Universidad Tecnológica Metropolitana
Facultad de Ingeniería
Departamento de Informática y computación

Avance de Proyecto

Predicción de ventas según el género de videojuegos

Asignatura: Mineria de Datos
Integrantes: Daniel Silva Contreras
Fecha: 03/10/2017

Tabla de Contenido

1.-Introducción

La minería de datos o exploración de datos es un campo de la estadística y las ciencias de la computación referido al proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos. Utiliza los métodos de la inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos.

El objetivo general del proceso de minería de datos consiste en extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior.

En el transcurrir de los años las tecnologías y los avances con relación a la minería de datos se vieron involucrados en diferentes procesos de negocios y la industria de los videojuegos no se quedó atrás en este campo, la necesidad por conocer a sus consumidores y el gusto de estos es parte fundamental para sobrevivir en un ambiente tan competitivo como lo es este, se necesitan de diferentes datos para antes de siquiera comenzar la idea de proyecto en un nuevo videojuego.

Grandes compañías desarrolladoras han caído bajo el manto de cancelaciones, pérdidas, fracasos y en casos hasta la misma quiebra por el mal manejo de la información, por lo que en los últimos años estas empresas desarrolladoras han contratado servicios de minería de datos para poder presentar productos de calidad.

Para obtener la data necesaria para la realización de este análisis, se usó la plataforma online Kaggle, la cual proporciona un repositorio para que cada usuario o compañía publiquen sus datos y desde ahí se comienza un concurso abierto para que los expertos en Data Mining descarguen los datos y propongan soluciones a los problemas de las compañías que subieron el dataset.

La dataset elegida está relacionada con los datos de ventas de videojuegos por todo el mundo, estos datos contienen datos de juegos de diferentes plataformas y diferentes géneros como lo son shooter, acción, aventuras, entre otros. Se usarán estos datos para predecir que género y en que plataforma se realizaran mayores ventas en los años próximos.

2.- Marco teórico

2.1.- Metodología

Para realizar un análisis de minería de datos existen diferentes metodologías para la realización de este análisis, una de estas y la que se ocupará para este análisis es la metodología CRISP-DM.

CRISP-DM (Cross Industry Standard Process for Data Mining) proporciona una descripción normalizada del ciclo de vida de un proyecto estándar de análisis de datos, de forma análoga a como se hace en la ingeniería del software con los modelos de ciclo de vida de desarrollo de software. El modelo CRISP-DM cubre las fases de un proyecto, sus tareas respectivas, y las relaciones entre estas tareas. En este nivel de descripción no es posible identificar todas las relaciones; las relaciones podrían existir entre cualquier tarea según los objetivos, el contexto, y el interés del usuario sobre los datos.

La metodología CRISP-DM contempla el proceso de análisis de datos como un proyecto profesional, estableciendo así un contexto mucho más rico que influye en la elaboración de los modelos. Este contexto tiene en cuenta la existencia de un cliente que no es parte del equipo de desarrollo, así como el hecho de que el proyecto no sólo no acaba una vez se halla el modelo idóneo (ya que después se requiere un despliegue y un mantenimiento), sino que está relacionado con otros proyectos, y es preciso documentarlo de forma exhaustiva para que otros equipos de desarrollo utilicen el conocimiento adquirido y trabajen a partir de él.
El ciclo de vida del proyecto de minería de datos consiste en seis fases mostradas en la Ilustración siguiente:

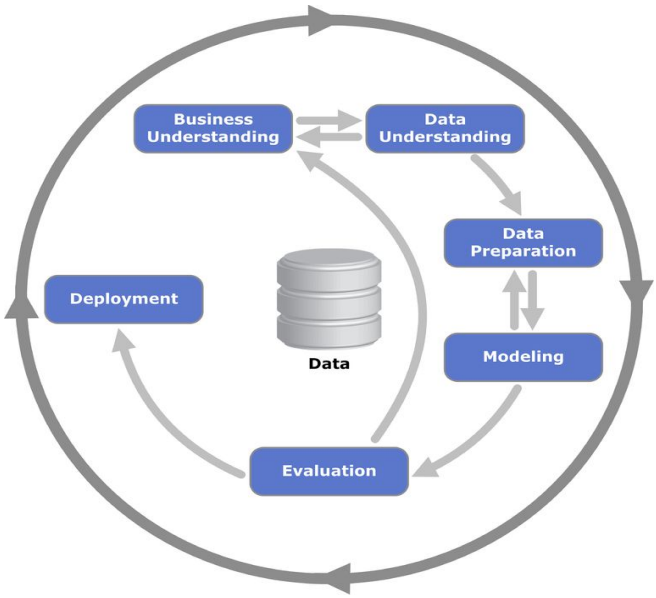

Ilustración 1.- Metodología CRISP-DM

La secuencia de las fases no es rígida: se permite movimiento hacia adelante y hacia atrás entre diferentes fases. El resultado de cada fase determina qué fase, o qué tarea particular de una fase, hay que hacer después. Las flechas indican las dependencias más importantes y frecuentes.

A continuación vamos a describir brevemente cada una de las fases.

1. **Comprensión del negocio:**
   Esta fase inicial se enfoca en la comprensión de los objetivos de proyecto. Después se convierte este conocimiento de los datos en la definición de un problema de minería de datos y en un plan preliminar diseñado para alcanzar los objetivos.

2. **Comprensión de los datos:**
   La fase de entendimiento de datos comienza con la colección de datos inicial y continúa con las actividades que permiten familiarizarse con los datos, identificar los problemas de calidad, descubrir conocimiento preliminar sobre los datos, y/o descubrir subconjuntos interesantes para formar hipótesis en cuanto a la información oculta.

3. **Preparación de datos:**
   La fase de preparación de datos cubre todas las actividades necesarias para construir el conjunto final de datos (los datos que se utilizarán en las herramientas de modelado) a partir de los datos en bruto iniciales. Las tareas incluyen la selección de tablas, registros y atributos, así como la transformación y la limpieza de datos para las herramientas que modelan.

4. **Modelado:**
   En esta fase, se seleccionan y aplican las técnicas de modelado que sean pertinentes al problema (cuantas más mejor), y se calibran sus parámetros a valores óptimos. Típicamente hay varias técnicas para el mismo tipo de problema de minería de datos. Algunas técnicas tienen requerimientos específicos sobre la forma de los datos. Por lo tanto, casi siempre en cualquier proyecto se acaba volviendo a la fase de preparación de datos.

5. **Evaluación:**
   En esta etapa en el proyecto, se han construido uno o varios modelos que parecen alcanzar calidad suficiente desde la una perspectiva de análisis de datos.
   Antes de proceder al despliegue final del modelo, es importante evaluar a fondo y revisar los pasos ejecutados para crearlo, comparar el modelo obtenido con los objetivos de negocio. Un objetivo clave es determinar si hay alguna cuestión importante de negocio que no haya sido considerada suficientemente. Al final de esta fase, se debería obtener una decisión sobre la aplicación de los resultados del proceso de análisis de datos.

