Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
Computación: Ciencia y Tecnología del Mundo Digital

Tarea 2

Profesor Denis Parra
10 de mayo de 2015

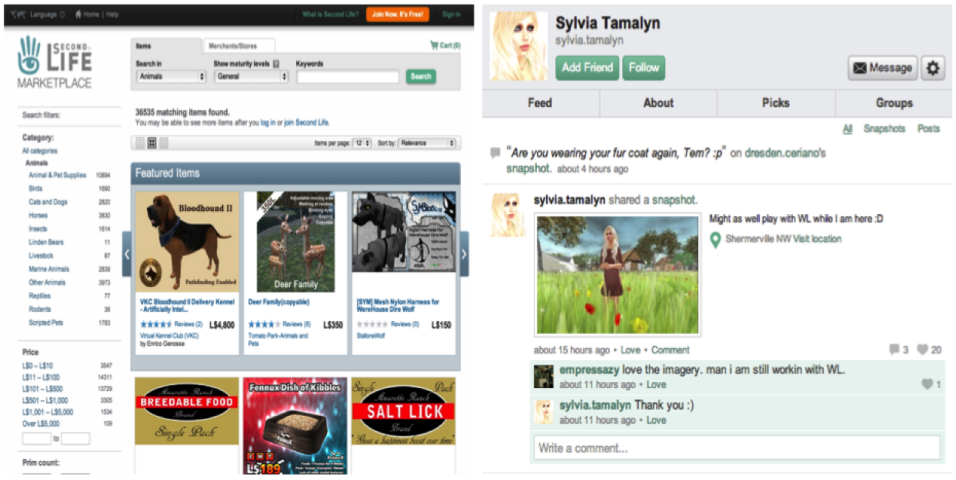
Indicaciones

- Fecha de Entrega: 10 de mayo de 2014
- Debes entregar la tarea en el repositorio git privado asignado al grupo.
- Cada hora de atraso descuenta 5 décimas de la nota que obtengas.
- La tarea es en pareja. La copia será sancionada con una nota 1.1 en el la tarea, además de las sanciones disciplinarias correspondientes.

Objetivo

El objetivo de esta tarea es que aprendas a:

- Hacer consultas básicas en una base de datos en SQL a través de Python.
- Filtrar la información relevante de la base de datos, y almacenarla en un archivo csv.
- Utilizar una librería en Python para realizar tareas de clasificación (minería de datos).
- Evaluar distintos "features" para seleccionar el modelo más apropiado para la tarea de clasificación.
- Utilizar git para trabajar concurrentemente en un grupo de trabajo.



(a) SecondLife store (b) SecondLife social stream

Fig. 1. Examples for a store in the marketplace and a social stream of an user in the online social network of the virtual world SecondLife.

Figure 1: Imágenes de Second Life

Parte I: Base de Datos

Instrucciones

Vas a trabajar sobre un dataset de Second Life. El dataset contiene información de los perfiles (avatares) de usuarios de Second Life. Todos los avatares participan en el *marketplace*, además pueden tener publicaciones en su *feed*, dejar *reviews* de productos de las tiendas, comentar *reviews*.

Información que debes obtener de SQL

Antes de trabajar con el dataset, debes conocerlo, por lo que tu programa debe imprimir en consola la siguiente información:

1. Número de usuarios
2. Número de tiendas
3. Número de categorías de productos
4. Número promedio de :
    - *Reviews* por producto
    - *Comments* por producto

- *Loves* por usuario
- *Comments* por usuario
- *Wallpost* por usuario
- *Snapshot* por usuario

Archivo que debes generar

Debes generar un archivo con la siguiente información **por usuario**:

1. Algún identificador único para los usuarios (i.e. Avatar)
2. Cantidad de:
    - *Loves*
    - *Comments*
    - *Wallpost*
    - *Snapshot*
3. Average *subjectivity* y desviación estándar de *subjectivity* de:
    - Los status del usuario en la red social[1]
    - Los comentarios del usuario en la red social
    - Los reviews del usuario en el marketplace
    - Los comentarios del usuario en el marketplace
4. Average *polarity* y desviación estándar de *polarity* de:
    - Los status del usuario en la red social
    - Los comentarios del usuario en la red social
    - Los reviews de cadal en el marketplace
    - Los comentarios del usuario en el marketplace

Por ejemplo, una línea del archivo podría ser:

```
"aang";10;45;37;86;0.3;0.199;0.76;0.52;0.07;0;0.01;0.04;-0.41;0.3;0.24;
   -0.86;0.12;0.9;0.05;0
```

[1] Feed.

Sentiment Analysis

Para calcular *subjectivity* y *polarity* puedes usar esta librería.

Subjectivity

Subjectivity es un valor que indica el nivel de subjetividad de un conjunto de palabras (i.e. una oración), según los adjetivos que contenga. Se mide como un número entre 0 y 1, donde 0 significa que la oración es muy objetiva y 1 indica que es muy subjetiva.

Polarity

Polarity mide la polaridad de una oración. Una oración se considera de polaridad positiva (de valor más cercano a 1), si los términos que aparecen en la oración evocan algo positivo[2]. En el caso contrario, una oración se asocia a una polaridad negativa[3].

Entrega

Debes entregar un archivo de código en Python (.py), en el repositorio git, junto con un Readme, que explique como ejecutar tu código. **No debes entregar** los archivos que tu programa genere.

