Assignment 2: Fragile Families Challenge

Neamah Hussein
Department of Computer Science
Princeton University
nhussein@princeton.edu

Devansh Gupta
Department of Electrical Engineering
Princeton University
devanshg@princeton.edu

Abstract

The Fragile Families Challenge is a collaborative effort to use a dataset that has been collected over fifteen years to better understand the factors that affect disadvantaged children and families in the United States. In this paper, we use classification and regression methods to predict six outcomes for the families in this study: child's GPA, child's grit, child's material hardship, parent' job loss, parents' job training and eviction.

1 Introduction

Nearly a third of all children born in the United States today are born to unmarried parents, with this proportion even higher in some minority and poor populations [7]. The "Fragile Families and Child Wellbeing Study" refers to these new parents and their children as "fragile families" because of the multiple risk factors associated with non-marital childbearing and to signify the vulnerability of the relationships within these families [7]. The aim of this paper is to utilize data collected about these "fragile families" over multiple years by using machine learning algorithms to model and predict certain key outcomes about them, and draw conclusions from the results.

The three continuous outcomes that we regress for are: the GPA of the child at age 15, a quantified measure of the material hardship a child has experienced at age 15, and a quantified measure of a child's grit at age 15. The binary outcomes that we classify are: whether or not a parent of the child has received any additional job training in the past three years, whether or not a parent has been laid off in the past three years, and whether or not the family has been evicted from their home in the past three years.

2 Related Work

2.1 Fragile Families

The Fragile Families and Child Wellbeing study [8] has been ongoing since 1998. The study follows approximately 5000 children from families from 20 different cities in the United States. The data has been collected in 6 waves so far: at birth, age 1, agree 3, age 5, age 9, age 15. The data is collected by means of interviews with mothers, fathers, primary caregivers, teachers and the children themselves. Several researchers (e.g.: [10], [12], [9]) have been able to use the data to explore sociological questions with empirical analyses to study the effects of measured factors on disadvantaged families and their children. The current state of the data uses data collected over the past 15 years to predict the six aforementioned outcomes at age 15.

2.2 Collaborative Projects

In today's digitized and connected world, the concept of crowd-sourcing has become increasingly popular not only in commercial applications, but also in research projects [3]. These research

projects tap into the collective knowledge of the masses and crowd-source man-power as well as expertise for the purposes of tackling a problem. Some examples of such projects are the human genome project [11]- which is the world's largest collaborative biology research project, the Atlas Collaboration's observation of the Higgs Boson [1] and the Netflix Prize, which was a competition on collaborative filtering to predict which movies customers would like based on their previous movie ratings [2]. Fragile Families is another such challenge that aims to bring together the best prediction algorithms for the six outcomes to build one large prediction model.

3 Methodology

We compare the accuracy of five classifiers to predict our binary outcomes of job loss, job training and eviction: *Random Forest(Gini Scoring), Gaussian Process Classification (RBF kernel), K-Nearest Neighbors Classification(23 neighbors), Boosting (Decision Tree base estimator) and Quadratic Discriminant Analysis*.

We compare the accuracy of five regressors to predict the continuous outcomes of gpa, grit and material hardship: *Epsilon-Support Vector Regression (RBF Kernel), K-Nearest Neighbors Regression (23 neighbors), Lasso (alpha = 0.96), Gaussian Process Regression (RBF Kernel) and Kernel Ridge Regression(alpha = 0.9).*

We implemented these using SciKit Learn [5] and all parameters are the default unless specified.

Our training data consisted of 4242 samples (i.e.: 4242 families). Each sample was a vector of over 13,000 features.Each of these features correspond to an answer from the interviews that were conducted over the 15 years of data collection, in 6 waves.

Exactly half of our training data was labeled. 2121 of our samples were labeled with the six outcomes, three of which are continuous and three are binary.

3.1 Missing Data

Since the data is collected using interviews over an extended period of 15 years, there are several instances of "item non-response" and "survey non-response". These, as explained by the Fragile Families team are instances when respondents either refuse to answer questions on the surveys, or do not participate in a particular survey. For these reasons, there are several values in the data that are codified for the various reasons that an actual answer has not been recorded. These codes are negative integers ranging from -1 to -9.

We did not focus the bulk of our work on sophisticated imputation methods. Instead, for all the missing values, we replaced the codes with the modal response item to that particular question. Future work would benefit from implementing more sophisticated means of imputation, such as a matching method.

The label data also contained several missing values. These, however, were simply coded with "NA". We removed all samples from our training data that had missing labels. This left us with around 1100 samples to train on.

3.2 Spotlight: Gaussian Process

Gaussian Process is an alternative approach to regression problems. We use Gaussian Process for Regression as well as for Classification. We will first discuss how a Gaussian Process works and then extend that to the notion of classification.

The usual structure of a Bayesian regression algorithm involves knowing a prior distribution that we assume fits the data, and then updating this the parameters of this distribution with each additional sample that the model encounters to produce a posterior distribution. This, however, requires the fundamental assumption that our data fits the shape of a specific type function. A Gaussian Process (GP) is a non-parametric approach, which given training inputs X and labels y, infers a distribution over all possible functions, $p(f|X, y)$ and then uses this to make predictions y_* given new inputs x_*, i.e., to compute [4]:

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int p(y_*|f, \mathbf{x}_*)p(f|\mathbf{X}, \mathbf{y})df$$

Thus, GP defines a prior over functions which is converted to posterior distribution over functions after the test data is seen. To limit the set of functions considered, constraints are imposed such as only considering the behavior of functions within the domain that contains our data and, only considering functions whose output mean is a specific value (in our case, 0) and functions that are smooth, i.e.: functions that ensure that inputs that are close together lead to output values that are close together. **The smoothness of the function is ensured using a covariance matrix, which along with the mean function that generates the expected value of $f(x)$ defines a Gaussian Process.**

As explained in [4], in order to represent a distribution over a function rather than over a parameter, we only need to know the distribution over a function's values "at a finite, but arbitrary set of points" [4], $x_1, ..., x_N$.

A Gaussian Process assumes that the distribution of $p(f(x_1), ..., f(x_N))$ is jointly Gaussian, i.e.: a Multivariate Gaussian Distribution. This joint Gaussian has parameters mean, $\mu(x)$, and covariance, $\sum(x)$, which is the matrix result of a transformation over the input x by a positive definite kernel function. The formula below shows the joint probability distribution of the continuous outcomes, where f are the outcomes to our train set and f_* are the outcomes to our test set.

$$\begin{pmatrix} f \\ f_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu \\ \mu_* \end{pmatrix}, \begin{pmatrix} K & K_* \\ K_*^T & K_{**} \end{pmatrix}\right)$$

Our goal is to find this conditional probability: $p(f_*|x_*, x, f)$. We ultimately end up with the mean μ_* and covariance matrix $\sum_*$ that define this distribution: $f_* \sim \mathcal{N}(\mu_*, \sum_*)$. Before sampling from this distribution, we need to express the standard normal equivalent of our multivariate Gaussian distribution in this form: $f_* \sim \mu + \mathcal{B}\mathcal{N}(0, I)$. $\mathcal{B}$ is the square root of the covariance matrix, which we get by using the Cholesky decomposition.

4 Results

Since we only had access to 1100 training labels and not the remaining test labels, we estimated the performance of our classifiers and regressions using 1) mean-square-error (MSE) values from 10-fold cross validation 2) 95% confidence interval of MSE values found from 1000 bootstrap (Cross Validation) samples of data for each classifer/regression, and 3) time taken to fit classifier/regression on training data.

Classifier	Eviction			Layoff			Job Training		
	CV MSE	MSE 95% CI	Time (s)	CV MSE	MSE 95% CI	Time (s)	CV MSE	MSE 95% CI	Time (s)
RF	**0.0601**	0.0418-0.0836	0.0966	0.2101	0.1791-0.2448	0.0841	0.2801	0.2478-0.3254	0.0856
KNN	0.0602	**0.0418-0.0835**	**0.0928**	0.2111	0.1791-0.2478	**0.0047**	0.2693	0.2358-0.3134	**0.0028**
GP	0.0602	0.0418-0.0806	1.5061	**0.1983**	**0.1642-0.2299**	1.9539	**0.2633**	**0.2239-0.3045**	1.4237
Boosting	0.0790	0.0567-0.1015	0.3828	0.2210	0.1940-0.2687	0.3749	0.3067	0.2687-0.3522	0.3831
QDA	0.3286	0.0567-0.3582	0.1221	0.4773	0.4000-0.5104	0.0261	0.3028	0.2687-0.3582	0.0216

Table 1: **Results from 5 classifiers on the 3 binary outcomes (Eviction, Layoff, Job Training) of approx. 1100 children, who comprise the training set, while selecting 50 features for each outcome.** For each classifier, we report the MSE found by 10-fold cross validation, the 95% Confidence Interval of MSE (Cross-Val Bootstrap), and wall clock time in seconds taken to fit the classifier.

Table 1 shows the performance of 5 classifiers on the binary outcomes. Iterating over multiple amounts of features selected, we found that 50 features gave us the optimal results of decreasing fitting time and improving accuracy by avoiding overfitting, for most continuous outcomes. The classifications performed equally well on a small number of features as well as a large number of features. Looking at Table 1, we found that RF classifier gave the lowest MSE during 10-fold CV

for Eviction, with its CI of MSE also being narrowly around that value, showing the dependable performance of the classifier. For Layoff and Job Training, we found that GP managed lowest MSE, though fitting time was considerably more than other other classifiers, implying that this method might not scale well to larger datasets with more samples or higher features selected. QDA had the worst performance, perhaps as it has no hyperparameters that we could tune to fit the data better.

Regressor	GPA			Grit			Material Hardship		
	CV MSE	MSE 95% CI	Time (s)	CV MSE	MSE 95% CI	Time (s)	CV MSE	MSE 95% CI	Time (s)
KNN	0.2266	0.2028-0.2631	**0.0037**	0.4112	0.3664-0.4836	0.2524	0.0231	0.0188-0.0286	**0.0023**
SVR	**0.2287**	**0.1977-0.2588**	0.1109	0.4083	0.3662-0.4634	**0.1358**	0.0238	0.1967-0.0283	0.0311
KR	6.3887	5.9230-7.1451	0.0672	7.2229	6.8584-7.5682	0.1386	0.0332	0.0267-0.0400	0.0687
Lasso	0.2317	0.2012-0.2658	0.0332	**0.4057**	**0.3654-0.4500**	0.1486	**0.0226**	**0.0185-0.0273**	0.0069
GP	22.542	10.151-61.105	0.2445	3.2663	2.8508-3.7019	0.2572	0.0240	0.0197-0.0288	0.2180

Table 2: **Results from 5 regressions on the 3 continuous outcomes (GPA, Grit, Mat. Hard.) of approx. 1100 children, who comprise the training set, while selecting 46 features for GPA, and 20 features each for Grit and Mat. Hard.** For each regression, we report the MSE found by 10-fold cross validation, the 95% Confidence Interval of MSE (Cross-Val Bootstrap), and wall clock time in seconds taken to fit the regressor.

Table 2 shows the performance of 5 regressions on the continuous outcomes. Iterating over multiple amounts of features selected, we found that 46 selected features was optimal for GPA, whereas 20 features was sufficient for better performance on Grit and Mat. Hardship. We found that SVR regression gave the lowest MSE during 10-fold CV for GPA, with its lower and higher bound of MSE CI also being lower than the other regressions. However, Lasso performed better with lower MSE for Grit and Material Hardship. Also, GP and KR returned substantially higher error rates than the other regression methods.

Thus, based on the above two tables and 3 performance indicators specified earlier, we picked the above mentioned classifiers and regressors (with tuned hyper parameters and optimal number of selected features) specifically for each of the 6 outcomes, and got the following error rates on the held-out test data on the FFC website:

	GPA	Grit	Mat. Hardship	Eviction	Layoff	Job Training
Loss Function Score	0.68714	0.53353	0.02792	0.05660	0.22453	0.27736

Table 3: MSE results on the leaderboard

5 Extensions

5.1 Using a More Sophisticated Model

We decided to use Gaussian Process Regression (featured in section 3.2) for the 3 continuous outcomes and Gaussian Process Classification to predict the 3 binary outcomes. Further, we used GLMs like Lasso, Ridge regression etc. as well.

5.2 Predicting Prediction Results

Cross Validation

Not having access to the test set meant that we would only be able to assess the true performance of our model once we uploaded our predictions to the website. In order to optimize on our submissions and predict how well our model might do, we did 10-fold cross validation on all our models used to predict the 6 outcomes. We report the Mean Squared Error values, averaged over the 10 cross-validations, in section 4.

We also used cross validation to optimize the hyper parameters for the models we trained.

Confidence Intervals - Cross Validated Bootstrap

While the average MSE from 10-fold cross validation provided some insight regarding the true performance of our model, we also wanted to know how variable this error was for a particular model. This helped in assessing the performance of a model before we chose to use it for our predictions.

