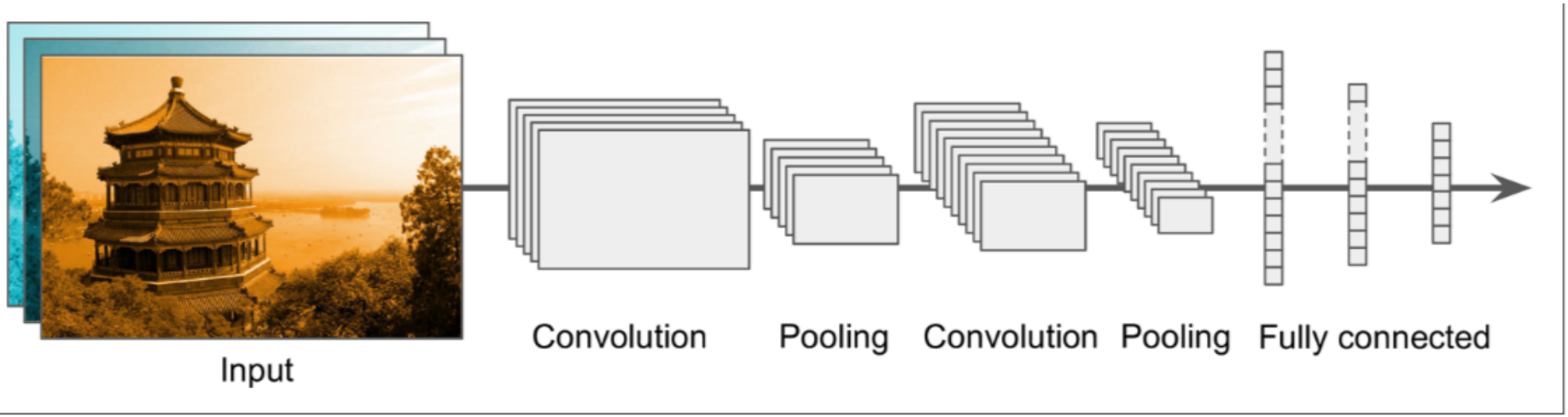


Image Processing

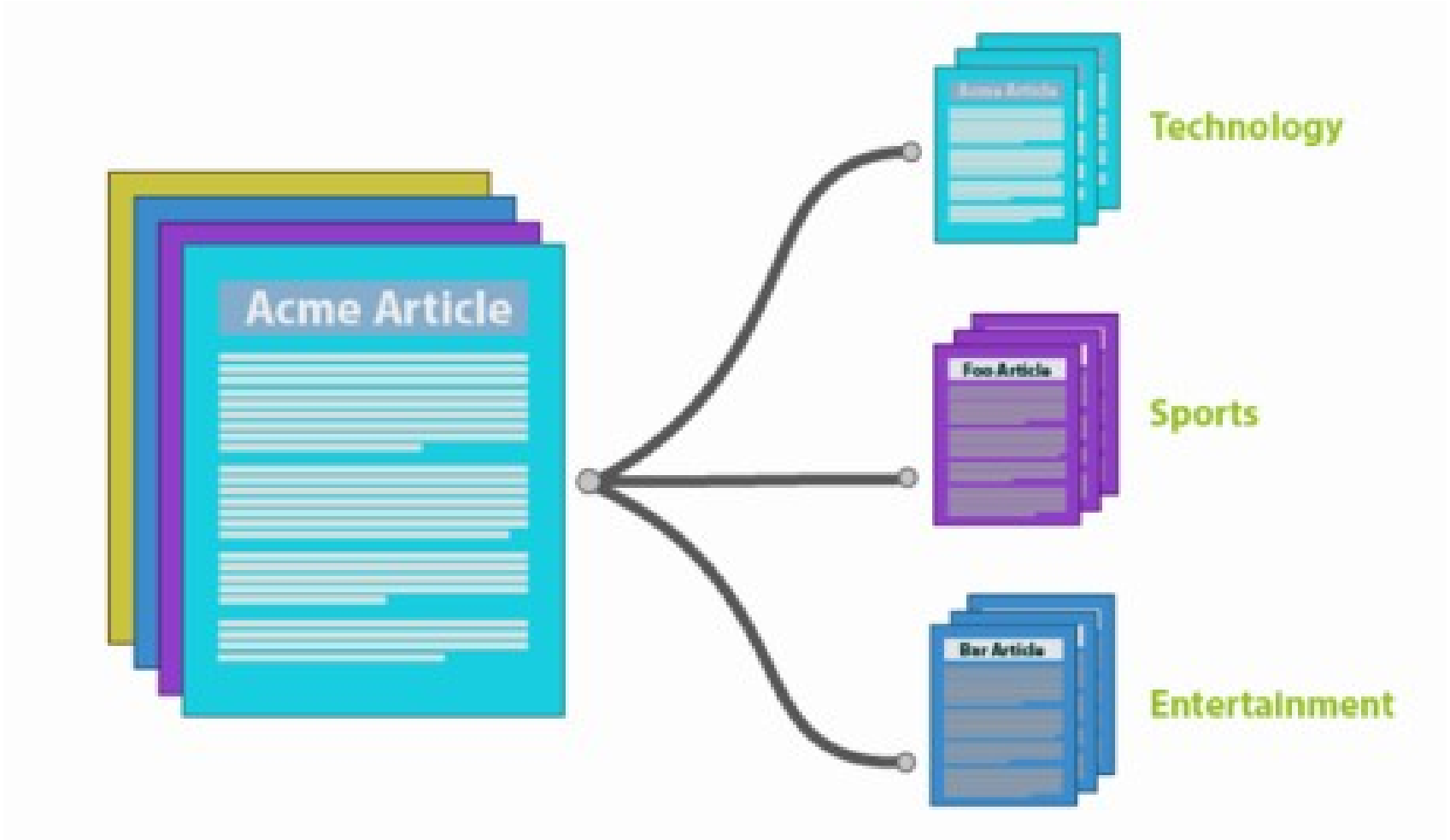


- Analyzing images of products on a production line to automatically classify them
- Detecting tumors in brain scans

