Estadística III para Ingenieros de Sistemas

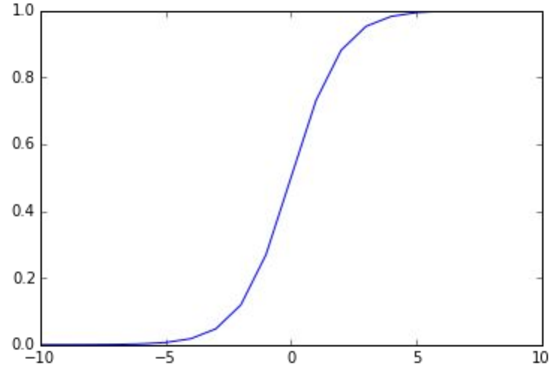
Jose Daniel Ramirez Soto 2023
jdr2162@columbia.edu

Supervisado, Regresión y regresión Logística

Regresión logística, encontrar los coeficientes W de una función, objetivo de reducir los errores en la clasificación.

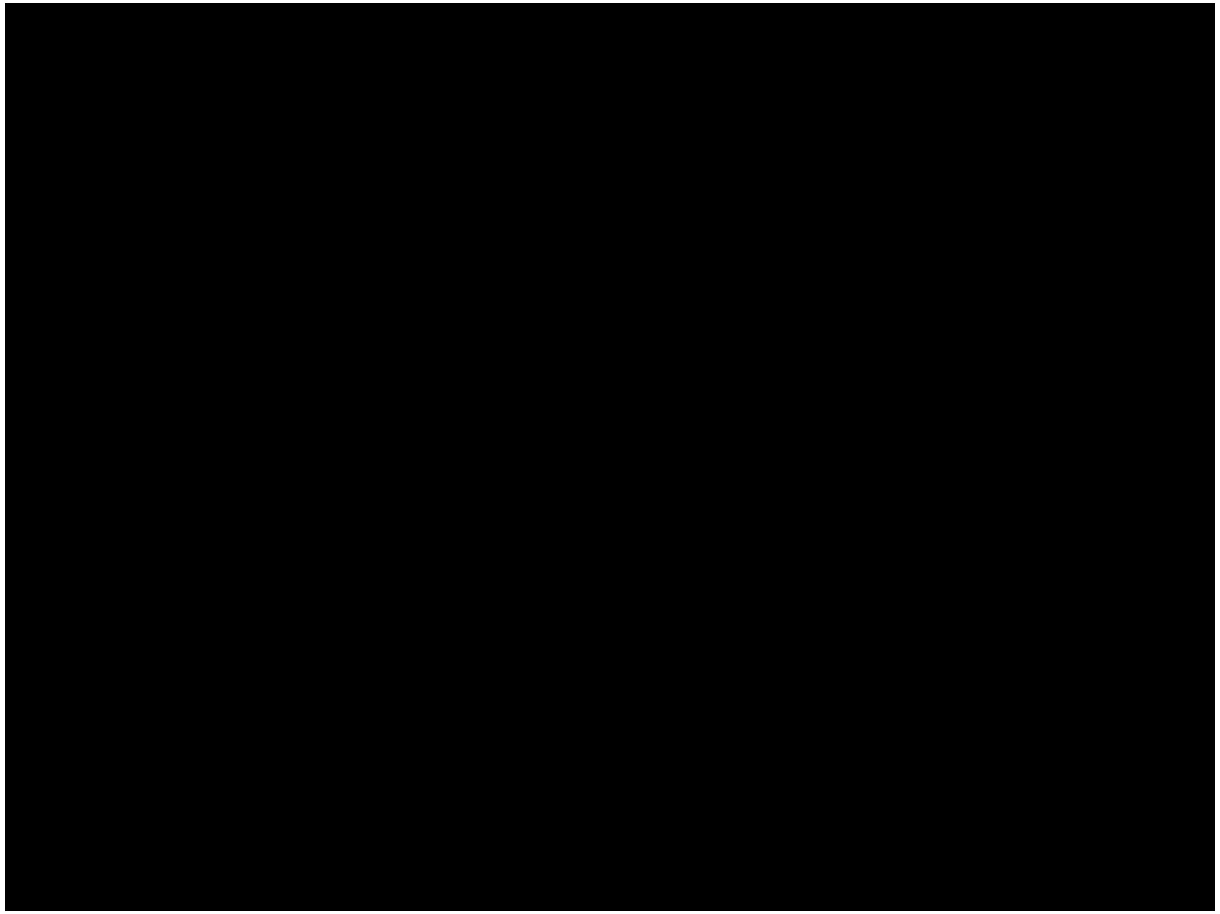
$$\tilde{y} = \frac{1}{1 + e^{-\sum_{i=0}^{p} w_i x_i} + b}$$

