WineRec
Entrega Final Proyecto
IIC 3633 - Sistemas Recomendadores

Paula Navarrete Campos & Astrid San Martín Jiménez
Departamento de Ciencias de la Computación
Facultad e Ingeniería
Pontifica Universidad Católica
pcnavarr@uc.cl, aesanmar@uc.cl

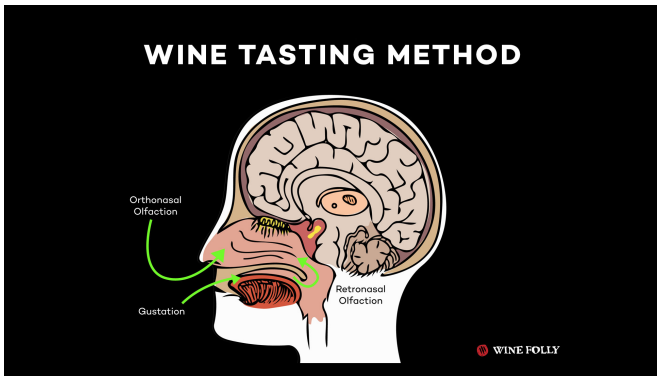
***Abstract*— Chile, as one of the leading wine producer countries, generates great interest in the national and international community. People have increasingly come closer to the wine market, which is expanding and lowered its elitized barriers. On-line sales have positioned each year with a greater presence. Many novice consumers are restlessness regarding wich wine to buy, what wine to try or which will be the wine that will fit their palate. To address these opportunities we decided to design a wine recommendation system, WineRec. To train our algorithm we use the data from the page *https://www.cellartracker.com/*. Our goal is to be able to serve users with a recommendation that will help them discern and make faster and better choices in the wine buying process.**

I. Introducción

En el rubro vitivinícola nos encontramos con una amplia variedad de características que definen las cualidades de un vino o licor, por ejemplo sus cepas, denominación de origen, viña productora, etc. Quienes se especializan como catadores, al evaluar un vino toman en cuenta una diversidad de características objetivas y percepciones organolépticas, las cuales califican según su experiencia y gusto desarrollado a través del tiempo. Junto con esto, una componente primordial en el contexto de valorar un vino, o tomar en cuenta sus recomendaciones, tiene que ver con su temporalidad, es decir, año de cosecha, antiguedad de la recomendación, incluso tiene que que ver con la antiguedad como catador.

Las recomendaciones en torno al rubro vitivinícola se enmarcan dentro de las tareas donde las preferencias del usuario son difíciles de predecir, por ejemplo algunos usuarios tienen preferencia por los sabores ácidos y valorarán estas características subjetivas por sobre las de otros tintos, mientras que otros discriminarán por región productiva. Con esto, vislumbramos que un sistema recomendador es un *must-have* para el comercio vitivinícola on-line. Buenas recomendaciones entimularán el comportamiento de compra de los usuarios. Basándonos en esto, apuntamos a desarrollar un modelo que nos permita hacer recomendaciones de vinos y licores a usuarios nuevos y antiguos.
Para un consumidor común, los parámetros que los llevan escoger un vino varían según la valoración personal que dan a ciertas características, algunos consumidores prefieren los vinos blancos, otros los ensamblajes, otros valoran aquellos que son orgánicos por sobre otros, etc. Así, cuando un consumidor poco experimentado ingresa al círculo vitivinícola, tomar la decisión sobre qué vino probar se hace dificíl, y probablemente tome en consideración la sugerencia de algún consumidor más ”experimentado” o de un usuario que considere ideosincráticamente similar a

ella o él.



Es aquí donde los sistemas recomendadores pueden entregar una buena solución a la hora de recomendar personalizadamente un vino o estimar la valoración que probablemente le atribuirá un consumidor no tan docto en la materia.
Al rededor del mundo existen sitios web dedicados a la revisión de vinos con una amplia gamma de variedades, con comentarios de expertos catadores y/o de usuarios comúnes. Con toda esta información disponible, encontramos los ingredientes perfectos para poder entregar un servicio de utilidad con recomendaciones basadas en lo que conocemos sobre los gustos de los consumidores y usuarios de dichas páginas.

La estructura de este documento es la siguiente, en la sección II presentamos un breve estado del arte y en la sección III exponemos la solución al problema trabajado. En la Sección IV detallamos del dataset utilizado. En la sección V comentamos sobre la metodología utilizada y en la sección VI proferimos un análisis de los parámetros. En la sección VII se exponen los resultados de la solución y metodología propuesta. Finalmente en la sección VIII explicamos nuestras conclusiones y trabajo futuro.

II. Estado del Arte

Las técnicas de Machine Learning detectan patrones en los datos y usan esos patrones para predecir resultados futuros, ayudando así a la toma de decisiones bajo incertidumbre [2]. Por lo tanto, del mismo modo que los expertos, al confiar en sus observaciones de correlaciones entre experiencias pasadas y sus resultados subsecuentes, desarrollan y adaptan de forma colectiva reglas simples y únicas para dirigir la toma de decisiones, también lo hacen las reglas supervisadas de Machine Learning, basadas en el reconocimiento de patrones entre multiples variables y su variable de respuesta [1].

Por lo tanto, al igual que el desarrollo de heurísticas individuales (ej. los vinos del valle de Casablanca generalmente son mas ácidos) puede ser modelado como un proceso de actualización Bayesiana, es decir, observaciones recurrentes de ciertos eventos dados, que preceden a la ocurrencia específica de un outcome, son 'aprendidos' como un eventos predictivos, las técnicas de machine Learning realizan iteraciones que prueban relaciones potenciales entre parámetros y un resultado, 'aprendiendo' qué parámetros son predictores más consistentes del resultado especificado.

El uso de sistemas recomendadores para entregar recomendación de productos o predecir cuál será la valoración de un usuario a cierto item de acuerdo a con las calificaciones entregadas a distintos items o respecto del comportamiento de usuarios similares, es uno de los problemas que podemos encontrar en ésta área [6]. El objetivo de obtener el perfil de un usuario, se puede abordar de dos formas: ya sea que el usuario entregue explicitamente la información ó reunir información de manera implicita que se relacione con el usuario.
Tenemos según Burke et al. 2007b [7] seis tipos diferentes de enfoques para recomendacion:

- Recomendación basada en contenidos: sugerir items basados en las preferencias historicas del usuario.
- Filtrado colaborativo: recomendación basada en usuarios que presentaron gustos similares.
- Recomendación basada en conocimiento: se recomiendan items basados en el conocimiento de un área específica sobre cierta carácteristica que aborda las necesidades y preferencias del usuario, un sentido de utilidad para el usuario.
- Demogáfico
- Recomendación basada en la comunidad
- Sitema híbrido de recomendación

Los sistemas de recomendación son una de las aplicaciones más exitosas y extendidas de las tecnologías de Machine Learning en negocios y proveen sugerencias a un usuario para apoyar su toma de decisiones. En particular, pueden ayudar a determinar si un usuario estaría interesado en adquirir un *vino o licor* en particular o no. Los algoritmos de Machine Learning en sistemas de recomendación se clasifican generalmente en dos categorías: métodos de filtrado basados en contenido (*content based*) y colaboración (*collaborative filtering*), aunque los recomendadores modernos combinan ambos

enfoques [3]. Los métodos basados en el contenido se basan en la similitud de los atributos de los itemes y los métodos colaborativos calculan la similitud de las interacciones delos usuarios con los itemes.

Los métodos de *collaborative filtering* recopilan calificaciones de itemes de muchas personas y utilizan técnicas de *Nearest Neighbor* para hacer recomendaciones a un usuario sobre nuevos itemes. Sin embargo, estos no toman en cuenta la cantidad significativa información que generalmente está disponible sobre la naturaleza de cada item ni su temporalidad, dejando irresuelta la pregunta de qué rol puede jugar el contenido en el proceso de recomendación. Por el contrario, métodos del tipo *content-based* aceptan información que describe la naturaleza de un ítem, y basadose en una muestra de las preferencias del usuario, aprenden a predecir qué elementos gustarán al usuario. Ambos se pueden considerar problemas de aprendizaje cuyo objetivo es aprender una función que pueda tomar una descripción de un usuario y un item y predecir las preferencias del usuario con respecto a ese item.

III. Solución al problema trabajado

El objetivo de este trabajo es desarrollar un recomendador que pueda explotar tanto los ratings como la información del contenido de los items. A diferencia del enfoque tradicional de *collaborative filtering*, enmarcamos el problema como uno de calificación de itemes ayudándonos de la calificación de estos. Por otro lado, diferimos de los métodos *content based*, en que utlizaremos la información social, en forma de calificaciones de otros usuarios, en el proceso de aprendizaje inductivo. En particular, formalizaremos el problema de recomendar un *vino* como un problema de aprendizaje; específicamente, el problema de aprender una función f que toma como entrada a un $usuario$ y un $vino$ y produce como resultado un score que indica si el *vino* sería de su agrado (y por lo tanto lo comprará) o no, es decir:

$$f(\langle user, wine \rangle) \longrightarrow \{score\} \tag{1}$$

IV. Dataset

A. Obtención del Dataset

Usamos web scraping para extraer datos del sitio web *CellarTracker.com* a través de un proceso automatizado. CellarTracker fue creado en marzo de 2003 por Eric LeVine [1] como una forma de mantener un registro de su propia bodega mientras estaba trabajando en Microsoft, lanzando públicamente el sitio en abril de 2004. Hoy en día CellarTracker es la principal herramienta de gestión de bodegas personales con cientos de miles de coleccionistas que rastrean más de 75 millones de botellas. CellarTracker también se ha convertido en la base de datos más grande de notas de cata con más de 5.8 millones de notas registradas a fin de 2016. Anualmente millones de entusiastas del vino visitan el sitio para leer reseñas y obtener recomendaciones de vinos. Aún así, CellarTraker al día de hoy no presta ningún tipo de inteligencia que le permita recomendar personalizadamente un vino a algún usuario, solo es capaz de entregar estadísticas básicas para cada vino (rating promedio, y promedio like/dislike). El panorama local es menos prometedor, los sitios chilenos de venta de vino online son muy rudimentarios, disponen un sistema simple de reviews y no cuentan con una cantidad de reviews significativa.

