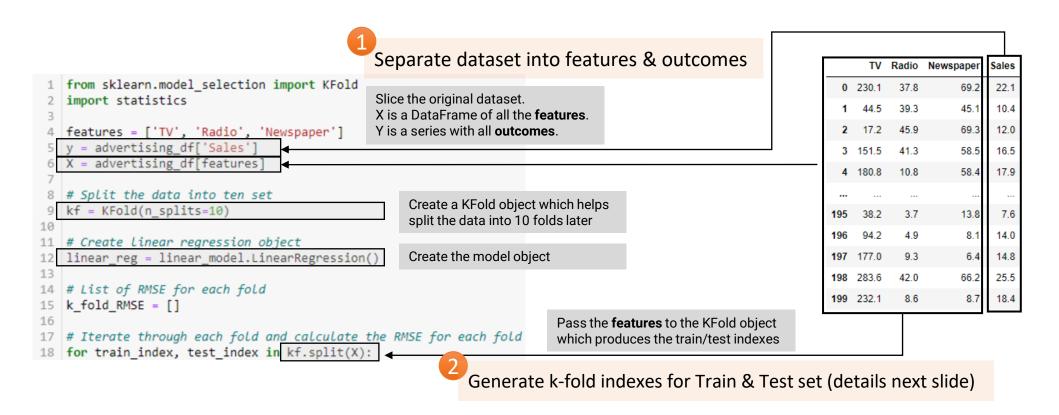


Now, we will cover K-fold cross validation. In the next few slides, we will breakdown the cross-validation process line-by-line.

```
1 from sklearn.model selection import KFold
 2 import statistics
 4 | features = ['TV', 'Radio', 'Newspaper']
 5 X = advertising df[features]
                                                                           Separate dataset into features & outcomes
 6 y = advertising df['Sales']
 8 # Split the data into ten set
 9 kf = KFold(n splits=10)
10
11 # Create linear regression object
12 linear reg = linear model.LinearRegression()
13
14 # List of RMSE for each fold
15 k_fold_RMSE = []
16
17 # Iterate through each fold and calculate the RMSE for each fold
                                                                           Generate k-fold indexes for Train & Test set
18 for train index, test index in kf.split(X):
19
20
       # Extract the training and test data
                                                                           Split dataset into Train/Test sets for the particular iteration
       X train, X test = X.iloc[train index], X.iloc[test index]
21
22
       y train, y test = y.iloc[train index], y.iloc[test index]
23
24
       # Fit model
                                                                           Train model & generate predictions for model evaluation
25
       linear reg model = linear reg.fit(X train, y train)
       y_pred = linear_reg_model.predict(X_test)
26
27
       #Calculate RMSE for the fold and append it
28
       RMSE = mean squared error(y test, y pred, squared=False)
29
       k_fold_RMSE.append(RMSE)
30
31
32 print('The RMSE for each fold is:', k_fold_RMSE)
33 print('The average RMSE is:', statistics.mean(k fold RMSE))
```





<sup>\*</sup> For the Kfold() function, there are optional parameters shuffle=True, random\_state=\_\_ which we recommend for you to use in your personal projects.

```
Generate k-fold indexes for Train & Test
14 # List of RMSE for each fold
15 k fold RMSE = []
                                          set
16
   # Iterate through each fold and calculate the RMSE for each fold
   for train index, test index in kf.split(X):
19
20
       # Extract the training and test data
       X train, X test = X.iloc[train index], X.iloc[test index]
21
22
       y train, y test = y.iloc[train index], y.iloc[test index]
23
24
       # Fit model
25
       linear_reg_model = linear_reg.fit(X_train, y_train)
       y_pred = linear_reg_model.predict(X_test)
26
27
       #Calculate RMSE for the fold and append it
28
29
       RMSE = mean squared error(y test, y pred, squared=False)
30
       k fold RMSE.append(RMSE)
31
   print('The RMSE for each fold is:', k fold RMSE)
33 print('The average RMSE is:', statistics.mean(k fold RMSE))
```

- Each iteration, kf.split(X) returns us 2 lists:
  - The first list comprises 9 folds of the DataFrame row index (we store these indices in train index)
  - The second list comprises 1-fold of the DataFrame row index (we store these indices in test index)
- Observe that in iteration 1, the chosen test index were 0 20, and in iteration 2, it was 20 - 39. This carries on until iteration 10, where all the 200 rows of data would have a chance at being the test data.

