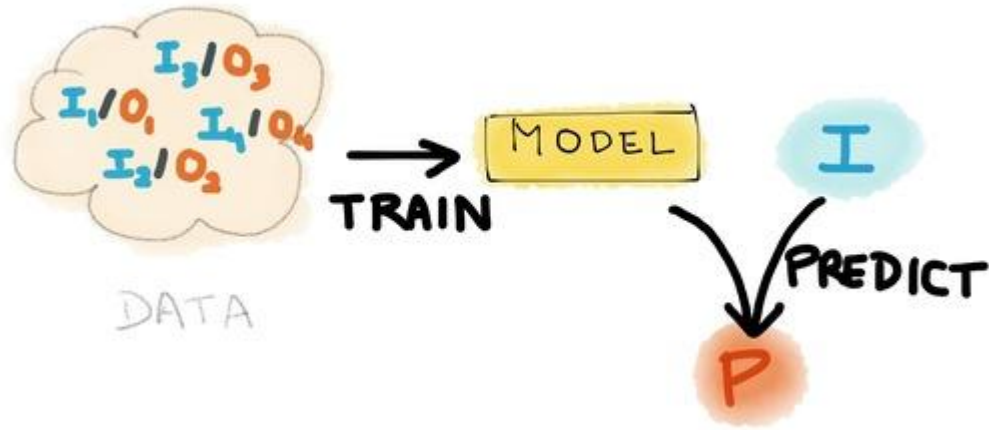# Machine Learning

Machine learning using scikit-learn
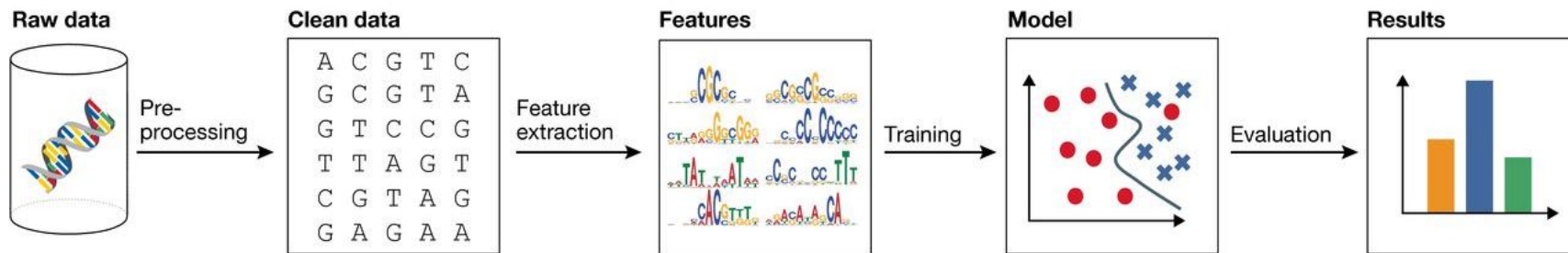
Contents

- What is ML
- Pipeline steps
- Learning?
- Many ML models
- Evaluating
- Example
- Hackathon suggestions

What is machine learning?

