```
/* ------- import of the dataset ------- */
proc import datafile="/home/u59902206/Spotify/data_o.csv"
        out=spotify
        dbms=csv
        replace;
        getnames=yes;
run;

/* --------- create the appropriate library --------  */
LIBNAME spotify '/home/u59902206/';
    DATA spotify.data;
    SET spotify;
RUN;

/* ------ check for missing values ------- */
proc means data=spotify N Nmiss;
run;
/* no missing values  */

/* create new variable  */
DATA spotify.new;
set spotify;
if popularity < 50 then popular="0";
else
    if popularity <70 then popular="1";
    else popular="2";
run;

proc print data=spotify.new;
run;

/* ---------------------- statistical analysis ---------------------- */

/* --- frequency table ---- */
proc freq data=spotify.new;
table explicit popular;
run;

/* --- Crosstabulation matrix ---- */
proc freq data=spotify.new;
table explicit*popular;
run;

/* --- boxplot ---  */
title "Popularity and explicit content";
PROC SGPLOT  DATA = spotify.new;
   VBOX popularity / category = explicit;
RUN;
options gstyle;
goptions reset=symbol;
title;

/* --- descriptive statistics ---- */
PROC MEANS DATA = spotify.new;
var acousticness danceability duration_ms instrumentalness loudness speechiness popularity;
RUN;


/* Test Normality */
Proc univariate data=spotify.new;
    HIST acousticness / normal;
    HIST danceability / normal;
    HIST duration_ms / normal;
    HIST instrumentalness / normal;
    HIST loudness / normal;
    HIST speechiness / normal;
    HIST popularity / normal;
run;

/* multicollinearity */
proc corr data=spotify.new;
var acousticness danceability duration_ms instrumentalness loudness speechiness popularity;
run;
```

```
/* ----------------- split of the dataset ---------------- */

/* --- train data ---  */

proc surveyselect data=spotify.new
    out=train_spotify
    method=srs
    sampsize=119000
    seed=1;
run;

/*--- test data--- */

proc surveyselect data=spotify.new
    out=test_spotify
    method=srs
    sampsize=51000
    seed=2;
run;


/* --- knn --- */

proc discrim data=train_spotify
testdata=test_spotify
testout=testout
method=npar
k=5;
class popular;
var acousticness danceability duration_ms instrumentalness loudness speechiness;
run;

/* --- logistic regression ---  */

proc logistic data = train_spotify descending;
  class explicit / param=glm;
  model explicit = acousticness danceability duration_ms instrumentalness loudness speechiness;
  output out=outdata p=pred_prob lower=low upper=up;
run;


/* ------------------------- Efficient Programming ---------------------------------------  */

DATA spotify_efficient;
set spotify;
keep acousticness danceability duration_ms instrumentalness loudness speechiness popularity explicit popular;
if popularity < 50 then popular="0";
else
    if popularity <70 then popular="1";
    else popular="2";
run;

/* --- train data effictient ---  */

proc surveyselect data=spotify_efficient
    out=train_spotify_eff
    method=srs
    sampsize=119000
    seed=1;
run;

/* --- test data efficient --- */

proc surveyselect data=spotify_efficient
    out=test_spotify_eff
    method=srs
    sampsize=51000
    seed=2;
run;


/* --- knn efficient --- */

proc discrim data=train_spotify_eff
testdata=test_spotify_eff
```

```
testout=testout_eff
method=npar
k=5;
class popular;
var acousticness danceability duration_ms instrumentalness loudness speechiness;
run;

/* --- logistic regression efficient ---   */

proc logistic data = train_spotify_eff descending;
  class explicit / param=glm;
  model explicit = acousticness danceability duration_ms instrumentalness loudness speechiness;
  output out=outdata p=pred_prob lower=low upper=up;
run;
```