# What kf.split(X) does

```
Iteration 1
     182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199
                                     9 10 11 12 13 14 15 16 17 18 19]
test_index (1 fold)
```

#### **Iteration 2**

```
81 82
182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199
```

**Iteration 10** 

33 print('The average RMSE is:', statistics.mean(k fold RMSE))

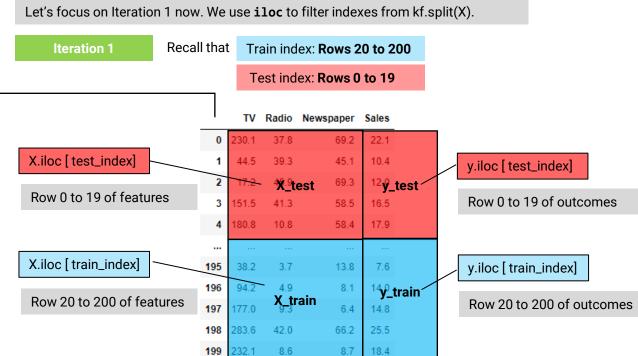
```
Split dataset into Train/Test sets for the particular iteration
```

14 # List of RMSE for each fold 15 k fold RMSE = [] 16 Let's focus on Iteration 1 now. We use **iloc** to filter indexes from kf.split(X). Details next slide. 17 # Iterate through each fold and calculate the RMSE for each fold for train index, test index in kf.split(X): 19 # Extract the training and test data **Iteration 1** 20 21 X train, X test = X.iloc train index], X.iloc test index] 22 y train, y test = y.iloc train index], y.iloc test index] 23 24 # Fit model 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 25 linear\_reg\_model = linear\_reg.fit(X\_train, y\_train) 26 y pred = linear reg model.predict(X test) 27 #Calculate RMSE for the fold and append it 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 28 29 RMSE = mean squared error(y test, y pred, squared=False) 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199] 30 k fold RMSE.append(RMSE) 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19] test\_index (1 fold) 31 32 print('The RMSE for each fold is:', k\_fold\_RMSE)

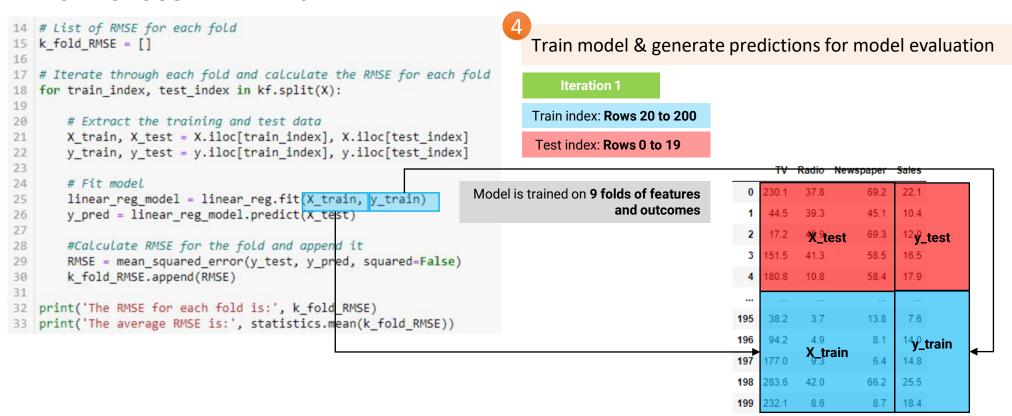


```
Split dataset into Train/Test sets for the particular iteration
```

