Introduction to tsfresh

AKLab, 22.03.2024, ABI House

Andreas W. Kempa-Liehr

Department of Engineering Science and Biomedical Engineering

ENGINEERING

22.03.2024 – AKLab

Outline

Applications of tsfresh

Activity recognition

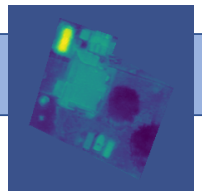
Sensor data

Non-destructive measurement

Spectra

Image classification

Satelite Images

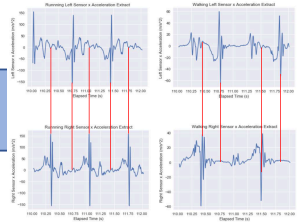


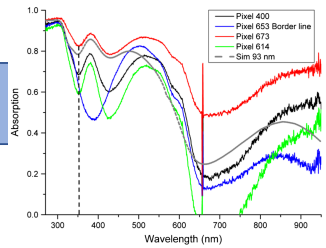
Authorship Attribution

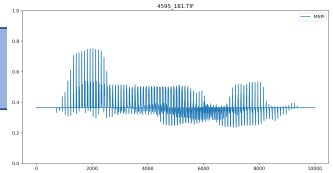
Short Text Samples

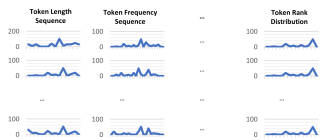
Text	Class
This process, however, afforded me...	EAP
It never once occurred to me that...	HPL
...	...
He laid a gnarled claw on my shoulder...	HPL

Time-series Engineering







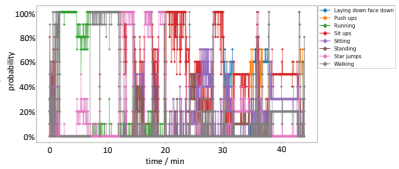


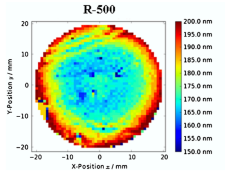
tsfresh

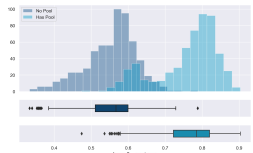
Systematic Feature Extraction

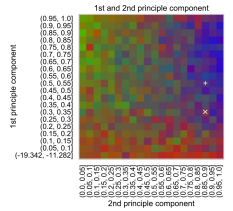
Feature Selection

Supervised Machine Learning



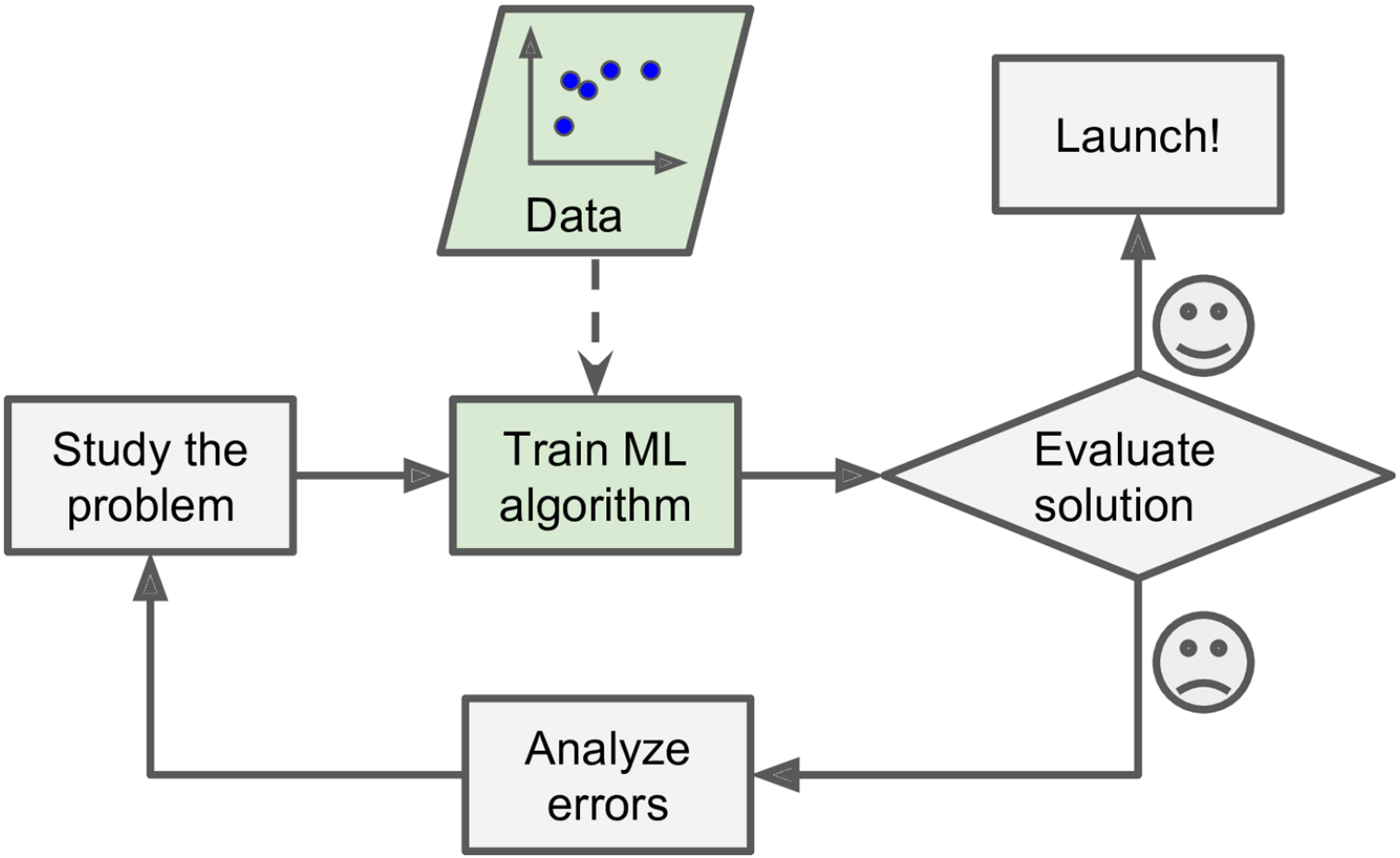






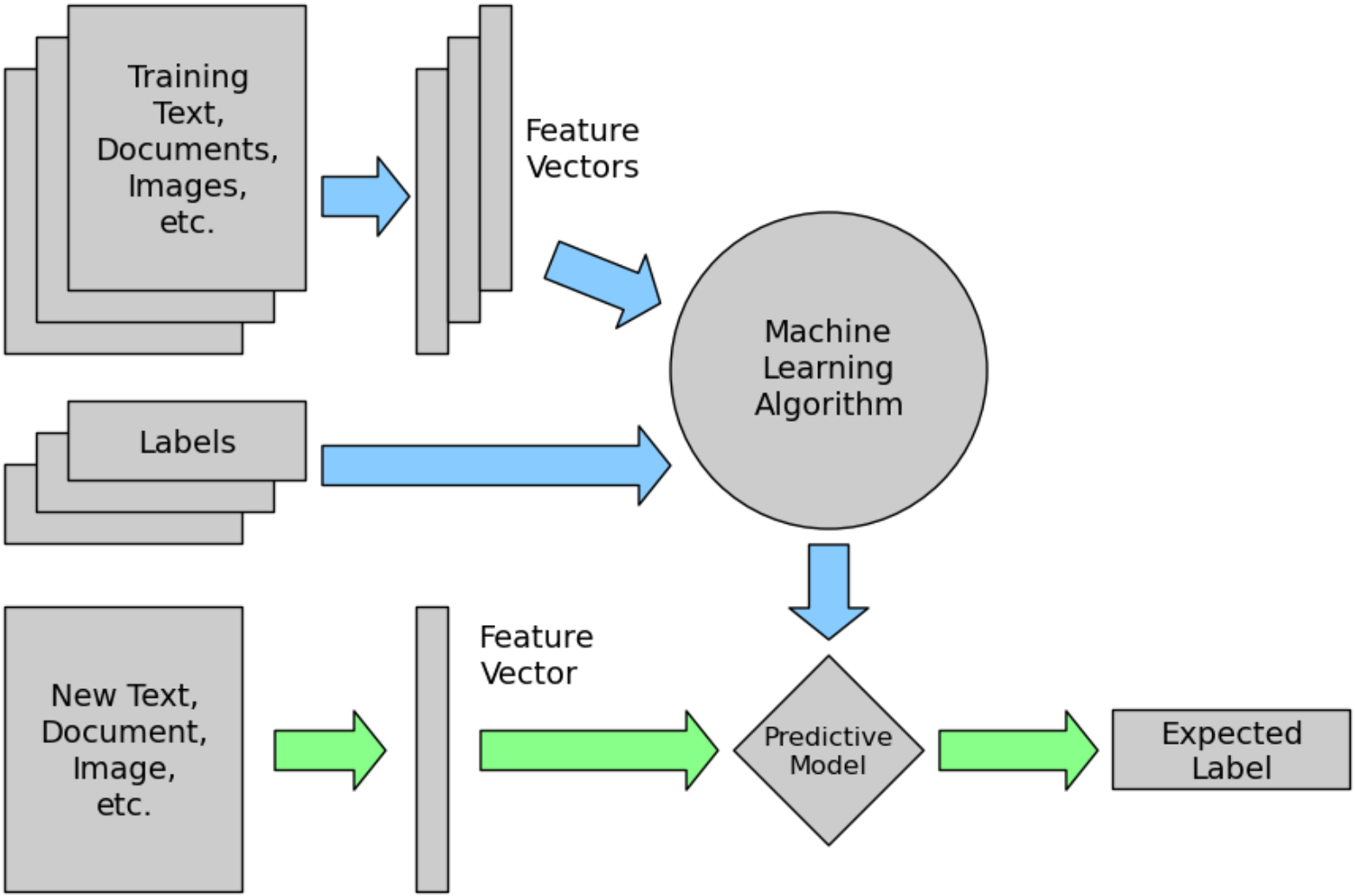
Machine learning

Training a ML algorithm is also called *fitting*.



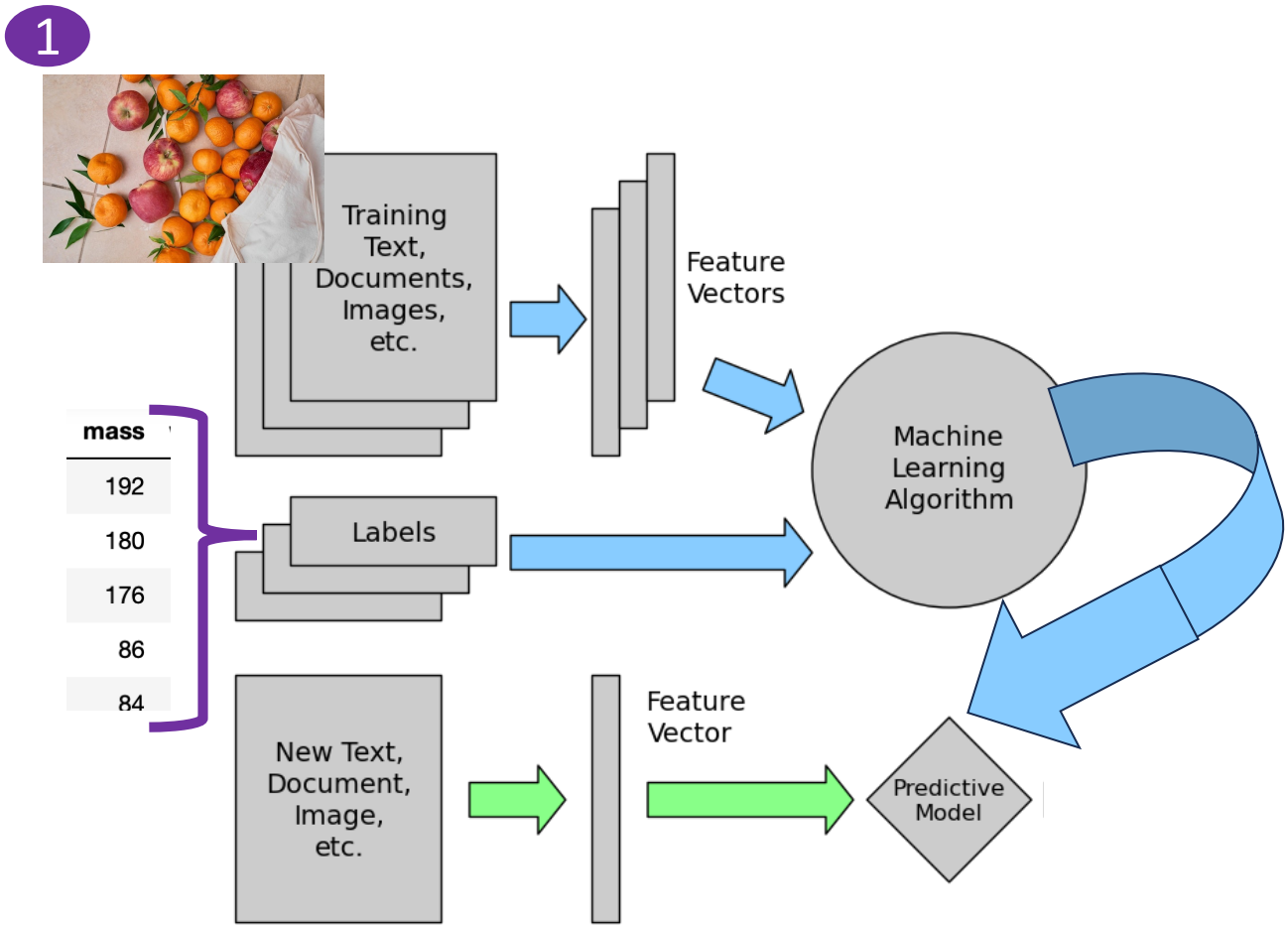
Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. 4th release. Sebastopol, CA, United States: O'Reilly Media, 2017, Fig. 1-2

