



Seif El-Din El-Moghazy  
Technical Assessment Report

# 1. Summary

-In my technical assessment, I did the following:

- 1) Trained the provided model and saved it using pickle in order to use in my API
  - 2) Used Flask to develop an API that takes data from users, communicate it to the machine learning model (loaded by pickle), get the top recommended products, and send it back to users
  - 3) Used Flask Restx to aid in generating a swagger documentation for my api
- Documentation includes explanation of the POST request developed, with explanations of the request parameters, the response parameters, and the different responses (with http codes) the user might receive

```
Request-Fields {
  sexo* string
    example: V
    Customer's sex
    Enum:
      > Array [ 2 ]
  age* integer
    Customer's Age
  segmento* string
    example: 01 - TOP
    segmentation: 01 - VIP, 02 - Individuals 03 - college graduated
    Enum:
      > Array [ 3 ]
  antigüedad* integer
    Customer seniority (in months)
  tiprel_1mes* string
    example: 1
    Customer type at the beginning of the month ,1 (First/Primary customer), 2 (co-owner ),P (Potential),3 (former primary), 4(former co-owner)
    Enum:
      > Array [ 4 ]
  renta* integer
    Gross income of the household
  pais_residencia* string
    Customer's Country residence
  ind_actividad_cliente* string
    example: 1
    Activity index (1, active customer; 0, inactive customer)
    Enum:
      > Array [ 2 ]
  nomprav* string
    Province name
}
```

Figure A Request fields documentation

Code	Description
200	Success
	Example Value   Model
	<pre>{   "Prededctions": "string" }</pre>
400	Validation Error

Figure B Response Documentation

- The code consists of 3 main parts:
  - 1) The Flask API
  - 2) Data processing class, which is responsible of processing the data received into the form accepted by the model
  - 3) Data prediction class, which is responsible of loading the model with the extension passed at the instant creation, in addition to predicting the data received from the processing class
- After building the API, I wrote the dockerfile responsible of generating the docker image for a container to run it

## 2. Request-Response Pairs

- To run requests, use either the swagger documentation or Postman with the following url and request body

1) Request URL: <http://localhost:5000/Predictor/Predict>

Request Body:

```
{
  "sexo": "H",
  "age": 22,
  "segmento": "03 - UNIVERSITARIO",
  "antiguedad": 34,
  "tiprel_1mes": "I",
  "renta": 106885.8,
  "pais_residencia": "ES",
```

```
"ind_actividad_cliente": "0",  
"nomprav": "BALEARs, ILLES"  
}
```

Response:

```
{  
  "Predictions": "ind_cco_fin_ult1 ind_recibo_ult1  
ind_nom_pens_ult1 ind_cno_fin_ult1 ind_nomina_ult1  
ind_ecue_fin_ult1 ind_reca_fin_ult1"  
}
```

2) Request URL: <http://localhost:5000/Predictor/Predict>

Request Body:

```
{  
  "sexo": "V",  
  "age": 56,  
  "segmento": "01 - TOP",  
  "antiguedad": 256,  
  "tiprel_1mes": "A",  
  "renta": 326124.9,  
  "pais_residencia": "ES",  
  "ind_actividad_cliente": "0",  
  "nomprav": "MADRID"  
}
```

Response Body:

```
{ "Predictions": "ind_cco_fin_ult1 ind_tjcr_fin_ult1  
ind_recibo_ult1 ind_ecue_fin_ult1 ind_nom_pens_ult1  
ind_nomina_ult1 ind_dela_fin_ult1" }
```

3) Request URL: <http://localhost:5000/Predictor/Predict>

Request Body:

```
{  
  "sexo": "H",  
  "age": 36,
```

```
"segmento": "02 - PARTICULARES",
"antiguedad": 34,
"tiprel_1mes": "I",
"renta": -99,
"pais_residencia": "ES",
"ind_actividad_cliente": "0",
"nomprav": "ALICANTE"
}
```

Response Body:

```
{
  "Predictions": "ind_cco_fin_ult1 ind_recibo_ult1
ind_nom_pens_ult1 ind_nomina_ult1 ind_cno_fin_ult1
ind_tjcr_fin_ult1 ind_ecue_fin_ult1"
}
```

Note that -99 in field “renta” is equivalent to NA or ‘ ‘ since that the field is set to a Float in Request-fields model

### 3. Bonus Tasks

- a) Please find attached in the github repo the dockerfile that will be used to build a docker image
- b) For API expansion to serve multiple ML models, dataPrediction class which receives a model’s name (or extension) which can grab the specified model from either our server or a cloud storage such as that of AWS.  
In addition, in order to facilitate the model selection of the user, we would ask the user to enter his/her bank name , which will be mapped to our models , with other data.

Diagram:

