Maestría en Explotación de Datos
y Descubrimiento del Conocimiento
Universidad de Buenos Aires

Maestría en Explotación de Datos y Descubrimiento del Conocimiento

Aprendizaje Automático

Trabajo Práctico Nro. 1

Predicción de Nominados al Oscar

Lucas Tomasini

Comisión 1

29 de Abril de 2019

Resumen

El objetivo de este artículo es predecir qué películas recibirán nominaciones al Oscar en las categorías más importantes, en base a características de las mismas: género, rating, críticas y otros premios prestigiosos que recibieron de la industria del cine. Se utilizaron datos de ejemplos de 1235 películas lanzadas entre el año 2000 y 2018 y dos tipos de algoritmos de aprendizaje automático para clasificación: Árboles de Decisión y Naive Bayes. Para árboles de decisión también se realizó un análisis del comportamiento frente a la falta de datos y a la presencia de ruido. Finalmente, se compararon ambos métodos evaluando su performance sobre nuevas películas utilizadas como datos de prueba.

Los resultados que arrojan ambos modelos son similares: un área bajo la curva ROC promedio de 0.8173 para árboles de decisión y 0.8231 para naive Bayes, utilizando clasificación en tres niveles: ninguna nominación, una nominación y más de una nominación.

El árbol de decisión resulta ser robusto a datos faltantes y ruido en los datos de entrada. Su performance se ve poco afectada salvo para valores altos de cantidad de datos faltantes imputados y ruido agregado.

Introducción

A lo largo de este trabajo se arman diferentes modelos utilizando técnicas de aprendizaje automático para predecir qué películas recibirán nominaciones a las categorías más importantes en los premios Oscar. Para construir dichos modelos nos basamos en las características de las películas, su éxito con los espectadores (recaudación, rating en IMDB, etc.) y otras nominaciones que recibieron a premios prestigiosos de la industria.

Se han realizado varios intentos de predecir quiénes serían los ganadores para las diferentes categorías (ver [1] y [2], por ejemplo), aunque usualmente estos trabajos implican el uso de redes neuronales y se enfocan en los distintos ganadores. Nosotros preferimos focalizarnos en las nominaciones definiendo tres tipos de clases: no reciben nominaciones, reciben sólo una, o más de una.

Con este objetivo, se utilizaron dos tipos de modelos: *Árboles de Decisió*n y *Naive Bayes*. En el primer caso, se testearon diferentes longitudes de árboles con diferentes medidas de impureza (Gini e Information Gain), así como la tolerancia de éstos a la cantidad de datos faltantes y al ruido en los datos. Para naive Bayes se analizaron diferentes distribuciones de probabilidad para ver cuál se adaptaba mejor a nuestro conjunto de datos. Por último, se realizó una comparación de la capacidad predictiva de ambos métodos basándose en el área debajo de la curva ROC y en el accuracy.

El trabajo se encuentra dividido en varias partes, donde se explican los diferentes pasos que tuvimos en cuenta para la construcción de los modelos. En la sección *Datos* se describen los diferentes atributos del dataset, así como las diferentes transformaciones que realizamos para poder utilizarlos. En la sección *Metodología* se detallan los diferentes algoritmos utilizados con sus respectivos hiperparámetros, así como las transformaciones realizadas a los datos para probar su tolerancia a valores faltantes y ruido. Por último, se muestran los resultados obtenidos en cada caso en la sección *Resultados.*

Datos

El conjunto de datos original con el que trabajamos fue obtenido a partir de un trabajo realizado por Alison Daley en el que usa Deep Learning para predecir los ganadores de los premios Oscar 2019 en diferentes categorías [1]. Los datos consisten en 1235 películas lanzadas entre el año 2000 y 2018 con 119 atributos, algunos de los cuales son características propias de las películas (título, duración, género, el rating que la gente le da en el sitio especializado IMDB [3], el rating de Metascore, el dinero recaudado, etc.) y otros son atributos relacionados al

desempeño de las películas (nominaciones y triunfos) en los Oscar y otros premios importantes como los Golden Globes, BAFTA, Screen Actors Guild y Critics Choice Awards, entre otros.

