Applications of Supervised Learning Techniques on Undergraduate Admissions Data

Thomas Lux
Roanoke College
221 College Lane
Salem, Virginia 24153 U.S.A
thlux@mail.roanoke.edu

Randall Pittman
Roanoke College
221 College Lane
Salem, Virginia 24153 U.S.A
rbpittman@mail.roanoke.edu

Maya Shende
Roanoke College
221 College Lane
Salem, Virginia 24153 U.S.A
mshende@mail.roanoke.edu

Anil Shende
Roanoke College
221 College Lane
Salem, Virginia 24153 U.S.A
shende@roanoke.edu

ABSTRACT

In making undergraduate admissions decisions, colleges and universities must take a large amount of data into consideration for each applicant. Surprisingly, there is almost no work reported in the literature for a systematic, automated use of the wealth of data gathered by an institution over the years; such a system could guide admissions offices in targeting applicants so that their yield (the applicants who enroll) is maximized by effectively distributing resources (counselors' time and energy) across applicants.

We discuss the use of supervised learning techniques, namely perceptrons and support vector machines, in predicting admission decisions and enrollment based on historical applicant data. We show through experimental results that a classifier, trained and validated on previous years' data, can identify with reasonable accuracy (1) those applicants that the admissions office is likely to accept (based on historical decisions made by the admissions office), and (2) of the accepted applicants, those ones that are likely to enroll at the institution. Additionally, the results from our feature selection experiments can inform admissions offices of the significance of applicant features relative to acceptance and enrollment, thus aiding the office in future data collection and decision making.

CCS Concepts

•**Computing methodologies** → **Machine learning; Supervised learning by classification;** *Feature selection;*

DOI: 10.1145/1235

Keywords

Neural Network; SVM; Machine Learning; Prediction; Classification; Undergraduate Admissions

1. INTRODUCTION

Undergraduate admissions offices across the country read thousands of applications every year trying to efficiently determine which they should accept and reject. A "success" for an admissions office is an instance of an applicant accepting an offer of admission. Admissions resources could be optimally utilized if they could target these applicants.

Data from previous years can predict admissions decisions for an upcoming year, and these predictions can be used to aid admissions offices in making decisions that benefit their institution's mission. Surprisingly, there is very little literature on systematic uses of this data to help admissions counselors make informed decisions; instead, admissions counselors rely primarily on their experience and "gut feeling".

In this paper, we discuss applying machine learning techniques to help admissions counselors focus their efforts on applicants who are more likely to enroll. We first describe, in Section 2, the typical admissions process. Our literature survey, in Section 3, shows the lack of research supporting the admissions process. Then, in Section 4, we discuss our methodology to make predictions that guide admissions counselors. Besides providing useful information on whether an applicant should be pursued further, our results also reveal the need for better, more uniform coding mechanisms, and suggest the relative importance of certain applicant features over others. We present our results, and discuss their implications in Section 5. We conclude in Section 6 with a discussion on possible future directions for this work.

2. THE ADMISSIONS PROCESS

The admissions process typically follows these stages:

Stage 0 Applicants complete their applications.

Stage 1 Admissions counselors read the applications and make an offer to certain applicants.

Stage 2 Some of these applicants accept their offer.

In Stage 0, applicants provide data on themselves. Some of this data can be coded quantitatively (e.g., SAT scores, High School grades, etc.), while other pieces are qualitative (e.g., a personal essay, letters of recommendation, etc.). Admissions counselors, based on their experience, code some of the qualitative data in the applications.

According to The College Board, there are three main approaches to how the decision in Stage 1 is made: (1) based on the institutional mission, (2) based on the institution's definition of a "successful student" through the student's tenure at the institution, and (3) based on what characteristics the institution would like to see in its incoming class [10]. Besides the information from the application forms, there are some other pieces of information that are taken into consideration, e.g., a formal or informal interview with the applicant, the number of times the applicant contacted his/her admissions counselor, etc. Based on all of the above information, the admissions office makes offers, typically accompanied by a scholarship package, to some of the applicants.

