Detection of student attrition using machine learning

L. M. Reigoto

***Abstract*— This work presents a method to predict students evasion rate. We accomplish that by using data analysis and machine learning techniques. The model, trained using a database from Clearwater State University, adopts a probalistic approach that can estimate the likelihood of a student attrition. We also extract a equation that shows the contribution of each input variable to the overall risk of evasion. The results obtained from the model can be used to devise strategies aiemd at minimizing the student dropout rate.**

***Keywords*— Student Attrition, evasion, Dropout Rate, Machine Learning, Data Analysis, Support Vector Machine, Predictive Modeling, University, Education.**

I. Introduction

School dropout is a significant problem that affects the educational system and it leads to various consequences in society, such as exacerbating social inequality and wasting resources in higher education. This study proposes to do an analysis of student dropout using data from the ClearWater University. The universities from Brazil have a high student attrition rate[1], as we can see on Table I.

Student Attrition - Brazil (2014-2019)			
Year	**Institution type**	**In-Person Course**	**Online Course**
2014	Private	27.9%	32.5%
	Public	18.3%	26.8%
2015	Private	28.6%	34.2%
	Public	18.4%	28.7%
2016	Private	30.1%	36.6%
	Public	18.5%	30.4%
2017	Private	28.5%	34.9%
	Public	18.6%	27.9%
2018	Private	29.4%	37.0%
	Public	18.5%	31.6%
2019	Private	30.7%	35.4%
	Public	18.4%	31.6%

TABLE I: Student attrition on universities on Brazil - SEMESP[1]

We can use the data from universities in Rio de Janeiro and see from Table II and Table III that the number of people completing courses is much lower than the number of people enrolling in them.

The objective of this research is to estimate a percentual evasion risk for each student at ClearWater University using machine learning techniques. We aim to provide a predictive model that may be utilized for devising strategies to mitigate the high attrition rate

This work is organized as follows. Section II describes the main theorical aspects of this method and previous studies in the literature. Section III has the proposed methodology. SectionIV shows the obtained results and discuss about them. Conclusions are in Section V.

Student Attrition - Rio de Janeiro (2015-2019) - In-Person courses			
Year	**Institution type**	**Incoming Students**	**Graduates**
2015	Private	164170	55006
	Public	39300	16036
2016	Private	145228	60063
	Public	40672	15299
2017	Private	141536	56186
	Public	38627	18283
2018	Private	123732	60614
	Public	39837	17989
2019	Private	114642	58518
	Public	41553	17998

TABLE II: Student attrition on universities on Rio de Janeiro - SEMESP (in-person courses)[2]

Student Attrition - Rio de Janeiro (2015-2019) - Online courses			
Year	**Institution type**	**Incoming Students**	**Graduates**
2015	Private	53552	7358
	Public	11230	1128
2016	Private	62609	10038
	Public	11076	1396
2017	Private	72885	12844
	Public	11494	2528
2018	Private	103045	18218
	Public	12194	2506
2019	Private	114250	22914
	Public	12588	1978

TABLE III: Student attrition on universities on Rio de Janeiro - SEMESP (online courses)[2]

II. Theoretical Framework and Literature Review

A. Support Vector Machine

Support Vector Machine (SVM) [5], [6] is a popular and powerful algorithm know for its robustness. Its strength lies its ability to accommodate a high degree of complexity using a lower complexity degree. This characteristic can be attributed to its support vectors and *Kernel* trick. This makes this algorithm more resistent to the *curse of dimensionality*.

The idea behind SVM is to use the hyperspace of the input parameters and choose the best hyperplane that splits two classes. This hyperplane maximizes the margin between these classes, which is calculated by the support vectors (Figure 1).Support vectors are the data points that touch the margin of the hyperplane. This margin classifier can also deal with non-linearly separable classes transforming the data to a higher-dimensional space phi, which in practice is implemented by *Kernel* functions, where the most common ones are polynomial *Kernel* and *Radial Basis Function* (RBF).

