
SKILLING
CENTER
TECMILENIO

SKILLING CENTER TECMILENIO

Inteligencia artificial y machine learning

Modelos de predicción con regresión

Introducción

Las instituciones financieras requieren tomar decisiones asertivas, por lo que es común analizar bases de datos detallando la relación que existe entre la edad del solicitante de un crédito y el factor riesgo. Para generar esta relación es posible utilizar un modelo de regresión simple tomando en cuenta que sólo requiere estas dos variables y una puede ser clasificada como variable dependiente (riesgo) y la otra independiente (edad).

Introducción

Los modelos de regresión son algunos de los algoritmos de aprendizaje automático más comunes en la industria. Los tipos de modelos dentro de este paradigma pueden ser simples, tales como la regresión lineal, o altamente complejos, como una arquitectura de redes neuronales. En algunos casos, se puede trabajar con estos modelos ignorando su complejidad y enfocándose directamente en el problema que resuelven. En el caso de modelos de predicción con regresión, se predice una variable numérica generalmente definida en el conjunto de números reales. Esto permite atender un conjunto amplio de problemas, desde la predicción de precios de acciones hasta el periodo de retención de un cliente usando un servicio empresarial.

En esta experiencia educativa, conocerás sobre los conceptos principales de los modelos de regresión, la aplicación de cada uno de ellos y lo más importante, un caso real de uso dentro de las finanzas.

Explicación

Cuando se tiene un conjunto de datos en un problema de aprendizaje automático, generalmente interesa entender y pronosticar el comportamiento de una variable objetivo. De forma estadística, es posible tomar los distintos valores de la variable objetivo y analizar su distribución.

Un modelo de regresión permite representar una variable objetivo-numérica, o variable dependiente, en función de un conjunto de variables independientes.

De acuerdo con Ferre (2019), los modelos de regresión permiten modelar de forma más precisa la variable objetivo al incorporar información adicional mediante las variables independientes para explicar el comportamiento de la variable dependiente.

Explicación

La implementación de los modelos de regresión depende específicamente del tipo de modelo (lineal o no-lineal) que se esté utilizando. Conociendo el modelo en particular, se puede diseñar un algoritmo de entrenamiento que se ajuste a las especificaciones del modelo. De forma general, es común utilizar un algoritmo de entrenamiento con base en la metodología del gradiente descendente, el cual detalla el mínimo valor que puede obtener una función. En este sentido, los pasos para diseñar un modelo de regresión son los siguientes:

Explicación

Los modelos de regresión son altamente utilizados en el ámbito financiero. En principio, la capacidad para representar el comportamiento de una variable numérica en función de otro conjunto de variables, lo vuelve una herramienta muy atractiva en la industria.

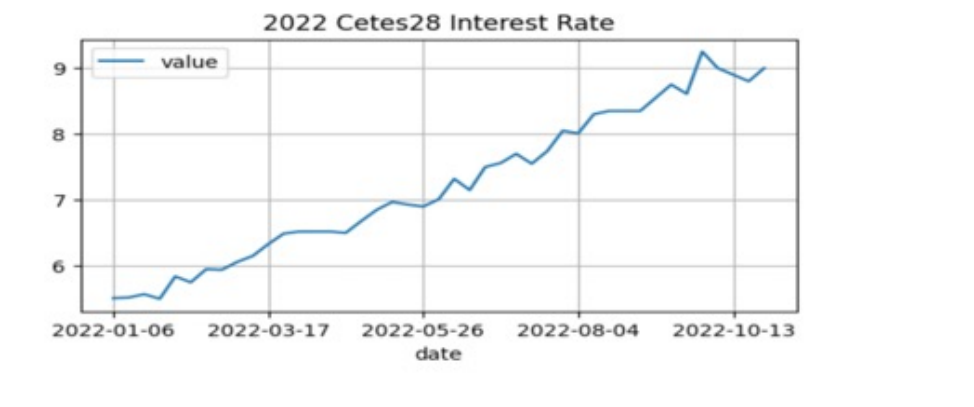
De acuerdo con Sidelov (2021) y Usachev (s.f.), dentro de los casos de uso de modelos de regresión más comunes en la industria financiera se encuentran los siguientes:

- **Predicciones de variables financieras y series de tiempo.**
- **Modelos de riesgos crediticios.**
- **Valuación de activos y administración de portafolios financieros.**

Explicación

Ejemplo práctico

Ana está ahorrando para solicitar un crédito hipotecario y comprar un departamento en los siguientes meses. Le preocupa un aumento potencial de la tasa de interés. El siguiente ejemplo muestra un modelo autoregresivo para estimar los cambios de la tasa de interés de referencia (CETES a 28 días).



