Tema 2.

Aprendizaje

Concepto: convertir experiencia en habilidad o conocimiento.

Entonces, un aprendizaje automático es programas a las computadoras para que aprendan de la entrada dada sin instrucciones explícitas. La entrada es la experiencia y la salida el conocimiento.

Marco de aprendizaje

Son los conjuntos X (variables de entrada) y Y (etiquetas del espacio). Se extrae S que es un conjunto de entrenamiento de N registros definidos en el espacio XxY, o sea S(X,Y) = {(xi,yi)} de i=1 hasta N. Este conjunto tiene una distribución de probabilidad conjunta, pero es desconocida.

La función que hace esto se llama predictor, hipótesis o clasificador y depende que aprendizaje hacemos. Pero se define como h: X -> Y. Se puede observar que la función relaciona los datos de entrada con la etiqueta, que es el dato objetivo o target. La finalidad? estimar o predecir nuevos ejemplos.

Se entiende que podemos tener diferentes hipótesis, porque podemos entrenar diferentes modelos. Entonces, este forma un conjunto H de todas las posibles hipotesis, de este conjunto tenemos una función correcta que etiqueta de manera correcta la relación entre X y Y, o sea f: X-Y, relaciona las variables con el target, pero lo importante es que esto es de la población total. Como no podemos hacer eso, entonces tenemos que tomar un conjunto de entrenamiento, lo cual es llamado S. Entonces el propósito es obtener un conjunto S y usar una función h de H a partir de S.

Midiendo el error

Esta es la forma de saber si se está aprendiendo correctamente. Se define como la probabilidad de no estimar correctamente la etiqueta correcta de un ejemplo aleatorio generado por la distribución de probabilidad del conjunto S. O sea la probabilidad de que h(x)!=f(x).

Formalmente, sea A un subconjunto aleatorio de X, la distribución de probabilidad P asigna un valor continuo P(A), determina la probabilidad de observar un ejemplo x que pertenezca a A. Se expresa mediante la función

$$\mathcal{L}(h(\mathbf{x}), f(\mathbf{x})) = \begin{cases} 1, & \text{Si } h(\mathbf{x}) \neq f(\mathbf{x}) \\ 0, & \text{Si } h(\mathbf{x}) = f(\mathbf{x}) \end{cases}$$

L es la función de pérdida del algoritmo de aprendizaje.

También se suele usar la notación de P:

$$\mathcal{R}_{P,f}(h) = \mathbb{P}_{\mathbf{x}\sim P}[h(\mathbf{x}) \neq f(\mathbf{x})] = P(\{\mathbf{x} : h(\mathbf{x}) \neq f(\mathbf{x})\}).$$

P,J indica que el error es sobre la distribución P y la función de etiquetado f. R es llamado como error de generalización, riesgo, error verdadero de h.

El objetivo es encontrar una función h tal que se minimice el error R.

$$h^* = \arg\min_{h\in\mathcal{H}} \mathcal{R}_{(P,f)}(h).$$

Que es encontrar el argumento mínimo dado h.

Riesgo empírico

El riesgo anterior no puede ser calculado porque al estar basado en f implica conocer el conjunto poblacional y solo tenemos S para hacer el entrenamiento, Se usa una aproximación que es conocida como riesgo empírico. Resulta promediar la función de perdida L sobre el conjunto S, con N ejemplos.

$$\mathcal{R}_{emp}(h) = \frac{\{i \in [N] : h(\mathbf{x}_i) \neq f(\mathbf{x}_i)\}}{N}$$

Se aplica la misma lógica que la anterior, o sea que se debe de minimizar el riesgo empírico eligiendo una hipótesis h que lo minimice.

$$\mathcal{R}_{emp}(h) = \frac{\{i \in [N] : h(\mathbf{x}_i) \neq f(\mathbf{x}_i)\}}{N}$$

Es lo mismo que el error de la población R. Ahora busca generalizar en base a S, se llama también aprendizaje inductivo.

Generalización, sobreajuste y desajuste

Si la función seleccionada para etiquetar ejemplos no vistos lo hace de manera correcta, o sea que el modelo hace estimaciones exactas sobre datos no vistos, entonces se dice que es capaz de generalizar bien el problema. Siendo este el objetivo al construir un modelo.

