Fragile Families Data Set Challenge

Author
Jelani Denis
jdenis@princeton.edu

Abstract

In this paper, we apply a number of machine learning frameworks to the prediction tasks posed as the Fragile Families Challenge, which involves using a large survey-based dataset of over 4,000 samples and more than 13,000 features. Our best performance for continuous-valued labels was achieved by our Linear Regression model with a MSE of 0.377, 0.216, and 0.026 for gpa, grit, and material hardship respectively on the held-out data. Our best performance for binary-valued labels was achieved by our SVC-type SVM with a Brier Loss Score of 0.133, 0.233, and 0..233 for eviction, layoff, and jobTraining respectively on the held-out data.

1 Introduction

The Fragile Families Challenge is an open-source competition proposed by the Fragile Families & Child Wellbeing Study hosted by the Bendheim-Thoman Center for Research on Child Wellbeing at Princeton University. The study examined and closely followed the development of disadvantaged "fragile families," many of whom are low income and unmarried parents, over the course of 9 years from the date of the birth of their child. Data were collected at regular intervals over a wide range of categories, including parent survey data, primary caregiver surveys, in-home assessments, child development assessments, and others. Six key outcomes were collected in a follow-up wave at year 15, including 3 continuous valued and 3 binary valued outcomes. The challenge is thus posed as this: Develop and train a model which most accurately predicts the year 15 labels from the rest of the data.

The challenge is quite daunting for many reasons. Firstly, the data is extremely sparse, and there is wide variety of data types to consider. Preprocessing and intelligent imputation of the data, in its own right, is a challenging problem alone. Even if one were to do this optimally, there is still the issue of subsetting the data and choosing the right features and the right models to accompany features in order to make the best predictions.

2 Description of Data

The dataset consists of 4,242 samples with over 13,000 features each. The features (which are usually responses to survey questions or assessment results) can be of 5 types: categorical variables, unordered categorical variables, continuous, binary, and string. Moreover, there is no consistency with respect to the type of data collected for each year from year 0 to year 9. That is to say, the range of variables assessed/collected in year 1 is different from year 3, year 5 and year 9. Even variables that are meant to be identical across years, upon closely inspection of the metadata, phrase questions slightly differently according to year. There is metadata for each variable (feature) available in the form of a csv or text file, and upon inspecting these it is easy to see how even variables of the same type have vastly different range and distribution of responses.

The dataset is provided to us as multiple files. ”Prediction.csv” contains a dummy prediction output for the 6 outcome variables for all 4,242 samples. However, half of these are provided with labels to use as training samples and can be found in ”train.csv”. All samples and labels are relatable by the column ”ChallengeID”. About 1/3 of the training labels were completely removed in order to protect privacy, so in reality the training set consists of about 1,466, rather than 2,121 samples. The training labels themselves are skewed, which is important to take into account when trying to generalize a fitted model. For example, the all the binary labels are mainly 0’s, the ”materialHardship” outcome is skewed very far to the left, while the grit outcome is skewed to the right and the gpa outcome has a loosely bimodal gaussian distribution. These distribution and biases of the training labels have a large impact on model performance.

Moreover, the year 0-9 data which can be found in ”background.csv” is extremely sparse. About 1/2 of all data entries are missing, and sometimes for good reason. Any negative number or NaN value indicates missing data, with a variety of meanings. For example, -9 implies that the entry is missing because the family did not participate in the corresponding survey wave. -6 implies that a respondent was intentionally not asked a question, usually because it does not apply to that respondent based on prior information. -2 means ”Don’t Know” and -1 means ”Refused”.

3 Methods

3.1 Selecting Features

Our first order of business was to narrow down the number of features we were willing to work with for our models. At first there are about 13,027 features in our data set. We then remove constant variables as listed in ”constantVariables.txt” which are features which have the same value across all samples. This brings the feature count down to 10,595. Next, we trim the remaining features based on the proportion of missing data for those features. We find that about 5,336 of the features at this point have more than 80% of their data missing. We remove all of these features. Finally, we are left with 5,259 features, which is still larger than our number of training samples by a factor of 4.

At this point we ran across a recommendation from the Challenge creators that teams focus only on the constructed variables, which are around 600 features crafted directly from the survey data. These are ”constructed metrics” based on the survey data which social scientists have deemed particularly relevant to the year 15 outcomes, and therefore have strong semantic weight in the feature set. Rather than continue to narrow down our 5,259 candidates with some complex Factor Analysis of Mixed Data, therefore, we simply selected whatever constructed variables were left in the 5,259 candidates to move forward. That left us with 484 features.

