

Grau en Enginyeria en Tecnologies Industrials

Machine Learning aplicat
al Data Warehouse tradicional

ANNEX

Autor: Adrià Recort I Fernandez
Director: Daniela Tost Padreny
Convocatòria: Gener 2020



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Sumari

SUMARI	2
1. SCRIPTS PL/SQL	3
1.1. Exemple de neteja de dades	3
1.2. Exemple d'ús de diccionaris	3
1.3. Job automatitzat a la base de dades	4
1.4. Script desat a la base de dades.....	5
1.5. Script per a l'extracció de dades.....	9
1.6. Scripts emprats per SAP BO:	9
1.6.1. Leads.....	9
1.6.2. Expedients al centre d'atenció telefònica	10
1.6.3. Taller.....	11
1.6.4. Enquestes de satisfacció	11
2. SCRIPTS PYTHON	13
2.1. Script iteració sobre paràmetres:	13
2.2. Script per a la validació creuada del model:	14
2.3. Script per l'avaluació de l'entrenament:	15
2.4. Script per a la utilització del predictor:	16
2.5. Script per a la instal·lació de l'entregable:	16
3. SCRIPT .BAT	17

1. Scripts PL/SQL

1.1. Exemple de neteja de dades

```

SELECT COD_CLIENTE,
       UPPER(                                     --Convertim a majúscules
              REGEXP_REPLACE(
              REGEXP_REPLACE(
              REGEXP_REPLACE(
              REGEXP_REPLACE(
              REGEXP_REPLACE(
                     PROVINCIA, '[àáäâ]', 'a' )    --Emprem regexp per a
                     '[èéëê]', 'e' )           --substituir els accENTS i
                     '[ìíïî]', 'i' )           --repetim per a les 5 vocals
                     '[òóöô]', 'o' )
                     '[ùúüû]', 'u' )
              ) PROVINCIA_OUT
FROM ODS_CLIENTES_PROVINCIAS

```

1.2. Exemple d'ús de diccionaris

```

SELECT COD_CLIENTE,
       COALESCE(C2.COLUMN1, --Retorna el primer valor
                 C3.COLUMN1, --no null de la selecció
                 C4.COLUMN1, --i seleccionem sempre la
                 C5.COLUMN1, --columna 1 de cada unió
                 C6.COLUMN1) PROVINCIA_LIMPIA
FROM ODS_CLIENTES_PROVINCIAS ORIGEN
LEFT JOIN DICCIONARIO_PROVINCIAS C2
ON ORIGEN.PROVINCIA = C2.COLUMN2
LEFT JOIN DICCIONARIO_PROVINCIAS C3
ON ORIGEN.PROVINCIA = C3.COLUMN3
LEFT JOIN DICCIONARIO_PROVINCIAS C4
ON ORIGEN.PROVINCIA = C4.COLUMN4
LEFT JOIN DICCIONARIO_PROVINCIAS C5
ON ORIGEN.PROVINCIA = C5.COLUMN5
LEFT JOIN DICCIONARIO_PROVINCIAS C6
ON ORIGEN.PROVINCIA = C6.COLUMN6;

```

1.3. Job automatitzat a la base de dades

```

DECLARE
    user_name varchar2(128);
BEGIN
    select user into user_name from dual;
    execute immediate 'alter session set current_schema = CLIMAPROODS';
    BEGIN
        SYS.DBMS_JOB.REMOVE(141);
        execute immediate 'alter session set current_schema = ' || user_name ;
    EXCEPTION
        WHEN OTHERS THEN
            execute immediate 'alter session set current_schema = ' || user_name ;
            RAISE;
    END;
    COMMIT;
END;
/

