EUREKATHON

Challenging Data for Zero Hunger

Powered by

Evaluation Deliverable template

This document is one of the items that you need to submit and that will be evaluated. \ Please fill in the relevant information below, by placing in each cell all the information related to its title, if pertinent to the problem you have chosen to address. The title of each cell corresponds to certain eveluation criterion. You can find more details on the evaluation process by consulting the *Evaluation Criteria* document. \ \ Information that you should be providing in this document may include description of approaches or data that was used, images and plots that illustrate your results and whatever you see fit to demonstrate your analysis. You may also link here any other relevant information used in solving the case. You may additionally include any other topic that you think is relevant to your approach. The code for solving the problem should be submitted in a separate notebook, and should not be included in this one (check the deliverables list on the *Participant Guide*.

Note: Please ensure that you provide the .html file and not the .ipynb itself (for compatibility reasons). You can do `File >> Export Notebook as >> HTML` on your jupyter notebook.

Fill in your group name

FeedingTheFuture

I. Impact, applicability and creativity

1. State clearly the problem you are solving

As the number of people calling for help is increasing, it is important to speed up the process of evaluation and prioritization of the families. Our solutions allows for triage of the cases.

2. Give a brief overview of the proposed solution

Create a model that predicts probability of a request being approved.

3. Describe how would you apply your solution in practice

Based on the data from the questionaire filled on Banco Alimentar website and the information of whether historical request where approved or not, the model can be built to predict the outcome of a request. Based on that, the request can be prioritized. Insights from exploratory data analysis, outcome of a model and feature importance cnan be used to refine the model in the future.

II. Analytical component

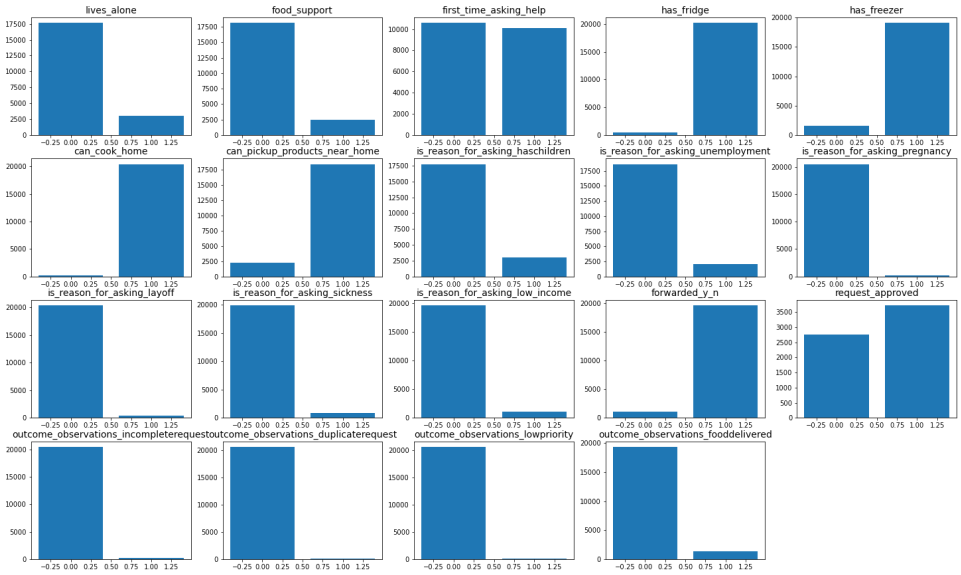
1. Exploratory Data Analysis

bin_features ="lives_alone", "food_support", "first_time_asking_help", "has_fridge", "has_freezer", "can_cook_home", "can_pickup_products_near_home", "is_reason_for_asking_haschildren", "is_reason_for_asking_unemployment", "is_reason_for_asking_pregnancy", "is_reason_for_asking_layoff", "is_reason_for_asking_sickness", "is_reason_for_asking_low_income", "request_approved"] num_features = "age_bin", "number_people_household", "number_kids_less_10yr" cat_features = "zinf", "county", "cp4", 'district',"forwarding_month_id", "month_id", 'request_reason'

