Towards Using Machine Learning To Improve Human Arithmetic Learning

Tessa Hall 19775539
Supervisor Dr H. Kamper

PROJECT AIM

Basic arithmetic is an essential skill in almost all careers, but many people worldwide do not have a strong foundation in this area. The development of personalised E-learning systems that focus on arithmetic could help to improve the basic numeracy ability of people from all walks of life. This personalisation can be achieved through the use of Machine Learning techniques.
The overarching aim of this research is to create an application that will accurately detect which areas of basic arithmetic a user struggles with, in order to generate targeted questions that may improve their abilities. Specifically, this project aims to determine the most accurate machine learning models to successfully classify future questions that a user is likely to get wrong. We hope that this technology will allow for the improvement of current arithmetic E-learning systems.

MODELS

Four classification models are investigated in this project and are implemented using the Scikit-learn machine learning library:

Decision Trees: similar to a flow chart, consisting of multiple nodes that test the point of interest against predetermined conditions to sort it into a class.

K-Nearest Neighbours: uses the classes of the K-nearest data points to determine the class of the point of interest. K is chosen as 3.

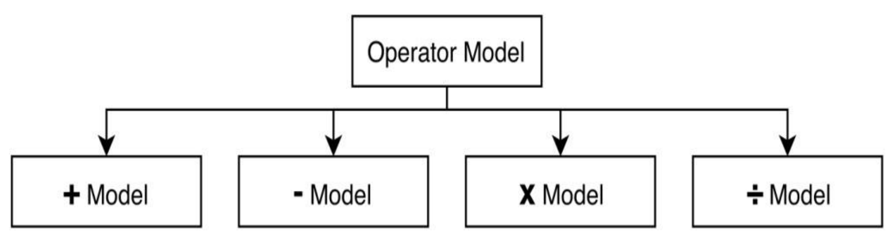
Logistic Regression: uses the sigmoid function to determine the probability that the point of interest lies in a specific class.

Neural Network: uses a Multi-Layer Perceptron (MLP) classifier with one hidden layer consisting of 100 neurons and the Adam optimization algorithm.

STRUCTURES

Non-Hierarchical: One model is trained on all types of question data and can be used to predict the expected results for any question operator.

Hierarchical: Five separate models are trained on segments of the question data as shown below. An initial model is trained using only the operator data from the questions. The other four models are assigned to different subdivisions of the training data according to operator (addition, subtraction, multiplication and division).



FEATURES

Three different feature vector representations that are derived from the arithmetic questions are evaluated for each model:

Base: the first four features provide a binary representation of the arithmetic operators. The remaining two features contain the arithmetic question numerical components.

High-Level: base features plus twelve additional features that provide information about potential prime factors of the arithmetic question numerical components. Aimed at improving the performance of Decision Trees.

Polynomial: base features plus seven additional polynomial features that are obtained by multiplying the question's numerical components by themselves and each other up to the third degree. Aimed at improving the performance of Logistic Regression models.

Arithmetic Questions

Agent Data Generator | Feature Vector Generator | Human Data Collection Application

Binary Classification Vectors | High level | Base | Polynomial | Binary Classification Vectors

