Kartik Mathur

Mkartik477@gmail.com

Prepared for Peter Vo

Director of Data and Insights  
Data2Intel

analysis of Numeracy risk in Primary Schools

Analytical Report

Table of Contents

[Executive Summary 2](#_Toc200039777)

[Business Question 2](#_Toc200039778)

[Approach 2](#_Toc200039779)

[Exploratory Data Analysis (EDA) 3](#_Toc200039780)

[Distribution of Target Variable 3](#_Toc200039781)

[Data Processing 4](#_Toc200039782)

[Correlation between features 5](#_Toc200039783)

[Differences According to Risk 5](#_Toc200039784)

[5. Preprocessing 7](#_Toc200039785)

[Irrelevant Columns 7](#_Toc200039786)

[Unexpected Values 7](#_Toc200039787)

[Standardising Values 7](#_Toc200039788)

[Class Imbalance 7](#_Toc200039789)

[Correlated Features 7](#_Toc200039790)

[6. Model Development and Evaluation 8](#_Toc200039791)

[Supervised Learning (Classification Models) 8](#_Toc200039792)

[Logistic regression 8](#_Toc200039793)

[Support Vector Classifier 9](#_Toc200039794)

[Evaluation Metrics: 10](#_Toc200039795)

[Evaluation of Logistic Regression 11](#_Toc200039796)

[Evaluation of Support Vector Classifier 11](#_Toc200039797)

[Comparison with Un-optimised Logistic Regression 12](#_Toc200039798)

[Precision-Recall Curve Analysis 13](#_Toc200039799)

[Unsupervised Learning (Clustering Analysis) 14](#_Toc200039800)

[Evaluation Metrics 14](#_Toc200039801)

[K-Means Clustering 15](#_Toc200039802)

[Solution Recommendation. 18](#_Toc200039803)

[Technical Recommendations 18](#_Toc200039804)

[Development and Testing Environment 18](#_Toc200039805)

[Model Deployment Considerations 18](#_Toc200039806)

# Executive Summary

This report investigates 2,000 primary school students, leveraging machine learning techniques to predict students at risk of underperforming in numeracy in their third year. We analyse student demographics, academic performance, and disability conditions to find patterns and employ predictive and clustering models to gain valuable insights and develop early intervention strategies to improve student outcomes.

# Business Question

* **Objective:** Identify students at risk of underperforming in numeracy in their third year using early academic data.
* **Value Proposition:** Identify risk factors early, allowing for timely prioritisation to improve overall educational outcomes.

# Approach

The prediction target (Year3\_Numeracy\_At\_Risk) is a binary column in which only 25% of the values are True, and the rest are False.

* **Problem Type:** Binary Classification on Unbalanced Data(predicting).
* **Target Variable:** Boolean column Year3\_Numeracy\_At\_Risk
* **Techniques Used:**
  + Supervised Learning: Classification models (Logistic Regression and Support Vector Classification)
  + Unsupervised Learning: Clustering analysis (K-Means clustering)

# Exploratory Data Analysis (EDA)

### Distribution of Target Variable

A graph showing a number of students

AI-generated content may be incorrect.

The number of students who are at risk of underperforming (“Students at risk” hereafter) is significantly lower than those who are not. This is of course a good thing, but it means that our ML model will have to compensate for the imbalance.

This imbalance can be addressed by resampling the data using a technique like SMOTE or Random Under Sampling.

