Title
Applying Artificial Intelligence to Predict Instability in Foster Care Outcomes

Authors
Daniel Oldham
Mihhail Berezovski

Declarations of Interest
None

Keywords
child welfare, foster care, data mining, data science, machine learning, artificial intelligence

Abstract
The Department for Children and Families (DCF) and associated child welfare service organizations record a number of details about the children and adults who participate in their services. This data presents an opportunity for analysis which could improve the operations of these organizations and the outcomes of the children and families who participate in them. We were provided eight years of data totaling over 40,000 childrens' foster care cases from an organization working with the DCF in Florida. Using artificial intelligence techniques and programming languages like Python and R we were able to predict a child's chance of having multiple removals from their current placement within the foster care system as part of their case, signifying less stability and a generally poorer overall outcome for the child's well-being. This paper aims to document the features that had the most impact on prediction and how these were engineered from the data we were given, and then present a risk calculator program that could be used by a practitioner to gain insight into the risk level of a child entering the foster care system with regards to susceptibility to multiple removals and subsequent instability.

Introduction

As part of a data science research initiative, Embry-Riddle Aeronautical University (ERAU) partnered with a child welfare services organization based out of northern Florida. This organization shared with ERAU eight total years of anonymized data involving foster care children, their associated DCF cases, and the adults who participated in their cases, in the form of Excel spreadsheet files. This data is not released with this research as part of a privacy agreement with the organization. Overall there were approximately 40,000 foster care childrens' cases in this data within the years 2010 to 2017 and 130,000 other unique participants associated with these cases. These participants include case workers, parents, relatives who participated in the case, and other children in the same home if applicable. The children in this data were 50.2% male and 49.8% female. Most primary caregivers were female (78%). An average year contained about 29,000 new lines of records, with a significant spike in the years 2015 and 2016.

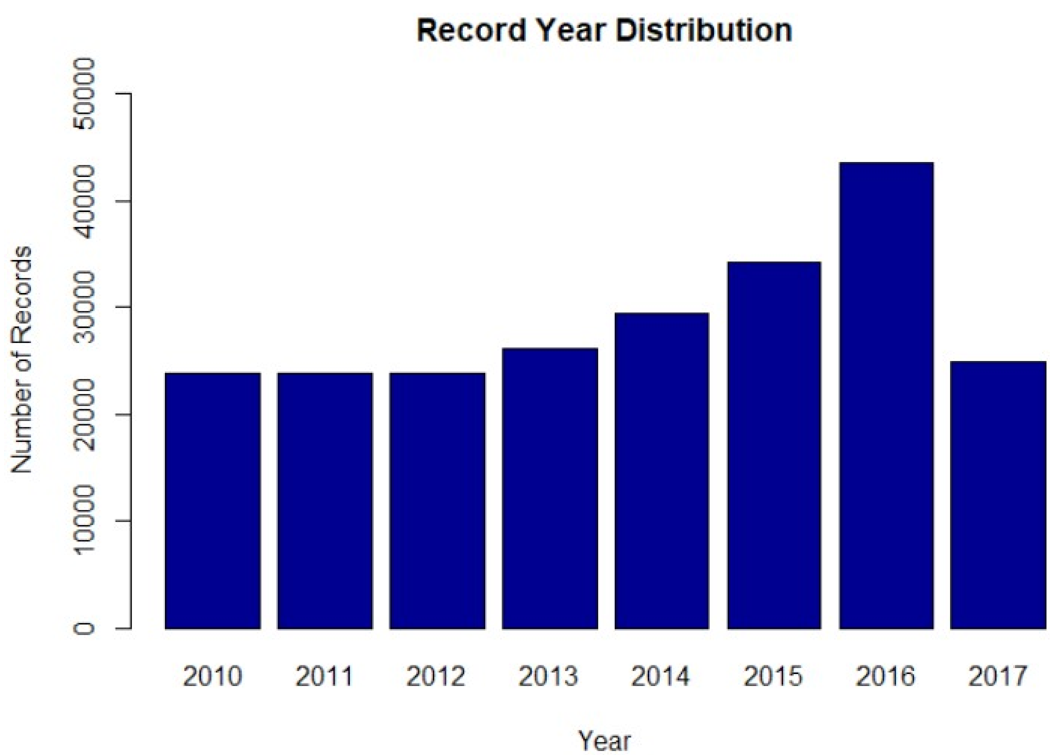

Figure 1: Distribution of record count per year

Out of 26 total ethnicity classifications, the top ten represented in the data are below. It is clear that ethnicities are mostly absent or not recorded, with well over half the records as "Other", "Unable to Determine", or "Unknown". That being said, we did use ethnicities in the model by marking some of the most frequently recorded ones outside of those mentioned (African American / Black, Hispanic / Latino and Eastern European) as unique numeric values for the model to incorporate.

Ethnicity	Percent of Participant Records
Other	56.2 %

African American / Black	20.5 %
Unable to Determine	10.5 %
Unknown	7.0 %
Hispanic / Latino	3.8 %
Eastern European	0.7 %
Mexican / Chicano / Mexican American	0.4 %
Puerto Rican	0.1 %
Haitian	0.1 %
Italian	< 0.1 %
(all other classifications)	~ 0.6 %

Figure 2: Table of top ten ethnicity classifications and percent of the raw participant records files they fill

In our prior research, we analyzed the movements of children through their case in the foster care system, for example, where a child may start (ex. foster care with a relative), and how their associated cases beyond this initial case differed (ex. removal from relative and placed into non-relative home). We "weighted" the stability and overall outcomes of these childrens' cases in order to predict which characteristics lead to more favorable outcomes, including exiting the foster care system and having no further cases. This prior research mostly focused on movement, how different placements lead to different outcomes and exiting, and which characteristics of the child's case influenced their placement settings. It was obvious that the very first placement influenced the outcome as a whole and this insight along with the overall results of the prior research were used in the design of this analysis [1].

Now, we aim to focus on prediction of a child's risk of multiple removals in their case. As a child is removed from their current placement, they not only lose stability in their lives but this reflects back that the decision to place them into that situation they are now being removed from was not the right one. If practitioners are able to assess the risk of this happening before a placement and removal event (and possibly many recurring events like this), they could prioritize the at-risk children who need a different placement. There are obviously real-world roadblocks to this logic, as not every child has the option to be put into a more optimal placement like foster care with a relative. Many children in this dataset continuously moved into and out of similar placements and to and from the same parents they were removed from originally. These cases are obviously difficult for the practitioner and participants, and it would benefit both parties if these could be predicted from the beginning. The goal is to predict which child is going to end up with multiple removals over a period of time and therefore is more susceptible to a continuous loop of removals and placements with limited stability and constant upheaval. Additionally, we aimed to integrate the model into a simple program that would allow the user to enter in details about a child and instantly output a result as "at risk" or "not at risk" for this sort of situation during their time in foster care.

In data science terms, this is nothing more than supervised learning to solve a classification problem. Since we can extrapolate from the data which children actually ended up in unstable situations, we can mark these children as the problematic cases. Thus, we have a sample set of results that we want a machine learning model to look at and predict as "at risk". All the other children who did not have multiple removals can be marked as "not at risk" overall. We are simply trying to classify a child as “not at risk” or “at risk” for multiple removals as accurately as possible, and using details that would be readily available to an experienced child welfare services practitioner at the DCF or an associated organization.

Many of the details that are used to build the machine learning model with are derived from the data. This means that we were not explicitly given a characteristic, for example the length of a case, but we can extrapolate this value by having the year of the start and ending dates of the case. We then add this new value to a new column of the dataset for each case. Feature engineering like this of new variables significantly boosts the usability of the data that we have because we can use these derived features to predict off of by telling the program to use them as part of the predictive model building process. With data like the records we have from the DCF which can be convoluted yet hide many important details, feature engineering is absolutely imperative to add to the dimension and depth of the data.

Background

Machine learning is a subset of artificial intelligence which aims to “train” a computer using an algorithm to make accurate predictions based on the patterns of features in the data. In this research we harness “supervised” learning, a type of machine learning where the program has access to a subset of the data including the variable that it is trying to predict so that it can train a model using a reinforcement learning process to predict those same results. This model is then released onto a different subset of the data *without* the prediction variable available and the subsequent accuracy is analyzed [2]. Splitting the data into two parts to allow the machine to learn patterns before moving on to actual prediction is widely accepted practice in the data science community to achieve high quality model results and accurate predictions.

The Random Forest model is a popular ensemble machine learning model which utilizes a multitude of smaller decision trees to make either classification (a class like “Yes” or “No”) or regression (numeric value) predictions. Decision trees are designed to predict a target variable’s results as each feature in the data varies by finding “rules” that split apart the data appropriately. For example, a decision tree could formulate a rule when trying to estimate if it will rain by the color of the clouds in the sky. The tree could split at if the clouds are marked as “dark”, if yes, rain is predicted; if no, no rain is predicted. Then each time the model attempts to predict rain, the tree is traversed based on if the clouds are dark or not to find the resulting rain prediction. This sort of simple splitting logic can be scaled to any extent but the model is at risk of becoming too rigid and “over-fitting” to provide highly accuracy predictions while training the model, but this accuracy is then not present when testing the model on the second portion of the data.

It should seem apparent how this rule-building logic is applicable to our foster care data predictions. We desire this sort of rule building to take place where a child is ultimately marked as “at-risk” or “not at risk” based on how childrens’ outcomes in the dataset vary based on their characteristics as a whole like their age, time in the foster care system, number of removals, and any feature that can be used or derived from the data that will contribute to the accuracy of the model.

Decision Tree models by themselves are easy to implement and understand but often over-fit the data leading to low accuracy. Random Forest models work by fitting many decision trees to subsets of the same data, and then aggregating the results of these individual trees to provide a single prediction. This reduces the over-fitting tendency of individual decision trees and boosts accuracy of the model as a whole. The Random Forest model is well known for its robustness for new modeling with a variety of variables, promptness in getting results, and ease of interpreting the results and therefore presents a great opportunity for our foster care application [3]. We also had success using a Random Forest model in our prior research, where this model type was the most accurate out of a number of machine learning models [1]. Since this is the same data we are working with, another Random Forest model makes sense to experiment with before branching out to other model types if desired.

In our use case we are classifying children as “at-risk” or “not at-risk”. To interpret the results of the model, we must analyze feature importance, which aims to rank and assess the influence of individual features of the data on the whole model output. We chose permutation importance and SHAP values as two methods to interpret the model output. Permutation importance works by assigning a value to each feature that is calculated by shuffling that feature’s column and comparing how the model output is effected, thus showing how much that feature could influence the whole prediction [4]. SHAP Values (Shapely Additive Explanations) allow us to decompose in which way each feature effected a prediction, not just at what importance each feature was [4]. SHAP values are very helpful in visualizing the specific influence of each feature, instead of just the importance of each one like permutation importance shows us. SHAP Values were used specifically for the risk calculator as a visual component to analyze to understand a prediction better. Both of these methods are widely used in data science to interpret and analyze model results like Random Forests and other model types.

Materials and Methods

The data wrangling, feature engineering, and machine learning data frame was done in R using R Studio using the Tidyverse package, and the machine learning model and risk classifier programs were written in Python with the help of the scikit-learn machine learning package and made interactive using Jupyter Notebooks.

In order to effectively build an accurate model, we needed to create a complete data frame with no empty values, consistent formatting, and useful derived values. Immediately after importing the data into R, the rows with empty values that we need for analysis were removed and the data was further filtered down for quality and completeness. Unfortunately, this means that many childrens’ cases were filtered out during the data cleaning and quality control process simply due to missing data. The final data frame ready for a machine learning model contained just shy of 3,000 childrens’ cases, down from the 40,000 initial children who were in the full dataset. Roughly 650 of these 3,000 children were marked as “at-risk” (~22%). Similarly, only about 20,000 participants aside from these children had records with the required level of completeness where they could be useful for our analysis, which is no where near the 130,000 participants that are recorded in the data.

This final cleaned model-ready data frame contained the following columns:

Feature	Description	Derived	Details
id	The unique ID of the child	No	
zip	The zip code of the child’s first record	No	If multiple zip codes are recorded for a single child, we take the first one as primary location
zip_count	The number of cases within that zip code	Yes	Useful for assessing a city vs. rural area

number_removals	The number of removals a child experienced	Yes	Based off of unique removal dates per child's ID; not used in model due to correlation with variable we are predicting
number_placements	The number of different placements a child was put into	Yes	Based off of unique placement begin dates per child's ID; not used in model due to correlation with variable we are predicting
number_participants	The number of unique participants included in a child's case	Yes	Also includes other children in the home in this count
case_duration_yrs	The total length of the child's case	Yes	Based off of case open and case closed dates; decimals are allowed as this is represented in years
number_caregivers	The unique number of caregivers and parents in a child's case	Yes	Counts Service Role of Primary/Secondary Caregiver, and Parents in and out of the home
age_child	The age of the child	Yes	Data only gave month of child's birth so date of birth was set to be first of that month during age calculation
avg_age_caregiver	The average age of a caregiver associated with a child's case	Yes	Since in many cases there are multiple caregivers we take the average of all of them
med_gross_income_zip	The median gross income of the zip code of the child's case (*zip*)	N/A	This is from another dataset (IRS 2018 tax return data), and linked via zip code to our dataset
first_placement	The first placement of the child when they entered into the foster care system	Yes	This is a numeric value ranking based off of prior research on best placements. 4-Pre-Adoptive Home 3-Foster with Relative 2-Foster Non-Relative 1-All Others
multiple_removals	0 if the child did not have multiple removals; 1 if the child had greater than 1 removal as part of their case	Yes	Binary ranking, either 0 or 1, this is the value the model is attempting to predict. This is derived off of *number_removals.*
gender	The gender of the child	No	1 for Female 0 for Male

ethnicity	The ethnicity of the child	No	Ranking based on most common, non-blank ethnicities in dataset: 3-African American/Black 2-Hispanic/Latino 1-Eastern European 0-All Others
perc_life	The total percent of the child's life that they spent within the foster care system	Yes	Useful for assessing younger children's cases which take up a significant portion of their entire lives
first_place_duration	The duration of the child's first placement after initially being placed in the system	Yes	This is calculated in years; decimals allowed

Figure 3: Table of features and their details in final machine learning ready data frame

A Random Forest model was built with Python scikit-learn RandomForestClassifier to predict the *multiple_removals* feature based off of 12 features. Since *number_removals* and *number_placements* both correlated with the *multiple_removals*, these features were not used as part of the predictive model. The zip codes vary widely and were not incorporated into the model. It is worth noting that many of these features do not need to be converted into numeric values (ex. gender) like we did here, the reason for this was so that we would be able to run other types of models beyond just Random Forest that require numeric inputs.

A “calculator” script was built in Python in which a user inputs these 12 features' values of a child's case and is given a decimal number between 0.00 and 1.00 regarding the child's risk of multiple removals (0 for low risk, 1 for high risk). This prediction value is made by the model built in scikit-learn, which is exported as its own file and then imported into the calculator program. The calculator therefore doesn't need the raw data files and instead imports the model built off of those files; this makes the calculator a “lightweight” and portable solution for prediction that would be easier to distribute and share than the full R data cleaning and associated programs.

Results

The Random Forest model resulted in an overall accuracy of about 82%, with the precision of predicting “not at-risk” children being 83%, and “at-risk” children being 70%. The importance level of features for model prediction are listed below, 1 being the most important variable for prediction, and 12 being the least important for prediction. This ranking is based off of permutation importance of each of these individual features as part of the Random Forest model and is also shown in the Python model output below with actual weighted values.

1. first_placement
2. case_duration_yrs
3. number_participants
4. zip_count

5. first_place_duration
6. age_child
7. number_caregivers
8. perc_life
9. med_gross_income_zip
10. avg_age_caregiver
11. ethnicity
12. gender

```
Random Forest Mean Accuracy:
 81.60%


Confusion Matrix:
 [[539  27]
 [109  64]]

Classification Report:
               precision    recall  f1-score   support

           0       0.83      0.95      0.89       566
           1       0.70      0.37      0.48       173

    accuracy                           0.82       739
   macro avg       0.77      0.66      0.69       739
weighted avg       0.80      0.82      0.79       739
```

Weight	Feature
0.0376 ± 0.0141	first_placement
0.0333 ± 0.0173	case_duration_yrs
0.0241 ± 0.0125	number_participants
0.0208 ± 0.0134	zip_count
0.0203 ± 0.0179	first_place_duration
0.0179 ± 0.0040	age_child
0.0135 ± 0.0110	number_caregivers
0.0119 ± 0.0052	perc_life
0.0103 ± 0.0072	med_gross_income_zip
0.0078 ± 0.0118	avg_age_caregiver
0.0011 ± 0.0040	ethnicity
-0.0027 ± 0.0024	gender

Figure 4: Random Forest model accuracy details and permutation importance output in Python

The calculator was successful not only at making predictions based off of inputted data, but also at visualizing the influences of the individual characteristics per each child. The model weighs certain features more than others and this weighing is represented by the blue and red bars pushing the overall prediction towards 0 or 1. These features will be weighed in different ways depending on the whole case details inputted into the model, so it would be difficult to make simple statements about each feature, but the visualization helps to interpret these influences. In data science terms, these influences are quantified by SHAP Values.

Here is an example of a prediction that is leaning strongly towards “not at-risk” (0.12 / 1.00):

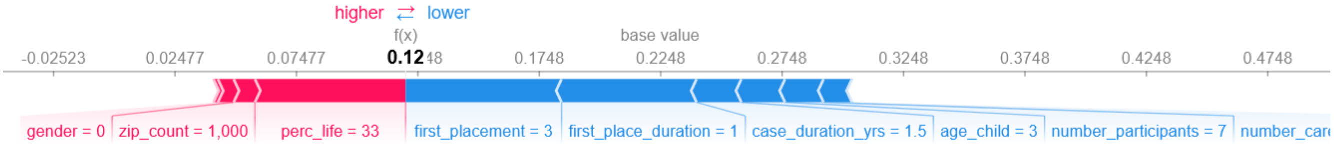

Figure 5: SHAP Values visualization output of calculator prediction program (not at-risk)

Here is an example of a prediction leaning strongly towards "at-risk" (0.52 / 1.00):

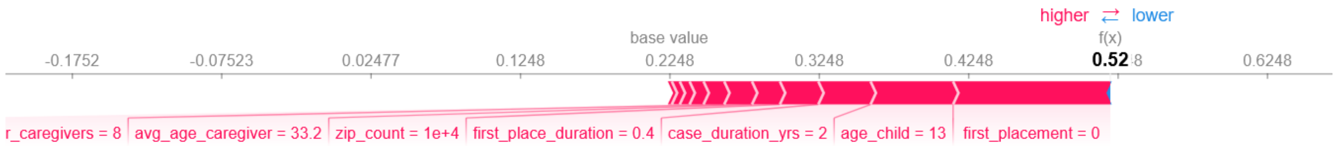

Figure 6: SHAP Values visualization output of calculator prediction program (at-risk)

Discussion

The model results can consistently predict children at risk of multiple removals at an acceptable accuracy. It is worth pointing out again the size of the data was very small from a machine learning perspective (only about 3,000 rows), and this is why feature engineering is so important, as well as good data cleaning. For an organization with more data (and/or cleaner data), modeling could result in even better results.