```



```

DECLARE
    X NUMBER;
    user_name varchar2(128);
BEGIN
    select user into user_name from dual;
    execute immediate 'alter session set current_schema = CLIMAPROODS';
    BEGIN
        SYS.DBMS_JOB.SUBMIT
        (
            job      => X
            ,what     => 'P_DATASET_MONTHLY(TO_NUMBER(TO_CHAR(SYSDATE, ''YYYY'')));

            ,next_date => to_date('22/04/2020 14:07:00','dd/mm/yyyy hh24:mi:ss')
            ,interval   => 'TRUNC(LAST_DAY(SYSDATE)) + 1 + 8/24 + 30/1440'
            ,no_parse   => FALSE
        );
        SYS.DBMS_OUTPUT.PUT_LINE('Job Number is: ' || to_char(x));
        execute immediate 'alter session set current_schema = ' || user_name ;
    EXCEPTION
        WHEN OTHERS THEN
            execute immediate 'alter session set current_schema = ' || user_name ;
            RAISE;
    END;
    COMMIT;
END;
/

```

1.4. Script desat a la base de dades

```

CREATE OR REPLACE PROCEDURE P_DATASET_MONTHLY(V_DATE IN INT) IS

v_date_i DATE := to_date(V_DATE-5||'0401','yyyymmdd');

v_date_f DATE := to_date(V_DATE-1||'0630','yyyymmdd');

BEGIN

DELETE FROM TFG_TRAIN_DATASET;

INSERT INTO TFG_TRAIN_DATASET
select dm.cod_persona_climapro_i, ped.cod_pedido, fecha_baja_cons, modelo_interes_c,
decode(dm.cod_sexo,'H',0,'M',1,-1) sexo, dm.desc_provincia, edad, veh.kilometraje,
round(months_between(fecha_baja_cons, fecha_matriculacion)/12) antiguedad,
case when pel1.cod_persona_climapro_i is null then 0 else 1 end flag_lead,
case when pel2.cod_persona_climapro_i is null then 0 else 1 end flag_expediente,
case when cab.bastidor is null then 0 else 1 end flag_taller,
case when opcionales like '%EA3%' or opcionales like '%EA5%' or opcionales like '%EA6%'
      or opcionales like '%EA8%' then 1 else 0 end acc1,
case when opcionales like '%3FJ%' or opcionales like '%3FU%' or opcionales like '%PFU%'
      or opcionales like '%PS1%' or opcionales like '%PS2%' or opcionales like '%PS6%'
      or opcionales like '%PS8%' or opcionales like '%PS9%' or opcionales like '%WYC%' then 1 else 0 end acc2,
case when opcionales like '%6T1%' or opcionales like '%8BD%' or opcionales like '%8G4%'
      or opcionales like '%8IE%' or opcionales like '%8IM%' or opcionales like '%8IP%' or
      opcionales like '%8IT%' or opcionales like '%8IU%' or opcionales like '%8IX%' or
      opcionales like '%8IV%' or opcionales like '%8K3%' or opcionales like '%8S6%' or
      opcionales like '%8S7%' or opcionales like '%8SH%' or opcionales like '%8SK%' or
      opcionales like '%8SP%' or opcionales like '%8SQ%' or opcionales like '%8VG%' or
      opcionales like '%8VH%' or opcionales like '%8WH%' or opcionales like '%9C7%' or
      opcionales like '%9I5%' or opcionales like '%9TJ%' or opcionales like '%9TK%' or
      opcionales like '%KL1%' or opcionales like '%P01%' or opcionales like '%PED%' or
      opcionales like '%PLP%' or opcionales like '%PX1%' or opcionales like '%PXA%' or
      opcionales like '%PXB%' or opcionales like '%PXC%' or opcionales like '%PXE%' or
      opcionales like '%W73%' or opcionales like '%W7N%' or opcionales like '%W91%' or
      opcionales like '%WLH%' or opcionales like '%WLL%' or opcionales like '%WXE%' or
      opcionales like '%WXF%' then 1 else 0 end acc3,
case when opcionales like '%2FK%' or opcionales like '%2JX%' or opcionales like '%2PK%' or
      opcionales like '%2PU%' or opcionales like '%5J3%' or opcionales like '%6KP%' or
      opcionales like '%7M3%' or opcionales like '%7M9%' or opcionales like '%FN7%' or
      opcionales like '%N0H%' or opcionales like '%N0V%' or opcionales like '%N5K%' or
      opcionales like '%N5X%' or opcionales like '%N5Y%' or opcionales like '%N6L%' then 1 else 0 end acc4;

