Regresión Logística

Delta Analytics construye capacidad técnica alrededor del mundo.

El contenido de este curso está siendo desarrollado activamente por Delta Analytics, una organización sin fines de lucro 501(c)3 del Área de la Bahía que apunta a capacitar a las comunidades para aprovechar sus datos.

Por favor comuníquese con cualquier pregunta o comentario a inquiry@deltanalytics.org.

Descubre más sobre nuestra misión aquí.

Rápida visión general

Regresión & Clasificación

- ML estudia cómo **aprender automáticamente** para hacer predicciones acertadas basados en **observaciones pasadas.**

- Dos tipos de tareas supervisadas, regresión y clasificación.

Regresión
Variable continua

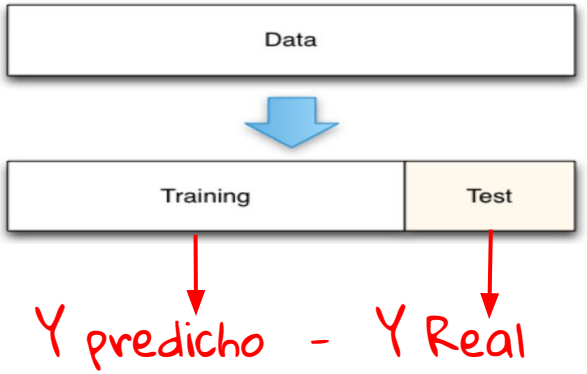
Clasificación
Variable categórica

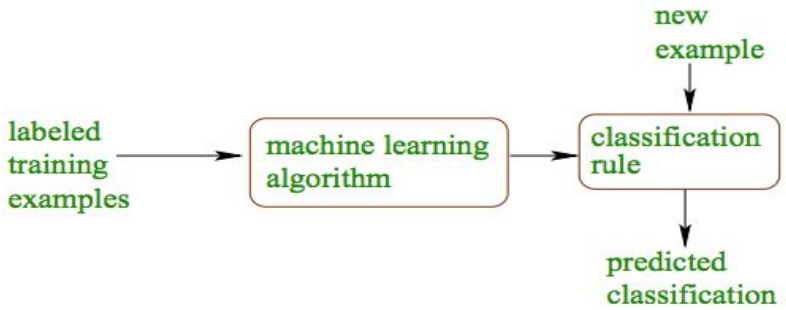
Regresión de mínimos cuadrados (OLS)

Regresión logística

Desempeño del modelo y evaluación

- Habilidad de generalizar a datos no observados:





Fuente: Machine Learning Algorithms for Classification, Schapire (2016)

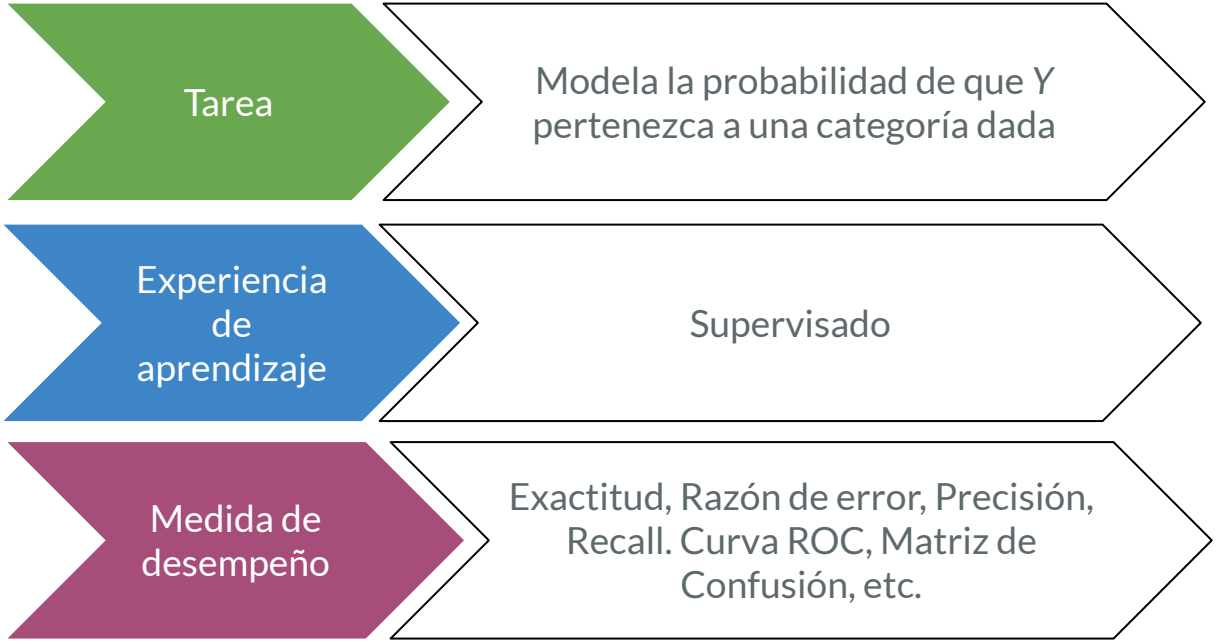
- Pasos generales:
  - Divide datos en sets de “entrenamiento” y “test”.
  - Usa resultados de regresión/clasificación del set de “entrenamiento” para predecir el set de“test”.
  - Compara “Y predicho” con “Y real”

