Aprendizaje Automático Relacional

Canales Twitch Afiliados

Guillermo Rodríguez Carrillo
dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
guirodcar@alum.us.es

Álvaro Vázquez Conejo
dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
alvvazcon@alum.us.es

***Resumen*— Este documento aborda un problema de clasificación relacional, centrándose en la creación de una función capaz de clasificar correctamente nodos de una red en diferentes clases, basándose en sus atributos. Específicamente, el problema se relaciona con determiner los canales de twitch dados son afiliados a la plataforma.**

Para ello utilizaremos librerías como sklearn, networkx o pyTorch

I. Introducción

II. Las plataformas de streaming han transformado la manera en que disfrutamos del contenido audiovisual en la actualidad. A través de servicios como Netflix, Amazon Prime Video y Disney+, podemos acceder a una amplia variedad de películas, series y programas en cualquier momento y lugar. Estas plataformas ofrecen comodidad, personalización y flexibilidad, permitiéndonos elegir qué ver y cuándo hacerlo. Además, han dado paso a la producción de contenido original de alta calidad, atrayendo a creadores y talentos de renombre. En consecuencia, las plataformas de streaming se han vuelto indispensables en el entretenimiento contemporáneo, cambiando la forma en que consumimos y disfrutamos de nuestras películas y series favoritas.

En nuestro caso vamos a comprobar cuantos creadores de contenido de la plataforma Twitch están afiliados a dicha plataforma y para ello nos vamos a centrar en los siguientes atributos

- Views
- Para todos los públicos
- Tiempo en directo
- Fecha creación
- Fecha de edición
- Id del canal
- Cuenta activa
- Idioma
- Afiliados
- Follows (Relaciones)

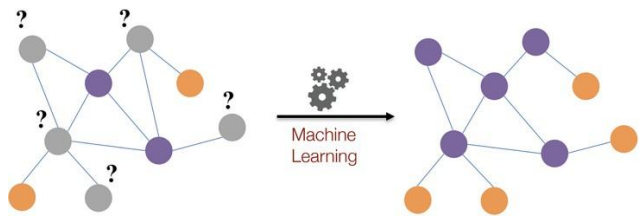
El objetivo de este estudio es determinar qué características son relevantes y qué nivel de relación con otros canales de la plataforma implican que un canal sea afiliado
Para lograrlo, se dispone de dos archivos en formato CSV: uno que contiene los atributos de cada nodo y otro las relaciones entre dichos canales mediante el id del canal. Estos archivos proporcionarán los datos necesarios para realizar el análisis y la predicción en base a los patrones identificados.

II Preliminares

A. Métodos empleados

- GCN: GCN (Graph Convolutional Network) es una técnica que se utiliza para analizar y comprender datos que tienen una estructura de relaciones, como una red social o un mapa de conexiones. Ayuda a identificar patrones y relaciones entre los elementos del conjunto de

datos. Las GCN se basan en un enfoque que permite aprender de forma automática las conexiones y las influencias entre los elementos, lo que ayuda a tomar decisiones más precisas y contextuales.

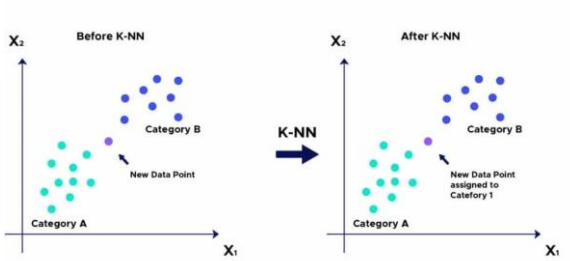


- Naive Bayes: Naive Bayes es un algoritmo de aprendizaje automático que se utiliza para clasificar datos. Se basa en el teorema de Bayes y asume la independencia condicional entre las características. En nuestro trabajo con los canales de Twitch, podemos aplicar Naive Bayes para clasificar si un canal está vinculado o no. Usando características como vistas, cobertura de toda la audiencia y cuándo se creó un canal, podemos calcular la probabilidad de que se vincule un canal. Este enfoque nos permitirá analizar y predecir la asociación de nuevos canales en función de estas características.

