# Set up default random forest regressor to train model

from sklearn.ensemble import RandomForestRegressor

rf = RandomForestRegressor(random\_state=42)

rf.fit(X\_train,y\_train)

# Required library

from sklearn.metrics import mean\_squared\_error, r2\_score, mean\_absolute\_error

from math import sqrt

from sklearn import model\_selection

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.model\_selection import KFold

from sklearn import preprocessing

from sklearn.preprocessing import StandardScaler

from sklearn.preprocessing import MinMaxScaler

# Compute RMSE on training / test / Entire datasets

y\_pred\_train\_rf = rf.predict(X\_train)

y\_pred\_test\_rf = rf.predict(X\_test)

# Construct a vector of errors for train and test data sets

rf\_err\_train = abs(y\_pred\_train\_rf - y\_train)

rf\_err\_test = abs(y\_pred\_test\_rf - y\_test)

# Compute MSE for Train set, Test set and Entire set

mse\_rf\_train = mean\_squared\_error(y\_train, y\_pred\_train\_rf)

mse\_rf\_test = mean\_squared\_error(y\_test, y\_pred\_test\_rf)

# Compute RMSE for Train set, Test set and Entire set

rmse\_rf\_train = sqrt(abs(mse\_rf\_train))

rmse\_rf\_test = sqrt(abs(mse\_rf\_test))

# Cross Validation Random Forest MSE

seed = 42

kf = model\_selection.KFold(n\_splits=10, random\_state=seed)

rf\_kf = RandomForestRegressor(random\_state=42)

scoring\_kf = 'neg\_mean\_squared\_error'

results\_kf\_train = model\_selection.cross\_val\_score(rf\_kf, X\_train, y\_train, cv=kf, scoring=scoring\_kf)

results\_kf\_test = model\_selection.cross\_val\_score(rf\_kf, X\_test, y\_test, cv=kf, scoring=scoring\_kf)

# Computing RMSE

rmse\_kf\_train = sqrt(abs(results\_kf\_train.mean()))

rmse\_kf\_test = sqrt(abs(results\_kf\_test.mean()))

# Compute Accuracy

acc\_rf\_train = rf.score(X\_train, y\_train)

acc\_rf\_test = rf.score(X\_test, y\_test)

# Print Accuracy & RMSE

print('\nRMSE Summary for Linear Regression')

print('- - - - - - - - - - - - - - - - - - - -')

print("RMSE - Lin Reg - train: %.4f" % rmse\_rf\_train)

print("RMSE - Lin Reg - test: %.4f" % rmse\_rf\_test)

print('\nACCURACY Summary')

print('- - - - - - - - - - - - - - - - - - - -')

print('Accuracy: Lin Reg - train: ', round(100 \* acc\_rf\_train, 2), '%')

print('Accuracy: Lin Reg - test: ', round(100 \* acc\_rf\_test, 2), '%')

print('\nRMSE Summary for 10-cv')

print('- - - - - - - - - - - - - - - - - - - -')

print("RMSE - 10cv - train: %.4f" % rmse\_kf\_train)

print("RMSE - 10cv - test: %.4f" % rmse\_kf\_test)