Prueba 2

- Para realizar esta prueba debes haber estudiado todo el material disponibilizado del módulo.
- Una vez terminada la prueba:
  - Realiza screenshots de la verificación de instalación de cada herramienta
  - Adjunta los archivos necesarios de cada ejercicio
  - Comprime la carpeta y sube el `.zip` a la sección correspondiente.

> Puntaje total = **26 puntos** (El alumno debe tener un total de 17 puntos para aprobar)

Descripción

Yelp es un directorio de servicios a nivel mundial, que permite a sus usuarios el evaluar los servicios (restaurants, bancos, clínicas, gimnasios, entre otros) para encontrar y sugerir mejores servicios.

Para esta prueba utilizaremos los datos disponibilizados por *Yelp* para:

- Identificar usuarios molestosos.
- Probabilidad de cierre de los negocios.

> Los datos se disponibilizaron en la página https://www.yelp.com/dataset/challenge.

Para efectos prácticos de la prueba:

- Los archivos disponibles se encuentran en el bucket del módulo, con la dirección `s3://bigdata-desafio/yelp-data/`.
- Dentro de esta dirección encontrará distintos archivos `json` con:
  - Registros respecto al negocio (`business.json`)
  - check-ins del usuario en un negocio (`checkin.json`)
  - Fotos asociadas al review (`photo.json`)
  - Reseñas de un usuario sobre el servicio (`review.json`)
  - Sugerencias del usuario sobre el servicio (`tip.json`)
  - Información del usuario (`user.json`)

La definición de la estructura de datos de cada `json` se encuentra alojada en la siguiente dirección provista por Yelp: https://www.yelp.com/dataset/documentation/main. (Dentro de este link se encontrarán las definiciones de las columnas y el tipo de registro).

Ejercicio 1: Identificando usuarios molestosos (12 puntos)

> Utilizando el archivo `user.json`

Desde Yelp están interesados en identificar a aquellos usuarios que se pueden considerar como molestosos. Para ello, tienen la siguiente definición de un usuario molestoso:

1. Un usuario molestoso es aquél que su promedio de evaluaciones es menor o igual a 2.
2. Un usuario molestoso es aquel que tiene en promedio menos de 100 reviews.
3. Un usuario molestoso es aquél que tiene cero fans.

A partir de esta definición, se le solicita los siguientes puntos:

- Identifique en una variable dummy todos los usuarios que se puedan clasificar como molestosos acorde al criterio.
- Recodificaciones en el archivo `user.json`:
  - `friends`, que corresponde a un string con todos los user_id de otros usuarios j que siguen al usuario i. El objetivo es contar la cantidad de amigos existentes.
  - `elite`, que corresponde a un string con todos los años en los que el usuario i fue considerado como un reviewer de elite. El objetivo es contar la cantidad de años en los cuales se consideró como elite.
  - Asegúrese de eliminar los siguientes registros: `friends`, `yelping_since`, `name`, `elite`, `user_id`.