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

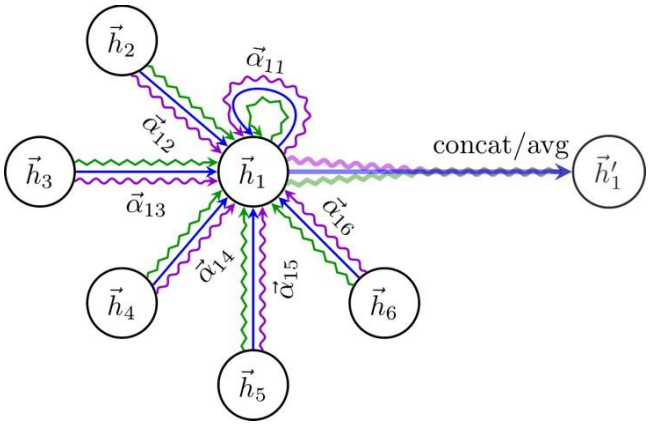
- Matriz de adyacencia: La matriz cuadrada representa la siguiente relación entre canales en Twitch. Cada celda de la matriz contiene valores que indican la presencia o ausencia de seguimiento entre los dos canales. Si el valor de una celda es 1, significa que el canal correspondiente sigue al canal representado por esta columna. Por otro lado, si el valor es 0, indica que no existe una relación de seguimiento entre los canales en cuestión. La matriz de adyacencia nos brinda información sobre cómo se conectan los canales y quién sigue a quién en la plataforma Twitch.

- kNN: (k-Nearest Neighbors) es un algoritmo de aprendizaje supervisado que se utiliza para clasificar y predecir nuevos ejemplos en función de su similitud con ejemplos previos. No requiere una etapa de entrenamiento explícita, ya que se basa en la información proporcionada por los vecinos más cercanos en el conjunto de datos. El método kNN se utiliza para problemas de clasificación y regresión, y su objetivo principal es asignar una etiqueta o valor a un nuevo ejemplo basado en las características de los ejemplos cercanos. Es especialmente útil cuando no se conocen las distribuciones subyacentes de los datos y se requiere una aproximación basada en la similitud.



- GAT: Graph Attention Network (GAT) es un método de aprendizaje automático que se utiliza para analizar gráficos y redes. A diferencia de los métodos de convolución de gráficos tradicionales, GAT asigna pesos de atención a los vecinos de un nodo en función de la importancia relativa de su información para el nodo en cuestión. Esto permite que los nodos se centren en la información más relevante de los nodos vecinos durante el proceso de aprendizaje. GAT utiliza mecanismos de atención basados en el aprendizaje automático para determinar estos pesos de atención y luego combina la información del vecindario ponderado de acuerdo con estos pesos para obtener una representación mejorada del nodo raíz. De esta forma, el GAT puede capturar relaciones no

lineales y modelar de manera eficiente la estructura y las interacciones en el gráfico.

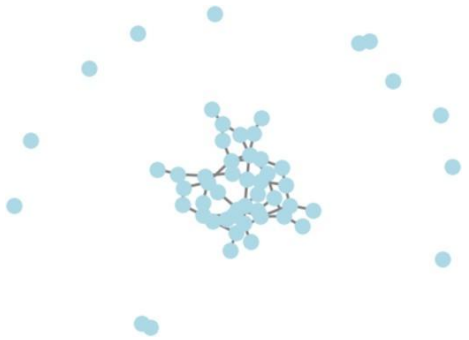


III. METODOLOGÍA

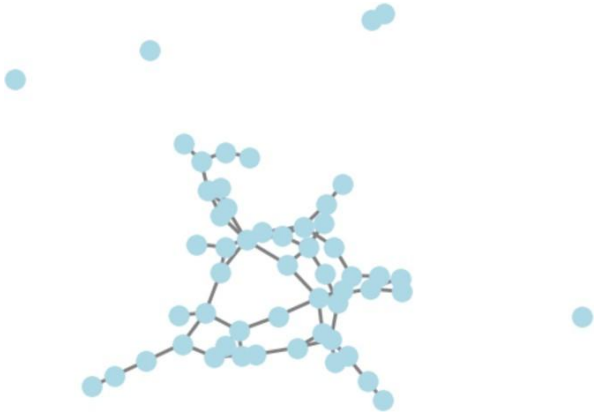
En primer lugar, realizamos la lectura de los datos almacenados en dos archivos CSV: uno que contiene los atributos de los nodos y otro que registra las relaciones entre ellos y creamos un grafo a partir de estos datos.

views	mature	life_time	created_at	updated_at	numeric_id
7879	1	969	2016-02-16	2018-10-12	0
500	0	2699	2011-05-19	2018-10-08	1
382502	1	3149	2010-02-27	2018-10-12	2
386	0	1344	2015-01-26	2018-10-01	3
2486	0	1784	2013-11-22	2018-10-11	4

dead_account	language	affiliate
0	EN	1
0	EN	0
0	EN	1
0	EN	0
0	EN	0



Fig[1,2] subgrafo para n vértices



A continuación codificamos los atributos “created_at”, “updated_at” y “language” mediante el método HotEncoder, el cual nos permite codificar atributos categóricos en una representación de números y concatenamos con los atributos numéricos.

