

Manual de Usuario de la herramienta:

Segmentación Inteligente como Potenciador



Indicaciones y recursos

Seguir los siguientes pasos para el acceso y uso de la herramienta compuesta por un dashboard que describe las características destacadas de cada grupo de clientes por reservación y una API para predecir el grupo al que pertenece un cliente nuevo.

1. Ingresar al siguiente link, donde se encuentran los archivos necesarios para procesar una base de datos nueva y generar el análisis con base en ella (se debe descargar el .zip y extraer, para posteriormente abrir toda la carpeta en el entorno de programación de su elección.

https://drive.google.com/drive/folders/1VbnImwr0G6GDdSARrt8EQD_yd_1q9wg6?usp=sharing

- a. Sin embargo, en caso de desear utilizar el análisis derivado de la información actual, sólo es necesario ingresar a la siguiente liga y acceder con las credenciales *user user* (usuario y contraseña, respectivamente):

<https://dashk2py-ahqkpr6jhqv8h5vir7lm6e.streamlit.app/>

2. En la terminal del entorno de programación, crear un ambiente virtual “venv” con los siguientes comandos:

```
python -m venv .venv
```

3. Con esto, estaremos dentro del ambiente virtual y continuaremos con la instalación de *Kedro*, *MLflow* y *Kedro-MLflow*.

- a. Kedro:

Es una herramienta para el manejo y desarrollo de proyectos de machine-learning. Se instala en la terminal del entorno de programación, mediante el comando:

```
pip install kedro
```

o

```
conda install -c conda-forge kedro
```

En la siguiente dirección se puede encontrar documentación y tutoriales para ello:

<https://kedro.org/#get-started>

- b. MLflow

Al igual que Kedro, se instala mediante el uso de comandos en terminal, y ayuda con el ciclo de vida completo de este tipo de proyectos, para hacerlos manejables y reproducibles. Se utiliza el comando siguiente:

```
pip install mlflow
```

Se pueden encontrar tutoriales y apoyos en la siguiente liga:

<https://mlflow.org/docs/latest/getting-started/intro-quickstart>

c. *Kedro-MLflow*

Lo análogo, con el comando

```
pip install kedro-mlflow
```

4. Una vez en el entorno de programación, con la carpeta del proyecto abierta, se debe cargar el archivo .csv que contiene los datos a procesar en la subcarpeta “data/01_raw”. Las columnas de este archivo deben ser las siguientes para que coincidan con la base de datos original y el proyecto corra sin problemas:

ID_Reserva	h_num_adu	aa_h_tot_hab	ID_canal	h_fec_sda	aa_Reservacio n	h_nom
Fecha_hoy	aa_h_num_adu	ID_Programa	h_fec_lld	h_fec_sda_ok	ID_estatus_res ervaciones	h_tfa_total
h_res_fec	h_num_men	ID_Paquete	h_fec_lld_ok	h_fec_sda_okt	h_can_res	aa_h_tfa_total
h_res_fec_ok	aa_h_num_me n	ID_Segmento_ Comp	h_fec_lld_okt	ID_Pais_Orige n	h_cod_reserva	moneda_cve
h_res_fec_okt	h_num_noc	ID_Agencia	h_fec_reg	Cliente_Dis	h_edo	h_ult_cam_fec
h_num_per	aa_h_num_noc	ID_empresa	h_fec_reg_ok	aa_Cliente_Dis p	h_codigop	h_ult_cam_fec _ok
aa_h_num_per	h_tot_hab	ID_Tipo_Habita cion	h_fec_reg_okt	Reservacion	h_correo_e	h_ult_cam_fec _okt

Regresar a la terminal para correr el comando:

```
kedro run
```

5. De esta manera, se generan los archivos utilizados en los productos finales:
 - a. “reservas_clustered.parquet”, en el que se guardan, se pueden observar y manipular los datos de la base ya procesada y etiquetada por grupo (la columna que indica el cluster se llama “cluster_gmm”). Adicionalmente, se incluyen columnas derivadas de valor como el día de la semana de distintas fechas, la tarifa por noche, la cantidad de días de anticipación con la que se realizó la reservación, entre otras. Se ubica en la carpeta “data/07_model_output”
 - b. “svm_model.pkl”, en el que se guarda la información necesaria para ejecutar el modelo con nuevos datos y que posteriormente se utiliza en la API, donde hay una interfaz que le permite al usuario realizar predicciones con información nueva (su uso con la información original se puede realizar mediante las instrucciones en el punto 1 b). Se ubica en la carpeta “data/06_models”.

API de predicción:

El funcionamiento de la API, se basa en tres archivos que trabajan en conjunto de forma que haya una experiencia de usuario interactiva y enlazada al modelo predictivo previamente entrenado.

Paso 1: Inicio de sesión

Al abrir la aplicación web el usuario accede a la pantalla de autenticación. En la cual deberá ingresar un usuario y contraseña para continuar. Como ejemplo las credenciales predeterminadas fueron

- Usuario: admin
- Contraseña: dechava

El archivo correspondiente a controlar este proceso es el del código principal del servidor Flask: [app.py](#), puesto que se validan las credenciales en la ruta principal.

El archivo *login.html* se encarga de el formulario visual de inicio de sesión



Iniciar sesión

Usuario

Contraseña

Entrar

Figura 1. Visualización de inicio de sesión para la API

Una vez autenticado el usuario se redirige al formulario de predicción, en el cual se piden los siguientes campos:

- Fecha de reservación (formato aaaa-mm-dd)
- Fecha de llegada (formato aaaa-mm-dd)
- Número de adultos
- Número de menores
- Número de noches
- ID tipo de habitación
- ID paquete
- Canal de reservación

El archivo *formulario.html* se encarga de la interfaz visual para capturar los datos de reservación.



TCA software[®]
solutions

Predicción de Cluster

Fecha de reservación:

Fecha de llegada:

Figura 2. Visualización del formulario

Después del procesamiento, el sistema muestra en pantalla el número de clúster al que pertenece la reservación ingresada



1

Canal para reservación:

00

Predecir

Resultado: 2

Figura 2. Visualización de resultados