Universidad Francisco Marroquín

Machine Learning Models

José Alejandro Guzmán Zamora

Youtube Trending Videos Analysis

Introducción

Youtube.com es un sitio web de contenido multimedia altamente popular en el que los usuarios comparten videos de temas variados. El fácil acceso a la plataforma y el flujo sencillo de acciones para compartir un video motiva a que cualquier tipo de persona lo haga en cualquier momento. Debido a esto, la cantidad de contenido que se está subiendo a la plataforma cada cierta cantidad de tiempo es masivo. Parte fundamental de la sostenibilidad del sitio es el consumo del contenido que se comparte en el mismo. Este constante flujo de información ha creado una red mundial de mercadeo y entretenimiento que conectan al mundo día a día. En esta ocasión se le concederá especial atención al segmento de tendencias dentro de la plataforma en función. Esta pestaña de tendencias mantiene un listado de los videos más populares de Youtube en un momento preciso. El algoritmo o método de decisiones que actualiza esta pestaña es desconocido, sin embargo, se podría intuir que hay una variedad de factores involucrados, entre estos la cantidad de vistas, la proporción de "me gusta" y "no me gusta", las veces que se ha compartido cada video, el número de comentarios y el porcentaje de tiempo visto en promedio. El presente reporte tiene como objetivo utilizar información histórica de videos que han formado parte de la tendencia en distintos países del mundo para hacer el intento de predecir si la tendencia de cada uno se consider a nacional (a nivel estadounidense) o internacional.

Para cumplir con el objetivo propuesto se utilizará el dataset "Trending Youtube Video Statistics", mantenido por Mitchell, J. dentro de la plataforma Kaggle. Este dataset contiene una cantidad considerable de observaciones a lo largo de meses en relación con videos que estuvieron en tendencia en distintos países. Se comienza con una breve descripción de los datos iniciales y las transformaciones que se realizaron para poder generar una columna de target. Esta columna de target es la que se usa posteriormente para entrenar un conjunto de modelos con la intención de encontrar al mejor predictor. Antes de entrar en el contexto de los modelos y el entrenamiento de cada uno, se explica el proceso de limpieza y exploración inicial de los datos. Una vez preparados los datos relevantes, se describe el proceso de construcción, entrenamiento, refinamiento y evaluación de los siguientes tipos de modelos: Árboles de decisión, Random Forests, Bagging y Boosting desde 2 perspectivas distintas. Todos estos métodos tienen en común una estructura inicial basada en árboles para tomar las decisiones que conllevan a la predicción.

Datos

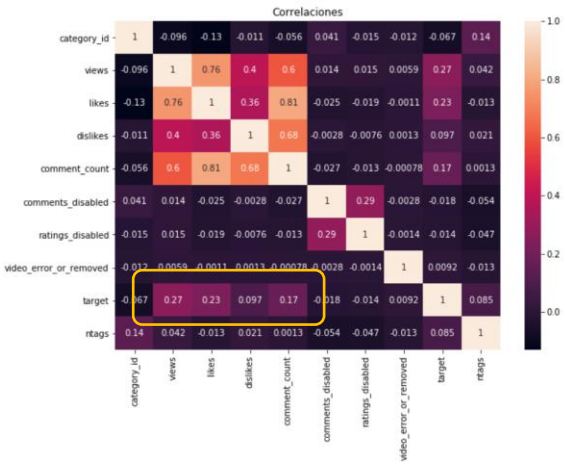
Anteriormente se mencionó que el dataset contiene información general de los videos que logran estar en la pestaña de tendencias. Esta información incluye la fecha en la que el video obtuvo una posición en la pestaña, el título del video, el título del canal que publicó el video, un identificador de categorías, la hora de publicación, los tags utilizados, la cantidad de visualizaciones al momento de consumir el API, la cantidad de likes, cantidad de dislikes, un conteo de comentarios, unas banderas y la descripción del video. La mayoría de estos atributos pueden llegar a ser intuitivos para cualquier persona que haya utilizado la plataforma, sin embargo, por motivos de claridad se define un *like* y un *dislike* como la opinión binaria que un usuario le puede dar al video que visualizó. Asimismo, es importante mencionar que las banderas incluyen información de videos que se eliminaron, videos que tienen los comentarios desactivados o los ratings (like,dislike) desactivados.

