

LASSO Models

Justin Post

Recap

- Judge the model's effectiveness using a **Loss** function
- Often split data into a training and test set
 - Perhaps 70/30 or 80/20
- Cross-validation gives a way to use more of the data while still seeing how the model does on test data
 - Commonly 5 fold or 10 fold is done
 - Once a best model is chosen, model is refit on entire data set

Recap

- Judge the model's effectiveness using a **Loss** function
- Often split data into a training and test set
 - Perhaps 70/30 or 80/20
- Cross-validation gives a way to use more of the data while still seeing how the model does on test data
 - Commonly 5 fold or 10 fold is done
 - Once a best model is chosen, model is refit on entire data set
- Often use both! Let's see why by introducing a model with a **tuning parameter**

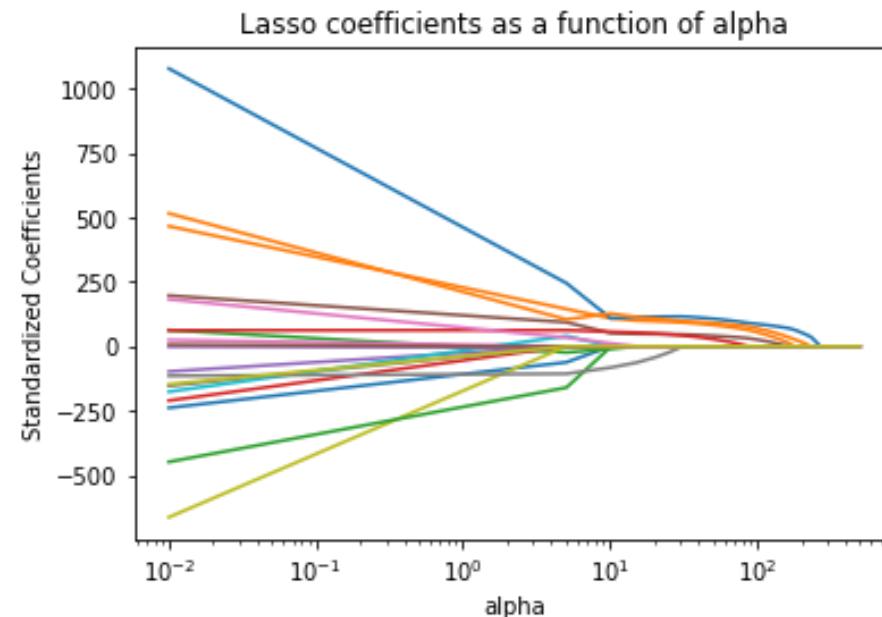
LASSO Model

- Least Angle Subset and Selection Operator or LASSO
 - Similar to Least Squares but a penalty is placed on the sum of the absolute values of the regression coefficients
 - $\alpha (>0)$ is called a tuning parameter

$$\min_{\beta' s} \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{1i} + \dots + \beta_p x_{pi}))^2 + \alpha \sum_{j=1}^p |\beta_j|$$

LASSO Model

- Least Angle Subset and Selection Operator or LASSO
 - Similar to Least Squares but a penalty is placed on the sum of the absolute values of the regression coefficients
 - Sets coefficients to 0 as you 'shrink'!



Tuning Parameter

- When choosing the tuning parameter, we are really considering a **family of models!**
- Consider an $\alpha = 0.1$ (small amount of shrinkage here)

```
from sklearn import linear_model
lasso = linear_model.Lasso(alpha=0.1)
lasso.fit(bike_data[["year", "log_km_driven"]].values, bike_data["log_selling_price"].values)

print(lasso.intercept_, lasso.coef_)

## -164.6120947286609 [ 0.08761607 -0.11092474]
```

Tuning Parameter

- When choosing the tuning parameter, we are really considering a **family of models!**
- Consider an $\alpha = 0.1$ (small amount of shrinkage here)

```
from sklearn import linear_model
lasso = linear_model.Lasso(alpha=0.1)
lasso.fit(bike_data[["year", "log_km_driven"]].values, bike_data["log_selling_price"].values)

print(lasso.intercept_, lasso.coef_)

## -164.6120947286609 [ 0.08761607 -0.11092474]
```