Flowchart of supervised learning

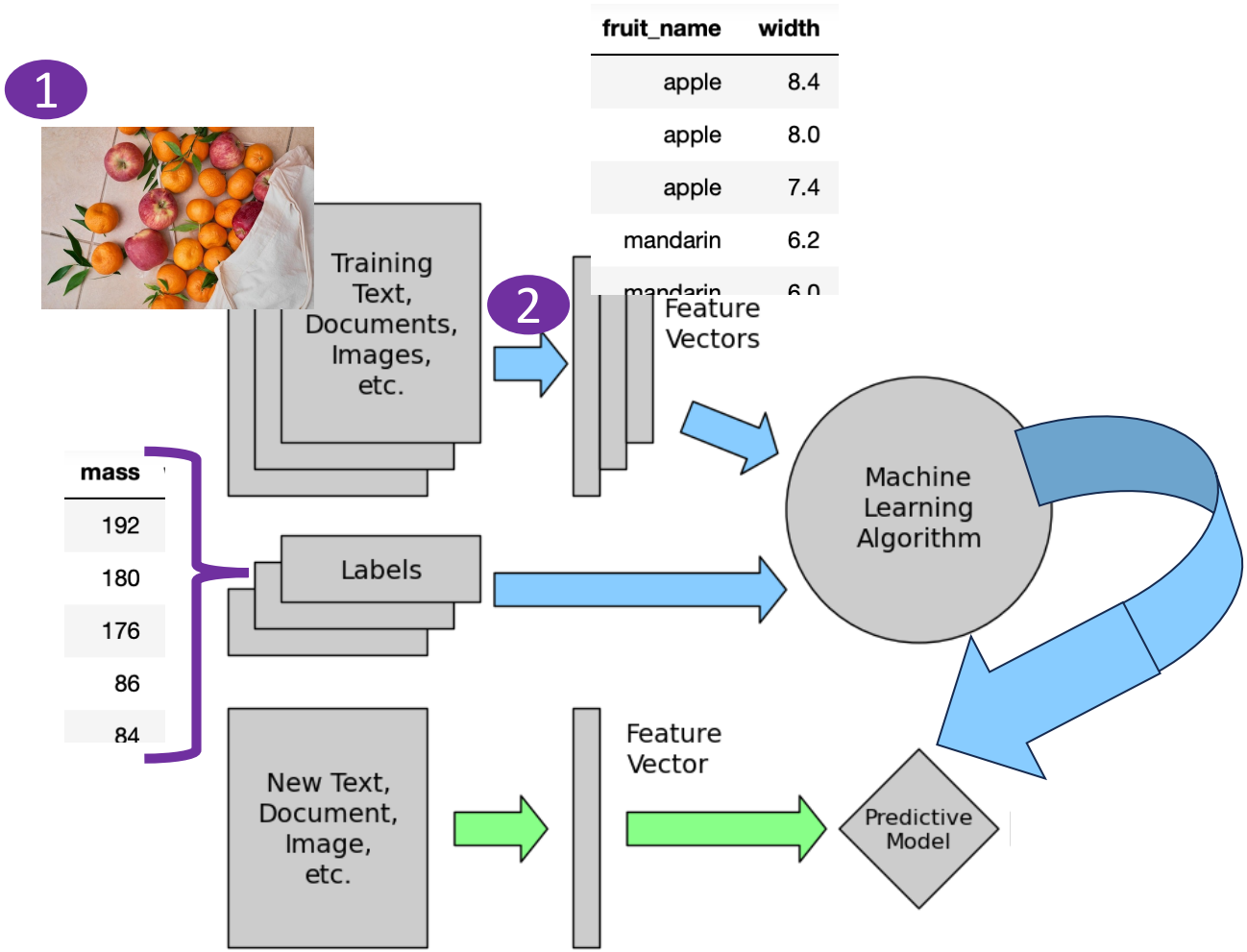


Jakob VanderPlas. *Astronomy with scikit-learn*. Release Scipy2012. Seattle: University of Washington, 2012. URL: http://astroML.github.com/sklearn_tutorial

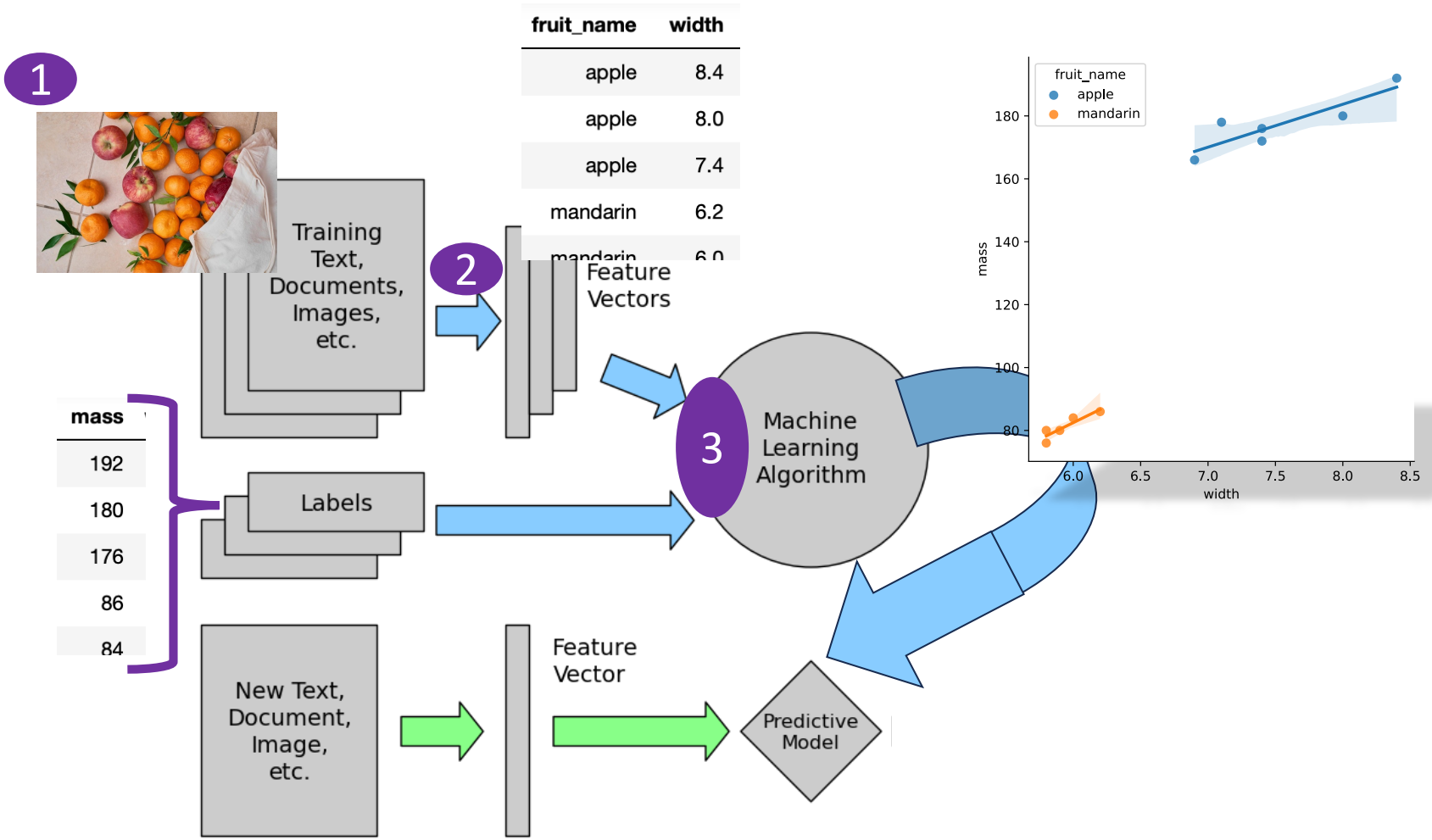
Machine learning in action - (1) Data collection