Con este entendimiento ligeramente más claro de los datos, se pueden empezar a seleccionar las variables que verdaderamente serán relevantes para nuestro análisis. Esta selección se realizó de manera inversa, es decir, se comenzó por eliminar las variables inútiles. Un identificador único no va a aportar nada a un modelo, de igual manera todas las columnas de texto como el título del video y la descripción del mismo no se pueden agregar al tipo de modelos que se van a usar. No obstante, hay que saber que sí se podría hacer un análisis sobre estos conjuntos de oraciones, pero esta fuera del alcance de este proyecto. Además, las banderas demostraron un desequilibrio que no vale la pena intentar arreglar, esto es porque hay unas que tienen 3 registros de un tipo y el resto son del otro tipo; es decir, prácticamente son columnas de un solo valor. Por último, para aprovechar la información de los tags, se hizo un Split() por cada registro y se creó una columna calculada de la cantidad de tags que cada video utilizó. Esta acción se realizó con la intención de aprovechar las features disponibles lo más posible. Esto nos deja con las siguientes variables relevantes hasta el momento: categoría, visualizaciones, likes, dislikes, número de comentarios y número de tags.

Hasta el momento no se ha comentado mucho el tema del target, ¿Cómo se va a realizar la predicción con la información que tenemos disponible hasta el momento? En este punto es cuando entra la importancia de los múltiples archivos del dataset. En la introducción se define un propósito dirigido a hacer predicciones de si un video estará en tendencia nacional o internacional; para obtener esa response variable se comenzó con utilizar el subconjunto de observaciones de Estados Unidos como base. Posteriormente se eliminaron las filas repetidas y se realizó una búsqueda de cada uno de los identificadores del dataset inicial en los demás datasets. Para agilizar esta operación, la comparación se hizo directamente sobre el identificador de cada video, por lo que, al tener un arreglo de identificadores en una región, simplemente se puede preguntar *if identificador in arreglo* y se hace la sumatoria correspondiente. En otras palabras, si el identificador de un video de Estados Unidos está presente en cualquier otro archivo se le suma a la cantidad de países que el video cubre. Por último, se consideraron a todos aquellos registros con cobertura de más de 1 país como internacionales, y por lo tanto el resto es de cobertura nacional. Esta columna calculada binaria se convierte en nuestra variable a predecir. El dataset resulta tener 3,461 videos que solo fueron tendencia en Estados Unidos y 2,890 videos que fueron tendencia internacionalmente.

Métodos

Si bien es cierto, al momento de hacer la exploración inicial de los datos se eliminaron algunas variables que se consideraron inútiles, todavía hay que hacer un análisis adicional para ingresar los datos a los diferentes modelos. Para esto se realizó un mapa de calor que muestra las correlaciones que cada variable tiene con cada otra variable en el dataset:



La visualización podrá parecer complicada, sin embargo, solo hay que considerar el recuadro amarillo. Este nos indica las variables que tienen mayor correlación en relación con el target. Es decir, el número de visualizaciones, de likes, dislikes y comentarios. De igual manera se utilizó un segmento del algoritmo de Random Forest para corroborar el resultado anterior. Al entrenar el modelo con todas las variables, asignó la mayor cantidad de importancia exactamente a las 4 variables que visualizamos hace unos momentos. Tomando en cuenta que ya se tocó el punto de Random Forest, es importante empezar a describir de manera breve qué algoritmos se utilizaron.

Como primer punto se entrenó un modelo de Decisión Tree. Un árbol de decisiones es un modelo de clasificación que se visualiza a partir de lo que es un árbol dentro del campo de estructuras de datos en computación. A grandes rasgos un árbol de decisión sigue un flujo recursivo para su construcción, esto es porque el árbol comienza por seleccionar la variable que mejor separe a los datos disponibles. En el caso de este proyecto, la métrica para seleccionar esta variable fue el coeficiente de Gini. El coeficiente de Gini es un cómputo que en pocas palabras describe lo siguiente:

La probabilidad de que, basado en una variable de separación, cada registro pertenezca a alguna de las clases descritas. Por ejemplo, si se decidió que la separación se realiza por videos con número de comentarios mayor o igual a 50,000 y un Gini de 0.60, esto significa que hay un 60 porciento de probabilidad de que cualquier video que tenga más de esa cantidad de comentarios pertenezca a la clase internacional. El algoritmo de decisión trees selecciona la variable con un mayor valor de Gini con la intención de facilitar la clasificación de niveles posteriores; así se va de manera recursiva hasta llegar a las hojas del árbol, en esta situación cada hoja representa una clasificación realizada por un camino específico en el árbol. Con esta breve aclaración de lo que es un árbol de decisión, ya se puede indagar sobre el algoritmo de Random Forest. Lo que hace este algoritmo es juntar los resultados de un conjunto finito de árboles de decisión, de esa idea surge el nombre de bosque. La parte estocástica del algoritmo se involucra con la selección de variables. Al momento de hacer las decisiones dentro de un árbol específico, el Random Forest no utiliza todas las variables disponibles en el dataset. Lo que hace es selecciones aleatoreas hasta encontrar las mejores decisiones. A causa de esta aleatoriedad y múltiple selección de variables es que se puede utilizar para computar el *feature importance* que anteriormente nos reveló que las variables numéricas que se seleccionaron inicialmente son las más relevantes para el dataset disponible. El siguiente modelo utilizado fue el clasificador de Bagging. El concepto de este modelo es muy similar al de Random Forest; la diferencia principal es que Bagging hace una mezcla de los conceptos de bootstrapping y agregación. En el caso del clasificador de Bagging hay una selección aleatoria de subdatasets a lo largo de las filas. Es decir, los clasificadores que forman parte del Bagging van a ser una mezcla aleatoria de distintas dimensiones (a lo largo de las filas) sacada del dataset original. En distintos términos se podría describir el proceso de bagging como el de construcción de datasets aleatorios (con todas las variables originales) y la sumatoria de los mismos para la votación final de clasificación. Por el otro lado, la aleatoriedad del Random Forest es directamente sobre las variables y de igual manera se hace una agregación de los clasificadores para obtener un mejor resultado. Es importante mencionar que los modelos de Random Forest y Bagging serán computacionalmente más caros que los árboles de decisión; esto es muy al considerar que los modelos más complejos van a formar grupos de árboles de decisión para aumentar la precisión y la exactitud de las clasificaciones. Este detalle se debe de considerar para aplicaciones que dependan mucho del tiempo o de recursos computacionales limitados.