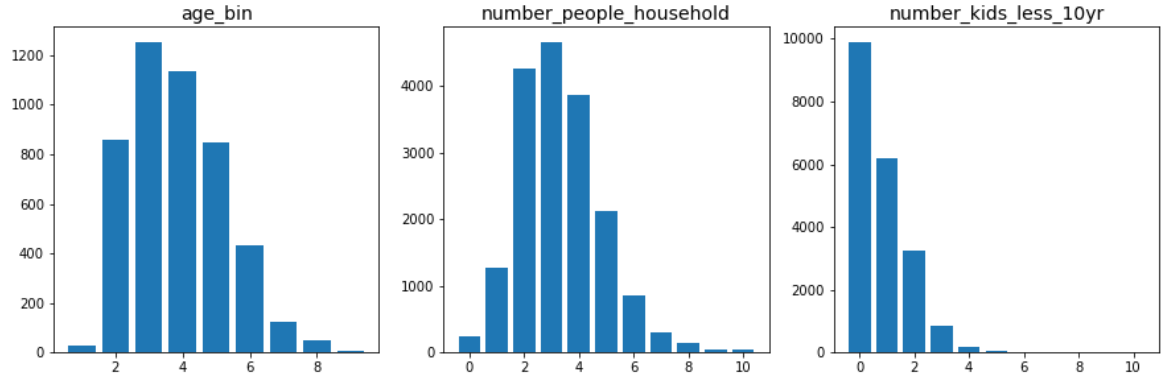
```
requests[num_features].describe()
```

	age_bin	number_people_household	number_kids_less_10yr
count	4735.000000	17835.000000	20496.000000
mean	3.845829	3.320549	0.824990
std	1.414322	1.540097	1.047616
min	1.000000	0.000000	0.000000
25%	3.000000	2.000000	0.000000
50%	4.000000	3.000000	1.000000
75%	5.000000	4.000000	1.000000
max	9.000000	10.000000	10.000000

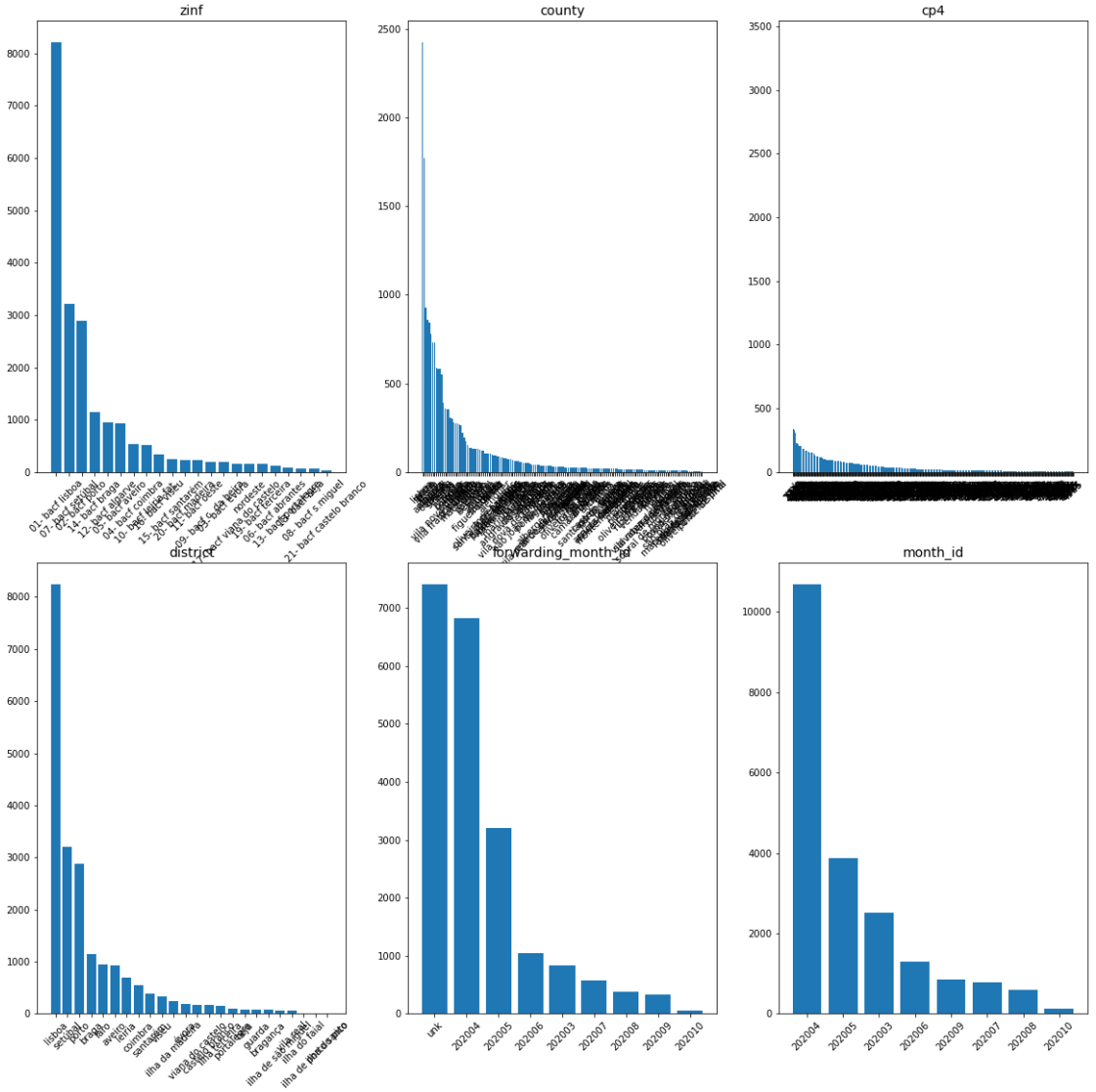
Categorical features:

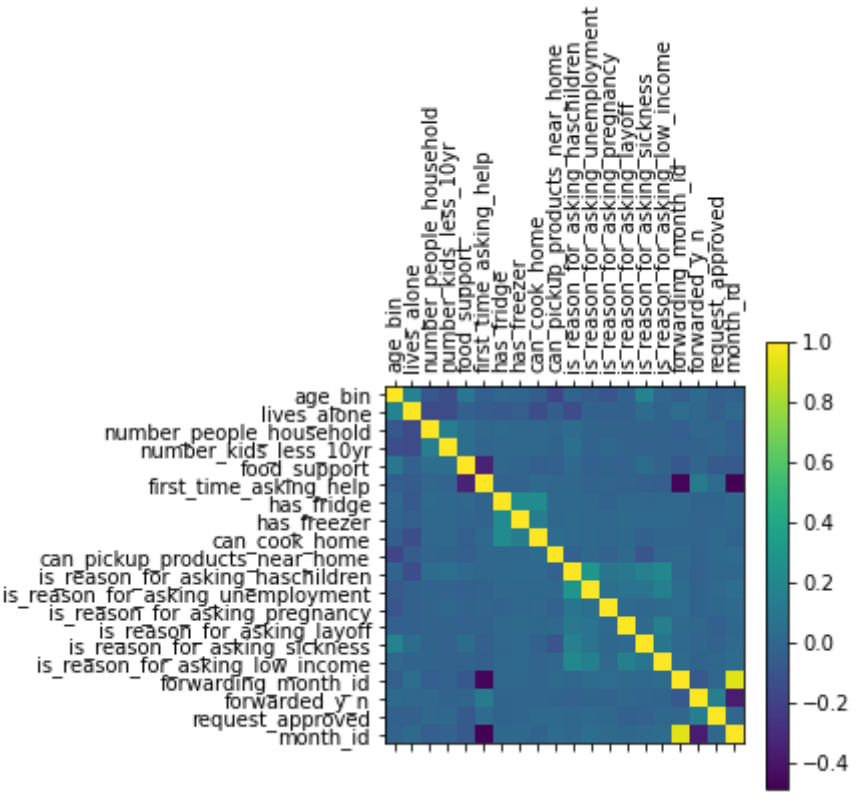


Numerical features:



Categorical features:


zinf
county
cp4
district
forwarding_month_id
month_id


age_bin
lives_alone
number_people_household
number_kids_less_10yr
food_support
first_time_asking_help
has_fridge
has_freezer
can_cook_home
can_pickup_products_near_home
is_reason_for_asking_haschildren
is_reason_for_asking_unemployment
is_reason_for_asking_pregnancy
is_reason_for_asking_layoff
is_reason_for_asking_sickness
is_reason_for_asking_low_income
forwarding_month_id
forwarded_y_n
request_approved
month_id

2. Variables chosen for problem solving

Following features were chosen to build the model:'zinf', 'county', 'cp4', 'age_bin', 'lives_alone', 'number_people_household', 'number_kids_less_10yr', 'food_support','first_time_asking_help', 'has_fridge', 'has_freezer', 'can_cook_home', 'can_pickup_products_near_home', 'request_reason', 'is_reason_for_asking_haschildren', 'is_reason_for_asking_unemployment', 'is_reason_for_asking_pregnancy', 'is_reason_for_asking_layoff', 'is_reason_for_asking_sickness', 'is_reason_for_asking_low_income', 'district', 'forwarding_month_id', 'forwarded_to', 'forwarded_y_n', 'request_approved', 'month_id'

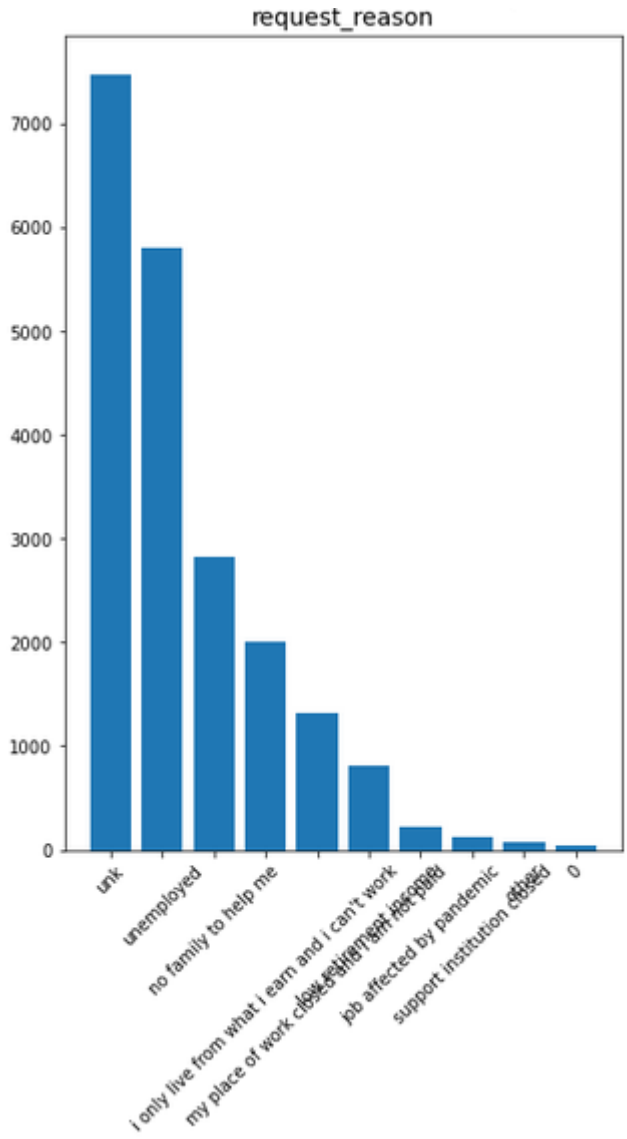
Categorical features with less than 30 categories was converted into dummy variables resulting in 84 columns. Categorical features that were not converted into dummy variables were excluded.

