PREDICTING DROPOUTS USING MACHINE LEARNING MODELS

Viksar Dubey
San Jose, CA, United States of America

ABSTRACT

High dropout rates in high schools and universities have become a major complication in many countries following the upsurge of the Covid 19 pandemic. A student's experience at school is one of, if not, the largest contributors to the likelihood that they drop out. However, with the growing quantity of students in school, and the sheer variety of those who drop out, manually pinpointing possible dropouts becomes extremely challenging. Therefore, by using a machine learning model, administrators would be able to more efficiently identify a possible dropout and provide the necessary resources to ensure their success. The data, acquired from the UCI Machine Learning Depository, had 24 inputs, and 3630 samples. For this particular project, Logistic Regression, K-Nearest Neighbors, Naive Bayes, Extra Trees Classifier, and an MLP Neural Network models were tested. Among these models, the most accurate was the Extra Trees Classifier model, with a percentage accuracy of 77.41%, a loss of 7.85, and an F1 score of 82.35%. With the accuracy provided, this model should be held in high regard as it serves as an efficient method for predicting possible college dropouts in university.

INTRODUCTION

For the concision of this paper, I will be using the term "dropout" to characterize students who stop attending school before completion, and in no way will and should this term be used in a derogatory manner.

A student's decision to drop out greatly affects the student themselves, the institution, and the economy. Studies show that a student who drops out of school normally has much more trouble finding a job, and even when doing so, college dropouts are commonly paid less than $30,000 compared to what graduates are paid. Not only do they face more adversity whilst searching for work, but tend to be more involved in criminal activities. Based on a study in 2018, the unemployment rate for student dropouts was 18.6% compared to a measly sub-4% unemployment rate for all individuals eligible for the workforce that year. Student dropouts also vastly affect an institution's success. Universities lose an estimated combined 3.8 billion dollars annually due to students dropping out.

So then the question arises, why even drop out? Well, there are many reasons why a student drops out. The Colorado Board of Education defines four broad classes of dropouts, three of which, "fade outs," "push outs," and "failing to succeed," are directly linked to their experience at school. "Fade out" students are those who typically do well at school, but soon become frustrated or bored with the curriculum and drop out. "Push out" students are those who are perceived to be dangerous and difficult. "Failing to succeed" students are those who attend schools that don't accompany them with the right environment and recognition to succeed. With the ability to identify possible dropouts, more attention can be given to these students by professors and counselors earlier, and new regulations could be devised to ensure the completion, and furthermore, the success of students' educational careers.

So how can machine learning be used to assess this problem? Machine learning can be defined as the ability of computers to make conclusions based on data, without the use of specific, man-made instructions or restrictions, and instead through training and repetition. When given enough data on existing dropouts or graduates and their various attributes, the machine learning models tested were able to make predictions on students who haven't dropped out or graduated yet.

To approach the binary classification problem of predicting possible student dropouts, six different artificial intelligence models were tested, including Logistic Regression, K-Nearest Neighbors, Naive Bayes, Extra Trees Classifier, and an MLP Neural Network. The model with the highest accuracy, highest F1 score, and lowest loss was considered the most effective in predicting students who drop out and students who graduate.

BACKGROUND

Previous studies focusing on academic improvement include the following: predicting a student's numeric academic output, designing intelligent tutoring services, designing student-specific lesson plans, and more. However, only a couple of years ago has the broad objective of ameliorating a student's performance been narrowed down to the most basic measure of their success in school, graduation.

As of now in most schooling systems, we are forced to find patterns and probabilities among input features such as grades and attendance. However, this methodology becomes very insufficient as the number of possible input features and biases change. For example, just skimming over a student's grades and attendance may give one a general understanding of if a student may drop out or not. However, the dataset used has many cases of students having excellent test scores and still dropping out. Finding connections between multiple features is what guarantees a higher accuracy and lower loss, and can mainly only be achieved using machine learning models.

A machine learning based study (Tan & Shao, 2015), exploring possible models that could predict student dropouts in a online learning environment, obtained results that stated that Decision Trees were the most effective model out of Desicion Trees, Naive Bayes, and a Artifical Neural Network. Another study (Aulck, Velagapudi, Blumenstock, & West, 2016) obtained results proving that in the dataset provided, the Logistic Regression model was the most effective compared to Random Forest Classifier and K-Nearest Neighbors.

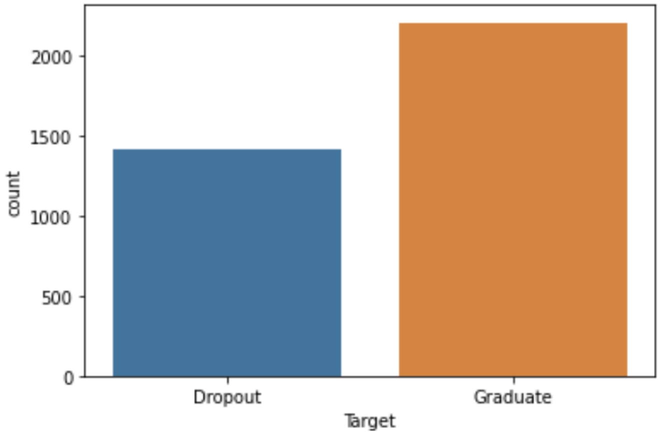
The primary models tested for this project are defined and visualized in the "Models/Methodology" section of this paper.

DATASET

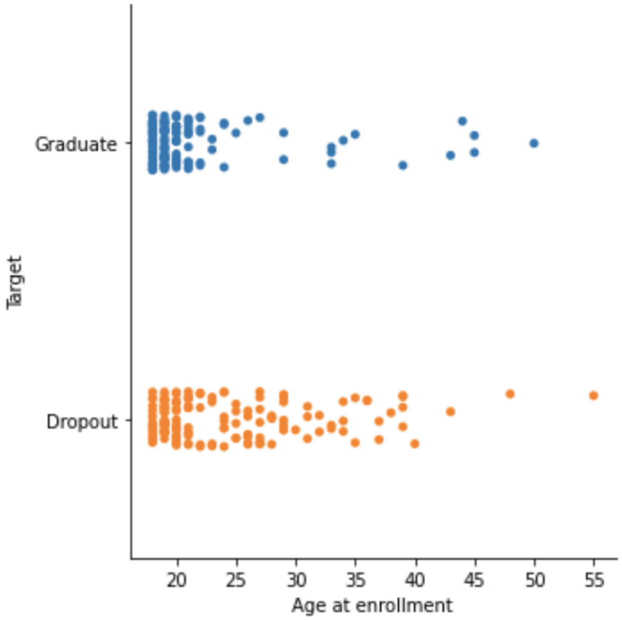
The raw dataset was found in the UCI Machine Learning Depository and was acquired from several disjoint databases. Its collection was supported by the SATDAP - Capacitação da Administração Pública under grant POCI-05-5762-FSE-000191, Portugal [Translated to SATDAP - Capacity Building of Public Administration under grant POCI-05-5762-FSE-000191, Portugal].

The original dataset contained 4424 different samples, which held 36 different features that would be specific to the student. Initially, three different labels were to be predicted: "Dropout", "Enrolled", and "Graduate." However, as more data preprocessing was performed, the "Enrolled" label was dropped. A currently enrolled student would still have the chance of either dropping out or graduating and so its maintenance would only distort appropriate results. Additionally, the 36 different input features were condensed into just 24. Among those dropped were "Curricular units 1st sem (approved)" and "Curricular units 2nd sem (approved)." The objective of this project was for models to predict based on factors defined before the student's collegiate career, and the factors presented were dependent on the student's performance in the first and second semesters during their collegiate careers. After dropping these features, the accuracy score dropped a staggering 16% from about 93% accuracy to 77% accuracy. However, this procedure was necessary to develop user confidence. The final dataset contained 3630 different student samples with 24 features. Within these samples, 2209 were graduates and 1421 were dropouts.

(Figure 1)

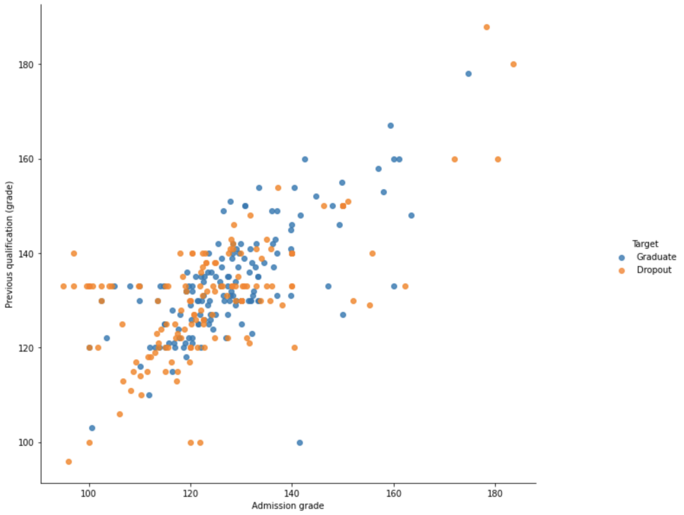


(Figure 2)



Some of the 24 input features affected the final prediction more than others. For example, “Age at enrollment”, “Admission grade”, and “Previous qualification (grade)” had significantly different ranges of values between graduates and dropouts. Data in the following plots has been reduced to display 150 graduates and 150 dropouts and their respective inputs to help evenly represent both labels.

(Figure 3)



As shown by Figure 2, many more dropouts are spread throughout the age range of 15 years to 40 years, while graduates are denser within the 15 years to 25 years range. The mean age at enrollment for graduates was 21.78, while the mean age at enrollment for graduates was 26.07. In Figure 3, many more graduates (represented in blue) are spread towards the top right of the graph, exhibiting higher admission scores and previous scores, while dropouts are placed closer to the bottom left. On average, the admission grade for graduates was 128.79, while the admission grade for dropouts was 124.96. The average previous qualification (grade) for graduates was 134.08, while the previous qualification (grade) for dropouts was 131.11.

Data was split on a 4 to 1 ratio, with the training size being 80% of the total dataset and the testing size being, therefore, 20%, and data was scaled to proportion using StandardScaler by sklearn.

MODELS/METHODOLOGY

For the dataset used, two labels were to be output based on the sample, resulting in a binary classification problem, in which many models, such as Logistic Regression, Random Forest, K-Nearest Neighbors, Naive Bayes, Extra Trees Classifier, and an MLP Neural Network models were fit to be tested.

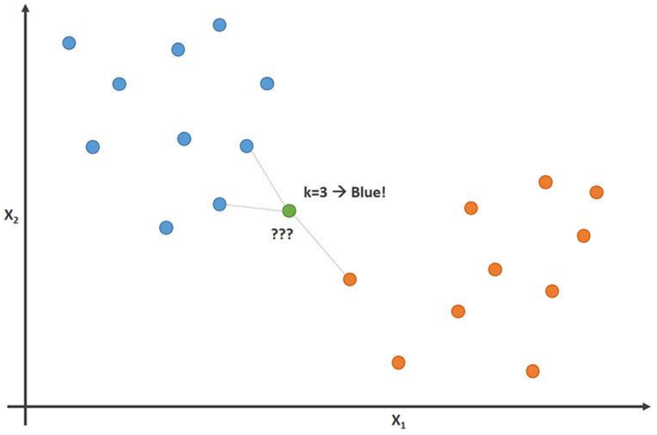
A. *Classification using Logistic Regression*

A Logistic Regression model is a classification model that outputs a label based on the probability that the predicted label is equivalent to the true label. This probability is derived from a weighted sum of input features and a bias term. Logistic Regression models output a 1 if the probability is above 0.5, or a 0 if the probability is below 0.5. Variables, such as “Age at enrollment”, “Admission grade”, and “Previous qualification (grade)”, would likely have a greater effect on the probability, and therefore label, output by a logistic regression model.

B. *Classification using K-Nearest Neighbors*

K-Nearest Neighbors, also known as KNN, is a memory-based learning method that relies on different points within a dataset to make predictions on other points within a dataset. For example, if K in K-Nearest Neighbors is set to 3, the model takes a look at the 3 closest points with respect to the unknown point and makes a prediction based on this. For this project, K was set to 9, as it resulted in the highest accuracy possible.

(Figure 4)



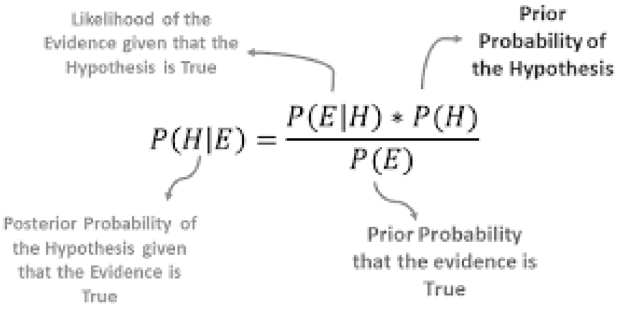
Retrieved from rapidminer.com

However, because this model is memory-based, and therefore, stores a training set rather than going through a training stage, its accuracy and efficiency diminish as a dataset gains in size. The dataset used in this project was not relatively large, however, this setback did make a difference in the accuracy and F1 score.

C. *Classification using Naive Bayes*

Naive Bayes is a machine learning algorithm that utilizes Baye's Theorem (as shown below) to make independent assumptions on input features.Naive Bayes Classifier is used mostly to make predictions on smaller datasets with fewer features. For this particular project, "verbose" was set to 1, "cv" was set to 10, "n_jobs" was set to -1, and other parameters remained default.

(Figure 5)

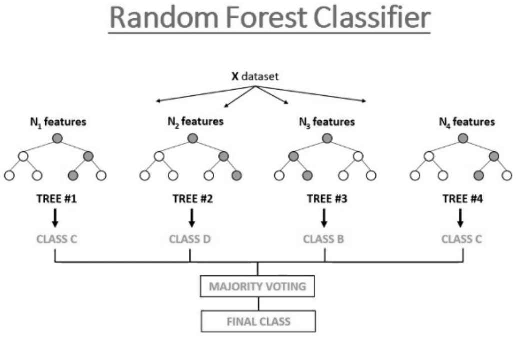

Likelihood of the Evidence given that the Hypothesis is True
Prior Probability of the Hypothesis
P(H|E) = P(E|H) * P(H) / P(E)
Posterior Probability of the Hypothesis given that the Evidence is True
Prior Probability that the evidence is True

Retrieved from medium.com

D. *Classification using Extra Trees Classifier*

To first utilize the Extra Trees Classifier, one would need to understand the Random Forest Classifier. The Random Forest Classifier model is defined by a large group of Decision Trees, which are trained on the bagging method. The bagging method is used to reduce volatility within a dataset that contains a large range of values. Random Forest models generally search for the best feature among a subset of features, rather than a decision tree, which searches for the best feature in nodes.

(Figure 6)



Retrieved from freecodecamp.com

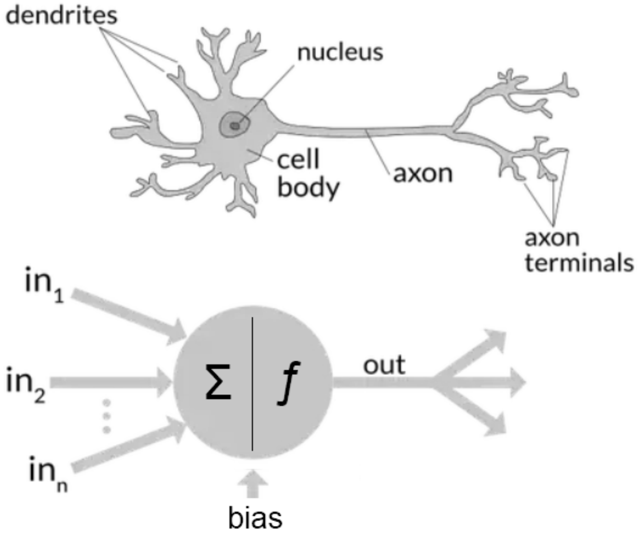
Random Forest Classifiers are beneficial to use because they handle large groups of data well, and are able to make classification decisions based on a majority vote within the trees. For this particular project, however, the Random Forest Classifier was not used as the evaluation metrics used could not properly analyze the Random Forest Classifier's performance.

The Extra Trees Classifier is very similar to the Random Forest Classifier, the only major difference being that the Extra Trees Classifier is practically a more randomized form of the Random Trees Classifier. Rather than searching for the best possible threshold for a feature, as done in a Random Forest Classifier, this threshold is randomized. This quality of Extra Trees Classifier allows it to be much quicker to run, therefore allowing more instances to be completed in a certain amount of time, when, for example, fine-tuning. For this particular project, "max_depth" was set to 20, "max_features" was set to "None", "min_samples_leaf" was set to 10, "min_samples_split" was set to 20, "n_estimators" was set to 6387, and other parameters remained default.

E. *Classification using Neural Networks and MLP*

A neural network is a machine learning model that mirrors the process of retrieving and making predictions off of data within the human brain.

(Figure 7)



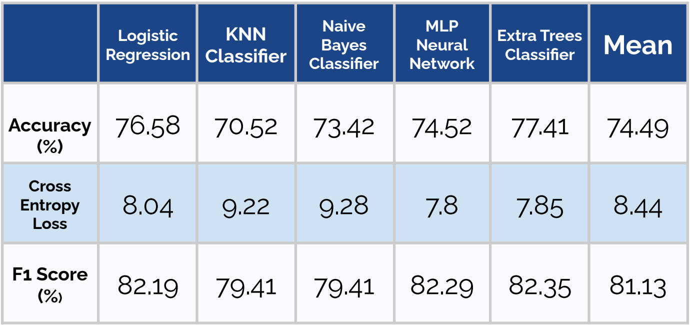
Retrieved from towardsdatascience.com

An MLP, also known as the Multilayer Perceptron, Classifier is a type of artificial neural network, composed of an input layer, multiple hidden layers, and an output layer. Within each layer is a bias, which may impact the final prediction more than other biases. Contrary to K-Nearest Neighbors, an MLP Classifier works best with larger datasets and favors accuracy. For this particular project, “hidden_layer_sizes” was set to (7,3), “random_state” was set to 1, “max_iter” was set to 10000000, and other parameters remained default.

All six models were tested regardless of their each own preferable scenarios or datasets. Each was fine-tuned appropriately and was compared based on accuracy score, cross-entropy loss, and F1 score. Accuracy score, as stated by the metric’s name, just displays the model's accuracy, based on how many times it correctly predicted either “Graduate” or “Dropout.” Cross-entropy loss represents the total sum of the false prediction possibilities of each student tested. The F1 score is the mean between the precision and recall values. These metrics alone provide a reasonable standard for effectiveness in a model.

RESULTS AND DISCUSSION

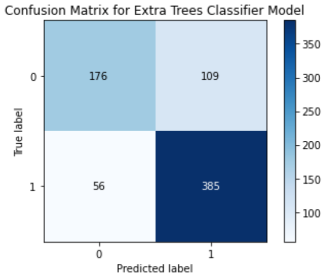
(Figure 8)



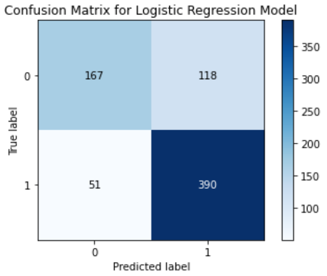
	Logistic Regression	KNN Classifier	Naive Bayes Classifier	MLP Neural Network	Extra Trees Classifier	Mean
Accuracy (%)	76.58	70.52	73.42	74.52	77.41	74.49
Cross Entropy Loss	8.04	9.22	9.28	7.8	7.85	8.44
F1 Score (%)	82.19	79.41	79.41	82.29	82.35	81.13

As seen in the table above, the Extra Trees Classifier model had an accuracy score of 77.41% above the mean accuracy of 74.49%. Its loss was 7.85, coming second to the MLP Neural Network model, and 0.59 below the mean. Finally, the model had the highest F1 score at 82.35%, compared to a mean of 81.13%.

(Figure 9)



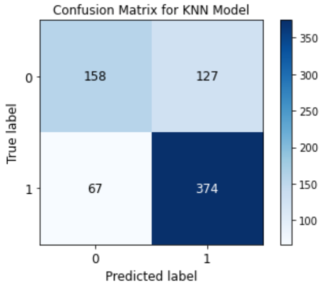
(Figure 10)



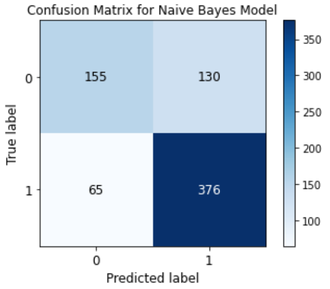
Dropout numerically represented by 0, Graduate numerically represented by 1

The Logistic Regression model was among the most successful of the models, receiving the second-highest accuracy score of 76.58%, a loss of 8.04, and an F1 score of 82.19%. The KNN model, on the other hand, did not return, receiving the second-highest accuracy score of 76.58%, a loss of 8.04, and an F1 score of 82.19%.

(Figure 11)

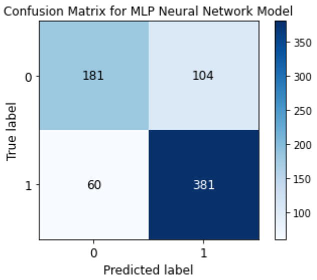


(Figure 12)



The Naive Bayes Classifier, similar to the KNN model, was not very effective compared to the mean values. It received an accuracy of 73.42%, a loss of 9.28, and an F1 score nearly identical to the KNN model, with a 79.41%. Lastly, the Neural Network model was slightly above the mean with an accuracy of 74.52%. However, its loss was actually 0.05 fewer than the loss of the Extra Trees Classifier, and its F1 score was only 0.06% less than the Extra Trees Classifier, making this model the second most effective in predicting possible dropouts.

(Figure 13)



The Extra Trees Classifier model likely worked so much more efficiently than the other models because of its ability to fit complex datasets. Both performances by the KNN model and the Naive Bayes model were expected as both models do not do well as dataset size increases. Though the dataset used to test the models wasn't relatively large, it definitely made an impact on the accuracy, loss, and F1 score of either model. For KNN models specifically, each individual point has to be computed using the nearest neighbors, leading to increased computing time and less efficiency. Naive Bayes models make predictions based on strong assumptions that wear down efficiency as data volume increases. The Logistic Regression model worked as expected as well as the Neural Network model.

Potential sources of error could've arisen through the possibility that the parameters used in the models were not fine-tuned well enough when testing. However, GridSearchCV was used to fine-tune the models, so this potential error is not very likely. Still, the models tested may not be complex enough to understand finer patterns and connections within this dataset. Another possibility, as well as a limitation, is that the dataset, which was retrieved from only one database, may have biases towards certain models.

CONCLUSION

Throughout the experimental process, the Extra Trees Classifier acquired the best scores based on the evaluation metrics tested with a 77.41% accuracy, 7.85 cross-entropy loss, and an F1 score of 82.35%. The KNN Classifier had the worst performance by a large margin, with an accuracy of 70.52%. To summarize, the order of most effective to least effective models goes as such: Extra Trees Classifier, MLP Neural Network, Logistic Regression, Naive Bayes, and KNN. The Extra Trees Classifier model, in regard to its evaluation metrics, can be considered a method effective in predicting probable dropouts. However, there is still lots of room for improvement, and the usage of advanced models could lead to increased accuracy and lowered losses.

As already mentioned before, each model's accuracy increased by around 15% to 20% when keeping inputs such as "Curricular units 1st sem (approved)" and "Curricular units 2nd sem (approved)" in the dataset, but to maintain trust and achieve realistic results, these inputs were not taken into account. Possible extensions could include using advanced models, such as models imported from *huggingface.co* which would be fine-tuned to fit the specific dataset. Further ideas could be finding the specific effect of certain input features on the likelihood that a student drops out or a model that returns a numeric value in correspondence to a student's academic success or lack of it.

ACKNOWLEDGEMENTS

I would like to acknowledge Eric Bradford and the InspiritAI Organization for providing lots of guidance in performing research and writing this paper. I would also like to thank the UCI Machine Learning Depository for providing me with the dataset used. I would also like to thank the creators of all the various references used for providing this paper with images and setting the basis of knowledge.

REFERENCES

V. Realinho, J. Machado, L. Baptista, M. V. Martins. (2021). "Predict students' dropout and academic success" (1.0) [Data set]. Zenodo. DOI: 10.5281/zenodo.5777340

M. V. Martins, D. Tolledo, J. Machado, L. M. T. Baptista, V. Realinho. (2021) "Early prediction of student's performance in higher education: a case study" Trends and Applications in Information Systems and Technologies, vol.1, in Advances in Intelligent Systems and Computing series. Springer. DOI: 10.1007/978-3-030-72657-7_16

Géron Aurélien, *Hands-on machine learning with scikit-learn, Keras, and tensorflow concepts, tools, and techniques to build Intelligent Systems*. Beijing: O'Reilly, 2020.

I. Mierswa, "K-nearest neighbors: A simple machine learning algorithm," *RapidMiner*, 05-May-2022. [Online]. Available: https://rapidminer.com/blog/k-nearest-neighbors-laziest-machine-learning-technique/. [Accessed: 30-Oct-2022].

D. David, "Random Forest classifier tutorial: How to use tree-based algorithms for machine learning," *freeCodeCamp.org*, 13-Aug-2020. [Online]. Available: https://www.freecodecamp.org/news/how-to-use-the-tree-based-algorithm-for-machine-learning/. [Accessed: 30-Oct-2022].

B. Gamal, "Naïve Bayes algorithm," *Medium*, 11-Apr-2021. [Online]. Available: https://medium.com/analytics-vidhya/na%C3%AFve-bayes-algorithm-5bf31e9032a2. [Accessed: 05-Nov-2022].

"The differences between artificial and Biological Neural Networks." [Online]. Available: https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7. [Accessed: 30-Oct-2022].

Tan & Shao, 2015, Mingjie Tan, Peiji Shao
Prediction of Student Dropout in E-Learning Program Through the Use of Machine Learning Method
2015

Aulck, Velagapudi, Blumenstock, & West, 2016, Lovenoor Aulck, Nishant Velagapudi, Joshua Blumenstock, Jevin West

Predicting Student Dropout in Higher Education

2016