**NN**

----------------------------------------------------------------------------------

[¿Cómo medir el éxito de un sistema de recomendación? (analyticsindiamag.com)](https://analyticsindiamag.com/how-to-measure-the-success-of-a-recommendation-system/)

###### *Métricas de precisión predictiva*

Las medidas de precisión predictiva o predicción de calificación abordan el tema de qué tan cerca están las calificaciones estimadas de un recomendador de las calificaciones genuinas de los usuarios. Este tipo de medida se utiliza ampliamente para evaluar las calificaciones no binarias.

Es el más adecuado para escenarios de uso en los que la predicción precisa de las clasificaciones de todos los productos es fundamental. El Error Absoluto Medio (MAE), el Error Cuadrático Medio (MSE), el Error Cuadrático Medio (RMSE) y el Error Absoluto Medio Normalizado (NMAE) son las medidas más importantes para este propósito.

En comparación con la métrica MAE, MSE y RMSE emplean desviaciones al cuadrado y, en consecuencia, enfatizan errores más grandes. El error es descrito por MAE y RMSE en las mismas unidades que los datos obtenidos, mientras que MSE produce unidades al cuadrado.

Para que los resultados sean comparables entre recomendadores con diferentes escalas de calificación, NMAE normaliza la medida MAE al rango de la escala de calificación apropiada. En la competencia de Netflix, se utilizó la medida RMSE para determinar la mejora en comparación con el algoritmo de Cinematch, así como el ganador del premio.

--------------------------------------------------------------------------------------------

----------------------------------------------------------------------------------

<https://medium.com/@rishabhbhatia315/recommendation-system-evaluation-metrics-3f6739288870>

La precisión se mide utilizando dos valores.

1. **MAE (error absoluto medio)**

*def MAE (predicciones):   
devuelve precisión.mae (predicciones, detallado = Falso)*

2. **RMSE (error cuadrático medio)**

def RMSE (predicciones):   
devuelve precisión.rmse (predicciones, detallado = Falso)

**3. Tasa de aciertos**

Genere las n recomendaciones principales para un usuario y compárelas con las que el usuario calificó, le gustaron o vio. Si coinciden, aumente la tasa de aciertos en 1, haga esto para el conjunto de entrenamiento completo para obtener la tasa de aciertos.

Tomamos el umbral y la calificación prevista y, primero, filtramos los valores según el umbral. Después de eso, ordenamos los valores en orden descendente y devolvemos los n valores principales predichos para cada usuario que usa esta función.

Para entrenar nuestro modelo, podríamos usar todo el conjunto de datos como nuestro conjunto de datos de entrenamiento y usar todo el conjunto de datos como nuestro conjunto de datos de prueba, pero esto no será útil ya que generalmente no desea evaluar un modelo con los datos con los que fue entrenado. Como podríamos predecir la lista top n con la que se entrenó a un usuario y daría una tasa de acierto del 100 por ciento, piénselo. Por lo tanto, podemos usar el método de validación cruzada de dejar uno fuera, en el que eliminamos un dato de la lista de datos de entrenamiento principales de cada usuario y usamos esos datos para evaluar nuestro modelo.Si la lista top-n generada contiene los datos excluidos del conjunto de entrenamiento, entonces lo consideramos como un acierto. Mire el código para obtener más explicaciones.

4. **Tasa de aciertos acumulada**

Es similar a la tasa de aciertos que se muestra arriba, pero agregamos un umbral para el conjunto de prueba para que podamos probar las películas que realmente le gustan a la persona.

**5. Tasa de aciertos recíprocos promedio**

Esta métrica también es similar a la tasa de aciertos. Pero para cada usuario para el cual tenemos un acierto, sumamos el recíproco de la clasificación del movieId coincidente en la lista de los n primeros y las predicciones omitidas en la lista de los n primeros para cada usuario y lo dividimos por el número de usuarios. Esta es una métrica centrada en el usuario, ya que las personas tienden a concentrarse más en lo que ven al comienzo de las n listas principales. Por eso le damos más peso a los resultados que aparecen en la parte superior de la lista.

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

<https://medium.com/@m_n_malaeb/recall-and-precision-at-k-for-recommender-systems-618483226c54>

El RMSE se puede calcular comparando la calificación prevista con la calificación real para cada par de usuario-elemento con una etiqueta conocida.

**Solo estamos interesados ​​en calcular la precisión y recuperar en k** . [La precisión y la recuperación](https://en.wikipedia.org/wiki/Precision_and_recall) son métricas binarias que se utilizan para evaluar modelos con salida binaria. Por lo tanto, necesitamos una forma de traducir nuestro problema numérico (calificaciones generalmente de 1 a 5) en un problema binario (elementos relevantes y no relevantes).

Para hacer la traducción asumiremos que cualquier **calificación real** superior a 3,5 corresponde a un ítem relevante y cualquier **calificación real** inferior a 3,5 es irrelevante. Un elemento relevante para un par de elemento-usuario específico significa que este elemento es una buena recomendación para el usuario en cuestión.

3,5 es sólo un valor umbral que elegí. Hay varias formas de establecer este valor umbral, como tener en cuenta el historial de calificaciones otorgadas por el usuario. En aras de la simplicidad, nos ceñiremos al umbral de 3,5.

## Configurando 'k'

En el contexto de los sistemas de recomendación, lo más probable es que estemos interesados ​​en recomendar los N elementos principales al usuario. Por lo tanto, tiene más sentido calcular la precisión y recuperar métricas en los primeros N elementos en lugar de en todos los elementos. De ahí la noción de *precisión y recuperación en k,* donde kes un número entero definible por el usuario que el usuario establece para que coincida con el objetivo de las N principales recomendaciones.

# Los elementos relevantes ya se conocen en el conjunto de datos

Elemento relevante: Tiene una calificación verdadera/real >= 3,5

Elemento irrelevante: Tiene una calificación verdadera/real < 3,5

# Los elementos recomendados se generan mediante un algoritmo de recomendación

Elemento recomendado: tiene una calificación prevista >= 3,5

Elemento no recomendado: tiene una calificación prevista < 3,5

## Precisión y recuperación en k: definición

*La precisión en k es la proporción de elementos recomendados en el conjunto top-k que son relevantes*

Su interpretación es la siguiente. Supongamos que mi precisión en 10 en un problema de recomendaciones entre las 10 principales es del 80%. Esto quiere decir que el 80% de las recomendaciones que hago son relevantes para el usuario.

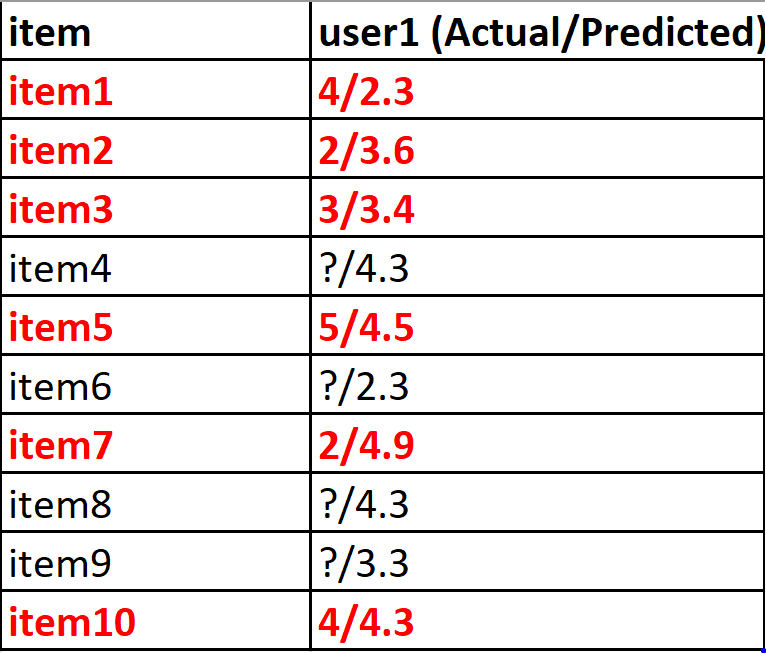
Matemáticamente precision@k se define de la siguiente manera:

Precisión@k = (# de artículos recomendados @k que son relevantes) / (# de artículos recomendados @k)

Supongamos que calculamos el recuerdo en 10 y encontramos que es del 40% en nuestro sistema de recomendaciones principales. Esto significa que el 40% del número total de elementos relevantes aparecen en los primeros k resultados.

