

Minería de datos Predicción



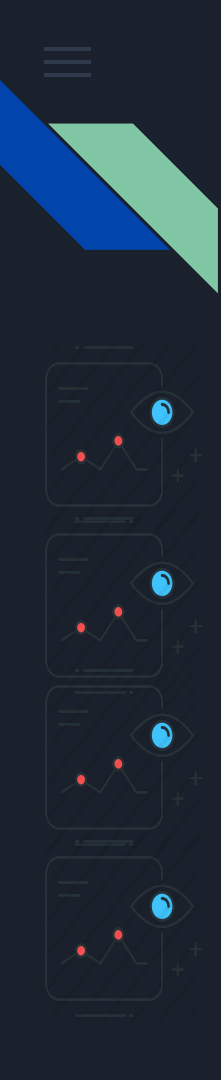
Isidro Garza Villarreal	1818012
José Luis Buendía Meza	1813456
Eliezer Gamaliel Castillo Alcantar	1684521
Mauricio Enrique Espinosa Martínez	1740483
Laura Estefany Rodríguez de los Reyes	1588292

Predicción

La predicción es una técnica que se utiliza para proyectar los tipos de datos que se verán en el futuro o predecir el resultado de un evento.

En muchos casos, el simple hecho de reconocer y comprender las tendencias históricas es suficiente para trazar una predicción algo precisa de lo que sucederá en el futuro.





Existen cuestiones
relativas a la relación
temporal de las
variables de entrada
o predictores de la
variable objetivo

Los valores son
generalmente
continuos

Las predicciones
son a menudo
(no siempre)
sobre el futuro

Variables
independientes



Atributos ya
conocidos

Variables de
respuesta



Lo que queremos
saber



Relación con otras técnicas

Cualquiera de las técnicas utilizadas para la clasificación y la estimación puede ser adaptada para su uso en la predicción mediante el uso de ejemplos de entrenamiento donde el valor de la variable que se predijo que ya es conocido, junto con los datos históricos de esos ejemplos.

Los datos históricos se utilizan para construir un modelo que explica el comportamiento observado en los datos. Cuando este modelo se aplica a nuevas entradas de datos, el resultado es una predicción del comportamiento futuro de los mismos.

Aplicaciones

Revisar los historiales crediticios de los consumidores y las compras pasadas para predecir si serán un riesgo crediticio en el futuro



Predecir el precio de venta de una propiedad



Predecir si va a llover en función de la humedad actual




Predecir la puntuación de cualquier equipo durante un partido de fútbol








Técnicas



La mayoría de las técnicas de predicción se basan en modelos matemáticos:

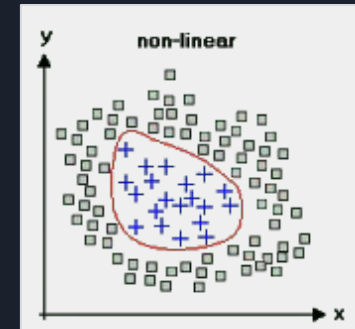
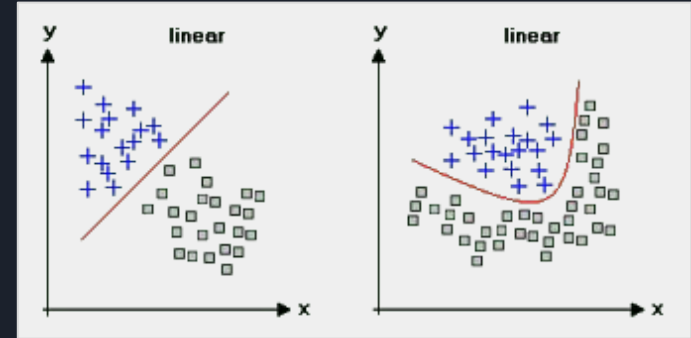
- 
- 
- Modelos estadísticos simples como regresión
 - Estadísticas no lineales como series de potencias
 - Redes neuronales, RBF, etc.



Todo basado en ajustar una curva a través de los datos, es decir, encontrar una relación entre los predictores y los pronosticados.

Tipos de métodos de regresión

- 01 Regresión lineal
- 02 Regresión lineal multivariante
- 03 Regresión no lineal
- 04 Regresión no lineal multivariante



Regresión Lineal

Linear Regression: Single Variable

$$\hat{y} = \beta_0 + \beta_1 x + \epsilon$$

Predicted output Coefficients Input Error

The diagram shows the equation $\hat{y} = \beta_0 + \beta_1 x + \epsilon$. The predicted output \hat{y} is enclosed in a red box with a label 'Predicted output' below it. The coefficients β_0 and β_1 are grouped by a green bracket with a label 'Coefficients' below it. The input x is enclosed in a blue box with a label 'Input' below it. The error term ϵ is enclosed in an orange box with a label 'Error' below it.

El objetivo del Análisis de regresión es determinar una función matemática sencilla que describa el comportamiento de una variable dados los valores de otra u otras variables.