- Métricas de validación (OLS):

** R^2 ** Adjusted R2 ** MSE

Module 3.0:
Regresión Logística

Visión general de regresión logística:



Fuente: Deep Learning Book - Chapter 5: Introduction to Machine Learning

Tarea

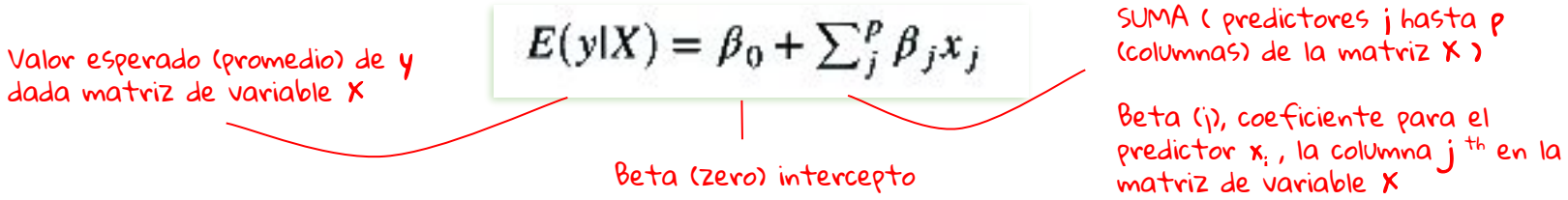
Checklist del módulo

- ❏ Regresión Logística
  - ❏ Tarea
    - ❏ Dilema usando OLS
    - ❏ Odds ratio
    - ❏ Función link
    - ❏ Umbrales de probabilidad
  - ❏ Experiencia de aprendizaje
    - ❏ Función de costo
    - ❏ Proceso de optimización
  - ❏ Desempeño
    - ❏ Matriz de Confusión
    - ❏ ROC y AUC

Tarea

Dilema al usar OLS

- Una regresión lineal con matriz de variables ***X*** variable objetivo ***y*** es formulado como:

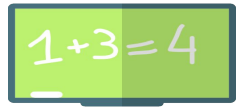


- Con regresión lineal, es difícil asignar un valor observado x a una categoría y.
- Ejemplo:
  - Predice admisiones a Colleges usando GRE, GPA, y prestigio del college
    - Cuál sería el valor de la categoría de salida “College Admissions”?

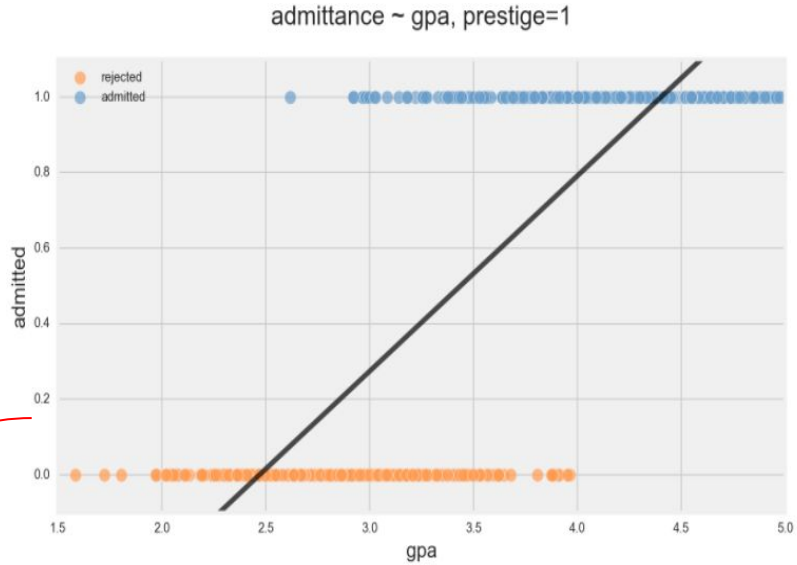
Tarea

Dilema al usar OLS

Prediciendo admisión de college con gpa, gre y prestigio de college



	admit	gre	gpa	prestige
0	0	380.0	3.61	3.0
1	1	660.0	3.67	3.0
2	1	800.0	4.00	1.0

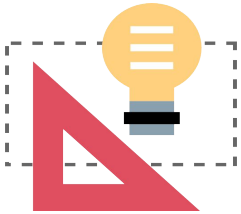


Houston tenemos un problema!!

