Lab_9_enunciado

October 25, 2024

Laboratorio 9: Optimización de modelos

MDS7202: Laboratorio de Programación Científica para Ciencia de Datos - Primavera 2024

0.0.1 Cuerpo Docente:

- Profesores: Ignacio Meza, Sebastián Tinoco
- Auxiliar: Eduardo Moya
- Ayudantes: Nicolás Ojeda, Melanie Peña, Valentina Rojas

0.0.2 Equipo: SUPER IMPORTANTE - notebooks sin nombre no serán revisados

- Nombre de alumno 1: Fabrizzio Pezzolla
- Nombre de alumno 2: Rodrigo Molina

0.0.3 Link de repositorio de GitHub: Link

0.0.4 Temas a tratar

- Predicción de demanda usando `xgboost`
- Búsqueda del modelo óptimo de clasificación usando `optuna`
- Uso de pipelines.

0.0.5 Reglas:

- **Grupos de 2 personas**
- Cualquier duda fuera del horario de clases al foro. Mensajes al equipo docente serán respondidos por este medio.
- Prohibidas las copias.
- Pueden usar cualquer matrial del curso que estimen conveniente.
- Código que no se pueda ejecutar, no será revisado.

0.0.6 Objetivos principales del laboratorio

- Optimizar modelos usando `optuna`
- Recurrir a técnicas de *prunning*
- Forzar el aprendizaje de relaciones entre variables mediante *constraints*
- Fijar un pipeline con un modelo base que luego se irá optimizando.

El laboratorio deberá ser desarrollado sin el uso indiscriminado de iteradores nativos de python (aka “for”, “while”). La idea es que aprendan a exprimir al máximo las funciones optimizadas que nos entrega `pandas`, las cuales vale mencionar, son bastante más eficientes que los iteradores nativos sobre DataFrames.

1 Importamos librerias útiles

[1]:
```
!pip install -qq xgboost optuna
```

2 El emprendimiento de Fiu

Tras liderar de manera exitosa la implementación de un proyecto de ciencia de datos para caracterizar los datos generados en Santiago 2023, el misterioso corpóreo **Fiu** se anima y decide levantar su propio negocio de consultoría en machine learning. Tras varias e intensas negociaciones, Fiu logra encontrar su *primera chamba*: predecir la demanda (cantidad de venta) de una famosa productora de bebidas de calibre mundial. Al ver el gran potencial y talento que usted ha demostrado en el campo de la ciencia de datos, Fiu lo contrata como data scientist para que forme parte de su nuevo emprendimiento.

Para este laboratorio deben trabajar con los datos `sales.csv` subidos a u-cursos, el cual contiene una muestra de ventas de la empresa para diferentes productos en un determinado tiempo.

Para comenzar, cargue el dataset señalado y visualice a través de un `.head` los atributos que posee el dataset.

[2]:
```
import pandas as pd
import numpy as np
from datetime import datetime

df = pd.read_csv('./sales.csv')

df.head()
```

[2]:
```
   id      date    city       lat      long     pop    shop        brand  \
0   0  31/01/12  Athens  37.97945  23.71622  672130  shop_1  kinder-cola
1   1  31/01/12  Athens  37.97945  23.71622  672130  shop_1  kinder-cola
2   2  31/01/12  Athens  37.97945  23.71622  672130  shop_1  kinder-cola
3   3  31/01/12  Athens  37.97945  23.71622  672130  shop_1   adult-cola
4   4  31/01/12  Athens  37.97945  23.71622  672130  shop_1   adult-cola

  container capacity  price  quantity
0     glass    500ml   0.96     13280
1   plastic    1.5lt   2.86      6727
2       can    330ml   0.87      9848
3     glass    500ml   1.00     20050
4       can    330ml   0.39     25696
```

[3]:
```
df['date'].shape
```

[3]:
```
(7456,)
```

2.1 1 Generando un Baseline (5 puntos)

Antes de entrenar un algoritmo, usted recuerda los apuntes de su magíster en ciencia de datos y recuerda que debe seguir una serie de *buenas prácticas* para entrenar correcta y debidamente su modelo. Después de un par de vueltas, llega a las siguientes tareas:

1. Separe los datos en conjuntos de train (70%), validation (20%) y test (10%). Fije una semilla para controlar la aleatoriedad. [0.5 puntos]
2. Implemente un `FunctionTransformer` para extraer el día, mes y año de la variable `date`. Guarde estas variables en el formato categorical de pandas. [1 punto]
3. Implemente un `ColumnTransformer` para procesar de manera adecuada los datos numéricos y categóricos. Use `OneHotEncoder` para las variables categóricas. `Nota:` Utilice el método `.set_output(transform='pandas')` para obtener un DataFrame como salida del `ColumnTransformer` [1 punto]
4. Guarde los pasos anteriores en un `Pipeline`, dejando como último paso el regresor `DummyRegressor` para generar predicciones en base a promedios. [0.5 punto]
5. Entrene el pipeline anterior y reporte la métrica `mean_absolute_error` sobre los datos de validación. ¿Cómo se interpreta esta métrica para el contexto del negocio? [0.5 puntos]
6. Finalmente, vuelva a entrenar el `Pipeline` pero esta vez usando `XGBRegressor` como modelo **utilizando los parámetros por default**. ¿Cómo cambia el MAE al implementar este algoritmo? ¿Es mejor o peor que el `DummyRegressor`? [1 punto]
7. Guarde ambos modelos en un archivo .pkl (uno cada uno) [0.5 puntos]

1.1. Separar los datos:

[4]:
```python
from sklearn import set_config
from sklearn.model_selection import train_test_split

set_config(transform_output="pandas")

# split all into train and test+dev
train_data, temp_data = train_test_split(df, test_size=0.3, random_state=42)
# split test and dev
dev_data, test_data = train_test_split(temp_data, test_size=1/3,␣
 ↪random_state=42)

def split_X_y(data, target='quantity'):
    X_ = data.drop(columns=target)
    y_ = data[target]
    return X_,y_

X_train, y_train = split_X_y(train_data)
X_test, y_test = split_X_y(test_data)
X_dev, y_dev = split_X_y(dev_data)
```

1.2. FuncTransformer:

[5]:
```python
from sklearn.preprocessing import FunctionTransformer
```

```python
def date_splitter(X):
    X = pd.to_datetime(X['date'], format='%d/%m/%y')
    return pd.DataFrame({
        'day': X.dt.day.astype('category'),
        'month': X.dt.month.astype('category'),
        'year': X.dt.year.astype('category')
    })

date_transformer = FunctionTransformer(date_splitter)
```

1.3. ColumnTransformer:

[6]:
```python
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, StandardScaler

col_transformer = ColumnTransformer(
    transformers=[
        ('date', date_transformer, ['date']),
        ('onehot', OneHotEncoder(sparse_output=False), ['city', 'shop',␣
 ↪'brand', 'container', 'capacity']),
        ('scaler', StandardScaler(), ['lat', 'long', 'pop', 'price'])
    ],
    remainder='passthrough'
)

# col_transformer.set_output(transform='pandas')
```

1.4. DummyRegressor:

[7]:
```python
from sklearn.pipeline import Pipeline
from sklearn.dummy import DummyRegressor

pipe_dummy = Pipeline([
    ('preprocessor', col_transformer),
    ('regressor', DummyRegressor(strategy='mean'))
])
```

1.5. Entrenar y obtener MAE:

[8]:
```python
pipe_dummy.fit(X_train, y_train)

from sklearn.metrics import mean_absolute_error

y_pred_dev = pipe_dummy.predict(X_dev)
mae_baseline = mean_absolute_error(y_dev, y_pred_dev)
mae_baseline
```

[8]:
```
13298.497767341096
```

1.6. XGBRegressor:

[9]:

```
from xgboost import XGBRegressor
import warnings

try:

    pipe_xgbreg = Pipeline([
        ('preprocessor', col_transformer),
        ('regressor', XGBRegressor(enable_categorical=False))
    ])

    pipe_xgbreg.fit(X = X_train, y = y_train)

except ValueError:
    warnings.warn("Can't use default arguments!")

    pipe_xgbreg = Pipeline([
    ('preprocessor', col_transformer),
    ('regressor', XGBRegressor(
        enable_categorical=True,
        random_state=42
        ))
    ])

    pipe_xgbreg.fit(X = X_train, y = y_train)
```

```
C:\Users\rodri\AppData\Local\Temp\ipykernel_7704\2232670368.py:14: UserWarning:
Can't use default arguments!
  warnings.warn("Can't use default arguments!")
```

[10]:

```
y_pred_dev = pipe_xgbreg.predict(X_dev)
mae_xgboost = mean_absolute_error(y_dev, y_pred_dev)
mae_xgboost
```

[10]:

```
2400.27751240516
```

1.7. pickle:

[11]:

```
import pickle

model_dict = {'dummy': pipe_dummy,
              'xgbreg': pipe_xgbreg}

for model_name, model_pipe in model_dict.items():
    with open(f'./pickle_1_7_{model_name}.pkl', 'wb') as pkl_file:
        pickle.dump(model_pipe, pkl_file)
```

2.2 2. Forzando relaciones entre parámetros con XGBoost (10 puntos)

Un colega aficionado a la economía le *sopla* que la demanda guarda una relación inversa con el precio del producto. Motivado para impresionar al querido corpóreo, se propone hacer uso de esta información para mejorar su modelo realizando las siguientes tareas:

1. Vuelva a entrenar el `Pipeline` con `XGBRegressor`, pero esta vez forzando una relación monótona negativa entre el precio y la cantidad. Para aplicar esta restricción apóyese en la siguiente documentación. [6 puntos]

   Hint 1: Para implementar el constraint se le sugiere hacerlo especificando el nombre de la variable. De ser así, probablemente le sea útil **mantener el formato de pandas** antes del step de entrenamiento.

   Hint 2: Puede obtener el nombre de las columnas en el paso anterior al modelo regresor mediante el método `.get_feature_names_out()`

2. Luego, vuelva a reportar el `MAE` sobre el conjunto de validación. [1 puntos]
3. ¿Cómo cambia el error al incluir esta relación? ¿Tenía razón su amigo? [2 puntos]
4. Guarde su modelo en un archivo .pkl [1 punto]

```
[12]: df_ = col_transformer.fit_transform(X_train)
      df_.head()
```

```
[12]:      date__day  date__month  date__year  onehot__city_Athens  \
      292         30            4        2012                  0.0
      3366        28            2        2015                  1.0
      3685        30            6        2015                  1.0
      2404        30            4        2014                  1.0
      2855        30            9        2014                  0.0

            onehot__city_Irakleion  onehot__city_Larisa  onehot__city_Patra  \
      292                      0.0                  0.0                 1.0
      3366                     0.0                  0.0                 0.0
      3685                     0.0                  0.0                 0.0
      2404                     0.0                  0.0                 0.0
      2855                     1.0                  0.0                 0.0

            onehot__city_Thessaloniki  onehot__shop_shop_1  onehot__shop_shop_2  \
      292                         0.0                  0.0                  0.0
      3366                        0.0                  1.0                  0.0
      3685                        0.0                  0.0                  0.0
      2404                        0.0                  1.0                  0.0
      2855                        0.0                  0.0                  1.0

            …  onehot__container_glass  onehot__container_plastic  \
      292   …                      0.0                        1.0
      3366  …                      0.0                        1.0
      3685  …                      0.0                        0.0
```

```
2404  …                      0.0                        0.0
2855  …                      0.0                        0.0

      onehot__capacity_1.5lt  onehot__capacity_330ml  onehot__capacity_500ml  \
292                      1.0                     0.0                     0.0
3366                     1.0                     0.0                     0.0
3685                     0.0                     1.0                     0.0
2404                     0.0                     1.0                     0.0
2855                     0.0                     1.0                     0.0

      scaler__lat  scaler__long  scaler__pop  scaler__price  remainder__id
292     -0.041439     -1.403508    -0.815231       1.659156            300
3366    -0.201218      0.416377     1.362723      -0.594186           3416
3685    -0.211468      0.389617     1.362723      -0.655752           3741
2404    -0.201218      0.416377     1.366906      -1.099033           2441
2855    -1.800022      1.726977    -0.936680      -0.778886           2898

[5 rows x 30 columns]
```

[13]:

```
pipe_xgbreg_restricted = Pipeline([
    ('preprocessor', col_transformer),
    ('regressor', XGBRegressor(
        enable_categorical=True,
        monotone_constraints={'scaler__price':-1},
        random_state=42
        ))
])

pipe_xgbreg_restricted.fit(X_train,y_train)
```

[13]:

```
Pipeline(steps=[('preprocessor',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('date',
FunctionTransformer(func=<function date_splitter at 0x000001AA50C6D080>),
                                                  ['date']),
                                                 ('onehot',
OneHotEncoder(sparse_output=False),
                                                  ['city', 'shop', 'brand',
                                                   'container', 'capacity']),
                                                 ('scaler', StandardScaler(),
                                                  ['lat', 'long', 'pop',
                                                   'price'])])),
                ('regressor',
                 XGBRegre…
                              importance_type=None,
                              interaction_constraints=None, learning_rate=None,
                              max_bin=None, max_cat_threshold=None,
                              max_cat_to_onehot=None, max_delta_step=None,
                              max_depth=None, max_leaves=None,
                              min_child_weight=None, missing=nan,
                              monotone_constraints={'scaler__price': -1},
                              multi_strategy=None, n_estimators=None,
                              n_jobs=None, num_parallel_tree=None,
                              random_state=42, …))])
```

2.2. MAE:

```
[14]: y_pred_dev = pipe_xgbreg_restricted.predict(X_dev)
      mae_xgboost_r = mean_absolute_error(y_dev, y_pred_dev)
      mae_xgboost_r
```

```
[14]: 2602.15612350863
```

2.3. No tenía razón. El MAE crece en comparación al caso en donde se fuerza la relación monotónica. Esto quiere decir que no ayuda al modelo asumir que hay una relación inversa entre cantidad de ventas y el precio del producto.