En el Análisis de regresión simple, se pretende estudiar y explicar el comportamiento de una variable que notamos y , y que llamaremos variable dependiente o variable de interés, a partir de otra variable, que notamos x , y que llamamos variable explicativa, variable de predicción o variable independiente.

Regresión Lineal Multivariante

Permite generar un modelo lineal en el que el valor de la variable dependiente o respuesta y , se determina a partir de un conjunto de variables independientes llamadas predictores x_1, x_2, x_3, \dots . Es una extensión de la regresión lineal simple.

The diagram shows the equation $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \varepsilon$ with the following labels and arrows:

- Y : response, dependent variable, observation, 'y-variable' (red arrow)
- β_1 : coefficient (orange arrow)
- x_1 : predictor, 'x-variable', independent variable, explanatory variable (green arrow)
- β_2 : coefficient (orange arrow)
- x_2 : predictor, 'x-variable', independent variable, explanatory variable (green arrow)
- β_p : coefficient (orange arrow)
- x_p : predictor, 'x-variable', independent variable, explanatory variable (green arrow)
- The entire sum $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$ is labeled "linear predictor" (blue bracket)
- ε : random error, "noise" (purple arrow)

Los modelos de regresión múltiple pueden emplearse para predecir el valor de la variable dependiente o para evaluar la influencia que tienen los predictores sobre ella.



Regresión No Lineal univariable y multivariable

Método para encontrar un modelo no lineal para la relación entre la variable dependiente y un conjunto de variables independientes.

La regresión no lineal es una regresión en la que las variables dependientes o de criterio se modelan como una función no lineal de los parámetros del modelo y una o más variables independientes.

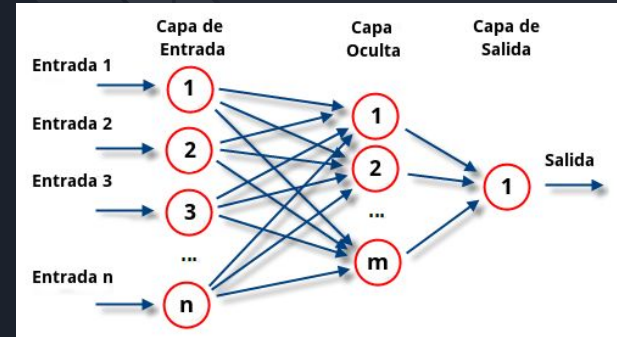
Se denomina regresión no lineal porque las relaciones entre los parámetros dependientes e independientes no son lineales

Redes neuronales

Utiliza los datos para modificar las conexiones ponderadas entre todas sus funciones hasta que sea capaz de predecir los datos con precisión.

Este proceso se conoce como entrenamiento de la red neuronal.

Las redes neuronales consisten generalmente de tres capas: de entrada, oculta y de salida.





Ejercicio

Pronóstico de ventas futuras

Usaremos los datos de los últimos días de noviembre 2018 para calcular las ventas de la primer semana de diciembre.

```
1 ultimosDias = df['2018-11-16':'2018-11-30']  
2 ultimosDias
```

fecha

2018-11-16 152

2018-11-17 111

2018-11-19 207

2018-11-20 206

2018-11-21 183

2018-11-22 200

2018-11-23 187

2018-11-24 189

2018-11-25 76

2018-11-26 276

2018-11-27 220

2018-11-28 183

2018-11-29 251


2018-11-30 189

Name: unidades, dtype: int64

Y ahora seguiremos el preprocesado de datos: escalando los valores, llamando a la función *series_to_supervised* sin incluir la columna de salida “Y” pues es la que queremos hallar. Por eso, verán en el código que hacemos `drop()` de la última columna.

```
1 values = ultimosDias.values
2 values = values.astype('float32')
3 # normalize features
4 values=values.reshape(-1, 1) # esto lo hacemos porque tenemos 1 sola dimension
5 scaled = scaler.fit_transform(values)
6 reframed = series_to_supervised(scaled, PASOS, 1)
7 reframed.drop(reframed.columns[[-7]], axis=1, inplace=True)
8 reframed.head(7)
```


	var1(t-7)	var1(t-6)	var1(t-5)	var1(t-4)	var1(t-3)	var1(t-2)	var1(t-1)
7	-0.24	-0.65	0.31	0.30	0.07	0.24	0.11
8	-0.65	0.31	0.30	0.07	0.24	0.11	0.13
9	0.31	0.30	0.07	0.24	0.11	0.13	-1.00
10	0.30	0.07	0.24	0.11	0.13	-1.00	1.00
11	0.07	0.24	0.11	0.13	-1.00	1.00	0.44
12	0.24	0.11	0.13	-1.00	1.00	0.44	0.07
13	0.11	0.13	-1.00	1.00	0.44	0.07	0.75



