Project Report

on

PREDICTING READING COMPREHENSION FROM GAZE BEHAVIOUR

Report submitted in partial fulfillment of the requirements

for the award of the degree of

Bachelor of Technology

in

Mechanical Engineering

Submitted by

Gollu Jikki Sravanthi

(19ME31010)

Under the guidance of

Prof. Manjira Sinha

CENTRE FOR EDUCATIONAL TECHNOLOGY

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

YEAR 2022-2023

DECLARATION

We certify that

a) The work contained in this report has been done by us under the guidance of our supervisor.

b) The work has not been submitted to any other Institute for any degree or diploma.

c) We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

d) Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

Date: May, 2023 (Gollu Jikki Sravanthi)

Place: Kharagpur (19ME31010)

CERTIFICATE

This is to certify that the project entitled, “Predicting Reading Comprehension from Gaze Behavior”, submitted by Gollu Jikki Sravanthi (19ME31010) is a record of bonafide research work carried out by her in the Centre for Educational Technology, Indian Institute of Technology Kharagpur under my supervision and guidance for the partial fulfillment for the award of degree of Bachelor of Technology in Mechanical Engineering during academic session 2022-2023 from Indian Institute of Technology, IIT Kharagpur.

Prof. Manjira Sinha

(Assistant Professor)

Centre for Educational technology

Indian Institute of Technology

E-mail: manjira.sinha@cet.iitkgp.ac.in

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Prof. Manjira Sinha for her tremendous help, support and guidance throughout the work. I deeply express my sincere thanks for encouraging and allowing me to work on the project under her able guidance. She has not only suggested the problem but also helped me in various other aspects of the project and provided me with the opportunities to expand my knowledge and experience. This work has been possible because of her inspiration, motivation and full freedom given to me to incorporate any ideas.

I am sincerely grateful to the Head of the Centre for Educational technology, Indian Institute of Technology, Kharagpur for providing all the necessary facilities for the successful carryout of the project.

Gollu Jikki Sravanthi

(19ME31010)

Fourth year Undergraduate student

Department of Mechanical Engineering

Indian Institute of Technology, Kharagpur

TABLE OF CONTENTS

INTRODUCTION

People generally move their eyes in a particular way while reading a text or responding to a question according to their understandability and knowledge. Individuals have a self-similar behavior, acting differently based on their understanding of the text or the complexity of a passage. These models push to decode those individuals' behavior from the readings of their eye movements like fixation duration, fixation time and area, etc. Eye movements can reflect cognitive processes like viewing, searching, reading, and thinking. Many numbers of researches are done regarding mental state from eye-tracking data. Most of them incorporated observations where people have to predict their unassigned tasks or an individual's characteristics, i.e., their personality. Few of them are concentrated on predicting their state during reading and their level of understanding of the text read. Here, supervised classification algorithms like K-nearest Neighbor, Naive Bayes, and more are used to predict a reader's difficulty level and nativity from eye fixation data and fixation durations.

Different approaches are used to classify the independent variables, reading difficulty, and whether the reader's first language is English from 4 of the given dependent variables. Fixation data is collected from eye-tracking data of 95 participants who read SAT practice passages. The participants are asked to respond to different comprehension questions followed by self-evaluation questions. Here difficulty and nativity are classified into two primary sub-groups by approximating exceptions.

The performances of the leading models: Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbor, Naive Bayes and Null Accuracy (without any model) are compared. Our main goal was to predict the reader's comprehension levels during reading from the fixation data obtained from their eye-tracking data and compare the accuracies with the original data to check the model's working. If this method proves successful, these models can help create intelligent systems that can give the mental state of a reader instantaneously without using questions and answers for evaluation. Thus, the models allow us to estimate the relationship between reading behavior and comprehension of a person from their reading patterns and explain excellent or bad understanding capability.

LITERATURE REVIEW

A large number of studies and research were done to estimate from the eye-tracking data. Boisvert and Bruce 2016 and Henderson et al. 2013 focused on predicting tasks like viewing or memorization. Al-Samarraie et al. 2017, 2018 are based on an individual's personality estimation.

It is found by Underwood et al. 1990 studies (one of the first studies to predict reading comprehension from eye tracking data) found that fixation duration helps in predicting the difficulty or comprehension level of a person reading the text. The model was trained and tested on their dataset but could not give generalizations over unknown or new datasets.

Some other studies focused on finding the literacy of an individual, Augereau et al. 2016 and Lou et al. 2017. They were able to find the literacy skill of an individual, but that is not related to our current model because that model is not based on eye movement or fixation measurements. In another study by Makowski et al. 2018, estimations of comprehension levels from eye-tracking data are done, but it still needs to produce better results.

Thus, whether a model could predict comprehension levels remains a question. It is still to be known if it can produce optimistic results. Therefore, here it is tried to classify our data into two variables, the difficulty of reading and nativity, and try to estimate the accuracies by the neural network's models.

DATA

The SAT dataset is one of the most critical and significant eye-tracking data of fixations. The eye movements of 95 participants reading four SAT passages are recorded. The data in each column includes all the information about fixation places, fixation time, rereading parts and timings, and more. It also consists of answers to comprehension and self-evaluation questions of every individual and evaluated performances. The passages, questions, and data are collected from the SAT dataset available publicly at https://github.com/ahnchive/SB-SAT. Participants were asked to read the text from the passages displayed individually to analyse the reading and answer the comprehension questions. Five minutes are given to read the passage, and no time limit is held for answering the questionnaire. The eye positions were recorded using an Eye Link 1000 sampling, and the fixations were parsed from the gaze coordinates.

Here, the models try to find the accuracy of the reader's difficulty in understanding and their nativity. Thus, the comprehension questions, the answers, and the levels of overall comprehension are not calculated. It is focused on the sequences of 21 fixation-location (x, y), fixation duration, and features of the pupil size extracted from the dataset. These recorded values are sent as inputs for the current model to predict an individual's difficulty and nativity level.

The responses of the participants decide the level of difficulty of the passages. They were asked to rate the difficulty where the passages are considered to be of a high level if the participants rated them as hard or very hard. They are considered to be of a low level if the participants rated them as easy or very easy. The final data is arranged based on whether the participant is a native speaker of English.

It is observed that people with higher understanding levels have shorter fixation durations and a faster reading rate. The people who rated the difficulty as hard tended to read slowly and had more extensive fixation durations than others. It is also observed that native speakers took smaller fixation durations and had a high reading rate. There were few changes in the observations with the pupil sizes. It showed no difference across the levels. Thus, these results jointly show the levels of understanding through different sequences of eye movements and their behaviour during reading.

PRE-PROCESSING

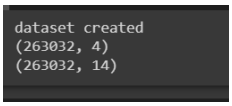
The dataset has many data samples (dependent variables) for every individual. The labels of the datasets are stored, and the corresponding information from the tables is changed from 4 levels to 2 simple levels - 0 or 1, according to the half they belong, to separate the high and low levels. The main four oculomotor features - Current fixation of X, Current fixation of y, Current fixation of Pupil, and Current Fixation Duration from each data are normalized to values between 0 and 1. The data retrieved is grouped into a window of consecutive fixations and fixation durations, without overlaps, to pass them as inputs for the model training. The dataset has to be labelled for differentiation and to address the results of every individual separately. But here, labelling could become a considerable problem if done one by one during compilation or manually, as the dataset is vast. So, the labels for the data are inherited directly from the label of the passage from which the window came and the label of that particular individual. The independent variables' difficulty and nativity values are extracted from the tables and stored in a data frame for accessing them simultaneously with the readings of the four dependent variables. It is assumed that reading at a local scale is the same as reading at a more global scale.

The data is taken in order of the individual's label and then in order of the passage they read. After slicing the four primary columns from the original eye-tracking data, they are assigned the individual's name and the passage using the function generate_window. All these values are stored as a tuple of fixations.

The predictions from the fixations are made in three different manners: prediction of new reading fixation windows, fixation windows on new passages, and fixation windows on new subjects.

Prediction on New Reading Windows (Record wise):

New windows mean the model has just seen those data. So, the model only worked on this data after, and it has to predict the output for this new set of windows. Thus, the original dataset of 2,63,032 samples is randomly divided into three proportions. The proportions are 80% for training, 10% for validation, and 10% for testing datasets.



Prediction on New Passages (Book wise):

This method is to predict if the model could generalize from new passages. This method tries to predict the comprehension level of a reader for an unseen passage based on the training from two datasets and validating through one dataset. So, the dataset is divided again into three proportions, where data from two passages, any 2 of the four, are randomly allocated for the training dataset, one passage data for validation, and one passage data is left for testing the model. Random splits are generated for different readers, so there would be less chance of two or more readers getting the same passages in the dataset.

```
list of books: ['dickens', 'flytrap', 'genome', 'northpole']
dataset created
(132447, 4)
(132447, 14)
dataset created
(64450, 4)
(64450, 14)
dataset created
(66135, 4)
(66135, 14)
train book # ['reading-dickens-1', 'reading-dickens-2', 'reading-dickens-3', 'reading-dickens-4', 'reading-dickens-5', 'reading-genome-1', 'reading-genome-2', 'reading-genome-3', 'rea
valid book # ['reading-northpole-1', 'reading-northpole-2', 'reading-northpole-3', 'reading-northpole-4', 'reading-northpole-5']
test book # ['reading-flytrap-1', 'reading-flytrap-2', 'reading-flytrap-3', 'reading-flytrap-4', 'reading-flytrap-5', 'reading-flytrap-6']
```

Prediction on New Readers (Subject wise):

This method tests whether the model could be generalized for new readers. The preliminary dataset of 95 participants is again split into three proportions. 80% of 95 participants, i.e., 76 participants' windows, are included in the training dataset, 10%, i.e., 10 participants' windows in the validation dataset, and the rest, 9 participants' windows in the testing dataset. The comprehension levels for the unseen 9 participants are predicted after training.

```
train subj # 76
valid subj # 10
test subj # 9
dataset created
(209557, 4)
(209557, 14)
dataset created
(29256, 4)
(29256, 14)
dataset created
(24219, 4)
(24219, 14)
```

The values in every dataset are joined together into an array, and thus, three new datasets are formed using the function create_dataset. The information of every data and their labels are stored in three data frames to access whenever needed. The models for the datasets are saved, and the data frames are copied into a CSV file for future purposes.

MODEL

After inputting the datasets for training, validation, and testing, two levels of a predicted variable, a high or low level of comprehension are found and compared for accuracies and errors.

Null Model:

The Null model is calculated for a lower bound. It simply returns the outputs based on the weights of the classes, i.e., how frequently the required class occurs in a dataset. It calculates the weights of each class (0 or 1) in the training dataset and outputs them along with the loss weight. The null accuracy is calculated from the maximum occurrence of the weights in the testing dataset.

Logistic Regression:

Logistic regression is a statistical model, commonly used to analyze binary outcomes. In the context of reading comprehension, logistic regression can be used to predict whether a reader has difficulty in understanding the comprehension, etc., based on their gaze behavior while reading the text. The model is loaded and used to predict the validation and testing accuracies.

Decision Tree:

A Decision Tree is a machine learning model and a supervised learning technique used for classification and regression problems but is mainly preferred in classification problems. The classifier is a tree structure, where internal nodes represent features, branches represent the decision taken, and each leaf node represents the outcomes. Here, the classifier is used to predict the accuracies. Fine-tuning is done to find the best parameters to avoid over-fitting or under-fitting models for the dataset. The max_depth and min_samples_split are derived.

Random Forest:

A Random Forest classifier is a machine learning algorithm and a supervised learning technique used for both classification and regression problems. It uses the concept of ensemble learning where multiple classifiers are combined to solve a complex problem and improve its performance. It contains a number of decision trees and different subsets and takes average to improve the predicted accuracy. Here, the classifier is used to predict the

accuracies and fine-tuning is done to improve the accuracy scores using the validation dataset. The parameters n_estimators, max_depth, min_samples_split, min_samples_leaf and random_state are derived.

K-Nearest Neighbor:

K-Nearest Neighbor is a machine learning algorithm that uses a supervised learning technique to assume similarity between available data and new data and classifies the new data into a category most similar to the available ones. It can be used for regression and classification problems but primarily for classification. It takes time to learn from the training set. Instead, it stores the dataset and performs an action on the dataset at the time of classification. Here, the classifier is used to predict the accuracies, and fine-tuning is done to improve the accuracy scores. The parameters n_neighbors is derived.

Naive Bayes:

The Naïve Bayes algorithm is a supervised learning algorithm used in machine learning to solve classification problems. It is primarily used in text classification that includes a high-dimensional training dataset. The probabilistic classifier helps make quick predictions based on an object's probability by assuming that a particular feature's occurrence is independent of the event of other features and follows Bayes' theorem. Here, the Gaussian Naive Bayes classifier is used to predict the accuracies and the Bernoulli Naive Bayes classifier improves the classification accuracies.

Fine-tuning using GridSearchCV:

GridSearchCV belongs to the sklearn model selection class, used to find the best parameters for a given model with given data. Exhaustive search is processed over some specified parameter values for the estimator. The parameters used to apply these methods are optimized by cross-validated grid search over the parameter grid. It implements a “fit” and a “score” method to find the best_estimator_ and best_params_ attributes for the given model.

The best parameters obtained from the above search are then used to fine-tune the model from which we derive validation and testing accuracies of the given features.

RESULTS

The plots consist of confusion matrix for testing dataset and result of model

Accuracy Scores found for Difficulty variable

Algorithm	Accuracy_type	Subject_wise	Record_wise	Book_wise
Logistic Regression	Validation	0.66	0.66	0.67
	Testing	0.72	0.66	0.70
Decision tree	Validation – b	0.56	0.57	0.55
	Validation – a	0.66	0.66	0.63
	Testing	0.67	0.66	0.66
Random Forest	Validation – b	0.62	0.64	0.60
	Validation – a	0.66	0.67	0.63
	Testing	0.66	0.67	0.65
K-Nearest Neighbor	Validation – b	0.60	0.61	0.58
	Validation – a	0.63	0.64	0.61
	Testing	0.62	0.65	0.63
Naïve Bayes	GaussianNB	0.65	0.66	0.64
	BernoulliNB	0.66	0.66	0.64
	Testing	0.72	0.66	0.64

b – before fine-tuning a – after fine-tuning

Accuracy Scores found for Nativity variable

Algorithm	Accuracy_type	Subject_wise	Record_wise	Book_wise
Logistic Regression	Validation	0.63	0.66	0.67
	Testing	0.72	0.66	0.67
Decision tree	Validation – b	0.53	0.59	0.58
	Validation – a	0.60	0.66	0.68
	Testing	0.67	0.67	0.67
Random Forest	Validation – b	0.56	0.66	0.65
	Validation – a	0.58	0.68	0.68
	Testing	0.68	0.68	0.68
K-Nearest Neighbor	Validation – b	0.56	0.64	0.63
	Validation – a	0.58	0.66	0.66
	Testing	0.65	0.66	0.67
Naïve Bayes	GaussianNB	0.63	0.65	0.67
	BernoulliNB	0.63	0.66	0.67
	Testing	0.72	0.66	0.67

Final Accuracy Scores found for Difficulty and Native Variables

Prediction Variable	Difficulty			Native		
	Subject_wise	Record_wise	Book_wise	Subject_wise	Record_wise	Book_wise
Null Accuracy	0.72	0.66	0.69	0.72	0.66	0.67
Logistic Regression	0.72	0.66	0.69	0.72	0.66	0.67
Decision Tree	0.67	0.66	0.65	0.67	0.67	0.67
Random Forest	0.66	0.67	0.65	0.67	0.68	0.68
K-Nearest Neighbor	0.62	0.65	0.63	0.65	0.66	0.67
Naive Bayes	0.72	0.66	0.64	0.72	0.66	0.67

PREDICTING ON NEW WINDOWS – RECORD WISE

NULL ACCURACY

DIFFICULTY acc=0.66%

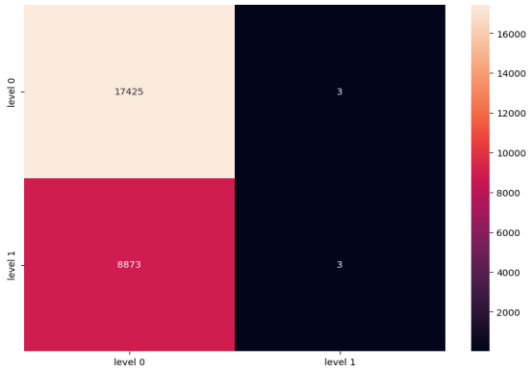
```
##### data description #####
# of classes:    2
input shape is:  4
# of samples for training is:    210425
# of samples for validation is:  26303
# of samples for prediction is:  26304
# of total sampels:      263032

##### data imbalances #####
0    0.659917
1    0.340083
Name: difficulty, dtype: float64

##### loss weight #####
{0: 0.7576712299172566, 1: 1.4702286129510076}

##### null acc for test dataset #####
0.6625608272506083
```

NATIVE acc=0.66%

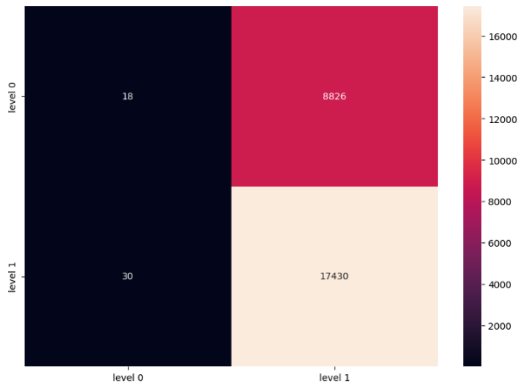
```
##### data description #####
# of classes:    2
input shape is:  4
# of samples for training is:    210425
# of samples for validation is:  26303
# of samples for prediction is:  26304
# of total sampels:      263032

##### data imbalances #####
0    0.340482
1    0.659518
Name: native, dtype: float64

##### loss weight #####
{0: 1.468504871172152, 1: 0.7581298323233342}

##### null acc for test dataset #####
0.6637773722627737
```

LOGISTIC REGRESSION

```
predicted variable: difficulty

              precision    recall  f1-score   support

     level 0       0.66      1.00      0.80     17428
     level 1       0.50      0.00      0.00      8876

    accuracy                           0.66     26304
   macro avg       0.58      0.50      0.40     26304
weighted avg       0.61      0.66      0.53     26304

confusion matrix:
 [[17425     3]
 [ 8873     3]]
Balanced acc score: 0.500082926647127
Balanced error rate: 0.49991707335287305
```




```
predicted variable: native

              precision    recall  f1-score   support

     level 0       0.38      0.00      0.00      8844
     level 1       0.66      1.00      0.80     17460

    accuracy                           0.66     26304
   macro avg       0.52      0.50      0.40     26304
weighted avg       0.57      0.66      0.53     26304

confusion matrix:
 [[   18  8826]
 [   30 17430]]
Balanced acc score: 0.5001585325481309
Balanced error rate: 0.4998414674518691
```

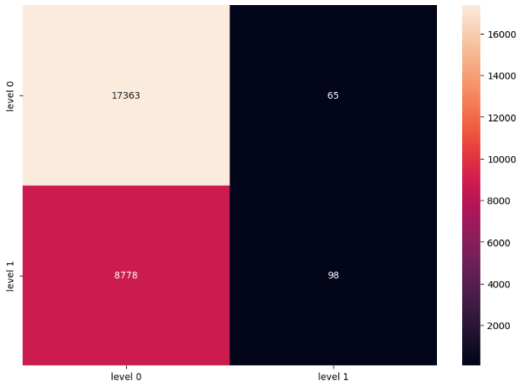


DECISION TREE

predicted variable: difficulty

	precision	recall	f1-score	support
level 0	0.66	1.00	0.80	17428
level 1	0.60	0.01	0.02	8876
accuracy			0.66	26304
macro avg	0.63	0.50	0.41	26304
weighted avg	0.64	0.66	0.54	26304

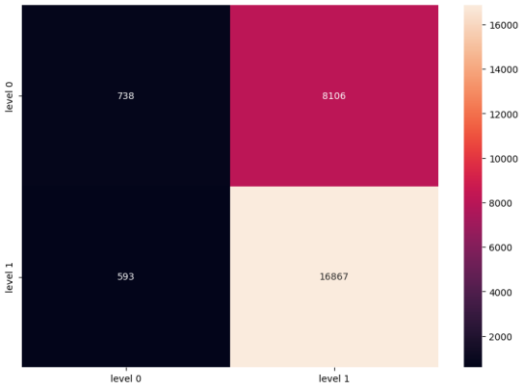
confusion matrix:
[[17363 65]
[8778 98]]
Balanced acc score: 0.5036556894920172
Balanced error rate: 0.4963443105079828



predicted variable: native

	precision	recall	f1-score	support
level 0	0.55	0.08	0.15	8844
level 1	0.68	0.97	0.79	17460
accuracy			0.67	26304
macro avg	0.61	0.52	0.47	26304
weighted avg	0.63	0.67	0.58	26304

confusion matrix:
[[738 8106]
[593 16867]]
Balanced acc score: 0.5247415297769198
Balanced error rate: 0.4752584702230802

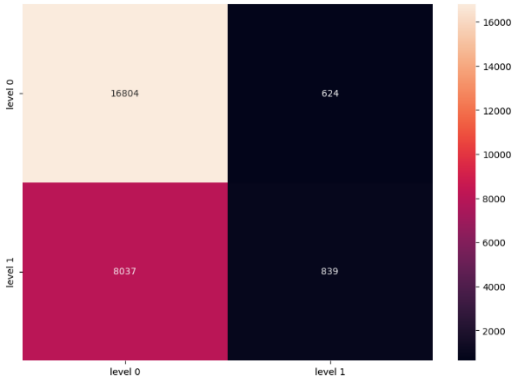


RANDOM FOREST

predicted variable: difficulty

	precision	recall	f1-score	support
level 0	0.68	0.96	0.80	17428
level 1	0.57	0.09	0.16	8876
accuracy			0.67	26304
macro avg	0.62	0.53	0.48	26304
weighted avg	0.64	0.67	0.58	26304

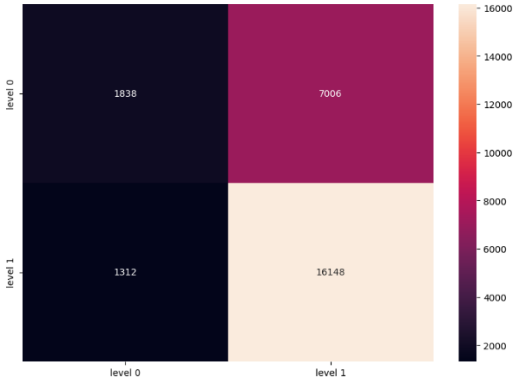
confusion matrix:
[[16804 624]
[8037 839]]
Balanced acc score: 0.5293600540039426
Balanced error rate: 0.4706399459960574



predicted variable: native

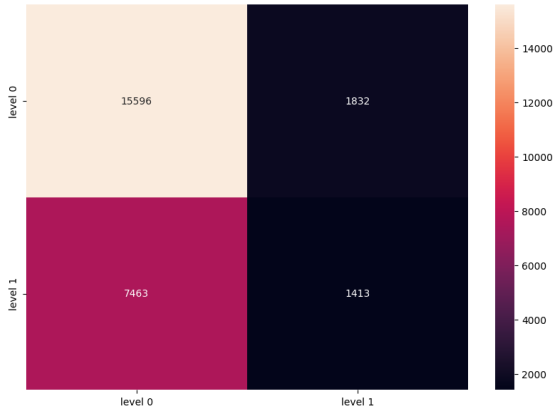
	precision	recall	f1-score	support
level 0	0.58	0.21	0.31	8844
level 1	0.70	0.92	0.80	17460
accuracy			0.68	26304
macro avg	0.64	0.57	0.55	26304
weighted avg	0.66	0.68	0.63	26304

confusion matrix:
[[1838 7006]
[1312 16148]]
Balanced acc score: 0.5663406646865641
Balanced error rate: 0.4336593353134359

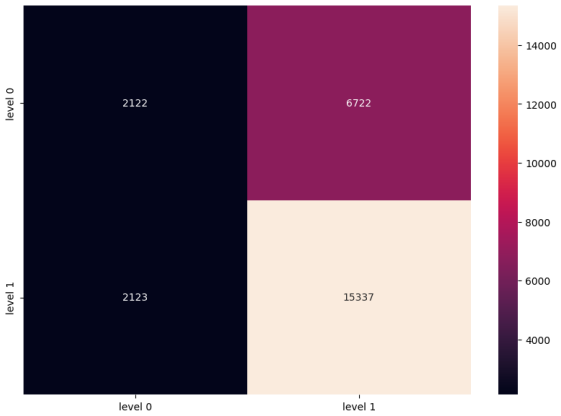


K-NEAREST NEIGHBOR

```
predicted variable: difficulty

              precision    recall  f1-score   support

     level 0       0.68      0.89      0.77     17428
     level 1       0.44      0.16      0.23      8876

    accuracy                           0.65     26304
   macro avg       0.56      0.53      0.50     26304
weighted avg       0.60      0.65      0.59     26304

confusion matrix:
 [[15596  1832]
 [ 7463  1413]]
Balanced acc score: 0.5270375648661181
Balanced error rate: 0.4729624351338819
```

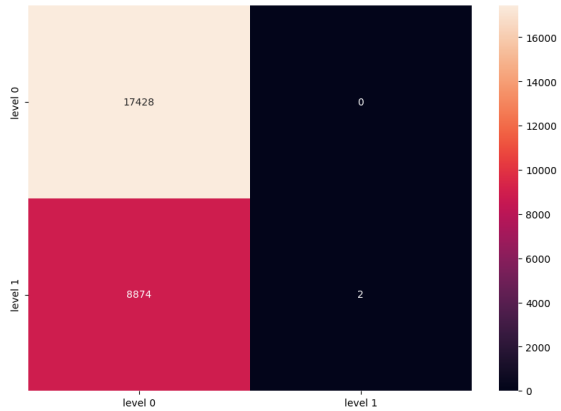



```
predicted variable: native

              precision    recall  f1-score   support

     level 0       0.50      0.24      0.32      8844
     level 1       0.70      0.88      0.78     17460

    accuracy                           0.66     26304
   macro avg       0.60      0.56      0.55     26304
weighted avg       0.63      0.66      0.62     26304

confusion matrix:
 [[ 2122  6722]
 [ 2123 15337]]
Balanced acc score: 0.5591722347338596
Balanced error rate: 0.4408277652661404
```

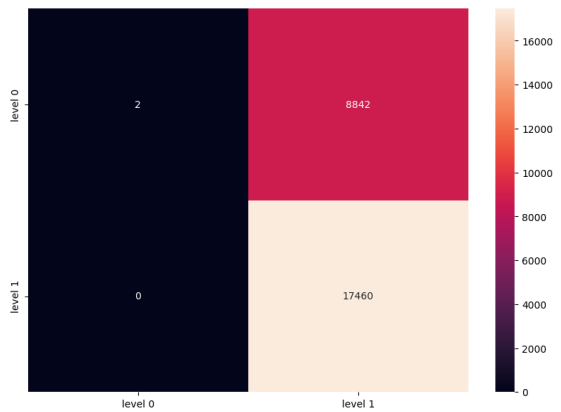


NAÏVE BAYES

```
predicted variable: difficulty

              precision    recall  f1-score   support

     level 0       0.66      1.00      0.80     17428
     level 1       1.00      0.00      0.00      8876

    accuracy                           0.66     26304
   macro avg       0.83      0.50      0.40     26304
weighted avg       0.78      0.66      0.53     26304

confusion matrix:
 [[17428     0]
 [ 8874     2]]
Balanced acc score: 0.5001126633618748
Balanced error rate: 0.49988733663812523
```




```
predicted variable: native

              precision    recall  f1-score   support

     level 0       1.00      0.00      0.00      8844
     level 1       0.66      1.00      0.80     17460

    accuracy                           0.66     26304
   macro avg       0.83      0.50      0.40     26304
weighted avg       0.78      0.66      0.53     26304

confusion matrix:
 [[    2  8842]
 [    0 17460]]
Balanced acc score: 0.5001130710085934
Balanced error rate: 0.49988692899140663
```



PREDICTING ON NEW PASSAGES – BOOK WISE

NULL ACCURACY

DIFFICULTY acc=0.69%

```
##### data description #####
# of classes:    2
input shape is:  4
# of samples for training is:    132447
# of samples for validation is:  64450
# of samples for prediction is:  66135
# of total sampels:      263032

##### data imbalances #####
0    0.637432
1    0.362568
Name: difficulty, dtype: float64

##### loss weight #####
{0: 0.7843969867102552, 1: 1.379052914349972}

##### null acc for test dataset #####
0.6936569138882589
```

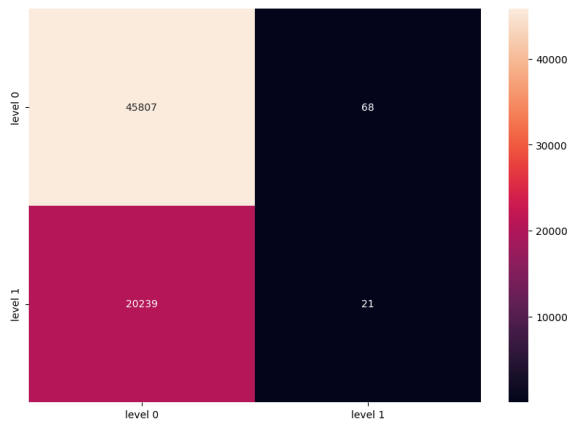
NATIVE acc=0.67%

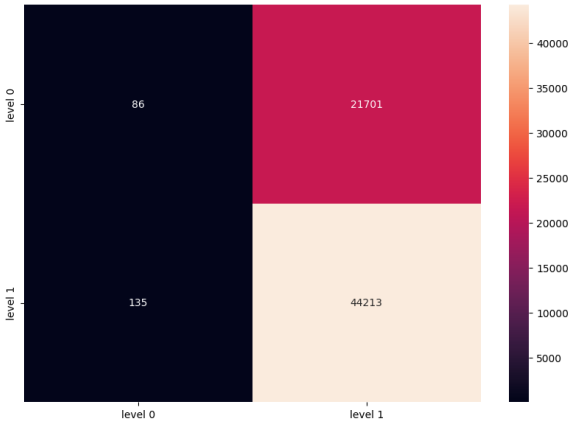
```
##### data description #####
# of classes:    2
input shape is:  4
# of samples for training is:    132447
# of samples for validation is:  64450
# of samples for prediction is:  66135
# of total sampels:      263032

##### data imbalances #####
0    0.352571
1    0.647429
Name: native, dtype: float64

##### loss weight #####
{0: 1.4181532004197273, 1: 0.7722857142857142}

##### null acc for test dataset #####
0.6705677780297875
```

LOGISTIC REGRESSION

```
predicted variable: difficulty

              precision    recall  f1-score   support

     level 0       0.69      1.00      0.82     45875
     level 1       0.24      0.00      0.00     20260

    accuracy                           0.69     66135
   macro avg       0.46      0.50      0.41     66135
weighted avg       0.55      0.69      0.57     66135

confusion matrix:
 [[45807    68]
 [20239    21]]
Balanced acc score: 0.49977711817220816
Balanced error rate: 0.5002228818277918
```

```
predicted variable: native

              precision    recall  f1-score   support

     level 0       0.39      0.00      0.01     21787
     level 1       0.67      1.00      0.80     44348

    accuracy                           0.67     66135
   macro avg       0.53      0.50      0.40     66135
weighted avg       0.58      0.67      0.54     66135

confusion matrix:
 [[   86 21701]
 [  135 44213]]
Balanced acc score: 0.5004516011591668
Balanced error rate: 0.49954839884083324
```



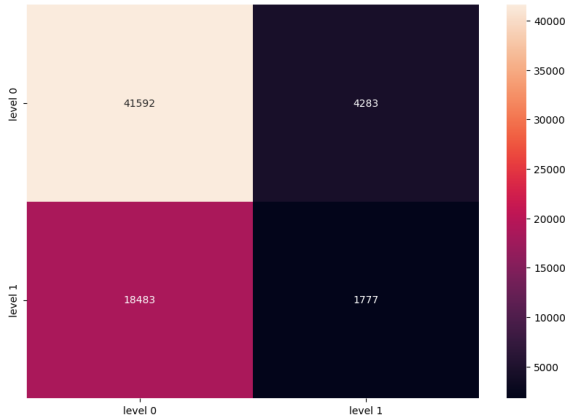


DECISION TREE

predicted variable: difficulty

	precision	recall	f1-score	support
level 0	0.69	0.91	0.79	45875
level 1	0.29	0.09	0.14	20260
accuracy			0.66	66135
macro avg	0.49	0.50	0.46	66135
weighted avg	0.57	0.66	0.59	66135

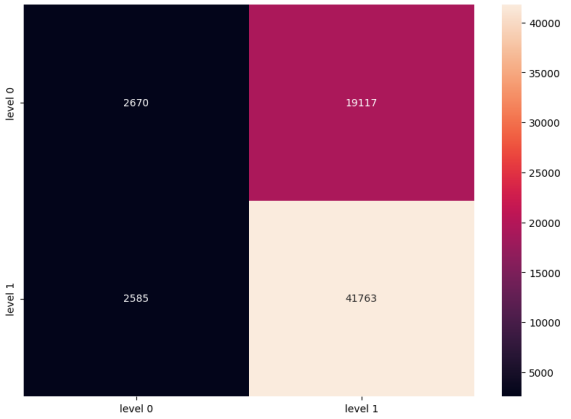
confusion matrix:
[[41592 4283]
[18483 1777]]
Balanced acc score: 0.49717368756573266
Balanced error rate: 0.5028263124342673



predicted variable: native

	precision	recall	f1-score	support
level 0	0.51	0.12	0.20	21787
level 1	0.69	0.94	0.79	44348
accuracy			0.67	66135
macro avg	0.60	0.53	0.50	66135
weighted avg	0.63	0.67	0.60	66135

confusion matrix:
[[2670 19117]
[2585 41763]]
Balanced acc score: 0.53213057873981
Balanced error rate: 0.46786942126019004

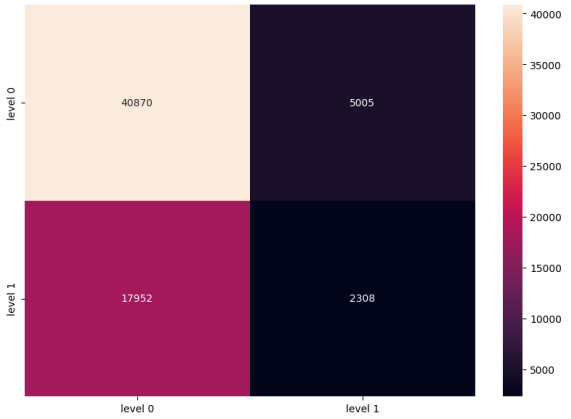


RANDOM FOREST

predicted variable: difficulty

	precision	recall	f1-score	support
level 0	0.69	0.89	0.78	45875
level 1	0.32	0.11	0.17	20260
accuracy			0.65	66135
macro avg	0.51	0.50	0.47	66135
weighted avg	0.58	0.65	0.59	66135

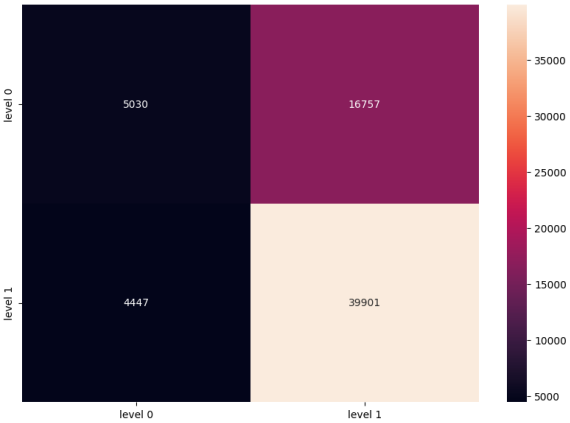
confusion matrix:
[[40870 5005]
[17952 2308]]
Balanced acc score: 0.502409117440575
Balanced error rate: 0.49759088255942496



predicted variable: native

	precision	recall	f1-score	support
level 0	0.53	0.23	0.32	21787
level 1	0.70	0.90	0.79	44348
accuracy			0.68	66135
macro avg	0.62	0.57	0.56	66135
weighted avg	0.65	0.68	0.64	66135

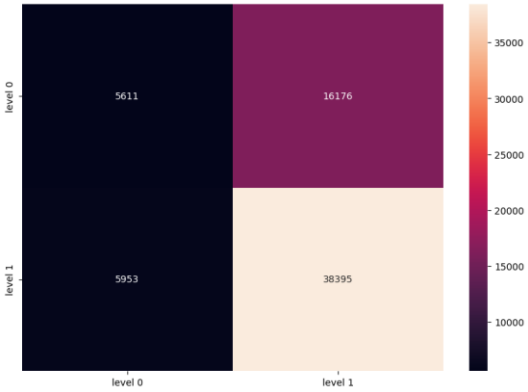
confusion matrix:
[[5030 16757]
[4447 39901]]
Balanced acc score: 0.565298261865417
Balanced error rate: 0.434701738134583



K – NEAREST NEIGHBOR

```
predicted variable: difficulty

              precision    recall  f1-score   support

     level 0       0.69      0.84      0.76     45875
     level 1       0.31      0.17      0.22     20260

    accuracy                           0.63     66135
   macro avg       0.50      0.50      0.49     66135
weighted avg       0.58      0.63      0.59     66135

confusion matrix:
 [[38341  7534]
 [16857  3403]]
Balanced acc score: 0.5018687767469759
Balanced error rate: 0.49813122325302406
```

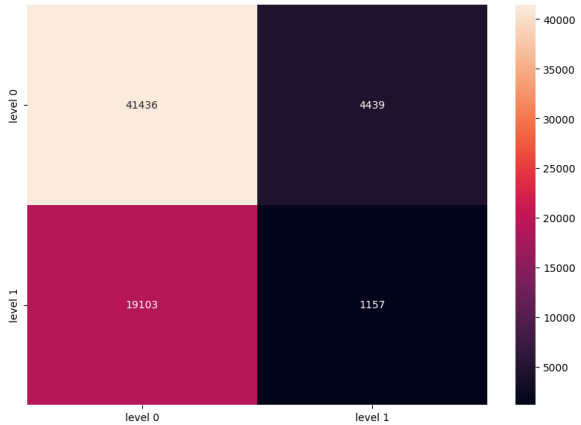
```
predicted variable: native

              precision    recall  f1-score   support

     level 0       0.49      0.26      0.34     21787
     level 1       0.70      0.87      0.78     44348

    accuracy                           0.67     66135
   macro avg       0.59      0.56      0.56     66135
weighted avg       0.63      0.67      0.63     66135

confusion matrix:
 [[ 5611 16176]
 [ 5953 38395]]
Balanced acc score: 0.561652556012582
Balanced error rate: 0.438347443987418
```

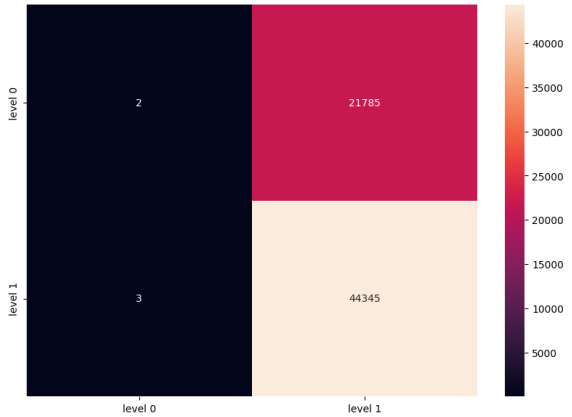


NAÏVE BAYES

```
predicted variable: difficulty

              precision    recall  f1-score   support

     level 0       0.68      0.90      0.78     45875
     level 1       0.21      0.06      0.09     20260

    accuracy                           0.64     66135
   macro avg       0.45      0.48      0.43     66135
weighted avg       0.54      0.64      0.57     66135

confusion matrix:
 [[41436  4439]
 [19103  1157]]
Balanced acc score: 0.48017232920265435
Balanced error rate: 0.5198276707973457
```

```
predicted variable: native

              precision    recall  f1-score   support

     level 0       0.40      0.00      0.00     21787
     level 1       0.67      1.00      0.80     44348

    accuracy                           0.67     66135
   macro avg       0.54      0.50      0.40     66135
weighted avg       0.58      0.67      0.54     66135

confusion matrix:
 [[    2 21785]
 [    3 44345]]
Balanced acc score: 0.5000120755337839
Balanced error rate: 0.4999879244662161
```





PREDICTING ON NEW READERS – SUBJECT WISE

NULL ACCURACY

DIFFICULTY acc=0.72%

```
##### data description #####
# of classes:    2
input shape is:  4
# of samples for training is:    209557
# of samples for validation is:  29256
# of samples for prediction is:  24219
# of total sampels:      263032

##### data imbalances #####
0    0.653622
1    0.346378
Name: difficulty, dtype: float64

##### loss weight #####
{0: 0.7649684969811128, 1: 1.4435083900476675}

##### null acc for test dataset #####
0.7214996490358809
```

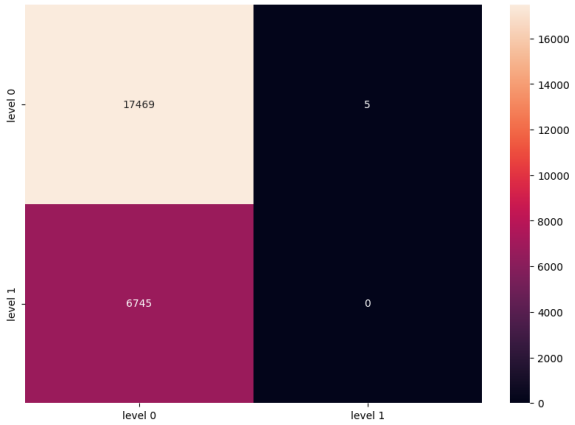
NATIVE acc=0.72%

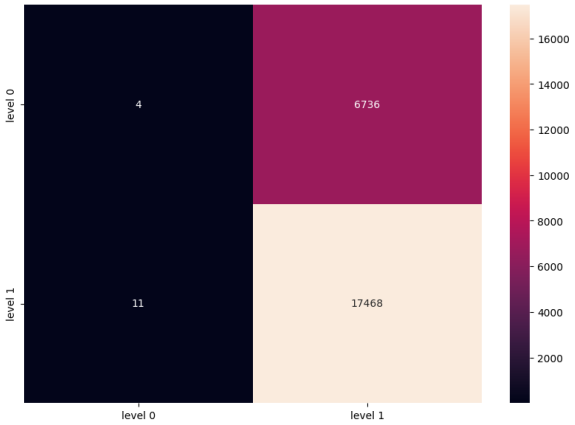
```
##### data description #####
# of classes:    2
input shape is:  4
# of samples for training is:    209557
# of samples for validation is:  29256
# of samples for prediction is:  24219
# of total sampels:      263032

##### data imbalances #####
0    0.343186
1    0.656814
Name: native, dtype: float64

##### loss weight #####
{0: 1.4569364684288832, 1: 0.7612503632664923}

##### null acc for test dataset #####
0.7217060985176927
```

LOGISTIC REGRESSION

```
predicted variable: difficulty

              precision    recall  f1-score   support

     level 0       0.72      1.00      0.84     17474
     level 1       0.00      0.00      0.00      6745

    accuracy                           0.72     24219
   macro avg       0.36      0.50      0.42     24219
weighted avg       0.52      0.72      0.60     24219

confusion matrix:
 [[17469     5]
 [ 6745     0]]
Balanced acc score: 0.4998569302964404
Balanced error rate: 0.5001430697035596
```

```
predicted variable: native

              precision    recall  f1-score   support

     level 0       0.27      0.00      0.00      6740
     level 1       0.72      1.00      0.84     17479

    accuracy                           0.72     24219
   macro avg       0.49      0.50      0.42     24219
weighted avg       0.60      0.72      0.61     24219

confusion matrix:
 [[    4  6736]
 [   11 17468]]
Balanced acc score: 0.4999820725947865
Balanced error rate: 0.5000179274052134
```



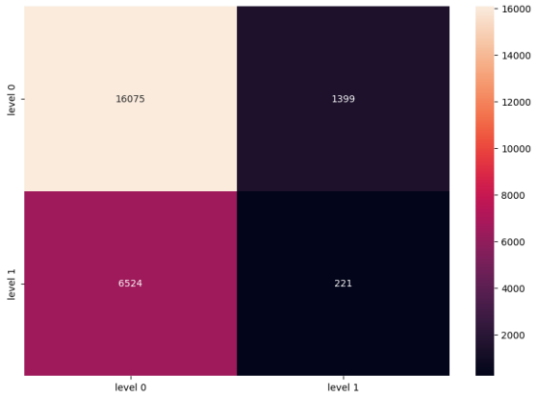


DECISION TREE

predicted variable: difficulty

	precision	recall	f1-score	support
level 0	0.71	0.92	0.80	17474
level 1	0.14	0.03	0.05	6745
accuracy			0.67	24219
macro avg	0.42	0.48	0.43	24219
weighted avg	0.55	0.67	0.59	24219

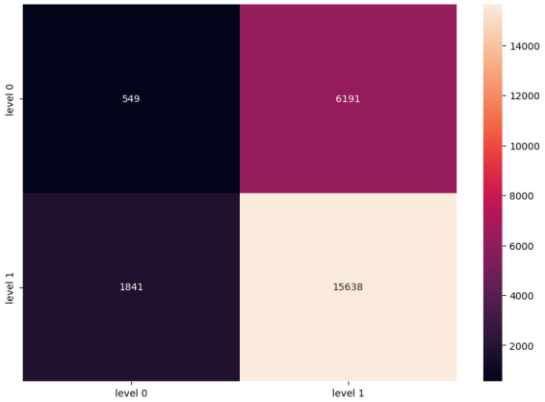
confusion matrix:
[[16075 1399]
[6524 221]]
Balanced acc score: 0.476351602503705
Balanced error rate: 0.523648397496295



predicted variable: native

	precision	recall	f1-score	support
level 0	0.23	0.08	0.12	6740
level 1	0.72	0.89	0.80	17479
accuracy			0.67	24219
macro avg	0.47	0.49	0.46	24219
weighted avg	0.58	0.67	0.61	24219

confusion matrix:
[[549 6191]
[1841 15638]]
Balanced acc score: 0.48806380713235703
Balanced error rate: 0.5119361928676429

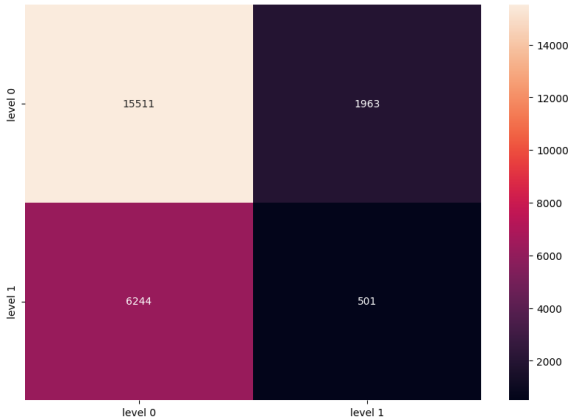


RANDOM FOREST

predicted variable: difficulty

	precision	recall	f1-score	support
level 0	0.71	0.89	0.79	17474
level 1	0.20	0.07	0.11	6745
accuracy			0.66	24219
macro avg	0.46	0.48	0.45	24219
weighted avg	0.57	0.66	0.60	24219

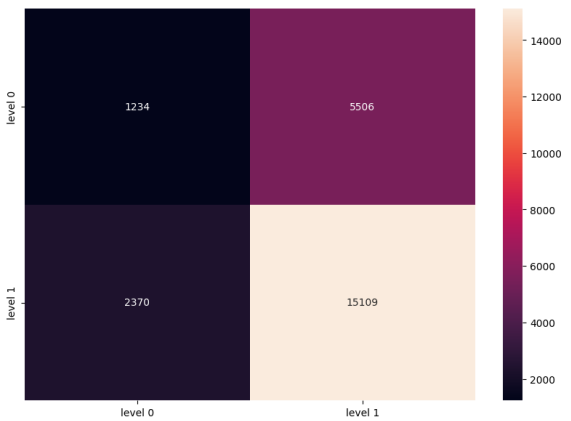
confusion matrix:
[[15511 1963]
[6244 501]]
Balanced acc score: 0.4809694555834007
Balanced error rate: 0.5190305444165992



predicted variable: native

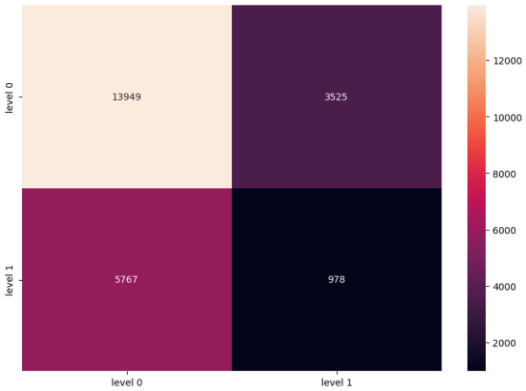
	precision	recall	f1-score	support
level 0	0.34	0.18	0.24	6740
level 1	0.73	0.86	0.79	17479
accuracy			0.67	24219
macro avg	0.54	0.52	0.52	24219
weighted avg	0.62	0.67	0.64	24219

confusion matrix:
[[1234 5506]
[2370 15109]]
Balanced acc score: 0.5237473862233664
Balanced error rate: 0.47625261377663364

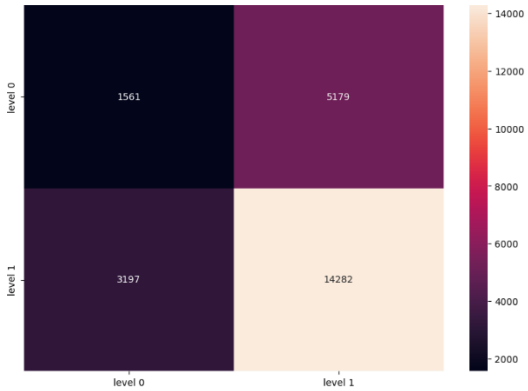


K – NEAREST NEIGHBOR

```
predicted variable: difficulty

              precision    recall  f1-score   support

     level 0       0.71      0.80      0.75     17474
     level 1       0.22      0.14      0.17      6745

    accuracy                           0.62     24219
   macro avg       0.46      0.47      0.46     24219
weighted avg       0.57      0.62      0.59     24219

confusion matrix:
 [[13949  3525]
 [ 5767   978]]
Balanced acc score: 0.4716340057658894
Balanced error rate: 0.5283659942341106
```

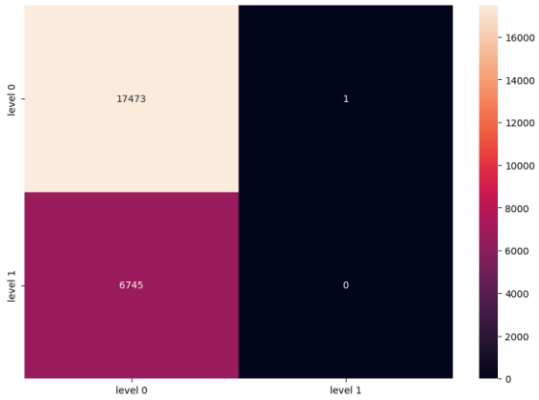



```
predicted variable: native

              precision    recall  f1-score   support

     level 0       0.33      0.23      0.27      6740
     level 1       0.73      0.82      0.77     17479

    accuracy                           0.65     24219
   macro avg       0.53      0.52      0.52     24219
weighted avg       0.62      0.65      0.63     24219

confusion matrix:
 [[ 1561  5179]
 [ 3197 14282]]
Balanced acc score: 0.5243485866804473
Balanced error rate: 0.4756514133955274
```

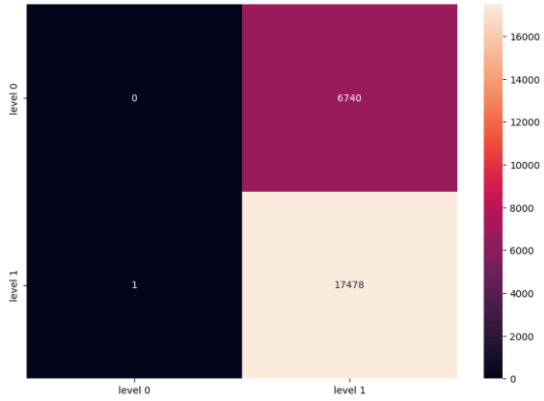


NAÏVE BAYES

```
predicted variable: difficulty

              precision    recall  f1-score   support

     level 0       0.72      1.00      0.84     17474
     level 1       0.00      0.00      0.00      6745

    accuracy                           0.72     24219
   macro avg       0.36      0.50      0.42     24219
weighted avg       0.52      0.72      0.60     24219

confusion matrix:
 [[17473     1]
 [ 6745     0]]
Balanced acc score: 0.4999713860592881
Balanced error rate: 0.500028613940712
```




```
predicted variable: native

              precision    recall  f1-score   support

     level 0       0.00      0.00      0.00      6740
     level 1       0.72      1.00      0.84     17479

    accuracy                           0.72     24219
   macro avg       0.36      0.50      0.42     24219
weighted avg       0.52      0.72      0.61     24219

confusion matrix:
 [[    0  6740]
 [    1 17478]]
Balanced acc score: 0.49997139424452197
Balanced error rate: 0.50002860575478
```



DISCUSSION

From the models, we can see that there is are still a lot of conditions that needs to be satisfied for improvement in the model's working. The models gave good accuracies, in some cases, near to the null accuracies. It is seen that the performance of the logistic regression model and naive bayes model have better accuracies (equal to the null accuracies) for both difficulty and native variables in all three types of datasets. But it is also observed that these two models have very few false positive and false negatives in the confusion matrix, which may seem to over-fit the given data. The validation datasets are used to fine-tune the model and improve the accuracies, it is observed that for new windows (record wise), there is no much improvement in the accuracies when compared to new readers and new passages. It is also observed that Naive Bayes model shows the same trait like there is no much improvement in the accuracy after fine-tuning the model.

These machine learning models have better performance when compared to the sequential models of RNN and CNN. The accuracies obtained are near to the null accuracies and the performance remains consistent regardless of the given dataset format. In RNN and CNN, the performance of record wise dataset is quite low, but that problem is resolved in these models. The training of the dataset is also improved as 80% of dataset is grouped for training, while in RNN and CNN models, only 60% dataset is used for training. This helped in better prediction of the new data given to the model.

The major limitation of this approach is the assumption that ground truth labels are assigned to local fixation windows based on the passage-level labels. This is done to avoid the task of labelling manually. But this problem needs to be solved for better performance of the models. A feasible solution can be to learn a weighting of local features that optimizes global prediction. This can be done by computer vision. A study is already going on this issue to construct reading detectors without using explicit ground truth labels for local fixation windows.

It is also observed that the results generated from the models vary as the dataset is randomly created. Thus, the accuracies listed here may not be optimal, but with few increases or decreases from the optimal accuracies. Most of cases, the accuracies obtained are almost equal to the null accuracy. Still, these models' performance needs to be improved and is to be worked upon.

CONCLUSION

These models are built to study and analyse approaches to classify a reader's comprehension level. It worked on mainly two prediction variables: the reader's difficulty and whether the reader is a native speaker of English. The supervised learning models, logistic regression, decision tree, random forest, k-nearest neighbour, and Naive Bayes, inputted with windows of consecutive fixations of an SAT passage, tried to predict the accuracy of the prediction variables. It is observed that prediction on new passages, new readers, and new windows for both difficulty and native variables gives better performance when compared to previous RNN and CNN models. Predictions on new windows for difficulty and native does not improve much from fine-tuning the model.

There is still a lot of advancement to be done for the model. Thus, predicting comprehension from eye-movement data is a problem that needs to be worked upon. The model shows the difference in statistical testing and prediction. Though the models did not include all the dependent features of the eye-tracking data, they still predicted accuracies with a better success rate. Thus, these models are being used in various fields, and every improvement in the predictions can make a significant difference in diverse contexts. Reading comprehension evaluation can be improved further by using the BERT model. The model takes text as input and classifies the text into high or low with high efficiency. A pre-trained model of BERT can be taken and fine-tuned following our data to input the passages and assessment questions and answers, which in turn outputs the required assessment.

Thus, it concludes the working and implementation of the models for predicting reading comprehension from gaze behavior.

REFERENCES

https://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501

https://omdena.com/blog/machine-learning-classification-algorithms/

Hosam Al-Samarraie, Atef Eldenfria, and Husameddin Dawoud. 2017. The impact of personality traits on users' information-seeking behavior. Information Processing & Management 53, 1 (2017), 237–247.

Hosam Al-Samarraie, Samer Muthana Sarsam, Ahmed Ibrahim Alzahrani, and Nasser Alalwan. 2018. Personality and individual differences: the potential of using preferences for visual stimuli to predict the Big Five traits. Cognition, Technology & Work (2018), 1–13.

Olivier Augereau, Hiroki Fujiyoshi, and Koichi Kise. 2016. Towards an automated estimation of English skill via TOEIC score based on reading analysis. In 2016 23rd International Conference on Pattern Recognition (ICPR). IEEE, 1285–1290.

Ralf Biedert, Jörn Hees, Andreas Dengel, and Georg Buscher. 2012. A robust real-time reading-skimming classifier. In Proceedings of the Symposium on Eye Tracking Research and Applications. ACM, 123–130.

Jonathan FG Boisvert and Neil DB Bruce. 2016. Predicting task from eye movements: On the importance of spatial distribution, dynamics, and image features. Neurocomputing 207 (2016), 653–668.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks 18, 5-6 (2005), 602–610.

John M Henderson. 2003. Human gaze control during real-world scene perception. Trends in cognitive sciences 7, 11 (2003), 498–504.

John M Henderson, Svetlana V Shinkareva, Jing Wang, Steven G Luke, and Jenn Olejarczyk. 2013. Predicting cognitive state from eye movements. PloS one 8, 5 (2013), e64937.

Shoya Ishimaru, Kensuke Hoshika, Kai Kunze, Koichi Kise, and Andreas Dengel. 2017. Towards reading trackers in the wild: detecting reading activities by EOG glasses and deep neural networks. In Proceedings of the 2017 ACM

International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 704–711.

Conor Kelton, Zijun Wei, Seoyoung Ahn, Aruna Balasubramanian, Samir Das, Dimitris Samaras, and Gregory Zelinsky. 2019. Reading Detection in Real-time. In In 2019 Symposium on Eye Tracking Research and Applications (ETRA '19).

Ya Lou, Yanping Liu, Johanna K Kaakinen, and Xingshan Li. 2017. Using support vector machines to identify literacy skills: Evidence from eye movements. Behavior research methods 49, 3 (2017), 887–895.

Silvia Makowski, Lena A Jäger, Ahmed Abdelwahab, Niels Landwehr, and Tobias Scheffer. 2018. A Discriminative Model for Identifying Readers and Assessing Text Comprehension from Eye Movements. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 209–225.

Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. Psychological bulletin 124, 3 (1998), 372.

Keith Rayner. 2009. Eye movements and attention in reading, scene perception, and visual search. The quarterly journal of experimental psychology 62, 8 (2009), 1457–1506.

Geoffrey Underwood, Alison Hubbard, and Howard Wilkinson. 1990. Eye fixations predict reading comprehension: The relationships between reading skill, reading speed, and visual inspection. Language and speech 33, 1 (1990), 69–81.

Zijun Wei, Hossein Adeli, Minh Hoai Nguyen, Greg Zelinsky, and Dimitris Samaras. 2016. Learned Region Sparsity and Diversity Also Predicts Visual Attention. In Advances in Neural Information Processing Systems. 1894–1902.