PREDICTING STUDENT GRADES, A CLASSIFICATION PROBLEM

JIMMY GOH HAN JIE (23A464A) submitted for
Specialist Diploma in Applied Artificial Intelligence
Nanyang Polytechnic
School of Information Technology
Ang Mo Kio Avenue 8 Singapore 569830

ABSTRACT

In the study "Predicting Student Grades" conducted, I have leveraged on dataset from AIAP's project assessment, comprising features detailing students' characteristics and final test score. The objective was to ascertain if students' characteristics could be used to predict student score, and decide if to render additional support for students with poorer grades. Classical machine learning random forest and deep learning feed forward neural networks are developed. The performance metrics, such as Accuracy, Precision, Recall and F1, were utilized to gauge the model's efficacy, Models will be tuned to enhance Recall, as it is costlier to have False Negative (Students that need additional support are predicted as no additional support required). McNemar's statistical test indicates there is significant difference in prediction performance between the two models. Forward neural networks performed significantly better.

Comparison between deep learning and classical machine algorithms

Classical Machine Learning(ML) model comparison against Neural Network(NN) will be discussed in following 4 pillars

Data and Resource Factors

ML is suitable for smaller datasets and less computationally demanding. Effective when computational resources are limited. NN requires larger datasets and substantial computational power (e.g., GPUs/TPUs). Ideal for tasks with abundant data and scalability.

Model Complexity and Performance

ML's simpler models may underperform on complex, non-linear relationships. Effective with simpler data structures. NN excels at capturing complex, non-linear patterns and high-dimensional data, achieving state-of-the-art performance on challenging tasks.

Interpretability and Domain Knowledge Factors

ML offers greater interpretability, making it suitable when model transparency is crucial. Allows for explicit feature engineering. NN Often considered "black boxes" with limited interpretability. Requires less manual feature engineering and can automatically learn relevant features from raw data.

Task and Problem Characteristics

ML is effective for tasks with simpler data and known feature engineering. Preferred for problems where model interpretability is vital. NN is ideal for complex tasks, including image recognition, NLP, and speech recognition, where intricate patterns and high performance are essential.

In summary, the choice between classical machine learning models and deep learning neural network models depends on the specific problem, available data, computational resources, and the trade-offs between model interpretability and predictive power. Classical models are preferred when interpretability and efficiency are crucial, while deep learning models excel in tasks where capturing complex patterns and high performance are essential, given sufficient data and computational resources.

1. Introduction

The client U.A secondary School wants me to build a model that can predict the students' O-level mathematics examination scores to identify weaker students prior to the examination timely.

Additional support can then be rendered to the students to ensure they are prepared for the upcoming exam. I will be given access to U.A Secondary School's past students' performance dataset.

2. ML formulation

Additional Support will be rendered to students at borderline or below passingl, B5 and below, Singapore O's level grading. Students with the score of 64 and below will be classified as add_support = 1, score of 65 and above = 0. Using past students' characteris dataset to train Machine Learning models to predict if current batch of student require add_support

ML : Random Forest supervised classifier Machine learning will be used to predict if add_support is required

NN: Feed Forward Fully Connected Neural Network model will be used to predict if add_support is required

Dataset fields and descriptions

Attribute	Description
student_id	Unique ID for each student
number_of_siblings	Number of siblings
direct_admission	Mode of entering the school
CCA	Enrolled CCA
learning_style	Primary learning style
tuition	Indication of whether the student has a tuition
final_test	Student's O-level mathematics examination score
n_male	Number of male classmates
n_female	Number of female classmates
gender	Gender type
age	Age of the student
hours_per_week	Number of hours student studies per week
attendance_rate	Attendance rate of the student (%)
sleep_time	Daily sleeping time (hour:minutes)
wake_time	Daily waking up time (hour:minutes)
mode_of_transport	Mode of transport to school
bag_color	Colour of student's bag

3. Data preparation & feature engineering

The bedrock of any machine learning project lies in the quality and relevance of the data at hand. This was vital because missing or inconsistent data could lead to biased or inaccurate predictions. Our goal was a pure dataset, devoid of such inconsistencies.