2.4.

```
[15]: with open(f'./pickle_2_4_xgbreg_const.pkl', 'wb') as pkl_file:
          pickle.dump(pipe_xgbreg_restricted, pkl_file)
```

3 3. Optimización de Hiperparámetros con Optuna (20 puntos)

Luego de presentarle sus resultados, Fiu le pregunta si es posible mejorar *aun más* su modelo. En particular, le comenta de la optimización de hiperparámetros con metodologías bayesianas a través del paquete `optuna`. Como usted es un aficionado al entrenamiento de modelos de ML, se propone implementar la descabellada idea de su jefe.

A partir de la mejor configuración obtenida en la sección anterior, utilice `optuna` para optimizar sus hiperparámetros. En particular, se pide que su optimización considere lo siguiente:

- Fijar una semilla en las instancias necesarias para garantizar la reproducibilidad de resultados
- Utilice `TPESampler` como método de muestreo
- De `XGBRegressor`, optimice los siguientes hiperparámetros:
  - `learning_rate` buscando valores flotantes en el rango (0.001, 0.1)
  - `n_estimators` buscando valores enteros en el rango (50, 1000)
  - `max_depth` buscando valores enteros en el rango (3, 10)
  - `max_leaves` buscando valores enteros en el rango (0, 100)
  - `min_child_weight` buscando valores enteros en el rango (1, 5)
  - `reg_alpha` buscando valores flotantes en el rango (0, 1)
  - `reg_lambda` buscando valores flotantes en el rango (0, 1)
- De `OneHotEncoder`, optimice el hiperparámetro `min_frequency` buscando el mejor valor flotante en el rango (0.0, 1.0)

Para ello se pide los siguientes pasos: 1. Implemente una función `objective()` que permita minimizar el `MAE` en el conjunto de validación. Use el método `.set_user_attr()` para almacenar el mejor pipeline entrenado. [10 puntos] 2. Fije el tiempo de entrenamiento a 5 minutos. [1 punto] 3. Optimizar el modelo y reportar el número de *trials*, el `MAE` y los mejores hiperparámetros encontrados. ¿Cómo cambian sus resultados con respecto a la sección anterior? ¿A qué se puede deber esto? [3 puntos] 4. Explique cada hiperparámetro y su rol en el modelo. ¿Hacen sentido los rangos de optimización indicados? [5 puntos] 5. Guardar su modelo en un archivo .pkl [1 punto]