In Stage 2, students with offers decide whether or not to enroll at the institution. The admissions office has very little information, if any, about how students make their decisions.

Understandably, admissions offices strive to make the most attractive offers to students who are most likely to enroll, while maintaining desirable characteristics of the incoming class and adhering to the institutional mission.

3. RELATED WORKS

The literature contains many studies which make predictions about student retention at colleges and universities [3, 5]. These studies focus on defining which factors (socio-demographic, academic, environment, etc.) contribute the most to retaining students. However, little has been done in making predictions about whether a given applicant will gain admission to, and then enroll at an institution.

Waters and Miikkulainen [12] used machine learning to predict graduate admissions in the computer science department at the University of Texas at Austin. They developed a statistical machine learning system that makes admissions predictions based on historical admissions data. Their predictions were restricted to whether the department accepted applicants or not based on previous acceptance data. They found that the features used most by their classifier (e.g., Undergraduate GPA, previous institutions, GRE scores) were those used by human reviewers during the review process.

Moore [8] worked on predicting decisions for graduate admissions as well as success rates of those admitted. Moore reports that the optimal algorithm used only academic and professional information to determine whether a given applicant may be accepted and/or succeed. This work did break the predictions into two stages, however the second stage involved predictions of success after a student had been admitted.

There is a middle stage (what we refer to as stage 2 in Section 2) – whether a student accepted by an institution will choose to attend the institution – which was not studied in either of the two projects mentioned above. Little research has been done in making predictions in this stage. There is literature on the use of probability models of admission and enrollment at a private, liberal arts college in an effort to pursue quality improvement [2]. However, they analyzed past admissions decisions and outcomes of those decisions to determine whether the selection process of that selective liberal arts college was in line with the college's institutional goals.

Our work goes one step beyond that which has been done in the aforementioned research. We make the assumption that the admissions office for the institution made past decisions with the institutional mission in mind, and our models use these past decisions to predict future decisions that the admissions office may make (stage 1), as well as the outcome of these decisions (stage 2).

4. METHODS

Admissions offices have a rich data set of information on past applicants including previous decisions made in stage 1 and decisions made by the applicants in stage 2. Using this data, we employ supervised machine learning techniques to classify applicants for each of stages 1 and 2 in the hope that the classifiers can assist admissions offices by improving their ability to allocate resources.

We use data provided to us by a small private liberal arts college. We use a standard Multi-Layer Perceptron (MLP) with a sigmoid activation function and back-propagation as well as a Single-Layer "Perceptron" with a linear activation function to make predictions. We also present predictions from a Support Vector Machine (SVM) implementation that utilizes three different kernel mappings : linear, polynomial, and RBF [6]. The implementations of the MLP and Perceptron are our own and can be obtained by contacting any of the authors, while the SVM's can be obtained from the publicly available Scikit-Learn package for Python3 [9].

There are three phases to our predictive modeling, they are: *data preprocessing* in which we transform the raw data set provided by the admissions office into a numerical form accepted by each classifier, *classification* where we train each classifier in an attempt to make useful predictions about future applicants, and *feature selection* where we use a classifier to attempt to identify which features in an application are most influential in the stage 1 and 2 predictions.

4.1 Data Preprocessing

Certain variables in our data set, as shown in Table 1, are created from the combination of other variables in the original data set provided by admissions. In order to determine the stage 1 target variable, we check whether the "acceptance date" variable[1] is non-empty. However, a positive value in the stage 2 target variable requires that the applicant made a deposit and has not withdrawn from the institution before beginning the first semester. Thus, an applicant is considered to be attending the institution if and only if the deposit date is nonempty and the withdrawal date is empty.

The decile variable is created from the combination of a normal decile ranking with an additional e-decile variable.[2] According to the admissions office, the decile variable is a more confident ranking of potential applicants, but a large portion of the data for the column is missing. Thus, as advised by the admissions office, we use the value from e-

[1]The date an institution accepts an applicant.

[2]The decile variable is provided by the applicant's high-school and represents class rank percentage truncated to the nearest 10% interval. The e-decile is the decile value that the institution estimates if the high-school does not provide a ranking.