Para el modelo de regresión de Ana, la variable objetivo será la tasa de interés y las variables independientes, en este caso solamente 1 será el valor inmediato anterior en la serie de tiempo, también conocido como rezago. El siguiente código permite calcular los rezagos, atendiendo hasta el caso general de “n” rezagos.

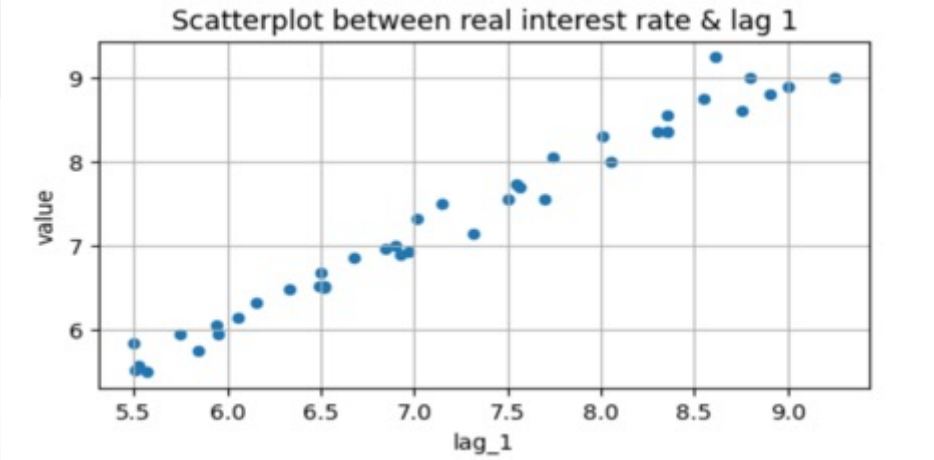
```
Def data_processing(df:pd.DataFrame, target_col:str,date_col:str,nlags:int = 1):
   dtaser = df.copy().sort_values(by=date_col, axis=0, accending=False)
   for i in range(1, nlags + 1):
      dataset[f”lag”{i}] = df[target_col].shift(periods=i=
   return dataser.reset_index(drop=true)[:nlags]
Dataser = dta_processing(df=data, target_col= “value”, date_col =“date”
Dataser.head ()
```

Explicación

	Date	Value	Lag_1
0	2022-10-27	9.00	8.80
1	2022-10-20	8.80	8.90
2	2022-10-13	8.90	9.00
3	2022-10-16	9.00	9.25
4	2022-09-29	9.25	8.61

Al usar un gráfico de dispersión, es evidente que existe una relación entre los valores anteriores de la serie de tiempo y los valores actuales. En el caso general, podrías utilizar más de un rezago y determinar la contribución de cada rezago para el valor actual. Por el momento, los modelos simples utilizarán solamente un rezago.

Explicación



Ana propone utilizar un modelo lineal, lo cual implica definir n+1 coeficientes para “n” variables. En este caso, la variable independiente es el rezago inmediato anterior, por lo cual solamente tendrá dos coeficientes. Así mismo, puede definir la función de error como la diferencia al cuadrado de la estimación y el valor real de la tasa de interés.

```
def get_linear_model {beta_0: float, beta_1: float}:
   def closure (x: float) => float:
     return beta_0 + beta_1*x
  return closure
def get_error_funtiona(beta_0: float, beta_1: foat):
    linear_model = get_linear_model (beta_0, beta_1)
    def closure(df: pd.DataFrame)  => float:
       retur sum((row[“y”]  =  linear_model(x=row)[“x”]))**2 for_, row in
df.iterrows() )
     return closure
```