Debido a la magnitud del conjunto de datos original, fue necesario transformarlo para que se adapte mejor a nuestras necesidades. El primer punto a considerar fueron las categorías de premios a predecir. Los datos originales tienen información de muchas categorías, por lo cual decidimos mantener solamente aquellas que son usualmente consideradas como las más importantes: mejor película, mejor director, mejor actor principal, mejor actriz principal, mejor actor de reparto, mejor actriz de reparto, mejor guion original y mejor guion adaptado. A continuación, tuvimos que realizar la misma limpieza de categorías para los otros siete premios considerados a parte de los Oscar.

El segundo punto a considerar en la limpieza de los datos fue el género de las películas. El conjunto de datos original trae una columna en la cual se listan diferentes tipos de géneros que tiene cada película en un campo de tipo *string* separados por un caracter '|'. Para hacer su uso más adecuado a los algoritmos que utilizamos y dado que una película puede pertenecer a más de un género (por ejemplo, una película puede ser un drama y a la vez romántica), convertimos este campo en 23 campos booleanos para cada uno de los géneros.

El siguiente paso consistió en recalcular la cantidad de nominaciones para cada uno de los premios que el dataset original proveía. Éste tiene una columna con la cantidad de nominaciones que tuvo una película para cada uno de los diferentes premios (los campos *Oscar_nominated*, *Golden_Globes_nominated*, *BAFTA_nominated*, etc). Sin embargo, los valores de estos campos están originalmente mal computados (cuentan mal la cantidad de categorías para las que fue nominada la película) y además incluyen otras categorías que no son las de interés para este trabajo (mejor diseño de vestuario, mejor maquillaje y peinado, mejor documental corto, etc). Por este motivo, generamos nosotros mismos esos campos, contando la cantidad de nominaciones en cada premio para las categorías principales que definimos más arriba.

El último paso en la limpieza del dataset consistió en eliminar aquellos atributos que no son de interés para el análisis o que poseen una baja correlación con la variable de clase que se quiere predecir. Descartamos todos los relativos a premios que no tienen relación con las categorías principales (por ejemplo American Cinema Editors, Costume Designer Guild, Art Director Guild, etc., que premian cuestiones técnicas de otras disciplinas) y premios menores que son otorgados en círculos más chicos del cine (por ejemplo Austin Film Critics Association, Denver Film Critics Society, etc.) y que consideramos que no tienen un efecto en las nominaciones al Oscar que una película puede recibir. También eliminamos las columnas de géneros que no aplican para los premios a las principales categorías ya que poseen las suyas propias (por ejemplo: cortos, documentales, etc). Por último, viendo la correlación de cada una de las variables independientes con la dependiente (cantidad de nominaciones a los Oscar), eliminamos aquellas cuyo valor absoluto de la correlación es inferior a 0.1.

El conjunto de datos final con el que trabajamos consiste en 1235 películas con 19 atributos, de los cuales 12 son numéricos (*duration*, *rate*, *metascore*, *votes*, *Oscar_nominations*, *GG_nominations*, *BAFTA_nominations*, *ActorsGuild_nominations*, *CriticsChoice_nominations*, *DirectorsGuild_nominations*, *WritersGuild_nominations*, *critic_reviews*) y siete son categóricos (*Action*, *Adventure*, *Animation*, *Drama*, *Family*, *Horror*, *Sci-Fi*).

Metodología

Como comentamos anteriormente, en el presente trabajo utilizamos dos tipos de modelos: Árboles de Decisión y Naive Bayes. En cada caso trabajamos con una partición del dataset usando 5-fold cross-validation sobre el conjunto de desarrollo para evaluar diferentes hiperparámetros mediante el área bajo la curva ROC (AUC ROC) y el accuracy como métrica de performance. Como se trata de una clasificación multiclase, el AUC ROC utilizado es el promedio de los AUC ROC de cada clase, considerando cada clase contra el resto ("binarizando" cada una). En el caso particular de los árboles de decisión también realizamos un análisis del comportamiento de los modelos frente a la falta de datos y la presencia de ruido. Para finalizar, realizamos una búsqueda de los hiperparámetros óptimos de cada algoritmo mediante *grid search* en base

al AUC ROC promedio y comparamos ambos métodos evaluando su performance sobre el conjunto de datos de prueba.