De este conjunto “ultimosDias” tomamos sólo la última fila, pues es la que correspondería a la última semana de noviembre y la dejamos en el formato correcto para la red neuronal con *reshape*:


```
1 values = reframed.values
2 x_test = values[6:, :]
3 x_test = x_test.reshape((x_test.shape[0], 1, x_test.shape[1]))
4 x_test
```

```
array([[[[ 0.11000001, 0.13 , -1. , 1. ,
0.44000006, 0.06999993, 0.75 ]]], dtype=float32)
```



Ahora crearemos una función para ir “rellenando” el desplazamiento que hacemos por cada predicción. Esto es porque queremos predecir los 7 primeros días de diciembre. Entonces para el 1 de diciembre, ya tenemos el set con los últimos 7 días de noviembre. Pero para pronosticar el **2 de diciembre** necesitamos los 7 días anteriores que **INCLUYEN** al 1 de diciembre y ese valor, lo obtenemos en nuestra predicción anterior. Y así hasta el 7 de diciembre.

```
1 def agregarNuevoValor(x_test,nuevoValor):
2     for i in range(x_test.shape[2]-1):
3         x_test[0][0][i] = x_test[0][0][i+1]
4     x_test[0][0][x_test.shape[2]-1]=nuevoValor
5     return x_test
6
7 results=[]
8 for i in range(7):
9     parcial=model.predict(x_test)
10    results.append(parcial[0])
11    print(x_test)
12    x_test=agregarNuevoValor(x_test,parcial[0])
```



Las predicciones están en el **dominio del -1 al 1** y nosotros lo queremos en nuestra escala “real” de unidades vendidas. Entonces vamos a “re-transformar” los datos con el objeto “scaler” que creamos antes.

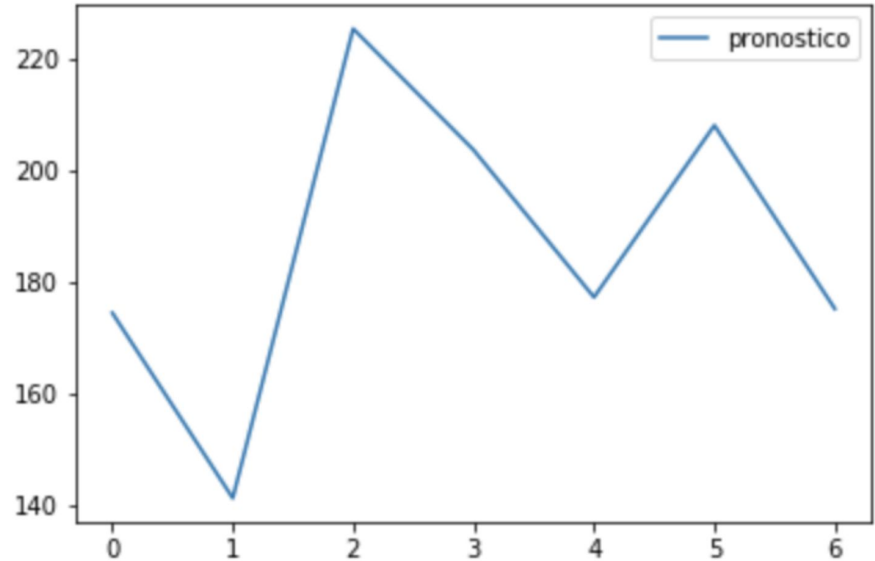
```
1 adimen = [x for x in results]
2 inverted = scaler.inverse_transform(adimen)
3 inverted
```

```
array([[174.48904094],
       [141.26934129],
       [225.49292353],
       [203.73262324],
       [177.30941712],
       [208.1552254 ],
       [175.23698644]])
```


Ya podemos crear un nuevo DataFrame Pandas por si quisiéramos guardar un nuevo csv con el pronóstico. Y lo visualizamos.

A partir de los últimos 7 días de noviembre 2018 y utilizando nuestra red neuronal, hicimos el siguiente pronóstico de venta de unidades para la primer semana de diciembre.

```
1 prediccion1SemanaDiciembre = pd.DataFrame(inverted)
2 prediccion1SemanaDiciembre.columns = ['pronostico']
3 prediccion1SemanaDiciembre.plot()
4 prediccion1SemanaDiciembre.to_csv('pronostico.csv')
```





Bibliografía

- <https://www.datasciencecentral.com/profiles/blogs/the-7-most-important-data-mining-techniques>
- <https://www.statisticssolutions.com/regression-analysis-nonlinear-regression/>
- https://www.cienciadedatos.net/documentos/25_regresion_lineal_multiple
- <http://www.cs.stir.ac.uk/courses/ITNP60/lectures/1%20Data%20Mining/4%20-%20Prediction.pdf>
- DATA MINING TECHNIQUES AND APPLICATIONS, Mrs. Bharati and M. Ramageri
- <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/1339/006312G216.pdf;jsessionid=9A3015C0766F561076782CDA1995E145?sequence=1>
- <https://www.aprendemachinelearning.com/pronostico-de-series-temporales-con-redes-neuronales-en-python/>



Gracias