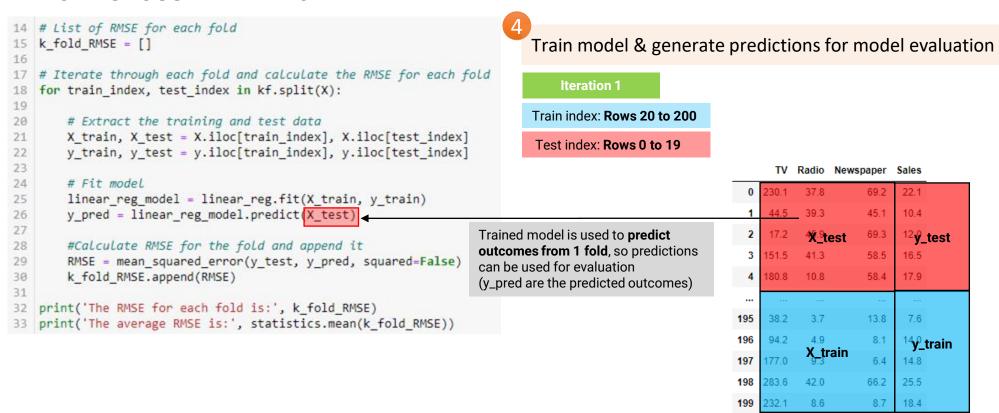
```
14 # List of RMSE for each fold
15 k fold RMSE = []
```













```
14 # List of RMSE for each fold
                                                                               Train model & generate predictions for model evaluation
15 k fold RMSE = []
16
17 # Iterate through each fold and calculate the RMSE for each fold
                                                                                   Iteration 1
18 for train index, test index in kf.split(X):
19
                                                                               Train index: Rows 20 to 200
        # Extract the training and test data
20
21
        X train, X test = X.iloc[train index], X.iloc[test index]
                                                                               Test index: Rows 0 to 19
        y train, y test = y.iloc[train index], y.iloc[test index]
22
23
                                                                                                                    TV Radio Newspaper Sales
24
        # Fit model
                                                                                                                        37.8
25
        linear_reg_model = linear_reg.fit(X_train, y_train)
26
        y pred = linear reg model.predict(X test)
                                                                                                                   44.5 39.3
                                                                                                                                  45.1
                                                                                                                                        10.4
27
                                                                                                                2
                                                                                                                         <sup>4</sup>X<sub>2</sub>test
                                                                                                                                  69.3
                                                                                                                   17.2
                                                                                                                                        <sup>12</sup>

√_test
        #Calculate RMSE for the fold and append it
28
                                                                       Now we calculate the error of
                                                                                                                        41.3
                                                                                                                3
                                                                                                                   151.5
                                                                                                                                  58.5
                                                                                                                                        16.5
29
        RMSE = mean_squared_error(y_test, y_pred, squared=False)
                                                                       the predictions (y_pred)
                                                                                                                                  58.4 17.9
30
        k fold RMSE.append(RMSE)
                                                                       versus the actual outcomes (y_test)
                                                                                                                4
                                                                                                                   180.8 10.8
31
32 print('The RMSE for each fold is:', k_fold_RMSE)
                                                                                                                   38.2
                                                                                                                         3.7
                                                                                                                                  13.8
                                                                                                                                        7.6
33 print('The average RMSE is:', statistics.mean(k_fold_RMSE))
                                                                                                              196
                                                                                                                   94.2
                                                                                                                        4.9
                                                                                                                                   8.1
                                                                                                                                        14y_train
                                                                                                                         X_train
                                                                                                                                   6.4
                                                                                                              197
                                                                                                                   177.0
                                                                                                                                        14.8
                                                                                                              198 283.6
                                                                                                                                       25.5
                                                                                                                        42.0
                                                                                                                                  66.2
                                                                                                                                       18.4
                                                                                                              199 232.1 8.6
                                                                                                                                   8.7
```

```
14 # List of RMSE for each fold
15 k fold RMSE = []
16
17 # Iterate through each fold and calculate the RMSE for each fold
18 for train index, test index in kf.split(X):
19
       # Extract the training and test data
20
21
       X train, X test = X.iloc[train index], X.iloc[test index]
       y_train, y_test = y.iloc[train_index], y.iloc[test_index]
22
23
24
       # Fit model
25
       linear_reg_model = linear_reg.fit(X_train, y_train)
26
       y pred = linear reg model.predict(X test)
27
       #Calculate RMSE for the fold and append it
28
29
       RMSE = mean_squared_error(y_test, y_pred, squared=False)
30
        k fold RMSE.append(RMSE)
31
32 print('The RMSE for each fold is:', k fold RMSE)
33 print('The average RMSE is:', statistics.mean(k fold RMSE))
```

Train model & generate predictions for model evaluation

. . . . . . . . . .

Iteration 1

Train index: Rows 20 to 200

Test index: Rows 0 to 19

Iteration 2

Train index: Rows 0 - 19, 40 - 200

.....

Test index: Rows 20 - 39

Finally, we append the RMSE of this fold to a list, ... and on to the next iteration where we deal with a new set of train / test indexes!