Explicación

Utilizando las ecuaciones de las derivadas parciales, puedes implementar las siguientes funciones:

```
def  partial_beta_0(beta0: float, beta_1: float);
     linear_model = get_linear_model (beta_0, beta_1)
     def closure(df:pd.DataFrame):
      return -2 *sum((row[“y”] * linear_model(x=row[“x”]) for_, row in df.interrows())
      return closure
def partial_beta_1(beta_0:float,beta_1: float):
    linear_model = get_linear_model((beta_0,beta_1)
    def closured (df.DataFrame):
      return -2 * sum((row[“y”] * linear_model(x=row[“x”])) * row[“x”] for_, row in df.iterrows()
)
```

Con esta implementación, Ana está lista para entrenar su modelo de regresión con los datos que obtuvo de Banxico respecto a la tasa de interés Cetes28. Para comenzar, se toma una muestra aleatoria de los datos que se usarán como conjunto de entrenamiento. En esta ocasión, se toma el 75 % de los datos.

Explicación

trainign_dataset = dataser.rename(columns= {“value”; “y”, “lag_1”: “x”}).sample(frac=0.75)
training_sataser.head()

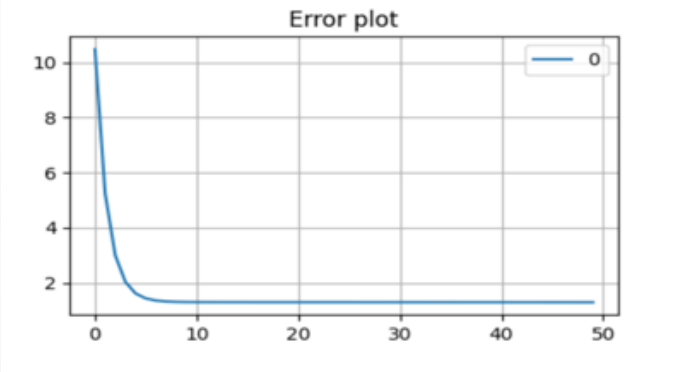
	Date	Y	X
0	2022-10-06	9.00	9.25
1	2022-08-25	8.85	8.35
2	2022-03-17	6.33	6.15
3	2022-07-14	7.55	7.70
4	2022-06-23	7.50	7.15

Explicación

Ana procede a implementar el algoritmo de gradiente descendente. Puntos clave a considerar:

1. Se inicializan los coeficientes del modelo con un valor aleatorio entre 0-1.
2. La constante de aprendizaje, *alpha,* permite controlar la magnitud de los pasos dentro del gradiente. Es recomendable usar un valor relativamente bajo para seguir al gradiente de forma cercana.
3. El número de iteraciones representa la cantidad de actualizaciones que se harán sobre los parámetros del modelo.

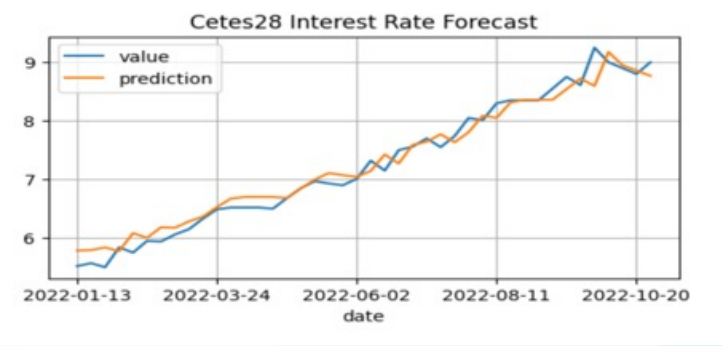
```
beta_0 = np.random.rand()
beta_1 = np.random.rand()
alpha = 0.0001
error = []
for _ in range (50):
    beta_0 = beta_0 – Alpha * partial_beta_0(beta_0, beta_1)(training_dataset)
    beta_1 = beta_1 – Alpha * partial_beta_1 (beta_0,beta_1)(training_dataset)
    errors.append(get_error_function (beta_0, beta_1) (training-sataset))
pd.DataFRAME(error).plot(grid=True, title= “Error plot”, figsize= (5, 3))
<AxesSubplot: tittle={´center´: Érror plot¨}>
```