1. Values inconsistency was cleaned up for CCA and tuition columns
2. Categorical columns with 2 values are converted to 1 and 0. direct_admission 1 for yes 0 for no, learning_style 1 for visual 0 for auditory, gender 1 for female 0 for male, tuition 1 for yes 0 for no. This will reduce input_shape for NN.
3. student_id and index columns are dropped
4. final_test and attendance_rate rows with NA values are dropped
5. add_support was created, 1 if final_test is below 65 and 0 for >= 65.
6. sleep_duration measured in hours was created using sleep_time and wake_time.
7. age has outliers (5,6,-4,-5) removed.
8. attendance_rate has steep drop off at 89, hence all 89 are below are grouped together
9. sleep_time has imbalance from 23.30 onwards, hence all 23.30 are grouped together
10. sleep_duration was highly unbalanced hence removed for machine learning

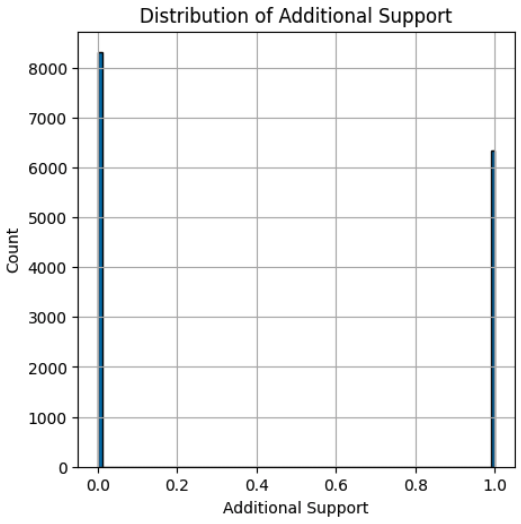
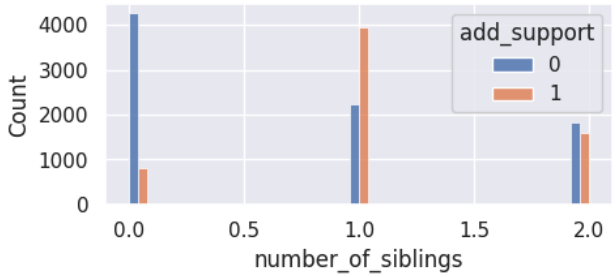
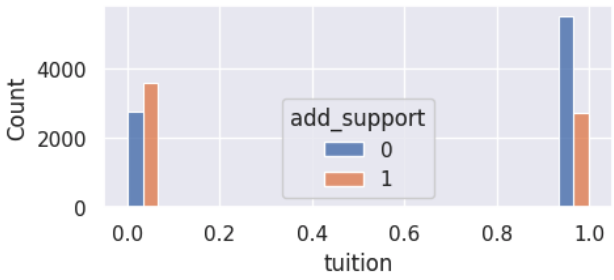
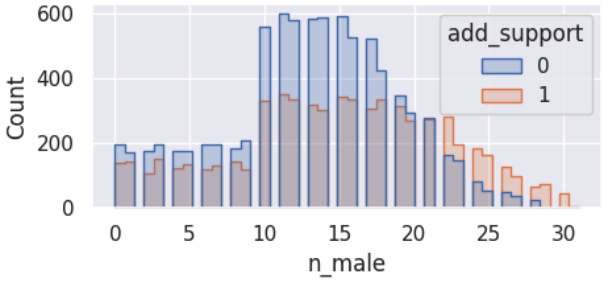


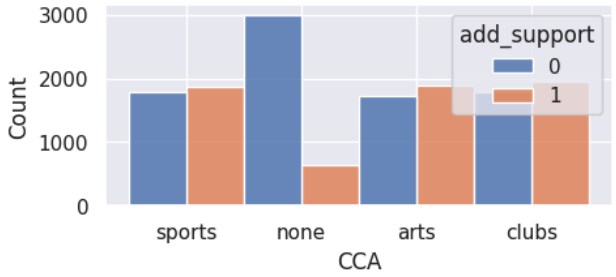
Figure 1 Additional Support distribution after cleanup

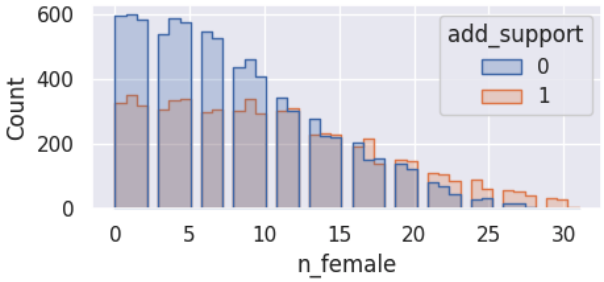

add_support
0
1
Count
number_of_siblings

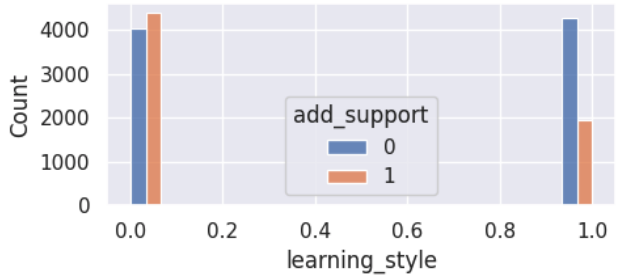

add_support
0
1
Count
tuition

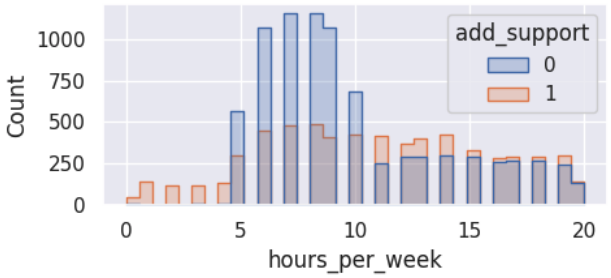

add_support
0
1
Count
direct_admission

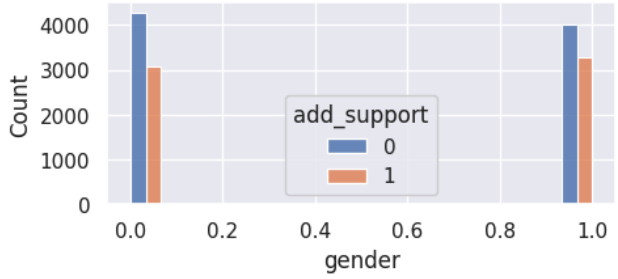

add_support
0
1
Count
n_male

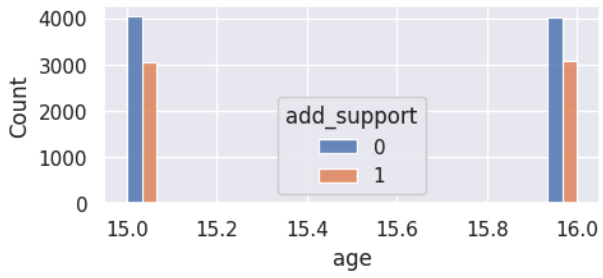

add_support
0
1
Count
sports
none
arts
clubs
CCA


add_support
0
1
Count
n_female


add_support
0
1
Count
learning_style


add_support
0
1
Count
hours_per_week


add_support
0
1
Count
gender


add_support
0
1
Count
age

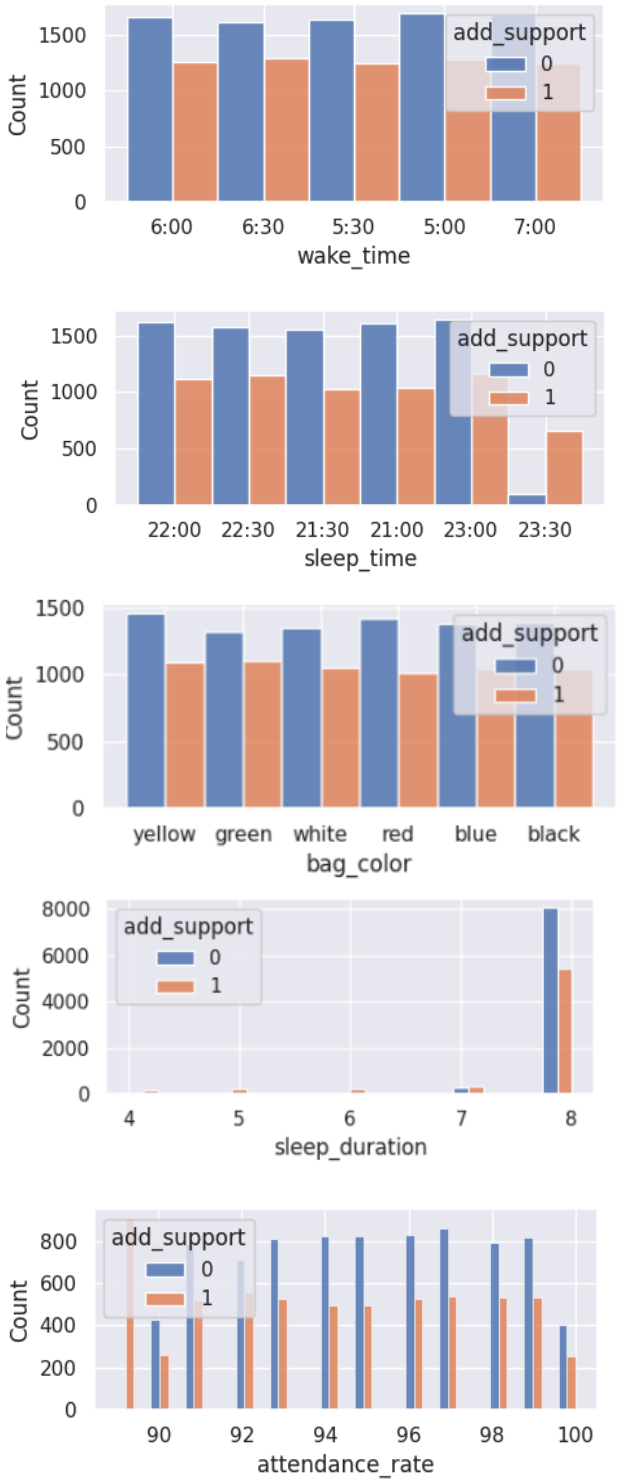


Figure 2 cleaned up dataset, Additional Support distribution per features

No other significant imbalance spotted in the dataset.

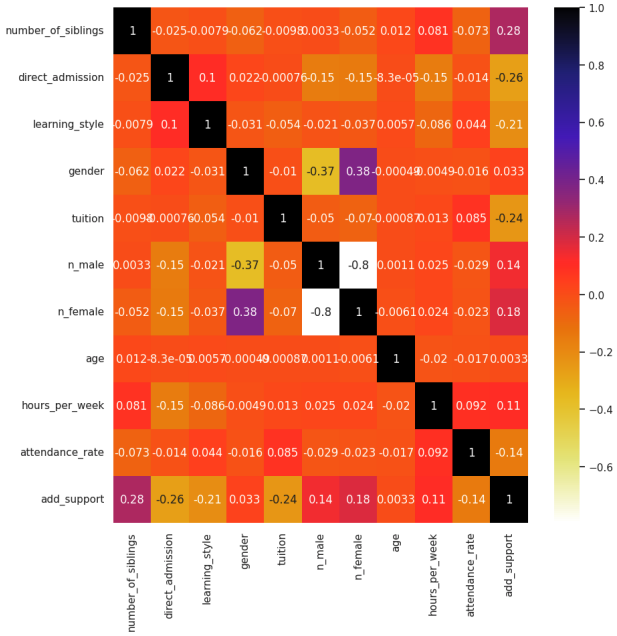


Figure 3 Heatmap to visualize correlation and multicollinearity

add_support	1.000000
number_of_siblings	0.276072
n_female	0.181618
n_male	0.137365
hours_per_week	0.106389
gender	0.032656
age	0.003286
attendance_rate	−0.144869
learning_style	−0.207154
tuition	−0.236190
direct_admission	−0.264201

Figure 4 Features pearson correlation score to Add_support

No significant relation or multicollinearity spotted in the dataset.

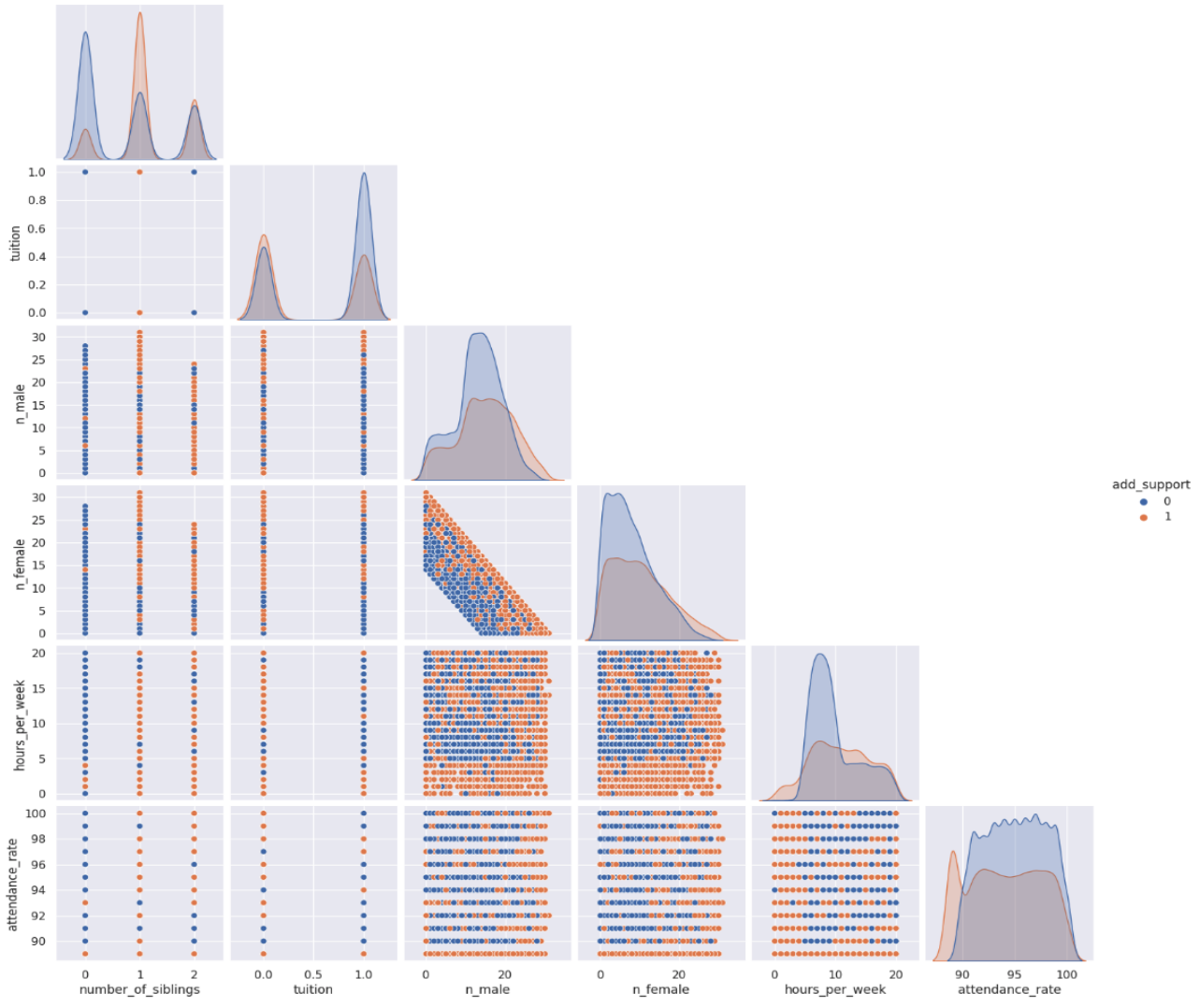


Figure 5 Relationship between numerical features

EDA Summary

- Target label add_support is considerably balanced. acceptable for modeling
- Those with 1 sibling have a higher proportion of requiring additional support, and those with 0 siblings do not require additional support.
- Those with lesser than 90 attendance_rate has higher proportion of requiring additional support
- Those with higher N_female and N_Male has higher proportion of requiring additional support
- Those with studies hours per weeks less than 5 has higher proportion of requiring additional support
- Those sleep on 23.30 and after has higher proportion of requiring additional support
- There seems to be a relationship between N_female and N_Male.
- There seem to have relation between number_of siblings,learning_style, tuition, direction_admission and add_support
- No significance multicollinearity observed among the features
- No significance imbalance of features observed
- Removed outliers records: age, 419 rows were removed

Data preparation for machine learning

In summation, the dataset is considerably balanced. acceptable for modeling

Final Dataset used, 14229 records with 14 features. Categorical features: 'CCA', 'mode_of_transport', 'bag_color','sleep_time','wake_time' encoded using oneHotEncoder to enhance machine learning.

direct_admission, learning_style, gender, tuition are in binary value and is suitable to machine learning

Numerical features : 'number_of_siblings', 'n_male','n_female','age','hours_per_week','attendance_rate' encoded using standardScaler, to enhance machine learning

Dataset is splitted into 3 categories, Train(10672), Validation(2489) and test(1068) for each ML modeling and NN modeling. After encoding 34 features and 1 target

---

4. MODELING AND EXPERIMENTS

4.1 Heuristic Modeling result

ML Random forest with the best param after Gridsearch.

```
best_params:  {'max_depth': 15, 'n_estimators': 100}
best_estimator:  RandomForestClassifier(max_depth=15,
Accuracy: 0.82
Precision: 0.81
Recall: 0.78
F1 Score: 0.79

Classification Report:
              precision    recall  f1-score   support

         0.0       0.83      0.86      0.84      1409
         1.0       0.81      0.78      0.79      1080

    accuracy                           0.82      2489
   macro avg       0.82      0.82      0.82      2489
weighted avg       0.82      0.82      0.82      2489
```

Figure 6 ML Random forest initial result

Our problem requires a better Recall score, it is costlier for False Negative to happen. hence changing threshold to 0.45 to have an increased recall while maintaining the same f1 score.

```
best_params:  {'max_depth': 15, 'n_estimators': 100}
best_estimator:  RandomForestClassifier(max_depth=15,
Accuracy: 0.82
Precision: 0.77
Recall: 0.83
F1 Score: 0.80
New threshold 0.45

Classification Report:
              precision    recall  f1-score   support

         0.0       0.86      0.82      0.84      1409
         1.0       0.77      0.83      0.80      1080

    accuracy                           0.82      2489
   macro avg       0.82      0.82      0.82      2489
weighted avg       0.82      0.82      0.82      2489
```

Figure 7 ML Random forest with 0.45 threshold

This random forest model will be used as heuristic to compare with NN model

4.2 Neural Network Modeling result

Used Keras tuner to find the best initial hyperparameter and fine tune it further. As this is a classification problem, the hyper parameters used are classification related.

NN tuner model hyper parameter list, optimizer['adam', 'rmsprop'], hp_learning_rate[min_value=1e-5, max_value=1e-1, sampling='log'], batch_size[min_value=32, max_value=512, step=32], hidden layers [min_value=1, max_value=4)], neurons per layers[min_value=32, max_value=512, step=32], final layer's sigmoid activation for range between 0 and 1, compile layer, binary_crossentropy for minimizing loss and maximizing Recall for performance metrics tracking.

L2 regularizer was used to reduce overfitting.

Early stopping with patience of 10, to hasten the run time. learning scheduler with patience of 5 before reducing the learning rate.

```
Best val_recall So Far: 0.9342592358589172
```

```
Best Hyperparameters:
{'optimizer':                  'adam',
'learning_rate':        0.0476950377,
'batch_size':  288,  'num_layers':  3,
'units_0':   448,    'units_1':   160,
'units_2':   352,    'units_3':   192,
'tuner/epochs':                  3,
'tuner/initial_epoch':           0,
'tuner/bracket': 3,'tuner/round': 0}
```

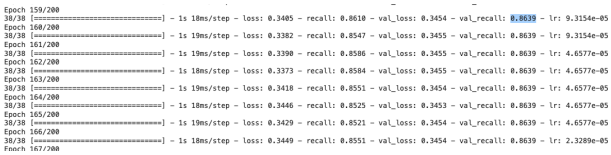
Created another NN model using the best hyperparameters of number of layers, neuron per layer, initial learning rate, batch size and optimizer.

Dropout layer of 0.5 was added to the model together with L2 regularizers to assist with overfitting.

learning rate scheduler was included, it does improve the model as epoch increase

early stopping was removed, as it prevented the model from reaching global minimum.

Model was fitted with 200 epochs, it reach peak validation dataset recall at 160th epoch, 0.8639



Test dataset Recall at 0.8638 is good hence no threshold adjustment.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_32 (Dense)             (None, 448)               15680

dropout_15 (Dropout)         (None, 448)               0

dense_33 (Dense)             (None, 160)               71840

dropout_16 (Dropout)         (None, 160)               0

dense_34 (Dense)             (None, 352)               56672

dropout_17 (Dropout)         (None, 352)               0

dense_35 (Dense)             (None, 1)                 353

=================================================================
Total params: 144545 (564.63 KB)
Trainable params: 144545 (564.63 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None

Evaluation Results:
Test Recall: 0.8638888888888889
Test Precision: 0.7920203735144312
Test F1 Score: 0.8263950398582817
```

Figure 8 NN sequential model tuned after auto tuner hyper parameters

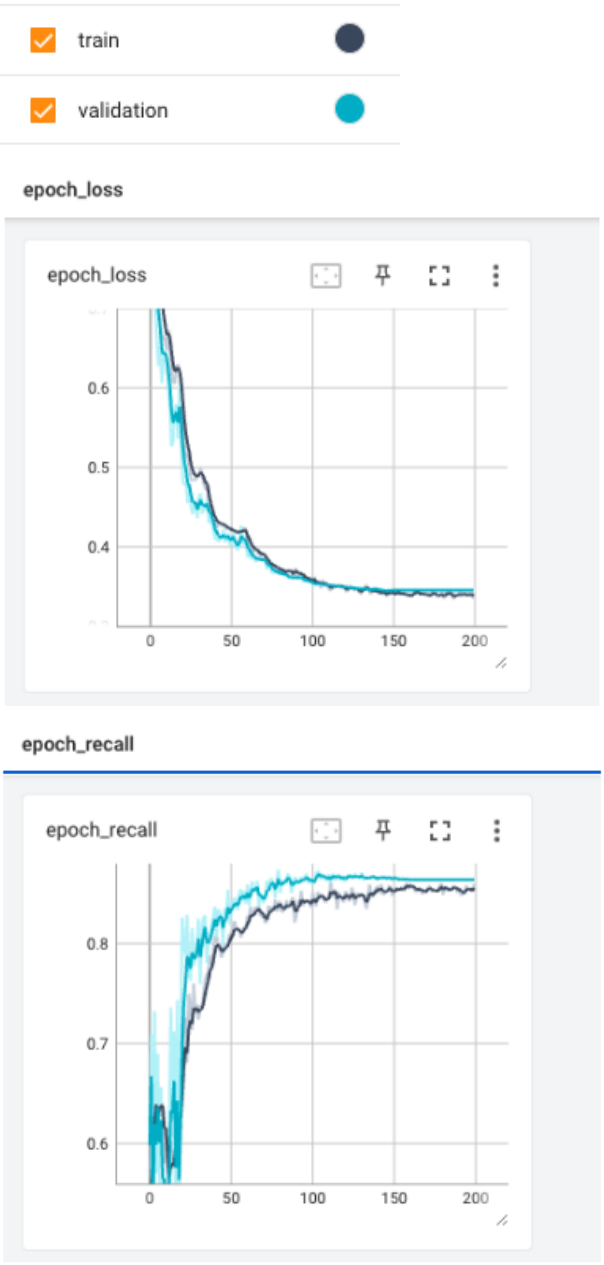


Figure 9 NN sequential model with 0.4 threshold

The training loss consistently decreases over epochs. The model is learning well from the training data, and the weights are adjusting to reduce loss.

Both training and validation recall are high and similar. The model is learning well from the training data and generalizing effectively to the validation set. This is a desirable scenario and deemed as good model performance.

4.3 McNemar's T-Test if there are significant difference

	Test 2 positive	Test 2 negative
Test 1 positive	a	b
Test 1 negative	c	d

a	864
b	26
c	68
d	110
Test type	Standard McNemar's test
Test statistic χ^2	18.766
p-value	0.00001477816

Your results are significant at the standard significance level of 5%! You can reject the null hypothesis and accept the alternative hypothesis.

Test 1- random forest model

Test 2 - Feed Forward Neural Network

Conclusion: There is a statistically significant difference between those 2 models. The Feed Forward Neural Network performed better.

5. SHAP for responsible AI uses

Both classical machine learning and Neural networks could be deemed as "black box", ongoing machine learning bias & fairness, ethical consideration and transparency are concerns of regulatory bodies.

