ESCUELA TÉCNICA SUPERIOR DE INGENIEROS

INFORMÁTICOS

UNIVERSIDAD POLITÉCNICA DE MADRID

Valoración de Películas Basada en Análisis de Redes Sociales

TRABAJO FIN DE MÁSTER

MÁSTER EN INTELIGENCIA ARTIFICIAL

AUTOR: Diego Martín Sanz

TUTORES: Emilio Serrano Fernández y Jacinto González Pachón

2016/2017

AGRADECIMIENTOS

Me gustaría agradecer a mis tutores, Emilio Serrano Fernández y Jacinto González Pachón todo el apoyo y dirección ofrecida para la realización de este proyecto, así como el hecho de haberme brindado la oportunidad de adentrarme en el interesante mundo del aprendizaje automático.


POLITÉCNICA


Departamento Inteligencia Artificial

Dedicado a vosotros M. M. S y A. S. M, mis padres, y especialmente a tí B. P. M., por haberme acompañado en esta andadura de manera intachable.


POLITÉCNICA


Departamento Inteligencia Artificial

RESUMEN

En este trabajo de fin de máster se estudiará la aplicación del Análisis de Redes Sociales (SNA) para conseguir predecir la valoración de películas en IMDb, antes de que éstas aparezcan en cartelera y basándonos para ello únicamente en los grafos que definen la estructura de red social de las mismas, donde un nodo es un personaje y una arista une dos personajes que interaccionan entre sí.

Se intentará predecir si la valoración de una película será positiva o negativa (considerando una valoración > 7 como positiva y una valoración <= 7 como negativa) y se intentará conseguir una predicción de las valoraciones en 5 categorías más precisas, siendo conscientes no obstante de la dificultad que presenta este segundo objetivo. Para ello aplicaremos métodos de clasificación típicos de la minería de datos, como son las implementaciones de Weka de *J48, Random Forests, Adaboost, NaiveBayes y OneR.*

En los resultados obtenidos se ha conseguido hasta casi un 74% de valor predictivo positivo (*precision*) en la predicción de películas con valoraciones positivas (mayores que 7). Sin embargo, no se ha encontrado correlación en los datos para conseguir predicciones de valoraciones de películas bajo el planteamiento de clasificación multi-clase, lo cual era de esperar, dada la desafiante naturaleza del problema.

Se han definido nuevos e innovadores conceptos basándonos en métricas de SNA como han sido el "Número de Protagonistas", "Popularidad de los Protagonistas" y "Relevancia de los Protagonistas", y hemos comprobado que con ellos se ha mejorado la calidad de los datos. Para ello hemos creado 4 definiciones de lo que podría ser un protagonista de una película, basándonos también en métricas de SNA.

La desafiante naturaleza de los objetivos planteados, unido a las contribuciones que este trabajo va a aportar al conocimiento de la materia, justifican con creces la realización de este trabajo.

SUMMARY

In this master's degree essay, it will be studied the application of Social Network Analysis (SNA) to aim to predict the movie ratings in IMDb, before they appear on the billboard and based solely on the graphs that define the movies social network structure, where a node is a character and an edge joins two characters that interact with each other.

It is our purpose to predict whether a movie rating will be positive or negative (considering a rating > 7 as positive and a rating <= 7 as negative) and to achieve a rating prediction in 5 more precise categories, being conscious of the difficulty of this second objective. We will apply classification methods typical of data mining, such as the Weka implementations of J48, Random Forests, Adaboost, NaiveBayes and OneR.

Up to almost 74% of accuracy has been achieved in predicting movies with positive ratings (greater than 7). However, no correlation was found in the data to obtain movie rating predictions under the multi-class classification approach, which was to be expected given the challenging nature of the problem.

New and innovative concepts have been defined based on SNA metrics such as "Number of Protagonists", "Popularity of Protagonists" and "Relevance of Protagonists", and we have verified that they have improved the dataframe quality. We have created 4 definitions of what could be a movie protagonist, also based on SNA metrics

The challenging nature of the objectives set, together with the contributions that this work will bring to the knowledge of the subject, further justify the accomplishment of this work.

TABLA DE CONTENIDOS

LISTADO DE FIGURAS

LISTADO DE TABLAS

1. INTRODUCCIÓN Y OBJETIVOS

El análisis de redes sociales es un paradigma de investigación basado en el uso de redes y teoría de grafos que actualmente se aplica a un gran número de disciplinas. Algunas de estas aplicaciones incluyen análisis de comportamiento de clientes, estudio de la propagación de rumores, sistemas de recomendación o detección de células terroristas.

En esta tesis de fin de máster se estudiará la aplicación del Análisis de Redes Sociales (SNA) para intentar predecir la valoración de películas en IMDb, antes incluso de que éstas aparezcan en cartelera, basándonos únicamente en los grafos que definen la estructura de red social de las mismas, donde un nodo es un personaje y una arista une dos personajes que interaccionan entre sí.

El objetivo que se persigue con la realización de este trabajo es el de intentar predecir si la valoración de una película será positiva o negativa (considerando una valoración > 7 como positiva y una valoración <= 7 como negativa). También se intentará predecir las valoraciones en 5 categorías más precisas, siendo conscientes no obstante de la dificultad que presenta este segundo objetivo.

Los resultados que hemos conseguido han sido de hasta casi un 74% de valor predictivo positivo (*precision)* en la predicción de películas con valoraciones positivas (mayores que 7), si bien es cierto que para conseguir esta precisión se sacrifica mucha sensibilidad (*recall*). Sin embargo, no se ha encontrado correlación en los datos para conseguir predicciones de valoraciones de películas bajo el planteamiento de clasificación multi-clase, lo cual era de esperar, dada la desafiante naturaleza del problema. Esto se ha conseguido mediante la aplicación de métodos de clasificación típicos de la minería de datos, como son las implementaciones de Weka [1] de *J48, Random Forests, Adaboost, NaiveBayes y OneR.*

De las contribuciones más interesantes que se han conseguido, se encuentran las siguientes:

1. Se han utilizado únicamente métricas de SNA correspondientes al grafo de estructura de red social de películas, para predecir las valoraciones de las mismas.
2. En algunos casos se han conseguido predicciones de valoraciones de películas antes de que se haya estrenado, publicado trailers y de que se hayan hecho comentarios sobre ellas en los medios sociales.
3. Se han creado 4 definiciones de Protagonista de una película, basadas en métricas de SNA.
4. Se han definido nuevos atributos, como son el Número de Protagonistas, la Popularidad de un Personaje y la Relevancia de un Personaje, en función de métricas de SNA.

La desafiante naturaleza de los objetivos planteados, unido a las contribuciones que este trabajo va a aportar al conocimiento de la materia, justifican con creces la realización de este trabajo.

El documento está estructurado en varias secciones. La primera de ellas es la sección en la que nos encontramos. En la segunda sección se recoge un estudio de los trabajos realizados durante los últimos años sobre el tema que nos aborda, dando así una visión del ámbito de esta temática y se detalla la metodología que se ha seguido para la elaboración de este proyecto. La sección tres, recoge una explicación de los experimentos realizados y se analizan los resultados obtenidos. Por último, se establecen las conclusiones del trabajo realizado, se listan las contribuciones aportadas al conocimiento y se proponen los puntos de acción que deberían ser abordados en un futuro trabajo.

2. DESARROLLO

En esta sección se recoge un estudio de los trabajos realizados durante los últimos años sobre el tema que nos aborda, dando así una visión del ámbito de esta temática, que es la predicción de las puntuaciones de películas, basado en el SNA y aplicando técnicas de minería de datos, proporcionando así una visión global de dónde nos encontramos ahora mismo, desde el punto de vista de la investigación en esta temática.

Además, se detalla la metodología que se ha seguido para la elaboración de este proyecto y de describen las diferentes técnicas con las que se ha experimentado y de las que hemos evaluado sus bondades y desventajas.

2.1. ESTADO DEL ARTE

En esta subsección, se presenta el estudio previo de los trabajos realizados hasta el momento sobre la temática principal del TFM, que es la predicción de las puntuaciones de películas, basado en el SNA y aplicando técnicas de minería de datos.

De esa forma, se presentan trabajos realizados sobre predicción de diferentes aspectos de una película, como pueden ser su calificación en IMDb [2], su éxito social [3], su popularidad [4], sus precios [5] o su éxito en los "*Academy Awards*" [6]. Además, se incluyen otros trabajos que han aplicado otras técnicas como son el SNA [5] y técnicas de *Sentiment Analysis* [6].

Se seguirá una estructura común para cada una de las subsecciones siguientes, siguiendo ésta el siguiente orden:

- ***Keywords*:**
  - Incluirá las palabras clave del trabajo estudiado.
- **Metodología:**
  - Técnicas y metodologías aplicadas en el trabajo en cuestión.
- **Resultados:**

  - Colección de los resultados obtenidos en los experimentos que se han llevado a cabo en cada uno de los trabajos.
- **Contribución de la Tesis**
  - Descripción de la aportación del TFM al trabajo en cuestión. Se dará una visión de cómo esta tesis mejora el estudio analizado, y qué contribución se va a hacer al conocimiento en esa materia.

2.1.1. Predicción de la Calificación de Películas en IMDb

Esta sección contiene el análisis realizado sobre la predicción de la calificación de películas en IMDb, utilizando los medios sociales [2].

Keywords:

Social media, predicting, movie rating, linear regression, WEKA, IMDB, YouTube, Twitter

Metodología:

Colección de Datos:

- Los autores se han hecho con una colección de datos de 70 películas obtenidos de IMDb [7]:
  - 10 para extracción de características textuales.
  - 60 para *testing.*
- Además, utilizan Datos de YouTube (metadatos y comentarios de los trailers) + Datos de Twitter (1,6 tweets en el año 2011)

Métodos Utilizados:

Para predecir los *ratings* de las películas utilizan *regression:*

- *Linear Regression* en WEKA sobre una colección de datos [8], [9].
- *10-fold cross validation* para calcular los resultados de la regresión.

Medidas de Regresión:

- Spearman´s ρ: Cuanto más alto sea el valor de ρ, mejor.
- MAE (*Mean Absolute Error*) y RMSE (*Root Squared Mean Error*): Cuanto más bajos sean, mejor.

Resultados

Las características que han provocado los mejores resultados de predicción, han sido el coeficiente "*likes*/*dislikes*" de los trailers en *YouTube*, combinado con las características textuales obtenidas de *Twitter*.

Contribución de la Tesis

Analizando los resultados de este trabajo, parece que finalmente los datos más concluyentes que se utilizan para "predecir" el rating de la película, son datos aportados por la gente después de haber visto el tráiler de la misma, con lo que no parece que se realice una "predicción" en el sentido completo de la palabra.

La contribución que aportará este TFM con respecto a este trabajo, es que se pretende realmente predecir el rating de una película, únicamente a partir de la red que define la interacción social en la propia película (relaciones entre personajes, información sobre actores, etc..), antes incluso de que se hayan visualizado trailers y de que se hayan analizado opiniones sociales sobre la misma.

Además, se pretende simplificar el problema definiendo unos umbrales que nos ayuden a predecir una clasificación, en vez de una regresión como se plantea en este trabajo.

2.1.2. Aplicación de la Minería de Datos para la Calificación de Películas

Esta sección contiene el análisis realizado sobre la utilización de Minería de Datos en la construcción de aplicaciones para la calificación de películas [10].

Keywords:

Data mining, movie classification, film rating, movie rating, decision tree J48, T-Score, mid-points, WEKA, IMDb

Metodología:

1. Realizan un trabajo previo que consiste en lo siguiente:
    a. *Movie Rating:* se centran en la clasificación de películas como:
        - **G**: *General Audience*
        - **PG**: *Parental Guidance Suggested*
        - **PG-13**: *Parents Strongly Cautioned*
        - **R**: *Restricted*

    b. *Data Mining*: 3 pasos/fases [11] . En la Figura 1, se detallan los pasos seguidos en el proceso de Data Mining.

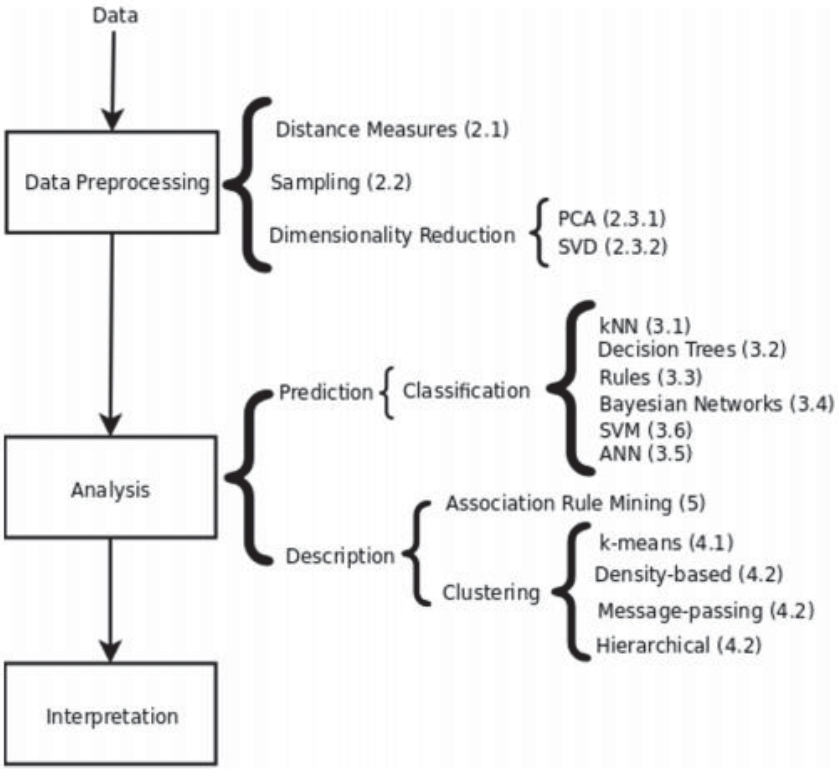

Figura 1. Pasos y métodos en un proceso de Data Mining.

- **Pre-procesado de datos**: esta es la fase más importante
  - Análisis estadístico (*correlation*) para analizar las relaciones entre los atributos.
  - Realizan "Limpiado", "Filtrado" y "Modificaciones" de los datos.

- **Análisis de datos**: las técnicas dependerán del objetivo que se tenga, esto es, clasificación, *clustering*, *association rule mining*, etc… Ellos se centran en el *clustering*.

- **Interpretación de los resultados**: Para cuantificar la calidad del clasificador, utilizan las siguientes medidas:

  - Porcentaje de Predicciones Correctas *(Accuracy)*:

$$\frac{(TP + TN)}{(TP + TN + FP + FN)}$$

  - Valor Predictivo Positivo *(Precision)*:

$$\frac{TP}{(TP + FP)}$$

  - Sensibilidad *(Recall)*:

$$\frac{TP}{(TP + FN)}$$

  - Medida F *(F-measure)*:

$$\frac{2TP}{(2TP + FN + FP)}$$

, siendo:

- **TP** (*True Positive*)
- **TN** (*True Negative*)
- **FP** (*False Positive*)
- **FN** (*False Negative*)

2. Selección de los Atributos. Métodos Utilizados:
    - *Decision tree (J48)* de WEKA [9].
    - *4-fold cross validation* para dividir los datos en "datos de entrenamiento" y "datos de testing" [9].
    - Extraen los siguientes atributos de los datos obtenidos de IMDB:
        - Actores
        - Actrices
        - Directores
        - Productores
        - Escritores
        - Presupuesto
        - Recaudación
        - Género
        - Fecha
        - Argumento

    - Del argumento, extraen una clasificación de palabras en las siguientes categorías:
        - Malas palabras
        - Sexual
        - Terror

- Religión
- Violencia
- Drogas

- Utilizan *T-Score* y *mid-points* de WEKA, para "*levelize*" los grupos de palabras.

- Selección de Atributos: Utilizan WEKA, "*decision tree J48*" + "*cross validation*" [9], y obtienen como características del árbol de decision J48:
  - Género - Familiar, animación, fantasía, thriller, drama, crimen, acción, comedia…
  - Los Grupos de Palabras - Malas palabras, sexual, terror, drogas.

Resultados

En la Tabla 1, se muestra una comparativa de los resultados obtenidos con ambos enfoques. En la Tabla 2, se detallan los resultados de "*precision*", "*recall*" y "*F-measure*" obtenidos con el modelo "*mid-point*".

Level approaches	Weka Selected Attributes		Correctness from 240 movies			
	Selected Genres	Selected word groups	Correct		Incorrect	
Mid-point	Family Animation Drama Sci-fi Comedy Romance	Bad word Terror	175	73%	65	27%
T-Score	Family Animation Fantasy Thriller Drama Crime Action Comedy	Bad word Sexual Terror Drug	161	67%	79	33%

Tabla 1. Comparativa de ambos enfoques con Datos de Entrenamiento.

Rating	Precision (%)	Recall (%)	F-measure (%)
PG	88.7%	78.8%	83.4%
PG-13	57.5%	76.3%	65.6%
R	81%	63.8%	71.3%
Average	**73%**	**76%**	**80%**

Tabla 2. Precision, Recall, F-Measure para Datos de Test.

Contribución de la Tesis

En este trabajo se plantea la clasificación de películas mediante la aplicación de técnicas como los árboles de decisión *J48* de WEKA, lo cual es de interés para aplicarlo en la tesis en la que se está trabajando, de cara a seleccionar los atributos

más influyentes de la colección de datos. Sin embargo, con la realización de este TFM no se pretende únicamente clasificar películas, sino que lo que se pretende es predecir qué valoración van a tener.

Este es el valor que aporta este TFM respecto de este trabajo, el utilizar métodos de clasificación para generar una predicción.

Además, nos parecen interesantes las métricas de calidad que utilizan (todas ellas métricas de calidad de clasificación), por lo que se aplicarán en este TFM, ya que son las más utilizadas en problemas de clasificación.

Por último, otro aspecto de interés es la clasificación de películas que definen, esto es, G, PG, PG-13 y R. Se va a extrapolar esta definición a las necesidades de este TFM y se definirán una serie de clases para clasificar la predicción de las películas. Un ejemplo de la clasificación que se realizará podría ser el siguiente:

- Valoración > 8
- Valoración < 6
- 6 <= Valoración <= 8

Donde Valoración $\in$ [0,10]

Esto no sería más que un ejemplo, pues esta definición de clases dependerá del conjunto de datos con el que se cuente finalmente, pues habrá que definir una clasificación que sea acorde con estos datos. La definición de clase deberá estar balanceada.

2.1.3. Predicción del Éxito de una Película basado en datos de IMDb

Esta sección contiene el análisis realizado sobre la predicción del éxito de una película, basado en datos de IMDb. [3]

Keywords:

Data mining, Logistic Regression, SVM Regression, Linear Regression, IMDB

Metodología:

En la Figura 2, se detalla el diseño general de la metodología seguida en este trabajo.

