1 Introduction[1]

Machine learning uses data examples to predict a label or value for a ________example.

Supervised vs. Unsupervised Learning

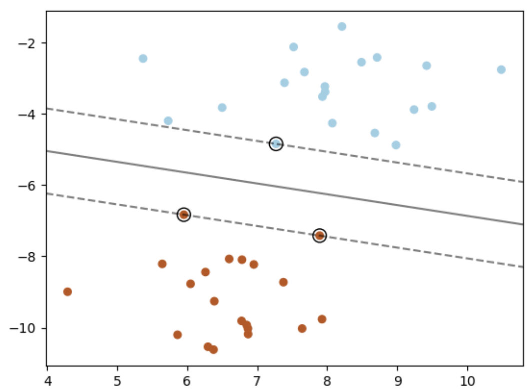
- In *supervised learning*, the dataset is a collection of labeled examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i = \left[x_1^{(i)}, \ldots, x_D^{(i)}\right]$ is a D-dimensional *feature vector*.

  e.g. Here are data with $N =$ ____ and $D =$ ____ from three kids of $\{(\mathbf{x}_i = [height, weight], y_i = age)\}$:

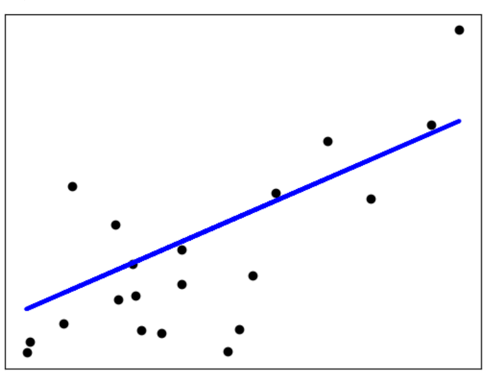
	height	weight	age
1	44	70	7
2	45	75	9
3	38	40	4

  We create a model to map new examples to suitable labels, e.g.:

  - A *support vector machine* (SVM) is a ____________classifier that uses a line to separate points in a plane into two groups (or it separates D-dimensional points with a $(D-1)$-dimensional hyperplane). e.g.[2]

  - *Linear regression* predicts a __________________label given an unlabeled example as $y \leftarrow f_{\mathbf{w},b}(\mathbf{x}) = \mathbf{wx} + b$ for scalar y, vector $\mathbf{x}$, and parameter vector $\mathbf{w}$. e.g.

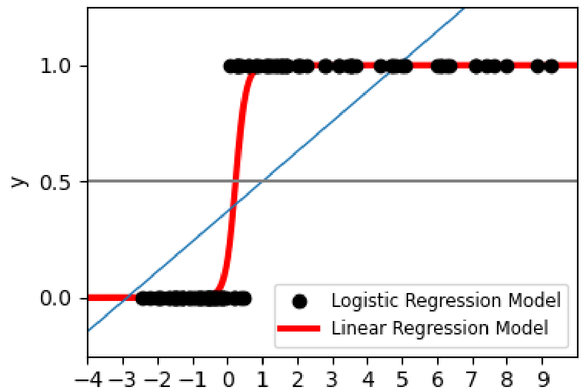
    

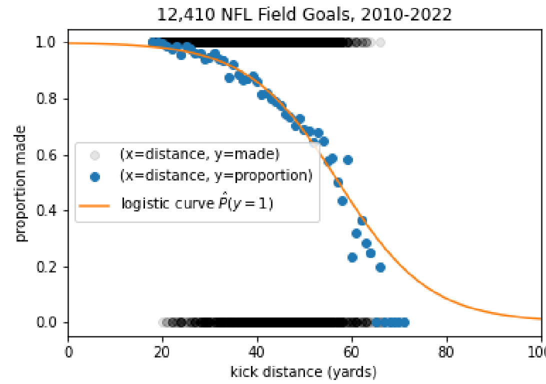
---

[1]These notes are based on Andriy Burkov's "The Hundred-Page Machine Learning Book" (`http://themlbook.com`).

