

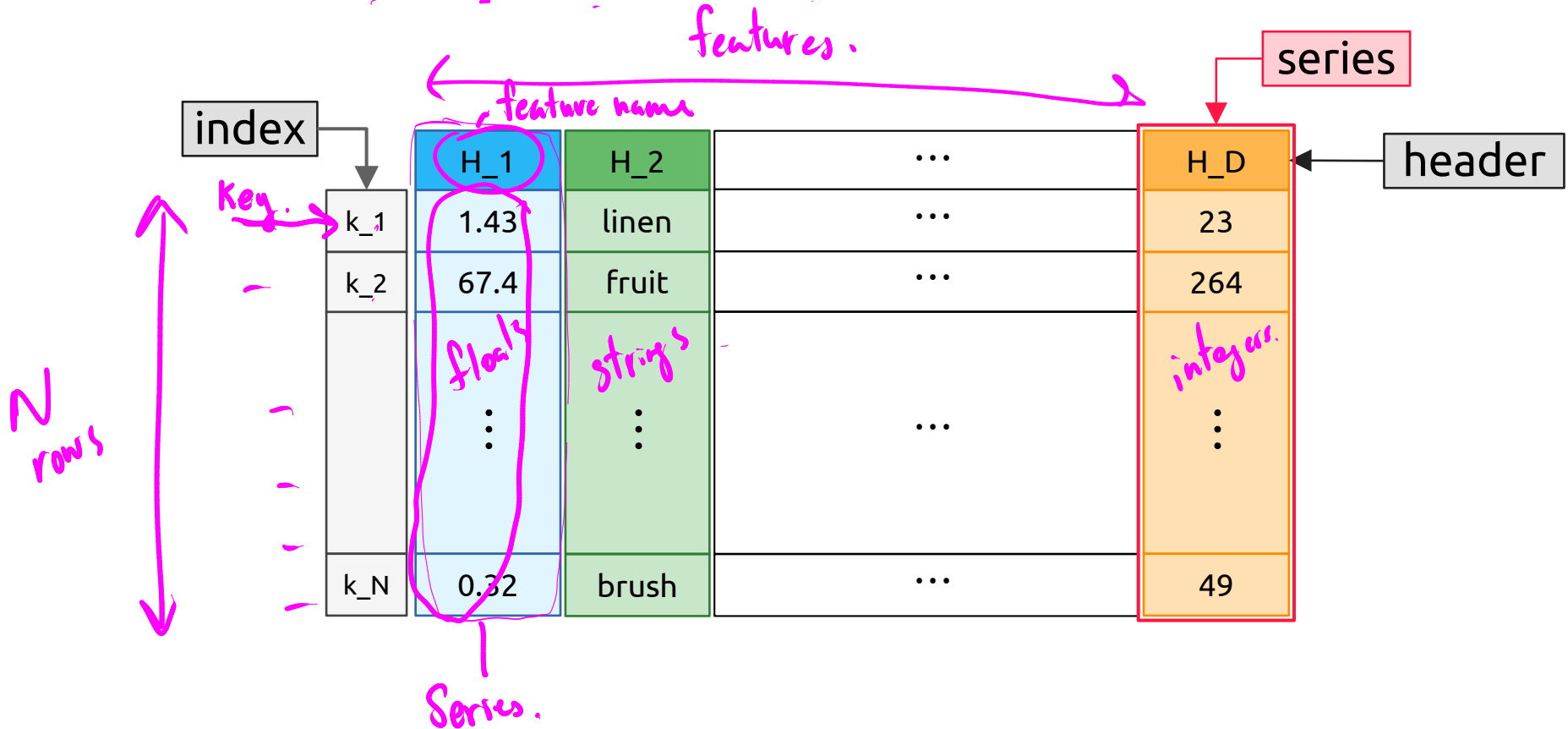


Statistics and Data Science for Engineers E178 / ME276DS

pandas
&
scikit-learn

pandas: A package for tabular data

- **DataFrame**: (index, {header:series})



Querying a DataFrame

- **Column selectors:** `[]`
 - `X["H1"]` ... single column
 - `X[["H1","H2"]]` ... multiple columns
- **Selecting rows by index:** `.loc[]`
 - `X.loc[k1]` ... single row
 - `X.loc[[k1,k2]]` ... multiple rows
- **`loc[]` also accepts a column selector**
 - `X.loc[k1,"H1"]`
 - `X.loc[k1,["H1","H2"]]`
 - `X.loc[[k1,k2],"H1"]`
 - `X.loc[[k1,k2],["H1","H2"]]`

Querying a DataFrame (cntd.)

