UNIVERSIDAD
NACIONAL
DE COLOMBIA

APRENDIZAJE DE MÁQUINAS

Métodos Supervisados

JOHN W. BRANCH

Profesor Titular
Departamento de Ciencias de la Computación y de la Decisión Director del Grupo de I+D en Inteligencia Artificial – GIDIA
jwbranch@unal.edu.co

Tomado y actualizado de presentaciones del docente:

Carlos A. Mera B.

Departamento de Ciencias de la Computación y de la Decisión Investigador del Grupo de I+D en Inteligencia Artificial – GIDIA

Contenido

1. **Regresión**
   a. Regresión Lineal
   b. Regresión Polinomial
   c. Regresión Lasso, Ridge y Elastic-net
   d. Árbol de Decisión para Regresión
   e. Red Neuronal para Regresión

2. **Clasificación**
   a. Regresión Logística
   b. K-nn
   c. Máquina de Vectores de Soporte
   d. Algoritmo XGBoost para Clasificación
   e. Red Neuronal para Clasificación

Regresión Lineal

CORRELACIÓN ENTRE DOS VARIABLES:

Se considera que dos variables cuantitativas (x e y) están correlacionadas cuando una de ellas (y) varía sistemáticamente con respecto a los valores de la otra (x).

Por ejemplo:

- ¿Hay una correlación entre la Temperatura y el número de Helados Vendidos en una Heladería?
- **¿Puede identificar otras correlaciones?**

Claro está, si sabemos que la variable x está correlacionada con y, quiere decir que podemos **predecir** la viarble y a partir de x.

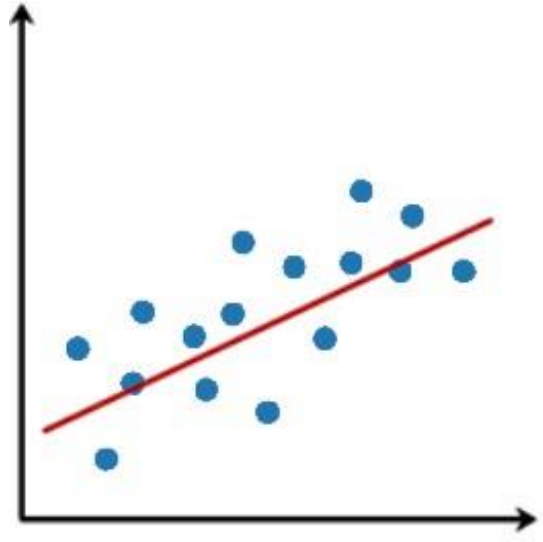
Estamos en el terreno de la PREDICCIÓN!

Gidia Grupo de I+D en Inteligencia Artificial
UNIVERSIDAD NACIONAL DE COLOMBIA

Regresión Lineal

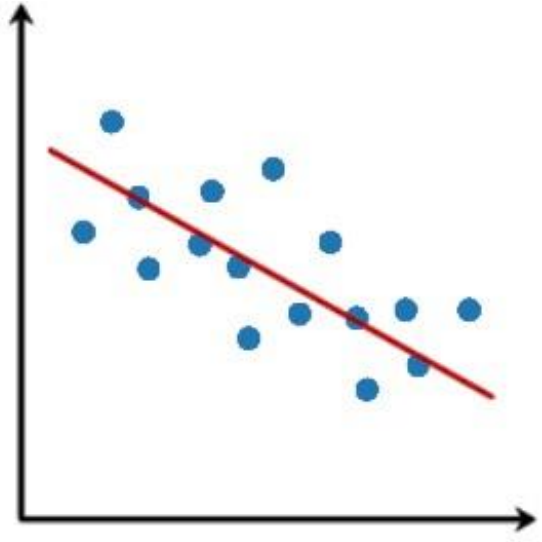
- **Tipos de Correlación:**

Hay tres tipos básicos de correlación: positiva, negativa y nula (sin correlación).