3.2 Imputation

We did some quick computation over the remaining feature set and found that there was still about 30% missing data, encoded as negative sub-entires in the associated data frame. We thought that 30% was still too high, and rather than just perform a dummy imputation the same for all data types, we decided to write a complex loop to replace missing values intelligently. To do this, we read in the meta-data file associated with this data set and use it to determine what the data type of each of the remaining constructed variable columns were. We decided to convert the ”string” data type to ”uc” via Label Encoding, under certain restrictions on the number of unique encoding for any one variable. We also decided to view ”bin” data as ”uc” data as well. Therefore, for every ”uc”, ”bin”, or ”string” column inspected, we converted all possible missing values to unordered categories. That is to say, we decided to view missing data entries as unordered categories, since we thought there was much underlying information that could be drawn from a response such as ”Don’t Know” or ”Refused” rather than overwriting this information with a mean or mode. For the continuous valued data types, however, namely ”cont” and ”oc” we had no choice but to impute with mean and mode respectively.

3.3 Separation and Scaling

Next we separated the data into (unordered) categorical and continuous types. We had to remove a few string-based features that could not be converted, so the total number of features is 476 at this stage of the code. 262 are unordered categorical features, and 213 are continuous-valued ones. The reason we do this is to target the categorical features for transformation via One-Hot Encoding. This is the standard way to deal with categorical features before sending them into machine learning models, since otherwise the models would try and interpret unordered categories as having some numerical meaning and interactions. This is kinda of like converting each unordered categorical feature into k binary ones, where k equals the number of unordered categories. In addition, we scaled the continuous valued-data all to [0,1] range using Sci-Kit Learn's MixMaxScaler. The reasoning was so that we could interpret the size of coefficients derived from logistic and linear regression more easily. Finally we combine both types of data into a single matrix that can be fed directly into machine learning models.

4 Spotlight Model: Logistic Regression

The model I will spotlight is Logistic Regression (LR), which is a type of Generalized Linear Model, or GLM, that is specialized for predicting binary-valued responses. Indeed, LR assumes that the distribution of the response is a Bernoulli, so either 0 (negative) or 1 (positive). The output of each prediction is not 0 or 1, but a probability estimate which takes the form of a mean parameter estimation. The probability is typically that for which the target sample falls under the positive label. In order to get an output in this range, we use the Logistic Function, as a function of the input which is a linear combination of the sample data points and the weight coefficients for this model. The choice of logistic function can be justified as the inverse of the link function of derived from the pdf of a Logistic Regression model when written in exponential family form. In this sense, the logistic is identified as the canonical response function for the Bernoulli distribution.

Like all GLM's, the probability density function of a Logistic Regression model can be written in exponential family form. We start out with the pdf of a Bernoulli with respect to its mean parameter π. Then we transform the pdf so that it is expressed as a function of some input x. The relationship between x and the mean parameter is defined by some response function "f".

The pdf of a Bernoulli is as follows:

$$P(y|\mu) = \mu^y(1-\mu)^{1-y} \tag{1}$$

Here, μ stands for the conditional mean of the response distribution, which in this case is the mean parameter of a Bernoulli. Exponential family form is as follows:

$$P(y|x) = h(y)exp(\eta^T t(y) - a(\eta)) \tag{2}$$
$$\eta = \psi(\mu) \tag{3}$$
$$\mu = f(\beta^T x) \tag{4}$$

When we convert the pdf for a Bernoulli into exponential form, we get the following.

$$P(y|\mu) = h(y)exp(y\log(\frac{\mu}{1-\mu}) + \log(1-\mu)) \tag{5}$$
$$\eta = \log(\frac{\mu}{1-\mu}) \tag{6}$$
$$\mu = f(\beta^T x) \tag{7}$$

Now the reason we convert to exponential family form is because it helps us choose a very convenient choice for the response function f. We choose f to be the inverse of the link function which generates the natural parameter η so that the pdf becomes a function of the input x, which enters the model via a linear combination with the coefficient parameters, hence the name "GLM". It

turns out the inverse of the log odds ratio is the logistic function. Hence our choice for μ is as follows.