3.1., 3.2., 3.3

[16]:

```python
import optuna
from optuna.samplers import TPESampler
import sklearn.datasets
import sklearn.ensemble
import sklearn.svm

optuna.logging.set_verbosity(optuna.logging.DEBUG)

def objective(trial):

    pipe_param = {
        'learning_rate': trial.suggest_float('learning_rate',.001,.1),
        'n_estimators': trial.suggest_int('n_estimators',50,1000),
        'max_depth': trial.suggest_int('max_depth',3,10),
        'max_leaves': trial.suggest_int('max_leaves',0,100),
        'min_child_weight': trial.suggest_int('min_child_weight',1,5),
        'reg_alpha': trial.suggest_float('reg_alpha',0,1),
        'reg_lambda': trial.suggest_float('reg_lambda',0,1)
    }

    col_transformer = ColumnTransformer(
        transformers=[
            ('date', date_transformer, ['date']),
            ('onehot',
                OneHotEncoder(
                    sparse_output=False,
                    min_frequency = trial.suggest_float('min_frequency',.0,.1)),
                ['city', 'shop', 'brand', 'container', 'capacity']),
            ('scaler', StandardScaler(), ['lat', 'long', 'pop', 'price'])
        ],
        remainder='passthrough'
    )

    pipe_xgbreg_ = Pipeline([
        ('preprocessor', col_transformer),
        ('regressor', XGBRegressor(
            enable_categorical=True,
            random_state=42,
            **pipe_param
            ))
    ])

    # X_train, X_val, y_train, y_val = sklearn.model_selection.
 ↪train_test_split(X, y, random_state=42)

    pipe_xgbreg_.fit(X_train, y_train)
    y_pred = pipe_xgbreg_.predict(X_dev)

    error = sklearn.metrics.mean_absolute_error(y_dev, y_pred)

    return error  # An objective value linked with the Trial object.

study = optuna.create_study(sampler=TPESampler())  # Create a new study.
study.optimize(objective, timeout=5*60)  # Invoke optimization of the objective␣
 ↪function.
```

```
[I 2024-10-25 00:00:59,238] A new study created in memory with name: no-
name-400f65da-627f-4f59-b12d-c0f51f9ba37f
[I 2024-10-25 00:01:00,665] Trial 0 finished with value: 2109.6047680942424 and
parameters: {'learning_rate': 0.07300853010389087, 'n_estimators': 573,
'max_depth': 5, 'max_leaves': 65, 'min_child_weight': 4, 'reg_alpha':
0.7530611989562601, 'reg_lambda': 0.07198627501001542, 'min_frequency':
0.0791332886704157}. Best is trial 0 with value: 2109.6047680942424.
[I 2024-10-25 00:01:00,998] Trial 1 finished with value: 9515.607919653337 and
parameters: {'learning_rate': 0.004702099603579871, 'n_estimators': 142,
'max_depth': 8, 'max_leaves': 17, 'min_child_weight': 1, 'reg_alpha':
0.9810716857110336, 'reg_lambda': 0.6148643673841265, 'min_frequency':
0.08096051282910686}. Best is trial 0 with value: 2109.6047680942424.
[I 2024-10-25 00:01:02,533] Trial 2 finished with value: 3342.075292903253 and
parameters: {'learning_rate': 0.005696749471875801, 'n_estimators': 815,
'max_depth': 5, 'max_leaves': 82, 'min_child_weight': 2, 'reg_alpha':
0.6216036030187583, 'reg_lambda': 0.7561313558368218, 'min_frequency':
0.05789453949006275}. Best is trial 0 with value: 2109.6047680942424.
[I 2024-10-25 00:01:03,416] Trial 3 finished with value: 2640.276770649941 and
parameters: {'learning_rate': 0.04282308360210183, 'n_estimators': 950,
'max_depth': 3, 'max_leaves': 86, 'min_child_weight': 4, 'reg_alpha':
0.48841681712601026, 'reg_lambda': 0.8373227370445386, 'min_frequency':
0.0927708909589483}. Best is trial 0 with value: 2109.6047680942424.
[I 2024-10-25 00:01:05,453] Trial 4 finished with value: 2047.5548127632985 and
parameters: {'learning_rate': 0.03726375889098917, 'n_estimators': 918,
'max_depth': 8, 'max_leaves': 70, 'min_child_weight': 5, 'reg_alpha':
0.039965732821498845, 'reg_lambda': 0.2731489631427, 'min_frequency':
0.07811875115173475}. Best is trial 4 with value: 2047.5548127632985.
[I 2024-10-25 00:01:06,057] Trial 5 finished with value: 2291.142388838238 and
parameters: {'learning_rate': 0.03653851224090242, 'n_estimators': 535,
'max_depth': 7, 'max_leaves': 30, 'min_child_weight': 3, 'reg_alpha':
0.7080913291532012, 'reg_lambda': 0.698361556268623, 'min_frequency':
0.09761940760049179}. Best is trial 4 with value: 2047.5548127632985.
```

```
[I 2024-10-25 00:01:06,490] Trial 6 finished with value: 2862.878885290912 and
parameters: {'learning_rate': 0.01796425131275789, 'n_estimators': 396,
'max_depth': 5, 'max_leaves': 88, 'min_child_weight': 3, 'reg_alpha':
0.31800590810696805, 'reg_lambda': 0.8210880784532824, 'min_frequency':
0.097195158616879}. Best is trial 4 with value: 2047.5548127632985.
[I 2024-10-25 00:01:07,285] Trial 7 finished with value: 2336.0857434646946 and
parameters: {'learning_rate': 0.03701695965865949, 'n_estimators': 947,
'max_depth': 4, 'max_leaves': 13, 'min_child_weight': 2, 'reg_alpha':
0.205264135431241, 'reg_lambda': 0.250353648178927, 'min_frequency':
0.0419107519503224}. Best is trial 4 with value: 2047.5548127632985.
[I 2024-10-25 00:01:07,653] Trial 8 finished with value: 3234.7413619084296 and
parameters: {'learning_rate': 0.036095296202756595, 'n_estimators': 410,
'max_depth': 3, 'max_leaves': 92, 'min_child_weight': 4, 'reg_alpha':
0.4333478448709589, 'reg_lambda': 0.028439240769341567, 'min_frequency':
0.0803157748634823}. Best is trial 4 with value: 2047.5548127632985.
[I 2024-10-25 00:01:08,056] Trial 9 finished with value: 5189.382202455456 and
parameters: {'learning_rate': 0.03454635768710781, 'n_estimators': 946,
'max_depth': 7, 'max_leaves': 2, 'min_child_weight': 1, 'reg_alpha':
0.8404796703370951, 'reg_lambda': 0.9860121256677118, 'min_frequency':
0.0014318526956674682}. Best is trial 4 with value: 2047.5548127632985.
[I 2024-10-25 00:01:09,114] Trial 10 finished with value: 2099.23233754742 and
parameters: {'learning_rate': 0.09752897058072088, 'n_estimators': 751,
'max_depth': 10, 'max_leaves': 51, 'min_child_weight': 5, 'reg_alpha':
0.09494403932831118, 'reg_lambda': 0.36420973654483846, 'min_frequency':
0.048757575174085084}. Best is trial 4 with value: 2047.5548127632985.
[I 2024-10-25 00:01:10,307] Trial 11 finished with value: 2090.497977775507 and
parameters: {'learning_rate': 0.09894980035488185, 'n_estimators': 712,
'max_depth': 10, 'max_leaves': 50, 'min_child_weight': 5, 'reg_alpha':
0.027098526843680706, 'reg_lambda': 0.3719157067096011, 'min_frequency':
0.04422746606806926}. Best is trial 4 with value: 2047.5548127632985.
[I 2024-10-25 00:01:11,579] Trial 12 finished with value: 2087.625618948063 and
parameters: {'learning_rate': 0.06304240138451461, 'n_estimators': 726,
'max_depth': 10, 'max_leaves': 54, 'min_child_weight': 5, 'reg_alpha':
0.03716948851806057, 'reg_lambda': 0.42539202378227947, 'min_frequency':
0.03467651843789368}. Best is trial 4 with value: 2047.5548127632985.
[I 2024-10-25 00:01:12,892] Trial 13 finished with value: 2094.444494276859 and
parameters: {'learning_rate': 0.06162272677696042, 'n_estimators': 680,
'max_depth': 9, 'max_leaves': 58, 'min_child_weight': 5, 'reg_alpha':
0.20630012425753685, 'reg_lambda': 0.4932236756076102, 'min_frequency':
0.022155179368849816}. Best is trial 4 with value: 2047.5548127632985.
[I 2024-10-25 00:01:14,684] Trial 14 finished with value: 2013.2740174772593 and
parameters: {'learning_rate': 0.06078824887047014, 'n_estimators': 834,
'max_depth': 9, 'max_leaves': 71, 'min_child_weight': 5, 'reg_alpha':
0.010347255180686177, 'reg_lambda': 0.1880939942697797, 'min_frequency':
0.0632408431696149}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:16,584] Trial 15 finished with value: 2069.8672038742875 and
parameters: {'learning_rate': 0.07972461920060392, 'n_estimators': 849,
'max_depth': 8, 'max_leaves': 71, 'min_child_weight': 4, 'reg_alpha':
```

```
0.214859839324815, 'reg_lambda': 0.18846511177560266, 'min_frequency':
0.06331898869703377}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:16,863] Trial 16 finished with value: 2890.475631785505 and
parameters: {'learning_rate': 0.05797465952207412, 'n_estimators': 101,
'max_depth': 8, 'max_leaves': 36, 'min_child_weight': 5, 'reg_alpha':
0.3597887999312945, 'reg_lambda': 0.22593477489971736, 'min_frequency':
0.06586043129439678}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:18,519] Trial 17 finished with value: 2173.1246463153925 and
parameters: {'learning_rate': 0.022483481096204842, 'n_estimators': 859,
'max_depth': 6, 'max_leaves': 99, 'min_child_weight': 4, 'reg_alpha':
0.14996174409103358, 'reg_lambda': 0.13921318098306112, 'min_frequency':
0.07348319920474022}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:20,701] Trial 18 finished with value: 2080.7864532573044 and
parameters: {'learning_rate': 0.05206168885533392, 'n_estimators': 995,
'max_depth': 9, 'max_leaves': 77, 'min_child_weight': 3, 'reg_alpha':
0.012014571072675544, 'reg_lambda': 0.34014426418468047, 'min_frequency':
0.027243683700063265}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:21,777] Trial 19 finished with value: 2092.0106922459236 and
parameters: {'learning_rate': 0.07618667919317465, 'n_estimators': 588,
'max_depth': 9, 'max_leaves': 67, 'min_child_weight': 5, 'reg_alpha':
0.3356785212498069, 'reg_lambda': 0.5628243803577642, 'min_frequency':
0.056490920652948334}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:22,478] Trial 20 finished with value: 2201.2054537511526 and
parameters: {'learning_rate': 0.04763236786396755, 'n_estimators': 409,
'max_depth': 7, 'max_leaves': 35, 'min_child_weight': 2, 'reg_alpha':
0.269601089415572, 'reg_lambda': 0.30654250573285996, 'min_frequency':
0.07109049262995354}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:24,038] Trial 21 finished with value: 2050.9011613627076 and
parameters: {'learning_rate': 0.08676060953926898, 'n_estimators': 842,
'max_depth': 8, 'max_leaves': 72, 'min_child_weight': 4, 'reg_alpha':
0.10989671893322668, 'reg_lambda': 0.18854953186516213, 'min_frequency':
0.05812802067130182}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:25,598] Trial 22 finished with value: 2070.7194835680752 and
parameters: {'learning_rate': 0.08498012112257783, 'n_estimators': 859,
'max_depth': 8, 'max_leaves': 75, 'min_child_weight': 4, 'reg_alpha':
0.15079454810488277, 'reg_lambda': 0.11695013464941133, 'min_frequency':
0.08732151213971048}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:26,805] Trial 23 finished with value: 2038.1525263233043 and
parameters: {'learning_rate': 0.06885670904316461, 'n_estimators': 641,
'max_depth': 9, 'max_leaves': 64, 'min_child_weight': 5, 'reg_alpha':
0.0903441623699152, 'reg_lambda': 0.26098644060960385, 'min_frequency':
0.05595135891066373}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:28,096] Trial 24 finished with value: 2072.5925122676163 and
parameters: {'learning_rate': 0.06896196519792967, 'n_estimators': 640,
'max_depth': 9, 'max_leaves': 61, 'min_child_weight': 5, 'reg_alpha':
0.07458712154317289, 'reg_lambda': 0.4510420456620375, 'min_frequency':
0.06862689988220685}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:29,225] Trial 25 finished with value: 2067.8929568622675 and
```

```
parameters: {'learning_rate': 0.05342677740874982, 'n_estimators': 778,
'max_depth': 9, 'max_leaves': 42, 'min_child_weight': 5, 'reg_alpha':
0.02299916500563104, 'reg_lambda': 0.008058131895408216, 'min_frequency':
0.05504230930392686}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:29,891] Trial 26 finished with value: 2514.9374867368274 and
parameters: {'learning_rate': 0.023127435491008562, 'n_estimators': 271,
'max_depth': 10, 'max_leaves': 79, 'min_child_weight': 5, 'reg_alpha':
0.5682879545829104, 'reg_lambda': 0.2511487121726603, 'min_frequency':
0.08659635536671426}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:30,614] Trial 27 finished with value: 2097.830910921257 and
parameters: {'learning_rate': 0.06778194558428928, 'n_estimators': 464,
'max_depth': 6, 'max_leaves': 45, 'min_child_weight': 4, 'reg_alpha':
0.15300848218901458, 'reg_lambda': 0.29288662447634695, 'min_frequency':
0.07435408789959394}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:31,739] Trial 28 finished with value: 2090.2869584040704 and
parameters: {'learning_rate': 0.04569013369004782, 'n_estimators': 647,
'max_depth': 9, 'max_leaves': 62, 'min_child_weight': 5, 'reg_alpha':
0.4101445772536411, 'reg_lambda': 0.14815466922886764, 'min_frequency':
0.04832968605615902}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:32,904] Trial 29 finished with value: 2063.4417027883446 and
parameters: {'learning_rate': 0.0697099255728873, 'n_estimators': 580,
'max_depth': 7, 'max_leaves': 65, 'min_child_weight': 4, 'reg_alpha':
0.26172172692238493, 'reg_lambda': 0.08122341251104859, 'min_frequency':
0.06398089590895474}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:34,773] Trial 30 finished with value: 2098.805869559807 and
parameters: {'learning_rate': 0.05391473547483721, 'n_estimators': 933,
'max_depth': 8, 'max_leaves': 96, 'min_child_weight': 5, 'reg_alpha':
0.1104632616714601, 'reg_lambda': 0.3953178457940947, 'min_frequency':
0.08597650405077473}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:36,251] Trial 31 finished with value: 2084.722813606902 and
parameters: {'learning_rate': 0.09090809042507597, 'n_estimators': 798,
'max_depth': 8, 'max_leaves': 70, 'min_child_weight': 4, 'reg_alpha':
0.09177248409085424, 'reg_lambda': 0.19481328470699236, 'min_frequency':
0.05964362724522408}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:38,048] Trial 32 finished with value: 2040.2209312188234 and
parameters: {'learning_rate': 0.08303045424825715, 'n_estimators': 898,
'max_depth': 8, 'max_leaves': 73, 'min_child_weight': 4, 'reg_alpha':
0.0014057927634517561, 'reg_lambda': 0.0851526807404833, 'min_frequency':
0.07709842927806872}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:39,955] Trial 33 finished with value: 2081.9477157541605 and
parameters: {'learning_rate': 0.07846165555098446, 'n_estimators': 903,
'max_depth': 9, 'max_leaves': 82, 'min_child_weight': 5, 'reg_alpha':
0.018895169739778922, 'reg_lambda': 0.06518903810717554, 'min_frequency':
0.08063129928447957}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:41,571] Trial 34 finished with value: 2100.8608350952068 and
parameters: {'learning_rate': 0.07437312917891195, 'n_estimators': 985,
'max_depth': 8, 'max_leaves': 57, 'min_child_weight': 4, 'reg_alpha':
0.9612875136022984, 'reg_lambda': 0.28867422808700394, 'min_frequency':
```

```
0.07430062367849069}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:43,276] Trial 35 finished with value: 2075.6492383663963 and
parameters: {'learning_rate': 0.06326389455770529, 'n_estimators': 906,
'max_depth': 7, 'max_leaves': 85, 'min_child_weight': 3, 'reg_alpha':
0.1697138755640851, 'reg_lambda': 0.10771942356476674, 'min_frequency':
0.07713872252463932}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:44,434] Trial 36 finished with value: 2076.319252428634 and
parameters: {'learning_rate': 0.08330759725961975, 'n_estimators': 802,
'max_depth': 6, 'max_leaves': 63, 'min_child_weight': 3, 'reg_alpha':
0.00392768765240952, 'reg_lambda': 0.5902125600080238, 'min_frequency':
0.0923031020536921}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:46,164] Trial 37 finished with value: 2084.6715692920543 and
parameters: {'learning_rate': 0.05751443536963845, 'n_estimators': 888,
'max_depth': 10, 'max_leaves': 76, 'min_child_weight': 5, 'reg_alpha':
0.26866416132708415, 'reg_lambda': 0.044103138283321644, 'min_frequency':
0.05244414682820604}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:47,601] Trial 38 finished with value: 2100.1789316628942 and
parameters: {'learning_rate': 0.029493067902779016, 'n_estimators': 762,
'max_depth': 9, 'max_leaves': 69, 'min_child_weight': 4, 'reg_alpha':
0.07240982137458853, 'reg_lambda': 0.6552705363529506, 'min_frequency':
0.038787737365860483}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:48,329] Trial 39 finished with value: 2320.7406219543827 and
parameters: {'learning_rate': 0.045760760206792137, 'n_estimators': 218,
'max_depth': 7, 'max_leaves': 88, 'min_child_weight': 3, 'reg_alpha':
0.7153883352294521, 'reg_lambda': 0.15745676917407145, 'min_frequency':
0.0682953934874843}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:49,611] Trial 40 finished with value: 2153.437894947812 and
parameters: {'learning_rate': 0.09256067173096365, 'n_estimators': 493,
'max_depth': 8, 'max_leaves': 82, 'min_child_weight': 5, 'reg_alpha':
0.5543624237768143, 'reg_lambda': 0.2503997458468144, 'min_frequency':
0.083493729043225}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:51,226] Trial 41 finished with value: 2036.8923312007462 and
parameters: {'learning_rate': 0.08844956548975329, 'n_estimators': 828,
'max_depth': 8, 'max_leaves': 73, 'min_child_weight': 4, 'reg_alpha':
0.11650607525006183, 'reg_lambda': 0.17160446318715133, 'min_frequency':