Tarea

Dilema al usar OLS

Formulando la idea en términos de clasificación



- Tenemos un problema de clasificación “binaria” básico
  - *1 = admitido y 0 = rechazado*

- Ten en cuenta que regresión logística igual obtiene un valor estimado. En clasificación binaria este valor esperado es la probabilidad de una clase:

$$E[y \in 0, 1] = P(y = 1)$$

- En lenguaje de regresión tendríamos:

Preguntas?

$$P(y = 1) = \beta_0 + \sum_j^p \beta_j x_j$$

Estima la probabilidad en lugar del número real!!!

- Hay un problema con esta ecuación: Queremos estimar la probabilidad en lugar de un número real.
  - *Necesitamos que y esté en el rango [-infinity, infinity] para que la regresión sea válida!*

$$P(y = 1) = \beta_0 + \sum_j^p \beta_j x_j$$

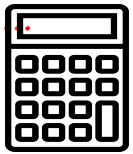
y en el rango ⊏ -infinity, infinity ⊐

- Es aquí donde la “función link” viene a nuestro rescate!!

Tarea

Odds ratio

Probabilidad de membresía a clase.



- Regresión logística es una variación de regresión lineal con variables objetivo categóricas, donde en lugar resolver para el promedio de ***y***, regresión logística resuelve a ***la probabilidad de pertenencia a clase y***.
- Cómo hace esto? Usa una **función link** para describir una función lineal entre la probabilidad y la variable independiente

La función link es una función del valor esperado de la variable objetivo

$$logit(E(y|X)) = \beta_0 + \sum_j^p \beta_j x_j$$

Tarea

Odds ratio

Modifica la ecuación de regresión

- Cuál es nuestra función link en el caso de regresión logística?
- Nuestra función link usará algo llamado **odds ratio**

El odds ratio de una probabilidad p es una medida de cuánto más probable es que el caso negativo

$$\text{odds ratio}(p) = \frac{p}{1-p}$$

→

- When `p = 0.5`: `odds ratio = 1`
  - it is equally likely to happen as it is to not happen.
- When `p = 0.75`: `odds ratio = 3`
  - it is 3 times more likely to happen than not happen.
- When `p = 0.40`: `odds ratio = 0.666..`
  - it is 2/3rds as likely to happen than not happen.

Preguntas?

Tarea

Odds ratio

En nuestro ejemplo...

Prediciendo admisión al college

	admit	gre	gpa	prestige
0	0	380.0	3.61	3.0
1	1	660.0	3.67	3.0
2	1	800.0	4.00	1.0

Probabilidades de admisión por prestigio de colleges

```
admissions.prestige.unique()
```

```
array([ 3.,  1.,  4.,  2.])
```

```
y_p1 = admissions[admissions.prestige == 1].admit.values
y_p2 = admissions[admissions.prestige == 2].admit.values
y_p3 = admissions[admissions.prestige == 3].admit.values
y_p4 = admissions[admissions.prestige == 4].admit.values
```

```
print 'P(admit | prestige = 1):', np.mean(y_p1)
print 'P(admit | prestige = 2):', np.mean(y_p2)
print 'P(admit | prestige = 3):', np.mean(y_p3)
print 'P(admit | prestige = 4):', np.mean(y_p4)
```

```
P(admit | prestige = 1): 0.540983606557
P(admit | prestige = 2): 0.358108108108
P(admit | prestige = 3): 0.231404958678
P(admit | prestige = 4): 0.179104477612
```

Odds ratios de admisión por prestigio de college

```
def odds_ratio(p):
    return (float(p) / (1 - p))
```

```
print 'odds(admit | prestige = 1):', odds_ratio(np.mean(y_p1))
print 'odds(admit | prestige = 2):', odds_ratio(np.mean(y_p2))
print 'odds(admit | prestige = 3):', odds_ratio(np.mean(y_p3))
print 'odds(admit | prestige = 4):', odds_ratio(np.mean(y_p4))
```

```
odds(admit | prestige = 1): 1.17857142857
odds(admit | prestige = 2): 0.557894736842
odds(admit | prestige = 3): 0.301075268817
odds(admit | prestige = 4): 0.218181818182
```