```

```

        or opcionales like '%P06%' or opcionales like '%P08%' or opcionales like '%PA5%'
        or opcionales like '%PL2%' or opcionales like '%PL8%' or opcionales like '%PLH%'
        or opcionales like '%VT2%' or opcionales like '%W00%' or opcionales like '%W01%'
        or opcionales like '%W05%' or opcionales like '%W06%' or opcionales like '%W07%'
        or opcionales like '%W12%' or opcionales like '%W17%' or opcionales like '%W21%'
        or opcionales like '%W25%' or opcionales like '%W2P%' or opcionales like '%W31%'
        or opcionales like '%W32%' or opcionales like '%W59%' or opcionales like '%W60%'
        or opcionales like '%W67%' or opcionales like '%W74%' or opcionales like '%W87%'
        or opcionales like '%W93%' or opcionales like '%WID%' or opcionales like '%WIE%'
        or opcionales like '%WII%' or opcionales like '%WIJ%' or opcionales like '%WIR%'
        or opcionales like '%WLF%' or opcionales like '%WLG%' or opcionales like '%WOG%'
        or opcionales like '%WSR%' or opcionales like '%WYD%'

        then 1 else 0 end acc4,
case when opcionales like '%G1A%' or opcionales like '%G1C%' or opcionales like '%G1D%'
      or opcionales like '%G1G%'
      then 1 else 0 end acc5,
case when pedidos.persona is null then 0 else 1 end flag_renovador
from dim_pedidos ped
inner join dim_vehiculos veh
  on ped.cod_vehiculo_climapro = veh.cod_vehiculo_climapro
inner join dim_relaciones_imp rel
  on ped.cod_vehiculo_climapro = rel.cod_vehiculo_climapro
  and ped.cod_conductor = rel.cod_persona_climapro_i
  and rel.cod_tipo_relacion in (1,2)
inner join dim_personas_imp dm
  on ped.cod_conductor = dm.cod_persona_climapro_i
inner join dim_personas_atr atr
  on dm.cod_persona_climapro_i = atr.cod_persona_climapro_i
inner join ods_apex_dic_modelos_interes dicc
  on veh.desc_segmento_sli = upper(dicc.modelo_interes)
left join ods_personas_en_fuente_1 pel1
  on dm.cod_persona_climapro_i = pel1.cod_persona_climapro_i
  and pel1.cod_fuente_origen = 'SFI'
  and fecha_es_lead between add_months(fecha_baja_cons, -
4) and add_months(fecha_baja_cons,4)
left join ods_personas_en_fuente_1 pel2
  on dm.cod_persona_climapro_i = pel2.cod_persona_climapro_i
  and pel2.cod_fuente_origen = 'SCC'
left join ods_dmt_or_cabecera cab
  on veh.bastidor = cab.bastidor
left join(
    select cod_pedido,cod_conductor persona,fecha_alta_cliente
    from dim_pedidos
    where cod_marca = '36V'
    and fecha_alta_cliente between add_months(v_date_i, -
8) and add_months(v_date_f,6)