Matemáticamente, recordar@k se define de la siguiente manera:

Recall@k = (# of recommended items @k that are relevant) / (total # of relevant items)



Para empezar, ignoraremos todas las calificaciones en las que no se conozca el valor real. No se pueden utilizar valores sin una calificación verdadera conocida.

Ordenaremos el resto de los elementos según la calificación de predicción descendente. Los resultados serán los siguientes:

será el siguiente:

artículo/real/previsto   
artículo7/2/4.9   
artículo5/5/4.5   
artículo10/4/4.3   
artículo2/2/3.6   
artículo2/3/3.4   
artículo1/4/2.3

**Artículos relevantes:**

El número de ítems relevantes son los ítems con calificación real mayor o igual a 3,5.

Artículos relevantes: artículo 5, artículo 10 y artículo 1   
# total de artículos relevantes = 3

**Artículos recomendados @ 3:**

Los artículos recomendados en 3 sonitem7, item5 and item10

Artículos recomendados @ 3: artículo 7, artículo 5 y artículo 10   
# de artículos recomendados en 3 = 3

**Artículos recomendados y relevantes @ 3**

Es la intersección entre Recomendado@3 y Relevante@3 que son

Recomendado@3 INTERSECCIÓN Relevante: elemento5 y elemento10   
# de elementos recomendados que son relevantes @3= 2

**Precisión @ 3:**

Podemos calcular la precisión que es 66,67%. Aquí podemos interpretar que sólo el 66,67% de mis recomendaciones son realmente relevantes para el usuario.

Precisión@3   
=(# de artículos recomendados que son relevantes @3)/(# de artículos recomendados en 3)   
= 2/3   
= 66,67%

**Recuerde @ 3:**

Aquí podemos interpretar que el 66,67% de los elementos relevantes se recomendaron entre los k elementos principales.

Recordar@3   
= (# de artículos recomendados que son relevantes @3)/(# total de artículos relevantes)   
= 2/3   
= 66,67%

## Casos límite

En el cálculo de precision@k, dividimos por la cantidad de elementos recomendados en la recomendación top-k. Si no hay artículos recomendados. es decir, el número de elementos recomendados en k es cero, no podemos calcular la precisión en k ya que no podemos dividir por cero. En ese caso establecemos la precisión en k en 1. Esto tiene sentido porque en ese caso no tenemos ningún elemento recomendado que no sea relevante.

De manera similar, al calcular el recuerdo @k podríamos enfrentarnos a una situación similar cuando el número total de elementos relevantes sea cero. En ese caso, establecemos el recuerdo en k como 1. Esto también tiene sentido porque no tenemos ningún elemento relevante que no esté identificado en nuestros resultados top-k.

## Nota final

Finalmente me gustaría agregar que lo que expliqué anteriormente es sólo una forma de calcular la precisión y recuperar en k. Otras variaciones de estas métricas de evaluación están disponibles en la literatura y pueden ser más complejas.

Puede encontrar una implementación en Python de las métricas explicadas anteriormente en la sección de preguntas frecuentes de la Biblioteca Surprise. --> <https://surprise.readthedocs.io/en/latest/FAQ.html#how-to-compute-precision-k-and-recall-k>

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

<https://medium.com/@mnshonco/recommender-systems-made-easy-build-and-evaluate-from-the-scratch-16c386aa8a61>

**Prueba 3. ¿Qué recomienda el modelo** ?

Una vez entrenadas las incrustaciones, es una tarea sencilla encontrar películas recomendadas. La multiplicación de matrices y el cambio de escala generarán puntuaciones para cualquier combinación de par de usuario-película. Alternativamente, al utilizar la incrustación de todos los usuarios y todas las películas, podemos recuperar las calificaciones previstas para todas las películas para todos los usuarios en forma de una matriz densa. A continuación, para el usuario seleccionado, simplemente ordene las calificaciones previstas para obtener las mejores recomendaciones.

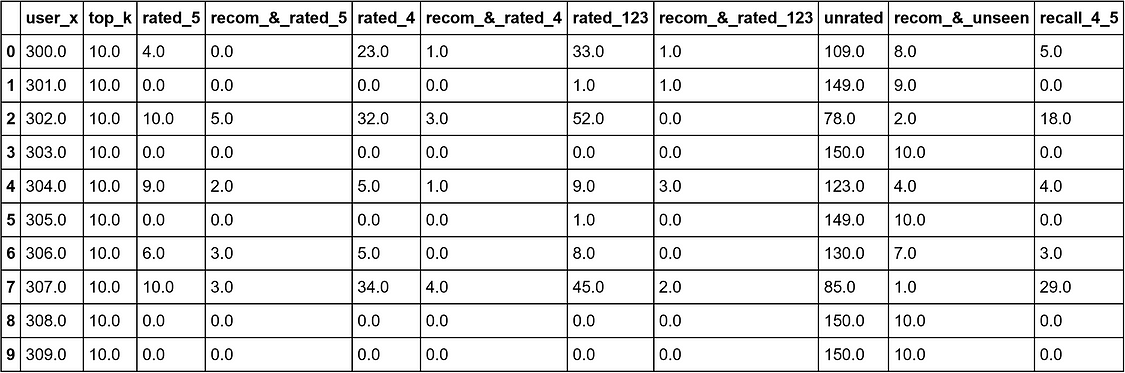
Como ejemplo, aquí hay recomendaciones para el usuario 302, las 10 mejores películas se seleccionan del conjunto de prueba. A la izquierda están las películas que calificó con 5 y a la derecha están las películas que le recomendaron. Como se ve en la tabla a continuación, de las 10 recomendaciones principales, 5 películas en realidad recibieron una calificación de 5. Otras 3 películas en realidad obtuvieron una calificación de 4 y las 2 películas restantes aún no las había visto o no las había calificado este usuario. Ninguna de las películas recomendadas tuvo calificaciones bajas (1, 2 o 3).



Rated\_5 son aquellos que en realidad calificaron con 5 por este usuario. A la derecha se encuentran las películas recomendadas, de las cuales cinco películas han recibido una calificación de 5.

**Prueba 4. Precisión y recuperación** :

El marco de datos a continuación captura todas las recomendaciones (top\_k = 10) para cada usuario en el conjunto de prueba. La tabla también muestra la intersección de esta recomendación con las calificaciones reales del usuario. Sin embargo, la parte importante de las recomendaciones no tiene calificaciones. A partir de los datos disponibles, podemos calcular la precisión promedio media (MPA) y la recuperación del modelo.



Datos de los primeros 10 usuarios.

Código para construir el marco de datos de pandas y obtener la lista de recomendaciones

La función anterior devuelve una tabla con todas las estadísticas y una lista con películas altamente calificadas y películas recomendadas para cada usuario. De la lista de películas calificadas/recomendadas, podemos recuperar las películas usando los índices siguientes.

Evaluación: La precisión promedio media (MAP) y la recuperación son las principales métricas utilizadas para evaluar los sistemas de recomendación. A continuación se muestran las ecuaciones para derivar la precisión para todos y cada uno de los usuarios.

----------------------------------------------------------------------------------

**CFC**

....

**FC**

----------------------------------------------------------------------------------

<https://medium.com/decathlondigital/building-a-rnn-recommendation-engine-with-tensorflow-505644aa9ff3>

El filtrado colaborativo se utiliza habitualmente para los sistemas de recomendación. Sin embargo, resulta ser un marco rígido que impide el uso de cualquier otro metadato útil. Además, se supone que los usuarios y los elementos dependen sólo linealmente. Trabajos más recientes ( [Netflix](https://netflixtechblog.com/supporting-content-decision-makers-with-machine-learning-995b7b76006f) , [YouTube](https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45530.pdf) , [Spotify](https://tbtech.co/how-deep-learning-is-central-to-spotifys-strategy/) ) han demostrado que los marcos de aprendizaje profundo superan a los métodos de factorización clásicos gracias al uso de **transformaciones no lineales más ricas de los datos de entrada** , así como a la adición de **metadatos útiles** .

----------------------------------------------------------------------------------

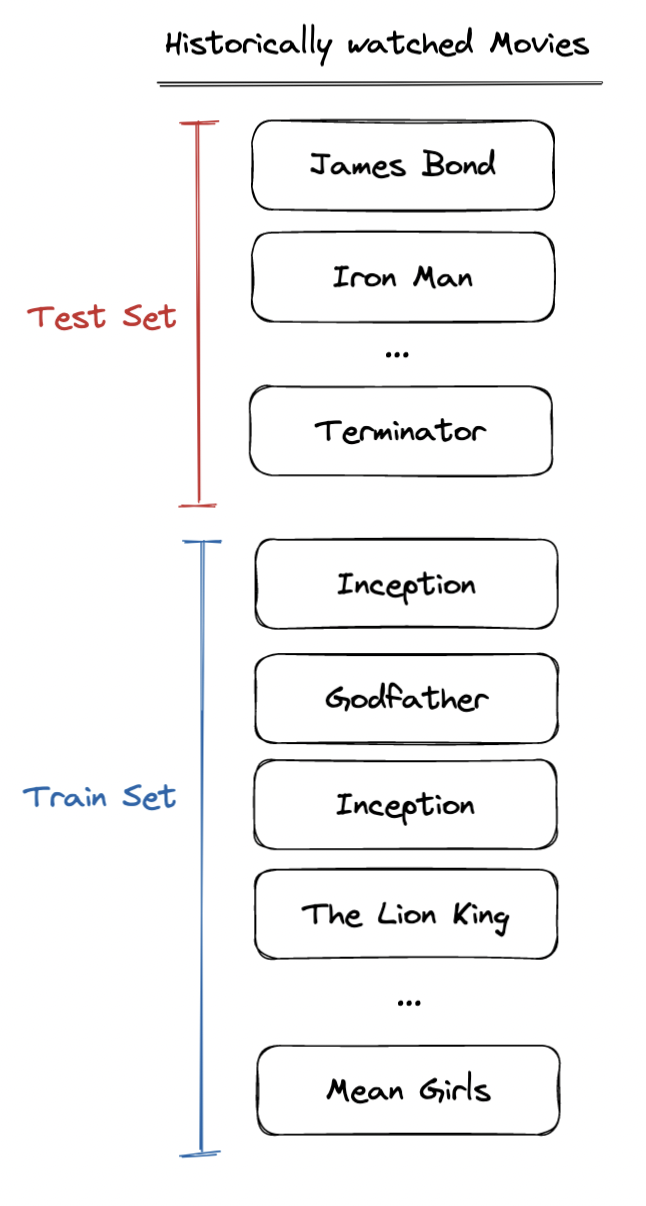
**PARA LOS TRES**

----------------------------------------------------------------------------------

[**https://www.shaped.ai/blog/evaluating-recommendation-systems-part-1**](https://www.shaped.ai/blog/evaluating-recommendation-systems-part-1)

¿Cómo podemos medir objetivamente qué algoritmo de recomendación es mejor para su usuario? Una forma de evaluar los algoritmos fuera de línea [2] es a través de un proceso llamado "validación cruzada", donde la lista de vigilancia histórica (es decir, la lista de elementos relevantes) se divide cronológicamente [3] en conjuntos de tren y prueba. Los algoritmos de recomendación se entrenan en el conjunto de trenes y las métricas de rendimiento se evalúan en el conjunto de prueba. Estas métricas de rendimiento se pueden utilizar para medir objetivamente qué algoritmo es más relevante.

Continuando con el ejemplo, supongamos que el conjunto de prueba para el usuario en cuestión contiene: Terminator, James Bond, Iron Man y otras 3 películas no relacionadas. Podemos medir el rendimiento comparando las coincidencias entre el conjunto de evaluación y ambas listas de recomendaciones. Dos métricas clásicas que se utilizan son: Precision@k y Recall@k .

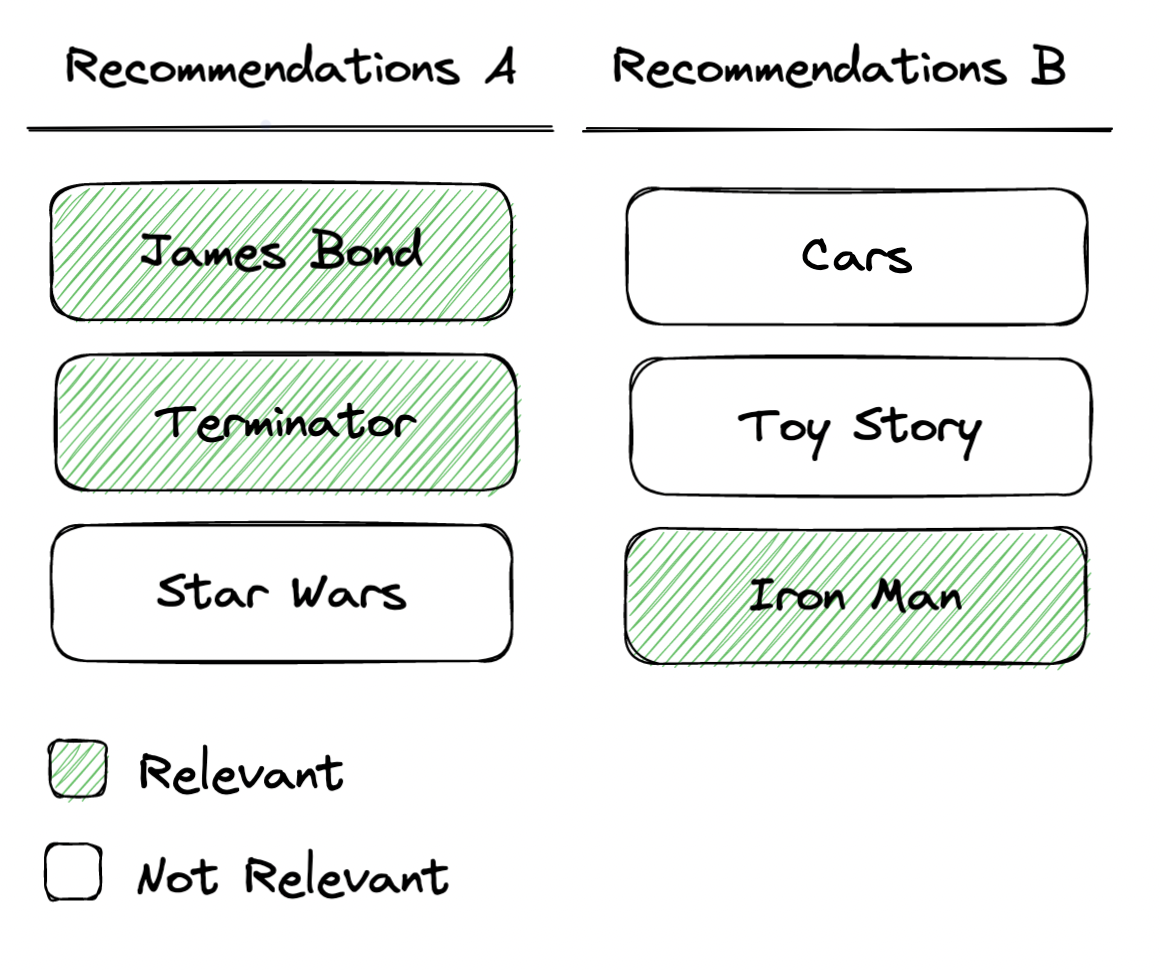


### 

### Precisión@k

‍Precision @k mide la proporción de artículos recomendados relevantes en una lista de recomendaciones de tamaño k . Para la primera lista de recomendaciones (The Terminator, James Bond y Love Actually), podemos ver que hay 2 coincidencias de los 3 elementos. Para la segunda lista, hay 1 coincidencia de los 3 elementos. Por lo tanto:

1. Algoritmo A, Precisión@3 = 2/3 = 0,666
2. Algoritmo B, Precisión@3 = 1/3 = 0,333



Medida con respecto al total de recomendaciones.

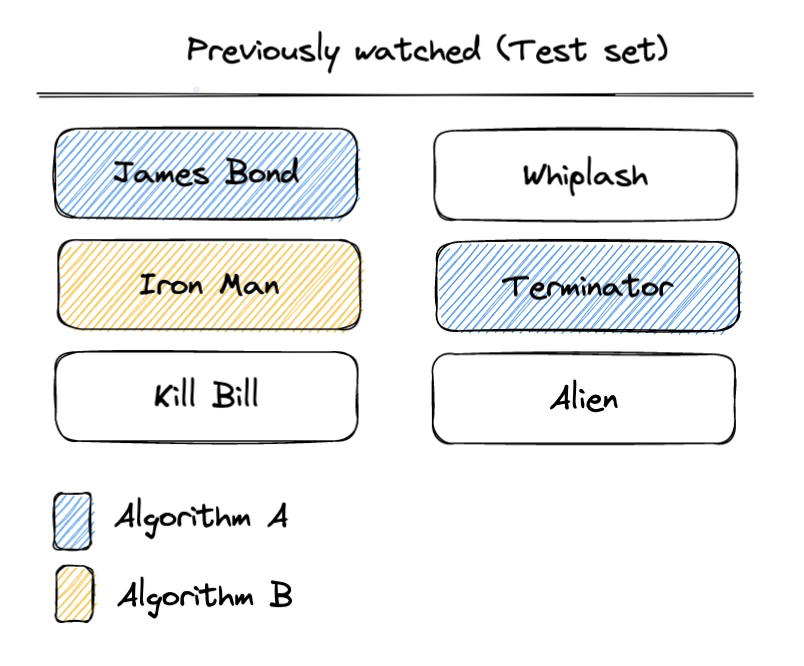
‍ Recall

@k

La otra forma en que podemos definir coincidencias se basa en la proporción de elementos recomendados relevantes (en la lista de recomendaciones de tamaño k) y el número total de elementos relevantes. Esta métrica se llama recordar@k . Por ejemplo, el usuario ha visto 6 películas y en la primera lista de recomendaciones 2 de ellas son relevantes. En la segunda lista, 1 de ellos es relevante. Por lo tanto:

Algoritmo A, Recuperación@3 = 2/6 = 0,333

Algoritmo B, Recuperar @ 3 = 1/6 = 0,166



Medida con respecto al total de la lista de relevantes

Precision@k nos brinda una comprensión interpretable de cuántos elementos son realmente relevantes en las k recomendaciones finales que mostramos a un usuario. Si confía en su elección de k, normalmente se corresponde bien con la recomendación final y es fácil de comunicar. El problema es que está fuertemente influenciado por la cantidad de elementos relevantes que tenga el usuario. Por ejemplo, imagine que nuestro usuario solo vio 2 elementos relevantes, la máxima Precisión@3 que podría lograr con un sistema de recomendación perfecto tiene un límite de: Precisión@3 = 2/3 = 0,666. Esto causa problemas al promediar el resultado entre varios usuarios.

Lo mejor de ambos mundos: R-Precision

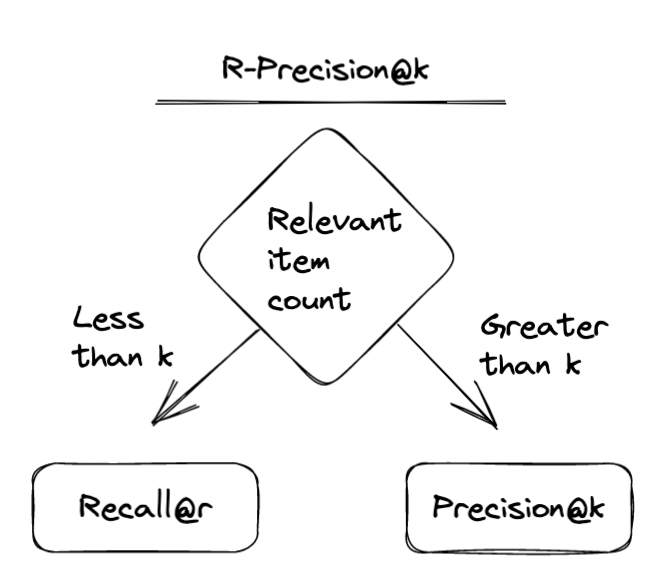
Una métrica que resuelve estos problemas con Precision se llama R-Precision. Ajusta k a la longitud de los elementos relevantes del usuario. Para nuestro ejemplo del último párrafo (donde el usuario solo ha visto 2 películas), esto significa que tenemos R-Precision=2/2=1, para un sistema de recomendación perfecto, que es lo que esperaríamos.

R-Precisión = (número de elementos top-r relevantes recomendados) / r

Donde r es el número total de elementos relevantes.

Tenga en cuenta que cuando r = k, Precision@k, Recall@k y R-Precision son todos iguales.

Cuando tenemos un presupuesto fijo de recomendaciones k, como en el ejemplo que estamos ejecutando, probablemente quieras limitar k. Esta métrica limitada, R-Precision@k, se puede considerar como Recall@r cuando el número de elementos relevantes es menor que k y Precision@k cuando es mayor que k. Nos brinda lo mejor de ambos mundos de las dos métricas y promedia bien entre los usuarios.



‍

Las métricas que hemos repasado hoy: Precision@k, Recall@k, R-Precision, son formas clásicas de evaluar la precisión de dos conjuntos desordenados de recomendaciones con k fijo. Sin embargo, el orden de clasificación puede ser crucial para evaluar la calidad de las recomendaciones a sus usuarios, particularmente para casos de uso de recomendaciones donde hay una superficie mínima para mostrar sus recomendaciones y cada posición de clasificación es importante. También es necesario considerar objetivos alternativos, como la diversidad y el sesgo, más allá de las métricas de precisión de relevancia de las que hablamos hoy. Finalmente, todas estas métricas son excelentes para la evaluación fuera de línea con validación cruzada, pero es importante no olvidarse de la evaluación en línea, es decir, medir los resultados después de haber comenzado a presentar las recomendaciones directamente a los usuarios.