Tarea

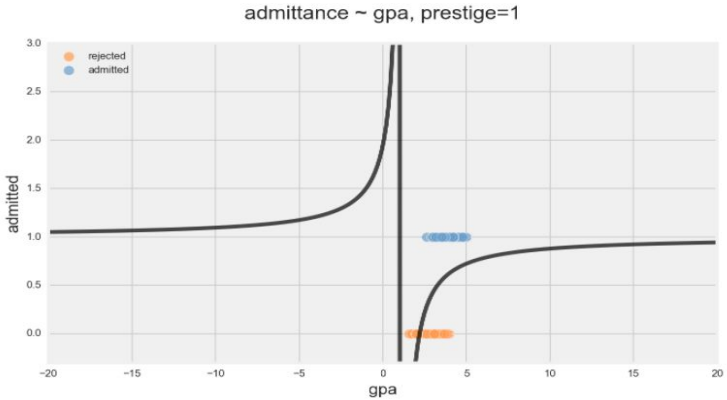
Odds ratio

Modifica la ecuación de regresión

- Si ponemos odds ratio en lugar de probabilidad en la ecuación, **el rango de *odds ratio***, nuestro valor predicho, está ahora en **[0, infinito]**

$$\frac{P(y=1)}{1-P(y=1)} = \beta_0 + \sum_j^p \beta_j x_j$$

- Y gráficamente se ve así:

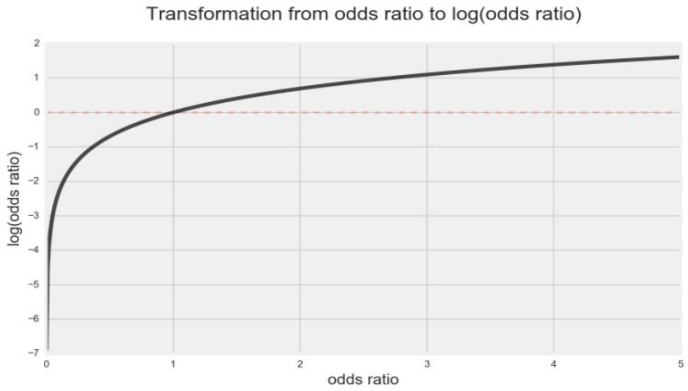


Tarea

Logit link function

Modifica la ecuación de regresión

- Si tomamos el logaritmo natural de una variable con rango de 0 a infinito, obtenemos una variable en el rango de menos infinito a más infinito
  - Por qué? Porque tomar el logaritmo de números menores a uno dá valores negativos.
- Y ahora nuestro gráfico se ve así:



... así es como debe lucir..

Tarea

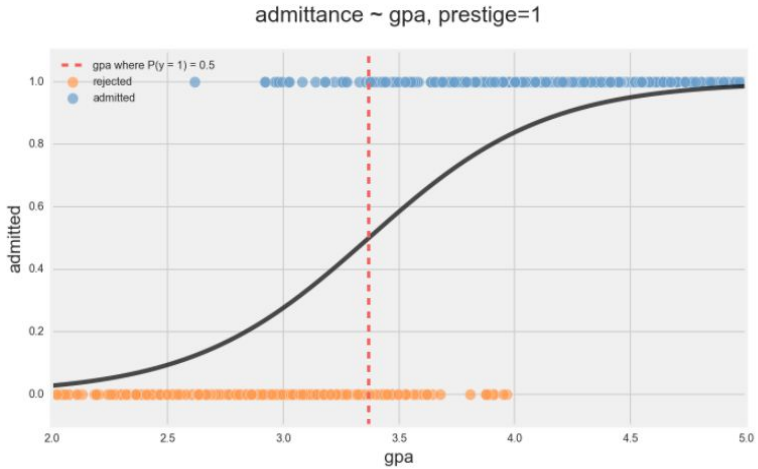
Logit link function

Modifica la ecuación de regresión

- La combinación de pasar de probabilidad a odds ratio y luego tomar el logaritmo se llama **función logit link**, y es lo que regresión usa para estimar probabilidades:

$$\text{logit}\Big(E[y]\Big) = \text{logit}\Big(P(y=1)\Big) = log\Big(\frac{P(y=1)}{1-P(y=1)}\Big) = \beta_0 + \sum_j^p \beta_j x_j$$

- Gráficamente se ve así:



Houston resolvimos el problema!