6. **Despliegue:**
   Generalmente, la creación del modelo no es el final del proyecto. Incluso si el objetivo del modelo es de aumentar el conocimiento de los datos, el conocimiento obtenido tendrá que organizarse y presentarse para que el cliente pueda usarlo. Dependiendo de los requisitos, la fase de desarrollo puede ser tan simple como la generación de un informe o tan compleja como la realización periódica y quizás automatizada de un proceso de análisis de datos en la organización.

Esta metodología para proyectos de minería de datos no es la “más actual” o “la mejor”, pero es muy útil para comprender esta tecnología o extraer ideas para diseñar o revisar métodos de trabajo para proyectos de similares características.

2.2.- Análisis Descriptivo

Uno de los objetivos de la Estadística es el de describir en unas pocas medidas resumen las principales características de un amplio conjunto de datos, de forma que estas medidas reflejen lo más fielmente las principales peculiaridades de dicho conjunto. A esta rama de la Estadística se la denomina Estadística Descriptiva.

Otro de los objetivos de la Estadística es realizar conjeturas acerca de las medidas resumen de un conjunto de datos conociendo tan sólo una parte del mismo; esta rama se denomina Estadística Inferencial.

2.3.- Analisis Factorial

El Análisis Factorial es una técnica multivariada que nos permite reducir el “tamaño" de un problema sin “demasiada pérdida de información".

Supongamos que disponemos de un problema en el que queremos estudiar el comportamiento de X1, X2,..., Xn variables. Si n es un valor muy alto, dicho estudio será bastante dificultoso, por lo tanto parece razonable buscar una técnica que nos permitiera trabajar con un número de variables considerablemente menor.

En la mayoría de los casos, no todas las variables aportarán información relevante e incluso puede haber varias variables que miden conceptos “similares".

Por ejemplo, si disponemos de un conjunto de variables económicas en las que haya dos grupos claramente diferenciados, aunque utilicemos un número considerable de variables, realmente estaremos midiendo solamente dos “factores" diferentes, por lo tanto si conseguimos utilizar esos dos factores se simplifica mucho nuestro problema.

3.- Definición del problema

El problema es saber que tipo o género de juegos se venderán más en el futuro para crear más juegos de este tipo, haciendo la distinción de la plataforma en que se desarrollarán estos juegos, ya que el desarrollo para las diferentes consolas de juego es distinto y varían en costos.

4.- Solución Propuesta

La solución que se propone es crear un modelo que haga Minería de datos y analice los datos que se tienen del comportamientos de las ventas de los juegos en los años anteriores para poder predecir que género de juegos y en que plataforma producirán más ventas en el mundo, esto es para dar énfasis en ese género y así generar más ventas y con esto más ingresos para la empresa.
Para esto se tiene un dataset con los datos que se necesitan para entregárselo al modelo que se creará, por lo que se debe:

- Comprender de buena manera el negocio.
- Tener buen conocimientos de los datos del dataset.

4.1.- Objetivo General

Se plantea realizar un estudio predictivo de la dataset enfocándose en la principal premisa que es: “Determinar un modelo predictivo para pronosticar las ventas futuras de videojuegos”, buscando una solución óptima a través de sus factores o variables que tienen los datos extraídos y podrían influir en el desarrollo de nuevos juegos.

4.2.- Objetivos Especificos

Los objetivos específicos necesarios para lograr el objetivo principal son los siguientes:

- Preparar los datos.
- Análisis descriptivo.
- Análisis factorial.
- Generar un modelo.
- Evaluar los resultados.
- Despliegue.

5.- Hipotesis

La hipótesis predictiva para este proyecto es “De acuerdo a ciertas relaciones entre las variables, que tipo de juegos (género), basados en la plataforma que se deberían desarrollar”, cuya relación debe estar retroalimentada por la predicción de las ventas.

6.- Compresión de los Datos

6.1.- Dataset

- Rank - Ranking de las ventas totales
- Name - El nombre de los juegos
- Platform - Plataforma de lanzamiento de juegos (es decir, PC, PS4, etc.)
- Year - Año del lanzamiento del juego
- Genre - Género del juego
- Publisher - Editor del juego
- NA_Sales - Ventas en América del Norte (en millones)
- EU_Sales - Ventas en Europa (en millones)
- JP_Sales - Ventas en Japón (en millones)
- Other_Sales - Ventas en el resto del mundo (en millones)
- Global_Sales - Total de ventas en todo el mundo.

6.2.- Variables

6.2.1.- Descripción de las variables Input

- **Platform**: Describe las diferentes plataformas que existen para los distintos videojuegos. El dataset cuenta con 29 tipos de plataformas en la ilustración 2 se puede visualizar la cantidad de veces que cada plataforma se repite en todo el dataset.

```
DS      2163
PS2     2161
PS3     1329
Wii     1325
X360    1265
PSP     1213
PS      1196
PC       960
XB       824
GBA      822
GC       556
3DS      509
PSV      413
PS4      336
N64      319
SNES     239
XOne     213
SAT      173
WiiU     143
2600     133
GB        98
NES       98
DC        52
GEN       27
NG        12
WS         6
SCD        6
3DO        3
TG16       2
PCFX       1
GG         1
Name: Platform, dtype: int64
```

Ilustración 2: Variable Platform

- **Year**: Describe el año de lanzamiento del juego. En la Ilustración 3 se muestra cuantas veces se repiten los años en el dataset.

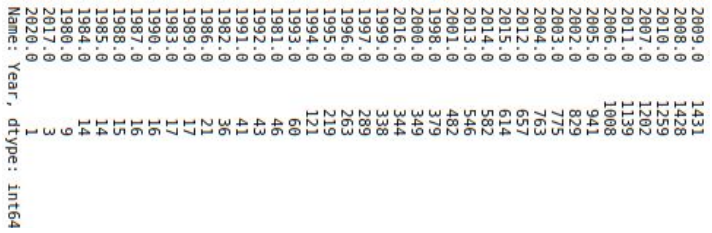

Ilustración 3: Variable Year

- **Genre**: Describe a que tipo de juego pertenece (shooter, Acción, aventura, etc). Se describen 12 géneros los cuales se muestran en la ilustración 4.

Ilustración 4: Variable Genero

- **Global_Sales**: Esta variable es importante porque contiene las ventas del juego en todo el mundo. Estas ventas para hacer mas visual el numero estan en millones, es decir si por ejemplo el valor de una celda del dataset es 82.64 ese valor es igual a 82.640.000 de ventas de ese juego.

6.2.2.- Descripción de las variables Output

- **Genre**: Como se mencionó en las variables input, esta variable cuenta con 12 tipos de juegos, los cuales pueden ser Action, Sports, Adventure, entre otros.

- **Platform**: Como se mencionó anteriormente esta variable contiene las distintas plataformas donde se pueden jugar o para la cual está desarrollado el tipo de juego.

- **Global_Sales**: Como se mencionó anteriormente, describe las ventas en millones de los videojuegos en todo el mundo.

7.- Preparación de los datos

Una vez comprendidos los datos y definidas las variables Input y Output, se comienzan a preparar los datos para entregarlos al modelo y este pueda analizarlos.
Lo que se hace es lo siguiente:

- Pasar los datos no numéricos a numéricos, es decir asignarles un número a cada grupo de datos no numérico, ya que el modelo no funciona con datos no numéricos.
- Las variables a cambiar son las siguientes:
    - Plataforma
    - Genero

Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Genero	Platafor...
Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74	0.0	26.0
NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	1.0	11.0
Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82	7.0	26.0
Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.0	0.0	26.0
GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.0	31.37	11.0	5.0
GB	1989.0	Puzzle	Nintendo	23.2	2.26	4.22	0.58	30.26	6.0	5.0
DS	2006.0	Platform	Nintendo	11.38	9.23	6.5	2.9	30.01	1.0	4.0
Wii	2006.0	Misc	Nintendo	14.03	9.2	2.93	2.85	29.02	4.0	26.0
Wii	2009.0	Platform	Nintendo	14.59	7.06	4.7	2.26	28.62	1.0	26.0
NES	1984.0	Shooter	Nintendo	26.93	0.63	0.28	0.47	28.31	8.0	11.0
DS	2005.0	Simulation	Nintendo	9.07	11.0	1.93	2.75	24.76	9.0	4.0
DS	2005.0	Racing	Nintendo	9.81	7.57	4.13	1.92	23.42	7.0	4.0
GB	1999.0	Role-Playing	Nintendo	9.0	6.18	7.2	0.71	23.1	11.0	5.0
Wii	2007.0	Sports	Nintendo	8.94	8.03	3.6	2.15	22.72	0.0	26.0
Wii	2009.0	Sports	Nintendo	9.09	8.59	2.53	1.79	22.0	0.0	26.0

Figura 1: Transformación de datos

En la siguiente imagen se puede visualizar la transformación de las variables Genre y Platform de un nombre a un número, para lo cual se crearon 2 columnas nuevas Género y Plataforma donde están dichos datos transformados.

8.- Analisis

8.1.- Multivariante

Para ingresar la base de datos a SPSS primero se tuvo que crear nuevas variables las cuales fueron una transformación a datos numéricos de las variables Genre y Platform, para que el spss los pueda leer de buena manera. Esta transformación se realizó mediante Jupyter, donde se crearon 2 nuevas columnas y se le asignó un número a cada género y plataforma.

Estadísticos

		Year	index	Rank	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Genero	Plataforma
N	Válido	16301	16598	16598	16572	16572	16572	16572	16572	16572	16572
	Perdidos	297	0	0	26	26	26	26	26	26	26
Media		2006,40	8298,50	8300,61	,2651	,1469	,0778	,0481	,5382	4,60	15,79
Error estándar de la media		,046	37,192	37,194	,00635	,00393	,00240	,00147	,01209	,028	,065
Mediana		2007,00	8298,50	8300,50	,0800	,0200	,0000	,0100	,1700	4,00	16,00
Moda		2009	0[a]	1[a]	,00	,00	,00	,00	,02	2	4
Desviación estándar		5,828	4791,574	4791,854	,81726	,50572	,30953	,18873	1,55613	3,548	8,395
Varianza		33,966	22959183,50	22961864,11	,668	,256	,096	,036	2,422	12,587	70,476
Rango		40	16597	16599	41,49	29,02	10,22	10,57	82,73	11	30
Mínimo		1980	0	1	,00	,00	,00	,00	,01	0	0
Máximo		2020	16597	16600	41,49	29,02	10,22	10,57	82,74	11	30
Suma		32706275	137738503	137773446	4392,75	2433,97	1290,01	797,64	8918,95	76193	261677

Figura 2: Estadisticos Descriptivos

8.1.1.- Análisis de datos

- La Base de datos contiene un total de 16.599 filas con datos de ventas de videojuegos.
- El valor máximo de ventas mundiales es de 82.74 millones.
- En promedio se realizan 0.5382 millones de ventas mundialmente.
- El Género de juego que tiene el máximo de las ventas es el “Platform”.
- La Plataforma que tiene el máximo de ventas es la “XOne.”

8.1.2.- Analisis Multivariante

A continuación, aplicado el Procedimiento Correlaciones bivariadas se obtiene la matriz de la matriz de correlaciones con los coeficientes de significación de cada correlación

Correlaciones

		V1	index	Rank	Year	NA_Sales	EU_Sales	Other_Sales	JP_Sales	Genero	Plataforma	Global_Sales
V1	Correlación de Pearson	1	1,000**	1,000**	,177**	-,401**	-,379**	-,333**	-,268**	,040**	-,085**	-,427**
	Sig. (bilateral)		,000	,000	,000	,000	,000	,000	,000	,000	,000	,000
	N	16598	16598	16598	16301	16572	16572	16572	16572	16572	16572	16572
index	Correlación de Pearson	1,000**	1	1,000**	,177**	-,401**	-,379**	-,333**	-,268**	,040**	-,085**	-,427**
	Sig. (bilateral)	,000		,000	,000	,000	,000	,000	,000	,000	,000	,000
	N	16598	16598	16598	16301	16572	16572	16572	16572	16572	16572	16572
Rank	Correlación de Pearson	1,000**	1,000**	1	,177**	-,401**	-,379**	-,333**	-,268**	,040**	-,085**	-,427**
	Sig. (bilateral)	,000	,000		,000	,000	,000	,000	,000	,000	,000	,000
	N	16598	16598	16598	16301	16572	16572	16572	16572	16572	16572	16572
Year	Correlación de Pearson	,177**	,177**	,177**	1	-,091**	,006	,041**	-,169**	-,010	,167**	-,074**
	Sig. (bilateral)	,000	,000	,000		,000	,408	,000	,000	,210	,000	,000
	N	16301	16301	16301	16301	16301	16301	16301	16301	16301	16301	16301
NA_Sales	Correlación de Pearson	-,401**	-,401**	-,401**	-,091**	1	,768**	,635**	,450**	-,022**	,042**	,941**
	Sig. (bilateral)	,000	,000	,000	,000		,000	,000	,000	,004	,000	,000
	N	16572	16572	16572	16301	16572	16572	16572	16572	16572	16572	16572
EU_Sales	Correlación de Pearson	-,379**	-,379**	-,379**	,006	,768**	1	,726**	,436**	-,012	,047**	,903**
	Sig. (bilateral)	,000	,000	,000	,408	,000		,000	,000	,125	,000	,000
	N	16572	16572	16572	16301	16572	16572	16572	16572	16572	16572	16572
Other_Sales	Correlación de Pearson	-,333**	-,333**	-,333**	,041**	,635**	,726**	1	,290**	-,019*	,055**	,748**
	Sig. (bilateral)	,000	,000	,000	,000	,000	,000		,000	,012	,000	,000
	N	16572	16572	16572	16301	16572	16572	16572	16572	16572	16572	16572
JP_Sales	Correlación de Pearson	-,268**	-,268**	-,268**	-,169**	,450**	,436	Efectúe una doble pulsación para activar		,084**	-,078**	,612**
	Sig. (bilateral)	,000	,000	,000	,000	,000	,00			,000	,000	,000
	N	16572	16572	16572	16301	16572	16572	16572	16572	16572	16572	16572
Genero	Correlación de Pearson	,040**	,040**	,040**	-,010	-,022**	-,012	-,019*	,084**	1	-,054**	-,001
	Sig. (bilateral)	,000	,000	,000	,210	,004	,125	,012	,000		,000	,887
	N	16572	16572	16572	16301	16572	16572	16572	16572	16572	16572	16572
Plataforma	Correlación de Pearson	-,085**	-,085**	-,085**	,167**	,042**	,047**	,055**	-,078**	-,054**	1	,029**
	Sig. (bilateral)	,000	,000	,000	,000	,000	,000	,000	,000	,000		,000
	N	16572	16572	16572	16301	16572	16572	16572	16572	16572	16572	16572
Global_Sales	Correlación de Pearson	-,427**	-,427**	-,427**	-,074**	,941**	,903**	,748**	,612**	-,001	,029**	1
	Sig. (bilateral)	,000	,000	,000	,000	,000	,000	,000	,000	,887	,000	
	N	16572	16572	16572	16301	16572	16572	16572	16572	16572	16572	16572

**. La correlación es significativa en el nivel 0,01 (bilateral).

*. La correlación es significativa en el nivel 0,05 (bilateral).

Figura 3: Matriz de Correlaciones bivariada

En cuanto a las correlaciones, podemos decir lo siguiente. Un p-valor (sig.) pequeño indica que se rechaza la hipótesis r=0 (no hay relación lineal entre las variables) y, por tanto, existe relación entre las variables.

Esto ocurre, por ejemplo, entre las variables Plataforma y Global_Sales: r=0.029 y p-valor=0.00 < 0.05.

8.2.- Factorial

Matriz de correlaciones[a]

		Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Genero	Plataforma
Correlación	Rank	1,000	,177	-,400	-,379	-,269	-,332	-,427	,038	-,086
	Year	,177	1,000	-,091	,006	-,169	,041	-,074	-,010	,167
	NA_Sales	-,400	-,091	1,000	,769	,451	,634	,941	-,021	,042
	EU_Sales	-,379	,006	,769	1,000	,436	,726	,903	-,011	,047
	JP_Sales	-,269	-,169	,451	,436	1,000	,291	,613	,084	-,079
	Other_Sales	-,332	,041	,634	,726	,291	1,000	,748	-,019	,055
	Global_Sales	-,427	-,074	,941	,903	,613	,748	1,000	,000	,029
	Genero	,038	-,010	-,021	-,011	,084	-,019	,000	1,000	-,055
	Plataforma	-,086	,167	,042	,047	-,079	,055	,029	-,055	1,000
Sig. (unilateral)	Rank		,000	,000	,000	,000	,000	,000	,000	,000
	Year	,000		,000	,204	,000	,000	,000	,105	,000
	NA_Sales	,000	,000		,000	,000	,000	,000	,004	,000
	EU_Sales	,000	,204	,000		,000	,000	,000	,079	,000
	JP_Sales	,000	,000	,000	,000		,000	,000	,000	,000
	Other_Sales	,000	,000	,000	,000	,000		,000	,009	,000
	Global_Sales	,000	,000	,000	,000	,000	,000		,492	,000
	Genero	,000	,105	,004	,079	,000	,009	,492		,000
	Plataforma	,000	,000	,000	,000	,000	,000	,000	,000	

a. Determinante = 1,125E-6

Figura 4: Matriz de Correlaciones

El determinante de la matriz se emplea como índice del tamaño de las correlaciones. Cuando su valor es elevado, las correlaciones dentro de la matriz son bajas, mientras que un determinante bajo indica que hay algunas correlaciones altas en la matriz. En este ejemplo su valor es 1.125E-6, es decir muy bajo lo que nos dice que existe una alta correlación entre las variables.

Prueba de KMO y Bartlett

Medida Kaiser-Meyer-Olkin de adecuación de muestreo		,350
Prueba de esfericidad de Bartlett	Aprox. Chi-cuadrado	223223,563
	gl	36
	Sig.	,000

Figura 5: Prueba KMO y Bartlett

La prueba de esfericidad de Bartlett contrasta la hipótesis de que los elementos fuera de la diagonal principal (las correlaciones) de la matriz de correlaciones sean cero. En este caso el valor del estadístico es 223.223,563 con un p-valor p=0, por lo que indica que se rechaza la hipótesis y se puede proseguir con el cálculo.

Otro índice es la medida de Kaiser-Meyer-Olkin que tiene en cuenta las correlaciones y las correlaciones parciales entre variables. Es aconsejable obtener valores grandes (más de 0.5) para que el análisis factorial pueda realizarse con garantías. En este ejemplo nos encontramos un valor de 0.350, que es aceptable pero que no entrega muchas garantías, ya que está por debajo del valor aceptable.

Comunalidades

	Inicial	Extracción
Rank	1,000	,435
Year	1,000	,736
NA_Sales	1,000	,815
EU_Sales	1,000	,837
JP_Sales	1,000	,513
Other_Sales	1,000	,682
Global_Sales	1,000	,975
Genero	1,000	,750
Plataforma	1,000	,472

Método de extracción: análisis de componentes principales.

Figura 6: Comunalidades

- Las comunalidades representan la varianza de cada variable explicada por los factores o las componentes principales.
- La comunalidad de una variable es la suma de las cargas factoriales asociadas a ella elevadas al cuadrado.
- Su cómputo se realiza a partir de la matriz de cargas factoriales. En el caso de componentes principales, cuando se retienen todas las variables, la comunalidad es siempre 1.
- Cuando se emplean otros métodos de extracción, la comunalidad inicial es el coeficiente de correlación múltiple entre cada variable y todas las demás.

Varianza total explicada

Componente	Autovalores iniciales			Sumas de extracción de cargas al cuadrado		
	Total	% de varianza	% acumulado	Total	% de varianza	% acumulado
1	3,919	43,541	43,541	3,919	43,541	43,541
2	1,266	14,068	57,610	1,266	14,068	57,610
3	1,030	11,445	69,055	1,030	11,445	69,055
4	,916	10,175	79,229			
5	,684	7,602	86,832			
6	,627	6,965	93,797			
7	,352	3,908	97,705			
8	,207	2,295	100,000			
9	7,713E-6	8,570E-5	100,000			

Método de extracción: análisis de componentes principales.

Figura 7: Varianza total explicada

Todos los datos, autovalores y varianzas explicadas antes y después de la extracción y después de la rotación se ve en esta figura.

