Lecture 7: Machine Learning
Overfit & Cross Validation

Big Data and Machine Learning en el Mercado Inmobiliario
Educación Continua

Ignacio Sarmiento-Barbieri

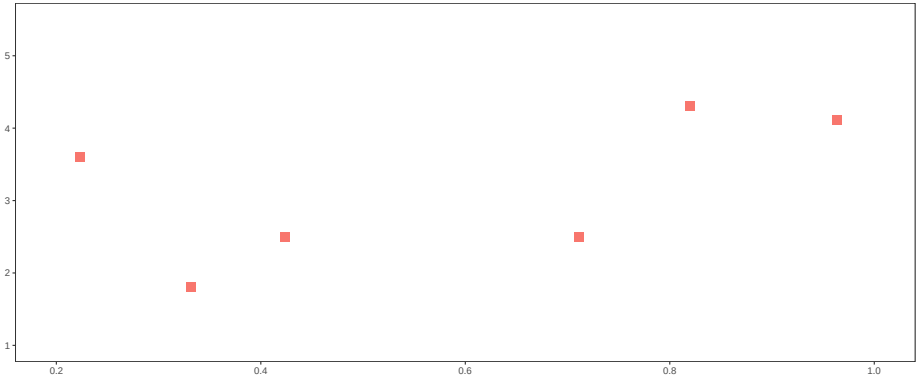
Universidad de los Andes

March 29, 2022

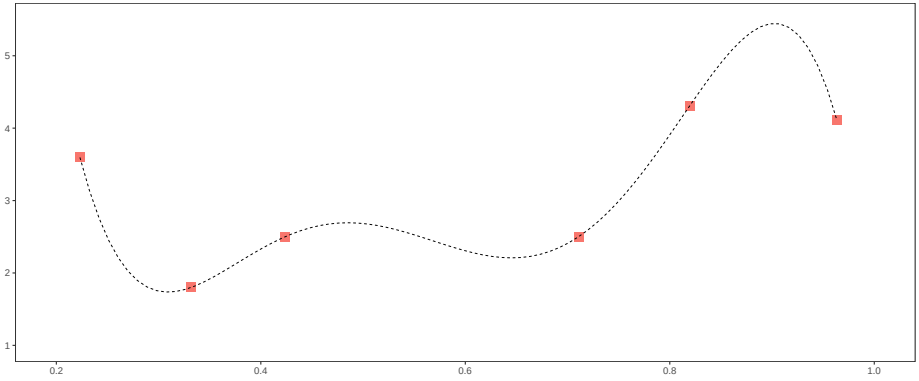
Agenda

1. Recap
   - Overfit y Predicción fuera de Muestra
2. Error de predicción y métodos de remuestreo
   - Enfoque de conjunto de validación
   - LOOCV
   - Validación cruzada en K-partes
3. Para seguir leyendo
4. Break