0.06149035421706409}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:52,962] Trial 42 finished with value: 2124.7298928663126 and
parameters: {'learning_rate': 0.040528073941074015, 'n_estimators': 825,
'max_depth': 9, 'max_leaves': 91, 'min_child_weight': 4, 'reg_alpha':
0.0675091892909365, 'reg_lambda': 0.08520000561756248, 'min_frequency':
0.06378451839106256}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:54,824] Trial 43 finished with value: 8287.698865065393 and
parameters: {'learning_rate': 0.001043668088638483, 'n_estimators': 704,
'max_depth': 8, 'max_leaves': 66, 'min_child_weight': 5, 'reg_alpha':
0.12828943110568528, 'reg_lambda': 0.19802224877662475, 'min_frequency':
0.07742988618621219}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:56,807] Trial 44 finished with value: 2097.4896092047074 and
parameters: {'learning_rate': 0.09222669136769046, 'n_estimators': 966,
'max_depth': 10, 'max_leaves': 74, 'min_child_weight': 4, 'reg_alpha':
0.20888594827832313, 'reg_lambda': 0.30181401002624975, 'min_frequency':
0.06027003150305813}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:58,277] Trial 45 finished with value: 2124.9825694073293 and
parameters: {'learning_rate': 0.07249824285938272, 'n_estimators': 919,
'max_depth': 7, 'max_leaves': 55, 'min_child_weight': 2, 'reg_alpha':
0.052892748275074286, 'reg_lambda': 0.24596694079616319, 'min_frequency':
0.04933585999109508}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:01:59,418] Trial 46 finished with value: 2075.6049554160904 and
parameters: {'learning_rate': 0.0798905638026098, 'n_estimators': 742,
'max_depth': 5, 'max_leaves': 81, 'min_child_weight': 5, 'reg_alpha':
0.17426112613128977, 'reg_lambda': 0.9577315978492751, 'min_frequency':
0.04553534234671429}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:02:00,253] Trial 47 finished with value: 2902.1703338520706 and
parameters: {'learning_rate': 0.012528072429466838, 'n_estimators': 640,
'max_depth': 8, 'max_leaves': 21, 'min_child_weight': 1, 'reg_alpha':
0.060500953343696415, 'reg_lambda': 0.3466765315351603, 'min_frequency':
0.09231807669614883}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:02:01,135] Trial 48 finished with value: 2163.0072678093297 and
parameters: {'learning_rate': 0.08820388975504426, 'n_estimators': 876,
'max_depth': 4, 'max_leaves': 60, 'min_child_weight': 3, 'reg_alpha':
0.24759994205346308, 'reg_lambda': 0.011255559866309411, 'min_frequency':
0.003263931027962505}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:02:02,475] Trial 49 finished with value: 2079.468321648642 and
parameters: {'learning_rate': 0.08252084278300062, 'n_estimators': 816,
'max_depth': 9, 'max_leaves': 47, 'min_child_weight': 5, 'reg_alpha':
0.1324728584980148, 'reg_lambda': 0.14847270829304368, 'min_frequency':
0.06887180205890286}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:02:03,688] Trial 50 finished with value: 2074.4195676991635 and
parameters: {'learning_rate': 0.09532199221904594, 'n_estimators': 690,
'max_depth': 10, 'max_leaves': 52, 'min_child_weight': 4, 'reg_alpha':
0.003088535586831098, 'reg_lambda': 0.4557111580405873, 'min_frequency':
0.0988693666447148}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:02:05,510] Trial 51 finished with value: 2082.774605969148 and
parameters: {'learning_rate': 0.08639571461179232, 'n_estimators': 827,
'max_depth': 8, 'max_leaves': 73, 'min_child_weight': 4, 'reg_alpha':
0.09775875161855428, 'reg_lambda': 0.2042987314578943, 'min_frequency':
0.053599439852325854}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:02:07,243] Trial 52 finished with value: 2067.269896232866 and
parameters: {'learning_rate': 0.06609029820354434, 'n_estimators': 945,
'max_depth': 8, 'max_leaves': 71, 'min_child_weight': 4, 'reg_alpha':
0.12231619521513841, 'reg_lambda': 0.16280871147637344, 'min_frequency':
0.06015944253176766}. Best is trial 14 with value: 2013.2740174772593.
[I 2024-10-25 00:02:08,882] Trial 53 finished with value: 2005.805549769974 and
parameters: {'learning_rate': 0.05920571817513656, 'n_estimators': 851,
'max_depth': 8, 'max_leaves': 67, 'min_child_weight': 5, 'reg_alpha':
0.053499146552752674, 'reg_lambda': 0.1140720685013228, 'min_frequency':
0.05721645251367464}. Best is trial 53 with value: 2005.805549769974.
```

```
[I 2024-10-25 00:02:10,675] Trial 54 finished with value: 2031.4389419197475 and
parameters: {'learning_rate': 0.05734150797136964, 'n_estimators': 999,
'max_depth': 7, 'max_leaves': 65, 'min_child_weight': 5, 'reg_alpha':
0.055909544294461586, 'reg_lambda': 0.11007029226493349, 'min_frequency':
0.04168489234442039}. Best is trial 53 with value: 2005.805549769974.
[I 2024-10-25 00:02:12,409] Trial 55 finished with value: 2056.970247592325 and
parameters: {'learning_rate': 0.058960099709298615, 'n_estimators': 989,
'max_depth': 7, 'max_leaves': 66, 'min_child_weight': 5, 'reg_alpha':
0.04572162505952951, 'reg_lambda': 0.11725968171216276, 'min_frequency':
0.03293572664767331}. Best is trial 53 with value: 2005.805549769974.
[I 2024-10-25 00:02:13,678] Trial 56 finished with value: 2046.2055285816461 and
parameters: {'learning_rate': 0.07239286765529607, 'n_estimators': 876,
'max_depth': 6, 'max_leaves': 59, 'min_child_weight': 5, 'reg_alpha':
0.189101621487213, 'reg_lambda': 0.05219750742109326, 'min_frequency':
0.03890955155320476}. Best is trial 53 with value: 2005.805549769974.
[I 2024-10-25 00:02:15,250] Trial 57 finished with value: 2057.049387798143 and
parameters: {'learning_rate': 0.04961210994849556, 'n_estimators': 773,
'max_depth': 7, 'max_leaves': 78, 'min_child_weight': 5, 'reg_alpha':
0.3038319370248432, 'reg_lambda': 0.08862692885880868, 'min_frequency':
0.04520535309941192}. Best is trial 53 with value: 2005.805549769974.
[I 2024-10-25 00:02:16,810] Trial 58 finished with value: 2224.650487560781 and
parameters: {'learning_rate': 0.06513605746426193, 'n_estimators': 540,
'max_depth': 9, 'max_leaves': 0, 'min_child_weight': 5, 'reg_alpha':
0.05320878307080972, 'reg_lambda': 0.22028057263322107, 'min_frequency':
0.05252758334827187}. Best is trial 53 with value: 2005.805549769974.
[I 2024-10-25 00:02:18,579] Trial 59 finished with value: 2021.887148738787 and
parameters: {'learning_rate': 0.06100612628795549, 'n_estimators': 953,
'max_depth': 8, 'max_leaves': 68, 'min_child_weight': 5, 'reg_alpha':
0.0016202872979955013, 'reg_lambda': 0.04182576736861712, 'min_frequency':
0.01770387884962845}. Best is trial 53 with value: 2005.805549769974.
[I 2024-10-25 00:02:20,333] Trial 60 finished with value: 2058.179913792652 and
parameters: {'learning_rate': 0.05623296087842478, 'n_estimators': 958,
'max_depth': 9, 'max_leaves': 55, 'min_child_weight': 5, 'reg_alpha':
0.8156893508967058, 'reg_lambda': 0.043613366537998285, 'min_frequency':
0.015952762239702278}. Best is trial 53 with value: 2005.805549769974.
[I 2024-10-25 00:02:22,836] Trial 61 finished with value: 1970.6179534891642 and
parameters: {'learning_rate': 0.061495828095354074, 'n_estimators': 926,
'max_depth': 8, 'max_leaves': 68, 'min_child_weight': 5, 'reg_alpha':
0.0028755461558962053, 'reg_lambda': 0.12085974287977863, 'min_frequency':
0.012681898035110887}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:24,928] Trial 62 finished with value: 2050.618957069238 and
parameters: {'learning_rate': 0.06021598864721086, 'n_estimators': 999,
'max_depth': 8, 'max_leaves': 68, 'min_child_weight': 5, 'reg_alpha':
0.0919577195695246, 'reg_lambda': 0.12119980873752306, 'min_frequency':
0.01232522839282887}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:27,268] Trial 63 finished with value: 2065.289267669821 and
parameters: {'learning_rate': 0.05482937545683258, 'n_estimators': 934,
'max_depth': 7, 'max_leaves': 64, 'min_child_weight': 5, 'reg_alpha':
```

```
0.036376897930156145, 'reg_lambda': 0.006427717047271048, 'min_frequency':
0.01256413168426102}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:29,313] Trial 64 finished with value: 2065.1422761831404 and
parameters: {'learning_rate': 0.061263638761574965, 'n_estimators': 861,
'max_depth': 9, 'max_leaves': 68, 'min_child_weight': 5, 'reg_alpha':
0.07986237158451401, 'reg_lambda': 0.15867678623353348, 'min_frequency':
0.0050828542431179645}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:32,278] Trial 65 finished with value: 2096.4025561245076 and
parameters: {'learning_rate': 0.05045151889178739, 'n_estimators': 965,
'max_depth': 8, 'max_leaves': 85, 'min_child_weight': 5, 'reg_alpha':
0.03614414203759003, 'reg_lambda': 0.12092427074637475, 'min_frequency':
0.024260945162873877}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:34,390] Trial 66 finished with value: 2051.6714536895856 and
parameters: {'learning_rate': 0.06408358657215124, 'n_estimators': 846,
'max_depth': 9, 'max_leaves': 78, 'min_child_weight': 5, 'reg_alpha':
0.22898978506885148, 'reg_lambda': 0.18245079195050465, 'min_frequency':
0.020259027160393146}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:35,951] Trial 67 finished with value: 2075.951144857426 and
parameters: {'learning_rate': 0.06938483603560774, 'n_estimators': 730,
'max_depth': 8, 'max_leaves': 63, 'min_child_weight': 5, 'reg_alpha':
0.1447689451678916, 'reg_lambda': 0.2287686572256394, 'min_frequency':
0.006216963543670219}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:37,429] Trial 68 finished with value: 2097.7436043670878 and
parameters: {'learning_rate': 0.04118981591445275, 'n_estimators': 791,
'max_depth': 7, 'max_leaves': 58, 'min_child_weight': 5, 'reg_alpha':
0.47147947870158124, 'reg_lambda': 0.2675684931006546, 'min_frequency':
0.008702843055024626}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:39,133] Trial 69 finished with value: 2053.705083856001 and
parameters: {'learning_rate': 0.061000081671373496, 'n_estimators': 918,
'max_depth': 6, 'max_leaves': 49, 'min_child_weight': 5, 'reg_alpha':
0.11787328968978993, 'reg_lambda': 0.060162280970820575, 'min_frequency':
0.05673432015392172}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:40,946] Trial 70 finished with value: 2083.992114634421 and
parameters: {'learning_rate': 0.051670127839651254, 'n_estimators': 606,
'max_depth': 8, 'max_leaves': 75, 'min_child_weight': 5, 'reg_alpha':
0.03038382822629273, 'reg_lambda': 0.7646405016538113, 'min_frequency':
0.02843760787890263}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:43,227] Trial 71 finished with value: 2042.9365219638141 and
parameters: {'learning_rate': 0.07459782510238727, 'n_estimators': 899,
'max_depth': 8, 'max_leaves': 72, 'min_child_weight': 5, 'reg_alpha':
0.004585933412959391, 'reg_lambda': 0.09350034775785422, 'min_frequency':
0.00028442939324703487}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:45,859] Trial 72 finished with value: 2082.2087020822855 and
parameters: {'learning_rate': 0.07790896807219064, 'n_estimators': 885,
'max_depth': 8, 'max_leaves': 70, 'min_child_weight': 4, 'reg_alpha':
0.0005393945321612442, 'reg_lambda': 0.027837382279932443, 'min_frequency':
0.06639207668805826}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:47,612] Trial 73 finished with value: 2076.9247052955757 and
```

```
parameters: {'learning_rate': 0.06778125570064475, 'n_estimators': 977,
'max_depth': 9, 'max_leaves': 61, 'min_child_weight': 5, 'reg_alpha':
0.07788648295828593, 'reg_lambda': 0.12632915505217057, 'min_frequency':
0.061707187693751345}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:48,502] Trial 74 finished with value: 2120.0055291056233 and
parameters: {'learning_rate': 0.056284539179410446, 'n_estimators': 302,
'max_depth': 7, 'max_leaves': 66, 'min_child_weight': 5, 'reg_alpha':
0.037112907179167126, 'reg_lambda': 0.17178156692818175, 'min_frequency':
0.020064129489600083}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:50,563] Trial 75 finished with value: 2039.9389632063212 and
parameters: {'learning_rate': 0.07107960198635047, 'n_estimators': 939,
'max_depth': 9, 'max_leaves': 81, 'min_child_weight': 4, 'reg_alpha':
0.09887664424917646, 'reg_lambda': 0.07234689693016147, 'min_frequency':
0.07086790831792736}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:52,889] Trial 76 finished with value: 2047.1362031236902 and
parameters: {'learning_rate': 0.07116555877573598, 'n_estimators': 949,
'max_depth': 10, 'max_leaves': 79, 'min_child_weight': 5, 'reg_alpha':
0.1648866477813876, 'reg_lambda': 0.06972377185888409, 'min_frequency':
0.07085194158784519}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:55,297] Trial 77 finished with value: 2068.183724744299 and
parameters: {'learning_rate': 0.065785733638906, 'n_estimators': 926,
'max_depth': 9, 'max_leaves': 87, 'min_child_weight': 4, 'reg_alpha':
0.1065776113796004, 'reg_lambda': 0.033151273126808156, 'min_frequency':
0.05726771276196584}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:57,765] Trial 78 finished with value: 2140.2757279353204 and
parameters: {'learning_rate': 0.05818317486500565, 'n_estimators': 838,
'max_depth': 9, 'max_leaves': 81, 'min_child_weight': 3, 'reg_alpha':
0.6335554008344553, 'reg_lambda': 0.10690771131510583, 'min_frequency':
0.06476705033982864}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:02:59,590] Trial 79 finished with value: 2108.3954968417274 and
parameters: {'learning_rate': 0.06269085591501114, 'n_estimators': 669,
'max_depth': 10, 'max_leaves': 92, 'min_child_weight': 2, 'reg_alpha':
0.06360935051753873, 'reg_lambda': 0.13839655779630236, 'min_frequency':
0.015927962808423957}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:01,779] Trial 80 finished with value: 2049.649355115065 and
parameters: {'learning_rate': 0.04570270177777472, 'n_estimators': 871,
'max_depth': 8, 'max_leaves': 64, 'min_child_weight': 5, 'reg_alpha':
0.39509556061607914, 'reg_lambda': 0.3277082063951113, 'min_frequency':
0.05140702199011529}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:04,703] Trial 81 finished with value: 2082.8869091829624 and
parameters: {'learning_rate': 0.08917339930275274, 'n_estimators': 908,
'max_depth': 8, 'max_leaves': 74, 'min_child_weight': 4, 'reg_alpha':
0.02674587987968853, 'reg_lambda': 0.07497687219140536, 'min_frequency':
0.07137220921273964}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:08,160] Trial 82 finished with value: 2011.6945406160924 and
parameters: {'learning_rate': 0.08214814824829593, 'n_estimators': 964,
'max_depth': 9, 'max_leaves': 76, 'min_child_weight': 4, 'reg_alpha':
0.09383861629748964, 'reg_lambda': 0.0005271336070599014, 'min_frequency':
```

```
0.07587146880780173}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:11,311] Trial 83 finished with value: 2093.9970775171864 and
parameters: {'learning_rate': 0.0997511626188342, 'n_estimators': 1000,
'max_depth': 9, 'max_leaves': 83, 'min_child_weight': 4, 'reg_alpha':
0.08888012466585599, 'reg_lambda': 0.001747833658771449, 'min_frequency':
0.05517508107530304}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:14,080] Trial 84 finished with value: 2007.9753602998198 and
parameters: {'learning_rate': 0.08034376746101944, 'n_estimators': 966,
'max_depth': 9, 'max_leaves': 69, 'min_child_weight': 4, 'reg_alpha':
0.188409518677491, 'reg_lambda': 0.5195462614317472, 'min_frequency':
0.06212067893846268}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:15,961] Trial 85 finished with value: 2069.0236122136466 and
parameters: {'learning_rate': 0.05353971649862247, 'n_estimators': 973,
'max_depth': 9, 'max_leaves': 68, 'min_child_weight': 5, 'reg_alpha':
0.18319290962564316, 'reg_lambda': 0.5501928915059621, 'min_frequency':
0.06204794690522615}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:18,019] Trial 86 finished with value: 2072.6448084666854 and
parameters: {'learning_rate': 0.09436803385777091, 'n_estimators': 952,
'max_depth': 10, 'max_leaves': 77, 'min_child_weight': 4, 'reg_alpha':
0.15181301951063242, 'reg_lambda': 0.520259208380021, 'min_frequency':
0.04782428763181995}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:19,596] Trial 87 finished with value: 2003.5216098020414 and
parameters: {'learning_rate': 0.08172782714241941, 'n_estimators': 802,
'max_depth': 9, 'max_leaves': 70, 'min_child_weight': 5, 'reg_alpha':
0.1367805803297798, 'reg_lambda': 0.4068071181918589, 'min_frequency':
0.04107821695872244}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:21,220] Trial 88 finished with value: 2088.9522905663143 and
parameters: {'learning_rate': 0.08489357552565256, 'n_estimators': 801,
'max_depth': 8, 'max_leaves': 72, 'min_child_weight': 3, 'reg_alpha':
0.1347040880411583, 'reg_lambda': 0.3853155071724766, 'min_frequency':
0.04167860955523832}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:21,519] Trial 89 finished with value: 2650.3926647449803 and
parameters: {'learning_rate': 0.07987987747620151, 'n_estimators': 69,
'max_depth': 9, 'max_leaves': 70, 'min_child_weight': 5, 'reg_alpha':