B. Dataset

Para obtener el dataSet recopilamos datos de vinos y licores para las regiones de aproximadamente 80 paises[2]. Para cada país recolectamos las primeras 5 páginas de resultados con información de cada item (aprox 125 items) para las regiones de cada país[3]. También recopilamos los reviews de la comunidad activa para cada uno de los vinos (items) recolectados, Al momento, hemos recopilado la información señanada para items en un rango de precios de 20 a 40 dólares la botella[4].

En una primera etapa contamos con dos archivos, uno que contiene la información relativa a cada item o *vino* con la url que aloja los reviews hechos a ese item y otro que contiene los reviews de los usuarios a cada vino de la lista, ambos fueron unificados dando origen a un archivo de reviews que contiene cada uno de los reviews añadida la información relativa al item que genera el review. A la fecha contamos con la infomación de aproximadamente 18,300 items. Luego de eliminar los registros inconsistentes, replicados y

[1] https://www.cellartracker.com/content.asp?iContent=3

[2] Recopilamos la información para los paises donde CellarTraker tiene información.

[3] Recolectamos las 5 primeras páginas por restricciones de tiempo. La base de datos es extensa y para ciertas regiones que se destacan por su producción vitivinícola, muchas paginas quedaron sin ser consideradas en este dataset.

[4] Uno de los contratiempos a los que nos vimos expuestas fue la politica que cellartracker de bloquear IP que sospecha están realizando scrapping de sus datos, lo cual ralentizó en gran medida el proceso de obtención del dataset.

que cuentan con toda la información necesaria, el set completo se redujo a 15748 reviews. que luego fue dividido en un 70% para training y 30% para testing.

1) Items: Para cada vino o licor (*item*) contamos con la información (tabla I) referente a la puntuación otorgada por los usuarios (puntuación de 1 a 100), url que aloja los reviews del vino, su cosecha, tipificación, productor, cepas utilizadas, designación, viña, país, region, subregión y denominación de origen.

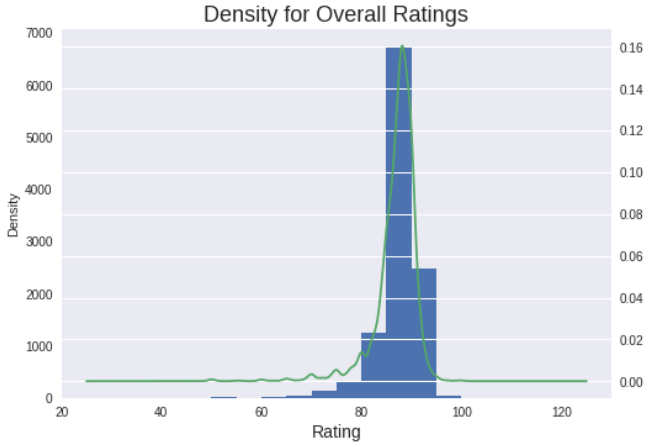
La Tabla II ilustra la estructura mencionada con valores reales.

2) Reviews: Recopilamos información relativa a los reviews disponibles para los items (tabla III) como identificador del usuario, comentarios, fecha y score. La Tabla IV muestra un ejemplo con información real de la estructura mencionada.

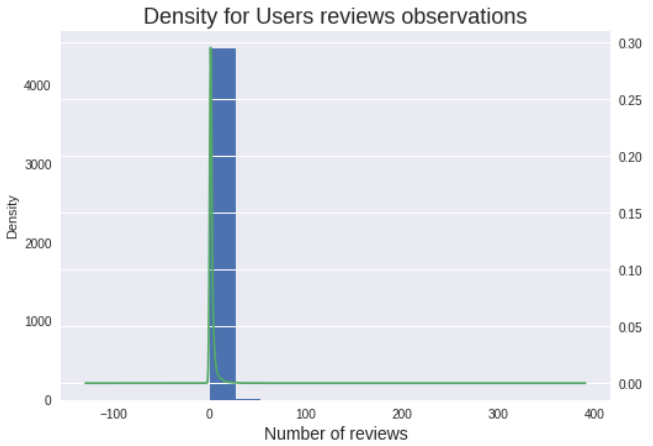
3) Users: Recopilamos información relativa a los usuarios(tabla V) como identificador del usuario, experiencia, comentarios y puntunción de cada review hecho.

La Tabla VI ilustra la estructura mencionada.

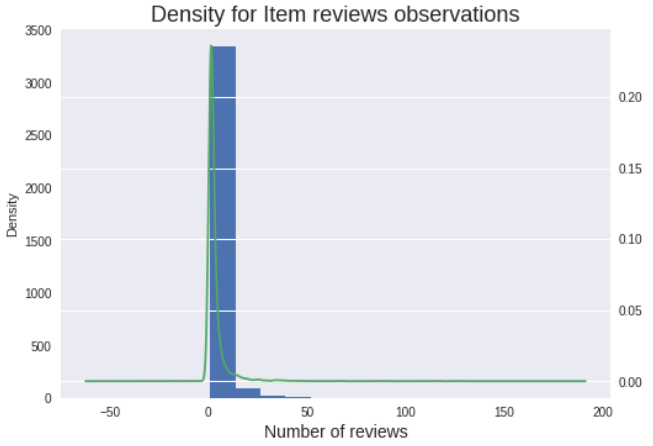
La tabla VII ilustra la cantidad de diferentes valores para cada una de las variables recopiladas. En la siguiente figura podemos apreciar la distribución total de los ratings, cuya media es de 87 puntos, con una desviación estándar de 4 puntos.