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

[¿Cómo medir el éxito de un sistema de recomendación? (analyticsindiamag.com)](https://analyticsindiamag.com/how-to-measure-the-success-of-a-recommendation-system/)

**Precisión media media @ K y media media de recuerdo @ K**

Para cada usuario del conjunto de pruebas, un sistema de recomendación normalmente genera una lista ordenada de recomendaciones. MAP@K indica la relevancia de la lista de elementos recomendados, mientras que MAR@K indica lo bien que el recomendador puede recordar todos los elementos del conjunto de prueba que el usuario ha calificado positivamente.

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

[Clase 4. Métricas de Evaluación para Sistemas de… | by grupo6 | Medium](https://medium.com/@Grupo6/clase-4-514bfb795f4)

Recall: de todos los que me gustan, cuántos me ha mostrado el recomendador.

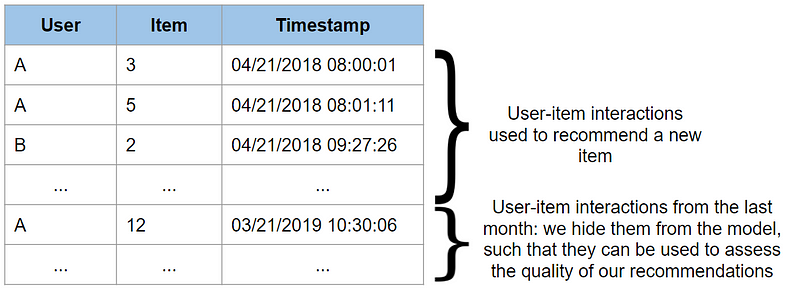
Precisión: de todos los que me ha mostrado el recomendador, cuántos me gustan.

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

[https://medium.com/decathlondigital/personalize-your-app-or-website-using-your-catalog-of-images-e5a921868413ç](https://medium.com/decathlondigital/personalize-your-app-or-website-using-your-catalog-of-images-e5a921868413%C3%A7)

Durante el desarrollo de un sistema de recomendación, generalmente verificamos la calidad de nuestro trabajo ocultando de nuestro modelo las interacciones más recientes entre el usuario y el elemento: por ejemplo, podemos ocultar de nuestro modelo las interacciones entre el usuario y el elemento que ocurrieron durante el último mes. . Por lo tanto, estas interacciones representan el “futuro” de nuestro sistema de recomendación, y su comparación con las sugerencias dadas por el sistema de recomendación representa un gran indicador de su calidad.



La precisión es la fracción de elementos recomendados con los que nuestros usuarios han interactuado durante el último mes. El recuerdo es, hasta cierto punto, lo contrario: es la fracción de elementos con los que los usuarios han interactuado durante el último mes que se pueden encontrar en las recomendaciones. Finalmente, la cobertura es la fracción de artículos de nuestro catálogo que han sido recomendados al menos a un usuario. Para todas estas métricas, ¡cuanto más altas, mejor para nuestro sistema de recomendación

----------------------------------------------------------------------------------

<https://flowthytensor.medium.com/some-metrics-to-evaluate-recommendation-systems-9e0cf0c8b6cf>

**Ganancia acumulada descontada normalizada (NDCG) @ K**

NDCG mide la recompensa general en todas las posiciones que tienen un artículo relevante. La recompensa es un registro inverso de la posición (es decir, rangos más altos para elementos relevantes conducirían a una mejor recompensa, según se desee).

**Ganancia acumulada descontada normalizada (NDCG) @ K**

NDCG mide la recompensa general en todas las posiciones que tienen un artículo relevante. La recompensa es un registro inverso de la posición (es decir, rangos más altos para elementos relevantes conducirían a una mejor recompensa, según se desee).

----------------------------------------------------------------------------------

**Common**

----------------------------------------------------------------------------------

<https://towardsdatascience.com/evaluation-metrics-for-recommendation-systems-an-overview-71290690ecba>

Cosas a considerar al evaluar los sistemas de recomendación:

Planeo publicar artículos sobre estos temas pronto y, en ese caso, los vincularé aquí.

Sesgo de popularidad

Puede haber una gran cantidad de sesgo de popularidad en los sistemas de recomendación y es difícil de detectar y corregir porque los elementos populares son relevantes y obtienen una puntuación alta en la mayoría de las métricas estándar. Hay varias formas de medir y corregir el sesgo de popularidad y planeo elaborar un artículo que hable sobre ellas.

Imagen por autor

2. Sesgo de posición

El sesgo de posición se produce cuando los elementos situados más arriba en una lista tienen más probabilidades de ser vistos o comprados, independientemente de su relevancia real. Como resultado, los elementos con clasificaciones más bajas reciben menos participación. Esto también podría afectar otras métricas y existen métodos para mitigarlo.

3. Bucle de retroalimentación degenerado

Cuando los usuarios se limitan a interactuar con los elementos sugeridos y los algoritmos se basan en los comentarios de los usuarios sobre estas sugerencias para la capacitación, puede surgir un ciclo de retroalimentación negativa. Este bucle refuerza la visualización de elementos que se han mostrado anteriormente, lo que potencialmente resulta en una experiencia de usuario negativa con el tiempo debido a que una parte importante de elementos relevantes permanecen sin descubrir por parte del usuario.

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

**Chen, M., & Liu, P. (2017). Performance Evaluation of Recommender Systems. International Journal of Performability Engineering, 13(8), 1246-1256. DOI: 10.23940/ijpe.17.08.p7.12461256**

[**Article (utdallas.edu)**](https://paris.utdallas.edu/IJPE/Vol13/Issue08/IJPE-2017-08-07.pdf)

En la literatura de sistemas de recomendación, se utilizaron una variedad de algoritmos de aprendizaje automático para predecir las calificaciones de los usuarios, incluyendo regresión, SVD, PCA, inferencia de probabilidad, redes neuronales, entre otros [28,32]. Por lo tanto, la precisión de la predicción desde la perspectiva del aprendizaje automático se aplica naturalmente a la evaluación de la precisión de las calificaciones de los usuarios de los sistemas de recomendación. La métrica de precisión de la predicción se trata esencialmente del error de predicción. Esta métrica es común en la evaluación de varios algoritmos de aprendizaje automático, como la regresión o la clasificación. Esta métrica utilizada en la evaluación de sistemas de recomendación se usa principalmente para medir la capacidad de predecir los comportamientos de los usuarios. La precisión de la predicción es la métrica más importante en el análisis fuera de línea de los sistemas de recomendación. Esta métrica se utiliza comúnmente en los primeros documentos de investigación de sistemas de recomendación para discutir la precisión de diferentes algoritmos de recomendación.