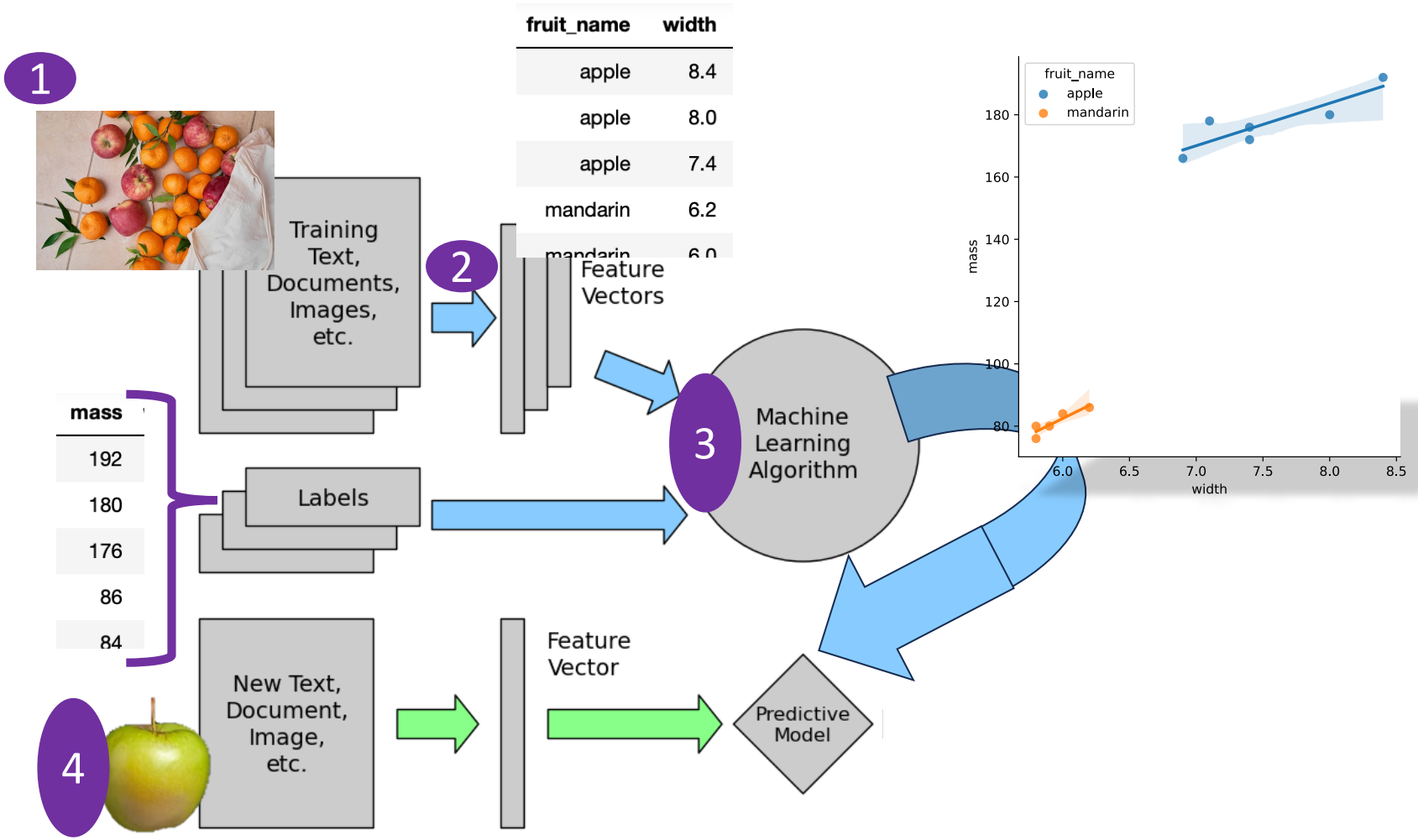
Machine learning in action - (2) Feature extraction



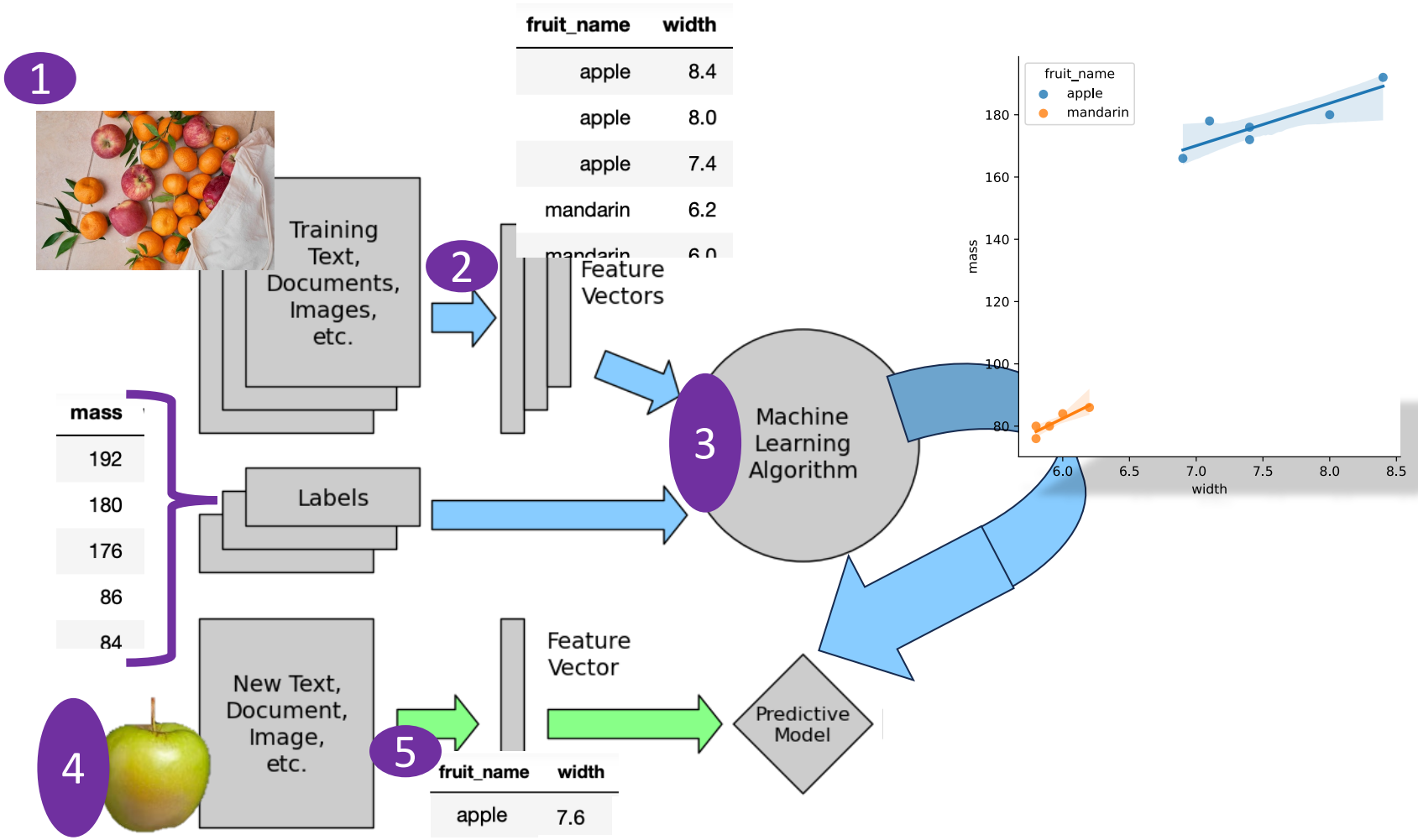
Machine learning in action - (3) Model fitting/training



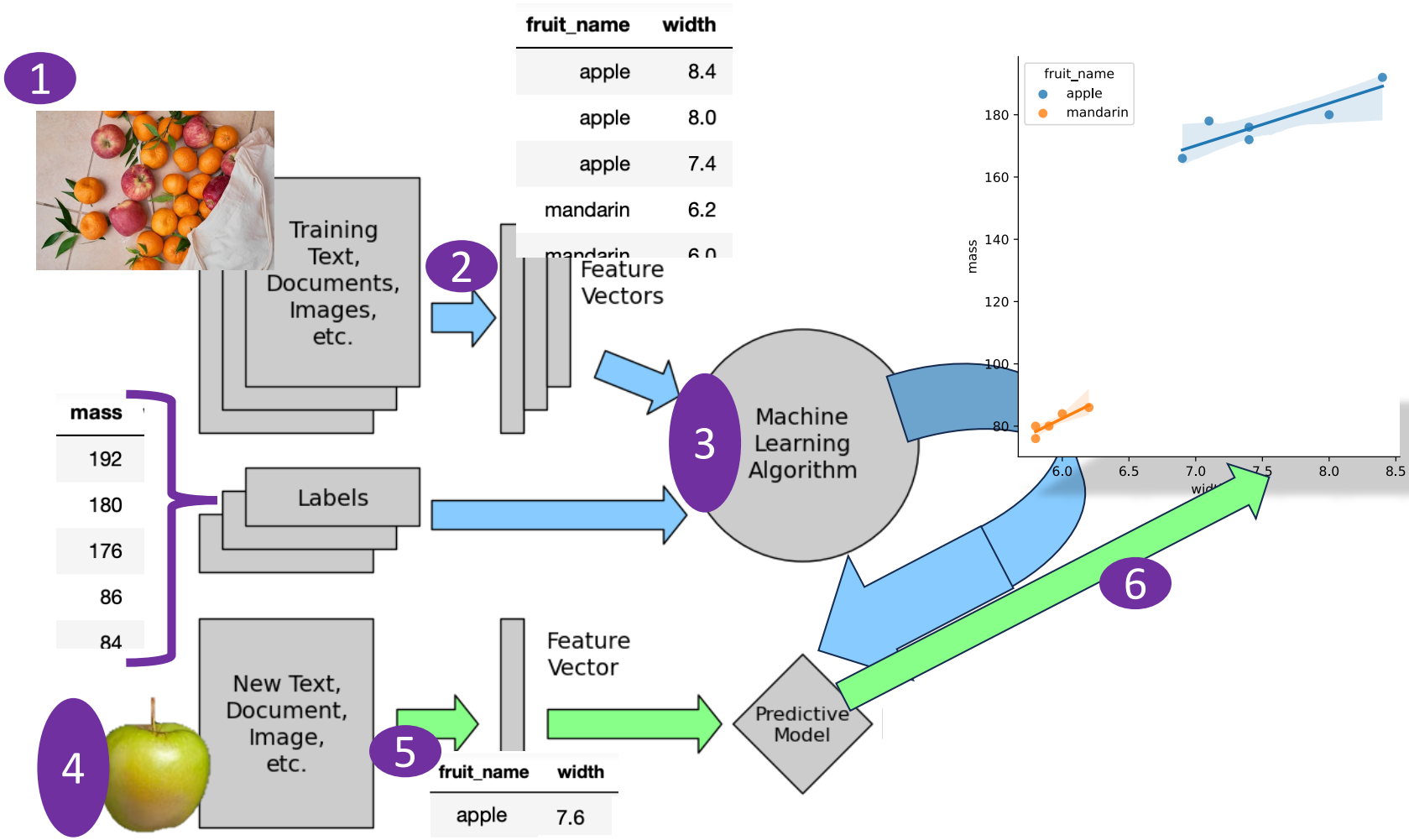
Machine learning in action - (4) New observation