Table 1: Variables/types used in the data set.

Variable	Data Type
Stage 1 Target	Binary
Stage 2 Target	Binary
Age	Integer
Gender	Binary
SAT Cumulative	Integer
SAT Verbal	Integer
SAT Math	Integer
Recalculated GPA	Float
Predicted GPA	Float
Decile	Float
Number of Visits	Integer
Number of Interviews	Integer
Number of institutional contacts	Integer
EFC	Integer
Has EFC	Binary
AP Honors	Integer
US Citizen	Binary
TR-TP	Float between 0 and 1
HSS-TP	Float between 0 and 1
HST-TP	Float between 0 and 1

decile whenever decile is missing, and if both are missing then we use the value 0.

The variable "Has EFC" is derived from the EFC variable (Expected Family Contribution) because of the numerous missing entries in the EFC data. Some of the EFC entries are blank, however many other entries for which their family cannot contribute contain the value 0. Inserting 0 into the EFC variable for all blank entries would cause the data to no longer represent the difference between blank and 0 values, and thus the "Has EFC" variable is introduced to help distinguish between missing data and 0 valued entries. Those applicants missing the EFC column are given a value of 0 in the "Has EFC" column.

The three target probability variables listed in the table (TR-TP, HSS-TP, HST-TP) are created from the variables "Territory" (TR), "High-School State" (HSS), and "High-School Type" (HST). The territory variable contains coded combinations of two or three letters denoting the region (as determined by the admissions office) in which each applicant resided. The HSS variable contains the abbreviation for the state of the applicant's high-school, and the HST variable lists the type of high-school the applicant attended. In order to transform these letter coded variables into numbers, we calculate how often an applicant with a specific TR, HSS, or HST value had a positive value in each of the stage 1 and stage 2 targets for the training and validation data. Thus we compute for each possible letter code in HSS the probability that an applicant with that letter code is accepted by the institution, and the probability that applicant would attend the institution given an offer of admission. The same calculation is also performed for TR and HST. This process yields the new numeric variables TR-TP, HSS-TP, and HST-TP. Observe also that since these new target probability variables require knowledge of the target variable, the TR-TP variable for stage 1 will be different from the TR-TP for stage 2, and similarly for HSS-TP and HST-TP. In order to transform these letter codes for the testing data (current applicants), we use the probabilities obtained from the training and validation data.

As mentioned, this data contains many missing entries. In order to maximize our prediction performance, we automatically generate "models" based on which features are not present for a particular applicant. For subsets of applicants that are large enough, we create a unique model for that subset. Our definition of large enough is that a model must contain at least 300 applicants. If a model does not contain at least that many applicants we do not attempt to classify those applications. To demonstrate what we mean by unique model, consider the following. A significant number of applicants do not have SAT data, so we create two models for each stage – one with all features present and one with all features without SAT data. An applicant is placed into a particular model based on which data features are missing from her/his application.

The last step of our data preprocessing is to make all of the columns of the data have a mean of zero and unit variance. We do this in order to improve the rate at which our classifiers converge on good predictions. Since we do not know the mean and standard deviation of testing data in practice, we record the mean and standard deviation for each column of the training and validation data, and then use those values to transform our testing data.

4.2 Classification

As stated previously, we use classifiers to predict two binary target variables (Stage 1 and Stage 2) relevant to the admissions process. The first represents whether or not the institution accepts an applicant, while the second represents whether an accepted applicant attends the institution. For the second stage we remove the data entries for all applicants who are not accepted in the first stage, since only accepted applicants can enroll in the institution.

For the purposes of predicting stage 1 and stage 2 results during an application cycle, the classifiers are trained and validated on the data from previous years and then tested on the current year's data. The values of the target variables are known for training and validation, and hence our use of supervised learning algorithms. Our data set comes from a small institution, therefore we use four years of application data (2010 - 2013) to train and validate the classifiers and produce a "predicted" performance, while the 2014 data is treated as current and we use it to test the "actual" performance of each classifier. For our institution we know the actual values of the target columns for 2014 (i.e., the decisions made by the admissions office), but retain this setup in order to simulate the effectiveness of our procedure as if it were implemented during the 2014 application cycle. Thus, our results have been compared to those of the human reviewers for the test year. We use a random selection of 80% of the 2010-2013 data to train, and the remaining 20% to validate each classifier and produce "predicted" performance measures.

In order to identify which classifier will be "best" for predicting stages 1 and 2 in the current application cycle (2014), we train and validate each of our classifiers on 1000 randomized 80-20 splits of the past data (2010 - 2013). For each of

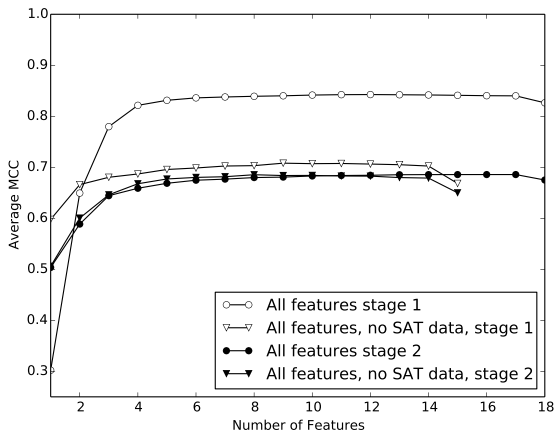

Figure 1: MCC values for varying numbers of features.

our five classifiers, we select the trained classifier with the highest performance ratings in the validation set for its type and use that classifier for testing on the current data.

4.3 Feature Selection

Once we have found the best overall performing classifier using the validation data, we use that classifier to do the computationally expensive task of feature selection. For our feature selection we use a variant of the simple (l,r) search originally proposed by [11] – we use 2 forward steps and 1 backwards step. An overview of this feature selection algorithm as well as a comparison of it with others can be found in [4]. We chose to use the (l,r) search method because it allows us to easily collect a sampling of performance for all feature subsets.

In order to reduce the random noise experienced during the evaluation of a feature subset, we execute 10 independent trials of classification and use the average performance of those trials as our indicator for the classifier's performance on a particular feature subset. Each trial involves training and validation on a new randomized 80-20 split of the data. These repeated trials allow the feature subsets to be chosen with greater confidence in their performance.

Lastly, there is random variation in the selected features from one execution of the (l,r) search to another. In order to identify the most influential features with respect to our target variables, we collect the occurrence of each feature over the number of possible feature sets in which it could have potentially appeared. For each feature this gives a number in the range 0 to 100 which represents the frequency of occurrence of that feature. Note that these percentages across all features are not expected to add up to 100 because we are combining the results of multiple different length subsets of features into a single ranking. We compute the frequency of each feature's occurrence over 100 independent trial runs of the (l,r) search as a measure of feature importance for each model. We use 100 trials for two reasons: beyond 100 trials the reduction in variance of results is not substantial, and this computation is still feasible, requiring approximately 8 hours to run on 25 64-bit computers each utilizing 4-cores at 3.10GHz.

5. RESULTS

We used the Matthews Correlation Coefficient (MCC) [7] to rate the quality of our classifiers. Our variables are equally weighted, and the data entries (individual applicants) are independent, thus satisfying the independence and equivalence assumptions required to use the MCC. Also, our data set is unbalanced, and thus simple accuracy, specificity, and sensitivity ratings can be misleading. We used the MCC in our results because it acts as a combination of each of these measures, works well for unbalanced data sets, and under the independence and equivalence assumptions is a better predictor of success than the other measures [1].

Table 3 displays the results of each classifier for each model. The underlined entries are indepently the maximal "predicted" and "actual" metric values across each row. The far left column provides general information about each model including the sample size. Note that this table emphasizes the importance of using an appropriate metric to evaluate a classifier's performance. There are multiple "100.0" entries for some metrics, but we know that these are a result of a trivial separation of the data.