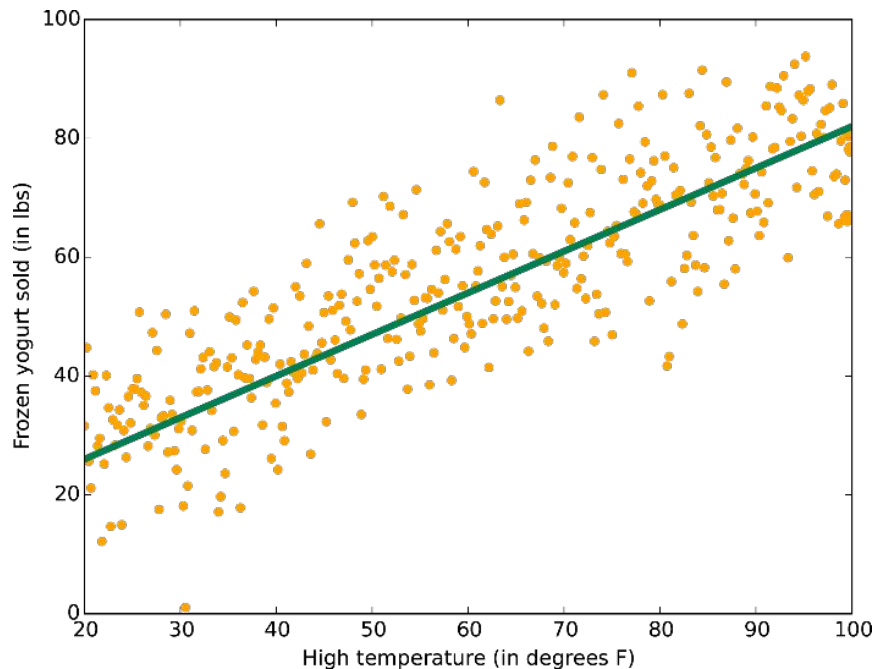# Machine learning pipeline

Raw data → Pre-processing → **Clean data** → Feature extraction → **Features** → Training → **Model** → Evaluation → **Results**

Clean data:
```
A C G T C
G C G T A
G T C C G
T T A G T
C G T A G
G A G A A
```

Cleaning data

- Clean data
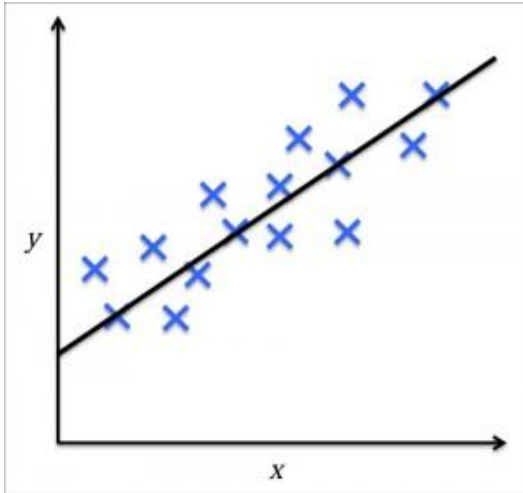 - Fill missing values
  - Drop incomplete rows

Learning a function
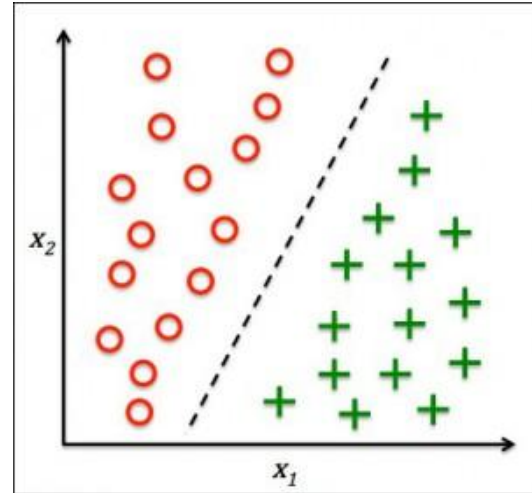
- Learn function $f(x) \rightarrow y$