Overfit



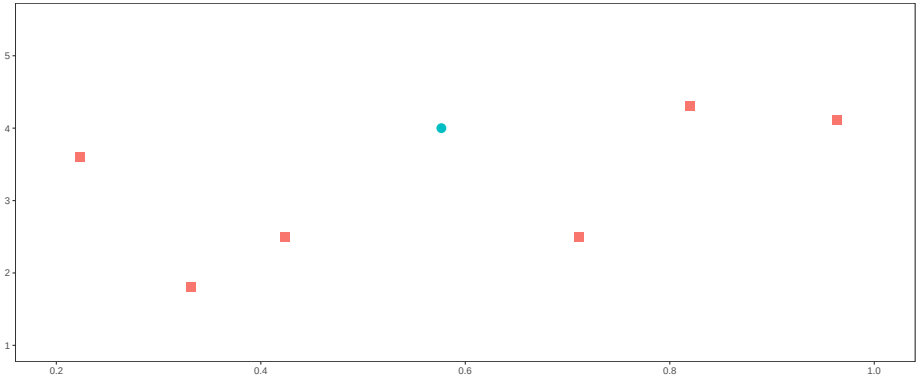
Overfit



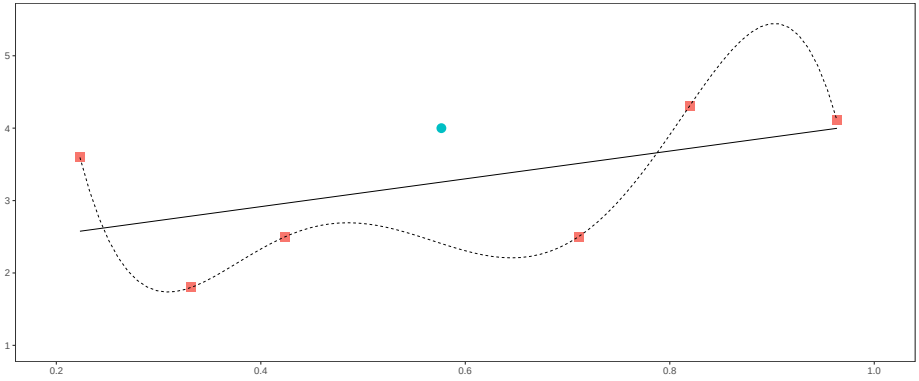
Overfit y Predicción fuera de Muestra

- ML nos interesa la predicción fuera de muestra
- Overfit: modelos complejos predicen muy bien dentro de muestra, pero tienden a hacer un trabajo fuera de muestra
- Hay que elegir el nivel adecuado de complejidad

Overfit



Overfit



Métodos de remuestreo

- Los métodos de resampleo son una herramienta indispensable de la estadística moderna.
- Estos envuelven sacar muestras aleatorias de nuestra muestra y reajustar el modelo de interés en cada muestra para obtener información adicional del modelo.
- Quizás el método más conocido por ustedes es el de bootstrap.
- Nosotros vamos a discutir la validación cruzada (cross-validation)

Error de Prueba y de Entrenamiento

- Dos conceptos importantes
  - *Error de Prueba*: es el error de predicción en la muestra de prueba (test)

$$Err_{\mathcal{T}est} = MSE[(y, \hat{y})|\mathcal{T}est] \tag{1}$$

  - *Error de Entrenamiento*:es el error de predicción en la muestra de entrenamiento (training)

$$Err_{\mathcal{T}rain} = MSE[(y, \hat{y})|\mathcal{T}rain] \tag{2}$$

- Cómo elegimos $\mathcal{T}est$?

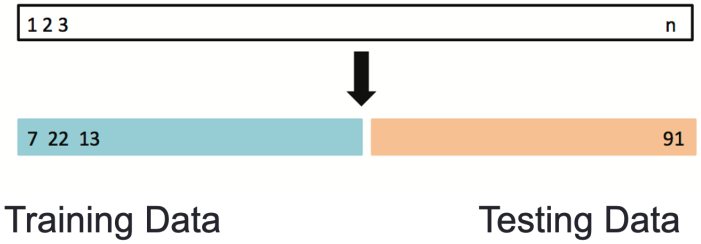
Qué son los Métodos de Remuestreo?

- Herramientas que implican extraer repetidamente muestras de un conjunto de entrenamiento y reajustar el modelo de interés en cada muestra para obtener más información sobre el modelo.
- Evaluación del modelo: estimar el error de predicción en la muestra de prueba
- Selección de modelo: seleccione el nivel apropiado de flexibilidad del modelo
- ¡Son computacionalmente costosos! Pero en estos días tenemos computadoras poderosas

Enfoque de conjunto de validación

- Suponga que nos gustaría encontrar un conjunto de variables que den el menor error de predicción en la muestra de prueba (no de entrenamiento)
- Si tenemos muchos datos, podemos lograr este objetivo dividiendo aleatoriamente los datos en partes de entrenamiento y validación (prueba)
- Luego usaríamos la parte de entrenamiento para construir cada modelo posible (es decir, las diferentes combinaciones de variables) y elegimos el modelo que dio lel menor error de predicción en la muestra de prueba