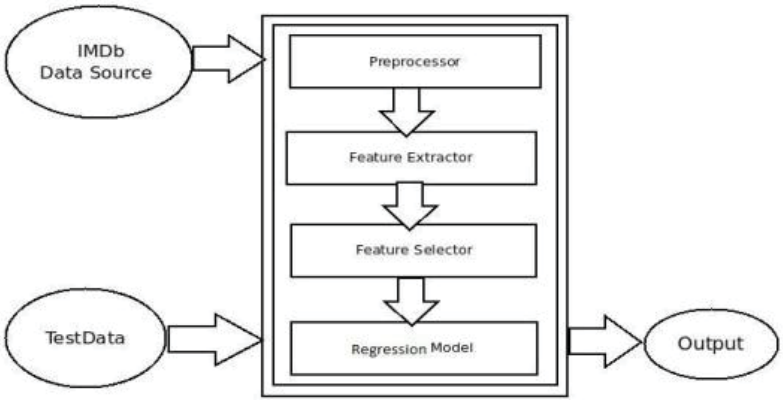

Figura 2. Diseño General de la Metodología Seguida.

- Obtienen datos sobre 1050 películas de IMDB, *Rotten Tomatoes* y Wikipedia.
- Siguen el siguiente procedimiento:
    1. Pre-procesado de datos [12, pp. 85-124].
    2. Integración y transformación de los datos.
    3. Selección de Atributos:

       Utilizan "*Supervised Learning*" y los siguientes modelos de predicción:

        - *Linear Regression Model*
        - *Logistic Regression Model*
        - *SVM Regression Model*

Resultados

No obtienen buenos resultados, ya que la mejor exactitud que consiguen en sus predicciones es del 50.7% con "*Linear Regression*". A continuación, se muestran los mismos en la Tabla 3:

Model	Linear regression	Logistic regression	SVM regression
Tolerance	20 %	12.5%	20%
Success Rate	50.7%	42.2%	39.0%
Correlation	0.965		0.965

Tabla 3. Tabla de Resultados.

Contribución de la Tesis

En este trabajo se detallan conclusiones que no han sido contrastadas adecuadamente. Los autores presentan resultados que no son comparables, ya que están mezclando técnicas de clasificación, como son *Logistic Regression* y *SVM*, con técnicas de regresión, como es *Linear Regression*. Además de que estas técnicas se utilizan para fines diferentes (las técnicas de clasificación predicen categorías -discretas-, la regresión se utiliza para predecir datos numéricos [12, pp. 15-23] están utilizando las mismas medidas de calidad para evaluar los resultados (% de exactitud), en lo que también discrepamos, ya que según [8, p. 417], la medida de calidad para clasificación es el % de precisión, mientras que la medida para la regresión suele ser el error cuadrático medio, medida de exactitud (también se suele usar la correlación como medida de calidad de la regresión).

La contribución que hará esta tesis con respecto a este trabajo, será el de evaluar adecuadamente las medidas para cada técnica aplicada, esto es, medidas de precisión para técnicas de clasificación y medidas de exactitud para técnicas de

regresión. No obstante, la intención es aplicar técnicas de clasificación al problema a resolver.

2.1.4. Predicción de la Popularidad de Películas mediante técnicas de Machine Learning

Esta sección contiene el análisis realizado sobre el uso de técnicas de "*Machine Learning*" para predecir la popularidad de las películas [4].

Keywords:

Machine learning, classification, movies, logistic, IMDB

Metodología:

Han creado un conjunto de datos y los han transformado. Después han aplicado diferentes enfoques de "*machine learning*" para construir modelos eficientes que puedan predecir la popularidad de las películas.

La Figura 3, muestra la metodología seguida:

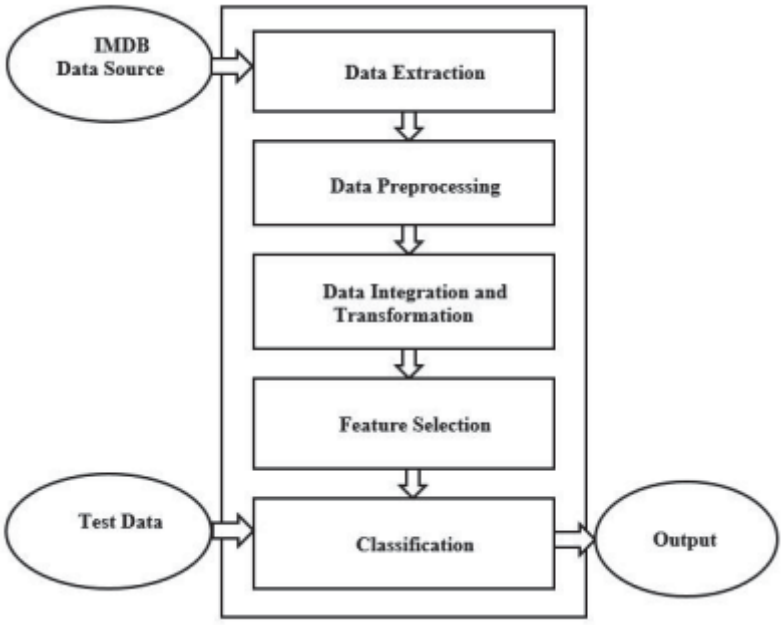

Figura 3. Diseño General de la Metodología Seguida.

1. Extracción de Datos:
   - Extraen datos de IMDB con un script en python y C# y lo almacenan en una BBDD MySQL.
   - Consiguen un Dataset de 2000 elementos.

2. Pre-procesado de Datos:
   - Realizan un "limpiado" y optimización de los datos. [3].
3. Integración y Transformación de Datos [13], [14]:
   a. En la siguiente tabla (Tabla 4) se muestran las variables que han utilizado en sus experimentos:

Name of Variable	No. of Values	Values
Rating	4	Terrible, Poor, Average, Excellent
MPAA Rating	5	R, PG, PG-13, G, NR
	20	Action, adventure, thriller, biography, crime, drama, horror, comedy, fantasy, animation, mystery, music, war ,documentary, romance, Sci-fi, Westren, Family, sport,short
Awards	4	Oscar Won, Oscar Nominee, Golden Globe won, Golden globe nominee
Screens	1	Positive integers
Opening weekend	9	1,2,3,4,5,6,7,8,9
Metascore	1	Positive integers
Number of votes	1	Positive integers
Budget	9	1,2,3,4,5,6,7,8,9

Tabla 4. Resumen de Variables Independientes.

4. Selección de Atributos

Utilizan "*Information Gain*" [12, pp. 327-442]

5. Clasificación:
   - "*Supervised Learning*".
   - Han experimentado con los siguientes clasificadores de WEKA:
     - *Logistic regression.*
     - *Simple Logistic.*
     - *Multilayer perceptron.*
     - *J48.*
     - *Naive bayes.*
     - *PART.*
   - Cada uno de ellos testeado con "*10-fold cross validation*".

Resultados:

Los mejores resultados los obtienen con "*simple logistic*", "*logistic regression*" y "*J48*", con una precisión del 84,34%, 84,15% y 82,42%, respectivamente. Ver Figura 4:

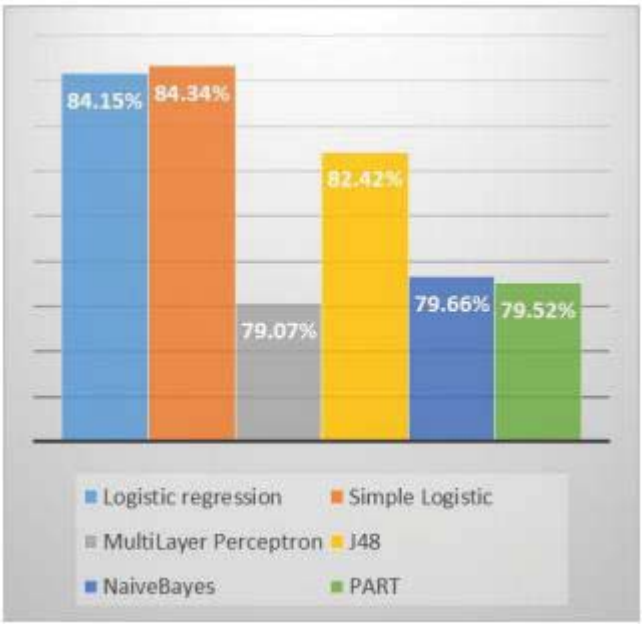

Figura 4. Resultados de la Clasificación.

Contribución de la Tesis

De este trabajo es interesante la definición de clases que han llevado a cabo, para cada una de las variables evaluadas. Como ya se ha comentado en secciones anteriores, se extrapolará esta idea para definir las clases a predecir en esta tesis.

Además, la contribución que aporta estas tesis con respecto a este trabajo, es que se intentará predecir una característica más tangible como es la valoración de la película en IMDb y no tan abstracta como es la popularidad de la película.

2.1.5. Predicción del Precio de Películas mediante SNA

Esta sección contiene el análisis realizado sobre la aplicación del Análisis de Redes Sociales y *Sentiment Analysis* para la predicción del precio de las películas [5].

Lo que se pretende en este trabajo, es predecir la recaudación en taquilla, en las 4 semanas siguientes del estreno, que producirían nuevas películas. Para ello se basan en la predicción del precio de las películas en el *Hollywood Stock Exchange* [15], que es un mercado de predicción de los ingresos brutos que producirá una película, y predice el ratio entre el ingreso bruto de la película y el presupuesto de producción de la misma.

Los autores intentan clasificar las películas en 3 grupos, dependiendo de si la recaudación de la película con respecto al coste de producción de la misma es:

- Menor.
- Un poco más alto.

- Mucho más alto.

Keywords:

Dynamic social network analysis, trend prediction, movie analysis

Metodología:

1) Obtener y formatear datos. Para extraer estos datos, utilizan diferentes métricas:

   a. *Web Metrics.*

   b. *SNA Metrics.*

      Hacen uso de la intermediación (*betweenness centrality*) de varios conceptos. La intermediación de un concepto en una red social se define como una aproximación de su influencia en la discusión en general. "Monitoriza el número de caminos geodésicos a través de toda la red, que pasan por el concepto del que se está evaluando su influencia" [16]. En otras palabras, los conceptos con alta intermediación, actúan como porteros/guardas entre diferentes dominios.

   c. *Sentiment Metrics.*

      Hacen uso de los siguientes algoritmos: "*Basic Sentiment Algorithm*" y "*Dynamic Adaptation of Bag-of-Words*".

2) Modelado de los datos:

   Una vez que han recopilado los datos, los autores desarrollan un modelo para predecir los futuros precios en HSX. Para ello, consideran cinco enfoques diferentes:

a. *Direction Prediction:*

Realizan una breve investigación sobre si el precio de la película subirá o bajará al día siguiente. No consiguen cuantificar la magnitud de este cambio, por lo que este método no les es muy útil para poder concluir decisiones valiosas.

b. *Linear Regression:*

En su primer intento por predecir la magnitud de la acción de una película, los autores utilizan *linear regression* para predecir el precio de la acción al día siguiente. Para evaluarlo, comparan las predicciones hechas, contra los precios actuales de las acciones en esos días. Especialmente evalúan el "Error Medio", la "Desviación Estándar del Error", el "Error Cuadrático Medio", la "Correlación Media" y la "Desviación Estándar de la correlación".

Obtienen resultados con errores razonablemente bajos, indicando que pueden ser capaces de predecir los cambios diarios en los precios de las acciones. En la Tabla 5 se muestran los resultados que obtienen:

IV Name	*Modeling-Days*	*Prediction-Days*	Mean Error	Error Standard Deviation	Mean Squared Error	Correlation Coefficient Mean	Correlation Coefficient Standard Deviation
Number of theaters showing film	8	2	0.006315	3.380257	10.387	0.765246	0.1445
IMDb votes for score of six	2	5	0.136708	3.348881	10.533	0.428571	0.937614
Percentage of IMDb votes for score of three	8	14	-1.366	3.134408	10.989	-0.46962	0.521188
IMDb mean score	3	11	-1.575	3.155722	11.609	-0.652	0.500621
IMDb votes for score of five	2	4	-0.2703	3.530512	11.759	0.428571	0.937614
Percentage of IMDb votes for score of three	9	14	-1.93602	2.972594	11.905	-0.50907	0.519679
IMDb weighted average score	3	12	0.1785	3.685474	12.709	-0.29391	0.756806
IMDb votes for score of six	2	8	-0.97831	3.556334	12.761	-0.07692	1.037749
IMDb mean score	4	11	-1.52655	3.359891	12.813	-0.63204	0.356732
IMDb votes for score of five	2	8	-0.17524	3.706572	12.853	0.384615	0.960769

Tabla 5. Top 10 de los resultados de predicción con linear regression, ordenados por error cuadrático medio.

Como el objetivo es predecir el precio de cierre final de la acción en el HSX, los autores estudian otros métodos aplicando regresión a los datos.

c. *Multilinear and Non-Linear Regression:*

Los planteamientos aplicando *linear regression*, les proporciona unos fundamentos muy básicos, por lo que sospechan que la combinación de varias variables independientes de manera conjunta, producirán mejores resultados de predicción.

Los autores no presentan resultados de la aplicación de esta metodología, por lo que sus suposiciones de mejora en los resultados de predicción, en nuestra opinión se quedan en eso, en una suposición que no llegan a concluir.

d. Slope of Smoothed SNA and Sentiment Variables:

Los autores creen que este enfoque debería proporcionar percepciones especialmente significativas sobre el precio de una película. Será otro planteamiento que tratarán de testear en un futuro.

e. Classifying Movies as Successes or Flops Based on Gross-to-Production-Budget Ratios:

En su último modelo, los autores categorizan el éxito de las películas en función del ratio entre los ingresos generados y los costes de producción. Así, definen 3 grupos de categorización:

- Grupo I: películas con ratio < 1. Estas películas se consideran un fracaso, ya que ni siquiera recuperan la inversión inicial.
- Grupo II: películas con 1 <= ratio <= 2. Películas que al menos recuperan la inversión inicial.
- Grupo III: películas con ratio > 2. Se consideran un éxito en taquilla, teniendo que en cuenta que prácticamente han doblado la inversión inicial.

En primer lugar, clasifican cada una de las 30 películas con valores de intermediación y "*sentiment*" tomados diariamente entre las 2 semanas antes del estreno y 1 semana después del estreno.

Después, calculan el producto de los valores de la intermediación y el "*sentiment*" de cada día y los suman (valores positivos de "*sentiment*" son números positivos y valores negativos de "*sentiment*" son números negativos). La suma neta representa el sentimiento general sobre la película, una semana después de su lanzamiento.

A continuación, toman el valor absoluto y el registro de las sumas para convertirlos en puntuaciones de "peso-del-sentimiento" de escala comparable. Suponen que estas puntuaciones se distribuyen uniformemente en cada categoría, siguiendo una distribución normal.

Utilizando una política "*leave-one-out*", pasan por cada uno de los 30 conjuntos de datos (es decir, películas) y dejan uno fuera cada vez, para *checking*. Con los conjuntos que quedan, calculan la media y la desviación estándar de las puntuaciones para cada categoría. Teniendo en cuenta estos cálculos y aplicando el teorema de Bayes en teoría de probabilidad, si *g* es el grupo para el conjunto de datos *x*, entonces $P(g|x) = \frac{P(x|g)P(g)}{P(x)}$, concluyen que las clasificaciones son correctas en un 53% de las veces, que es mejor que una clasificación aleatoria (33%). La siguiente tabla (Tabla 6) muestra con qué tasa de éxito han sido categorizadas las películas.

Group	Percentage of Movies Classified Correctly Into This Group
Group I	72.7%
Group II	16.7%
Group III	85.7%

Tabla 6. Tasa de éxito de clasificación de películas en grupos correctos.

La Tabla 7 es más útil, muestra con qué tasa ha sido clasificada una película como perteneciente a un grupo, estando realmente en ese grupo.

Group Movie was Classified as Being In	Percentage of Those Movies Actually In That Group
Group I	56.5%
Group II	100%
Group III	55.4%

Tabla 7. Porcentajes comparando el número de películas clasificadas correctamente con respecto a falsos positivos.

Por último, la Tabla 8 muestra lo que quizá sea el resultado más valioso, esto es, que una película clasificada como perteneciente al Grupo III sea realmente improbable que resulte pertenecer al Grupo I, y que una película identificada como del Grupo I sea realmente improbable que esté en el Grupo III.

Percentage of Movies Classified as Group III, But Actually In Group II or III	82.4%
Percentage of Movies Classified as Group I, But Actually In Group I or II	88.9%

Tabla 8. Esta tabla muestra que las películas identificadas como éxitos generalmente lo hicieron bien o muy bien, pero rara vez terminaron siendo un fracaso. Del mismo modo, las películas clasificadas como fracasos fueron fracasos, pero rara vez se convirtieron en éxitos.

Resultados

Los resultados preliminares que obtienen, empleando diferentes métodos de predicción (*multilinear regression* y *non-linear regression*), han sido conseguir predecir la recaudación en taquilla generada por las películas, al menos tan bien como el Mercado de Predicción HSX.

Por otro lado, en este trabajo los autores han establecido una base de cómo clasificar películas como "taquillazos" o "fracasos" una semana después de haber sido estrenadas, siendo esta clasificación correcta el 80% de las veces.

Contribución de la Tesis

A pesar de que en la tesis no se van a utilizar métricas como las presentadas en este trabajo (*Web Metrics* y *Sentiment Metrics*) debido a que en la tesis se va a intentar predecir la valoración de una película sin necesidad de evaluar opiniones externas, sí que es de interés el uso que hacen de "*SNA Metrics*". Más concretamente, el uso

que ellos hacen del concepto de intermediación (*betweenness centrality*), nos da la idea de utilización de, no solo la intermediación, sino de muchas otras medidas de centralidad en una red.

La evaluación de la intermediación de un concepto, permite monitorizar la importancia de ese concepto/nodo dentro de la red que define a esa película.

En esta tesis se intentará extraer medidas de centralidad en cada una de las películas, de manera que se pueda añadir a la colección de datos sobre los que aplicaremos técnicas de minería de datos. Las medidas de centralidad que se aplicarán son, entre otras:

- La centralidad de grado (*degree centrality*).
- La cercanía (*closeness*).
- La intermediación (*betweenness centrality*).
- La centralidad de vector propio (*eigenvector centrality*).

Todas estas medidas de centralidad, son medidas a nivel de nodo de una red. En este TFM lo que se pretende es extraer información a nivel de película, por lo que se deberá realizar un tratamiento de los valores de centralidad de los nodos de una red, de manera que se calculen valores medios, máximos y mínimos de cada una de estas métricas. De esta manera, lo que se consigue es generalizar los datos obtenidos de los nodos y extrapolarlos a nivel de película.

Para explicarlo con mayor claridad, en lugar de interpretar los valores de centralidades de todos los nodos de cada una de las redes (una red por cada película), lo que haremos será calcular los valores medios, máximos y mínimos de estas métricas, e incorporar esta información como atributos dentro de nuestro *dataframe* (se incluye un ejemplo de una de las métricas, como es la intermediación):

- Intermediación Media de una Película: lo calcularemos como el valor medio de los valores de intermediación de todos los nodos de la red que representa a esta película.

- Intermediación Máxima de una Película: lo calcularemos como el valor máximo de los valores de intermediación de todos los nodos de la red que representa a esta película.

- Intermediación Mínima de una Película: lo calcularemos como el valor mínimo de los valores de intermediación de todos los nodos de la red que representa a esta película.

Se ampliarán estas definiciones a todas las métricas de SNA que se utilicen. Estas métricas serán definidas en posteriores secciones.

