2015

# Machine Learning for Predictive Analytics Made Easy – A Case Study

David Hansen
*George Fox University*, dhansen@georgefox.edu

# Machine Learning for Predictive Analytics Made Easy – A Case Study

David M. Hansen, Ph.D.
Department of Computer Science and Electrical Engineering
George Fox University
Newberg, OR

## Abstract

*Machine Learning and Artificial Neural Networks are mature, easy-to-use technologies that remain under-utilized. We present a case study demonstrating the ease-of-use and effectiveness of freely available open-source Artificial Neural Network software to predict prospective student matriculation for University admissions. We discuss data collection, formatting and transformation, and assess our results.*

## 1. Background and Motivation

In recent years our University has relied predictive analytics to help predict the likelihood of prospects matriculating at the University. While the University enrolls only 500-600 new undergraduate students each year, those students are drawn from a pool of 35,000-70,000 "prospects" often consisting of little more than basic demographic information (e.g., name, address, age, gender, race, high school, academic interest) obtained via a variety of sources. Predictive analytics are used to rank prospects against historical students; a higher ranking implying that a prospect is more similar to the average historical student, and therefore more likely to matriculate, but otherwise saying little about the relative strength or weakness of the prospect as a student.

The most obvious use of this sort of predictive data is to help optimize the recruiting and admissions process, focusing on those prospects most likely to matriculate as well as on desirable prospects that may be predicted to be somewhat less likely to matriculate.

A less obvious use of the data is to aid in the promotion of new initiatives at the University such as the resurrection of an intercollegiate Football program or a recently implemented "Honors Program." In both instances the programs are expected to enroll students with lower predictive rankings because they don't "look like" students currently attending the University. Nonetheless, the use of rankings helps identify prospects who are relatively similar to existing students and therefore likely to matriculate.

In a general sense, the goal of predictive analytics at the University is to optimize the process of finding and enrolling new students (customers) for our courses and majors (product).

Others have applied machine learning to aspects of University admissions for predicting the enrollment of "applicants" [1, 2]. Chang gives a comparison of logistical regression vs. neural networks with SPSS for considering University applicants, finding neural networks to be more accurate [2]. We go beyond these efforts to develop, apply, and assess predictive models for "applicants" and "admitted" students as well as much earlier in the recruiting process, using much less data, for "prospects."

## 2. Traditional Statistical Techniques

Most recently our University has relied on a commercial 3rd-party service to generate predictive rankings for prospects. Each fall two years of historical data is extracted from University databases and sent to a service that uses traditional statistical techniques, such as linear-regression, to painstakingly build a predictive model that can be applied to prospective students for the coming year.

Although the process is demonstrably effective (see comparison in Section 4), it is somewhat limited and inflexible as it relies on very little data and generates a single prediction for each prospect.

## 3. Experimenting With Machine Learning

After suggesting that artificial neural networks (ANNs) might be as effective, we were given access to the University's prospect data and invited to experiment with machine learning and ANNs to 1) see if the technology worked with this data so that we

might 2), replicate the 3rd-party service and 3), develop additional models that could be run throughout the recruitment process as a prospect moved on to become an "applicant" and finally "admitted."

### 3.1. Machine Learning Software

Being Computer Science faculty and programmers, we chose to use a low-level ANN framework called FANN - an open-source C-library for building machine learning applications [3, 4]. The essential code for constructing, training, and saving a network amounts to 15 lines of C-code that invokes 9 FANN functions (specifically we build a 3-layer ANN that has #inputs/2 nodes in the hidden layers using the `FANN_ELLIOT` output activation function and trained with the `FANN_TRAIN_RPROP` training algorithm).

FANN uses a simple file format where each training datum is represented by a vector of numeric input values followed by one or more output values.