- Consider an $\alpha = 1.05$ (a larger amount of shrinkage)

```
lasso = linear_model.Lasso(alpha=1.05)
lasso.fit(bike_data[["year", "log_km_driven"]].values, bike_data["log_selling_price"].values)

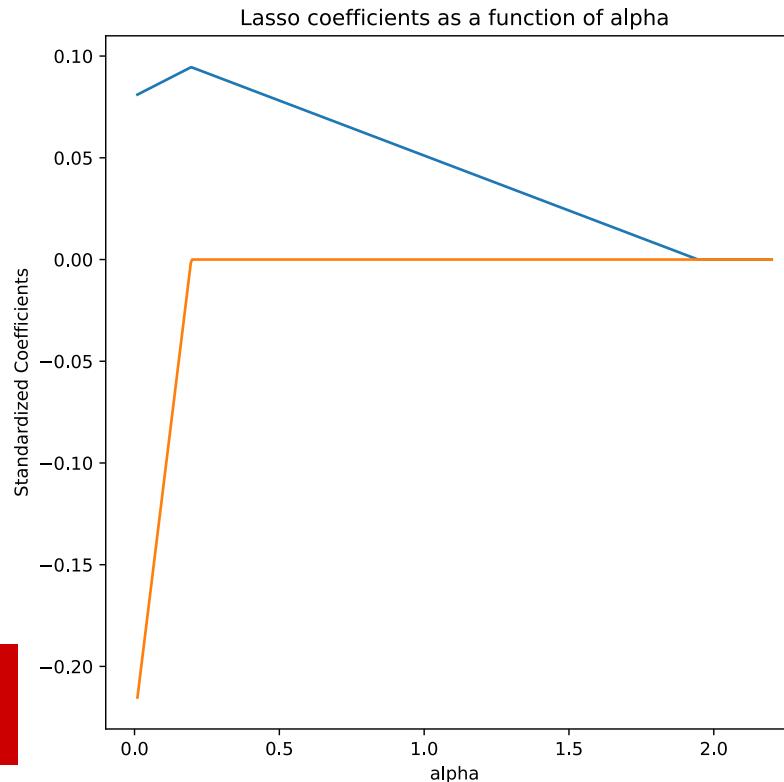
print(lasso.intercept_, lasso.coef_)

## -86.65630892150766 [ 0.04835598 -0. ]
```

LASSO Fits Visual

- Perfect place for CV to help select the best α !

```
## (-0.0995000000000003, 2.3095000000000003, -0.23074900910594778, 0.10996726863779523)
```



Using CV to Select the Tuning Parameter

- Return the optimal α using LassoCV

```
from sklearn.linear_model import LassoCV
lasso_mod = LassoCV(cv=5, random_state=0, alphas = np.linspace(0,2.2,100)) \
    .fit(bike_data[["year", "log_km_driven"]].values,
        bike_data["log_selling_price"].values)

## C:\python\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:634: UserWarning: Coordinate descent without L1 regularization did not converge
##     model = cd_fast.enet_coordinate_descent_gram(
## C:\python\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:634: UserWarning: Coordinate descent without L1 regularization did not converge
##     model = cd_fast.enet_coordinate_descent_gram(
## C:\python\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:634: UserWarning: Coordinate descent without L1 regularization did not converge
##     model = cd_fast.enet_coordinate_descent_gram(
## C:\python\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:634: UserWarning: Coordinate descent without L1 regularization did not converge
##     model = cd_fast.enet_coordinate_descent_gram(
## C:\python\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:634: UserWarning: Coordinate descent without L1 regularization did not converge
##     model = cd_fast.enet_coordinate_descent_gram(
## C:\python\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:634: UserWarning: Coordinate descent without L1 regularization did not converge
##     model = cd_fast.enet_coordinate_descent_gram(
## C:\python\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:1771: UserWarning: With alpha=0, this algorithm does not converge
##     model.fit(X, y)
## C:\python\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:648: UserWarning: Coordinate descent with no regularization did not converge
##     model = cd_fast.enet_coordinate_descent(
## C:\python\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:648: ConvergenceWarning: Objective did not converge
##     model = cd_fast.enet_coordinate_descent(
```