- Ahora que tenemos una probabilidad, cómo clasificamos los datos?
- Escoge una probabilidad dependiendo en la clasificación que intentas resolver:

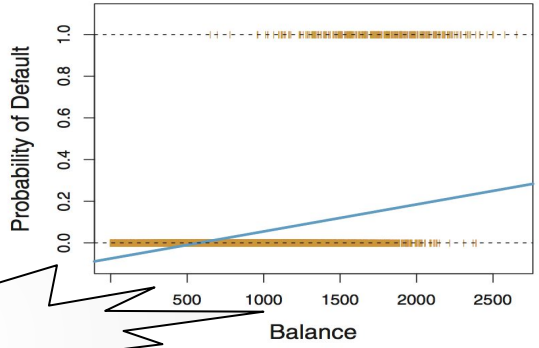
$$y = \begin{cases} 0 & if\ p < 0.5 \\ 1 & if\ p \geq 0.5 \end{cases}$$

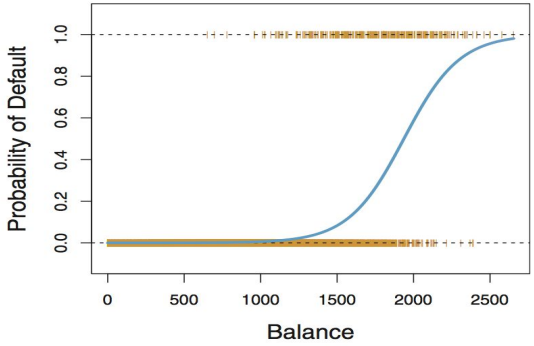
En este caso, 0.5 es el umbral de probabilidad. El umbral puede ser ajustado por el modelo.

Tarea

Revisemos nuestro entendimiento de regresión logística

- He aquí un ejemplo de clasificación, donde el balance de la cuenta es usado para predecir la probabilidad de desfalco.
- Puedes establecer cuál es el método de clasificación correcto, regresión (izquierda) y regresión logística (derecha)?





Preguntas?

Metodología de aprendizaje

- Regresión logística, como OLS, es resulta al minimizar una función de pérdida, también llamada **función de costo**
- La función de costo para OLS era la RSS (residual sum of squares), pero la función de costo para regresión logística es:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$
$$= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

- Veamos el detalle. Queremos minimizar la función de costo, J

1. Para cada clasificador que hemos predicho...

2. Suma el "costo" de la predicción, donde h(x) es la predicción e y es la clase real

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

- La función de costo debiese ser mayor cuando nuestras predicciones son erróneas y menor cuando aciertan

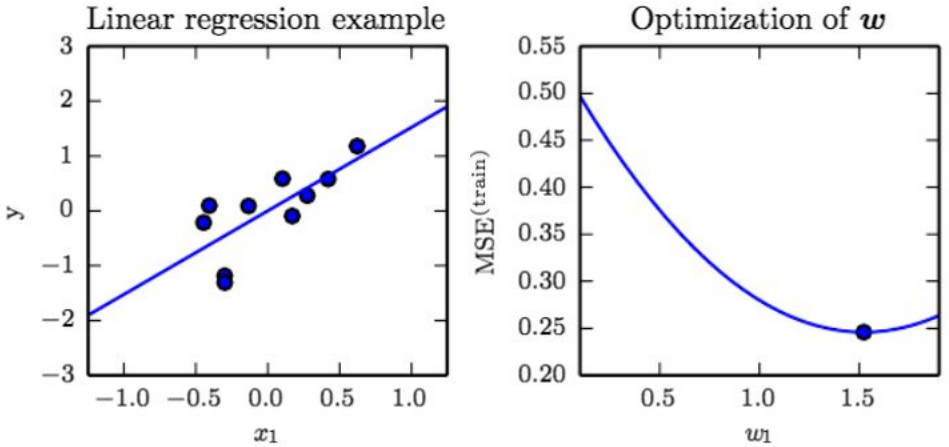
$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$
$$= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

- La función de costo satisface nuestra necesidad! Cuando h(x) = 1 e y(0) , la función de costo es infinita

- Como OLS, regresión logística **aprende por gradient descent** para minimizar la función de costo
- Recordatorio de gradient descent de OLS

Preguntas?

