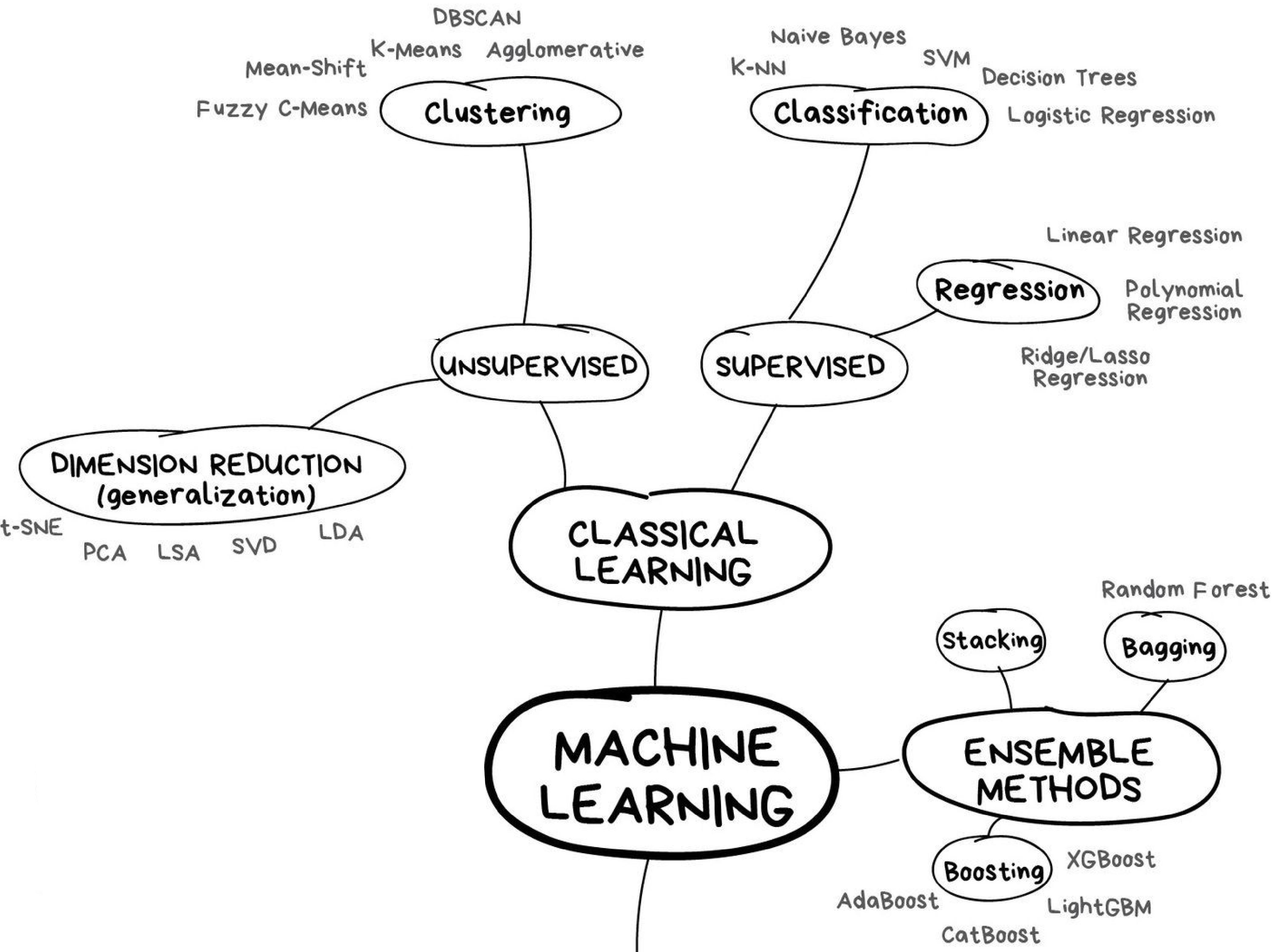
Aprendizaje automático con scikit


DBSCAN
K-Means
Agglomerative
Mean-Shift
Fuzzy C-Means
Clustering
Naive Bayes
SVM
K-NN
Decision Trees
Classification
Logistic Regression
Linear Regression
Regression
Polynomial Regression
Ridge/Lasso Regression
UNSUPERVISED
SUPERVISED
DIMENSION REDUCTION (generalization)
t-SNE
PCA
LSA
SVD
LDA
CLASSICAL LEARNING
Random Forest
Stacking
Bagging
MACHINE LEARNING
ENSEMBLE METHODS
Boosting
XGBoost
AdaBoost
LightGBM
CatBoost

scikit learn

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, … — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, … — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, … — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

Representing Data

$$
\begin{array}{rc}
\text{one sample} & \\
X = & \left(\begin{array}{ccccc}
1.1 & 2.2 & 3.4 & 5.6 & 1.0 \\
6.7 & 0.5 & 0.4 & 2.6 & 1.6 \\
2.4 & 9.3 & 7.3 & 6.4 & 2.8 \\
1.5 & 0.0 & 4.3 & 8.3 & 3.4 \\
0.5 & 3.5 & 8.1 & 3.6 & 4.6 \\
5.1 & 9.7 & 3.5 & 7.9 & 5.1 \\
3.7 & 7.8 & 2.6 & 3.2 & 6.3
\end{array}\right) \\
 & \text{one feature}
\end{array}
\qquad
y = \left(\begin{array}{c}
1.6 \\ 2.7 \\ 4.4 \\ 0.5 \\ 0.2 \\ 5.6 \\ 6.7
\end{array}\right)
$$

outputs / labels

NumPy

Simple API

- estimator.fit(X, [y])
- estimator.predict
  - Classification
  - Regression
  - Clustering

- transformer.fit(X)
- transformer.transform
  - Preprocessing
  - Dimensionality reduction
  - Feature selection
  - Feature extraction

Regression

- **model =** LinearRegression()
- **y_pred** = model.fit.(**X_train, y_train**).score**(X_test, y_test**)
- **y_pred = model.predict(X_new)**

Classification

- **model =** LogisticRegression()
- **y_pred** = model.fit.(**X_train, y_train**).score(**X_test, y_test**)
- **y_pred = model.predict(X_new)**

Clustering

- **model = KMeans()**
- **y_pred** = model.fit.(**X**).predict(X)
- **y_pred = model.predict(X_new)**

Preprocessing

- X = > X_mod

- tranformer = StandardScaler()
- X_mod = transformer.fit(X).transform(X)

Dimensionality Reduction

- X = > X_mod

- tranformer = PCA()
- X_mod = transformer.fit(X).transform(X)