3. Statement of assumptions made

Data was filtered by feature "reason not approved". Simply rejectin someone does not mean that they are not eligible for help. Following reasons were assumed to not to make a family eligible for help:

```
requests["reason_not_approved"].value_counts()
```

```
Could not contact the person                                454
Analysis in process                                         425
Already receives support from another institution           422
other (please specify in the observations)                  330
Address of the request is outside our action zone           232
Does not require help anymore                               211
Invalid request (please specify in the observations)        206
                                                             84
Lack of capacity (Food/HR/Distribution)                      43
Lack of stock/food                                           33
Lack of capacity (HR/Distribution)                           12
No justification for support                                  2
Family is already receiving food support                      2
```



4. Overview of modelling approaches used

Class counts in the target variable "request_approved":

```
1.0    3704
0.0    1380
```

Based on that data was balanced- the majority "approved" class was undersampled

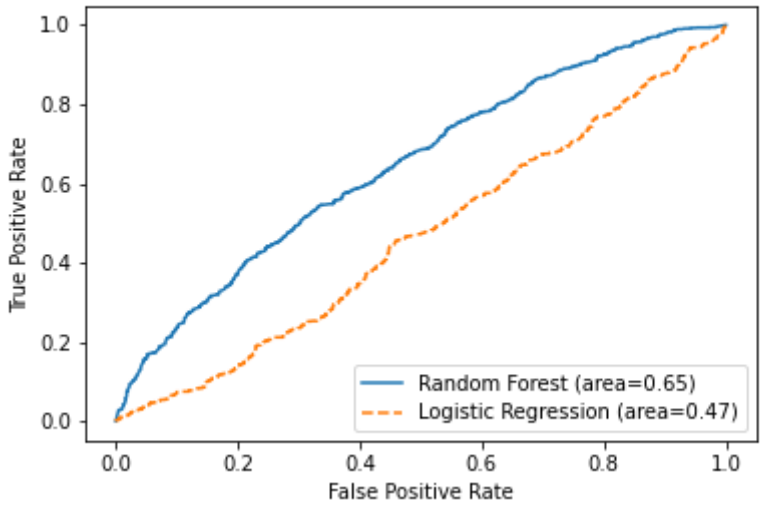
Logistic regression model was build as a benchmark, followed by Random Forest model.

5. Evaluation and discussion of results and model performance

The performance of the Random Forest classifier was evaluated by crossvalidated F1 and AUC scores:

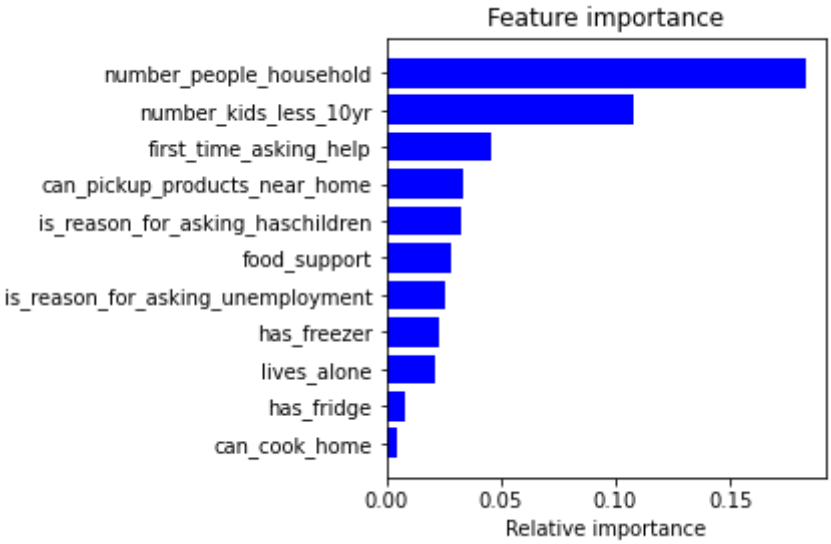
```
F score: 0.678 (0.022)
AUC: 0.715 (0.022)
```

The performance of both models was plotted on ROC curve:



6. Model interpretability and fairness

Random Forest models allows for a great deal of interpretability.



7. Data and results visualization

Donor Profiling

In NOS dataset, we analysed the available NOS data collections in order to find useful information to build our donor profile.