En promedio, los usuarios emiten 2 reviews, con una desviación estándar de 5, siendo el mínimo un review y un máximo de 261 reviews. La siguiente figura ilustra la función de densidad asociada a la distribución de ratings por usuario.



Respecto de los items (vinos), éstos obtienen un promedio de 3 reviews, con una desciación estándar de 6, siendo el mínimo un review y un máximo de 128 reviews. La siguiente figura ilustra la función de densidad asociada a la distribución de ratings por item.



V. Metodología Utilizada

Para modelar el proceso en el que los individuos desarrollan reglas de decisión utilizaremos un sistema recomendador, una técnica de aprendizaje supervizado que utiliza un set pre-clasificado de observaciones (resultados de interés) como un set de entrenamiento, que genera recomendaciones correlacionadas con un resultado de *performance* de interés que es este caso es su calificación. Con este set de entrenamiento, el sistema computa interralaciones entre usuarios e itemes para clasificarlos, maximizando la capacidad de un conjunto de reglas iniciales (o features) para predecir la clasificación correcta de los resultados.

En este caso, los datos de entrada para el sistema recomendador preceden en el tiempo a los datos de resultado, lo que se asemeja a la forma en que los individuos infieren causalidad a través de un proceso de actualización bayesiano. Además, de forma consistente

Información	Descripción
Score	puntuación de 1 a 100.
Reviews	url que aloja todos los reviews del item.
Vintage	año de producción.
Type	tipificación del vino, Ej: rojo, blanco, rosé, etc.
Producer	nombre del productor (quien hizo el vino).
Variety	cepa(s) utilizadas para producir el vino. Ej: Cabernet, Syrah, merlot, etc.
Designation	designación dentro de la viña de donde provienen las uvas que elaboraron el vino.
Vineyard	viña donde se produjo el vino.
Country	país de producción.
Region	indica de dónde se obtuvieron las uvas para producir el vino.
Subregion	area específica dentro de una Región de dónde se obtuvieron las uvas para producir el vino.
Appellation	denominación de origen. Indica el valle de dónde se obtuvieron las uvas para producir el vino.

TABLE I: Información para cada vino o licor (*item*).

Score	Reviews	Vintage	Type	Producer	Variety	Designation	Vineyard	Country	Region	Subregion	Appellation	Pricelevel
88	https://www.cellartracker.com/notes.asp?iWine=858606	2006	Red	'Tilda	Syrah Blend	Petulance	n/a	USA	Washington	Columbia Valley	Columbia Valley	1
	https://www.cellartracker.com/editnote.asp?iWine=2638585	2016	Rosé - Sparkling	14 Hands	Champagne Blend	n/a	n/a	USA	Washington	Columbia Valley	Yakima Valley	1
75.5	https://www.cellartracker.com/notes.asp?iWine=2630939	2015	Red	100 MILE VINEYARD	Merlot	n/a	n/a	USA	California	Central Valley	Lodi	1
86	https://www.cellartracker.com/notes.asp?iWine=2914131	2016	Red	100 MILE VINEYARD	Zinfandel	n/a	n/a	USA	California	Central Valley	Lodi	1
	https://www.cellartracker.com/notes.asp?iWine=2987125	2016	Red	1000 Stories	Carignan	Bourbon Barrel Aged	n/a	USA	California	North Coast	Mendocino County	1
	https://www.cellartracker.com/editnote.asp?iWine=3008228	2016	Red	1000 Stories	Red Blend	Gold Rush Red Bourbon Barrel Aged	n/a	USA	California	n/a	California	1
87.7	https://www.cellartracker.com/notes.asp?iWine=726338	2007	White	12 Mile Trail	Chardonnay	n/a	merryvale	USA	California	Napa Valley	St. Helena	1

TABLE II: Item Dataset Structure

Información	Descripción
UserID	Identificador del usuario.
Review	texto con el comentario con la experiencia de cata del vino.
Web_Page_URL	url que aloja el comentario.
Timestamp	fecha en que fue referido el comentario.
Score	puntuación que el usuario ha dado al vino.

TABLE III: Información relativa a los reviews (*item*).

UserID	Review	Web_Page_URL	Score
Dukeies21	Very good	https://www.cellartracker.com/notes.asp?iWine=1903563	97
dssinger	Very pleasant!	https://www.cellartracker.com/notes.asp?iWine=2022206	87
Villa D	Not bad for a twist off cap. Robert Rex's winery does a very good job of offering "clean wines" (little as possible sulfites) which my wife likes because of no after drinking. Still, a very good every day wine.	https://www.cellartracker.com/notes.asp?iWine=2022206	89
ellahazard	Yummy and inexpensive, good gift?Tastes more expensive than it is...	https://www.cellartracker.com/notes.asp?iWine=2022206	
cnr128	Love those Central Coast Syrahs with the peppery berry thing going on...in this case in a Rhone (SMG) blend. Like this quite a bit, especially for the price.	https://www.cellartracker.com/notes.asp?iWine=773039	86

TABLE IV: Reviews Dataset Structure

Información	Descripción	Para c/review de usuario	
User_ID	Identificador del usuario.	**Review**	texto con el comentario con la experiencia de cata.
Intake_qty	cantidad de botellas consumidas.	**Score**	puntuación que el usuario sobre esa experiencia.
Reviews_qty	cantidad de reviews del usuario.		