Model Training

Non-Hierarchical Structure

Decision Trees | K-Nearest Neighbours | Logistic Regression | Neural Networks

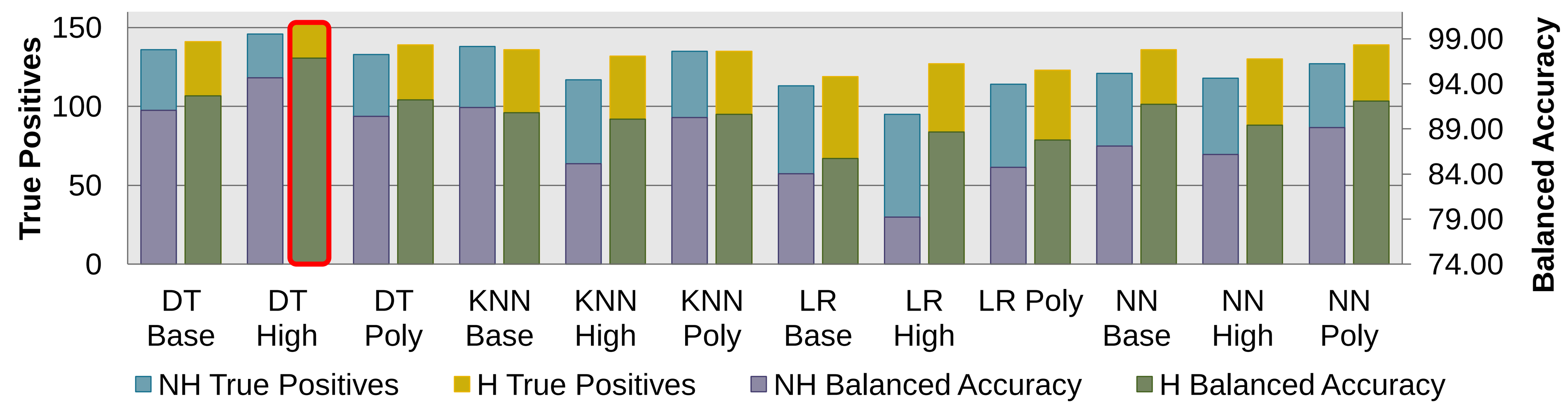
Hierarchical Structure

Addition / Subtraction / Multiplication / Division

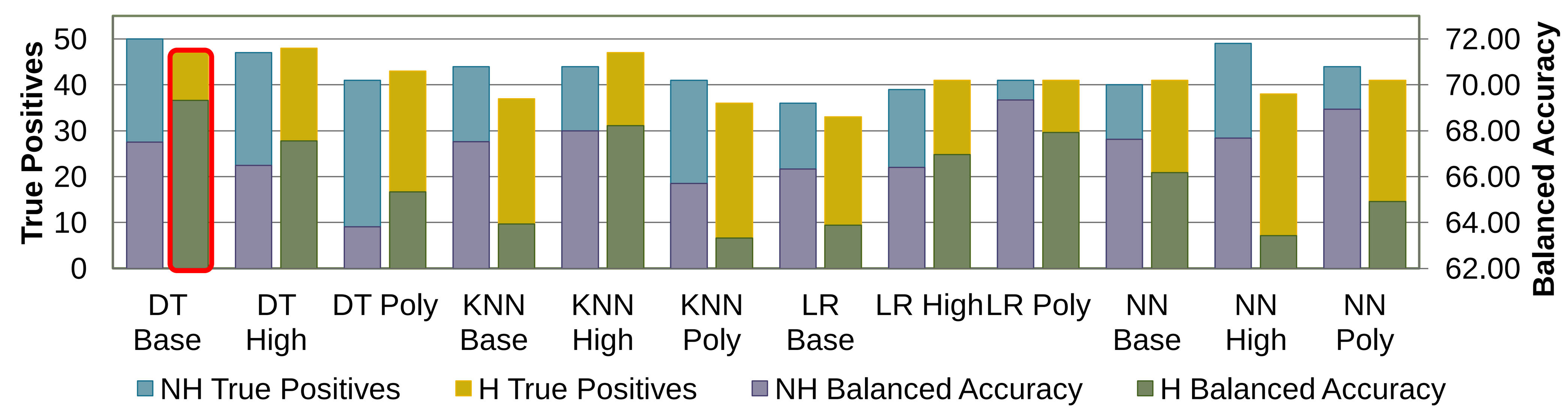
Decision Trees | K-Nearest Neighbours | Logistic Regression | Neural Networks