1. *Partición de los datos*

El primer paso para poder entrenar y probar los modelos con dos conjuntos diferentes fue partir los datos en un conjunto de desarrollo y un conjunto de prueba. A su vez, el conjunto de desarrollo se partirá en conjunto de entrenamiento y validación una proporción 80/20 con el objetivo de ajustar los hiperparámetros aplicando 5-fold cross-validation.

Para el conjunto de prueba se tomaron las películas correspondientes a los años 2016, 2017 y 2018 (10% del dataset). A su vez, el conjunto target ('Oscar_nominations') fue convertido en tres clases: cero nominaciones (clase 0), una nominación (clase 1) y más de una nominación (clase 2).

2. Árboles de decisión

El primer método que probamos fue el de árboles de decisión, probando con diferentes alturas y con dos medidas de impureza: Information Gain y Gini Gain. Para decidir la mejor combinación de dichos hiperparámetros entrenamos el modelo utilizando el conjunto de entrenamiento, y evaluamos su performance tanto sobre el conjunto de validación como sobre el de entrenamiento mediante el accuracy y la AUC ROC promedio. En todos los casos se utilizó 5-fold cross-validation.

Las diferentes combinaciones de altura y medida de impureza fueron las siguientes:

- Altura 3 y la medida de impureza por default (Gini)
- Altura 3 e Information Gain
- Altura 6 y Gini Gain
- Altura 6 e Information Gain
- Altura sin máximo preestablecido y Gini Gain
- Altura sin máximo preestablecido e Information Gain

3. Datos Faltantes

Para analizar la robustez de estos árboles de decisión frente a la presencia de datos faltantes eliminamos datos de manera aleatoria (entre 0% y 80% en incrementos de 5%) e imputamos siguiendo diferentes estrategias. Los atributos que seleccionamos para aplicar este proceso fueron los que están más correlacionados con la función objetivo o clase (que en gran parte también coinciden con los atributos más cercanos a la raíz del árbol). Éstos son: *GG_nominations*, *metascore*, *CriticsChoice_nominations*, *critic_reviews*, *duration*, *BAFTA_nominations*, *ActorsGuild_nominations*, *WritersGuild_nominations*, *rate* y *votes*. Como estrategia de imputación utilizamos moda, media y mediana de toda la columna, como así también moda, media y mediana según la clase.

A continuación partimos el conjunto de desarrollo en conjunto de entrenamiento y validación (80%/20%) y entrenamos un árbol de decisión de altura 6 con una medida de impureza Gini Gain para cada porcentaje de datos faltantes y estrategia de imputación. Dicha combinación de altura y medida de impureza es la que mejor performance logró de las pruebas anteriores. Luego analizamos el Accuracy, AUC ROC y la cantidad de nodos hoja para cada árbol en función del porcentaje de datos faltantes.

4. Tolerancia al Ruido

El siguiente punto que queremos analizar es el comportamiento del modelo frente a la presencia de ruido. Con este fin agregamos ruido gaussiano de media cero y varianza igual a la varianza de la columna multiplicada por el porcentaje de ruido deseado. Como el dataset contiene sólo valores positivos, se truncó a cero cualquier valor de celda que haya quedado negativo luego del agregado del ruido.

De nuevo, al igual que en el tratamiento de faltantes, las columnas consideradas a las cuales se les agregó ruido fueron *GG_nominations*, *metascore*, *CriticsChoice_nominations*, *critic_reviews*, *duration*, *BAFTA_nominations*, *ActorsGuild_nominations*, *rate*, *votes* y *WritersGuild_nominations*. También, como en el caso anterior, se entrenó un árbol de decisión para porcentajes de ruido entre 0% y 80% en incrementos de 5% y se analizó el accuracy, la AUC ROC y el número de nodos hoja en función de los diferentes porcentajes.

5. Naive Bayes

Luego de una primera prueba con árboles de decisión, utilizamos una segunda con el algoritmo de Naive Bayes para predecir las nominaciones. Evaluamos diferentes distribuciones de probabilidad (gaussiana, multinomial y de Bernoulli) y encontramos que la de Bernoulli fue la que mejores resultados obtuvo con nuestro conjunto de datos.