```

```

        union
            select cod_pedido,cod_propietario persona,fecha_alta_cliente
            from dim_pedidos
            where cod_marca = '36V'
                and fecha_alta_cliente between add_months(v_date_i,-
8) and add_months(v_date_f,6)
            ) pedidos
            on ped.cod_conductor = pedidos.persona and ped.cod_pedido != pedidos.cod_pedido
            and pedidos.fecha_alta_cliente between add_months(fecha_baja_cons,-
8) and add_months(fecha_baja_cons,6)
            where ped.flag_registro_inactivo = 0
                and rel.flag_registro_inactivo = 0
                and ped.cod_marca = '36V'
                and ped.cod_tipo_cliente = 7
                and fecha_baja_cons between v_date_i and v_date_f
                and dm.flag_cliente_inactivo is null
                and dm.cod_sexo is not null
                and dm.cod_tipo_persona = 'F'
                and dm.desc_provincia is not null
                and dm.desc_provincia != 'NO VALIDO'
                and edad is not null
                and edad < 90
;
;

DELETE FROM TFG_EVAL_DATASET;

INSERT INTO TFG_EVAL_DATASET
select dm.cod_persona_climapro_i, cod_pedido, fecha_baja_cons, modelo_interes_c,
decode(dm.cod_sexo,'H',0,'M',1,-1) sexo, dm.desc_provincia, edad, veh.kilometraje,
round(months_between(fecha_baja_cons, fecha_matriculacion)/12) antiguedad,
case when pel1.cod_persona_climapro_i is null then 0 else 1 end flag_lead,
case when pel2.cod_persona_climapro_i is null then 0 else 1 end flag_expediente,
case when cab.bastidor is null then 0 else 1 end flag_taller,
case when opcionales like '%EA3%' or opcionales like '%EA5%' or opcionales like '%EA6%'
      or opcionales like '%EA8%'
      then 1 else 0 end acc1,
case when opcionales like '%3FJ%' or opcionales like '%3FU%' or opcionales like '%PFU%'
      or opcionales like '%PS1%' or opcionales like '%PS2%' or opcionales like '%PS6%'
      or opcionales like '%PS8%' or opcionales like '%PS9%' or opcionales like '%WYC%'
      then 1 else 0 end acc2,
case when opcionales like '%6T1%' or opcionales like '%8BD%' or opcionales like '%8G4%'
      or opcionales like '%8IE%' or opcionales like '%8IM%' or opcionales like '%8IP%'
      or opcionales like '%8IT%' or opcionales like '%8IU%' or opcionales like '%8IX%'
      or opcionales like '%8IY%' or opcionales like '%8K3%' or opcionales like '%8S6%'
      or opcionales like '%8S7%' or opcionales like '%8SH%' or opcionales like '%8SK%'
      or opcionales like '%8SP%' or opcionales like '%8SQ%' or opcionales like '%8VG%'
```

```

        or opcionales like '%8VH%' or opcionales like '%8WH%' or opcionales like '%9C7%'
        or opcionales like '%9I5%' or opcionales like '%9TJ%' or opcionales like '%9TK%'
        or opcionales like '%KL1%' or opcionales like '%P01%' or opcionales like '%PED%'
        or opcionales like '%PLP%' or opcionales like '%PX1%' or opcionales like '%PXA%'
        or opcionales like '%PXB%' or opcionales like '%PXC%' or opcionales like '%PXE%'
        or opcionales like '%W73%' or opcionales like '%W7N%' or opcionales like '%W91%'
        or opcionales like '%WLH%' or opcionales like '%WLL%' or opcionales like '%WXE%'
        or opcionales like '%WXF%'

    then 1 else 0 end acc3,
case when opcionales like '%2FK%' or opcionales like '%2JX%' or opcionales like '%2PK%'
      or opcionales like '%2PU%' or opcionales like '%5J3%' or opcionales like '%6KP%'
      or opcionales like '%7M3%' or opcionales like '%7M9%' or opcionales like '%FN7%'
      or opcionales like '%N0H%' or opcionales like '%N0V%' or opcionales like '%N5K%'
      or opcionales like '%N5X%' or opcionales like '%N5Y%' or opcionales like '%N6L%'
      or opcionales like '%P06%' or opcionales like '%P08%' or opcionales like '%PA5%'
      or opcionales like '%PL2%' or opcionales like '%PL8%' or opcionales like '%PLH%'
      or opcionales like '%VT2%' or opcionales like '%W00%' or opcionales like '%W01%'
      or opcionales like '%W05%' or opcionales like '%W06%' or opcionales like '%W07%'
      or opcionales like '%W12%' or opcionales like '%W17%' or opcionales like '%W21%'
      or opcionales like '%W25%' or opcionales like '%W2P%' or opcionales like '%W31%'
      or opcionales like '%W32%' or opcionales like '%W59%' or opcionales like '%W60%'
      or opcionales like '%W67%' or opcionales like '%W74%' or opcionales like '%W87%'
      or opcionales like '%W93%' or opcionales like '%WID%' or opcionales like '%WIE%'
      or opcionales like '%WII%' or opcionales like '%WIJ%' or opcionales like '%WIR%'
      or opcionales like '%WLF%' or opcionales like '%WLG%' or opcionales like '%WOG%'
      or opcionales like '%WSR%' or opcionales like '%WYD%'

    then 1 else 0 end acc4,
case when opcionales like '%G1A%' or opcionales like '%G1C%' or opcionales like '%G1D%'
      or opcionales like '%G1G%'

    then 1 else 0 end acc5
from dim_pedidos ped
inner join dim_vehiculos veh
  on ped.cod_vehiculo_climapro = veh.cod_vehiculo_climapro
inner join dim_relaciones_imp rel
  on ped.cod_vehiculo_climapro = rel.cod_vehiculo_climapro
  and ped.cod_conductor = rel.cod_persona_climapro_i
  and rel.cod_tipo_relacion in (1,2)
inner join dim_personas_imp dm
  on ped.cod_conductor = dm.cod_persona_climapro_i
inner join dim_personas_atr atr
  on dm.cod_persona_climapro_i = atr.cod_persona_climapro_i
inner join ods_apex_dic_modelos_interes dicc
  on veh.desc_segmento_sli = upper(dicc.modelo_interes)
left join ods_personas_en_fuente_l pel1
  on dm.cod_persona_climapro_i = pel1.cod_persona_climapro_i
  and pel1.cod_fuente_origen = 'SFI'