0.056309358998134273, 'reg_lambda': 0.6182813773653215, 'min_frequency':
0.04156194248176838}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:23,234] Trial 90 finished with value: 2072.754109291164 and
parameters: {'learning_rate': 0.07653159515383584, 'n_estimators': 851,
'max_depth': 9, 'max_leaves': 61, 'min_child_weight': 4, 'reg_alpha':
0.198567599145839, 'reg_lambda': 0.516228583696726, 'min_frequency':
0.03347982416879801}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:24,790] Trial 91 finished with value: 2050.740860691493 and
parameters: {'learning_rate': 0.08085143256507947, 'n_estimators': 750,
'max_depth': 9, 'max_leaves': 76, 'min_child_weight': 5, 'reg_alpha':
0.11614258700224321, 'reg_lambda': 0.46911599948504945, 'min_frequency':
0.05900010357183616}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:26,518] Trial 92 finished with value: 2048.6509303215125 and
parameters: {'learning_rate': 0.07569790631399445, 'n_estimators': 888,
'max_depth': 10, 'max_leaves': 67, 'min_child_weight': 5, 'reg_alpha':
0.06312713241677662, 'reg_lambda': 0.22589436973549407, 'min_frequency':
0.03756631978003868}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:27,507] Trial 93 finished with value: 2094.8511413533283 and
parameters: {'learning_rate': 0.06713882482756428, 'n_estimators': 443,
'max_depth': 8, 'max_leaves': 57, 'min_child_weight': 5, 'reg_alpha':
0.029774524515541176, 'reg_lambda': 0.4820083231486471, 'min_frequency':
0.06656140502516951}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:29,201] Trial 94 finished with value: 2045.8276228006057 and
parameters: {'learning_rate': 0.08934404652873101, 'n_estimators': 918,
'max_depth': 9, 'max_leaves': 64, 'min_child_weight': 5, 'reg_alpha':
0.22999324473967417, 'reg_lambda': 0.4429320323545977, 'min_frequency':
0.04736258718374027}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:30,813] Trial 95 finished with value: 2054.673676335356 and
parameters: {'learning_rate': 0.05924243742199553, 'n_estimators': 822,
'max_depth': 9, 'max_leaves': 70, 'min_child_weight': 5, 'reg_alpha':
0.0786715900652565, 'reg_lambda': 0.20589758310531578, 'min_frequency':
0.0548670744144701}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:31,790] Trial 96 finished with value: 2185.533838283608 and
parameters: {'learning_rate': 0.08212119866030823, 'n_estimators': 976,
'max_depth': 4, 'max_leaves': 74, 'min_child_weight': 5, 'reg_alpha':
0.13741649131059575, 'reg_lambda': 0.02990907397930659, 'min_frequency':
0.061168375876213814}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:33,338] Trial 97 finished with value: 2046.7743513489793 and
parameters: {'learning_rate': 0.08474247620548242, 'n_estimators': 765,
'max_depth': 8, 'max_leaves': 66, 'min_child_weight': 5, 'reg_alpha':
0.16316523031007668, 'reg_lambda': 0.2769175496109073, 'min_frequency':
0.05115140998454875}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:35,104] Trial 98 finished with value: 2033.8970761417463 and
parameters: {'learning_rate': 0.06427509500615579, 'n_estimators': 940,
'max_depth': 9, 'max_leaves': 62, 'min_child_weight': 5, 'reg_alpha':
0.013016164642016242, 'reg_lambda': 0.4254738782680608, 'min_frequency':
0.08903657708199819}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:35,873] Trial 99 finished with value: 2604.5059053867476 and
parameters: {'learning_rate': 0.047911100204193084, 'n_estimators': 934,
'max_depth': 3, 'max_leaves': 72, 'min_child_weight': 5, 'reg_alpha':
0.017706890999950216, 'reg_lambda': 0.6660721293678658, 'min_frequency':
0.09073084242160306}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:37,457] Trial 100 finished with value: 2059.8374008700644
and parameters: {'learning_rate': 0.054797433410816665, 'n_estimators': 965,
'max_depth': 7, 'max_leaves': 52, 'min_child_weight': 3, 'reg_alpha':
0.04828840674744923, 'reg_lambda': 0.4266508504477807, 'min_frequency':
0.0290336398098436}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:39,235] Trial 101 finished with value: 2025.503943910861 and
parameters: {'learning_rate': 0.06409905350973653, 'n_estimators': 860,
'max_depth': 9, 'max_leaves': 62, 'min_child_weight': 5, 'reg_alpha':
0.10403508077551524, 'reg_lambda': 0.41913428580020295, 'min_frequency':
0.07950879355126794}. Best is trial 61 with value: 1970.6179534891642.
```

```
[I 2024-10-25 00:03:40,816] Trial 102 finished with value: 2059.952371619037 and
parameters: {'learning_rate': 0.06399964825202757, 'n_estimators': 864,
'max_depth': 9, 'max_leaves': 60, 'min_child_weight': 5, 'reg_alpha':
0.10666998229260977, 'reg_lambda': 0.40627275583702815, 'min_frequency':
0.08124313404941537}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:41,576] Trial 103 finished with value: 2524.666180514074 and
parameters: {'learning_rate': 0.06230786133464204, 'n_estimators': 906,
'max_depth': 9, 'max_leaves': 8, 'min_child_weight': 5, 'reg_alpha':
0.013239061185324199, 'reg_lambda': 0.42041613856479265, 'min_frequency':
0.08447286418446118}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:43,145] Trial 104 finished with value: 2060.8967370302544
and parameters: {'learning_rate': 0.05791060307295621, 'n_estimators': 785,
'max_depth': 10, 'max_leaves': 62, 'min_child_weight': 5, 'reg_alpha':
0.07344526545390456, 'reg_lambda': 0.35882074173009276, 'min_frequency':
0.09642007326361669}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:44,833] Trial 105 finished with value: 2044.0738100477868
and parameters: {'learning_rate': 0.0870077815100972, 'n_estimators': 890,
'max_depth': 9, 'max_leaves': 69, 'min_child_weight': 5, 'reg_alpha':
0.04793564749972146, 'reg_lambda': 0.5397218909809577, 'min_frequency':
0.07627413344804787}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:46,295] Trial 106 finished with value: 2060.598229350059 and
parameters: {'learning_rate': 0.06446129967335297, 'n_estimators': 834,
'max_depth': 8, 'max_leaves': 56, 'min_child_weight': 5, 'reg_alpha':
0.9984848937399021, 'reg_lambda': 0.5900316119019189, 'min_frequency':
0.08190238706602937}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:48,151] Trial 107 finished with value: 2079.2991168036865
and parameters: {'learning_rate': 0.05635546969740924, 'n_estimators': 986,
'max_depth': 9, 'max_leaves': 65, 'min_child_weight': 4, 'reg_alpha':
0.3039521293098436, 'reg_lambda': 0.48890942574978064, 'min_frequency':
0.010098339797060641}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:50,046] Trial 108 finished with value: 2069.9471284084716
and parameters: {'learning_rate': 0.060771927590834544, 'n_estimators': 951,
'max_depth': 8, 'max_leaves': 76, 'min_child_weight': 5, 'reg_alpha':
0.021232407408783054, 'reg_lambda': 0.37179183212141104, 'min_frequency':
0.09034765297599873}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:51,854] Trial 109 finished with value: 2114.295422600708 and
parameters: {'learning_rate': 0.05243903291294706, 'n_estimators': 929,
'max_depth': 9, 'max_leaves': 68, 'min_child_weight': 1, 'reg_alpha':
0.1213006044032152, 'reg_lambda': 0.10183139621072744, 'min_frequency':
0.0680417069946591}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:53,629] Trial 110 finished with value: 2072.0011249135437
and parameters: {'learning_rate': 0.06011167928454076, 'n_estimators': 860,
'max_depth': 9, 'max_leaves': 73, 'min_child_weight': 4, 'reg_alpha':
0.09179799576684117, 'reg_lambda': 0.31660916673397754, 'min_frequency':
0.0793592369663846}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:55,179] Trial 111 finished with value: 2088.246574171592 and
parameters: {'learning_rate': 0.0693998548196981, 'n_estimators': 812,
'max_depth': 9, 'max_leaves': 63, 'min_child_weight': 5, 'reg_alpha':
```

```
0.06600132963853524, 'reg_lambda': 0.13678746000539094, 'min_frequency':
0.09493933710627277}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:55,895] Trial 112 finished with value: 2149.1048509481366
and parameters: {'learning_rate': 0.06734861829225551, 'n_estimators': 354,
'max_depth': 8, 'max_leaves': 58, 'min_child_weight': 5, 'reg_alpha':
0.0017300989662657207, 'reg_lambda': 0.17713391698573613, 'min_frequency':
0.07336224215790713}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:57,573] Trial 113 finished with value: 1983.2214140965584
and parameters: {'learning_rate': 0.07361270915961372, 'n_estimators': 876,
'max_depth': 9, 'max_leaves': 71, 'min_child_weight': 5, 'reg_alpha':
0.0438216929178959, 'reg_lambda': 0.15065483753430686, 'min_frequency':
0.0632960003978513}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:03:59,386] Trial 114 finished with value: 2057.3379641455504
and parameters: {'learning_rate': 0.07801952864044674, 'n_estimators': 877,
'max_depth': 10, 'max_leaves': 70, 'min_child_weight': 5, 'reg_alpha':
0.03909195613412718, 'reg_lambda': 0.15319325255730432, 'min_frequency':
0.06329414770716202}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:01,261] Trial 115 finished with value: 2037.4912124111859
and parameters: {'learning_rate': 0.07337717491400673, 'n_estimators': 911,
'max_depth': 9, 'max_leaves': 67, 'min_child_weight': 5, 'reg_alpha':
0.08418424435597573, 'reg_lambda': 0.019566522227984565, 'min_frequency':
0.058277535024753524}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:02,890] Trial 116 finished with value: 2015.4953939129568
and parameters: {'learning_rate': 0.08308472826662913, 'n_estimators': 847,
'max_depth': 8, 'max_leaves': 71, 'min_child_weight': 5, 'reg_alpha':
0.05384736414778812, 'reg_lambda': 0.5106457948428818, 'min_frequency':
0.016142277225904868}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:04,703] Trial 117 finished with value: 2028.1985652521903
and parameters: {'learning_rate': 0.08121882810456951, 'n_estimators': 895,
'max_depth': 9, 'max_leaves': 71, 'min_child_weight': 5, 'reg_alpha':
0.9297899607768404, 'reg_lambda': 0.46497502796153384, 'min_frequency':
0.024866521634506422}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:06,179] Trial 118 finished with value: 2020.615592808151 and
parameters: {'learning_rate': 0.08229560754575453, 'n_estimators': 854,
'max_depth': 7, 'max_leaves': 75, 'min_child_weight': 5, 'reg_alpha':
0.488699118443425, 'reg_lambda': 0.5187500449249345, 'min_frequency':
0.01681579411799872}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:07,940] Trial 119 finished with value: 2095.1350428416854
and parameters: {'learning_rate': 0.08277579069435356, 'n_estimators': 843,
'max_depth': 9, 'max_leaves': 79, 'min_child_weight': 5, 'reg_alpha':
0.6546894816095139, 'reg_lambda': 0.5254327628251266, 'min_frequency':
0.018553485020101367}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:09,748] Trial 120 finished with value: 2073.248364863661 and
parameters: {'learning_rate': 0.07656261645184118, 'n_estimators': 893,
'max_depth': 8, 'max_leaves': 75, 'min_child_weight': 5, 'reg_alpha':
0.9279011447689944, 'reg_lambda': 0.5961209360200419, 'min_frequency':
0.023984441176938577}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:11,612] Trial 121 finished with value: 2046.265506941388 and
```

```
parameters: {'learning_rate': 0.07980136078499143, 'n_estimators': 806,
'max_depth': 7, 'max_leaves': 71, 'min_child_weight': 5, 'reg_alpha':
0.5258687029357801, 'reg_lambda': 0.568817192023015, 'min_frequency':
0.015770104066562735}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:12,877] Trial 122 finished with value: 2065.9923708920187
and parameters: {'learning_rate': 0.08538418675827829, 'n_estimators': 863,
'max_depth': 6, 'max_leaves': 77, 'min_child_weight': 5, 'reg_alpha':
0.725227875897911, 'reg_lambda': 0.504117220410822, 'min_frequency':
0.01820162322610266}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:14,467] Trial 123 finished with value: 2046.990127614646 and
parameters: {'learning_rate': 0.08207065026208499, 'n_estimators': 960,
'max_depth': 7, 'max_leaves': 40, 'min_child_weight': 5, 'reg_alpha':
0.7688726527857297, 'reg_lambda': 0.5017521994281223, 'min_frequency':
0.025290966202616345}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:15,620] Trial 124 finished with value: 2068.5409019874455
and parameters: {'learning_rate': 0.07895183566443877, 'n_estimators': 924,
'max_depth': 8, 'max_leaves': 26, 'min_child_weight': 5, 'reg_alpha':
0.6098870778765884, 'reg_lambda': 0.4647897152162822, 'min_frequency':
0.013404998302473499}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:17,200] Trial 125 finished with value: 2053.7675283471663
and parameters: {'learning_rate': 0.09138656902053804, 'n_estimators': 881,
'max_depth': 7, 'max_leaves': 72, 'min_child_weight': 5, 'reg_alpha':
0.36779971594684585, 'reg_lambda': 0.5640811402263534, 'min_frequency':
0.013823164942221553}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:19,321] Trial 126 finished with value: 2107.34568773317 and
parameters: {'learning_rate': 0.08390690769865532, 'n_estimators': 1000,
'max_depth': 9, 'max_leaves': 69, 'min_child_weight': 2, 'reg_alpha':
0.8424298480571469, 'reg_lambda': 0.05308335421595526, 'min_frequency':
0.009715697269746373}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:20,666] Trial 127 finished with value: 2078.817907444668 and
parameters: {'learning_rate': 0.07480617835641465, 'n_estimators': 841,
'max_depth': 6, 'max_leaves': 66, 'min_child_weight': 5, 'reg_alpha':
0.043419314209764616, 'reg_lambda': 0.4818236492601326, 'min_frequency':
0.02179124042759203}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:21,202] Trial 128 finished with value: 2234.9575951804577
and parameters: {'learning_rate': 0.0711168879572634, 'n_estimators': 176,
'max_depth': 7, 'max_leaves': 80, 'min_child_weight': 5, 'reg_alpha':
0.673565424403988, 'reg_lambda': 0.4403765766976068, 'min_frequency':
0.015532534773897374}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:23,088] Trial 129 finished with value: 2095.212446390583 and
parameters: {'learning_rate': 0.08102883273891183, 'n_estimators': 903,
'max_depth': 8, 'max_leaves': 84, 'min_child_weight': 5, 'reg_alpha':
0.8646136025933228, 'reg_lambda': 0.9126018604288602, 'min_frequency':
0.03675087700907256}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:24,855] Trial 130 finished with value: 2065.863373600981 and
parameters: {'learning_rate': 0.08692797588061418, 'n_estimators': 820,
'max_depth': 9, 'max_leaves': 73, 'min_child_weight': 5, 'reg_alpha':
0.48441173528267145, 'reg_lambda': 0.09335091726400319, 'min_frequency':
```

```
0.007944462610349284}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:26,668] Trial 131 finished with value: 2011.8446712174086
and parameters: {'learning_rate': 0.06191780533003459, 'n_estimators': 942,
'max_depth': 9, 'max_leaves': 65, 'min_child_weight': 5, 'reg_alpha':
0.03132920923713043, 'reg_lambda': 0.5432843929580629, 'min_frequency':
0.011248376320179217}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:28,564] Trial 132 finished with value: 2055.704653375985 and
parameters: {'learning_rate': 0.06208816942082162, 'n_estimators': 978,
'max_depth': 9, 'max_leaves': 68, 'min_child_weight': 5, 'reg_alpha':
0.05787923600925608, 'reg_lambda': 0.5395803013761313, 'min_frequency':
0.011262026274281176}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:29,801] Trial 133 finished with value: 2033.5443862720595
and parameters: {'learning_rate': 0.07724246923455408, 'n_estimators': 953,
'max_depth': 5, 'max_leaves': 75, 'min_child_weight': 5, 'reg_alpha':
0.022608805621352146, 'reg_lambda': 0.5112003633392534, 'min_frequency':
0.0030103831348469168}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:31,565] Trial 134 finished with value: 2056.839585527488 and
parameters: {'learning_rate': 0.0571717881346086, 'n_estimators': 923,
'max_depth': 9, 'max_leaves': 65, 'min_child_weight': 5, 'reg_alpha':
0.09118531934808713, 'reg_lambda': 0.46339921666902234, 'min_frequency':
0.017931040343650978}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:33,305] Trial 135 finished with value: 2023.0223219198 and
parameters: {'learning_rate': 0.059427303300144084, 'n_estimators': 870,
'max_depth': 9, 'max_leaves': 70, 'min_child_weight': 5, 'reg_alpha':
0.45442141424418764, 'reg_lambda': 0.396856191561002, 'min_frequency':
0.031115704925179698}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:35,104] Trial 136 finished with value: 2043.3907790820444
and parameters: {'learning_rate': 0.0664870394622387, 'n_estimators': 856,
'max_depth': 9, 'max_leaves': 71, 'min_child_weight': 5, 'reg_alpha':
0.4291487155180525, 'reg_lambda': 0.38626067603924574, 'min_frequency':
0.026256537770381026}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:36,967] Trial 137 finished with value: 2069.6590154533974
and parameters: {'learning_rate': 0.0728959784110833, 'n_estimators': 879,
'max_depth': 9, 'max_leaves': 77, 'min_child_weight': 5, 'reg_alpha':
0.5849719429368432, 'reg_lambda': 0.40243334689439475, 'min_frequency':
0.02998218519650774}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:38,555] Trial 138 finished with value: 2050.4054581394616
and parameters: {'learning_rate': 0.059786075423683784, 'n_estimators': 784,
'max_depth': 10, 'max_leaves': 70, 'min_child_weight': 5, 'reg_alpha':
0.5230183377432162, 'reg_lambda': 0.5556941459890073, 'min_frequency':
0.023506502957227664}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:40,472] Trial 139 finished with value: 2026.832103788093 and
parameters: {'learning_rate': 0.08182045945457296, 'n_estimators': 900,
'max_depth': 9, 'max_leaves': 74, 'min_child_weight': 5, 'reg_alpha':
0.446114669058096, 'reg_lambda': 0.4492963763330353, 'min_frequency':
0.02164044628836572}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:42,314] Trial 140 finished with value: 2011.3627752845196
and parameters: {'learning_rate': 0.08992115231985963, 'n_estimators': 833,
'max_depth': 10, 'max_leaves': 74, 'min_child_weight': 5, 'reg_alpha':
0.5163331970532105, 'reg_lambda': 0.4892677468232907, 'min_frequency':
0.030657385415713626}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:44,033] Trial 141 finished with value: 2077.300263986261 and
parameters: {'learning_rate': 0.09014117989157315, 'n_estimators': 834,
'max_depth': 10, 'max_leaves': 74, 'min_child_weight': 5, 'reg_alpha':
0.44887040227532404, 'reg_lambda': 0.529405708970526, 'min_frequency':
0.021213397974163342}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:45,998] Trial 142 finished with value: 2111.374376630879 and
parameters: {'learning_rate': 0.09360028922635887, 'n_estimators': 871,
'max_depth': 10, 'max_leaves': 78, 'min_child_weight': 5, 'reg_alpha':
0.5129939738730854, 'reg_lambda': 0.6275912473778413, 'min_frequency':
0.00712990318127665}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:47,699] Trial 143 finished with value: 2033.6074690329476
and parameters: {'learning_rate': 0.09616962303089928, 'n_estimators': 848,
'max_depth': 9, 'max_leaves': 68, 'min_child_weight': 5, 'reg_alpha':
0.4466978877080218, 'reg_lambda': 0.4884529640809426, 'min_frequency':
0.029905459750041212}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:49,357] Trial 144 finished with value: 2074.496666195087 and
parameters: {'learning_rate': 0.08773420647616272, 'n_estimators': 813,
'max_depth': 10, 'max_leaves': 73, 'min_child_weight': 5, 'reg_alpha':
0.5458386897032252, 'reg_lambda': 0.5793924257048714, 'min_frequency':
0.013076070571529224}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:51,398] Trial 145 finished with value: 2054.438650293689 and
parameters: {'learning_rate': 0.08393788911852372, 'n_estimators': 912,
'max_depth': 9, 'max_leaves': 75, 'min_child_weight': 5, 'reg_alpha':
0.48385240522344714, 'reg_lambda': 0.3526250649118964, 'min_frequency':
0.031781579082554294}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:53,055] Trial 146 finished with value: 2038.2205659084716
and parameters: {'learning_rate': 0.06290463044130694, 'n_estimators': 796,
'max_depth': 8, 'max_leaves': 67, 'min_child_weight': 5, 'reg_alpha':
0.4559129832738357, 'reg_lambda': 0.4433330509163934, 'min_frequency':
0.017717552243599253}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:54,838] Trial 147 finished with value: 2049.193673270089 and
parameters: {'learning_rate': 0.08592110784868781, 'n_estimators': 944,
'max_depth': 9, 'max_leaves': 64, 'min_child_weight': 5, 'reg_alpha':
0.5496431697103313, 'reg_lambda': 0.4228664717527459, 'min_frequency':
0.019691197139658383}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:56,626] Trial 148 finished with value: 2079.8673633598464
and parameters: {'learning_rate': 0.07852746033085156, 'n_estimators': 869,
'max_depth': 8, 'max_leaves': 69, 'min_child_weight': 5, 'reg_alpha':
0.5023837324537712, 'reg_lambda': 0.5068098537604778, 'min_frequency':
0.014905968558424508}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:04:58,369] Trial 149 finished with value: 2038.9638745559294
and parameters: {'learning_rate': 0.06561117851619029, 'n_estimators': 895,
'max_depth': 9, 'max_leaves': 72, 'min_child_weight': 5, 'reg_alpha':
0.03317793855280632, 'reg_lambda': 0.5490105282692408, 'min_frequency':
0.010834873663702001}. Best is trial 61 with value: 1970.6179534891642.
```

```
[I 2024-10-25 00:05:00,310] Trial 150 finished with value: 2068.850729703743 and
parameters: {'learning_rate': 0.059037970803922604, 'n_estimators': 932,
'max_depth': 10, 'max_leaves': 80, 'min_child_weight': 5, 'reg_alpha':
0.5727016620237566, 'reg_lambda': 0.00017789411132262148, 'min_frequency':
0.04322031356261356}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:02,329] Trial 151 finished with value: 2047.2227556419246
and parameters: {'learning_rate': 0.0821616849394541, 'n_estimators': 896,
'max_depth': 9, 'max_leaves': 71, 'min_child_weight': 5, 'reg_alpha':
0.39195779746289827, 'reg_lambda': 0.4571471214852356, 'min_frequency':
0.03494332321745657}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:04,179] Trial 152 finished with value: 2051.030742560834 and
parameters: {'learning_rate': 0.08125147099436791, 'n_estimators': 848,
'max_depth': 9, 'max_leaves': 76, 'min_child_weight': 5, 'reg_alpha':
0.3417994762706972, 'reg_lambda': 0.46744522668742006, 'min_frequency':
0.023462621465896585}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:05,754] Trial 153 finished with value: 2028.1224280906795
and parameters: {'learning_rate': 0.08024690434075656, 'n_estimators': 829,
'max_depth': 9, 'max_leaves': 66, 'min_child_weight': 5, 'reg_alpha':
0.4070536573811909, 'reg_lambda': 0.41287062933947516, 'min_frequency':
0.02210808626623471}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:07,524] Trial 154 finished with value: 2116.471118376928 and
parameters: {'learning_rate': 0.054972109524036396, 'n_estimators': 769,
'max_depth': 9, 'max_leaves': 67, 'min_child_weight': 5, 'reg_alpha':
0.4065486259743252, 'reg_lambda': 0.3982290126319703, 'min_frequency':
0.0170821644123142}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:09,081] Trial 155 finished with value: 2049.779991308528 and
parameters: {'learning_rate': 0.07504396675873273, 'n_estimators': 823,
'max_depth': 8, 'max_leaves': 65, 'min_child_weight': 5, 'reg_alpha':
0.43084976740186537, 'reg_lambda': 0.41606973956596754, 'min_frequency':
0.021383040864810847}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:10,591] Trial 156 finished with value: 2045.2331315366155
and parameters: {'learning_rate': 0.08333772934213676, 'n_estimators': 833,
'max_depth': 9, 'max_leaves': 60, 'min_child_weight': 5, 'reg_alpha':
0.4573982423108956, 'reg_lambda': 0.3836740299178421, 'min_frequency':
0.019631081004952602}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:12,465] Trial 157 finished with value: 2055.9153547587607
and parameters: {'learning_rate': 0.07905339316001868, 'n_estimators': 859,
'max_depth': 9, 'max_leaves': 70, 'min_child_weight': 5, 'reg_alpha':
0.4975932828303857, 'reg_lambda': 0.33812268846262156, 'min_frequency':
0.06624707698298857}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:14,122] Trial 158 finished with value: 2036.1015628274858
and parameters: {'learning_rate': 0.06172079063793027, 'n_estimators': 912,
'max_depth': 9, 'max_leaves': 63, 'min_child_weight': 5, 'reg_alpha':
0.06651788919649819, 'reg_lambda': 0.04515564161700923, 'min_frequency':
0.07615387968144241}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:15,716] Trial 159 finished with value: 2047.4309547061011
and parameters: {'learning_rate': 0.08528737205266389, 'n_estimators': 797,
'max_depth': 8, 'max_leaves': 74, 'min_child_weight': 5, 'reg_alpha':
```

```
0.0029583998789206797, 'reg_lambda': 0.44267074793392336, 'min_frequency':
0.06294075342956014}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:17,660] Trial 160 finished with value: 2042.2408127148306
and parameters: {'learning_rate': 0.06971740351754278, 'n_estimators': 967,
'max_depth': 9, 'max_leaves': 69, 'min_child_weight': 5, 'reg_alpha':
0.4756869359732519, 'reg_lambda': 0.5282690182661923, 'min_frequency':
0.01425903036188756}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:19,445] Trial 161 finished with value: 2065.72868428016 and
parameters: {'learning_rate': 0.08129980699876564, 'n_estimators': 884,
'max_depth': 9, 'max_leaves': 72, 'min_child_weight': 5, 'reg_alpha':
0.41367858222241005, 'reg_lambda': 0.4936777484720258, 'min_frequency':
0.027246922045142542}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:21,111] Trial 162 finished with value: 2006.6069263890636
and parameters: {'learning_rate': 0.07988736122527137, 'n_estimators': 868,
'max_depth': 9, 'max_leaves': 66, 'min_child_weight': 5, 'reg_alpha':
0.3662360910450958, 'reg_lambda': 0.4858826345695933, 'min_frequency':
0.026927927665699805}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:22,867] Trial 163 finished with value: 2043.817510040713 and
parameters: {'learning_rate': 0.07766858093720042, 'n_estimators': 855,
'max_depth': 9, 'max_leaves': 66, 'min_child_weight': 5, 'reg_alpha':
0.348624698760067, 'reg_lambda': 0.48132209095520573, 'min_frequency':
0.059757460258282075}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:24,552] Trial 164 finished with value: 2100.3399362647237
and parameters: {'learning_rate': 0.0885447231102101, 'n_estimators': 874,
'max_depth': 9, 'max_leaves': 67, 'min_child_weight': 5, 'reg_alpha':
0.3205264119274339, 'reg_lambda': 0.4294690484629597, 'min_frequency':
0.027682274778268533}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:26,061] Trial 165 finished with value: 1992.8014932040157
and parameters: {'learning_rate': 0.08375102124578716, 'n_estimators': 824,
'max_depth': 8, 'max_leaves': 62, 'min_child_weight': 5, 'reg_alpha':
0.39057921067144397, 'reg_lambda': 0.48016193450867595, 'min_frequency':
0.021869627162201363}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:27,921] Trial 166 finished with value: 2061.006072895708 and
parameters: {'learning_rate': 0.08499595965948924, 'n_estimators': 931,
'max_depth': 8, 'max_leaves': 61, 'min_child_weight': 5, 'reg_alpha':
0.10544078148473306, 'reg_lambda': 0.5148687658187955, 'min_frequency':
0.016685348263388654}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:29,265] Trial 167 finished with value: 2061.0217522583416
and parameters: {'learning_rate': 0.09098684938232454, 'n_estimators': 728,
'max_depth': 8, 'max_leaves': 63, 'min_child_weight': 5, 'reg_alpha':
0.3868235038634221, 'reg_lambda': 0.4821955764294437, 'min_frequency':
0.035185368455282875}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:30,969] Trial 168 finished with value: 2063.565786810708 and
parameters: {'learning_rate': 0.08382634536601248, 'n_estimators': 811,
'max_depth': 8, 'max_leaves': 69, 'min_child_weight': 3, 'reg_alpha':
0.3743463783210693, 'reg_lambda': 0.1313293383881121, 'min_frequency':
0.06508831646271274}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:32,774] Trial 169 finished with value: 2080.8121276487045
```

```
and parameters: {'learning_rate': 0.08686567081210375, 'n_estimators': 910,
'max_depth': 8, 'max_leaves': 59, 'min_child_weight': 5, 'reg_alpha':
0.289944357234988, 'reg_lambda': 0.5333934174241647, 'min_frequency':
0.03957065661705488}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:34,555] Trial 170 finished with value: 2025.9402944358862
and parameters: {'learning_rate': 0.0759218524427595, 'n_estimators': 844,
'max_depth': 8, 'max_leaves': 74, 'min_child_weight': 5, 'reg_alpha':
0.137567698855561, 'reg_lambda': 0.4527328046178137, 'min_frequency':
0.03168203318320002}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:36,274] Trial 171 finished with value: 2027.0077068858463
and parameters: {'learning_rate': 0.07599220394385876, 'n_estimators': 843,
'max_depth': 8, 'max_leaves': 74, 'min_child_weight': 5, 'reg_alpha':
0.1440622843312391, 'reg_lambda': 0.4625945928578028, 'min_frequency':
0.031173538761206033}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:38,232] Trial 172 finished with value: 2056.4701470542004
and parameters: {'learning_rate': 0.07414389431039183, 'n_estimators': 869,
'max_depth': 8, 'max_leaves': 77, 'min_child_weight': 5, 'reg_alpha':
0.17571128772927225, 'reg_lambda': 0.4494312092291246, 'min_frequency':
0.03356423679943231}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:39,846] Trial 173 finished with value: 2022.7908849581866
and parameters: {'learning_rate': 0.07951341061265337, 'n_estimators': 831,
'max_depth': 8, 'max_leaves': 72, 'min_child_weight': 5, 'reg_alpha':
0.029202626132984064, 'reg_lambda': 0.4976688953363113, 'min_frequency':
0.026921744624937453}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:41,568] Trial 174 finished with value: 2032.720649744823 and
parameters: {'learning_rate': 0.07886366959865106, 'n_estimators': 784,
'max_depth': 8, 'max_leaves': 71, 'min_child_weight': 5, 'reg_alpha':
0.04051444727509746, 'reg_lambda': 0.5025298219199733, 'min_frequency':
0.03179211552724413}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:43,292] Trial 175 finished with value: 2053.7849219339473
and parameters: {'learning_rate': 0.05999490207974164, 'n_estimators': 838,
'max_depth': 8, 'max_leaves': 68, 'min_child_weight': 5, 'reg_alpha':
0.021756834344411585, 'reg_lambda': 0.516384044804974, 'min_frequency':
0.026325400505900234}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:45,209] Trial 176 finished with value: 2103.009907262586 and
parameters: {'learning_rate': 0.07690931322963236, 'n_estimators': 813,
'max_depth': 8, 'max_leaves': 72, 'min_child_weight': 5, 'reg_alpha':
0.08215713716320196, 'reg_lambda': 0.4823919717724118, 'min_frequency':
0.08281271812720735}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:47,167] Trial 177 finished with value: 2063.1654028991657
and parameters: {'learning_rate': 0.06356274721284638, 'n_estimators': 856,
'max_depth': 8, 'max_leaves': 76, 'min_child_weight': 5, 'reg_alpha':
0.1259918440206297, 'reg_lambda': 0.550553445928384, 'min_frequency':
0.028749689624405644}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:48,642] Trial 178 finished with value: 2051.8581840980783
and parameters: {'learning_rate': 0.06824466707315849, 'n_estimators': 759,
'max_depth': 8, 'max_leaves': 64, 'min_child_weight': 5, 'reg_alpha':
0.04927442281194527, 'reg_lambda': 0.5029190332921798, 'min_frequency':
```

```
0.012032885772060221}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:50,218] Trial 179 finished with value: 2062.346640552953 and
parameters: {'learning_rate': 0.05785064766657109, 'n_estimators': 825,
'max_depth': 8, 'max_leaves': 70, 'min_child_weight': 5, 'reg_alpha':
0.0712369088018788, 'reg_lambda': 0.02137010503086102, 'min_frequency':
0.07948594281276698}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:52,011] Trial 180 finished with value: 2059.964592240946 and
parameters: {'learning_rate': 0.0796112642175627, 'n_estimators': 883,
'max_depth': 8, 'max_leaves': 79, 'min_child_weight': 5, 'reg_alpha':
0.02302561396528491, 'reg_lambda': 0.5687574809328849, 'min_frequency':
0.03021845522236441}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:53,904] Trial 181 finished with value: 2041.4016182053258
and parameters: {'learning_rate': 0.08171694865005613, 'n_estimators': 949,
'max_depth': 9, 'max_leaves': 73, 'min_child_weight': 5, 'reg_alpha':
0.006780664418685305, 'reg_lambda': 0.4357019226361658, 'min_frequency':
0.02554117174095323}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:55,692] Trial 182 finished with value: 2056.179335289078 and
parameters: {'learning_rate': 0.08307316771122476, 'n_estimators': 852,
'max_depth': 9, 'max_leaves': 74, 'min_child_weight': 5, 'reg_alpha':
0.00024232328115074246, 'reg_lambda': 0.07272188190884182, 'min_frequency':
0.02006748667006493}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:56,961] Trial 183 finished with value: 2064.0452938988096
and parameters: {'learning_rate': 0.07212757335648971, 'n_estimators': 554,
'max_depth': 9, 'max_leaves': 69, 'min_child_weight': 5, 'reg_alpha':
0.46623519981171174, 'reg_lambda': 0.47671575301692787, 'min_frequency':
0.022981132480217197}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:05:59,083] Trial 184 finished with value: 2047.5027616873115
and parameters: {'learning_rate': 0.07729502886600881, 'n_estimators': 897,
'max_depth': 9, 'max_leaves': 75, 'min_child_weight': 5, 'reg_alpha':
0.03844529020376852, 'reg_lambda': 0.45257426501280756, 'min_frequency':
0.03686096989811482}. Best is trial 61 with value: 1970.6179534891642.
[I 2024-10-25 00:06:00,826] Trial 185 finished with value: 2033.2575172715983
and parameters: {'learning_rate': 0.08074429432237107, 'n_estimators': 874,
'max_depth': 9, 'max_leaves': 72, 'min_child_weight': 5, 'reg_alpha':
0.5274211169228081, 'reg_lambda': 0.5233978411815451, 'min_frequency':
0.02751600560863465}. Best is trial 61 with value: 1970.6179534891642.
```