A continuación entrenamos el modelo usando el 98% del conjunto de desarrollo e informamos las probabilidades a priori (de clase). El 2% lo dejamos para predecir e informar las probabilidades condicionales de observaciones nuevas, es decir, probabilidad de que una observación pertenezca a una determinada clase dado que se conocen sus atributos o *features*. También calculamos el accuracy y la AUC ROC para evaluar la performance del modelo.

6. Comparación entre árboles de decisión y naive Bayes

Para la comparación entre los modelos construidos con árboles de decisión y naive Bayes determinamos primero los mejores hiperparámetros a utilizar en cada caso aplicando *grid search* con 5-fold cross-validation sobre el conjunto de desarrollo. Para el árbol de decisión, los hiperparámetros considerados fueron la altura del árbol, la medida de impureza, el número mínimo de muestras para separar un nodo, el número mínimo de muestras por hoja y el máximo número de hojas. Para naive Bayes solamente consideramos el parámetro *alpha*, que regula la cantidad de suavizado (regularización) que se le aplica al algoritmo.

Como cierre del trabajo, comparamos la performance de cada modelo entrenando ambos algoritmos con los datos de desarrollo completos y calculando el accuracy y la ROC AUC sobre los datos de prueba que separamos al principio (nominaciones al Oscar de 2016 a 2018).

Resultados

1. Partición de los datos

Se trata de un dataset cuyas clases están desbalanceadas. Del conjunto total, hay 868 películas que no fueron nominadas a ningún Oscar, 146 que fueron nominadas a uno solo y 185 que fueron nominadas a más de uno. Por este motivo, utilizamos 5-fold estratificado, que selecciona las particiones al azar pero teniendo en cuenta la proporción de datos pertenecientes a las diferentes clases (0, 1 o 2).

2. Árboles de decisión

Se detalla a continuación tablas con los valores de accuracy y ROC AUC para el conjunto de entrenamiento y validación evaluados sobre cada uno de los *folds.* Se muestra también el promedio total junto con su desvío estándar. Cada tabla corresponde a una combinación dada de altura y medida de impureza.

	Medición	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Media	Desvío
0	Training Accuracy	0.8686	0.8724	0.8782	0.8898	0.8770	0.8772	0.0080
1	CV Accuracy	0.8664	0.8698	0.8419	0.8465	0.8605	0.8570	0.0123
2	Training ROC AUC	0.8355	0.7890	0.8077	0.8138	0.8002	0.8092	0.0174
3	CV ROC AUC	0.8333	0.7895	0.7743	0.7355	0.7967	0.7859	0.0355

Tabla 1.1 - Altura 3 y la medida de impureza por default (Gini)

	Medición	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Media	Desvío
0	Training Accuracy	0.8709	0.8643	0.8701	0.8701	0.8666	0.8684	0.0028
1	CV Accuracy	0.8664	0.8605	0.8698	0.8512	0.8605	0.8616	0.0071
2	Training ROC AUC	0.7989	0.7604	0.7843	0.7783	0.7722	0.7788	0.0143
3	CV ROC AUC	0.7837	0.7608	0.7926	0.7514	0.7673	0.7712	0.0168

Tabla 1.2 - Altura 3 e Information Gain

	Medición	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Media	Desvío
0	Training Accuracy	0.9512	0.9524	0.9443	0.9513	0.9490	0.9496	0.0032
1	CV Accuracy	0.8894	0.8512	0.8465	0.8651	0.8791	0.8663	0.0182
2	Training ROC AUC	0.9478	0.9392	0.9181	0.9147	0.9249	0.9289	0.0141
3	CV ROC AUC	0.8634	0.8185	0.7946	0.7917	0.8502	0.8237	0.0323

Tabla 1.3 - Altura 6 y Gini Gain

	Medición	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Media	Desvío
0	Training Accuracy	0.9442	0.9478	0.9455	0.9420	0.9443	0.9448	0.0021
1	CV Accuracy	0.8802	0.8558	0.8465	0.8791	0.8419	0.8607	0.0180
2	Training ROC AUC	0.9033	0.9149	0.9275	0.9095	0.9287	0.9168	0.0111
3	CV ROC AUC	0.8067	0.8043	0.8033	0.8094	0.7925	0.8032	0.0065