We will utilize SHAP to interpret features' importance in machine learning. advantage lies in its simplicity; it can turn complex data scripts into shareable and interactive web applications with minimal effort, making it an ideal choice for our objectives.

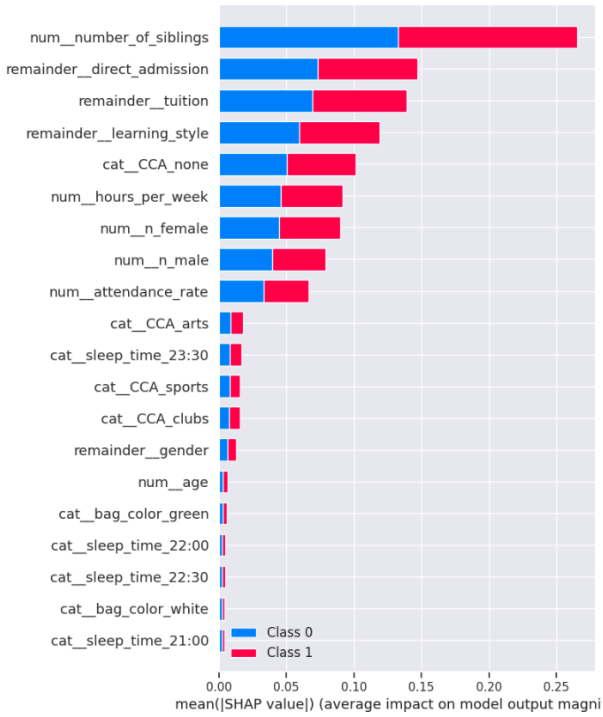


Figure 8 features and bias for classifier outcome.

There is no bais feature.

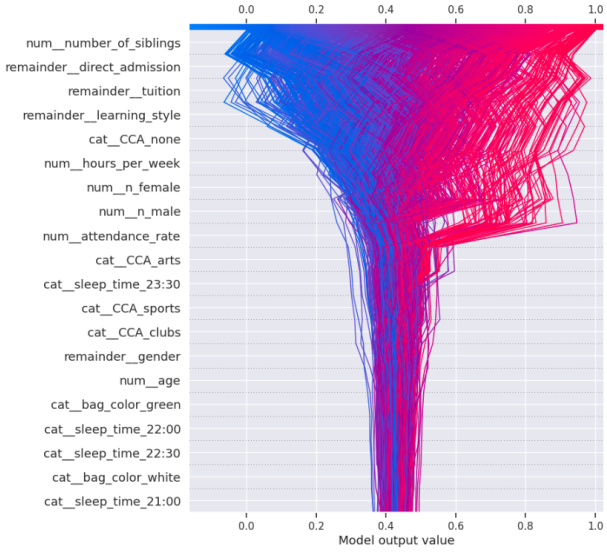


Figure 9 features importance in machine learning.

Number of siblings has a high impact if students require support to prepare for the examination.

6. CONCLUSION

This project successfully addressed the goal of predicting student scores based on a set of characteristics. The comparison between the Random Forest model and the Feedforward Neural Network revealed a statistically significant advantage for the neural network, indicating its superior performance in this predictive task. The ability to accurately identify students in need of additional study support has important implications for educational institutions. However, it's crucial to recognize the limitations of our study, including that prediction probability is at 86% accurate. Future work could explore further tuning of the Feed Forward Neural Network. Overall, this project demonstrates the potential of machine learning in enhancing educational outcomes and offers valuable insights for educators and policymakers.

7. REFERENCES

[1] data source: https://techassessment.bob.core.windows.net/aiap-preparatory-bootcamp/score.db
[2] project challenge: https://github.com/aisingapore/AIAP-Technical-Assessment-Past-Years-Series/tree/main/StudentScorePrediction
[3] https://www.tensorflow.org/tutorials/keras/keras_tuner
[4] https://en.wikipedia.org/wiki/Academic_grading_in_Singapore
[5] https://keras.io/guides/keras_tuner/visualize_tuning/
[6] https://www.datacamp.com/tutorial/introduction-to-shap-values-machine-learning-interpretability
[7] https://www.yourdatateacher.com/2021/05/17/how-to-explain-neural-networks-using-shap/
[8] McNemar’s test. https://towardsdatascience.com/mcnemars-test-to-evaluate-machine-learning-classifiers-with-python-9f26191e1a6b#:~:text=McNemar's%20test,have%20a%20lot%20of%20data.
[9] responsible AI use. https://www.microsoft.com/en-us/ai/principles-and-approach/