Logística de envíos: ¿Cuándo llega?

Mentoría DiploDatos 2019

Integrantes:
Alini, Walter
Salina, Noelia

Motivación

En la actualidad, cada vez más productos se comercializan a través de una plataforma online. Una de las principales ventajas de este sistema es que el usuario puede recibir el producto en su domicilio en una fecha determinada. Pero, ¿cómo sabemos qué día va a llegar? ¿A partir de qué datos podemos predecir la demora del envío? En este práctico se trabajará con datos de envíos de MercadoLibre, el e-commerce más grande de Latinoamérica, analizando y modelando el problema de logística de envíos para poder responder ¿cuándo llega?

Descripción del dataset

Datos: El conjunto de datos que recibimos corresponde a un muestreo de 500.000 envíos de MercadoLibre. Estos envíos fueron realizados en Brasil en el período comprendido entre Octubre de 2018 y Abril de 2019.

El dataset presenta la siguiente información:

- **sender_state**: Estado de Brasil de donde sale el envío.
- **sender_zipcode**: Código postal (de 5 dígitos) de donde sale el envío.
- **receiver_state**: Estado de Brasil a donde llega el envío.
- **receiver_zipcode**: Código postal (de 5 dígitos) a donde llega el envío.
- **shipment_type**: Método de envío (normal, express, super).
- **quantity**: Cantidad de productos en un envío.
- **service**: Servicio del correo con el cual se realizó un envío.
- **status**: Estado del envío (set: listo para ser enviado, sent: enviado, done: entregado, failed: no entregado, cancelled: cancelado).
- **date_created**: Fecha de creación del envío.
- **date_sent**: Fecha y hora en que se realizó el envío (salió del correo).
- **date_visit**: Fecha y hora en que se entregó el envío al destinatario.
- **shipment_days**: Días hábiles entre que el envío fue enviado (salió del correo) y que fue entregado.

Análisis y Curación de datos

De los 500.000 datos originales, eliminamos menos del 2% de los datos, posteriores a su análisis, y que corresponden a datos no confiables, por una o más de las siguientes características:

- Son datos “potencialmente duplicados”: si bien no tenemos una forma de determinar su duplicidad a partir de los datos, los consideramos duplicados por una alta probabilidad de que se refieran a envíos ya considerados;
- Son datos con información faltante: envíos que aún no terminaron (por lo tanto, no son buena información para una predicción de envíos)
- Son envíos creados después de ser enviados
- Son envíos creados después de ser recibidos
- Son envíos por fuera del período de análisis en cuestión (y luego, posiblemente erróneos)

Contamos con 490.439 datos para el análisis presente.

Target propuesto

El target (definición de lo que debemos estimar) propuesto es:
“Dado un envío que tenga como estado de origen Sao Paulo, cantidad de días hábiles que tardará el correo en hacerlo llegar a destino”.

Features importantes

Dividimos los features de acuerdo a la importancia a priori para el análisis de información, de la siguiente manera:

Alta importancia	Baja importancia	A evaluar incorporar
- receiver_state - receiver_zipcode - sender_zipcode - shipment_type - service - shipment_days*	- date_created - date_sent* - date_visit* - quantity - status	- Handling time: Tiempo que tarda el vendedor en despachar el envío - Cantidad de envíos a un lugar (estado, zona, área, etc.) - Distancia a destino - Fechas especiales: Hot Sale, feriados importantes, etc.

Los marcados con * no es información que contaremos al momento de estimar (sí al momento de entrenar y testear)

Clustering

A partir de estos datos, se corrieron algoritmos de clustering, como primeros acercamientos a baselines en términos de modelos de Machine Learning para la resolución de la problemática. Se dividió el target en 6 categorías (“0-1 días”, “2-3 días”, “4-5 días”, “6-7 días”, “8-9 días” y “10 o más días”) y se obtuvieron los siguientes resultados:

kNN

Si bien no es un algoritmo de clustering, sino de clasificación, lo utilizamos para correr las primeras pruebas alrededor de la problemática, con los siguientes resultados:

Mejor par de features encontrado: aquellos con los que mejor se puede predecir el target: **service** y **receiver_zipcode**.
Mínimo error medio para 10 vecinos cercanos: 0.4748 (precisión 0.5252)

A partir de este par de features, intentamos obtener distintos valores de error para encontrar un valor de N que sea razonable, con estos mejores resultados:

N = 50, error medio = 0.4475 (precisión 0.5525)
N = 100, error medio = 0.4476 (precisión 0.5524)

K-Means

Utilizamos K-Means como primera prueba de concepto en busca de una solución del problema. La mejor performance fue de un error medio de 0.6196 (precisión de 0.3804) que la encontramos con:
n_init = 10 (corridas distintas del algoritmo)
max_iter = 500 (número máximo de iteraciones de una corrida)
init = random (método de inicialización)
tol = 0.01 (precisión para convergencia)
algorithm = full (algoritmo utilizado)

Conclusiones y próximos pasos

- Contamos con una cantidad suficiente de datos para el problema en cuestión (490k aproximadamente)
- Analizamos y limpiamos la información de valores de poca confianza
- Definimos el target
- Tenemos un primer baseline de precisión en desarrollo, de 55% aproximadamente
- Hicimos las primeras pruebas sobre algoritmos de clasificación y de clustering.
- Próximos pasos:
    - Investigar más features
    - Investigar nuevos features
    - Ajustar modelos probados
    - Probar nuevos modelos