La función **Sigmoid**, tiene forma de “S” y valores entre [0,1]. Por defecto retorna 1 si $\tilde{y}>0.5$



* A Course of Machine Learning http://ciml.info/

Ejemplo: utilizando datos históricos del año anterior. Predecir si un alumno pasará el curso este año. ¿Cuál es el porcentaje de personas que el modelo predice que pasaron el curso?



Supervisado, Regresión y regresión Logística

Regresión logística, los resultados de la regresión logística son similares a los de la regresión. Sin embargo, en los problemas de clasificación se utilizan otras métricas para medir el error:

Labels \ Predicción	No Paso	Paso
No Paso	4 (True Negative)	1 (False Positive)10
Paso	1 (False Negative)	14 (True Positive)

accuracy = (TN + TP)/(TN+FP+FN+TP) = 18/20=0.9
precisión = TP/(TP+FP)=14/15 (cols)=0.933 (false positive)
recall = TP/(FN + TP)=14/15 (rows)=0.933 (false negative)

* A Course of Machine Learning http://ciml.info/

Estadística III para Ingenieros de Sistemas

Jose Daniel Ramirez Soto 2023
jdr2162@columbia.edu

Agenda

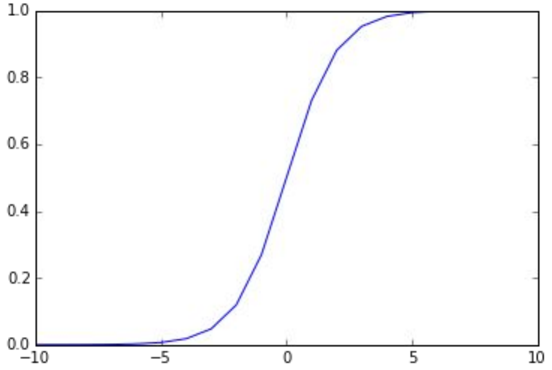
- **anuncios varios**
  - Parcial, revisión del parcial
  - Tarea se enviará este Sábado con fecha de entrega lunes 24 de Abril
- **modelos de analitica (machine learning-ML) Supervisado**
  - **Regresión logística**
    - **Matemática de la regresión logística**
      - **Gradiente descendiente**
    - **K-fold**
    - **Regularización**
    - **Feature selection o selección de features**
- **Práctica de regresión en Python**

Supervisado, Regresión y regresión Logística

Regresión logística, encontrar los coeficientes W de una función, objetivo de reducir los errores en la clasificación.

