Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
Primer Semestre de 2015
Benjamín Assael C.
Raimundo Herrera S.

Informe Regresión Logística

Análisis del modelo e interpretación de los resultados

---

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

   Para decidir que set de features es el mejor predictor, nos basamos en los diferentes clasificadores. Por un lado, para los clasificadores *precision* y *AUC*, el mejor predictor es *C1*, el correspondiente a *features* de tipo social. Por otro lado, para *recall* y *F1*, el clasificador que mejor se comporta es *C4*, que corresponde a todos las *features*.

   Si promediamos los 4 clasificadores para cada combinación, obtenemos que el que obtiene el mejor promedio es *C1* (*features* de tipo social). Consideramos para lo anterior que cada uno de los clasificadores es mejor a medida que su valor crece, por lo que, el grupo que tenga un promedio mayor debería ser el mejor. Sin embargo, características como *precision* y *recall* (y por consiguiente *F1*) no pueden ser tajantemente evaluadas como mejores o peores debido a que en ciertos casos puede ser preferible obtener mayores resultados en una u otra. Para este caso, nos interesa el mayor valor posible de ambos porque queremos obtener la mayor cantidad posible de vendedores que realmente lo son, al mismo tiempo que queremos obtener información de la mayor cantidad posible de usuarios.

   Observando el *dataset*, se puede comprobar la conclusión anterior sobre la superioridad de *C1*, ya que existe más información para la actividad social, que para la actividad del *market*, lo que respalda de mejor manera nuestras métricas, en base a un mejor entrenamiento de modelos. Para tener una referencia, la cantidad de reviews y de comentarios de productos del *market*, en total, asciende a 1541, mientras que el número de elementos del *feed* llega a 63968.

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?

   Considerando los 4 modelos generados, y los coeficientes que resultan de la regresión al utilizar todos los features, vemos que los features que más influyen, es decir, que tienen la mejor capacidad de predicción, son aquellos que al realizar e^{β_i} siendo β_i el coeficiente, son mayores. Estos corresponden a features de sentimiento en la red social, específicamente a la desviación estándar de la *subjectivity* de los *status* (sosn_sd_average_subjectivity_status), seguido por la desviación estándar de la *subjectivity* de los *comments* (sosn_sd_average_subjectivity_comments), los cuales son sustantivamente mayores que los otros features.

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?

   El detalle de los cálculos de medias y desviaciones estándar, está en el archivo *std_mean_analysis.py*. De esos resultados observamos que efectivamente varían significativamente ambos valores. En el caso de la

desviación, vemos que el mayor corresponde al del *out_degree*, con un valor de 175,602346989, mientras que el menor es el de *somp_average_polarity_comments* cuya desviación es de 0,00333129291415. Para el caso de las medias, también hay variaciones sustantivas, siendo los extremos los mismos *features*, con valores de 9,84825 y $4{,}89737482087 * 10^{-5}$ respectivamente.

En el uso del modelo propiamente tal, por la forma en que se calcula, no hay una influencia directa al normalizar las *features*, de hecho las métricas y $p-values$ son iguales con o sin normalización, por lo tanto la precisión de las predicciones sería igual en ambos casos. Sin embargo normalizar puede ayudar en el análisis de resultados y predicciones, ya que al hacerlo, la interpretación de "incrementar en una unidad" el valor de algún *feature*, es distinto ya que ahora significaría, por ejemplo, pasar de el mínimo al máximo valor posible, al pasar de 0 a 1.

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?

   En la interpretación de coeficientes tenemos que considerar e^{β} siendo β cada coeficiente, esto nos dará un número, por ejemplo 1,2, lo que significa que el *odds ratio* de una predicción aumentaría en un 20 % al aumentar una unidad del *feature* relacionado a ese coeficiente, estando el *odds ratio* para este caso definido como:

$$odds\ ratio = \frac{probabilidad\ vendedor}{probabilidad\ comprador}$$

   Los coeficientes exponenciados son:

Feature	e^{β_i}
sosn_average_subjectivity_statuses	8.429165323547416
sosn_average_subjectivity_comments	5.545537150867584
sosn_sd_average_subjectivity_statuses	12.647780185787607
sosn_sd_average_subjectivity_comments	10.656329452724696
sosn_average_polarity_statuses	0.9457071180213715
sosn_average_polarity_comments	2.6418891663839434
sosn_sd_average_polarity_statuses	1.5979698507249072
sosn_sd_average_polarity_comments	0.5982211280917515
somp_average_subjectivity_reviews	2.8158128411709535
somp_average_subjectivity_comments	1.0995064734173756
somp_sd_average_subjectivity_reviews	1.2104705001429106
somp_sd_average_subjectivity_comments	1.0513404192340408
somp_average_polarity_reviews	1.470182988465006
somp_average_polarity_comments	1.0274003814093071
somp_sd_average_polarity_reviews	1.6245369602606223
somp_sd_average_polarity_comments	1.0335468783982462
aosn_in_degree	1.12532605851146
aosn_out_degree	0.9990091310725805

   Lo anterior se interpreta como un factor para cada *feature*, que si lo multiplicamos por el odds ratio, corresponde a lo que valdrá el odds ratio al aumentar en uno el valor de la *feature* correspondiente.

   Para el caso del *in_degree*, si aumenta en una unidad, obtendremos que el *odds ratio* aumentará en aproximadamente un 12,5 %. Para complementar esta respuesta, hemos realizado un *testeo* en el archivo *logistic_regression.py*, donde se muestra una predicción con cierto valor en el *in_degree*, el *odds ratio* calculado, y luego ambos valores para la predicción aumentando en una unidad el valor del *in_degree*.

5. *Bonus:* Comparación con método alternativo.

   En el archivo *logistic_regression.py*, se termina imprimiendo todos los evaluadores, es decir, *precision, recall, F-1 y AUC*, para otro clasificador, en este caso, el elegido fue el *Decision Tree Classifier*. Si comparamos el promedio de los *scores* obtenidos por este clasificador, usando todos los features, vemos que supera levemente (alrededor de 0,066) a la regresión logística en el conjunto que considera todos los features. Solo es inferior al modelo de regresión en *precision*. La supuesta superioridad del modelo *Decision Tree Classifier*, fue solo evaluada en cuanto a los *scores*, pero sabemos que, por las características del modelo, puede no ser conveniente siempre usarlo, como por ejemplo en *datasets* poco balanceados, o porque a veces produce un *overfitting* indeseado.