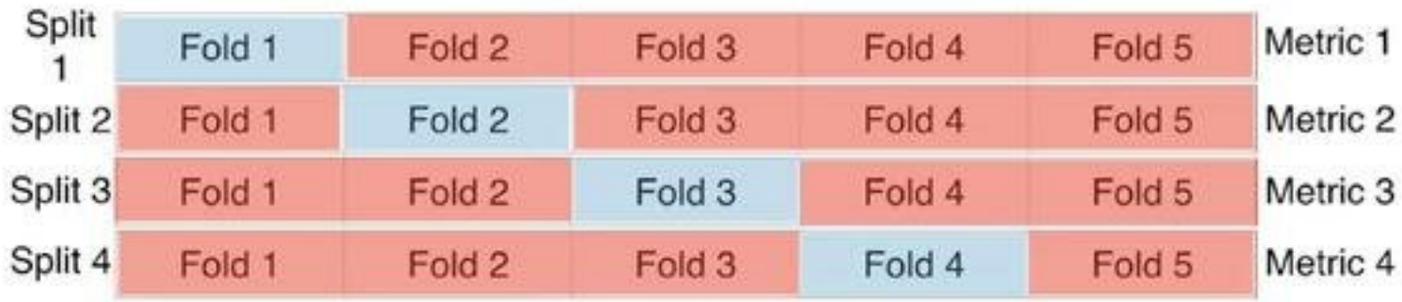
# Regression mechanics

- y = ax + b
  - y = target
  - x = single feature
  - a, b = parameters of model
- How do we choose a and b?
- Define an error functions for any given line
  - Choose the line that minimizes the error function

Basics of Linear Regression

# Cross-validation basics




Training data  Test data

# Why regularize?

- Recall: Linear regression minimizes a loss function
- It chooses a coefficient for each feature variable
- Large coefficients can lead to overfitting
- Penalizing large coefficients: Regularization



# Regularized regression

SUPERVISED LEARNING WITH SCIKIT-LEARN

# Why regularize?

- Recall: Linear regression minimizes a loss function
- It chooses a coefficient for each feature variable
- Large coefficients can lead to overfitting
- Penalizing large coefficients: Regularization



# Ridge regression

- Loss function = OLS loss function +

$$\alpha * \sum_{i=1}^{n} {a_i}^2$$

- Alpha: Parameter we need to choose
- Picking alpha here is similar to picking k in k-NN
- Hyperparameter tuning (More in Chapter 3)
- Alpha controls model complexity
  - Alpha = 0: We get back OLS (Can lead to overfitting)
  - Very high alpha: Can lead to underfitting



# Ridge regression in scikit-learn

```
from sklearn.linear_model import Ridge
X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size = 0.3, random_state=42)
ridge = Ridge(alpha=0.1, normalize=True)
ridge.fit(X_train, y_train)
ridge_pred = ridge.predict(X_test)
ridge.score(X_test, y_test)
```

```
0.699693827512731 79
```



# Lasso regression

- Loss function = OLS loss function +

$$\alpha * \sum_{i=1}^{n} |a_i|$$



# Lasso regression in scikit-learn

```
from sklearn.linear_model import Lasso
X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size = 0.3, random_state=42)
lasso = Lasso(alpha=0.1, normalize=True)
lasso.fit(X_train, y_train)
lasso_pred = lasso.predict(X_test)
lasso.score(X_test, y_test)
```

```
0.59502295353285506
```



# Lasso regression for feature selection

- Can be used to select important features of a dataset
- Shrinks the coefficients of less important features to exactly 0