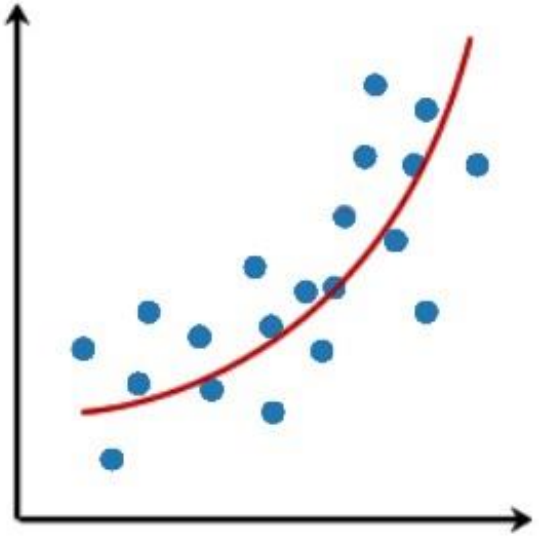
Correlación Lineal Positiva

Ocurre cuando una variable aumenta y la otra también

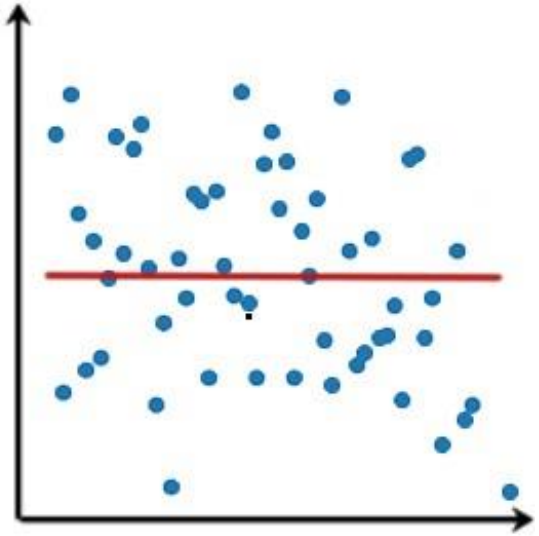


Correlación Lineal Negativa

Ocurre cuando una variable aumenta y la otra disminuye



Correlación No Lineal

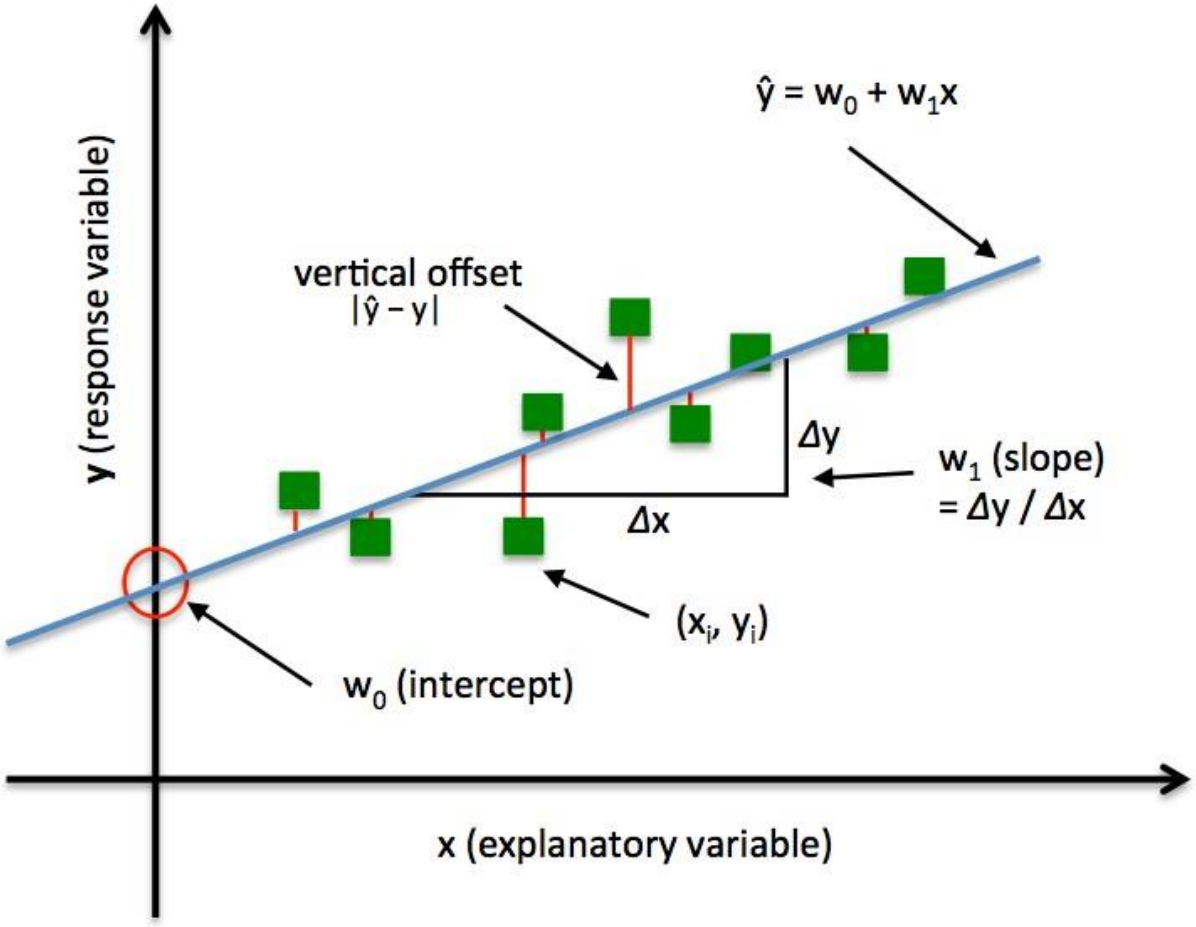


Sin Correlación

No hay una relación aparente entre las variables

Gidia Grupo de I+D en Inteligencia Artificial
Universidad Nacional de Colombia

Regresión Lineal



- **Generalidades:**

- Los métodos de regresión buscan modelar la relación entre 2 variables.
- El modelo se ajusta usando una medida de error sobre las predicciones que éste hace.
- En la **Regresión Lineal** el modelo a ajustar es una línea recta:

$$\hat{y} = w_0 + w_1 x$$

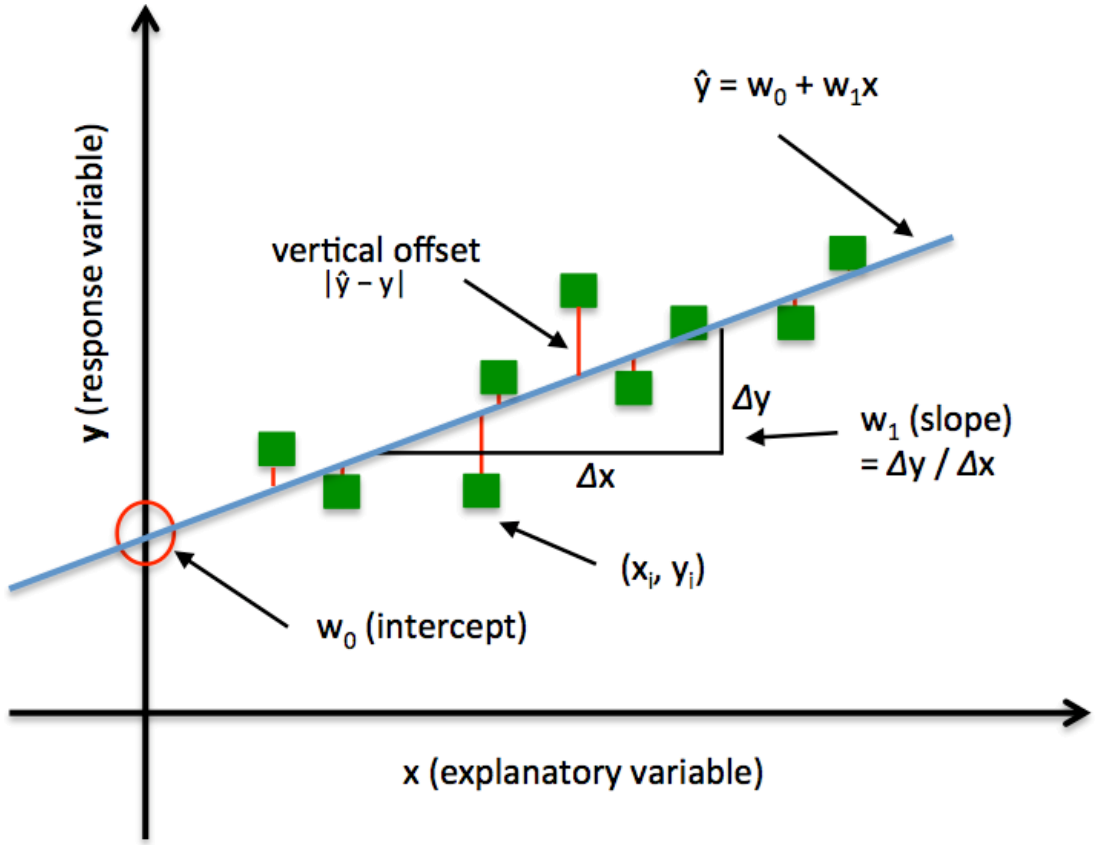
Puede haber múltiples líneas rectas dependiendo de los valores de intercepción y pendiente. Básicamente, lo que hace el algoritmo de regresión lineal es ajustar varias líneas y retornar la línea que produce el menor error.

Gidia Grupo de I+D en Inteligencia Artificial
UNIVERSIDAD NACIONAL DE COLOMBIA

Regresión Polinomial

La dependencia entre la *variable de respuesta* y la *regresora* frecuentemente no es lineal. ¿Cómo determinar la significancia de la desviación del supuesto de linearidad?
Una de las maneras más sencillas es usando la **regresión polinomial**, donde:

$$y = a + bx + cx^2 + dx^3 \ldots$$



Gidia Grupo de I+D en Inteligencia Artificial

UNIVERSIDAD NACIONAL DE COLOMBIA

Regresión Ridge, Lasso y Elastic-Net

Links de interés:

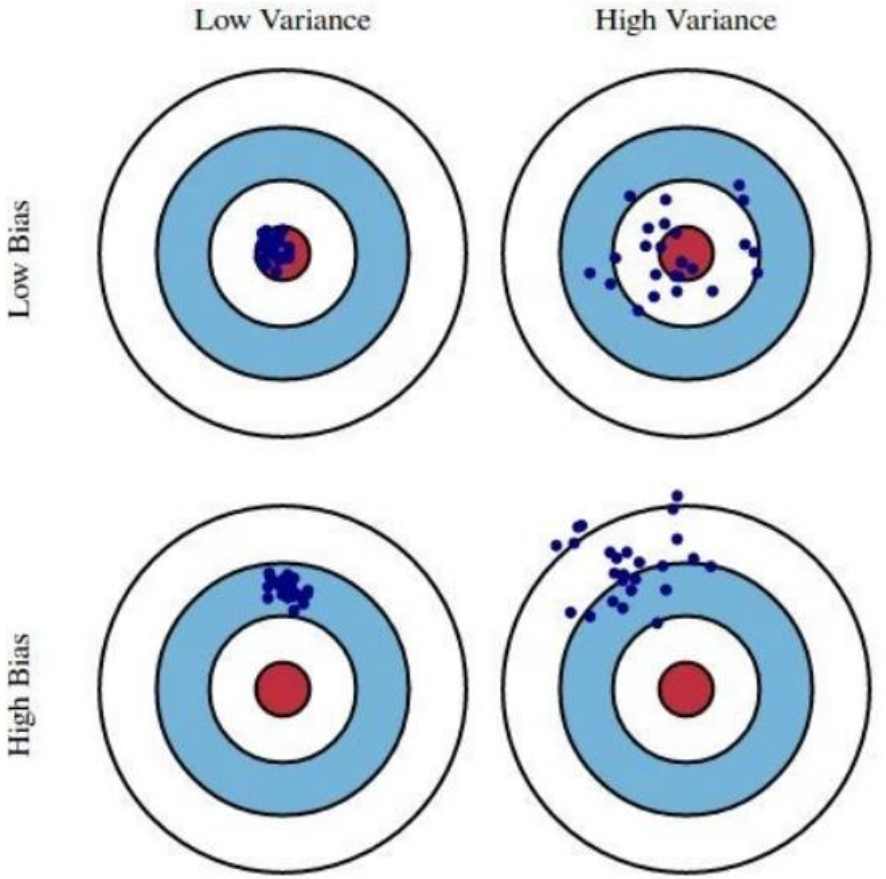
- https://hackernoon.com/an-introduction-to-ridge-lasso-and-elastic-net-regression-cca60b4b934f
- https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net

¿Qué es el Bias?

El sesgo (bias) es la diferencia entre la predicción media de nuestro modelo y el valor correcto que intentamos predecir. Los modelos con alto sesgo prestan muy poca atención a los datos de entrenamiento y simplifican en exceso el modelo. Siempre lleva a un alto error en los datos de entrenamiento y de prueba.

¿Qué es la Varianza?

La varianza es la variabilidad de la predicción del modelo para un punto de datos dado o un valor que nos dice la dispersión de nuestros datos. Los modelos con alta varianza prestan mucha atención a los datos de entrenamiento y no generalizan sobre los datos que no han visto antes. Como resultado, tales modelos funcionan muy bien en los datos de entrenamiento pero tienen altas tasas de error en los datos de prueba.



Variance/bias trade off (KDnuggets.com)

Gidia Grupo de I+D en Inteligencia Artificial

UNIVERSIDAD NACIONAL DE COLOMBIA

Regresión Ridge, Lasso y Elastic-Net

Regresión Ridge
Utiliza la regularización **L2** como forma de penalidad para el ajuste en la ecuación objetiva (Regresión lineal/polinomial).

El término L2 es igual al cuadrado de la magnitud de los coeficientes. En este caso si lambda(λ) es cero entonces la ecuación es la básica, pero si es mayor que cero entonces añadimos una restricción a los coeficientes. Esta restricción da como resultado unos coeficientes minimizados que tienden a cero cuanto mayor sea el valor de lambda.

$$+\lambda \sum_{j=0}^{p} w_j^0$$

L2 regularization penalty term

Gidia Grupo de I+D en Inteligencia Artificial
UNIVERSIDAD NACIONAL DE COLOMBIA

Regresión Ridge, Lasso y Elastic-Net

Regresión Lasso

Utiliza la regularización **L1** como forma de penalidad para el ajuste en la ecuación objetiva (Regresión lineal/polinomial).

Similar a la regresión Ridge, un valor lambda de cero nos da la ecuación básica. Sin embargo, dado un valor lambda adecuado la regresión de lazo puede llevar algunos coeficientes a cero. Cuanto mayor sea el valor de lambda, más características se reducen a cero.

$$+\lambda \sum_{j=0}^{p} |w_j|$$

L1 regularization penalty term

Gidia Grupo de I+D en Inteligencia Artificial
UNIVERSIDAD NACIONAL DE COLOMBIA

Regresión Ridge, Lasso y Elastic-Net

Regresión Elastic-Net

Utiliza la regularización **L1 y L2** como forma de penalidad para el ajuste en la ecuación objetiva (Regresión lineal/polinomial).