Además, en el trabajo analizado es de un gran interés todo el estudio que han realizado en el apartado de modelado de datos, y más concretamente, en el último de los 5 enfoques "*Classifying Movies as Successes or Flops Based on Gross-to-Production-Budget Ratios*". El análisis que hacen es muy similar al que se va a realizar en esta tesis, en el sentido de que se basan en la clasificación de películas en 3 categorías concretas. Es cierto que no es útil para esta tesis la aportación que hacen al estudio añadiendo datos de "*sentiment*", pero podría ser una aportación muy interesante como línea de trabajo futuro.

Nuestra contribución, a pesar de aplicar metodologías ya conocidas como son las *SNA Metrics*, será la de aplicarlas para conseguir predecir la valoración de una película antes de que se emita en las salas de cine.

Además, esta tesis aporta como contribución adicional el hecho de conseguir predicción **s**in opiniones externas (por ejemplo, las obtenidas de redes sociales). No se trata de explotar la información sobre lo que se habla en una red social (opiniones y comentarios de gente sobre una película), sino que se explotará la estructura social de la película propiamente dicha, es decir, la información que se extraiga de las

relaciones de los personajes entre sí, de las correlaciones que existan entre las diferentes redes sociales que definen a cada película (de la estructura de su grafo).

2.1.6. Predicción del éxito de Películas en los Academy Awards mediante SNA y Sentiment Analysis

Esta sección contiene el análisis realizado sobre la predicción del éxito de las películas en los "*Academy Awards*" mediante *Sentiment Analysis* y *SNA* [6].

El paper presenta un nuevo planteamiento de *Web Mining* que combina el SNA y el análisis del sentimiento automático (*Automatic Sentiment Analysis*).

Keywords:

Trend Prediction, Dynamic Social Network Analysis, Online Forum, Internet Movie Database, Oscar Awards, Condor, TeCFlow, LIWC

Metodología:

- Utilizan dos fuentes de datos, IMDb [7] y Box Office Mojo [17].
- Se basan en dar pesos a los *posts* en IMDb para predecir tendencias y eventos en el mundo del cine.
- Herramientas utilizadas para el análisis de los datos:
    - *Condor*: herramienta de análisis de redes sociales (formalmente llamada *TeCFlow*) [18]. *Condor* crea mapas visuales y muchas métricas gráficas de las relaciones encontradas en las redes sociales, mediante la minería de estructuras de los enlaces en sitios web, foros en línea y redes de correo electrónico. Por ejemplo, *Condor* puede crear vistas gráficas de enlaces estáticos de las comunicaciones entre usuarios en un foro web y calcular el índice de contribución de cada usuario, que proporciona pistas sobre la relevancia e importancia de los usuarios clave que contribuyen a la comunicación.

Los autores hacen uso de 2 características principales de *Condor*:

1) Permite analizar cambios temporales continuos en estructuras comunicativas de una web.
2) Soporta el análisis de contenido de términos que están siendo utilizados en foros de comunicación.

- *LIWC*: Es la versión online de "*Linguistic Inquiry and Word Count*", un software con características para calificar entradas de texto de acuerdo a sus propiedades emocionales [19].

- Primeramente, introducen un modelo para predecir las nominaciones a los Óscar en función de la estructura comunicativa de posts online.
  Después, aplican el mismo enfoque para examinar si hay una correlación entre la estructura comunicativa de los posts y el éxito de las películas en la taquilla. Examinan la estructura comunicativa de los posts (que son tratados por *Condor* como redes sociales) en los que la gente opina sobre las posibles candidaturas a los Óscar, con métricas externas como son índices de intensidad y positividad, que extraen de los comentarios sobre cada una de las películas, y concluyen que los patrones de discusión en IMDb les sirve para predecir:
  - La recaudación en taquilla.
  - Las nominaciones a los Premios Óscar.

Resultados:

Dos meses antes de entregar los Oscar, los autores fueron capaces de predecir 9 nominaciones a los Oscar, del top 10. En la Tabla 9 se muestra una comparativa de los valores obtenidos por el modelo frente a los resultados finales de los Oscar.

Top 10 Oscar Model	Model Value	Actual Result	LIWC Value
Departed	1,00	Oscar	3,85
Take the Lead	0,51	-	6,58
Dreamgirls	0,45	Oscar	7,57
The Queen	0,41	Oscar	6,96
Little Children	0,37	Nomination	3,44
Babel	0,30	Oscar	6,67
Little Miss Sunshine	0,28	Oscar	3,58
United 93	0,24	Nomination	6,26
Borat	0,24	Nomination	5,32
Blood Diamond	0,24	Nomination	2,82

Tabla 9. Valores del Modelo vs. Resultados de los Oscar.

Contribución de la Tesis

Uno de los planteamientos que presentan en este trabajo, es la aplicación de patrones de discusión en IMDb, pero después de que la película haya sido estrenada. El valor que se pretende aportar en esta tesis, es que se calificarán las películas antes de su estreno, sin tener en cuenta la opinión social ni ninguna otra información de tipo sentimental que pueda ser obtenida de cualquier otra fuente, que es lo en lo que se basa este trabajo, en el análisis de la estructura comunicativa de posts, en foros de opinión, que son cuantificados con métricas, como los índices de intensidad y positividad.

2.1.7. Otros trabajos analizados

Otros autores han realizado trabajos sobre la aplicación de técnicas de *Sentiment Analysis* [6] y [5], para aplicarlas a la predicción de diferentes tipos de características de películas.

También se ha analizado la aplicación de técnicas de minería y *Summarization* como en el trabajo de L. Zhuang, F. Jing y X. Zhu [20], un trabajo basado en la identificación de "*feature words*" y "*opinion words*" en una frase,

determinando la clase de "*feature words*" y la polaridad de "*opinion words*", para posteriormente relacionarlas en parejas.

En una primera fase, no se han considerado estos trabajos para el *grosso* de la realización del TFM, aunque se tendrán en cuenta como posible contenido de "Trabajo Futuro", como aportación adicional.

Por último, se ha analizado [21], con la intención de obtener una base de conceptos en cuanto a la diferencia del aprendizaje supervisado y no supervisado.

2.2. METODOLOGÍA Y EVALUACIÓN DE RIESGOS

Esta sección se explica la metodología que se ha seguido en la realización de este trabajo. La metodología que se ha seguido se basa en el proceso denominado *KDD* (*Knowledge Discovery from Data*), que se resume en una serie de pasos según [12]. Esta secuencia de pasos, nos ha servido para definir nuestro propio guión de trabajo:

1. **Obtención de Datos**, fase donde se extraerá la información de diferentes fuentes de datos.

2. **Pre-procesado de Datos**, donde se realizará una limpieza e integración de los datos obtenidos de diferentes fuentes.

3. **Integración y Transformación**, que será la fase en la que consolidaremos los datos en un formato adecuado.

4. **Selección de Datos**, donde se aplicarán métodos de selección de atributos para obtener los datos más relevantes para el análisis.

5. **Minería de Datos**, que será el paso en el que se aplicarán diferentes algoritmos de clasificación para intentar extraer un patrón.

6. **Evaluación e Interpretación de los resultados**, para identificar y analizar las medidas obtenidas y poder concluir con el conocimiento adquirido.

En las siguientes sub-secciones, se detalla el trabajo realizado en cada una de estas fases.

2.2.1. Frameworks Utilizados

En esta sub-sección vamos a presentar los *frameworks* que se han utilizado en la implementación de la metodología presentada.

- *WEKA* [1]: Weka es una colección de algoritmos de aprendizaje automático para tareas de minería de datos.
- *igraph* [22]: Es una colección de herramientas de análisis de redes.
- *NetworkX* [23]: Es un paquete de software escrito en python que se utiliza para la creación, manipulación y estudio de la estructura de redes complejas.
- *IMDbPY* [24]: Es un paquete de python útil para obtener y gestionar datos de la base de datos de películas IMDb [7], tales como información sobre películas, personas, personajes y compañías.

2.2.2. Obtención de Datos

El proceso de obtención de datos, ha consistido principalmente en la extracción de información de dos fuentes de datos diferentes, *IMDb* [7] y *Moviegalaxies* [25]. De estas dos fuentes, se extraerán los siguientes datos:

- Datos referentes a cada Película, como son el año y el país de producción, el presupuesto, el género y personajes principales, entre otros. Para ello se ha hecho uso de la versión 5.1 de *IMDbPY* [24].
- Métricas de SNA, a partir de los ficheros de grafos (en formato "*gexf*") que definen la estructura social de cada película. Para la obtención de estas métricas, en una primera instancia se pensó en utilizar *igraph* [22], sin embargo, finalmente esta opción fue desechada debido a que esta librería, en su versión para Python, no soporta la importación de ficheros de tipo "*gexf*".
  Se decidió utilizar, por tanto, *NetworkX* [23].

Los datos que se utilizarán para realizar el posterior análisis, serán los correspondientes a una colección de más de 750 películas, estructurados en forma de red social, y que han sido obtenidos de *Moviegalaxies* [25], mediante *web scrapping*. *Moviegalaxies* es un sitio web para descubrir los grafos sociales asociados a una película. Crea una nueva forma de experimentar con películas,

construyendo una tecnología, a través de la implementación de Algoritmos Inteligentes y la aplicación de procesamiento de datos, para visualizar películas.

Además, por cada una de las películas extraídas de *Moviegalaxies*, se obtendrán diferentes atributos de IMDb. Para todo ello se han implementado un conjunto de scripts en Python, que hacen uso de diferentes librerías para la interconexión con cada una de nuestras fuentes de datos.

A continuación, se lista la totalidad de atributos extraídos y se añade una breve descripción de su significado.

Datos relativos a la película y obtenidos de IMDb [7]**:**

- **Rating:** Valoración de la película en IMDb.
- **Presupuesto (en dólares).**
- **Año de producción.**
- **Director.**
- **Personajes.**
- **Guionista.**
- **Productor.**
- **País de producción.**
- **Géneros.**
- **Director de vestuario.**
- **Compañía de Sonido.**
- **Editor.**
- **Manager de Producción.**
- **Director de efectos especiales.**
- **Director de fotografía.**
- **Director de maquillaje.**
- **Compañía de Producción.**

Datos relativos a la estructura del grafo y obtenidos de *Moviegalaxies* [25]**:**

Los atributos que se detallan a continuación son métricas de análisis de redes sociales, y para los que se ha seguido la definición implementada en *NetworkX* [23]:

- **Grado:** Grado de cada nodo. Es el número de enlaces adyacentes a ese nodo.
- **Densidad:** Densidad de un grafo. La densidad de un grafo no dirigido se define como

$$d = \frac{2m}{n(n-1)}$$

  Donde *n* es el número de nodos y *m* es el número de enlaces en el grafo.

- **Número de nodos:** Número de nodos en el grafo.
- **Tamaño:** Número de enlaces en el grafo.
- **Excentricidad:** Excentricidad de los nodos en el grafo. La excentricidad de un nodo *n* es la distancia máxima entre *n* y todos los nodos en el grafo.
- **Diámetro:** Diámetro del grafo. El diámetro es la máxima "*excentricidad*". Es la mayor distancia entre todos los pares de nodos del grafo.
- **Radio:** Radio en el grafo. El radio ese la mínima "*excentricidad*".
- **Centralidad de Grado (*degree centrality*):** Centralidad de grado de los nodos. La centralidad de grado de un nodo *n* es el número de nodos a los que *n* está conectado, es decir, corresponde al número de enlaces que *n* tiene con los demás nodos en el grafo.
  Este valor está normalizado por el valor máximo de grado en el grafo.
- **Cercanía (*closeness centrality*):** Centralidad de cercanía de los nodos. La cercanía de un nodo *u*, es el recíproco de la suma de las distancias del camino más corto entre el nodo *u* y todos los demás *n-1* nodos. Como la suma de las distancias depende del número de nodos en el grafo, el valor de cercanía es normalizado por la suma de las mínimas distancias posibles *n-1*.

$$C(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(v,u)}$$

Donde $d(v,u)$ es la distancia del camino más corto entre *v* y *u*, y *n* es el número de nodos en el grafo.

- **Intermediación (*betweenness centrality*):** La intermediación de un nodo *v* es la suma de la fracción de todos los caminos más cortos que pasan a través de *v*. En otras palabras, la intermediación es una medida que cuantifica la frecuencia o el número de veces que un nodo actúa como **puente** en el camino más corto entre otros dos nodos.

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

Donde *V* es el conjunto de nodos, $\sigma(s,t)$ es el número de caminos (s,t) más cortos, y $\sigma(s,t|v)$ es el número de esos caminos que pasan a través de un nodo *v*, distinto a *s*, *t*.

- **Intermediación de Enlace (*edge betweenness centrality*):** La intermediación de un enlace *e* es la suma de la fracción de todos los caminos más cortos que pasan a través de *e*. En otras palabras, la intermediación es una medida que cuantifica la frecuencia o el número de veces que un enlace actúa como **puente** en el camino más corto entre otros dos nodos.

$$c_B(e) = \sum_{s,t \in V} \frac{\sigma(s,t|e)}{\sigma(s,t)}$$

Donde *V* es el conjunto de nodos, $\sigma(s,t)$ es el número de caminos (s,t) más cortos, y $\sigma(s,t|e)$ es el número de esos caminos que pasan a través de un enlace *e*.

- **Centralidad de Vector Propio (*eigenvector centrality*):** La centralidad de vector propio nos da la centralidad para un nodo basada en la centralidad de sus vecinos. La centralidad de vector propio de un nodo *i* es:

$$\mathbf{Ax} = \lambda \mathbf{x}$$

Donde A es la matriz adyacente del grafo G con valor de vector propio λ. Según el teorema de *Perron-Frobenius*, existe una única y positiva solución si λ es el mayor valor de vector propio asociado al vector propio de la matriz adyacente A [26, p. 169].

Esta métrica mide la influencia de un nodo en la red [27].

- **Centralidad de Comunicabilidad (*communicability centrality*):** También conocida como Centralidad de Subgrafo, la centralidad de comunicabilidad de un nodo *n* es la suma de caminos cerrados de todas las longitudes que empiezan y terminan en el nodo *n*.

  Esta métrica se puede calcular utilizando una descomposición espectral de la matriz adyacente [28], [29],

$$SC(u) = \sum_{j=1}^{N} (v_j^u)^2 e^{\lambda_j},$$

  Donde v_j es un vector propio de la matriz adyacente A del grafo G, correspondiente al valor de vector propio λ_j.
- **Intermediación de Comunicabilidad (*communicability betweenness centrality*):** Esta medida hace uso de número de caminos que conectan cada par de nodos como base de una medida de intermediación.
- **Centralidad de Carga (*load centrality*):** La centralidad de carga de un nodo es la fracción de todos los caminos más cortos que pasan a través de ese nodo.
- **Conectividad de un Nodo (*node connectivity*):** Es igual al mínimo número de nodos que deben eliminarse para desconectar un grafo G.
- **Grado de *Asortatividad* (*degree assortativity*):** La *asortatividad* mide la similaridad de las conexiones en el grafo con respecto al grado del nodo.
- **Número de Cliques:** Número de máximos cliques para cada nodo.
- **Triángulos:** Número de triángulos que incluyen un nodo como vértice.
- **Transitividad:** Es la fracción de todos los posibles triángulos presentes en un grafo G.

Los posibles triángulos se identifican por el número de "*triads*" (dos enlaces con un vértice común).

$$T = 3\frac{\#triangles}{\#triads}$$

- **Clustering:** El coeficiente de *clustering* de un nodo *u* es la fracción de posibles triángulos a través de ese nodo,

$$c_u = \frac{2T(u)}{deg(u)(deg(u)-1)}$$

Donde $T(u)$ es el número de triángulos a través del nodo *u* y $\deg(u)$ es el grado de *u*.

- **Longitud del Camino más corto:** Longitud de los caminos más cortos en el grafo.

2.2.3. Pre-Procesado de Datos

Después de la extracción de información de diferentes fuentes de datos, será necesario realizar un pre-procesado de los mismos, pues la mayoría de ellos no se encuentran en un formato directamente manejable. Esta es, por tanto, la fase donde se ha realizado una integración y limpieza de los datos obtenidos de la fase anterior.

Explicaremos el tipo de dato en que se obtiene cada atributo y qué procesado le hemos dado a cada uno de ellos.

Como ya se ha comentado en la subsección anterior, los datos que se han obtenido son de 2 fuentes de datos diferentes. Por un lado, datos relativos a la película y obtenidos de IMDb [7], y por otro, datos relativos a la estructura de grafo de la película, obtenidos de *Moviegalaxies* [25]**.**

Este primer subconjunto de datos, se ha obtenido a través de *IMDbPY* [24], una librería desarrollada en Python que permite obtener todos los datos de una película de IMDb [7] en un mismo objeto (*movie object*). Este objeto contiene una serie de

atributos, cada uno de ellos en un formato concreto, y sobre los que se ha tenido que implementar un pre-procesado para poder incluirlos en nuestro *dataframe*:

- **Rating:** Se obtiene en formato "*string*" y se transforma en formato "*float*". Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).
- **Presupuesto (en dólares):** Este atributo se ha obtenido realizando *scrapping* sobre la página de la película en IMDb. Este dato está disponible en este formato "$15,000,000" y se realiza un pre-procesado para transformarlo en un *float*, del tipo "15.000.000". Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).
- **Año de producción:** Se obtiene en formato "*string*". Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).
- **Director:** Este atributo se obtiene como una "lista de personas", que son todas las personas que participan en la dirección de la película, ordenados por orden de aparición en los títulos.

  Nosotros nos quedamos únicamente con el primer elemento de la lista, y además con el atributo "*name*" del elemento. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).
- **Personajes:** Este atributo se obtiene como una "lista de personas", que son todas las personas que forman parte del *casting* de la película, ordenados por orden de aparición en los títulos.

  Nosotros nos quedamos únicamente con los primeros 3 elementos de la lista, y además con el atributo "*name*" del elemento. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).
- **Guionista:** Este atributo se obtiene como una "lista de personas", que son todas las personas que forman parte en la escritura del guión de la película, ordenados por orden de aparición en los títulos.

  Nosotros nos quedamos únicamente con el primer elemento de la lista, y además con el atributo "*name*" del elemento. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).

- **Productor:** Este atributo se obtiene como una "lista de personas", que son todas las personas que forman parte en la producción de la película, ordenados por orden de aparición en los títulos.

  Nosotros nos quedamos únicamente con el primer elemento de la lista, y además con el atributo "*name*" del elemento. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).
- **País de producción:** Este atributo se obtiene como una lista de *strings*, que contiene los países que han participado en la producción de la película, ordenados por orden de importancia.

  Nosotros nos quedamos únicamente con el primer elemento de la lista. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).
- **Géneros:** Este atributo se obtiene como una lista de *strings*, que contiene los géneros dentro de los que se engloba la película, ordenados por orden de importancia.

  Nosotros nos quedamos únicamente con los 3 primeros elementos de la lista. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).
- **Director de vestuario:** Este atributo se obtiene como una "lista de personas", que son todas las personas que han participado en la dirección del vestuario de la película, ordenados por orden de aparición en los títulos.

  Nosotros nos quedamos únicamente con el primer elemento de la lista, y además con el atributo "*name*" del elemento. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).
- **Director de Sonido:** Este atributo se obtiene como una "lista de personas", que son todas las personas que han participado en la dirección de sonido de la película, ordenados por orden de aparición en los títulos.

  Nosotros nos quedamos únicamente con el primer elemento de la lista, y además con el atributo "*name*" del elemento. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).