### 3.2. Data Collection and Transformation

The collection and transformation of data to create numeric data suitable for use with FANN accounts for much of the work. At their simplest, ANNs take a vector of numeric inputs and generate one or more numeric outputs. The process of training an ANN is a matter of repeatedly presenting the ANN with vectors of inputs together with known correct output(s) so that the ANN gradually "learns" how the output(s) are a function of the inputs [5]. Before an ANN can be developed, data must be suitably transformed into numeric input values.

We began by augmenting the small demographic data available to us from the University database with U.S. Census data, pulling zipcode-specific data via a U.S. Census web service as we retrieved prospect data from our University database [6]. This yielded a collection of input data that included the prospect's name, gender, race, state, zipcode, number of visits to the University, how and when we obtained their name, along with zipcode-based population, level of education, ethnic diversity, median age, income, home value, etc.

The second task was transforming the data into a numeric vector format suitable for use with FANN.

Transforming numeric data is generally straightforward and often requires no additional work. Nonetheless, we did "normalize" all of our numeric data to the range `0.0-1.0`, a common practice when using ANNs. For example, we took the median household income of the prospect's zipcode and used the value `min(1.0, income/250000.0)`; similarly we used the value `min(1.0, age/60.0)` for the zipcode's median age. Other values are expressed as simple ratios such as the percentage of the population with a college education.

Non-numeric data takes a number of forms and it is important that transformations be done carefully to avoid introducing irrelevant artifacts into the data. Three common forms of non-numeric "categorical" data are *ordinal*, *interval*, and *nominal* data [7].

Ordinal categorical data are collections of named values that have an *intrinsic ordering* among the values. Students are familiar with the ordinal category "grade" with values `{A, B, C, D, F}`. Ordinal category values can generally be transformed into a numeric value fairly easily (e.g., `{1.0, 0.75, 0.5, 0.25, 0.0}`), however one must be careful to avoid the assumption that the distances between the values are always uniform – relative differences between values are significant when using ANNs. Our data contains no ordinal categorical data.

Like ordinal data, interval data are ordered but implicitly by fixed-sized units of measure. Dates, for example, can be treated as a sort of numeric data and the interval between two dates used in a meaningful way. Our data includes a number of dates – when the prospect entered the system, when they applied for admission, etc. While the dates themselves are not meaningful numeric values (even though internally represented as such), the distance between dates (size of the interval) may be. A prospective student is counted as "matriculated" if they are enrolled in classes after a particular future date (e.g., the first Friday of the fall semester). Using that date as an end-point, we compute an interval for every date that is the distance from the end-point. These distances are normalized to the range `0.0-1.0` by assuming that prospects become students within at most one or two years. So we would represent the "applied date" of a prospect for the 2014 academic year who submitted their application on November 1, 2013 by computing the number of days between November 1, 2013 and the matriculation date of September 5, 2014 and dividing that number by 365. This use of intervals satisfies our intuition that prospects who apply at different times may matriculate at different rates; converting dates to intervals allows the ANN to determine whether or not such a relationship exists.

Nominal categorical data are generally non-numeric enumerated values that lack any order among the values (e.g., gender `{Male, Female}`). It is a mistake to map nominal categorical data onto numeric values as we did above for grades because the ANN will try to make sense of the meaningless notion that the gender "Male" is somehow less than or greater than the gender "Female." In fact, some data masquerade as

numeric data with the false implication that order is relevant. We treat zipcode, although represented as a number, as a nominal category lest the ANN be confused by the fact that zip 10101 is not in any meaningful sense "less than" zip 10102.

One way to deal with nominal categorical data is to create a binary value for each of the possible categorical values setting one of the values to `1.0` and the rest to `0.0`. E.g., instead of a "gender" category we introduce mutually exclusive "male" and "female" categories. This approach works for categories with small numbers of values but becomes cumbersome for many-valued categories (e.g., the hundreds of High Schools from which our prospects are drawn).

