SCIKIT-LEARN:A BASIC TOOL FOR MACHINE LEARNING IN PYTHON

INTELLIGENT SYSTEMS

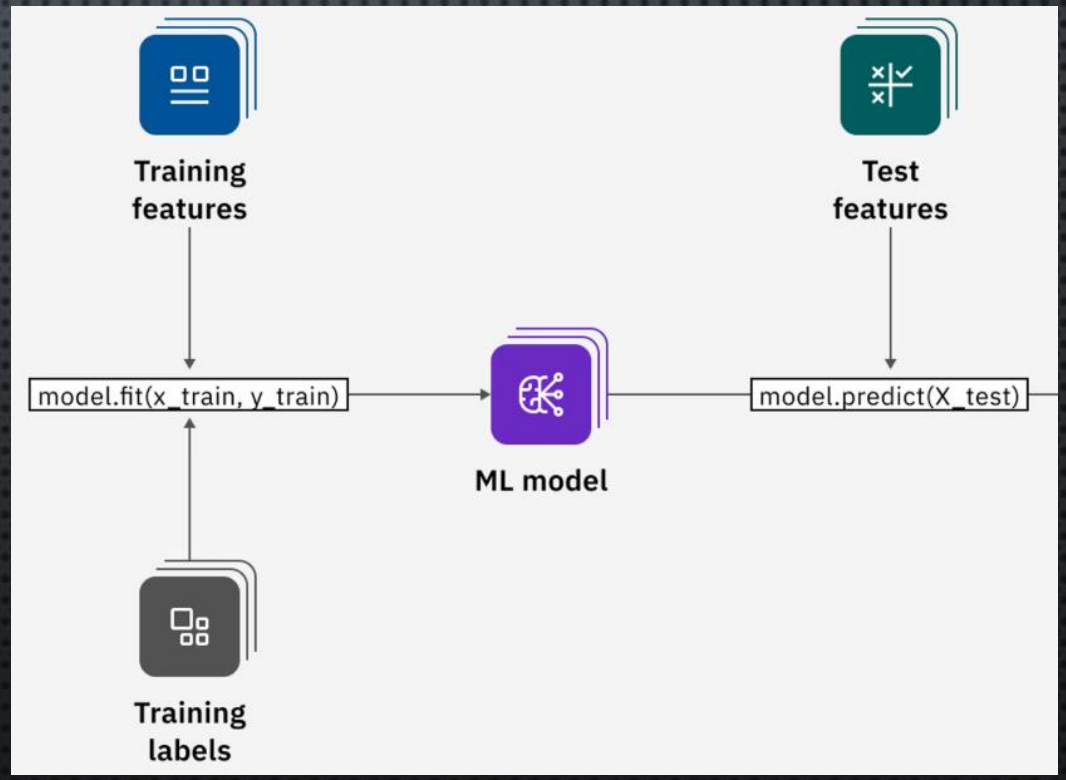
VEDRAN MITIĆ 2123

What is scikit-learn?

- The most popular Python library for machine learning
- Built on NumPy, SciPy, Cython and Matplotlib
- Open source
- Focus on data modeling

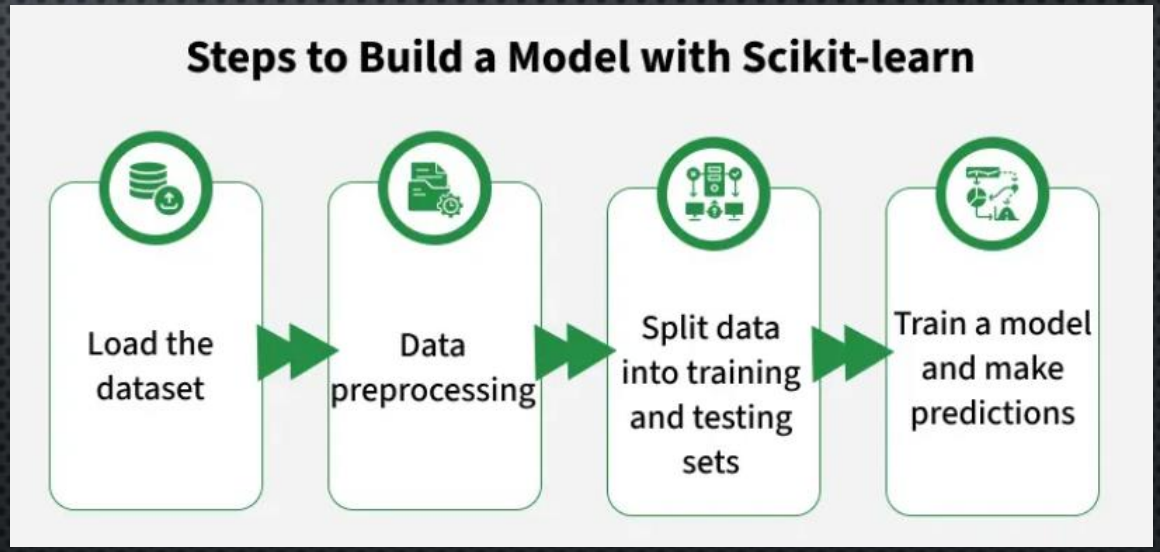
WHY SCIKIT-LEARN?

