

INFO5002: Intro to Python for Info Sys

Machine Learning

DSfs 153-163



Northeastern
University

Model

- Aims to tell a relationship between variables.
- Is imperfect.

ML — model fitting

- Machine Learning is the process of fitting models to given data to minimise a given loss function.
- Can think of fitting a linear line on a scatterplot.
- Every model has different **parametres** and these are learned from the data provided.

Types of training

- Supervised: give data and labels.
- Unsupervised: give data no labels.
- Semi-supervised: some data has labels.
- Online: keep learning as new data comes in.
- Reinforcement: use feedback on performance to update.

How to train

- We split our data into training and test datasets ($\approx 2:1$).
- Fit our model to the training data.
- Test the accuracy of our model on the test dataset.

```
import random

def split(xs: list, ys: list, percent: float):
    assert len(xs) == len(ys)

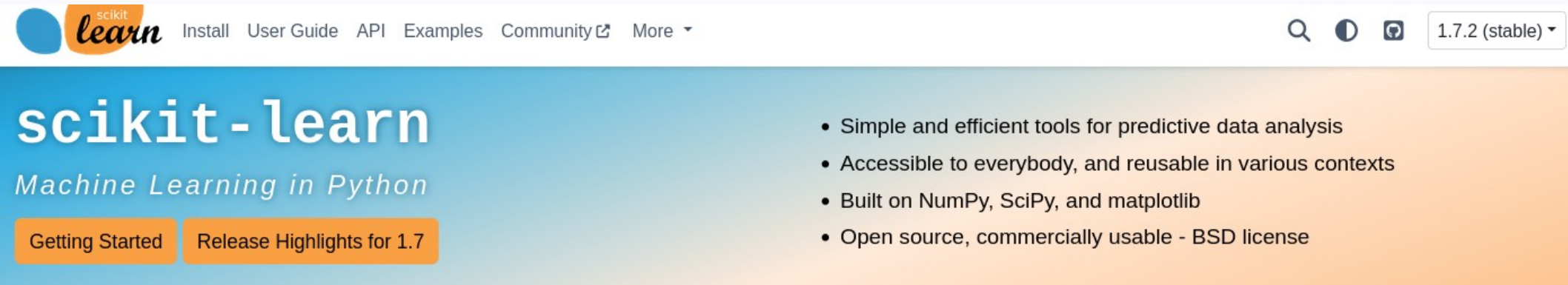
    idxs = [i for i in range(len(xs))]
    random.shuffle(idxs)

    idx = int(len(idxs) * percent)
    train_idx = idxs[:idx]
    test_idx = idxs[idx:]

    return (
        [xs[i] for i in train_idx],
        [xs[i] for i in test_idx],
        [ys[i] for i in train_idx],
        [ys[i] for i in test_idx]
    )

x_train, x_test, y_train, y_test = split(x, y, 0.8)
```

Is there a library?



The screenshot shows the top portion of the scikit-learn website. On the left is the scikit-learn logo. To its right is a navigation menu with links for 'Install', 'User Guide', 'API', 'Examples', 'Community' (with an external link icon), and 'More' (with a dropdown arrow). On the far right of the navigation bar are icons for search, a dark mode toggle, and a GitHub repository icon, followed by a version selector showing '1.7.2 (stable)' with a dropdown arrow. Below the navigation bar is a blue header section containing the text 'scikit-learn' in large white font, 'Machine Learning in Python' in a smaller white font, and two orange buttons: 'Getting Started' and 'Release Highlights for 1.7'. To the right of this header is an orange section with a bulleted list of features.

scikit-learn

Install User Guide API Examples Community [↗](#) More ▾

1.7.2 (stable) ▾

scikit-learn

Machine Learning in Python

Getting Started Release Highlights for 1.7

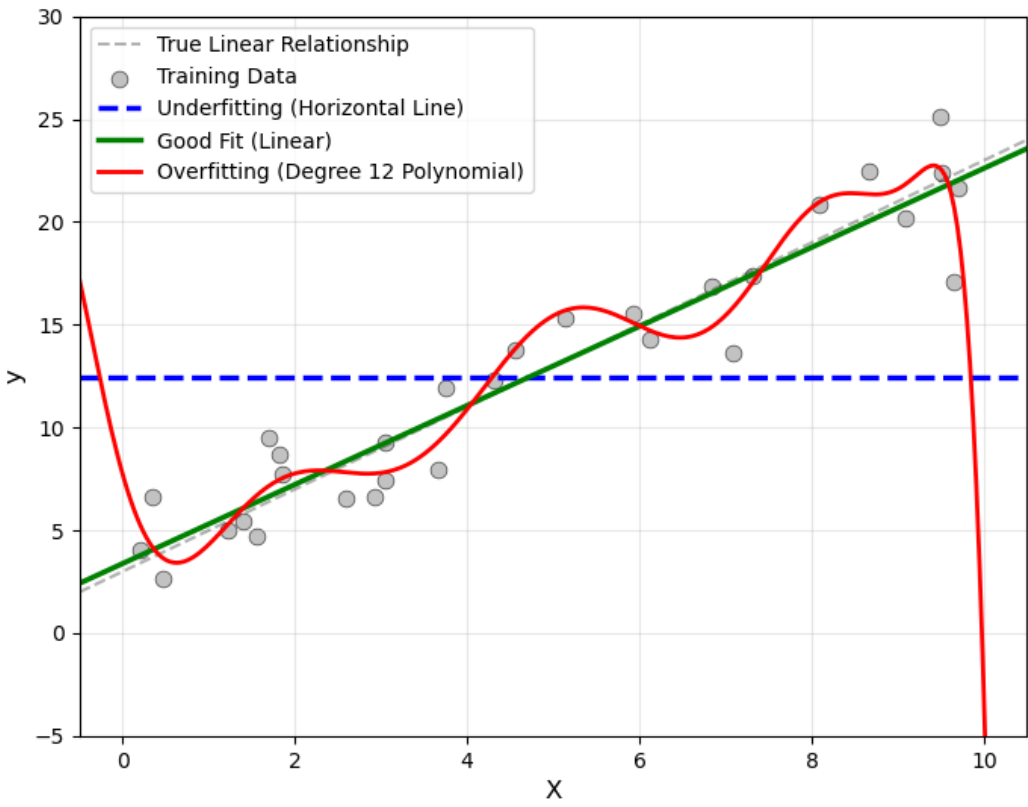
- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