For many-valued nominal category fields, such as High School attended, we instead convert these fields into another common type of numeric data – a *ratio*. We compute two ratios for the field based on our historical data. The first is the ratio of how often the field takes on the same value among all historical prospects. The second is the ratio of how often prospects with that same value matriculated. So, for example, a small High School may appear infrequently in our historical data, but it may be the case that a large percentage of the time that it does appear, the prospect matriculates. These two ratios attempt to capture these two aspects of multi-valued nominal data.

Finally, we leveraged a zipcode database providing latitude and longitude to compute an as-the-crow-flies distance from the University for each prospect.

The end result is a 28-value vector of numeric input values.

### 3.3. Training and Prediction

To develop a predictive ANN for a given year, we pull five years of historical data from the University database, augmented by U.S. Census data, for training (training on smaller three-year histories gave poorer results). For example, we trained an ANN on 238,623 prospects from 2007-2011 to predict the likelihood of matriculation for each of the 58,276 prospects for the 2012 academic year, of which 447 (0.8%) prospects matriculated. The training process takes a matter of minutes on an Intel Core i5-based MacBook Pro; generating predictive matriculation values for the all of the 58,276 prospects using the trained ANN takes fractions of a second.

## 4. Analysis of Initial Results

To test the efficacy of machine learning and ANNs for this task, we initially built an ANN to predict matriculation for the fall of 2012. We did this for two reasons. First, we began this work in the summer of 2013 allowing us to compare the predictive output of the model with the ground-truth of the previous academic year. Second, the 3rd-party service employed by the University had generated predictions for the fall of 2012, allowing us to directly compare our ANN-based predictions with the 3rd-party service's linear-regression-based model.

As shown in Table 1, the 3rd-party service reported their accuracy for each cohort of ~1600 prospects. In the top cohort, 161 prospects matriculated, accounting for 36% of the total number of matriculating students.

By contrast, our ANN correctly identified 346 prospects in the top cohort, accounting for 77% of the total.

**Table 1. Predictive accuracy by cohort**

| Cohort | Commercial Service | ANN |
|---|---|---|
| 1-1600 | 161 | 346 |
| 1601-3200 | 92 | 76 |
| 3201-4800 | 65 | 19 |
| 4801-6400 | 38 | 6 |
| 6401-8000 | 30 | 0 |
| 8001-9600 | 19 | 1 |
| 9601-11200 | 19 | 0 |
| 11201-12800 | 13 | 0 |
| 12801-13400 | 7 | 0 |
| 13401-16000 | 3 | 0 |

As Figure 1 shows, the ANN did a much better job of accurately predicting matriculation, correctly ranking 77% of matriculating students in the top cohort and identifying nearly 100% within the top three cohorts.

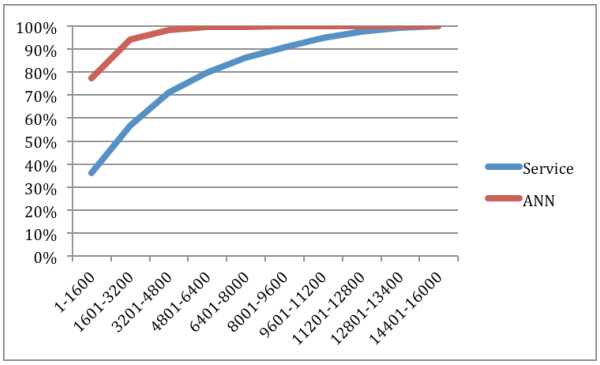


**Figure 1. Model comparison**

Based on this initial success we generated a predictive model for the fall of 2013 for use by University admissions counselors. We also generated

and validated the model for 2011 and most recently began validating predictions we have generated for the fall of 2014 (temporarily treating students who have made a housing deposit as "matriculated").

Figure 2 shows the consistent year-to-year accuracy of what we have come to call our "Fall" model (as described in Section 3.2) at a higher resolution over cohorts of 500 prospects (approximately one centile of all prospects). The graph depicts the number of prospects from each centile that matriculated, with the top centile of prospects consistently matriculating at a rate of 40-50%.

