Applying Machine Learning Methods in Missing People Classification

Carlos Gomes de Oliveira*, Eduardo Terra Morelli†, and Raul Senna Ferreira‡
Instituto INFNET
Escola Superior da Tecnologia da Informação
Rio de Janeiro, RJ
Email: *czgrqg@gmail.com, †emorelli1966@gmail.com, ‡raul.ferreira@prof.infnet.edu.br

***Abstract*—The missing people database offers several challenges regarding its analysis. The database may be incomplete since usually there is not a commitment from the authorities to keep consolidated and detailed information on missing persons. There are many apparent uncorrelated attributes, as different persons have different attribute groups. Many of these records contain attributes with incorrect and missing information.**

This paper intends to demonstrate the use of machine learning methods as Decision Trees, Random Forests, Logistic Regression, and Extreme Gradient Boosting to classify the data in appropriate classes. The selection of these methods is not random, one of the reasons is that they allow us to determine the most useful attributes and have this usefulness quantified.

The classification of the data gives us insight into the groups within our studied population, with quantified metrics and rules obtained from the models.

***Index Terms*—classification, knowledge acquisition, Decision Trees, Extreme Gradient Boosting, XGBoost, Random Forests, Logistic Regression, missing people**

I. Introduction

Within any given country there are some regional databases, and sometimes a national database, however, it is not mandatory. When the authorities act, it is usually on a local basis, and the evidence that could solve the case may be in another database of another state or even country. It is usually a task left to the families to visit offices from several different governmental areas, search the Internet, on-line databases, social networks, hospitals, and morgues. For these families, any help is welcome.

The task of dealing with this data is daunting as it is difficult to find patterns. Besides, the records are usually incomplete and most data is sparse. Even when official databases exist on a local or national basis it is rarely mandatory for the authorities to register there the information available. Different databases have different structure and organization.

The objective of this paper is to demonstrate how useful it would be to find patterns and individuals in this data and to determine what features or missing people attributes are critical to establishing if they will go missing or be found. In any missing people database there is a special group of people we need to find, people with risk social profiles (mainly murdered for drug trafficking related reasons), another critical group is children or women, due to human trafficking.

In order to accomplish that, it is necessary to gain insight into the data, to find the boundaries among the groups in the community represented by the data, the rules, and characteristics that distinguish them. This can be accomplished through machine learning methods.

II. Related Work

Machine learning algorithms are widely used for prediction, for example, Cornel and Mirela used Decision Trees [1] to predict economic forecasts for a university [2], by the American Rheumatism Association in 1987 to revise criteria for the classification of rheumatoid arthritis [3], and to predict rates of relapse in subgroups of male and female smokers [4].

Random Forests have been used from Gene Classification [5] to Image Classification [6].

Logistic Regression have been use to predict the performance of habitat models [7], for landslide susceptibility assessment [8] and Cancer Classification [9]. It is a great classification and prediction tool, and we used it as a baseline for algorithm accuracy comparison.

Extreme Gradient Boosting [10] have been used to study fish species richness in the oceans surrounding New Zealand [11] and to classify remotely sensed imagery [12].

III. Machine Learning Methods in Missing People Classification

In this paper, we used Decision Trees (DT), Random Forests (RF), Logistic Regression (LR) and eXtreme Gradient Boosting or XGBoost (XGB).

Logistic Regression and Decision Trees were used for its long history, and well documented results as a baseline for comparison to other methods.

Since RF introduction in 2001 [13], it has been widely used in data classification and regression, and more recently XGBoost has shown great results and its acceptance is growing as demonstrated in several Kaggle [14] competitions.

Before training any model the data should be analyzed, cleansed, and imputed. Data cleansing is a process of transforming the original data in a way that keeps its accuracy and improves its usability by programs, specially machine learning algorithms, that are very sensitive to discrepancies in data. One data cleansing procedure is described in (Real-world data is dirty: Data cleansing and the merge/purge problem) [15]. Finally, imputation is a statistical method to fill in missing values [16].