Using CV to Select the Tuning Parameter

- Return the optimal α using `LassoCV`

```
pd.DataFrame(zip(lasso_mod.alphas_, lasso_mod.mse_path_), columns = ["alpha_value", "MSE_by_fold"])

##      alpha_value          MSE_by_fold
## 0    0.000000 [0.5496710875578059, 0.6805679103740427, 0.500...
## 1    0.022222 [0.5496710875578059, 0.6805679103740427, 0.500...
## 2    0.044444 [0.5496710875578059, 0.6805679103740427, 0.500...
## 3    0.066667 [0.5496710875578059, 0.6805679103740427, 0.500...
## 4    0.088889 [0.5496710875578059, 0.6805679103740427, 0.500...
## ..
## 95   2.111111 [0.30465461356828655, 0.3626276356362589, 0.19...
## 96   2.133333 [0.2998496943347467, 0.35411026477928403, 0.18...
## 97   2.155556 [0.29626758731409036, 0.3464595664117418, 0.18...
## 98   2.177778 [0.2939082925063059, 0.3396755405336323, 0.182...
## 99   2.200000 [0.2927721424754243, 0.33375857421410604, 0.18...

## [100 rows x 2 columns]
```

Using CV to Select the Tuning Parameter

- Return the optimal α using `LassoCV`

```
fit_info = np.array(list(zip(lasso_mod.alphas_, np.mean(lasso_mod.mse_path_, axis = 1))))  
fit_info[fit_info[:,0].argsort()]
```

```
## array([[0.          ,  0.26832555],  
##        [0.02222222,  0.26904464],  
##        [0.04444444,  0.27086204],  
##        [0.06666667,  0.27377741],  
##        [0.08888889,  0.27779157],  
##        [0.11111111,  0.28290414],  
##        [0.13333333,  0.28911508],  
##        [0.15555556,  0.29642441],  
##        [0.17777778,  0.30483239],  
##        [0.2          ,  0.30967893],  
##        [0.22222222,  0.31098715],  
##        [0.24444444,  0.31159763],  
##        [0.26666667,  0.31226415],  
##        [0.28888889,  0.31298674],  
##        [0.31111111,  0.31376537],  
##        [0.33333333,  0.31460007],  
##        [0.35555556,  0.31549081],  
##        [0.37777778,  0.31643761],  
##        [0.46666667,  0.32078535],  
##        [0.49999999,  0.32399458],  
##        [0.53333333,  0.32718391],  
##        [0.56666667,  0.33035334],  
##        [0.6          ,  0.33350381],  
##        [0.63333333,  0.33663434],  
##        [0.66666667,  0.34074501],  
##        [0.7          ,  0.34483581],  
##        [0.73333333,  0.34890671],  
##        [0.76666667,  0.35295771],  
##        [0.8          ,  0.35698881],  
##        [0.83333333,  0.36099991],  
##        [0.86666667,  0.36498111],  
##        [0.9          ,  0.36894241],  
##        [0.93333333,  0.37287481],  
##        [0.96666667,  0.37678831],  
##        [1.          ,  0.38068281],  
##        [1.03333333,  0.38455841],  
##        [1.06666667,  0.38841521],  
##        [1.1          ,  0.39225321],  
##        [1.13333333,  0.39607241],  
##        [1.16666667,  0.40087281],  
##        [1.2          ,  0.40565441],  
##        [1.23333333,  0.41041721],  
##        [1.26666667,  0.41515221],  
##        [1.3          ,  0.41986841],  
##        [1.33333333,  0.42456601],  
##        [1.36666667,  0.42924501],  
##        [1.4          ,  0.43390521],  
##        [1.43333333,  0.43854661],  
##        [1.46666667,  0.44317021],  
##        [1.5          ,  0.44777501],  
##        [1.53333333,  0.45236101],  
##        [1.56666667,  0.45692821],  
##        [1.6          ,  0.46147661],  
##        [1.63333333,  0.46599621],  
##        [1.66666667,  0.47049601],  
##        [1.7          ,  0.47497601],  
##        [1.73333333,  0.47943701],  
##        [1.76666667,  0.48387921],  
##        [1.8          ,  0.48830261],  
##        [1.83333333,  0.49270721],  
##        [1.86666667,  0.49709301],  
##        [1.9          ,  0.50145901],  
##        [1.93333333,  0.50579621],  
##        [1.96666667,  0.51011461],  
##        [2.          ,  0.51441321],  
##        [2.03333333,  0.51869201],  
##        [2.06666667,  0.52294101],  
##        [2.1          ,  0.52716921],  
##        [2.13333333,  0.53137761],  
##        [2.16666667,  0.53556521],  
##        [2.2          ,  0.53973201],  
##        [2.23333333,  0.54387801],  
##        [2.26666667,  0.54799321],  
##        [2.3          ,  0.55208761],  
##        [2.33333333,  0.55616121],  
##        [2.36666667,  0.56021401],  
##        [2.4          ,  0.56424601],  
##        [2.43333333,  0.56825701],  
##        [2.46666667,  0.57224721],  
##        [2.5          ,  0.57621701],  
##        [2.53333333,  0.58016621],  
##        [2.56666667,  0.58410501],  
##        [2.6          ,  0.58792301],  
##        [2.63333333,  0.59172101],  
##        [2.66666667,  0.59549801],  
##        [2.7          ,  0.59925401],  
##        [2.73333333,  0.60298901],  
##        [2.76666667,  0.60669301],  
##        [2.8          ,  0.61037601],  
##        [2.83333333,  0.61403801],  
##        [2.86666667,  0.61767901],  
##        [2.9          ,  0.62129801],  
##        [2.93333333,  0.62489601],  
##        [2.96666667,  0.62847201],  
##        [3.          ,  0.63202601],  
##        [3.03333333,  0.63555301],  
##        [3.06666667,  0.63905801],  
##        [3.1          ,  0.64254101],  
##        [3.13333333,  0.64599901],  
##        [3.16666667,  0.64943501],  
##        [3.2          ,  0.65284801],  
##        [3.23333333,  0.65623801],  
##        [3.26666667,  0.65960601],  
##        [3.3          ,  0.66294201],  
##        [3.33333333,  0.66624501],  
##        [3.36666667,  0.66951601],  
##        [3.4          ,  0.67275401],  
##        [3.43333333,  0.67595901],  
##        [3.46666667,  0.67914101],  
##        [3.5          ,  0.68229901],  
##        [3.53333333,  0.68543501],  
##        [3.56666667,  0.68855701],  
##        [3.6          ,  0.69165701],  
##        [3.63333333,  0.69473601],  
##        [3.66666667,  0.69780101],  
##        [3.7          ,  0.69984601],  
##        [3.73333333,  0.70286601],  
##        [3.76666667,  0.70586401],  
##        [3.8          ,  0.70884101],  
##        [3.83333333,  0.71180001],  
##        [3.86666667,  0.71473701],  
##        [3.9          ,  0.71765101],  
##        [3.93333333,  0.72054301],  
##        [3.96666667,  0.72341301],  
##        [4.          ,  0.72625101],  
##        [4.03333333,  0.72904801],  
##        [4.06666667,  0.73181301],  
##        [4.1          ,  0.73453501],  
##        [4.13333333,  0.73721501],  
##        [4.16666667,  0.74085301],  
##        [4.2          ,  0.74343801],  
##        [4.23333333,  0.74598101],  
##        [4.26666667,  0.74848101],  
##        [4.3          ,  0.75092601],  
##        [4.33333333,  0.75332101],  
##        [4.36666667,  0.75566601],  
##        [4.4          ,  0.75794101],  
##        [4.43333333,  0.76018601],  
##        [4.46666667,  0.76239101],  
##        [4.5          ,  0.76455601],  
##        [4.53333333,  0.76668101],  
##        [4.56666667,  0.76876601],  
##        [4.6          ,  0.77080101],  
##        [4.63333333,  0.77280601],  
##        [4.66666667,  0.77477101],  
##        [4.7          ,  0.77668601],  
##        [4.73333333,  0.77858101],  
##        [4.76666667,  0.78044601],  
##        [4.8          ,  0.78228101],  
##        [4.83333333,  0.78410601],  
##        [4.86666667,  0.78592101],  
##        [4.9          ,  0.78764601],  
##        [4.93333333,  0.78945101],  
##        [4.96666667,  0.79122601],  
##        [5.          ,  0.79296101],  
##        [5.03333333,  0.79468601],  
##        [5.06666667,  0.79638101],  
##        [5.1          ,  0.79804601],  
##        [5.13333333,  0.79968101],  
##        [5.16666667,  0.80127601],  
##        [5.2          ,  0.80281101],  
##        [5.23333333,  0.80432601],  
##        [5.26666667,  0.80581101],  
##        [5.3          ,  0.80726601],  
##        [5.33333333,  0.80868101],  
##        [5.36666667,  0.80998601],  
##        [5.4          ,  0.81128101],  
##        [5.43333333,  0.81254601],  
##        [5.46666667,  0.81378101],  
##        [5.5          ,  0.81497601],  
##        [5.53333333,  0.81614101],  
##        [5.56666667,  0.81727601],  
##        [5.6          ,  0.81838101],  
##        [5.63333333,  0.81946601],  
##        [5.66666667,  0.82052101],  
##        [5.7          ,  0.82154601],  
##        [5.73333333,  0.82255101],  
##        [5.76666667,  0.82353601],  
##        [5.8          ,  0.82448101],  
##        [5.83333333,  0.82540601],  
##        [5.86666667,  0.82631101],  
##        [5.9          ,  0.82717601],  
##        [5.93333333,  0.82800101],  
##        [5.96666667,  0.82880601],  
##        [6.          ,  0.82956101],  
##        [6.03333333,  0.83026601],  
##        [6.06666667,  0.83092101],  
##        [6.1          ,  0.83152601],  
##        [6.13333333,  0.83210101],  
##        [6.16666667,  0.83264601],  
##        [6.2          ,  0.83314101],  
##        [6.23333333,  0.83358601],  
##        [6.26666667,  0.83401101],  
##        [6.3          ,  0.83442601],  
##        [6.33333333,  0.83482101],  
##        [6.36666667,  0.83519601],  
##        [6.4          ,  0.83554101],  
##        [6.43333333,  0.83586601],  
##        [6.46666667,  0.83616101],  
##        [6.5          ,  0.83641601],  
##        [6.53333333,  0.83664101],  
##        [6.56666667,  0.83684601],  
##        [6.6          ,  0.83703101],  
##        [6.63333333,  0.83720601],  
##        [6.66666667,  0.83734101],  
##        [6.7          ,  0.83744601],  
##        [6.73333333,  0.83752101],  
##        [6.76666667,  0.83757601],  
##        [6.8          ,  0.83761101],  
##        [6.83333333,  0.83764601],  
##        [6.86666667,  0.83766101],  
##        [6.9          ,  0.83764601],  
##        [6.93333333,  0.83762101],  
##        [6.96666667,  0.83757601],  
##        [7.          ,  0.83751101],  
##        [7.03333333,  0.83744601],  
##        [7.06666667,  0.83736101],  
##        [7.1          ,  0.83725601],  
##        [7.13333333,  0.83713101],  
##        [7.16666667,  0.83700601],  
##        [7.2          ,  0.83684101],  
##        [7.23333333,  0.83664601],  
##        [7.26666667,  0.83642101],  
##        [7.3          ,  0.83614601],  
##        [7.33333333,  0.83584101],  
##        [7.36666667,  0.83549601],  
##        [7.4          ,  0.83514101],  
##        [7.43333333,  0.83472601],  
##        [7.46666667,  0.83428101],  
##        [7.5          ,  0.83374601],  
##        [7.53333333,  