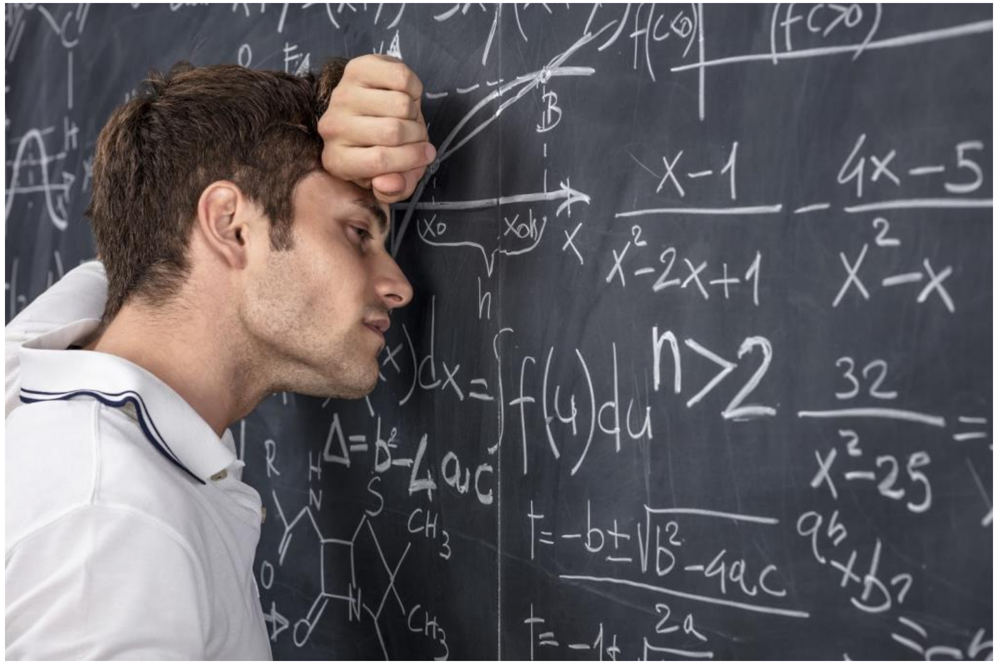
ASSESSING MATHEMATICS LEARNING IN HIGHER EDUCATION BY USING MACHINE LEARNING AND DEEP LEARNING

Presented by:
Abhishek Pandit
Mustafa Alsenaidi
Anand Aggarwal
Yug Patel

STRUCTURE OF PRESENTATION

INTRODUCTION

- Understanding and supporting student performance is a key challenge in education
- Predictive models enable early identification of struggling students
- Machine Learning (ML) and Deep Learning (DL) can uncover patterns from educational data
- **Goal:** Accurately assess student performance when learning mathematics to enable timely interventions.



CodaKid. (n.d.). Struggling with Math? Retrieved April 29, 2025, from https://codakid.com/struggling-with-math/

[1] Moubayed et al., A Deep Learning Approach Towards Student Performance Prediction, 2024

LITERATURE REVIEW

Name of Author and Research Title	Dataset	Methodology & Best-performing Results	Advantages / Pros	Limitations / Cons
Felipe E. Arévalo-Cordovilla(2024). Comparative Analysis of Machine-Learning Models for Predicting Student Success in Online Programming Courses.	591 Moodle LMS + demographics (UNEMI, 2022–2023)	**Logistic Regression** achieved the best AUC-ROC of 0.9354 with 87% accuracy, while **Random Forest** delivered the highest overall accuracy of 89% among LogReg, RF, SVM, and MLP models.	Interpretable; early grades + engagement give strong discrimination; easy to deploy.	Single course/institution; lacks psychosocial factors
Edmund F. Agyemang et al. (2024). Predicting Students Academic Performance Via Machine-Learning Algorithms.	xAPI-Edu-Data click-stream from Kaggle (secondary-school interactions)	Among RF, KNN, DT, LogReg, and SVM models, **Random Forest** performed best with an **accuracy of 85.42%** and a **G-Mean of 0.9243**, indicating strong balanced classification performance.	Balanced sensitivity & specificity; highlights actionable engagement features; broad literature context.	One MOOC-style context; RF less interpretable; generalizability flagged
Naveed U. R. Junejo et al. (2024) SAPPNet: Students' Academic Performance Prediction during COVID-19 using Neural Network	Jordan Univ. questionnaire (demographics, digital-tool use pre/post COVID-19, sleep, psych., GPA)	The hybrid deep learning model **SAPPNet** outperformed all ML/DL baselines, achieving **93%** in accuracy, precision, recall, and F1-score.	Captures static & temporal traits; highest multiclass GPA prediction; strong generalization on val. Data.	Computation heavy; pandemic-specific, single institution; transferability untested
Shuping Li and Taotang Liu (2021). Performance Prediction for Higher Education Students Using Deep Learning.	The dataset used in the study was collected from a multidisciplinary university, comprising 83,993 students, 4,699 courses from 2007 to 2019.	A hybrid deep learning model combining 1D CNN and LSTM was used with quantile transformation and **min-max scaling** for preprocessing. The model achieved high prediction accuracy with an MAE of **59%** and RMSE of **78%**	The model efficiently combines CNN and LSTM to handle complex, time-dependent student data, leading to more accurate predictions. It also benefits from effective preprocessing techniques that enhance model performance.	The approach is computationally intensive and requires careful tuning to avoid overfitting. Additionally, the model's complexity can make it harder to interpret and apply in practical educational settings.
A Deep Learning Approach Towards Student Performance Prediction in Online Courses: Challenges Based on a Global Perspective	The paper uses three datasets from different global universities, Students are labeled as Good, Fair, or Weak based on their final grade percentage (≥70%, 51–69%, ≤50%).	The study compared CNN and RNN-LSTM architectures with traditional models (SVM, KNN, Naive Bayes, RF); **RNN-LSTM** achieved **82%** on Dataset 1, **KNN** performed best on Dataset 2 with **94%**, and **CNN** led Dataset 3 with **91%** accuracy.	Enables early identification of weak students Works across different countries & courses Outperforms traditional ML in 2/3 datasets Leverages sequence (RNN) and feature extraction (CNN)	Not generalizable — no single model fits all datasets Sensitive to class imbalance (e.g., Dataset 2 had few weak students) Requires more compute and tuning than traditional ML