### Data Processing

A graph of colorful boxes

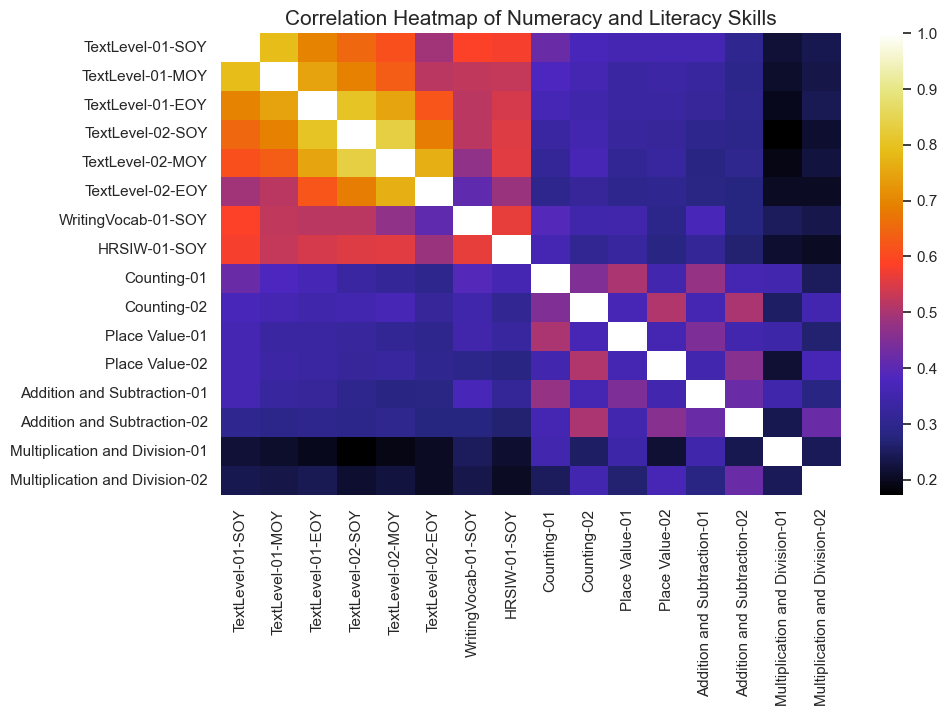
AI-generated content may be incorrect.

Above is a box plot showing the distribution of the data regarding the literacy skills of the students.

An analysis of the data distribution revealed outlier values in the 'Text Level' column exceeding the expected range (0-31), potentially introducing bias into the model. These entries were removed to maintain accuracy.

Further analysis revealed that the columns “HRSIW-01-SOY” contained negative values, and “NumAbvYear9” contained a single entry with a value of 3. These entries were also removed.

# Correlation between features



The correlation heatmap shows a moderate to strong correlation between some academic performance features. This indicates multi-collinearity which may bias our model. This can be reduced using Principal Component Analysis.

### Differences According to Risk

We are interested in how students who are at risk differ from those who are not. First, we will look at the differences between the means of each column.

Looking into these differences may reveal which columns are likely the most important features to consider in our model, allowing for effective feature engineering.

The table below shows the top ten values of the differences between the means of each column according to whether the student is at risk.

A table of data with numbers

AI-generated content may be incorrect.

As expected, the greatest difference comes from the columns pertaining to academic performance. However, this method of analysis does not account for the differences in scale of the values between columns.

A table with numbers and symbols

AI-generated content may be incorrect.

In the table above, we conduct a T-Test to check the statistical significance of this difference. A P-Value below 0.05 indicates statistical significance. The table shows the columns with the top ten P-Values. Observe that 7 columns have a P-Value that does not indicate significance. These columns will likely add noise to the model and will thus be removed.

It should be noted that the difference in scale between columns can also add bias to the model. Considering this, the values will be standardized.

# 5. Preprocessing

The EDA process revealed the following properties of the data:

1. Irrelevant columns
2. Values outside the given range for a column
3. Inconsistent scale between columns
4. Imbalance in the distribution of classes in the target variable
5. Correlation between some columns

Considering this, the following preprocessing steps were taken.

### Irrelevant Columns

According to the T-Test analysis, five columns showed no statistically significant difference in their values. These columns have been removed from the training data.

### Unexpected Values

To ensure that the model is not biased by impossible data, all entries containing abnormal data have been removed from the training data.

### Standardising Values

To mitigate scale discrepancies between columns, standardisation has been applied, ensuring a mean of 0 and unit variance across all features. This makes the training data compliant with the assumptions of Logistic Regression and mitigates bias.

### Class Imbalance

With 75% of students classified as 'not at risk,' the model risks bias toward this majority class. To address this, Random Under Sampling was employed, balancing the dataset to improve prediction accuracy. The purpose of this model is to accurately predict which students are at risk, removing this bias .

### Correlated Features

Columns that are correlated with each other can also introduce bias. To address this, we use Principal Component analysis to reduce potentially correlated columns into a few columns with no correlation.