- **Selecting rows with a conditional**

- `X.loc[boolean_mask]`

... syntactic sugar: `X[<boolean mask>]`

- `X.loc[boolean_mask, column_selector]`

- **Ordered rows (integer index)**

- `X[slice]`

- `X.loc[slice, column_selector]`

- **Ordered rows and columns: `.iloc[]`**

- `X.iloc[row_slice]`

- `X.iloc[row_slice, col_slice]`

Loading and saving data

Single object files


	text	pickle
numpy	<code>np.savetxt(filename,A)</code> <code>A = np.loadtxt(filename)</code>	<code>np.save(filename,A)</code> <code>A = np.load(filename)</code>
pandas	<code>DF.to_csv(filename)</code> <code>DF = pd.read_csv(filename)</code>	<code>DF.to_pickle(filename)</code> <code>DF = pd.read_pickle(filename)</code>

Multiple object files

```
import pickle
```

```
with open("mypickle.pkl","wb") as f:  
    pickle.dump((A,D),f)
```

```
with open("mypickle.pkl", "rb") as f:  
    Anew, Dnew = pickle.load(f)
```


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

scikit-learn

Machine Learning in Python

[Getting Started](#)
[Release Highlights for 1.1](#)
[GitHub](#)

- 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

<https://scikit-learn.org>

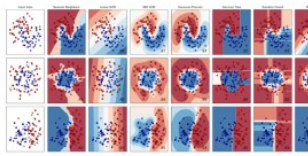
- Open source
- Nice API
- Extensible
- Used in industrial R&D

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...



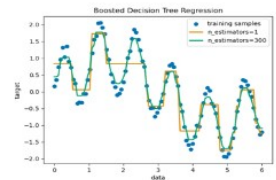
Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...



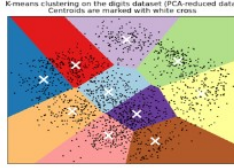
Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, and more...



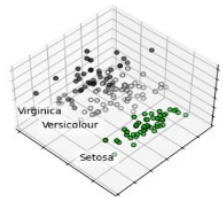
Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: PCA, feature selection, non-negative matrix factorization, and more...



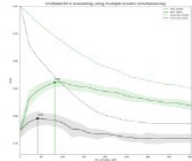
Examples

Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning

Algorithms: grid search, cross validation, metrics, and more...