TABLE V: Información relativa a los usuarios.

con la que los individuos descifran interralaciones entre variables predictivas y su respuesta asociada. El modelo de aprendizaje elegido es una *factorization machine* [4].

Utilizamos *one-hot econding* para binarizar las vari-

User_ID	tenure	Reviews_qty	Reviews
Dukeies21	459	5	$reviews = \{(\text{"Awful .."}, 75), \ldots, (\text{"this..."}, 89)\}$
dssinger	874	384	$reviews = \{(\text{"Very .."}, 87), \ldots, (\text{"Super ..."}, 76)\}$
Villa D	103	23	$reviews = \{(\text{"The .."}, 75), \ldots, (\text{"At th..."}, 89)\}$
ellahazard	85	58	$reviews = \{(\text{"Fruity .."}, 74), \ldots, (\text{"Tastes ..."}, 84)\}$

TABLE VI: User Dataset Structure

	# instancias
users	4486
items	3469
vintage	24
type	15
producer	1022
variety	244
designation	1142
vineyard	106
country	4
region	88
subregion	46
appellation	338
ratings	44

TABLE VII: Cantidad de valores distintos para cada variable.

ables categóricas mencionadas en la sección anterior, en particular, generamos una variable dummy por cada valor distinto en cada una de las categorías mencionadas en la sección anterior.

1) Métodos: Aplicaremos varios algoritmos para luego compararlos en cuanto a su resultados. El primer método escogido para la realización de éste proyecto que es el de *Factorization Machines* [4]. Aquí usamos parámetros factorizados, con lo cual podemos conocer la interacción, aún cuando nos encontramos con poca información (*"sparse data"*), escenario donde escala en $O(kn)$

Existen distintas tareas de predicción que podemos realizar: regresión, clasificador binario y ranking [4]. Según nuestros objetivos fijados usaremos *factorization machine* con regresion, es decir entregar la puntuación que el usuario entregaria.

Como puntos de comparación implementamos los siguientes algoritmos disponibles en la libreria *pyReclab* [13]:

- UserKnn
- ItemKnn
- SlopeOne
- SVD

2) Evaluación: Para evaluar el performance de cada uno de los métodos calculamos dos métricas para cada uno:

- Mean Absolute Error (MAE)[8]
- Mean Squared Error (RMSE) [9]
- Tiempo de procesamiento

VI. Implementación y Análisis de Parámetros

En primera instancia, separamos nuestro set de datos en un 70% para set de entrenamiento y 30% para conformar un set de testeo, utilizamos la libreria *train_test_split* de *scikit-learn* [14] para este fin.

Manipulamos los datos para obtener la respresentación ilustrada en la tabla VIII implementamos one-of-K or one-hot coding para binarizar las variables categóricas a través de la librería *OneHotEncoder* de la misma librería.

Para cada uno de los experimentos variamos los parámetros que se señalan a continuación para determinal el modelo que mejor se ajusta:

1) **UserKnn:** variamos el numero de vecinos de 2 a 15 incrementando de uno a la vez.
2) **ItemKnn:** de la misma forma, variamos el numero de vecinos de 2 a 15 incrementando de uno a la vez.
3) **SlopeOne:** Este método no requiere el seteo de parámetros en particular para mejorar el performance.
4) **SVD:** Cambiaremos dos parametros en este modelo, el número de factores y el número de iteraciones. Incrementaremos el número de factores de 10 a 100 y 1000 y variamos el numero de iteraciones de 100 a 200. Realizamos todas las combinaciones entre estos valores para elegir aquellos que mejor ajustan el modelo.

Evaluamos el tiempo de procesamiento requerido por cada algoritmo. Y así obtener una evaluación sobre qué estrategia resulta con un mejor balance entre tiempo de entrenamiento y mejor predicción.

1) Implementación: La implementación del modelo se realizó con fastFM [5], una libreria de *factorization machine* en *Python*, usando de base *Tensorflow.* Una ventaja de usar esta libreria es que pudimos usar una implementación especial en GPU, y asi aprovechar en su totalidad la capacidad que nos entrega *Google Collaboratory.*. La tabla VIII ilustra un bosquejo de la estructura de datos propuesta como input. En el github del proyecto dejamos está disponible la implementación de todos los experimentos.

VII. Resultados Obtenidos

Primero que todo, es importante destacar que en el proceso de desarrollo nos dimos cuenta que no es trivial la elección de features predictivas con las que alimentar la factorization machine, por la estructura del algoritmo [4] es importantísimo normalizarlas o binarizarlas ya que de otra forma, las métricas se tornan incomparables con nuestro baseline. Las unidades del RMSE corresponden a la unidad de la variable dependiente, no así por ejemplo con el R^2 por ejemplo, ya que es una proporción. Por lo tanto, numéricamente, RMSE puede cambiar arbitrariamente (cambiando las unidades de medida de las variables predictivas, por ejemplo de gramos a kilos) mientras se mantiene el R^2 constante, cambiando la unidad de la variable dependiente, los números serán diferentes pero el significado es exactamente el mismo.