A frequent concern is to be sure if the model is not overfitting. As stated by Douglas M. Hawkins (The problem of overfitting) [17], overfitting is the use of models that include more terms than are necessary or use more complicated approaches than are necessary.

A. Random Forests

We start with the use of Random Forests (RF) to perform classification analysis on missing person data. RF is a powerful and resilient algorithm in comparison to other top performer algorithms. It is a variation of the basic supervised learning model implementing decision trees, but creating a multitude of trees, hence its name.

The resulting class is collected from these trees and the most frequent classification of regression is a step in the right direction of obtaining a pure (correct) result. Since it was branded and introduced by Breiman (Breiman, 2001), it has proved its usefulness, especially on its strengths as outlined in his original paper [13]:

- Its accuracy is considered as good as or better than Adaboost.
- It is quite robust to outliers points and noise in data.
- I also return extremely useful information about the model, such as the variable importance, internal estimates of error, strength, and correlation.
- It is considered faster than traditional bagging or boosting.
- It can be easily parallelized and it is simple to use.
- As a result of the consolidation of the parallel trees, it does not overfit.
- And finally, it performs similarly on both continuous and categorical features in the dataset. All these characteristics are mentioned by those who increasingly use it in fields from image analysis and genetics to application log and business data classification and regression analysis.

B. eXtreme Gradient Boosting - XGBoost

XGBoost [18] can be seen in a similar way to RF, since its structure is also based on the basic model of supervised learning. It also has an objective function that describes the training loss and the regularization function, and similarly to RF it uses trees as a learning mechanism.

Differently from RF, which creates several trees at the same time and consolidates the result, XGBoost creates one tree at a time, learns from its results, minimizing loss and complexity (optimizing the objective function) and builds a new tree. It has a surprisingly good rate for error reduction at each subsequent tree creation, improving performance and result accuracy.

- It is flexible, as it supports regression, classification, ranking, and user defined objectives.[18]
- It is portable, it runs on Windows, Linux and OS X, as well as various cloud Platforms.[18]
- It supports multiple languages, such as C++, Python, R, Java, Scala, Julia.[18]
- It is extremely tunable, with several parameters controlling most weights, methods and algorithms used for learning, training loss reduction and regularization.[19]
- It learns faster than other algorithms.[19]

C. Logistic Regression

Logistic regression is a model where the dependent variable is categorical and has two values, usually binary and either representing two states (state-A or state-B), or one state and off that state (state-A or NOT-state-A). If we have more than two categories or classes use the multinomial logistic regression or the ordinal logistic regression, if the categories are ordered.

This method was developed in 1958 by David Cox [20], and is still widely used. In this study, it will be used as a baseline model for efficiency analysis.

IV. Experiments and Results

The objective of this experiment is to analyze data collected from several missing people databases and find patterns in the data. We used the person found/missing attribute as a prediction class and built a prediction model on a selection of the other features to predict this class. We expect to find in attribute relevance to the model accuracy an important insight into the data and the problem of missing people. Besides, it is possible to extract the tree structure and learn the rules it used to predict the outcome.

This database was obtained from several other missing people databases all over Brazil. It has a relatively small number of records, as it reflects the lack of information on missing people. Several attributes (features) are missing, with incorrect value or wrong scale/unity, which is also a measure of how much these individuals are not relevant to the authorities.

Besides the prediction class, found/missing, there are two groups of features that can be seen in Table I. They can be quantitative, such as age or a number of days; or qualitative, as the sex or skin color. The first group is people characteristic features, such as age, sex, eye color, hair color, weight, height and mental impairment. The second is the time-frame the person went or is still missing, such as the number of days the person has gone missing.

Since this is a very heterogeneous database, at first some feature engineering had to be performed. The first phase is data cleansing. We had numeric features that contained text and special characters, they had to be parsed in order to get the actual value of the feature. Some categoric features also contained special characters and the categories had aliases. Another problem was quantitative features with mixed unities, such as meters and centimeters or kilograms and grams, so we had to perform unity conversion based on value range. Finally, I converted the values types so that they could be consistent with the feature data-type.