In instances where the data are not linearly separable, SVM employs a technique known as the *Kernel* trick. The *Kernel* trick transforms the data parameters from the input space into a new space where they are linearly separable. The algorithm

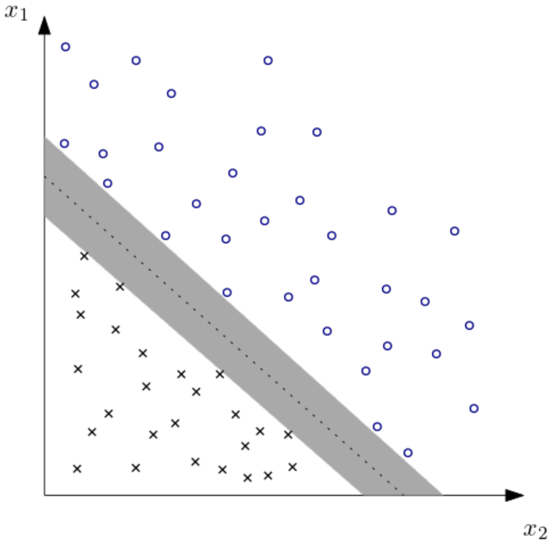

Fig. 1: An example of the margin defined by SVM between two classes in the x_1 vs x_2 graph. The two classes represented by 'X' and 'O' are separated by a plane (in gray) that maximizes the distance margin between these classes.

then operates within this new space (Figure 2), which is often of higher dimension, ensuring the data can be separated linearly.

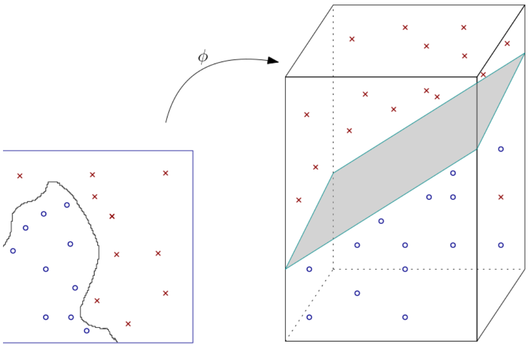

Fig. 2: A *Kernel* function transforms a non-linear 2D input space into a 3D feature space where the data is linear.

B. K-fold Cross Validation

The K-fold method consists in distributing the data between K different folds. Then we train our data K times, each time removing one folder from the training data and using it to evaluate the model. The final evaluating metrics are the average from all the K models. The final model is the combination of all the models.

For classification problems, the final prediction can be the class most frequently predicted across all models. It is common to evaluate a model with K-fold and then train a final model using all the folds for training. Provided a sufficient amount of data and a suitable number of folds are used to reduce noise from specific samples, the theory of generalization [4] posits that our final model is expected to behave similarly or better than the combination of the K original ones.

C. Literature reviews

Previous studies were made using machine learning to predict student attrition. In the study *"Preventing Student Dropout in Distance Learning Using Machine Learning Techniques"*[10] multiple machine learning techniques (*Naive Bayes, Support Vector Machine, Logistic Regression, Artificial Neural Networks and Decision Tree* were utilized to predict student dropout in a specific dataset. The author of this study ended up getting the best performance with *Naive Bayes*.

Another article *Predicting Students Drop Out: A Case Study* [7], uses *Decision Tree, Random Forest, Bayan Classifier, Logistic Model* and *Rule-Based Learner* to predict student evasion on the first semester. The authors concludes that is feasible to learn and predict student evasion looking for just one semester. He also point outs that simple classifiers (with lower complexity) ended up reaching high-accuracy for this problem and more complex classifiers had difficult to beat these models.

In *Mining Educational Data to Reduce Dropout Rates of Engineering Students*[9] the author study a method to predict dropout rates of Engineering Students using machine learning. The author ends up reaching best results with a decision tree classifier.

III. Methodology

A. Original Dataset

The dataset used in this study consists of 3400 student entries with 56 column, 31 of which we are missing data in some entries. It has information from the first two semesters from the analised students and has a information if the student had dropped out by the 2nd semester. This information it's the target that we want to predict.

B. Data Cleaning

1) Categorical data: For categorical data we analysed each attribute mainly for the occurence of each value and for missing data. The data cleaning conclusion and choices can be seen on Table IV

2) Numerical data: For numerical data we analysed each attribute mainly for its distribution, quartiles, scatter plots and missing data. We also generated a new feature by combination of previous features. The data cleaning conclusion and choices can be seen on Table V

C. Training, validation and testing

The dataset was divided into training and testing. The training set was further divided into K-folds and they were used to grindsearch the SVM parameters. In each fold it was used SMOTE (*Synthetic Minority Oversampling Technique*) [8] operation to balance the target classes.

Attribute	Modification
Gender	Replaced 'M' with 1 and 'F' with 0 (binary encoded)
BackGround	1. Created new value "unspecified/others". 2. Replaced the attributes ("BG7", "BG5", "BG8" (few data) with unspecified/others". 3. One Hot Encoded the remaining values
In_State_Flag	Replaced 'Y' with 1 and 'N' with 0 (binary encoded)
International_Sts	Removed the attribute. (3373 entries = 'N' and 27 entries = 'Y')
Major	Many values had few observations. (eg: Liberal Arts - 1, Geology - 4, French - 4) 1. Changed the values with fewer than 60 occurrences to "Undeclared" (undeclared and others). 2. One Hot Encoded the remaining values
Minor	Removed the attribute. 3160 observations with value "N" the other had too few occurrences (top3: Spanish - 26, Psychology - 19, Music - 19)
Housing_Sts	1. Binary encoded
Father_Edu_Desc	Removed the attribute (already encoded as Father_Edu_Cd)
Father_Edu_Cd	1. 4 ('other/unknown') replaced to 0. 2. Filled missing data with 0
Mother_Edu_Desc	Removed the attribute (already encoded as Mother_Edu_Cd)
Mother$_E du\_Cd$	1. 4 ('other/unknown') replaced to 0. 2. Filled missing data with 0
Course_Name *	Attribute removed.(High dimensionality and many attributes have low observations)
Course_Grade *	Attribute removed. We removed the course linked to each grade and will have some correlations with some attributes (earned credit and attempted credits)
High_Schl_Name	Attribute removed. (High Dimensionality and many values with few observations)
Degree_Cd	Attribute removed. Mostly of the data cointains "B", the others values have to few occurrences ("B" - 3384, "A" - 12, "V" - 4)
Degree_Desc	Attribute removed (it's the description of the Degree_Cd)

TABLE IV: Categorical Attributes Modifications

Attribute	Modification
Student Age	Attribute removed. Centered around 18 with a few outliers. Another possibility to separate in 3 bins (bellow 18, 18 and over 18), but that would result in huge outliers.
First/Second Term	Attributes removed.
Test Entrances	Attributes removed. Huge correlation between attributes and all of them missing too many data.
Distance Home	Removed outliers.
High SCHL GPA	No changes except removing the few missing data.
Attempted hours	Some data had earned hours greather than attempted hours. This was probally a mistake in the data. Attempted hours and earned hours values were exchanged for those data.
Reproved hours	New feature generated from the subctration of attempeted hours and earned hours.
Earned hours	Attribute removed. High correlation with attempted hours. The new attribute reproved hours replaced it.
Gross Fin Need	Removed outliers.
Unmet Need	Attribute removed.
Cost of attend	No modifications
Est Fam Contribut	Attribute removed.

TABLE V: Numerical Attributes Modifications

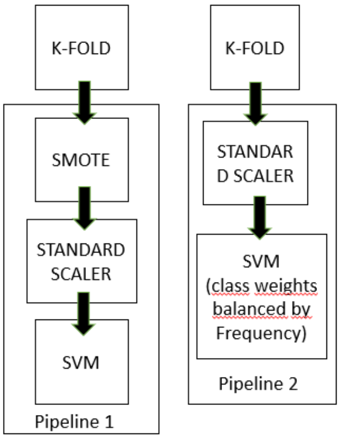

Fig. 3: Block Diagram.

SMOTE is a technique to deal with imbalanced data, it create a synthetic data by interpolating two minority class observations in the same neighborhood (of the feature space). This is done by randomly choosing one of the k-nearest neighbor minority class example of another minority class observation. Then the synthetic data is created in the feature space between these two data points (multiplied by a random value between 0 and 1). The objective of this technique its to make the minority class more representative in the training sets of supervised machine learning models

A standard scaler operation follows the SMOTE operation on the training folds. Both preprocessings can be implemented easily in each training fold with the help of a pipeline creation that evokes the preprocessing technique before running the training in each interation. If we preprocessed the data before splitting the folds, this techniques would fit to the validation set generating bias toward it. The pipeline utilized can be seen on Figure 3.

With the right parameters the SVM was retrained with 5 folds to calibrate an estimated probability for the data classification.

After the training we evaluated the test set and generated a confusion matrix.

We also tried one approach replacing SMOTE with class weights balanced by their frequency.

Only the linear kernel was tested for the SVM. Polynomials and RBF (*Radial Basis Function*) Kernels could yeld better results, but its easier to determine how each variable affects the prediction using the linear kernel. Its important to know how each variable contribute to the prediction to be able to determine which kind of actions have to be taken for each student.

All the experiments above were repeated replacing SMOTE with balanced class weights, by their ocurrences, as an alternative to deal with the imbalanced data.

IV. Results and Discussion

A. Using SMOTE and default Sklearn config for SVM

This model with the hiperameter C = 1[3] was used as a baseline model. The results can be seen on Table VI

The overall accuracy of this model was 70,12%.

B. Improving the model with SMOTE

Grindsearch was utilized to determine the better value for the hyperameter C of the linear SVM. The returned value was

Support Vector Machine ǀ C=1) ǀ SMOTE		
True Label	**Predicted Label = 0**	**Predicted Label = 1**
0	32	40
1	58	198

TABLE VI: Confusion Matrix of the SVM with SMOTE and C = 1

C = 0.01. We also utilized 5 folds to calibrate a percentual estimative of the student attrition. The values obtained with it can be see on Table VII

Support Vector Machine ǀ C=0.01 ǀ SMOTE		
True Label	**Predicted Label = 0**	**Predicted Label = 1**
0	31	41
1	55	201

TABLE VII: Confusion Matrix of the SVM with SMOTE and C = 0.01

The overall accuracy of this model was 70,73%.

Then we returned the calculated coefficients (for each attribute) that delimites the hyperplane that separates the class. The coefficients can be seen on Figure 4. The classification of the class is made considering the signal value of the coefficients multiplied by the input. With that in mind we can track which features are contributing to the evasion risk.

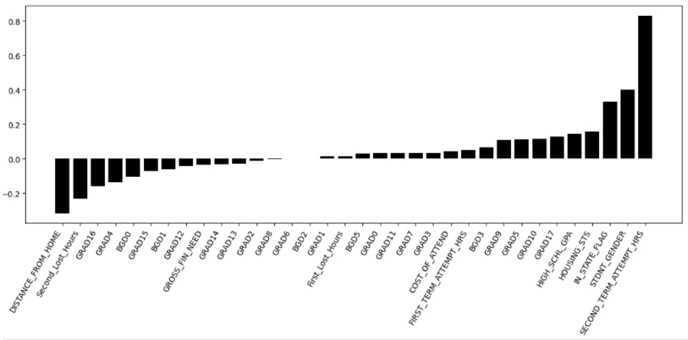

Fig. 4: Coefficients generated by the optimized SVM method using SMOTE.

C. Using SVM (C=1) with balanced class weights

This model with the hiperameter C = 1[3] was used as a baseline model. The results can be seen on Table VIII.

Support Vector Machine ǀ C=1 ǀ Balanced Weights)		
True Label	**Predicted Label = 0**	**Predicted Label = 1**
0	29	43
1	33	223

TABLE VIII: Confusion Matrix of the SVM with balanced weights and C = 1

The overall accuracy of this model was 76,82%.

D. Improving the model with balanced class weights

The experiments realized in Subsection IV-B were repeated for the balanced class weights. The results can be seen on Table IX and Figure 5. The overall accuracy of the obtained model was 78,65%.

Support Vector Machine ǀ C=0.01 ǀ Balanced Weights)		
True Label	**Predicted Label = 0**	**Predicted Label = 1**
0	29	43
1	27	229

TABLE IX: Confusion Matrix of the SVM with balanced weights and C = 1

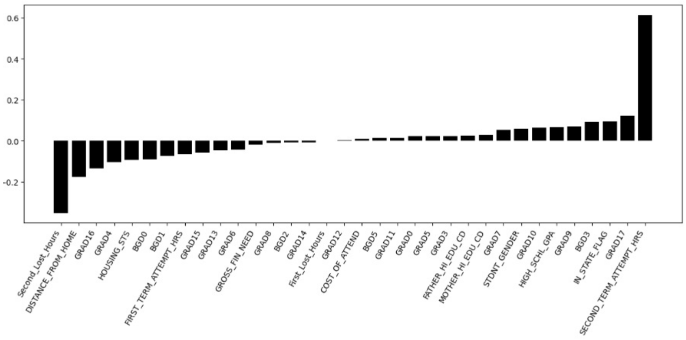

Fig. 5: Coefficients generated by the optimized SVM method using Balanced Weights.

V. Conclusion and future work

The results presented in this work shows that it is possible to identify students in risk of evasion which could enable earlier actions to reduce the number of students attrition rate. Althought the dataset should be improved to make a solid generalization.

The dataset contains 3400 students, 2677 that didnt evade until the 2nd semester and 723 that evaded in the first 2 semesters. One of the attributes in this dataset are the number of attempted credits and earned credits in the second semester. It contains 197 students with empty fields in those attributes, 172 of those students evaded the course after the second semester. Most likely those students already evaded after the first semester generating high bias toward those fields.

This can be seen on Figures 4-5. This probally means that our model isn't learning well to generalize and is highly biasing toward this attribute. This is a problem caused by the dataset.

A. Future Work

One approach to deal with this dataset could be to try to make one evasion predict for each semester. First predict if the student will evade at the first semester and then predict on the remaining students if they will evade at the second semester

Since we adressed the problem of universities in Brazil, the ideal dataset would be of brazilian universities. The students behavement and motivations could change between different cultures. The ideal dataset would also have dataset from different universities to avoid making the model representing bias of one specific university.

Since we can predict the probability of a student attrition we could save them in a database (including their ID) and query the ID of students with attrition percentage above one threshold.

Another improvement would be to multiply the input attributes by the SVM coefficients and then return the contribution of each attribute for the attrition rate. With that in mind would be possible to focus on specific solutions to each student.

References

[1] Evasão - dados brasil - 11º mapa do ensino superior. https://www.semesp.org.br/mapa/edicao-11/brasil/evasao/. Acessado em 15 de julho de 2023.

[2] Rio de janeiro - 11º mapa do ensino superior - instituto semesp. https://www.semesp.org.br/mapa/edicao-11/regioes/sudeste/rio-de-janeiro/. Acessado em 15 de julho de 2023.

[3] sklearn.svm.svc - scikit-learn 1.3.0 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. Acessado em 15 de julho de 2023.

[4] Yaser S Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning From Data*, volume 4. AMLBook, 2012.

[5] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In David Haussler, editor, *Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT'92)*, pages 144–152, Pittsburgh, PA, USA, July 1992. ACM Press.

[6] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 1 edition, 2000.

[7] Gerben Dekker, Mykola Pechenizkiy, and Jan Vleeshouwers. Predicting students drop out: A case study. pages 41–50, 01 2009.

[8] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh Chawla. Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, 61:863–905, 04 2018.

[9] Saurabh Pal. Mining educational data to reduce dropout rates of engineering students. *International Journal of Information Engineering and Electronic Business*, 4, 04 2012.

[10] Mingjie Tan and Peiji Shao. Prediction of student dropout in e-learning program through the use of machine learning method. *International Journal of Emerging Technologies in Learning (iJET)*, 10(1):pp. 11–17, Feb. 2015.