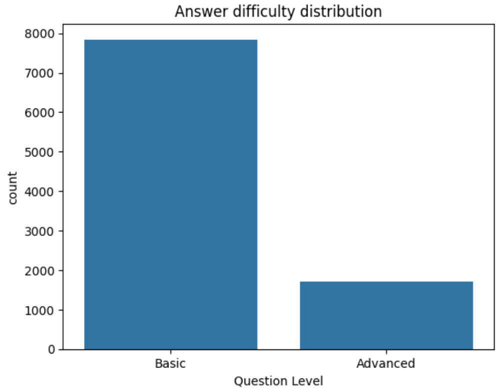
MATHE DATASET

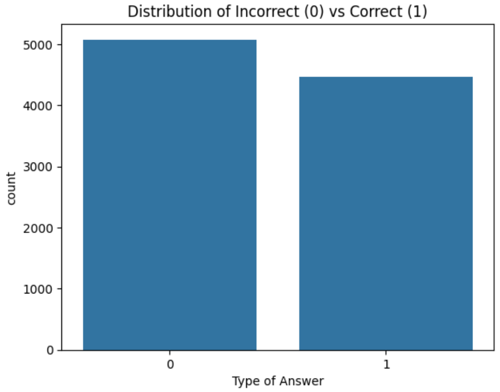
- Released early 2024, MathE platform is used by students in multiple countries.
- The dataset has 9546 record, with 8 features per record.
- 1. **Student ID**
- 2. **Student Country** (8 countries)
- 3. **Question ID** (833 distinct questions)
- 4. **Type of Answer** (Correct-1 or Incorrect-0)
- 5. **Question Level** (Basic and Advanced, rated by instructors)
- 6. **Topic** (15 distinct topics)
- 7. **Subtopic** (25 distinct subtopics)
- 8. **Keywords** (194 distinct keywords)

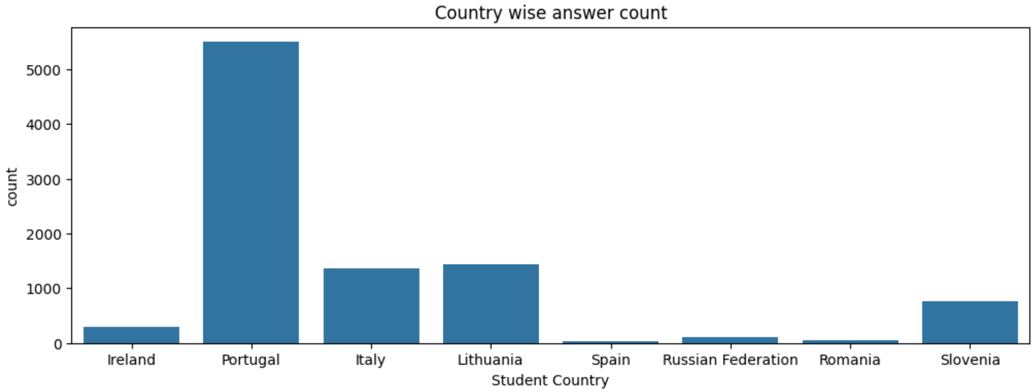
Student ID;Student Country;Question ID;Type of Answer;Question Level;Topic;Subtopic;Keywords			
647;Ireland;77;0;Basic;Statistics;Statistics;Stem and Leaf diagram	Relative frequency	Sample	Frequency
41;Portugal;77;1;Basic;Statistics;Statistics;Stem and Leaf diagram	Relative frequency	Sample	Frequency
340;Portugal;77;1;Basic;Statistics;Statistics;Stem and Leaf diagram	Relative frequency	Sample	Frequency
641;Italy;77;0;Basic;Statistics;Statistics;Stem and Leaf diagram	Relative frequency	Sample	Frequency
669;Portugal;77;1;Basic;Statistics;Statistics;Stem and Leaf diagram	Relative frequency	Sample	Frequency