A modo de referencia, esta es una entrada original:

```
[Row(average_stars=4.03, compliment_cool=1, compliment_cute=0,
compliment_funny=1, compliment_hot=2, compliment_list=0, compliment_more=0,
compliment_note=1, compliment_photos=0, compliment_plain=1,
compliment_profile=0, compliment_writer=2, cool=25, elite='2015,2016,2017',
fans=5, friends='c78V-rj8NQcQjOI8KP3UEA, alRMgPcngYSCJ5naFRBz5g,
ajcnq75Z5xxkvUSmmJ1bCg, BSMAmp2-wMzCkhTfq9ToNg, jka10dk9ygX76hJG0gfPZQ,
dut0e4xvme7QSlesOycHQA, l4l5lBnK356zBua7B-UJ6Q, 0HicMOOs-M_gl2eO-zES4Q,
_uI57wL2fLyftrcSFpfSGQ, T4_Qd0YWbC3co6WSMw4vxg, iBRoLWPtWmsI1kdbE9ORSA,
xjrUcid6Ymq0DoTJELkYyw, GqadWVzJ6At-vgLzK_SKgA, DvB13VJBmSnbFXBVBsKmDA,
vRP9nQkYTeNioDjtxZlVhg, gT0A1iN3eeQ8EMAjJhwQtw, 6yCWjFPtp_AD4x93WAwmnw,
1dKzpNnib-JlViKv8_Gt5g, 3Bv4_JxHXq-gVLOxYMQX0Q, ikQyfu1iViYh8T0us7wiFQ,
f1GGltNaB7K5DR1jf3dOmg, tgeFUChlh7v8bZFVl2-hjQ, -9-9oyXlqsMG2he5xIWdLQ,
Adj9fBPVJad8vSs-mIP7gw, Ce49RY8CKXVsTifxRYFTsw, M1_7TLi8CbdA89nFLlH4iw, wFsNv-
hqbW_F5-IRqfBN6g, 0Q1L7zXHocaUZ2gsG2XJeg, cBFgmOCBdhYa0xoFEAzp_g,
VrD_AgiFvzqtlR15vir3SQ, cpE-7HK514Sr5vpSen9CEQ, F1UYelhPFB-zIKlt0ygIZg,
CQAL1hvsLMCzuJf9AglsXw, 1KnY1wr15WfEWIRLB9IS6g, QWFQ-kXBiLbid-lm5Jr3dQ,
nymT8liFugCrM16lTy0ZfQ, qj69bdd885heDvUPCyHd2Q, DySCZZcgbdrlHgEovk5y9w,
lZMJIDuvhT9Dy4KyquLXyA, b_9Gn7wS93AoPZPR0dIJqQ, N07g1IaLh0_6sUjtiSRe4w,
YdfPX_7DxSnKvvdCJ57iOw, 8GYryZPD22W7WgQ8kvMkEQ, cpQmAgOWatghp14h1pn1dQ,
EnchhymLYMqftCRjqvVWmw, -JdfKhFktE7Zs9BMDFcPeQ, uWhC9eof98zPkvsalgaqJw,
eyTlNDDaiPatfe6mheIZ0g, VfHq0o73aKsODvfAhwAQtg, kvD5tICngLAaQDujSFWupA,
dXacwEhqi9-3_XT6JeH0Og, NfU0zDaTMEQ4-X9dbQWd9A, cTHWBdjSKbctSUIvWsgFxw,
3IEtCbSDF5t7RkZ20T6s9A, HJJXTrp6UybYyPdQ9DA0JA, JaXogQFVjzGRAeBvzamBHg,
NUonfKkjS1iVqnNITtgXZg, D5vaJAYp0sOrGfsj9qvsMA, H27Ecbwwu4FGAlLgICourw,
S8HrLmMiE4u8FWYWkNEoTw, Io36Y3xWQcIX9rYvPcYfXQ, J5mcqh8KxYpqjaLBNlwcig, -
nTB3_08g06fD0GT8AtDBQ, wMpFA46lihK8oFns_5p65A, RZGFJHeomGJCWp3xcL3ejA,
ZoQSzzXoSP1RxOD4Amv9Bg, qzM0EB0SkuuGIFv0adjQAQ, HuM6vvuveken-fPZ7d4olA,
H3oukHpGpn9n_mJwSDSQyQ, PkmsJsQ8FIZe8eh8c_u96g, wSByVbwME4MzgkJaFyfvNg,
YEVqknoDmrHAoUbHX0nPnA, li3vsK1XAPmeJYAUTYflHQ, MKc8yXi0glbPYt0Qb4PECw,
fQPH6W9fksi27gkuUPnFaA, amrCMrDsoRetYFg2kwwdFA, 84dVQ6n6r2ezNaTuc7RkKA,
yW9QjWY0olv5-uRKv3t_Kw, 5XJDj7c3eoidfQ3jW18Zgw, txSc6a6pIDctvwyBeu7Aqg,
HFbbDCyyqP9xPkUlcxeIdg, hTUv5oh2do6Z3OppPuuiJA, gSqonG9J4fNM-fl_fE71AA,
pd9mgTFpBTg5F9x-MsczNg, j3VE22V2GcHiH8UZxfFLfw, NYXlMW-T-3V4Jqr4r-i0Wg,
btxgAZedxX8IWhMifA7Xkg, -Hp5mPLiRJNFnyeX5Ygzag, P6-DwVg6-t2JuQwIUEk0iQ,
OI2TvxYvZrAodBG_RF53Xw, bHxf_VPKmZur1Bier-6A2A, Et_Sb39cVm81_Xe9HDM8ZQ,
5HwGl2UyYbaRq8aD6YC-fA, ZK228WMcCKLo5thcjD7rdw, iTf8wojwfm0NOi7dOiz3Nw,
btYRxQYNJjpecflNHtFH0A, Kgo42FzpW_dXFgDKoewbtg, MNk_1Q_dqOY3xxHZKeO8VQ,
AlwD504T9k0m5lkg3k5g6Q', funny=17, name='Rashmi', review_count=95, useful=84,
user_id='l6BmjZMeQD3rDxWUbiAiow', yelping_since='2013-10-08 23:11:33')
```

Y esta es la misma entrada posterior a la recodificación (**Importante:** Esta entrada no contiene la recodificación del vector objetivo [si es que el usuario se puede considerar molestoso]):

```
[Row(average_stars=4.03, compliment_cool=1, compliment_cute=0,
compliment_funny=1, compliment_hot=2, compliment_list=0, compliment_more=0,
compliment_note=1, compliment_photos=0, compliment_plain=1,
compliment_profile=0, compliment_writer=2, cool=25, fans=5, funny=17,
review_count=95, useful=84, friend_count=99, elite_count=3)]
```

Requerimientos

> Todos los objetivos se deben resolver utilizando `pyspark`.