The second phase of feature engineering related to feature value correction and imputing. The missing values of some features were imputed from relations obtained from other features. For instance, height and weight features relate to

TABLE I
FEATURES USED IN MODEL

Feature Name	Feature Type
Days Missing	Quantitative
Height	Quantitative
Weight	Quantitative
Age	Quantitative
Physical Type	Categorical
Skin Color	Categorical
Eye Color	Categorical
Sex	Categorical
Hair Color	Categorical
Mental Impairment	Categorical

each other through the BMI (Body Mass Index). Even though the BMI have some limitations [21], it was our best shot for data imputation on weight and height features. A routine was implemented to fill in missing values of these two features using this formula.

When both height and weight were missing, we used census statistics [22] on the target population to establish weight and height values that could be imputed for the corresponding age groups.

We also found out that some numeric features had too many distinct values and the ratio of the number of records to each distinct value was too small, sometimes 1 for specific value ranges. This lead some models to overfit over these features and to reduce the accuracy of the model. We used binning methods to solve this problem. For features with a limited range, as height, weight or age, the values were rounded up to the nearest 3 or 5 multiple. In addition, for features with longer ranges and varying density of values for each data sub-range, those values were rounded up to the nearest tenth power, chosen to smooth the data density in each data sub-range. For example, the missing days feature has a very wide range, and commonly with very few or even only 1 record per value, which resulted in overfitting the training data. After the analysis of how value density varied, we came up with this binning method. For less than 1.000 days, values were rounded up by 50 days; between 1.000 and 6.000, rounded up by 100 days; between 6.000 and 10.000, rounded up by 1.000 days; and finally, those between 10.000 and 100.000 were rounded by 10.000 days. With this adjustment, the records were grouped in fewer values of the missing days feature, enough to avoid individual identification and still maintaining prediction value. It was this very adjustment that increased accuracy by almost 5%. Additionally, the accuracy value became much more stable.

Initially, all features with numeric or categoric values were selected. Then, as the algorithms returned the feature importance in relation to each other and how much they improved or worsened the prediction accuracy. View Table III for Decision Trees, Table IV for Random Forests, Table V for Logistic Regression and Table VI for feature importance

TABLE II
MODEL ACCURACY

Algorithm	Test#1	Test#2	Test#3	Test#4
Decision Trees	69.82%	68.39%	68.68%	67.24%
Random Forest	**80.46%**	79.02%	**81.32%**	77.58%
Logistic Regression	75.86%	72.70%	75.29%	71.55%
XGBoost	**81.03%**	79.02%	79.31%	77.59%

TABLE III
FEATURE IMPORTANCE FOR DECISION TREES

Importance	Feature Name
1	Mental Impairment
2	Days Missing
3	Age
4	Sex
5	Eye Color
6	Hair Color
7	Skin Color
8	Physical Type
9	Weight
10	Height

in XGBoost. This information was used to select a new group of features. So, after each prediction iteration, the importances were assessed and a new mix of features was selected.

We built a single routine to perform all data cleansing and feature engineering. The clean dataset was written to disk in CSV format.

Finally, after the data clean-up and imputing comes the model execution. Another routine loaded the clean dataset. In this step, we had the one-hot encoding routine. One-hot encoding is a technique that comes from digital computer architecture, as described in David and Sarah Harris in Digital design and computer architecture[23]. It takes a feature with several distinct values, and create several features, one for each distinct value. For example, the feature sex can have the values male and female, after this technique is applied, the result is two features, sex-male with values of true or false, and sex-female, with the same range of values.

We found out that the generic use of this routine was skewing the results, and that without it, the accuracy increased. Then we executed all models, training over 80% of the data and prediction over 20%, we calculated an accuracy rate of this prediction, and obtained a feature importance graph.

Decision trees reached an accuracy of 65% to 70%, Logistic Regression 70% to 75%, Random Forests and XGBoost 77% to 81%. Considering Logistic Regression as our baseline, we had both Random Forests and XGBoost reaching similar results, although we expected more of XGBoost as Random Forests reached similar and sometimes better accuracy. Some results can be seen in Table II. In all model training and prediction, we used the same split of 80/20 for train and test data. The same features were used for prediction in all models.