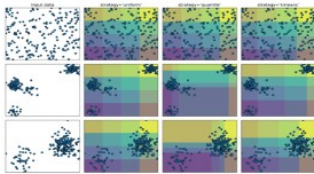
Examples

Preprocessing

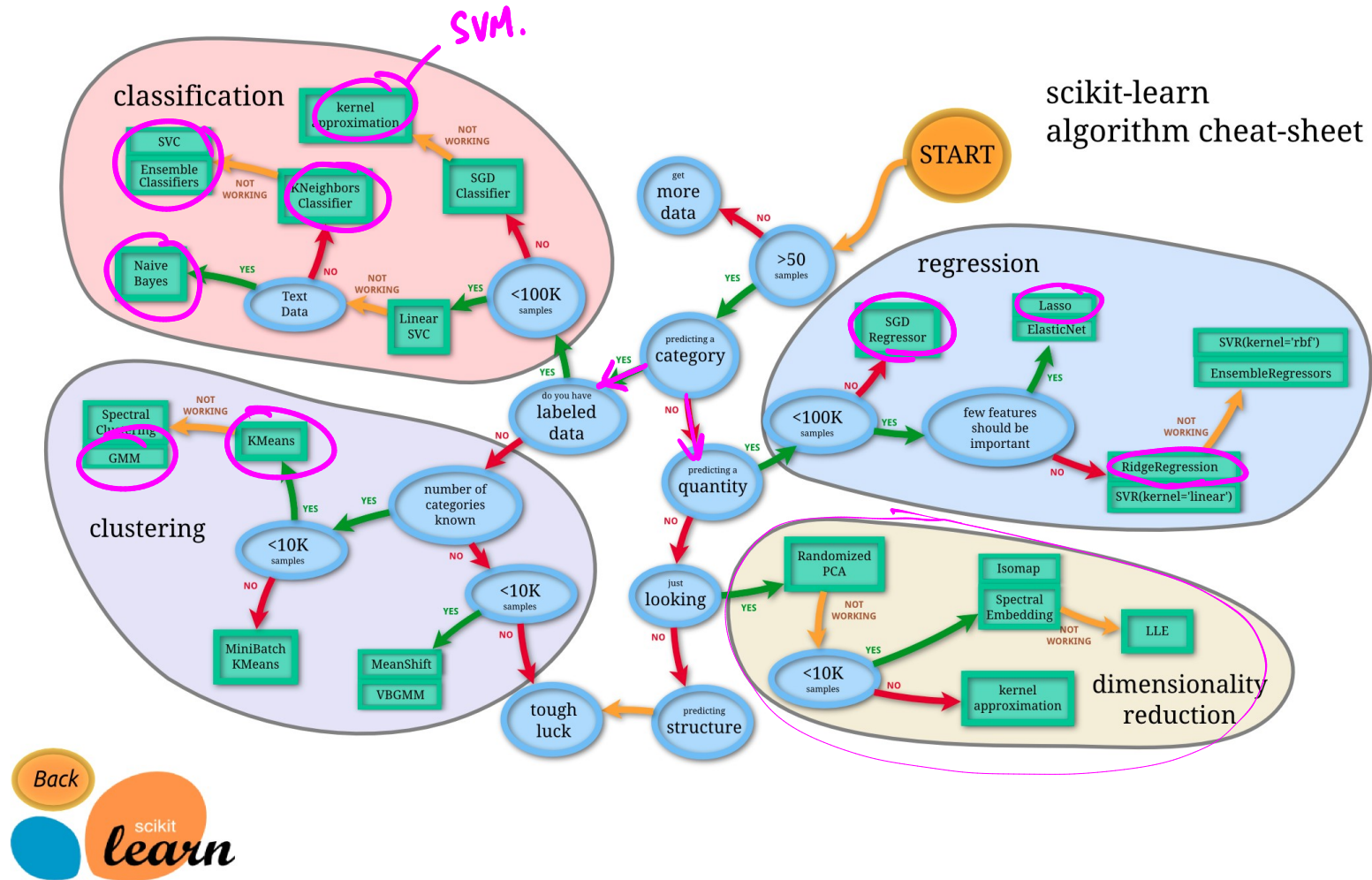
Feature extraction and normalization.

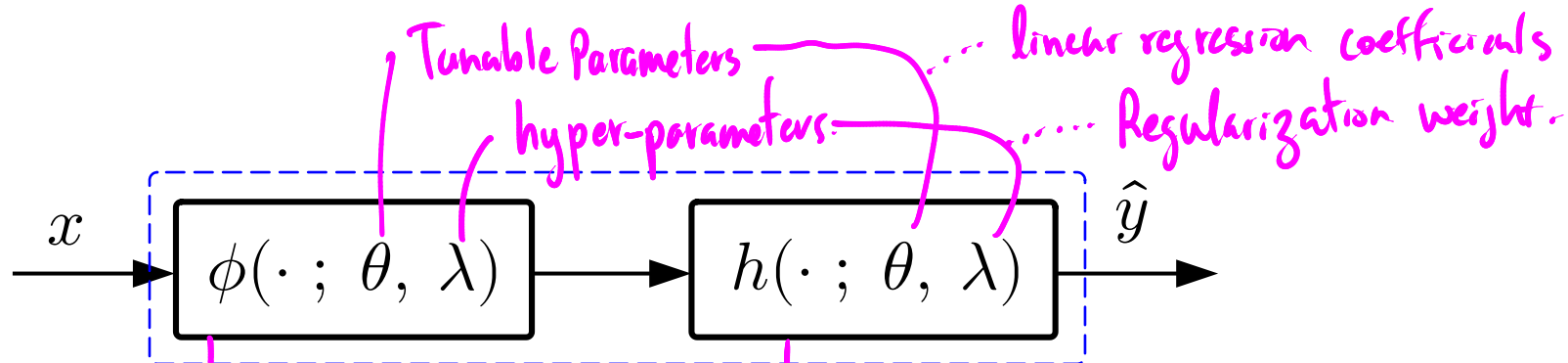
Applications: Transforming input data such as text for use with machine learning algorithms.