Al calcular la precisión de la predicción, se necesita un conjunto de datos sin conexión que contenga las calificaciones de los usuarios, como las calificaciones de los usuarios para un producto o una película. El conjunto de datos se divide en conjunto de entrenamiento y conjunto de prueba. Se entrena un modelo de predicción de calificaciones de usuarios en el conjunto de entrenamiento, luego se calcula la predicción de las calificaciones de usuarios en el conjunto de prueba. El error es la desviación entre la calificación predicha y la calificación real. Hay tres métricas para medir la precisión de la predicción: Error Absoluto Medio (MAE), Error Cuadrático Medio (MSE) y Error Cuadrático Medio Raíz (RMSE), y las fórmulas son las siguientes:

Error Absoluto Medio: 𝑀𝐴𝐸= 1 |𝑄| ∑|𝑟𝑢𝑖−𝑟𝑢𝑖| (𝑢,𝑖)∈𝑄 (1)

Error Cuadrático Medio: 𝑀𝑆𝐸= 1 |𝑄| ∑(𝑟𝑢𝑖−𝑟𝑢𝑖)2 (𝑢,𝑖)∈𝑄 (2)

Error Cuadrático Medio Raíz: 𝑅𝑀𝑆𝐸=√ 1 |𝑄| ∑(𝑟𝑢𝑖−𝑟𝑢𝑖)2 (𝑢,𝑖)∈𝑄 (3)

MAE es el más simple, pero no tiene en cuenta la dirección del error (error positivo o error negativo). MSE tiene una penalización mayor en errores grandes y el error cuadrático no tiene un significado intuitivo. Por lo tanto, el RMSE se usa más ampliamente en el cálculo de la precisión de predicción del sistema de recomendación.

3.2. La perspectiva de la recuperación de información

Los sistemas de recomendación también se consideran como un caso especial de sistemas de recuperación de información, que recuperan la información relacionada de los datos históricos del usuario [14,25]. Aunque la precisión de predicción es una métrica importante para evaluar un sistema de recomendación, a veces al usuario no le importa la precisión numérica específica, como si al usuario le importa si la calificación de una película es 4.9 o 4.8; al usuario le preocupa si la película es maravillosa o no. Por lo tanto, es necesario estudiar las métricas de apoyo a la decisión y clasificación. El objetivo del soporte de decisión es ayudar a los usuarios a elegir elementos "buenos", mientras que las métricas de clasificación están relacionadas con el orden de los elementos recomendados que a los usuarios les gustan.

Un sistema de recomendación generalmente recomienda una lista de elementos para los usuarios. Generalmente están dispuestos horizontal o verticalmente. Los usuarios generalmente solo se preocupan por las partes delanteras de varios elementos y pocos usuarios se preocuparán por las partes traseras de los elementos, como los elementos enumerados en la página 20. Esta forma de recomendación se llama recomendación Top-n. Podemos usar la precisión, recall y F-Measure del campo de recuperación de información para evaluar el rendimiento del sistema de recomendación. Las fórmulas son las siguientes:

Precisión: 𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛=𝑁𝑟𝑠 𝑁𝑠 o 𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛@𝑛=𝑁𝑟𝑠@𝑛 𝑛 (4)

Recall: 𝑅𝑒𝑐𝑎𝑙𝑙=𝑁𝑟𝑠 𝑁𝑟 (5)

F-Measure: 𝐹−𝑀𝑒𝑎𝑠𝑢𝑟𝑒=2×𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛×𝑅𝑒𝑐𝑎𝑙𝑙 𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛+𝑅𝑒𝑐𝑎𝑙𝑙 (6)

donde 𝑁𝑟𝑠 es el número de elementos recomendados que prefiere el usuario, 𝑁𝑠 es el número de elementos recomendados y n es el primer n elementos recomendados. 𝑁𝑟 es el número de elementos que prefiere el usuario. La precisión de la recomendación describe la proporción de elementos que prefieren los usuarios, y el recall describe la proporción de los elementos favoritos del usuario que no se pierden. F-Measure es un compromiso entre la precisión y el recall. En la aplicación de la recomendación, los usuarios generalmente prestan más atención a la precisión del top-n, en lugar del recall.

3.2.4. Rango Medio Recíproco

El concepto de Rango Medio Recíproco (MRR) proviene del sistema de recuperación de información, lo que significa que los resultados de recuperación más relevantes deben clasificarse primero. Cuando se aplica la métrica de MRR al sistema de recomendación, puede medir si el sistema de recomendación coloca los elementos favoritos del usuario al principio. El MRR se define de la siguiente manera:

𝑀𝑅𝑅= ∑ 1/𝑟𝑎𝑛𝑘𝑖 𝑄 𝑞=1 𝑄 (9)

donde Q es el número de recomendación, 𝑟𝑎𝑛𝑘𝑖 son los elementos favoritos del usuario en la lista de clasificación de recomendación. Obviamente, cuanto mayor sea el valor de MRR, mejor será el rendimiento del sistema de recomendación.

3.2.6. Ganancia Acumulada Descuentada Normalizada

El coeficiente de correlación de rango de Spearman en realidad no considera la posición de las clasificaciones, en otras palabras, la penalización por todas las clasificaciones incorrectas es la misma. Por lo tanto, en los últimos años, la Ganancia Acumulada Descuentada Normalizada (nDCG) se ha utilizado ampliamente en la evaluación de sistemas de recomendación [12]. El nDCG se define de la siguiente manera:

𝑛𝐷𝐶𝐺= 𝐷𝐶𝐺(𝑟) 𝐷𝐶𝐺(𝑟𝑝𝑒𝑟𝑓𝑒𝑐𝑡) (11)

𝐷𝐶𝐺(𝑟)=∑𝑑𝑖𝑠𝑐(𝑟(𝑖))𝑢(𝑖) (12)

donde disc(r(i)) es una función de descuento basada en la clasificación, que hace que la clasificación de los elementos anteriores sea más importante; u(i) es la utilidad de los elementos en la lista de recomendaciones, como la calificación o clic, navegación o compra del usuario. DCG(rperfect) representa una clasificación perfecta de ganancia acumulada descontada.

3.2.7. Cobertura

En el sistema de recomendación, la cobertura de los elementos se refiere a la proporción de elementos recomendados respecto al total de elementos [26]. La cobertura de los elementos se define de la siguiente manera:

𝑐𝑜𝑣𝑒𝑟𝑎𝑔𝑒=⋃ 𝐼(𝑢) 𝑢∈𝑈 𝐼 (13)

donde 𝐼(𝑢) es el número de elementos recomendados para un usuario, 𝐼 representa el número total de elementos. La cobertura es una métrica de evaluación importante de los sistemas de recomendación porque puede describir la capacidad de explotación del rango de elementos recomendados.

Un buen sistema de recomendación requiere no solo una alta precisión de predicción sino también una alta cobertura.

3.3.1. Diversidad

En los sistemas de recomendación, la definición de la diversidad de elementos es justo lo contrario a la de la similitud. En algunas situaciones, no tiene sentido recomendar elementos similares a los usuarios en la práctica. Por ejemplo, si el usuario ya ha comprado un reloj deportivo con función de GPS, por lo general, no estará interesado en otros relojes deportivos similares de otras marcas recomendados por el sistema de recomendación. Pero si el sistema de recomendación puede recomendarle otros elementos, como el cinturón de frecuencia cardíaca o un CD de música deportiva, puede obtener resultados de recomendación mucho mejores. Por lo tanto, al diseñar un sistema de recomendación, no solo debe preocuparse por la precisión de la predicción, sino también por la diversidad de los productos recomendados, para satisfacer diferentes requisitos de los usuarios.

3.3.3. Novedad

La novedad del sistema de recomendación significa recomendar elementos con los que los usuarios no están familiarizados. En una aplicación que necesita recomendar elementos novedosos, el método más obvio y fácil de realizar es filtrar los elementos que el usuario ya ha comprado o calificado. Sin embargo, en muchas situaciones, los usuarios no le dirán al sistema de recomendación que ya conocen todos estos elementos. Este método más simple no puede filtrar todos los elementos que los usuarios ya conocen de manera eficiente. Otro método para mejorar la novedad es utilizando la popularidad promedio de los elementos recomendados, lo que significa que cuanto más popular sea un elemento recomendado, menos novedad tendrá y recomendar algunos elementos menos populares hará que los usuarios sientan novedad. En la actualidad, la novedad del sistema de recomendación se puede adquirir a través de encuestas a los usuarios. Un sistema de recomendación comercial debe equilibrar la precisión de predicción, la diversidad y la novedad en la recomendación.

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

[**https://www.shaped.ai/blog/evaluating-recommendation-systems-part-1**](https://www.shaped.ai/blog/evaluating-recommendation-systems-part-1)

Si alguien ha visto anteriormente películas como: Avengers, Top Gun y Star Wars, probablemente disfrutará más de las películas de la primera lista. Asumimos esto porque nuestra comprensión previa de las películas (por ejemplo, género, reparto, escenario) nos permite evaluar qué lista es más similar a la lista de observación histórica. Sin embargo, imagine que conoce a la mayoría de las personas que anteriormente vieron las mismas películas y que también disfrutaron de las películas de la segunda lista de recomendaciones, podría concluir que la segunda lista de recomendaciones es en realidad más relevante. Estas dos formas de decidir la relevancia son lo que los sistemas de recomendación pretenden aprender de sus datos [1].

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

[**https://medium.com/@karina.condeixa/building-a-ranking-based-recommendations-system-e3adeb2c00f7**](https://medium.com/@karina.condeixa/building-a-ranking-based-recommendations-system-e3adeb2c00f7)

La evaluación fuera de línea de los sistemas de recomendación es uno de los criterios más críticos para comparar diferentes algoritmos y seleccionar el modelo de mejor rendimiento para implementar en producción.

# ¿Cómo evaluar?

Como los usuarios solo ven los elementos recomendados en el top k, los sistemas de recomendación generalmente se evalúan utilizando métricas *de alto nivel* que asignan puntuaciones más altas a los modelos que clasifican la mayoría de los elementos relevantes en el top k. Una propiedad muy pesada de una métrica implica que la métrica otorga puntuaciones más altas a un algoritmo que funciona muy bien en los primeros elementos y descarta (o le da menor importancia) las puntuaciones de otros elementos.

----------------------------------------------------------------------------------

[¿Cómo medir el éxito de un sistema de recomendación? (analyticsindiamag.com)](https://analyticsindiamag.com/how-to-measure-the-success-of-a-recommendation-system/)

Cualquier modelo predictivo o sistema de recomendación, sin excepción, depende en gran medida de los datos. Hacen recomendaciones confiables basadas en los hechos que tienen. Es natural que los mejores sistemas de recomendación provengan de organizaciones con grandes volúmenes de datos, como Google, Amazon, Netflix o Spotify. Para detectar puntos en común y sugerir elementos, los buenos sistemas de recomendación evalúan los datos de los elementos y los datos de comportamiento del cliente. El aprendizaje automático se nutre de los datos; Cuantos más datos tenga el sistema, mejores serán los resultados.

Los datos cambian constantemente, al igual que las preferencias de los usuarios, y su negocio cambia constantemente. Es mucha información nueva. ¿Podrá su algoritmo mantenerse al día con los cambios? Por supuesto, las recomendaciones en tiempo real basadas en los datos más recientes son posibles, pero también son más difíciles de mantener. El procesamiento por lotes, por otro lado, es más fácil de administrar, pero no refleja los cambios recientes en los datos.

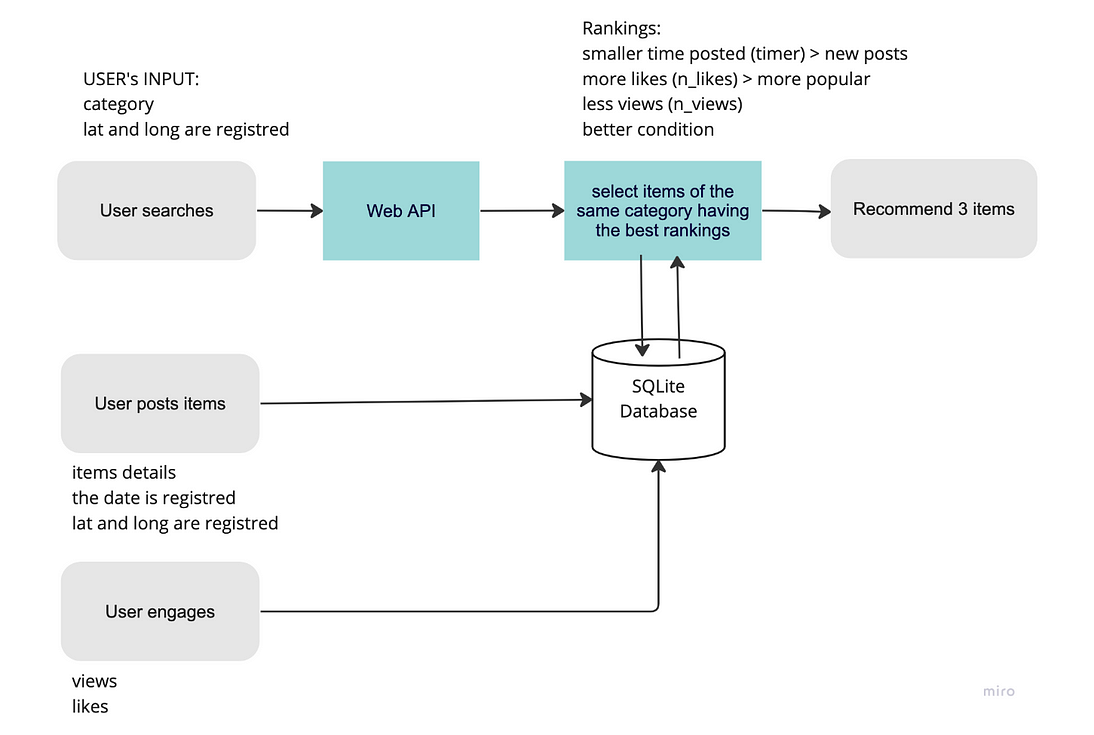
El sistema de recomendación debería seguir mejorando a medida que pasa el tiempo. Las técnicas de aprendizaje automático ayudan al sistema a "aprender" los patrones, pero el sistema aún requiere instrucciones para dar resultados adecuados. Debes mejorarlo y asegurarte de que cualquier ajuste que hagas continúe acercándote a tu objetivo comercial.

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------[**https://medium.com/@karina.condeixa/building-a-ranking-based-recommendations-system-e3adeb2c00f7**](https://medium.com/@karina.condeixa/building-a-ranking-based-recommendations-system-e3adeb2c00f7)

## Flujo de datos

El flujo de datos se proporciona de la siguiente manera: la categoría del elemento buscado por el usuario y los puntos de geolocalización se guardan (usando la API PythonAnywhere y la API de Google Maps). Cuando los usuarios publican información sobre artículos que encontraron o donaron, ingresan la condición del artículo y la fecha y los puntos de geolocalización se almacenan en una base de datos SQLite. Las interacciones del usuario, como ver publicaciones y dar me gusta a los elementos, se rastrean en la base de datos mediante contadores. El sistema de recomendación utiliza los datos de la API de Google Maps como variables y consulta la base de datos mediante SQL para clasificar los elementos, devolviendo las tres recomendaciones principales.

