AUTORES

Julen Rodríguez Meneses
Miguel Sagaseta de Ilúrdoz Sánchez

Predictor automatizado para la selección del entretenimiento electrónico

El cine, los libros e incluso los videojuegos pueden ser considerados como la octava maravilla del mundo. Queremos dar, por tanto, una recomendación fiel a los gustos del usuario para su disfrute superlativo.

MOTIVACIÓN

Los videojuegos son usados como hobbie o profesión por todo tipo de personas. Es muy importante que cada uno de ellos tenga la mejor experiencia posible y se ajuste a las preferencias personales.

¿Seremos capaces de predecir correctamente?

Conjunto de datos

	Nombre	Fecha Salida	Genero	Plataforma	Portable	Desarrolladora	Rating ESRB	metaScore	userScore	Multijugador	Online
0	The Legend of Zelda: Skyward Sword	2011	general	[Wii]	0	Nintendo	E10+	93.00	8.1	0	0
1	Total War: Shogun 2	2011	general	[PC]	0	Creative Assembly	T	90.00	8.4	1	1
2	Crysis	2007	action	[PC, PlayStation 3, Xbox 360]	0	Crytek	M	84.33	7.6	0	1
3	Allegiance	2000	sci-fi	[PC]	0	Microsoft Game Studios	E	86.00	7.5	1	0

He aquí una muestra de los datos empleados. Cada uno de estos consta de quince atributos. Trataremos de predecir el nombre del juego en cuestión. No obstante, contamos con muchos datos a tratar.

Por lo tanto, necesitamos realizar unas aclaraciones previas:

- La mayoría de variables, **genero**, **portable**, **online**, ..., son binarias, fácilmente tratables.
- Existen una serie de variables las cuales provienen como resultado de otras distintas, estas son **el Rating ESRB** y ambos **Scores**. Nos centraremos en clasificadores especializados para estas dos variables particulares.

División de datos para predictores previos (**5424 juegos totales**)

	mild_fantasy_violence	cartoon_violence	mild_cartoon_violence	animated_violence	mild_animated_violence	edutainment	crude_humor	mature_humor	comic_mischief	Rating
...	0	0	0	0	0	0	0	0	1	E10+
...	0	0	0	0	0	0	0	0	0	T
...	0	0	0	0	0	0	0	0	0	M
...	0	0	0	1	0	0	0	0	0	E

	Review	Score
...	I've been following the total war series since tw: rome. i have to say shogun 2 is probably the best tw series of the bunch. playing on normal, individual battles are generally not a challenge for the experienced player (most battles end with a heroic victory) but still, you can tell that the battlefield ai is much improved over previous releases. the campaign ai is a lot better too.	9
...	this would have been a great game . . . if they'd finished developing it. buggy, crash prone, and utterly unplayable. don't buy it for at least a few months - the fact that they're apparently using fans of the series to do play testing rather than doing it themselves is frustrating, to say the least.	4
...	great game.... thank you ca for an awesome game graphics are superb, gameplay is very awesome, what more can you ask for? if you like the total war series then dont even hesitate to buy this game.	10
...	poor battle ai. they just mindlessly swarm you or hill camp. also the ai just loves to chase your light cav with their yari inf. as a result fought out battles turn into a disgusting micro work. no tactics, no strategy, no soul. after a couple of turns miraculously all ai units wander around with 4-5 exp chevrons. the campaign ai is only there to keep you busy with no strategic personal goals. it's just a product of the fear by the devs that the game would possibly be too easy and pose no "challenge".	5

X = Datos **binarios** (0/1) // y = **strings**

Columna derecha como y
Resto columnas como X

X = **reviews** // y = **int** [0-10]

train_test_split

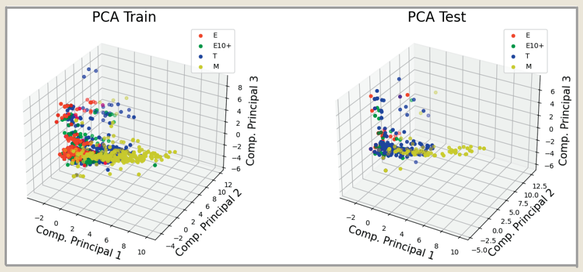
- Datos **train** -> 84,99% (**4610 juegos**)
- Datos **test** -> 15,01% (**814 juegos**)

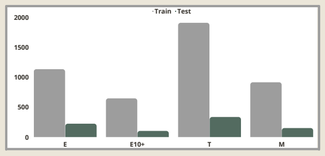
Predictor ESRB

Datos a utilizar

Existen un total de 48 características. Haremos un tratamiento previo:

- **Escalamiento** para tener datos en rangos parecidos.
- Reducción de dimensionalidad **PCA** hasta una dimensión representable.





Se observan varios datos:

- Clases **E** y **M** claramente diferenciadas
- Clase **T** en punto medio
- Clase **E10+** prácticamente indiferenciable

Usamos **GridSearchCV** para encontrar los mejores híper-parámetros y predictores.
Se usar **StratifiedKFold** para la división de *train* y *validación* dentro (2 splits).

Utilizamos todo tipo de clases de predictores:

- **Supervisados**:
  - **Clasificación**: Regresión Logística, Naive Bayes, Red Neuronal, SVMs, Ensembles.
  - **Regresión**: Regresión lineal.
- **No Supervisados**:
  - Kmeans.

Resultados obtenidos

	Predictor	Accuracy
0	svm	0.706388
1	ovo	0.706388
2	randomForest	0.705160
3	bagging	0.703931
4	adaboost	0.701474
5	decisionTree	0.696560
6	ova	0.681818
7	regresionLogistica	0.675676
8	redNeuronal	0.662162
9	nb	0.624079
10	kmeans	0.276413
11	regresionLineal	0.199017

Hiper-parametros *SVM*

- **C**: 1
- **gamma**: 5
- **kernel**: rbf
- **decision_function_shape**: *ovo*

Hiper-parametros *RandomForest*

- **n_estimators**: 20
- **criterion**: gini
- **min_samples_leaf**: 2
- **min_samples_split**: 10
- **max_features**: *None*
- **class_weight**: *None*
- **bootstrap**: *True*

- Los hiper-parámetros obtenidos en *SVM* indican que se intenta **obtener** una **varianza baja**.
- El ***kernel*** utilizado, ***rbf***, se utiliza para poder ajustar al máximo los parámetros.
- Los hiper-parámetros obtenidos en el mejor *Ensemble* indican que se necesitan **varios datos iniciales** para crear unos arboles decentes.
- Un claro **uso** del ***bootstrap*** para poder crear un predictor que **no sobre-aprenda**.

Resultados *SVM*

Obtenemos las siguientes conclusiones:

- Como cabe de esperar, aumenta el porcentaje de acierto al aumentar las iteraciones.
- La clase *E10+* tiene un acierto nefasto, lo cual es, sin duda, la razón del bajo porcentaje de acierto en el predictor.

	precision	recall	f1-score	support
E	0.69	0.74	0.72	223
E10+	0.55	0.23	0.32	104
M	0.87	0.77	0.82	151
T	0.68	0.80	0.73	336
accuracy			0.71	814
macro avg	0.70	0.64	0.65	814
weighted avg	0.70	0.71	0.69	814

Observamos el **recall** decente, menos el de la clase *E10+*. *Se* trata un **problema** del **mundo real**. Las otras clases son medianamente diferenciables ya que son juegos para niños(*E*), jovenes(*T*), y adultos(M). La clasificación *E10+*, sin embargo, se solapa con sus vecinas.

Usamos el **accuracy** como medida de calidad. **Necesitamos tener la certeza de que, si nos asigna una clase, sea verdaderamente la propia**. Podríamos usar el **recall**, con ella nos aseguraríamos de acertar todas las clases, pero **nadie nos asegura que la clase predicha sea la correcta**.

Predictor Reviews

Datos a utilizar

Realizamos nueva división para obtener datos de *validación* manuales

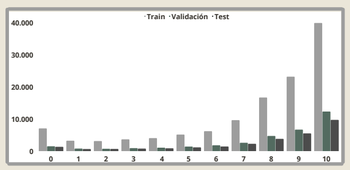
train_test_split

- Datos **train** -> 67,99%
- Datos **validación** -> 17%
- Datos **test** -> 15,01%

Obtenemos una gran cantidad de reviews. Se necesita realizar un preprocesamiento previo.

Se usa la librería **nltk**:

- **Stop-words and non-words removal**
- **Lemmatización**
- **Stemming**



Bag of Words

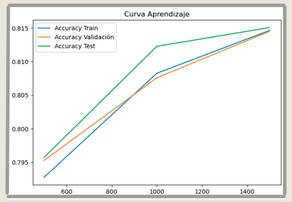
Usamos un predictor basado en **Naive Bayes Multinomial**, siguiendo el algoritmo ***Bag of Words***

Necesitamos extraer las palabras mas representativas, los **tokens** a analizar. Usaremos la función *CountVectorizer* y *TfidVectorizer*.

Hiper-parametros *vectorizer*

- **strip_accents**: ascii
- **analyzer**: word
- **max_features**: [500, 1000, 1500]
- **binary**: *True*

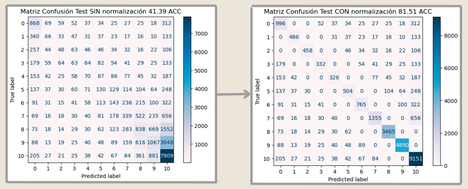
Realizamos 3 iteraciones distintas para saber cual es el mejor numero de palabras para extraer.



Se observan los siguientes datos:

- Con el aumento de atributos, aumenta también el acierto
- No es un porcentaje de cambio tan drástico, pero al ser datos grandes, la diferencia entre la primera aproximación y la ultima en cuanto al conjunto *Test*, es de un aumento de acierto de **551 review**

Resultados *NB Multinomial -> Normalizada*



La **normalización** se ha realizado por lo siguiente:

- Son valoraciones subjetivas por lo que no existe una distinción clara entre clases.
- Al ser 10 notas (clases) distintas, la diferencia entre valorar un juego como 8 o 9 es muy pequeña, por lo que consideraremos como correcta una predicción que se desvíe como mucho dos punto de la nota real.

De los **resultados** podemos decir lo siguiente:

- Existe un mayor porcentaje de acierto en los extremos.
- Por otro lado, aquellos que se encuentran en "puntos muertos", como valoraciones entre el 3 y el 7 devuelven una peor precisión. Podemos achacar esto a la complejidad subjetiva del problema.

Conclusión

KMeans

Tras haber entrenado los predictores previos creamos el definitivo, **recomendador de videojuegos**.

Hiper-parametros *SVM*

- **n_clusters**: len(listaJuegos)/2
- **init**: k-means++
- **algorithm**: lloyd

Características *X*:
añoLanzamiento / genero / portable / caracteristicasESRB / review / multijugador / online

Ejemplo a recomendar: [2002, 'fantasy', 1, 'T', 10, 1, 0]

Se utiliza un clasificador no supervisado ya que no nos interesa obtener un "porcentaje de acierto", sino tener juegos clasificados.

- Utilizamos la **mitad de clusters** de la **cantidad total de juegos**.
- Usamos un algoritmo como ***KMeans*** para poder obtener recomendaciones. Obtenemos el **cluster** al que más **cerca se encuentre** y damos los j**uegos que se encuentren en ese cluster** como **recomendaciones**.

- **Ambos valores han sido dados con los predictores anteriores.**

Recomendado:

	Nombre	Plataforma	Desarrolladora	Sinopsis
257	Spy Hunter	[PlayStation 2, GameCube, Xbox, Game Boy Advance, PlayStation Vita]	Paradigm Entertainment	Based on the arcade classic, Spy Hunter combines explosive combat, action-racing, and mission-based game play as you drive the G-115 Interceptor capable of morphing on the fly into a jet-powered speed boat, jet watercraft, or motorcycle to escape and evade enemy fire. [Mofwa]

Líneas Futuras

- Viendo el porcentaje de acierto obtenido en el **predictor ESRB**, necesitaríamos de alguna forma mejorar este número. Se podría realizar mediante la eliminación de algunas características o su unión en grupos. Por otro lado, hacer otro tipo de estandarización/normalización y/o utilizar otro tipo de reducción de dimensión.
- La clase **E10+** ha demostrado ser especialmente conflictiva, afectando al rendimiento de todo el sistema. Concluimos con el ya mencionado problema del mundo real, es una clase que de forma instintiva se solapa con las demás, produciendo fallos.
- Por último, en el **Bag of Words**, podríamos hacer un preprocesamiento distinto u otro tipo de predictor no basado en porcentajes para mejorar la precisión. Para ello se debería usar una menor cantidad de ejemplos para no realizar un sobre-aprendizaje.