# Automatización del Test de Bechdel para el análisis de la representación femenina en guiones de cine y creación de un modelo predictivo basado en las características de las películas.

## INTRODUCCIÓN

El test de Bechdel, también conocido como test de Bechdel/Wallace o the rule, es un método para evaluar si un guion de película, serie, cómic u otra representación artística cumple con los estándares mínimos para evitar la brecha de género. Se originó en el cómic Unas lesbianas de cuidado (en inglés Dykes to Watch Out For o DTWOF), obra de Alison Bechdel. Su invención se atribuye a Liz Wallace, una amiga de la autora del cómic.

En la tira cómica «The Rule», uno de los personajes dice que ella únicamente acepta ver una película si cumple con los siguientes requisitos:

- Aparecen al menos dos personajes femeninos.
- Estos personajes hablan una a la otra en algún momento.
- Esta conversación trata de algo distinto a un hombre (no limitado a relaciones románticas, por ejemplo dos hermanas hablando de su padre no supera el test).

Una variante exige que, además, las dos mujeres sean personajes con nombre.

A pesar de que, a priori, estas restricciones parecen laxas, y se puede presuponer que la mayor parte de las películas las cumplen, es sorprendente la gran cantidad de películas famosas que no pasan el test, poniendo de manifiesto la escasa (o en ocasiones nula) representación femenina que hay en el cine.

## OBJETIVO DEL PROYECTO

La forma que había hasta ahora de determinar si una película pasa el test de Bechdel, era viéndola, cosa que, teniendo en cuenta la enorme cantidad de películas existentes a día de hoy, se antoja imposible de realizar con todas ellas.

De esta idea parte el proyecto, cuyo objetivo es automatizar este test, para que sea posible clasificar una película en base a su representación femenina sin necesidad de verla. Para cumplir este objetivo, se han seguido dos aproximaciones distintas:

- Por un lado, se ha creado un algoritmo en Python que tiene como entrada los guiones (en inglés) de las películas, y como salida un 0 o un 1 en función de si una película no pasa o si pasa el test.
- Por otro, se han aplicado técnicas de Machine Learning, utilizando algoritmos de clasificación, para identificar si una película pasa el test o no, en base a sus características.

# DESARROLLO

El desarrollo de este TFM ha seguido la estructura tradicional de los proyectos de Data Science, siendo este: extracción de los datos, limpieza de los datos, aplicación de algoritmos (en este caso tanto un algoritmo propio, creado específicamente para este trabajo, como de algoritmos de Machine Learning), validación de los algoritmos y extracción de resultados.

Para explicar el proceso seguido, iré desarrollando en los siguientes subapartados cada uno de estos pasos.

## 1. Extracción de los datos

La extracción de los datos para este proyecto se ha llevado a cabo directamente desde la web, debido a los datos que eran necesarios para conseguirlos.

### Obtención de los datos básicos

Para poder aplicar el algoritmo clasificación que utiliza los guiones de las películas para determinar si una película pasa el test de Bechdel o no, era necesario contar, por un lado, con una dataset que contuviera el mayor número de guiones posibles, y por otro, era necesario tener otro dataset con la etiqueta de si una película en concreto pasa el test de Bechdel o no. Para obtener estos datos, se utilizó la extensión de Chrome Web Scraper para extraer todos los guiones contenidos en la página web http://www.imsdb.com/, obteniendo un total de 1107 guiones; esta misma herramienta de Chrome fue utilizada para extraer el dataset con los títulos de todas las películas contenidas https://bechdeltest.com/, etiquetados por los usuarios de la página en base a si pasan el test de Bechdel o no, obteniendo un total de 7678 películas etiquetadas.

### Obtención de información sobre el género de los personajes

Para conseguir información sobre el género de los personajes presentes en cada uno de los guiones que se han analizado, se realizó una nueva extracción de datos con Web Scraper, también del sitio web https://bechdeltest.com/, en esta ocasión para extraer la URL de la página en https://www.imdb.com/ de cada película, donde está contenida toda la información relativa a una película. Esta URL fue procesada en nuestro código para que estuviera direccionada a la página en https://www.imdb.com/ que contiene una tabla con el elenco de la película, de cara a poder guardar esta información para utilizarla posteriormente como entrada del código de R creado para este TFM *results_for_scrapping.R,* que utiliza el paquete rvest para hacer la extracción del elenco de cada película (contiene el nombre del personaje y el actor/actriz que lo interpreta), guardando un csv por película con esta información.

### Obtención de datos para el análisis de las intervenciones femeninas

Para hacer el análisis de las intervenciones femeninas en los guiones, se han extraído dos listas de nombres con Web Scraper, desde la web http://www.randomnames.com, una lista que contiene los nombres masculinos en inglés más comunes (un total de 1804), y otra que contiene los nombres femeninos en inglés más comunes (un total de 2288).

Con este fin, también se ha extraído un listado de las profesiones más comunes en inglés, utilizando Web Scraper, de la página web http://www.lingolex.com.

### Obtención del género de las películas analizadas

Para extraer el género de cada una de las películas que han sido analizadas aplicando algoritmos de Machine Learning, se ha utilizado un código de Python creado específicamente para este fin (*Genre_of_film.ipynb*) que utiliza las librerías request y bs4, haciendo scraping de la web https://www.imdb.com/.

## 2. Limpieza de los datos

La parte más significativa de la limpieza de los datos ha sido la limpieza y estructuración de los guiones en un formato estándar que pudiera ser utilizado por el algoritmo desarrollado con este fin. Los guiones extraídos en bruto estaban en formato de texto, pero para poder realizar un análisis correcto era necesario estructurarlos como intervenciones de personajes, eliminando todas las partes relativas a descripciones de escenas, quedándome sólo con los diálogos. Se han aplicado diversas limpiezas del texto de los guiones, basadas en las reglas de estilo que tienen que cumplir este tipo de documentos: las descripciones de las escenas forman párrafos en sí mimas, así como las intervenciones de los personajes, que tienen la estructura *nombre del personaje que habla (en mayúscula)-salto de línea-frases que dice (separadas por saltos de línea).* Es decir, si se separa el texto por párrafos, y el guión está escrito siguiendo la estructura correcta, se obtendrán las intervenciones de los personajes, con el nombre del personaje en mayúscula, separadas de las descripciones de escena (que en este caso no nos interesan). Aplicando esta idea, el resultado final de este código de limpieza es un diccionario cuyas claves son los títulos de las películas analizadas, y cuyo valor es un dataframe que contiene las intervenciones de los personajes de dos columnas (primera columna: nombre del personaje, segunda columna: frases que recita). Las intervenciones siguen el mismo orden que en el guión original, para poder detectar diálogos entre personajes.

En el caso ideal de que todas los guiones analizados cumplieran estas normas de estilo, el diccionario resultante tendría 588 guiones; sin embargo, debido a que no todos los guiones seguían esta estructura estándar, ha sido necesario realizar una segunda limpieza: primero, se han eliminado aquellas películas cuyo dataframe de intervenciones tenía una longitud de menos de 10 intervenciones (se ha asumido para esto que no hay ninguna película en la que los personajes hablen menos de 10 veces en total, pensando que esta longitud se debía a una incorrecta limpieza de los datos en la limpieza previa). Por otro lado, viendo la distribución de la cantidad de intervenciones en todas las películas, se ha visto que en el 75% de las películas, la cantidad de intervenciones es inferior a 115. Explorando detenidamente aquellas películas que superaban este número, se ha visto que, debido a que sus guiones no cumplían las reglas de formato, la limpieza previa no se ha realizado correctamente, existiendo dataframes en los que estaban incluidas descripciones de escenas (probablemente debido a que éstas no estaban separadas de las intervenciones por cambio de párrafo, si no sólo por saltos de línea). Para evitar problemas en el análisis posterior, se han seleccionado sólo aquellas películas con menos de 115 intervenciones.

Tras esta limpieza, me quedo con un total de 677 guiones bien procesados.

## 3. Algoritmo de análisis de los guiones

### Identificación de intervenciones femeninas

En el apartado anterior se ha hablado de la limpieza de los guiones de forma genérica, para que tengan una estructura que permita ser analizados. Sin embargo, una parte fundamental del algoritmo desarrollado se basa en identificar únicamente aquellas intervenciones de personajes femeninos en nuestros guiones, ya que son las que nos van a dar información sobre si una película pasa el test o no.

Para llevar a cabo esto, se han utilizado los csv obtenidos con el código de R explicado anteriormente, que contiene los nombres de los personajes relacionados con el nombre de la actriz u actor que lo interpretan. Estos csv, tras procesarlos para seleccionar sólo aquellos personajes femeninos en base al nombre de la actriz que lo interpreta, han sido guardados en un diccionario cuyas claves son los títulos de las películas y cuyos valores son los dataframes resultantes de este procesamiento.

Posteriormente, se han seleccionado sólo aquellas películas de las que, por un lado, se tiene información sobre su elenco, y por otro, se tiene su guión correctamente procesado, obteniendo un total de 319 películas, que serán analizadas por el algoritmo desarrollado.

El siguiente paso en esta limpieza ha sido seleccionar sólo aquellas intervenciones femeninas de nuestro guión, identificando si cada personaje que aparece en nuestro dataframe de intervenciones, aparece también en el dataframe que tiene listados los nombres de los personajes femeninos.

Por último, se ha realizado un último paso, ya que, al haber hecho la selección de intervenciones femeninas en base al nombre de la actriz que lo interpreta, se ha visto que no todas las intervenciones identificadas tras esta limpieza cumplen la condición de que los personajes femeninos tengan nombre propio: identifica, por un lado, personajes masculinos (esto es debido a que hay nombres de actores que son identificados como femeninos, como es el caso de Alex o Chris), también identifica personajes que no tienen nombre propio (sólo identificados por su profesión, o por su relación con otro personaje 'friend', 'mum', 'mother'.

Lo que se ha hecho en este paso ha sido enriquecer la lista de nombres femeninos con nombres de personajes contenidos en los daraframes de intervenciones femeninas (hay que tener en cuenta que hay personajes femeninos con nombres inventado para la película), eliminando aquellos nombres de personajes masculinos mal identificados como femeninos en el paso anterior y aquellos personajes femeninos sin nombre propio (identificados sólo por su profesión o por su relación con otro personaje). Esta lista se utilizará en el algoritmo desarrollado para identificar correctamente a aquellos personajes femeninos que aparecen en un guión, y que tienen nombre propio.

**Aplicación del algoritmo desarrollado**

Llegados a este punto, ya tenemos un diccionario que contiene 319 dataframes, uno por cada película analizada, que contiene todas las intervenciones de personajes femeninos de la película.

El algoritmo desarrollado se basa en comprobar si estos dataframes cumplen las tres condiciones del test de Bechdel:

1. Hay más de un personaje femenino con nombre propio en la película. Para ello, se sirve de la lista de nombres femeninos enriquecida en el paso previo, con la que identifica estos personajes y los cuenta. Si no se cumple esta condición, esa película

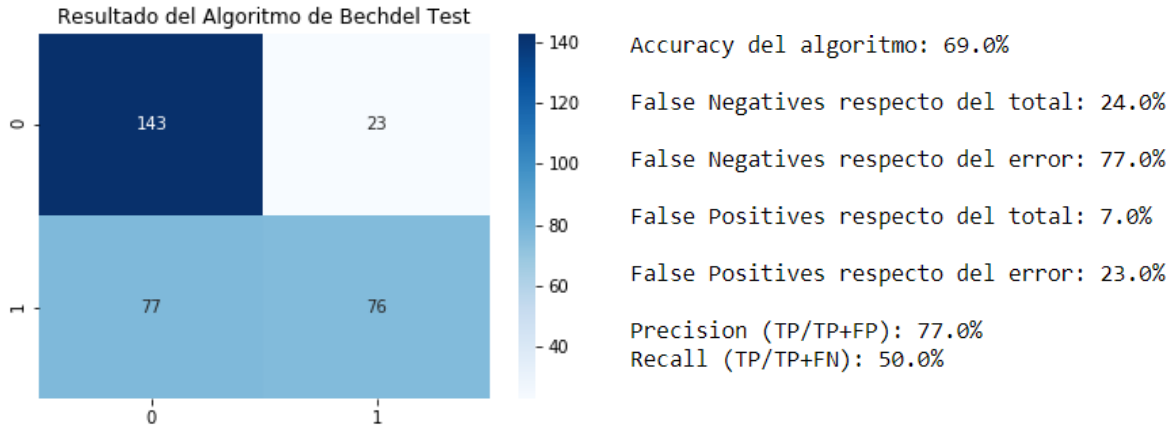
pasa a ser identificada etiquetada como 0 (no cumple el test). Si cumple la condición, la película pasa al siguiente paso del algoritmo.

2. Estos personajes femeninos hablan entre sí. Para ello, teniendo en cuenta que el índice del dataframe que contiene las intervenciones femeninas no ha sido modificado tras eliminar las intervenciones masculinas, es posible identificar si dos intervenciones femeninas son consecutivas en el guión, y por tanto, constituyen un diálogo entre personajes. También hay que indicar que el algoritmo discrimina entre intervenciones consecutivas del mismo personaje (lo que no constituiría un diálogo), o de personajes femeninos diferentes (lo que sí que constituiría un diálogo). Además, el algoritmo no tiene en cuenta sólo dos intervenciones consecutivas, sino que es capaz de identificar un diálogo completo, contenga el número de intervenciones que contenga, entre dos personajes femeninos, basándose en la consecutividad de los índices. Las películas que no cumplen la condición de tener al menos un diálogo entre dos personajes femeninos son etiquetadas como 0. Si la cumplen, pasan al siguiente paso del algoritmo.
3. La conversación trata de algo que no sea un hombre. Para este último paso, se ha enriquecido la lista de nombres masculinos con términos relativos a un personaje masculino, para identificar si el diálogo entre personajes femeninos versa sobre un hombre o no ('him', 'his', 'brother', 'father', 'husband', 'he', etc). Se comprueba si en las palabras dichas por los personajes aparece alguna de estas palabras (utilizando nltk para comprobar sólo los nombres propios, los nombres comunes y las preposiciones). En el caso de que aparezca alguna de ellas, se pasa a comprobar el siguiente diálogo, y en el caso de que no aparezca, se identifica la película como 1 (la película ha cumplido las tres condiciones). Si en todos los diálogos aparece alguno de estos términos o algún nombre masculino, la película se etiqueta como 0, ya que no cumpliría la última condición.

Una vez aplicado el algoritmo a todas las películas que se están analizando, se obtiene un dataframe que contiene el nombre de la película y resultado tras el análisis.

**Análisis de resultados**

Para analizar los resultados, se comprueba el accuracy del resultado del algoritmo respecto a la etiqueta real extraída de la web, además de hacer una matriz de confusión. También se mira precision y recall. Los resultados obtenidos son los siguientes:




```
Accuracy del algoritmo: 69.0%

False Negatives respecto del total: 24.0%

False Negatives respecto del error: 77.0%

False Positives respecto del total: 7.0%

False Positives respecto del error: 23.0%

Precision (TP/TP+FP): 77.0%
Recall (TP/TP+FN): 50.0%
```

**Conlusiones**

Como podemos ver, falla un 30% de las veces, lo que supone un resultado interesante, aunque no especialmente bueno. Se observa también que la mayoría de los fallos (casi un 80%), los realiza al etiquetar como 0 una película cuando es positiva. Esto implica que el algoritmo no detecta bien las tres condiciones, pudiendo hallarse el error en cualquiera de ellas: no selecciona correctamente todos los personajes femeninos, no identifica bien los diálogos, o no es capaz de identificar correctamente de qué versan los diálogos. Debido a la cantidad de procesamiento previo realizado, y a que las listas han sido creadas a mano, mi conclusión es inclinarme hacia la identificación de personajes femeninos, que puede que no se esté haciendo todo lo bien que se podría. Este sería un punto interesante de mejora, conseguir una mejor precisión del algoritmo a la hora de identificar los personajes femeninos con nombre propio.

En cuanto al análisis de precision y recall, podemos ver que el algoritmo tiene un valor significativo de precision (de las películas que etiqueta como positivas, acierta un 77% de las veces), mientras que el recall es bastante mediocre (sólo acierta un 50% de las películas reales que pasan el test).

## 4. Clasificación aplicando algoritmos de Machine Learning

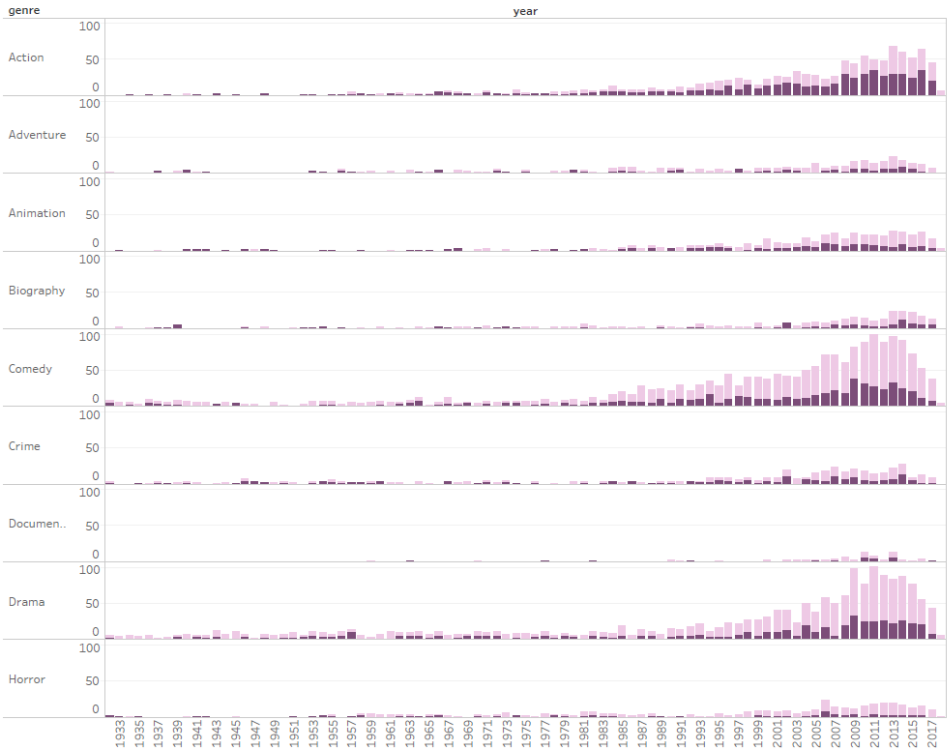
La siguiente parte de este TFM se basa en realizar la misma clasificación que en el apartado 3, pero en lugar de hacerlo mediante el análisis de guiones, se busca tratar de predecir si una película pasará el test o no en base a sus características.

En este apartado, vamos a utilizar, a priori, las 7678 películas etiquetadas que hemos extraído de la web https://bechdeltest.com/ , aunque, como veremos más adelante, va a ser necesario eliminar algunas de ellas de nuestro análisis por diversos motivos, salvo aquellas cuyo género no ha sido extraído correctamente con el código de *Genre_of_Films.ipynb*. Nos quedamos, finalmente, con un total de 7411 películas para realizar nuestro análisis.

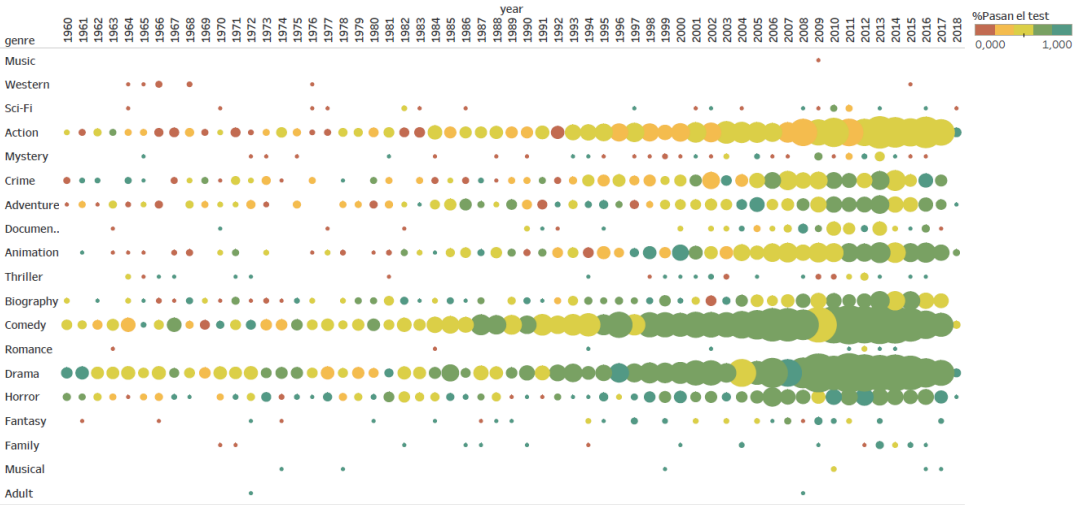
**Selección de Características**

Las características seleccionadas han sido Género y Año. Esto es debido porque, aparentemente, estas características podrían contener la información necesaria para determinar si una película pasa el test de Bechdel o no, como sugieren estas gráficas obtenidas con Tableau:

Evolución temporal (géneros más significativos)




Porcentaje de películas que pasan el test por año y género




Estas gráficas nos permiten ver varias cosas:

- Por un lado, los géneros de las películas que vamos a predecir son: Music, Western, Sci-Fi, Action, Mystery, Crime, Adventure, Documental, Animation, Thriller, Biography, Comedy, Romance, Drama, Horror, Fantasy, Family, Musical, Adult. Se puede ver, en cuanto a la distribución de los datos, que no tenemos el mismo número de películas por cada género, siendo los géneros que más películas tienen Action, Comedy y Drama. Esta limitación es a la hora de interpretar los resultados obtenidos en este apartado. También respecto a Género se observa, además, que acertadamente está seleccionada como característica, ya que podemos observar que el porcentaje de películas que pasan el test es diferente dependiendo del género (vemos que en Action, apenas el 50% de las

películas lo pasan, mientras que en Comedy y en Drama, este porcentaje aumenta hasta el 75%).

- Por otro lado, vemos que estamos analizando películas desde 1933 hasta 2018. También en este caso la distribución de películas es muy desigual, concentrándose las películas en la franja entre el 2000 y el 2018, lo que también habrá que tener en cuenta a la hora de analizar los resultados. También en este caso se ve una diferencia en el porcentaje de películas que pasan el test entre las distintas décadas, acumulándose mayor porcentaje a partir de la década de los 2000, por lo que aparentemente esta característica también tiene sentido que haya sido seleccionada.

Dicho esto, me gustaría recalcar que soy consciente de la limitación y el sesgo del dataset utilizado, lo que probablemente se traducirá en pobres resultados tras la aplicación de los modelos de clasificación; sin embargo, puesto que el objetivo principal del TFM es demostrar el conocimiento de las técnicas y no obtener un buen resultado en sí, se han obviado estas limitaciones y se ha realizado en análisis con los datos de los que dispongo.

Debido a que ambas características son variables categóricas, se planteó la pregunta de cómo introducirlas en el modelo. En un primer momento, se pensó en transformarlas en variables dummies, lo que aumentaba la cantidad de características muchísimo, y luego aplicar un PCA. Tras ver los resultados obtenidos mediante esta estrategia de selección de variables, finalmente opté por otra: convertir cada variable categórica en una probabilidad de aparecer en el dataset, quedándome así solo con dos características (Año y Género), cuyos valores eran continuos (entre el 0 y el 1, dependiendo de la probabilidad que tuviera esa variable de aparecer en el dataset).

**Preparación de Train, Test y Validación**

Para validación, se han seleccionado las mismas películas utilizadas en el apartado 3, de cara a poder comprobar cuál de las dos estrategias es mejor para identificar si una película pasa el test o no. Las películas restantes (7092) se han dividido en una proporción de 25% para el test y 75% para el train.

**Selección del modelo**

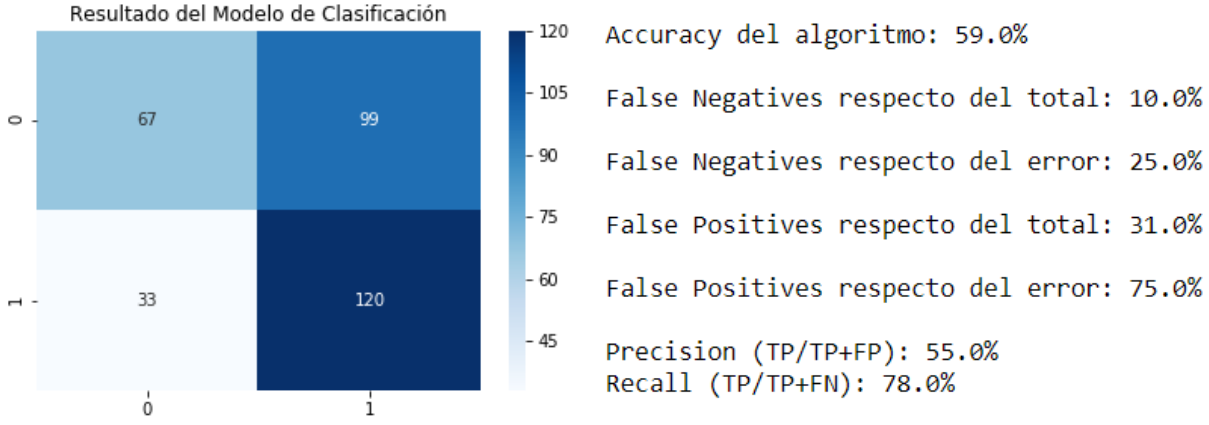
- La selección del modelo de clasificación que mejores resultados obtiene se ha realizado utilizando los siguientes modelos: xgboost, RandomForestClassifier, KNeighborsClassifier y Gaussian Naïve Bayes. En aquellos modelos en los que era preciso, se ha hecho una selección de hiperparámetros utilizando GridSearchCV. Todos los modelos han sido entrenados con los datos de train, y comprobados con los datos de test. La métrica utilizada para comprobar la validez de los modelos ha sido *accuracy*, ya que en este caso nos interesa cómo de preciso es un modelo a la hora de etiquetar una película, independientemente de que la etiquete como positiva o como negativa. Los resultados obtenidos han sido los siguientes (cabe destacar que, debido a la componente aleatoria de algunos de estos modelos, existe la posibilidad de que, al ejecutar el código, estos resultados varíen; sin embargo, en general se mantienen más o menos constantes).
- ***xgboost***:
    - Hiperparámetros seleccionados: 'max_depth': 5, 'eta': 0.5, 'silent': 1, 'objective': 'binary:logistic', 'eval_metric':'error'
    - Accuracy: 0.6142131979695431

- ***RandomForestClassifier:***
    - Hiperparámetros seleccionados: n_estimators=18, max_depth=3.
    - Accuracy: 0.61590524534686975
- ***KNeighborsClassifier:***
    - Hiperparámetros seleccionados: n_neighbors= 23
    - Accuracy: 0.60180485053581501
- ***GaussianNaiveBayes:***
    - Accuracy: 0.61872532430908067

En general, a pesar de que en estos resultados se sugiera lo contrario, el modelo que mejores resultados ha obtenido siempre (en todas las iteraciones que se han llevado a cabo a lo largo del desarrollo) ha sido ***RandomForestClassifier***, por lo que ha sido el modelo seleccionado para la validación del resultado.

**Validación del resultado**

Aplicando el modelo entrenado de RandomForestClassifier con los hiperparámetros arriba indicados a los datos de validación, se obtienen los siguientes resultados:




**Conclusiones**

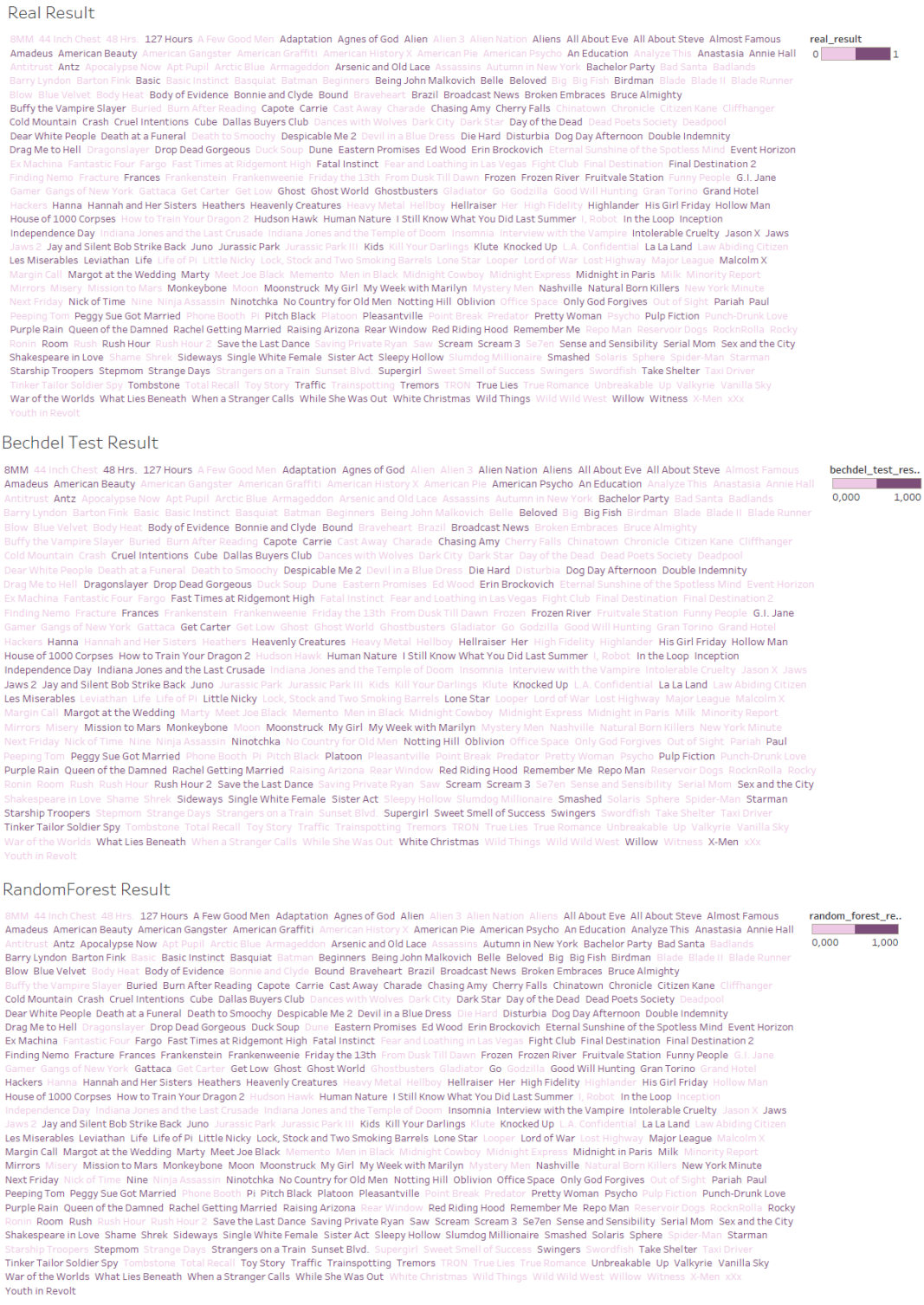
Como podemos observar, la precisión del algoritmo es poco significativa, ya que acierta poco más de la mitad de las veces (60%). Cabe destacar que, en este caso, en oposición a lo que sucedía en el apartado 3 con el algoritmo de análisis de guiones, lo que mejor identifica este modelo son las películas que no cumplen el test de Bechdel (Negatives), constituyendo únicamente el 25% del error que comete el modelo. Si nos fijamos en precision, vemos que, por cada película que el modelo etiqueta como 1, falla etiquetando otra que debería ser 0. En cuanto al recall, sin embargo, sí que se ve que acierta etiquetando como positivas el 78% de las películas que lo son en realidad.

Es evidente que el modelo no es bueno, ya que tiene muy poca precisión a la hora de etiquetar las películas. Esto, sin duda, se debe a las limitaciones mencionadas anteriormente. Como propuesta de mejora, si se contara con una base de datos mayor, con una distribución igual de las películas en lo relativo a año y a género, podrían mejorarse los resultados. También podrían mejorarse añadiendo más características sobre las películas (director, guionistas, productora,

etc), ya que, si bien las características utilizadas parecen prometedoras, parece que no explican totalmente la variable dependiente.

## 5. Representación de los resultados totales del proyecto.

Como último apartado de esta memoria, he decidido utilizar una visualización generada con Tableau para poder ver visualmente las diferencias entre las clasificaciones hechas mediante el algoritmo de análisis de guiones y el modelo de Machine Learning.



Como podemos ver, la representación visual concuerda con los resultados explicados antes: mientras que el algoritmo de análisis de guiones se queda “corto” a la hora de identificar películas que sí pasan el test (se observa una significativa disminución en las películas consideradas como 1 respecto de la imagen que contiene la clasificación real), el modelo predictivo de Machine Learning se pasa a la hora de identificar como positivas películas que no lo son (hay muchas más películas en oscuro, es decir, clasificadas como 1, de las que hay en la visualización de la clasificación real).