```

```

        and fecha_es_lead between add_months(fecha_baja_cons,-
4) and add_months(fecha_baja_cons,4)
        left join ods_personas_en_fuente_1 pel2
            on dm.cod_persona_climapro_i = pel2.cod_persona_climapro_i
            and pel2.cod_fuente_origen = 'SCC'
        left join ods_dmt_or_cabecera cab
            on veh.bastidor = cab.bastidor
        where ped.flag_registro_inactivo = 0
            and rel.flag_registro_inactivo = 0
            and ped.cod_marca = '36V'
            and ped.cod_tipo_cliente = 7
            and fecha_baja_cons between v_date_f and sysdate
            and dm.flag_cliente_inactivo is null
            and dm.cod_sexo is not null
            and dm.cod_tipo_persona = 'F'
            and dm.desc_provincia is not null
            and dm.desc_provincia != 'NO VALIDO'
            and edad is not null
            and edad < 90
;
COMMIT;
END;

```

1.5. Script per a l'extracció de dades

```

SELECT * FROM TFG_TRAIN_DATASET;

SELECT * FROM TFG_EVAL_DATASET;

```

1.6. Scripts emprats per SAP BO:

1.6.1. Leads

```

select ped.cod_conductor,
case when asistentes.cod_persona_climapro_i is null then 0 else 1 end flag_lead,
case when pedidos.persona is null then 0 else 1 end flag_renovador
from tmp_arecort_renovadores ped
    LEFT join (
        select cod_pedido,cod_conductor persona,fecha_alta_cliente from dim_pedidos
        where cod_marca = '36V'
        and fecha_alta_cliente > date'2017-01-01'
        union