To ensure optimal performance, the correct number of features must be created. With too few features, we lose out on crucial data. However, as the number of features increases, the model must operate in higher and higher dimensions, causing performance to suffer.

A screenshot of a graph

AI-generated content may be incorrect.

Using the graph above, we can select the optimal number of features.

# 6. Model Development and Evaluation

## Supervised Learning (Classification Models)

### Logistic regression

The model was tested using the following pipeline:

logreg\_pipe\_1 = Pipeline(

        [

            ("resample", RandomUnderSampler(random\_state=42)),

            ("scaling", StandardScaler()),

            ("pca", PCA(n\_components=8, random\_state=42)),

            ("classifier", LogisticRegression(

C=0.1,

solver='newton-cg',

random\_state=42

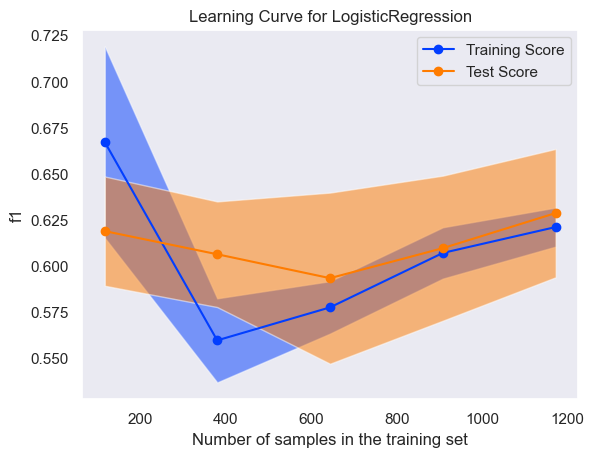
)

)

        ]

    )

The resampling technique, number of PCA components, and model hyperparameters were selected through cross validation analysis on the training data.



The above graph shows the scalability of the model. As the number of training samples increases, the training and test scores converge and trend upwards. Thus, the model is likely to maintain or improve performance with more data.

### Support Vector Classifier

The model was tested using the following pipeline:

svc\_pipe\_1 = Pipeline(

        [

            ("resample", RandomUnderSampler(random\_state=42)),

            ("scaling", StandardScaler()),

            ("pca", PCA(n\_components=4, random\_state=42)),

            ("classifier", SVC(

C=0.1,

gamma=0.001,

kernel='rbf',

probability=True

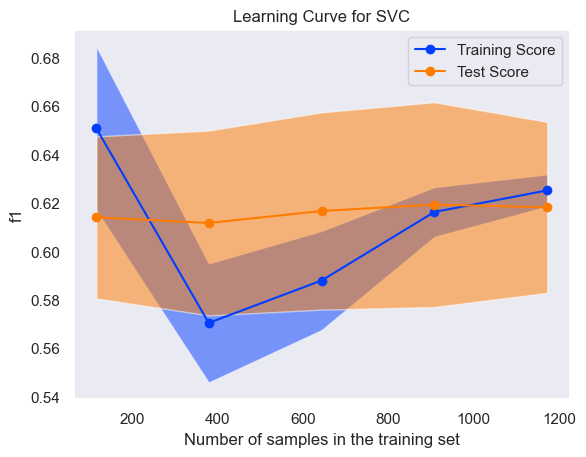
)

)

        ]

    )

Just as with the Logistic regression model, the resampling technique, number of PCA components, and model hyperparameters were selected through cross validation analysis on the training data.



The training and test scored do converge as the number of samples increases, however there is no upward trend. Given that our current test size is approximately 350 entries, the model is likely to underperform during this test. Performance will likely improve as the number of samples increases.

### Evaluation Metrics:

The models will be evaluated according to the following metrics:

* 1. **Confusion Matrix:** A visual representation of predicted classifications vs actual classifications.
  2. **Recall Score:** Represents the proportion of correctly identified at-risk students out of all actual cases.
  3. **F1-Score:** Represents the harmonic mean of the recall score and precision score. Precision score being the ratio of true positives to predicted positives.
  4. **Precision-Recall Curve:** Represents the relationship between precision and recall at different probability thresholds.

Our priority is to correctly identify students who are at risk. This means that we want to minimise cases where the true label is 1 and the predicted label is 0, which represents the number of students who are at risk that our model missed.

Considering this, the Recall score and the F1-Score are the most relevant metrics.

### Evaluation of Logistic Regression

A chart of a color chart

AI-generated content may be incorrect.

Recall Score: 0.739

F1-score: 0.589

### Evaluation of Support Vector Classifier

A chart of a blue yellow and purple box

AI-generated content may be incorrect.

Recall Score: 0.656

F1-Score: 0.565

### Comparison with Un-optimised Logistic Regression

**A chart of a color chart

AI-generated content may be incorrect.**

Recall Score: 0.354

F1-Score: 0.472

The above scores and confusion matrix are the result of running a completely unoptimized, trained model on the testing data.

The outlined preprocessing and hyperparameter selection techniques were not applied to this model, resulting in severely reduced performance. The optimised Logistic Regression model boasts a recall score of 208% that of the unoptimized model.

### Precision-Recall Curve Analysis

**A graph of a logistic regression

AI-generated content may be incorrect.**

The precision-recall curve evaluates the model at different probability thresholds, that being the probability at which the model classifies an entry as positive. A lower threshold means more positive predictions and less negative predictions.

This means that there is often a direct trade-off between precision and recall, as is evident from the above curves.

The legend shows the model’s name, the model’s AP (Average Precision), and the model’s optimal threshold.

The Logistic regression model outperforms SVC, and that is explained by this precision-recall curve. The model has higher AP and a curve that shows consistently higher precision and recall values, indicating superior performance.

## Unsupervised Learning (Clustering Analysis)

### Evaluation Metrics

Clusters will be evaluated using the Silhouette Score.

The Silhouette Score measures how similar a data point is to its own cluster compared to other clusters. It considers two aspects:

1. Cohesion: How close a data point is to other points in its own cluster.
2. Separation: How far a data point is from the points in the nearest neighbouring cluster.

The graph below shows the Silhouette score achieved by K-Means Clustering on the dataset by cluster count.

A graph with a line

AI-generated content may be incorrect.

Using K-Means Clustering with 3 clusters produces the optimal Silhouette score.

### K-Means Clustering

A diagram of a cluster of dots

AI-generated content may be incorrect.

The plot above shows the results of the clustering analysis. Each data point has been separated into distinct, non-overlapping clusters.

To further understand each cluster, we will look at the means of each column according to cluster.

A graph of a bar chart

AI-generated content may be incorrect.

Cluster 0 contains most of the students at risk of underperforming at numeracy in year 3. Clusters 1 and 2 seem quite like each other in this respect. Investigating other columns will reveal the distinctions.

A graph of multiple colored rectangular objects

AI-generated content may be incorrect.

A graph of numbers and numbers

AI-generated content may be incorrect.

The graph above shows the students’ literacy skills by cluster. As expected, students belonging to cluster 0 tend to have lower academic performance. A distinction between clusters 1 and 2 also arises, with cluster 2 having higher academic performance on average.

Next, we look at the disability profile of the students in each cluster.

A graph of different colored bars

AI-generated content may be incorrect.

Since the disability columns in this dataset are marked as either 1 or 0, the bars in this graph represent the probability that a student belonging to that cluster has a certain disability.

The results for most disability types are as expected, with cluster 0 having the highest probability and cluster 2 having the lowest probability. However, students in cluster 2 seems to have the highest probability of having a physical disability as compared to the others.

We also observe that students in cluster 0 are more likely to receive NCCD funding.

# Solution Recommendation.

Among the models tested, the Logistic Regression approach demonstrated the strongest performance on the training data. In addition, its scalability and reliability make it the preferred choice for deployment.

# Technical Recommendations

## Development and Testing Environment

* **Programming Language:** Python
* **Libraries Used:** Pandas, numpy, Scikit-learn, Seaborn, Matplotlib, Imblearn, scipy
* **Computing Environment:** Jupyter Notebook

## Model Deployment Considerations

* Follow the preprocessing pipeline outlined earlier in the document.
* Conduct regular cross validation and learning curve analysis to maintain performance and scalability.