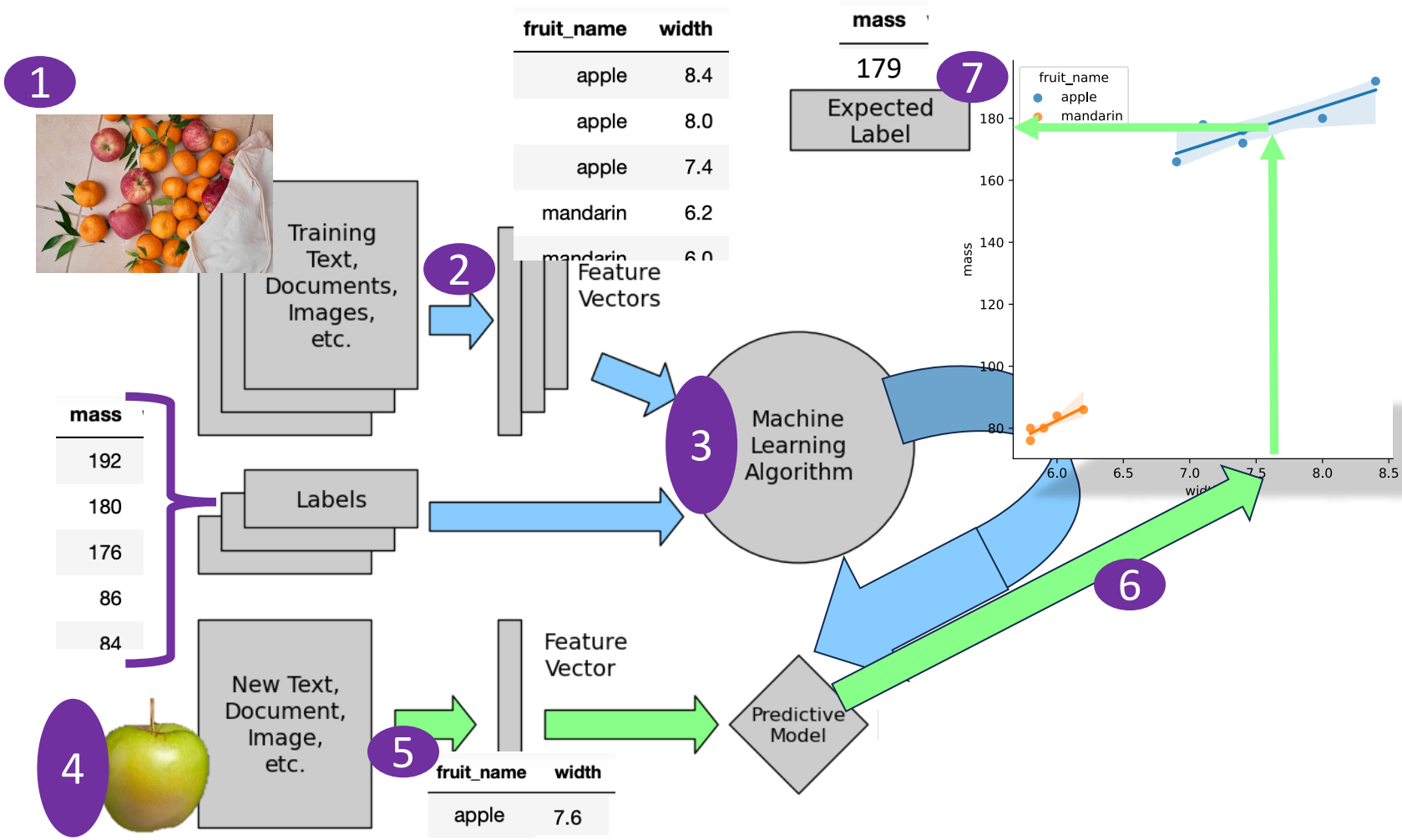
Machine learning in action - (5) Feature extraction



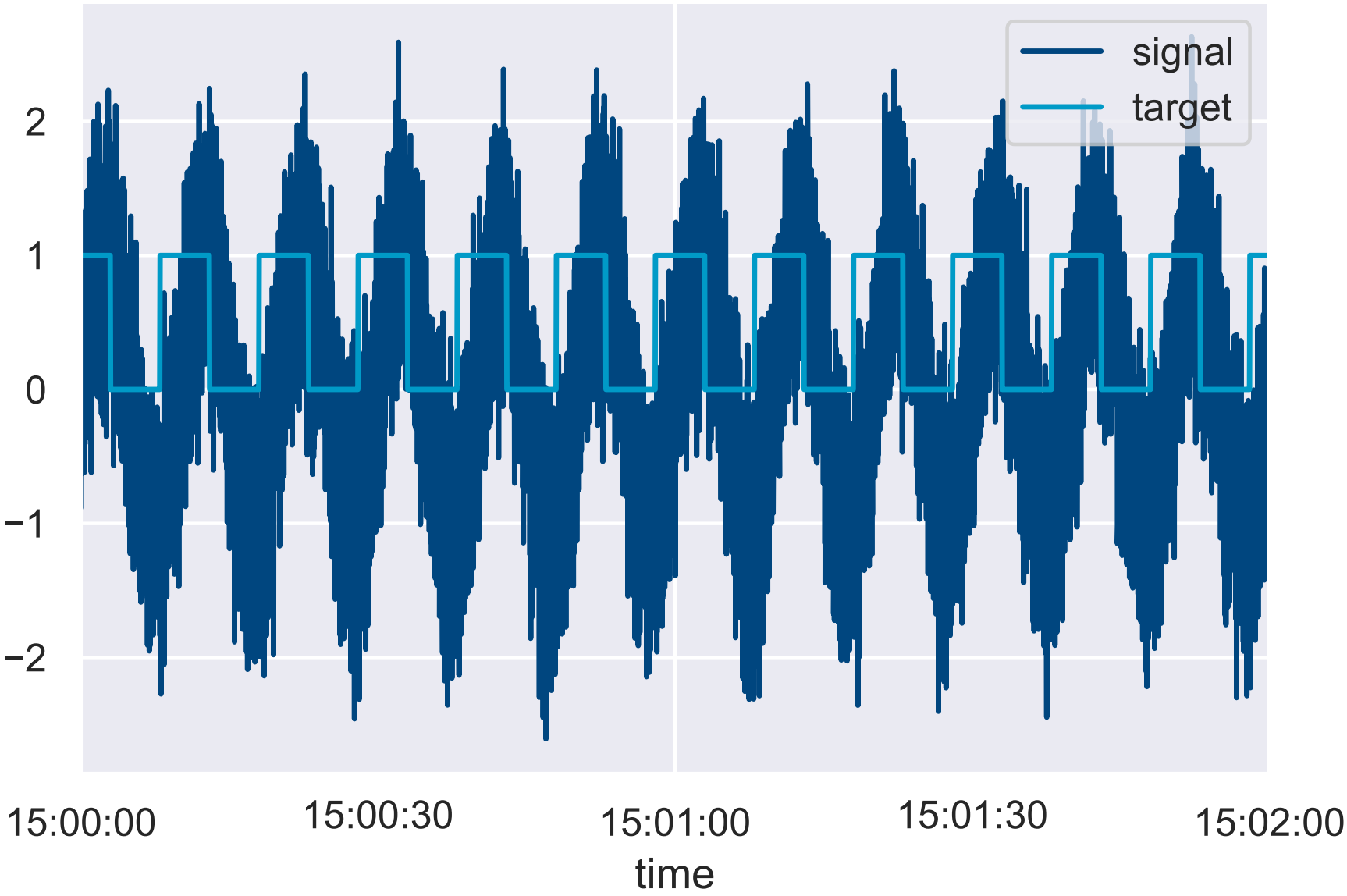
Machine learning in action - (6) Use model