```

```

select cod_pedido,cod_propietario persona,fecha_alta_cliente from dim_pedidos
where cod_marca = '36V'
and fecha_alta_cliente > date'2017-01-01'
) pedidos on
ped.cod_conductor = pedidos.persona and ped.cod_pedido != pedidos.cod_pedido
and pedidos.fecha_alta_cliente between add_months(ped.fecha_baja_cons,-8) and add_months(ped.fecha_baja_cons,6)
left join (
select distinct cod_persona_climapro_i, fecha_es_lead
from ods_sfi_pef pef
inner join ods_personas_en_fuente_1 pel
on pef.cod_persona_en_fuente = pel.cod_persona_en_fuente
) asistentes on
ped.cod_conductor = asistentes.cod_persona_climapro_i
and fecha_es_lead between add_months(fecha_baja_cons,-4) and add_months(fecha_baja_cons,4)

```

1.6.2. Expedients al centre d'atenció telefònica

```

select ped.cod_conductor,
case when expedientes.cod_persona_climapro_i is null then 0 else 1 end flag_expediente,
desc_categoria,
case when pedidos.persona is null then 0 else 1 end flag_renovador
from tmp_arecort_renovadores ped
LEFT join (
select cod_pedido,cod_conductor persona,fecha_alta_cliente from dim_pedidos
where cod_marca = '36V'
and fecha_alta_cliente > date'2017-01-01'
union
select cod_pedido,cod_propietario persona,fecha_alta_cliente from dim_pedidos
where cod_marca = '36V'
and fecha_alta_cliente > date'2017-01-01'
) pedidos on
ped.cod_conductor = pedidos.persona and ped.cod_pedido != pedidos.cod_pedido
and pedidos.fecha_alta_cliente between add_months(ped.fecha_baja_cons,-8) and add_months(ped.fecha_baja_cons,6)
left join (
select distinct pel.cod_persona_climapro_i,desc_categoria
from ODS_SCC_PEF scc
inner join ods_personas_en_fuente_1 pel

```



```

on scc.cod_persona_en_fuente = pel.cod_persona_en_fuente
inner join ods_scc_expedientes expe
on scc.ID_CLIENTE = expe.COD CUENTA
and expe.cod_marca = '36V'
) expedientes on
ped.cod_conductor = expedientes.cod_persona_climapro_i

```

1.6.3. Taller

```

select distinct tmp2.cod_vehiculo_climapro,tmp2.pasos,tmp2.gasto,
CASE WHEN PEDIDOS.PERSONA IS NULL THEN 0 ELSE 1 END Flag_renovador
from(
select tmp.cod_vehiculo_climapro,tmp.cod_conductor,tmp.fecha_baja_cons,tmp.c
od_pedido,
      count(dmt.cod_or) pasos,
      sum(dmt.imp_total_or) gasto
  from tmp_arecort_renovadores tmp
  left join dim_vehiculos veh
  on tmp.cod_vehiculo_climapro = veh.cod_vehiculo_climapro
  left join ods_dmt_or_cabecera dmt
  on veh.bastidor = dmt.bastidor and cod_accion <> 'B'
  and flag_facturada = 1 and flag_c_manoobra = 1
  group by tmp.cod_vehiculo_climapro,tmp.cod_conductor,tmp.fecha_baja_cons,tm
p.cod_pedido) tmp2
  LEFT join (
  select cod_pedido,cod_conductor persona,fecha_alta_cliente from dim_pedidos
  where cod_marca = '36V'
  and fecha_alta_cliente > date'2017-01-01'
  union
  select cod_pedido,cod_propietario persona,fecha_alta_cliente from dim_pedid
os
  where cod_marca = '36V'
  and fecha_alta_cliente > date'2017-01-01'
  ) pedidos on
  tmp2.cod_conductor = pedidos.persona and tmp2.cod_pedido != pedidos.cod_pe
ido
  and pedidos.fecha_alta_cliente between add_months(tmp2.fecha_baja_cons,-
8) and add_months(tmp2.fecha_baja_cons,6)