Types of algorithms

Regression
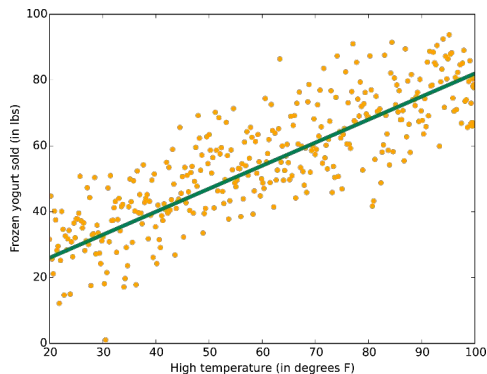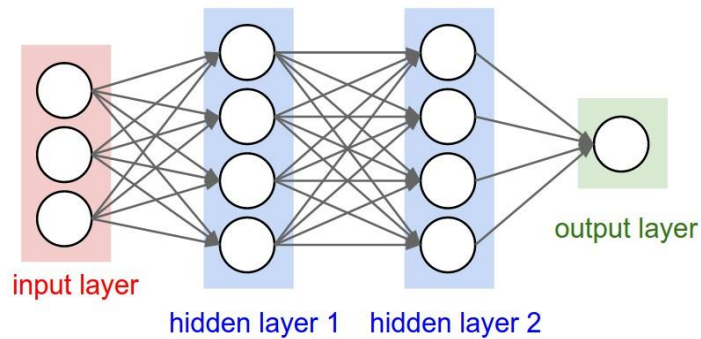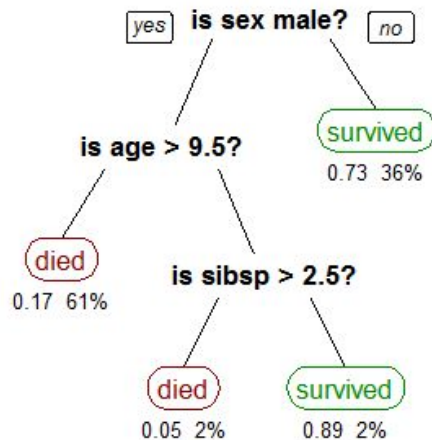

Classification

Types of algorithms

- Different approaches (models)

How does learning work?

- y = *Intercept* + *Slope**x
  - Learn *Intercept*, *Slope*

- Loss - e.g. MSE

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

Regression line errors
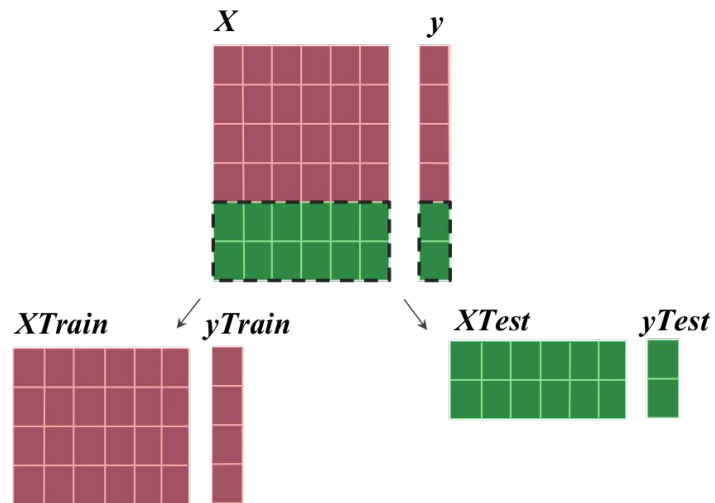
Legend: Regression line, Errors, Samples

How does learning work?

