PMII Python Cheat Sheet

Import Data	import pandas as pd datadf = pd.read_excel('datadf.xlsx') datadf = pd.read_csv('datadf.csv') datadf = pd.read_table('datadf.txt') *# drop null values, change encoding type* pd.read_csv('datadf.csv', na_values='?', encoding= 'utf-8').dropna()
Read Data info	print(datadf.info()) print(datadf.shape) print(datadf.head())
Split Data	*# split into two parts* train, test = train_test_split(datadf, test_size=0.2, random_state=4) *# split into four parts* X_train, X_test, y_train, y_test = train_test_split(datadf[x_columns], datadf[y_column], train_size = 0.6, random_state = 1)
Logit Model	from sklearn.linear_model import LogisticRegression *# build the logit regression model, using the training dataset* logreg = LogisticRegression() logreg.fit(X_train, y_train) *# generate predicted label for the test dataset* y_pred = logreg.predict(X_test) *# generate predicted probability* pred_probs = logistic_model.predict_proba(datadf[x_columns]) *# create a new column to store the probability* datadf["Prediction"] = pred_probs[:,1]
Decision Tree	from sklearn.tree import DecisionTreeClassifier tree = DecisionTreeClassifier(max_depth = 3) tree.fit(X_train, y_train) y_pred = tree.predict(X_test)
Random Forest	from sklearn.ensemble import RandomForestClassifier rfc = RandomForestClassifier(max_features=10, random_state=1) rfc.fit(X_train, y_train) y_pred = rfc.predict(X_test)
Support Vector	from sklearn.svm import SVC, LinearSVC svc = SVC(C= 1.0, kernel='linear') svc.fit(X_train, y_train) y_pred = svc.predict(X_test)
Neural Network	from sklearn.neural_network import MLPClassifier mlp = MLPClassifier(hidden_layer_sizes=(30,30,30)) mlp.fit(X_train,y_train) y_pred = mlp.predict(X_test)
Model Evaluation	confusion_matrix(y_test, y_pred) tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel() print("Sensitivity = %s" %(tp/(tp+fn))) print("Specificity = %s" %(tn/(tn+fp))) print("Accuracy = %s" %((tn+tp)/(tn+tp+fn+fp))) print(classification_report(y_test, y_pred)) auc_score = roc_auc_score(datadf["actual_label"], datadf["Prediction"])