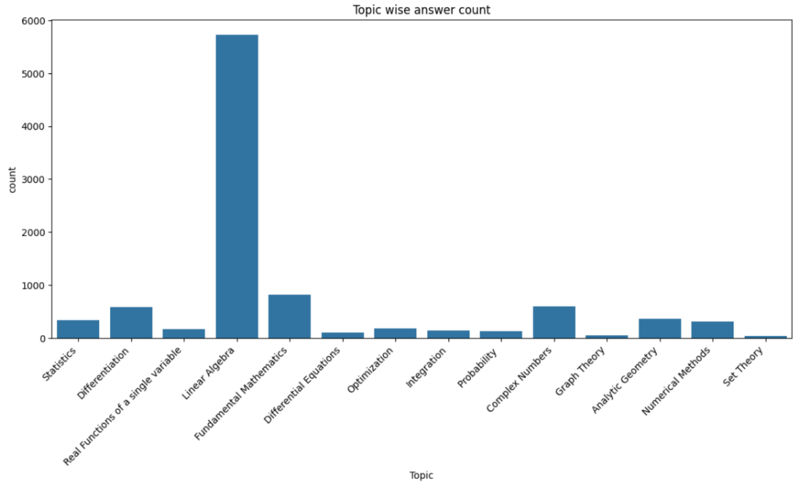
[2] P. Faria, D. Vieira, A. Albuquerque, and M. Dinis, "Dataset for Assessing Mathematics Learning in Higher Education," UCI Machine Learning Repository, University of California, Irvine, 2021. [Online]. Available: https://archive.ics.uci.edu/dataset/1031/dataset+for+assessing+mathematics+learning+in+higher+education

