SCIKIT-LEARN CHEATSHEET:

PYTHON MACHINE LEARNING TUTORIAL

In this step-by-step Python machine learning cheatsheet, you'll learn how to use Scikit-Learn to build and tune a supervised learning model!

Scikit-Learn, also known as sklearn, is Python's premier general-purpose machine learning library. While you'll find other packages that do better at certain tasks, Scikit-Learn's versatility makes it the best starting place for most ML problems.

To see the most up-to-date full tutorial, as well as installation instructions, visit the online tutorial at elitedatascience.com.

SETUP

Make sure the following are installed on your computer:

- Python 2.7+ or Python 3
- NumPy
- Pandas
- Scikit-Learn (a.k.a. sklearn)

*We strongly recommend installing Python through Anaconda (installation guide). It comes with all of the above packages already installed.

IMPORT LIBRARIES AND MODULES

```
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.ensemble import RandomForestRegressor
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.externals import joblib
```

LOAD RED WINE DATA

```
dataset_url = 'http://mlr.cs.umass.edu/ml/machine-learning-databas
              es/wine-quality/winequality-red.csv'
data = pd.read_csv(dataset_url, sep=';')
```

SPLIT DATA INTO TRAINING AND TEST SETS

```
y = data.quality
X = data.drop('quality', axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,
                                                    random_state=123,
                                                    stratify=y)
```

DECLARE DATA PREPROCESSING STEPS

```
pipeline = make_pipeline(preprocessing.StandardScaler(),
                         RandomForestRegressor(n_estimators=100))
```

DECLARE HYPERPARAMETERS TO TUNE

```
hyperparameters = { 'randomforestregressor__max_features' : ['auto',
                    'sqrt', 'log2'],
                    'randomforestregressor__max_depth':
                    [None, 5, 3, 1]}
```

TUNE MODEL USING CROSS-VALIDATION PIPELINE

```
clf = GridSearchCV(pipeline, hyperparameters, cv=10)

clf.fit(X_train, y_train)
```

REFIT ON THE ENTIRE TRAINING SET

```
# No additional code needed if clf.refit == True (default is True)
```

EVALUATE MODEL PIPELINE ON TEST DATA

```
pred = clf.predict(X_test)
print r2_score(y_test, pred)
print mean_squared_error(y_test, pred)
```

SAVE MODEL FOR FUTURE USE

```
joblib.dump(clf, 'rf_regressor.pkl')
# To load: clf2 = joblib.load('rf_regressor.pkl')
```