DSC550 Week7

*# split into training and test data again*

# drop some columns

df\_reduced **=** dfcopy**.**drop(['horsepower', 'cylinders'], axis**=**1)

*#df\_reduced*

# use standard scaler

*#make the features standardized distribution*

**from** sklearn.preprocessing **import** StandardScaler

sc **=** StandardScaler()

cols **=** ['displacement','weight','acceleration', 'model year', 'origin']

df\_reduced[cols]**=** sc**.**fit\_transform(df\_reduced[cols])

df\_reduced**.**head()

# Split and train

**from** sklearn.model\_selection **import** train\_test\_split

X **=** dfcopy**.**drop(columns**=**['mpg'])

y **=** dfcopy['mpg']

*#split the data*

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.2, random\_state**=**42)

*#Train an ordinary linear regression on the training data.*

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

regression **=** LinearRegression()

*#fit the regression*

model **=** regression**.**fit(X\_train, y\_train)

evaluate on training set

train\_predicted **=** model**.**predict(X\_train)

evaluate on test set

train\_predicted **=** model**.**predict(X\_test)

reference notebook

https://github.com/shtanriverdi/Clustering-Project-of-Amyotrophic-Lateral-Sclerosis-ALS-/blob/master/Clustering\_Implementation.ipynb

reference on how to do clustering chart

<https://medium.com/@sarahalalawi0/evaluate-the-models-with-different-number-of-clusters-4a8c5109a57>

<https://www.youtube.com/watch?v=DDrSPPwvGtQ&t=205s>