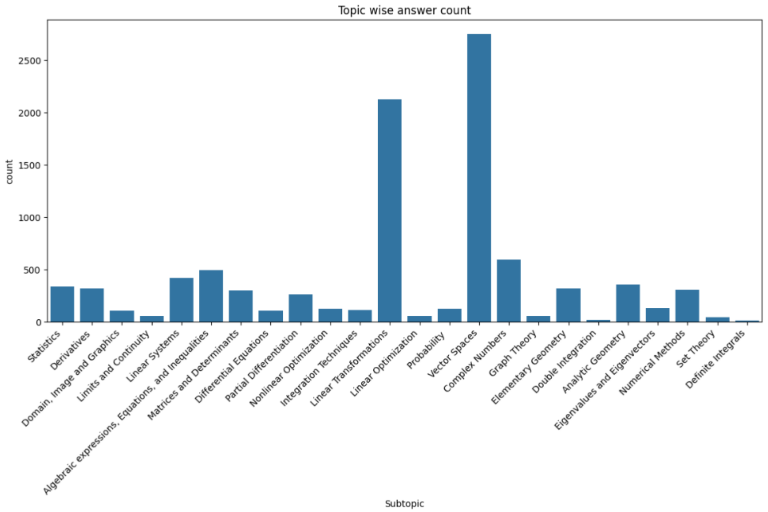
DATA VISUALIZATION





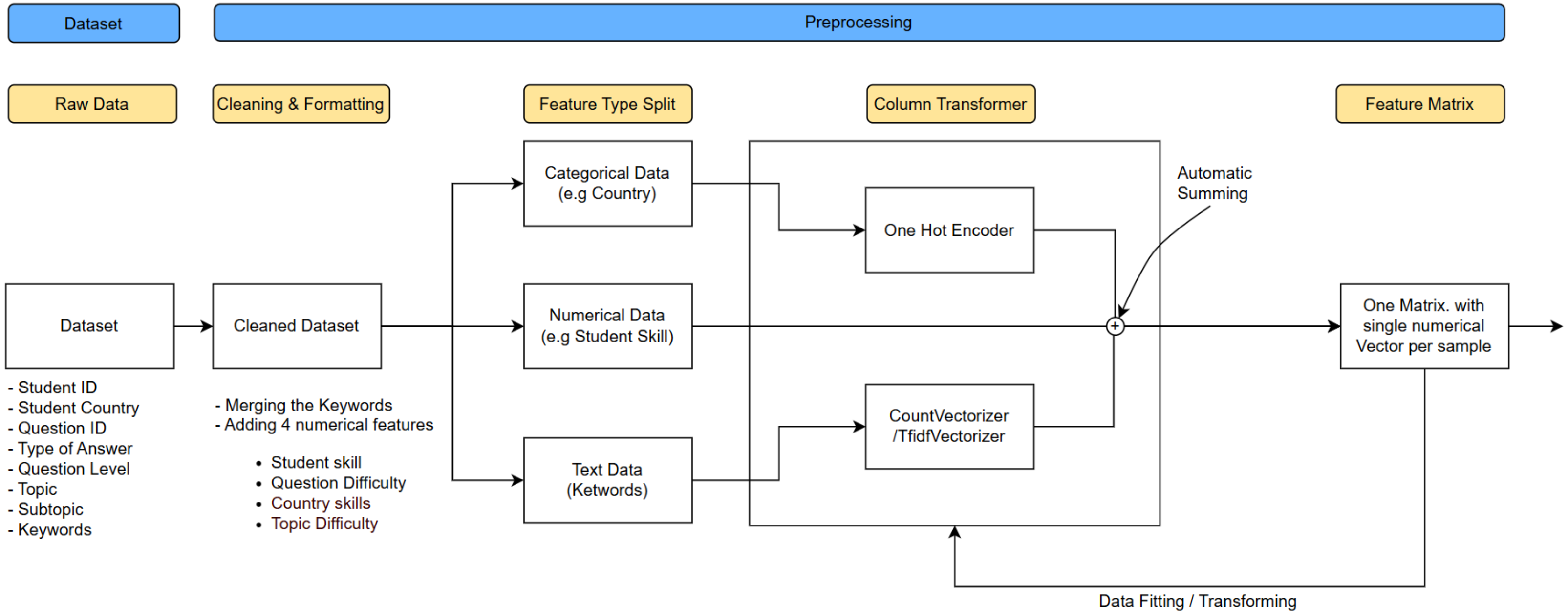






PREPROCESSING

- Data Cleaning & Formatting
- Feature Split
- Embedding
- Column Transformer

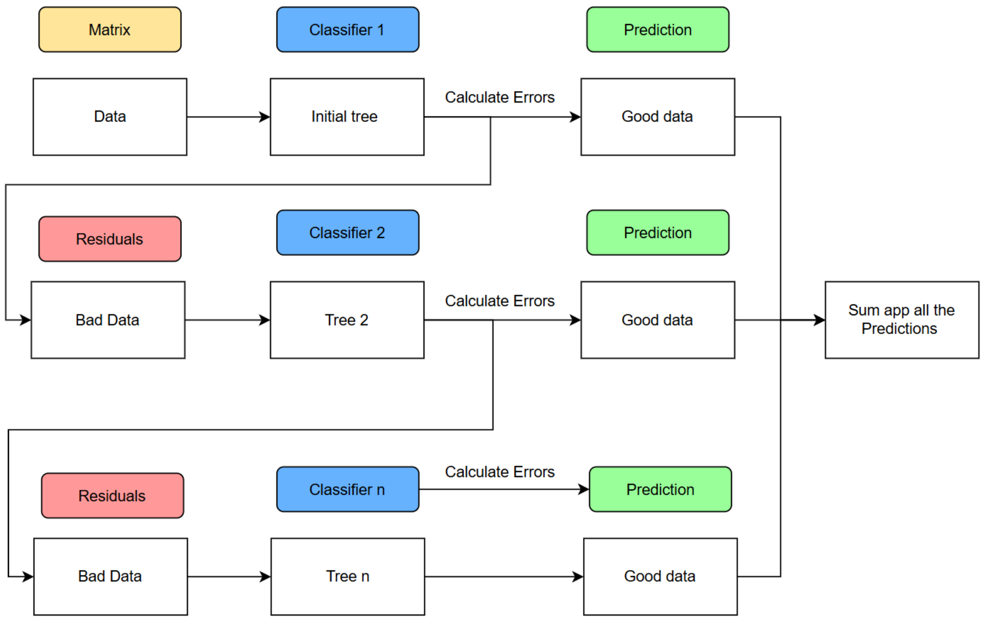


[3] Scikit-learn developers, "`sklearn.compose.ColumnTransformer`," *scikit-learn: Machine Learning in Python*, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.compose.ColumnTransformer.html.
[4] Scikit-learn developers, "`sklearn.preprocessing.OneHotEncoder`," *scikit-learn: Machine Learning in Python*, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html.
[5] Scikit-learn developers, "`sklearn.preprocessing.StandardScaler`," *scikit-learn: Machine Learning in Python*, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.
[6] Scikit-learn developers, "`sklearn.feature_extraction.text.TfidfVectorizer`," *scikit-learn: Machine Learning in Python*, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.
[7] Scikit-learn developers, "`sklearn.feature_extraction.text.CountVectorizer`," *scikit-learn: Machine Learning in Python*, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

THE MODEL

- eXtreme Gradient Boosting Classifier (XGBoost)
- Uses gradient descent to minimize loss function.

