

scikit-learn Cheatsheet

scikit-learn is a popular open-source machine learning library for Python. It provides a wide range of tools for building and evaluating machine learning models, including classification, regression, clustering, and more. This cheatsheet provides a quick reference for some of scikit-learn's unique features, including code blocks for loading data, preprocessing, model selection, and more. Additionally, it includes a list of resources for further learning.

Loading Data

```
from sklearn.datasets import load_digits

# Load the digits dataset
digits = load_digits()

# Split the data into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target,
test_size=0.25, random_state=42)
```

Preprocessing

```
# Scale the data to have zero mean and unit variance
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Encode categorical variables as integers
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
y_train_encoded = encoder.fit_transform(y_train)
y_test_encoded = encoder.transform(y_test)
```

Model Selection

```
# Train a support vector machine classifier
from sklearn.svm import SVC
clf = SVC(kernel='linear', C=1.0, random_state=42)
clf.fit(X_train_scaled, y_train_encoded)

# Evaluate the classifier
from sklearn.metrics import accuracy_score
y_pred = clf.predict(X_test_scaled)
accuracy_score(y_test_encoded, y_pred)
```

Cross-Validation

```
# Perform k-fold cross-validation
from sklearn.model_selection import cross_val_score
scores = cross_val_score(clf, X_train_scaled, y_train_encoded, cv=5)
```

Grid Search

```
# Perform a grid search over hyperparameters
from sklearn.model_selection import GridSearchCV
param_grid = {'C': [0.1, 1.0, 10.0], 'kernel': ['linear', 'rbf']}
grid_search = GridSearchCV(clf, param_grid, cv=5)
grid_search.fit(X_train_scaled, y_train_encoded)
```

Other Useful Features

```
# Train a decision tree classifier
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(max_depth=3, random_state=42)
clf.fit(X_train_scaled, y_train_encoded)

# Visualize the decision tree
from sklearn.tree import plot_tree
plot_tree(clf)

# Save and load a model
import joblib
joblib.dump(clf, 'model.joblib')
clf = joblib.load('model.joblib')
```

Resources

- [scikit-learn documentation](#)
- [scikit-learn tutorials](#)
- [Python Data Science Handbook](#)