Model Results

Computer-generated Agent Results



Human User Results



Targeted Question Generation

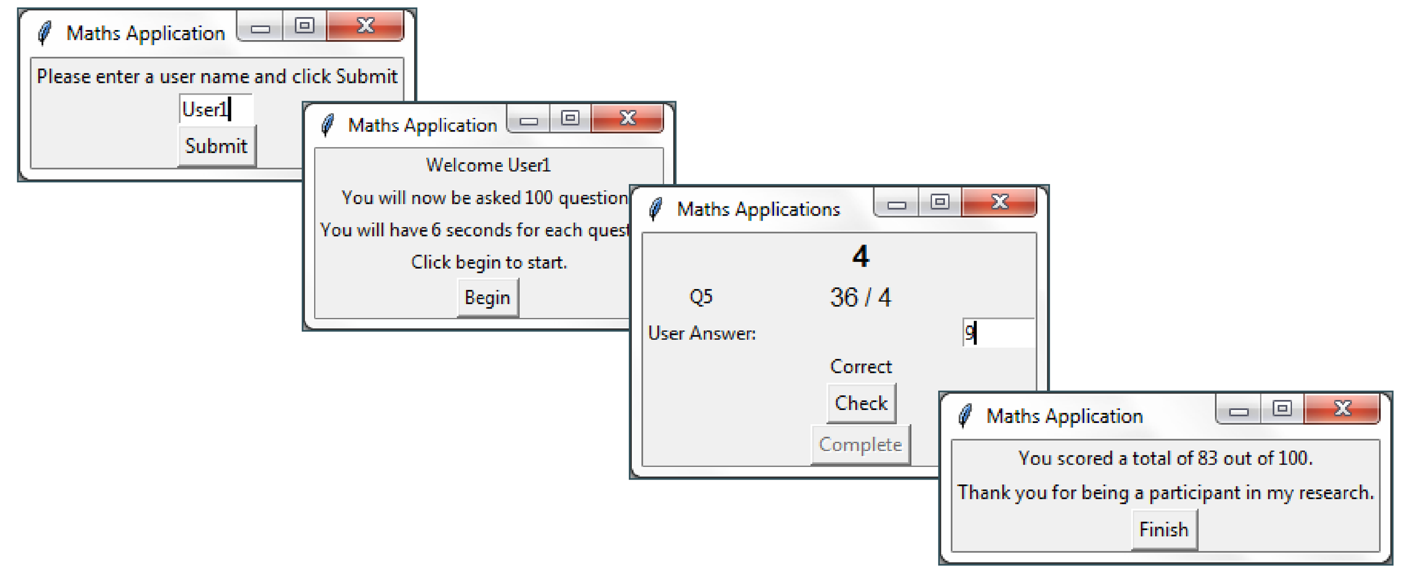
TRAINING DATA

The models are trained using a set of 50 randomly generated arithmetic questions and evaluated using another set of 50 questions. Two types of data are obtained during this process in order to train and evaluate the classification models:

Feature data: Contains information about each arithmetic question.

Binary classification vector: Incorrect answers are assigned a value of 0 and correct answers a value of 1 in this vector. This vector is obtained from two sources:

- **Computer-Generated Agent:** simulates human user responses and consistently answers specific types of questions incorrectly.
- **Human User:** answers questions using a computer application, which generates the binary classification vector based on the correctness of the user's answers.



EVALUATION

Confusion Matrix: a true positive result (0) indicates that the model successfully predicted that a question would be answered incorrectly. It is the most important of the four confusion matrix metrics.

	Actual Positive (0)	Actual Negative (1)
Predicted Positive (0)	True Positive (TP)	False Positive (FP)
Predicted Negative (1)	False Negative (FN)	True Negative (TN)

Balanced Accuracy: calculated using the metrics from the confusion matrix as follows:

$$\text{Balanced Accuracy }(\%) = 0.5 \times \left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \times 100$$

TARGETED QUESTIONS

The best model implementations are used to generate targeted questions. Arithmetic questions are randomly generated until the model predicts a predetermined number of questions that will be answered incorrectly.

Computer-Generated Agents: 17 of the 18 targeted questions matched the error patterns that the agents were designed to produce.

Human Users: the generated questions corresponded well with error patterns that were manually identified in the user answers.

CONCLUSION

The hierarchical decision tree model produced the best results for both the computer-generated agents and human users. The high-level feature system had a very high balanced accuracy of 96.9% on agent data. On human user data, the base feature system produced the most promising results with a balanced accuracy of 69.3%, still leaving some room for improvement.