Tabla 1.4 - Altura 6 e Information Gain

	Medición	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Media	Desvío
0	Training Accuracy	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0000
1	CV Accuracy	0.8571	0.8512	0.8326	0.8651	0.8372	0.8486	0.0136
2	Training ROC AUC	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0000
3	CV ROC AUC	0.8076	0.8201	0.7867	0.8164	0.8118	0.8085	0.0131

Tabla 1.5 - Altura sin máximo preestablecido y Gini Gain

	Medición	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Media	Desvío
0	Training Accuracy	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0000
1	CV Accuracy	0.8756	0.8465	0.8419	0.8837	0.8279	0.8551	0.0236
2	Training ROC AUC	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0000
3	CV ROC AUC	0.8374	0.7877	0.8069	0.8234	0.7783	0.8067	0.0245

Tabla 1.6 - Altura sin máximo preestablecido e Information Gain

Nótese que para los casos de árboles con altura 3, los valores de accuracy y ROC AUC calculados sobre el conjunto de entrenamiento son similares, aunque levemente mayores, a los calculados sobre el conjunto de validación. Por otro lado, con árboles de altura 6 la performance aumenta para ambos conjuntos. El árbol se ajusta mejor a los datos de entrenamiento y, al mismo tiempo, generaliza mejor al clasificar instancias no observadas. En cambio, en los dos últimos

casos donde no hay un límite prestablecido en la altura máxima, vemos que el árbol se ajusta perfectamente al set de entrenamiento pero, sin embargo, tanto el accuracy como el ROC AUC del set de validación disminuyen respecto del caso anterior. En este caso el algoritmo está sobreajustando los datos de entrenamiento, disminuyendo su capacidad de clasificar datos nuevos. Por lo tanto, la mejor combinación de todas es el árbol de altura 6 con criterio Gini, que es levemente superior al de Information Gain.

A modo ilustrativo graficamos el árbol para el primer caso (altura 3 con criterio Gini), ya que para árboles más grandes se dificulta la lectura de los atributos que participan:

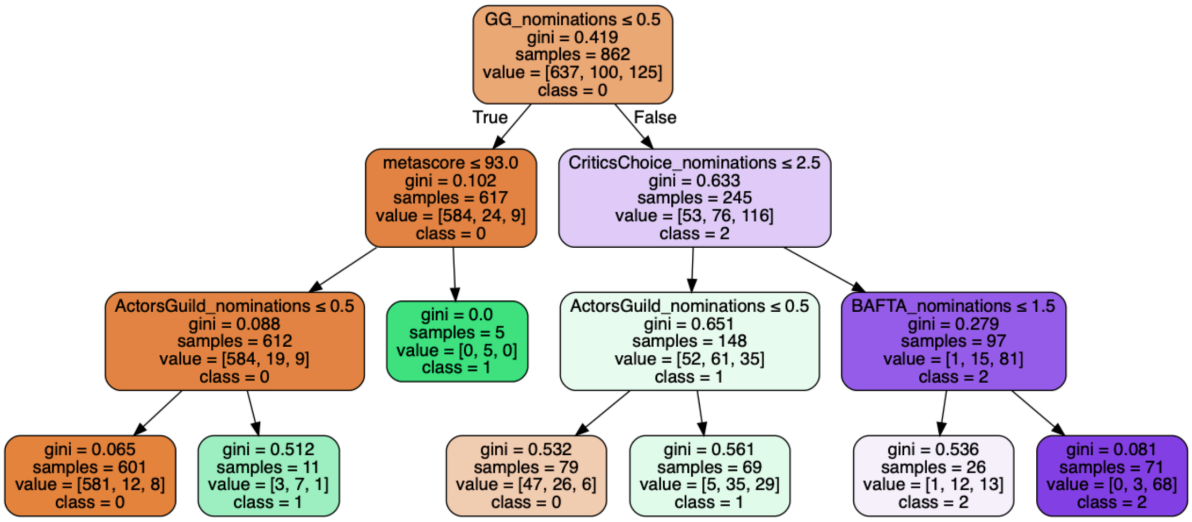


Figura 1 – Árbol de decisión altura 3 con Gini

Vemos que *GG_nominations*, *ActorsGuild_nominations*, *metascore*, *BAFTA_nominations* y *CriticsChoice_nominations* participan del árbol. Justamente estos atributos poseen una correlación alta con la clase y están entre los que seleccionamos para rellenar con datos faltantes y agregar ruido.