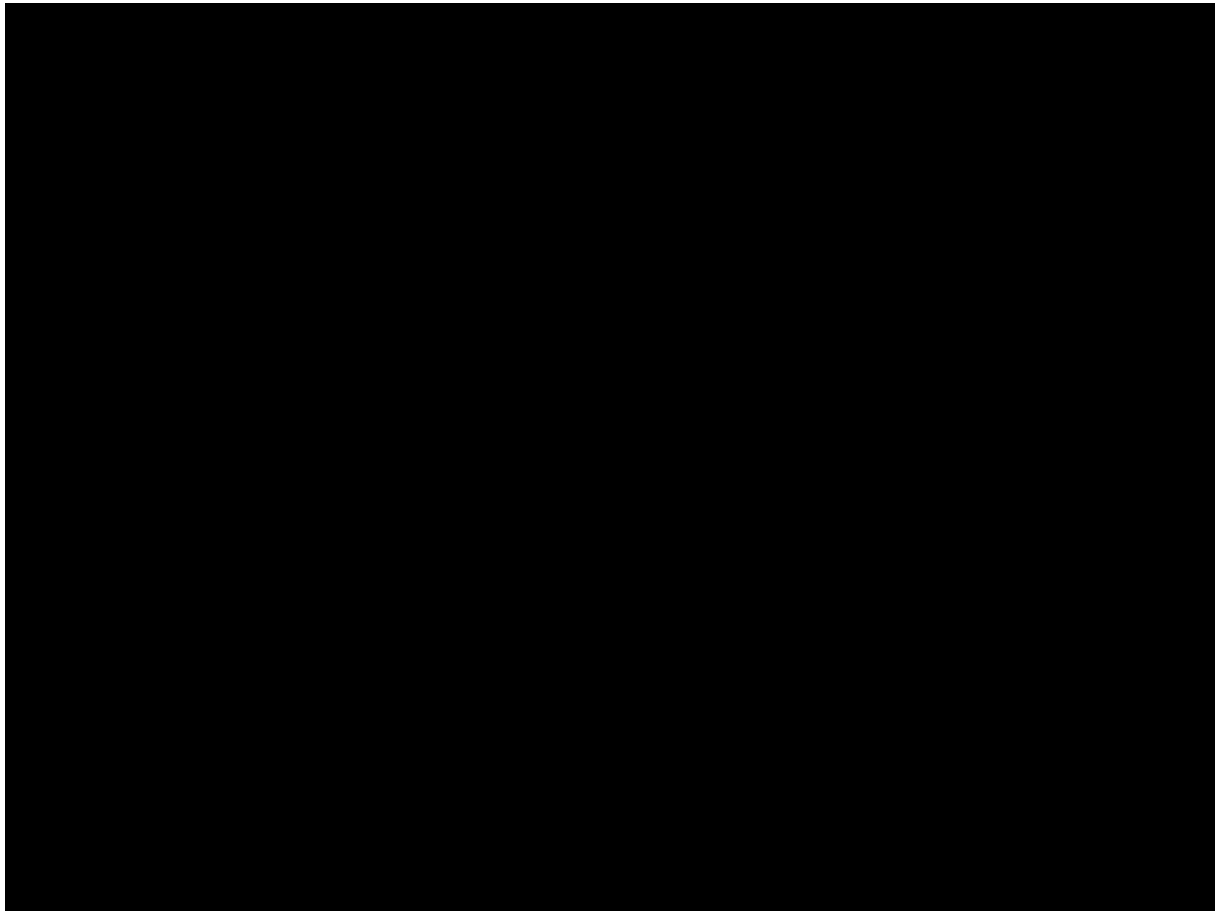
$$\tilde{y} = \frac{1}{1 + e^{-\sum_{i=0}^{p} w_i x_i} + b}$$

La función **Sigmoid**, tiene forma de "S" y valores entre [0,1]. Por defecto retorna 1 si $\tilde{y}>0.5$



* A Course of Machine Learning http://ciml.info/

Ejemplo: utilizando datos históricos del año anterior. Predecir si un alumno pasará el curso este año. ¿Cuál es el porcentaje de personas que el modelo predice que pasaron el curso?