Explicación

1. Como puede observarse, el modelo minimiza el valor de la función de error con cada iteración de entrenamiento. Este tipo de gráficas suelen indicar un buen entrenamiento cuando adquieren una forma tradicional de “codo”.
2. Los parámetros resultantes del entrenamiento se pueden utilizar directamente en el modelo lineal para generar predicciones de la tasa de interés. En este caso, la estimación de la tasa de interés sigue la tendencia de los valores originales. El modelo de Ana predice una ligera reducción de la tasa de interés para el siguiente periodo.

```
linear_model = get_linear_model(beta_0,beta_1)
predictions = dataser.assign(
  prediction=lamda df: np.vectorize(linear:model) (df.lag_1)
)[[“date”, “value”, “prediction”]]
predictions.sort:values(by=“date”, axis=0, ascending=True).plot(
  x= “date”
  y = [“value”, “prediction”],
  title= “”Cetes28 Interest Rate Forecast”,
  grid= True
  figsize=(6,3)
)
current_interest_rate =
dataset.ser_index(“dste”).T.to_dict()[dataset.date.max()][“value”]
interest_rate_forecast = linear_model(current_interest_rate)
print(“Current_interest rate:”, round(current_interest_rate, 4))
print(“Next period interest rate forecast:”, round(interés_rate_forecast,
4))
current interest rate: 9.0
next period interest rate forecast: 8.9487
```



Explicación

¿Cómo Ana puede mejorar su modelo predictivo de regresión? En principio se puede mejorar el rendimiento de este modelo en particular agregando más variables independientes, en este caso, más rezagos, y trabajando directamente con las diferencias en la tasa de interés en lugar de valores absolutos.

De forma general, los modelos suelen beneficiarse de algunos factores:

1. Un volumen amplio de muestras (observaciones).
2. Incorporación de variables independientes con contribuciones relevantes.
3. Uso de un modelo suficientemente flexible para representar los efectos esperados en la variable de respuesta.

Actividad

Un modelo de regresión lineal para el análisis de datos es una herramienta muy utilizada en la toma de decisiones.

Como analista financiero buscas crear una regresión lineal a fin de facilitar la toma de decisiones. Para iniciar la creación de tu modelo, requieres información precisa. Por ello, debes indagar:

1. ¿Cuáles serían los mejores modelos de regresión lineal para tomar una buena decisión financiera? Menciona al menos tres.
2. ¿Cuál sería un número óptimo de variables, a fin de generar el cruce de información correspondiente?
3. ¿Qué herramientas de *machine learning* puede apoyar en su construcción?
4. ¿De qué forma se puede evaluar la efectividad del modelo?

Cierre

Para construir un modelo de regresión simple es importante identificar la siguiente información:

1. El objetivo principal de un modelo de regresión es conocer la relación entre una y más variables.

2. Las opciones para estimar un modelo de regresión, de entre los que destaca, la facilidad de aplicación e interpretación.

3. Obtener la interpretación del resultado de acuerdo con la magnitud del cambio de la variable dependiente.

Bibliografía

- Ferre, M. (2019). *FEIR 40: modelos de regresión*. Recuperado de https://gauss.inf.um.es/feir/40/
- Sidelov, P. (2021). *Machine learning in banking: top use cases*. Recuperado de https://sdk.finance/top-machine-learning-use-cases-in-banking
- Usachev, D. (s.f.). *10 use cases of machine learning for finance*. Recuperado de https://fayrix.com/blog/machine-learning-in-finance#use-cases


SKILLING
CENTER
TECMILENIO

SKILLING CENTER TECMILENIO

Inteligencia artificial y machine learning

Modelos de predicción con clasificación

Introducción

Dentro de las decisiones que las instituciones financieras deben tomar se encuentra aceptar o rechazar algún crédito. Por lo general se realiza una consulta en diferentes bases de datos, principalmente el buró de crédito, el cual permite conocer el historial crediticio de un candidato. Sin embargo, esto no basta para tomar una decisión, pues es relevante conocer la capacidad de pago. Por ello, es necesario una solicitud de información que pueda ayudar a la institución a categorizar al cliente, es decir, detallar el nivel de riesgo que para ella representa. Así mismo, considerando el último punto (riesgo), es necesario establecer la tasa de interés y el plazo para dicha solicitud (crédito).