3. *Datos Faltantes*

En los siguientes gráficos se visualiza el accuracy, ROC AUC y cantidad de nodos hoja en función del porcentaje de valores faltantes que fueron imputados. Los gráficos de la izquierda corresponden a imputaciones que no tienen en cuenta la clase, mientras que los de la derecha a imputaciones según la clase. Ambos graficados sobre la misma escala en el eje y para facilitar la comparación. Los diferentes colores representan la estrategia de imputación: por moda, por media o por mediana.

Figura 2.1 – Accuracy en función del % de valores imputados

Figura 2.2 – ROC AUC promedio en función del % de valores imputados

Figura 2.3 – Cantidad de hojas en función del % de valores imputados

De las figuras 2.1 y 2.2 vemos que los métodos de imputación según la clase son más efectivos que las imputaciones simples, ya que la performance decrece con menor pendiente a medida que aumenta el porcentaje de valores imputados. El accuracy en la figura 2.1 disminuye en promedio alrededor de un 10% en el gráfico de la izquierda y de un 4% en el de la derecha. Una variación similar ocurre también con el ROC AUC en la figura 2.2. Nótese que las diferentes estrategias de imputación (moda, media o mediana) siguen tendencias similares, aunque la imputación por moda en el caso simple afecta más la performance para valores altos de porcentaje de imputación.

De la figura 2.3 vemos que la cantidad de hojas del árbol disminuye a medida que aumenta el porcentaje y esta disminución es más pronunciada en el caso de imputación según la clase. Esto tiene su explicación por el hecho de que al imputar por clase estamos disminuyendo el nivel de impureza o desorden que existe en ese atributo para cada valor de la clase, con lo cual aumenta el poder de clasificación que tiene ese atributo y, por lo tanto, menos cantidad de hojas son requeridas. Por ejemplo, en el caso extremo de 80% de valores imputados, dicho atributo tendrá casi todos los valores iguales dentro de una misma clase, lo cual hace que su entropía o impureza Gini dentro de esa clase sea cercana a 0, maximizando la ganancia que se obtiene al clasificar con dicho atributo. Esto no significa que la clasificación mejore, de hecho la performance ya vimos que disminuye

4. Tolerancia al Ruido

En los siguientes gráficos se visualiza accuracy, ROC AUC y cantidad de nodos hoja en función del porcentaje de ruido agregado.

