COMPSCI 596 Independent Study
ML Model Based Students Online Exam Behaviors Prediction

Yujin Qin, Yi Ding, Atif Abedeen

Introduction

As the COVID-19 arose, most of the courses in school were transitioned into online courses and it has created unique challenges for the educational system. These changes may have impacted the prevalence of cheating in different ways, as it became much easier for students to use external resources for help. In addition to the occurrence of ChatGPT this year, students are more likely to cheat or search for solutions during quizzes or exams. The purpose of this project is to build an application that uses data analysis and machine learning to predict if students cheat on their exams from their clicking streaming data.

This project is relevant to both major areas of study and our college experiences. As Computer Science major students, we are very interested in Machine Learning and how it could be utilized and make a better academic community environment. Having an application that could possibly predict and detect cheating in an exam would make students more compliant with the academic dishonesty policy and focus on learning experience instead of depending on external resources.

Literature Review

The paper that we looked at is called the "Analysis of Learning Records to Detect Student Cheating on Online Exams: Case Study during COVID-19 Pandemic" by Antonio Balderas and Juan Antonio Caballero-Hernández. This paper focuses on using learning records to detect student cheating during online exams, particularly in the context of the COVID-19 pandemic.

The authors address the challenge of maintaining academic integrity in online exams, where students may have increased opportunities to engage in dishonest practices. They propose an approach that leverages learning records, which capture various data points related to student behavior and performance, to identify potential instances of cheating. They consider various factors such as completion times, answer patterns, and consistency across exams to detect irregularities. They ordered the sequences of students who took the exam and found that students who took the exams later did better and faster on the exam compared to the students who started earlier.

Another study we looked at was the "The NAEP EDM Competition: On the Value of Theory-Driven Psychometrics and Machine Learning for Predictions Based on Log Data" by Fabian Zehner, Scott Harrison, and others. The focus of the competition was to predict efficient test-taking behavior using log data, and the authors present their top-down approach to feature engineering based on psychometric modeling, with the aim of utilizing machine learning for predictive classification. The authors used various techniques for feature engineering, including the Log-Normal Response Time Model to estimate latent person speed and the Generalized Partial Credit Model to estimate latent person ability. They also adopted an n-gram feature approach for event sequences. Instead of using the provided binary target label, they differentiated between inefficient test takers who were going too fast and those who were going too slow, training a multi-label classifier.

Methodology

One potential possible way to build our model is to use one of the machine learning algorithms - random forest. We'll create different decision trees to sort the data, and then the decisions are made after computing all features. Each tree will give us the probability of the student being "cheating" or "honest". When the subtrees result is combined, the model will mark students as "cheating" or "honest". Second possible solution would be using the logistic regression method which is a supervised learning algorithm. The decisions are categorized into "cheating" or "honest" and the probability of cheating is calculated after checking a set of conditions. We also used Neural Networks (NNs) which are very useful in pattern recognition and mapping non-linear features. And finally, we used Support Vector Machines (SVMs) which

is very useful in finding an optimal hyperplane in a high-dimensional feature space to separate different classes, maximizing the margin between them. After comparing all the models and finding the best performance, the conclusion will be drawn.

Data

The data set we worked with is provided by Educational Testing Service, with permission from The Nation's Report Card, also known as the National Assessment of Educational Progress (NAEP). This dataset provides us the Response Process Data From the 2017 NAEP Grade 8 Mathematics Assessment. It includes not only the answers that the students selected or entered, but also their actions or events initiated during the digital assessment, such as their utilization of the onscreen calculator, clicking on response options, removing response options, and key presses.

Study Procedure

1. Data Collection: Collect and gather the data for the model.
2. Data Preprocessing: Clean, prepare, and transform the data to ensure it is in the correct format for analysis. This may involve tasks such as data normalization, handling missing values, and feature engineering.
   Since our data doesn't include pre-include labels. We labeled our data using the total completion time that the students spent on the exam based on the normal distribution. We assigned 0 for the 50% students in the middle of the bell curve which means "non cheating " and the two ends of the bell as 1 meaning "cheating".
3. Split Data: Split the data into two sets: a training set and a testing set. The training set is used to train the model, while the testing set is used to evaluate the performance of the model.
4. Feature Selection: Select the features that will be used in the ML model. This involves identifying the most relevant and important features that will help the model make accurate predictions.