Posteriormente se aplicó el método de boosting. Tal como se mencionó al principio de esta sección, todos los modelos utilizados se basan en la estructura de un árbol. En el caso de boosting también hay clasificaciones por medio de nodos, sin embargo, estos algoritmos son iterativos y determinan la clasificación final de una manera ligeramente distinta. Los algoritmos anteriores seleccionaban un subconjunto del dataset para hacer una agregación de clasificadores y posteriormente una votación para la creación del mejor clasificador posible. En el caso de boosting, se comienza con la selección de un subconjunto del dataset (con todo y selección de variables), se empieza la clasificación y la siguiente selección para la decisión depende de los errores de la anterior. Antes de hacer la decisión correspondiente, se evalúa el desempeño de la decisión que se acaba de realizar. Lo que ocurre es que se les asignan pesos mayores a las observaciones clasificadas incorrectamente, en las siguientes iteraciones de

reelección de observaciones y variables, las que tienen mayor peso van a recibir atención especial para que el clasificador reduzca la proporción del error. Cuando se concluye una ronda de clasificación inicial, la siguiente incluyen nuevas observaciones; estas nuevas observaciones entran con un peso de inicialización homogéneo, sin embargo, el cambio en los pesos de las anteriores causa un desface que obliga al clasificador a generalizar para observaciones que anteriormente no se pudieron clasificar correctamente. Dentro de las especificaciones técnicas de los modelos utilizados, se presentan dos variaciones de boosting. La primera, AdaBoost indica un cambio adaptivo, que obedece al proceso que se describe anteriormente, la votación se va a adaptar en base a clasificaciones anteriores. La segunda variación se denomina XGBoost, esta es una implementación especial de los gradient boost clasifiers; lo que hacen este tipo de clasificadores es que abordan el problema de clasificación como uno de optimización en base a pesos, estos se minimizan con técnicas similares al descenso del gradiente y otros algoritmos más avanzados.

En términos generales se desea encontrar el mejor clasificador utilizando árboles de decisión. La manera en la que se construyen cada uno de los clasificadores varía dependiendo del modelo que se vaya a entrenar. A continuación, se presentan los resultados de aplicar todos estos conceptos al dataset de videos en tendencia de Youtube.

Resultados

Con el objetivo de evitar la presencia de overfitting en los resultados de los modelos, se optó por hacer el entrenamiento con el método de cross validation dividido. A este método de entrenamiento se le conoce como KFold Cross Validation, esto es porque se seleccionan N parejas de conjuntos de datos de entrenamiento y de pruebas que engloban a todos as observaciones de un dataset. Se escogió una división de 10, es decir, se realizan 10 iteraciones en las que se escoge de manera aleatoria un 10% del dataset para ser evaluado como el subconjunto de prueba. A lo largo de las iteraciones se guardan los resultados y al final se puede obtener una métrica promedio para verificar el comportamiento del modelo. En cuanto a la métrica de evaluación se escogió la de exactitud; en términos de una matriz de confusión, la exactitud es la división de la suma de verdaderos negativos y verdaderos positivos con la suma de todas las observaciones. En pocas palabras es el porcentaje de observaciones que se predijeron correctamente. En la tabla a continuación se presentan los resultados por modelo:

Modelo	Detalles Adicionales	Exactitud con Entrenamiento	Exactitud con Prueba
Decision Tree	Todas las variables, Profundidad máxima de 5	0.776	0.755
Decision Tree	Con las variables seleccionadas Anteriormente, profundidad Máxima de 5	0.777	0.760
Random Forest	Utilizando 110 estimadores	0.777	0.763
Bagging Classifier	Utilizando 200 estimadores	0.776	0.765