Supervisado, Regresión y regresión Logística, Calcular B1

$$y = \frac{1}{(1+e^{-(b1x_1+b2x_2+b0)})}$$

Cada dato : $p(y = "\ c\ " \mid experimentos) = p(y_1 = "\ C\ ") * p(y_2 = "\ C\ ")$

$$p(y = 1|d) = 1/n \prod_{i=1}^{n} (1 - p(y))^{1-y}. (p(y))^{y}$$

$$l = \prod_{i=1}^{n} (1 - \frac{1}{(1+e^{-(b1x+b2x_2+b_0)})})^{(1-y)} * (\frac{1}{(1+e^{-(b1x+b2x_2+b_0)})})^{y}$$

$$\log(l) = \sum_{i=1}^{n} +(1-y)\log(1 - \frac{1}{(1+e^{-(b_1x+b_2x_2+b_0)})}) + y\log(\frac{1}{(1+e^{-(b_1x+b_2x_2+b_0)})})$$

* A Course of Machine Learning http://ciml.info/

Supervisado, Regresión y regresión Logística, Calcular B1

$$\frac{dl}{db_1} = \frac{dl}{du}\frac{du}{db_1}$$

$$\frac{dl}{db_1} = y\log([1+e^{-(b_1x+b_2x_2+b_0)}]^{-1}) + (1-y)\log(\frac{e^{-(b_1x+b_2x_2+b_0)}}{1+e^{-(b_1x+b_2x_2+b_0)}})$$

$$\frac{dl}{db_1} = -y\log(1+e^{-(b_1x+b_2x_2+b_0)}) + (1-y)[\log(e^{-(b_1x+b_2x_2+b_0)}) - \log(1+e^{-(b_1x+b_2x_2+b_0)})]$$

$$\frac{dl}{db_1} = \log(e^{-(b_1x+b_2x_2+b0)}) - \log(1+e^{-(b_1x+b_2x_2+b_0)}) - y\log(e^{-(b_1x+b_2x_2+b_0)})$$

$$\frac{dl}{db_1} = \frac{-(b_1x_1+b_2x_2+b0)}{db_1} - \frac{\log(1+e^{-(b_1x_1+b_2x_2+b_0)})}{db_1} - \frac{y(-(b_1x_1+b_2x_2+b_0))}{db_1}$$

* A Course of Machine Learning http://ciml.info/

Supervisado, Regresión y regresión Logística, Calcular B1

$$= -x_1 - (1 + e^{-(b_1x_1+b_2x_2+c)})^{-1}(e^{-(b_1x_1+b_2x_2+c)})(-x_1)) + yx_1$$

$$= -x_1 - (-x_1)(\frac{1}{1+e^{(b_1x_1+b_2x_2+c)}}) + yx_1$$

$$= -x_1 - (-x_1)(\frac{1}{1+e^{(b_1x_1+b_2x_2+c)}}) + yx_1$$

$$= \frac{-x_1+x_1+-x_1e^{(b_1x_1+b_2x_2+c)}}{1+e^{(b_1x_1+b_2x_2+c)}}) + yx_1$$

$$= x_1(y - \frac{e^{(b_1x_1+b_2x_2+c)}}{1+e^{(b_1x_1+b_2x_2+c)}})$$

$$\frac{dl}{db_1} = x_1(y - \frac{1}{1+e^{-(b_1x_1+b_2x_2+c)}})$$

* A Course of Machine Learning http://ciml.info/

Supervisado, Regresión y regresión Logística, Calcular B0

Calcular b_0. ¶

$$\frac{dl}{db0} = \log(e^{-(b_1x_1+b_2x_2+b_0)}) - \log(1+e^{-(b_1x_1+b_2x_2+b_0)}) - y\log(e^{-(b_1x+b_2x_2+b_0)})$$

$$\frac{dl}{d0} = \frac{-(b_1x_1+b_2x_2+b_0)}{dc} - \frac{\log(1+e^{-(b_1x_1+b_2x_2+b_0)})}{db0} - \frac{y(-(b_1x_1+b_2x_2+b_0))}{db0}$$

$$\frac{dl}{d0} = -1 - (-1)(\frac{1}{1+e^{(b_1x_1+b_2x_2+b_0)}}) + y$$

$$\frac{dl}{d0} = y - \frac{e^{(b_1x_1+b_2x_2+b_0)}}{1+e^{(b_1x_1+b_2x_2+b_0)}})$$

$$\frac{dl}{d0} = y - \frac{1}{1+e^{-(b_1x_1+b_2x_2+b_0)}}$$

* A Course of Machine Learning http://ciml.info/

Aplicar el gradiente descendiente

Algorithm 21 $\textsc{GradientDescent}(\mathcal{F}, K, \eta_1, \dots)$

$z^{(0)} \leftarrow \langle 0, 0, \dots, 0 \rangle$ // initialize variable we are optimizing

for $k = 1 \dots K$ **do**

  $g^{(k)} \leftarrow \nabla_z \mathcal{F}|_{z^{(k-1)}}$ // compute gradient at current location