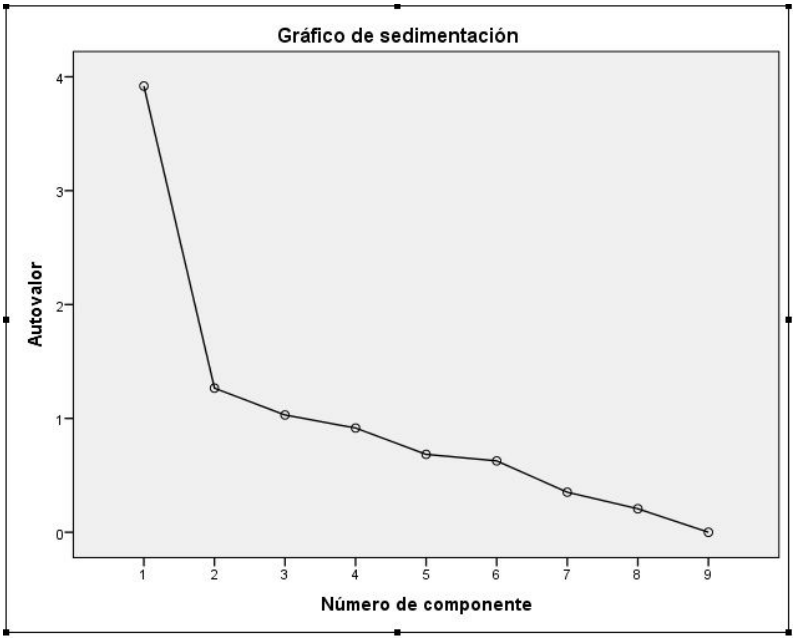

Figura 8: Gráfico de sedimentación.

Este gráfico e un método alternativo para la selección del número de factores, en este gráfico se representan gráficamente los autovalores (eje de ordenadas) para cada factor (eje de abscisas). El análisis visual del gráfico de se centra en la búsqueda de un punto de inflexión en la gráfica, lo que habitualmente se produce con valores por debajo de 1.

Matriz de componente[a]

	Componente		
	1	2	3
Rank	-,540	,095	,365
Year	-,106	,740	,422
NA_Sales	,902	,029	,002
EU_Sales	,903	,124	,077
JP_Sales	,615	-,351	,104
Other_Sales	,796	,208	,077
Global_Sales	,986	,011	,056
Genero	-,005	-,285	,818
Plataforma	,046	,668	-,154

Método de extracción: análisis de componentes principales.

a. 3 componentes extraídos.

Figura 9: Matriz de componente

Se pueden observar 3 factores de comportamiento, y los valores nos indican que las variables Year y Plataforma tienen alta carga en el segundo factor, mientras que las variables que marcan las ventas como lo son EU_Sales, NA_Sales, Global_Sales entre otros tienen una carga alta en el primer factor. Se podria decir que se usa un mecanismo diferente en cada factor.

En la matriz se visualiza una relación de cada componente con las variables arrojadas por el método y las variables que dan un valor muy negativo significa que su relación es baja con esa componente.

9.- Modelos

9.1.- Visualización de Datos

A partir del entorno interactivo de Jupyter Notebook trabajaremos con el fin de visualizar los resultados simultáneamente. En este trabajo se realizó una hoja de trabajo para cada tipo de algoritmo a aplicar al modelo, así como otra para la visualización de datos.

El primer paso es importar las librerías que son necesarias para el análisis de datos y que hemos descrito anteriormente. Se simplifica el nombre de las librerías, por lo que cada vez que empleemos alguna de las funciones o comandos, por ejemplo pandas, se especificará pd.”Nombre de la función”

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Figura 10: Librerias

Para conocer que datos posee el conjunto necesitamos abrir el archivo que contiene el conjunto de datos denominado “juegos” mediante el comando de Pandas “pd.read_csv" y el directorio donde se encuentra el archivo.
Este archivo es al cual se le aplicaron las transformaciones de caracteres a números de las variables “Genre” y “Platform”.

```
#Traer datos del archivo
dt = pd.read_csv('juegos.csv')
df = pd.DataFrame(dt)
df
```

Figura 11: Apertura de datos

Se define en un dataframe, mostrándose un pequeño conjunto de datos a partir de dt.head(), que permite mostrar las cinco primeras filas del dataset, ya que la tabla completa es demasiado extensa:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Genero	Plataforma
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74	0	26
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	1	11
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82	7	26
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00	0	26
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37	11	5

Figura 12: Matriz de datos

Para averiguar de qué tamaño es el conjunto de datos se emplea el siguiente código, significa que tenemos 15 variables (columnas) y 16.598 instancias o registros (filas):

```
#Tamaño de la muestra
dt.shape
```

```
(16598, 15)
```

Figura 13: Tamaño Muestra

Usamos el comando *dt.describe()* para el análisis de los estadísticos de cada variable del Dataset, en el cual se muestra el número de muestras, el valor medio, la desviación típica, el valor mínimo y máximo, y los distintos percentiles para cada variable cuantitativa:

```
#Describir las variables
dt.describe()
```

	Unnamed: 0	index	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Genero	Plataforma
count	16598.000000	16598.000000	16598.000000	16327.000000	16598.000000	16598.000000	16598.000000	16598.000000	16598.000000	16598.000000	16598.000000
mean	8298.500000	8298.500000	8300.605254	2006.406443	0.264667	0.146652	0.077782	0.048063	0.537441	4.595011	15.797988
std	4791.574219	4791.574219	4791.853933	5.828981	0.816683	0.505351	0.309291	0.188588	1.555028	3.545754	8.392298
min	0.000000	0.000000	1.000000	1980.000000	0.000000	0.000000	0.000000	0.000000	0.010000	0.000000	0.000000
25%	4149.250000	4149.250000	4151.250000	2003.000000	0.000000	0.000000	0.000000	0.000000	0.060000	2.000000	7.000000
50%	8298.500000	8298.500000	8300.500000	2007.000000	0.080000	0.020000	0.000000	0.010000	0.170000	4.000000	16.000000
75%	12447.750000	12447.750000	12449.750000	2010.000000	0.240000	0.110000	0.040000	0.040000	0.470000	8.000000	21.000000
max	16597.000000	16597.000000	16600.000000	2020.000000	41.490000	29.020000	10.220000	10.570000	82.740000	11.000000	30.000000

Figura 14: Estadisticos descriptivos

9.2.- Gráficas de las variables a evaluar

Se procede a la visualización de los distintos datos, con el fin de obtener mayor información del dataset. Representamos la variable “Genero”

El siguiente gráfico hace referencia al número de usuarios totales, “count”, por cada tipo de género, donde cabe destacar el alto nivel de usuarios en el género 2 es decir “Action”.

```
#Otra grafica
sns.countplot(x="Genero", data= dt, palette = "Greens_d")
plt.show()
```

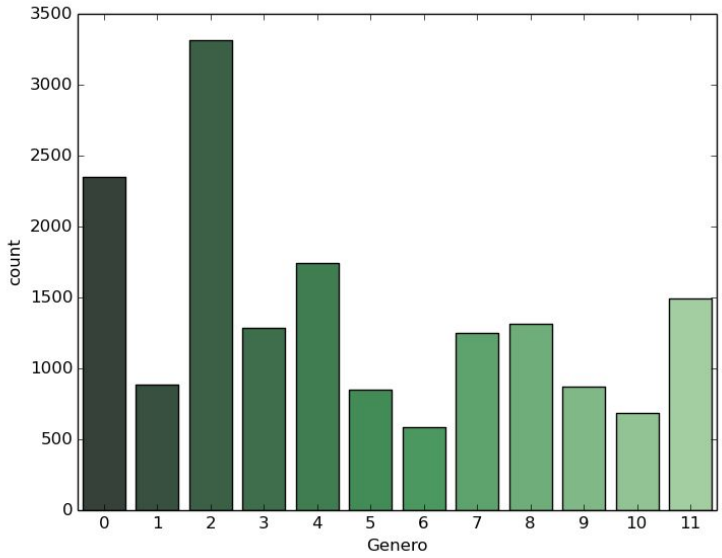

Figura 15: Nivel de ventas por género

A continuación se presenta un gráfico que relaciona el número de usuarios totales de ventas, “Global_Sales”, en cada genero, tambiéen por plataforma y año:

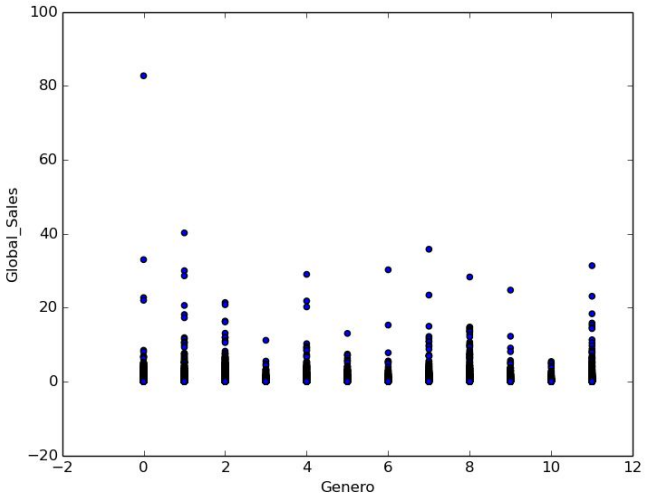

Figura 16: Genero vs Ventas

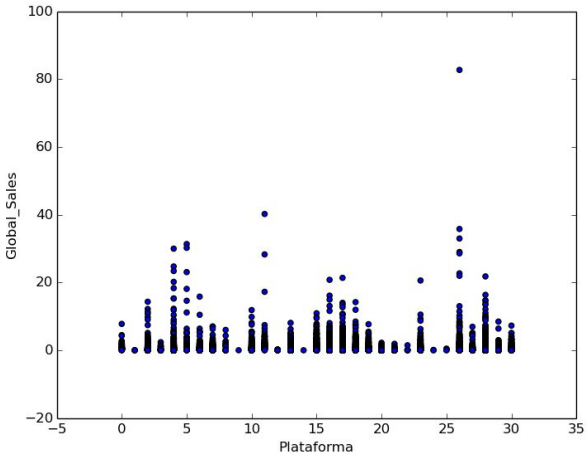

Figura 17: Plataforma vs ventas

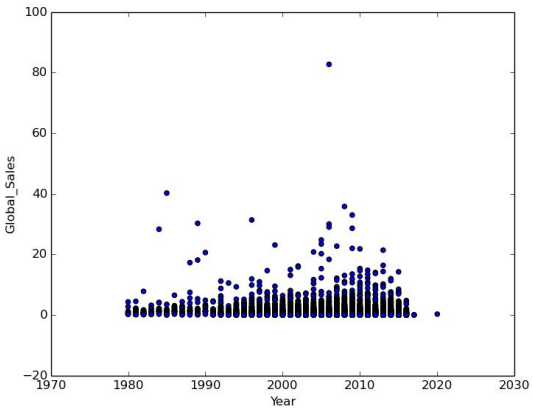

Figura 18: Años vs ventas

Por último, se establece la relación entre las distintas variables del modelo por la función de correlación entre las variables a partir del código:

```
##Correlacion de varibles
corrvariables = dt[["Rank","Year","NA_Sales","EU_Sales","Other_Sales",
                    "Global_Sales","Genero","Plataforma"]].corr()
mask = np.array(corrvariables)
mask[np.tril_indices_from(mask)] = False
fig,ax = plt.subplots()
fig.set_size_inches(20,10)
sns.heatmap(corrvariables, mask = mask,vmax = .10,
            square = True, annot = True)
plt.show()
```

Figura 19: Correlación variables

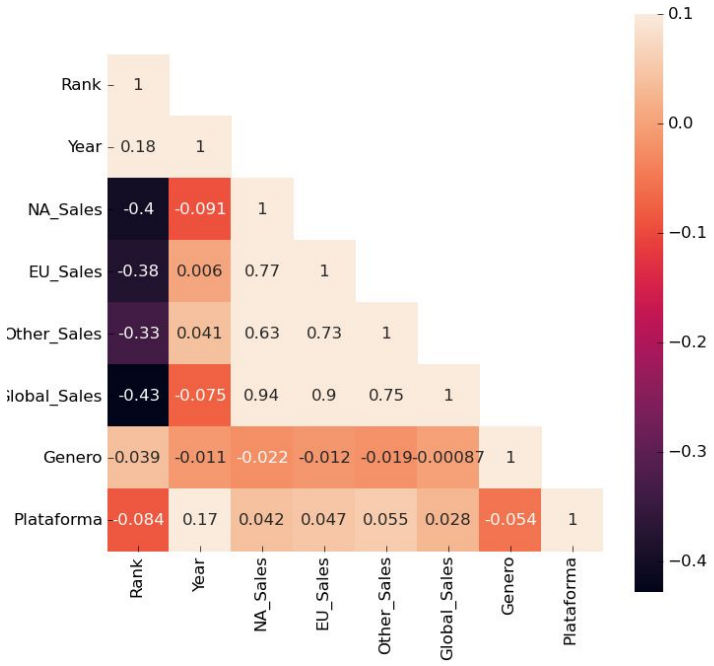

Figura 20: Correlación entre las distintas variables

Analizando la Figura 20 se aprecia relación entre las variables de Genero, Year y Plataforma, con el total de ventas (Global_Sales), siendo mayor en este último tipo, ya que las ventas de los juegos dependen mucho de la plataforma de dicho juego no tanto así del género y el año.

10.- Definición de algoritmos a usar

10.1.- Arbol de decisión

Es modelo predictivo que emplea técnicas de análisis discriminantes, la desviación, para poder predecir los valores a partir de una función establecida por el conjunto de variables predictoras.

La metodología de resolución es mediante un grafo etiquetado e implantado en todos sus nodos, basándose en la división de nodos divididos a su vez en subnodos a partir de ciertos criterios definidos por los conceptos adquiridos basados en los históricos, que reduzcan al máximo la varianza. Están constituidos de las siguientes partes:

- Nodos interiores (atributos)
- Arcos (posibles valores del nodo origen)
- Hojas (valor de la regresión)

El procedimiento que lleva a cabo la preparación del árbol de decisión es:

- **Sobreajuste**: a partir de los ejemplos de entrenamiento del modelo, si existe una hipótesis que se ajusta peor pero actúa mejor sobre la distribución completa de las instancias o atributos que presentan una aparente regularidad pero no son relevantes en realidad.

- **Ruido**: errores presentes en la base de datos a tratar o causado por sobreajuste del modelo.

- **Poda**: evita ruidos y errores de sobreajustes a partir de evitar el desarrollo del árbol antes de que se ajuste perfectamente, o transformar a condiciones de regla directamente en la aplicación del árbol, ambas son poda a posteriori.