- **Editor:** Este atributo se obtiene como una "lista de personas", que son todas las personas que han participado en la edición de la película, ordenados por orden de aparición en los títulos.

  Nosotros nos quedamos únicamente con el primer elemento de la lista, y además con el atributo "*name*" del elemento. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).
- **Manager de Producción:** Este atributo se obtiene como una "lista de personas", que son todas las personas que han participado en la gestión de producción de la película, ordenados por orden de aparición en los títulos.

  Nosotros nos quedamos únicamente con el primer elemento de la lista, y además con el atributo "*name*" del elemento. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).
- **Director de efectos especiales:** Este atributo se obtiene como una "lista de personas", que son todas las personas que han participado en la dirección de efectos especiales de la película, ordenados por orden de aparición en los títulos.

  Nosotros nos quedamos únicamente con el primer elemento de la lista, y además con el atributo "*name*" del elemento. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).
- **Director de fotografía:** Este atributo se obtiene como una "lista de personas", que son todas las personas que han participado en la dirección de fotografía de la película, ordenados por orden de aparición en los títulos.

  Nosotros nos quedamos únicamente con el primer elemento de la lista, y además con el atributo "*name*" del elemento. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).
- **Director de maquillaje:** Este atributo se obtiene como una "lista de personas", que son todas las personas que han participado en la dirección de maquillaje de la película, ordenados por orden de aparición en los títulos.

  Nosotros nos quedamos únicamente con el primer elemento de la lista, y además con el atributo "*name*" del elemento. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).

- **Compañía de Producción:** Este atributo se obtiene como una "lista de compañías", que son todas las compañías que han participado en la producción de la película, ordenadas por orden de aparición en los títulos. Nosotros nos quedamos únicamente con el primer elemento de la lista, y además con el atributo "*name*" del elemento. Si alguna película no tiene este atributo, le asignamos "?" (valor desconocido).

Datos relativos a la estructura del grafo y obtenidos de *Moviegalaxies* [25]**:**

Los atributos que se detallan a continuación son métricas de análisis de redes sociales. A continuación, se detalla qué tipo de pre-procesado se ha realizado a cada uno de ellos.

Dentro de estos datos, existe un subconjunto que son métricas a nivel de red, por lo que apenas ha sido necesario ningún pre-procesado, solamente aquellos datos de tipo *float*, se han formateado para que tengan 3 dígitos decimales. Estos atributos son los siguientes: *Densidad*, *Número de nodos*, *Tamaño*, *Diámetro*, *Radio*, *Grado de Asortatividad* (*degree assortativity*), *Número de Cliques* y *Transitividad*.

Todas las demás métricas que se han obtenido de la estructura de red, son métricas que la librería utilizada proporcionaba a nivel de nodo. Lo verdaderamente interesante y útil (para conseguir mayor generalización en los datos) es extraer información a nivel de película, para lo que se ha realizado un pre-procesado y tratamiento de estos datos. Lo que se ha decidido es calcular los valores medios, máximos y mínimos de cada una de estas métricas. De esta manera, lo que se consigue es generalizar los datos obtenidos de los nodos y extrapolarlos a nivel de película.

Para explicarlo con mayor claridad, en lugar de interpretar los valores de estas métricas de todos los nodos de cada una de las redes (una red por cada película), lo que haremos será calcular los valores medios, máximos y mínimos de estas métricas, e incorporar esta información como atributos dentro de nuestro *dataframe* (se incluye un ejemplo de una de las métricas, como es la intermediación):

- Intermediación Media de una Película: lo calcularemos como el valor medio de los valores de intermediación de todos los nodos de la red que representa a esta película.

- Intermediación Máxima de una Película: lo calcularemos como el valor máximo de los valores de intermediación de todos los nodos de la red que representa a esta película.

- Intermediación Mínima de una Película: lo calcularemos como el valor mínimo de los valores de intermediación de todos los nodos de la red que representa a esta película.

Los datos pre-procesados bajo este planteamiento, han sido: Grado, Excentricidad, Centralidad de Grado (*degree centrality*), Cercanía (*closeness centrality*), Intermediación (*betweenness centrality*), Intermediación de Enlace (*edge betweenness centrality*), Centralidad de Vector Propio (*eigenvector centrality*), Centralidad de Comunicabilidad (*communicability centrality*), Intermediación de Comunicabilidad (*communicability betweenness centrality*), Centralidad de Carga (*load centrality*), Conectividad de un Nodo (*node connectivity*), Triángulos, Clustering, Longitud del Camino más corto.

2.2.4. Integración y Transformación de Datos

Se ha decidido crear un conjunto de datos en un formato tal que nos permitiese su fácil exportación al formato *arff* de WEKA, herramienta con la que se procederá al proceso de minería de datos. El formato elegido fue el *csv*, creando así una tabla de datos que esté conformada por varias columnas, una por cada atributo obtenido en fases anteriores, y en el que cada fila contendrá los datos de cada una de la instancias utilizada en la fase de entrenamiento y test, esto es, de cada una de las películas que conforman nuestro dataframe, con la intención de tener una estructura de datos tal que pueda ser explotada con herramientas de minería de datos, como WEKA.

Además, se va a añadir una columna más, que será la CLASE (o tipo), a la que pertenecerá cada una de las instancias. A continuación, se detalla cómo se ha definido esta clase.

Definición de Clases de Predicción

Tras haber diseñado y conformado el *dataframe* de los datos que intentaremos explotar, el siguiente paso será hacer una definición de clase, en la que nos basaremos para realizar una primera clasificación de las películas.

Para ello nos hemos basado en el *rating* de cada película en IMDb. Se ha querido realizar una definición de clase lo más balanceada posible, teniendo en cuenta la distribución de instancias de los datos.

Así, se decide plantear dos tipos de definición de clase, Binaria (Tabla 10) y Multi-clase (Ver Tabla 11):

- **Definición Binaria**:
  - **C-** → para instancias con ***rating*** **<=7**

    Dentro de esta categoría tenemos un total de 369 instancias
  - **C+** → para instancias con ***rating*** **> 7**

    Dentro de esta categoría tenemos un total de 377 instancias

Clase	Definición
C-	rating <= 7
C+	rating > 7

Tabla 10. Definición de Clase Binaria.

- **Definición Multi-clase:**
  - **C1** → para instancias con valores de *rating*, tal que ***rating*** < **6.5**

    Dentro de esta categoría tenemos un total de 198 instancias.
  - **C2** → para instancias con valores de *rating*, tal que **6.5** <= ***rating*** < **7**

    Dentro de esta categoría tenemos un total de 138 instancias.
  - **C3** → para instancias con valores de *rating*, tal que **7** <= ***rating*** < **7.5**

    Dentro de esta categoría tenemos un total de 157 instancias.
  - **C4** → para instancias con valores de *rating*, tal que **7.5** <= ***rating*** < **8.0**

    Dentro de esta categoría tenemos un total de 129 instancias.
  - **C5** → para instancias con valores de *rating*, tal que ***rating*** >= **8.0**

    Dentro de esta categoría tenemos un total de 124 instancias.

Clase	Definición
C1	rating < 6,5
C2	6,5 <= rating < 7
C3	7 <= rating < 7,5
C4	7,5 <= rating < 8,0
C5	rating >= 8

Tabla 11. Definición Multi-clase.

Se ha querido hacer una primera definición binaria, definiendo como **clase negativa** aquellas instancias con un *rating* menor o igual a 7, y como **clase positiva** aquellas instancias con un *rating* mayor a 7. De esta manera, se maximiza el número de casos por clase para mejorar la generalización de cara a los primeros experimentos, la validación puede ser más rigurosa (ya que se podrían reservar más casos para testado), y se simplifica el problema a clasificación binaria.

Se realizará una primera fase de experimentos, aportando al conjunto de datos la definición de clase binaria, y si éstos presentan buenos resultados, pasaríamos a realizar otra fase de experimentos con el conjunto de datos clasificados en base a la definición Multi-Clase definida anteriormente.

2.2.5. Selección de Atributos

Antes de ejecutar la fase de minería de datos y con la intención de quedarnos con los atributos más relevantes del dataframe, se decide aplicar métodos de selección de atributos.

Se han utilizado dos métodos de evaluación de atributos diferentes, aplicando a cada uno de ellos distingos métodos de búsqueda. Los métodos de evaluación aplicados han sido "*CfsSubsetEval*" y "*ClassifierSubsetEval*" de WEKA combinándonos con los siguientes métodos de búsqueda: "*BestFirst*", "*GreedyStepwise*", "*LinearForwardSelection*", y "*GeneticSearch*".

Los evaluadores de subconjuntos toman un subconjunto de atributos y devuelven una medida numérica que guía la búsqueda [30, pp. 420-425].

CfsSubsetEval evalúa la capacidad predictiva de cada subconjunto de atributos por separado y el grado de redundancia entre ellos, eligiendo conjuntos de atributos que están altamente correlacionados con la clase, pero que tienen una baja intercorrelación entre ellos [30, pp. 420-425].

ClassifierSubsetEval utiliza un clasificador, especificado en el editor de objetos como parámetro, para evaluar conjuntos de atributos en los datos de entrenamiento o en un conjunto separado de retención [30, pp. 420-425]. En nuestro caso se ha utilizado *J48* como clasificador. Cuando se utiliza este clasificador en selección de atributos, se construye un árbol a partir de los datos dados. Se supone que todos los atributos que no aparecen en el árbol, son irrelevantes. Por consiguiente, el conjunto

de atributos que aparecen en el árbol, forma el subconjunto reducido de atributos [12].

Los métodos de búsqueda recorren el espacio de atributos para encontrar un buen subconjunto. La calidad es medida por el evaluador de subconjuntos de atributos elegidos [30, p. 423]. Cada método de búsqueda se ha aplicado con la configuración por defecto de WEKA.

BestFirst realiza lo que se conoce como "*hill climbing*" con backtracking. Se puede especificar cuántos nodos consecutivos de no mejora deben encontrarse antes de que el sistema retroceda. Puede buscar hacia adelante desde el conjunto vacío de atributos, hacia atrás desde el conjunto completo, o comenzar en un punto intermedio (especificado por una lista de índices de atributos) y buscar en ambas direcciones al considerar todas las posibles adiciones y supresiones de un solo atributo. Los subconjuntos que se han evaluado se almacenan en caché para obtener mayor eficiencia. El tamaño de caché también es parametrizable [30, p. 423].

GreedyStepwise busca exhaustivamente en el espacio de subconjuntos de atributos. Al igual que *BestFirst*, puede avanzar desde el conjunto vacío o hacia atrás desde el conjunto. A diferencia de *BestFirst*, no retrocede, pero termina tan pronto como disminuya la métrica de evaluación, al agregar o eliminar el mejor atributo. De modo alternativo, clasifica los atributos recorriendo el espacio de vacío a completo (o viceversa) y registrando el orden en el que se seleccionan los atributos. Se puede especificar el número de atributos a retener o establecer un umbral por debajo del cual se descarten [30, pp. 423,424].

Con *LinearForwardSelection,* también es posible generar una lista ordenada de atributos, continuando la competición has que se hayan seleccionado todos los atributos: la clasificación se establece en el orden en el que se agregan estos atributos [30, p. 424.].

GeneticSearch utiliza un algoritmo genético simple [31]. Los parámetros incluyen el tamaño de la población, el número de generaciones y las probabilidades de cruce

y mutación. Se puede especificar una lista de índices de atributos como punto de partida, que se convierte en un miembro de la población inicial [30, p. 424].

2.2.6. Minería de Datos

En esta fase, y de cara a llevar a cabo la clasificación de atributos, se investigará con varios métodos de clasificación, experimentado con los siguientes clasificadores implementados en WEKA:

- *J48.*
- *Random Forests.*
- *Adaboost.*
- *NaiveBayes.*
- *OneR.*

J48 es la implementación en Java del algoritmo C4.5 [32]. Este algoritmo se engloba dentro de los algoritmos de inducción recursiva (*top-down*) de árboles de decisión. Construye una estructura de diagrama de flujo donde cada nodo interno representa un test de un atributo, cada rama se corresponde con el resultado de cada test y cada nodo externo (hoja) representa una predicción de clase. En cada nodo, el algoritmo elige el mejor atributo por el que particionar los datos en clases individuales [12].

Random Forests son una combinación de predictores de árboles, de manera que cada árbol depende de los valores de un vector aleatorio muestreado independientemente y con la misma distribución para todos los árboles en el bosque. El error de generalización para los bosques converge a un límite, a medida que el número de árboles en el bosque va aumentando. El error de generalización de un bosque de clasificadores de árboles depende de la fuerza de cada uno de los árboles en el bosque y la correlación entre ellos. *Random Forests* utiliza una selección aleatoria de características para dividir cada nodo, dando lugar a tasas de error que se consideran favorables a las obtenidas mediante *Adaboost*, pero son más robustas con respecto al ruido.

Estimaciones internas monitorizan el error, la fuerza y la correlación y éstas se utilizan para mostrar la respuesta al aumento del número de características utilizadas en la división. Las estimaciones internas también se utilizan para medir la importancia de la variable. [33, pp. 5-32].

Adaboost [34] es un meta-algoritmo que, teóricamente, puede utilizarse para reducir significativamente el error de cualquier algoritmo de aprendizaje que genera clasificadores cuyo rendimiento es un poco mejor que la predicción aleatoria, pero que son denotados como algoritmos de aprendizaje débiles.

NaiveBayes [35] implementa el clasificador probabilístico "*Naïve Bayes Classifier*". El método está diseñado para su uso en tareas de inducción supervisadas, en las que el objetivo es predecir con precisión la clase a la que pertenecen una serie de instancias de prueba y en la que las instancias de entrenamiento incluyen información de la clase.

Los clasificadores bayesianos son clasificadores estadísticos. Pueden predecir las probabilidades de pertenencia a una clase, como la probabilidad de que una *tupla* (lista ordenada de n elementos) determinada pertenezca a una clase particular.

La clasificación bayesiana se basa en el teorema de Bayes, que para problemas de clasificación se puede ver como:

Sea X una tupla de n atributos y sea H una hipótesis tal que la tupla X pertenezca a la clase C, se busca la probabilidad de que la tupla X pertenezca a la clase C, dado que conocemos la descripción de X. [12, pp. 350-355]

$$P(H|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|H)P(H)}{P(\boldsymbol{X})}$$

Estudios que comparan algoritmos de clasificación han encontrado un clasificador bayesiano simple conocido como el clasificador bayesiano "*naïve*", que es comparable en rendimiento a los árboles de decisión y a determinados clasificadores basados en redes neuronales.

Los clasificadores bayesianos "*naïve*" asumen que el efecto de un valor de atributo en una clase dada es independiente de los valores de los otros atributos. Esta suposición se llama independencia condicional. Se hace para simplificar los cálculos implicados y, en este sentido, se considera "naïve" [12].

Este clasificador se puede ver como una forma determinada de red bayesiana, llamada "*naive*" porque se basa en dos importantes suposiciones simplificadoras. En particular, asume que los atributos predictivos son condicionalmente independientes dada la clase, y postula que ningún atributo oculto o latente influye en el proceso de predicción. Así, cuando se representa gráficamente, un clasificador bayesiano "*naive*" tiene la forma mostrada en la Figura 5, en la cual todos los arcos son dirigidos desde el atributo de clase a los atributos predictivos [35].

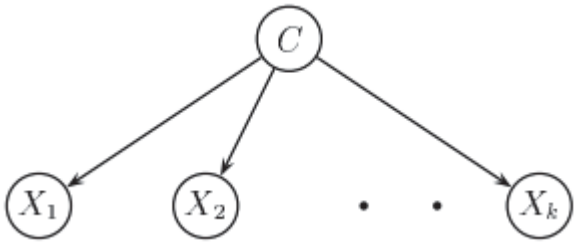

Figura 5. Clasificador NaiveBayesian representado como una Red Bayesiana en la cual los atributos representados ($X_1, X_2, \ldots X_K$) son condicionalmente independientes dado el atributo clase (C).

Estas suposiciones soportan algoritmos muy eficientes, tanto para la clasificación como para el aprendizaje.

OneR es una implementación para construir y usar clasificadores conocidos como 1R (1 regla). Estos clasificadores utilizan el atributo "error-mínimo" para predicción, discretizando atributos numéricos [36]. De [30, pp. 84-88] se recoge una descripción más detallada del método. Este método genera un árbol de decisión de un nivel, expresado como un conjunto de reglas que testean un atributo en particular. Resulta que las reglas simples con frecuencia alcanzan una precisión sorprendentemente alta. Tal vez esto se debe a que la estructura que subyace en muchos conjuntos de datos del mundo real es

bastante rudimentaria, y sólo un atributo es suficiente para determinar la clase de una instancia con bastante precisión. En cualquier caso, siempre es una buena opción para llevar a cabo unos primeros experimentos simples.

La idea es que se generan reglas que testean un solo atributo y, en consecuencia, una sola rama. Cada rama corresponde a un valor diferente del atributo. Para dilucidar cuál es la mejor clasificación que podemos dar a cada rama, se usa la clase que se da con más frecuencia en los datos de entrenamiento. El cálculo de la tasa de error es simple, pues basta con contar los errores que se producen con los datos de entrenamiento, es decir, el número de instancias que no pertenecen a la clase mayoritaria.

En conclusión, cada atributo genera un conjunto distinto de reglas, una por cada valor del atributo. A continuación, se evalúa la tasa de error para el conjunto de reglas de cada atributo y se elige el mejor. En la Figura 6 se muestra un ejemplo del pseudocódigo de este algoritmo.

```
For each attribute,
  For each value of that attribute, make a rule as follows:
    count how often each class appears
    find the most frequent class
    make the rule assign that class to this attribute-value.
  Calculate the error rate of the rules.
Choose the rules with the smallest error rate.
```

Figura 6. Pseudocódigo de OneR.

Para evaluar el rendimiento cada uno de los algoritmos detallados con anterioridad, se ha utilizado "*10 fold – cross validation*". Según [9, pp. 149-151], *10 fold cross-validation* es la manera estándar de predecir la tasa de error de un algoritmo de aprendizaje. Los datos se dividen aleatoriamente en 10 partes, en las que la clase a queda representada aproximadamente en las mismas proporciones que en el conjunto completo de datos. Se toma cada una de las partes y se entrena al esquema de aprendizaje con las 9 partes restantes. Después se calcula la tasa de error con la parte tomada inicialmente. Este

proceso se ejecuta un total de 10 veces (una por cada división de datos), con diferentes conjuntos de entrenamiento. Finalmente, se promedian las 10 estimaciones de error para obtener una estimación global del error.

Además, WEKA por defecto implementa lo que se conoce como "*stratified*" *cross-validation*. Esto quiere decir que, durante el proceso de división de datos, se asegura que cada división tenga la proporción adecuada de cada valor de la clase, tanto en el conjunto de datos de entrenamiento, como en el conjunto de datos de test.

En la Figura 7, se adjunta de manera resumida la descripción de *10 fold – cross validation*.

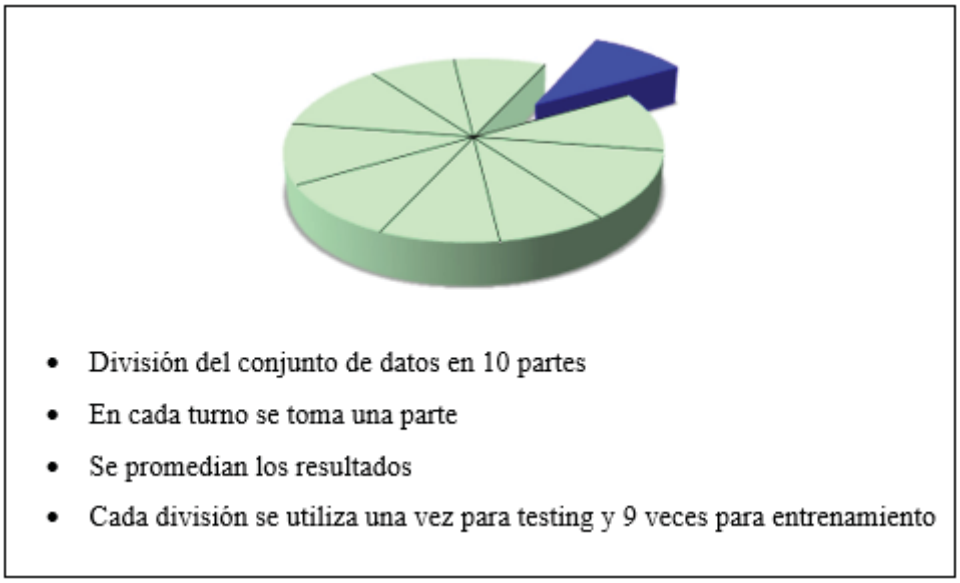

Figura 7. Stratified 10 fold cross-validation.

2.2.7. Evaluación e Interpretación de los Resultados

Para cuantificar la calidad del clasificador utilizado en cada experimento, se analizarán medidas de evaluación del rendimiento, tales como "*Accuracy*" (Porcentaje de Predicciones Correctas), "*Precision*" (Valor Predictivo Positivo), "*Recall*" (Sensibilidad) y "*F-Measure*" (Medida F). Todas ellas están basadas en los cuatro posibles resultados: *TP*, *TN*, *FP* y *FN*.

- ***TP*** (*True Positive*): Clasificaciones correctas.
- ***TN*** (*True Negative*): Clasificaciones correctas.
- ***FP*** (*False Positive*): Se da cuando el resultado es predicho como positivo (sí) incorrectamente, cuando realmente sería un resultado negativo (no).
- ***FN*** (*False Negative*): Se da cuando el resultado es predicho como negativo (no) cuando realmente sería un resultado positivo (sí).

Podemos expresar estos conceptos en términos de la matriz de confusión (Figura 8). La matriz de confusión es una herramienta muy útil que se utiliza para analizar cuán bien puede un clasificador reconocer instancias de diferentes clases. *TP* y *TN* nos dicen cuándo el clasificador está haciendo las cosas bien, mientras que *FP* y *FN* nos dice cuándo el clasificador se está comportando de manera errónea [12, pp. 364-376].

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	*P*
	no	*FP*	*TN*	*N*
	Total	P'	N'	$P + N$

Figura 8. Matriz de Confusión, con totales para tuplas positivas y negativas.

Para entender mejor estos conceptos, los explicaremos sobre un ejemplo [37]. Suponiendo un sistema de clasificación que ha sido entrenado para distinguir entre gatos, perros y conejos, se presenta la matriz de confusión de la Figura 9, que resume los resultados de la prueba. Hay que tener en cuenta que, de los 27 animales, 8 son gatos, 6 perros y 13 son conejos.

		Valor Predicho		
		Gato	Perro	Conejo
Valor Real	Gato	5	3	0
	Perro	2	3	1
	Conejo	0	2	11

Figura 9. Ejemplo de Matriz de Confusión.

Asumiendo esta matriz de confusión, la tabla de confusión para la clase "gato", sería la mostrada en la Tabla 12.

5 *TP* (gatos reales que fueron correctamente clasificados como gatos)	**3 *FN*** (gatos que fueron incorrectamente clasificados como perros)
2 *FP* (animales no-gatos que fueron incorrectamente clasificados como gatos)	**17 *TN*** (resto de animales, correctamente clasificados como no-gatos)

Tabla 12. Tabla de confusión para la clase "gato".

A continuación, se definen las medidas de evaluación que se utilizarán:

- *Accuracy* (Porcentaje de Predicciones Correctas): Se define como el porcentaje de instancias clasificadas correctamente.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

- *Precision* (Valor Predictivo Positivo): Se puede tomar como una medida de exactitud (qué porcentaje de instancias clasificadas como positivas son realmente positivas)

$$Precision = \frac{TP}{(TP + FP)}$$

- *Recall* (Sensibilidad): Es una medida de sensibilidad (qué porcentaje de instancias positivas son clasificadas como positivas)

$$recall = \frac{TP}{(TP + FN)}$$

- *F-measure* (Medida F): Es la media armónica del Valor Predictivo Positivo (*Precision*) y la Sensibilidad (*recall)*.

$$F = \frac{2 \times precision \times recall}{precision + recall} = \frac{2{\times}TP}{(2{\times}TP + FN + FP)}$$

Donde:

- **TP** (*True Positive*)
- **TN** (*True Negative*)
- **FP** (*False Positive*)
- **FN** (*False Negative*)

3. RESULTADOS

En esta sección se recogerán los resultados de los experimentos realizados. Se llevarán a cabo varias iteraciones/fases, cada una de ellas con alguna modificación en los datos explotados.

3.1. EXPERIMENTOS FASE 1

En esta fase, los experimentos han sido realizados con el *dataframe* de datos que se adjunta el ANEXO I. DataFrame de Datos (fase 1 y 2 de experimentos) más el atributo "*rating*", erróneamente incluido, es decir, un dataframe con 64 atributos. Además, se ha hecho distinción en los experimentos entre clasificación binaria (C+, C-) y clasificación Multi-Clase (C1, C2, C3, C4 y C5).

Los resultados que se han obtenido en esta fase han sido inesperadamente buenos, obteniendo una precisión del 100% con métodos de clasificación simples como son *J48* y *OneR*. Al intentar dilucidar la razón de estos perfectos resultados, nos dimos cuenta de que tanto *J48* como *OneR* estaban basando su clasificación en un único atributo, "*rating*". Por consiguiente, detectamos que habíamos cometido un error, y es que habíamos incluido en la colección de datos, el atributo en el que precisamente nos habíamos basado para definir nuestra CLASE. Los siguientes experimentos los realizaremos eliminando este atributo del dataframe.

3.2. EXPERIMENTOS FASE 2

En esta fase, los experimentos han sido realizados con el *dataframe* de datos que se adjunta en ANEXO I. DataFrame de Datos (fase 1 y 2 de experimentos), es decir, un dataframe con 63 atributos. Además, se hará distinción en los experimentos entre clasificación binaria (C+, C-) y clasificación Multi-Clase (C1, C2, C3, C4 y C5). Se comentan los resultados obtenidos en las siguientes sub-secciones.

3.2.1. Resultados de Experimentos con Clasificación Binaria

En la Tabla 13, se recogen los resultados obtenidos en los experimentos realizados. En la primera columna se detalla el tipo de Algoritmo aplicado, en la segunda y tercera el porcentaje de clasificaciones correctas e incorrectas, respectivamente. En las últimas cinco columnas se resumen los valores de algunas métricas de calidad, para cada una de las clasificaciones.

ALGORITMO	% Clas. OK	% Clas. KO	CLASE	Precision	Recall	F-Measure	ROC Area
OneR	50,6	49,4	C+	0,506	0,891	0,645	0,502
			C-	0,506	0,114	0,186	0,503
			Media	0,506	0,506	0,418	0,502
J48	51,81	48,18	C+	0,513	0,878	0,648	0,52
			C-	0,549	0,152	0,238	0,524
			Media	0,531	0,518	0,445	0,522
RandomForest	61,75	38,25	C+	0,612	0,662	0,636	0,672
			C-	0,624	0,572	0,597	0,676
			Media	0,618	0,617	0,617	0,674
NaiveBayes	57,32	42,68	C+	0,723	0,25	0,372	0,664
			C-	0,541	0,902	0,677	0,664
			Media	0,633	0,573	0,523	0,664
AdaBoost	64,16	35,84	C+	0,663	0,59	0,624	0,699
			C-	0,624	0,694	0,657	0,702
			Media	0,644	0,642	0,641	0,701

Tabla 13. Resultados Experimentos FASE 2 Clasificación Binaria.

En general, podemos concluir que los resultados obtenidos son bastante malos. Los algoritmos que mejor resultado presentan son "*AdaBoost*" con un 64.16% de clasificaciones correctas, seguido de "*RandomForest*" con un porcentaje de acierto del 61.75%.

3.2.2. Resultados de Experimentos con Clasificación Multi-Clase

En la Tabla 14, se recogen los resultados obtenidos en los experimentos realizados. En la primera columna se detalla el tipo de Algoritmo aplicado, en la segunda y tercera el porcentaje de clasificaciones correctas e incorrectas, respectivamente. En las últimas cinco columnas se resumen los valores de algunas métricas de calidad, para cada una de las clasificaciones.

ALGORITMO	% Clas. OK	% Clas. KO	CLASE	Precision	Recall	F-Measure	ROC Area
OneR	22,04	77,96	C1	0,278	0,051	0,086	0,5
			C2	0,25	0,079	0,12	0,512
			C3	0,22	0,877	0,351	0,531
			C4	0	0	0	0,478
			C5	0,364	0,065	0,11	0,521
			Media	0,226	0,22	0,136	0,509
J48	27,01	72,99	C1	0,272	0,959	0,424	0,525
			C2	0	0	0	0,501
			C3	0	0	0	0,481
			C4	0	0	0	0,526
			C5	0,24	0,098	0,139	0,524
			Media	0,112	0,27	0,135	0,511
RandomForest	34,68	65,32	C1	0,363	0,772	0,494	0,691
			C2	0,313	0,179	0,227	0,563
			C3	0,336	0,26	0,293	0,561
			C4	0,218	0,092	0,13	0,519
			C5	0,408	0,236	0,299	0,699
			Media	0,33	0,347	0,306	0,611
NaiveBayes	27,42	72,58	C1	0,36	0,619	0,455	0,654
			C2	0,197	0,329	0,246	0,537
			C3	0,333	0,006	0,013	0,542
			C4	0,189	0,215	0,201	0,536
			C5	0,35	0,057	0,098	0,711
			Media	0,292	0,274	0,221	0,598
AdaBoost	31,85	68,15	C1	0,292	0,954	0,447	0,546
			C2	0	0	0	0,535
			C3	0	0	0	0,498
			C4	0	0	0	0,495
			C5	0,49	0,398	0,439	0,611
			Media	0,158	0,319	0,191	0,534

Tabla 14. Resultados Experimentos FASE 2 Clasificación Multi-Clase.

En esta ocasión, intentando clasificar los datos del *dataframe* en una definición Multi-clase, observamos que los resultados obtenidos son bastante peores que los obtenidos con una definición de clase binaria. Los algoritmos que mejores resultados presentan siguen siendo "*AdaBoost*" y "*RandomForest*", aunque no se pueden considerar como resultados ni mucho menos concluyentes.

3.2.3. Resultados tras Selección de Atributos

Al detectar que los resultados obtenidos en la FASE 2 de experimentos no era muy satisfactorios, se decidió repetir estos experimentos tras aplicar métodos de selección de atributos, para intentar quedarnos con los atributos más relevantes.

Se han utilizado dos métodos de evaluación de atributos diferentes, aplicando a cada uno de ellos distingos métodos de búsqueda. Los métodos de evaluación aplicados han sido "*CfsSubsetEval*" y "ClassifierSubsetEval" de WEKA combinándonos con los siguientes métodos de búsqueda: "*BestFirst*", "*GreedyStepwise*", "*LinearForwardSelection*", y "*GeneticSearch*". Estos métodos han sido detallados en la sección 2.2.5 Selección de Atributos.

Los resultados obtenidos al ejecutar los experimentos con los atributos seleccionados, no mejoraron los obtenidos en las fases anteriores. Se resume en la Tabla 15 cuáles fueron los atributos seleccionados por cada método:

Evaluador	Método de Búsqueda	Clasificación	# Atributos Selec.	Atributos Seleccionados
CfsSubsetEval	BestFirst, GreedyStepwise, LinearForwardSelection	Binaria	4	2,5,6,7
		Multi-Clase	1	6
CfsSubsetEval	GeneticSearch	Binaria	13	2,3,4,5,7,8,9,14,16,17,20,56,57
		Multi-Clase	10	1,4,5,6,7,18,19,20,56,57
ClassifierSubsetEval (con J48)	BestFirst	Binaria	20	2,10,11,22,24,26,27,29,33,34,38, 46,47,49,50,51,52,58,61,62
		Multi-Clase	6	3,7,22,25,44,45
ClassifierSubsetEval (con J48)	GeneticSearch	Binaria	21	1,10,22,28,29,30,32,34,35,37,40, 42,43,44,46,50,51,53,54,60,61
		Multi-Clase	16	7,8,17,18,22,25,26,27,28,30,36, 40,45,47,60,62
ClassifierSubsetEval (con J48)	GreedyStepwise	Binaria	8	2,10,11,22,26,38,50,52
		Multi-Clase	6	3,7,22,25,44,45
ClassifierSubsetEval (con J48)	LinearForwardSelection	Binaria	13	2,10,11,32,34,38,41,44,49,50,51, 52,61
		Multi-Clase	6	3,7,22,25,44,45

Tabla 15. Resumen de Atributos Seleccionados.

Adicionalmente, se detallan más estos datos en los Anexos ANEXO II. Selección de Atributos en Clasificación Binaria y ANEXO III. Selección de Atributos en Clasificación Multiclase.

3.3. EXPERIMENTOS FASE 3

En esta fase se incluirán los resultados de los experimentos realizados en una 2ª iteración de minería de datos, en la que se ha decidido modificar el dataframe inicial con el que se realizaron la fase 1 y 2 de experimentos. Los malos resultados obtenidos en la fase de experimentos anterior, nos hace pensar que tenemos muchos atributos de tipo nominal con un número de instancias tal, que provocan que esos datos sean poco generalizables. Por este motivo y con la intención de obtener mayor generalización en nuestros datos, se decide eliminar del dataframe aquellos atributos nominales cuyo número de instancias se acerque a la mitad del número de instancias totales.

Adicionalmente, se decide también no incluir los atributos "*secondary_genre*" y "*third_genre*" para ganar en generalidad en el género de la película. Se eliminan del dataframe, por tanto, los atributos detallados en la Tabla 16:

ATRIBUTO	# INSTANCIAS DIFERENTES
secondary genre	21
third genre	19
main director	470
main character	464
secondary character	620
third character	658
main writer	597
main producer	551
main costume designer	332
main sound	409
main editor	431
main production manager	568
main visual effects	436
main cinematographer	377
make up	483
main production company	316

Tabla 16. Listado de atributos eliminados del dataframe.

Por otro lado, y buscando mejorar la calidad de nuestro dataframe, se decide definir nuevos atributos que serán añadidos al dataframe. A continuación, se explica cómo se han obtenido estos nuevos atributos.

Se ha intentado aprovechar al máximo la información que podríamos obtener de la estructura de red que posee cada película. Es por esto que decidimos crear una definición de protagonista de una película, en función de las métricas propias del SNA, con la intención extraer más atributos, pero solo de estos protagonistas, pensando en que éstos serán los más relevantes.

En la siguiente subsección, se hace una descripción detallada del proceso de definición de protagonista, así como de nuevos atributos que se han definido.

3.3.1. Definición de Protagonista y nuevos Atributos

Para proponer una definición de protagonista de una película, tras analizar varias métricas propias del SNA, se ha estimado que los atributos más relevantes para este fin debían ser:

- **Grado de un Personaje**: Número de enlaces que tiene un personaje con otros personajes de la película.
- **Intermediación** (*betweenness centrality*): medida que cuantifica la frecuencia o el número de veces que un personaje (nodo) actúa como un puente a lo largo del camino más corto entre otros dos personajes (nodos).
- **Centralidad de Vector Propio** (*Eigenvector Centrality*): Mide la influencia de un personaje (nodo) en una película (red).
  Intuitivamente, los personajes (nodos) que poseen un valor alto de esta medida de centralidad están conectados a muchos personajes (nodos) que a su vez están bien conectados. Esta medida asigna puntuación a todos los personajes (nodos) de la película (red), basado en el concepto de que conexiones a personajes (nodos) con alta puntuación, contribuyen más a la puntuación del personaje (nodo) en cuestión.

Para realizar el análisis de estos atributos desde el punto de vista de una red, se han tomado 5 películas como referencia:

- El Padrino.
- El Señor de los Anillos: La Comunidad del Anillo.
- Forrest Gump.
- Gladiador.
- Cadena Perpetua.

A continuación, se ha hecho uso de "*Gephi*" [38] para realizar un primer estudio gráfico de la estructura de la red, y "NetworkX" [23] para exportar los atributos necesarios de cada una de estas películas.

En función de estos atributos, se proponen las siguientes definiciones de protagonista de una película:

Definición 1. Un personaje es protagonista si:

$$\frac{Grado\ del\ Personaje}{Max.\ Grado\ en\ la\ Película} > 0.5$$

Definición 2. Un personaje es protagonista si:

$$Intermediación\ del\ Personaje > Intermediación\ Media\ de\ la\ Película$$

Definición 3. Un personaje es protagonista si:

$$\frac{Grado\ del\ Personaje}{\#\ Personajes\ en\ la\ Película} > 2 * Centr.Media\ de\ Vector\ Propio\ Norm.de\ la\ Película$$

Donde "*Centr. Media de Vector Propio Norm.*" Es la media de todos los valores de Centralidad de Vector Propio de la red, normalizados por el valor máximo en todos ellos.

Además, se define este cociente como un nuevo atributo (que incluiremos en el dataframe), al que definiría como: Popularidad del Personaje.

$$Popularidad = \frac{Grado\ del\ Personaje}{\#\ Personajes\ en\ la\ Película}$$

Definición 4. Un personaje es protagonista si:

$$\frac{Grado\ del\ Personaje}{Max\ Grado\ en\ la\ Película} > 2 * Centr.Media\ de\ Vector\ Propio\ Norm.de\ la\ Película$$

Donde "*Centr. Media de Vector Propio Norm*." Es la media de todos los valores de Centralidad de Vector Propio de la red, normalizados por el valor máximo en todos ellos.

Además, se definirá este cociente como un nuevo atributo (que incluiremos en el dataframe), al que nombraremos como: Relevancia del Personaje.

$$Relevancia = \frac{Grado\ del\ Personaje}{Max\ Grado\ en\ la\ Película}$$

En la siguiente tabla (Tabla 17) se resumen los resultados obtenidos con cada una de las definiciones propuestas, detallando el número de protagonistas obtenidos y quiénes son estos protagonistas.