Features Used for Derived Feature Modeling

Features	Description
Grade (in Percentage)	A integer that represents students grades in percentage
Calculator	A count variable indicating the total number of times students opened the calculator tool
Too Fast	A count variable indicating the number of times that student is in the fastest 5% of all respondents for a single question
Rapid Answering	Over 15% of the all test questions（5 questions）, the time spent on these questions were very fast and the answer was correct at the same time
Question Attempts	A count variable indicating the avg number of times students try to answer the question or modify exist answer
No Second Attempt	Finished all questions at once without second attempt or revision on answers and get 70% or higher on the test
Too Much Attempts	Attempting to answer the question or changing the answers ＞2.5 times.
No Calculator Attempt	Answered a question that required the usage of a calculator without any attempt yet getting it correct

5. Model Training: Train the logistic regression model using the training set. This involves fitting the model to the data and optimizing the parameters to achieve the best performance.

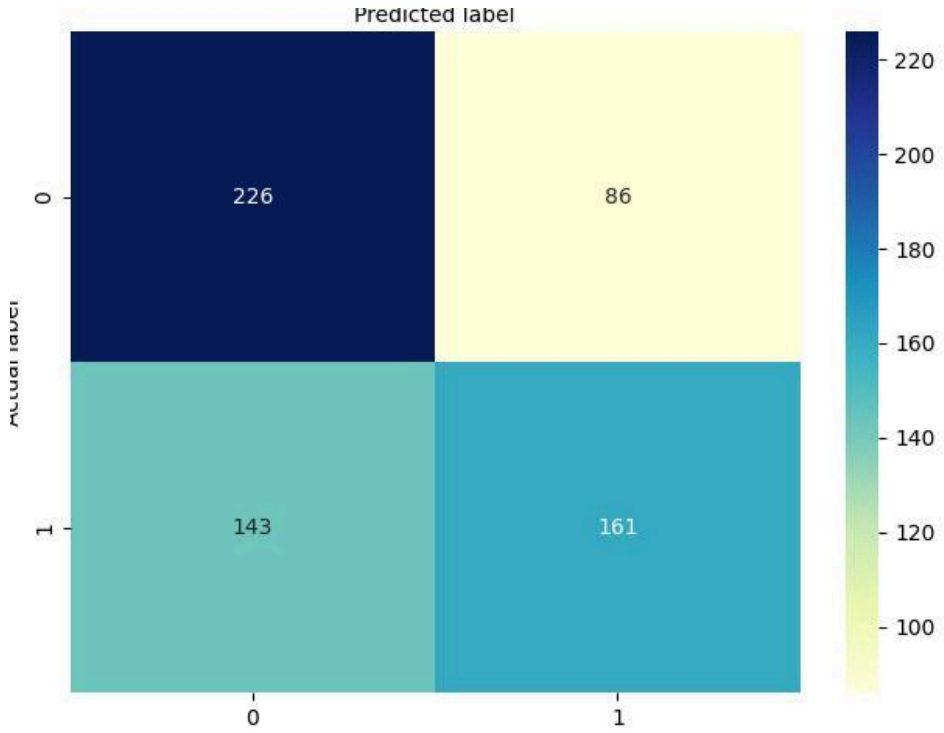
6. Model Evaluation: Evaluate the performance of the logistic regression model using the testing set. This involves calculating metrics such as accuracy, precision, recall, and F1 score to measure the model's ability to make correct predictions.

7. Model Improvement: If the model performance is not satisfactory, iterate and improve the model by adjusting the features, tuning the hyperparameters, or using more advanced modeling techniques.

Results

LOGISTIC REGRESSION

Confusion Matrix

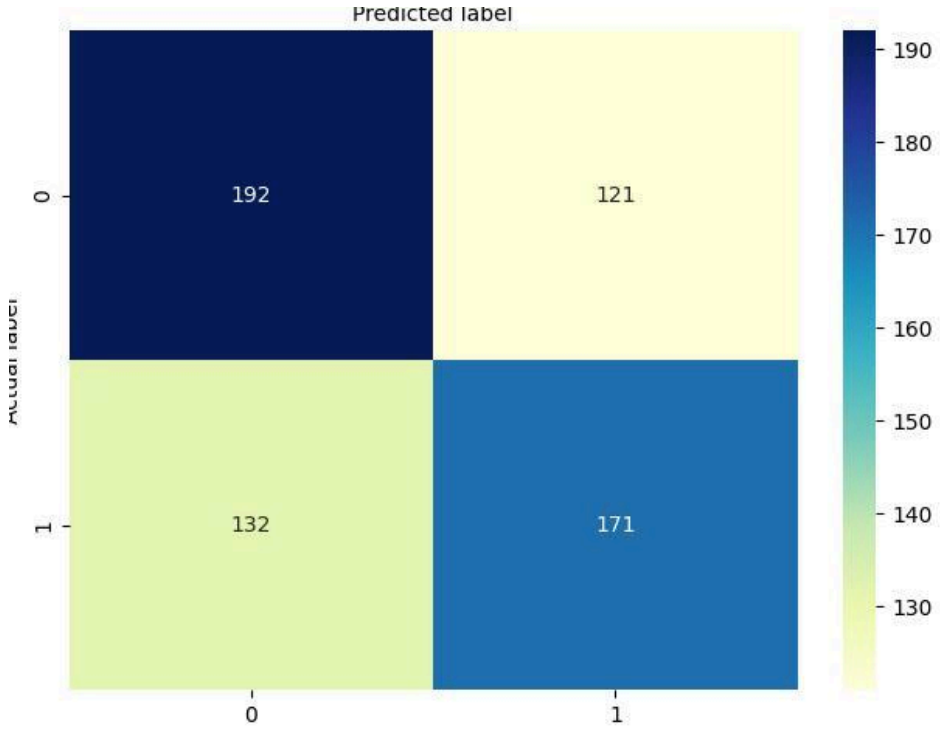


METRICS:

```
Accuracy: 0.6233766233766234
Precision: 0.648
Recall: 0.5294117647058824
F1 Score: 0.5827338129496403
AUC Score: 0.6227703984819734
```

RANDOM FOREST

Confusion Matrix of Random Forest

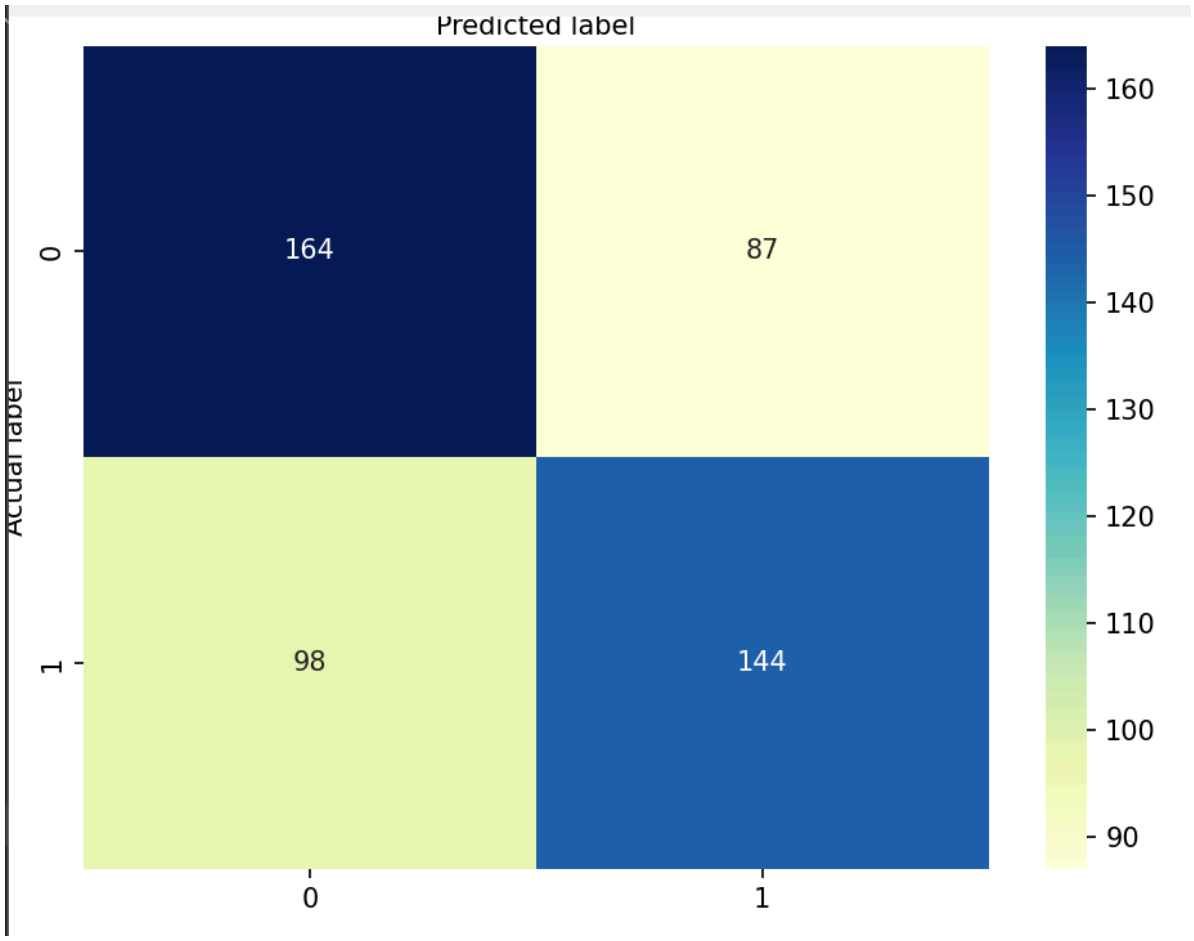


METRICS:

```
Accuracy: 0.586038961038961
Precision: 0.577922077922078
Recall: 0.5874587458745875
F1 Score: 0.5826513911620295
AUC Score: 0.5860616413079007
```

NEURAL NETWORKS

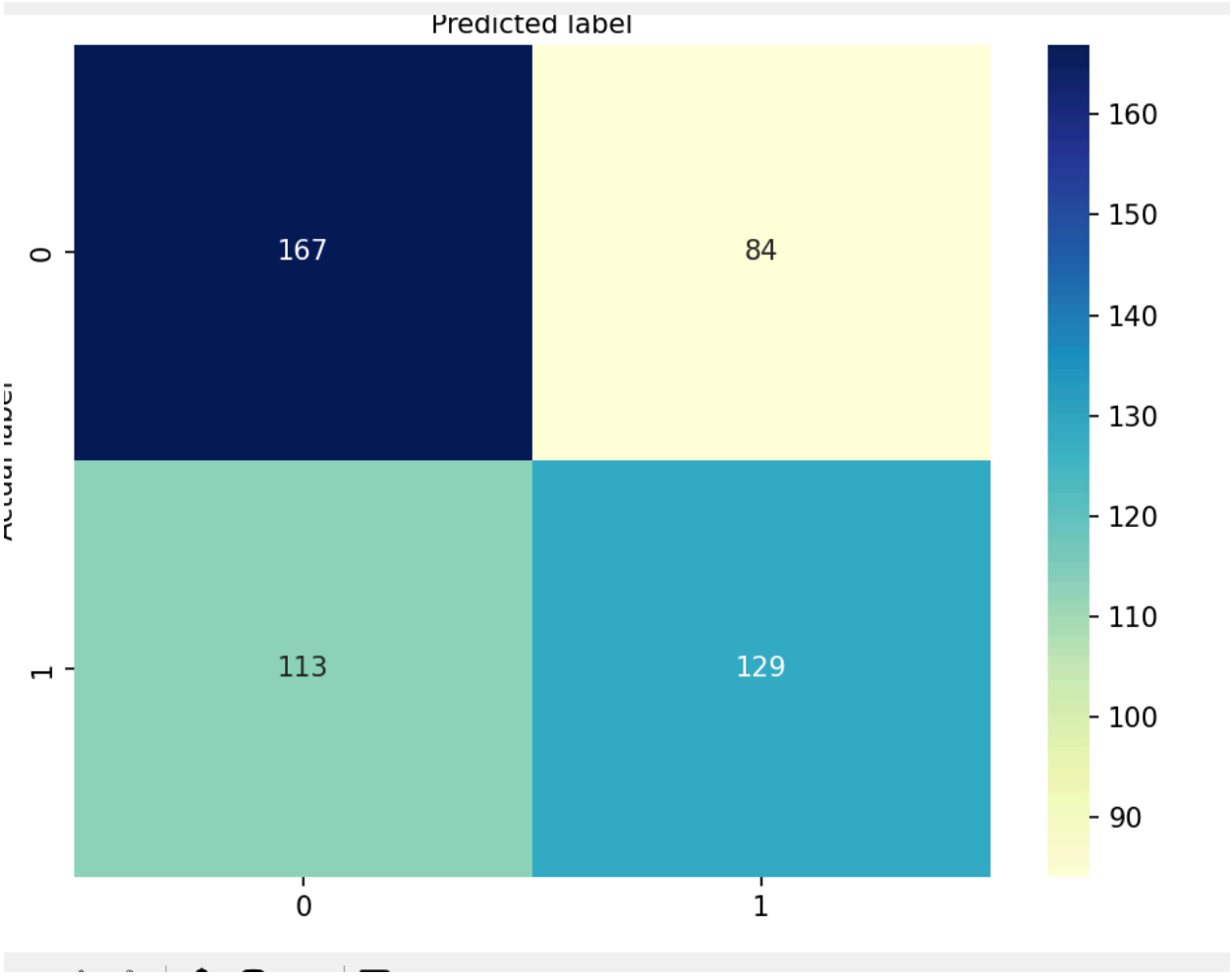
(Confusion Matrix of NN with 2 layers with 8 nodes each)



METRICS:

```
Accuracy: 0.624746450304259 7
Precision: 0.6233766233766234
Recall: 0.5950413223140496
F1 Score: 0.6088794926004228
AUC Score: 0.6242138882486582
```

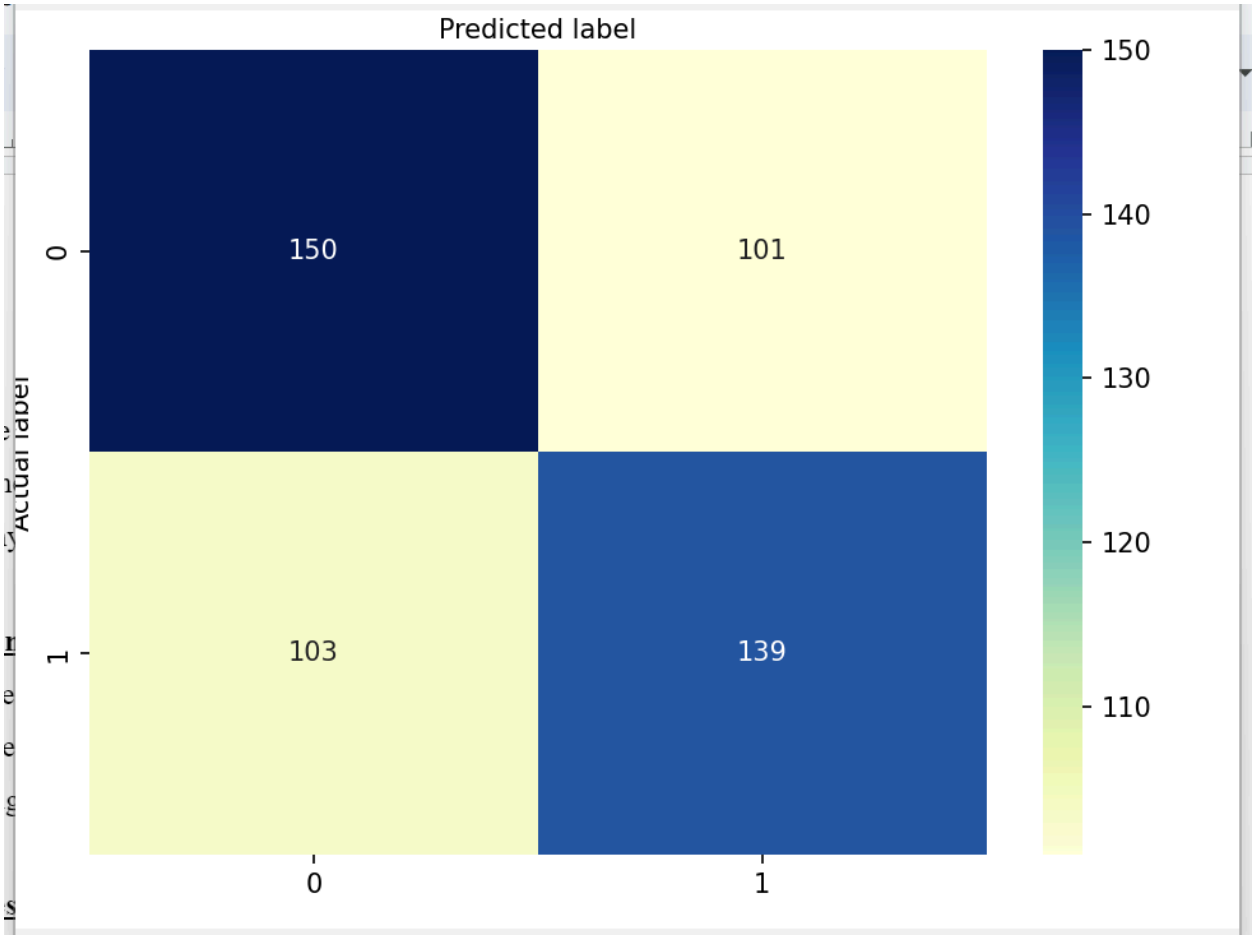
(Confusion Matrix of NN with 2 layers with 4 nodes each)



METRICS:

```
Accuracy: 0.6004056795131846
Precision: 0.6056338028169014
Recall: 0.5330578512396694
F1 Score: 0.567032967032967
AUC Score: 0.599198248328998
```

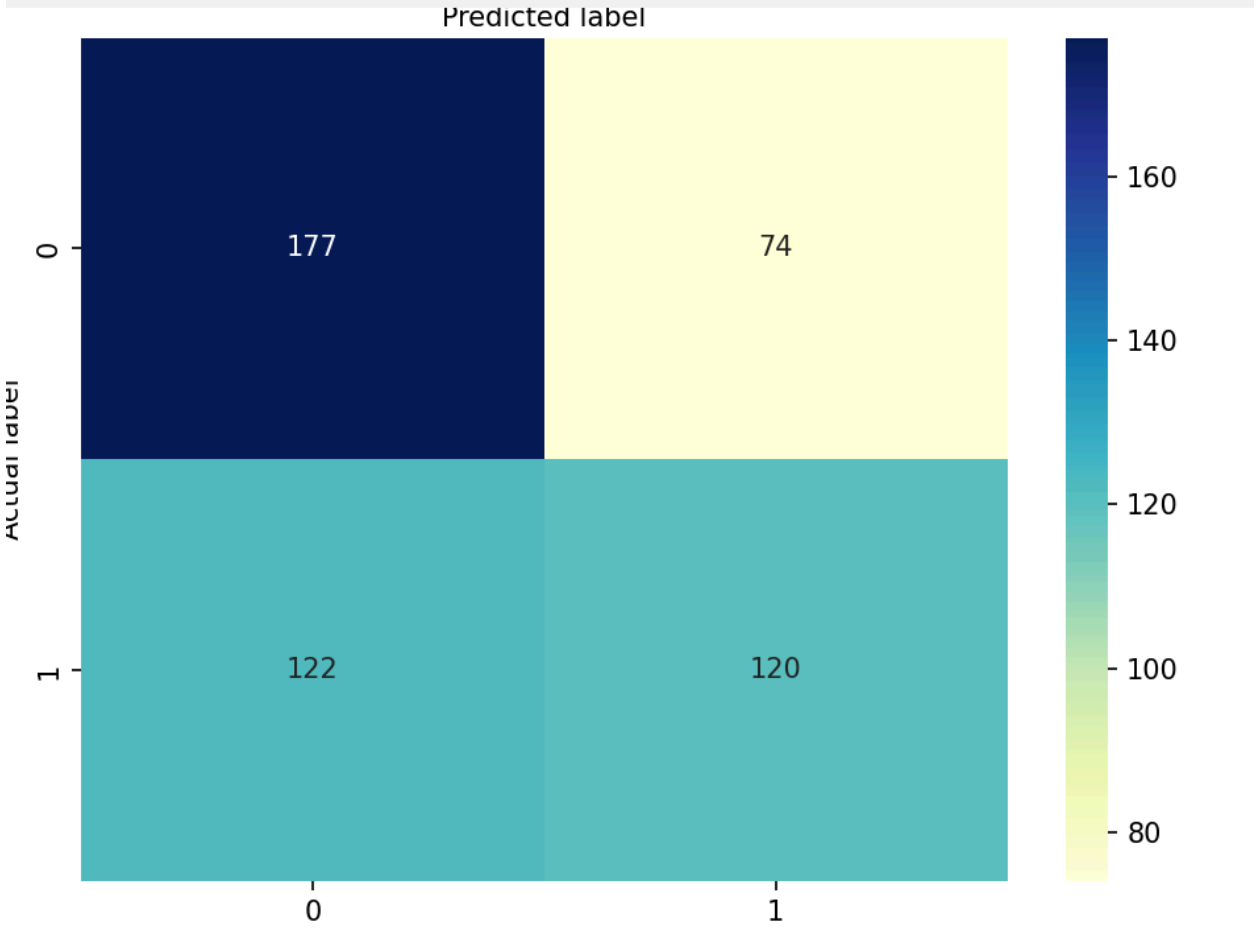
(Confusion matrix of NN with 3 layers with 4 nodes each)



METRICS:

```
Accuracy: 0.5862068965517241
Precision: 0.5791666666666667
Recall: 0.5743801652892562
F1 Score: 0.5767634854771785
AUC Score: 0.5859948635211221
```

SVM (Kernel = Linear)

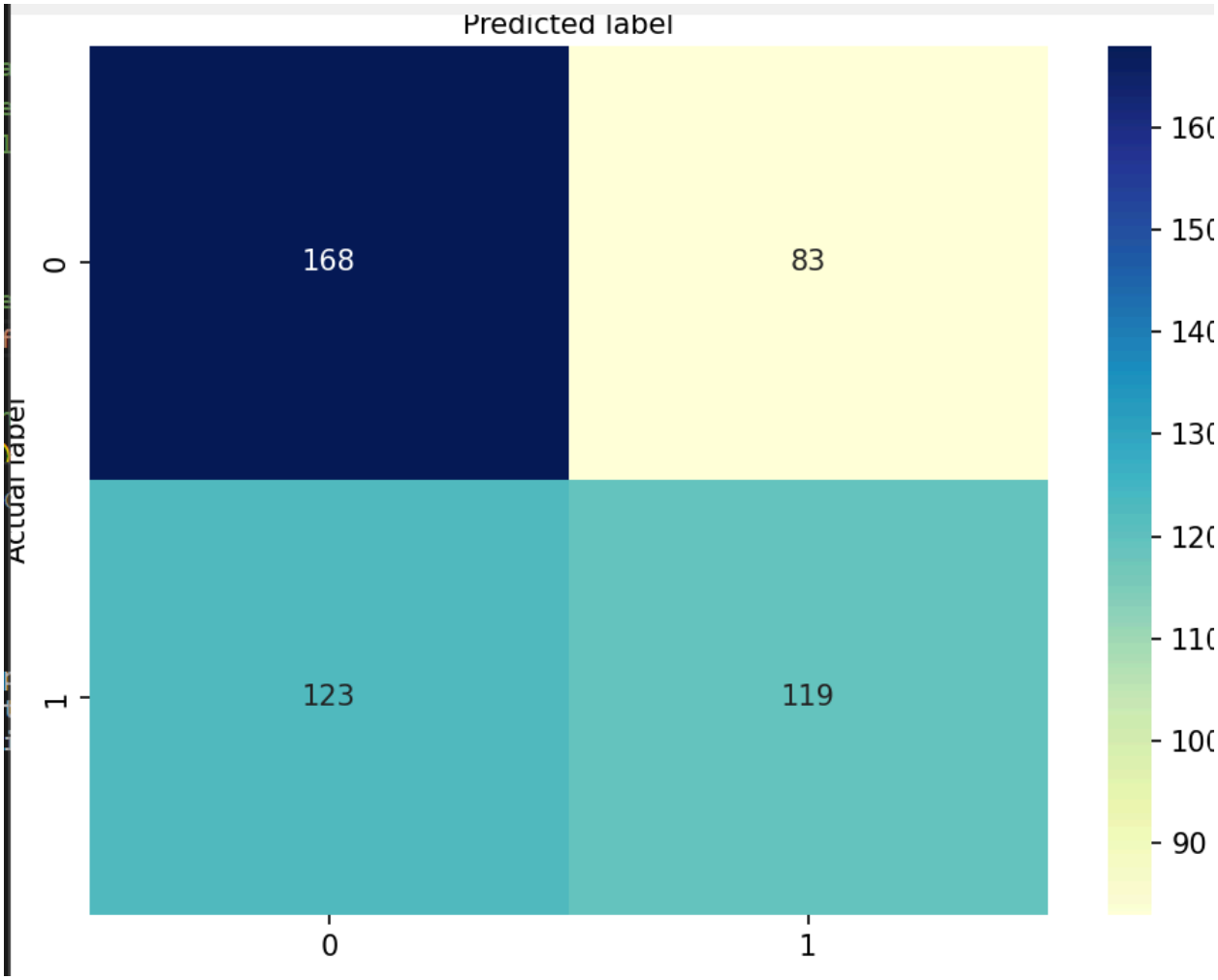


METRICS:

```
Accuracy: 0.6247464503042597
Precision: 0.6185567010309279
Recall: 0.49586776859504134
F1 Score: 0.5504587155963302
AUC Score: 0.6005235257317837
```

KERNEL=rbf

:



METRICS:

```
Accuracy: 0.6247464503042597
Precision: 0.5891089108910891
Recall: 0.49173553719008267
F1 Score: 0.5360360360360361
AUC Score: 0.5805291231767147
```

Interpretation of the results

We had tested on 4 different machine learning algorithms namely logistic regression, random forests, neural networks, and SVMs. For neural networks, we tried using different layers and different nodes. Looking at the metrics, we observed that the F1-score was the highest for the NN with 2 layers and 4 nodes each. The AUC score was highest for this neural network as well. In the case of SVMs, the highest F1-score and AUC was achieved under the linear kernel which was about the same that we achieved under NNs.

Limitations of the Study

We had a couple of challenges while conducting this study. The first one was labeling the data and separating them into two classes, ‘cheating’ and ‘non-cheating’. We had used only one feature which was if the student finished the exam too fast or too slow as explained in a previous section. We believe we could have used more features to further refine our labeling. For example, one idea we had was to compare the ‘wrong’ answers for each question that the student got incorrect with other students and if there were multiple incorrect answers that were the same among them, we would classify them as cheating. Given the large nature of the dataset, this would be computationally complex to achieve. Another challenge we faced was finding features to train our ML models with. We had found 9 features but we believe we could have used the click stream data to find more features. For example, one idea was to take two or three students that achieved the same grade and check if at any given time if they were at the same question. Then we could check how long they took to complete that question. If they finished it too fast, did not use a calculator, submitted the answer at the same time, or presented with any other suspicious behavior, then we could conclude that they had received external help. Again, given the complex nature of this idea, we decided not to pursue this method. The reason why it was so complex was because there were different types of questions in the exam and as such, there were many different types and ways to answer the questions. Given the time constraints we had, it was difficult for us to devise an algorithm that could compare the answers of the different questions.

Conclusion

Based on the results achieved using the simple machine learning algorithms, we found out that click-stream data can be quite detrimental in predicting whether a person has committed any sort of academic dishonesty or not. The algorithms we used achieved about 60% accuracy while testing. Our study only looked at individual students and their behavior but this study can be extended into comparing different students and have complex ML algorithms predict if they helped each other or not. Again, with the emergence and fast development of AI and the fact that it is readily available to everyone, it is very important to ensure that tighter regulations are kept in place to prevent students from receiving external help. This study has the chance to be the base for any other studies conducted in the future with more refined features and complex algorithms.

References

[1] Credit Card Fraud Detection Using Machine Learning ... - IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8123782

[2] Fraud Detection Using Machine Learning - Stanford University. https://cs229.stanford.edu/proj2018/report/261.pdf

[3] Hussein, Fairouz, et al. “Advances in Contextual Action Recognition: Automatic Cheating Detection Using Machine Learning Techniques.” MDPI, Multidisciplinary Digital Publishing Institute, 31 Aug. 2022, https://www.mdpi.com/2306-5729/7/9/122

[4] Kamalov, Firuz, et al. “Machine Learning Based Approach to Exam Cheating Detection.” PLOS ONE, Public Library of Science, https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0254340