Enfoque de conjunto de validación



Enfoque de conjunto de validación

Ejemplo

- Los datos vienen de Properati

```
#cargar librerias
import pandas as pd
import numpy as np
```

```
df = pd.read_csv('co_properties.csv.gz', compression='gzip'
, header=0,    sep=',', quotechar='"')
```

	id	ad_type	start_date	end_date	created_on	\
0	KsjahK62rxcYKXXQjOdkqw==	Propiedad	2020-10-07	2021-10-09	2020-10-07	
1	Y+gsBZYq1zu5NoR3V5oUGA==	Propiedad	2020-10-07	2021-01-06	2020-10-07	
2	Jpzqxj8/Vgf3Aa5ASxUBNg==	Propiedad	2020-10-07	2020-10-07	2020-10-07	
3	ieuFnkFx/yHDD66iMV14Gw==	Propiedad	2020-10-07	2021-04-12	2020-10-07	
4	g4u5JM+hAHEk8SukRSjMzg==	Propiedad	2020-10-07	9999-12-31	2020-10-07	

Enfoque de conjunto de validación

Ejemplo

- Los datos vienen de Properati

	lat	lon	l1	l2	l3	...	bathrooms	\
0	3.9210	-76.506000	Colombia	Valle del Cauca	NaN	...	7.0	
1	3.3577	-76.541811	Colombia	Valle del Cauca	Cali	...	7.0	
2	3.3577	-76.541811	Colombia	Valle del Cauca	Cali	...	7.0	
3	3.3640	-76.538000	Colombia	Valle del Cauca	Cali	...	8.0	
4	3.3910	-76.517000	Colombia	Valle del Cauca	Cali	...	9.0	

Enfoque de conjunto de validación

Ejemplo

- Los datos vienen de Properati

	surface_total	surface_covered	price	currency	price_period	\
0	NaN	NaN	1.300000e+09	COP	NaN	
1	NaN	NaN	2.800000e+09	COP	NaN	
2	NaN	NaN	2.800000e+09	COP	Mensual	
3	NaN	NaN	3.500000e+09	COP	NaN	
4	NaN	NaN	4.800000e+08	COP	NaN	

Enfoque de conjunto de validación

Ejemplo

- Los datos vienen de Properati

```
                                          title  \
0  Casa Campestre en venta en darien 3469064
1                       Casa en ciudsd jardin
2                       Casa en ciudsd jardin
3              Casa en venta en pance 1630426
4             CASA EXTERNA BARRIO CIUDAD 2000
```

Enfoque de conjunto de validación

Ejemplo

- Los datos vienen de Properati

```
                                                description  property_type  \
0  HERMOSA CASA CAMPESTRE, &Aacute;REA 6,000 MT, ...           Casa
1  Casa independiente con posiciona en ciudad jar...           Casa
2  Casa independiente con posiciona en ciudad jar...           Casa
3  EXCELENTE CASA - LOTE 6,373 MT, EN OBRA GRIS U...           Casa
4  CASA EXTERNA EN EL BARRIO CIUDAD 2000,CONSTRUI...           Casa

  operation_type
0          Venta
1          Venta
2          Venta
3          Venta
4          Venta
```

Enfoque de conjunto de validación

- Filtramos y nos quedamos con casas en venta en Bogotá, que no tienen faltantes en las siguientes variables

	price	rooms	bedrooms	bathrooms	surface_total
873	1.750000e+08	3.0	3.0	2.0	63.0
1311	8.000000e+08	5.0	5.0	3.0	175.0
1557	1.800000e+09	3.0	3.0	4.0	550.0
8163	3.490000e+08	4.0	4.0	3.0	90.0
8171	9.500000e+08	4.0	4.0	4.0	322.0

	surface_covered	lat	lon
873	63.0	4.753	-74.112
1311	253.0	4.703	-74.057
1557	369.0	4.724	-74.024
8163	180.0	4.745	-74.064
8171	279.0	4.702	-74.060

Enfoque de conjunto de validación

```
from sklearn.model_selection import train_test_split

train, test = train_test_split(df, test_size=0.2,random_state=123)
```

```
y_train= train['price']
X_train=train.drop(columns=['price'])

y_test= test['price']
X_test=test.drop(columns=['price'])
```

Enfoque de conjunto de validación

```
from sklearn.linear_model import LinearRegression

model1 = LinearRegression().fit(X_train[['rooms']],y_train)
```