PELÍCULA	# PROTAGONISTAS				PROTAGONISTAS			
	Def. 1	Def. 2	Def. 3	Def. 4	Def. 1	Def. 2	Def. 3	Def. 4
El Padrino	4	8	3	6	MICHAEL	MICHAEL	MICHAEL	MICHAEL
					DON CORLEONE	DON CORLEONE	HAGEN	HAGEN
					HAGEN	HAGEN	DON CORLEONE	DON CORLEONE
					SONNY	SONNY		SONNY
						SOLLOZZO		CLEMENZA
						CLEMENZA		SOLLOZZO
						MAN		
						CARLO		
El Señor de los Anillos: La Comunidad del Anillo	11	8	2	8	FRODO	FRODO	FRODO	FRODO
					SAM	SAM	SAM	SAM
					GANDALF	GANDALF		GANDALF
					PIPPIN	PIPPIN		PIPPIN
					MERRY	MERRY		MERRY
					ARAGORN	ARAGORN		ARAGORN
					GIMLI	SARUMAN		BOROMIR
					BOROMIR	ARWEN		GIMLI
					STRIDER			
					LEGOLAS			
					ELROND			
Forrest Gump	2	4	3	3	FORREST	FORREST	FORREST	FORREST
					JENNY	JENNY	JENNY	JENNY
						LT DAN	LT DAN	LT DAN
						NEWSMAN		
Gladiator	5	6	2	5	MAXIMUS	MAXIMUS	MAXIMUS	MAXIMUS
					PROXIMO	PROXIMO	COMMODUS	COMMODUS
					COMMODUS	COMMODUS		PROXIMO
					LUCILLA	LUCILLA		LUCILLA
					GAIUS	ASSASSIN #3		GAIUS
						CASSIUS		
Cadena Perpetua	5	6	5	6	RED	RED	RED	RED
					ANDY	ANDY	ANDY	ANDY
					HEYWOOD	HEYWOOD	HEYWOOD	HEYWOOD
					HADLEY	HADLEY	HADLEY	HADLEY
					FLOYD	NORTON	FLOYD	FLOYD
						BROOKS		SNOOZE

Tabla 17. Ejemplo de protagonistas obtenidos con las definiciones creadas.

Si revisamos los resultados obtenidos, observamos que, bajo las 4 definiciones de protagonista, hay un mínimo de ellos que se repiten, lo que nos da a entender que las definiciones propuestas son buenas definiciones de protagonista.

Adicionalmente, se van a incluir los siguientes nuevos atributos en el dataframe de datos, para realizar otra iteración más de minería de datos:

1. # protagonistas en la película.
2. Popularidad máx. de los protagonistas de la película.
3. Popularidad media de los protagonistas de la película.

4. Popularidad min de los protagonistas de la película.
5. Relevancia máx. de los protagonistas de la película.
6. Relevancia media de los protagonistas de la película.
7. Relevancia min. de los protagonistas de la película.
8. Centralidad de Vector Propio Normalizada máx. de la película.
9. Centralidad de Vector Propio Normalizada media de la película.
10. Centralidad de Vector Propio Normalizada min de la película.

Añadiremos estos 10 nuevos atributos calculados para cada una de las definiciones propuestas, lo que hará un total de 31 nuevos atributos.

Esto hará que tengamos un dataframe con un total de 77 atributos (Ver ANEXO IV. DataFrame de Datos (fase 3 de experimentos)).

Previo a la realización de los experimentos, se aplicará al dataframe los métodos de selección de atributos que han sido detallados en secciones anteriores, con el objetivo de extraer los atributos más relevantes e intentar obtener unos mejores porcentajes de aciertos en la clasificación de las películas.

3.3.2. Resultados de Experimentos con Clasificación Binaria

En la Tabla 18, se recogen los resultados obtenidos en los experimentos realizados. En la primera columna se detalla el tipo de Algoritmo aplicado, en la segunda y tercera el porcentaje de clasificaciones correctas e incorrectas, respectivamente. En las últimas cinco columnas se resumen los valores de algunas métricas de calidad, para cada una de las clasificaciones.

Estos experimentos han sido realizados tras aplicar los métodos de selección detallados en la sección 2.2.5.

ALGORITMO	% Clas. OK	% Clas. KO	CLASE	Precision	Recall	F-Measure	ROC Area
OneR	48,86	51,14	C+	0,491	0,483	0,487	0,489
			C-	0,487	0,495	0,491	0,489
			Media	0,489	0,489	0,489	0,489
J48	61,58	38,42	C+	0,618	0,616	0,617	0,603
			C-	0,614	0,616	0,615	0,606
			Media	0,616	0,616	0,616	0,605
RandomForest	65,87	34,13	C+	0,656	0,672	0,664	0,69
			C-	0,661	0,645	0,653	0,69
			Media	0,659	0,659	0,659	0,69
NaiveBayes	54,88	45,12	C+	0,738	0,157	0,259	0,602
			C-	0,526	0,944	0,676	0,598
			Media	0,632	0,549	0,467	0,6
AdaBoost	62,38	37,62	C+	0,651	0,541	0,591	0,654
			C-	0,605	0,707	0,652	0,653
			Media	0,628	0,624	0,621	0,654

Tabla 18. Resultados Experimentos FASE 3 con clasificación Binaria.

Tras aplicar los métodos de selección detallados en la sección 2.2.5 y hacer experimentos con el subconjunto de atributos obtenidos con cada uno de ellos, se han obtenido los mejores resultados con la colección de atributos resultante de aplicar el evaluador "*ClassifierSubsetEval (J48)*" y "*BestFirst*" como método de búsqueda.

Con este método de selección de atributos se han seleccionado 28, que se listan a continuación:

- *year*
- *main genre*
- *max_degree*
- *density*
- *number_of_nodes*
- *size*
- *diameter*
- *radius*

- *max_degree_centrality*
- *min_degree_centrality*
- *average_closeness_centrality*
- *average_betweenness_centrality*
- *min_eigenvector_centrality*
- *average_communicability_centrality*
- *min_communicability_centrality*
- *max_communicability_betweenness_centrality*
- *min_communicability_betweenness_centrality*
- *number of cliques*
- *min_eigen_cent_norma*
- *avr_prota_pop_def_1*
- *min_prota_pop_def_1*
- *min_prota_rel_def_1*
- *num_protas_def_2*
- *min_prota_rel_def_2*
- *min_prota_rel_def_3*
- *num_protas_def_4*
- *avr_prota_pop_def_4*
- *min_prota_rel_def_4*

Cabe reseñar que, de los 28 atributos seleccionados, casi la mitad de ellos corresponden a los nuevos atributos que se definieron en la sub-sección anterior, lo que nos hace ver que los conceptos de "Número de Protagonistas", "Popularidad de los Protagonistas" y "Relevancia de los Protagonistas" han tenido importancia y han sido seleccionados por el método de selección aplicado.

Todos los algoritmos han sido ejecutados con "*10-fold Cross Validation*" y la configuración que WEKA asigna por defecto.

Por otro lado, los mejores resultados reportados en la Tabla 18 son los conseguidos con los algoritmos "*RandomForest*", "*J48*" y "*AdaBoost*", con un porcentaje de clasificaciones correctas del 66%, 66% y 63%, respectivamente. Se ha de notar también que se ha conseguido hasta un 74% de Valor Predictivo Positivo (*precision*) a la hora de decidir si una película tiene una valoración mayor que 7 (clase positiva), si bien para conseguir esta precisión se sacrifica mucha sensibilidad (*recall*). Esto quiere decir que habrá muchos casos positivos que serán clasificados como negativos (falsos negativos), sin embargo, este modelo da bastante seguridad en la predicción de verdaderos positivos. Esto se consigue basándonos únicamente en una representación en grafo de la estructura social de la película, donde un nodo es un personaje y una arista une dos personajes que interaccionan entre ellos.

La configuración de *RandomForest* con la que se ha conseguido los mejores resultados ha sido la que WEKA ofrece por defecto, esto es:

- *masDepth*: 0
- *numFeatures*: 5
- *numTrees*: 100
- *seed*: 1

Con esta configuración RandomForest construye 100 árboles, cada uno de ellos considerando 5 atributos aleatorios.

Para los experimentos realizados bajo *J48*, también se ha probado con varias configuraciones, pero la configuración con la que mejores resultados hemos obtenido ha sido la que WEKA asigna por defecto, esto es:

- *binarySplits*: False
- *confidenceFactor*: 0.25
- *debug*: False
- *minNumObj*: 2
- *numFolds*: 3
- *reducedErrorPruning*: False

- *saveInstanceData*: False
- *seed*: 1
- *subtreeRaising*: True
- *unpruned*: False
- *useLaplace*: False

Con esta configuración se obtiene el árbol de tamaño 204 y con 110 hojas. Se adjunta este árbol en el ANEXO V. Árbol Generado por J48 en fase 3 de experimentos (clasificación binaria). Los atributos que ha elegido el algoritmo para particionar los datos en clases individuales a la hora de generar las ramas han sido los siguientes:

- main genre
- year
- number of cliques
- size
- num_protas_def_4
- min_degree_centrality
- density
- radius
- min_prota_rel_def_2
- average_betweenness_centrality
- min_prota_rel_def_1
- min_eigen_cent_norma
- avr_prota_pop_def_1
- max_degree_centrality
- min_prota_pop_def_1
- max_degree
- avr_prota_pop_def_4
- num_protas_def_2
- average_communicability_centrality
- max_degree_centrality
- min_prota_rel_def_3

- max_communicability_betweenness_centrality

Para la aplicación de *Adaboost*, los mejores resultados se han obtenido con la configuración por defecto que aplica WEKA:

- *Classifier*: DecisionStump
- *Debug*: false
- *numIterations*: 10
- *seed*: 1
- *useResampling*: false
- *weightThreshold*: 100

En este caso, los atributos que han sido más relevantes para *Adaboost* se listan a continuación:

- year
- main genre
- min_prota_rel_def_3

En la Tabla 19, se muestra una comparación de los mejores resultados obtenidos en la Fase 2 y 3 de experimentos, en términos de precisión (*Accuracy*), bajo una definición de clasificación binaria.

Prescisión (*Accuracy*) en Clasif. Binaria					
RandomForest		AdaBoost		J48	
FASE 2	FASE 3	FASE 2	FASE 3	FASE 2	FASE 3
62%	66%	64%	63%	52%	66%

Tabla 19. Comparativa de mejores resultados obtenidos en fases 2 y 3 bajo clasificación Binaria.

Se observa que el porcentaje de clasificaciones correctas conseguidas con el algoritmo "*AdaBoost*" tiene un valor bastante parejo en las fases 2 y 3.

Sin embargo, en los resultados mejoran sustancialmente con "*RandomForest*" y "*J48*", con los que se observa una mejora de un 4% y del 14%, respectivamente en la tasa de acierto en la Fase 3 con respecto al conseguido en la Fase 2.

Se concluye por tanto que la inclusión en el dataframe de los nuevos atributos definidos (*Número de Protagonistas*, *Popularidad de los Protagonistas* y *Relevancia de los Protagonistas*) han supuesto una mejora en los resultados, del 4% con *RandomForest* y del 14% con *J48*.

3.3.3. Resultados de Experimentos con Clasificación Multi-Clase

En la Tabla 20, se recogen los resultados obtenidos en los experimentos realizados. En la primera columna se detalla el tipo de Algoritmo aplicado, en la segunda y tercera el porcentaje de clasificaciones correctas e incorrectas, respectivamente. En las últimas cinco columnas se resumen los valores de algunas métricas de calidad, para cada una de las clasificaciones.

Estos experimentos han sido realizados tras aplicar los métodos de selección detallados en la sección 2.2.5.

ALGORITMO	% Clas. OK	% Clas. KO	CLASE	Precision	Recall	F-Measure	ROC Area
OneR	21,28	78,72	C1	0,265	0,355	0,303	0,498
			C2	0,147	0,142	0,144	0,473
			C3	0,186	0,224	0,203	0,487
			C4	0,12	0,069	0,088	0,482
			C5	0,294	0,213	0,205	0,497
			Media	0,206	0,213	0,205	0,497
J48	24,09	75,91	C1	0,322	0,34	0,331	0,554
			C2	0,168	0,17	0,169	0,501
			C3	0,256	0,211	0,231	0,519
			C4	0,153	0,185	0,167	0,49
			C5	0,288	0,258	0,272	0,587
			Media	0,245	0,241	0,242	0,531
RandomForest	30,79	69,21	C1	0,378	0,59	0,461	0,642
			C2	0,25	0,163	0,197	0,541
			C3	0,244	0,211	0,226	0,545
			C4	0,167	0,115	0,136	0,481
			C5	0,344	0,339	0,341	0,715
			Media	0,284	0,608	0,287	0,587
NaiveBayes	27,17	72,83	C1	0,334	0,57	0,421	0,605
			C2	0,197	0,369	0,257	0,527
			C3	0,5	0,007	0,013	0,504
			C4	0,202	0,162	0,179	0,54
			C5	0,417	0,121	0,188	0,65
			Media	0,333	0,272	0,226	0,566
AdaBoost	32,26	67,74	C1	0,296	0,96	0,543	0,549
			C2	0	0	0	0,534
			C3	0	0	0	0,508
			C4	0	0	0	0,534
			C5	0,495	0,395	0,439	0,606
			Media	0,161	0,323	0,194	0,541

Tabla 20. Resultados Experimentos FASE 3 Clasificación Multi-Clase.

Tras aplicar los métodos de selección detallados en la sección 2.2.5 y hacer experimentos con el subconjunto de atributos obtenidos con cada uno de ellos, se han obtenido los mejores resultados con la colección de atributos resultante de aplicar el evaluador "*ClassifierSubsetEval (J48)*" y "*GeneticSearch*" como método de búsqueda.

Con este método de selección de atributos se han seleccionado 33, que se listan a continuación:

- *year*
- *main country*
- *average_degree*
- *density*
- *size*
- *average_degree_centrality*
- *max_degree_centrality*
- *max_betweenness_centrality*
- *min_betweenness_centrality*
- *max_edge_betweenness_centrality*
- *min_eigenvector_centrality*
- *average_communicability_centrality*
- *average_communicability_betweenness_centrality*
- *number of cliques*
- *transitivity*
- *average_clustering*
- *average_shortest_path_length*
- *min_eigen_cent_norma*
- *num_protas_def_1*
- *min_prota_pop_def_1*
- *max_prota_rel_def_1*
- *min_prota_rel_def_1*
- *num_protas_def_2*
- *max_prota_pop_def_2*
- *avr_prota_pop_def_2*
- *avr_prota_rel_def_2*
- *num_protas_def_3*

- *max_prota_pop_def_3*
- *avr_prota_rel_def_3*
- *min_prota_rel_def_3*
- *num_protas_def_4*
- *max_prota_pop_def_4*
- *avr_prota_pop_def_4*

Cabe reseñar que, de los 33 atributos seleccionados, la mitad de ellos (al igual que ha ocurrido en el enfoque de Clasificación Binaria) corresponden a los nuevos atributos que se definieron en la sub-sección anterior, lo que nos hace ver que los conceptos de "Número de Protagonistas", "Popularidad de los Protagonistas" y "Relevancia de los Protagonistas" han tenido importancia y han sido seleccionados por el método de selección aplicado.

Todos los algoritmos han sido ejecutados con "*10-fold Cross Validation*" y la configuración que WEKA asigna por defecto, ya que así es como se han conseguido los mejores resultados.

En esta fase de experimentos bajo una definición Multi-clase, observamos que los resultados obtenidos son bastante peores que los obtenidos con una definición de clase binaria. Los algoritmos que mejores resultados presentan siguen siendo "*AdaBoost*" y "*RandomForest*", aunque no se pueden considerar como resultados ni mucho menos concluyentes.

En la Tabla 21, se muestra una comparación de los resultados obtenidos en la Fase 2 y 3 de experimentos, en términos de precisión (*Accuracy*), bajo una definición de clasificación Multi-Clase.

Prescisión (*Accuracy*) en Clasif. Multi-Clase			
RandomForest		AdaBoost	
FASE 2	FASE 3	FASE 2	FASE 3
35%	31%	32%	32%

Tabla 21. Comparativa de mejores resultados obtenidos en fases 2 y 3 bajo clasificación Multi-Clase.

Se observa que, en este caso, no se aprecia ninguna mejora en los resultados obtenidos, incluso hay un empeoramiento del 4% en la precisión conseguida con *RandomForest*, lo que nos hace concluir que bajo el planteamiento de clasificación multi-clase, el hecho de haber añadido nuevos atributos al dataframe, no nos ha hecho conseguir mejores resultados.

Esto puede darse debido a que tenemos un caso de "*overfitting"* o *"underfitting"*, ya que estas son las 2 causas más comunes de malos rendimientos en algoritmos de *Machine Learning*.

3.4. ANÁLISIS DE OVERFITTING VS. UNDERFITTING

Como regla general para saber si tenemos un caso de "*overfitting*" o "*underfitting*", se pueden aplicar estas premisas:

- Tendremos un caso de *Overfitting*, cuando se obtiene un buen rendimiento del algoritmo con los datos de entrenamiento y un pobre rendimiento con otros datos de test (pobre generalización). Si esto ocurre, es señal de que el modelo comienza a "memorizar" datos de entrenamiento en lugar de "aprender" para generalizar.

- Tendremos un caso de *Underfitting*, cuando obtenemos un mal rendimiento del algoritmo con los datos de entrenamiento y también un pobre rendimiento con otros datos (pobre generalización).

Para comprobar esto, una buena práctica es comparar el rendimiento de los algoritmos utilizando como opciones de test, por un lado, "*conjunto de entrenamiento*" y por otro "*cross-validation"*. Esto normalmente da una indicación de si el modelo está "sobreajustando" los datos de entrenamiento. Compararemos el error obtenido con "*conjunto de entrenamiento*" del obtenido con "*cross-validation*". Si existe una gran diferencia en estos dos valores, será indicativo de que existe un "sobreajuste" de los datos.

Para realizar la comparación, analizaremos los resultados obtenidos con los algoritmos *J48 y RandomForest* (ya que éstos son los que mejor rendimiento han presentado).

Análisis bajo Clasificación Binaria

Para el caso de clasificación binaria, al comprobar el rendimiento de *J48*, cuando ejecutamos el test aplicándole un conjunto de datos de entrenamiento, obtenemos una precisión del 95,45% de acierto, mientras que aplicándole validación cruzada obtenemos una precisión del 62%.

En las ejecuciones con *RandomForest*, conseguimos una precisión del 100% ejecutando el test con un conjunto de datos de entrenamiento y una precisión del 66% cuando lo ejecutamos con validación cruzada.

Podemos observar que para ambos algoritmos (en clasificación binaria) hay una gran diferencia en el rendimiento, cuando comparamos los resultados obtenidos con un conjunto de datos de entrenamiento respecto de los obtenidos con validación cruzada, siendo ésta de alrededor de un 33%.

Análisis bajo Clasificación Multi-clase

Para el caso de clasificación multi-clase, al comprobar el rendimiento de *J48*, cuando ejecutamos el test aplicándole un conjunto de datos de entrenamiento, obtenemos una precisión del 89,7% de acierto, mientras que aplicándole validación cruzada obtenemos una precisión del 24,1%.

En las ejecuciones con *RandomForest*, conseguimos una precisión del 100% ejecutando el test con un conjunto de datos de entrenamiento y una precisión del 30,8% cuando lo ejecutamos con validación cruzada.

Podemos observar que para ambos algoritmos (en clasificación multi-clase) hay una diferencia más que considerable en el rendimiento, siendo ésta de más de un 65%.

En la siguiente tabla, se resume el análisis realizado.

		Precisión *(Accuracy)*	
		BINARIA	**MULTICLASE**
J48	training set	95,45%	89,70%
	cross-val	61,58%	24,10%
RandomForest	training set	100%	100%
	cross-val	65,86%	30,80%

Tabla 22. Comparativa de rendimientos.

Si nos fijamos en la precisión del porcentaje de acierto cuando ejecutamos los algoritmos con un conjunto de datos de entrenamiento, vemos que éste es bastante alto, lo que nos hace rechazar el planteamiento de que tengamos un caso de *underfitting*.