Antes de proceder con los metodos elegidos, dividimos los datos en conjuntos de entrenamiento, validación y prueba. Para finalizar comenzamos con los métodos elegidos:

IV. RESULTADOS

- GCN: El método GCN aplicado a nuestro caso nos permitirá analizar y comprender de manera efectiva las relaciones y conexiones entre los canales, lo que nos permitirá realizar clasificación o predicción.
  Como el grafo que obtenemos a partir del CSV es demasiado grande, tomamos una muestra menorpara el entrenamiento, validación y pruebas

```
La predicción para el modelo por el método GCN será:
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
El accuracy conseguido mediante este método será: 0.55
```

- Naive Bayes: Naive Bayes se utiliza para clasificar o predecir características en función de atributos disponibles. En n puede ayudar a clasificar canales en categorías o predecir su popularidad. Es útil para características categóricas o discretas.

```
La predicción para el método Naive Bayes será:
[1 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0]
El accuracy para el método Naive Bayes será: 0.55
```

- kNN: El modelo kNN podría ser muy útil para dar una recomendación personalizada sobre los canales que podrían interesarte según otros canales que sigues

```
Las predicciones mediante el modelo kNN con k = 3 son:
[1 0 1 1 1 1 1 0 0 1 0 0 1 0 0 0 0 0 1 0]
El accuracy mediante el modelo kNN con k = 3 es: 0.6
```

V ELECCION DE METODO

Debido a la similitud de las precisiones de todos los métodos utilizado es complicado elegir un método de mejor resolución. Sin embargo, el método que más se aleja tanto en la precisión media obtenida como en las predicciones es el kNN para k = 3.
Por lo tanto podemos afirmar que este es el que más se aleja de la realidad.

VI CONCLUSIONES

Además, los métodos utilizados en este trabajo, como Graph Convolution (GCN), Naive Bayes y k-Nearest Neighbors (kNN), han demostrado fortalezas y aplicaciones específicas en el análisis de datos, ya sea Twitch. El GCN fue eficaz para modelar la estructura de la red y capturar interacciones entre canales, lo que ayuda a descubrir comunidades y patrones de conectividad. Por otro lado, Naive Bayes es muy útil para el análisis de características y la clasificación de canales en función de la probabilidad condicional.

Por su parte, kNN destaca por identificar similitudes y grupos entre secuencias. Usando métricas de distancia y considerando k vecinos más cercanos, kNN puede asignar nuevas instancias a las clases respectivas en función de las características de los canales previamente etiquetados. Esta capacidad de clasificación y predicción basada en la similitud nos ha permitido comprender mejor la distribución y clasificación de los canales en función de sus propiedades.

En conjunto, estos métodos de aprendizaje automático brindan información valiosa sobre los datos de Twitch, lo que brinda información valiosa para la toma de decisiones a nivel de plataforma. Los resultados obtenidos no difirieron significativamente entre los diferentes métodos, lo que indica consistencia en los patrones y comportamiento observado en los canales de Twitch. A demás podemos observar que ninguno de los atributos marca una diferencia considerable con respecto a los demás.

Es importante enfatizar que si bien estos métodos han demostrado ser útiles en este trabajo, existen oportunidades de mejora y refinamiento en futuras investigaciones. La exploración de otros algoritmos de aprendizaje automático, la optimización de hiperparámetros y la incorporación de nuevas variables pueden conducir a una mayor previsibilidad y precisión. En resumen, este trabajo proporciona una base sólida para futuros estudios y aplicaciones de aprendizaje automático en el contexto de nuestro dataset. Los resultados obtenidos a través de GCN, Naive Bayes y kNN brindan información valiosa sobre las relaciones entre los canales y sus características, así como oportunidades para mejorar la comprensión de la plataforma y tomar decisiones informadas, más transparencia en la estrategia de marketing, recomendaciones de contenido y comunidad de Twitch. gestión.

VI Referencias

[1] Scikit-Learn Machine Learning in Python [scikit-learn.org]
[2] Networkx Documentation [networkx.org/]
[3] Librería Pandas: [https://pandas.pydata.org/pandasdocs/stable/index.html]
[4] Librería PyTorch [https://pytorch.org/docs/stable/index.html]
[5] GCN [https://paperswithcode.com/method/gcn]
[6] Algoritmos de aprendizaje automatico [https://keepcoding.io/blog/tipos-de-algoritmos-de-machine-learning]