

Nombre. – Lopez Chavez Pablo

Materia. – Inteligencia Artificial 1

Regresión. - regresión lineal multivariable, una regresión polinómica y el cálculo por la ecuación de la normal

Links

<https://colab.research.google.com/drive/1UXwaLUE2Zw5S5iXBSPQqCJR4RbxHlpuk?usp=sharing>

<https://drive.google.com/file/d/1QU2N-9ejlfFJv6HMM4ds9ZZ0zmmhNcWAq/view?usp=sharing>

## Regresión lineal multivariable

Del data set usaremos 10 columnas para la regresión, el resto las eliminaremos como también eliminaremos la fila 1

(#0 yearID InnOuts PO A E H ER BAOpp ERA) columnas a usar

Del data set usaremos 10 columnas para la regresión, el resto las eliminaremos como también eliminaremos la fila 1

Seleccionaremos los valores para x como para y

Obtendremos la cantidad de ejemplos imprimir X, y, y m para verificar, Imprimir algunos puntos de datos

creamos una copia del conjunto de datos original para no modificar los datos originale, Inicializa un array de ceros para las medias ( $\mu$ ) de cada columna, Inicializa un array de ceros para las desviaciones estándar ( $\sigma$ ) de cada columna, Calcula la media de cada columna (característica) en el conjunto de datos, Calcula la desviación estándar de cada columna (característica) en el conjunto de datos, Normaliza cada característica restando la media y dividiendo por la desviación estándar, Devuelve el conjunto de datos normalizado, las medias y las desviaciones estándar

llama featureNormalize con los datos cargados

calcula el costo (o error) de un modelo de regresión lineal multivariable dado un conjunto de datos, los parámetros actuales del modelo ( $\theta$ ), y los valores reales de salida ( $y$ ), m son los valores de entrenamiento, J como costo inicia en 0, La fórmula general para la función de costo en la regresión lineal es la que se esta usando en j

descenso por gradiente para un modelo de regresión lineal multivariable (también conocida como regresión lineal múltiple),  $\alpha$  valor de aprendizaje,  $\text{num\_iters}$  número de iteraciones,  $\theta$  vector de parámetro, se crea una copia de  $\theta$  para no modificar la original, calcula el error entre las predicciones y el valor real,  $(\alpha / m) * (\text{np.dot}(X, \theta) - y) \cdot \text{dot}(X)$ : Multiplica el error por la tasa de aprendizaje  $\alpha$  y por la matriz de características  $X$ , escalada por el número de ejemplos  $m$ . Este término ajusta  $\theta$  en la dirección que minimiza el error.

inicializamos  $\theta$  con la cantidad de columnas de  $x$ , llamamos a `gradientedescens`, imprimimos el costo y el descenso de la gradiente, damos valores a  $X\_array$  y al final calculamos el precio

## Regresión polinómica

Del data set usaremos 10 columnas para la regresión, el resto las eliminaremos como también eliminaremos la fila 1

(#0 yearID InnOuts PO A E H ER BAOpp ERA) columnas a usar

Seleccionaremos los valores para  $x$  como para  $y$ , Obtendremos la cantidad de ejemplos imprimir “ $X, y$ ,” y  $m$  para verificar, Imprimir algunos puntos de datos

Imprimir los puntos la grafica con los valores de  $x$  y “ $y$ ”

Multiplicamos  $x$  por  $\text{sig}$  misma y lo unimos al array de  $x$

Realizamos la normalización de  $x$  mediante la desviación estándar, donde  $\mu$  es la media de cada característica,  $\sigma$  la desviación estándar y en  $X\_norm$  se guarda los valores donde se utiliza  $\mu$  y  $\sigma$

Después de `featureNormalize` la función es probada, se añade el termino de intersección a  $X\_norm$ : Se vuelve a graficar pero ahora con  $X\_norm$

Aquí agregamos una columna de 1 a la matriz  $X\_norm$

calcula el costo (o error) de un modelo de regresión lineal multivariable dado un conjunto de datos, los parámetros actuales del modelo ( $\theta$ ), y los valores reales de salida ( $y$ ),  $J$  como costo inicia en 0,  $h$  tiene el valor de calcula el producto punto (o producto matricial) de  $X$  y  $\theta$ . donde  $h$  es la predicción,

descenso por gradiente para un modelo de regresión lineal multivariable (también conocida como regresión lineal múltiple),  $\alpha$  valor de aprendizaje,  $\text{num\_iters}$  número de iteraciones,  $\theta$  vector de parámetro, se crea una copia de  $\theta$  para no modificar la

original, calcula el error entre las predicciones y el valor real,  $(\alpha / m) * (np.dot(X, \theta) - y).dot(X)$ : Multiplica el error por la tasa de aprendizaje  $\alpha$  y por la matriz de características  $X$ , escalada por el número de ejemplos  $m$ . Este término ajusta  $\theta$  en la dirección que minimiza el error.

Estamos usando solo 1000 ejemplos, y  $\alpha$  con 0.01, iteraciones 10000, inicializamos  $\theta$  con la cantidad de columnas de  $x$ , llamamos a `gradientedescens`, imprimimos el costo, damos valores a `X_array` y al final calculamos el precio

## Ecuación de la normal

Sacamos el dataset, eliminamos las columnas que no se usaran, como la fila 0, damos los valores a  $x$  y a " $y$ ", sacamos una copia de  $x$ , agregamos las columnas donde será  $x*x$ , agregamos la columna de sesgo

damos los valores de  $\theta$  con la cantidad de columnas de  $x$ , aplicamos la ecuación de la normal

llamamos a la función, imprimimos los datos, damos valores a `X_array` y multiplicamos por  $\theta$

Imprimimos los valores