Un conjunto de hipótesis con una gran variedad de hipotesis, se elige una que sea consistente con la muestra y noi cometa errores. SI es menos compleja es posible tener errores en al muestra de entrenamiento.

Pero escoger el que menos error tenga en el conjunto de entrenamiento nos puede llevar a que el modelo esté memorizando, lo cual es diferente a generalizar.

Es importante tanto el tamaño de la muestra como la complejidad de las familias de hipótesis. Si tenemos una muestra pequeña, y si se toma una familia muy compleja entonces habrá mala generalización y se tendrá un sobreajuste. Esto significa que el modelo se ajusta mucho a las particularidades del conjunto de entrenamiento, o sea funciona excelente para entrenar pero muy malo en nuevos datos.

También se tiene el caso contrario donde hay desajuste, esto ocurre con modelos muy simples. Provoca que no se capture todos los aspectos y variabilidad de los datos, y como resultado funciona mal no solo con datos nuevos sino hasta con los de entrenamiento.

Algoritmos paramétricos y no paramétricos

Se sabe que existe un conjunto de hipotesis, pero no se conoce realmente la forma de la función correcta. Entonces, se debe de evaluar diferentes algoritmos para decidir cual se acerca más a f.

Se tiene diferentes modelos porque cada uno hace diferentes suposiciones y sesgos en el conjunto de entrenamiento, estas suposiciones se aplican a la forma de la función y en cómo aprende.

Un algoritmo paramétrico es aquel que describe los datos mediante un conjunto de parámetros fijos, que son los pesos, esto es independiente de la cantidad de muestras. Por ende, hace suposiciones de la forma de la función, un ejemplo es regresión lineal, de manera que estos parámetros son ajustados o aprendidos mediante el conjunto de entrenamiento.

Los no paramétricos no hacen suposiciones así de fuertes como los anteriores. Por ende, aprenden de forma libre los datos de entrenamiento. Un ejemplo es knn. Estos algoritmos buscan ajustarse a los datos de entrenamiento en la construcción de la función de mapeo.

Riesgo estructural

Debemos de ver que en sí un modelo h seleccionado de un conjunto de datos finito, siempre se va tener problemas con sobreajuste.

Este tipo de riesgo va abordar este problema, o sea el sobreajuste. Lo que hace es balancear la complejidad del modelo contra el sobreajuste durante el entrenamiento. Se tiene que minimizar lo siguiente:

$$\mathcal{R}_{srm}(h) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(h(\mathbf{x}_i), f(\mathbf{x}_i)) + \lambda\Omega(h)$$

El primero es el riesgo empírico porque es el promedio de la función de perdida mas una función de regularización, la cual se determina su importancia de esa función con lambda.

Regularización: es una modificación de un algoritmo que reduce el error sobre los datos no vistos, pero no en los datos de entrenamiento. Solo afectando a los nuevos.

Se va suponer que la funcion h es paramétrica, o sea está representada por parámetros. Tal que la función de regularización tiene que ser de tal modo que penalice los parámetros que producen alta complejidad en el modelo. El término termina penalizando pesos mayores.

Minimizar la fuinción de regularizaciójn limita la capacidad de los subconjuntos del espacio de parámetros, cointrola la compensación entre minimizar el error de entrenamiento y la brea del error de entrenamiento con el error de prueba.

Lambda lo que hace es dar más o menos importancia al término de regularización tal que un valor grande fomenta ponderaciones más dispersa, un valor pequeño relaja la regularización.

Complejidad del modelo

Es la variedad existente en la familia de funciones de conjuntos de hipótesis.

Tema 3.

Regresión Lineal

Algoritmo de aprendizaje supervisado, calcula una relación lineal entre X y Y.

La variable dependiente es es la variable respuesta que es Y, mientras que las variables explicativas son X, pueden ser una o más variables, son las independientes.

La relación es modelada con funciones predictoras lineales, pero sus parámetros son desconocidos y son ajustados usando los datos de la muestra, o sea S. Predice variables continuas.

Tiene dos categorías de aplicaciones. Se puede aplicar regresión lineal para ajustar un modelo predictivo a un conjunto de datos S. Una segunda aplicación es que sirve para explicar la variación de la respuesta que se atribuye a la variación en las variables explicativas, regresión lineal podría cuantificar la fuerza de la relación entre la respuesta y la variable.