- CONSISTENCY
- DOCUMENTATION
- ROBUSTNESS



WORKFLOW

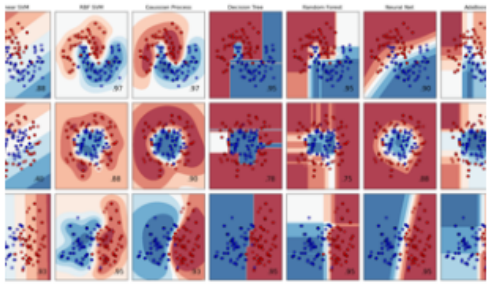
- Data preprocessing
- Estimation
- Training
- Prediction and model evaluation



Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...

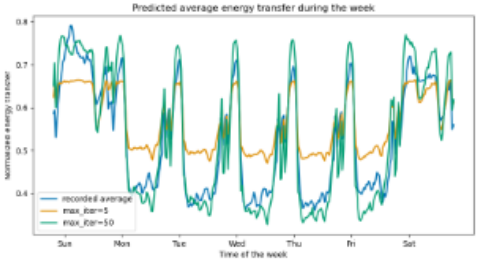


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, stock prices.
Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...

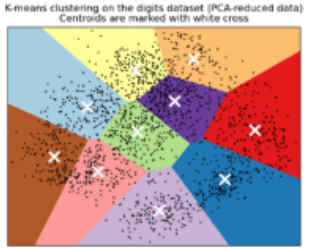


Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, grouping experiment outcomes.
Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...

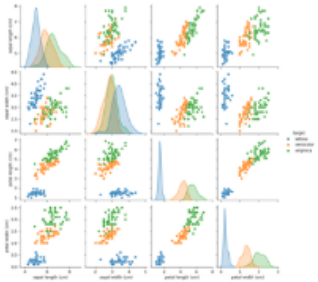


Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, increased efficiency.
Algorithms: PCA, feature selection, non-negative matrix factorization, and more...

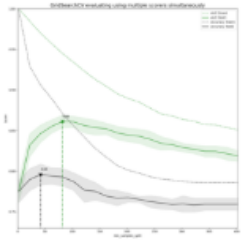


Examples

Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning.
Algorithms: Grid search, cross validation, metrics, and more...