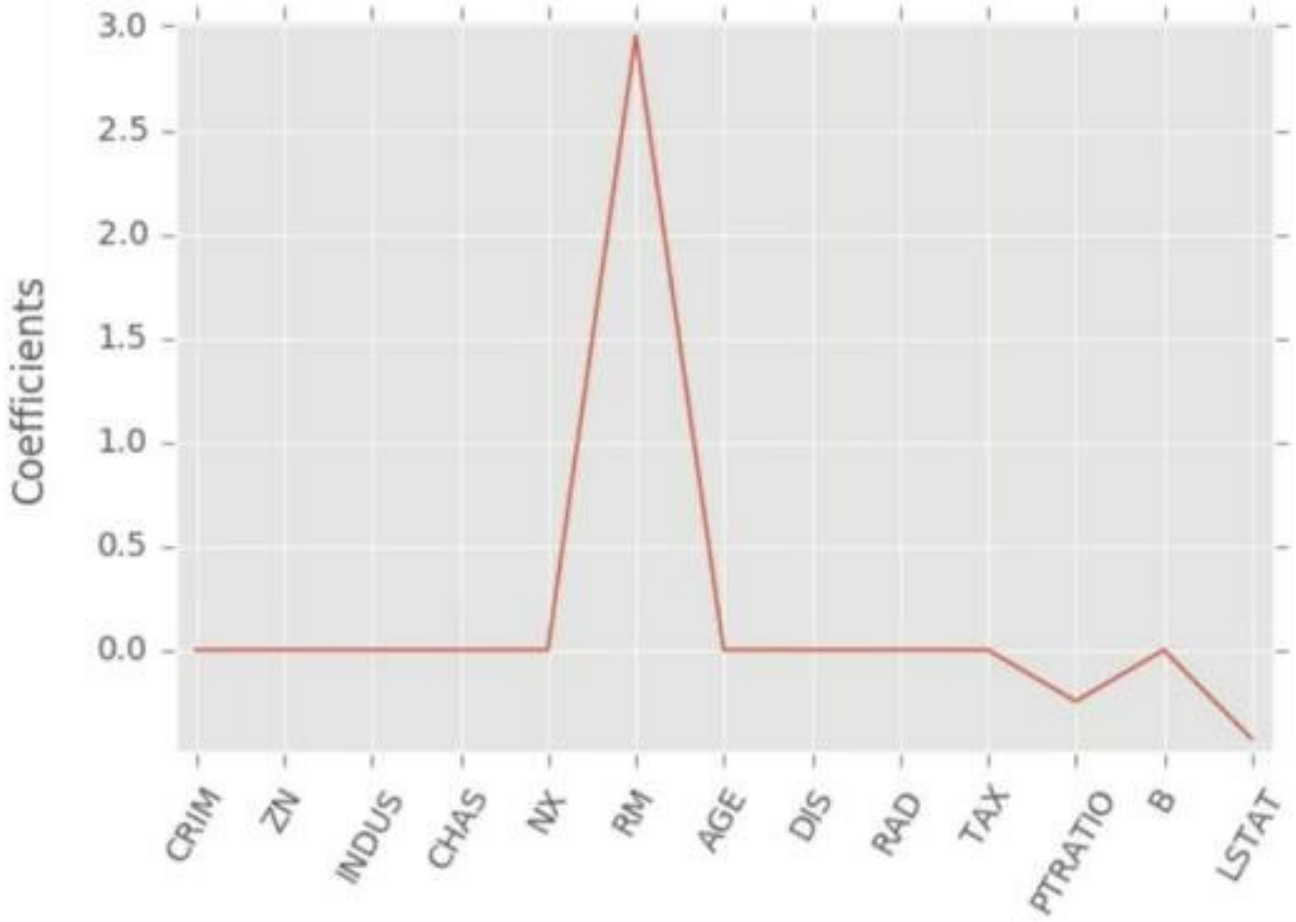


# Lasso for feature selection in scikit-learn

```
from sklearn.linear_model import Lasso

names = boston.drop('MEDV', axis=1).columns

lasso = Lasso(alpha=0.1)

lasso_coef = lasso.fit(X, y).coef_

_ = plt.plot(range(len(names)), lasso_coef)
_ = plt.xticks(range(len(names)), names, rotation=60)
_ = plt.ylabel('Coefficients')
plt.show()
```



# Lasso for feature selection in scikit-learn






Regularized Regression

Share with people and groups

Add people and groups

smit shah (you)
smitshah397@gmail.com — Owner

Anirban Chakraborty
chakraborty.anrbn@gmail.com — Editor

Decodr Technologies
getdecodr@gmail.com — Editor

Jagriti Kumari
jagriti@decodr.in — Editor

Send feedback to Goog

Link copied

Done

Get link

**Anyone** on the internet with this link can view

Change

Copy link

**Example:** The average commute time of employees of a company to and fro office is 35 minutes

**What is Hypothesis Testing?**

Page 1 / 21

Type here to search

6:28 PM
11/28/2020