Para la búsqueda del mejor árbol de decisión se examina el árbol más simple y corto, con el menor número de atributos es capaz de predecir la decisión. El nodo terminal devuelve un valor promedio de la salida, por lo que se validará el modelo mediante la validación cruzada evaluando el error cuadrático medio.

10.2.- Random Forest

Es un conjunto de árboles de regresión aleatorios, de amplio uso, ya que presenta un rendimiento especialmente bueno para datos de alta dimensionalidad. Se ajusta a un número de árboles de decisión en varias submuestras del conjunto de datos, para utilizar un valor promedio que mejora la precisión de la predicción y controla el ajuste excesivo, consiguiendo así mejores resultados que con el árbol de decisión. Cabe destacar, que el tamaño de la muestra es el mismo que el tamaño del dataset de entrada.

La metodología para el modelo de predicción es la siguiente:

1. Decidir qué variables van a ser los datos de entrada al bosque para obtener el valor de la variable a predecir, que depende de la definición de los datos de entrada.
2. Aleatoriamente se crea el conjunto de entrenamiento a partir del conjunto de datos de entrada.
3. Aplicación de “cross validation” (Validación cruzada) al conjunto de entrenamiento con el fin de eliminar el sobreajuste del modelo.
4. En cada división del árbol en cada nodo, la búsqueda de la mejora variable para dividir los datos no se realiza sobre todas las variables sino sobre el subconjunto de las mismas, que se realiza de forma aleatoria.
5. Los anteriores procesos se repiten iterativamente hasta llegar tener un conjunto de árboles de decisión entrenados sobre diferentes datos y atributos del conjunto.
6. Una vez entrenado el algoritmo, la evaluación de la predicción se realiza con el conjunto de árboles, teniendo en cuenta el valor promedio de los resultados.

Por otro lado, la técnica de random forest puede ser paralelizado eficazmente puesto que cada árbol puede construirse de manera independiente.

Además, el comportamiento del algoritmo es en función del número de árboles que incorpora en el algoritmo entrenado. Con lo cual un incremento del número de árboles permite una mayor diversidad de los mismos consiguiendo una reducción del error, aunque cabe destacar que a partir de un determinado número de árboles la mejora se estanca.

Cuanto mayor es la correlación entre dos árboles cualesquiera, menor será el error producido por el algoritmo.

11.- Implementación de modelos

11.1.- Árbol de decisión

Para comenzar con la aplicación del Árbol de Decisión se debe importar de la librería Sklearn el algoritmo básico. A continuación se representa la acción de importar la librería, y leer el archivo con los datos:

```
#Librerias
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
import os
import matplotlib.pylab as plt
from sklearn.cross_validation import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
import sklearn.metrics
```

```
#Leemos el archivo
dt=pd.read_csv('juegos.csv')
dt.head()
```

Figura 21: Librerías árbol de decisión

A continuación se muestran una serie de comandos para la preparación de los datos antes de realizar el arbol de decision:

```
#Eliminamos datos missing
data_clean = dt.dropna()
```

```
#Indicamos las variables predictorias y objetivo
predictors = data_clean[['Plataforma','Genero','Global_Sales','Year']]
targets = data_clean.Genero
```

```
#Muestra de entrenamiento al 60%
pred_train,pred_test,tar_train,tar_test = train_test_split(predictors,targets,
                                        test_size = 0.6)
```

```
#Comprobamos tamaño de las muestras
pred_train.shape
pred_test.shape
tar_train.shape
tar_test.shape
```

```
(9775,)
```

```
#Construccion del arbol de decision
classifier = DecisionTreeClassifier()
classifier = classifier.fit(pred_train, tar_train)
```

Figura 22: Preparación de datos y construcción de árbol de decisión

La figura anterior nos muestra que se debe realizar una eliminación de datos *missing,* es decir datos que faltan, también indicamos cuales son las variables predictoras y cual es el target o variable objetivo de nuestra predicción.
Luego se crea la muestra de entrenamiento que será al 60% (Porcentaje variable), se comprueban los tamaños de las muestras y se procede a construir el árbol.

A continuación se predicen los valores del grupo test y se pide la matriz de confusión, esta última, es una herramienta que permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado.

```
#Predecimos para los valores del grupo Test
predictions = classifier.predict(pred_test)
```

```
#Pedimos la matriz de confusión de las predicciones del grupo Test
sklearn.metrics.confusion_matrix(tar_test, predictions)
```

Figura 23: Valores Predictivos y matriz de confusión

Luego de esto sacamos el índice de *Accuracy Store,* el cual mide la precisión, el cual resume la Matriz de Confusión y la cantidad de aciertos.

```
#Sacamos el índice Accuracy Score
sklearn.metrics.accuracy_score(tar_test, predictions)
```

```
1.0
```

Figura 24: Indice de Accuracy Store

Este índice indica el porcentaje de certeza del modelo, para este modelo dicho índice es del 100%, lo que puede ser contraproducente, ya que puede haber un sobreentrenamiento del modelo.

Por último se importan las librerías para dibujar el árbol de decisión, el cual se crea en un archivo aparte con extensión .dot.

```
#Para dibujar el árbol hay que importar otra serie de cosas
from sklearn import tree
from io import StringIO
from IPython.display import Image
```

```
#Pintamos el árbol
with open("tree.dot", "w") as f:
    f = tree.export_graphviz(classifier, out_file=f)
```

Figura 25: Dibujar árbol de decisión

Finalmente se presenta el árbol que dio como resultado:

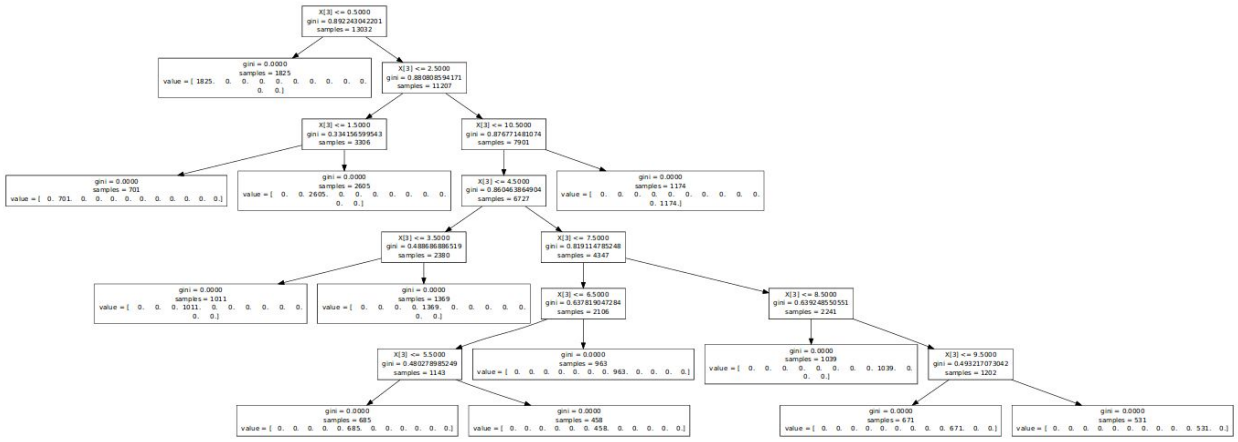

Figura 26: Árbol de decisión

11.2.- Random Forest

Método denominado “bosque aleatorio”, es la aplicación del algoritmo anterior en varias ocasiones. Para aplicarlo es necesario importar una serie de librerías, primero para, como se muestra en la siguiente imagen, además del resto de procedimientos anteriormente descritos en los algoritmos como es la partición del conjunto de entrenamiento:

```
#Librerias
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
import os
import matplotlib.pylab as plt
from sklearn.cross_validation import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
import sklearn.metrics
from sklearn import datasets
from sklearn.ensemble import ExtraTreesClassifier
```

```
dt=pd.read_csv('juegos.csv')
dt.head()
```

Figura 27: Librerías lectura de datos

A continuación eliminamos los datos con valores missing, ya que Python no puede hacer árboles con datos missing, es decir valores o datos que faltan. También indicamos las variables con que las que se predecirá y nuestro target, es decir el objetivo.

Y por último creamos nuestra muestra de entrenamiento y prueba, la cual será al 80%, es decir tomará el 80% de los datos del Dataset.

```
#Eliminando valores missing
data_clean = dt.dropna()
```

```
#Indicamos las variables predictorias y objetivo
predictors = data_clean[['Rank','NA_Sales','EU_Sales','Genero','JP_Sales',
                         'Other_Sales','Global_Sales','Plataforma','Year']]
targets = data_clean.Genero
```

```
#Muestra de entrenamiento al 80%
pred_train,pred_test,tar_train,tar_test = train_test_split(predictors,targets,
                                          test_size = 0.8)
```

Figura 28: Preparación de datos

A continuación se importan las librerías para el algoritmo Randon Forest propiamente, luego de eso iniciamos el algoritmo con una cantidad de 25 árboles.

Luego se construye el modelo sobre los datos entrenados, tanto para las variables predictoras como para el target. Luego se realizan las predicciones correspondientes para los valores del grupo test y se pide la matriz de confusión, al igual que en el algoritmo anterior (Árbol de decisión).

```
#Importamos librerias para random forest
from sklearn.ensemble import RandomForestClassifier
```

```
#Iniciamos algoritmo con 25 arboles
classifier=RandomForestClassifier(n_estimators=25)
```

```
#Construimos modelo sobre datos entrenados
classifier=classifier.fit(pred_train,tar_train)
```

```
#Predecimos para los valores del grupo Test
predictions=classifier.predict(pred_test)
```

```
#Pedimos la matriz de confusión
sklearn.metrics.confusion_matrix(tar_test,predictions)
```

Figura 29: Construcción algoritmo

Seguimos obteniendo el índice de accuracy store al igual que en el algoritmo anterior, el cual resume la Matriz de Confusión y la cantidad de aciertos.

Luego iniciamos el ExtraTreesClassifier, esto es para obtener la importancia de cada variable, y después de iniciado el modelo, se ajusta.

```
#Indice de Accuracy_score
sklearn.metrics.accuracy_score(tar_test, predictions)
```

```
0.93718455872323014
```

```
#iniciamos ExtraTreesClassifier
model = ExtraTreesClassifier()
```

```
#Ajustamos el modelo
model.fit(pred_train,tar_train)
```

```
ExtraTreesClassifier(bootstrap=False, compute_importances=None,
           criterion='gini', max_depth=None, max_features='auto',
           min_density=None, min_samples_leaf=1, min_samples_split=2,
           n_estimators=10, n_jobs=1, oob_score=False, random_state=None,
           verbose=0)
```

Figura 30: Iniciación de modelo

En la figura 30 se muestra el índice de accuracy store, el cual nos da con 93%, lo que es bastante bueno, ya que nos dice que el modelo no estará sobreentrenado y es un buen porcentaje de certeza.

Luego listamos la importancia de cada variable de nuestro dataset.

```
#Pedimos que nos muestre la importancia de cada variable
list(model.feature_importances_)
```

```
[0.032971853864155715,
 0.021373529899622774,
 0.022227936057063045,
 0.02337671266662162,
 0.014814434554160216,
 0.026815637468276365,
 0.031648071717375735,
 0.78655507968879168,
 0.040216744083932901]
```

Figura 31: Importancia de cada variable

Como se puede ver hay una variable que es bastante importante con respecto a las demás, esto es porque esa variable además de ser una predictora, es la variable objetivo, es decir “Genero”, por ende la importancia de esta es mucho mayor que las demás.

A continuación se presenta una gráfica donde se puede visualizar la diferencia que hay entre la importancia de cada variable.

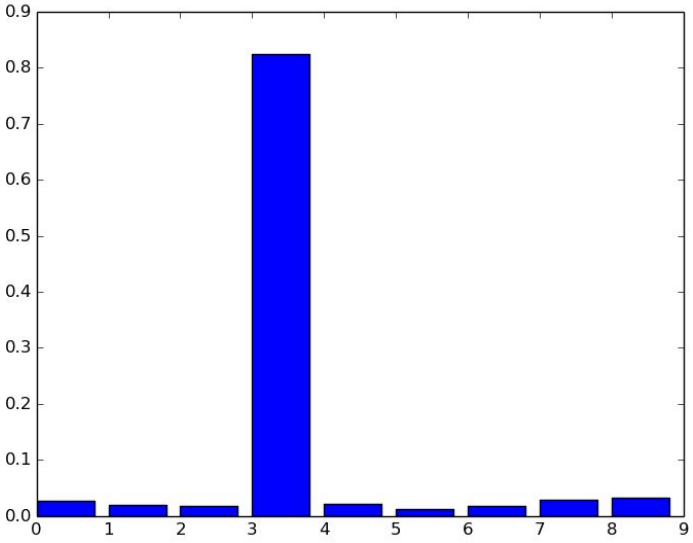

Figura 32: Gráfico de importancia de las variables

Por último, obtenemos el gráfico que nos indica si los árboles que se tuvieron que construir fueron suficientes o no:

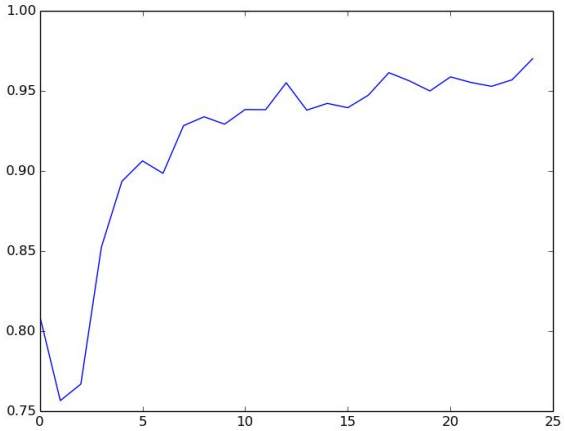

Figura 33: Gráfico árboles suficientes

Como punto negativo, en Random Forest no se interpretan los árboles que se construyen. Lo que obtenemos es solamente la importancia de las variables explicativas.
Random Forest se utiliza solo para clasificar la importancia de las variables en la predicción de la variables objetivo.

Sabemos cuales son las variables predictivas más importantes, pero no necesariamente sus relaciones entre sí.