Trabajo Práctico 2 - Machine Learning

[7506] Organización de Datos
Primer cuatrimestre de 2019
24 Junio 2019

Alumna: ROLDAN, Maria Cecilia
Número de padrón: 101939
Repositorio: https://github.com/Cceecilia/OrganizacionDatosTP2

Índice

1. Introduction **2**

2. Procesamiento de los datos **2**

3. Features **3**
3.1. Subastas 3
3.2. Installs 3
3.3. Eventos 4
3.4. Clicks 4
3.5. Agregados 4

4. Algoritmos Utilizados **5**
4.1. Xgboost 5
4.2. Random Forest 5
4.3. Gradient Boosting 5
4.4. AdaBoost 5

5. El Mejor Resultado **6**

1. Introduction

El segundo trabajo práctico es una competencia de Machine Learning en donde cada grupo debe intentar determinar, para cada dispositivo presentado por Jampp, el tiempo que transcurrirá hasta que el mismo aparezca nuevamente en una subasta, y el tiempo hasta que el usuario del mismo decida instalar una nueva aplicación.

2. Procesamiento de los datos

Lo primero que hice fue dividir los cuatro archivos (auctions, events, clicks, installs) en ventanas de tres días: 18 a 20, 19 a 21, 20 a 22, 21 a 23, 22 a 24, 23 a 25 y 24 a 26, con lo cual me quedaron 28 archivos en total.

Para los archivos de los días 18-20, 19-21, 20-22, 21-23 y 24-26 creé nuevos archivos para poder entrenar con los features de cada archivo auctions, events, clicks e installs. Para los features no tuve en cuenta las columnas de los archivos originales que tuvieran más de la mitad de los datos nulos. Y para los archivos de los días 21-23, 22-24, 23-25 y 24-26 creé los archivos para poder testear auctions e installs.

A partir de esos archivos pude hacer los sets de entrenamiento para predecir los tiempos de las subastas y de las instalaciones. Para predecir las subastas hice un full join entre auctions, events y el archivo de testeo de auctions, además agregé los registros de clicks e installs que coincidieran con los anteriores para tener información extra. Para predecir las instalaciones hice un full join entre events, clicks, installs y el archivo que testea installs y agregé las subastas que coincidían. En ambos casos, cuando quedaban referencias a tiempos nulos los reemplazaba por el tiempo máximo (3 días en segundos), y las demás referencias nulas las reemplazaba por 0.

En el caso de los archivos para predecir las subastas, viendo que los errores al predecir eran mayores a los de las instalaciones, creé nuevos archivos de entrenamiento con distintos features y los comparé con los resultados anteriores.

Con estos 4 sets de entrenamiento (uno por cada ventana de tres días) pude entrenar los datos para encontrar buenos valores de hiperparámetros para usar en las predicciones. La forma de evaluar resultados que utilizo es el cálculo del error cuadrático medio. Al entrenar, lo calculo en cada set de datos y tomo el promedio en total.

En el caso de los features, use One Hot Encoding para pasar una variable categórica de las subastas en dos binarias, sin embargo no fueron significantes en las predicciones. En la mayoría de los casos uso Mean Encoding tomando la cantidad de veces que aparecen los valores y normalizándolos.

Para predecir los tiempos finales creé los archivos a predecir juntando los archivos de la misma forma que al hacer los sets de entrenamientos, pero usando los target_id del objetivo del trabajo en vez de los archivos creados para testear.

3. Features

3.1. Subastas

1. s_source_id: probabilidad de que aparezca el source_id del dispositivo en una subasta. Es uno de los features más importante a la hora de predecir las subastas.
2. s_ref_type1: es 1 si el ref_type del dispositivo es 1 y 0 sino. No resultó útil a la hora de predecir subastas.
3. s_ref_type7: es 1 si el ref_type del dispositivo es 7 y 0 sino. No resultó útil a la hora de predecir subastas.
4. s_t_min: tiempo mínimo en el cual apareció el dispositivo en una subasta. Es uno de los features más importante a la hora de predecir las subastas.
5. s_t_prom: tiempo promedio en el cual apareció el dispositivo en subasta.
6. s_t_max: tiempo máximo en el cual apareció el dispositivo en una subasta. Es uno de los features más importante a la hora de predecir las subastas.
7. s_subastas: cantidad de subastas de un dispositivo.

3.2. Installs

1. i_t_min: tiempo mínimo en el cual un dispositivo instaló una aplicación.
2. i_t_prom: tiempo promedio en el cual un dispositivo instaló una aplicación.
3. i_t_max: tiempo máximo en el cual un dispositivo instaló una aplicación.
4. i_app_id: probabilidad de que aparezca la application_id del dispositivo en una instalación.
5. i_installs: cantidad de instalaciones de un dispositivo.
6. i_model: probabilidad de que aparezca el modelo del dispositivo en una instalación.
7. i_language: probabilidad de que aparezca el lenguaje del dispositivo en una instalación. No resultó útil para predecir instalaciones.
8. i_type: probabilidad de que aparezca el tipo del dispositivo en una instalación.
9. i_wifi: cantidad de instalaciones con wifi del dispositivo. No resultó útil para predecir instalaciones.
10. i_implicit: cantidad de instalaciones con implicit de un dispositivo. No resultó útil para predecir instalaciones.
11. i_attributed: cantidad de instalaciones atribuidas a japp de un dispositivo. No resultó útil para predecir instalaciones.