0.83319101],  
##        [7.56666667,  0.83253601],  
##        [7.6          ,  0.83184101],  
##        [7.63333333,  0.83109601],  
##        [7.66666667,  0.83024101],  
##        [7.7          ,  0.83024101],  
##        [7.73333333,  0.83009601],  
##        [7.76666667,  0.82984101],  
##        [7.8          ,  0.82944601],  
##        [7.83333333,  0.82899101],  
##        [7.86666667,  0.82843601],  
##        [7.9          ,  0.82774101],  
##        [7.93333333,  0.82698601],  
##        [7.96666667,  0.82612101],  
##        [8.          ,  0.82514601],  
##        [8.03333333,  0.82419101],  
##        [8.06666667,  0.82313601],  
##        [8.1          ,  0.82204101],  
##        [8.13333333,  0.82084601],  
##        [8.16666667,  0.81954101],  
##        [8.2          ,  0.81814601],  
##        [8.23333333,  0.81664101],  
##        [8.26666667,  0.81503601],  
##        [8.3          ,  0.81334101],  
##        [8.33333333,  0.81164601],  
##        [8.36666667,  0.80983601],  
##        [8.4          ,  0.80794101],  
##        [8.43333333,  0.80593601],  
##        [8.46666667,  0.80382601],  
##        [8.5          ,  0.80161601],  
##        [8.53333333,  0.80029101],  
##        [8.56666667,  0.79885601],  
##        [8.6          ,  0.79731101],  
##        [8.63333333,  0.79575601],  
##        [8.66666667,  0.79418101],  
##        [8.7          ,  0.79250601],  
##        [8.73333333,  0.79081601],  
##        [8.76666667,  0.78901601],  
##        [8.8          ,  0.78711601],  
##        [8.83333333,  0.78519601],  
##        [8.86666667,  0.78316601],  
##        [8.9          ,  0.78102601],  
##        [8.93333333,  0.77887601],  
##        [8.96666667,  0.77661601],  
##        [9.          ,  0.77424601],  
##        [9.03333333,  0.77186601],  
##        [9.06666667,  0.76947601],  
##        [9.1          ,  0.76707601],  
##        [9.13333333,  0.76466601],  
##        [9.16666667,  0.76224601],  
##        [9.2          ,  0.76071601],  
##        [9.23333333,  0.75817601],  
##        [9.26666667,  0.75562601],  
##        [9.3          ,  0.75306601],  
##        [9.33333333,  0.75049601],  
##        [9.36666667,  0.74791601],  
##        [9.4          ,  0.74532601],  
##        [9.43333333,  0.74272601],  
##        [9.46666667,  0.74011601],  
##        [9.5          ,  0.73749601],  
##        [9.53333333,  0.73486601],  
##        [9.56666667,  0.73222601],  
##        [9.6          ,  0.72957601],  
##        [9.63333333,  0.72691601],  
##        [9.66666667,  0.72424601],  
##        [9.7          ,  0.72156601],  
##        [9.73333333,  0.71887601],  
##        [9.76666667,  0.71617601],  
##        [9.8          ,  0.71346601],  
##        [9.83333333,  0.71074601],  
##        [9.86666667,  0.70801601],  
##        [9.9          ,  0.70527601],  
##        [9.93333333,  0.70252601],  
##        [9.96666667,  0.70076601],  
##        [10.          ,  0.69899601],  
##        [10.03333333,  0.69621601],  
##        [10.06666667,  0.69342601],  
##        [10.1          ,  0.69062601],  
##        [10.13333333,  0.68781601],  
##        [10.16666667,  0.68499601],  
##        [10.2          ,  0.68216601],  
##        [10.23333333,  0.67932601],  
##        [10.26666667,  0.67647601],  
##        [10.3          ,  0.67361601],  
##        [10.33333333,  0.67074601],  
##        [10.36666667,  0.66786601],  
##        [10.4          ,  0.66497601],  
##        [10.43333333,  0.66207601],  
##        [10.46666667,  0.65916601],  
##        [10.5          ,  0.65624601],  
##        [10.53333333,  0.65331601],  
##        [10.56666667,  0.65037601],  
##        [10.6          ,  0.64742601],  
##        [10.63333333,  0.64446601],  
##        [10.66666667,  0.64149601],  
##        [10.7          ,  0.63851601],  
##        [10.73333333,  0.63552601],  
##        [10.76666667,  0.63253601],  
##        [10.8          ,  0.62953601],  
##        [10.83333333,  0.62652601],  
##        [10.86666667,  0.62351601],  
##        [10.9          ,  0.62049601],  
##        [10.93333333,  0.61746601],  
##        [10.96666667,  0.61442601],  
##        [11.          ,  0.61137601],  
##        [11.03333333,  0.60831601],  
##        [11.06666667,  0.60524601],  
##        [11.1          ,  0.60216601],  
##        [11.13333333,  0.59907601],  
##        [11.16666667,  0.59597601],  
##        [11.2          ,  0.59286601],  
##        [11.23333333,  0.58974601],  
##        [11.26666667,  0.58661601],  
##        [11.3          ,  0.58347601],  
##        [11.33333333,  0.58032601],  
##        [11.36666667,  0.57716601],  
##        [11.4          ,  0.57399601],  
##        [11.43333333,  0.57081601],  
##        [11.46666667,  0.56762601],  
##        [11.5          ,  0.56442601],  
##        [11.53333333,  0.56121601],  
##        [11.56666667,  0.55799601],  
##        [11.6          ,  0.55476601],  
##        [11.63333333,  0.55152601],  
##        [11.66666667,  0.54827601],  
##        [11.7          ,  0.54501601],  
##        [11.73333333,  0.54174601],  
##        [11.76666667,  0.53846601],  
##        [11.8          ,  0.53517601],  
##        [11.83333333,  0.53187601],  
##        [11.86666667,  0.52856601],  
##        [11.9          ,  0.52524601],  
##        [11.93333333,  0.52191601],  
##        [11.96666667,  0.51857601],  
##        [12.          ,  0.51522601],  
##        [12.03333333,  0.51186601],  
##        [12.06666667,  0.50849601],  
##        [12.1          ,  0.50511601],  
##        [12.13333333,  0.50172601],  
##        [12.16666667,  0.49832601],  
##        [12.2          ,  0.49491601],  
##        [12.23333333,  0.49149601],  
##        [12.26666667,  0.48806601],  
##        [12.3          ,  0.48462601],  
##        [12.33333333,  0.48117601],  
##        [12.36666667,  0.47771601],  
##        [12.4          ,  0.47424601],  
##        [12.43333333,  0.47076601],  
##        [12.46666667,  0.46727601],  
##        [12.5          ,  0.46377601],  
##        [12.53333333,  0.46026601],  
##        [12.56666667,  0.45674601],  
##        [12.6          ,  0.45321601],  
##        [12.63333333,  0.44967601],  
##        [12.66666667,  0.44612601],  
##        [12.7          ,  0.44256601],  
##        [12.73333333,  0.43900601],  
##        [12.76666667,  0.43543601],  
##        [12.8          ,  0.43185601],  
##        [12.83333333,  0.42827601],  
##        [12.86666667,  0.42468601],  
##        [12.9          ,  0.42108601],  
##        [12.93333333,  0.41747601],  
##        [12.96666667,  0.41385601],  
##        [13.          ,  0.41022601],  
##        [13.03333333,  0.40656601],  
##        [13.06666667,  0.40288601],  
##        [13.1          ,  0.40018601],  
##        [13.13333333,  0.39646601],  
##        [13.16666667,  0.39272601],  
##        [13.2          ,  0
```

