**Learning Curve for Linear Regression**

***Importing Required Libraries***

**import** pandas **as** pd

**from** sklearn.linear\_model **import** LinearRegression

**from** sklearn.ensemble **import** RandomForestRegressor

**from** sklearn.model\_selection **import** learning\_curve

**import** matplotlib.pyplot **as** plt

***Importing the dataset in a variable and defining training sizes***

linear\_regression **=** pd.read\_csv('Training\_curve\_data.csv')

​train\_sizes**=**[1,50,500,750,1000,1500,2000,2500]

***Defining features and target variable***

features**=**['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8','PC9','PC10','PC11','PC12','PC13','PC14','PC15','PC16','PC17','PC18','PC19']

target **=** 'Overall satisfaction'

**Selecting the Model for running the learning curve**

train\_sizes,train\_scores,validation\_scores**=**learning\_curve(

estimator**=**LinearRegression(),

X**=**linear\_regression[features],

y**=**linear\_regression[target],train\_sizes**=**train\_sizes,cv**=**5,

scoring**=**'neg\_mean\_squared\_error')

**In order to see what learning curve returns**

print('Training scores:\n\n', train\_scores)

print('\n', '\_\_' **\*** 70)

print('\n Validation scores:\n\n', validation\_scores)

**Taking the mean value of each row and also flip the signs of the error scores**

train\_scores\_mean **=** **-**train\_scores.mean(axis **=** 1)

validation\_scores\_mean **=** **-**validation\_scores.mean(axis **=** 1)

print('Mean training scores\n\n', pd.Series(train\_scores\_mean, index **=** train\_sizes))

print('\n', '-' **\*** 20)

print('\nMean validation scores\n\n',pd.Series(validation\_scores\_mean, index **=** train\_sizes))

plt.style.use('seaborn')

plt.plot(train\_sizes, train\_scores\_mean, label **=** 'Training error')

plt.plot(train\_sizes, validation\_scores\_mean, label **=** 'Validation error')

plt.ylabel('MSE', fontsize **=** 20)

plt.xlabel('Training set size', fontsize **=** 20)

plt.title('Learning curves for a Linear regression model', fontsize **=** 22, y **=** 1.03)

plt.legend(fontsize**=**20)

**Learning Curve for Random Forest Classifier**

***Importing Required Libraries***

**import** pandas **as** pd

**from** sklearn.linear\_model **import** LinearRegression

**from** sklearn.ensemble **import** RandomForestRegressor

**from** sklearn.model\_selection **import** learning\_curve

**import** matplotlib.pyplot **as** plt

**Importing the dataset in a variable and defining training sizes**

linear\_regression **=** pd.read\_csv('Training\_curve\_data.csv')

train\_sizes**=**[1,50,500,750,1000,1500,2000,2500]

***Defining features and target variable***

features**=**['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8','PC9','PC10','PC11','PC12','PC13','PC14','PC15','PC16','PC17','PC18','PC19']

target **=** 'Overall satisfaction'

**Selecting a model for running the learning curve**

train\_sizes,train\_scores,validation\_scores**=**learning\_curve(

estimator**=**RandomForestRegressor(),

X**=**linear\_regression[features],

y**=**linear\_regression[target],train\_sizes**=**train\_sizes,cv**=**5,

scoring**=**'neg\_mean\_squared\_error')

train\_scores\_mean **=** **-**train\_scores.mean(axis **=** 1)

validation\_scores\_mean **=** **-**validation\_scores.mean(axis **=** 1)

plt.plot(train\_sizes, train\_scores\_mean, label **=** 'Training error')

plt.plot(train\_sizes, validation\_scores\_mean, label **=** 'Validation error')

plt.ylabel('MSE', fontsize **=** 20)

plt.xlabel('Training set size', fontsize **=** 20)

plt.title('Learning curves for a Random Forest regression model', fontsize **=** 22, y **=** 1.03)

plt.legend(fontsize**=**20)

​