Además de configurar y elegir una Elastic-Net de valor lambda también nos permite afinar el parámetro alfa donde $\boldsymbol{\alpha}$ = 0 corresponde a Ridge y $\boldsymbol{\alpha}$ = 1 a Lasso. En pocas palabras, si ponemos 0 para alfa, la función de penalización se reduce al término L1 (Ridge) y si ponemos alfa en 1 obtenemos el término L2 (Lasso).

$$\frac{\sum_{i=1}^{n}(y_i - x_i^J \hat{\beta})^2}{2n} + \lambda \left(\frac{1-\alpha}{2} \sum_{j=1}^{m} \hat{\beta}_j^2 + \alpha \sum_{j=1}^{m} |\hat{\beta}_j| \right)$$

Elastic net regularization

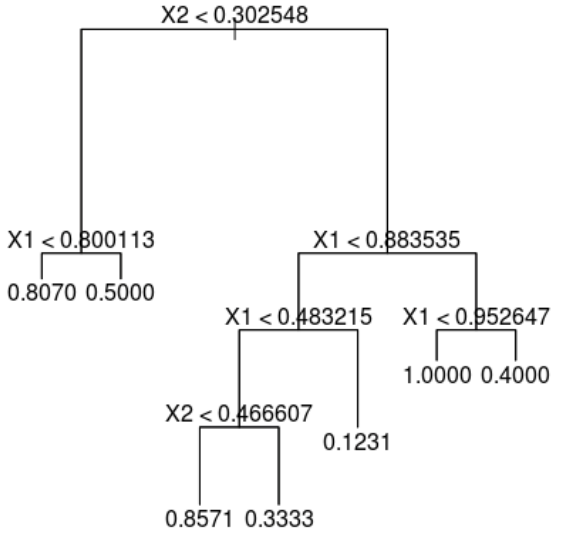
Gidia Grupo de I+D en Inteligencia Artificial
UNIVERSIDAD NACIONAL DE COLOMBIA

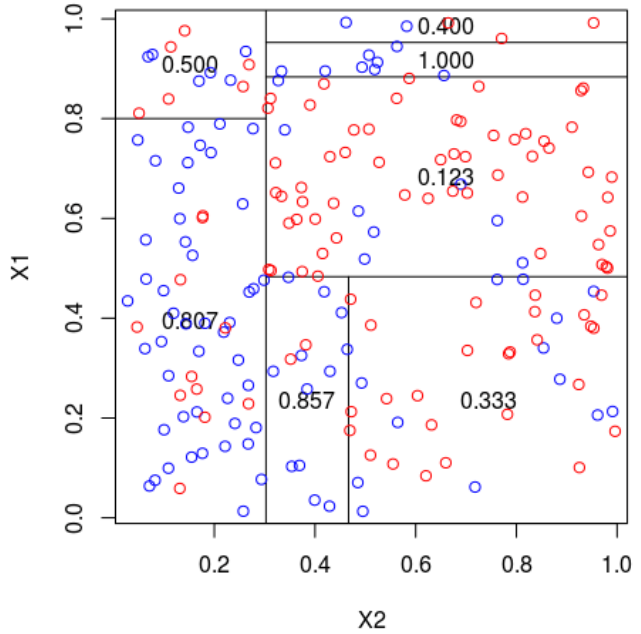
Árbol de Decisión para Regresión

Links de interés:
- https://gdcoder.com/decision-tree-regressor-explained-in-depth/

Los Árboles de Decisión pueden resumirse con los siguientes puntos:
- Los árboles de decisión son modelos predictivos que utilizan un conjunto de reglas binarias para calcular un valor objetivo.
- Cada árbol individual es un modelo bastante simple que tiene ramas, nodos y hojas.

Un árbol de decisión consisten en estimar un resultado haciendo una serie de preguntas a los datos, cada pregunta estrechando nuestros posibles valores hasta que el modelo tenga la suficiente confianza como para hacer una sola predicción. El orden de la pregunta así como su contenido son determinados por el modelo. Además, las preguntas formuladas están todas en forma de Verdadero/Falso.





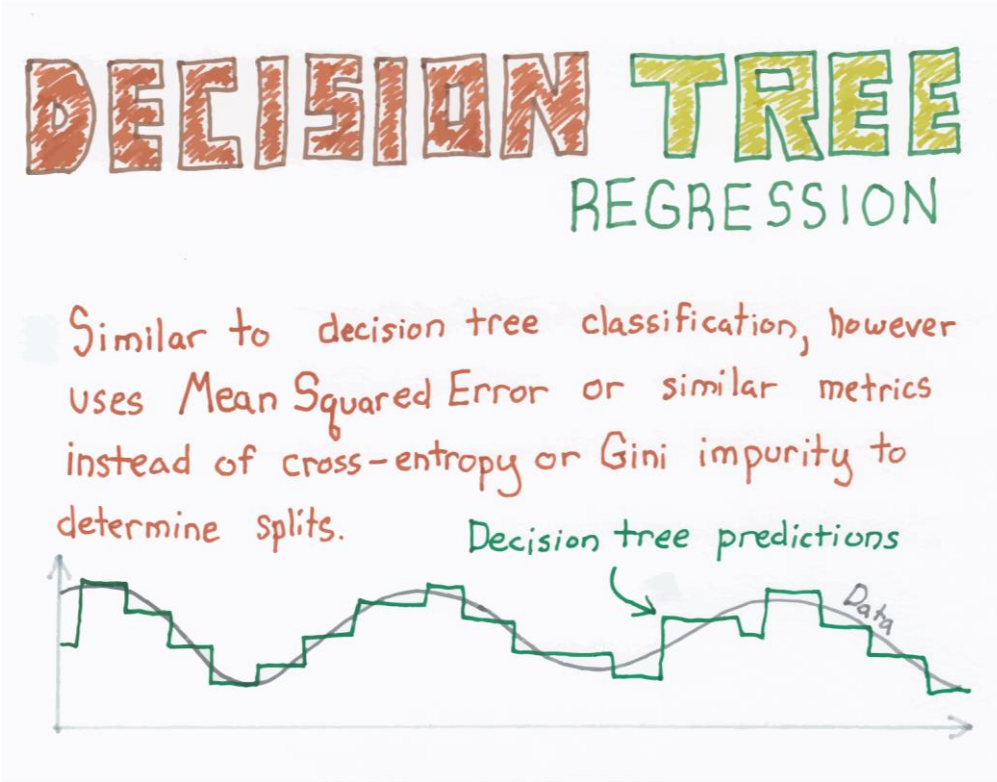
Gidia Grupo de I+D en Inteligencia Artificial