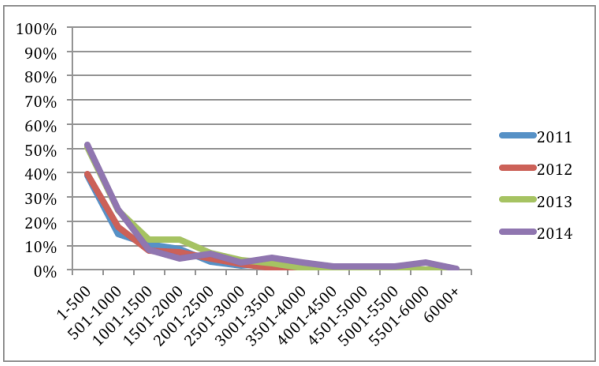


**Figure 2. Year-to-year model consistency**

Figure 3 shows the cumulative accuracy of our Fall model demonstrating that, of ~50,000 prospects in each year's input data set, the models capture ~70% of matriculating students within just the top two centiles (2%) of all the prospects ranked by the model.

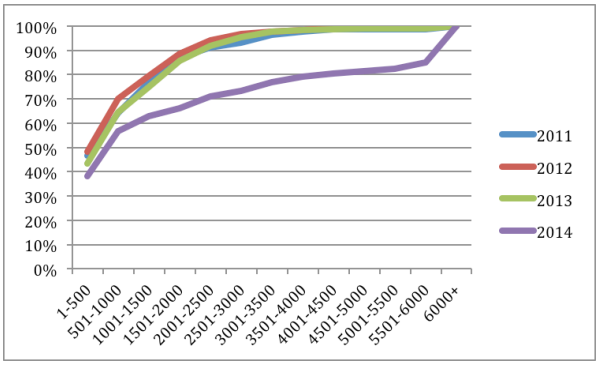


**Figure 3. Cumulative accuracy**

While the 2014 results in Figure 3 look less accurate than previous years, 2014's numbers are preliminary and include a fair number of students who have deposited, but will not actually matriculate. By the time we confirm the model's performance in the fall of 2014, we expect that some of those 15% in the tail end of the model will *not* have matriculated, confirming the model's prediction, making the accuracy more consistent with previous years.

## 5. New Models

Given our success in developing a Fall model relying on a relatively few demographic input values, we sought to exploit our access to additional University admissions data to build two additional models – Applied and Admitted models that can be used later in the recruiting and admissions cycle.

### 5.1. The Applied Model

The Applied model is an ANN that is constructed similarly to the Fall model, but restricted to historical data for prospects that completed an application for admission to the University (~4-8% of prospects become applicants) and only run for a prospect once they have completed an application for admission. For the 2012 Applied model, we trained an ANN using the 5858 applicants from 2007-2011 and generated predictions for each of the 2043 applicants for 2012.

Applications provide additional data including standardized test scores, high school GPA, and household financial information, growing our input vector from the 28 numeric values of the Fall model to 53 values for the Applied model.

Unlike the Fall model, whose predictions will remain relatively static, the Applied model's predictions will change as data incrementally becomes available. Prospects often complete an application before standardized test scores or financial information arrive at the University. To deal with this "missing data" we have taken the approach of using University averages for data such as test scores, and zipcode averages for data such as household income. Thus, as data continues to trickle in, a prospect's ranking under the Applied model may change; we anticipate integrating this model's output into a "dashboard" for Admissions counselors, updating a prospect's ranking on a regular basis.

Figure 4 presents the year-to-year consistency of our Applied model in deciles of prospects that applied.

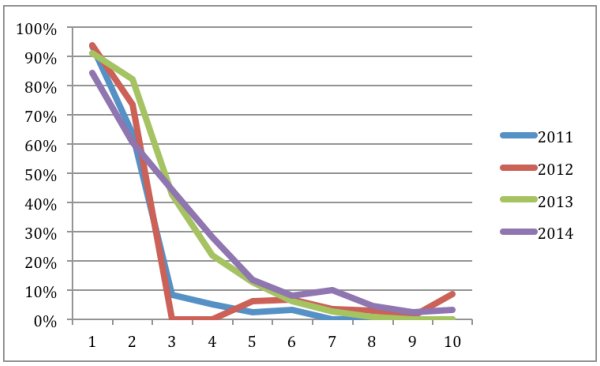

**Figure 4. Applied model consistency**

The Applied model is significantly more accurate than the Fall model due to the additional, more personalized data. Among the top two deciles, the Applied model is correct 70-90% of the time.

Figure 5 also demonstrates the accuracy of the Applied model as the top four deciles capture over 80% of matriculating prospects.

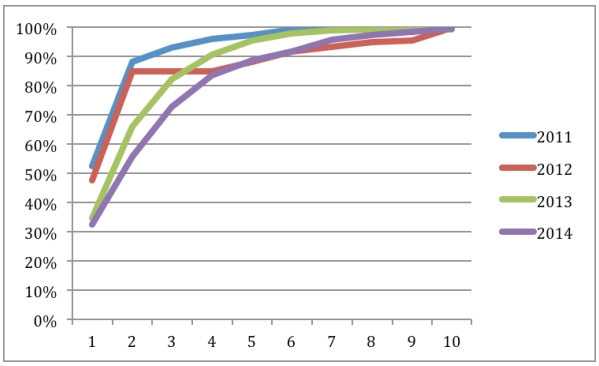


**Figure 5. Applied cumulative accuracy**

## 5.2. The Admitted Model

Like the Applied model, the Admitted model narrows the focus again, this time training the ANN on only those prospects who applied and were subsequently admitted; approximately 1/3 of admitted prospects will matriculate. The Admitted model adds one additional piece of data to the input used for the Applied model – the date the prospect made their housing deposit.

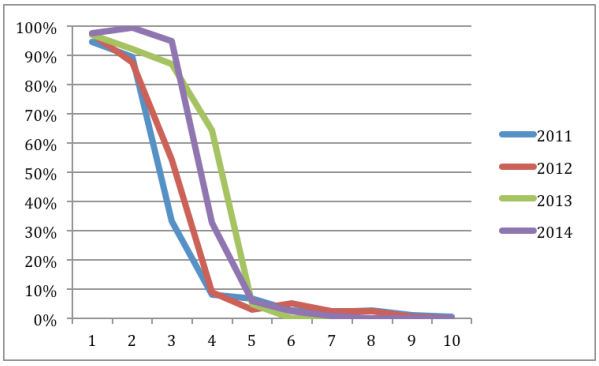


**Figure 6. Admitted model consistency**

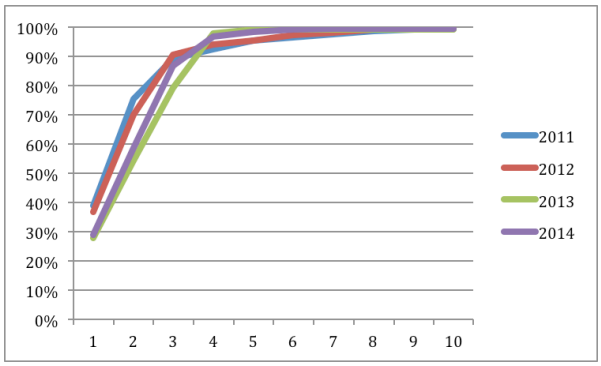


**Figure 7. Admitted model cumulative accuracy**

The Admitted model displays an almost binary separation between those the model predicts will matriculate and those that will not as shown in Figures 6 and 7. Using the Admitted model in the summer, we can very accurately identify the prospects who will enroll and attend in the fall and those that will not.