De hecho, las métricas basadas en errores como RMSE, MAE, etc., proporcionan la imagen real de la calidad de la predicción. Sin embargo, decidir un valor de umbral adecuado para estas métricas es realmente problemático. Por ejemplo, los valores altos de RMSE pueden deberse a la presencia de un pequeño número de predicciones de error altas (como se ve en los outliers).

Con esto, calculamos de todas formas el R^2 a asociado a la factorization machine, y vemos que a medida que aumentan las iteraciones éste mejora, pero RMSE y MAE aumentan. Creemos que esto es atribuible a lo reducido del dataset ya que al binarizar las features para alimentar la factorization machine, terminamos con una cantidad de features muy cercana al núemro de observaciones, lo que puede traer problemas y arrojar valores negativos de R^2.

Un R^2 negativo suele darse cuando la suma de cuadrados residual se acerca a la suma de cuadrados, lo que significa que la explicación de la respuesta es muy baja o insignificante. Entonces, el R^2 ajustado negativo traduce la insignificancia de las variables explicativas. Los resultados pueden mejorarse con el aumento del tamaño de la muestra. Las figuras 1 y 2 dan cuenta de esta situación.

La tabla IX expone los valores para los mejores RMSE, MAE y tiempos asociados a cada uno de los experimentos llevados a cabo. Vemos que en general utilizar en este problema una Factorization Machine no se traduce en una mejora significativa en términos de alguna de las métricas señaladas, pero aún así se mantiene competitiva con respecto a los otros algoritmos y tal vez podria mejorar al aumentar el tamaño del dataset. Las figuras 3, 4 y 5 a continuación muestran la tendecia para el RMSE y MAE para ItemKnn, UserKnn, SlopeOne y SVD variando los parámetros mencionados en la sección anterior.

VIII. Conclusiones

Observamos y experimentamos la importancia de poseer un buen y completo set de datos para poder realizar la tarea de recomendación. Incorporar features predictivas como tenure y cantidad de reviews anteriores, que reflejan la experiencia en términos del tiempo y como catador mejora el ajuste del modelo de maquinas de factorización. Observamos que la elección no tomar en cuenta las recomendaciones de binarización y normalización de las variables predictivas genera desajustes al modelo empobreciéndolo en su capacidad predictiva. El transformar la variable de timestamp a tenure mejoró considerablemente el modelo (RMSE y MAE). Vemos que el tiempo asociado a entrenar una factorization machine es un factor importante a tener en cuenta a la hora de elegir un modelo ya que sobrepasa con creces al del resto de los experimentos. Esto podría tener un revés al contar con un set de datos más cercano al real donde el sparsity se tangibiliza y los experimentos de baseline tienden a ser mas pobres.

Trabajo Futuro

Es interesante evaluar el comportamiento del modelo de factorization machines con un set de datos ampliado. Como trabajo futuro pretendemos seguir corriendo el scrapping de datos, mejorando el algoritmo de web scrapping, incorporando *Tor*, para no ser detectadas por los mecanismos antiscrapping de *CellarTracker*. También proponemos incorporar un sentiment analysis sobre el texto asociado a los reviews e incorporarlo como una feature predictiva y analizar si genera mejoras al modelo.

Por último, sería interesante evaluar qué tan predictivo es es modelo para usuarios segmentados por

	User				Wine				Other wines rated				Last wine rated				Score	
x_1	1	0	0	...	1	0	0	...	0.3	0	0.3	...	0	0	0	...	66	$\rightarrow y_1$
x_2	1	0	0	...	0	0	1	...	0.3	0	0.3	...	1	0	0	...	89	$\rightarrow y_2$
x_3	0	1	0	...	0	0	1	...	0	0.5	0.5	...	0	0	0	...	72	$\rightarrow y_3$
x_4	0	1	0	...	0	1	0	...	0	0.5	0.5	...	0	0	0	...	70	$\rightarrow y_3$
x_5	0	0	1	...	1	0	0	...	0.5	0	0.5	...	0	0	0	...	92	$\rightarrow y_3$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_n	0	0	1	...	0	0	1	...	0.5	0	0.5	...	0	0	1	...	68	$\rightarrow y_n$
	$\uparrow$	$\uparrow$	$\uparrow$		$\uparrow$	$\uparrow$	$\uparrow$		$\uparrow$	$\uparrow$	$\uparrow$		$\uparrow$	$\uparrow$	$\uparrow$			
	$Anna$	Bob	$Mary$	...	w^1	w^2	w^3	...	w^1	w^2	w^3	...	w^1	w^2	w^3			

TABLE VIII: Data Structure

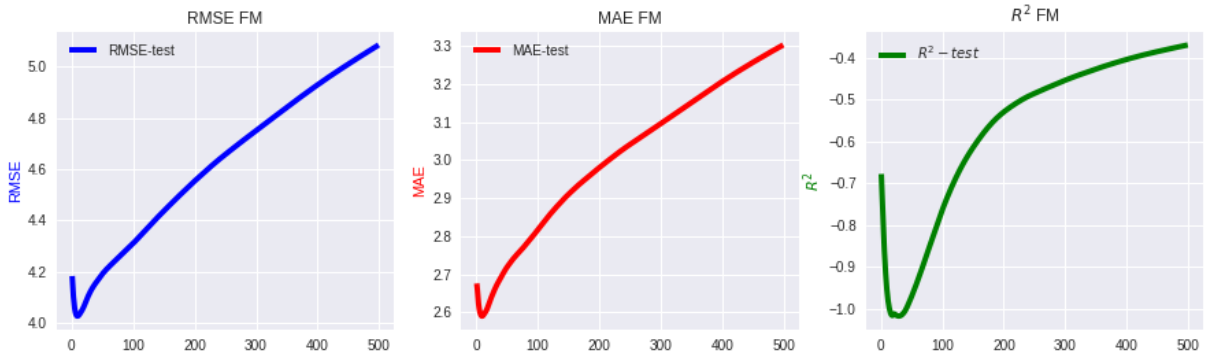


Fig. 1: RMSE, MAE y R^2 para FM con 500 iteraciones. Tendencia general.

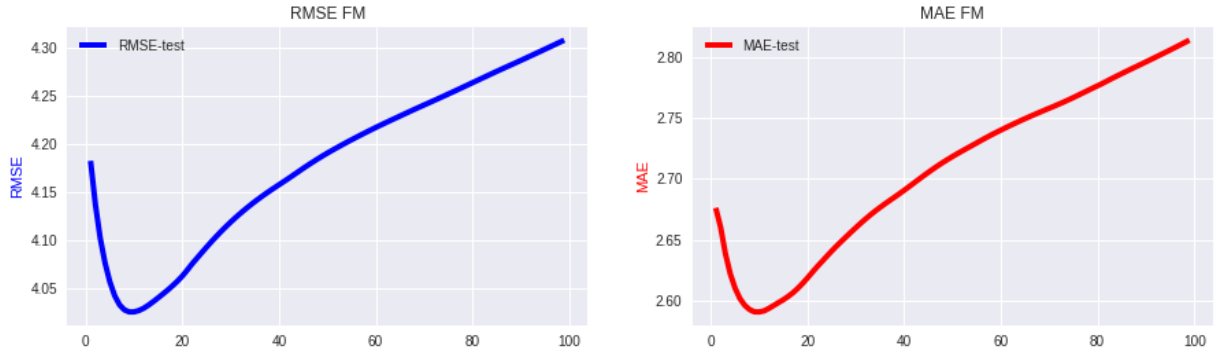


Fig. 2: RMSE, MAE para FM zoom

	RMSE	MAE	Time (segs)
ItemKnn	3.450	2.428	0.051
UserKnn	3.534	2.473	0.066
SlopeOne	56.731	40.161	0.001
SVD	3.000	1.993	0.021
FM	4.026	2.590	3.482

TABLE IX: Resultados obtenidos

zona geográfica, y analizar la predictivilidad para zonas específicas como Chile. O bien analizar cómo puede ser esto una herramienta de utilidad para ayudar a la toma de decisiones respecto de exportaciones de la producción local y fomentar el crecimiento de la industria vitivinícola nacional.

REFERENCES

[1] Leatherbee, M., del Sol, P. (2016). Predicting Entrepreneurial Performance: Simple Rules versus Expert Judgment. Working Paper. Available at: http://ctie.economia.cl/wp-content/uploads/2017/07/Predicting-Entrepreneurial-Performance-Si mple-Rules-2016. pdf (accessed May 2017).

[2] Murphy, K. (2012). Machine learning: a probabilistic approach. Massachusetts Institute of Technology, 1-21.

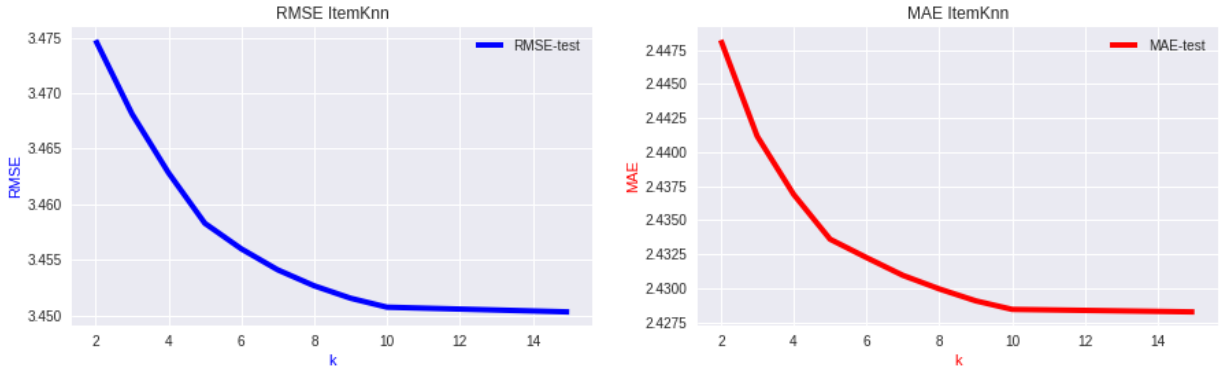


Fig. 3: Resultados para ItemKnn

[3] Schafer, J. B., Frankowski, D., Herlocker, J., Sen, S. (2007). Collaborative filtering recommender systems. In The adaptive web (pp. 291-324). Springer Berlin Heidelberg.