El modelo tiene una sola variable independiente. De manera que theta 1 es el peso para xi. Dado un conjunto S, entonces el algoritmo pretende encontrar f para tener la mejor linea, por medio de los parámetros theta 0 y 1. Segun el valor de theta, indica una relación directa o inversa entre x & y.

Esto se puede extrapolar a una regresión lineal multiple, donde se relacione la variable dependiente yi con el vector xi con m variables explicativas. O sea que tenemos m caracteristicas. Teniendo m thetas, que serían los parámetros del modelo. Esto mismo se puede resumir de manera vectorial como: yi = xi^T * Theta + epsilon, epsilon es una variable aleatoria que recoje factores que se asociacion al azar. Asi mismo se debe de considerar el término constante theta 0.

Suposiciones que se hacen en el conjunto de datos de entrenamiento:

1. Linealidad: X y Y deben estar relacionadas linealmente entre sí.
2. Independencia: la variable dependiente de una observación no depende de otra observación.
3. Homocedasticidad: las variables independientes, la varianza de los errores es constante. La cantidad de las variables independientes no tienen impacto en la varianza.
4. Sin multicolinealidad: sin alta correlación entre las variables independientes.

ECM

Es para medir la calidad de los ajustes. Siendo la función de costo o pérdida L para la regresión lineal. Esto no es otra cosa que la diferencia entre el valor predicho y el real, pero es una suma, porque necesitamos el promedio de la diferencia entre el valor predicho y el real al cuadrado.

Mínimos cuadrados

Método usado para ajustar los parámetros a los datos de entrenamiento. Es usado normalmente, debido a que es una función convexa. Este método lo que hace es minimizar la suma de cuadrados de las diferencias de los residuos. Siendo el residuo la diferencia entre los puntos generados de la función y los correspondientes a los datos de la muestra. O sea predicciones y valores reales.

Hay una versión lineal y no lineal.

El modelo tiene de la forma la función f, el modelo tiene m parámetros, o sea m x's, de manera que son ajustados por el vector theta. Entonces el método es minimizar la suma de los residuos, siendo: y - f(x,theta), siendo un residuo al cuadrado.

El mínimo de la suma de cuadrados se encuentra asignando el gradiente a cero, porque es función convexa. Se encuentra derivando para maximizar y se iguala a 0.

Se tiene uan versión con notación matricial.

$$\mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}; \boldsymbol{\theta}) = \|\mathbf{Y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 \quad = \quad \boldsymbol{\theta} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{Y}$$

De esta manera se encuentran los pesos theta que minimizan la función de pérdida.

Se puede hacer lo mismo para una versión cuadrática o hasta p potencias. Esto es usando una matriz de cambio de base Z. Donde cada valor de X se eleva desde 0 hasta p potencia.Y se hace lo mismo que la anterior expresión para hacer el cálculo. y=Z * theta.

Se puede tener hasta bases exponenciales, logaritmos, funciones trigonométricas. Una buena base mejora el rendimiento pero no es obvia cual base necesitamos.

Problemas con este método: los atributos son independientes, entonces X^T * X debe de ser invertible; sensible a valores atípicos; si son muy grandes los datos, no se puede almacenar X^T * X; el costo de X^T * X es O(nm^2), par asu inversi[on es O(m^3). Complejidad total O(nm^2 + m^3)

Descenso de gradiente

Algoritmo de optimización para encontrar valores de parámetros, minimizan la función de pérdida. Se emplea porque hay funciones de costo o pérdida que son complicadas de minimizar.

Tema 5.

Selección de modelos

Nos interesa cumplir con requisitos específicos, como mantenimiento, escalabilidad, complejidad, etc. Esto ocasiona que se debe de hacer simple, entre más simple es más fácil de entender pero el problema es que no es predictivo. Lo cual es lo que no queremos, entonces debemos de poner lo de predictivo que los anteriores de mantenimiento,etc.

La selección es el proceso de elegir un modelo entre una colección de modelos de aprendizajes candidatos, para un problema de modelado predictivo.

Complejidad es la alta capacidad de modelos para ajustar.

No es lo mismo la selección de modelo que evaluar el desempeño de un modelo. En la selección es para elegir el nivel adecuado de flexibilidad.

Siempre habrá error predictivo, no hay modelo perfecto sino lo suficientemente bueno.