1. **Start with a weak model**, like a shallow decision tree.
2. **Calculate the error** (difference b/w predicted and actual values).
3. **Compute the gradient** of the loss function w.r.t. predictions—
   this tells us the direction to minimize the error.
4. **Train a new model** to predict this gradient (the residuals).
5. **Add this new model to the overall prediction**,
   adjusting with a learning rate (a small step).
6. Repeat steps 2–5 for a number of iterations.



[8] N. Nikolaev, "XGBoost Algorithm: Everything You Need to Know," Neptune.ai, Mar. 21, 2022. [Online]. Available: https://neptune.ai/blog/xgboost-everything-you-need-to-know

XGB CLASSIFIER

- We gradually increased number of classifiers to 500
- Decreased learning rate from 0.080 to 0.009

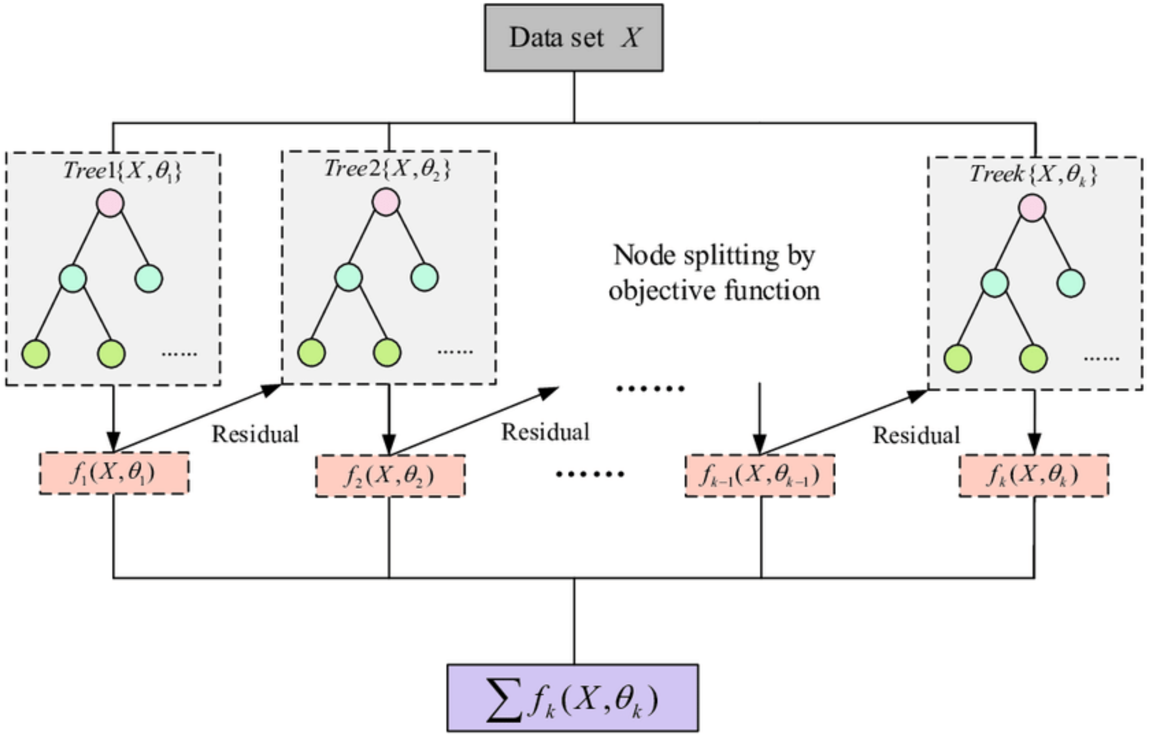
$$\text{Similarity Score (SS)} = \frac{(\sum \text{Residuals})^2}{\text{Number of Residuals} + \lambda}$$

$$\text{Gain} = \text{SS}_{\text{left}} + \text{SS}_{\text{right}} - \text{SS}_{\text{parent}}$$

$$\text{Output} = \frac{(\sum \text{Residuals})}{\text{Number of Residuals} + \lambda}$$

$$\lambda \uparrow \quad \Rightarrow \quad \text{SS} \downarrow \quad \Rightarrow \quad \text{Gain} \downarrow$$