It is difficult to classify the 2014 applicants in both Stage 1 and Stage 2 with classifiers trained on 2010-2013 applicants. As we can see in Table 3, all of the models achieve more than acceptable performance on the validation data, but have widely varying performance on the testing data. In the table we see that, according to the MCC, the MLP was predicted to be the best classifier for every model except stage 2 with all features. Contrary to predictions, the MLP was never the best classifier in testing, according to the MCC. The variation between predicted performance and actual performance is smaller in Stage 1, which leads us to believe that admissions counselors are relatively consistent from year to year in making offers. On the other hand, it is more difficult to confidently predict Stage 2 results, i.e., the choice of an applicant to enroll at the institution or not.

It is also clear from Table 3 that having a high rating for predicted classifier performance on the previous years' data does not guarantee similar performance when using the model for the next year. Regardless of the classifier, the results indicate that actual performance is likely to be less than predicted performance. The usual suspect for such variation in performance is that the classifiers overfit the training and validation data. In our data set this problem may arise from inconsistencies in the data collection between the 2010-2013 and 2014 academic years, but a more likely reason is the lack of crucial information for stage 2, the scholarship package. We discuss this further in Section 6. Thus, it is possible that a more consistent and rigid encoding scheme for data collection on admission applications would result in a reduction in overfitting as well as increased consistency between predicted and actual performance.

This classification process does provide admissions offices useful information. Note that sensitivity represents the true positive rate of predictions, while specificity indicates the true negative rate. Thus, whenever a model with high sensitivity predicts that an applicant will attend the college, it is more likely to be correct. The same holds true for a model with high specificity and applicants not attending.

On average the best performing classifier we had at the prediction stage was the MLP. We used the MLP for our feature selection and the MCC metric to compare performance as features were added or removed. Feature selection

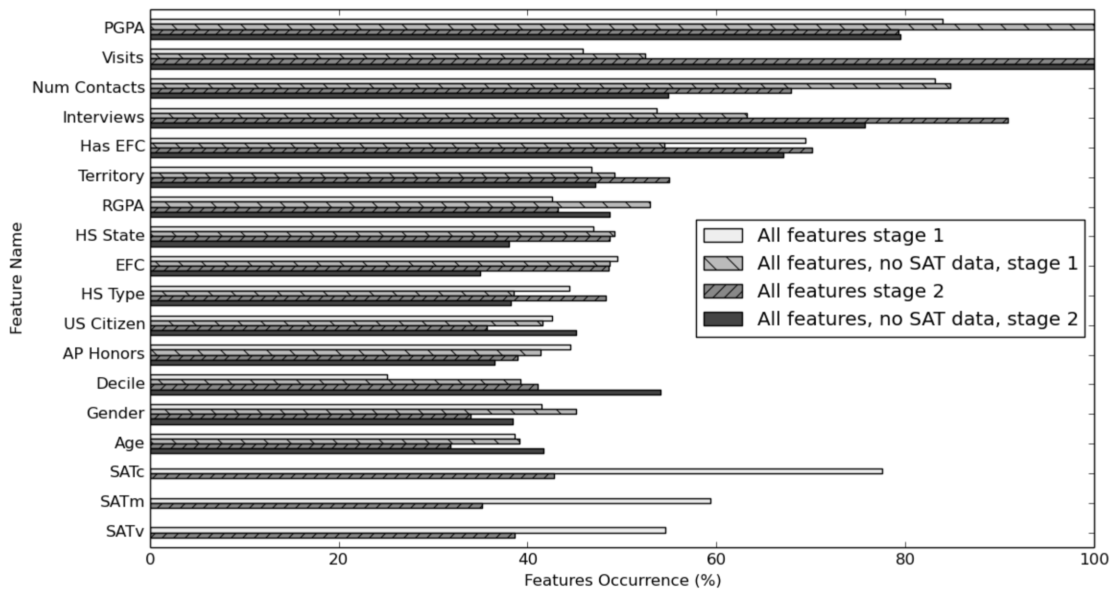

Figure 2: Feature ranking for each model sorted by sum of occurrence of each feature across all models.