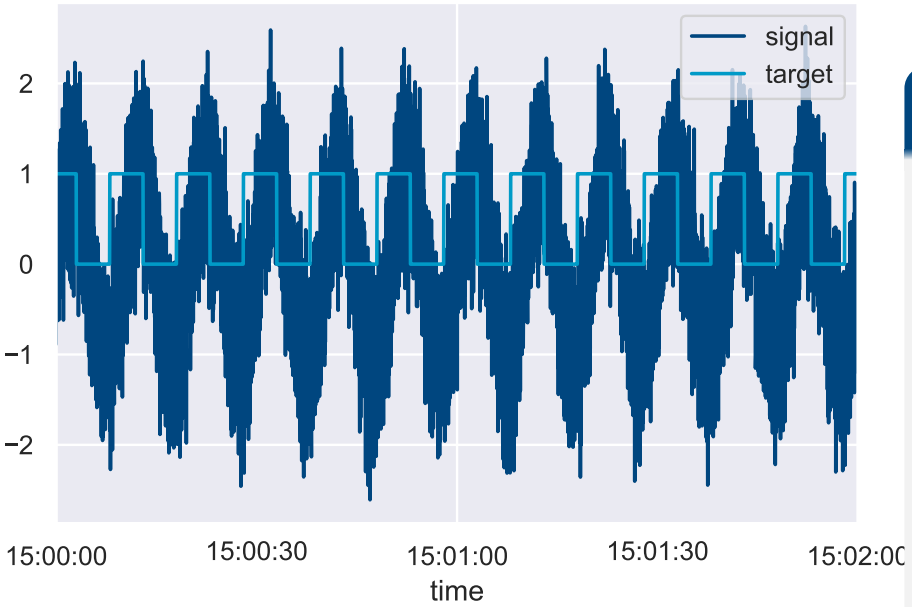
Machine learning in action - (7) Prediction



Noisy sine example



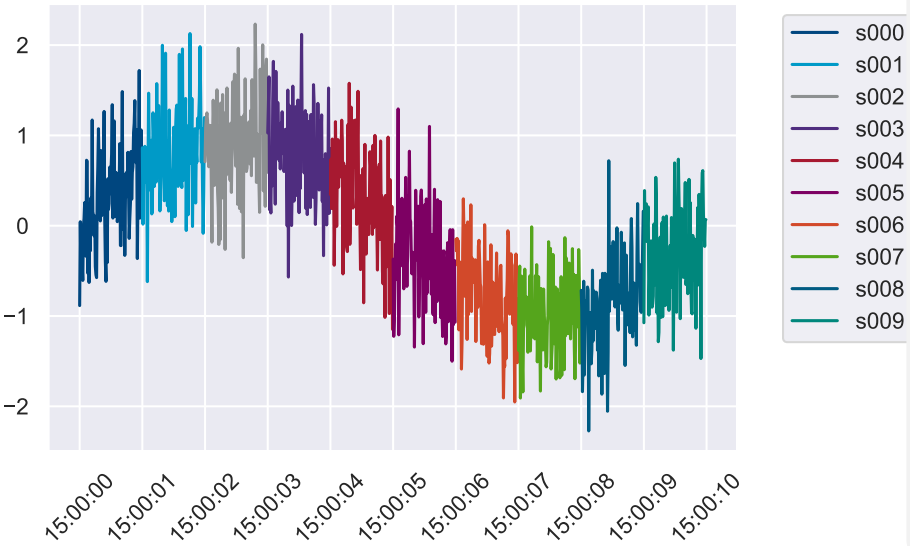
Signal Processing Example: Noisy Sine

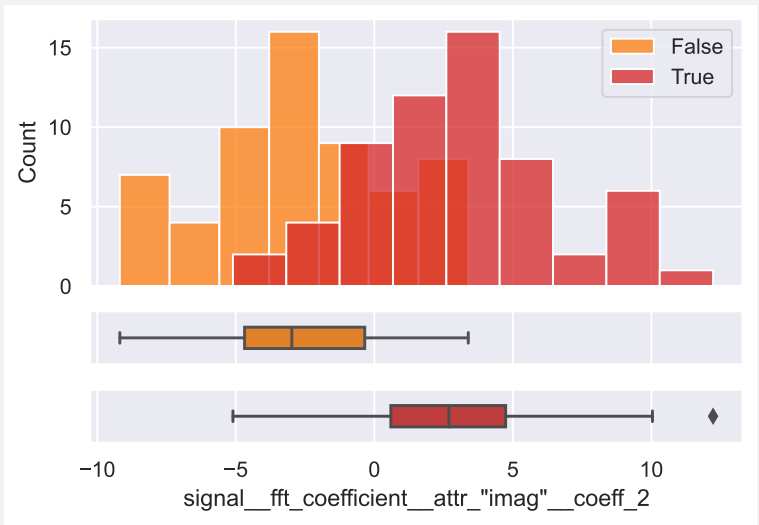


$T = 100$ $N = 120$ $K = 778$

Systematic feature engineering X_ϕ

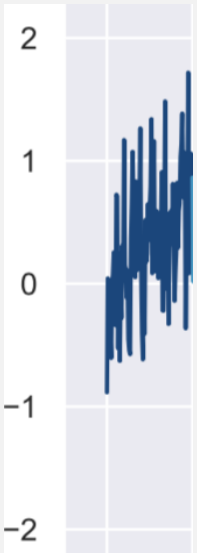
$$\begin{pmatrix} \phi_1(\vec{x}_1) & \dots & \phi_k(\vec{x}_1) & \dots & \phi_K(\vec{x}_1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \phi_1(\vec{x}_i) & \dots & \phi_k(\vec{x}_i) & \dots & \phi_K(\vec{x}_i) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \phi_1(\vec{x}_N) & \dots & \phi_k(\vec{x}_N) & \dots & \phi_K(\vec{x}_N) \end{pmatrix}$$





Signal classification - A special type of signal operator

Signal classification



input signal $\vec{x} \rightarrow$ Signal Operator $\rightarrow$ $y = \left(\mathbb{P}[y = \texttt{True}|\vec{x}] \geq \frac{1}{2}\right)$