Todo esto puede parecer sencillo, teniendo la información adecuada, sin embargo, resulta complejo si se considera que los datos recabados son proyecciones del futuro, que pudieran verse afectadas por diversas situaciones micro y macroeconómicas.

En esta experiencia educativa conocerás acerca de los modelos de predicción por clasificación, su aplicación y, lo más interesante, su ejecución en la práctica.

Explicación

Los modelos de regresión tradicional, específicamente las regresiones lineales, se enfocan en representar una variable objetivo-numérica como una combinación lineal de un conjunto de variables independientes. Wasserman (2010), explica cómo el modelo de regresión realmente permite representar el valor esperado de una distribución de probabilidad normal, usando una función que opera sobre las variables independientes. En este sentido, es posible comenzar a indagar sobre la generalización de este tipo de modelos, considerando los distintos elementos que pueden ser generalizables, tales como el tipo de distribución que representa los datos.

En el caso de una regresión lineal tradicional, tiene sentido utilizar una distribución normal dado que la variable objetivo es continua y definida sobre el conjunto de números reales entre *-inf* e *inf*. Sin embargo, se puede pensar en una variable objetivo cuya especificación requiere usar una definición diferente. Por ejemplo, determinar el número de casos en una pandemia (variable definida por números enteros positivos) o pronosticar la supervivencia de una enfermedad (variable binaria, definida por los valores exactos de 0 y 1) requieren utilizar una distribución diferente.

Explicación

El Instituto de Tecnología de Massachusetts explica, en el OpenCourseWare (2017), cómo tomar el concepto de una regresión lineal y generalizar sobre la distribución objetivo para resolver problemas de distinta naturaleza, pero manteniendo ciertas condiciones de un problema de regresión lineal tradicional. Este tipo de modelos se conocen como modelos lineales generalizados, o GLM (por sus siglas en inglés) y permiten agregar más flexibilidad al concepto de regresión lineal tradicional mediante la apertura de utilizar diferentes distribuciones de probabilidad.

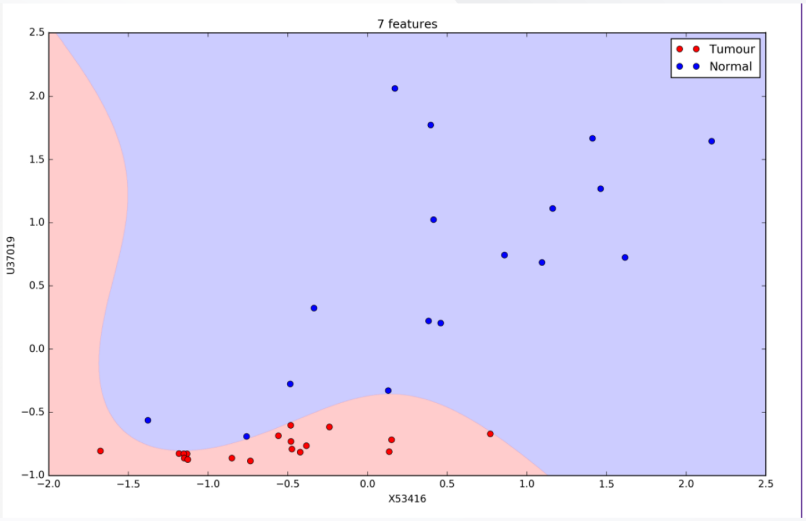
Los *modelos lineales generalizados*, particularmente para el caso donde la distribución de la variable objetivo es la distribución de Bernoulli, representan el caso más simple de los modelos de clasificación. Esta “simpleza” se deriva de que este modelo tiene una base lineal (combinación lineal de las variables independientes) para mostrar la variable objetivo, la cual representa un caso binario (0 ó 1). Este modelo se conoce como *la regresión logística*.