Beyond their useful predictive ability, we have found that the results of our ANNs are also interesting for the artifacts they reveal about the admissions process. For example, the slight drop off in accuracy for 2013 and 2014 seen in Figure 5 is interesting when taken in context (as noted previously, 2014's accuracy should improve once final numbers are known). Both 2013 and 2014 1) set new records for Freshman enrollment and 2), saw new initiatives at the University that emphasized recruiting students somewhat outside historical norms. In 2013 and 2014 the University admitted ~75 student-athletes each year to participate in our resurrected intercollegiate Football program and in 2014 the University initiated a new Honors Program for ~40 outstanding scholars. Both programs slightly change the makeup of the incoming classes, perhaps explaining why the Applied models for 2013 and 2014 are slightly less accurate than 2011 and 2012 models. Interestingly, Figures 6 and 7 show that the Admitted models for 2013 and 2014 are just as accurate as the 2011 and 2012 models. This suggests that although some of the students were a bit outside historical norms for applicants, when compared among previously admitted students, the Admitted model accurately predicts their matriculation. Going forward, these students will become part of the historical context used for building ANNs in subsequent years, helping to maintain accuracy in the future.

# 6. Conclusion

Application of machine learning and ANNs to our University admissions process has been simple to implement and quite effective at very low cost. The three models we have developed provide the

University with useful information about prospects throughout the admissions process, accurately identifying prospects that will matriculate the following fall. Our next application of machine learning will be retaining the students we've helped the University recruit by identifying Freshman who are at risk for not returning for their Sophomore year.

It is worth noting that we have done little to optimize our selection of data or the parameters of the FANN system. Our approach has been to use as much data as we reasonably can (we use only a few attributes from the U.S. Census data available) without thinking too much about whether or not the data is "useful" to the model. One benefit of ANNs is that input data that does not correlate with the output will be ignored to some degree.

More generally, machine learning and Artificial Neural Networks are a mature, robust, easy-to-use technology with the potential to greatly enhance data analytics for many organizations and enterprises.

In addition to FANN, mature software tools are available for machine learning and the development of ANNs including popular commercial tools such as Matlab, SAS, SPSS, etc. Among other open-source machine learning toolkits, the Java-based Weka toolkit (http://www.cs.waikato.ac.nz/ml/weka/) appears to offer help in automating portions of the data transformation task described in Section 3.2 [7]. More recently, web-services such as the Google Cloud Prediction API (https://cloud.google.com/products/prediction-api/) and BigML (https://bigml.com/) are beginning to offer online tools and APIs to facilitate predictive analytics.

## 7. References


[1] W.D. Amburgey and J.C. Yi, Using Business Intelligence in college Admissions: A Strategic Approach, International Journal of Business Intelligence Research, 2(1), 1-15, January-March 2011.

[2] L. Chang, Applying Data Mining to Predict College Admissions Yield: A Case Study, in J. Luan and C. Zhao (Eds.), New Direction for Institutional Research (131), Wiley, San Francisco, 2006.

[3] S. Nissen, Implementation of a Fast Artificial Neural Network Library (FANN), Department of Computer Science University of Copenhagen (DIKU), Tech. Rep., 2003.

[4] Fast Artificial Neural Network Library (FANN) from http://leenissen.dk/fann/wp/, 2013.

[5] Peter Norvig and S. Russell, Artificial Intelligence: A Modern Approach, Prentice Hall, 2009.

[6] U.S. Census Bureau, American Community Survey (ACS), from http://api.census.gov/data/2012/acs5/, 2012.

[7] I. Witten, E. Frank, and M. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2011.

[8] T. Chang, Data Mining: A Magic Technology for College Recruitment, Paper of Overseas Chinese Association for Institutional Research (www.ocair.org), 2008.

[9] D. Aadland, R. Godby, and J. Weichman, University of Wyoming Enrollment Project Final Report from http://www.uwyo.edu/accreditation/_files/docs/econometric_analysis_on_fin_aid_07.pdf, 2007.