Optimización de MSE:

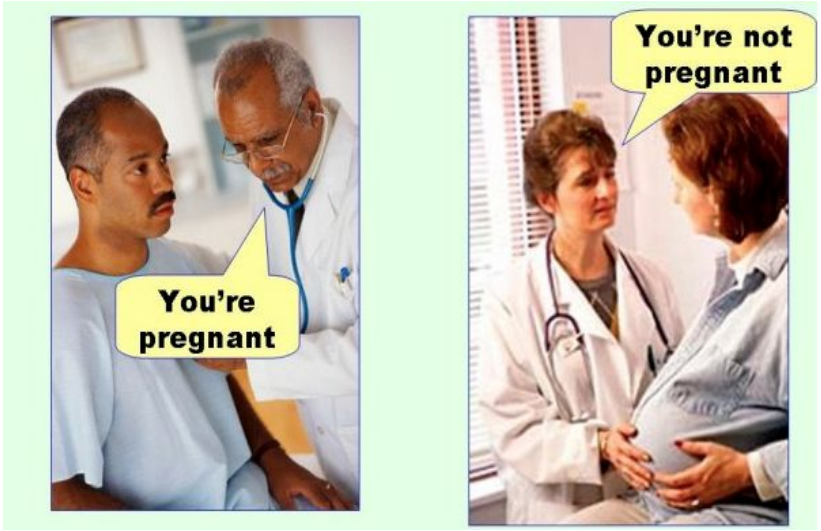


Fuente: Seeing Theory - Regression

Métricas de desempeño

Matriz de confusión

Imagina que vas al doctor y obtienes el diagnóstico equivocado



Fuente: [Computing for the Social Sciences, University of Chicago]

		predicted positive	predicted negative
truth	positive	tp	fn
	negative	fp	tn

Verdadero Positivo (tp): Los casos en que el modelo predice "si/positivo", y el valor real también es “si/positivo."

Verdadero Negativo (tn): Los casos en que el modelo predice "no/negativo", y el valor real también es “no/negativo."

Falso Positivo (fp): Los casos en que el modelo predice "si/positivo", y el valor real es “no/negativo."

Falso Negativo (fp): Los casos en que el modelo predice "no/negativo", y el valor real es “si/positivo."

Usando información de la matriz de confusión

		predicted positive	predicted negative
truth	positive	tp	fn
	negative	fp	tn

Número de datos:

n = tp + tn + fp + fn

Exactitud:

Qué tan seguido el clasificador está en lo correcto? => (tp + tn) / n

Razón de error:

Qué tan seguido el clasificador está equivocado? => (fp + fn) / n

Precisión:

Cuando el modelo predice “si”, qué tan seguido está en lo correcto? => tp / (tp + fp)

Recall (Razón de verdadero positivo):

Que tan seguido predice “si”, cuando es “si” realmente? => tp / (tp + fn)

Podemos predecir si una congresista es demócrata o republicana? Usemos la 1984 United States Congressional Voting Records Database

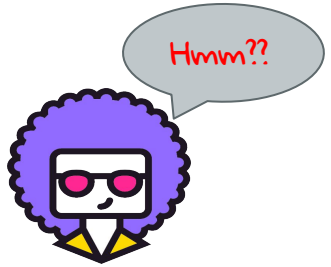
Asume que hemos seleccionado y entrenado un modelo (búsqueda de grid de hiperparámetros), etc y obtenemos el siguiente resultado:

```
Best estimators on the left out data:
LogisticRegression(C=6.1584821106602643, class_weight=None, dual=False,
          fit_intercept=False, intercept_scaling=2, max_iter=100,
          multi_class='ovr', n_jobs=1, penalty='l2', random_state=None,
          solver='liblinear', tol=0.0001, verbose=0, warm_start=False)

Best C / Regularization Param on the left out data:
6.15848211066

Best Params on hold out data (train):
{'C': 6.1584821106602643, 'intercept_scaling': 2, 'fit_intercept': False, 'solver': 'liblinear', 'penalty': 'l2', 'cl
ass_weight': None}

Best Score on left out data:0.964
```



Ahora, evalúa el modelo => sabiendo que si escogemos al azar del dataset, 61 % de las veces escogerás demócrata *(there are 267 democrats and 168 republicans in the dataset)*

Note: ***En esta clase, omitimos otros factores (imbalance de clase, como hacer búsqueda de grid de hiperparámetros, etc).***

Aquí está la matriz de confusión, calculemos algunos indicadores de desempeño

	Predict_Label_0 Republican	Predict_Label_1 Democrat
True_Label_0 Republican	49	2
True_Label_1 Democrat	2	78

Número de datos:

n = tp + tn + fp + fn => 49 +78 + 2 + 2 = 131

Exactitud:

Qué tan seguido el clasificador está en lo correcto? => (tp + tn) / n

=> (49+78) 131 => 0.9694 or 96.94%

Razón de error:

Qué tan seguido el clasificador está equivocado? => (fp + fn) / n

=> 4 / 131 => 0.03053 or 3.053%

Precisión:

Cuando el modelo predice “si”, qué tan seguido está en lo correcto? => tp / (tp + fp)

=> 49 / (49 + 2) => 97.5%

Recall (Razón de verdadero positivo):

Que tan seguido predice “si”, cuando es “si” realmente? => tp / (tp + fn)

=> 49 / (49 + 2) => 97.5%

Otros umbrales

Que tal si en lugar de mejorar la *exactitud* global, queremos mejorar una exactitud "clase-específica"?

- Este es el caso cuando queremos aumentar *sensitividad/recall* => aumentar la razón de verdadero positivo (TPR)
    - Razón de verdadero positivo = tp / (tp + fn) => 49 / (49 + 2) => 97.5%
- Por otro lado, si queremos aumentar la e*specificidad* tendremos que aumentar la razón de falsos positivos (TNR)
    - Razón de falsos positivos = fp / (fp + tn) => 2 / (2 + 78) => 2.5%

Cómo logramos esto?

- Estima un modelo mejor (logra mayor sensitividad y especificidad)
- Usa nuestro modelo actual para lograr uno de estos objetivos
    - Ajustando el umbral, o el *punto de corte* para clasificar individuales como "demócratas o republicanas"

Curva ROC

- Podemos graficar con varias selecciones de umbrales, y luego seleccionar el umbral en el punto en el que nos sentimos cómodos.
- El mejor enfoque es tener *conocimiento del dominio* en los beneficios de considerar un umbral (*trade off).*

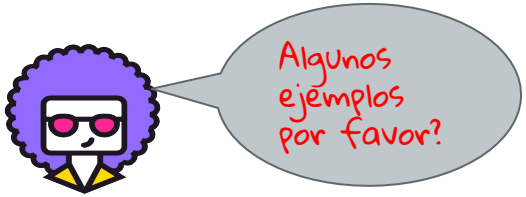
- Receiving Operating Characteristic (ROC) forma visual de determinar el desempeño de un *clasificador binario*

    - *En pocas palabras,con la curva ROC estamos midiendo el "trade off", o la razón, a la cual el modelo predice algo correctamente, con la razón a la cual el modelo predice algo incorrectamente.*
    - *A medida que el umbral aumenta para la clase positiva, la razón de falsos positivos y la razón de verdaderos positivos necesariamente aumentan.*

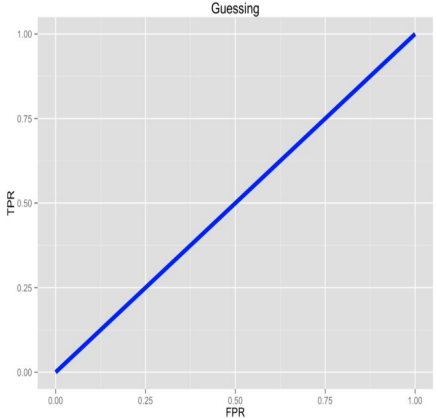
Desempeño

Evaluación de modelo

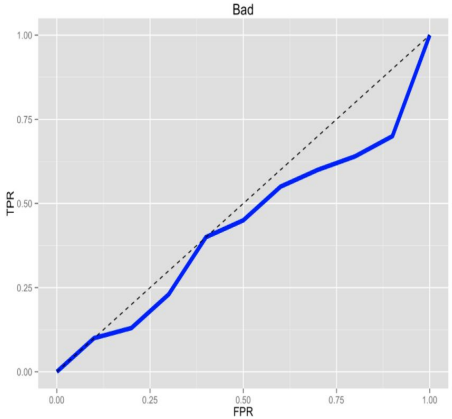
Curva ROC



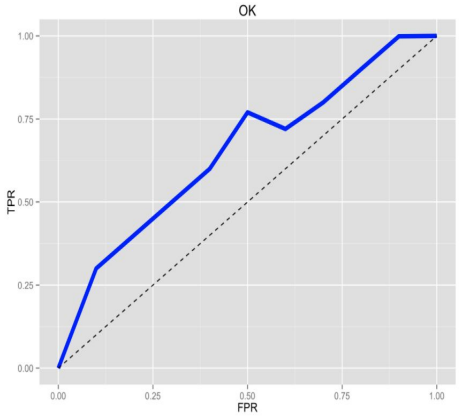
Clasificador adivina al azar (50/50)



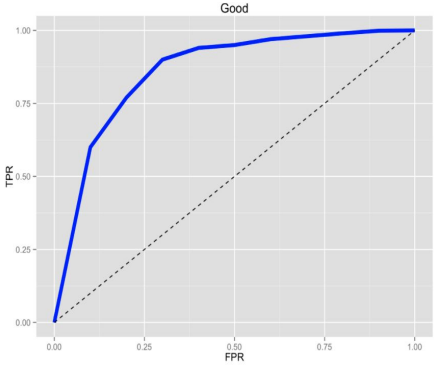
Peor que adivinar, la línea azul es menor a la segmentada



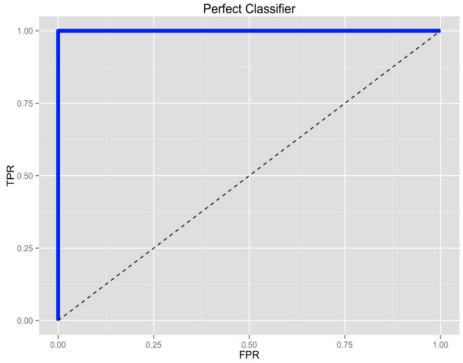
Clasificador mediocre, líneas que muestran bajas



Buen clasificador, El escenario ideal donde hay una curva con forma de joroba que crece continuamente



Un clasificador perfecto es el que muestra un trade-off perfecto entre TPR y FPR => TPR de uno y FPR de zero

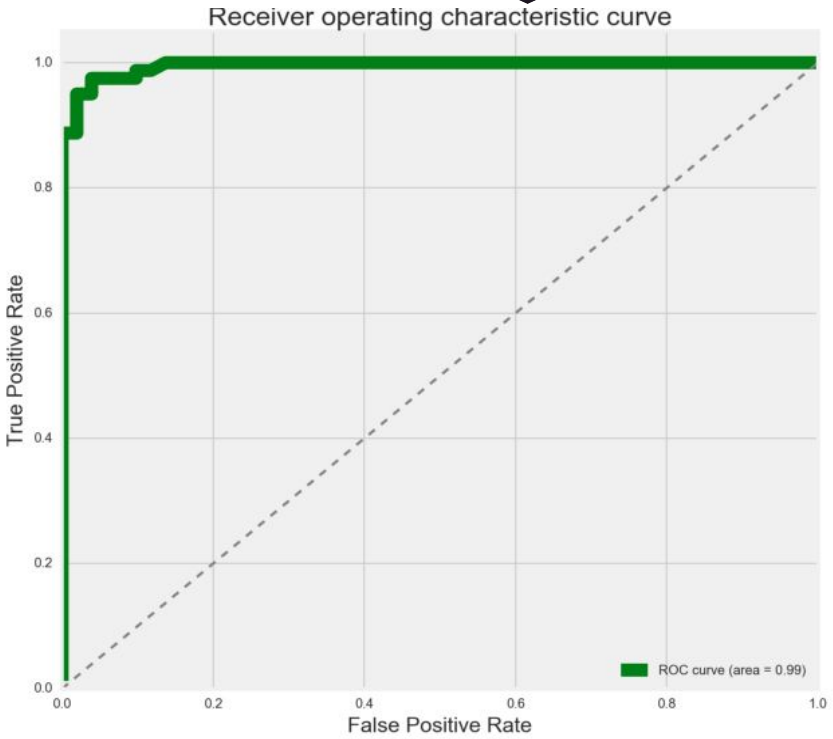


Source: yhat

Curva ROC y AUC

Hay un concepto extra que deberíamos saber:

- Área bajo la curva o AUC, es el área bajo la curva ROC.
- AUC muestra que tan bien TPR y FPR se muestran en conjunto.
- Mayor el área bajo la curva, muestra mayor calidad del modelo.
- Mayor el área bajo la curva, mayor la razón de verdaderos positivos a falsos positivos a medida que el umbral se vuelve más permisivo
  - AUC = 0 => MALO
  - AUC = 1 => BUENO



Checklist del módulo

- ✓ Regresión logística
  - ✓ Tarea
    - ✓ Dilema al usar OLS
    - ✓ Odds ratio
    - ✓ Logit link function
    - ✓ Umbrales de probabilidad
  - ✓ Experiencia de aprendizaje
    - ✓ Función de costo
    - ✓ Proceso de optimización
  - ✓ Desempeño
    - ✓ Matriz de confusión
    - ✓ ROC y AUC

Recursos Avanzados

Recursos adicionales:

- Libros
  - An Introduction to Statistical Learning with Applications in R (James, Witten, Hastie and Tibshirani): Chapters 4.1, 4.2 4.3

- Recursos en línea
  - Statistical learning: logistic regression - *MACS 30100 - Perspectives on Computational Modeling*
  - Simple guide to confusion matrix terminology
  - A Simple Logistic Regression Implementation

- Si está interesado en la búsqueda de hiperparámetros:
  - Tuning the hyper-parameters of an estimator
  - LogisticRegression (sklearn.linear_model)

Felicidades! ¡Terminaste el módulo!

Obtén más información sobre el machine learning de Delta para una buena misión aquí.