TABLE IV
FEATURE IMPORTANCE FOR RANDOM FORESTS

Importance	Feature Name
1	Days Missing
2	Age
3	Mental Impairment
4	Physical Type
5	Height
6	Skin Color
7	Sex
8	Eye Color
9	Weight
10	Hair Color

TABLE V
FEATURE IMPORTANCE FOR LOGISTIC REGRESSION

Importance	Feature Name
1	Mental Impairment
2	Eye Color
3	Physical Type
4	Height
5	Days Missing
6	Weight
7	Age
8	Skin Color
9	Sex
10	Hair Color

TABLE VI
FEATURE IMPORTANCE FOR XGBOOST

Importance	Feature Name
1	Days Missing
2	Age
3	Skin Color
4	Eye Color
5	Weight
6	Height
7	Sex
8	Mental Impairment
9	Physical Type
10	Hair Color

As mentioned before, we can extract the actual tree produced by the Random Forests and XGBoost algorithms. In this paper, we included some trees from XGBoost, they are small enough to read and understand. The identifiers in the figures included are generated by XGBoost, in Table VII we can relate the feature identifier in the graph to its description.

In Fig. 1, for example, we can see the first tree, where the first split is at feature mental impairment, if it is greater than 0.5, which means to have mental impairment or this value is undefined according to Table VIII. Then it goes through two splits over the feature missing days, if it is less than 4.550 and

TABLE VII
FEATURE IDENTIFIERS FOR XGBOOST

XGB Id	Feature Name
f0	Days Missing
f1	Height
f2	Age
f3	Weight
f4	Physical Type
f5	Skin Color
f6	Sex
f7	Hair Color
f8	Mental Impairment
f9	Eye Color

TABLE VIII
FEATURE MENTAL IMPAIRMENT VALUES IN MODEL

Mental Impairment	Value in Model
No	0
Yes or Undefined	1

TABLE IX
FEATURE SEX VALUES IN MODEL

Sex	Value in Model
Female	0
Male	1

if it is more than 1.450, then it splits on height; if it is greater than 145 cm this leaf ratio is the greatest in this tree with a value of 0.000714286.

What the algorithm implied is that people missing between 1.450 and 4.550 days and are more than 145 cm tall have better chances of being found than those who are less than 145 cm tall. When we looked into the data within these limits we saw that there are 1.465% more people found who are more than 145 cm tall than below this height, this information is in Table X. Looking at the data again, but this time with no limits on the missing days feature, the figures change as we can see in Table XI, the numbers on status as Found change little, but the numbers on status as Missing nearly double.

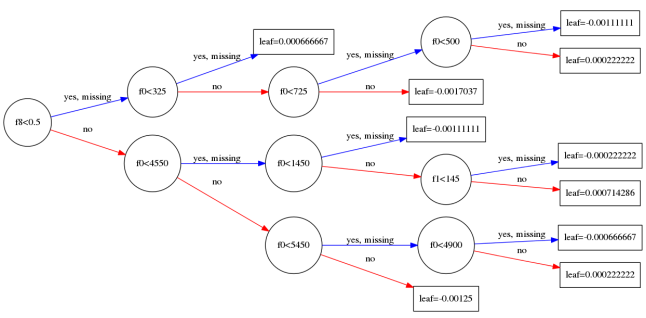

Fig. 1. XGBoost tree #1

TABLE X
DAYS MISSING BETWEEN 1.450 AND 4.550

Status	<145 cm	>145 cm	Variation
Missing	62	231	273%
Found	26	407	1.465%

TABLE XI
WITH NO LIMITATION ON DAYS MISSING

Status	<145 cm	>145 cm	Variation
Missing	107	462	332%
Found	26	465	1.688%

TABLE XII
AGE <31.5 AND WEIGHT <57.5

Status	Female	Male	Variation
Missing	210	231	10%
Found	281	174	-38%

We concluded that this range of missing days has an effect on whether people can be found or go missing, also depending on their height.

In Fig. 2, we have the same split if the person has mental impairment or it is undefined, the subsequent rules differ from the first tree. Here the next split is on ages below 31.5 years, if true the next split is on weight below 57.5, if true the last split is on sex, if the value of this feature is less than 0.5, then it is a female, as we can see from Table IX and receives the greatest leaf value in this tree.

Here we learned that women younger than 31.5 years and weighing less than 57.5 kilograms have better chances of being found than men. We can see this data in Table XII and these rules implied by the algorithm are supported by these numbers. As noticed how these splits work, this one suggests trying one-hot encoding with the feature sex.

In the next tree, in Fig. 3 , after the first usual split on mental impairment, we have a split on missing days below 4.900, the next is on age above 19.5, and finally a split on eye color value below 2.5, which means colors: undefined, blue, and brown, according to Table XIV, which leads to the greatest leaf value in this tree.

In Table XIII we see that eye colors blue and green have few values, so undefined color have little variation between missing and found. Brown and black have similar quantities

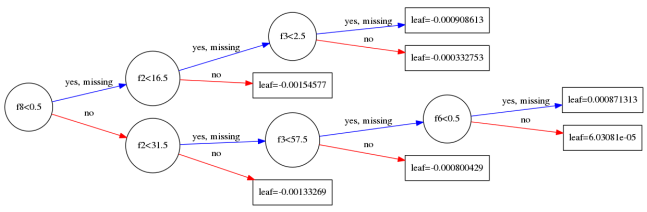

Fig. 2. XGBoost tree #2

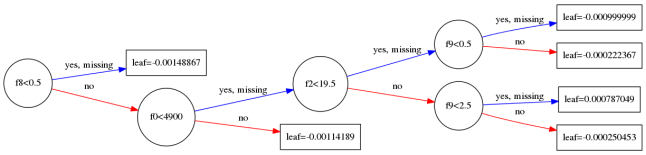

Fig. 3. XGBoost tree #3

TABLE XIII
DAYS MISSING <4.900 AND AGE >19.5

Eye Color	Missing	Found	Variation
Undefined	134	157	17%
Blue	1	5	400%
Brown	47	80	70%
Black	58	131	126%
Green	15	9	-40%

TABLE XIV
FEATURE EYE COLOR VALUES IN MODEL

Eye Color	Value in Model
Undefined	0
Blue	1
Brown	2
Black	3
Green	4

for missing, but black has more values for found, even though it was not considered in the tree.

Here we saw clearly that the limits of analyzing trees individually give a hint of how the algorithm worked to get to its results, but as it works with ranges on the splits, they may be reasonable and effective on continuous values features, such as number of days or years. However, they are much less effective on categorical data. We saw this is a hint to try one-hot encoding on this feature as well.

V. CONCLUSION AND FUTURE WORK

The accuracy results of Decision Trees and Logistic Regression was much lower than other models, as expected. Considering only the higher accuracy models, Random Forests and XGBoost, the most important features are missing days and age. Then we have the other features but the order is different depending on the algorithm. Since we have a relatively small dataset, we probably have different types of missing people skewing the results.

People may go missing due to sex trafficking, drug trafficking or they simply wanted to leave, and those reasons vary according to different ranges of age and sex, have different feature importances, and have different missing/found ratios. Besides missing days, the other features relate closely to the missing person intrinsic characteristics such as age and ethnicity.

When we looked into the trees, we noticed that segmenting the data to analyze specific groups may increase accuracy,

besides, some decisions made by the algorithm indicated that the use of one-hot encoding on these features may also increase accuracy and return more valuable insight on the data and the people groups represented.

We believe the objectives were met, as we built models which are able to predict if a person would go missing or be found from a set of features and learned much more from the results. We also found what the most important features are to reach this goal and how these features impact our results.

We also believe that future experiments could repeat this investigation against similar missing people databases from different countries, to find out differences and similarities. In addition, new experiments could be conducted with specific ranges of age and sex groups, in an effort to study the characteristics and prediction accuracy when we target known social problems such as sex and drug trafficking.