Enfoque de conjunto de validación

```
y_pred = model1.predict(X_test[['rooms']])
```

```
y_pred[0:10]
```

```
array([1.32764284e+09, 1.31508025e+09, 1.31298649e+09, 1.31926778e+09,
       1.31298649e+09, 1.32764284e+09, 1.31717402e+09, 1.31508025e+09,
       1.31089272e+09, 1.32136155e+09])
```

Enfoque de conjunto de validación

```
MSE = np.square(np.subtract(y_test,y_pred)).mean()
MSE
```

```
2.6487563763772257e+18
```

```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,y_pred)
```

```
2.648756376377227e+18
```

Enfoque de conjunto de validación

```
model2 = LinearRegression().fit(X_train[['rooms','surface_total']],y_train)
y_pred2 = model2.predict(X_test[['rooms','surface_total']])
```

```
model3 = LinearRegression().fit(X_train[['surface_total','rooms','bedrooms',
'bathrooms']],y_train)
y_pred3 = model3.predict(X_test[['surface_total','rooms','bedrooms',
'bathrooms']])
```

Enfoque de conjunto de validación

```
mean_squared_error(y_test,y_pred)
```

```
2.648756376377227e+18
```

```
mean_squared_error(y_test,y_pred2)
```

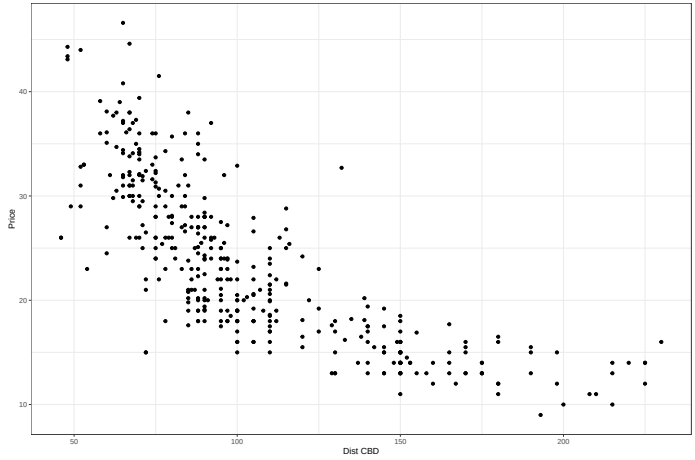
```
2.520184992914835e+18
```

```
mean_squared_error(y_test,y_pred3)
```

```
2.232920478011764e+18
```

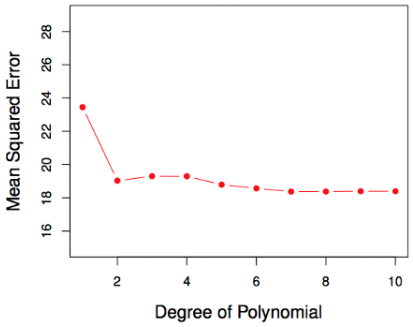
Ejemplo 2

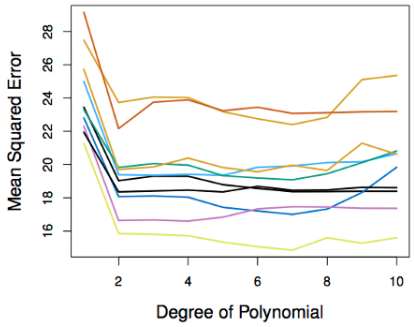
- Modelo $y = f(x) + u$ donde f es un polinomio de grado p^*.



Enfoque de conjunto de validación

- Modelo $y = f(x) + u$ donde f es un polinomio de grado p^*.