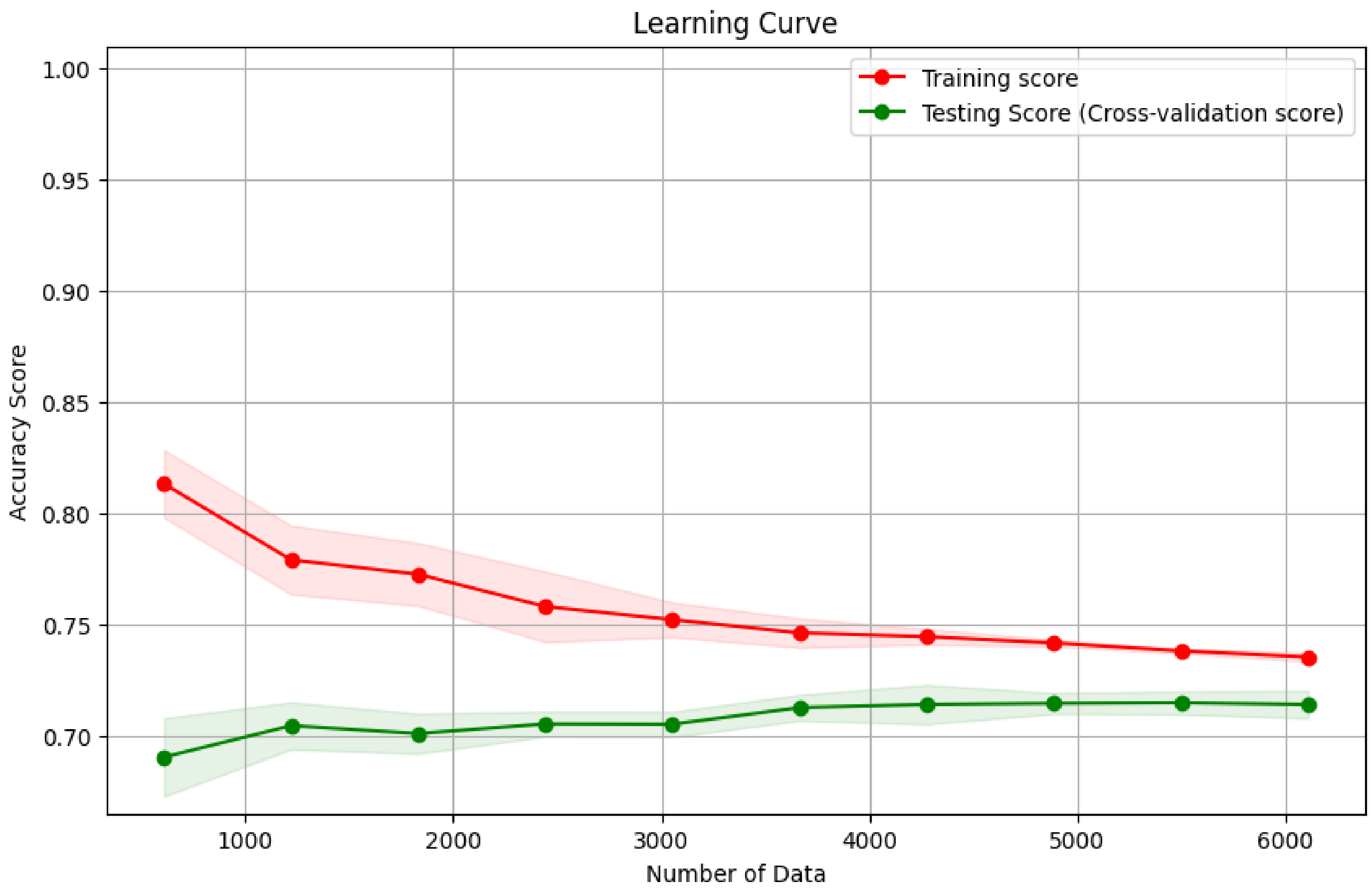
$$\lambda \uparrow \quad \Rightarrow \quad \text{Output} \downarrow$$



https://medium.com/@fraidoonomarzai99/xgboost-classification-in-depth-979f11ef4bf9

[8] N. Nikolaev, “XGBoost Algorithm: Everything You Need to Know,” Neptune.ai, Mar. 21, 2022. [Online]. Available: https://neptune.ai/blog/xgboost-everything-you-need-to-know

RESULT



Accuracy (with cross validation):
0.7235602094240837

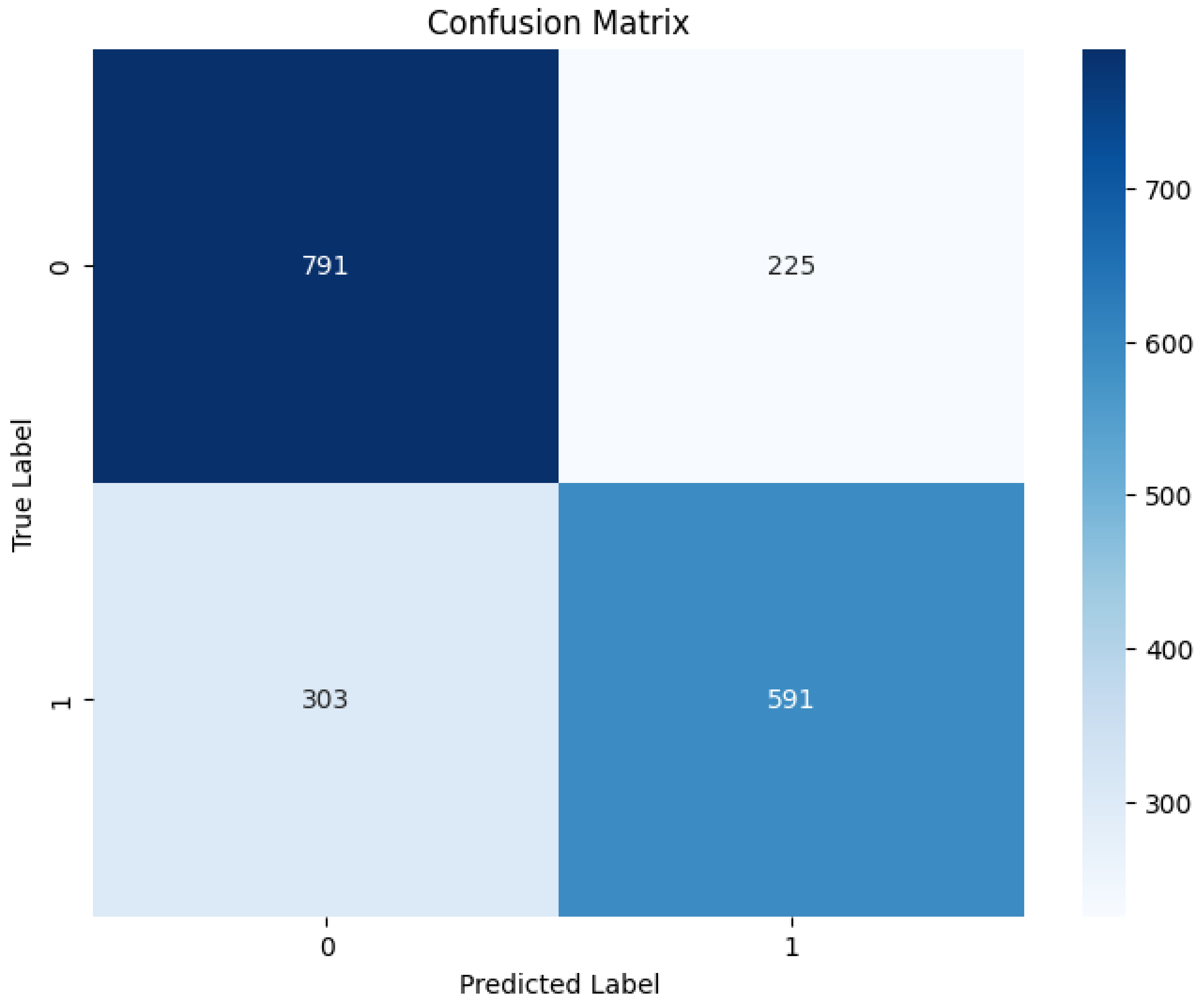
LEARNING CURVE

- TRAINING SCORE SHOWS THE MODEL'S ACCURACY ON THE TRAINING SET.
- IT STARTS HIGH (~0.82) AND SLOWLY DECREASES, STABILIZING AROUND 0.73 AS MORE DATA IS ADDED.

- TESTING SCORE REPRESENTS THE MODEL'S GENERALIZATION TO UNSEEN DATA.
- IT STARTS LOW (~0.69) AND SLOWLY INCREASES TO AROUND 0.72 AS TRAINING EXAMPLES INCREASE, INDICATING BETTER PERFORMANCE WITH MORE DATA.

THE FINAL VALIDATION ACCURACY IS 0.7236, WHICH IS THE AVERAGE TESTING SCORE

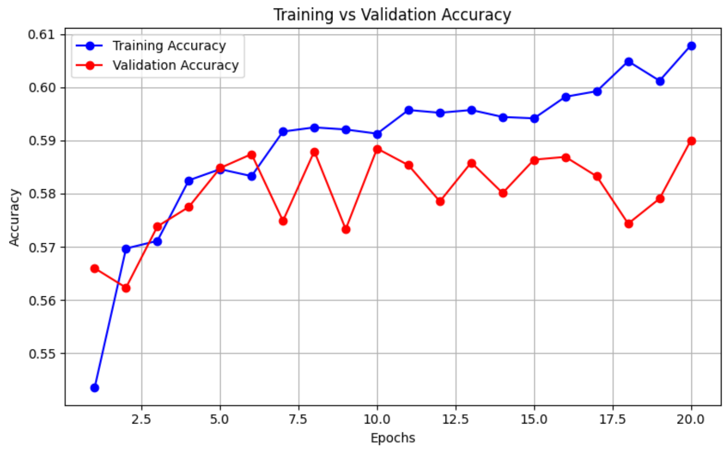
RESULT (continued…)

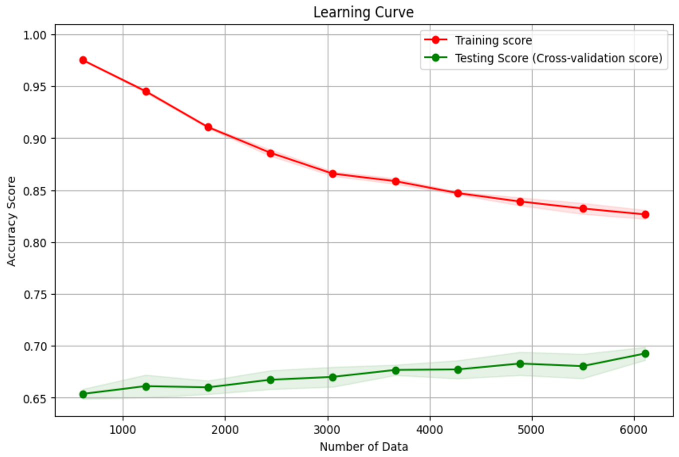


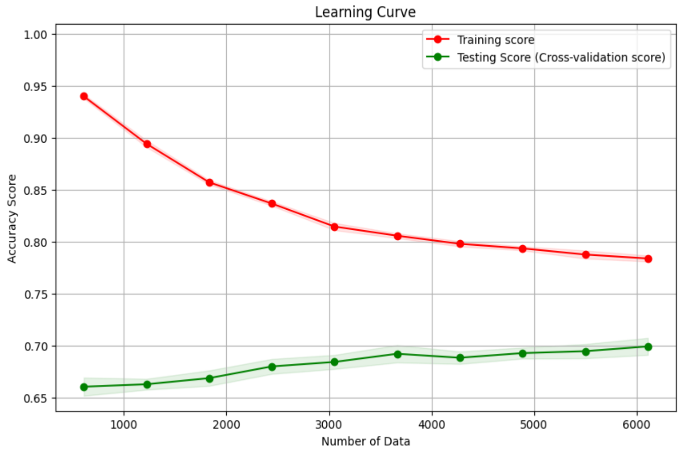
- **True Label 0, Predicted 0 (791)**: True negatives (TN) — The model correctly predicted class 0.
- **True Label 0, Predicted 1 (225)**: False positives (FP) — The model incorrectly predicted class 1 when it was actually class 0.
- **True Label 1, Predicted 0 (303)**: False negatives (FN) — The model incorrectly predicted class 0 when it was actually class 1.
- **True Label 1, Predicted 1 (591)**: True positives (TP) — The model correctly predicted class 1.

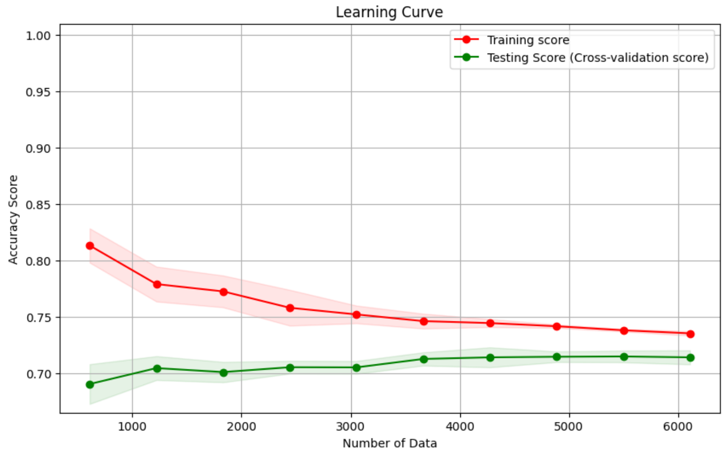
DISCUSSION

- 4 models – 1 Deep Learning, 3 XGB Classifier
- Deep Learning Model Accuracy: ~59%
- XGB Classifier Model Accuracy: ~68%, ~69%, **~72%**
- Changes in the number_of_classifiers and learning rate.
  - 1st Model: 400 Classifiers, Learning rate - 0.08
  - 2nd Model: 400 Classifiers, Learning rate 0.1
  - **3rd Model (Best): 500 Classifiers: Learning Rate: 0.0009**









TEAM RESPONSIBILITY

Team Members	Responsibility
Anand Aggarwal	- Visualization - Literature Review - Initial Models (DL)
Mustafa Alsenaidi	- Dataset Preprocessing - Literature Review - Visualizatoin
Yug Patel	- Preprocessing - Model Training - Initial Models
Abhishek Pandit	- Model Fitting - Parameter Tuning

CONCLUSION

- Searched for the Dataset and Related Paper
- Evaluated the Datasets
- Preprocessed the data using ColumnTransformer
- Fitted a model using XGB Classifier

Accuracy : ~72%

The End

Questions?