$$\eta = \log(\frac{\pi}{1-\pi}) \tag{8}$$

$$\mu = \frac{1}{1+e^{-\eta(x)}} \tag{9}$$

Now we can return to our Bernoulli pdf with a convenient choice for μ, and perform a Maximum Likelihood Estimation for the coefficient parameters β. By taking our objective to be the log likelihood (log of the pdf), and taking a derivative with respect to β we can define an update rule for β using an iterative process like Gradient Descent, since this is not nice anlaytical closed form for the optimal parameters. The derivative and update rule are as follows:

$$\frac{dL}{d\beta} = \sum_{1}^{n}[y_i - \mu_i(\beta^T x_i)]x_i \tag{10}$$

$$\beta^{t+1} = \beta^t + \rho\frac{dL}{d\beta} \tag{11}$$

5 Results

Mean Squared Error - Leaderboard			
	GPA	GRIT	HARDSHIP
Linear Regression	0.377 Elastic L1 Ratio = 1.0	0.216 Elastic L1 Ratio = 0.1	0.026 Elastic L1 Ratio = 0.1
SVM Regression	0.467 C=300, kernel="poly"	0.321 C=300, kernel="poly"	0.033 C=300, kernel="poly"

Brier Loss Score - Leaderboard			
	EVICTION	LAYOFF	JOB TRAINING
Logistic Regression	0.135 Regularizer = L2	0.244 Regularizer = L2	0.240 Regularizer = L2
SVM Classification	0.133 C=300, kernel="poly"	0.233 C=300, kernel="poly"	0.233 C=300, kernel="poly"

6 Discussion

Let's start with some important model-specific prediction caveats. Although we included one-hot encoding in our training sample matrix, we did not incur co-linearity problems under either linear or logistic regression since we also enforced some degree of regularization. With either L1 or L2, the one-hot feature columns the would have been co-linear are implicitly taken care of.

Also, as reviewed in precept, the binary labels are quite biased in favor of 0/negative label. In order to make our models more generalizable and anticipate the distribution of the held-out set, we

always set the ”class_weight” parameter to ”balanced”. This enabled our model to appropriately re-scale coefficients with respect to the frequency of class labels.

Lastly, we had to use makeshift probabilities for the SVM - SVC output, since SVM's do not naturally yield any theoretically sound / nicely interpretable probabilities. We worked around this problem by using the decision_function() output for sci-ki learn's svm.SVC class. The output of this function is the distance of a sample to the hyperplane, which we then scaled accordingly to derive a reasonable probability estimate, which actually performed better in terms of Brier Loss Score than our Logistic Regression model.

Finally, we decided to look at the largest coefficients derived from the trained Logistic and Linear regression models to get a sense of which constructed variables were most important for a particular outcome label type. We used the meta-data to group the 10-20 most important features for each of our 6 outcome label types into topics. Our results below show the most important feature topics (umbrellas) for each label type. It is notable that race, household income, and parental relationship status are considered important to nearly every outcome. This is not too surprising. What is more interesting is which feature topics were not found to be strongly associated with any given label type. For instance, Sex/Gender was only strongly associated with Grit, and Behavior was only strongly associated with Eviction status.

Most Important Feature Topics Per Label Type

	GPA	GRIT	HARD	LAYOFF	EVICT	JOB
Cognitive Skills	✓					✓
Height and Weight	✓			✓	✓	✓
Educational Attainment/Achievement	✓		✓		✓	✓
Parental Relationship Status		✓	✓	✓	✓	✓
Age					✓	✓
Race/Ethnicity		✓	✓	✓	✓	✓
New Partner Relationship Status						✓
Household Income/ Poverty	✓	✓		✓	✓	✓
Household Composition			✓	✓	✓	
Behaviour					✓	
Mental Health	✓	✓	✓	✓		
Fertility History				✓		
Sex/Gender		✓				

7 Conclusion

I think there are a few important extensions that could be done for this work. Namely, More complex models could be considered, such as Gaussian Process or Time Series that take relative time of each data point into account. Also, smarter imputation like Matching Method imputation could be performed, which would require some rigorous definition of similarity across samples and features. Lastly, one might ignore the constructed variables and try to use PCA / MCA instead to determine mathematically which features should be passed towards prediction models.

8 References

[1] Murphy, K. Machine Learning: A Probabilistic Perspective. MIT, in press. (MLAPP)

Acknowledgments

The following packages and tools from sklearn were utilized for this assignment:

linear_model.LogisticRegressionCV
linear_model.ElasticNetCV
preprocessing.MinMaxScaler
metrics.brier_score_loss
metrics.mean_squared_error
svm.SVC
svm.SVR
GridSearchCV