```

1.6.4. Enquestes de satisfacció

```

select ped.cod_conductor,
encuestados.cod_respuesta,

```

```
case when pedidos.persona is null then 0 else 1 end flag_renovador
from tmp_arecort_renovadores ped
    LEFT join (
    select cod_pedido,cod_conductor persona,fecha_alta_cliente,cod_concesionario from dim_pedidos
    where cod_marca = '36V'
    and fecha_alta_cliente > date'2017-01-01'
    union
    select cod_pedido,cod_propietario persona,fecha_alta_cliente,cod_concesionario from dim_pedidos
    where cod_marca = '36V'
    and fecha_alta_cliente > date'2017-01-01'
    ) pedidos on
ped.cod_conductor = pedidos.persona and ped.cod_pedido != pedidos.cod_pedido
and pedidos.fecha_alta_cliente between add_months(ped.fecha_baja_cons,-8) and add_months(ped.fecha_baja_cons,6)
left join (
select cod_persona_climapro_i, tns.periodo_encuesta, cod_respuesta
from ODS_TNS_PEF pef
inner join ods_personas_en_fuente_l pel
on pef.cod_persona_en_fuente = pel.cod_persona_en_fuente
inner join ODS_TNS_ENCUESTA_PV tns
on pef.cod_encuesta = tns.cod_encuesta and pef.periodo_encuesta = tns.periodo_encuesta
and cod_pregunta = 'Q4'
where pef.cod_marca = '36V'
) encuestados on
ped.cod_conductor = encuestados.cod_persona_climapro_i
```

2. Scripts Python

2.1. Script iteració sobre paràmetres:

```

import pandas as pd
from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor
from sklearn.metrics import mean_absolute_error
from datetime import datetime

x = pd.read_excel('TFG_TRAIN_DATASET.xlsx')
x_test_full = pd.read_excel('TFG_EVAL_DATASET.xlsx')

y = x.FLAG_RENOVADOR
x.drop(['FLAG_RENOVADOR'], axis=1, inplace=True)
x.drop(['COD_PERSONA_CLIMAPRO_I'], axis=1, inplace=True)
x.drop(['COD_PEDIDO'], axis=1, inplace=True)

X_train_full, X_valid_full, y_train, y_valid = train_test_split(x, y, train_size=0.8, test_size=0.2, random_state=0)

X_train = pd.get_dummies(X_train_full)
X_valid = pd.get_dummies(X_valid_full)
X_test = pd.get_dummies(x_test_full)
X_train, X_valid = X_train.align(X_valid, join='left', axis=1)
X_train, X_test = X_train.align(X_test, join='left', axis=1)

results = {}
n_est = 10000
n_learn = 0.05

for i in range(11):
    startTime = datetime.now()
    my_model = XGBRegressor(n_estimators = n_est, learning_rate = n_learn, n_jobs = 4)
    my_model.fit(X_train,y_train,
                  early_stopping_rounds = 5,
                  eval_set = [(X_valid, y_valid)],
                  verbose = False)

    predictions = my_model.predict(X_valid)

    for prediction in range(len(predictions)):
        if predictions[prediction] >= 0.5:

```

```

        predictions[prediction] = 1
    else:
        predictions[prediction] = 0

results[n_learn] = (str(mean_absolute_error(predictions, y_valid)),str(datetime.now() - startTime))

#n_est = n_est + 500
n_learn = round(n_learn - 0.2*n_learn,4)

print("\n learning_rate" + "          " + "error"+ "          " + "time")
print("_____" + "          _____")
for result in results:
    print(str(result) + "          " + str(results[result][0])+ "          " + str(results[result][1]))