Using CV to Select the Tuning Parameter

- Now fit using that optimal α

```
lasso_best = linear_model.Lasso(lasso_mod.alpha_) #warning thrown since we are using 0, but can ignore
lasso_best.fit(bike_data[["year", "log_km_driven"]].values, bike_data["log_selling_price"].values)

print(lasso_best.intercept_, lasso_best.coef_)

## -148.79329107788135 [ 0.0803366 -0.22686129]
```

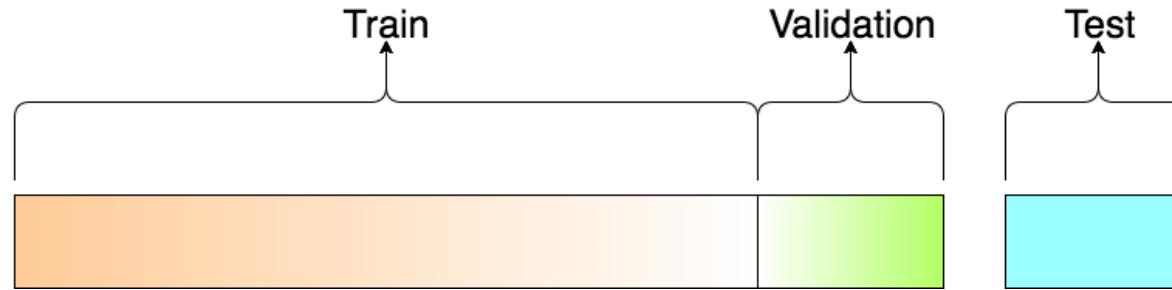
So Do We Just Need CV?

Sometimes!

- If you are only considering one type of model, you can use just a training/test set or k-fold CV to select the best version of that model
- When you have multiple types of models to choose from, usually use both!
 - When we use the test set too much, we may have '**data leakage**'
 - Essentially we end up training our models to the test set by using it too much

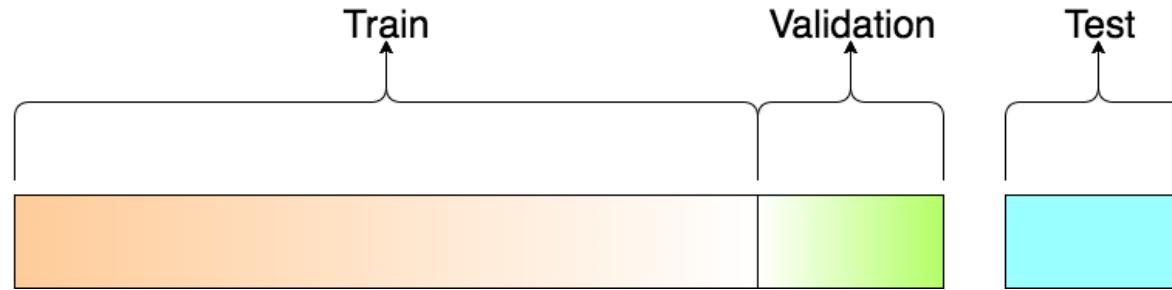
Training/Validation/Test or CV/Test

- Instead, we sometimes split into a training, validation, and test set
- CV can be used to replace the validation set!



Training/Validation/Test or CV/Test

- Instead, we sometimes split into a training, validation, and test set
- CV can be used to replace the validation set!



- Compare only the **best** model from each model type on the test set

Recap

- LASSO is similar to an MLR model but shrinks coefficients and may set some to 0
 - Tuning parameter must be chosen (usually by CV)
- Training/Test split gives us a way to validate our model's performance
 - CV can be used on the training set to select **tuning parameters**
 - Helps determine the 'best' model for a class of models
- With many competing model types, determine the best from each type check performance on the test set