UNIVERSIDAD NACIONAL DE COLOMBIA

Árbol de Decisión para Regresión

Durante el entrenamiento, el modelo se ajusta con cualquier dato histórico que sea relevante para el dominio del problema y el verdadero valor que queremos que el modelo aprenda a predecir. El modelo aprende cualquier relación entre los datos y la variable objetivo.

Después de la fase de capacitación, el árbol de decisión produce un árbol calculando las mejores preguntas, así como su orden para hacer las estimaciones más precisas posibles. Cuando se quiere hacer una predicción se debe proporcionar al modelo el mismo formato de datos para hacer una predicción. La predicción será una estimación basada en los datos del tren en el que se ha entrenado.



Gidia Grupo de I+D en Inteligencia Artificial

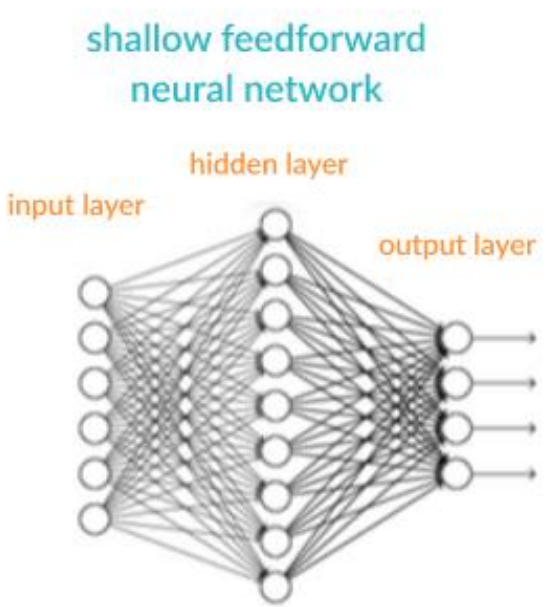
UNIVERSIDAD NACIONAL DE COLOMBIA

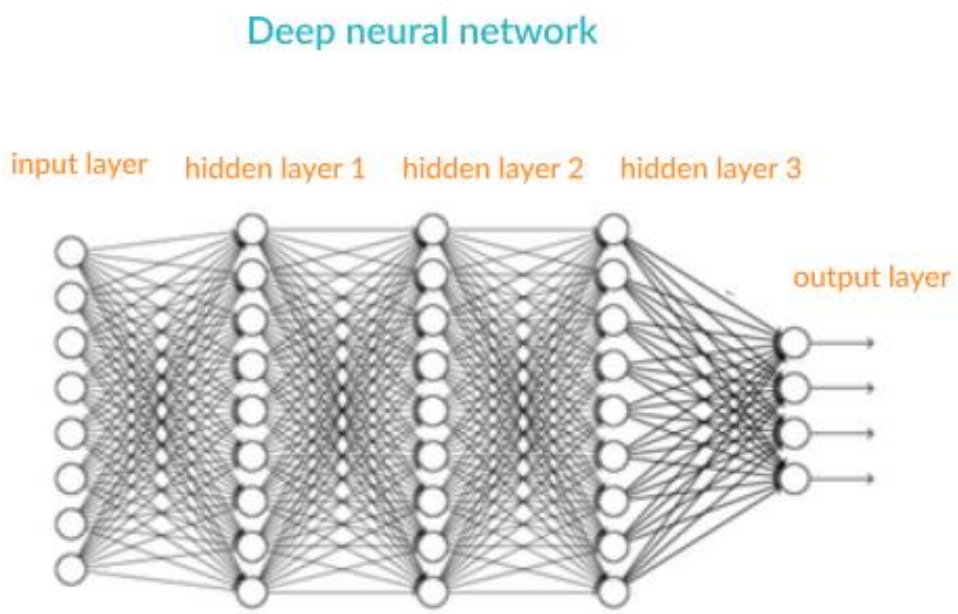
Redes Neuronales para Regresión

Links de interés:

- https://missinglink.ai/guides/neural-network-concepts/neural-networks-regression-part-1-overkill-opportunity/

Las Redes Neuronales Artificiales (ANN) están compuestas por elementos simples, llamados neuronas, cada una de las cuales puede tomar decisiones matemáticas simples. Juntas, las neuronas pueden analizar problemas complejos, emular casi cualquier función incluyendo las más complejas, y proporcionar respuestas precisas. Una red neuronal poco profunda tiene tres capas de neuronas: una capa de entrada, una capa oculta y una capa de salida. Una red neuronal profunda (DNN) tiene más de una capa oculta, lo que aumenta la complejidad del modelo y puede mejorar significativamente el poder de predicción.