- Genere la medición de usuarios molestos en base a los criterios expuestos. (2 puntos)

- Divida la muestra en conjuntos de entrenamiento (preservando un 70% de los datos) y validación (preservando un 30% de los datos). (1 punto)
- Entrene tres modelos (`LogisticRegression` , `GBTClassifier` y `DecisionTreeClassifier`) sin modificar hiperparámetros que en base a los atributos disponibles en el archivo `user.json` , clasifique los usuarios molestosos. (6 puntos)
- Reporte cuál es el mejor modelo en base a la métrica AUC. (1 punto)
- Identifique cuales son los principales atributos asociados a un usuario molestoso y repórtelos. (2 puntos)

Ejercicio 2: Identificando la probabilidad de cierre de un servicio (14 puntos)

> Utilizando el archivo `review.json`.

Desde Yelp están interesados en predecir la probabilidad de cierre de un servicio en base a los reviews y características de un negocio. Así, la primera iteración del modelo es generar una identificación de qué factores están asociados al cierre.

El equipo de desarrollo de Yelp le hace entrega de un archivo llamado `recoding_business_schema.py` que describe:

- Atributos a recodificar
- Atributos a mantener.

Este archivo sirve como guía **y no implementa la recodificación en el `pyspark.sql.dataframe.DataFrame`**, esto es tarea de usted.

De manera adicional, cabe destacar que éste archivo no incluye la recodificación del vector objetivo (`is_open`). Usted deberá recodificarla de manera tal de identificar como 1 aquellos servicios que cerraron y 0 el resto.

Para ejemplificar cómo debería quedar un registro posterior a la recodificación, tome el siguiente ejemplo **antes** de recodificar:

```
[Row(address='2818 E Camino Acequia Drive',
attributes=Row(AcceptsInsurance=None, AgesAllowed=None, Alcohol=None,
Ambience=None, BYOB=None, BYOBCorkage=None, BestNights=None, BikeParking=None,
BusinessAcceptsBitcoin=None, BusinessAcceptsCreditCards=None,
BusinessParking=None, ByAppointmentOnly=None, Caters=None, CoatCheck=None,
Corkage=None, DogsAllowed=None, DriveThru=None, GoodForDancing=None,
GoodForKids='False', GoodForMeal=None, HairSpecializesIn=None, HappyHour=None,
HasTV=None, Music=None, NoiseLevel=None, OutdoorSeating=None,
RestaurantsAttire=None, RestaurantsDelivery=None,
RestaurantsGoodForGroups=None, RestaurantsPriceRange2=None,
RestaurantsReservations=None, RestaurantsTableService=None,
RestaurantsTakeOut=None, Smoking=None, WheelchairAccessible=None, WiFi=None),
business_id='1SWheh84yJXfytovILXOAQ', categories='Golf, Active Life',
city='Phoenix', hours=None, is_open=0, latitude=33.5221425,
longitude=-112.0184807, name='Arizona Biltmore Golf Club',
postal_code='85016', review_count=5, stars=3.0, state='AZ')]
```

El mismo registro **posterior** a la recodificación. El registro presenta el vector objetivo (`is_open`) recodificado.

```
[Row(label=1, review_count=5, stars=3.0, accepts_insurance=0,
all_ages_allowed=0, alcohol_consumption=0, bitcoin_friendly=0, food_related=0,
finance_related=0, health_related=0, smoking=0, free_wifi=0, has_tv=0,
dog_friendly=0, kid_friendly=0, expensive_restaurant=0, loud_place=0,
happy_hour=0)]
```

Requerimientos

> Todos los objetivos se deben resolver utilizando `pyspark`.

- Implemente el esquema de recodificación. (2 puntos)
- Genere la recodificación del vector objetivo. (2 puntos)
- Divida la muestra en conjuntos de entrenamiento (preservando un 70% de los datos) y validación (preservando un 30% de los datos). (1 punto)
- Entrene tres modelos (`LogisticRegression` , `GBTClassifier` y `DecisionTreeClassifier`) sin modificar hiperparámetros que en base a los atributos recodificados del archivo `review.json` , clasifique aquellos servicios cerrados. (6 puntos)
- Reporte cuál es el mejor modelo en base a la métrica AUC. (1 punto)
- Identifique cuales son los principales atributos asociados al cierre de un servicio. (2 puntos)