The top five most important features for prediction were the first placement of the child, their case duration, the number of participants involved, the number of cases within their zip code, and the duration of their first placement. The first placement being the most important feature is of no surprise because that is what our prior research pointed directly to in terms of case success. Children who start in foster care with a relative (or who are immediately moved to a pre-adoption service) are much less at risk for staying in the system and being removed many times. Case duration and number of participants are also logically important influences on a child's stability and support. The SHAP values show that many of these characteristics can influence the outcome in both directions. It is interesting to see that the number of cases within the child's zip code influences their overall stability. It is possible that this is the influence of more crowded areas like cities making it more difficult to place children appropriately for some reason, or this could work the opposite way with the city having better resources and more practitioners available to work on the case. This is something that would need to be explored further to see the actual influence.

We can see from the Python model output details (*Figure 3*) that the model had an easier time predicting "0" (not at-risk), which makes sense due to the amount of those cases that are available in the data. Although it was less accurate at predicting the "at-risk" classification, as a whole these results are very acceptable for the scope of this prediction. We see in the confusion matrix that there are only 27 false positives. This means that there were 27 children that the model classified as "at-risk", but in reality they were not at risk. In the opposite corner, 109 children were *not* marked as "at-risk" when they should have been (false negatives). It is debatable if this ratio of of false positives to false negatives is good or bad: do we want to "cry wolf" on more children and potentially classify too many children as at-risk, or potentially miss more at-risk children? Again, considering the proportion of at-risk cases (only 22% of the data), it is understandable that the model is most accurate on the "not at-risk” children. The size of the training set of data could be increased to potentially boost accuracy of the “at-risk” children. Regardless, the majority of the predictions were true positive or true negative, which speaks to the accuracy of the Random Forest model.

The calculator application was successful in taking a number of characteristics regarding a child and giving a prediction about multiple removal risk. It is definitely possible to use this sort of calculator “in situ”; with an experienced practitioner in the loop this could aid decision-making regarding at-risk children. Since the calculator outputs decimals between 0 and 1, it is not just a binary decision-making app, it allows the user to see what extent the child is really at risk or not. This would allow an experienced user to supplement their own judgment about the case. As a final note, this calculator could operate as a portable and shareable solution for making predictions based off of the model because it does not require the raw data, data cleaning, and other complex programming to use. It simply opens up in Python or Jupyter Notebook and the user edits the parameters and runs the file.

Conclusions

We are overall content with the results of modeling the data with an overall accuracy of about 80% with our Random Forest model. However, more accurate results are definitely possible with data of higher quality and completeness. The top five most important features for prediction were the first placement of the child, their case duration, the number of participants involved, the number of cases within their zip code, and the duration of their first placement. The calculator program was a successful endeavor to make it simple to understand a prediction based off of a child’s case details. A route for further research would be to conglomerate more case, participant, and removal data from the DCF and other organizations to increase the size of the model-ready data frame.

References

[1] Oldham, Daniel; Foster, Nathan; and Berezovski, Mihhail (2019) "Data Mining and Machine Learning to Improve Northern Florida’s Foster Care System," Beyond: Undergraduate Research Journal: Vol. 3, Article 3. Available at: https://commons.erau.edu/beyond/vol3/iss1/3

[2] IBM Cloud Education (2020) “Supervised Learning,” IBM. Available at: https://www.ibm.com/cloud/learn/supervised-learning

[3] Storey, Dylan (2018) “Random Forests, Decision Trees, and Ensemble Methods Explained,” Oracle AI & Data Science Blog. Available at: https://blogs.oracle.com/datascience/random-forests%2c-decision-trees%2c-and-ensemble-methods-explained

[4] Becker, Dan (2021) “Machine Learning Explainability,” Kaggle. Available at: https://www.kaggle.com/learn/machine-learning-explainability

Appendix

For all source code in Python and R, visit: https://github.com/mathemacode/PSF_ERAU