- Training : Gradient descent
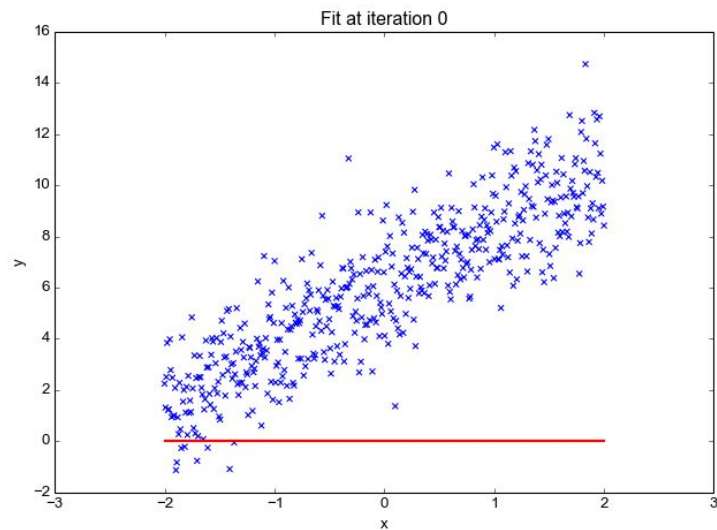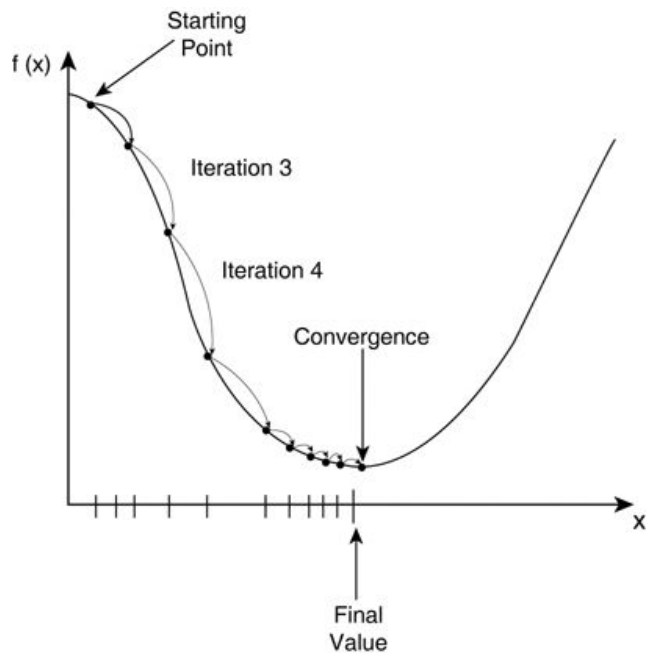
Evaluating

Simple: train/test split

Better: KFold Cross validation

What could possibly go wrong?

Underfitting   Just right!   overfitting

Fixed data size

Mean Error

Cross-validation error

Training error

High bias   High variance

Model Complexity
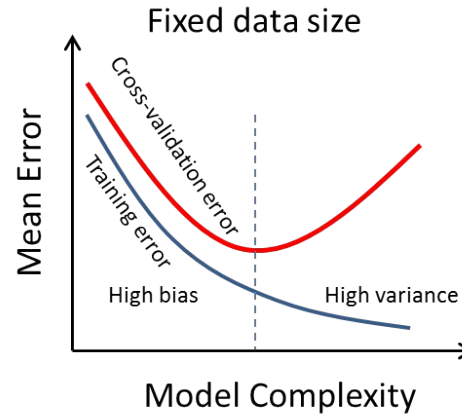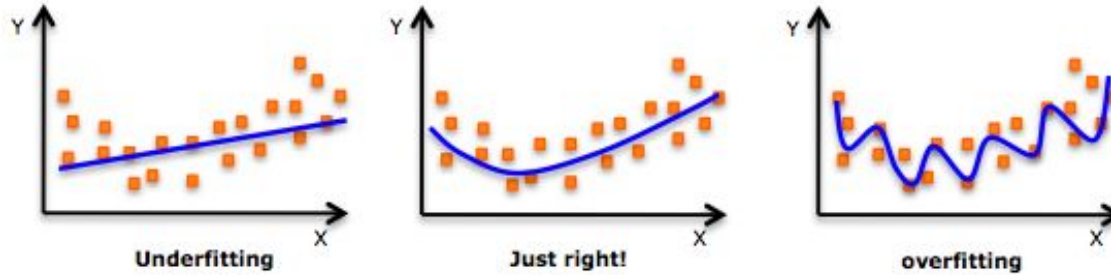
Python & scikit-learn

```python
from sklearn import datasets
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X,y = datasets.load_iris(return_X_y=True)

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

model = DecisionTreeClassifier(min_samples_split=8, min_samples_leaf=4)

model.fit(X_train, y_train)

y_predicted = model.predict(X_test)

print("Score: %.4f " % accuracy_score(y_test, y_predicted))
```

Example

- [Analysis, features & model NYC taxi trips](#)

Hackathon suggestions

- Find research questions / hypotheses
- Prepare data & engineer features
- Train & evaluate a model
- Compare models
- Learn & improve !

Questions ?

- Let's start hacking !