# Predicting Course Enrollment with Machine Learning and Neural Networks: A Comparative Study of Algorithms

**Bahgat Ayasi**

Computer Science Department, University of Jaén, Spain, ba000034@red.ujaen.es, https://orcid.org/0000-0003-3430-9260

**Mohammed Saleh**

Information Technology Center, Al-Istiqlal University, Palestine, msaleh@pass.ps, https://orcid.org/0009-0008-7254-8071

**Ángel M. García-Vico**

Dept. of Communication and Control Systems, UNED, Spain, agvico@ujaen.es, https://orcid.org/0000-0003-1583-2128

**Cristóbal Carmona**

Computer Science Department, University of Jaén, Spain, ccarmona@ujaen.es, https://orcid.org/0000-0001-7008-5378

**Abstract:** The digitization and collection of big data by higher education institutions can help administrators make informed decisions about resource allocation, specifically in enrollment management. This study explores the use of machine learning and neural network algorithms to predict future student enrollments in courses. Real data from the Arab American University in Palestine (AAUP) was used. Eight machine learning algorithms, in addition to the Multilayer Perceptron (MLP) neural network model, were used to predict which students are most likely to enroll in a specific course. Results show that ensemble-based and bagging algorithms outperform other classifiers, including neural network models, for individual-level prediction of student enrollments. The Random Forest algorithm achieves the highest accuracy of 94% and an F1 score of 79% after applying under-sampling techniques on the highly imbalanced dataset. The study recommends future research to develop a generalized model for predicting enrollment in any course at AAUP and highlights the effectiveness of these techniques for improving resource allocation and student support.

**Keywords:** classifiers, imbalanced dataset, hyperparameters tuning, F1 score, enrollment, higher education

## Introduction

Course enrollment predictive techniques using machine learning and neural networks can predict the number of students who will enroll in a course or program. These techniques help educational institutions and businesses make informed decisions

about resource allocation, staffing, and capacity planning. Machine learning and neural networks identify patterns and relationships based on academic historical data. These techniques optimize operations and better meet the needs of students.
The current enrollment prediction methodology at Arab American University in Palestine (AAUP) involves a manual model based on statistical methods. Academic department heads use a Cohort survival model to forecast student enrollment based on historical data from the past two or three academic years. The boundary level is set at 10% plus or minus the average rate of registration in the same course in the same semester. Additionally, there is a traditional programming application based on fixed logic used to expect the number of students at the course level.

The current enrollment prediction tool performs well for courses with specific conditions but is poor for free elective and general education courses due to unexpected student numbers and transfers/withdrawals between majors. The tool requires manual tuning and updates and gives different results at different times of the year. This calls for unconventional methods and techniques for courses like “Palestinian Studies” (PS), which have no fixed prerequisites and face uncertainty in enrollment numbers every academic semester.

The study aims to investigate the effectiveness of different machine learning and neural network techniques for predicting course enrollment at the individual level for all new intakes and continuing students in AAUP. Real data from students who have already registered from 2018 to 2022 will be used as a case study. AAUP is the first private university in Palestine that offers academic services for both national and international students in various disciplines and degree levels. AAUP represents a diverse range of national and regional universities and higher education institutes, making the findings of this research widely applicable to other similar institutions.

The objective of this research is to predict individual-level enrollment in a particular course using machine learning and Neural network techniques. The study uses real data from AAUP and explores the effectiveness of eight different classification models and a multilayer perceptron (MLP) neural network model for individual enrollment prediction. Specifically, we use the logistic regression algorithm (LR), Stochastic Gradient algorithm (SGD), K-Nearest Neighbors algorithm (KNN), Decision Tree algorithm (CART), Gradient Boosting algorithm (GB), Bagging Classifier (BG), Support Vector Machine algorithm (SVM), Random Forest algorithm (RF), and the multilayer perceptron (MLP) Neural network algorithm to predict individual course enrollment using students' historical academic data extracted from AAUP's student information registration database.
Because of the imbalanced nature of the dataset, we implement under-sampling techniques to address imbalanced datasets, which reduces the skew from a 10:1 to a 3.5:1 for the majority negative class 0 to the minority positive class 1. We evaluate the performance of our model using the F1 score with recall, precision, and accuracy metrics.

Our study builds upon previous work in the field of enrollment management. For example, (Shao et al., 2022) modeled course enrollment prediction by applying two tree-based algorithms, while (Esquivel & Esquivel, 2020) used logistic regression and support vector machines were used to predict enrollment at the individual and cohort level for applicants admitted to the University of New Mexico (UNM). Additionally, (Kardan et al., 2013) a neural network was used for course selection. Another study by (Doleck et al., 2020) conducted a comparative analysis of deep learning frameworks for predicting student performance in an undergraduate course. The authors use four deep learning frameworks (convolutional neural network, recurrent neural network, long short-term memory, and deep belief network) to predict student performance. The results showed that all four deep learning frameworks outperformed a traditional machine learning algorithm (logistic regression) in predicting student grades. The deep belief network had the highest accuracy among the four deep learning models tested.

Overall, the paper provides a useful overview of the current state of predictive analytics in education and highlights the potential of deep learning for improving student outcomes.

Furthermore, Wanjau et al. (2016) aimed to develop a data mining model to predict student enrollment in STEM courses in higher education institutions. The author emphasized the importance of feature selection and data preprocessing in improving the accuracy of predictive models. To develop the data mining model, the authors collected data on student demographics, academic performance, and course history from a public university in Kenya. The author used Decision Tree (CART), Naïve Bayes, k – Nearest Neighbor, Artificial Neural Networks (Multilayer Perceptron), Support Vector Machine (SMO), and Logistic Regression algorithms to predict student enrollment in STEM courses for the next academic year. The results showed that the decision tree algorithm (CART) had the highest accuracy in predicting student enrollment, with an overall percentage of correct classification of 85.2%
To address the imbalanced nature of the dataset, we employ under-sampling techniques and evaluate the performance of various classification algorithms, including linear classifiers and tree-based and bagging classifiers, and MLP algorithms. The F1 score with recall, precision, and accuracy metrics is used to evaluate the model's performance, especially on the minority positive class 1. We found that the literature is lacking in related research on predicting enrollment in a particular course, whereas there is much study on general enrollment at the institutional level.

Our study demonstrates that we can accurately predict course enrollment at the individual level using machine learning and neural network techniques. By utilizing these techniques, we were able to develop models that can effectively predict course enrollment with high precision and recall metrics. Furthermore, by implementing various data preprocessing techniques such as under sampling using the imbalanced-learn Python library, we were able to address the issue of imbalanced datasets and improve the F1 score, which is a more robust metric compared to accuracy when dealing with imbalanced datasets. Our study's contributions to the literature on imbalanced datasets and classification problems are noteworthy as it provides valuable insights into the development of effective models for predicting course enrollment in higher education institutions. The insights gained from our study have practical implications for academic and administrative decision-makers in higher education institutions. Finally, the development of a generalized model for all university-offered courses is a significant challenge for higher education institutions. Our study's success in developing accurate models for predicting course enrollment can pave the way for the development of generalized models for all university-offered courses.

The remainder of this paper is organized as follows: Section 2 gives the related works, Section 3 describes the Research Methodology, Section 4 Experimental Studies, and Section 5 discusses the results. We conclude in Section 6.

## Related Works

There is a lack of research in predicting student enrollment at an individual level using machine learning and neural networks. However, there have been several studies that have explored the use of machine learning and neural network algorithms to predict student enrollment at an institutional level based on historical admission applications and student academic characteristic features. We present here most of these studies that demonstrate the potential of machine learning and neural network algorithms in predicting student enrollment at an individual level.

The study (Shao et al., 2022) focused on using tree-based models, namely CART and random forest, to predict course enrollment at San Diego State University. The study used historical data in the form of student demographic information and academic performance. The authors also determined which factors were the most influential in predicting enrollment using a variable importance metric derived from tree-based algorithms. The authors found that the proposed decision tree approach was able to improve upon the current state-of-the-art conditional probability analysis slightly, and the proposed random forest model was able to further improve upon both methods. One limitation of the study is that it did not address the issue of imbalanced datasets, which is a common challenge in predicting course enrollment. Imbalanced datasets can result in biased models and misclassification errors. Furthermore, the study did not discuss the effect of hyperparameter tuning on the proposed models. Hyperparameter tuning is a critical step in model development as it can significantly affect the model's performance. In our study, we address these limitations.

Slim et al. (2018) focused on using a Support Vector Machine (SVM) algorithm and logistic regression with a classification approach to predict the enrollment of applicants at the University of New Mexico (UNM). The study found that a small set of factors related to student and college characteristics were highly correlated to the applicant's decision to enroll. The study also used the confusion matrix precision and recall metrics in addition to accuracy to test the performance of a classifier. The results showed that the SVM algorithm outperformed logistic regression, achieving an accuracy of 91.25%.

The neural network approach used in the study by Kardan et al. (2013) was able to predict class enrollment in online courses with high accuracy based on the experimental data. The study focused on identifying potential factors that affect student satisfaction with the online courses they select. The samples collected for this research included 714 courses over 16 academic terms. The findings revealed that the proposed model outperformed three well-known machine learning techniques. However, the study did not provide information about the specific neural network architecture used or the hyperparameter tuning process.

(Mia* et al., 2019) applied seven popular classifiers on a data set of more than a thousand students of a private university in Bangladesh. The classifiers used were SVM, Naive Bayes, Logistic, JRip, J48, Multilayer Perceptron, and Random Forest. Each record in the dataset contains five attributes. The authors computed six performance metrics, including accuracy, sensitivity, specificity, precision, false positive rate (FPR), and false negative rate (FNR), for each of the seven classifiers. The results indicate that SVM outperforms the other models in terms of prediction accuracy, achieving 85.76% accuracy. Random Forest achieved the lowest accuracy at 79.65%.

The researcher Shilbayeh and Abonamah (2021) also focused on predicting student enrollment behavior and identifying students who are at risk of dropping out using machine learning approaches. The study used data from the Abu Dhabi School of Management (ADSM) and developed a student enrollment model using Boosted regression trees. The model was able to identify student characteristics that influence enrollment decisions and those who are at risk of dropping out. The boosted regression model was tested using 10-fold cross-validation and achieved an accuracy of 89%, outperforming a single regression decision tree that achieved only 76% accuracy using the same validation method.

The researcher Saini and Jain (2013) used decision tree algorithms like ID3 and J48 for student enrollment in a specific stream, such as MCA. The study focused on using past academic performance as a predictor for enrollment suitability, which could help in identifying students who may not perform well in MCA. Also, the study reported the accuracy of the models, which can be useful for future research and implementation.

Another study by Akinode and Bada (2021) used a survey research approach to analyze factors influencing student enrollment in a Federal Polytechnic in South-West Nigeria. The dataset consisted of 560 students enrolled in various courses from 2017 to

2018. The study employed machine learning methods such as the decision tree algorithm (ID3) and support vector machine (SVM) to analyze the correlation of different factors on student enrollment. The results showed that the ID3 algorithm outperformed other ML algorithms, such as Artificial Neural Network and Logistic Regression, with the highest accuracy of 97%, while SVM, KNN, and Naïve Bayes had an accuracy of 95%, 85%, and 88%, respectively. In our study, we will also use these algorithms in addition to other boosting and ensemble algorithms to compare their performance in predicting student enrollment.

Esquivel and Esquivel (2020) a Logistic Regression model was developed to predict the likelihood of an admitted student enrolling in a Philippine university. The dataset was obtained from the university admissions office, and a descriptive research design was used to mine students and the university's characteristics. The study aimed to identify the relationships between the features or variables and the dependent variable of whether the student will enroll or not. Logistic Regression, a binary classification algorithm, was implemented to predict the enrollment decision of a student. The model achieved an accuracy rate of 80% using the selected attributes, indicating that machine learning techniques can aid Higher Education Institutions in making management decisions and providing estimates of class sizes with limited information about prospective students.

In summary, machine learning and neural network algorithms have shown promise in predicting class enrollment at the individual level. Additionally, under-sampling methods have been employed to address the issue of highly imbalanced datasets. Evaluation metrics such as accuracy, F1 score, precision, and Recall have been used to assess the effectiveness of these predictive models.

## Research Methodology

This research focuses on predicting future student enrollment in a particular course based on historical academic data in the context of AAUP. To achieve this goal, we used eight binary classification machine learning algorithms in addition to a multilayer perceptron MLP neural networks algorithm. The objective was to predict which students would enroll in one of the high-demand mandatory courses named Palestinian Studies (PS) in the next semester. The models were compared based on accuracy, precision, recall, and F1 score metrics.

### Data Collection

The AAUP student registration database provided the real data for this research, which was organized into a dataset. It contained individual registration information for nearly 9,000 undergraduate students enrolled in the high-demand mandatory course (PS), between Fall 2018 and Summer 2021. It included around 137,000 student registration records and was used for classification algorithms to predict the likelihood of students enrolling in PS in the upcoming semester based on historical registration data.

### Dataset Description

The dataset contains nine independent features and labels (dependent variable) that will be utilized for forecasting course enrollment at the student registration level. These features as shown in Table 1 include student characteristics, while the dependent feature indicates the enrollment status of the student in the PS course per student semester level.

Table 1 Student Registration Data

| Feature | Description |
|---|---|
| **STUDENT_ID** | A numeric variable that represents a pseudo-unique ID for each student. |
| **STUDENT_STATUS** | A categorical variable that indicates the student's academic status at the specified semester, one of 11 statuses (e.g., regular, dismissed, discontinued, graduate, etc..). |
| **SEMESTER** | A numeric variable that indicates the academic semester the student registered for the PS course |
| **STUDENT_LEVEL** | A categorical variable that indicates the student's academic level at the specified semester. |
| **GENDER** | A binary variable that indicates student gender possible outcomes Male or female |
| **FACULTY_NO** | A categorical variable that indicates the student’s college that semester. |
| **MAJOR_NO** | A categorical variable that indicates the student's major at that semester. |
| **Cum_GPA** | A numeric variable that represents the student's GPA for that semester. |
| **EARNED_HRS** | A numeric variable that represents the number of successfully earned credit hours up to that semester |
| **LABEL** | A binary variable indicating the student registration status of the course encoded as two classes, 0 for the negative class (not enrolled in the course) and 1 for the positive class (enrolled in the course). |

The categorical variables, such as STUDENT_STATUS, STUDENT_LEVEL, and GENDER were converted into numerical continuous variables using a one-hot encoding technique.

**Dataset preprocessing**

In a binary classification problem, the targets or labels in the dataset are binary random variables that take on values of either 0 or 1. The Bernoulli distribution is a common way to model this type of problem, where the model predicts a probability distribution for each example indicating the likelihood of it belonging to class 1 “enrolled” or class 0 “not enrolled”. To train a machine learning model on this data, we typically need to encode the class labels as numeric values. One common approach is to use label encoding, where we map the two class labels to 0 and 1, respectively, before providing them to the algorithm for modeling. This is a type of ordinal encoding, and sci-kit-learn provides the LabelEncoder class specifically for this purpose, in our example, the two class labels are "enrolled" and "not enrolled", which need to be mapped to binary values of 1 and 0, respectively. Once this encoding is done, we can use the data to train a binary classification model that predicts the likelihood of a student enrolling in the PS course in the specified semester. after the encoding, several preprocessing steps have been made, such as:

1-Random under-sampling is a technique used to resample imbalanced datasets where the number of examples in the minority class is much smaller than that in the majority class. In this technique, we randomly select examples from the majority class and delete them from the training dataset until a more balanced distribution is achieved. The imbalanced-learn Python library provides an implementation of the RandomUnderSampler class for performing random undersampling. This class can be used to randomly remove examples from the majority class until the desired balance between the two classes is achieved.

The dataset we are working with exhibits severe class imbalance, with a ratio of approximately 10:1 To address this issue, we conducted numerous experiments with oversampling and under-sampling techniques. Using accuracy, F1, and cross-validation metrics to evaluate model performance, we determined that under-sampling was more effective for our dataset. Undersampling mitigated the risk of overfitting and was preferable since the majority class contained a substantial number of instances, and removing a portion of them would not compromise data quality or Model performance. We selected the random under-sampling method provided by the imbalanced-learn Python library resulting in a ratio of approximately 3.5 to 1 for class 0 to class 1. This technique helped to mitigate bias towards the majority class and enhance the overall performance of the model.

2- To ensure the performance of our model is not affected, we identify and remove outliers using an automatic outlier detection method called the Local-Outlier-Factor model, which is available in the Python sci-kit-learn library. We prefer this method over statistical techniques because of the complexity and unknown inter-relationships between input variables. After applying this approach, our dataset is moderately imbalanced, with a total of 50,476 records. Class 1 has 11,128 records, while class 0 has 39,348 records, resulting in a ratio of approximately 3.5 to 1 for class 0 to class 1.

3- After evaluating the impact of missing data on the analysis, where the dataset is relatively large, we have determined that removing about 2000 records with missing values is the best approach.

4- There is a significant difference between the data points of the input variable according to the standard deviation measurements. So, we perform data normalization scaling on the input data (input features) and convert it from the original range into a new range between 0 and 1. We use the Python sci-kit-learn library object called "MinMaxScaler" which can be used for this purpose.

**Data Analysis**

In analyzing student enrollment in a particular course, our approach is to use binary classification machine learning algorithms to predict whether a student will enroll or not. The first step in this process is to split the historical data into a training set and a testing set. The training set is used to train the algorithm to identify patterns in the data and make accurate predictions, while the testing set is used to evaluate the accuracy of the predictions on unseen data.

We use the "train_test_split" object from the Python Scikit-learn library to split the dataset into training and testing sets. We set the "test_size" parameter to 0.2 to allocate 20% of the data to the testing set, and the "train_size" parameter to 0.8 to allocate 80% of the data to the training set. We also set the "shuffle" parameter in the train_test_spilt object to "True" to randomize the order of the data samples. Furthermore, we use the "stratify" parameter to ensure that the distribution of the target variable is represented in both the training and testing sets. As we are building a classification model to predict whether a student will enroll in the PS course or not, we use the stratify parameter on the target label to ensure that the distribution of the target variable is represented in both the training and testing sets. This helps to prevent bias in the model and improves its accuracy.

Because our dataset has a numerical input and categorical output, we used the ANOVA F-value correlation statistics technique in the feature selection process and applied it to our classification problem. Specifically, we used the f_classif () function from the Scikit-learn library to compute it. By selecting the top k most relevant features using the SelectKBest class, we were able to identify the 5 most significant features: STUDENT_STATUS, SEMESTER, STUDENT_LEVEL, FACULTY_NO EARNED_HRS. We excluded the irrelevant ones.

Additionally, plotting a correlation heatmap Figure 1 below is a helpful way to visualize the relationships between features and identify which ones are most strongly correlated with the target variable. The correlation heatmap identified the most correlated features, which matched the results obtained using the ANOVA F-value correlation statistics technique.

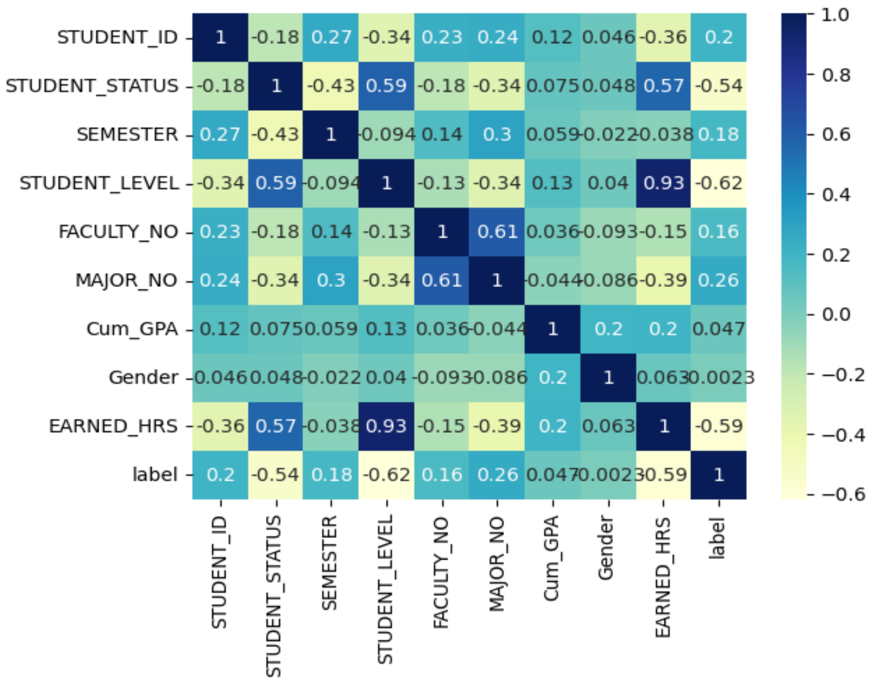


Figure 1: Correlation heatmap

When training an MLP neural network for binary classification, we employed similar techniques and methods as we did for other machine learning models. However, we deviated from our usual data normalization approach used with machine learning algorithms in this study and instead opted to use the Standard Scaler in the case of MLP. Our decision was based on experimentation, which revealed that the Standard Scaler slightly outperformed the MinMax scaler in terms of model performance.

To split the data into training and testing sets, we used the train_test_split object with a 20% testing set and 80% training set of dataset size. We set the shuffle parameter to true and the stratify parameter to ensure that the target variable's distribution is represented in the sample. Additionally, we removed outliers and selected the 5 most significant input features using the same methods as in the machine learning analysis above. We noted an improvement in the accuracy and performance of the model during experiments with and without data preprocessing.

**Models Evaluation methodology**

The evaluation methodology involves using stratified k-fold cross-validation with k=10 to prevent overfitting. Eight classifiers are trained on the training dataset, and their predictions on the test dataset are evaluated using accuracy, precision, recall, and F1 score metrics. The focus is on minimizing false negatives since missing a case or misclassifying the minority class is costlier than the majority class.

For imbalanced classification, accuracy may not be optimal since they are insensitive to skewed domains. Instead, the F1 score which is the harmonic mean of precision and recall, is preferred since it considers the model's performance for each class. In this particular highly imbalanced dataset, and after using under-sampling techniques we have a moderated imbalanced class distribution, where about 65% of instances belong to the negative outcome class, representing "Not enroll," and 35% of instances belong to the positive outcome class, representing the "enroll" status for the PS course.

In this specific problem, the majority class represents students who did not enroll in the PS course, and the minority class represents those who did. TN and TP are the cases correctly classified, while FN and FP are the cases misclassified by the model. we consider the following Confusion matrix analysis

| | **Negative prediction 0** | **Positive Prediction 1** |
|---|---|---|
| **Negative Class 0 (Student Not enrolled in PS course)** | True Negative (TN) | False Positive (FP) |
| **Positive Class 1 (student enrolled in PS Course)** | False Negative (FN) | True Positive (TP) |

Accuracy = number of corrected predictions/total number of predictions made.

c

Precision summarizes the fraction of examples assigned to the positive class that belongs to the positive class.

Precision = TP / (TP + FP)

Recall summarizes how well the positive class was predicted and is the same calculation as sensitivity.

Recall = TP / (TP + FN)

F-Measure is the harmonic mean of precision and recall.

F-measure = (2 * Precision * Recall) / (Precision + Recall)

## Experimental Studies

### Experiment 1: Predicting using the original Dataset before under sampling

A model was designed to gather historical enrollment records of students who enrolled in a particular course (PS) from Fall 2018 to Summer 2021. Nine input features were used, and the labeled column was named "enroll in PS course." The label was then converted using one-hot encoding to represent the positive class (minority class). Similarly, students who had not registered for the course in the same semester were labeled as "Not enrolled in PS course" and converted to 0 to represent the negative class (majority class).

The scikit-learn library was used to validate models using stratified k-fold cross-validation with k=10 splits to avoid overfitting. Since the problem is a classification problem, stratified k-fold cross-validation is preferred over regular k-fold cross-validation, as the splits are not completely random, and the ratio between target classes is maintained in each fold, similar to the full dataset.

Before validating the models, the model's parameters were tuned using the Grid search technique with the help of scikit-learn's GridSearchCV function. The goal was to find the optimal hyperparameters for the eight machine learning models used in the study, based on the F1 score and accuracy metrics.

The Grid Search technique was used to find the best hyperparameters for the original severely imbalanced dataset, which had about 137,000 records before resampling it to moderate imbalance. The dataset was split into an 80% training set and a 20% test set using stratified cross-validation with kfold=10. The eight classifiers were then trained and validated with the tuned hyperparameters on the training dataset, Table 2 shows the tuned hyperparameters for all algorithms.

and the best model was selected for prediction purposes based on performance metrics such as accuracy, F1 score, precision, and recall. Finally, the trained classifiers were tested on the test set, and their performance was recorded for comparison as in Table 3. The results of the experiment show that the best algorithm for the positive class 1 is the Random Forest algorithm,

Table 2: Best hyperparameters for the original dataset before resampling.

| **Model name (Classifier)** | **Best Hyperparameters** |
|---|---|
| **Random Forest Classifier (RF)** | {'bootstrap': True, 'max_features': 'sqrt', 'min_samples_split': 12, 'n_estimators': 1000} |
| **Logistic Regression (LR)** | {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'} |
| **Stochastic Gradient Descent (SGD)** | {'penalty': 'l2', 'loss': 'modified_huber', 'learning_rate': 'constant', 'eta0': 10, 'alpha': 100} |
| **K-Nearest Neighbors (KNN)** | {'metric': 'Manhattan', 'n_neighbors': 3, 'weights': 'uniform'} |
| **Super vector machine (SVM)** | {'C': 50, 'gamma': 'auto', 'kernel': 'poly'} |
| **Decision Tree Classifier (CART)** | {'criterion': 'entropy', 'max_depth': None, 'max_features': 10, 'min_samples_leaf': 6} |
| **Gradient Boosting Classifier (GB)** | {'learning_rate': 0.1, 'max_depth': 9, 'n_estimators': 1000, 'subsample': 0.7} |
| **Bagging Classifier (BG)** | {'n_estimators': 1000} |

The best F1 score achieved was about 55%, and the best accuracy was 94%, with a recall of 43% and a precision of 75%. The Random Forest algorithm is considered a bagging algorithm, which is a type of ensemble machine learning algorithm called Bootstrap Aggregation or Bagging.

Table 3: Comparison of algorithm performance with accuracy, F1 score

| **The Original Data Set before resampling** | | | |
|---|---|---|---|
| **Algorithm name** | **Accuracy on Train Set%, Accuracy on Test Set %** | **Precision%, Recall%, F1 score%** | **Precision%, Recall%, F1 Score%** |
| | | **class 0** | **class 1** |
| **Random Forest (RF)** | 94,93 | 94, 98,96 | 75,43,55 |
| **Logistic Regression (LR)** | 90,90 | 90,100,95 | 33,0,0 |
| **Stochastic Gradient (SGD)** | 90,90 | 91,95,99 | 43,5,8 |
| **K-Nearest Neighbors (KNN)** | 92,92 | 94,97,96 | 62,45,52 |
| **Decision Tree (CART)** | 92,90 | 94,95,95 | 49,44,47 |
| **Gradient Boosting (GB)** | 93,91 | 94,97,96 | 68,42,52 |
| **Bagging Classifier (BG)** | 93,92 | 94,97,96 | 61,47,53 |
| **Super Vector machine (SVM)** | 90,90 | 91,100,95 | 84,3,6 |

Furthermore, the results indicate that all classifiers are unstable (high variance) and have poor performance for this type of problem with a highly imbalanced dataset.

**Experiment 2**: **Predicting using the Dataset After under sampling**

By applying under-sampling techniques on the negative class in the Dataset, resulting in a ratio of approximately 3.5 to 1 for class 0 to class 1. Then, we used Grid search techniques again to find the best hyperparameters for each classifier. The results are summarized in Table 4. It appears that the best hyperparameters for the resampled dataset are different from the ones for the original imbalanced dataset for some algorithms. This is expected since the data distribution has changed after applying the under-sampling technique.

Table 4: Best hyperparameters for the resampled dataset

| **Model name (Classifier)** | **Best Hyperparameters** |
|---|---|
| **Random Forest Classifier(RF)** | (bootstrap=True,max_features='sqrt',min_samples_split= 8,min_samples_leaf= 8,n_estimators=1000) |
| **Logistic Regression (LR)** | {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'} |
| **Stochastic Gradient Descent (SGD)** | {'penalty': 'elastic net', 'loss': 'squared_hinge', 'learning_rate': 'in scaling', 'eta0': 10, 'class_weight': {1: 0.3, 0: 0.7}, 'alpha': 100} |
| **K-Nearest Neighbors (KNN)** | {'metric': 'Manhattan', 'n_neighbors': 3, 'weights': 'uniform'} |
| **Super vector machine (SVM)** | {'C': 50, 'gamma': 'auto', 'kernel': 'poly'} |
| **Decision Tree Classifier (CART)** | {'criterion': 'entropy', 'max_depth': None, 'max_features': 8, 'min_samples_leaf': 6} |
| **Gradient Boosting Classifier (GB)** | {'learning_rate': 0.1, 'max_depth': 9, 'n_estimators': 1000, 'subsample': 1.0} |
| **Bagging Classifier (BG)** | {'n_estimators': 100} |

We evaluated the performance of the classifiers on both training and test sets using cross-validation and recorded the performance metrics such as accuracy, F1 score, precision, and recall recorded as shown in Table 5. It has been found that the Random Forest classifier (RF) performed the best, according to the F1 score, for positive class 1. The model achieved an 84% recall, 88% precision, and 94% accuracy. The box plot in Figure 3 provides a visual comparison of the algorithms in terms of accuracy after the evaluation of the training set. Additionally, Figure 4 displays a comparison of the same algorithms in terms of F1 score after evaluation on the test set (unseen data). Finally, the confusion matrix for the selected RF model with tuned hyperparameters after predicting the test set can be found in Figure 2.

It is important to note that the F1 score has improved for all models compared to the original dataset, indicating that the resampling has helped to improve the performance of the classifiers. Overall, it seems that the RF model with the tuned hyperparameters provides the best performance for this kind of problem.

Table 5: Comparison of algorithms performance with accuracy, F1 score

| Algorithm name | The Data Set After Resampling | | |
|---|---|---|---|
| | Accuracy on Train Set%, Accuracy on Test Set % | Precision%, Recall%, F1 score% | Precision%, Recall%, F1 score% |
| | | class 0 | class 1 |
| **Random Forest Classifier (RF)** | 94,94 | 95,97,96 | 88,84,86 |
| **Logistic Regression (LR)** | 88,86 | 89,93,91 | 75,64,70 |
| **Stochastic gradient Classifier (SGD)** | 87,86 | 93,88,90 | 68,79,73 |
| **K-Nearest Neighbors (KNN)** | 91,90 | 93,95,94 | 83,77,80 |
| **Decision Tree Classifier (CART)** | 91.5,89 | 94,93,93 | 78,81,79 |
| **Gradient Boosting Classifier (GB)** | 94,93 | 95,95,95 | 84,83,83 |
| **Bagging Classifier (BG)** | 94, 93 | 94,96,95 | 84,83,84 |
| **Super vector machine (SVM)** | 90,88 | 92,94,93 | 76,71,74 |

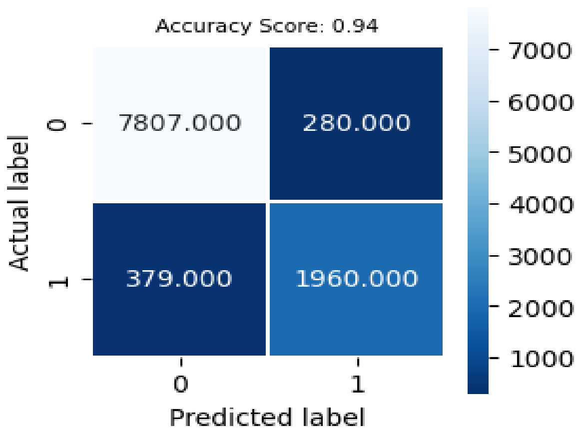


Figure 2: Confusing matrix of random forest on the testing set

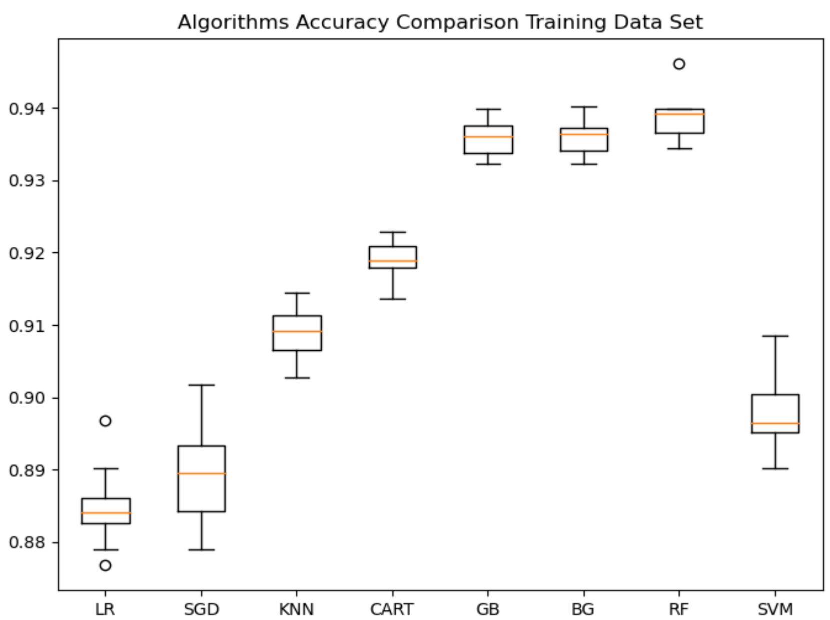


Figure 3: Algorithms Accuracy Comparison on Training Set

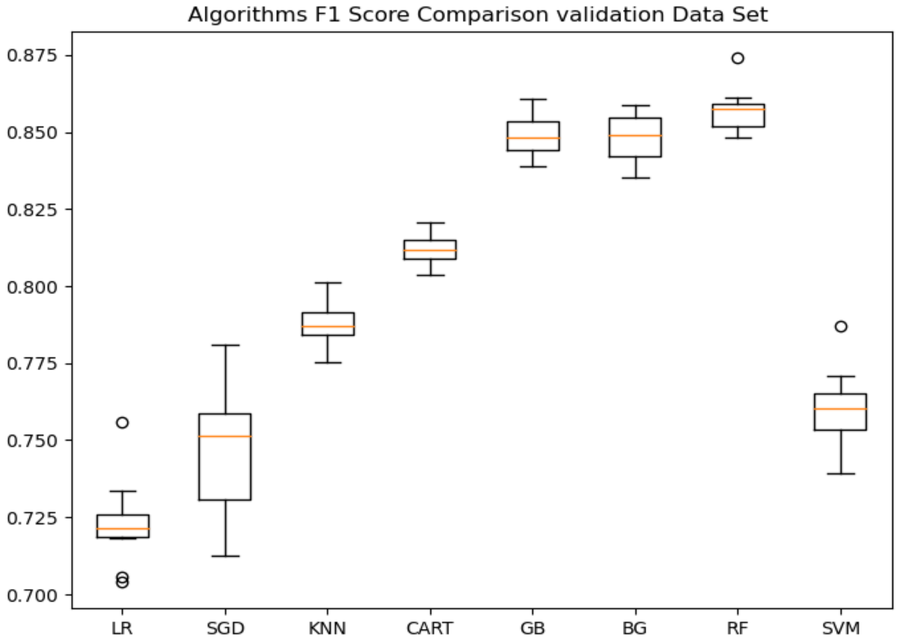

Figure 4: Algorithms F1- measure comparison on test set

**Experiment 3: Construct an MLP neural network model using the original dataset, before any resampling.**

We performed hyperparameter tuning using Bayesian optimization on an MLP neural network model. The hyperparameters optimized included the number of neurons, activation function, optimizer, learning rate, batch size, dropout rate, epochs, number of layers, batch normalization, and dropout layers. The optimization was carried out on the original dataset, which had 137,000 records, and the model was optimized for F1 scoring. The best hyperparameters and neural network structure were found to be *{'activation': LeakyReLU, 'batch_size': 472, 'dropout': 0.65, 'dropout_rate': 0.28, 'epochs': 50, 'layers1': 3, 'layers2': 3, 'learning_rate': 0.62, 'neurons': 39, 'normalization': 0.37, 'optimizer': Adagrad}.*

The neural network has 39 neurons in each hidden layer, with three hidden layers before the output layer. The first hidden layer is not followed by batch normalization because the normalization value is less than 0.5. The dropout rate is set at 0.28, meaning that 28% of the neurons are randomly dropped before the values are passed to the next three hidden layers. The output layer has one neuron containing the probability value.

As shown in Table 6 below, The MLP neural network model achieved an accuracy of 91% on the testing dataset and an F1 score of 30% for the positive class 1. This indicates relatively poor performance compared to the other classifiers used in experiment number 1. The precision was 65%, and the recall was 19%. These results suggest that the MLP neural network may not be the best model for this particular imbalanced dataset.

Table 6. Evaluation results of Experiment 3

| | **The original Data set before resampling** | | |
|---|---|---|---|
| **Algorithm name** | **Accuracy Train set /Test set** | **Precision/Recall/F1 score** | **Precision/Recall/F1 score** |
| | | **class 0** | **class 1** |
| **MLP Neural network** | 91%,91% | 92%,99%,95% | 65%,19%,30% |

**Experiment 4: Construct an MLP neural network model using a resampled dataset.**

In this experiment, we implemented Bayesian Optimization on a prepared and under-sampled dataset of about 51,000 records to optimize F1 scoring. A function was used to check different combinations of hyperparameters and neural network structures, including activation functions, optimizers, learning rate, batch size, number of epochs, number of neurons in each layer, dropout rate, and normalization value. The best result was achieved using the following hyperparameters and neural network structure: *{'activation': 'soft plus', 'batch_size': 999, 'dropout': 0.14, 'dropout_rate': 0.18, 'epochs': 59, 'layers1': 2, 'layers2': 2, 'learning_rate': 0.33, 'neurons': 86, 'normalization': 0.32, 'optimizer': Ftrl}.*

After preparing the dataset, we split it into training and testing sets using the train_test_split function from the scikit-learn library, with 80% of the data used for training and 20% for testing. We then implemented a neural network model using Keras and TensorFlow, with the best hyperparameters and neural network structure obtained through Bayesian Optimization. The model had 3 hidden layers, with the first layer having 86 neurons, followed by 2 hidden layers with 2 neurons each. We used the common practice " normal" weight initialization technique and early stopping technique to avoid overfitting. The output layer had a single node with a sigmoid linear activation function, predicting a value between 0 and 1 for binary classification.
We used Ftrl optimizer and binary_crossentropy as the loss function to train the model, which was trained for 59 epochs with a batch size of 999 samples. The model was evaluated on the test dataset using various metrics like Accuracy, Precision, Recall, and F1 score. The model was cross-validated with a k-fold equal to 10, the result in Table 7 shows that the model achieved an accuracy of 91%, precision of 81%, recall of 78%, and F1 score of 79% on the train set. We also tested the model on an unseen dataset, achieving the same accuracy of 91% and the same F1 score of 79%.

Table 7 Evaluation Results of Experiments 4

| | **The Resampled Dataset** | | |
|---|---|---|---|
| **Algorithm name** | **Accuracy**<br>**Train Set/Test set** | **Precision/recall/F1 score** | **Precision/recall/F1 score** |
| | | class 0 | class 1 |
| **MLP Neural network** | 91%,91% | 94%,94%,94% | 81%,78%,79% |

**Binary Cross-Entropy Loss**

Binary cross-entropy is a widely used loss function for binary classification problems because it is effective, easy to compute, and encourages the model to output probabilities that are close to the true labels. In Figure 5, we created two-line plots. The top plot shows the learning curves of the cross-entropy loss over epochs for the train (blue) and test (orange) sets. The bottom plot shows the classification F1 score over epochs for the train and test sets for experiment 4. In this case, we can see that the model learned the problem reasonably well, achieving about a 79% F1 score on the training dataset and about 79% on the test dataset. The scores are reasonably close, suggesting the model is probably not over or underfit. The plots suggest that the model has a good fit for the problem and that the training process converged well.

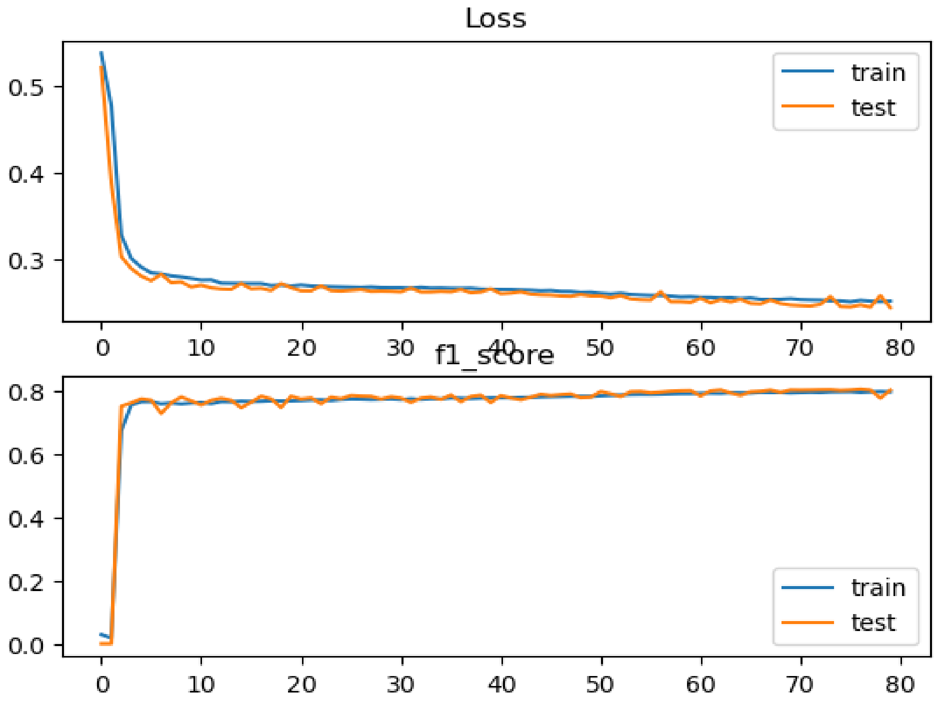


Figure 5: Binary cross-entropy loss for experiment 4

## Results and Discussion

The experiments conducted in this study focused on the classification of student enrollment data, to predict future enrollment using machine learning and MLP neural network algorithms. The experiments conducted in the previous sections show that traditional machine learning models especially tree-based and Bagging algorithms such as Random Forest, perform better than other linear-based classifiers and MLP neural network algorithms in classifying highly imbalanced datasets as well as for moderately imbalanced datasets concerning accuracy, recall, and F1 score. while MLP neural networks perform well on moderately imbalanced datasets.

The under-sampling technique significantly improves the performance of classification models without affecting the data quality. Additionally, in terms of feature selection, it was found that a small set of five input features can achieve the required accuracy and F1 score metrics for predicting enrolled students in a specified course using an individual forecasting approach. The use of standard scalar techniques also slightly improves the performance of models compared to MinMax scalar techniques. Overall, the study provides valuable insights into the use of different Machine learning algorithms and techniques for classification and forecasting in education-related applications.

## Conclusion

The paper presents a study on predicting course enrollment at AAUP using machine learning and MLP neural network techniques. The study focuses on developing individual-level classification models using only historical academic data with a limited set of input features. The study also explores the use of under-sampling techniques to address highly imbalanced datasets and evaluates the models' performance using accuracy, F1 score, Precision, and Recall metrics. The results show the effectiveness of these techniques in predicting course enrollment, with high accuracy and F1 scores achieved.

The study provides valuable contributions to the field of enrollment management in higher education and can inform enrollment management decisions to improve administrative performance. Future research can expand on these techniques and investigate the development of a generalized model for predicting student enrollment for any course in the institute.
Overall, the study highlights the potential of machine learning and neural network techniques in predicting course enrollment and improving enrollment management practices.

## Recommendations

1. For researchers and practitioners in enrollment management or data analysis in higher education, this article provides valuable insights into how machine learning and neural network techniques can be used to predict individual-level enrollment in a particular course. The article compares different algorithms and provides a performance evaluation of each, which can help readers understand the advantages and limitations of each algorithm.
2. For those interested in machine learning and neural networks, this article provides a real-world example of how these techniques can be applied in the context of enrollment management in higher education. The article also highlights the importance of addressing imbalanced datasets, which is a common issue in many real-world applications.
3. Consider using traditional machine learning algorithms such as Random Forest and Bagging algorithms for highly imbalanced datasets and moderately imbalanced datasets. These algorithms have shown better performance in terms of accuracy, recall, and F1 score than linear-based classifiers and MLP neural network algorithms.
4. Utilize under-sampling techniques to improve the performance of classification models without affecting the data quality. The study found that under-sampling techniques significantly improve the performance of classification models, especially for highly imbalanced datasets.
5. Use a small set of input features for predicting enrolled students in a specified course using an individual forecasting approach. The study found that only five input features can achieve the required accuracy and F1 score metrics.
6. Finally, for those interested in improving resource allocation and student support in higher education institutions, this article highlights the potential of machine learning and neural network techniques for optimizing operations and better meeting the needs of students. The article calls for further research to develop a generalized model for predicting enrollment in any course at AAUP, which can have broader implications for other similar institutions.

## Acknowledgments

The authors would like to thank the AAUP Administration for their support in facilitating using their real Data sets.

## References


Akinode, J.L., & Bada, O. (2021). Student enrollment prediction using machine learning techniques. In Proceedings of the 5th National Conference of the School of Pure & Applied Sciences Federal Polytechnic Ilaro, Nigeria (pp. 476-484).

Doleck, T., Lemay, D. J., Basnet, R. B., & Bazelais, P. (2020). Predictive analytics in education: a comparison of deep learning frameworks. Education and Information Technologies,25(3), 1951–1963. https://doi.org/10.1007/s10639-019-10068-4

Esquivel, J. A., & Esquivel, J. A. (2020). Using a binary classification model to predict the likelihood of enrolment in the undergraduate program of a Philippine University. *International Journal of Computer Trends and Technology*, *68*(5), 6–10. https://doi.org/10.14445/22312803/ijctt-v68i5p103

Kardan, A. A., Sadeghi, H., Ghidary, S. S., & Sani, M. a. F. (2013). Prediction of student course selection in online higher education institutes using neural network. Computers & Education, 65, 1–11. https://doi.org/10.1016/j.compedu.2013.01.015

Mia*, M. J., Biswas, A. A., Sattar, A., & Habib, M. T. (2019). Registration status prediction of students using machine learning in the context of the Private University of Bangladesh. International Journal of Innovative Technology and Exploring Engineering, 9(1), 2594–2600. https://doi.org/10.35940/ijitee.a5292.119119

Saini, P., & Jain, A. (2013). Prediction using Classification Technique for the Students' Enrollment Process in Higher Educational Institutions. International Journal of Computer Applications, 84(14), 37–41. https://doi.org/10.5120/14646-2966

Shao, L., Ieong, M., Levine, R. A., Stronach, J., & Fan, J. (2022, June 30). *Machine learning methods for course enrollment prediction.* Strategic Enrollment Management Quarterly. Retrieved April 23, 2023, from https://asir.sdsu.edu/Documents/SMGDocs/SEMQ-1002-Shao.pdf

Shilbayeh, S., & Abonamah, A. (2021). Predicting student enrolments and attrition patterns in higher educational institutions using machine learning. *The International Arab Journal of Information* Technology, 18(4). https://doi.org/10.34028/18/4/8

Slim, A. M., Hush, D., Ojah, T., & Babbitt, T. H. (2018). Predicting Student Enrollment Based on Student and College Characteristics. In Educational Data Mining. Retrieved April 23, 2023, from http://educationaldatamining.org/files/conferences/EDM2018/papers/EDM2018_paper_136.pdf

Wanjau, S. K., Okeyo, G., & Rimiru, R. (2016). Data Mining Model for predicting student enrolment in STEM Courses in Higher Education Institutions. *International Journal of Computer Applications Technology and Research*, *5*(11), 698–704. https://doi.org/10.7753/ijcatr0511.1004