```

2.2. Script per a la validació creuada del model:

```

from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer

my_pipeline = Pipeline(steps=[
    ('preprocessor', SimpleImputer()),
    ('model', XGBClassifier(n_estimators = 1000, learning_rate = 0.05, n_jobs
s = 4))
])

from sklearn.model_selection import cross_val_score

scores = -1 * cross_val_score(my_pipeline, X_train,y_train,
                               cv=5,
                               scoring='neg_mean_absolute_error')

print("Average MAE score:", scores.mean())

scores

```

2.3. Script per l'avaluació de l'entrenament:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor

x = pd.read_excel('TFG_TRAIN_DATASET.xlsx')
x_test_full = pd.read_excel('TFG_EVAL_DATASET.xlsx')

y = x.FLAG_RENOVADOR
x.drop(['FLAG_RENOVADOR'], axis=1, inplace=True)
x.drop(['COD_PERSONA_CLIMAPRO_I'], axis=1, inplace=True)
x.drop(['COD_PEDIDO'], axis=1, inplace=True)

X_train_full, X_valid_full, y_train, y_valid = train_test_split(x, y, train_size=0.8, test_size=0.2, random_state=0)

X_train = pd.get_dummies(X_train_full)
X_valid = pd.get_dummies(X_valid_full)
X_test = pd.get_dummies(x_test_full)
X_train, X_valid = X_train.align(X_valid, join='left', axis=1)
X_train, X_test = X_train.align(X_test, join='left', axis=1)

my_model = XGBRegressor(n_estimators = 1000, learning_rate = 0.04, n_jobs = 4)
my_model.fit(X_train,y_train,
             early_stopping_rounds = 5,
             eval_set = [(X_valid, y_valid)],
             verbose = False)

predictions = my_model.predict(X_valid)

X_valid['Flag_renovador'] = y_valid
X_valid['Predictions'] = predictions
X_valid.to_excel('TFG_TRAINING_OUTPUT.xlsx')
```

2.4. Script per a la utilització del predictor:

```

import pandas as pd
from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor

x = pd.read_excel('TFG_TRAIN_DATASET.xlsx')
x_test_full = pd.read_excel('TFG_EVAL_DATASET.xlsx')

y = x.FLAG_RENOVADOR
x.drop(['FLAG_RENOVADOR'], axis=1, inplace=True)
x.drop(['COD_PERSONA_CLIMAPRO_I'], axis=1, inplace=True)
x.drop(['COD_PEDIDO'], axis=1, inplace=True)
x_test_full.drop(['COD_PERSONA_CLIMAPRO_I'], axis=1, inplace=True)
x_test_full.drop(['COD_PEDIDO'], axis=1, inplace=True)

X_train_full, X_valid_full, y_train, y_valid = train_test_split(x, y, train_size=0.8, test_size=0.2, random_state=0)

X_train = pd.get_dummies(X_train_full)
X_valid = pd.get_dummies(X_valid_full)
X_test = pd.get_dummies(x_test_full)
X_train, X_valid = X_train.align(X_valid, join='left', axis=1)
X_train, X_test = X_train.align(X_test, join='left', axis=1)

my_model = XGBRegressor(n_estimators = 1000, learning_rate = 0.04, n_jobs = 4)
my_model.fit(X_train,y_train,
             early_stopping_rounds = 5,
             eval_set = [(X_valid, y_valid)],
             verbose = False)

predictions = my_model.predict(X_test)

x_test_full['Predictions'] = predictions
x_test_full.to_excel('TFG_DATASET_OUTPUT.xlsx')

```

2.5. Script per a la instal·lació de l'entregable:

Desar el contingut d'aquest enllaç en un fitxer anomenat **get-py.py**

<https://bootstrap.pypa.io/get-pip.py>



3. Script .bat

Script per a la instal·lació de les llibreries necessàries un cop es disposa de python al equip i del codi del apartat anterior desat als arxius corresponents. Per a fer-ho, simplement desar les següents línies de codi en un fitxer .bat i fer doble click.

```
python get-pip.py  
pip install pandas  
pip install sklearn  
pip install xgboost
```