Ideally, we want a model whose MSE does not swing erratically. We'd like to determine with 95% confidence that our average MSE over 10-fold cross validation will fall in a certain interval. In order to do this, we use the cross-validation bootstrap method to generate our 95% confidence intervals for MSE.

We sample-with-replacement around 730 samples from our training set of 1100 samples to create a new training set. The remaining samples make up our validation set. We train the model on our new training set and then evaluate the average MSE, μ_1 on the predictions for our *validation* set only. We repeat this k times, where $k = 1000$. Ultimately, we have k values of average MSE's from each iteration of the bootstrap: $\mu_1, ..., \mu_k$. We assume the average MSE from each iteration is a random variable μ that takes a normal distribution. We compute the confidence interval by finding the empirical quintiles. So, after sorting the k MSE's, we take the $\mu_2 5$ as the lower bound and $\mu_7 50$ as the upper bound of our confidence interval. The smaller the interval, the more sure we are of what performance to expect from our model.

6 Discussion and Conclusion

With a large data set with high dimensionality, the possibility of overfitting is very high. We chose to do feature selection in order to identify and only train on features that were deemed to to the outcome. We tried two different methods of feature selection: chi-square and ANOVA F-Value.

Chi-square feature selection measures the dependence between features and discards features that are likely to be independent of the outcome.

ANOVA F-Value feature selection method calculates the proportion of variance explained in the data by a particular feature. Features with high such proportions are selected.

We discovered that chi-square feature selection did not provide for much variety in the types of features selected. In particular, they all corresponded to similar questions across all waves of the data: "Approximately how much could you sell this home for today?" or "How much is owed on this house?". However, we discovered that this aligns with sociological theory that societal inequality stems primarily from discrepancies in *wealth* more than income [6].

The features selected using ANOVA F-Value feature selection were a lot more descriptive and relevant to the specific outcomes we were training for. For example, here are some examples of interesting features selected:

- Eviction:
  - m5f23c Did not pay full amount of rent/mortgage payments in past 12 months
  - m5f23k Telephone service disconnected because wasn't enough money in past 12 month
- Job Training:
  - cm1edu Mother baseline education (own report)
  - m5i2_2 I2_2. Attending GED or ABE program
  - m5i2_7 I2_7. Attending vocational/technical/trade school
- Material Hardship:
  - f2a3d How many nights has child spent with you since he/she was born?
  - m4c6a1 How many times have you refused to let father see child, last 2 yrs?
  - m4c37c3 How long did he (father) spend in jail/prison? (weeks)

We treated the number of features to train on like a hyper parameter and optimized for the best number of features to select for each outcome.

References

[1] Georges Aad, T Abajyan, B Abbott, J Abdallah, S Abdel Khalek, AA Abdelalim, O Abdinov, R Aben, B Abi, M Abolins, et al. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1):1–29, 2012.

[2] James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA, 2007.

[3] M Fathianathan, JH Panchal, and AYC Nee. A platform for facilitating mass collaborative product realization. *CIRP Annals-Manufacturing Technology*, 58(1):127–130, 2009.

[4] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[6] Fabian T Pfeffer and Robert F Schoeni. How wealth inequality shapes our future. *RSF*, 2016.

[7] Nancy Reichman, Julien Teitler, Irwin Garfinkel, and Sara McLanahan. Fragile families: Sample and design. In *Children and Youth Services Review, Vol. 23*, pages 303–326, 2001.

[8] Nancy E Reichman, Julien O Teitler, Irwin Garfinkel, and Sara S McLanahan. Fragile families: Sample and design. *Children and Youth Services Review*, 23(4-5):303–326, 2001.

[9] Lauren M Rich. Regular and irregular earnings of unwed fathers: Implications for child support practices. *Children and Youth Services Review*, 23(4-5):353–376, 2001.

[10] Lauren M Rich, Irwin Garfinkel, and Qin Gao. Child support enforcement policy and unmarried fathers' employment in the underground and regular economies. *Journal of Policy Analysis and Management*, 26(4):791–810, 2007.

[11] Mark P Sawicki, Ghassan Samara, Michael Hurwitz, and Edward Passaro. Human genome project. *The American journal of surgery*, 165(2):258–264, 1993.

[12] Robert C Whitaker, Shannon M Phillips, and Sean M Orzol. Food insecurity and the risks of depression and anxiety in mothers and behavior problems in their preschool-aged children. *Pediatrics*, 118(3):e859–e868, 2006.