Sin embargo, el hecho de obtener un pobre rendimiento al aplicar un conjunto de datos con validación cruzada, nos muestra que posiblemente estemos ante un caso de *overfitting*, siendo éste todavía más marcado en el enfoque de clasificación multi-clase.

Para prevenir casos de *overfitting*, se sugiere aplicar estas tres técnicas, que serán tratadas con mayor detalle en una próxima sección (CONCLUSIONES):

- Incrementar la cantidad de muestras en los datos de entrenamiento, es decir, intentar obtener un mayor número de instancias de películas e incluirlas en nuestro dataframe.
- Reducir el número de atributos del dataframe, ya que, aun aplicando métodos de selección de atributos, parece que esto no es suficiente y deberíamos ser más restrictivos en la fase de obtención de datos.
- Incrementar la regularización, en el sentido de mejorar la generalización en los datos del modelo de aprendizaje.

4. CONCLUSIONES Y TRABAJOS FUTUROS

4.1. CONCLUSIONES

En esta sección se establecerán las conclusiones del trabajo realizado, para lo que nos apoyaremos fundamentalmente en los datos y observaciones obtenidos durante todo el desarrollo.

1. Se ha conseguido hasta casi un 74% de Valor Predictivo Positivo (*precision*) en la predicción de películas con valoraciones positivas (mayores que 7), si bien es cierto que para conseguir esta precisión se sacrifica mucha sensibilidad (*recall*). Esto quiere decir que habrá muchos casos de falsos negativos (FN), sin embargo, este modelo da bastante seguridad en la predicción de verdaderos positivos (TP).
2. Hemos conseguido basarnos únicamente en la representación en grafo de la estructura social de la película, donde un nodo es un personaje y una arista une dos personajes que interaccionan entre ellos, para predecir las valoraciones.
3. No se ha encontrado correlación en los datos para conseguir predicciones de valoraciones de películas bajo el planteamiento de clasificación multi-clase, lo cual era de esperar, dada la desafiante naturaleza del problema.
4. La definición e inclusión de nuevos conceptos en el dataframe, como han sido "Número de Protagonistas", "Popularidad de los Protagonistas" y "Relevancia de los Protagonistas", ha mejorado de calidad de los datos.
5. Se ha detectado un caso de *overfitting* en el modelo de clasificación, más marcado si cabe en el problema de clasificación Multi-clase.
6. Es un planteamiento erróneo el hecho de incluir la clase a predecir en la colección de datos (*rating* en nuestro caso), pues esto provoca que métodos de clasificación simples, como son *J48*, *OneR*, proporcionen resultados con un 100% de precisión en las clasificaciones.

4.2. RESUMEN DE CONTRIBUCIONES

En esta sección se listan las nuevas contribuciones que este trabajo aporta al conocimiento, en lo que se refiere a la temática de predicción de valoraciones de películas.

1. Se han utilizado únicamente métricas de SNA correspondientes al grafo de estructura de red social de películas, para predecir las valoraciones de las mismas.
2. En algunos casos se han conseguido predicciones de valoraciones de películas antes de que se haya estrenado, publicado trailers y de que se hayan hecho comentarios sobre ellas en los medios sociales.
3. Se han creado 4 definiciones de Protagonista de una película, basadas en métricas de SNA
4. Se han definido nuevos atributos, como son el Número de Protagonistas, la Popularidad de un Personaje y la Relevancia de un Personaje, en función de métricas de SNA.
5. Se han evaluado adecuadamente las medidas de calidad de las técnicas de clasificación utilizadas, utilizando para ello el % de precisión (*accuracy*), tal y como se recomienda en [8, p. 417] y mejorando lo aportado por [3].

4.3. TRABAJOS FUTUROS

En esta sección discutiremos los puntos de acción que deberían ser abordados en un futuro trabajo, para llegar a unos resultados más satisfactorios para el problema planteado inicialmente y llevar a cabo una implantación real. Para intentar prevenir los problemas de *overfitting* comentados en la sección anterior, y también para llegar a unos resultados más satisfactorios para los problemas planteados inicialmente, se proponen las siguientes acciones como trabajos futuros a implementar:

1. Incrementar la cantidad de muestras en los datos de entrenamiento, es decir, intentar obtener un mayor número de instancias de películas e incluirlas en el dataframe.
2. Reducir el número de atributos del dataframe, ya que, aun aplicando métodos de selección de atributos, parece que esto no es suficiente y se debería ser más restrictivos en la fase de obtención de datos.

3. Incrementar la regularización, en el sentido de mejorar la generalización en los datos del modelo de aprendizaje. Por ejemplo, se propone implementar la siguiente generalización en estos atributos:
    - **Presupuesto**: Se podría implementar la definición de este atributo para que aceptase valores del tipo "Alto", "Medio" y "Bajo", donde:
        - Alto: presupuesto > 100.000.000 $
        - Medio: 15.000.000 $ <= presupuesto <= 100.000.000 $
        - Bajo: presupuesto < 15.000.000 $

      Bajo estas definiciones aumentaría la generalización en los valores de ese atributo, ya que, si nos fijamos en el histograma de la distribución de sus valores, más de un tercio de los mismos se encuentran cercanos al valor mínimo.
    - **Año**: Se podría implementar la definición de este atributo para que aceptase valores del tipo "Actual", "2000's" y "Antigua", donde:
        - Actual: año > 2010
        - 2000's: 1990 <= año <= 2010
        - Antigua: año < 1990

      Bajo estas definiciones aumentaría la generalización en los valores de ese atributo, ya que, si nos fijamos en el histograma de la distribución de sus valores, los rangos definidos tendrían el mismo número de instancias de este atributo.
    - **País**: Se podría implementar la definición de este atributo para que aceptase valores del tipo "USA", "UK", "France" y "Others", donde:
        - Others: serían todos los países con menos de 21 instancias.
    - **Género**: Se podría implementar la definición de este atributo para que aceptase valores del tipo "*Action*", "*Comedy*", "*Drama*", "*Crime*" y "Others", donde:
        - Others: serían todos los géneros con menos de 78 instancias.
4. Dar más relevancia a los personajes con mayor peso en sus enlaces para el cálculo de las métricas de SNA o en la definición de nuevos atributos. En definitiva, tener

en cuenta los pesos de los enlaces en cada uno de los grafos que representan cada película.

5. Investigar nuevas fuentes de información, de manera que podamos extraer nuevos atributos por cada película.
6. Analizar el "Resumen de la Trama" de IMDb e intentar extraer nuevos atributos a través de la aplicación de técnicas de *Sentiment Analysis*, como por ejemplo qué estados de ánimo (tristeza, alegría, desesperación, amor), que puedan provocar una cierta predisposición al espectador para sentirlos.

5. BIBLIOGRAFÍA

[1] The University of Waikato, «WEKA 3: Data Mining Software in Java,» [En línea]. Available: http://www.cs.waikato.ac.nz/ml/weka/.

[2] A. Oghina, M. Breuss, M. Tsagkias y M. de Rijke, «Predicting IMDB Movie Ratings Using Social Media,» de *DOI: 10.1007/978-3-642-28997-2_51,* 2012.

[3] V. R. Nithin, M. Pranav, B. Sarath y A. Lijiya , «Predicting Movie Success Based on IMDB Data,» *International Journal of Data Mining Techniques and Applications,* vol. 03, pp. 365-368, 2014.

[4] M. Hassan Latif y H. Afzal, «Prediction of Movies popularity Using Machine Learning Techniques,» *International Journal of Computer Science and Network Security,* vol. 16, nº 8, pp. 127-131, August 2016.

[5] L. Doshi, J. Krauss, S. Nann y P. Gloor, «Predicting Movie Prices Through Dynamic Social Network Analysis,» *Procedia Social and Behavioral Sciences,* pp. 6423-6433, 2010.

[6] J. S. Krauss, K. Fischbach, D. Simon y P. Gloor, «Predicting Movie Success and Academy Awards through Sentiment and Social Network Analysis,» *16th European Conference on Information Systems,* 2008.

[7] IMDb, «IMDb - Movies, TV and Celebrities - IMDb,» [En línea]. Available: http://www.imdb.com.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann y I. H. Witten, «The WEKA Data Mining Software: An Update,» *SIGKDD Explorations,* vol. 11, nº 1, pp. 10-18, 2009.

[9] I. H. Witten y E. Frank, Data Mining Practical Machine Learning Tools and Techniques, 2 ed., M. Kaufmann, Ed., ELSEVIER, 2005.

[10] S. Kabinsingha, S. Chindasorn y C. Chantatrapornchai, «A Movie Rating Approach and Application Based on Data Mining,» *International Journal of Engineering and Innovative Technology (IJEIT),* vol. 2, nº 1, pp. 77-83, 2012.

[11] X. Amatriain, A. Jaimes, N. Oliver y J. M. Pujol, «Data Mining Methods for Recommender Systems,» pp. 39-71.

[12] J. Han, M. Kamber y J. Pei, «Data Mining Concepts and Techniques,» 3rd ed., M. Kaufman, Ed., ELSEVIER, 2012, pp. 85-124.

[13] S. Saraee, S. White y J. Eccleston, «A data mining approach to analysis and prediction of movie ratings,» pp. 344-352, 2004.

[14] N. Apte, M. Forssell y A. Sidhwa, «Predicting Movie Revenue,» 2011.

[15] HSX, «HSX.com – Hollywood Stock Exchange. Trade movies, stars and more.,» [En línea]. Available: http://www.hsx.com/.

[16] S. Wassermann y K. Faust, Social Network Analysis, Cambridge University Press, 1994.

[17] Box Office Mojo, «Box Office Mojo,» [En línea]. Available: www.boxofficemojo.com.

[18] P. A. Gloor, R. Laubacher , S. B. C. Dynes y Y. Zhao, «Visualization of Communication Patterns in Collaborative Innovation Networks - Analysis of some W3C working groups,» 2003.

[19] LIWC, «LIWC - Linguistic Inquiry and Word Count,» [En línea]. Available: http://liwc.wpengine.com/.

[20] L. Zhuang, F. Jing y X. Zhu, «Movie Review Mining and Summarization,» pp. 43-50.

[21] P. Chaovalit y L. Zhou, «Movie Review Mining: a Comparison between Supervised and Unsupervised,» de *Proceedings of the 38th Hawaii International Conference on System Sciences*, 2005.

[22] The igraph core team, «igraph-The network analysis package,» 2003. [En línea]. Available: http://igraph.org/redirect.html.

[23] A. Hagberg, P. Swart y D. Schult, «NetworkX,» 2015. [En línea]. Available: http://networkx.readthedocs.io/en/networkx-1.10/index.html.

[24] A. Malagoli, «IMDbPY,» 2014. [En línea]. Available: http://imdbpy.sourceforge.net/index.html.

[25] J. Kaminski, M. Schober, R. Albaladejo, O. Zastupailo y C. Hidalgo, «Moviegalaxies - Social Networks in Movies,» August 2012. [En línea]. Available: http://moviegalaxies.com.

[26] M. Newman, Networks: An Introduction, Oxford University Press, 2010.

[27] P. Bonacich, «Factoring and weighting approaches to status scores and clique identification,» *The Journal of Mathematical Sociology*, pp. 113-120, 1972.

[28] E. Estrada y J. A. Rodriguez-Velazquez, «Subgraph Centrality in Complex Networks,» *Physical Review E 71, 056103*, 2005.

[29] E. Estrada y Hatano N., «Communicability in complex networks,» *Physical Review E 77, 036111,* 2008.

[30] I. H. Witten y E. Frank, «Data Mining-Practical Machine Learning Tools and Techniques,» Second Edition ed., M. Kaufmann, Ed., ELSEVIER, 2005, pp. 420-425.

[31] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[32] R. Quinlan, Programs for Machine Learning, San Mateo: Morgan Kaufmann, 1993.

[33] L. Breiman, Random Forest. Machine Learning, R. E. Shapire, Ed., 2001, pp. 5-32.

[34] Y. Freund y R. E. Shapire, «Experiments with a new boosting algorithm,» de *Thirteenth International Conference on Machine Learning*, San Francisco, 1996.

[35] G. H. John y P. Langley, «Estimating Continuous Distributions in Bayesian Classifiers,» de *Eleventh Conference on Uncertainty in Artificial Intelligence*, San Mateo, 1995.

[36] R. C. Holte, «Very simple classification rules perform well on most commonly used datasets.,» 1993, pp. 63-91.

[37] Wikipedia, «Confusion Matrix,» [En línea]. Available: https://en.wikipedia.org/wiki/Confusion_matrix.

[38] M. Bastian, S. Heymann y M. Jacomy, «Gephi: an open source software for exploring and manipulating networks,» de *International AAAI Conference on Weblogs and Social Media*, 2009.

[39] S. Yoo, R. Kanter y D. Cummings, «Predicting Movie Revenue from IMDb Data».

6. ANEXO I. DATAFRAME DE DATOS (FASE 1 Y 2 DE EXPERIMENTOS)

# INSTANCIAS TOTALES = 755				
ATRIBUTO	**TIPO**	**# INSTANCIAS DIFERENTES**	**# INSTANCIAS MISSING**	**COMENTARIOS**
'budget ($)'	Numeric	158	93	
'year'	Numeric	72	1	
'main director'	Nominal	470	4	
'main character'	Nominal	464	1	
'secondary character'	Nominal	620	2	
'third character'	Nominal	658	3	
'main writer'	Nominal	597	4	
'main producer'	Nominal	551	4	
'main country'	Nominal	16	2	
'main genre'	Nominal	17	1	
secondary genre'	Nominal	21	74	
'third genre'	Nominal	19	269	
'main costume designer'	Nominal	332	65	
main sound'	Nominal	409	6	
'main editor'	Nominal	431	9	
'main production manager'	Nominal	568	26	
'main visual effects'	Nominal	436	151	
'main cinematographer'	Nominal	377	12	
'make up'	Nominal	483	29	
'main production company'	Nominal	316	4	
'average_degree'	Numeric	585	0	
'max_degree'	Numeric	64	0	
'min_degree'	Numeric	4	0	
'density'	Numeric	290	0	
'number_of_nodes'	Numeric	76	0	
'size'	Numeric	234	0	
'diameter'	Numeric	9	0	
'radius'	Numeric	5	0	
'average_eccentricity'	Numeric	511	0	
'max_eccentricity'	Numeric	9	0	
'min_eccentricity'	Numeric	5	0	
'average_degree_centrality'	Numeric	290	0	
'max_degree_centrality'	Numeric	284	0	
'min_degree_centrality'	Numeric	69	0	

'average_closeness_centrality'	Numeric	249	0	
'max_closeness_centrality'	Numeric	270	0	
'min_closeness_centrality'	Numeric	245	0	
average_betweenness_centrality'	Numeric	73	0	
'max_betweenness_centrality'	Numeric	463	0	
'min_betweenness_centrality'	Numeric	1	0	Todos los valores son 0
'average_edge_betweenness_centrality'	Numeric	81	0	
'max_edge_betweenness_centrality'	Numeric	203	0	
'min_edge_betweenness_centrality'	Numeric	20	0	
'average_eigenvector_centrality'	Numeric	151	0	
'max_eigenvector_centrality'	Numeric	248	0	
'min_eigenvector_centrality'	Numeric	53	0	
'average_communicability_centrality'	Numeric	753	0	
'max_communicability_centrality'	Numeric	753	0	
'min_communicability_centrality'	Numeric	717	0	
'average_communicability_betweenness_centrality'	Numeric	246	0	
'max_communicability_betweenness_centrality'	Numeric	316	0	
'min_communicability_betweenness_centrality'	Numeric	55	0	
'average_load_centrality'	Numeric	73	0	
'max_load_centrality'	Numeric	482	0	
min_load_centrality'	Numeric	1	0	Todos los valores son 0
'node_connectivity'	Numeric	4	0	
'degree_assortativity'	Numeric	353	0	Valores negativos
'number of cliques'	Numeric	63	0	
'average_triangles'	Numeric	653	0	
'transitivity'	Numeric	393	0	
'average_clustering'	Numeric	309	0	
'average_shortest_path_length	Numeric	499	0	

7. ANEXO II. SELECCIÓN DE ATRIBUTOS EN CLASIFICACIÓN BINARIA

- **CfsSubsetEval + BestFirst, GreedyStepwise, LinearForwardSelection**:

  Atributos Seleccionados (4): 2,5,6,7

  year
  secondary character
  third character
  main writer

- **CfsSubsetEval +GeneticSearch:**

  Atributos Seleccionados (13): 2,3,4,5,7,8,9,14,16,17,20,56,57

  year
  main director
  main character
  secondary character
  main writer
  main producer
  main country
  main sound
  main production manager
  main visual effects
  main production company
  node_connectivity
  degree_assortativity

- **ClassifierSubsetEval (con J48) + BestFirst:**

  Atributos Seleccionados (20):
2,10,11,22,24,26,27,29,33,34,38,46,47,49,50,51,52,58,61,62

year
main genre
secondary genre
max_degree
density
size
diameter
average_eccentricity
max_degree_centrality
min_degree_centrality
average_betweenness_centrality
min_eigenvector_centrality
average_communicability_centrality
min_communicability_centrality
average_communicability_betweenness_centrality
max_communicability_betweenness_centrality
min_communicability_betweenness_centrality
number of cliques
average_clustering
average_shortest_path_length

- **ClassifierSubsetEval (con J48) + GeneticSearch:**

  Atributos Seleccionados (21):
  1,10,22,28,29,30,32,34,35,37,40,42,43,44,46,50,51,53,54,60,61
  budget ($)
  main genre
  max_degree
  radius
  average_eccentricity
  max_eccentricity
  average_degree_centrality

min_degree_centrality
average_closeness_centrality
min_closeness_centrality
min_betweenness_centrality
max_edge_betweenness_centrality
min_edge_betweenness_centrality
average_eigenvector_centrality
min_eigenvector_centrality
average_communicability_betweenness_centrality
max_communicability_betweenness_centrality
average_load_centrality
max_load_centrality
transitivity
average_clustering

- **ClassifierSubsetEval (con J48) + GreedyStepwise:**

  Atributos Seleccionados (8): 2,10,11,22,26,38,50,52

  year
  main genre
  secondary genre
  max_degree
  size
  average_betweenness_centrality
  average_communicability_betweenness_centrality
  min_communicability_betweenness_centrality

- **ClassifierSubsetEval (con J48) + LinearForwardSelection:**

  Atributos Seleccionados (13): 2,10,11,32,34,38,41,44,49,50,51,52,61

  year
  main genre

secondary genre

average_degree_centrality

min_degree_centrality

average_betweenness_centrality

average_edge_betweenness_centrality

average_eigenvector_centrality

min_communicability_centrality

average_communicability_betweenness_centrality

max_communicability_betweenness_centrality

min_communicability_betweenness_centrality

average_clustering

8. ANEXO III. SELECCIÓN DE ATRIBUTOS EN CLASIFICACIÓN MULTICLASE

- **CfsSubsetEval + BestFirst, GreedyStepwise, LinearForwardSelection**:

  Atributos Seleccionados (1): 6

  third character

- **CfsSubsetEval +GeneticSearch:**

  Atributos Seleccionados (10): 1,4,5,6,7,18,19,20,56,57

  budget ($)

  main character

  secondary character

  third character

  main writer

  main cinematographer

  make up

  main production company

  node_connectivity

  degree_assortativity

- **ClassifierSubsetEval (con J48) + BestFirst:**

  Atributos Seleccionados (6): 3,7,22,25,44,45

  main director

main writer

max_degree

number_of_nodes

average_eigenvector_centrality

max_eigenvector_centrality

- **ClassifierSubsetEval (con J48) + GeneticSearch:**

  Atributos Seleccionados (16): 7,8,17,18,22,25,26,27,28,30,36,40,45,47,60,62

  main writer

  main producer

  main visual effects

  main cinematographer

  max_degree

  number_of_nodes

  size

  diameter

  radius

  max_eccentricity

  max_closeness_centrality

  min_betweenness_centrality

  max_eigenvector_centrality

average_communicability_centrality

transitivity

average_shortest_path_length

- **ClassifierSubsetEval (con J48) + GreedyStepwise:**

  Atributos Seleccionados (6): 3,7,22,25,44,45

  main director

  main writer

  max_degree

  number_of_nodes

  average_eigenvector_centrality

  max_eigenvector_centrality

- **ClassifierSubsetEval (con J48) + LinearForwardSelection:**

  Atributos Seleccionados (6): 3,7,22,25,44,45

  main director

  main writer

  max_degree

  number_of_nodes

average_eigenvector_centrality

max_eigenvector_centrality

9. ANEXO IV. DATAFRAME DE DATOS (FASE 3 DE EXPERIMENTOS)

# INSTANCIAS TOTALES = 755				
ATRIBUTO	**TIPO**	**# INSTANCIAS DIFERENTES**	**# INSTANCIAS MISSING**	**COMENTARIOS**
'budget ($)'	Numeric	158	93	
'year'	Numeric	72	1	
'main country'	Nominal	16	2	
'main genre'	Nominal	17	1	
'average_degree'	Numeric	585	0	
'max_degree'	Numeric	64	0	
'min_degree'	Numeric	4	0	
'density'	Numeric	290	0	
'number_of_nodes'	Numeric	76	0	
'size'	Numeric	234	0	
'diameter'	Numeric	9	0	
'radius'	Numeric	5	0	
'average_eccentricity'	Numeric	511	0	
'max_eccentricity'	Numeric	9	0	
'min_eccentricity'	Numeric	5	0	
'average_degree_centrality'	Numeric	290	0	
'max_degree_centrality'	Numeric	284	0	
'min_degree_centrality'	Numeric	69	0	
'average_closeness_centrality'	Numeric	249	0	
'max_closeness_centrality'	Numeric	270	0	
'min_closeness_centrality'	Numeric	245	0	
average_betweenness_centrality'	Numeric	73	0	

'max_betweenness_centrality'	Numeric	463	0	
'min_betweenness_centrality'	Numeric	1	0	Todos los valores son 0
'average_edge_betweenness_centrality'	Numeric	81	0	
'max_edge_betweenness_centrality'	Numeric	203	0	
'min_edge_betweenness_centrality'	Numeric	20	0	
'average_eigenvector_centrality'	Numeric	151	0	
'max_eigenvector_centrality'	Numeric	248	0	
'min_eigenvector_centrality'	Numeric	53	0	
'average_communicability_centrality'	Numeric	753	0	
'max_communicability_centrality'	Numeric	753	0	
'min_communicability_centrality'	Numeric	717	0	
'average_communicability_betweenness_centrality'	Numeric	246	0	
'max_communicability_betweenness_centrality'	Numeric	316	0	
'min_communicability_betweenness_centrality'	Numeric	55	0	
'average_load_centrality'	Numeric	73	0	
'max_load_centrality'	Numeric	482	0	
'min_load_centrality'	Numeric	1	0	Todos los valores son 0
'node_connectivity'	Numeric	4	0	
'degree_assortativity'	Numeric	353	0	Valores negativos
'number of cliques'	Numeric	63	0	
'average_triangles'	Numeric	653	0	
'transitivity'	Numeric	393	0	
'average_clustering'	Numeric	309	0	
'average_shortest_path_length	Numeric	499	0	
'max_eigen_cent_norma'	Numeric	1	0	son todos 1
'avr_eigen_cent_norma'	Numeric	254	0	

'min_eigen_cent_norma'	Numeric	74	0	
'num_protas_def_1'	Numeric	17	0	
'max_prota_pop_def_1'	Numeric	308	0	
'avr_prota_pop_def_1'	Numeric	399	0	
'min_prota_pop_def_1'	Numeric	336	0	
'max_prota_rel_def_1'	Numeric	1	0	son todos 1
'avr_prota_rel_def_1'	Numeric	241	0	
'min_prota_rel_def_1'	Numeric	159	0	
'num_protas_def_2'	Numeric	18	0	Hay 2 películas con 0 protagonistas bajo esta definición
'max_prota_pop_def_2'	Numeric	308	2	
'avr_prota_pop_def_2'	Numeric	410	2	
'min_prota_pop_def_2'	Numeric	308	2	
'max_prota_rel_def_2'	Numeric	1	2	son todos 1
'avr_prota_rel_def_2'	Numeric	375	2	
'min_prota_rel_def_2'	Numeric	288	2	
'num_protas_def_3'	Numeric	14	0	Hay 33 películas con 0 protagonistas bajo esta definición
'max_prota_pop_def_3'	Numeric	298	33	
'avr_prota_pop_def_3'	Numeric	359	33	
'min_prota_pop_def_3'	Numeric	321	33	
'max_prota_rel_def_3'	Numeric	1	33	son todos 1
'avr_prota_rel_def_3'	Numeric	303	33	
'min_prota_rel_def_3'	Numeric	251	33	
'num_protas_def_4'	Numeric	26	0	Hay 5 películas con 0 protagonistas

				bajo esta definición
'max_prota_pop_def_4'	Numeric	308	5	
'avr_prota_pop_def_4'	Numeric	419	5	
'min_prota_pop_def_4'	Numeric	318	5	
'max_prota_rel_def_4'	Numeric	1	5	son todos 1
'avr_prota_rel_def_4'	Numeric	352	5	
'min_prota_rel_def_4'	Numeric	232	5	

10.ANEXO V. ÁRBOL GENERADO POR J48 EN FASE 3 DE EXPERIMENTOS (CLASIFICACIÓN BINARIA)

```
main genre = Comedy
|   year <= 1989
|   |   number of cliques <= 43
|   |   |   num_protas_def_2 <= 2: C- (3.0/1.0)
|   |   |   num_protas_def_2 > 2: C+ (33.0/4.0)
|   |   number of cliques > 43: C- (2.0)
|   year > 1989
|   |   size <= 97
|   |   |   num_protas_def_4 <= 1: C- (6.0)
|   |   |   num_protas_def_4 > 1
|   |   |   |   year <= 2003
|   |   |   |   |   max_degree <= 14: C- (3.0)
|   |   |   |   |   max_degree > 14
|   |   |   |   |   |   min_eigen_cent_norma <= 0.04: C+ (23.0/3.0)
|   |   |   |   |   |   min_eigen_cent_norma > 0.04: C- (2.0)
|   |   |   |   year > 2003
|   |   |   |   |   num_protas_def_4 <= 3: C+ (5.0/1.0)
|   |   |   |   |   num_protas_def_4 > 3
|   |   |   |   |   |   min_degree_centrality <= 0.048: C- (10.0)
|   |   |   |   |   |   min_degree_centrality > 0.048
|   |   |   |   |   |   |   density <= 0.372: C+ (2.0)
|   |   |   |   |   |   |   density > 0.372: C- (2.0)
|   |   size > 97
|   |   |   radius <= 2
|   |   |   |   min_prota_rel_def_2 <= 0.136
|   |   |   |   |   average_betweenness_centrality <= 0.026: C+ (9.0/1.0)
|   |   |   |   |   average_betweenness_centrality > 0.026: C- (3.0)
|   |   |   |   min_prota_rel_def_2 > 0.136
|   |   |   |   |   num_protas_def_4 <= 6: C- (28.0/1.0)
|   |   |   |   |   num_protas_def_4 > 6
|   |   |   |   |   |   min_prota_rel_def_1 <= 0.757
|   |   |   |   |   |   |   year <= 2008
|   |   |   |   |   |   |   |   min_eigen_cent_norma <= 0.001: C+ (3.0/1.0)
|   |   |   |   |   |   |   |   min_eigen_cent_norma > 0.001: C- (24.0)
|   |   |   |   |   |   |   year > 2008: C+ (4.0/1.0)
|   |   |   |   |   |   min_prota_rel_def_1 > 0.757: C+ (3.0)
|   |   |   radius > 2: C- (6.0)
```

```
main genre = Drama
|   year <= 1983: C+ (26.0)
|   year > 1983
|   |   radius <= 2
|   |   |   radius <= 1
|   |   |   |   avr_prota_pop_def_1 <= 0.78: C- (6.0)
|   |   |   |   avr_prota_pop_def_1 > 0.78: C+ (3.0)
|   |   |   radius > 1
|   |   |   |   num_protas_def_4 <= 9
|   |   |   |   |   max_degree_centrality <= 0.96
|   |   |   |   |   |   min_prota_pop_def_1 <= 0.85
|   |   |   |   |   |   |   size <= 129
|   |   |   |   |   |   |   |   number of cliques <= 27
|   |   |   |   |   |   |   |   |   number of cliques <= 23
|   |   |   |   |   |   |   |   |   |   max_degree <= 22
|   |   |   |   |   |   |   |   |   |   |   avr_prota_pop_def_4 <= 0.753
|   |   |   |   |   |   |   |   |   |   |   |   num_protas_def_2 <= 3: C+ (10.0)
|   |   |   |   |   |   |   |   |   |   |   |   num_protas_def_2 > 3
|   |   |   |   |   |   |   |   |   |   |   |   |   average_communicability_centrality <= 118.548
|   |   |   |   |   |   |   |   |   |   |   |   |   |   density <= 0.163: C+ (4.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   density > 0.163
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   density <= 0.294: C- (7.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   density > 0.294: C+ (2.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   average_communicability_centrality > 118.548:
C+ (6.0)
|   |   |   |   |   |   |   |   |   |   |   avr_prota_pop_def_4 > 0.753: C- (4.0)
|   |   |   |   |   |   |   |   |   |   max_degree > 22: C- (9.0/1.0)
|   |   |   |   |   |   |   |   |   number of cliques > 23: C- (7.0)
|   |   |   |   |   |   |   |   number of cliques > 27: C+ (5.0)
|   |   |   |   |   |   |   size > 129: C- (6.0)
|   |   |   |   |   |   min_prota_pop_def_1 > 0.85: C+ (13.0/1.0)
|   |   |   |   |   max_degree_centrality > 0.96: C- (7.0)
|   |   |   |   num_protas_def_4 > 9: C+ (8.0/1.0)
|   |   radius > 2
|   |   |   min_eigen_cent_norma <= 0.002: C+ (15.0)
|   |   |   min_eigen_cent_norma > 0.002
|   |   |   |   max_degree <= 17: C+ (3.0)
|   |   |   |   max_degree > 17: C- (3.0)
main genre = Adventure
|   min_prota_rel_def_4 <= 0.366: C+ (9.0)
|   min_prota_rel_def_4 > 0.366
```

```
|   |   max_degree <= 18: C+ (6.0)
|   |   max_degree > 18
|   |   |   density <= 0.138: C+ (4.0)
|   |   |   density > 0.138
|   |   |   |   number of cliques <= 18
|   |   |   |   |   max_degree_centrality <= 0.895: C+ (5.0)
|   |   |   |   |   max_degree_centrality > 0.895: C- (5.0/1.0)
|   |   |   |   number of cliques > 18: C- (14.0)
main genre = Action
|   year <= 1984
|   |   diameter <= 3
|   |   |   min_degree_centrality <= 0.034: C- (2.0)
|   |   |   min_degree_centrality > 0.034: C+ (7.0/1.0)
|   |   diameter > 3: C+ (10.0)
|   year > 1984
|   |   radius <= 2
|   |   |   radius <= 1
|   |   |   |   size <= 77: C- (6.0/1.0)
|   |   |   |   size > 77: C+ (2.0)
|   |   |   radius > 1
|   |   |   |   diameter <= 3
|   |   |   |   |   min_eigenvector_centrality <= 0.002: C- (6.26/0.13)
|   |   |   |   |   min_eigenvector_centrality > 0.002
|   |   |   |   |   |   min_eigenvector_centrality <= 0.003: C+ (3.13/0.07)
|   |   |   |   |   |   min_eigenvector_centrality > 0.003
|   |   |   |   |   |   |   year <= 2000
|   |   |   |   |   |   |   |   average_closeness_centrality <= 0.551
|   |   |   |   |   |   |   |   |   min_prota_rel_def_2 <= 0.393
|   |   |   |   |   |   |   |   |   |   year <= 1994: C- (2.0)
|   |   |   |   |   |   |   |   |   |   year > 1994: C+ (2.0)
|   |   |   |   |   |   |   |   |   min_prota_rel_def_2 > 0.393: C+ (6.0)
|   |   |   |   |   |   |   |   average_closeness_centrality > 0.551
|   |   |   |   |   |   |   |   |   min_eigenvector_centrality <= 0.013
|   |   |   |   |   |   |   |   |   |   min_prota_rel_def_1 <= 0.535: C- (2.0)
|   |   |   |   |   |   |   |   |   |   min_prota_rel_def_1 > 0.535: C+ (3.4/0.4)
|   |   |   |   |   |   |   |   |   min_eigenvector_centrality > 0.013: C- (5.4)
|   |   |   |   |   |   |   year > 2000
|   |   |   |   |   |   |   |   min_eigen_cent_norma <= 0.023: C- (12.0)
|   |   |   |   |   |   |   |   min_eigen_cent_norma > 0.023
|   |   |   |   |   |   |   |   |   average_betweenness_centrality <= 0.033: C+ (2.0)
|   |   |   |   |   |   |   |   |   average_betweenness_centrality > 0.033: C- (3.8/0.8)
```

```
|   |   |   |   diameter > 3
|   |   |   |   |   max_degree_centrality <= 0.619: C- (12.0)
|   |   |   |   |   max_degree_centrality > 0.619
|   |   |   |   |   |   average_communicability_centrality <= 595.986
|   |   |   |   |   |   |   number of cliques <= 29
|   |   |   |   |   |   |   |   year <= 2004
|   |   |   |   |   |   |   |   |   num_protas_def_4 <= 1: C- (3.0)
|   |   |   |   |   |   |   |   |   num_protas_def_4 > 1
|   |   |   |   |   |   |   |   |   |   avr_prota_pop_def_4 <= 0.511
|   |   |   |   |   |   |   |   |   |   |   average_communicability_centrality <= 124.204: C+ (4.0)
|   |   |   |   |   |   |   |   |   |   |   average_communicability_centrality > 124.204
|   |   |   |   |   |   |   |   |   |   |   |   average_communicability_centrality <= 385.014
|   |   |   |   |   |   |   |   |   |   |   |   |   min_prota_rel_def_3 <= 0.815: C- (9.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   min_prota_rel_def_3 > 0.815: C+ (3.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   |   average_communicability_centrality > 385.014: C+
(2.0)
|   |   |   |   |   |   |   |   |   |   |   avr_prota_pop_def_4 > 0.511: C+ (8.0)
|   |   |   |   |   |   |   |   |   year > 2004: C- (5.0)
|   |   |   |   |   |   |   number of cliques > 29: C+ (5.0)
|   |   |   |   |   |   average_communicability_centrality > 595.986
|   |   |   |   |   |   |   num_protas_def_2 <= 9: C- (22.0/2.0)
|   |   |   |   |   |   |   num_protas_def_2 > 9
|   |   |   |   |   |   |   |   min_prota_rel_def_3 <= 0.56: C+ (2.0)
|   |   |   |   |   |   |   |   min_prota_rel_def_3 > 0.56: C- (2.0)
|   |   radius > 2
|   |   |   min_eigenvector_centrality <= 0.001
|   |   |   |   min_communicability_betweenness_centrality <= 0.003
|   |   |   |   |   avr_prota_pop_def_4 <= 0.2: C+ (4.0/1.0)
|   |   |   |   |   avr_prota_pop_def_4 > 0.2
|   |   |   |   |   |   min_degree_centrality <= 0.027: C- (23.0/1.0)
|   |   |   |   |   |   min_degree_centrality > 0.027
|   |   |   |   |   |   |   average_betweenness_centrality <= 0.036: C- (2.0)
|   |   |   |   |   |   |   average_betweenness_centrality > 0.036: C+ (6.0/1.0)
|   |   |   |   min_communicability_betweenness_centrality > 0.003: C+ (2.0)
|   |   |   min_eigenvector_centrality > 0.001
|   |   |   |   min_eigenvector_centrality <= 0.003: C+ (6.0/1.0)
|   |   |   |   min_eigenvector_centrality > 0.003: C- (2.0)
main genre = Crime
|   year <= 1985: C+ (11.0)
|   year > 1985
|   |   number of cliques <= 24
```

```
|   |   |   max_degree <= 15
|   |   |   |   number_of_nodes <= 13: C- (2.0)
|   |   |   |   number_of_nodes > 13: C+ (10.0)
|   |   |   max_degree > 15
|   |   |   |   min_communicability_betweenness_centrality <= 0.03
|   |   |   |   |   diameter <= 3
|   |   |   |   |   |   num_protas_def_2 <= 3: C- (9.0)
|   |   |   |   |   |   num_protas_def_2 > 3
|   |   |   |   |   |   |   num_protas_def_4 <= 3: C+ (2.0)
|   |   |   |   |   |   |   num_protas_def_4 > 3: C- (4.0)
|   |   |   |   |   diameter > 3
|   |   |   |   |   |   max_degree <= 23: C- (5.0)
|   |   |   |   |   |   max_degree > 23: C+ (6.0/1.0)
|   |   |   |   min_communicability_betweenness_centrality > 0.03: C+ (3.0)
|   |   number of cliques > 24
|   |   |   average_closeness_centrality <= 0.495: C+ (18.0)
|   |   |   average_closeness_centrality > 0.495
|   |   |   |   number of cliques <= 29: C+ (4.0)
|   |   |   |   number of cliques > 29: C- (3.0)
main genre = Mystery: C+ (4.0/1.0)
main genre = Biography
|   year <= 2008
|   |   max_communicability_betweenness_centrality <= 0.918
|   |   |   max_communicability_betweenness_centrality <= 0.874: C+ (13.0/1.0)
|   |   |   max_communicability_betweenness_centrality > 0.874: C- (5.0)
|   |   max_communicability_betweenness_centrality > 0.918: C+ (18.0)
|   year > 2008: C- (7.0/1.0)
main genre = Sci-Fi
|   density <= 0.202: C+ (2.0)
|   density > 0.202: C- (2.0)
main genre = Animation
|   min_eigen_cent_norma <= 0.004
|   |   number of cliques <= 17: C+ (2.0)
|   |   number of cliques > 17: C- (4.0)
|   min_eigen_cent_norma > 0.004: C+ (7.0)
main genre = Fantasy: C- (5.0)
main genre = Horror
|   year <= 1986: C+ (6.0/1.0)
|   year > 1986: C- (33.0/1.0)
main genre = Documentary: C- (2.0)
main genre = Short: C+ (2.0/1.0)
```

```
main genre = Music: C+ (0.0)
main genre = Thriller: C- (1.0)
main genre = Western: C+ (1.0)
main genre = Romance: C- (1.0)
```