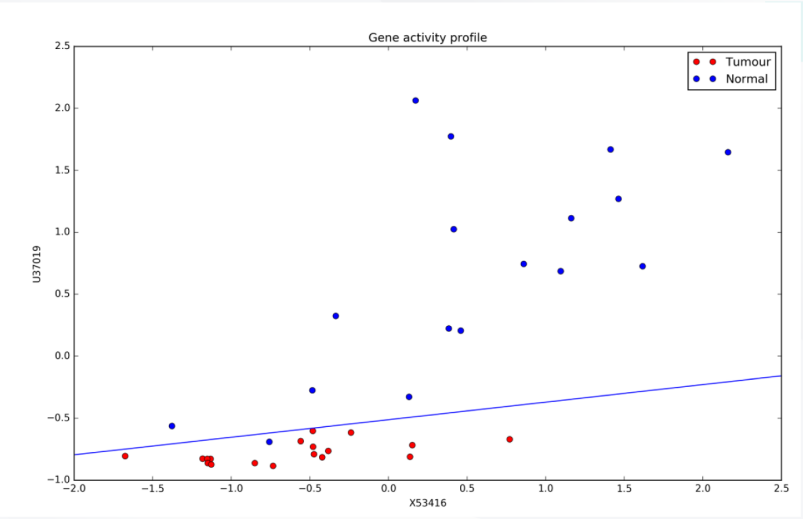
Explicación

Modelos más complejos, como redes neuronales, modelos de árboles de decisión, entre otros, se basan en un conjunto de transformaciones no-lineales para describir la variable objetivo. Sin embargo, ambos tipos de modelos (lineal y no-lineal) en el contexto de clasificación comparten el objetivo de describir una variable independiente con naturaleza categórica. El caso más común de esta variable categórica generalmente se basa en una clasificación binaria, sin embargo, existen casos de predicción multicategoría.

Explicación

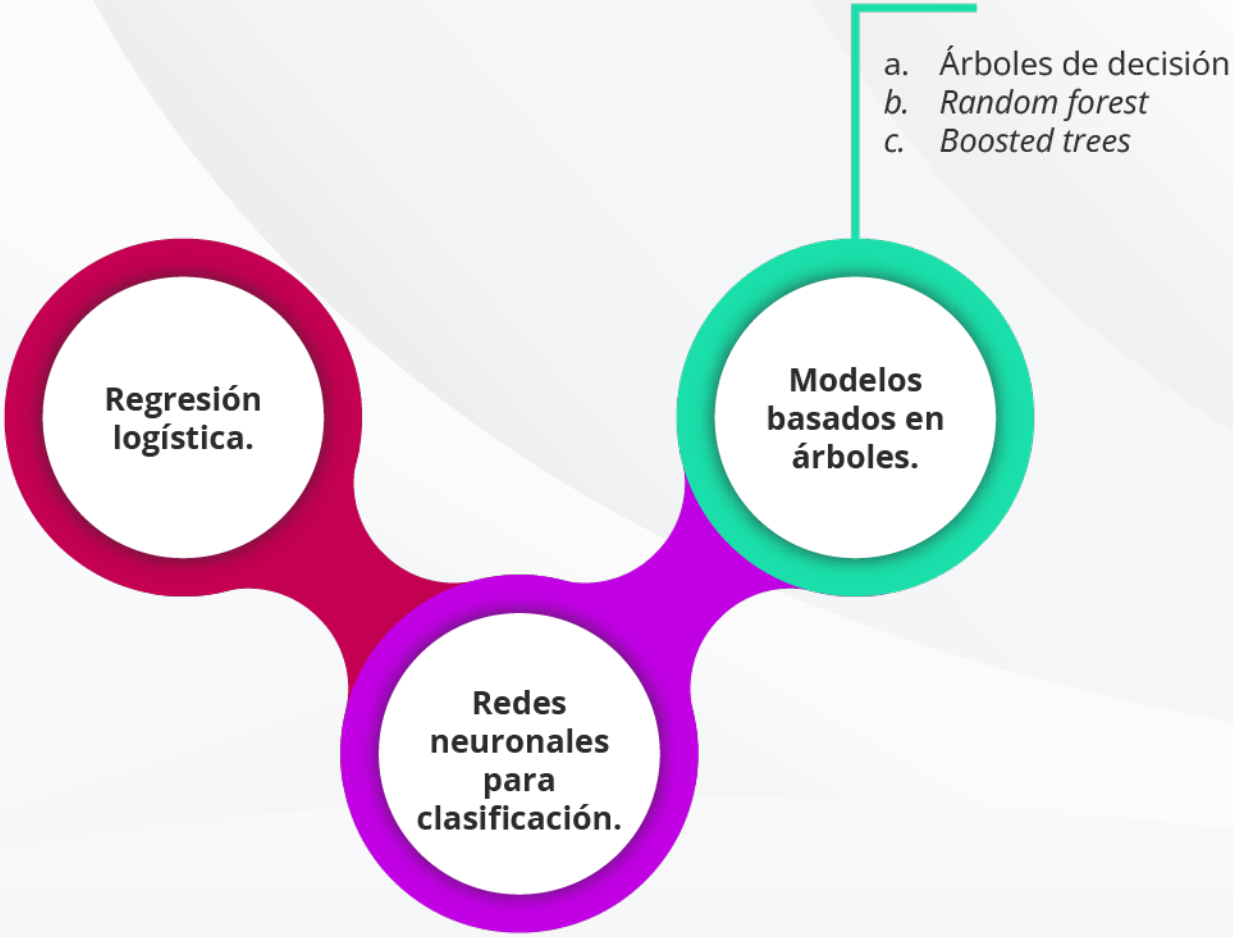
Al aplicar un modelo de clasificación lo que se intenta crear es una frontera de decisión para discernir entre las categorías de la variable de dependiente usando como insumo los valores de la variable independiente. Nuevamente, esta frontera de decisión puede tener una forma lineal (línea recta o hiperplano) o forma irregular derivado del uso de modelos no-lineales, como se muestra en las siguientes imágenes:





Explicación

Entre los modelos de clasificación más populares se tienen:



Explicación

Independientemente del modelo que se seleccione, todos reducen las predicciones a una respuesta binaria, generándose cuatro posibles casos.

Caso 1. Predicción positiva, valor real positivo (TP - *true positive*).
El modelo predice la clasificación positiva de forma correcta.

Caso 2. Predicción negativa, valor real negativo (TN - *true negative*).
El modelo predice la clasificación negativa de forma correcta.

Caso 3. Predicción positiva, valor real negativo (FP - *false positive*).
El modelo predice la clasificación positiva de forma incorrecta.

Caso 4. Predicción negativa, valor real positivo (FN - *false negative*).
El modelo predice la clasificación negativa de forma incorrecta.

Estos cuatro casos conforman los elementos de la matriz de confusión.

Actividad

1- Resuelve el siguiente análisis considerando la información disponible en la siguiente liga https://gist.github.com/RHDZMOTA/406bb08e1a33469eceb66e5d6bf78e27/

La descarga de la información se realiza programáticamente con *Python* utilizando el siguiente código:

```
import os
import pandas as pd

url _base = (
     “https://gist.github.com/RHDZMOTA/”
     “406bb08e1a33469eceb66e5d6bf78e27/raw/a5f5d03c731df8a08e0574b150fe877f0abcc564/”
)
dataset_names = [
    “annual – income – category – low.json”,
    “annual – income – category -medium.json”,
    “annual – income – category – hight.json”,
]
dataset = pd.contact ( [pd.read_json(f”{url_base}/{name}”). T for name in dataset_names]) \
.sample(frac=1 , random_state=888). reset_index (drop=true)

print(dataset.shape)
dataset.head()
(222416, 6)
```

El enlace es externo a la Universidad Tecmilenio, al acceder a él considera que debes apegarte a sus términos y condiciones.

Actividad

1. Realiza un análisis exploratorio de los datos anteriores sobre el conjunto de datos completo.
2. Particiona el conjunto de datos en subconjuntos de entrenamiento (70 % *train*) y prueba (30 % *test*) y contesta: ¿Por qué es importante dividir los datos en estos dos conjuntos?

Cierre

Los modelos de regresión lineal relacionan una variable con otra, por ejemplo, edad y salario.

Un modelo de regresión lineal generalizado analiza la relación entre diferentes variables, pudiendo ser dependientes o independientes, por ejemplo, edad, salario, historial crediticio, etcétera.

Un modelo de proyección por clasificación, permite no solo relacionar diferentes variables, sino crear una proyección de resultados bajo las consideraciones establecidas (algoritmos), por ejemplo, nivel de riesgo.

Bibliografía

- MIT OpenCourseWare. (2017, 17 de agosto). *21. Generalized Linear Models* [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=X-ix97pw0xY
- Wasserman, L. (2010). *All of statistics - a concise course in statistical inference*. Springer.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, estas se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.