- Scikit-learn implements many data science tools so that you do not need to re-implement from scratch.

```
from sklearn.model_selection import train_test_split  
  
X_train, X_test, y_train, y_test = train_test_split(  
    X, y, test_size=0.33, random_state=42)
```

Dangers of ML

- Underfitting: where the model performs poorly on our data.
- Overfitting: where the model performs very well on our provided data but fails when given new data (poor generalisation).



Training Set

What if I evaluate multiple models

- If you fit multiple models on the train data and then choose the model that performs best on the test data, you are **meta-training**.
- Test set restricted to performance **only!**
- Split data into train (for training), validation (choosing best model), test (to see performance).

Measuring performance in binary classification

- True Positive: Predict pos, is pos.
- False Positive (Type I error): Predict pos, is neg.
- False Negative (Type II error): Predict neg, is pos.
- True Negative: Predict neg, is neg.

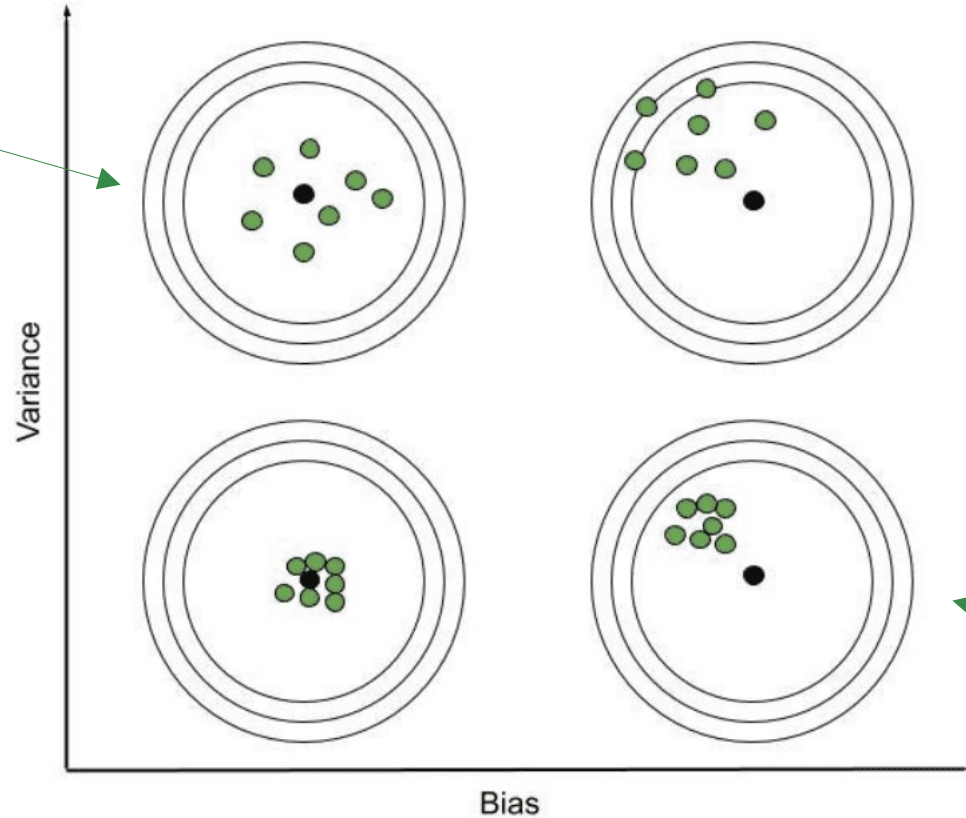
Can create a confusion matrix

	Pos	Neg
Predict pos	TP	FP
Predict neg	FN	TN

- Accuracy: $(TP + TN) / \text{total}$
- Precision: $TP / (TP + FP)$
- Recall (TPR): $TP / (TP + FN)$
- FPR: $FP / (FP + TN)$
- F1-score: $2 * p * r / (p + r)$

Bias vs Variance

overfitting



underfitting

Evaluating non-binary

- Accuracy: $\# \text{ correct} / \text{total}$
- Classification: giving the right label.
- Regression: within a given range from correct value.

Features

- Variables you give your model to predict an output.
- You should choose features carefully based on domain expertise.
- Sometimes features are not provided and you must extract it from some data.
- Sometimes you have too many features and can reduce it with **dimensionality reduction**.