[17]:

```python
study.set_user_attr('best_trial',study.best_trial)

best_params = study.best_params

col_transformer = ColumnTransformer(
    transformers=[
        ('date', date_transformer, ['date']),
        ('onehot',
            OneHotEncoder(
                sparse_output=False,
                min_frequency = best_params['min_frequency']),
            ['city', 'shop', 'brand', 'container', 'capacity']),
        ('scaler', StandardScaler(), ['lat', 'long', 'pop', 'price'])
    ],
    remainder='passthrough'
)

del best_params['min_frequency']

pipe_xgbreg_ = Pipeline([
    ('preprocessor', col_transformer),
    ('regressor', XGBRegressor(
        enable_categorical=True,
        random_state=42,
        device='cuda',
        **best_params
        ))
])

study.set_user_attr('best_pipeline', pipe_xgbreg_)
```

[18]:

```
study.best_value
```

[18]:

```
1970.6179534891642
```

El MAE es más bajo. Esto se debe a que el algoritmo de optuna busca la mejor combinación de parámetros respecto al MAE según la predicción del modelo seleccionado. Optuna ha encontrado el mejor punto medio (dentro de las pruebas) entre complejidad del árbol y generalización al conjunto de validación.

3.4.

[19]:

```
study.best_params
```

[19]:

```
{'learning_rate': 0.061495828095354074,
 'n_estimators': 926,
 'max_depth': 8,
 'max_leaves': 68,
 'min_child_weight': 5,
 'reg_alpha': 0.0028755461558962053,
 'reg_lambda': 0.12085974287977863,
 'min_frequency': 0.012681898035110887}
```

- `learning_rate` o *shrinkage* cuantifica la influencia de cada árbol sobre cada predicción. Modelos más conservativos tendrán *shrinkage* más pequeño para evitar el sobreajuste. Se buscaron valores entre 0.0 y 1.0, lo que se condice con la documentación.
- `n_estimators` se refiere a la cantidad de rondas de *boosting*. Más estimadores aumentan la cantidad de modelos más débiles que se utilizan para la estimación, el rendimiento del modelo y el tiempo de ejecución. El que el mejor valor haya resultado tan cerca de 1000 sugiere que

no se trataba de un rango no razonable.
- `max_depth` controla la máxima profundidad de cada árbol. Valores más altos hacen del modelo uno más complejo y propenso a sobreajustar. El valor por defecto es 6. Un valor de 10 parece suficientemente razonable.
- `max_leaves` se refiere a la cantidad de nodos que se agregan con cada árbol. Por defecto, este valor es cero. Valores más grandes aumentan la complejidad del modelo y el riesgo de sobreajuste. Optuna encontró que un valor de 35 es óptimo (incluso en un rango que contiene al valor por defecto), sugiriendo que un modelo más complejo es deseable.
- `min_child_weight` establece el peso mínimo que debe tener una hoja para que se produzca una división. Reducir este número permite evitar que la computación se realice con sets de datos demasiado específicos, llevando a sobreajuste. El rango de valores propuesto es considerablemente pequeño ante la cantidad de datos disponibles, lo que puede llevar a una elección de parámetros que sobreajuste.
- `reg_alpha` establece la regularización L1 a todos los pesos. Un valor de 1 reduciría el impacto de todos los pesos. Su valor por defecto es 1, y su rango es $[0, \text{inf}]$. Elegir valores en $[0, 1]$ significa probar con regularizaciones relativamente débiles.
- `reg_lambda` es equivalente a `reg_alpha` para regularización L2. Al igual que la anterior, números más grandes llevan a modelos más conservadores.
- `min_frequency`especifica la mínima frecuencia o cantidad que una categoría debiese tener para ser considerada infrecuente. Valores enteros representan valores absolutos, mientras que decimales representan porcentajes sobre la población total. Un valor de 0.08 indica que la categoría debe ser más común que el 8% por sobre el total para que se le asigne una columna. Este valor optimizado podría ser demasiado pequeño, al dejar fuera ciudades que aún sin tanta frecuencia entregan información importante para la toma de decisiones.