Table 2: Five most important features per model.

Stage 1 All features	Stage 1 No SAT	Stage 2 All features	Stage 2 No SAT
PGPA[3]	PGPA	Visits	Visits
Num Contacts	Num Contacts	Interviews	PGPA
SATc[4]	Interviews	PGPA	Interviews
Has EFC	Has EFC	Has EFC	Has EFC
SATm[5]	Visits	Num Contacts	Num Contacts

often yields improved predictions, but we were equally interested in which features of the data had the greatest impact on our predictions. As seen in Figure 1, the performance of each of the classifiers plateaus after approximately 6-8 features are added. This can be useful for our admissions office, because it provides insight as to which features of an application are most indicative of whether or not an applicant will be accepted or enroll at the institution. Also note the drop in performance when adding the last feature to each of the models. It is not immediately clear why we witness such a decrease in performance, but it is likely to be a side effect of randomness and greedy selection in our variant of the Forward Feature Selection algorithm.

Figure 2 shows the most important features as determined by their frequency of occurrence during the (l,r) search. Table 2 lists, in descending order, the best five features for each model. In many ways the most important features in Table 2 agree with the style of the institution from which our data originates, a smaller institution where personal connections are held as most valuable. An Office of Admissions can take these findings and use them to more efficiently utilize resources in the next application cycle.

When an applicant is made an offer by the institution, it is often accompanied by a scholarship package. According to the admissions counselors, the scholarship package is an important factor in the applicant's decision making process and thus is critical for stage 2 predictions. Our data set did not include this information. It seems likely that including this feature in a data set would improve the classifiers' ability to predict stage 2.

6. FUTURE WORK

This project used admissions data from a small institution with approximately 2000 students. Four previous academic years of data was collected to generate a large enough sample size to make confident predictions. However, a larger institution of 10,000+ students could use just the previous year's admissions data. This may also reduce the effects of the overfitting problem experienced by our classifiers.

This research could also be extended to any domain in which a standard application process is followed, such as jobs, scholarships, proposals, etc. Before attempting to implement a similar prediction scheme to the one presented here, it is very important that there exists a foundation of data on previous applicants and the relevant targets.

Lastly, it would be possible to add a third stage where the target variable indicates whether a student stayed beyond the first year to predict retention as well. While research has been done on predicting student retention, the incorporation of our stage 1 and 2 could be used to improve student retention predictions. At the same time, student retention predictions could inform stage 1 decisions for future years.

We would like to thank Dr. Brenda Poggendorf, Ms. Melissa Lacombe, and Mr. Courtney Penn, of the Admissions Office at Roanoke College, for providing us with the data used in this project.

Table 3: Predicted/Actual results for each model and stage for each classifier. MCC is a floating point number between -1 and 1 while accuracy, sensitivity, specificity, and precision are all percentages.