Logistic regression

- Let $X_\phi \in \mathbb{R}^{N \times \kappa}$ be the feature matrix of $1 \leq \kappa \leq K$ statistically significant time-series features, and $\phi_k(\vec{x}_1), \ldots, \phi_\kappa(\vec{x}_N)$ its k^{th} feature column.
- Let $y_1, \ldots, y_N$ be the labels of signals $\vec{x}_1, \ldots, \vec{x}_N$.
- Probability $\mathbb{P}[y = \text{True}|\vec{x}]$ that target y is True given observed signal $\vec{x}$ is predicted as

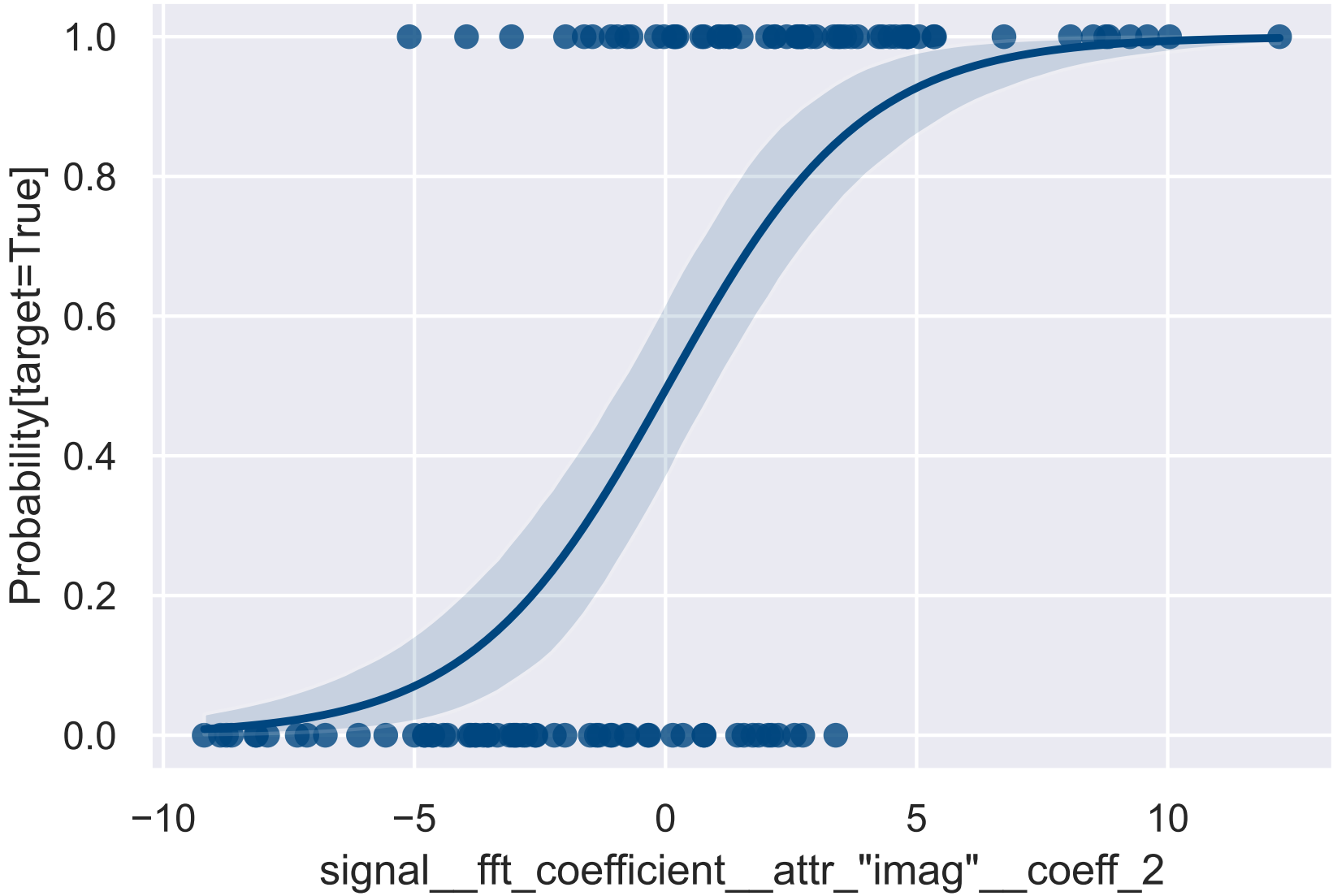
$$\mathbb{P}[y = \text{True}|\boldsymbol{x}] = \left(1 + \mathrm{e}^{-\left(\hat{w}_0 + \sum_{k=1}^{K} \hat{w}_k \phi_k(\boldsymbol{x})\right)}\right)^{-1}$$

$$\hat{w}_0, \hat{w}_1, \ldots, \hat{w}_k = \underset{w_0,\ldots,w_k}{\arg\min} \left[\sum_{i=1}^{N} -\log \mathbb{P}\left[y = y_i|\boldsymbol{x}_i\right] + \frac{1}{C}\sum_{k=0}^{K} |w_k|\right.$$

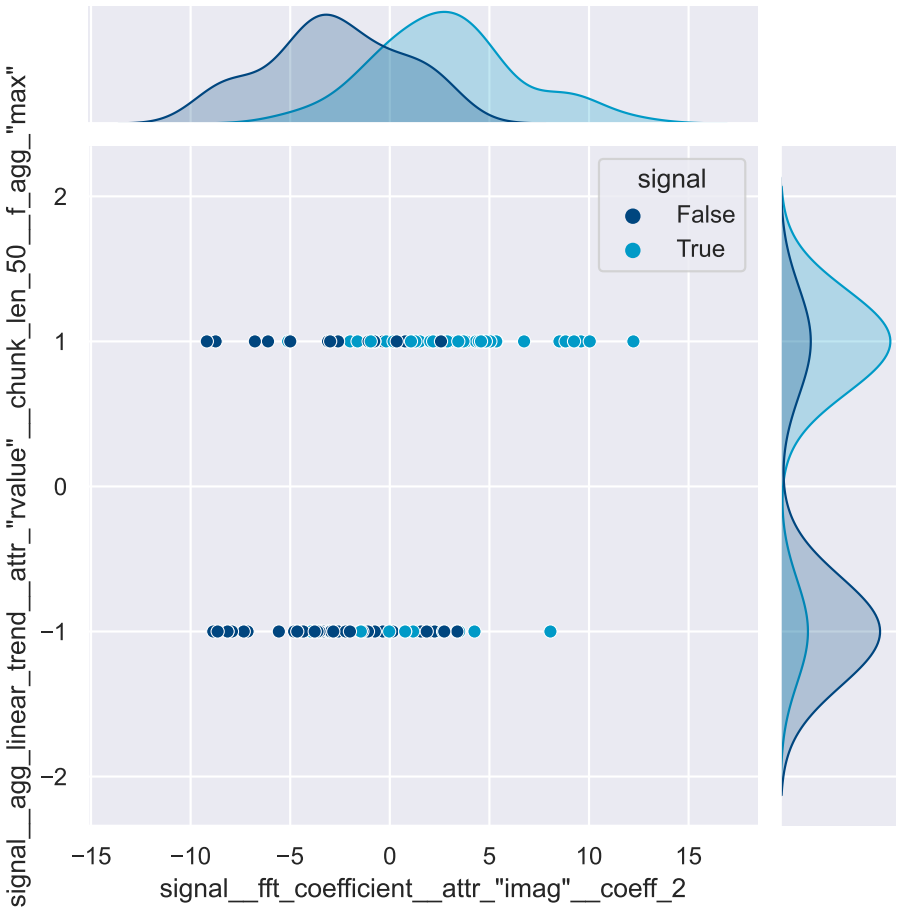
with C beging the inverse regularisation strength.

