Prueba - SQL para Data Science

- Para realizar esta prueba debes haber estudiado previamente todo el material disponibilizado correspondiente al módulo.
- Una vez terminada la prueba, comprime la carpeta que contiene el desarrollo de los requerimientos solicitados y sube el `.zip` en el LMS.
- Puntaje total: 10 puntos.
- Desarrollo prueba:
    - La prueba se debe desarrollar de manera Individual.
    - Para la realización de la prueba necesitarás apoyarte del archivo *Apoyo Prueba - SQL para Data Science*.

Desde **OkCupid** -aplicación de citas- solicitan el desarrollo de una serie de modelos predictivos.

Los datos a utilizar se registraron en base a una serie de perfiles públicos dentro de 25 millas de la ciudad de San Francisco activos durante el 2011.

Caveat: *Los permisos para obtener estos datos provinieron del presidente y co-fundador de OkCupid, Christian Rudder, con la condición de que se mantuvieran públicos.*

Requerimientos

Parte 1: Registro de los archivos en la base de datos. (3 Puntos)

- Generar una nueva base de datos con la siguiente nomenclatura: `apellido_nombre`.

- Importar en tablas los archivos `train_cupid.csv` y `test_cupid.csv` a un motor Postgres, **implementando solo la librería** `psycopg2`. Las tablas deben contener los nombres de las columnas y el total de los registros presente en cada archivo.

Parte 2: Entrenamiento de modelos (3.5 Puntos)

- Ingestar la tabla de training **mediante** `psycopg2` para el posterior entrenamiento del modelo.

- Entrenar los siguientes modelos (sin necesidad de ajustar por hiper parámetros):
  - `GradientBoostingClassifier`, `AdaBoostClassifer`, `RandomForestClassifier`, `SVC`, `DecisionTreeClassifier`, `LogisticRegression`, `BernoulliNB`.
  - Existen tres vectores objetivos a evaluar: single, seeing someone y available.

- Serializar el objeto y preservarlo por cada combinación de modelo entrenado y vector objetivo.

Parte 3: Exportación de predicciones (3.5 Puntos)

- Ingestar la tabla de testing **mediante** `psycopg2` para la posterior predicción del modelo.

- **En base a los objetos serializados**, predecir y evaluar cuatro queries específicas:
  - **Query 1:** 'atheism', 'asian', 'employed', 'pro_dogs', 'chinese'.
  - **Query 2:** 'income_over_75', 'french', 'german','orientation_straight', 'new york'.
  - **Query 3:** 'education_undergrad_university', 'body_type_regular', 'pro_dogs', 'employed'.
  - **Query 4:** 'taurus', 'indian', 'washington', 'income_between_50_75', 'hinduism'.

- Cada una de estas queries específicas debe ser registrada en la base de datos.

- La base de datos creada debe contener las tablas:
  - 2 que representan a training y testing.
  - 84 que representan a cada una de las combinaciones entre modelo, vector y query específica.

- A modo de referencia, la base de datos creada debe contener 86 tablas en total.

Archivos

La evaluación considera los siguientes documentos:

- Un archivo de funciones auxiliares con todas las funciones implementadas.
- Un notebook con el procedimiento implementado.
- Un `.zip` con los objetos serializados.
- Un archivo pgsql donde exportará la **base de datos** creada en el punto 1. Para generar el archivo, pueden implementar la siguiente línea **desde la consola**.

```
pg_dump -U nombre_usuario nombre_tabla > apellido_nombre_sql_test.pgsql
```

Aspectos adicionales a considerar

- Los archivos `csv` ya se encuentran preprocesados, **no es necesario realizar limpieza**. El procedimiento se encuentra en el archivo `Preprocesamiento de datos`.