Esquema de la base de datos

products	
product_id	int(11)
product_name	varchar(128)
product_link	varchar(256)
product_price	int(8)
product_description	varchar(1024
product_rating	float
product_ratings	int(11)
store_id	int(8)

categories	
category	varchar(128)
store_id	int(11)
count	int(11)

comment_stores	
comment_id	int(11)
store_id	int(11)

[2] Por ejemplo: "C es un lenguaje simple y bueno".
[3] "C++ es tedioso y difícil."

about	
Campo	Tipo
id	int(11)
avatar	varchar(256)
uuid	varchar(36)
birthday	varchar(256)
payment	varchar(36)
biography	varchar(2048)
real_biography	varchar(2048)
facebook	varchar(128)
twitter	varchar(128)
youtube	varchar(128)
plurk	varchar(128)
linkedin	varchar(128)
flickr	varchar(128)
homepage	varchar(256)
real_life_image	varchar(256)
avatar_image	varchar(256)
partner	varchar(128)
feed	varchar(3)

store	
store_id	int(11)
store_name	varchar(128)
store_location	varchar(256)
owner_name	varchar(128)
number_of_items	int(11)
join_date	varchar(64)

feed	
id	int(11)
source	varchar(256)
destination	varchar(256)
type	varchar(36)
data	varchar(2048)
time	datetime

groups	
id	int(11)
avatar	varchar(36)
groupUUID	varchar(36)

reviews	
id	int(11)
product_id	int(11)
name	varchar(128
date	date
rating	int(11)
comments	int(11)
review_id	int(11)
review	text

comments	
id	int(11)
rid	int(11)
name	varchar(128)
date	date
comment	text

customers	
name	varchar(120)

sellers	
name	varchar(128)

Indicaciones

- El dataset está en MySQL, por lo que debes instalar MySQL en tu computador para poder consultarlo. En la ayudantía del miércoles 22 de abril, se verá como instalar MySQL[4].

[4]Aunque si quieres saber antes, puedes envíarle un mail a los ayudantes

- Para poder consultar la base de datos con Python, te recomendamos usar esta librería.

Parte II: Data Mining

Instrucciones

Para esta parte de la tarea debes trabajar con el databaset que generaste en la parte anterior y agregar 2 nuevos features que serán explicados más abajo. Debes usar la librería de python **scikit-learn** para crear modelos usando regresión logística y con eso predecir si un usuario es un vendedor dentro de la red.
La regresión logística nos permitirá conocer la probabilidad de ser o no vendedor de un usuario, y puede expresarse como:

$$logit(P(vendedor_{sb} = SI)) = \beta_0 + X\beta + g_u \tag{1}$$

$$logit(p) = \log(\frac{p}{1-p}) \tag{2}$$

Donde X es un vector de variables que llamaremos *features* y β es un vector de coeficientes constantes asociados a cada *feature*. Junto con tu código, debes entregar un informe, el cual está detallado más adelante.

Set de Datos

El set de datos corresponde a un grafo que representa a una muestra de la red de usuarios de Second Life (SL). Para esta actividad deberás considerar 16 variables agrupadas en tres categorías. Las columnas (*features*) con las que vas a trabajar están a continuación.

El objetivo es predecir si usuario es o no un vendedor. Para predecir esto, haremos uso de *features* calculados para cada usuario. Todos estos features fueron calculados en la parte anterior, excepto el in y out degree. El in degree de un usuario se calcula como la suma de toda la actividad en Feed donde destination sea el usuario y el out degree lo mismo pero donde el source sea el usuario (es importante que no consideres dentro del cálculo del in y out degree los casos donde source y destination corresponden al mismo usuario).

Lo que debes hacer

Tienes que generar y probar 4 modelos de regresión logística: tres modelos de regresión usando las features de cada categoría (c1, c2 y c3) y uno que combine las

Categoría	*Features*
c1. Sentiment on social network	sosn_average_subjectivity_statuses
	sosn_average_subjectivity_comments
	sosn_sd_average_subjectivity_reviews
	sosn_sd_average_subjectivity_comments
	sosn_average_polarity_statuses
	sosn_average_polarity_comments
	sosn_sd_average_polarity_statuses
	sosn_sd_average_polarity_comments
c2. Sentiment on market place	somp_average_subjectivity_reviews
	somp_average_subjectivity_comments
	somp_sd_average_subjectivity_reviews
	somp_sd_average_subjectivity_comments
	somp_average_polarity_reviews
	somp_average_polarity_comments
	somp_sd_average_polarity_reviews
	somp_sd_average_polarity_comments
c3. Activity on social network	aosn_in_degree
	aosn_out_degree

Table 1: Variables (features) usadas para predecir si un usuario es o no un vendedor.

features de todas las categorías. Debes comparar los resultados usando las métricas a continuación.

- Precision
- Recall
- F-1
- AUC

Además debes generar un gráfico que muestre la curva ROC para cada modelo. Para todo el proceso debes utilizar k-fold cross validation con $k = 5$.

Informe

Debes escribir un informe relativamente breve. Al menos debes responder lo siguiente:

1. ¿Qué categoría de features (o combinación) es mejor predictor?

2. De los 4 modelos generados, ¿qué features (el sentimiento, actividad en la red social, etc..) explican que uno tenga mejor capacidad de predicción que el resto?

3. Calcule la media y desviación estándar de cada feature, ¿varían mucho las medias y desviaciones standard entre distintas features? ¿Qué efecto podría tener en el modelo de regresión normalizar las features?

4. Una interpretación de los coeficientes del modelo de regresión lógistica final (con todas las categorías). Particularmente responde la pregunta ¿Cuál es el efecto en la predicción de la variable vendedor si el in_degree aumenta en una unidad?

Bonus (1 punto)

Implementar el clasificador usando un método alternativo a Logistic Regression (L.R.), por ejemplo, Naive Bayes, Decision Trees o SVM y comparar el rendimiento con el modelo de L.R.