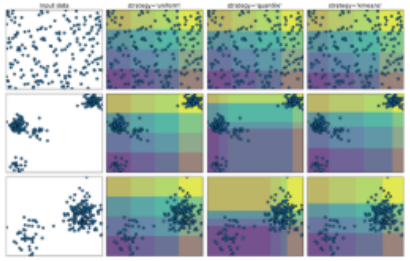


Examples

Preprocessing

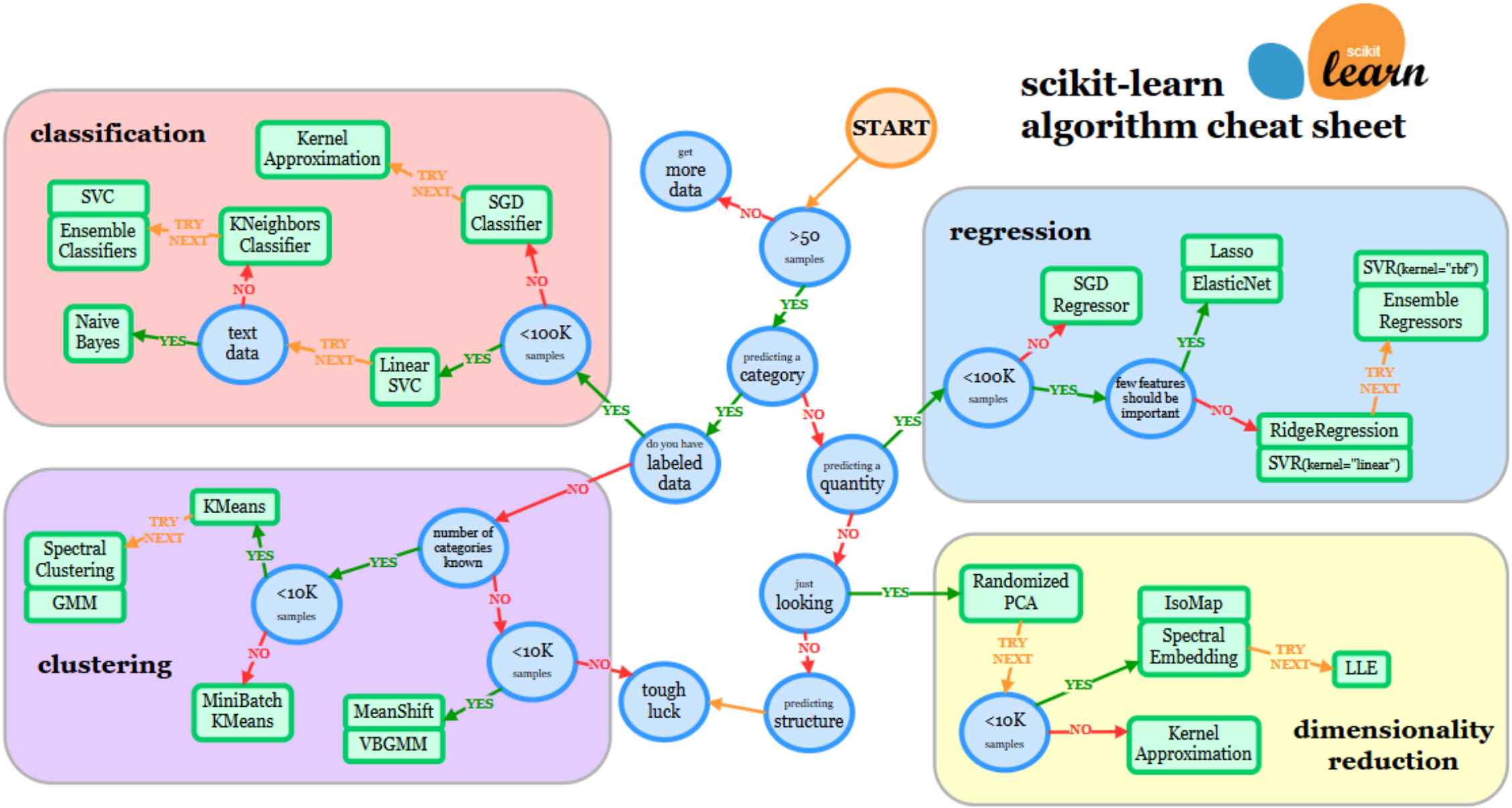
Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.
Algorithms: Preprocessing, feature extraction, and more...



Examples

scikit-learn algorithm cheat sheet



```python
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics

iris = load_iris()

X = iris.data
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=1)

log_reg = LogisticRegression(max_iter=200)
log_reg.fit(X_train, y_train)

y_pred = log_reg.predict(X_test)

print("Logistic Regression model accuracy:", metrics.accuracy_score(y_test, y_pred))

sample = [[3, 5, 4, 2], [2, 3, 5, 4]]
preds = log_reg.predict(sample)
pred_species = [iris.target_names[p] for p in preds]
print("Predictions:", pred_species)
```

PIPELINE

- Combining processing and modeling into a single object
- Data leakage
- Cleaner code
- GridSearch

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC

pipe = Pipeline([
    ('scaler', StandardScaler()),
    ('svm', SVC())
])

pipe.fit(X_train, y_train)
score = pipe.score(X_test, y_test)
```

Conclusion

- Classical machine learning
- Not a Deep Learning
- A good starting point
- https://scikit-learn.org/stable/#
- https://github.com/scikit-learn/scikit-learn/discussions