1. Exploratory Data Analysis

Description of included features

- client_list_eurekathon

features: "sa", "is_donor"

- client_mobile_data_eurekathon

features: "rate_plan"

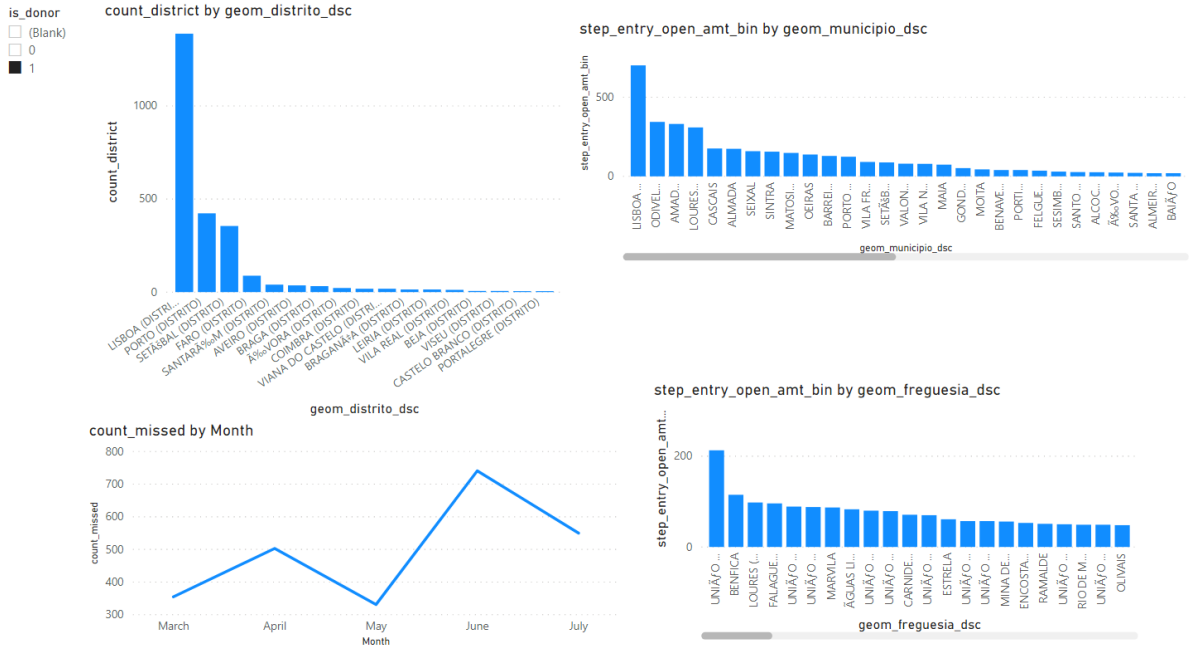
- client_tv_data_eurekathon

features: "segment", "package_type"

- client_profile_data_eurekathon

features: 'premium_a_qty', 'premium_b_qty', 'premium_c_qty', 'premium_d_qty', 'premium_a_amt_bin', 'premium_d_amt_bin', 'geom_distrito_dsc', 'geom_nuts_iii_dsc', 'pack_dsc', 'cp4'

2. Useful features count



Decision tree test result

```
Accuracy:  0.6476345840130505
Precision:  0.69
Recall:  0.6272727272727273
F1 score:  0.6571428571428571
```

Random Forest test result

```
Accuracy:  0.6818923327895595
Precision:  0.6955017301038062
Recall:  0.6090909090909091
F1 score:  0.6494345718901454
```