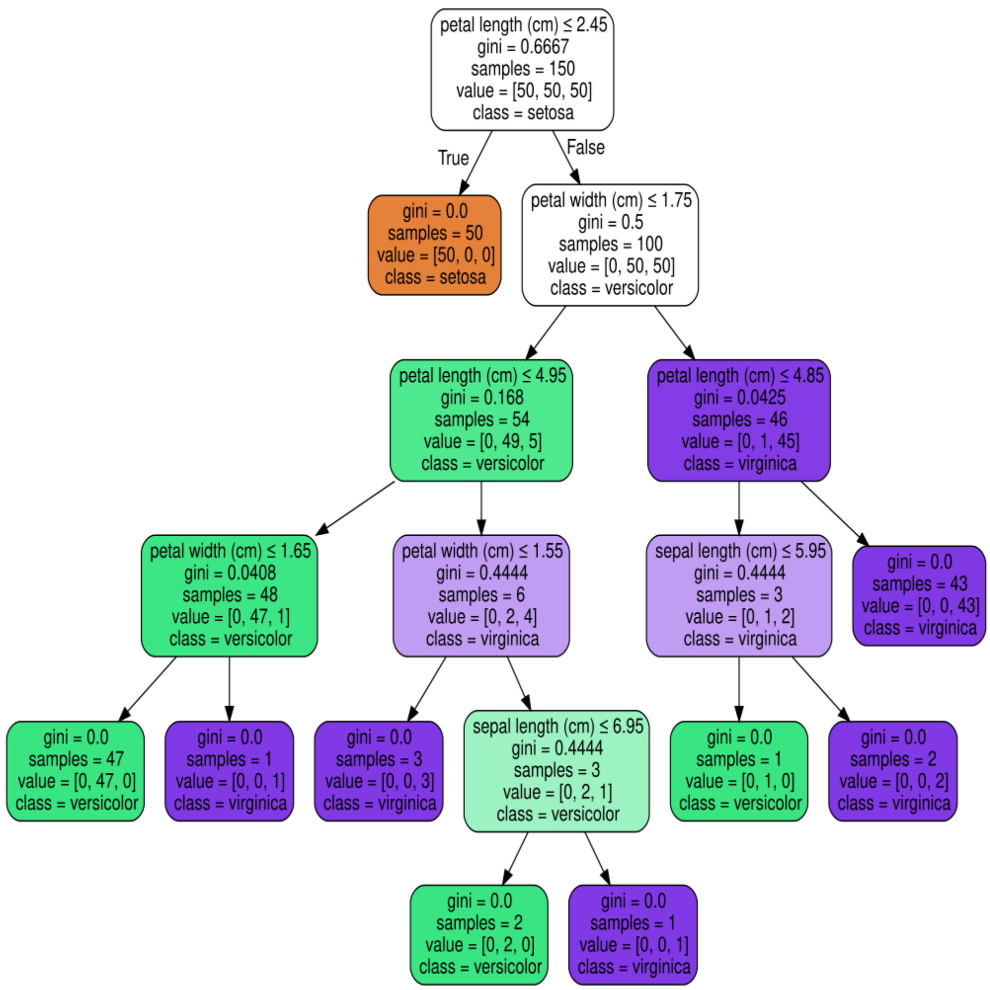
[2]from `https://scikit-learn.org/stable/modules/svm.html` and `https://scikit-learn.org/stable/_images/sphx_glr_plot_ols_001.png`

– *Logistic regression* models a ______________ in $[0, 1]$. e.g.[3]



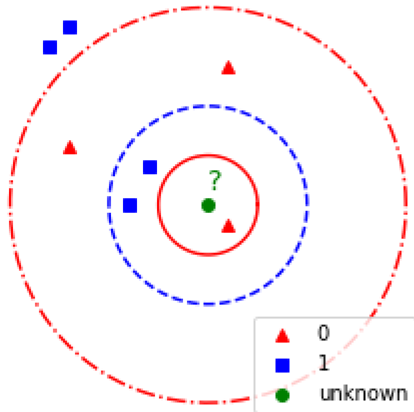


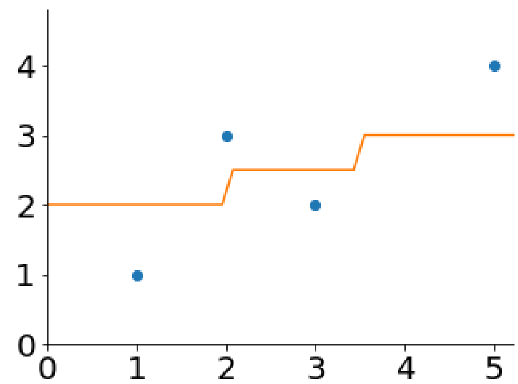
– A *decision tree* is a directed acyclic graph that we use like a ______________ to make a decision. At each node, if the value of some feature j is less than a ______________, the left branch is followed; otherwise the right branch is followed. e.g.[4]



[3]first plot is from `https://scikit-learn.org/stable/_images/sphx_glr_plot_logistic_001.png`
[4]from `https://scikit-learn.org/stable/_images/iris.svg`

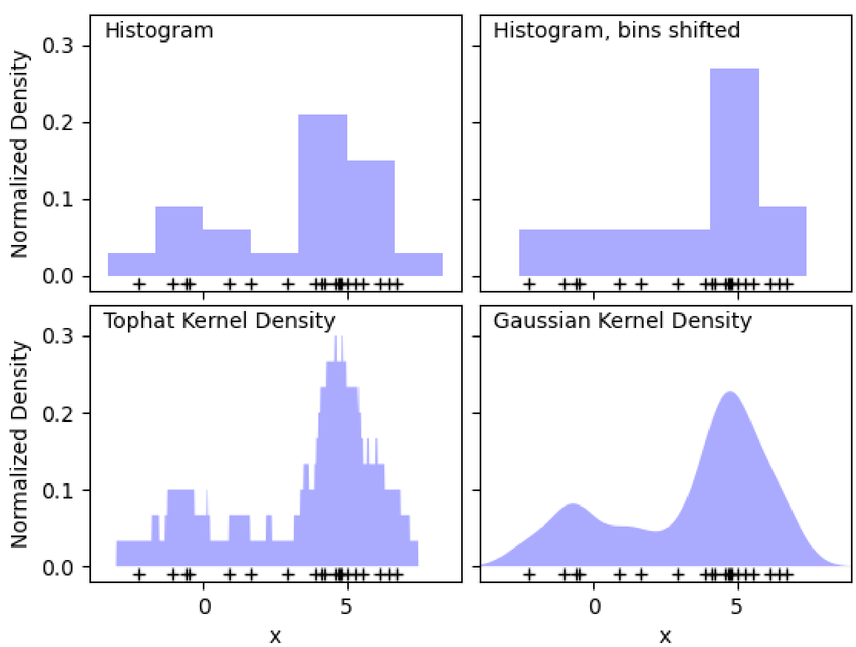
– *k-nearest neighbors* (k-NN) classification assigns a new $\mathbf{x}$ the ________________ label among its ____ nearest neighbors. k-NN regression assigns $\mathbf{x}$ the ____________ value among its k nearest neighbors. e.g.





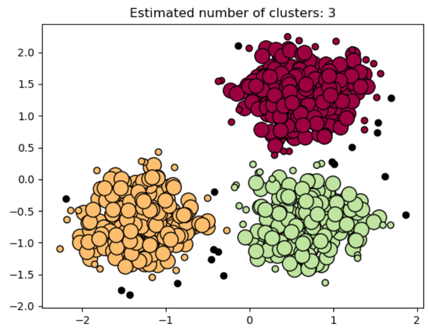
- In *unsupervised learning*, the dataset is a collection of ______________ examples $\{\mathbf{x}_i\}_{i=1}^N$ and we infer a function on $\mathbf{x}$ to solve a problem or find hidden structure in $\{\mathbf{x}_i\}$. e.g.:

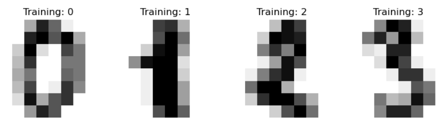
  – *Density estimation* models the probability density function of the (______________) distribution from which data were drawn. e.g.[5]

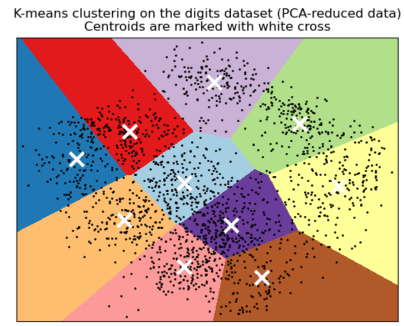


[5]from `https://scikit-learn.org/stable/_images/sphx_glr_plot_kde_1d_001.png`

– *Clustering* maps each unlabeled example **x** to a ______________. e.g.[6]

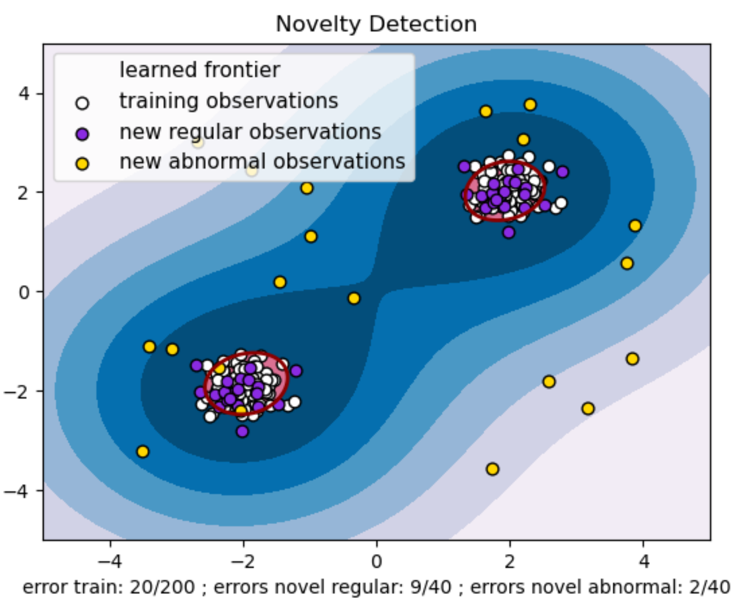






– *Dimensionality reduction* maps **x** into a vector with __________________ to remove ______________________ features, reduce ____________, ______________ data (since we can only see up to 3D), and facilitate simple interpretable models.

– *Outlier detection* quantifies how far **x** is from from ____________ examples. e.g.[7]



[6]from https://scikit-learn.org/stable/_images/sphx_glr_plot_dbscan_001.png and https://scikit-learn.org/stable/_images/sphx_glr_plot_digits_classification_001.png and https://scikit-learn.org/stable/_images/sphx_glr_plot_kmeans_digits_001.png

[7]from https://scikit-learn.org/stable/_images/sphx_glr_plot_oneclass_001.png

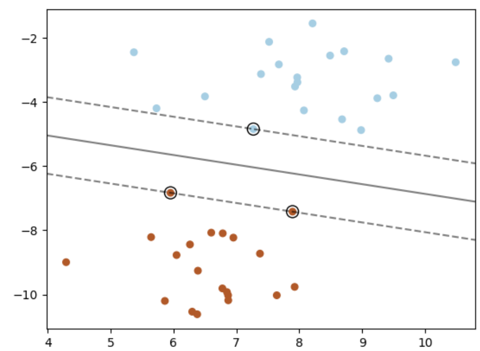
Support Vector Machine (SVM): The Linear Model

- A *hyperplane* in a D-dimensional space is a $(D-1)$-dimensional space. e.g. A hyperplane is a ______________ in 1D, a ______________ in 2D, and a ______________ in 3D.
- SVM using a *linear model* finds a hyperplane *decision boundary* specified by $\mathbf{wx} + b = 0$ that separates label $+1$ examples from label -1 examples.[8] (Note: $\mathbf{wx} = w^{(1)}x^{(1)} + \ldots + w^{(D)}x^{(D)}$.)
- Training learns optimal values $\mathbf{w}^*$ and b^*.
- The SVM labels a new $\mathbf{x}$ with $y = f(\mathbf{x}) =$ ______________ $(\mathbf{w}^*\mathbf{x} + b^*) \in \{-1, 1\}$.
- In the easiest *hard margin SVM* case where the two labeled subsets are linearly separable,[9] training consists of minimizing Euclidean norm $||\mathbf{w}|| = \sqrt{\sum_{i=1}^{D} \left(w^{(i)}\right)^2}$ subject to constraints $\begin{cases} \mathbf{wx}_i + b \geq 1 & \text{if } y_i = +1 \\ \mathbf{wx}_i + b \leq -1 & \text{if } y_i = -1 \end{cases}$, or equivalently subject to $y_i(\mathbf{wx}_i + b) \geq 1$, for $i = 1, \ldots, N$.
  (We omit the details of this *constrained optimization* problem.)
- Here we find the distance between the constraint boundaries.
  The parallel hyperplanes $\mathbf{wx} + b = 1$ and $\mathbf{wx} + b = -1$ have normal vector _______. Let $\mathbf{x}_1$ be any point in the first hyperplane. The normal line through $\mathbf{x}_1$ is $\mathbf{x}_1 + \mathbf{w}t$ for $t \in \mathbb{R}$. It intersects the second hyperplane when

  $$\mathbf{w}(\mathbf{x}_1 + \mathbf{w}t) + b = -1 \implies t = \frac{-(\mathbf{wx}_1 + b) - 1}{\mathbf{ww}} = \frac{-2}{||\mathbf{w}||^2}.$$

  The intersection point is $\mathbf{x}_2 = \mathbf{x}_1 + \mathbf{w}\left(\frac{-2}{||\mathbf{w}||^2}\right) = \mathbf{x}_1 - \frac{2\mathbf{w}}{||\mathbf{w}||^2}$.

  The distance from $\mathbf{x}_1$ to $\mathbf{x}_2$ is $\left|\left|\mathbf{x}_1 - \left(\mathbf{x}_1 - \frac{2\mathbf{w}}{||\mathbf{w}||^2}\right)\right|\right| = \frac{2}{||\mathbf{w}||}$.



- $||\mathbf{w}||$ is in the denominator of the distance, so minimizing $||\mathbf{w}||$ ________________ the *margin* between $+1$ and -1 support vectors.
- A sample on either of the constraint/margin boundaries is called a ______________ *vector*.

Coming in §3:

- An SVM can have a *hyperparameter* (parameter controlling learning; not trained) to penalize misclassification of outliers (positives on the negative side of the boundary or negatives on the positive side).
- An SVM can include a *kernel* that allows a ______________ decision boundary.

[8] Burkov uses $\mathbf{wx} - b = 0$. I use $\mathbf{wx} + b = 0$ to match scikit-learn.
[9] We return to SVMs in §3 to address some harder cases.

Python

- `from sklearn import svm` loads the `svm` module
- `clf = svm.SVC(kernel='linear', C=1)` gives a SVM support vector classification model. (A large `C`, like `C=1000`, gives ≈ the hard-margin version above; we will explore `C` more in §3.)
- `clf.fit(X, y)` fits the model to $\texttt{X}_{N \times D}$ and $\texttt{y}_{N \times 1}$.[10]
- `clf.coef_` gives $\mathbf{w}^*$ and `clf.intercept_` gives b^*
- `clf.predict(X)` does classification on examples in `X`
- `clf.score(X, y)` gives the average accuracy on `X` with respect to `y`

To learn more:

- User guide: `https://scikit-learn.org/stable/modules/svm.html`
- Reference manual:
  `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`
- Example:
  `https://scikit-learn.org/stable/auto_examples/svm/plot_separating_hyperplane.html`

[10] In the code, `X` is 2D while `y` is 1D.