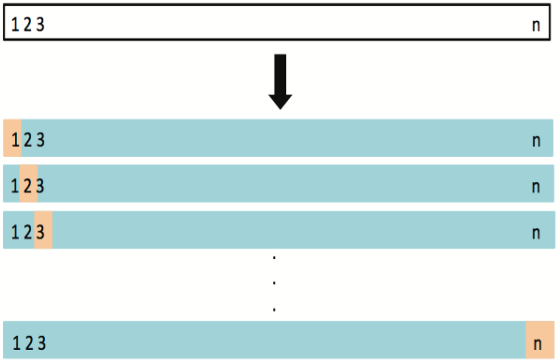
- Izquierda: error de predicción en la muestra de prueba para una sola partición
- Derecha: error de predicción en la muestra de prueba para varias particiones
- Hay un montón de variabilidad. (Necesitamos algo mas estable)

Enfoque de conjunto de validación

- Ventajas:
  - Simple
  - Fácil de implementar
- Desventajas:
  - El MSE de validación (prueba) puede ser altamente variable
  - Solo se utiliza un subconjunto de observaciones para ajustar el modelo (datos de entrenamiento). Los métodos estadísticos tienden a funcionar peor cuando se entrenan con pocas observaciones

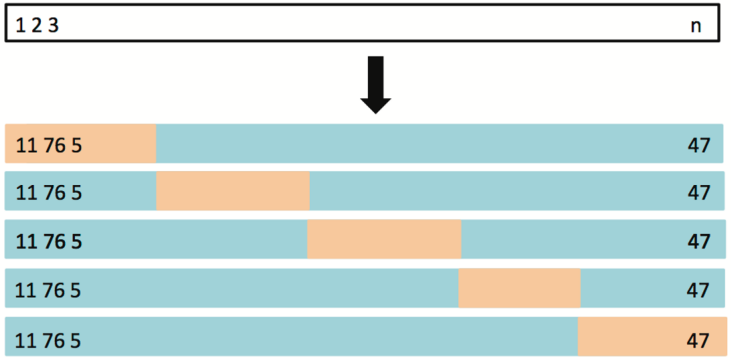
Leave-One-Out Cross Validation (LOOCV)

- Este método es similar al enfoque de validación, pero trata de abordar las desventajas de este último.



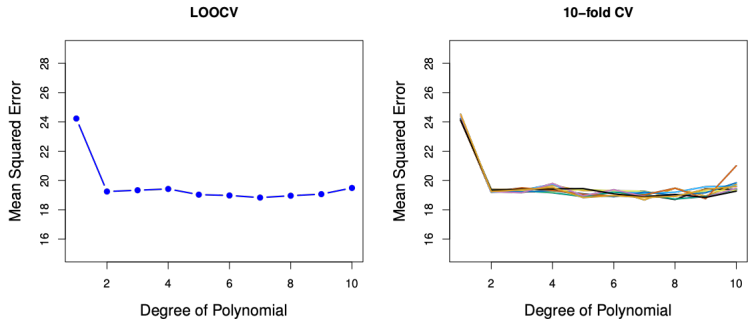
Validación cruzada en K-partes

- LOOCV es computacionalmente intensivo, por lo que podemos ejecutar k-fold Cross Validation



Validación cruzada en K-partes

- Izquierda: LOOCV error
- Derecha: 10-fold CV
- LOOCV es caso especial de k-fold, donde k = n
- Ambos son estables, pero LOOCV (generalmente) es mas intensivo computacionalmente!



Trade-off Sesgo-Varianza para validación cruzada en K-partes

- Sesgo:
  - El enfoque del conjunto de validación tiende a sobreestimar el error de predicción en la muestra de prueba (menos datos, peor ajuste)
  - LOOCV, agrega más datos $\rightarrow$ menos sesgo
  - K-fold un estado intermedio
- Varianza:
  - LOOCV promediamos los resultados de n modelos ajustados, cada uno está entrenado en un conjunto casi idéntico de observaciones $\rightarrow$ altamente correlacionado
  - K partes esta correlación es menor, estamos promediando la salida de k modelo ajustado que están algo menos correlacionados
- Por lo tanto, existe un trade-off
  - Tendemos a usar k-fold CV con (K = 5 y K = 10)
  - Se ha demostrado empíricamente que producen estimaciones del error de prediccion que no sufren ni de un sesgo excesivamente alto ni de una varianza muy alta Kohavi (1995)

K-Fold con datos geográficos

Para seguir leyendo

- Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, No. 2, pp. 1137-1145).
- Lovelace, R., Nowosad, J., & Muenchow, J. (2019). Geocomputation with R. CRC Press. (Chapters 2 & 6)

Volvemos en 5 min con Python