Algorithms: preprocessing, feature extraction, and more...



Examples





preprocessing.

- Scaling the input. (aka Normalizing).
- Encode. (Bag of words).
- Clustering (kmeans). (GMM).

modeling.

- Regression
 - Linear Regression
 - Ridge
 - Lasso.
 - KNN,
- Classification
 - Naive Bayes.
 - Logistic Regression.
 - KNN,

scikit-learn API

Estimators: ... *fit* : Runs the training algorithm. to compute θ

Transformers

$$\Phi = \phi(\mathbf{X} ; \theta, \lambda)$$

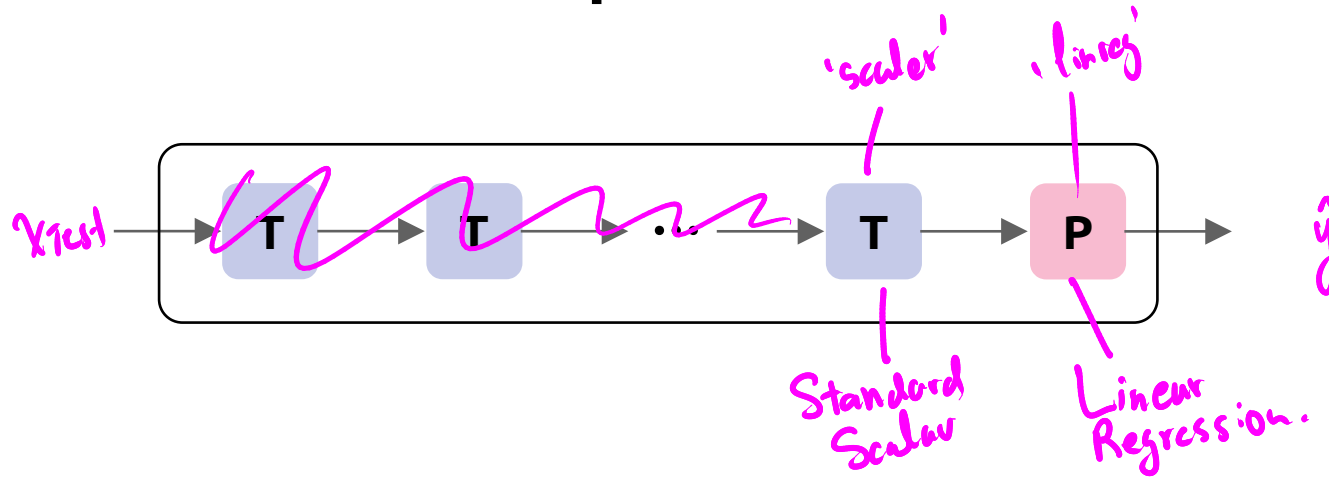
.transform(...)
.fit_transform(...)

Predictors

$$\hat{\mathbf{Y}} = h(\Phi ; \theta, \lambda)$$

.predict(...)
.fit_predict(...)

Pipelines



- A pipeline is an estimator of the same type as its downstream-most component.
- Strings together transformers and predictors.
- Clearly separates fitting and predicting functionality.