Natural Language Processing

Natural Language Processing

Automatically classifying news articles

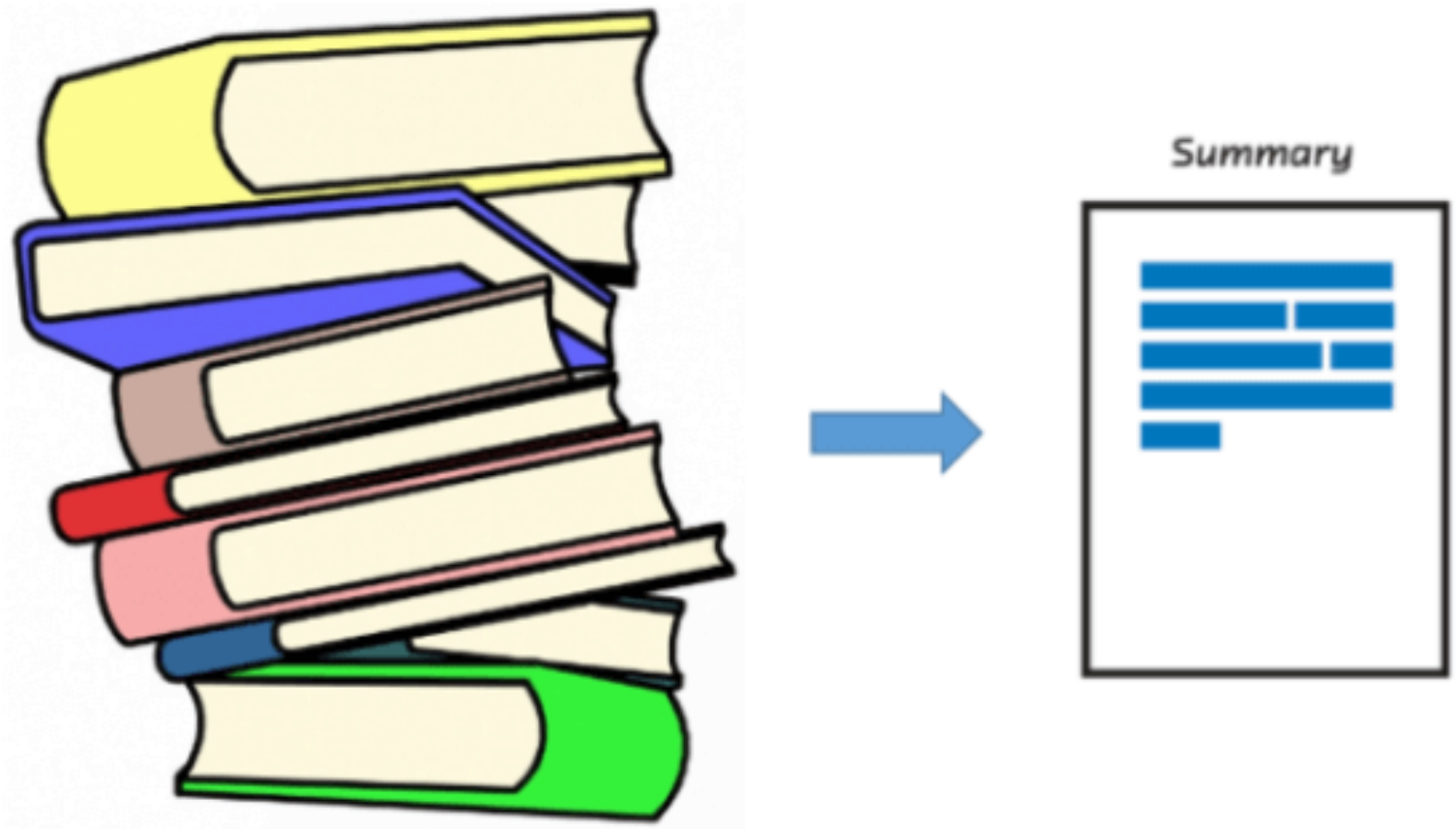


Natural Language Processing

Automatically flagging offensive comments on discussion forums

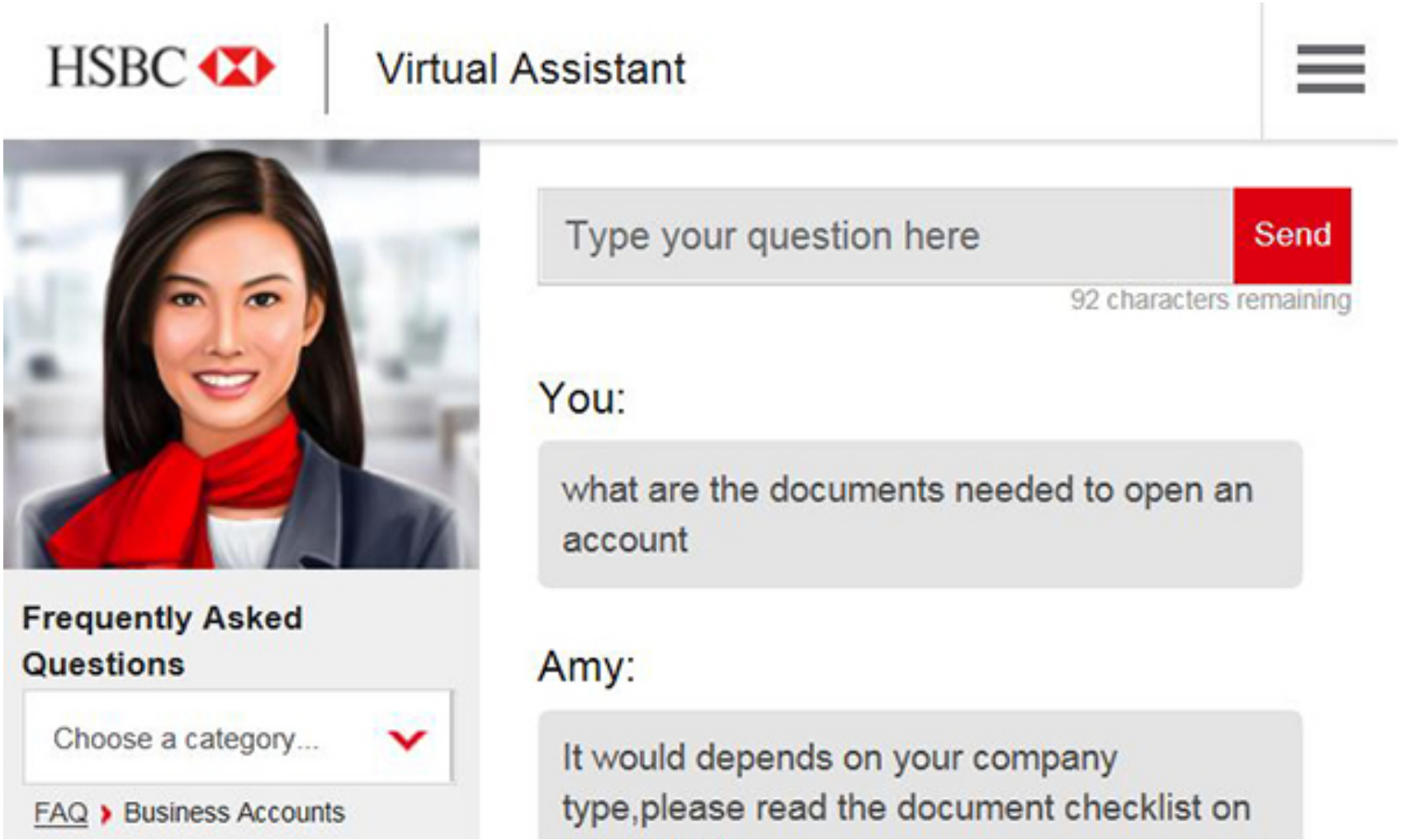
Natural Language Processing

Summarizing long documents automatically



Natural Language Processing

Creating a chatbot or a personal assistant



Business Applications

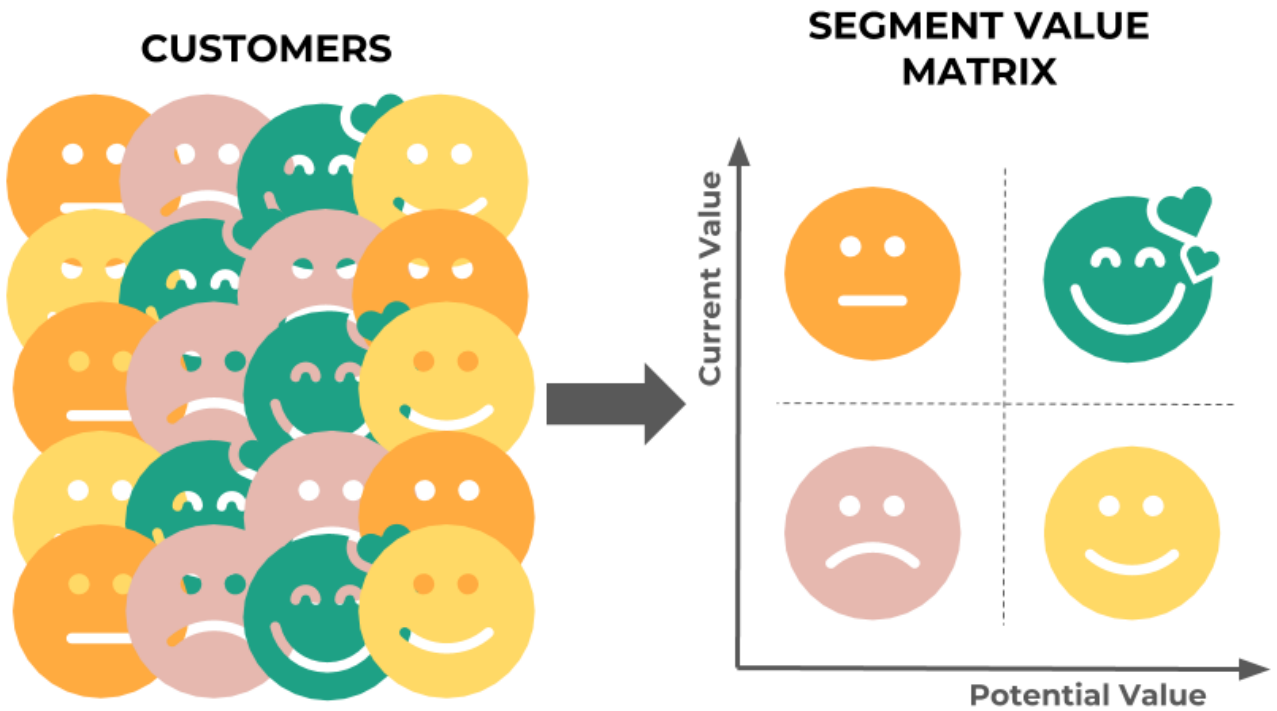
Forecasting your company's revenue next year, based on many performance metrics

Business Applications

Detecting credit card fraud

Business Applications

Segmenting clients based on their purchases



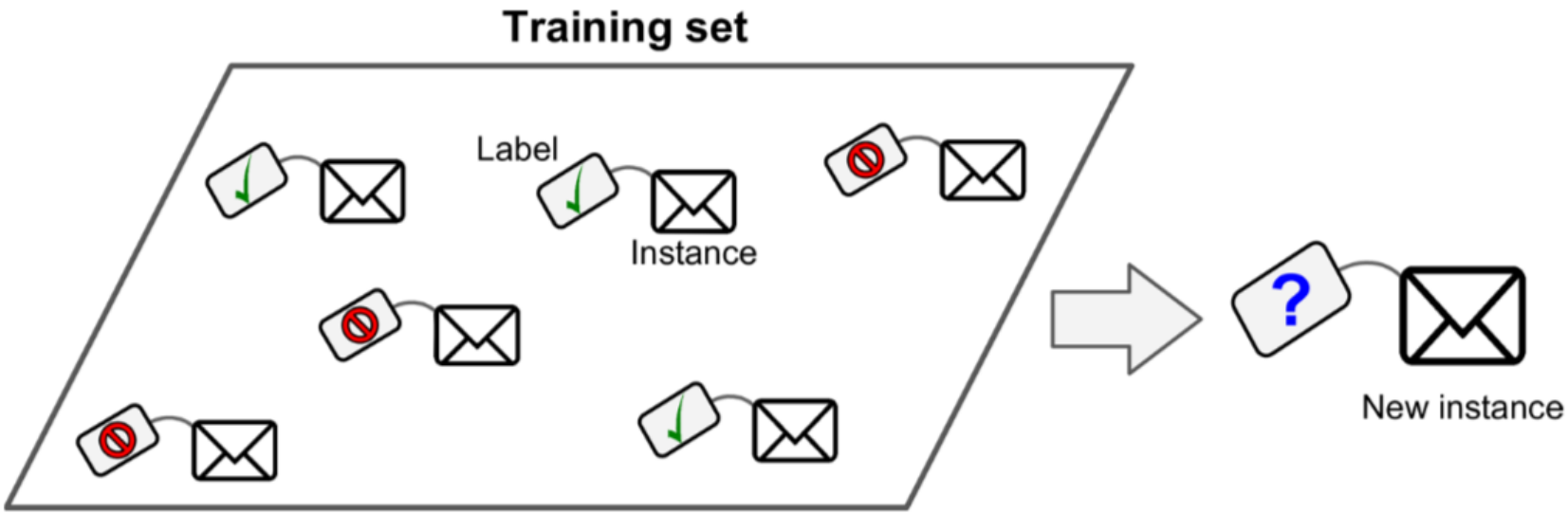
Business Applications

Recommending a product that a client may be interested in

Types of Machine Learning

- Supervised Learning
  - Labeled data
  - Direct feedback
  - Predict outcome/future
- Unsupervised Learning
  - No labels
  - No feedback
  - Find hidden structure in data
- Reinforcement Learning
  - Decision process
  - Reward system
  - Learn series of actions

Supervised Learning



In supervised learning, the training set you feed to the algorithm includes the desired solutions, called labels.

Examples of Supervised Learning

Examples of Supervised Learning include:

- spam detection
- medical diagnosis
- ad click prediction

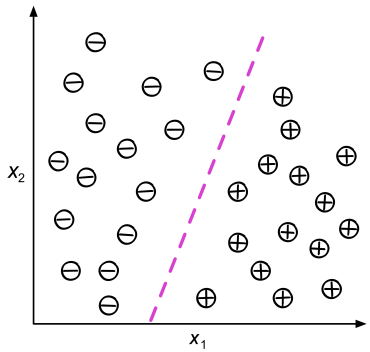


Figure: Classification

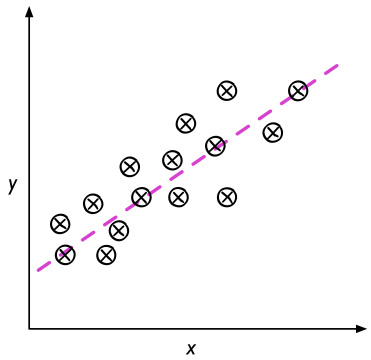
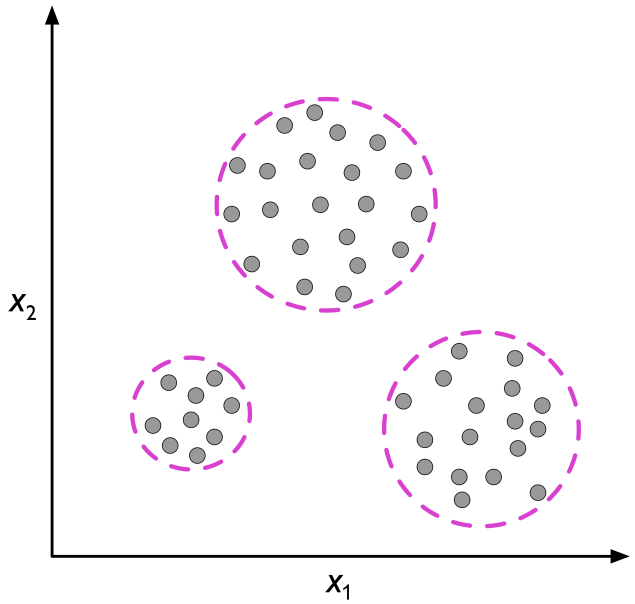


Figure: Regression

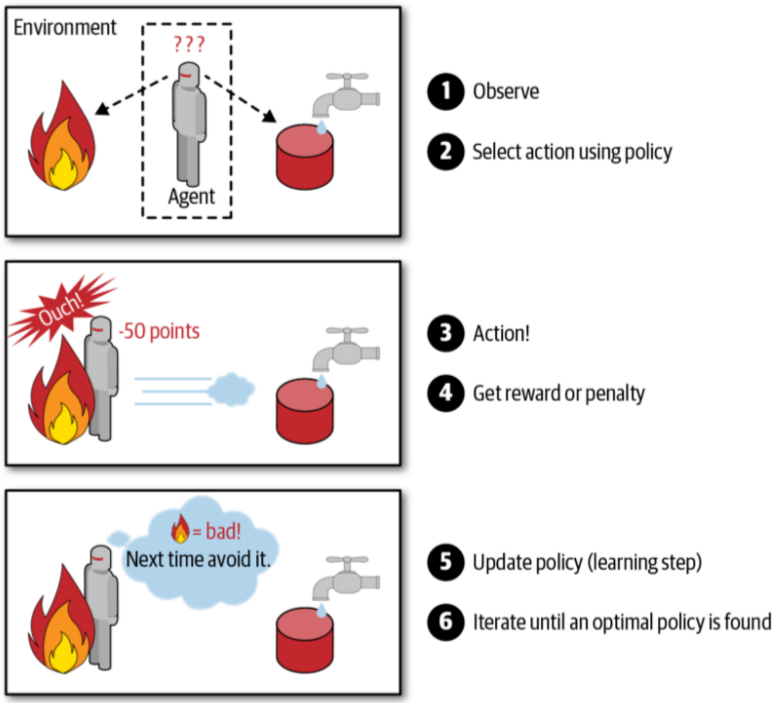
Unsupervised Learning

- In unsupervised learning, the training data is unlabeled. The system tries to learn without a teacher.
- For example, say you have a lot of data about your blog's visitors. You may want to run a clustering algorithm to try to detect groups of similar visitors.



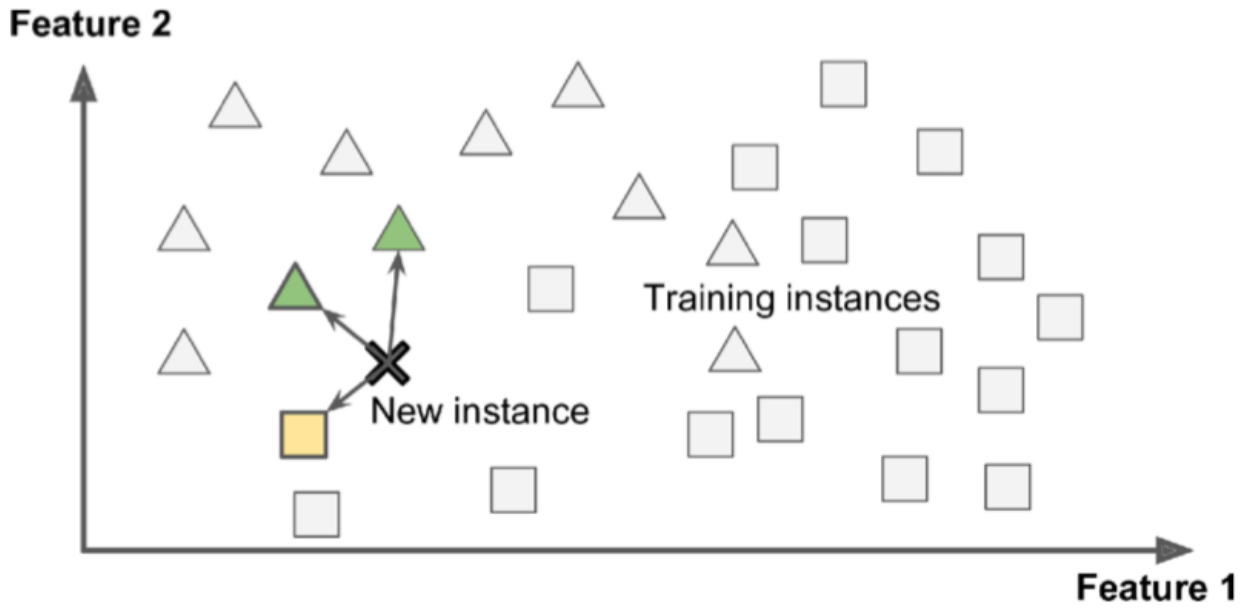
Reinforcement Learning

- The goal is to develop a system (agent) that improves its performance based on interactions with the environment
- The dataset uses a "rewards/punishments" system, offering feedback to the algorithm to learn from its own experiences by trial and error.

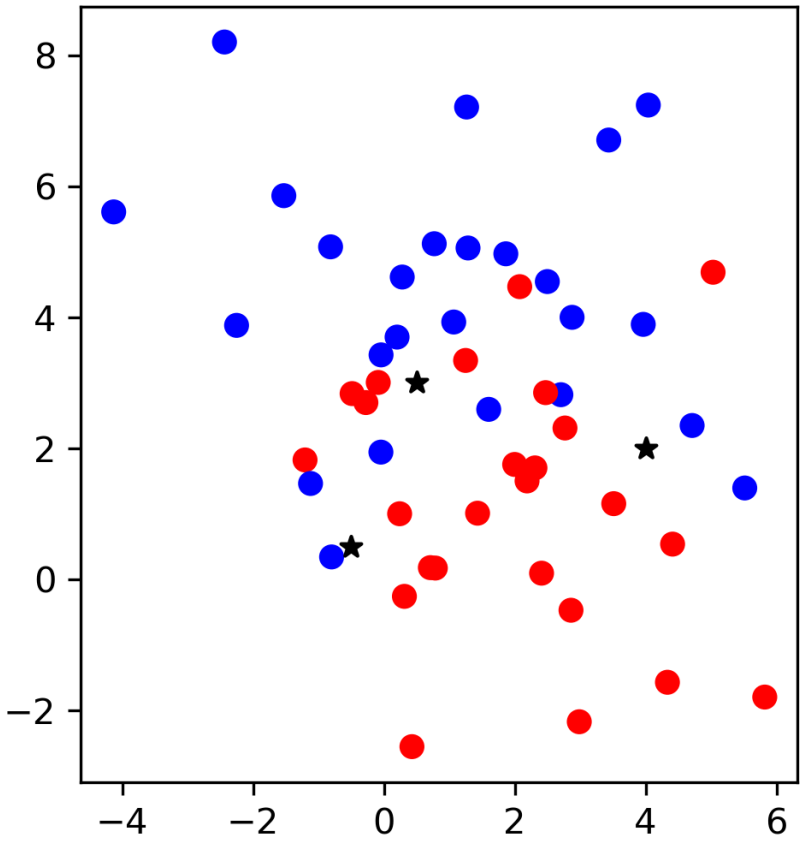


Instance Based Learning

With instance-based learning, the system learns the examples by heart, then generalizes to new cases by using a similarity measure to compare them to the learned examples (or a subset of them)

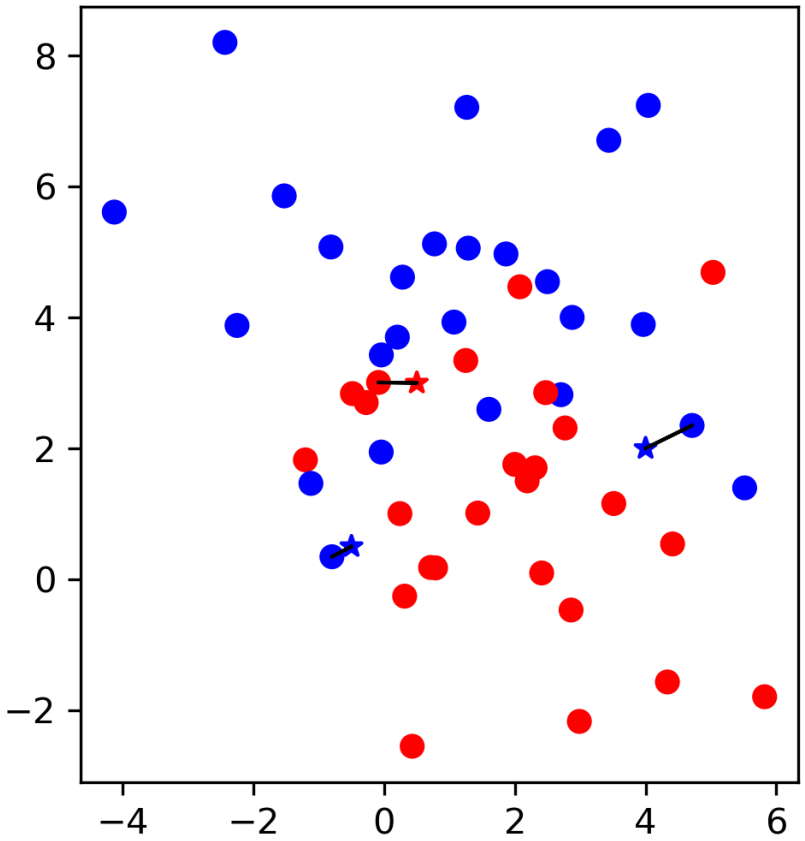


Nearest Neighbors



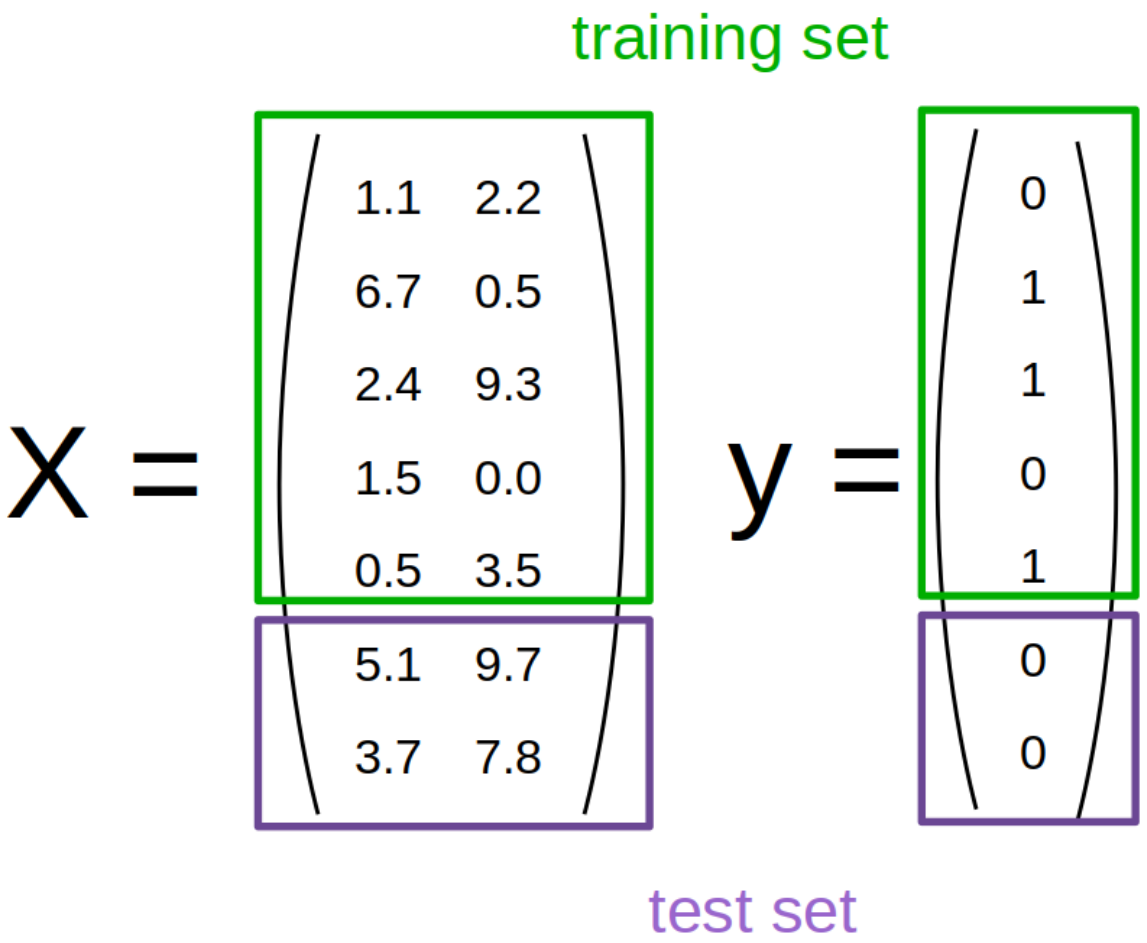
$$f(x) = y_i, i = \text{argmin}_j ||x_j - x||$$

Nearest Neighbors



$$f(x) = y_i, i = \text{argmin}_j ||x_j - x||$$

Splitting Data into Training and Test

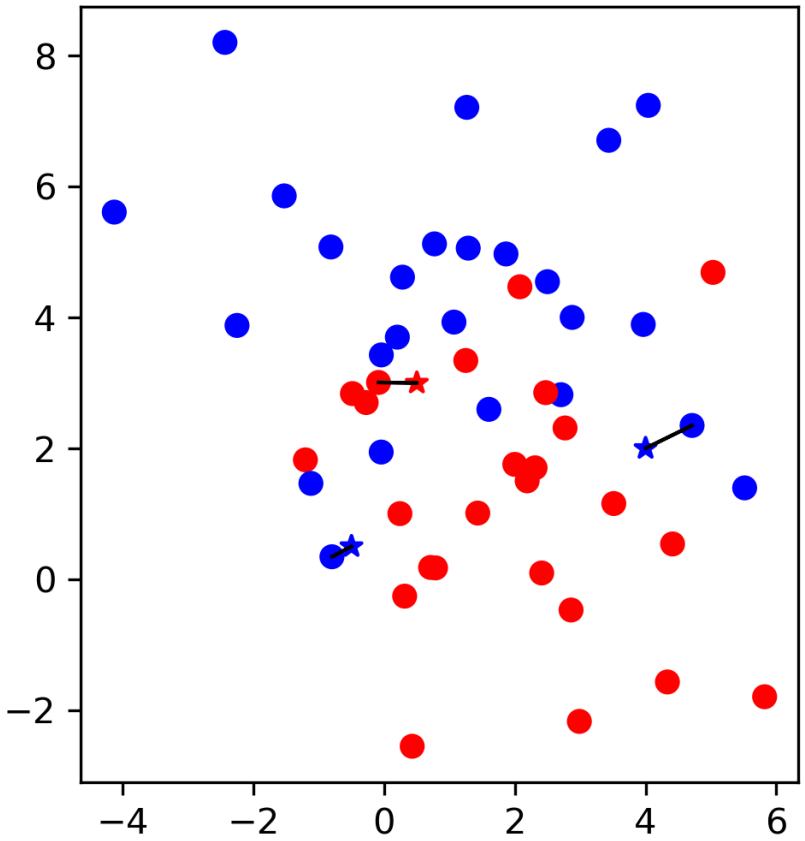


KNN with ScikitLearn

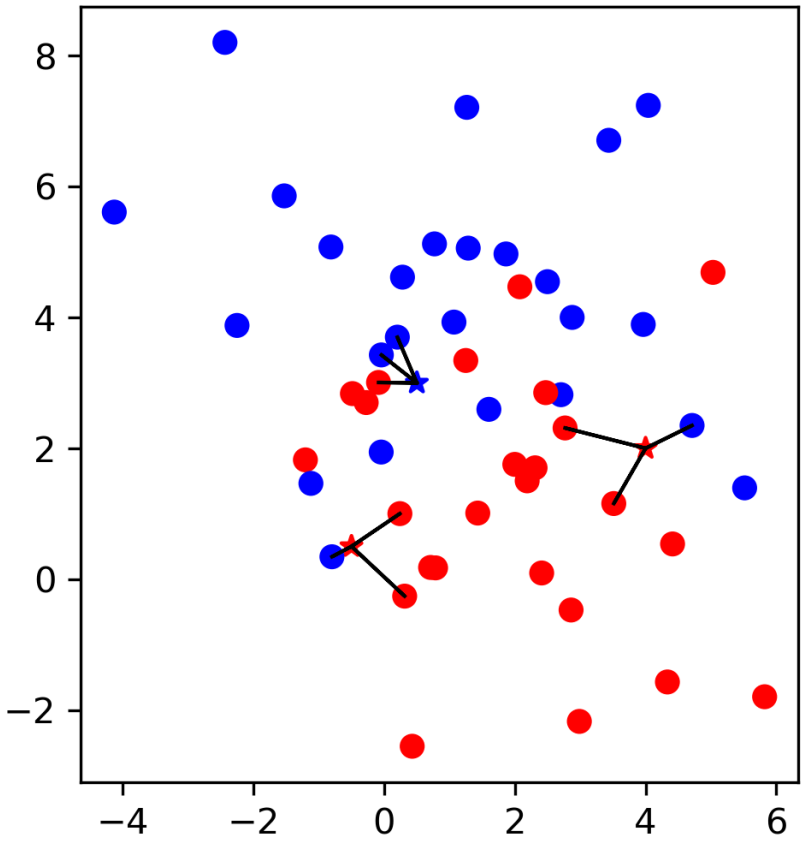
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y)