Para el desarrollo de un modelo ideal es: filtrado y transformación de datos, extracción y selección de características, configuración de hiper parámetros.

Como en la mayoría de veces las aplicaciones el suministro de datos para el entrenamiento y las pruebas es limitado. Por ende, para un conjunto de validación pequeño se tendrá uina estimación ruidosa del rendimiento predictivo. Por ende, existen diferentes criterios para obtener la mejor aproximación de un modelo ideal.

Errores de predicción

El error de predicción de cualquier algoritmo de aprendizaje se puede dividir en error de sesgo, varianza y el error irreductible.

Error irreductible: es el término de ruido que ningun modelo puede reducir, ya son el ruido del los datos. Solo se puede tener control del sesgo y varianza.

Error de sesgo

Es la diferencia entre la predicción esperada, promedio, y la real. El sesgo mide que tan lejos están las predicciones de los valores correctos.

Error varianza

Es la variabilidad de la predicción de un modelo para un ejemplo. Es cuánto varían las predicciones para un punto dado entre diferentes ejecuciones del modelo.

Descomposicion sesgo varianza

El error de generalización es la suma del error de sesgo, varianza y el irreductible. El chiste es que tenien una función que f que genera un doncjunto D, entonces queremos obtener un modelo f_hat que se aproxime a esa f original. Tal que el ecm sea el minimo.

El error esperado varia con respecto a diferentes opciones del conjunto D

El sesgo al cuadrado nos dice el error causado por suposiciones simplificadas, por ejemplo para un problema no lineal, suponemos una base, entonces esa suposicion no es algo 100% y por ende este sesgo al cuadrado mide esos errores que se crean por esas supociiones

Varianza es cuanto se va mover las predicciones con respecto a la media

La desv standard nos dice los errores asociados a los datos, como variables desconocidas que influyen en el mapeo

sobreajuste y desajuste

Esto es relacionado a sesgo y varianza, al agregar más parámetros entonces aumenta la complejidad del modelo, por ende el sesgo disminuye mientras que la varianza aumenta, entonces se debe de encontrar ese balance entre ambos, y es un punto de balance donde la varianza y y sesgo sean lo más mínimo y ninguno aumente. Pero encontrar eso no es tan fácil.

Lo optimo es llegar donde la disminucion del sesgo es el aumento de la varianza, entonces tenemos ese punto optimo. si nos pasamos de ese punto entonces es sobreajuste y si no llegamos al punto es sobreajuste

Es por esto que como no es fácil encontrar ese punto tenemos que hacer muchas pruebas de diferentes modelos para minimizar el sesgo a un punto óptimo o lo más cercano a este realmente.

Primero se toma una medida de error precisa. Después se prueban diferentes niveles de complejidad de un mismo modelo y se elige un complejidad que minimice el error general. Que es el error de descomposición de sesgo y varianza

Criterios de selección de modelos

El problema más común es que los modeladores reportan errores basados en los datos usados para entrenar, en lugar de reportar errores sobre datos nuevos. Una medida incorrecta del error puede llevar a una predicción inferior.

El optimismo de entrenamiento es una medida de qué tan bien se comporta el modelo con datos nuevos, a comparación de los datos de entrenamiento. Esta medida se puede medir como una función de complejidad del modelo, si aumenta la complejidad aumenta el optimismo, debido a que se ajusta cada vez más los datos de entrenamiento.

Error de predicción verdadero = Error de entrenamiento + f(Complejidad del modelo)

Basado en su error de predicción y complejidad se tienen diferentes medidas para seleccionar un modelo

Técnica basada en teoría de la información

Enfoque basado en la cantidad de información que se pierde entre un modelo candidato y el modelo real. Estos parten de un modelo paramétrico, se puede definir la verosimilitud de un conjunto de datos y parámetros, para definir una probabilidad de observar los datos dado los parámetros. Como una probabilidad condicional.

Si ajustamos los parámetros para maximizar la verosimilitud, obtener una estimación a esta misma. Usar esto para comparar entre modelos y diferentes complejidades.

Criterio de información de Akaike AIC

Función de probabilidad de un modelo específico y su cantidad de parámetros.

$\mathsf{AIC} = -\frac{2}{n}\ln(\mathsf{Likelihood}) + \frac{2k}{n}$ Se minimiza el AIC

Likelihood es la verosimilitud del modelo en el conjunto de entrenamiento, k es la cantidad de parámetros, n es la cantidad de registros del conjunto.