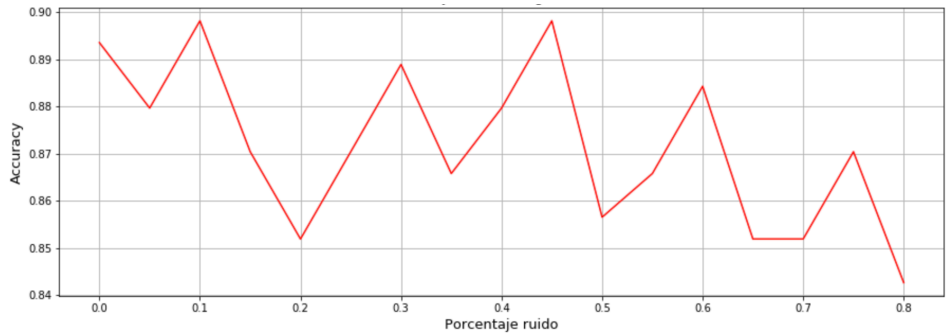


Figura 3.1 – Accuracy en función del % de ruido

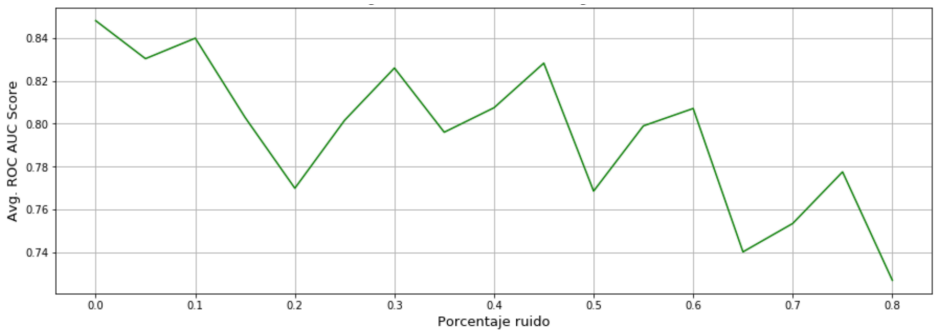


Figura 3.2 – AUC ROC promedio en función del % de ruido

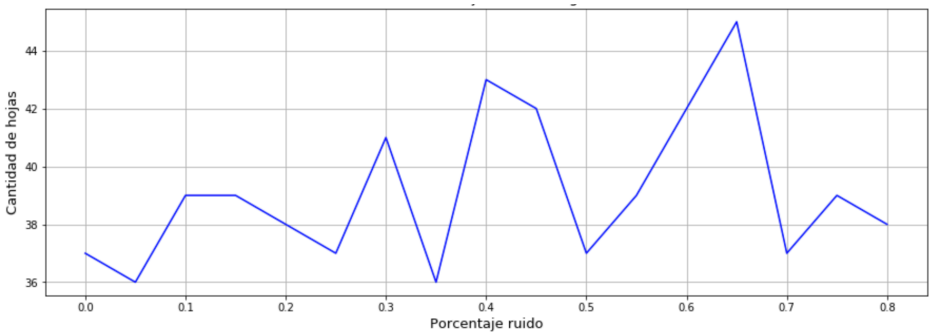


Figura 3.3 – Cantidad de hojas en función del % de ruido

Comparando las figuras 3.1 y 3.2 vemos que el accuracy se ve menos afectado por el ruido que el ROC AUC, aunque hasta un 40% de ruido la degradación es más bien leve. Recién entre el 60% y 80% se nota una caída más pronunciada de la performance. La cantidad de hojas es oscilante, aunque no parecería haber una tendencia significativa de aumento o disminución.

5. Naive Bayes

Medimos la performance para diferentes clasificadores naive Bayes, siendo el Bernoulli el que mejor clasifica. Mostramos la tabla con los resultados solamente para este caso:

	Medición	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Media	Desvío
0	Training Accuracy	0.8744	0.8794	0.8794	0.8794	0.8852	0.8795	0.0038
1	CV Accuracy	0.8986	0.8791	0.8744	0.8698	0.8558	0.8755	0.0156
2	Training ROC AUC	0.8288	0.8298	0.8365	0.8376	0.8421	0.8349	0.0056
3	CV ROC AUC	0.8586	0.8591	0.8263	0.7988	0.8041	0.8294	0.0288

Tabla 2 – Naive Bayes Bernoulli

Nótese que la performance sobre el conjunto de entrenamiento y sobre el conjunto de validación son similares, a diferencia del caso de árboles de decisión, donde el algoritmo predice mejor sobre el conjunto de entrenamiento. A su vez, naive Bayes resulta ser levemente superior al clasificar datos nuevos que el árbol de la Tabla 1.3.

Las probabilidades a priori para cada clase son: 0.7390 para la clase 0, 0.1160 para clase 1 y 0.1450 para clase 2. A modo ilustrativo mostramos las probabilidades a posteriori de 10 ejemplos de instancias no observadas:

	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10
Clase 0	0.0012	0.9989	1.0	0.9989	0.0000	0.9906	0.1401	0.9997	0.7498	0.9998
Clase 1	0.1251	0.0011	0.0	0.0011	0.4026	0.0093	0.7678	0.0003	0.2419	0.0002
Clase 2	0.8737	0.0000	0.0	0.0000	0.5973	0.0001	0.0920	0.0000	0.0083	0.0000

Tabla 3 – Probabilidades a posteriori de cada clase

6. Comparación entre árboles de decisión y naive Bayes

Los hiperparámetros que maximizan el ROC AUC, llegando a un valor de 0.8361 en el árbol de decisión, son:

- Criterio: Gini
- Máxima altura del árbol: 5
- Mínimo número de muestras para separar un nodo: 5
- Mínimo número de muestras por hoja: 5
- Máximo número de hojas: 40