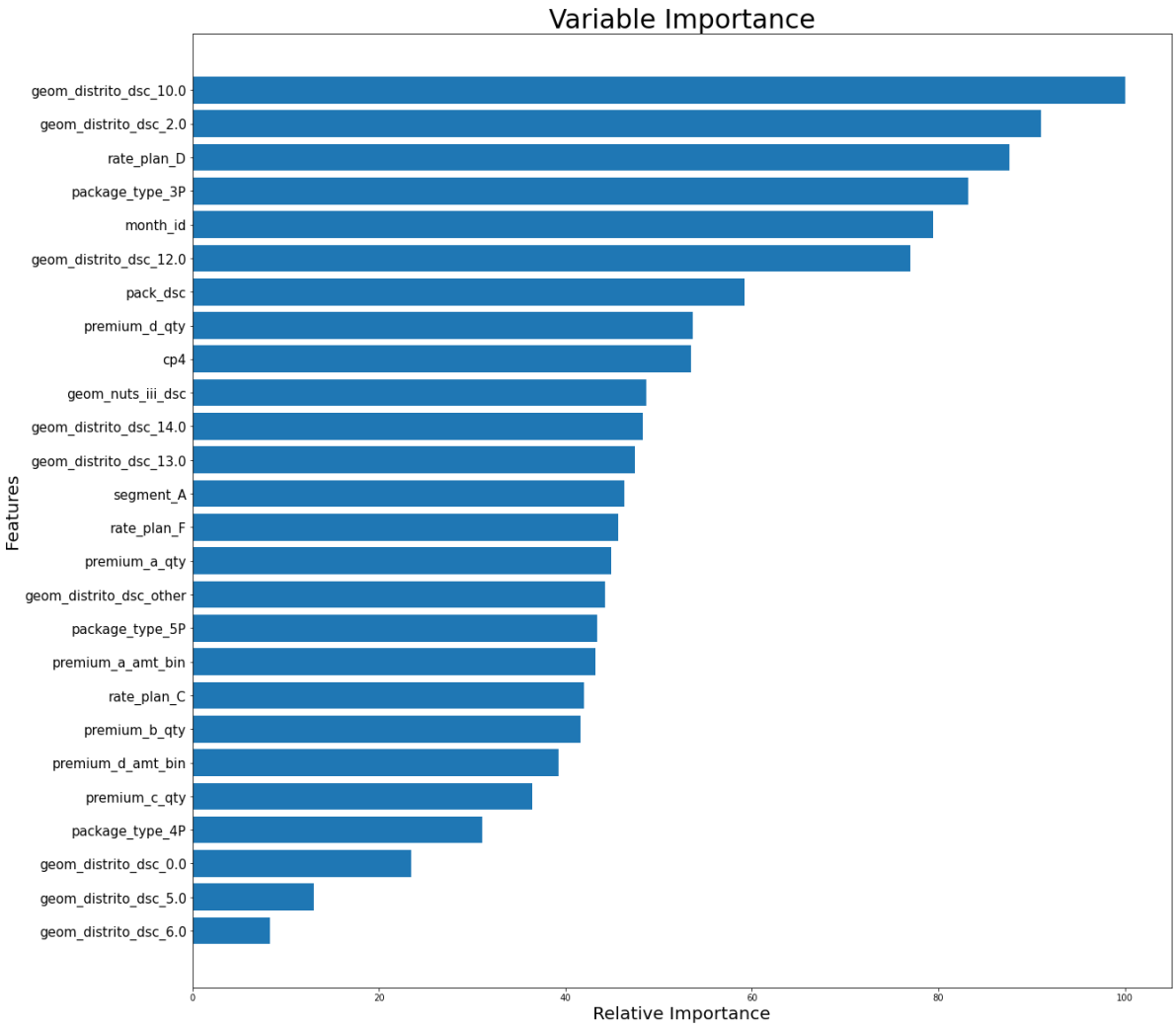
XGBoost test result

```
 {'n_estimators': 300, 'max_depth': 4, 'learning_rate': 0.1, 'subsample': 0.7, 'colsample_bytree': 0.7, 'eval_metric': 'log
loss', 'seed': 42, 'reg_alpha': 0.85, 'reg_lambda': 0.85}
Done! Elapsed time: 2.0

Model Train Report
Accuracy:  0.906927921623513
Precision:  0.9067055393586005
Recall:  0.9001447178002895
F1 score:  0.9034132171387074

CV Score: Mean - 0.6377 | Std - 0.0288 | Min - 0.6071 | Max - 0.6840

Model Test Report
Accuracy:  0.6786296900489397
Precision:  0.726962457337884
Recall:  0.6454545454545455
F1 score:  0.6837881219903693
```



XGBoost delivered the best results. The feature importance indicates that the geolocation of the clients is likely to be a good indicator of whether they are a donor or not.

In [ ]: