The Magic of the Sorting Hat - A Journey from Hogwarts to Data Analytics

Irdina NurBalqis Binti Abdul Rashid[1][030918-10-1276] and Tuan Hanis Naisha Binti Tuan Zaimi[2][030826-10-1818]

[1,2] Northern University of Malaysia, Sintok 06010 Bukit Kayu Hitam, Kedah
irdina_nurbalqis@soc.uum.edu.my, tuan_hanis_naisha@soc.uum.edu.my

Abstract. The Sorting Hat is a legendary artifact from the magical world of Harry Potter that divides students into four houses at Hogwarts School of Witchcraft and Wizardry according to their characteristics and personalities. For students at Universiti Utara Malaysia (UUM), this project uses data analytics to simulate the Sorting Hat's sorting procedure. In order to create a predictive model, we gather and examine student information, engineer pertinent elements, and use machine learning techniques. The model is evaluated for accuracy in placing pupils into one of four Hogwarts house-inspired comparable categories: Gryffindor, Hufflepuff, Ravenclaw, and Slytherin. This is done through training and testing. Developing a strong and trustworthy instrument that can forecast house assignments and reveal information about the capabilities and personalities of students is the ultimate goal. This creative method not only shows the value of data analytics, but it also infuses instructional data science with a little magic.

Keywords: Predictive Modeling, Student Classification, MLP Neural Networks, Naive Bayes, Random Forest, Harry Potter.

1 Introduction

The enthralling aspects and cherished characters of the Harry Potter universe have mesmerized millions of people. An enigma at Hogwarts School of Witchcraft and Wizardry, the Sorting Hat places incoming students into one of four houses Hufflepuff, Ravenclaw, Slytherin, and Gryffindor according to their traits and personalities. It is one of the most fascinating features of this world. Our initiative, which employs data analytics to place students at Universiti Utara Malaysia (UUM) into comparable residences, was inspired by this made-up sorting procedure and aims to add a little enchantment to the real world.

This work is significant because it has the ability to improve learning outcomes by creating customized learning environments. Understanding each student's distinct qualities allows educational institutions to design specialized support networks that encourage engagement and growth. Our goal is to create a prediction algorithm that reliably places students in house groups and offers information about their aptitudes and personalities.

In order to do this, we gathered a large amount of student data and used machine learning methods to mimic the Sorting Hat's process. Our task was to solve a technological problem in which strong feature engineering and model training were needed to accurately classify pupils based on a variety of intricate data points. We want to show how valuable data analytics can be in educational settings and investigate the creative nexus between fiction and practical applications by conquering these obstacles.

1.1 **Background**

The concept of categorizing people according to their personal characteristics has important practical ramifications in a number of domains, such as psychology, education, and human resources. Understanding student traits in the classroom can result in more personalized instruction and encouraging surroundings, which can eventually improve student performance.

The OCEAN (Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism) model, which is well-known in psychological research, is one of the well-established personality evaluation techniques that our project depends upon. To construct a comprehensive dataset, we also included academic performance, hobbies, behavioral tendencies, and other personal factors.

Artificial intelligence's machine learning field has shown success in categorization problems by using previous data to train algorithms to predict results based on input features. Several machine learning approaches were used in this study, such as Random Forest, Naive Bayes, and Multilayer Perceptron (MLP) neural networks. These techniques were selected due to their efficacy in generating precise classifications and their capacity to manage intricate, multi-dimensional data.

We used a survey to collect data from UUM students, covering a wide range of attributes, in order to build our model. In order to identify valuable characteristics from the raw data and use them to train our machine learning models, feature engineering was utilized. By assessing the models' output, accuracy and dependability in housing assignment prediction were guaranteed.

By combining these techniques, our project not only demonstrates the effectiveness of data analytics but also adds a magical touch to the field of educational data science by showing how innovative ways may produce insightful knowledge about the personalities and strengths of students.

2 Data

2.1 Source of Data

Our data was collected from 1 July 2024 until 8 July 2024 through Google Form survey with 151 respondents equivalent to 151 datasets. It was based on these questions, aimed among UUM students and were tested based on their characteristics such as:

- Personality traits: Based on the OCEAN test's result, the students' are required to choose Openness, Conscientiousness, Agreeableness, Extraversion and Neuroticism.
- Behavioral traits: This question aimed to determine most often how students' usually behave either Risk-Taking, Collaboration, Discipline or Independence.
- Hobbies: Based on their hobbies, students' are required to choose either, Physical, Cerebral, creative, Community Activities, Collecting or Making & Tinkering.
- Academic Performance: Based on their current CGPA, students are required to key in their latest academic pointer.
- Hometown: Based on the states they were born, grew or currently live in, students are required to choose one (Selangor, Kedah, Sabah etc.).
- Inasis: Based on the current UUM's residential college, students are required to choose one (BSN, MISC, Proton etc.)
- Leadership: Based on their activeness outside of class activity and the position they are holding or used to hold, students are required to choose Yes or No.
- Favorite Cuisine: Based on students' preferred choice of food, they are required to choose one (Malay, Chinese, Indian etc)
- Family Income: Based on students' parents' estimated income, they are required to key in in RM format.
- Number of Bestfriends on Campus: Based on the number of the students' best friends, they are required to key in how many they have
- Academic Programmes at UUM: Based on their courses' faculty names, students are required to choose one (SOC, TISSA, SBM etc.)

- Hogwarts House: Based on "The Harry Potter Night Sorting Hat quiz" test's result, the students' are required to choose the Hogwarts House that was assigned to them. (Ravenclaw, Hufflepuff, Gryffindor or Slytherin).

2.2 **Data Cleaning**

In this phase, our aim is to reduce the number of unique data produced while also keeping data's integrity. Therefore, we will only amend manually data for the Co-Curriculum part. This is due to the sparsity of the data input such as 'PERFORMING ARTS', 'performing arts' and 'seni lakonan' which refers to the same Co-Cu, Performing arts. Therefore, we will class each of the data by assigning them to their respectives factions according to Pusat KoK UUM Official website. After the categorization, all data will be encoded to numerical values representing their own class. This processes the data to the format that the neural network can accept, even though it is less indispensable for certain algorithms such as Naive Bayes compared to MLP. While also handle the missing values using `SimpleImputer` of scikit - learn by sub most frequent values into the missing place.

Table 1. Co-curriculum data encoding.

Initial Answer	After Categorize	After Encode
Painting Islamic Calligraphy (Khat) Malay Dance Photography Technology Landscaping Video Technology Graphics Technology Journalism	arts & creativity	0
Painting Islamic Calligraphy (Khat) Malay Dance Acting	visual & performing arts	0
Community Development Family Development Volunteerism	Community Services	1
Malaysian Red Crescent Society (PBSM) Relasis Bridgade St. John Ambulance Kor SISPA Fire & Rescue Brigade	emergency response	3

Siswaniaga Interior Design Tailoring Catering Franchise Veterinary Horticulture Aquaculture	Entrepreneurship & Agribusiness	4
Leadership Peer Mentoring Spirituality	Leadership Skills	5
Silat olahraga Karatedo Silat cekak Taekwondo Silat gayong	Martial Arts	6
Outdoor Recreation Scout Vehicle Maintenance Kor Siskor	Mechanical/Survival Skills	7
Brassband Orchestra Bagpipe Caklempong Angklung Keroncong Gamelan	Music	8
Sports Science Sports & Recreation Management Golf Badminton Table Tennis Archery Volleyball Swimming Cycling Futsal Squash	Sports	9
Palapes Suksis	Uniform Bodies	10

Club Related	Club	12

Table 2. Personality traits encoder

Features' Question Answer Selection	After Encode
Agreeableness	0
Openness	1
Conscientiousness	2
Neuroticism	3
Extraversion	4

Table 3. Behavioral traits encoder

Features' Question Answer Selection	After Encode
Risk-Taking	0
Collaboration	1
Discipline	2
Independence	3

Table 3. Hobbies encoder

Features' Question Answer Selection	After Encode
Physical	0
Cerebral	1
Creative	2
Community activities	3

Collecting	4
Making & Tinkering	5

Table 4. Leadership skills encoder

Features' Question Answer Selection	After Encode
Yes	0
No	1

Table 5. Favorite cuisine encoder

Features' Question Answer Selection	After Encode
Malay	0
Chinese	1
Indian	2
Western	3
Japanese	4
Korean	5
Thai	6
Exotic	7

Table 6. Faculty in UUM encoder

Features' Question Answer Selection	After Encode
SEFB	0
SOC	1

STML	2
IBS	3
SBM	4
SOB	5
TISSA	6
SOL	7
STHEM	8
SOIS	9
SQS	10
SOG	11
SOE	12
SMMTC	13

Table 7. Hogwarts house encoder.

Features' Question Answer Selection	After Encode
Gryffindor	0
Hufflepuff	1
Ravenclaw	2
Slytherin	3

Table 8. Academic performance encoder

Features' Question Answer Selection	After Encode	Encoding Meaning
CGPA >= 3.67 and CGPA <= 4.00	0	First Class Honours
CGPA >= 3.00 and CGPA < 3.67	1	Second Class Upper

CGPA >= 2.00 and CGPA < 3.00	2	Second Class Lower
CGPA < 2.00	3	Fail

Table 9. Estimated income encoder

Features' Question Answer Selection	After Encode	Encoding Meaning
Income > 11820	0	T20
Income >= 5250 and Income < 11820	1	M40
Income < 5250	2	B40

Table 10. Number of friends encoder

Features' Question Answer Selection	After Encode	Encoding Meaning
Friends Number > 5	0	Many
Income >= 3 and Income < 5	1	Medium
Friends Number < 3	2	Low

Table 11. Malaysia regional state encoder

Features' Question Answer Selection	After Encode	Encoding Meaning
Kedah Perlis Pulau Pinang Perak	0	Northern Region
Kelantan Terengganu Pahang Kuantan	1	Eastern Region
Selangor Kuala Lumpur Putrajaya Negeri Sembilan Melaka	2	Central Region

Johor	3	Southern Region
Sabah Sarawak Labuan	4	Borneo

Table 12. UUM residential college encoder

Features' Question Answer Selection	After Encode	Encoding Meaning
MAS MAYBANK TNB TRADEWINDS PROTON	0	Lane A
PETRONAS SIME DARBY TM" GRANTT MISC BSN	1	Lane B
YAB MUAMALAT	2	Lane C
BANK RAKYAT SME BANK	3	Lane D
SISIRAN TAMAN UNIVERSITI Rumah Sendiri (own house)	4	Lane F/Other

3 Methods

Data preparation also involves feature extraction from the dataset to obtain the independent variable and the dependent variable. Here is where we determined which column section of features we would like to be involved in for house prediction. For this project, after several tries and errors by using various methods including heatmap and chi-square, the best result would be to take all except for "Timestamp" and "From Who".

The primary issue identified is the imbalance in the data gathered for different classes. For instance, in the 'Hometown' column, even after classification, there is a significant bias towards the northern region being the most answered compared to other regions. This might cause the system to learn and catch info unrelated to the intended used model. To address this problem, we will handle it by using the Synthetic Minority Oversampling Technique (SMOTE).

Even though the technique was applied intended to address the imbalance in the minority class data sets, avoiding biases in training and potentially increasing accuracy by increasing the amount of low numbered data. The downside need to take note is that it doesn't consider the majority class while creating synthetic samples. By using SMOTE, it is as if we are gambling with the data. Either it will improve our accuracy or make it worse.

3.1 **Random Forest**

"Random Forest": RandomForestClassifier(n_estimators=7, criterion='entropy', random_state=7),

Random Forest is another technique of the group method for classification and regression analysis developed for use in data mining. During training, it evolves an enormous number of decision trees and uses them for prediction by integrating outcomes of these trees. Every tree in the forest works on the different subset of the data which is derived from the boot strap—the process in which data is sampled in a random fashion with replacement. This helps in controlling variance the aspect of overfitting is addressed since individual trees are less influenced with each other.

The Random Forest model is initialized with specific parameters: n_estimators = 7 mean that the given forest should contain 7 trees, criterion = ‘entropy’ means that the quality of splits is calculated by entropy, random_state =7 to make the different conditions reproducible give the seed for the random number generator. After that the classifier is trained on the training data using the fit method and the output of the predict method is obtained on the test set.

The model’s outcome is analyzed based on the various measures like accuracy score, classification report and confusion matrix. Decision maker also plots the ROC curves in order to analyze the performance of the classifier. The Random Forest Classifier optimizes itself to make credible decisions for classification problems by aggregating all the individual trees’ decisions with all votes in support of a certain classification. This work addresses the problem of overfitting and, in general, exhibits high potential in classification problems.

3.2 **Naive Bayes**

"Naive Bayes": GaussianNB(),

Naïve Bayes is another probabilistic classifier based on Bayes' formula, which determines the probability of a class label against the features. To express it in this framework, P(A|B) gives the posterior probability that class label A has occurred given that the features B have occurred. Bayes' Theorem is expressed as the following;

$$P(A|B) = (P(B|A) * P(A)) / P(B)$$

Such an expression shows that the prior probability of the class P(A) is the likelihood of the features given the class P(B|A) With P(B) being the marginal likelihood of the features. In the case of the NB classifier, the set A is the class label (e.g Hogwarts House) and B contains the features (e.g Personality traits and Hobbies).

Naive Bayes on the other hand makes the assumption that for any two features, they are statistically independent of each other given the class label. This simplifies the computation since the joint probability of the features can be arrived at by multiplying the individual probability values. Even though this tends to be the case when theorists are coming up with such assumptions, there is the likelihood that in real data, features could be related. The classifier estimates prior probabilities for each class from experiences in the training data and likelihoods statistically. For the continuous features, Gaussian Naive Bayes assumes a normal distribution; hence, the Gaussian Naive Bayes is implemented using the `GaussianNB` variant.

The `predict` class from the GaussianNB classifier is initialized and trained using the `fit` method respectively while the predictions are made with the help of `predict` method. A wide variety of classification evaluation methods including accuracy score, classification report and the confusion matrix are used to determine the efficiency of the classifier.

3.3 **MLP Neural Nets**

"MLP Neural Nets": MLPClassifier(solver='adam', activation='relu', alpha=1e-05, tol=1e-04, hidden_layer_sizes=(6,), random_state=1, max_iter=1000)

Multilayer Perceptrons are a form of feedforward ANN. They consist of multiple layers of neurons: An input layer, one or many hidden layers and an output layer to set up a network. The input layer is the layer that takes in the input characteristics like 'Personality Traits', 'Behavioural Traits', and 'Hobbies'. The hidden layer is the layer, where all the real computation takes place, and which is hidden from the user point of view. Every neuron within these layers is connected with neurons of the previous and/or subsequent layers in order to accommodate complex interactions and relationships. The final layer of the network is designated for giving the final result and the number of neurons in this layer equals the number of classes of the classification problem.

Activation function plays a significant role in the MLPs because it allows the introduction of nonlinearity into the model allowing it to learn complex patterns.

Some of the activation functions are ReLU, Sigmoid, Tanh; where ReLU maps the output as the input, Sigmoid maps the output between 0 and 1 and Tanh maps the output between -1 and 1. They are useful in capturing high-order structures and increase the accuracy of the model.

This is the process through which data is passed through the network from layer to another in the forward manner. In each neuron the weighted average of the input is calculated, passed through the activation function and transferred to the next layer. It is performed up to the last layer of a network which is the output layer of the neural network. Specifically, backpropagation is the training method where the network modifies its weights with regards to the error in the network's output. This enables the calculation of the loss (or error) and the network adjusts the weights to a value that minimizes the loss using trade models such as gradient descent.

Several strategies are applied to the modeling process to improve the training. The typical SGD (Stochastic Gradient Descent), Adam, RMSprop alter the weights in a gradual manner in order to decrease loss function metrics. Here, Adam is used because of the efficiency and effectiveness for cases with sparse gradients.

The MLPClassifier is set with parameters including solver = 'adam' for the optimization algorithm's preference to be Adam, activation = 'relu' for ReLU activation, and finally hidden_layer_sizes = (6,) for just one concealed layer with six units. Regularization is managed utilizing alpha = 1e-05 in an attempt to control the excessive complexity of the models. The model is initialized using the fit method in which it is trained by finding relationships and making assumptions to come up with results; the model is then tested on the test set. The presented model's performance is evaluated with various accuracy measures.

4 Evaluation/Results

This phase represents the outcomes of previous methods being implemented. Moving forward, we will analyze and discuss the impact of the good influence of SMOTE on making the data better versus not using SMOTE.

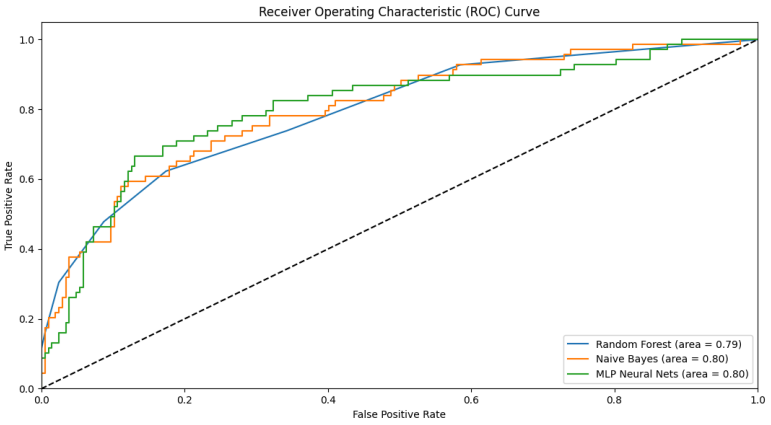

Fig. 1. Showing ROC graph for methods with SMOTE implementation

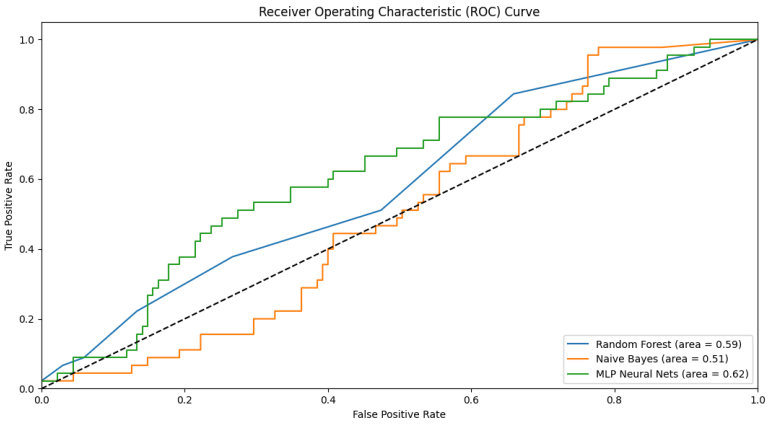

Fig. 2. Showing ROC graph for methods without SMOTE implementation

In order to depict the performance value, an ROC curve is used as shown above which depicts the relation between the true positive rate and the false positive rate of the classifiers. This illustration is useful to comprehend how effectively every classifier works.

- **Fig.1.**, Graph 1 (With SMOTE):
  - Random Forest: AUC = 0.79
  - Naive Bayes: AUC = 0.80
  - MLP Neural Nets: AUC = 0.80

- **Fig.2.**, Graph 2 (Without SMOTE):
    - Random Forest: AUC = 0.59
    - Naive Bayes: AUC = 0.51
    - MLP Neural Nets: AUC = 0.62

Applying SMOTE significantly improves model performance, as shown in **Fig.1.** where all models achieve AUC values close to 0.80. This indicates that the models are more effective at distinguishing between classes when SMOTE is applied. Without SMOTE, performance drops substantially, with AUC values around 0.51 to 0.62, highlighting the models' struggle to distinguish between classes without the synthetic balancing provided by SMOTE.

As SMOTE helps address class imbalance, without it, **Fig.2.** shows models tend to be biased towards the majority class, leading to poorer performance as indicated by the lower AUC values. The significant drop in AUC values without SMOTE suggests that the models are highly sensitive to class imbalance. Using SMOTE or other balancing techniques can help in training more robust models that perform consistently across different datasets. This underscores the importance of using techniques like SMOTE for datasets with imbalanced classes to improve the predictive power and reliability of the models.

4.1 **Evaluation/Results Analysis**

In the classification report, precision represents the accuracy of the positive predictions. Recall refers to the ability to capture all positive instances in the actual class. F1- score refers to the harmonic mean between precision and Recall. Support helps to identify imbalances among classes and evaluate each class performance, the lower the support indicates good performance.

For the Confusion matrix part, it is a type of table that is used for assessing the performance of any type of classification modeling technique. The rows of the matrix contain the instances in a predicted class while the columns contain the instances in an actual class or vice versa. The following matrices show the class assignment of the data as analyzed by each of the models.

Here, 0 refers to Gryffindor, 1 refers to Hufflepuff, 2 refers to Ravenclaw and 3 refers to Slytherin.

```
----------------------Result: MLP Neural Nets ----------------------

Accuracy score of MLP Neural Nets is 44.44%
Metric classification report: MLP Neural Nets -->
               precision    recall  f1-score   support

           0       0.43      0.55      0.48        11
           1       0.45      0.64      0.53        14
           2       0.45      0.29      0.36        17
           3       0.00      0.00      0.00         3

    accuracy                           0.44        45
   macro avg       0.33      0.37      0.34        45
weighted avg       0.42      0.44      0.42        45

Confusion Matrix: MLP Neural Nets -->
 [[6 3 2 0]
 [1 9 4 0]
 [5 7 5 0]
 [2 1 0 0]]

Process finished with exit code 0
```

Fig. 3. Showing result of MLP Neural Nets without SMOTE implementation

```
----------------------Result: MLP Neural Nets ----------------------

Accuracy score of MLP Neural Nets is 62.32%
Metric classification report: MLP Neural Nets -->
               precision    recall  f1-score   support

           0       0.59      0.56      0.57        18
           1       0.62      0.93      0.74        14
           2       0.56      0.43      0.49        21
           3       0.73      0.69      0.71        16

    accuracy                           0.62        69
   macro avg       0.63      0.65      0.63        69
weighted avg       0.62      0.62      0.61        69

Confusion Matrix: MLP Neural Nets -->
 [[10  2  4  2]
 [ 0 13  1  0]
 [ 6  4  9  2]
 [ 1  2  2 11]]
```

Fig. 4. Showing result of MLP Neural Nets with SMOTE implementation

The comparison between classification reports for **Fig.3.**, Report 1 and **Fig.4.**, Report 2. is as below,

- Overall Accuracy:
  - Report 1: 44.44%
  - Report 2: 62.32%
  - Report 2 shows a significantly higher accuracy.
- Precision, Recall, and F1-Score:
  - Gryffindor (0):
    - Report 1: Precision 0.43, Recall 0.55, F1-Score 0.48
    - Report 2: Precision 0.59, Recall 0.56, F1-Score 0.57
    - Report 2 performs better in terms of precision and F1-score.
  - Hufflepuff (1):
    - Report 1: Precision 0.45, Recall 0.64, F1-Score 0.53
    - Report 2: Precision 0.62, Recall 0.93, F1-Score 0.74
    - Report 2 has significantly better precision, recall, and F1-score.
  - Ravenclaw (2):
    - Report 1: Precision 0.45, Recall 0.29, F1-Score 0.36
    - Report 2: Precision 0.56, Recall 0.43, F1-Score 0.49
    - Report 2 again shows better performance across all metrics.
  - Slytherin (3):
    - Report 1: Precision 0.00, Recall 0.00, F1-Score 0.00
    - Report 2: Precision 0.73, Recall 0.69, F1-Score 0.71
    - Report 2 is vastly superior, indicating the model was unable to classify Slytherin in Report 1.
- Macro and Weighted Averages:
  - Macro and weighted averages for precision, recall, and F1-score are all higher in Report 2, indicating overall better model performance.

The comparison between confusion matrices for **Fig.5.**, Report 1 and **Fig.6.**, Report 2. is as below,

- Gryffindor (0):
  - Report 1: Predicted correctly 6 out of 11 times.
  - Report 2: Predicted correctly 10 out of 18 times.
- Hufflepuff (1):
  - Report 1: Predicted correctly 9 out of 14 times.
  - Report 2: Predicted correctly 13 out of 14 times.
- Ravenclaw (2):
  - Report 1: Predicted correctly 5 out of 17 times.
  - Report 2: Predicted correctly 9 out of 21 times.
- Slytherin (3):
  - Report 1: Predicted correctly 0 out of 3 times.
  - Report 2: Predicted correctly 11 out of 16 times.

Hence, we may conclude that the model performance for **Fig.4.** with SMOTE performs significantly better overall, with higher accuracy and better precision, recall, and F1-scores for all classes. **Fig.4.** also shows more balance and handles each class better, especially Slytherin, which **Fig.3.**'s model failed to classify correctly at all. From the confusion matrix we can see that Fig.4. makes fewer errors and has better classification accuracy across all classes. In conclusion, the **Fig.4.** MLP Neural Net model demonstrates better performance and reliability in classifying the four Hogwarts houses compared to the first model.

```
----------------------Result: Naive Bayes ----------------------

Accuracy score of Naive Bayes is 17.78%
Metric classification report: Naive Bayes -->
              precision    recall  f1-score   support

           0       0.50      0.09      0.15        11
           1       0.23      0.21      0.22        14
           2       0.29      0.12      0.17        17
           3       0.09      0.67      0.15         3

    accuracy                           0.18        45
   macro avg       0.28      0.27      0.17        45
weighted avg       0.31      0.18      0.18        45

Confusion Matrix: Naive Bayes -->
 [[1 4 1 5]
 [0 3 4 7]
 [1 5 2 9]
 [0 1 0 2]]
```

Fig. 5. Showing result of Naive Bayes without SMOTE implementation

```
----------------------Result: Naive Bayes ----------------------

Accuracy score of Naive Bayes is 59.42%
Metric classification report: Naive Bayes -->
               precision    recall  f1-score   support

           0       0.42      0.44      0.43        18
           1       0.65      0.79      0.71        14
           2       0.67      0.48      0.56        21
           3       0.67      0.75      0.71        16

    accuracy                           0.59        69
   macro avg       0.60      0.61      0.60        69
weighted avg       0.60      0.59      0.59        69

Confusion Matrix: Naive Bayes -->
 [[ 8  3  4  3]
 [ 3 11  0  0]
 [ 5  3 10  3]
 [ 3  0  1 12]]
```

Fig. 6. Showing result of Naive Bayes without SMOTE implementation

The comparison between classification reports for **Fig.5.**, Report 1 and **Fig.6.**, Report 2. is as below,

- Overall Accuracy:
  - Report 1: 17.78%
  - Report 2: 59.42%
  - Report 2 shows a significantly higher accuracy.
- Precision, Recall, and F1-Score:
  - Gryffindor (0):
    - Report 1: Precision 0.50, Recall 0.09, F1-Score 0.15
    - Report 2: Precision 0.42, Recall 0.44, F1-Score 0.43
    - Report 2 performs better in terms of recall and F1-score.
  - Hufflepuff (1):
    - Report 1: Precision 0.23, Recall 0.21, F1-Score 0.22
    - Report 2: Precision 0.65, Recall 0.79, F1-Score 0.71
    - Report 2 has significantly better precision, recall, and F1-score.
  - Ravenclaw (2):
    - Report 1: Precision 0.29, Recall 0.12, F1-Score 0.17
    - Report 2: Precision 0.67, Recall 0.48, F1-Score 0.56
    - Report 2 shows better performance across all metrics.
  - Slytherin (3):
    - Report 1: Precision 0.09, Recall 0.67, F1-Score 0.15
    - Report 2: Precision 0.67, Recall 0.75, F1-Score 0.71
    - Report 2 performs better in terms of precision and F1-score.

- Macro and Weighted Averages:
  - Macro and weighted averages for precision, recall, and F1-score are all higher in Report 2, indicating overall better model performance.

The comparison between confusion matrices for **Fig.5.**, Report 1 and **Fig.6.**, Report 2. is as below,

- Gryffindor (0):
  - Report 1: Predicted correctly 1 out of 11 times.
  - Report 2: Predicted correctly 8 out of 18 times.
- Hufflepuff (1):
  - Report 1: Predicted correctly 3 out of 14 times.
  - Report 2: Predicted correctly 11 out of 14 times.
- Ravenclaw (2):
  - Report 1: Predicted correctly 2 out of 17 times.
  - Report 2: Predicted correctly 10 out of 21 times.
- Slytherin (3):
  - Report 1: Predicted correctly 2 out of 3 times.
  - Report 2: Predicted correctly 12 out of 16 times.

Fig.6. shows that there is enhancement in the classifier performance where the accuracy has risen from 17. 78% to 59. 42%. It can be seen that for each house, there is a higher precision, recall, and F1-score of classifications for all classes in the **Fig.6.** than **Fig.5.**. The confusion matrix of **Fig.6.** indicates improved correctly classified instances in terms of the diagonal values as compared to the first report. All in all, we can conclude that the **Fig.6.** scenario is more suitable for the Naive Bayes classifier, as it exhibits higher accuracy and better precision, recall, and F1-scores for all classes.

```
----------------------Result: Random Forest ----------------------

Accuracy score of Random Forest is 33.33%
Metric classification report: Random Forest -->
              precision    recall  f1-score   support

           0       0.21      0.36      0.27        11
           1       0.50      0.64      0.56        14
           2       0.29      0.12      0.17        17
           3       0.00      0.00      0.00         3

    accuracy                           0.33        45
   macro avg       0.25      0.28      0.25        45
weighted avg       0.31      0.33      0.30        45

Confusion Matrix: Random Forest -->
 [[4 4 3 0]
 [3 9 2 0]
 [9 5 2 1]
 [3 0 0 0]]
```

Fig. 7. Showing result of Naive Bayes without SMOTE implementation

```
----------------------Result: Random Forest ----------------------

Accuracy score of Random Forest is 55.07%
Metric classification report: Random Forest -->
              precision    recall  f1-score   support

           0       0.42      0.61      0.50        18
           1       0.30      0.21      0.25        14
           2       0.59      0.48      0.53        21
           3       0.88      0.88      0.88        16

    accuracy                           0.55        69
   macro avg       0.55      0.54      0.54        69
weighted avg       0.55      0.55      0.54        69

Confusion Matrix: Random Forest -->
 [[11  3  3  1]
 [ 7  3  4  0]
 [ 7  3 10  1]
 [ 1  1  0 14]]
```

Fig. 8. Showing result of Naive Bayes with SMOTE implementation

The comparison between classification reports for **Fig.7.**, Report 1 and **Fig.8.**, Report 2. is as below,

- Overall Accuracy:
  - Report 1: 33.33%
  - Report 2: 55.07%
  - Report 2 shows a significantly higher accuracy.
- Precision, Recall, and F1-Score:
  - Gryffindor (0):
    - Report 1: Precision 0.21, Recall 0.36, F1-Score 0.27
    - Report 2: Precision 0.42, Recall 0.61, F1-Score 0.50
    - Report 2 performs better in terms of precision, recall, and F1-score.
  - Hufflepuff (1):
    - Report 1: Precision 0.50, Recall 0.64, F1-Score 0.56
    - Report 2: Precision 0.30, Recall 0.21, F1-Score 0.25
    - Report 1 has significantly better precision, recall, and F1-score.
  - Ravenclaw (2):
    - Report 1: Precision 0.29, Recall 0.12, F1-Score 0.17
    - Report 2: Precision 0.59, Recall 0.48, F1-Score 0.53
    - Report 2 shows better performance across all metrics.
  - Slytherin (3):
    - Report 1: Precision 0.00, Recall 0.00, F1-Score 0.00
    - Report 2: Precision 0.88, Recall 0.88, F1-Score 0.88

        - Report 2 is vastly superior, indicating the model was unable to classify Slytherin in Report 1.
    - Macro and Weighted Averages:
        - Macro and weighted averages for precision, recall, and F1-score are all higher in Report 2, indicating overall better model performance.

The comparison between confusion matrices for **Fig.7.,** Report 1 and **Fig.8.**, Report 2. is as below,

- Gryffindor (0):
    - Report 1: Predicted correctly 4 out of 11 times.
    - Report 2: Predicted correctly 11 out of 18 times.
- Hufflepuff (1):
    - Report 1: Predicted correctly 9 out of 14 times.
    - Report 2: Predicted correctly 3 out of 14 times.
- Ravenclaw (2):
    - Report 1: Predicted correctly 2 out of 17 times.
    - Report 2: Predicted correctly 10 out of 21 times.
- Slytherin (3):
    - Report 1: Predicted correctly 0 out of 3 times.
    - Report 2: Predicted correctly 14 out of 16 times.

Fig.8. Random Forest model shows the rise in accuracy, precision, recall and F1-scores over most of the classes than **Fig.7.** Random Forest model. The confusion matrix also displays similar improvements with more appropriate distribution of correct classifications. However, we observe that Hufflepuff (1) has a reduced accuracy, suggesting the model may require fine-tuning for better distribution of predictions among the classes.

5 Conclusion

In this project, we used data analytics and machine learning approaches to take an inventive approach to replicating the Sorting Hat from the Harry Potter books. We gathered information from 151 UUM students between July 1 and July 8, 2024, using a Google Form survey. A wide range of qualities were addressed by the survey, such as behavioral traits, personality traits, interests, academic achievement, and more.

We collected a thorough dataset by utilizing the OCEAN personality test, evaluating behavioral patterns, and looking at a variety of academic and personal characteristics. With the use of this data and the help of SMOTE, we were able to develop a strong prediction model that was used to assign students to the Hogwarts houses of Gryffindor, Hufflepuff, Ravenclaw, and Slytherin.

We used a variety of machine learning techniques, such as Random Forest, Naive Bayes, and Multilayer Perceptron (MLP) neural networks, to obtain accurate

categorization. These techniques were selected due to their shown performance in classification tests and their ability to handle complicated, multi-dimensional data.

Our research brings a little enchantment to the world of data analytics while also showcasing its potential in educational settings. We showed how machine learning may offer insightful information about students' personalities and strengths by modeling the Sorting Hat process. This can help create more customized and supportive learning environments.

Through this project, we demonstrated the possibilities of fusing real-world data science with creative inspiration, highlighting the inventive powers of predictive modeling in comprehending and improving student experiences.

6 Reflection

This effort demonstrated the effectiveness of fusing educational and psychological ideas with data science to produce insightful classifications. We gained knowledge about the need of carefully planning features and coordinating data gathering with the objectives of the model. In retrospective terms, we wish we had set aside more time for iterative testing and looked for additional domain knowledge to improve our methodology. For upcoming students, make sure to combine technical abilities with domain expertise, prepare data thoroughly, and be open to trying out different machine learning models. By following these procedures, your predictive models will perform better and produce more precise and useful insights.

This project imitates the Sorting Hat from Harry Potter to sort the UUM students into their correct house depending on their characteristics.. Based on data gathered, we derived variables and applied Random Forest, Naive Bayes and MLP Neural Nets to classify the existing data and predict new data for which Hogwarts House they belong to. The decision to perform data cleaning, feature encoding, and a special approach to the class imbalance issue through SMOTE categorization proved to be the most effective steps to take. SMOTE favored models that presented better results overall, which indicated that differential data analysis could be beneficial in developing approaching schematic environments and as a consequence deciding the proficiency of learners. Some of the future plans include gathering more realistic student data to minimize the use of synthetic data, to make sure it reflects the real world much better.

Acknowledgments. This study’s dataset was collected by the work of the whole SKIH2103 class. It was ideated from and supervised by Associate Prof. Dr. Azizi Bin Ab Aziz. Special thanks to Shahidatul Hidayah and Khairun Nisaa for helping by providing ideas, knowledge and support throughout the whole case study.

References

1. Amin, M. Z., & Ali, A. (2017, February). Application of multilayer perceptron (MLP) for data mining in healthcare operations. In 2017 3rd International Conference on Biotechnology, Lahore, Pakistan (p. 2e11).
2. Weirich, P. (2011). The bayesian decision-theoretic approach to statistics. Philosophy of Statistics, 233–261. https://doi.org/10.1016/b978-0-444-51862-0.50007-1
3. Gouda, P., & Cormican, M. (2016). Virtual sorting hat™technology for the matching of candidates to residency training programs. Medical Education, 50(12), 1249–1252. doi:10.1111/medu.13113