from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train, y_train)
print("accuracy: ", knn.score(X_test, y_test)))
y_pred = knn.predict(X_test)
```

accuracy: 0.77

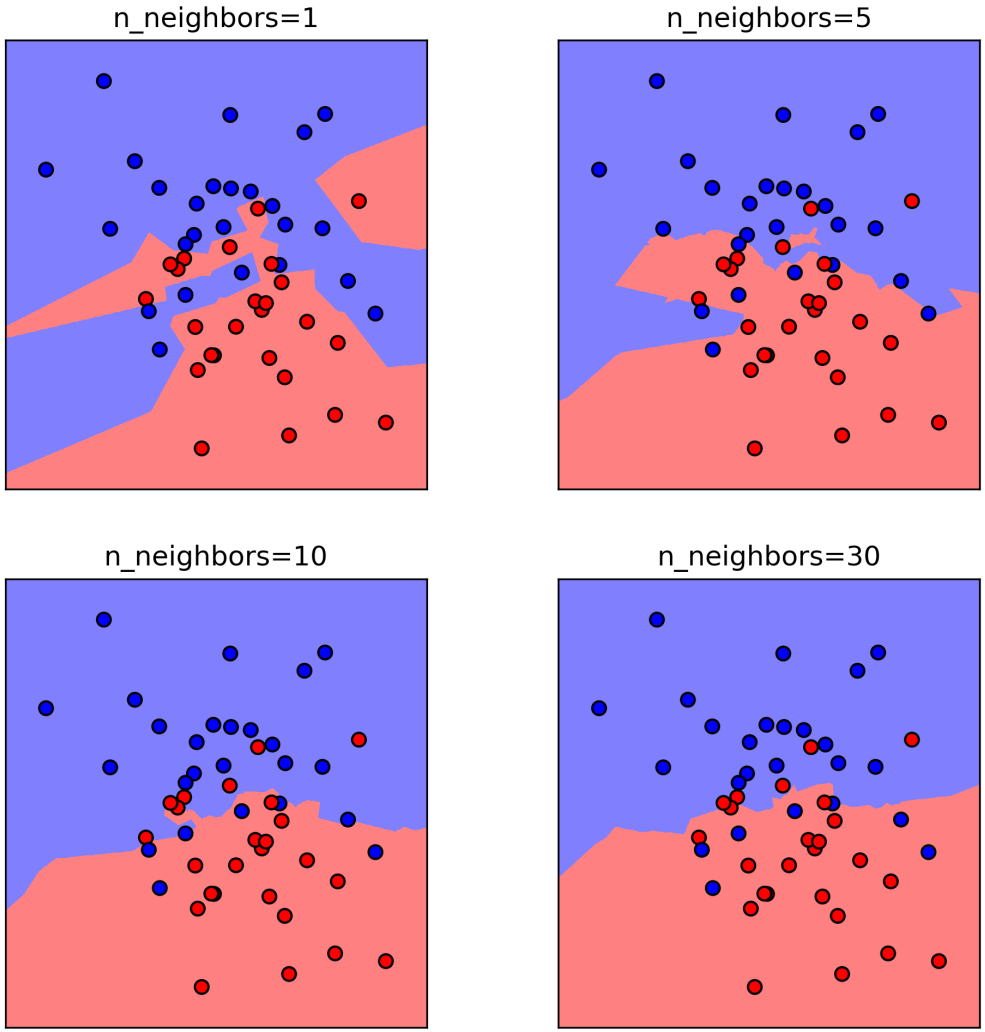
Influence of Number of Neighbors



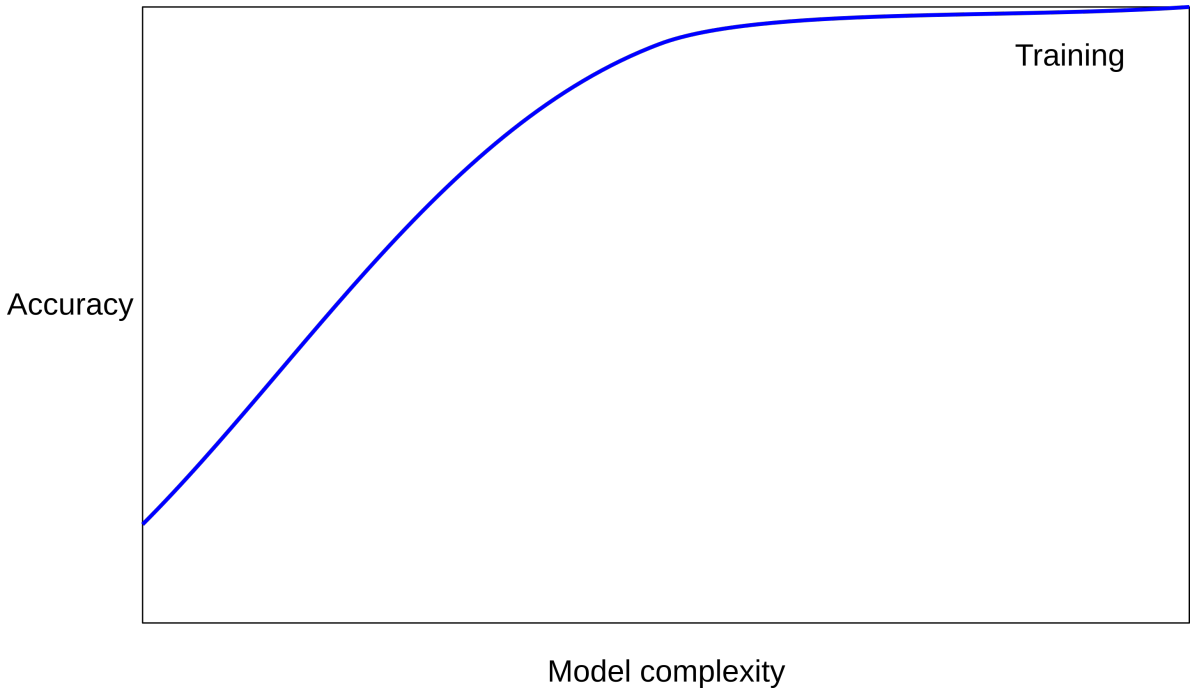
Influence of Number of Neighbors



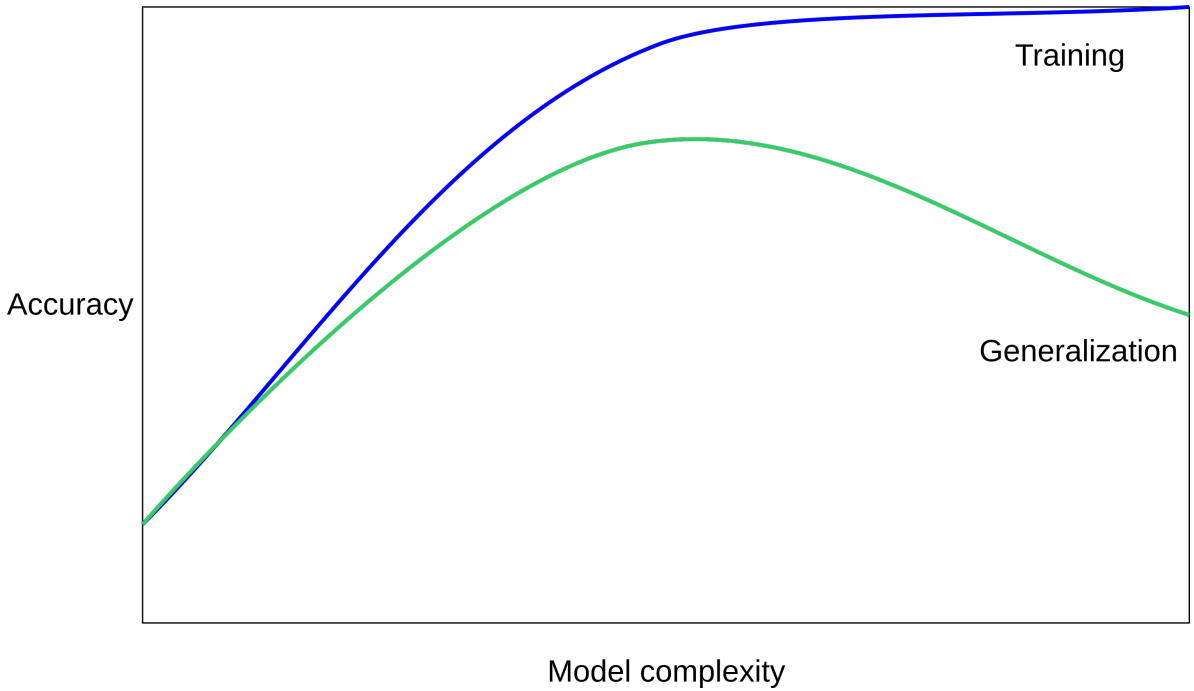
Influence of Number of Neighbors



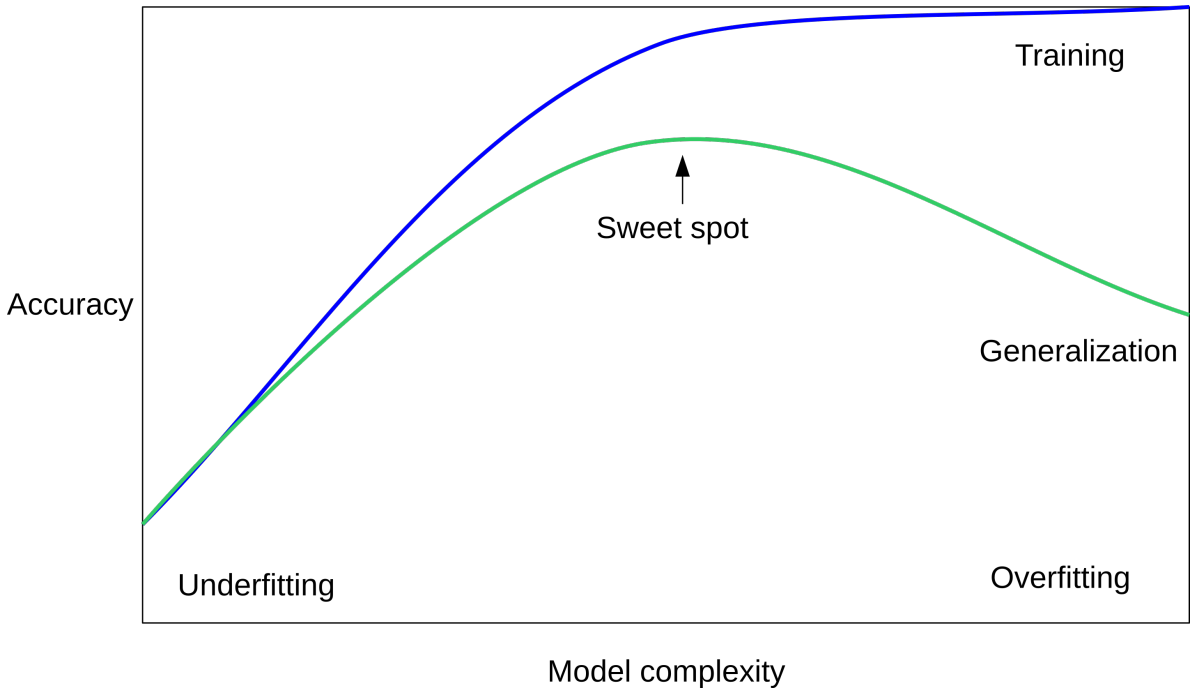
Overfitting and Underfitting