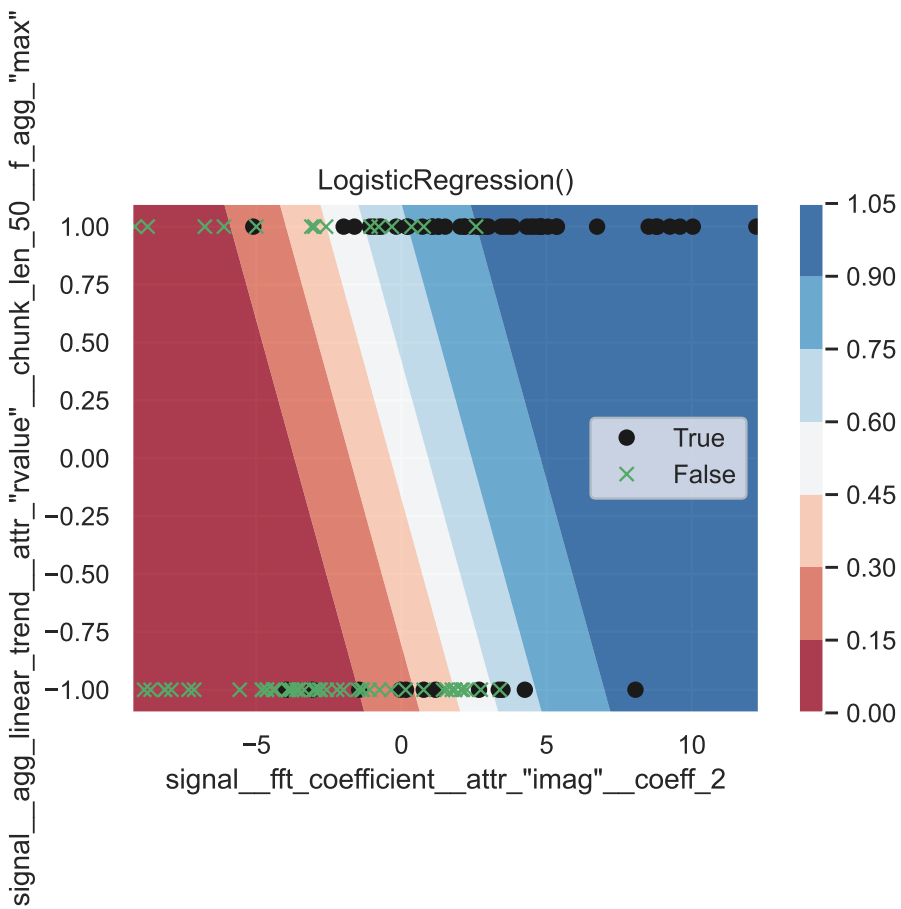
Logistic regression with one time-series feature

$C = 1.0, w_0 = -0.01387571, w_1 = 0.50363426$



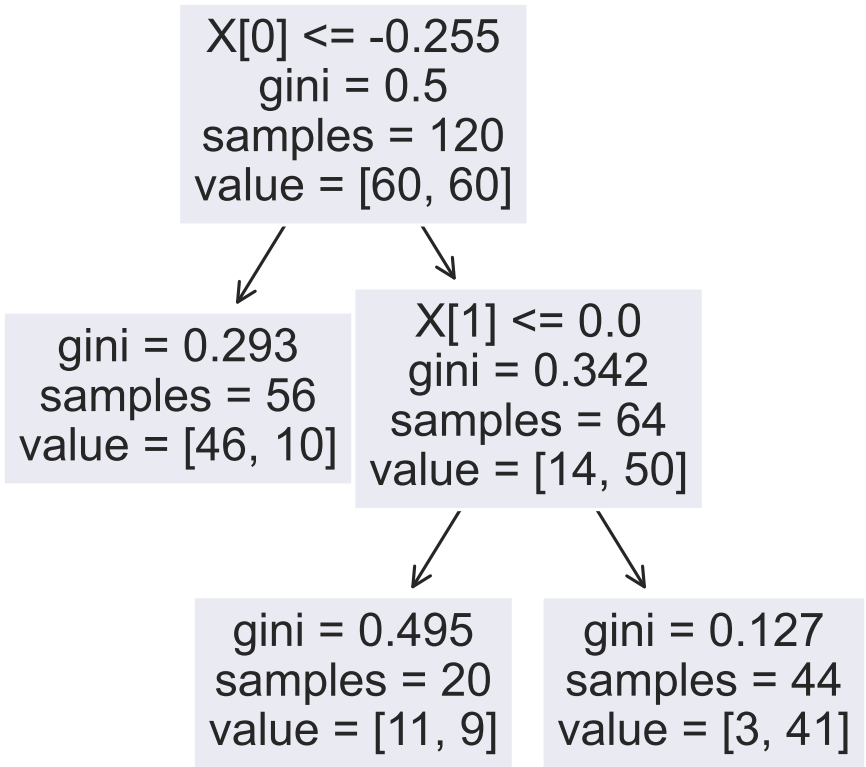
Logistic regression with two time-series features

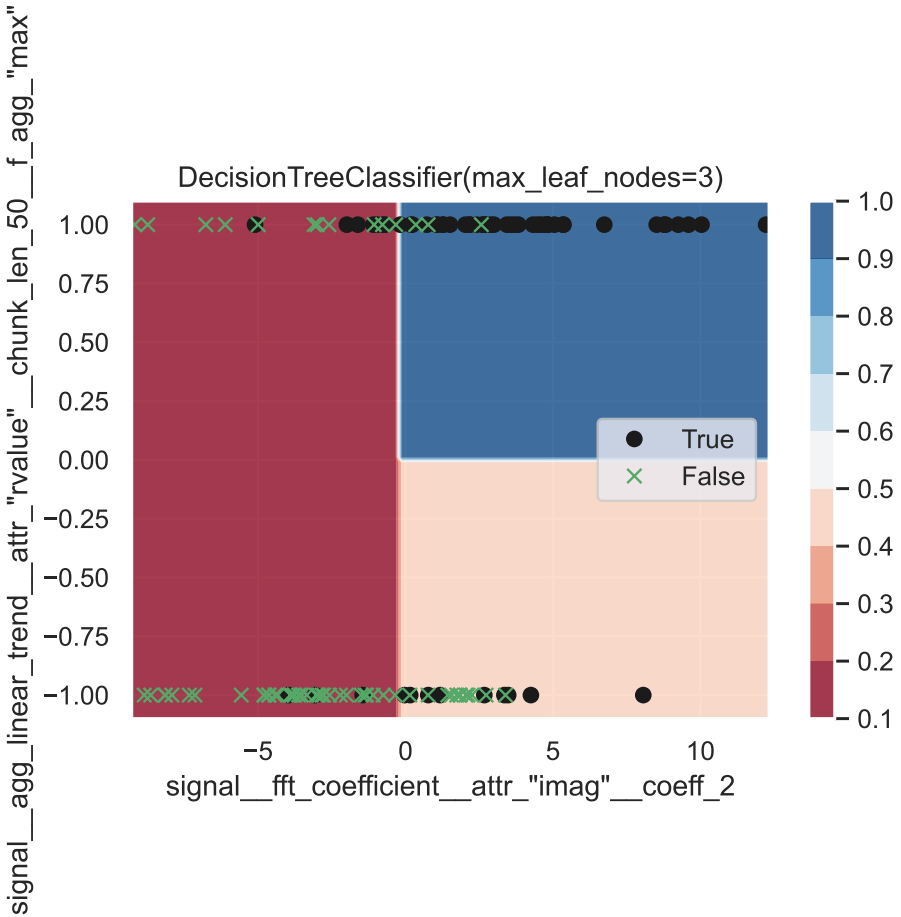




Decision tree with two time-series features

X[0] = `signal__fft_coefficient__attr_"imag"__coeff_2'`
X[1] = `signal__agg_linear_trend__attr_"rvalue"__chunk_len_50__f_agg_"max"`





Gini impurity bottom left node $G = 1 - \left(\frac{11}{20}\right)^2 - \left(\frac{9}{20}\right)^2 = 0.495$

Random Forests

Definition

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
Random decision forests correct for decision trees' habit of overfitting to their training set.

How does is work?

The Random Forest algorithm introduces extra randomness. Instead of searching for the very best feature when splitting a node, it search for the best feature among a random subset of features.

Random forest with two time-series features