  $z^{(k)} \leftarrow z^{(k-1)} - \eta^{(k)} g^{(k)}$ // take a step down the gradient

end for

return $z^{(K)}$

* A Course of Machine Learning http://ciml.info/
* Gareth James, An Introduction to Statistical Learning

Aplicar el gradiente descendiente

```
def train(self, x , y):
    # Copiamos las variables y hacemos la actualziacion despues de calcular
    b = self.b[:]
    c = self.c
    if np.array(x).ndim < 2 :
        b[0] = b[0] + self.lr * x[0]*(y - self.sigmoid(x))
        b[1] = b[1] + self.lr * x[1]*(y - self.sigmoid(x))
        if not self.is_norm:
            c = c + self.lr * (y - self.sigmoid(x))

    else:
        n_row = np.array(x).shape[0]
        b[0] = b[0] + self.lr *(sum([x[i][0]*(y[i] -self.sigmoid(x[i]))
                        for i in range(n_row)])/float(n_row))
        b[1] = b[1] + self.lr *(sum([x[i][1]*(y[i] -self.sigmoid(x[i]))
                        for i in range(n_row)])/float(n_row))
        if not self.is_norm:
            c = c + self.lr *(sum([y[i] -self.sigmoid(x[i])
                            for i in range(n_row)])/float(n_row))
    self.b = b
    self.c = c
```

* A Course of Machine Learning http://ciml.info/
* Gareth James, An Introduction to Statistical Learning

K-fold

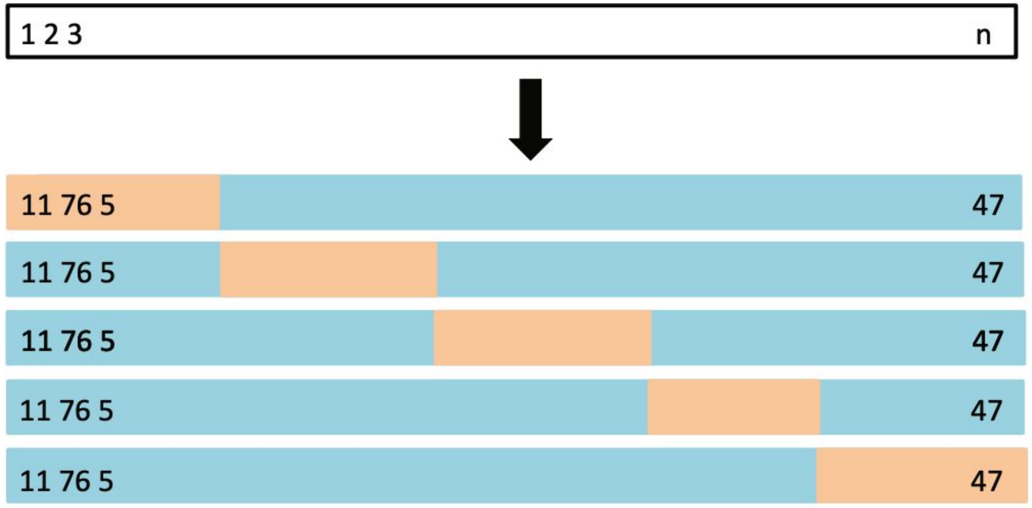


FIGURE 5.5. *A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.*

or linear discriminant analysis, or any of the methods discussed in later chapters. The magic formula (5.2) does not hold in general, in which case the model has to be refit n times.

