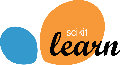
** Scikit-Learn Basics**

|  |  |
| --- | --- |
| **1. Load Data Sets**  **a) Built-in datsets**  **from sklearn import datasets**  **dir(datasets) –** will list the available data sets  **dset = datasets.load\_diabetes() –** this is a dictionary which contains actual data, target variable, feature names, and description    **b). Other datasets can be loaded in as Numpy arrays or data frames**  **pd.read\_csv(), pd.read\_table(), pd.read\_excel()**    **2. Data Preparation**  **Train/test split**  **from sklearn.model\_selection import train\_test\_split**  **X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y)**    **Standardization**  **from sklearn.preprocessing import StandardScaler**  **scaler = StandardScaler()**  **scaler.fit(X\_train)**  **X\_train = scaler.transform(X\_train)**  **X\_test = scaler.transform(X\_test)**    **Polynomial Features**  **from sklearn.preprocessing import PolynomialFeatures**  **poly = PolynomialFeatures()**  **X\_train\_poly = poly.fit\_transform(X\_train)**  **X\_test\_poly = poly.fit\_transform(X\_test)** | **3. Training a model**  **Few ML algorithms**  **from sklearn.cluster import Kmeans**  **from sklearn.linear\_model import LinearRegression**  **from sklearn.linear\_model import LogisticRegression**  **from sklearn.tree import DecisionTreeClassifier**  **from sklearn.svm import SVC**  **from sklearn.ensemble import RandomForestClassifier**  **Fitting the model & making predictions (Ex: Clustering)**  **kmc = Kmeans()**  **kmc.fit(X)**  **kmc.labels\_**  **Ex: Linear regression**  **lr = LinearRegression()**  **lr.fit(X\_train, y\_train)**  **y\_pred = lr.predict(X\_test) –** for predictions on test data  **y\_prob = lr.predict\_proba(X\_test) –** for probabilities on test data  **4. Evaluation**  **a) Regression**  **from sklearn.metrics import r2\_score, mean\_squared\_error**  **r2\_score(y\_test,y\_pred), mean\_squared\_error(y\_test,y\_pred)**  **b) Classification**  **from sklearn.metrics import accuracy\_score, recall\_score,**  **precision\_score, confusion\_matrix, roc\_auc\_score confusion\_matrix(y\_test,y\_pred), accuracy\_score(y\_test,y\_pred)**  **recall\_score(y\_test,y\_pred), precision\_score(y\_test,y\_pred)**  **roc\_auc\_score(y\_test,y\_prob)** |

***Dr.Palle,*** [***spalle@emory.edu***](mailto:spalle@emory.edu) ***, Big Data Analytics Program, ECE, Emory University, Atlanta, GA***