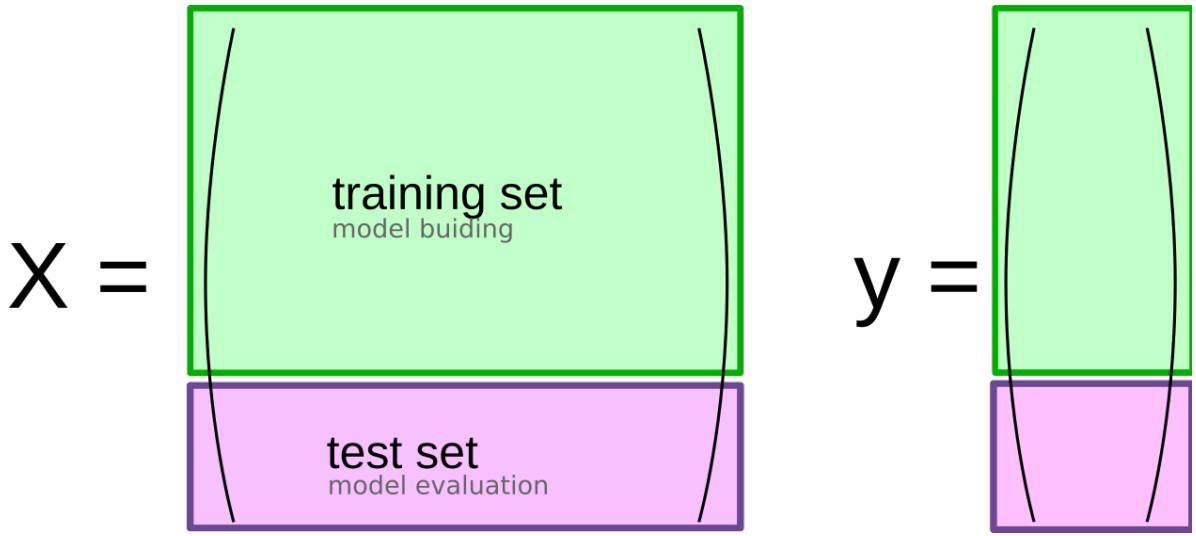
Overfitting and Underfitting



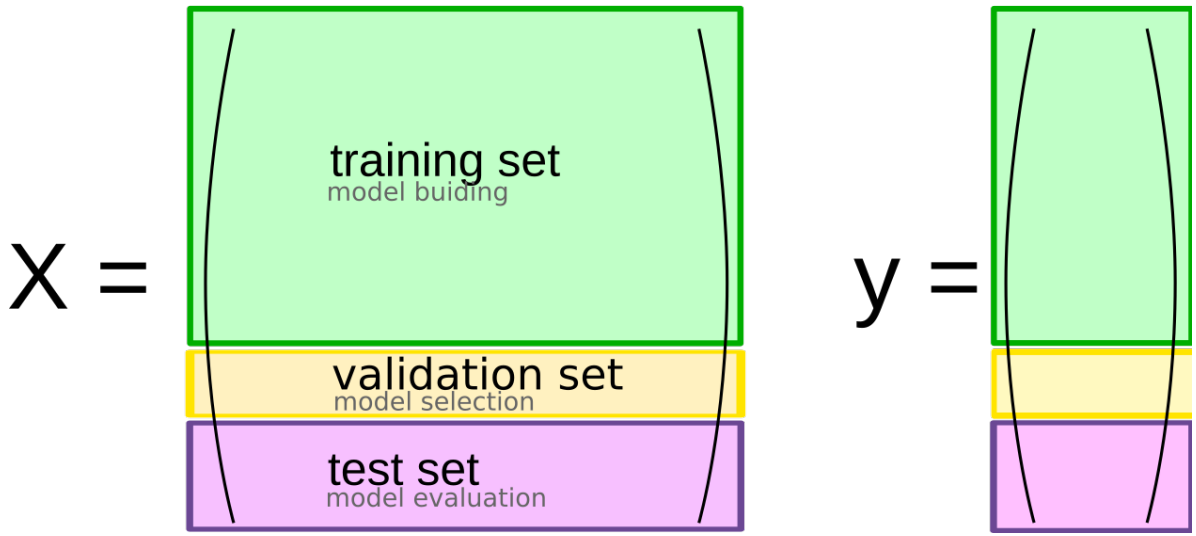
Overfitting and Underfitting



Threefold Split



Threefold Split

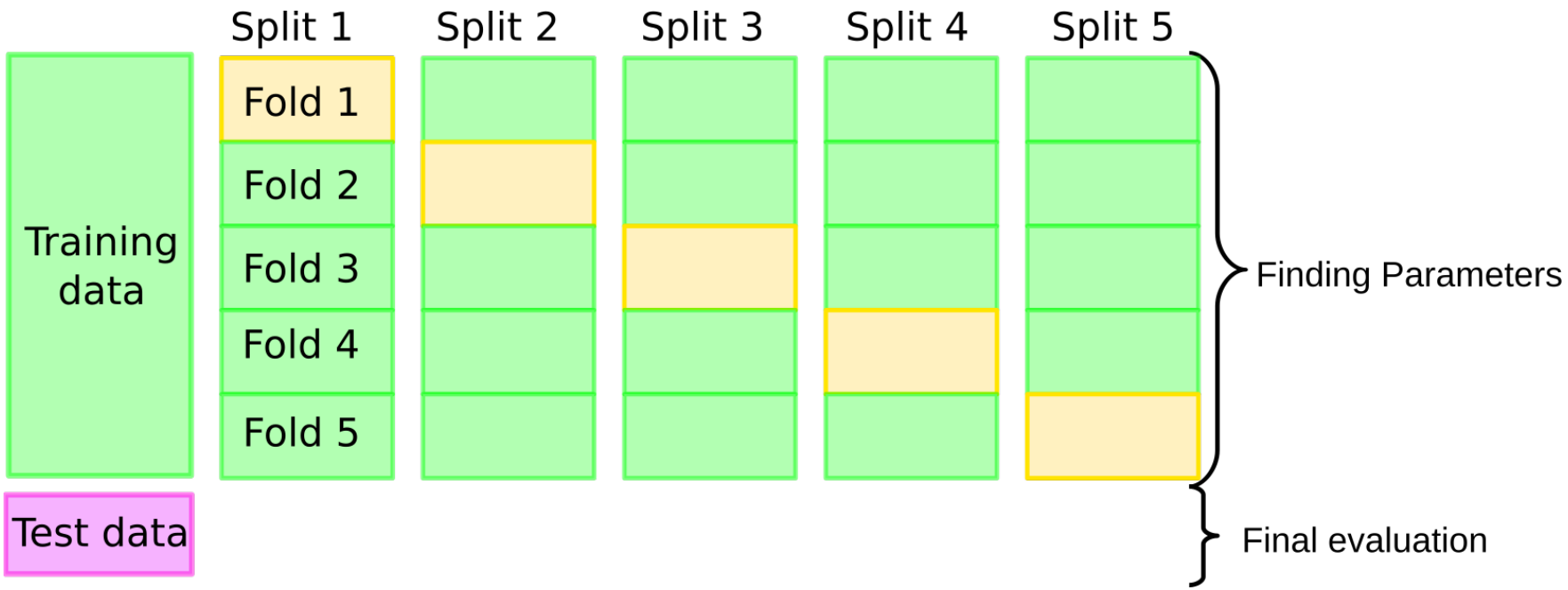


Threefold Split for Hyper-Parameters

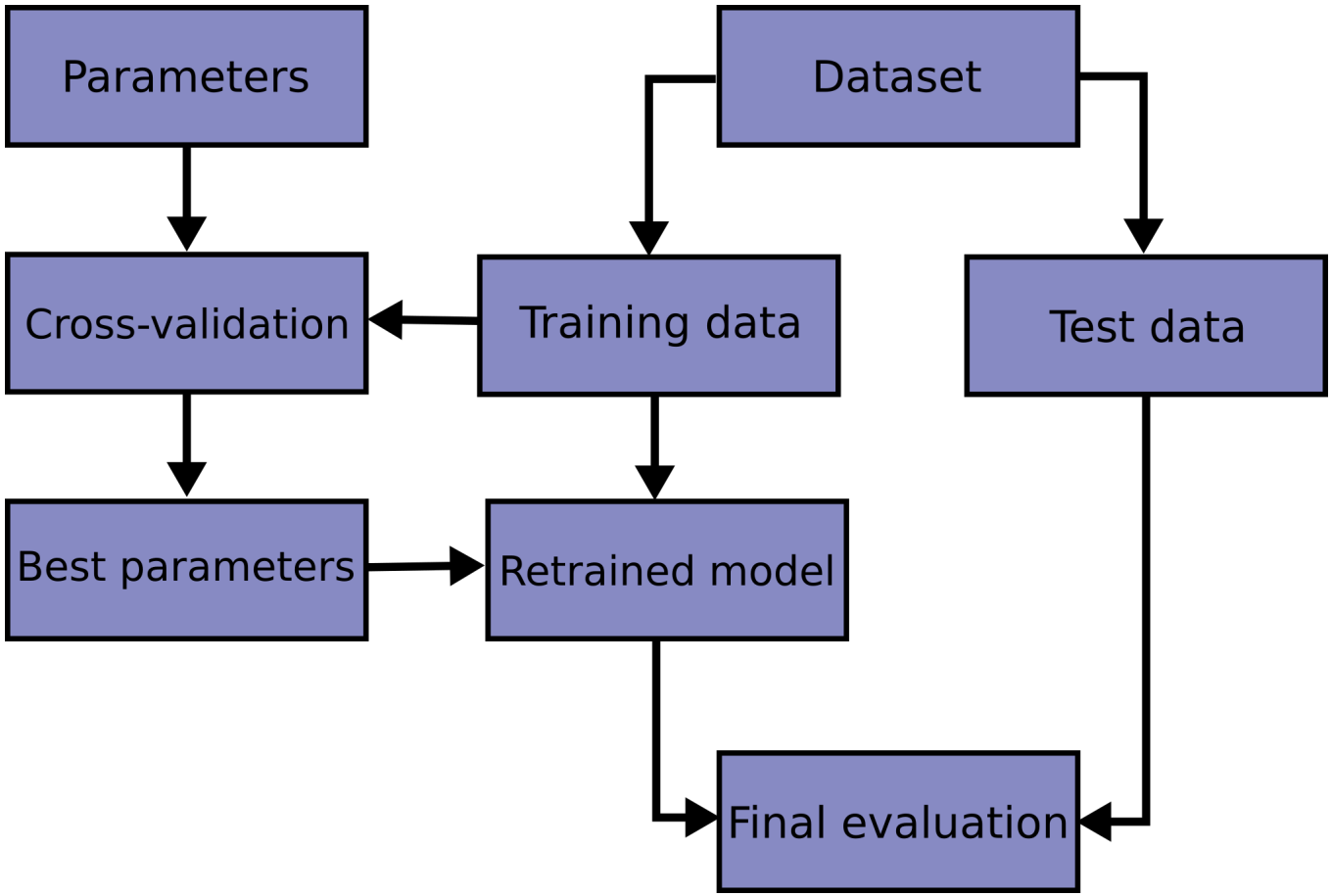
```
X_trainval, X_test, y_trainval, y_test = train_test_split(X, y)
X_train, X_val, y_train, y_val = train_test_split(X_trainval, y_trainval)