AdaBoost	Utilizando 130 estimadores	0.772	0.750
XGBoost	Con 10 iteraciones de boost	0.775	0.751

La tabla está ordenada conforme se explicaron los modelos en la sección anterior, se podría declarar que conforme se avanza en la tabla el modelo se hace más complejo en términos conceptuales, así como en el procesamiento necesario para llegar a una convergencia. El modelo que tuvo el mejor resultado es el clasificador de Bagging. A este clasificador le sigue el Random Forest. En ambos modelos hay un detalle adicional de estimadores, este detalle representa la cantidad de arboles de decisión que el modelo va a evaluar para determinar el mejor clasificador. Con esta explicación se manifiesta la diferencia de rendimiento que alguno de estos puede tener comparado a un árbol de decisión simple. En cuanto a los clasificadores que utilizan el método de boosting, se obtuvieron resultados comparables; no obstante, tomando en cuenta la complejidad mencionada de los mismos no vale la pena utilizarlos sobre este conjunto de datos de videos. Esto puede ocurrir a causa de la poca variabilidad que hay a lo largo del dataset, puede ser que estos algoritmos de boosting sean eficientes para conjuntos de datos con mayor cantidad de variables y una variabilidad mayor. Esto se menciona porque durante el proceso de entrenamiento con estos métodos, se tuvo que realizar relativamente bastante hyperparameter tunning para reducir el overfitting de los datos. Tomando en cuenta los factores mencionados de los modelos evaluados, la simpleza y eficiencia del árbol de decisión lo coloca como el modelo que conviene más utilizar para este dataset. Es fundamental recordar que hay 2 modelos de árboles de decisión que se utilizaron, el que obtuvo mejor resultado es el que considera solamente las variables relevantes.

Conclusiones y Recomendaciones

Cada día hay más fuentes de generación masiva de datos en el internet que se le pueden presentar a cualquier individuo, en este caso, la gran cantidad de información que Youtube maneja funciona como un ejercicio interesante de predicción del tipo de videos que se presentan en la pestaña de videos en tendencia de diferentes países. Algún youtuber podría obtener uno de los modelos realizados y evaluar la posibilidad de que después del momento en que uno de sus videos entre en tendencia, este se pueda esparcir de manera internacional e incluso a lo largo de todo el mundo. En cuanto al flujo de investigación de ciencias de los datos, este proyecto es útil para demostrar la importancia de la información “escondida” dentro de un grupo de datos. En aplicaciones cotidianas, la información es inconsistente y puede ser que los stakeholders de una solución de datos estén pidiendo métricas y detalles que no se presentan necesariamente como una variable dentro de un dataset. Al momento de hacer la selección de las variables relevantes del dataset, se pudo demostrar que, en este caso específico, los resultados de una evaluación visual de un diagrama de correlación son prácticamente las mismas que las del resultado de importancia de un Random Forest. A lo largo del proyecto se fueron desarrollando, entrenando y evaluando distintos modelos de clasificación basados en un diseño de árboles. Los resultados revelaron que a pesar de que Bagging tuvo el mejor rendimiento, su exactitud no es lo suficientemente mayor a la del clasificador simple del árbol de decisión; por lo tanto, este último se seleccionó como el modelo a utilizar para las predicciones necesarias.

Se podría recomendar indagar sobre la funcionalidad que el API de Youtube ofrece para obtener información de videos normales y poder predecir si un video común puede llegar a formar parte de las tendencias. Además, es importante realizar el análisis presente utilizando como base otros países que forman parte del dataset general para verificar si se puede generalizar el modelo a nivel mundial y mejorar los resultados. Por último, se pueden aprovechar los datos de tiempo y verificar si hay alguna manera de registrar la trazabilidad para hacer un detector de potencial de tendencia en tiempo real.

Apéndice

- **API de Datos de Youtube:**
  - **fuente importante para análisis a futuro y continuar a poblar el dataset actual.**
  - https://developers.google.com/youtube/v3/docs/search/list
- **Popular Ensemble Methods: An Empirical Study**
  - Este es un paper en el que se estudian distintos métodos ensemble, las conclusiones a las que se llegan apoyan de cierta manera a los detalles que se mencionan en el reporte. Por ejemplo, demuestra que Bagging va a resultar en un mayor rendimiento y que los métodos de boosting van a causar overfitting frecuente sobre datasets ruidosos.
  - https://arxiv.org/pdf/1106.0257.pdf
- **Explaining Feature Importance by example of a Random Forest**
  - Utilización de Random Forest con el approach de selección de variables
  - https://towardsdatascience.com/explaining-feature-importance-by-example-of-a-random-forest-d9166011959e
- **Trending Youtube Video Statistics Dataset**
  - Información original que se utilizó para el análisis
  - https://www.kaggle.com/datasnaek/youtube-new