Model	Statistics	MLP	Perceptron	SVM_{Linear}	SVM_{Poly}	SVM_{RBF}
Stage 1	MCC	0.8891/0.6424	0.8379/0.7811	0.8450/0.7942	0.8692/0.7105	0.8747/0.7300
All features	Accuracy	97.03/85.92	95.63/93.77	95.88/94.56	96.35/90.62	96.56/93.39
Training/Validation $N = 11783$	Sensitivity	98.48/84.03	97.37/94.56	97.58/96.33	98.87/90.48	98.62/97.49
Testing $N = 3146$	Specificity	89.47/96.45	86.51/89.35	86.83/84.76	84.35/91.44	86.40/70.56
	Precision	97.99/99.25	97.42/98.02	97.53/97.24	96.78/98.33	97.28/94.86
Stage 1	MCC	0.8945/0.3942	0.7757/0.4501	0.8483/0.5868	0.8056/0.3684	0.8042/0.0000
All features	Accuracy	98.24/74.73	95.73/82.50	96.98/91.17	97.49/78.08	97.24/92.69
No SAT data	Sensitivity	99.72/72.91	99.15/81.77	99.43/91.46	100.00/77.50	99.73/100.00
Training/Validation $N = 1986$	Specificity	84.21/97.92	69.57/91.67	78.72/87.50	66.67/85.42	69.70/0.00
Testing $N = 657$	Precision	98.36/99.78	96.14/99.20	97.21/98.93	97.35/98.54	97.33/92.69
Stage 2	MCC	0.7275/0.2515	0.7321/0.4535	0.7138/0.4729	0.6750/0.0122	0.7227/0.0000
All Features	Accuracy	91.34/41.24	92.20/69.78	91.49/71.84	90.88/16.69	92.10/83.39
Training/Validation $N = 9869$	Sensitivity	76.06/99.55	70.05/96.16	69.27/95.94	60.60/100.00	70.91/0.00
Testing $N = 2627$	Specificity	95.23/29.63	97.38/64.52	96.86/67.04	97.82/0.09	96.84/100.00
	Precision	80.26/21.98	86.18/35.06	84.18/36.70	86.43/16.62	83.39/100.00
Stage 2	MCC	0.8357/0.4607	0.8154/0.6653	0.8080/0.5926	0.6996/0.4849	0.7888/0.0000
All features	Accuracy	95.56/72.25	95.83/90.48	95.28/90.48	93.06/76.52	95.00/84.56
No SAT data	Sensitivity	78.69/94.68	86.67/78.72	74.55/52.13	68.63/90.43	71.70/0.00
Training/Validation $N = 1799$	Specificity	99.00/68.16	97.14/92.62	99.02/97.48	97.09/73.98	99.02/100.00
Testing $N = 609$	Precision	94.12/35.18	81.25/66.07	93.18/79.03	79.55/38.81	92.68/100.00
Overall	MCC	0.8907/0.3356	0.8861/0.6714	0.8885/0.7121	0.8867/0.2071	0.8884/0.7820
	Accuracy	94.57/66.87	94.32/83.40	94.45/85.34	94.36/60.39	94.44/88.80
	Sensitivity	94.36/84.32	93.71/92.32	94.05/94.41	93.87/89.51	94.01/84.16
	Specificity	94.82/46.51	95.04/72.99	94.93/74.74	94.94/26.39	94.95/94.21
	Precision	95.62/64.79	95.69/79.96	95.70/81.36	95.67/58.67	95.66/94.44

7. REFERENCES

[1] P. Baldi, S. Brunak, Y. Chauvin, C. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412 – 424, 2000.

[2] T. Bruggink and V. Gambhir. Statistical Models for College Admission and Enrollment: A Case Study for a Selective Liberal Arts College. *Research in Higher Education*, 37(2):221 – 240, April 1996.

[3] D. Delen. A comparitive analysis of machine learning techniques for student retention management. *Decision Support Systems*, pages 498 – 506, June 2010.

[4] F. Ferri, P. Pudil, M. Hatef, and J. Kittler. Comparative study of techniques for large-scale feature selection. *Pattern Recognition in Practice IV*, pages 403–413, 1994.

[5] Z. Kovačić. Early Prediction of Student Success: Mining Students Enrolment Data. In *Proceedings of Informing Science and IT Education Conference (InSITE)*, 2010.

[6] S. Marsland. *Machine Learning: An Algorithmic Perspective*. Chapman and Hall/CRC, 2014.

[7] B. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozome. *Biochimica et biophysics acta*, 405(2):442 – 451, October 1975.

[8] J. Moore. An Expert System Approach to Graduate School Admission Decisions and Academic Performance Prediction. *OMEGA - International Journal of Management Science*, 26(5):659 – 670, 1998.

[9] 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.

[10] G. Rigol. *Admissions Decision-Making Models, How U.S. Institutions of Higher Education Select Undergraduate Students*. The College Board, 2003.

[11] S. D. Stearns. On selecting features for pattern classifiers. In *Proceedings of the 3rd International Joint Conference on Pattern Recognition*, pages 71–75, 1976.

[12] A. Waters and R. Miikkulainen. Grade: Machine Learning Support for Graduate Admissions. *Association for the Advancement of Artificial Intelligence*, 2013.