val_scores = []
neighbors = np.arange(1, 15, 2)
for i in neighbors:
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train, y_train)
    val_scores.append(knn.score(X_val, y_val))
print(f"best validation score: {np.max(val_scores):.3}")
best_n_neighbors = neighbors[np.argmax(val_scores)]
print("best n_neighbors:", best_n_neighbors)

knn = KNeighborsClassifier(n_neighbors=best_n_neighbors)
knn.fit(X_trainval, y_trainval)
print(f"test-set score: {knn.score(X_test, y_test):.3f}")
```

```
best validation score: 0.991
best n_neighbors: 11
test-set score: 0.951
```

Cross Validation



Grid Search Work flow



GridSearchCV

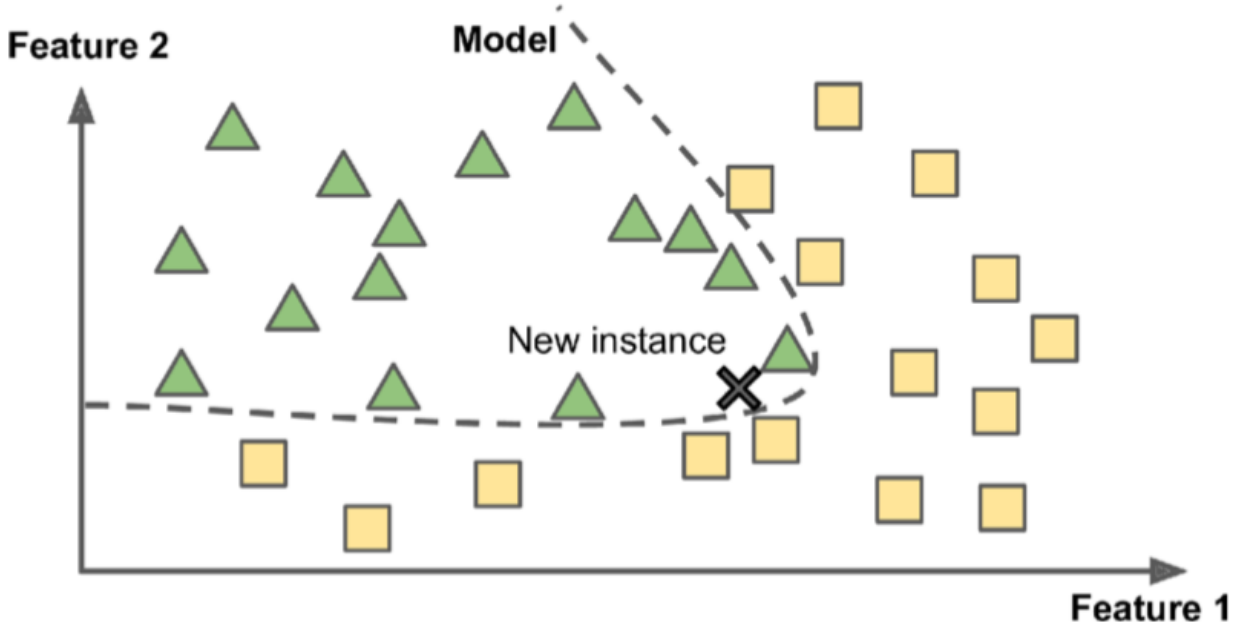
```
from sklearn.model_selection import GridSearchCV

X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y)


param_grid = {'n_neighbors':  np.arange(1, 30, 2)}
grid = GridSearchCV(KNeighborsClassifier(), param_grid=param_grid, cv=10,
                    return_train_score=True)
grid.fit(X_train, y_train)
print(f"best mean cross-validation score: {grid.best_score_}")
print(f"best parameters: {grid.best_params_}")
print(f"test-set score: {grid.score(X_test, y_test):.3f}")
```

```
best mean cross-validation score: 0.967
best parameters: {'n_neighbors': 9}
test-set score: 0.993
```

Model Based Learning

Another way to generalize from a set of examples is to build a model of these examples and then use that model to make predictions.

This is called model-based learning



Linear Regression (1)

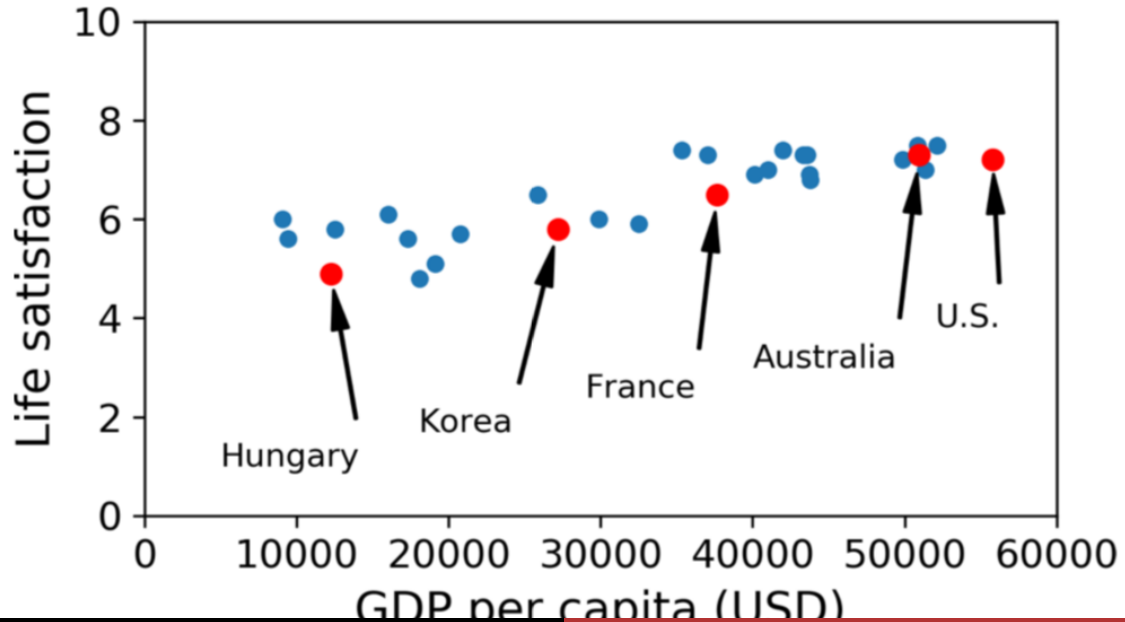
- Suppose you want to know if money makes people happy.
- So you download the Better Life Index data from the OECD's website and stats about gross domestic product (GDP) per capita from the IMF's website.
- Then you join the tables and sort by GDP per capita.

Country	GDP per capita (USD)	Life satisfaction
Hungary	12,240	4.9
Korea	27,195	5.8
France	37,675	6.5
Australia	50,962	7.3
United States	55,805	7.2

Linear Regression (1)

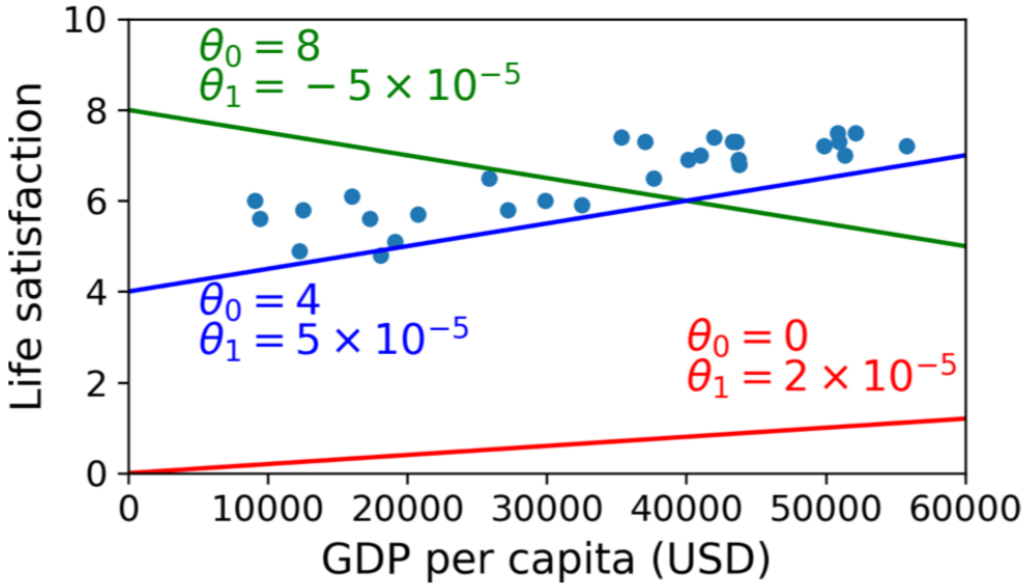
- Suppose you want to know if money makes people happy.
- So you download the Better Life Index data from the OECD's website and stats about gross domestic product (GDP) per capita from the IMF's website.
- Then you join the tables and sort by GDP per capita.

$$life\_satisfaction = \theta_0 + \theta_1 GDP\_per\_capita$$



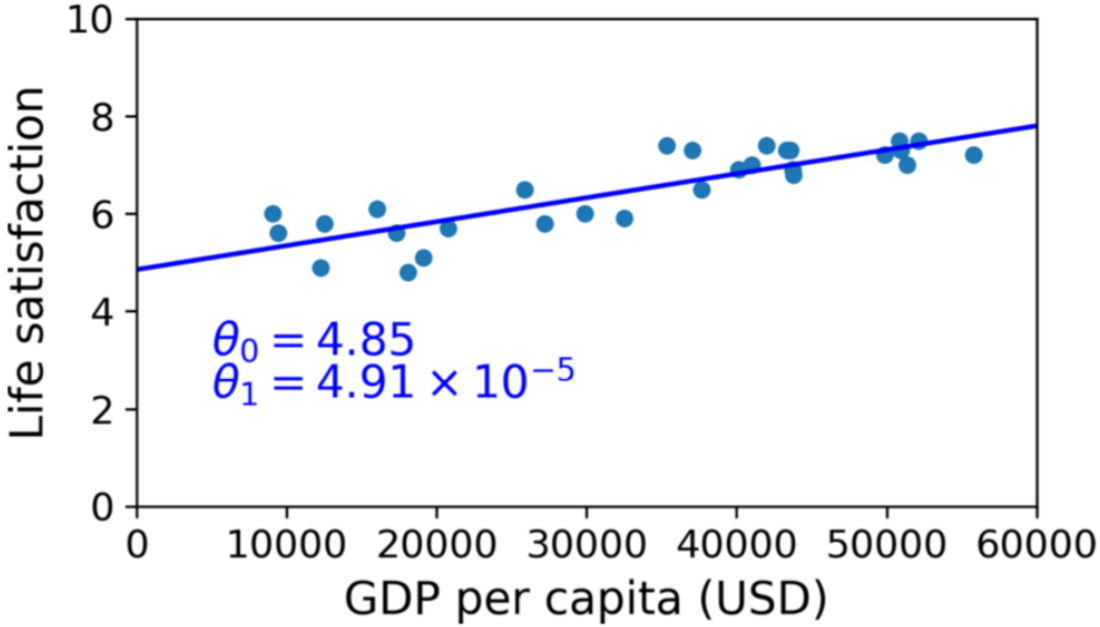
Linear Regression (2)

- This model has two model parameters, $_0$ and $_1$.
- By tweaking these parameters, you can make your model represent any linear function



Linear Regression (2)

- Before you can use your model, you need to define the parameter values θ_0 and θ_1.
- How can you know which values will make your model perform best?
  - Utility?
  - Cost?



Python Code for GDP Example

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import sklearn.linear_model

# Load the data
oecd_bli = pd.read_csv("oecd_bli_2015.csv", thousands=',')
gdp_per_capita = pd.read_csv("gdp_per_capita.csv",thousands=',',delimiter='\t',
                             encoding='latin1', na_values="n/a")
```