Primer término de la ec es la tasa de error del conjunto, el segundo es la penalizacion para ajustar la complejidad.

Criterio de información Bayesiano BIC

$$\mathsf{BIC} = -2\ln(\mathsf{Likelihood}) + k\ln(n)$$

Equilibra el modelo desde una perspectiva bayesiana. Penalizando con mayor fuerza la complejidad del modelo en comparación con el AIC, da modelos más simples a comparación de AIC.

Estos enfoques de teóricos son muy complejos, y dependen de muchas teorías. Entre sus ventajas es que son ffaciles de aplicar, y está integrado en la mayoría de programas de analisis avanzados. Entre sus desventajas es que no es comparables con diferentes aplicaciones, el modelo debe generar probabilidades.

Holdout sample

Este approach no parte de suposiciones, usando los datos propios. Se divide el conjunto inicial en dos grupos, uno usado para entrenar y el segundo para medir el modelo resultante.

Un error común es crear un conjunto de reserva, entrenar un modelo y luego ajustar con los datos de reserva de manera iterativa.

Pros: no tiene suposiciones, es preciso si hay suficientes datos, fácil de implementar y entender.

Contras:posible sesgo que ocasiona sobreajuste, si usamos el conjunto de reserva para entrenar entonces el modelo se contamina, se debe elegir un tamaño de reserva adecuado, lo más común es 70 y 30.

Validación cruzada

Técnica basada en remuestreo. Divide los datos en conjuntos de k pliegues. Para el caso de 5 pliegues con 100 ejemplos, se hacen 5 pliegues de 20 ejemplos cada uno, luego la construcción del modelo y estimación de errores se repite 5 veces, donde se combinan 4 grupos para entrenar el modelo, el pliegue restante es usado para estimar el error de predicción real. Se terminaría con 5 estimaciones de error, se promedia y se obtiene una estimación del error de predicción real.

Esta técnica es preferible cuando se tiene datos límitados.

Esta técnica también puede proporcionar estimaciones de la variabilidad de la estimación del error real.

Cuando menor sea el número de pliegues, más sesgado son las estimaciones de errores. Aunque se toma en cuenta que en cada iteración se crea un nuevo modelo, entonces es un proceso lento, por lo quie sería conveniente usar un número pequeño de pliegues de ser así.

La más usada es 5 a 10 pliegues, una versión de 5x2 ha sido sugerida.

Pros: sin suposiciones, datos suficientes precisos, simple

Contra: computacionalmente costos, elegir tamaño del pliege y hay posible sesgo conservador.

Este approachg da más confianza y seguridad en conclusiones resultantes.

Configuración de hjiperparámetros

Optimización de hiperparámetros: elegir un conjunto de hiperparámetros óptimo para un algoritmo.

Hiperparámetro: parámetro usado para controlar proceso de aprendizaje.Son puestos a la hora de entrenar y difieren de los parámetros que son aprendidos o ajustados durante el entrenamiento. Un ejemplo es la cantidad de vecinos en knn.

Una mala selección de hiperparametros puede el modelo no ajustarse o sobreajustarse.

Búsqueda manual

Usado cuando hay pocos hiperparámetros y el modelo es simple. El proceso es definir un conjunto de valores posibles para cada hiperparámetro, se van seleccionando y ajustando manualmente los valores hasta que el rendimiento sea bueno. Por ejemplo usar una tasa de aprendizaje de 0.1 e ir aumentando o decrementando.

Este método puede llevar mucho tiempo porque se trata de encontrar una combinación óptima. Siendo propenso a errores, puede pasar por alto combinaciones o no evaluar el impacto de cada hiperparámetro.

Técnicas de optimización con algoritmos automatizados

Estos son algoritmos automatizados, pueden explorar grandes espacios de hiperparámetros.

Se puede usar grid search, random search y optimizacion bayesiana.

Grid search

Implica entrenar el modelo con cada combinación posible de hiperparámetros del conjunto predefinido. Por cada modelo entrenado, se evalua usando una métrica específica, como F1 score o exactitud.

El modelo con mejor rendimiento es el elegido.

Pros: usado por simplicidad y eficacía.

Contras: mucho esfuerzo computacional, limita al conjunto predefinido.