ACKNOWLEDGMENT

This research was supported by Instituto INFNET. We would like to show our gratitude to Raul Ferreira for sharing his pearls of wisdom and his insights with us during the course of this research.

We would also thank our course coordinator Eduardo Morelli who provided support and guidance that greatly assisted the research.

REFERENCES

[1] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees, wadsworth international group, belmont, california, usa, 1984; bp roe et al., boosted decision trees as an alternative to artificial neural networks for particle identificatio n," *Nucl. Instrum. Meth. A*, vol. 543, p. 57, 2005.

[2] C. Lazăr and M. Lazăr, "Using the method of decision trees in the forecasting activity." *Petroleum-Gas University of Ploiesti Bulletin, Technical Series*, vol. 67, no. 1, 2015.

[3] F. C. Arnett, S. M. Edworthy, D. A. Bloch, D. J. McShane, J. F. Fries, N. S. Cooper, L. A. Healey, S. R. Kaplan, M. H. Liang, H. S. Luthra *et al.*, "The american rheumatism association 1987 revised criteria for the classification of rheumatoid arthritis," *Arthritis & Rheumatism*, vol. 31, no. 3, pp. 315–324, 1988.

[4] M. M. Ward, D. C. Elli, L. M. Jack *et al.*, "Differential rates of relapse in subgroups of male and female smokers," *Journal of Clinical Epidemiology*, vol. 46, no. 9, pp. 1041–1053, 1993.

[5] R. Díaz-Uriarte and S. A. De Andres, "Gene selection and classification of microarray data using random forest," *BMC bioinformatics*, vol. 7, no. 1, p. 1, 2006.

[6] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[7] J. Pearce and S. Ferrier, "Evaluating the predictive performance of habitat models developed using logistic regression," *Ecological modelling*, vol. 133, no. 3, pp. 225–245, 2000.

[8] I. Das, S. Sahoo, C. van Westen, A. Stein, and R. Hack, "Landslide susceptibility assessment using logistic regression and its comparison with a rock mass classification system, along a road section in the northern himalayas (india)," *Geomorphology*, vol. 114, no. 4, pp. 627–637, 2010.

[9] X. Zhou, K.-Y. Liu, and S. T. Wong, "Cancer classification and prediction using logistic regression with bayesian gene selection," *Journal of Biomedical Informatics*, vol. 37, no. 4, pp. 249–259, 2004.

[10] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[11] J. Leathwick, J. Elith, M. Francis, T. Hastie, and P. Taylor, "Variation in demersal fish species richness in the oceans surrounding new zealand: an analysis using boosted regression trees," *Marine Ecology Progress Series*, vol. 321, pp. 267–281, 2006.

[12] R. Lawrence, A. Bunn, S. Powell, and M. Zambon, "Classification of remotely sensed imagery using stochastic gradient boosting as a refinement of classification tree analysis," *Remote sensing of environment*, vol. 90, no. 3, pp. 331–336, 2004.

[13] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[14] 2016. [Online]. Available: https://www.kaggle.com/

[15] M. A. Hernández and S. J. Stolfo, "Real-world data is dirty: Data cleansing and the merge/purge problem," *Data mining and knowledge discovery*, vol. 2, no. 1, pp. 9–37, 1998.

[16] P. Royston *et al.*, "Multiple imputation of missing values," *Stata journal*, vol. 4, no. 3, pp. 227–41, 2004.

[17] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.

[18] 2016. [Online]. Available: https://xgboost.readthedocs.io/en/latest/

[19] T. Chen and T. He, "xgboost: extreme gradient boosting," *R package version 0.4-2*, 2015.

[20] D. R. Cox and E. J. Snell, *Analysis of binary data*. CRC Press, 1989, vol. 32.

[21] S. M. Garn, W. R. Leonard, and V. M. Hawthorne, "Three limitations of the body mass index." *The American journal of clinical nutrition*, vol. 44, no. 6, pp. 996–997, 1986.

[22] 2016. [Online]. Available: http://www.ibge.gov.br/

[23] D. Harris and S. Harris, *Digital design and computer architecture*. Elsevier, 2012.