[4] Rendle, S. (2010). Factorization machines. In Data Mining (ICDM), 2010 IEEE 10th International Conference on (pp. 995-1000). IEEE.

[5] Bayer, I. "fastFM: A Library for Factorization Machines" Journal of Machine Learning Research 17, pp. 1-5 (2016)

[6] Nilashi, M., Bagherifard, K., Ibrahim, O., Alizadeh, H. Collaborative Filtering Recommender Systems. Research Journal of Applied Sciences, Engineering and Technology 5(16): 4168-4182, 2013.

[7] Burke, R.D., 2007b. Hybrid web recommender systems. Lect. Notes Comput. Sc., 4321: 377-408.

[8] Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: 14th Conference on Uncertainty in Artificial Intelligence, pp. 43–52 (1998)

[9] Shardanand, U., Maes, P.: Social information filtering: algorithms for automating word of mouth. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1995, pp. 210–217. ACM Press/Addison-Wesley Publishing Co., New York (1995)

[10] Manning, C.D., Raghavan, P., Schtze, H.: Introduction to

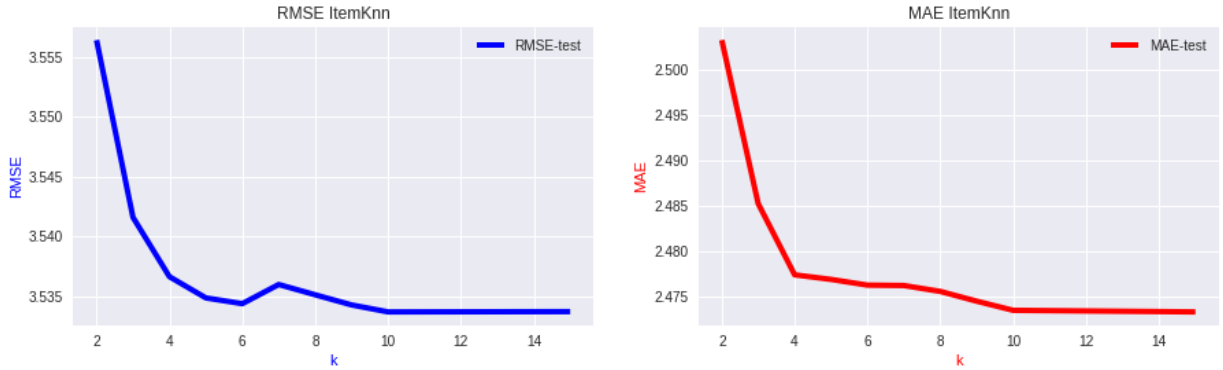


Fig. 4: Resultados para UserKnn

Information Retrieval. Cambridge University Press, New York (2008)

[11] J'arvelin, K., Kek'al'ainen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. 20, 422–446 (2002)

[12] Ziegler, C.-N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th International Conference on World Wide Web, WWW 2005, pp. 22–32. ACM, New York (2005)

[13] Sepulveda, G., Parra, D. (2017). pyRecLab: A Software Library for Quick Prototyping of Recommender Systems. arXiv preprint arXiv: 1706. 06291 (2017).

[14] Pedregosa, F. Varoquaux, G. and Gramfort, A. Michel, V.et al.(2011) Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. vol. 12. pp.2825-2830.

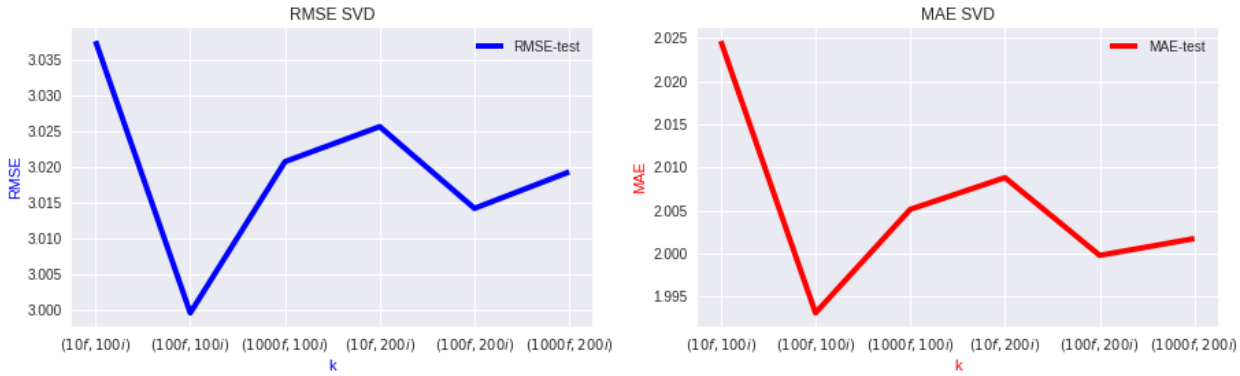


Fig. 5: Resultados para SVD