Random search

Implica seleccionar de forma random una combinación de hiperparámetros de un conjunto predefinido y se entrena el modelo usando esa selección.

Se debe de predefinir un conjunto para cada hiperparámetro, luego se escoge de manera aleatoria cada una. De la misma manera para cada modelo se evalua con una métrica como f1 score o precisión.

Este proceso se repite k veces.

Pros: facil de usar y de implementar

Contras: menos sistemático y puede no encontrar la combinación óptima.

Optimización bayesiana

Construye un modelo probabilistico de la función objetivo en función de los valores de hiperparámetros probados hasta el momento.

Tema 4. Regresión logística

Modelos lineales para clasificación

Toma un vector de entrada y lo asigna a k clases. De manera que es una función f que mapea cada entrada a una clase, de 1 a k clases.

La técnica es dada la entrada se divide en regiones de decisión para cada clase, estos limites se llaman límites o superficies de decisión. En caso de modelo lineal para clasificación, entonces las superficies de desicion son funciones lineales. Las clases deben de separase or decisiones lineales.

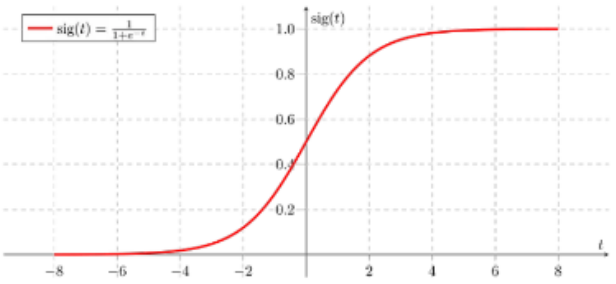
En el caso de regresión lineal no se puede usar tal cual para clasificación, lo que hace regresión es establecer una recta, y son valores reales. Entonces, para el caso de clasificación se tendrá en la recta las clases existentes, por lo que tendríamos que colocar un umbral para decir si es 0 o 1. El problema de esto es que regresión lineal tiende a ajustarse a los datos, entonces para cada dato nuevo que se agrega se tendría que redefinir el umbral. Siendo que no debería de funcionar así.

Aquí es donde entra la función sigmoide logística

Esto es para problemas de clasificación binaria, son dos grupos en y = {0,1}.

Lo que se desea es predecir etiquetas de clases o probabilidades posteriores en un rango de 0 a 1.

$$\sigma(a) = \frac{1}{1 + e^{-a}}. \qquad (1)$$



La característica principal es que es una función de comprensión, mapea todo el eje real en un intervalo finito. Por ende de todos los valores que tenemos en nuestro conjunto, podemos emplearlo para un intervalo acotado de 0 a 1.

De manera que se puede estimar una cantidad continua, en una probabilidad, se puede usar un umbral para definir en que punto considerar cuando es clase 0 y cuando clase 1.

Propiedades de la función sigmoide

1. Acotada de 0 a 1
2. Simétrica o sea que 1 - sigmoide(a) = sigmoide(-a)
3. Es diferenciable = sigmoide(a) * (1 - sigmoide(a))

La tarea es predecir la probabilidad de que x pertenezca a una clase y: p(y|x)

Para saber si pertenece a C1, entonces se escribe como p(C1 | x), para C2 es p(C2 | x) = 1 - p(C1 | x)

La decisión sobre qué clase aplicar a un ejemplo x, está dado por un umbral

1 Si P(y=1 | x) > 0.5

0 en otro caso

Función logit:

Esta nos dice que el logaritmo de la razón de probabilidades p(C1 | x)/p(C2 | x) es equivalente a la expresión de regresión lineal.

Sirve como vinculo entre la probabilidad y la expresión de regresión lineal. Oscila entre finito positivo y negativo, da un criterio adecuado para el cual se realiza la regresión lineal y el logit se convierte en probabilidades.

Función odds

Resulta después de exponenciar ambos lados de la función logit. La cual nos dice que la probabilidad de que la variable dependiente sea igual a un caso, es equivalente a la función exponencial de la regresión lineal. Dada una combinación lineal x de los predictores.

Función de costo

No se usa ECM porque crea complicaciones al encontrar un mínimo global con descenso de gradiente. Como los resultados son entre 0 a 1, es dificil hacer un seguimiento de errores por los valores flotantes.

La función a minimizar es la función de perdida log loss.