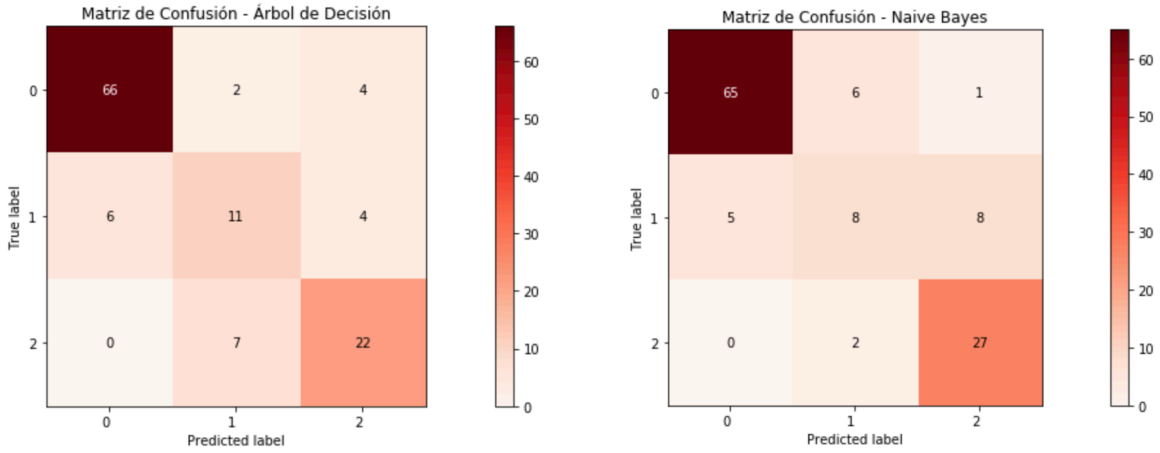
Para el caso de naive Bayes, el alpha óptimo es de 0.4, obteniéndose un ROC AUC de 0.8323.

La Tabla 4 muestra los resultados de la comparación final entre los dos modelos sobre el conjunto de testing. Ambos llegan a una performance similar, siendo naive Bayes ligeramente superior.

	Árbol de Decisión	Naive Bayes
Accuracy	0.8115	0.8197
ROC AUC promedio	0.8173	0.8231

Tabla 4 – Comparación entre ambos métodos

Como detalle final, es interesante visualizar también la matriz de confusión de ambos resultados. Como se puede apreciar en la Figura 4, tanto el árbol de decisión como naive Bayes tienen mayor dificultad en predecir películas que fueron nominadas exactamente a un Oscar. En cambio, logran una mejor clasificación para las clases 0 y 2, es decir, películas con ninguna nominación o películas con más de una.



Conclusiones

Teniendo en cuenta que el objetivo es encontrar un algoritmo capaz de clasificar en 3 clases distintas (lo cual es más desafiante que un clasificador binario), podemos concluir que ambos modelos dan una predicción bastante aceptable de la cantidad de nominaciones al Oscar (cero, una o más de una) que tendrá una película dentro de las categorías principales.

Como era de esperarse, limitar la altura del árbol de decisión u otros parámetros como el máximo número de hojas resulta efectivo a la hora de querer prevenir *overfitting* sobre los datos de entrenamiento. En el caso de naive Bayes esto es más fácil de controlar, ya que mientras los datos de entrenamiento sigan la misma distribución que los de validación y prueba, es muy difícil que ocurra un sobreajuste, hecho que queda en evidencia en la parte de *Resultados*, donde la performance sobre el conjunto de entrenamiento y validación son similares.

En concordancia con la teoría, el árbol de decisión resultó ser robusto a datos faltantes o ruido en los diferentes *features*, ya que su performance se vio poco afectada, salvo para valores considerablemente altos de cantidad de datos faltantes imputados y ruido agregado.

Ambos métodos tienen mayor dificultad en predecir películas de clase 1, es decir, nominadas exactamente a un Oscar. En cambio, logran clasificar mejor aquellas de clase 0 o 2, es decir, que no van a ser nominadas a ningún Oscar o van a tener más de una nominación.

Referencias

1. Daley, A., *Predicting the 2019 Oscars Winners with Machine Learning*, https://blog.bigml.com/2019/02/22/predicting-the-2019-oscars-winners-with-machine-learning/
2. King, A., *Approximating the Minds of 2019 Oscars Voters using Neural Networks*, https://towardsdatascience.com/approximating-the-minds-of-2019-oscars-voters-using-neural-networks-b922f3d6864c
3. Internet Movie Database, www.imbd.com