* A Course of Machine Learning http://ciml.info/
* Gareth James, An Introduction to Statistical Learning

Regularización

La suma de los cuadrados de los coeficientes Ridge y La suma de los valores absolutos Lasso

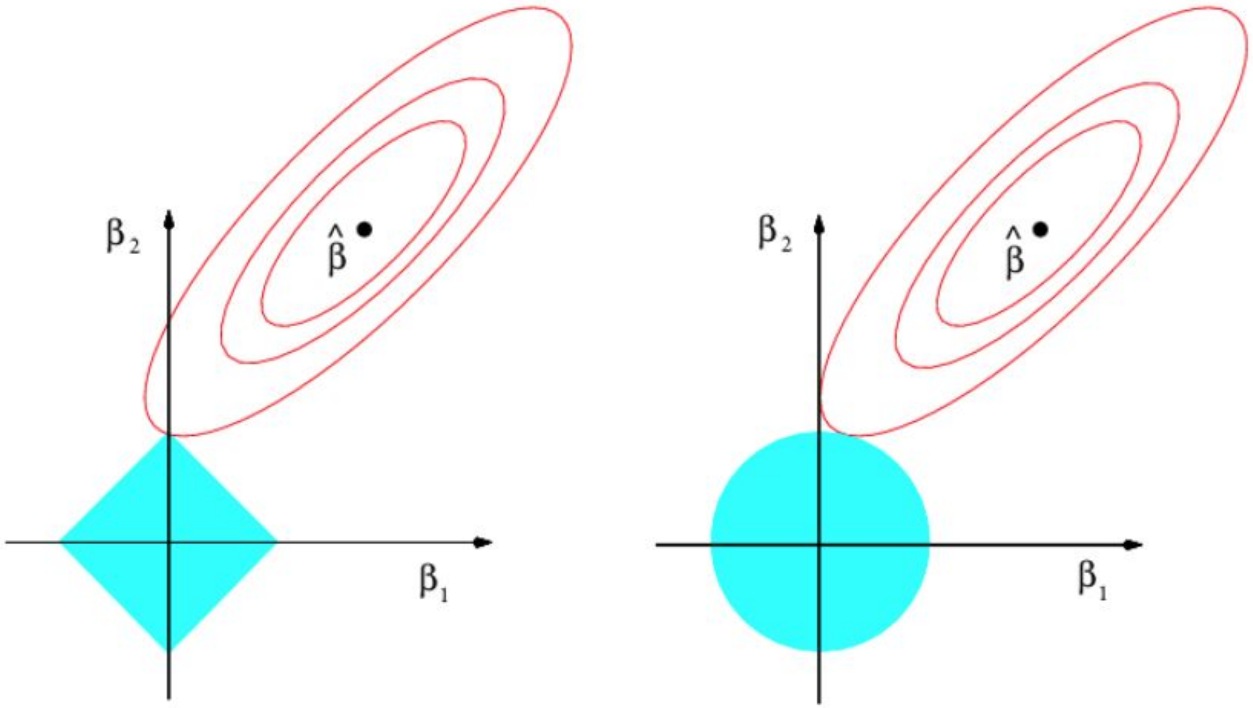


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$*, while the red ellipses are the contours of the RSS.*

Con ecuaciones simples

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s \tag{6.8}$$

and

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s, \tag{6.9}$$

Forma matricial

$$\mathcal{L}(\boldsymbol{w}, b) = \sum_n \exp\left[-y_n(\boldsymbol{w} \cdot \boldsymbol{x}_n + b) \right] + \frac{\lambda}{2} ||\boldsymbol{w}||^2$$

* A Course of Machine Learning http://ciml.info/

Feature Selection

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p - k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p - k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

* A Course of Machine Learning http://ciml.info/
* Gareth James, An Introduction to Statistical Learning

Feature Selection

Algorithm 6.3 *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

* A Course of Machine Learning http://ciml.info/
* Gareth James, An Introduction to Statistical Learning