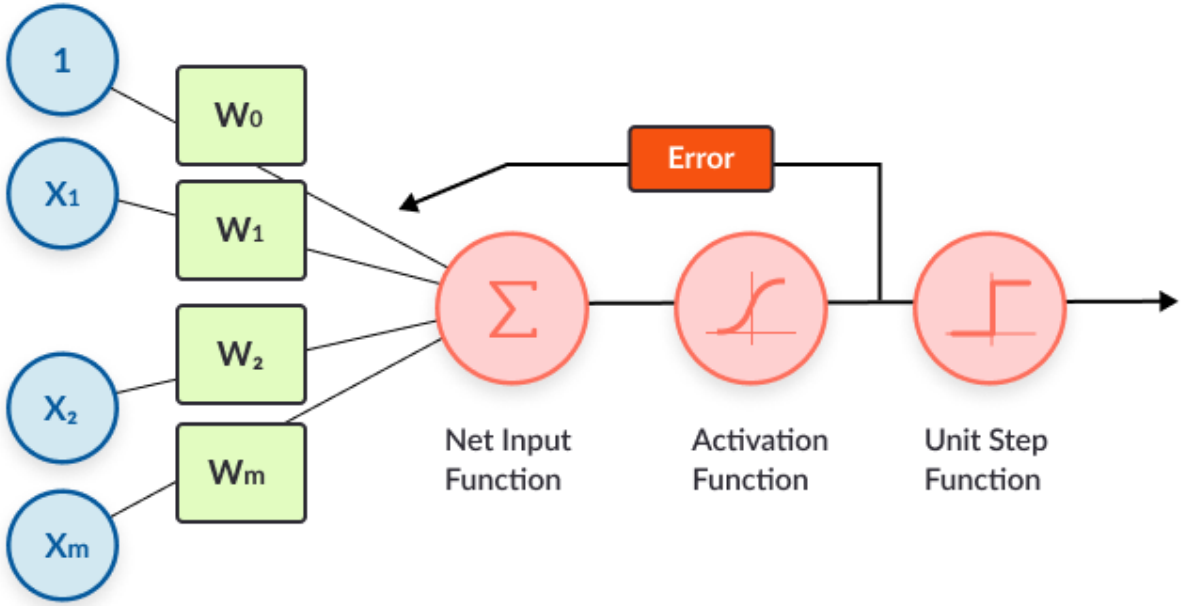
Gidia Grupo de I+D en Inteligencia Artificial

UNIVERSIDAD NACIONAL DE COLOMBIA

Redes Neuronales para Regresión

Las redes neuronales son reducibles a modelos de regresión: una red neuronal puede "fingir" ser cualquier tipo de modelo de regresión. Por ejemplo, esta red neuronal muy simple, con una sola neurona de entrada, una neurona oculta y una neurona de salida, es equivalente a una regresión logística. Toma varias variables dependientes igual al número de parámetros de entrada, las multiplica por sus coeficientes igual a sus pesos, los cuales pasan por una función de activación “sigmoidea” y una función de paso unitario, que se asemeja mucho a la función de regresión logística con su término de error.

La red aprenderá a través del descenso de gradiente **(backpropagation)** para encontrar coeficientes que sean mejores y se ajusten a los datos, hasta llegar a los coeficientes óptimos de regresión (o, en términos de red neuronal, los pesos óptimos para el modelo).



Gidia Grupo de I+D en Inteligencia Artificial

UNIVERSIDAD NACIONAL DE COLOMBIA

Ejemplo de Regresión

Links de interés:

- https://github.com/MGCodesandStats/datasets/blob/master/cars.csv
- https://datascienceplus.com/keras-regression-based-neural-networks/

Se necesita construir un modelo capaz de estimar la cantidad de dinero que un cliente podría gastar en la compra de un vehículo a través de diferentes atributos de la persona.

- Edad
- Género
- Promedio de millas conducidas al día
- Deuda personal
- Ingreso mensual

	age	gender	miles	debt	income	sales
2	28	0	23	0	4099	620
3	26	0	27	0	2677	1792
4	30	1	58	41576	6215	27754
5	26	1	25	43172	7626	28256
6	20	1	17	6979	8071	4438
7	58	1	18	0	1262	2102
8	44	1	17	418	7017	8520
9	39	1	28	0	3282	500
10	44	0	24	48724	9980	22997

Gidia Grupo de I+D en Inteligencia Artificial
UNIVERSIDAD NACIONAL DE COLOMBIA

Ejemplo de Regresión

Luego de realizar el respectivo pre-procesamiento de datos, se presenta el modelo propuesto. En este ejemplo, se muestra un red neuronal artificial poco profunda.