Python Code for GDP Example

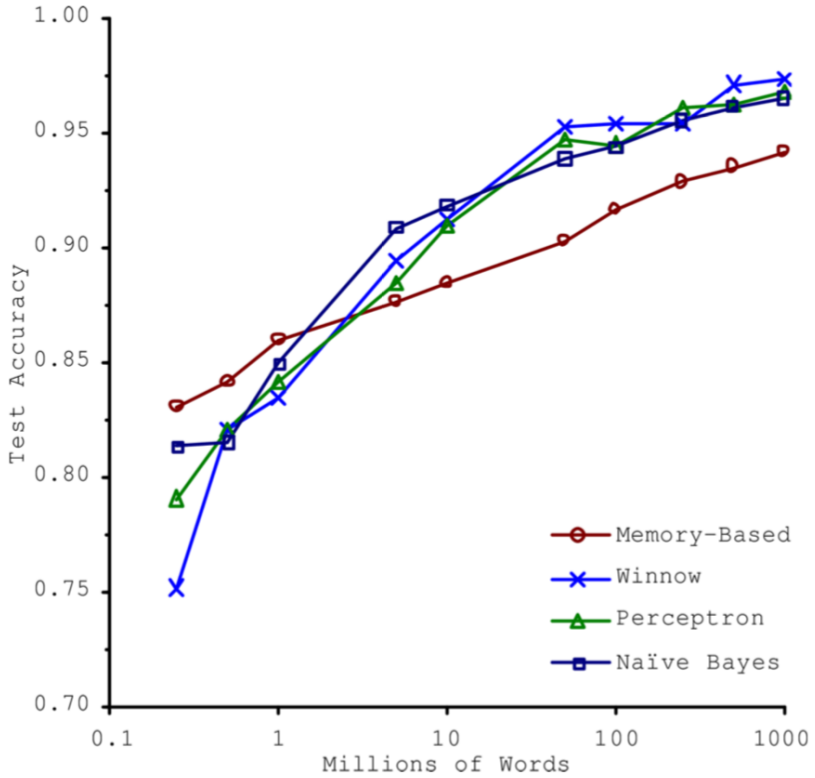
```
# Prepare the data
country_stats = prepare_country_stats(oecd_bli, gdp_per_capita)
X = np.c_[country_stats["GDP per capita"]]
y = np.c_[country_stats["Life satisfaction"]]

# Visualize the data
country_stats.plot(kind='scatter', x="GDP per capita", y='Life satisfaction')
plt.show()
```

Python Code for GDP Example

```python
# Select a linear model
model = sklearn.linear_model.LinearRegression()

# Train the model
model.fit(X, y)

# Make a prediction for Cyprus
X_new = [[22587]]  # Cyprus's GDP per capita
print(model.predict(X_new)) # outputs [[ 5.96242338]]
```

Main Challenges–Again!

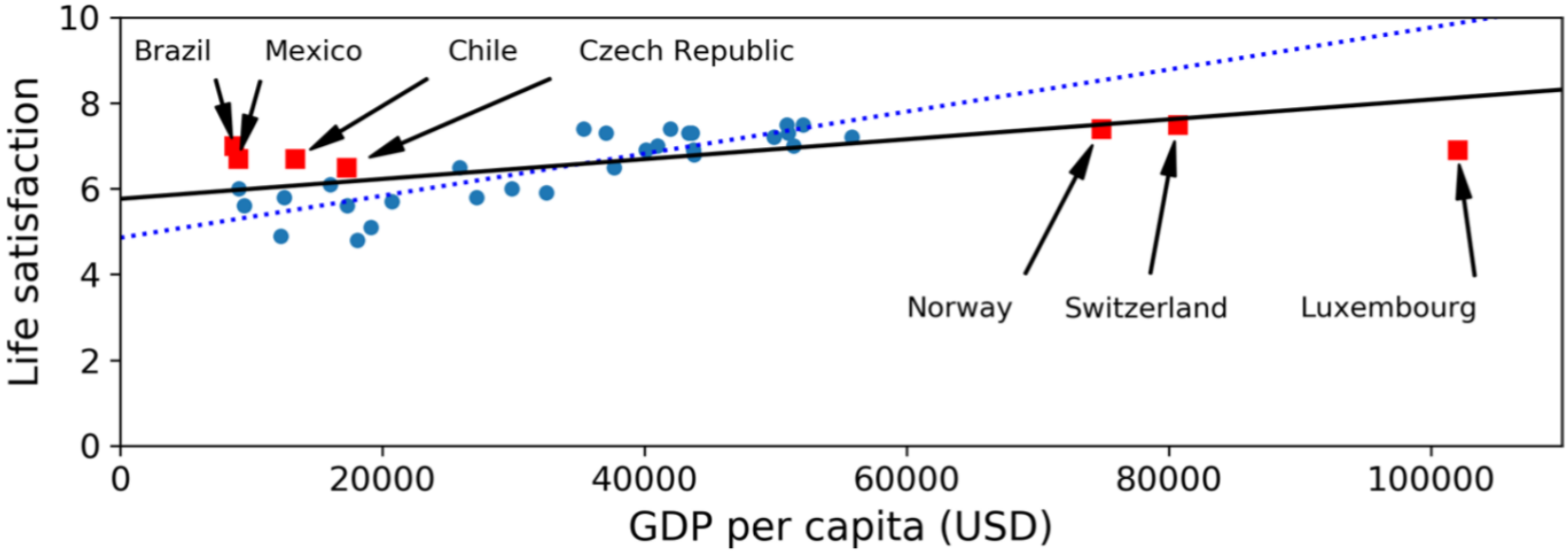
Insufficient Quantity of Training Data



"these results suggest that we may want to reconsider the trade- off between spending time and money on algorithm development versus spending it on corpus development."

Main Challenges–Again!

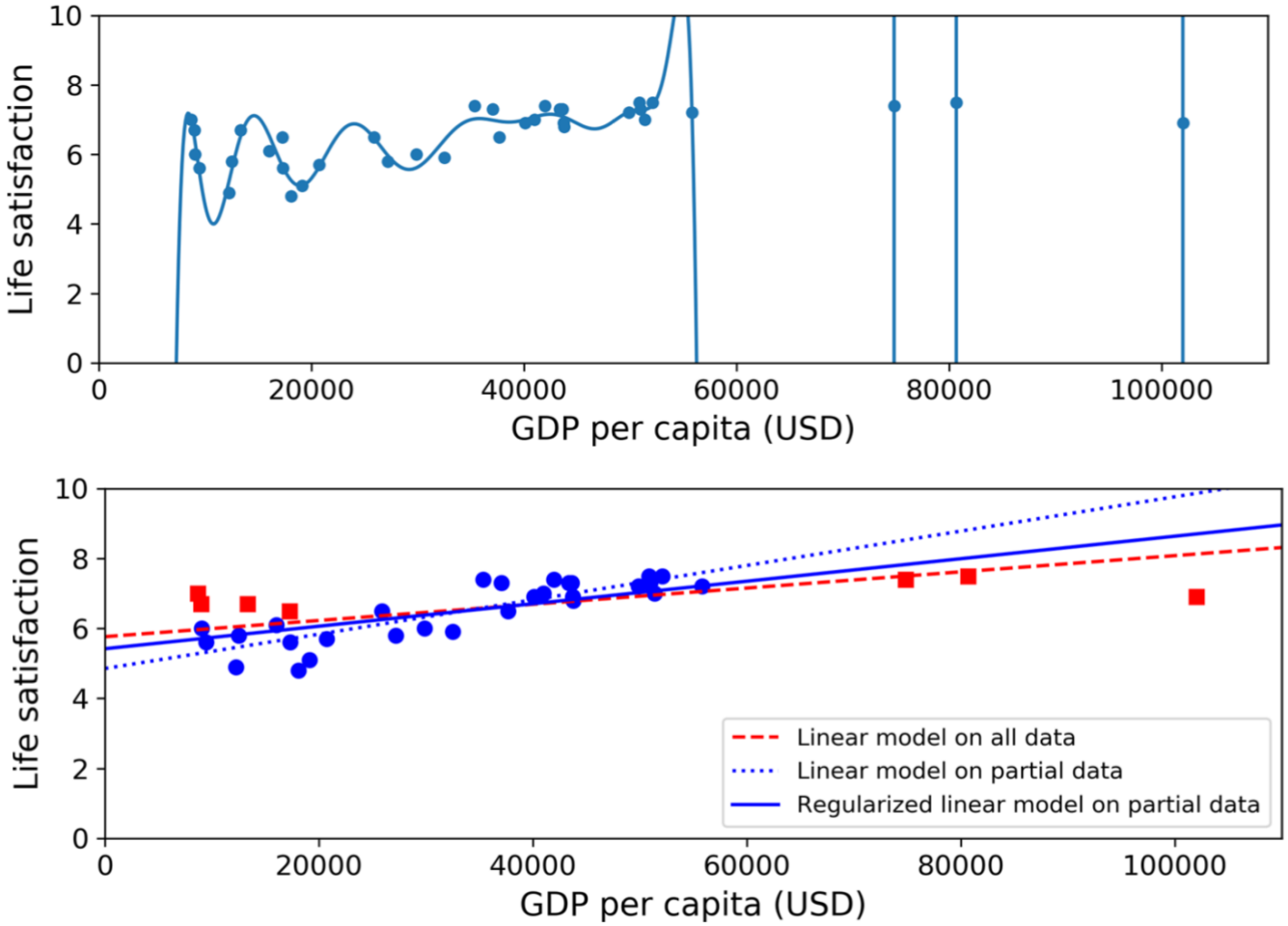
Nonrepresentative Training Data



- If the sample is too small, you will have sampling noise (i.e., nonrepresentative data as a result of chance).
- But even very large samples can be nonrepresentative if the sampling method is flawed. This is called sampling bias.

Main Challenges–Again!

Overfitting the Training Data and Regularization



Main Challenges–Again!

- Underfitting
  - Select a more powerful model, with more parameters.
  - Feed better features to the learning algorithm (feature engineering)
  - Reduce the constraints on the model (e.g., reduce the regularization hyperpara- meter).
- Poor-Quality Data
  - Outliers
  - Missing Data
- Irrelevant Features
  - Feature Selection
  - Feature Extraction

Summary

- Machine Learning is about making machines get better at some task by learning from data, instead of having to explicitly code rules.
- Machine Learning is used mainly for prediction
- There are two broad categories of ML algorithms: Supervised and Unsupervised Models
- To cope with over-fitting, use cross validation
- Hyper-parameter tuning is an important component of ML pipeline