3.3. Eventos

1. e_kind: probabilidad de que aparezca el kind del dispositivo en un evento.
2. e_app_id: probabilidad de que aparezca la aplicación del dispositivo en un evento. Es uno de los features más importante a la hora de predecir las instalaciones.
3. e_event_id: probabilidad de que aparezca el event_id de un dispositivo en un evento. No resultó útil a la hora de predecir subastas, pero sí para predecir instalaciones.
4. e_wifi: cantidad de eventos con wifi del dispositivo.
5. e_atribuidos: cantidad de eventos atribuidos a japp. No resultó útil para predecir.
6. e_eventos: cantidad de eventos de un dispositivo.
7. e_t_prom: es el tiempo promedio en el cual apareció un evento. Es uno de los features más importante a la hora de predecir las instalaciones.

3.4. Clicks

1. c_advertiser: probabilidad de que aparezca el aviso del dispositivo en un click.
2. c_osmin: probabilidad de que aparezca el os_mínimo del dispositivo en un click. No resultó útil para predecir instalaciones.
3. c_t_prom: tiempo promedio en el cual hizo click el dispositivo en un aviso.
4. c_clicks: cantidad de clicks de un dispositivo.
5. c_source: probabilidad de que aparezca el source del dispositivo en un click. No resultó útil para predecir.
6. c_wifi: cantidad de clicks con wifi del dispositivo. No resultó útil para predecir instalaciones.
7. c_type: probabilidad de que aparezca el tipo del dispositivo en un click.
8. c_timeToClick: promedio del tiempo que tarda el dispositivo en hacer click. No resultó útil para predecir instalaciones.
9. c_osmax: probabilidad de que aparezca el os_máximo del dispositivo en un click. No resultó útil para predecir instalaciones.
10. c_carrier: probabilidad de que aparezca el carrier del dispositivo en un click.

3.5. Agregados

1. subasta: es 1 si el dispositivo estuvo en al menos una subasta y 0 sino.
2. s_h_4-12: cantidad de veces que el dispositivo estuvo en subastas entre las 4 y 12 horas, en este rango de horas se estiman menos subastas.
3. s_h_12-20: cantidad de veces que el dispositivo estuvo en subastas entre las 12 y 20 horas.
4. s_h_20-4: cantidad de veces que el dispositivo estuvo en subastas entre las 20 y 4 horas, en este rango de horas se estiman más subastas.
5. e_eventosId: es la probabilidad global de que aparezca el evento_id multiplicado por la cantidad de eventos del dispositivo.

4. Algoritmos Utilizados

4.1. Xgboost

En una primera instancia busqué los mejores parámetros utilizando los archivos para predecir los tiempos de las subastas y las instalaciones por separado. Para predecir las subastas contaba solo con los datos de los archivos auctions y events, con los mejores parámetros que encontré el error era de 83 mil aproximado. Para predecir las instalaciones contaba con los archivos events, clicks e installs y con parámetros parecidos al anterior el error era 54 mil aproximado. Al subirlo a la competencia es score fue 111.400 aprox.

Para tratar de mejorar el resultado anterior, agregé los archivos clicks e installs para predecir las subastas, y el archivo auctions al de las instalaciones. Al subirlo nuevamente, mi score fue 111.200 con lo cual mejoré en 200, pero no fue una mejora significativa.

Por último, probé predecir las subastas con otro set de entrenamiento que contenía los features más importantes, pero su resultado fue peor al anterior.

4.2. Random Forest

Es el segundo algoritmo que utilicé. Tanto para entrenar como para predecir, utilice todos los archivos. Al predecir los tiempos de las subastas el error mínimo posible fue de 84 mil aproximado, y el error al predecir las instalaciones fue de 54 mil aproximado. Al probar con el otro set de entrenamiento para las subastas, el resultado no mejoró.

Varios de los parámetros obtenidos en las predicciones de las subastas coincidieron para predecir las instalaciones. Al subir las predicciones el score que obtuve fue 110.900 con lo cual mejoré el algoritmo anterior.

4.3. Gradient Boosting

Al utilizar este algoritmo decidí usar los mismos parámetros para predecir las subastas y las instalaciones, ya que en los algoritmos anteriores que probé sus parámetros no diferían mucho.

Al predecir los tiempos de las subastas el error que obtuve fue 87 mil aproximado y para los tiempos de las instalaciones fue 54 mil aproximado también. Al subir las predicciones el score fue 111.300, con lo cual empeoró las soluciones anteriores.

4.4. AdaBoost

Al entrenar este algoritmo, los errores en los resultados de las predicciones fueron bastante similares a los anteriores. Al principio tenía la intención de entrenarlo con XGBoost y RandomForest como estimadores base y poder comparar sus resultados, sin embargo entrenarlo con RandomForest tomó más tiempo del esperado.

Con XGBoost como estimador base de este algoritmo, el score de estas predicciones fue de 105.900 aproximado.

5. El Mejor Resultado

En los tres primeros casos las predicciones dieron resultados similares, por lo cual al intentar combinar sus objetivos y darle mayor peso al de mejor resultado terminó dando un score mayor al mínimo.

El algoritmo que mejor score tiene es AbaBoost usando como estimador base a XGBoost, con un valor de 105.900.