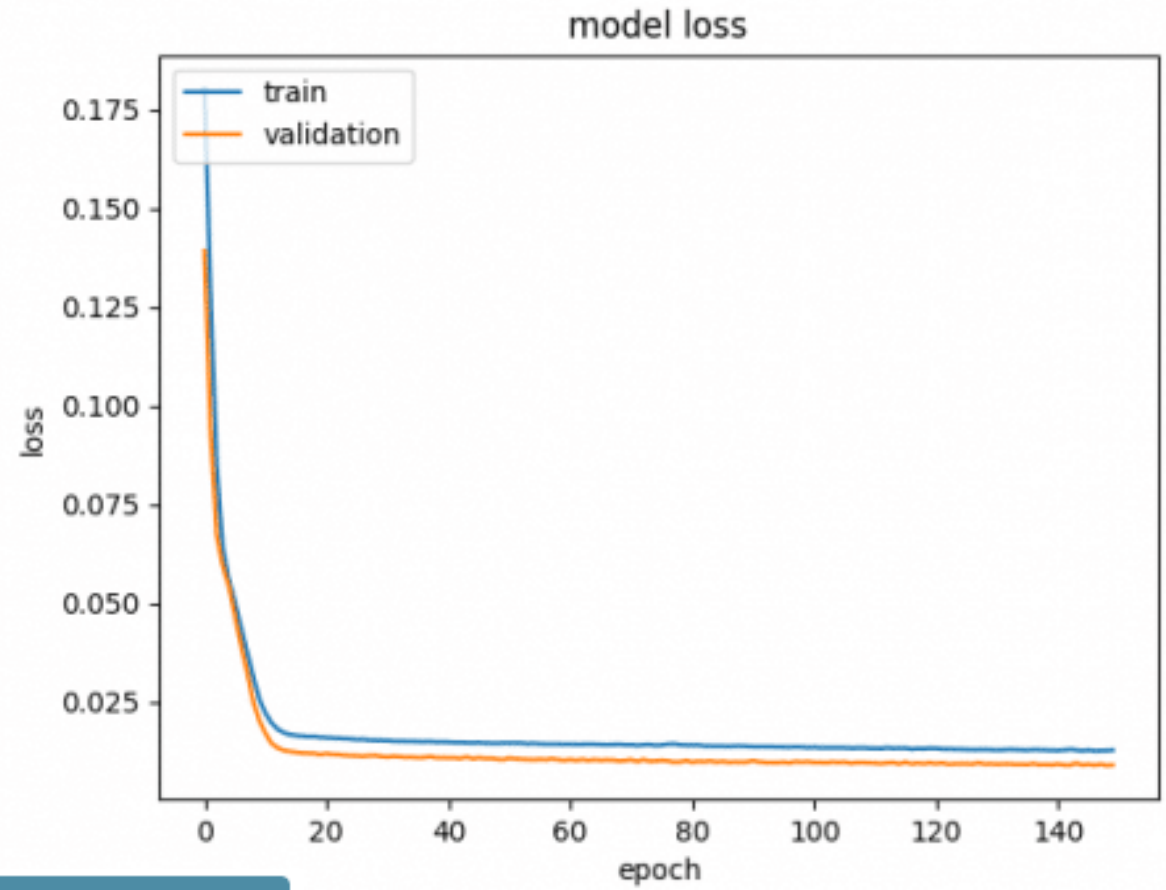
```
model = Sequential()
model.add(Dense(12, input_dim=5, kernel_initializer='normal', activation='re
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='linear'))
model.summary()
```

Para esto, se crea un modelo de 3 capas más 1 de entrada. Su composición es de 12 neuronas en su primera capa, 8 en su segunda capa y 1 en su capa de salida. Nótese que la capas diferentes a la neurona de salida utilizan la función de activación ‘relu’, a diferencia de la ultima capa que maneja una función de activación lineal. Se compila el modelo utilizando el optimizador ADAM y la función de pérdida MSE (Mean Squared Error).

```
model.compile(loss='mse', optimizer='adam', metrics=['mse','mae'])
```

Ejemplo de Regresión

En su gráfica de entrenamiento se puede observar como el modelo fue minimizando la función de pérdida (MSE) elegida anteriormente.



Aquí podemos observar un ejemplo nuevo que nunca ha visto el modelo, junto con su valor estimado

```
Xnew = np.array([[40, 0, 26, 9000, 8000]])
Xnew= scaler_x.transform(Xnew)
ynew= model.predict(Xnew)
#invert normalize
ynew = scaler_y.inverse_transform(ynew)
Xnew = scaler_x.inverse_transform(Xnew)
print("X=%s, Predicted=%s" % (Xnew[0], ynew[0]))
```

```
X=[  40.    0.   26. 9000. 8000.], Predicted=[13686.491]
```


Gidia Grupo de I+D en Inteligencia Artificial
UNIVERSIDAD NACIONAL DE COLOMBIA