3.5.

```
[20]: with open(f'./pickle_3.pkl', 'wb') as pkl_file:
          pickle.dump(study.user_attrs['best_pipeline'], pkl_file)
```

3.1 4. Optimización de Hiperparámetros con Optuna y Prunners (17 puntos)

Después de optimizar el rendimiento de su modelo varias veces, Fiu le pregunta si no es posible optimizar el entrenamiento del modelo en sí mismo. Después de leer un par de post de personas de dudosa reputación en la *deepweb*, usted llega a la conclusión que puede cumplir este objetivo mediante la implementación de **Prunning**.

Vuelva a optimizar los mismos hiperparámetros que la sección pasada, pero esta vez utilizando **Prunning** en la optimización. En particular, usted debe:

- Responder: ¿Qué es prunning? ¿De qué forma debería impactar en el entrenamiento? [2 puntos]
- Redefinir la función `objective()` utilizando `optuna.integration.XGBoostPruningCallback` como método de **Prunning** [10 puntos]
- Fijar nuevamente el tiempo de entrenamiento a 5 minutos [1 punto]
- Reportar el número de *trials*, el `MAE` y los mejores hiperparámetros encontrados. ¿Cómo cambian sus resultados con respecto a la sección anterior? ¿A qué se puede deber esto? [3 puntos]
- Guardar su modelo en un archivo .pkl [1 punto]

Nota: Si quieren silenciar los prints obtenidos en el prunning, pueden hacerlo mediante el siguiente comando:

`optuna.logging.set_verbosity(optuna.logging.WARNING)`

De implementar la opción anterior, pueden especificar `show_progress_bar = True` en el método `optimize` para *más sabor*.

Hint: Si quieren especificar parámetros del método .fit() del modelo a través del pipeline, pueden hacerlo por medio de la siguiente sintaxis: `pipeline.fit(stepmodelo__parametro = valor)`

Hint2: Este enlace les puede ser de ayuda en su implementación

Prunning es una técnica utilizada para detener tempranamente los ensayos de hiperparámetros que no muestran promesas de mejora. En lugar de esperar a que cada ensayo complete su entrenamiento, evalúa el rendimiento intermedio y decide si continuar o detener el ensayo. Esta técnica es especialmente útil en la optimización de hiperparámetros, donde el tiempo y los recursos computacionales son limitados.

La implementación de prunning en el entrenamiento de modelos debería tener un impacto significativo. Al detener los ensayos no prometedores de manera temprana, se pueden reasignar los recursos a ensayos más prometedores, acelerando el proceso de optimización. Esto no solo reduce el tiempo de entrenamiento, sino que también aumenta la eficiencia del proceso, permitiendo explorar un mayor número de configuraciones de hiperparámetros en el mismo período.

[21]:

```
#import optuna
#from optuna.integration import XGBoostPruningCallback
#from xgboost import XGBRegressor
#from sklearn.pipeline import Pipeline
#from sklearn.compose import ColumnTransformer
#from sklearn.preprocessing import OneHotEncoder, StandardScaler
#from sklearn.metrics import mean_absolute_error
#import pickle
#
## Silenciar los logs de Optuna para reducir la salida innecesaria
#optuna.logging.set_verbosity(optuna.logging.WARNING)
#
## Función objetivo para Optuna con prunning
#def objective(trial):
#    # Sugerencias de hiperparámetros para XGBRegressor
#    learning_rate = trial.suggest_float('learning_rate', 0.001, 0.1)
#    n_estimators = trial.suggest_int('n_estimators', 50, 1000)
#    max_depth = trial.suggest_int('max_depth', 3, 10)
#    max_leaves = trial.suggest_int('max_leaves', 0, 100)
#    min_child_weight = trial.suggest_int('min_child_weight', 1, 5)
#    reg_alpha = trial.suggest_float('reg_alpha', 0.0, 1.0)
#    reg_lambda = trial.suggest_float('reg_lambda', 0.0, 1.0)
#
#    # Sugerencia de hiperparámetro del OneHotEncoder
#    min_frequency = trial.suggest_float('min_frequency', 0.0, 1.0)
```

```
#
#    # Definir el preprocesador de columnas
#    col_transformer = ColumnTransformer(
#        transformers=[
#            ('date', date_transformer, ['date']),  # date_transformer ya␣
 ↪definido
#            ('onehot', OneHotEncoder(sparse_output=False,␣
 ↪min_frequency=min_frequency), ['city', 'shop', 'brand', 'container',␣
 ↪'capacity']),
#            ('scaler', StandardScaler(), ['lat', 'long', 'pop', 'price'])
#        ],
#        remainder='passthrough'
#    )
#
#    # Crear y ajustar el modelo XGBRegressor
#    pipe_xgbreg = Pipeline([
#        ('preprocessor', col_transformer),
#        ('regressor', XGBRegressor(
#            learning_rate=learning_rate,
#            n_estimators=n_estimators,
#            max_depth=max_depth,
#            max_leaves=max_leaves,
#            min_child_weight=min_child_weight,
#            reg_alpha=reg_alpha,
#            reg_lambda=reg_lambda,
#            enable_categorical=True,
#            random_state=42,
#            eval_metric="mae"  # Configuración de la métrica
#        ))
#    ])
#
#    # Callback de Optuna para prunning
#    pruning_callback = XGBoostPruningCallback(trial, "validation_0-mae")
#
#    # Ajustar el pipeline con early stopping y prunning (pasar los parámetros␣
 ↪al regressor dentro del pipeline)
#    pipe_xgbreg.fit(
#        X_train, y_train,
#        regressor__eval_set=[(X_dev, y_dev)],
#        regressor__early_stopping_rounds=50,
#        regressor__callbacks=[pruning_callback],
#        regressor__verbose=False  # Evitar el exceso de logs
#    )
#
#    # Predicciones y cálculo del MAE
#    y_pred = pipe_xgbreg.predict(X_dev)
#    error = mean_absolute_error(y_dev, y_pred)
```

```
#
#    # Almacenar el mejor modelo en el trial
#    trial.set_user_attr("best_model", pipe_xgbreg)
#
#    return error
#
## Crear el estudio de Optuna con prunning y configurar el tiempo de␣
 ↪optimización
#study = optuna.create_study(sampler=optuna.samplers.TPESampler(),␣
 ↪direction="minimize")
#study.optimize(objective, timeout=300, show_progress_bar=True)
#
## Reportar los resultados de la optimización
#print(f"Total de trials realizados: {len(study.trials)}")
#print(f"Mejor MAE: {study.best_value}")
#print(f"Mejores hiperparámetros: {study.best_params}")
#
## Recuperar el mejor modelo
#best_model = study.best_trial.user_attrs["best_model"]
#
## Guardar el mejor modelo en un archivo .pkl
#with open('best_xgboost_model_pruned.pkl', 'wb') as f:
#    pickle.dump(best_model, f)
#
```

[22]:

```
#study = optuna.create_study(sampler=TPESampler(), direction="minimize")

#study.optimize(objective, timeout=300, show_progress_bar=True)
```

[23]:

```
# No puedo aaaa
```

3.2 5. Visualizaciones (5 puntos)

Satisfecho con su trabajo, Fiu le pregunta si es posible generar visualizaciones que permitan entender el entrenamiento de su modelo.

A partir del siguiente enlace, genere las siguientes visualizaciones:

1. Gráfico de historial de optimización [1 punto]
2. Gráfico de coordenadas paralelas [1 punto]
3. Gráfico de importancia de hiperparámetros [1 punto]

Comente sus resultados:

4. ¿Desde qué *trial* se empiezan a observar mejoras notables en sus resultados? [0.5 puntos]
5. ¿Qué tendencias puede observar a partir del gráfico de coordenadas paralelas? [1 punto]
6. ¿Cuáles son los hiperparámetros con mayor importancia para la optimización de su modelo? [0.5 puntos]

[24]:
```
optuna.visualization.plot_optimization_history(study)
```

[25]:
```
optuna.visualization.plot_parallel_coordinate(study)
```

[26]:
```
optuna.visualization.plot_param_importances(study)
```

3.3 6. Síntesis de resultados (3 puntos)

Finalmente:

1. Genere una tabla resumen del MAE en el conjunto de validación obtenido en los 5 modelos entrenados desde Baseline hasta XGBoost con Constraints, Optuna y Prunning. [1 punto]
2. Compare los resultados de la tabla y responda, ¿qué modelo obtiene el mejor rendimiento? [0.5 puntos]
3. Cargue el mejor modelo, prediga sobre el conjunto de **test** y reporte su MAE. [0.5 puntos]
4. ¿Existen diferencias con respecto a las métricas obtenidas en el conjunto de validación? ¿Porqué puede ocurrir esto? [1 punto]

[29]:
```
mae_results = {
    'Modelo': ['Baseline (Dummy)', 'XGBoost', 'XGBoost con Constraints',␣
 ↪'XGBoost con Optuna'],
    'MAE en Validación': [mae_baseline, mae_xgboost, mae_xgboost_r, '1970.617']
}

mae_table = pd.DataFrame(mae_results)
print(mae_table)
```

```
                    Modelo MAE en Validación
0         Baseline (Dummy)      13298.497767
1                  XGBoost       2400.277512
2  XGBoost con Constraints       2602.156124
3       XGBoost con Optuna          1970.617
```

De la tabla anterior, el modelo que obtuvo el mejor rendimiento es XGBoost con Optuna, con un MAE de 1970.62 en el conjunto de validación. Esto indica que la optimización de hiperparámetros con Optuna mejoró el desempeño del modelo en comparación con las otras versiones. Posiblemente con prunning sea mejor.

[32]:
```
from sklearn.metrics import mean_absolute_error
import pickle

# Cargar el mejor modelo (XGBoost optimizado con Optuna)
with open('pickle_3.pkl', 'rb') as file:
    best_model = pickle.load(file)

# Predecir sobre el conjunto de test
y_test_pred = best_model.predict(X_test)

# Calcular el MAE en el conjunto de test
```

```
test_mae = mean_absolute_error(y_test, y_test_pred)
print(f'MAE en el conjunto de test: {test_mae}')
```

```
---------------------------------------------------------------------------
NotFittedError                            Traceback (most recent call last)
Cell In[32], line 9
      6     best_model = pickle.load(file)
      8 # Predecir sobre el conjunto de test
----> 9 y_test_pred = best_model.predict(X_test)
     11 # Calcular el MAE en el conjunto de test
     12 test_mae = mean_absolute_error(y_test, y_test_pred)

File c:
 ↪\Users\rodri\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\pipeline.
 ↪py:602, in Pipeline.predict(self, X, **params)
    600 if not _routing_enabled():
    601     for _, name, transform in self._iter(with_final=False):
--> 602         Xt = transform.transform(Xt)
    603     return self.steps[-1][1].predict(Xt, **params)
    605 # metadata routing enabled

File c:
 ↪\Users\rodri\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\utils\_set_
 ↪py:295, in _wrap_method_output.<locals>.wrapped(self, X, *args, **kwargs)
    293 @wraps(f)
    294 def wrapped(self, X, *args, **kwargs):
--> 295     data_to_wrap = f(self, X, *args, **kwargs)
    296     if isinstance(data_to_wrap, tuple):
    297         # only wrap the first output for cross decomposition
    298         return_tuple = (
    299             _wrap_data_with_container(method, data_to_wrap[0], X, self)
    300             *data_to_wrap[1:],
    301         )

File c:
 ↪\Users\rodri\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\compose\_co
 ↪py:973, in ColumnTransformer.transform(self, X, **params)
    947 """Transform X separately by each transformer, concatenate results.
    948 
    949 Parameters
   (…)
    970     sparse matrices.
    971 """
    972 _raise_for_params(params, self, "transform")
--> 973 check_is_fitted(self)
    974 X = _check_X(X)
    976 # If ColumnTransformer is fit using a dataframe, and now a dataframe is
    977 # passed to be transformed, we select columns by name instead. This
    978 # enables the user to pass X at transform time with extra columns which
    979 # were not present in fit time, and the order of the columns doesn't
    980 # matter.

File c:
 ↪\Users\rodri\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\utils\valida
 ↪py:1622, in check_is_fitted(estimator, attributes, msg, all_or_any)
   1619     raise TypeError("%s is not an estimator instance." % (estimator))
   1621 if not _is_fitted(estimator, attributes, all_or_any):
-> 1622     raise NotFittedError(msg % {"name": type(estimator).__name__})

NotFittedError: This ColumnTransformer instance is not fitted yet. Call 'fit'␣
 ↪with appropriate arguments before using this estimator.
```

4 Conclusión

Eso ha sido todo para el lab de hoy, recuerden que el laboratorio tiene un plazo de entrega de una semana. Cualquier duda del laboratorio, no duden en contactarnos por mail o U-cursos.