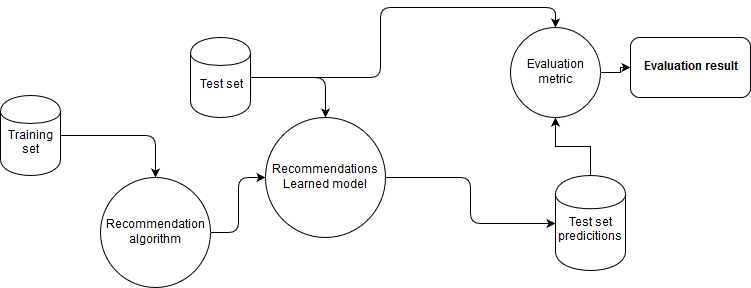


Se puede habalr del timestamp y si esto puede influir en el entrenmiento de los modelos dada la temporalidad. Tal vez según cuando voy a reocmendar peliculas elijo registros mas o menos antiguos.

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

<https://medium.com/@m_n_malaeb/recall-and-precision-at-k-for-recommender-systems-618483226c54>

- Flujo de trabajo de los sistemas de recomendación:  


Primero, se alimenta un conjunto de entrenamiento a un algoritmo de recomendación que produce un modelo de recomendación que puede usarse para generar nuevas predicciones. Para evaluar el modelo, se envía un conjunto de pruebas retenido al modelo aprendido donde se generan predicciones para cada par de usuario-elemento. **Luego, las predicciones con etiquetas conocidas (valor real)** se utilizan como entrada al algoritmo de evaluación para producir resultados de evaluación.

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

<https://medium.com/decathlondigital/personalization-strategy-and-recommendation-systems-at-d%C3%A9cathlon-canada-d9cb3d37f675>

Para lineas futuras usar tal ve RNN

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

<https://medium.com/decathlondigital/building-a-rnn-recommendation-engine-with-tensorflow-505644aa9ff3>

Para lineas futuras:

*Nuestro enfoque de aprendizaje profundo está inspirado en*[*técnicas recientes de generación de texto*](https://www.tensorflow.org/tutorials/text/text_generation)*que utilizan modelos RNN capaces de generar palabras posteriores a partir de una frase determinada (secuencia de palabras). Para nuestro problema de recomendación, reemplazamos la secuencia de palabras con una secuencia de elementos y aplicamos la misma metodología (la siguiente figura muestra la analogía de la PNL y la Recomendación).*

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

<https://medium.com/@rishabhbhatia315/recommendation-system-evaluation-metrics-3f6739288870>

Las métricas de evaluación para los sistemas de recomendación han evolucionado; Inicialmente, la precisión de las calificaciones previstas se utilizó como métrica de evaluación para los sistemas de recomendación. Netflix incluso inició una competencia para encontrar el algoritmo que pudiera mejorar su precisión en un 10 por ciento por un millón de dólares. Pero luego se dieron cuenta de que la precisión de las predicciones no tenía relación con si alguien estaría interesado en ver la película predicha, ya que una persona quiere ver películas nuevas que le gustaría ver, por lo que se debe colocar una lista de las n mejores películas al frente. de usuarios y deberíamos medir cómo reaccionan ante estas películas recomendadas. En este blog, me gustaría explicar estas métricas de precisión en detalle y ayudarlo a comprender cada una de ellas. También se mostrará el código detrás de ellas, que brindará una breve descripción general del código y cómo

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

<https://neptune.ai/blog/recommender-systems-metrics>

El poder de las recomendaciones de productos

Para aprovechar RS al máximo y mejorar la experiencia del usuario, debemos comprender y profundizar en las relaciones entre:

Usuario y producto : cuando el usuario tiene preferencia por productos específicos. Por ejemplo, a un usuario de Netflix le pueden gustar los thrillers mientras que a otro usuario le pueden gustar las comedias.

Producto y producto : cuando los artículos son similares. Por ejemplo, música o películas del mismo género.

Usuario y usuaria – cuando los usuarios tienen gustos iguales o diferentes respecto a un artículo específico. Por ejemplo, los adolescentes pueden diferir de los adultos en cuanto al contenido que consumen.

Tener en cuenta estas relaciones al diseñar un RS generará una experiencia agradable para los usuarios y, en consecuencia, aumentará su compromiso con dichos productos. Imaginemos YouTube sin videos recomendados que te gusten. ¡La mayoría de nosotros pasamos mucho tiempo allí simplemente porque las recomendaciones son muy precisas!

Las estrategias detrás de los sistemas de recomendación

Para seleccionar la mejor estrategia para dichos sistemas, primero debemos evaluar la cantidad de datos disponibles sobre usuarios y productos. A continuación se muestran algunas estrategias populares, ordenadas según la cantidad de datos necesarios en orden creciente:

1

Global: ofrecer al usuario los productos populares, de tendencia o comprados con más frecuencia. Pueden ser relevantes para cualquier usuario.

2

Contextual: basándose en los atributos del producto y los artículos comprados juntos, se pueden combinar con atributos básicos del usuario, como la geolocalización, y se pueden utilizar para dirigirse a un grupo.

3

Personalizado: requiere no solo datos de contexto sino también el comportamiento del usuario, como el historial de compras, clics, etc.

Estas estrategias deben combinarse unas encima de otras para fortalecer el desempeño de RS. Por ejemplo, las plataformas de compras online deben conocer el contexto del producto, así como el historial de compras de los usuarios. Si bien la estrategia "Ver juntos" solo será posible para un nuevo usuario, para los clientes antiguos la estrategia "Comprar juntos" es más adecuada.

----------------------------------------------------------------------------------

Dado estas metricas de avalución de modelos de recoemndacion:  
**3. Hit Rate**

Generate the top n recommendation for a user and compare them to those the user has rated, liked or watched. If they match then increase the hit rate by 1, do this for the complete training set to get the hit rate.

We are taking the threshold and the predicted rating and first, we are filtering values based on the threshold. After that, we sort the values in descending order and returning the top-n values predicted for each user using this function.

To train our model we could use the entire dataset as our training dataset and use the entire dataset as our testing dataset but this won’t be useful as you generally don’t want to evaluate a model with the data it was trained with. As we could just predict the top-n list a user was trained with and it would give a hit rate of 100 percent, think about it. So we can use leave one out cross validation method where we remove one data from each user's top-n training data list and use that data to evaluate our model. If the generated top-n list contains the data left out from the training set then we consider that as a hit. Look at the code for more explanation

4**.Cumulative hit rate**

It is similar to the hit rate shown above but we add a threshold for the test set so that we test on the movies that the person actually likes.

**5.Average Reciprocal Hit Rate**

This metric is also similar to the hit rate. But for each user for which we have a hit, we sum the reciprocal of the rank of the matched movieId in the top-n list and the left out predictions in the top n list for each user and divide by the number of users. This is a user-focused metric as people tend to concentrate on more what they see at the beginning of top-n lists. So we are giving more weight to those hits which are showing up at the top of the list.

**6. Rating hit Rate**

We calculate the rating score for each rating for finding which type of rating is getting more hits. We sum the number of hits for each type of rating in our top-N list and divide by the total number of movies of each rating in our top-N list.

**7.User Coverage**

In this metric, we try to find good recommendations in our top n list by setting a threshold which allows only good recommendations for each user in the top-n list and summing them and dividing by the number of users.

**8. Diversity**

This is calculated by first finding similarity for a set of users and then subtracting it from 1 to find the diversity. We calculate similarity for all the combinations of users and sum them and then divide by the number of combinations. To calculate similarity we need the inner ids of the users to find the similarity between the users, as the surprise library in python uses them for indexing similarity scores.

**9. Novelty**

We will take in a handy dictionary of popularity rankings of movies and go through each user's top-n list to find the average of all the popularity rankings for the movies predicted for that user.

Quiero que me las adaptes, modifiques o uses para un sistema de recoemndación que dado un para usuario-pelicula me calcule estas metricas