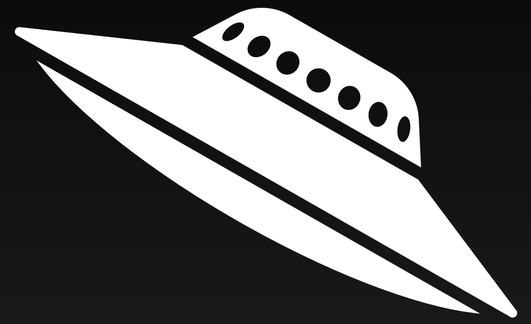


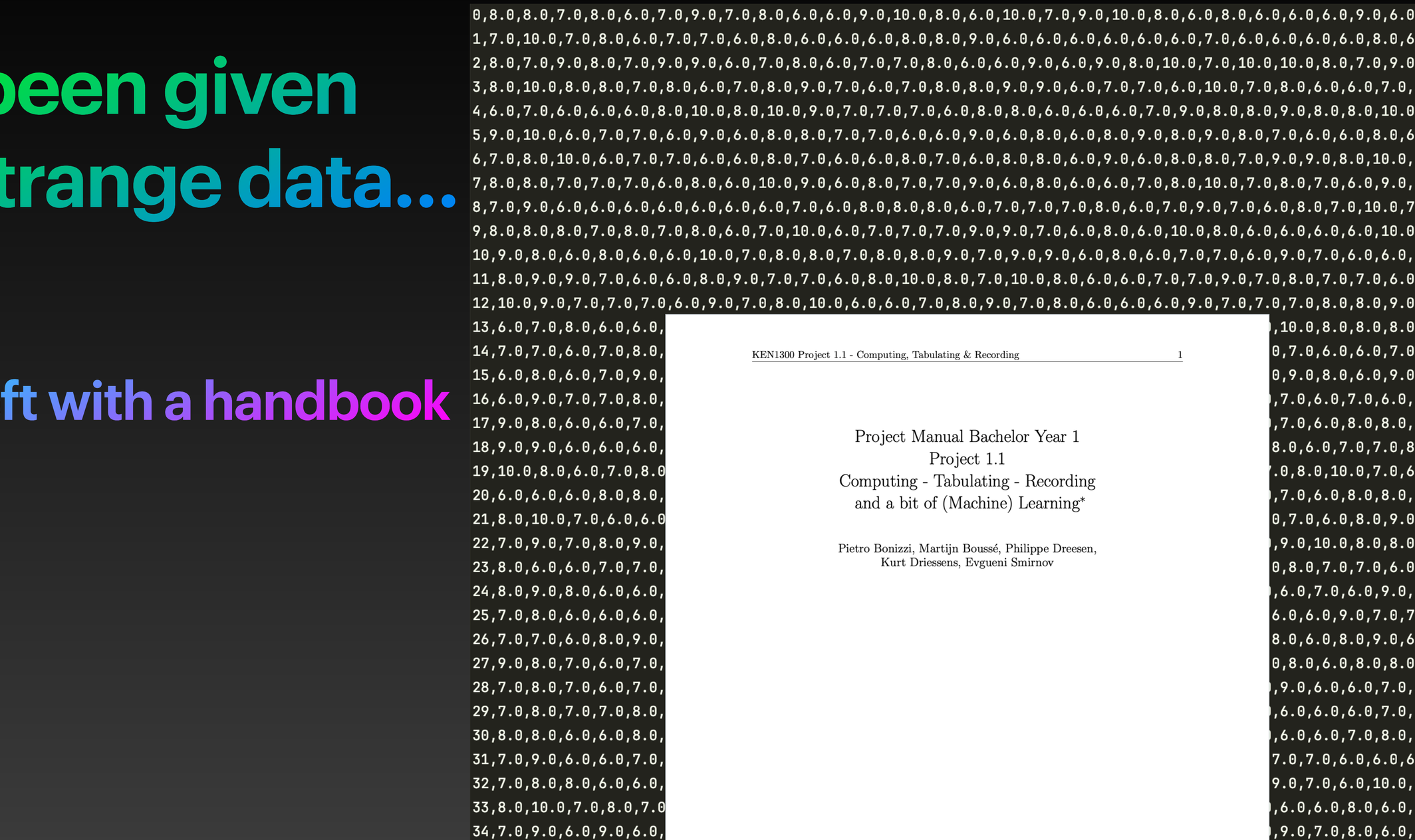
Alien-Students Grades

We’ve been invaded!

Project 1-1 by group KEN001
Octavian-Constantin Rujan, Ariannys Cermeño, Jakub Bujak, Jesse Hoydonckx, Kordian Klimas, Jakub Polichnowski

We've been given some strange data...

...and left with a handbook



KEN1300 Project 1.1 - Computing, Tabulating & Recording 1

Project Manual Bachelor Year 1
Project 1.1
Computing - Tabulating - Recording
and a bit of (Machine) Learning*

Pietro Bonizzi, Martijn Boussé, Philippe Dreesen,
Kurt Driessens, Evgueni Smirnov

Problem Statement

We cannot extract meaningful insights from the pure form of student grades data. The challenge lies within our capability of processing it in a way that allows us to understand and be able to interpret it.

Research Question

How can we derive and visualise comprehensive insights from those files across diverse subjects to improve our understanding of the data?

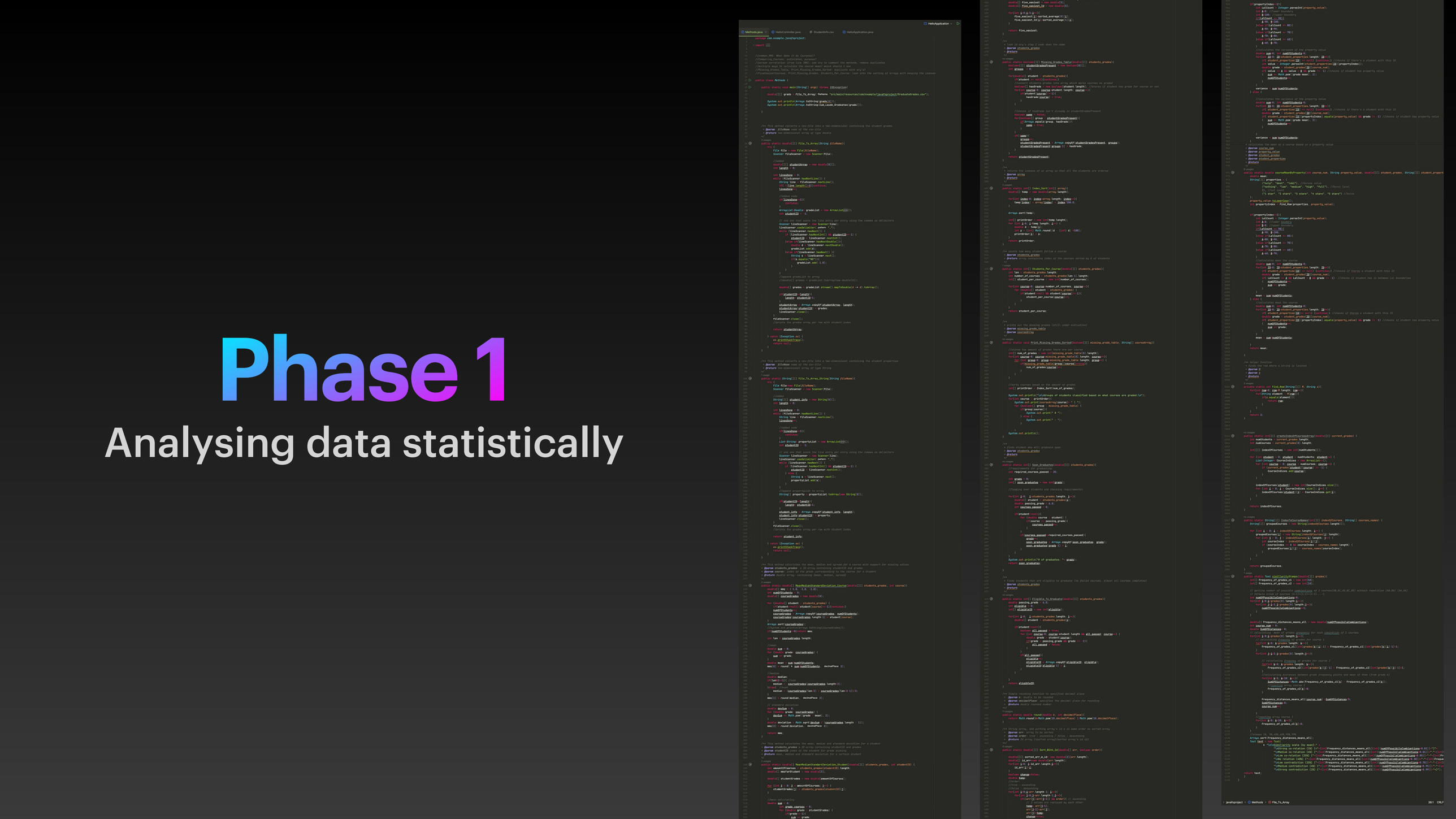
Table of Contents

1. Study Objectives
2. Approach
3. Methodology and Implementation
4. Outcomes and Implications

Study Objectives

- Identification of course difficulty levels.
- Identification of courses characteristics
- Predicting future student grades.
- Communicate Information Effectively

Our Approach


Phase 1
Analysing data statistically

Identification of course difficulty levels

Create and algorithm to Statistically analyse data

1. How can we effectively analyze the difficulty of courses?

2. How is it possible to identify the top-performing students?

3. How many students are eligible to graduate this year?

Identification of courses characteristics

Statistical Analysis and Analysis of Missing Information

4. Are there courses that seem similar or related?

5. Is there an order to courses in which they are taken?

Equations to statistically analyse data

Mean

$$\text{Mean } (\overline{x}) = \frac{\sum_{i=1}^{n} x_i}{n}$$

Standard Deviation

$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{n - 1}}$$

Variance

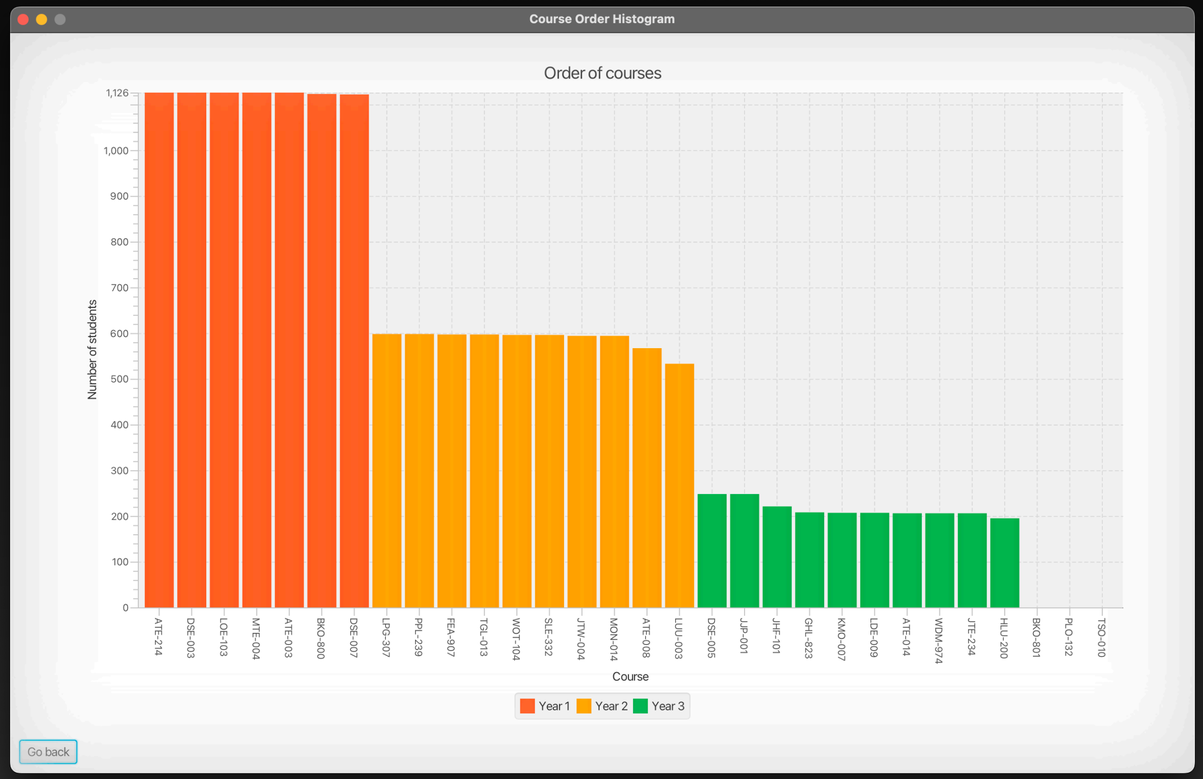
$$\sigma^2 = \frac{\sum (x_i - \mu)^2}{n}$$

Pearson Correlation

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

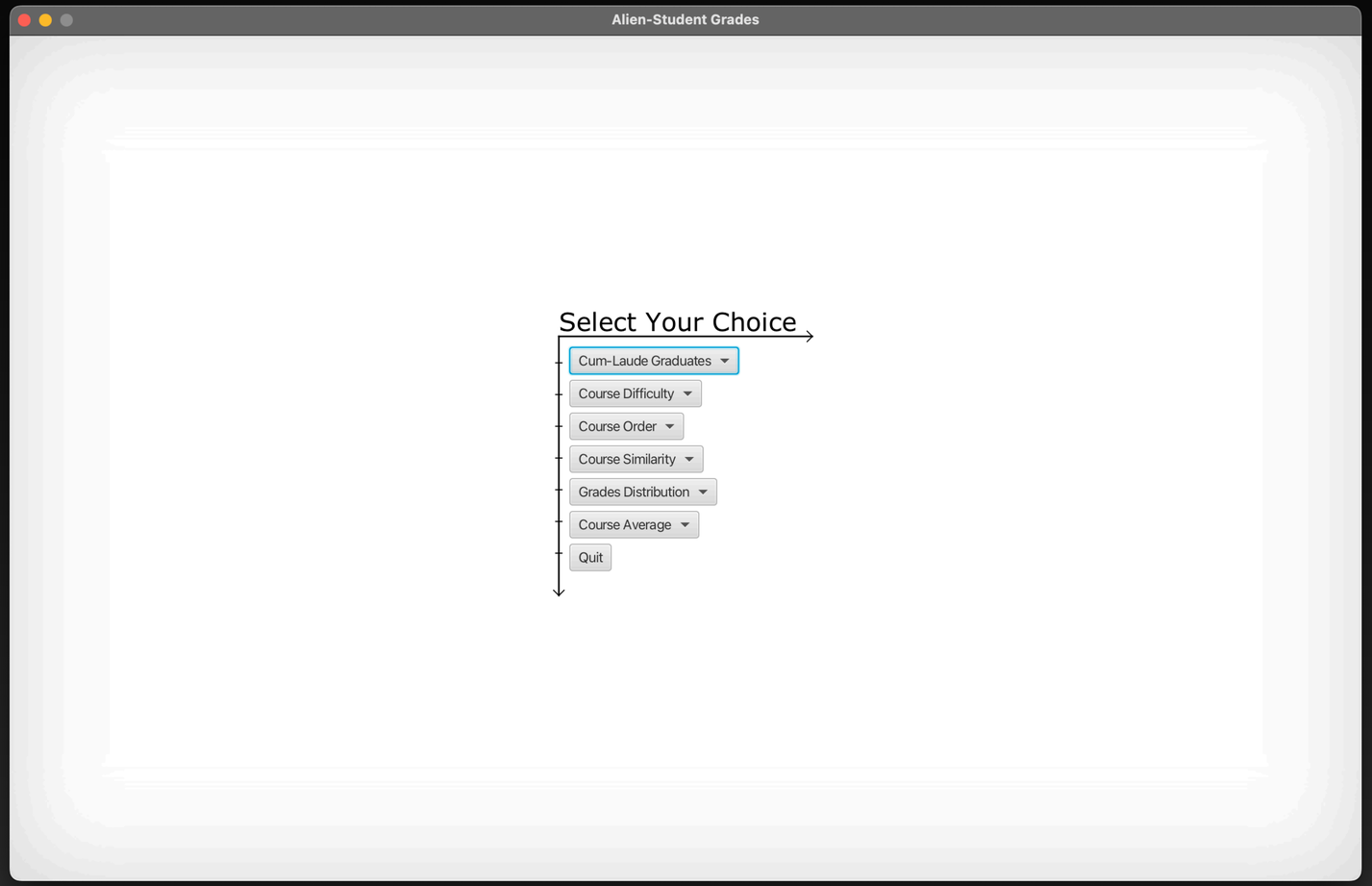
Analysis of Missing Information

- Attributing the Value -1
- Observing and analysing patterns in the data
- Drawing conclusions



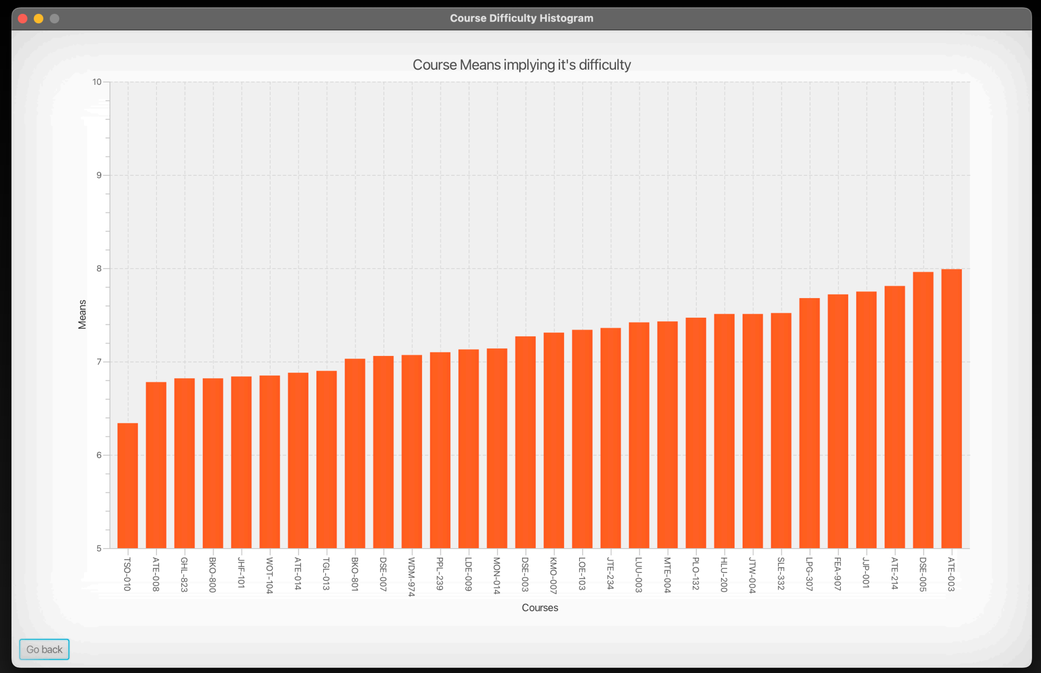
Phase 2

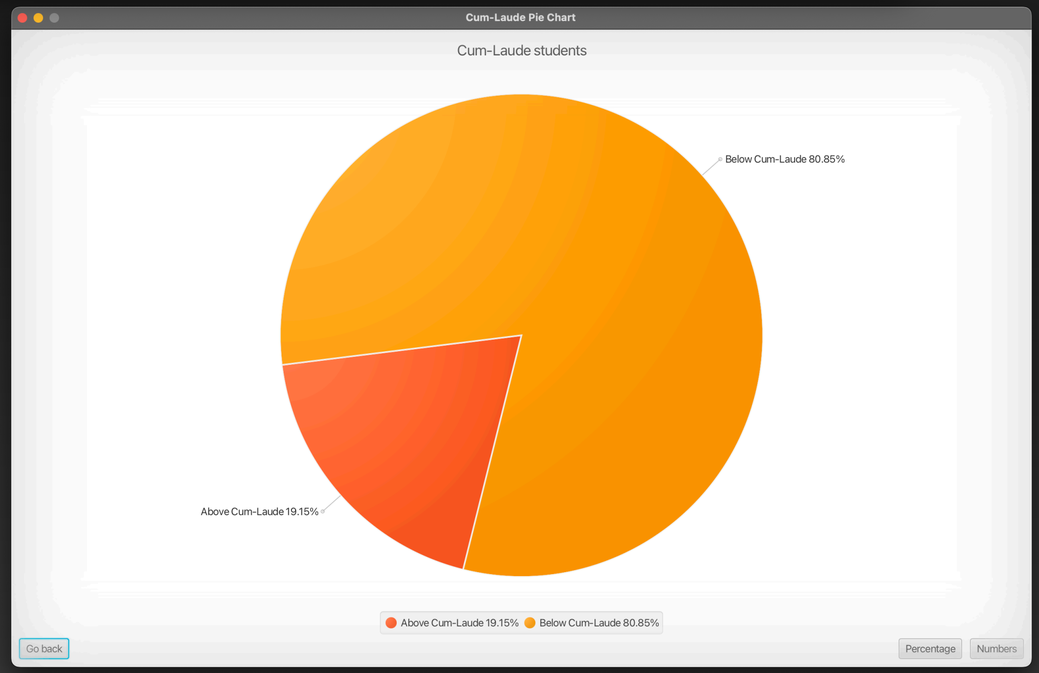
Communicate Information Effectively



Phase 2

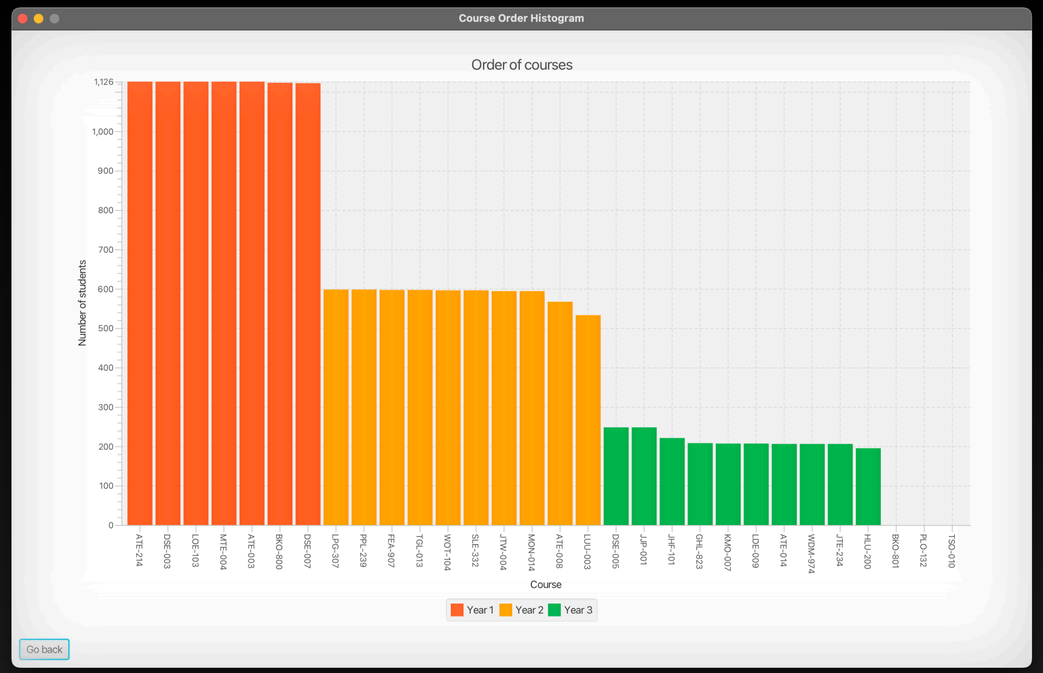
- Effectively analyse the difficulty of courses
- Identify the top-performing students

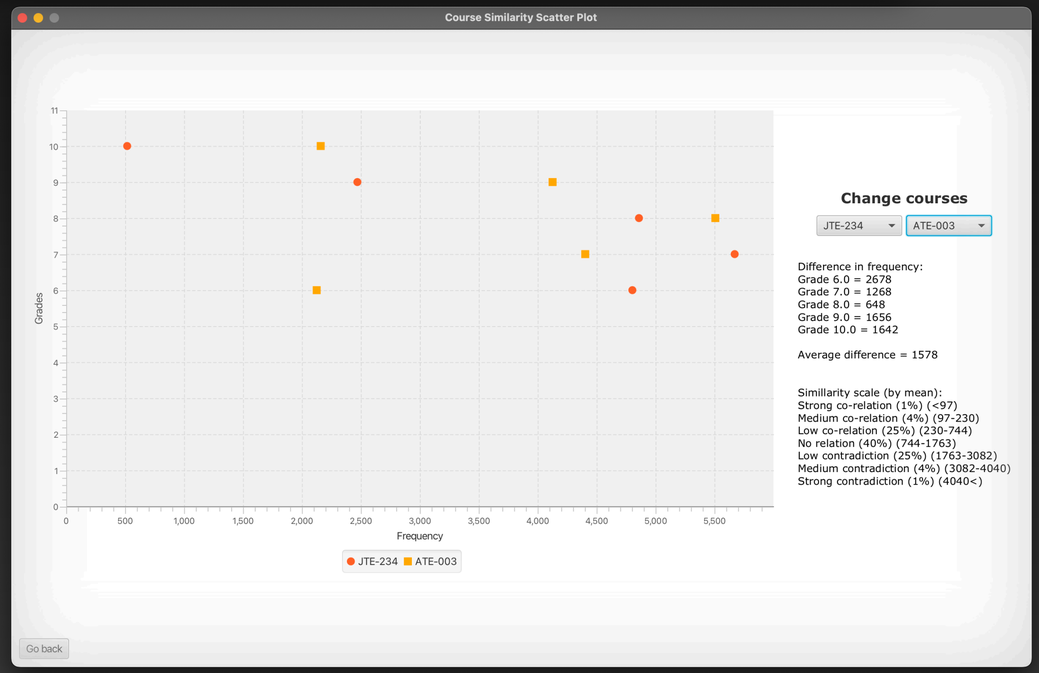




Phase 2

- Order of courses
- Courses that seem similar or related





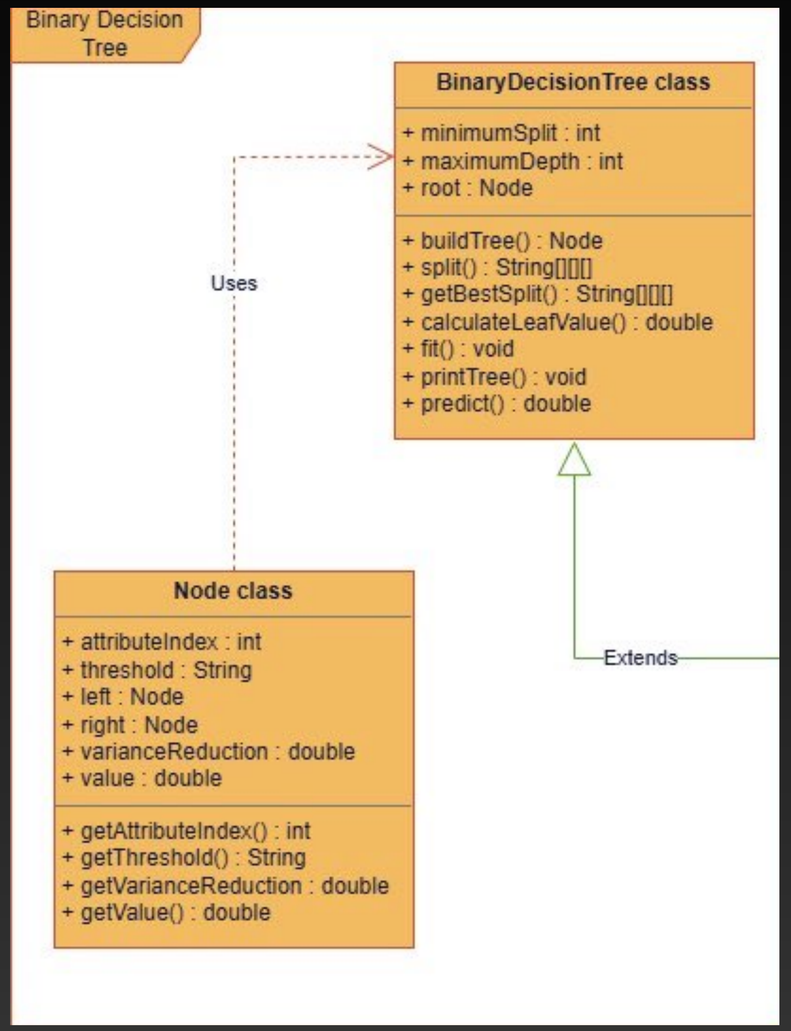
Phase 3

How can we predict the future performance of a student?

```
public class Phase3{

  public static void main(String[] args) {



        final int STUDENT_ID = 1000243;
        final int COURSE = 1;


        final int DEPTH = 3;
        final int SPLITS = 2;
        final boolean PRINT_TREE = false;


        final int FOREST_SIZE = 50;
        final int FOREST_DEPTH = 3;
        final int FOREST_SPLITS = 2;
        final int BOOTSTRAP_SIZE = 25;
        final int FOREST_VAR =  3;
        final boolean PRINT_FOREST = true;
```

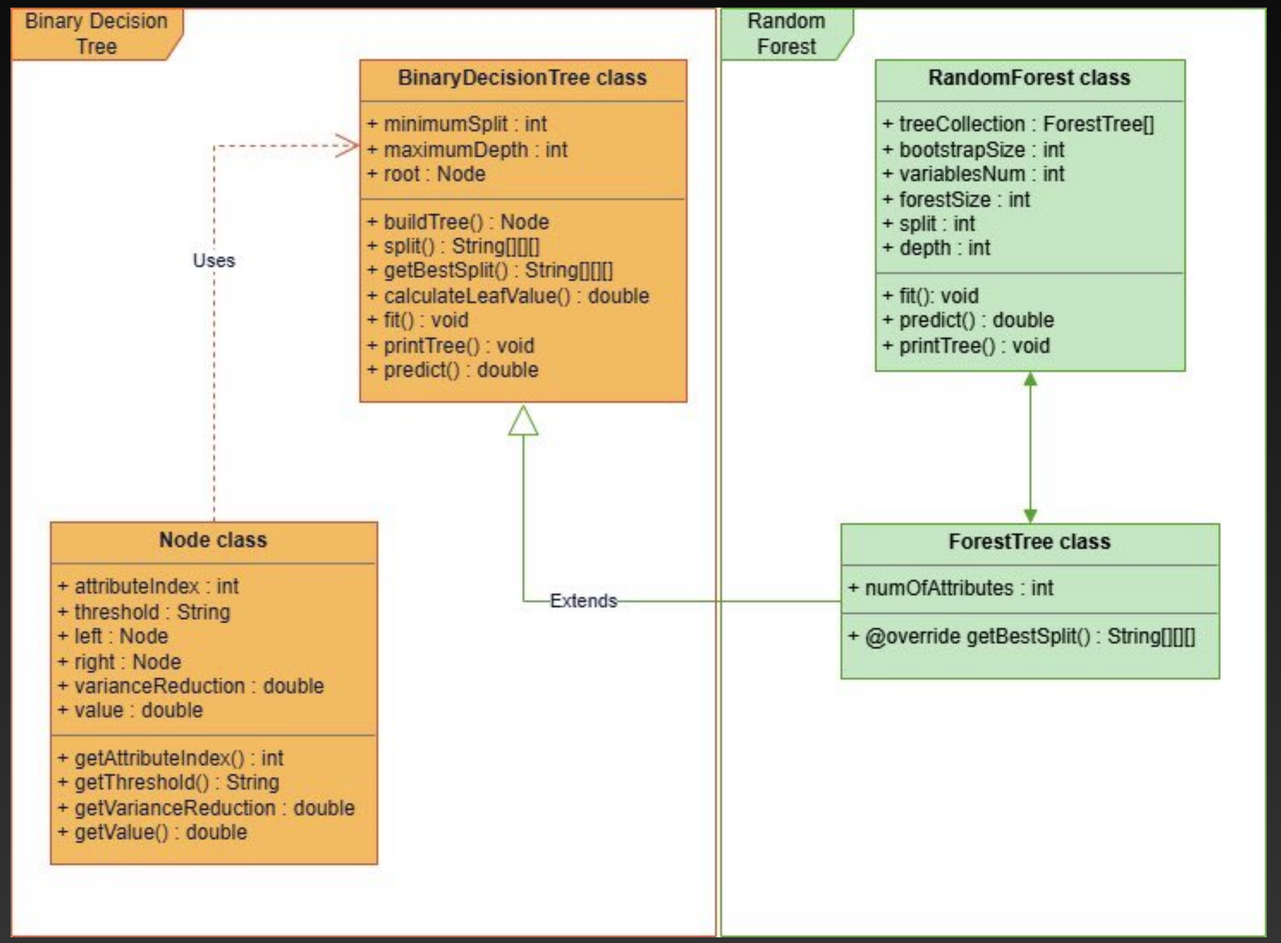
The Tree

- A customisable decision tree
  - Different depth
  - Multiple splits
- Predict a grade based on attributes



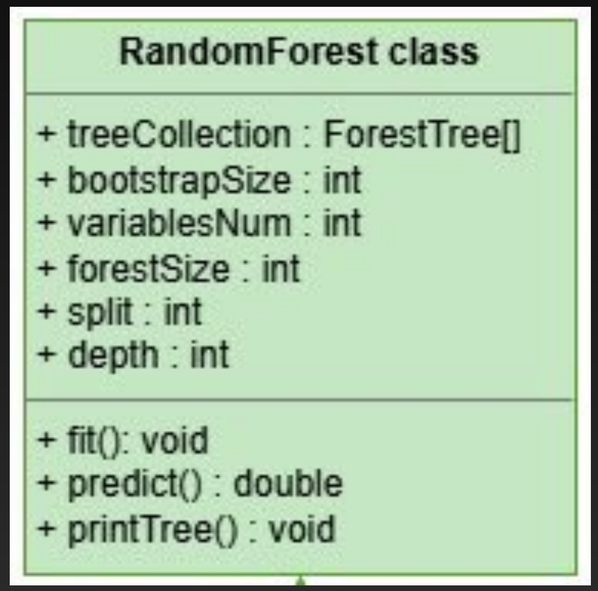
The Forest

- Consists of our decision trees
- Attribute sampling
- Bagging (bootstrap aggregating)



Configuring The Forest

- Size of the forest
- Number of selected attributes
- Size of bagging sets
- Depth of each tree
- Test to determine best value of each parameter



Conclusion & Outcomes

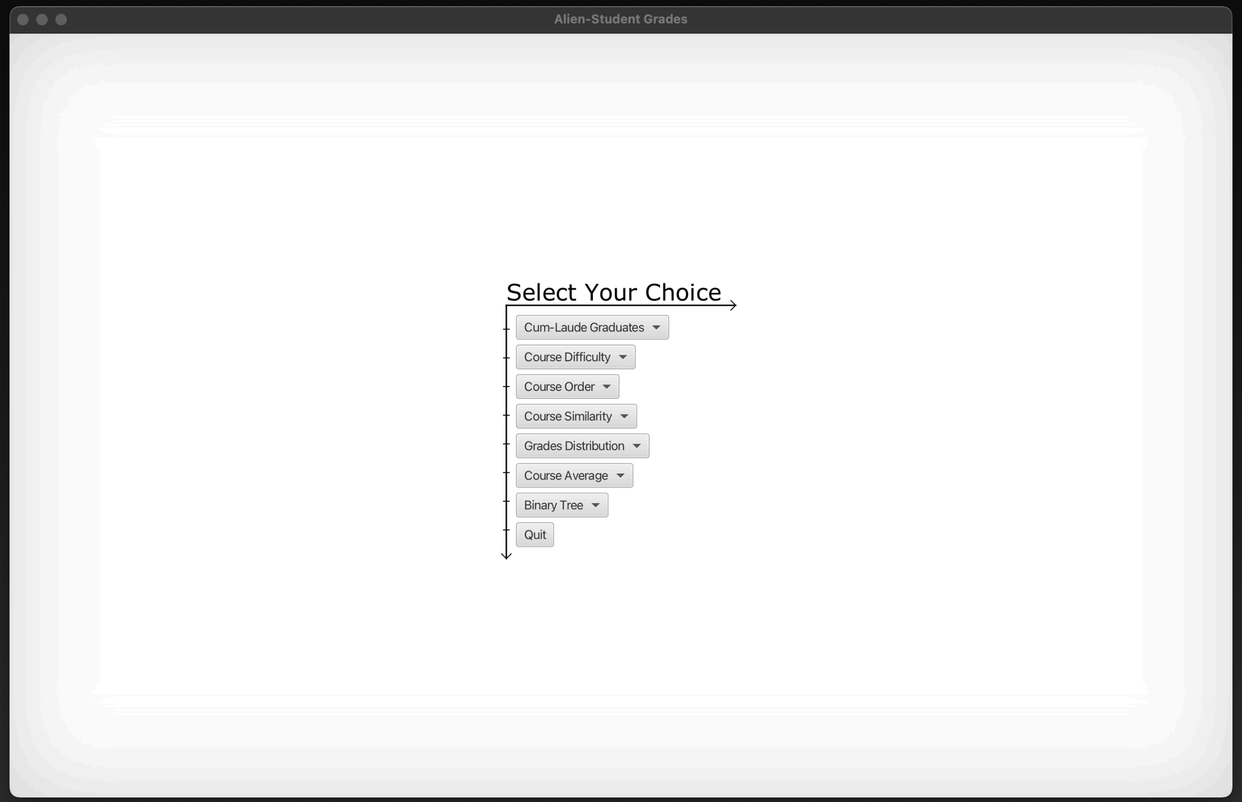
Predicted Grades

- Grades are predicted a random forest
- Accuracy is measured by testing the prediction against already existing grades

```
course: 1
properties: [lobi, nothing, 82, 1 star]
expected: 6.0
tree prediction: 6.246153846153846
forest prediction: 6.900841006850442

X_Hurni Level <= nothing ? 0.2302454155912277
left: X_Volta <= 3 stars ? 0.008310661460893987
  left: X_Suruna Value <= lobi ? 0.42173337372403374
    left: X_Volta <= 86 ? 0.01899485555995256
      left: 6.0
      right: 6.285714285714286
    right: X_Suruna Value <= nulp ? 0.2448979591836739
      left: 7.0
      right: 8.0
  right: X_Volta <= 62 ? 0.01702283730255516
    left: X_Suruna Value <= lobi ? 0.21301775147929009
      left: 6.0
      right: 7.0
    right: X_Suruna Value <= doot ? 0.39312514855448744
      left: 8.022727272727273
      right: 6.6976744186046515
right: X_Suruna Value <= lobi ? 0.38869482852043147
  left: X_Volta <= 77 ? 0.013643258310388484
    left: X_Hurni Level <= full ? 0.22278577943457678
      left: 8.344827586206897
      right: 7.239130434782608
    right: X_Hurni Level <= full ? 0.15663045260481767
      left: 7.897435897435898
      right: 7.020408163265306
  right: X_Suruna Value <= nulp ? 0.15457405036287963
    left: X_Hurni Level <= full ? 0.21045003786201721
      left: 9.063492063492063
      right: 8.016666666666667
    right: X_Hurni Level <= low ? 0.14621430208853026
      left: 8.366666666666667
      right: 9.281553398058252
tree accuracy: 0.5477779410566184
forest accuracy: 0.6347723066284731
```

The App

Final product of our work



Appendix

Who did what?

- Octavian-Constantin Rujan (Coding)
- Ariannys Cermeño (Presentation and Report)
- Jakub Bujak (Coding)
- Jesse Hoydonckx (Coding)
- Kordian Marcin Klimas (Coding)
- Jakub Tomasz Polichnowski (Coding and Presentation)

Octavian-Constantin Rujan

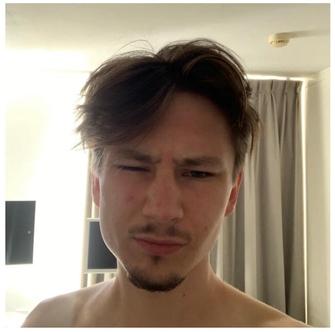
Antoni Starczynowski

Ariannys de los Angeles Cermeno Salas

Jakub Bujak

Jesse Hoydonckx

Kordian Marcin Klimas



Jakub Tomasz Polichnowski

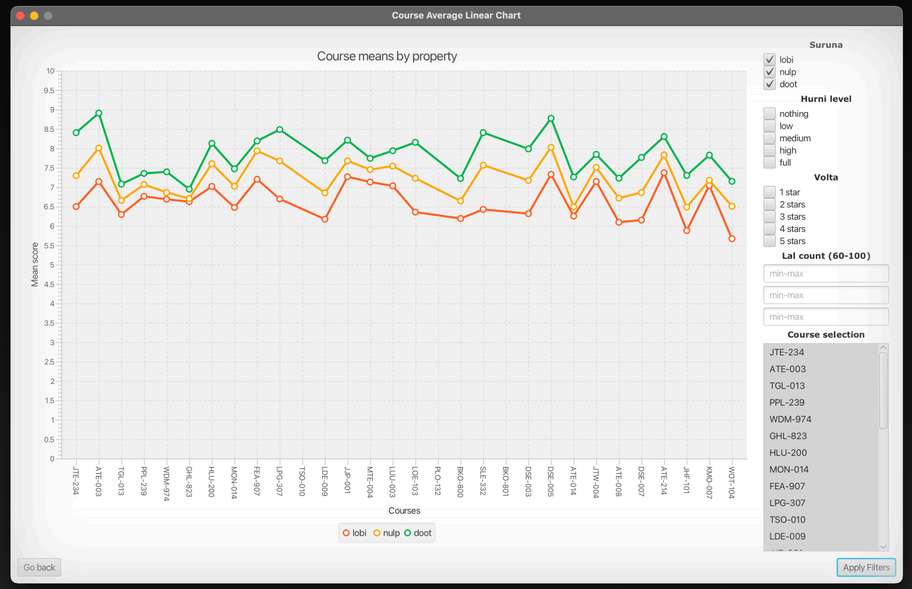
Limitations regarding the universality of the findings

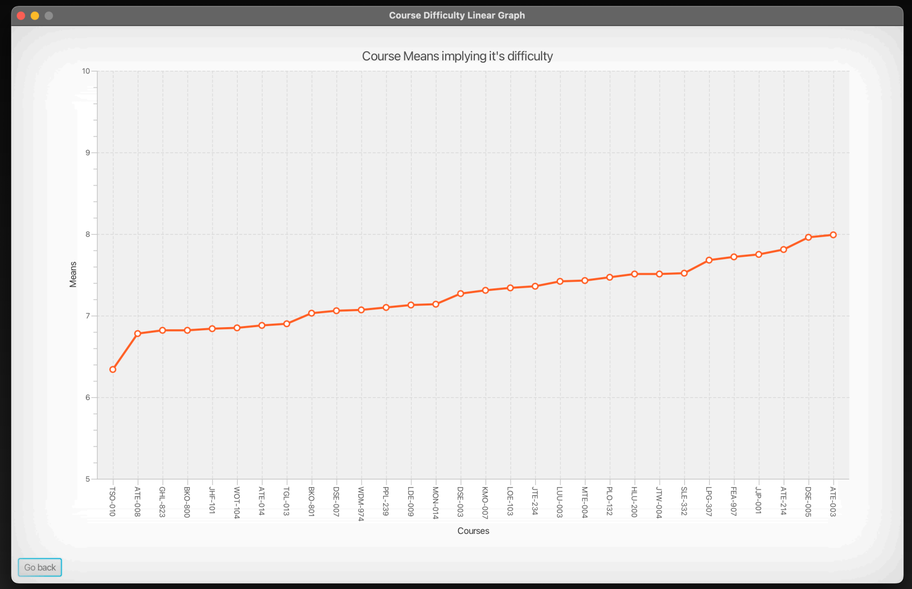
- Different grading systems
- Different data sets
- Limited Data provided

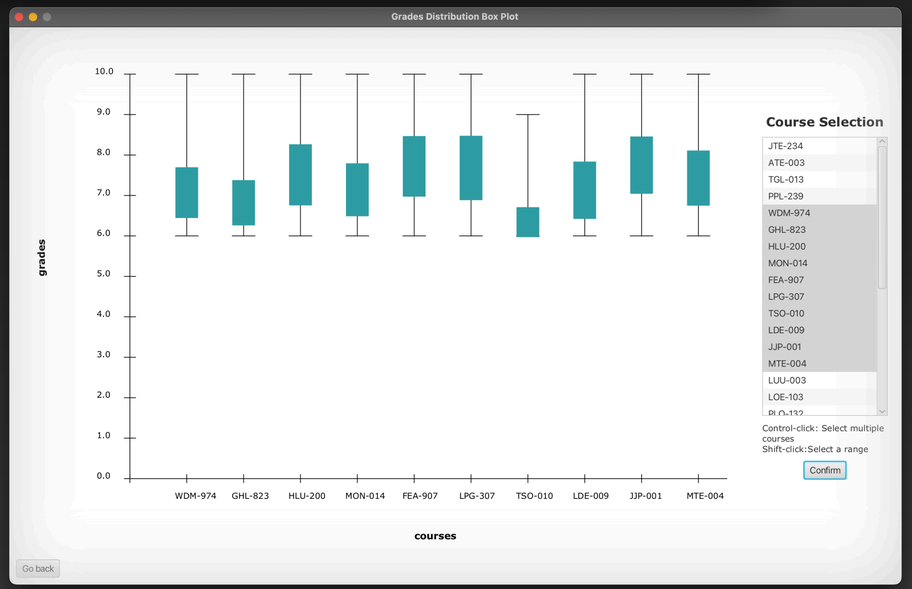
Why is the Java program necessary for educational objectives?

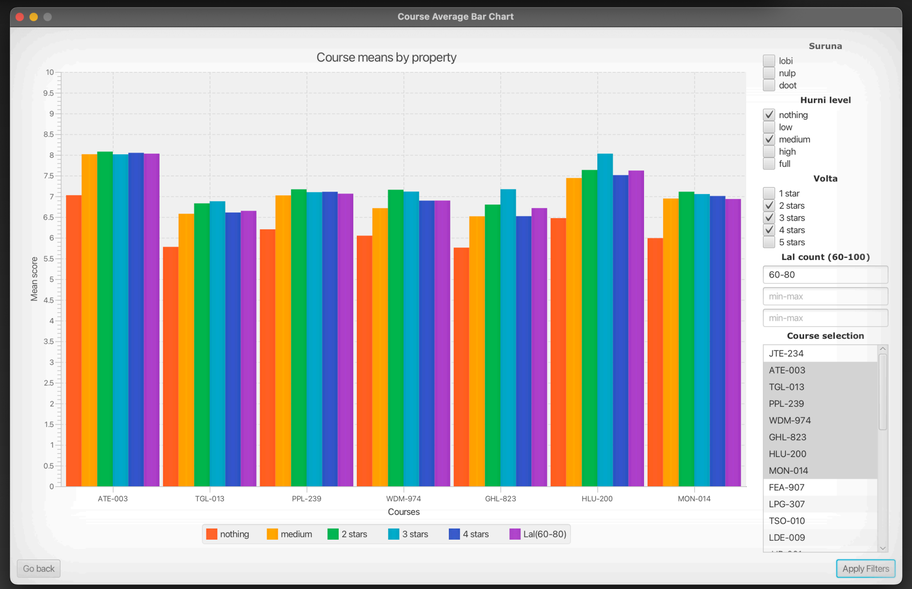
- Help educators offer support to students.
- Gives insight into the performance of teachers.
- Helps pinpoint subjects that schools should offer additional tutoring.
- Helps track the quality of the school system.

Other Graphs

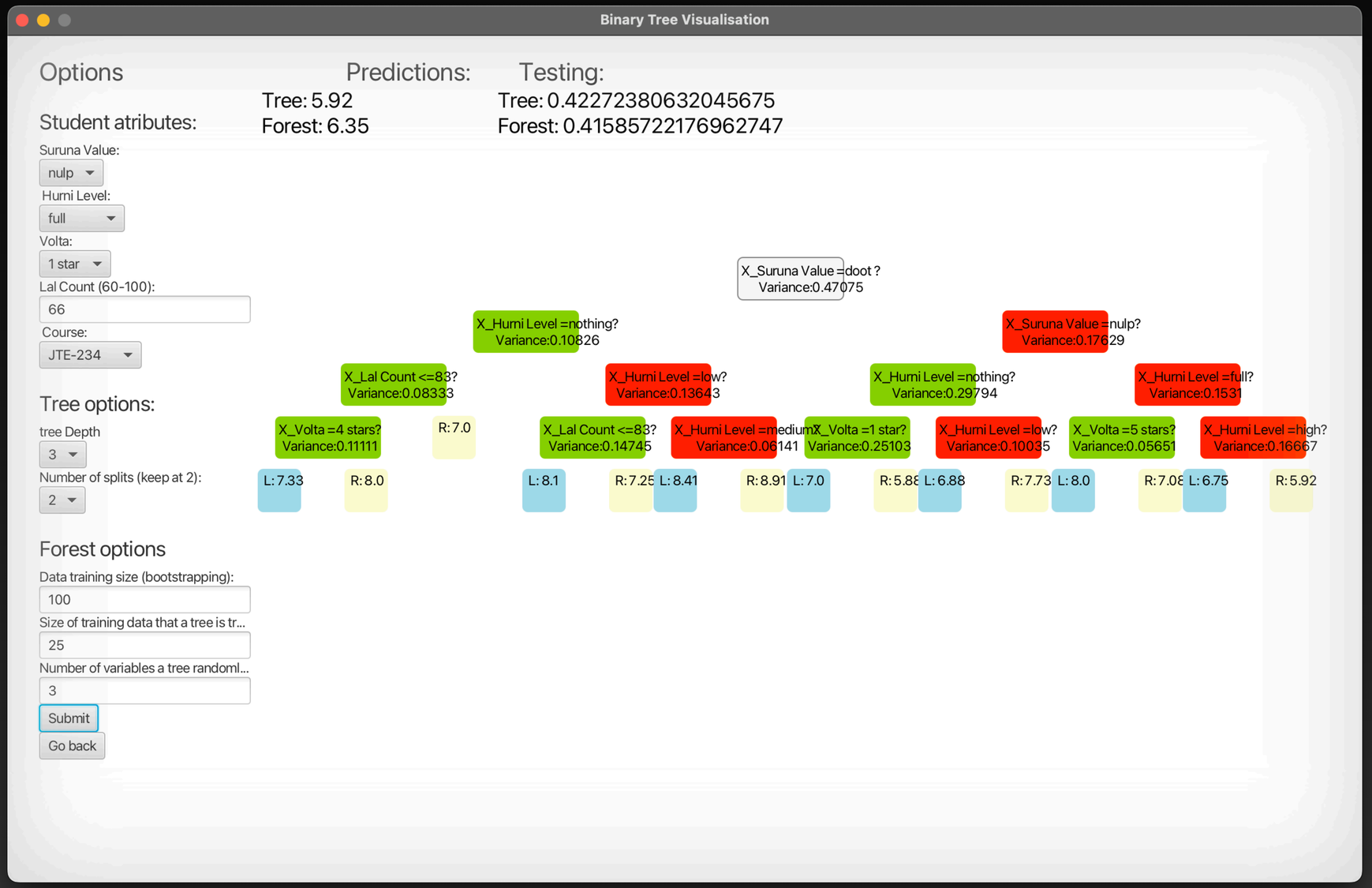








Binary Tree Visualisation



Test results forest variables

```
FOREST NUMBER OF VARIABLES TESTING
#variables: 1 -- accuracy: 0.6871435911090105
#variables: 2 -- accuracy: 0.6135426303238924
#variables: 3 -- accuracy: 0.5932685029495025
#variables: 4 -- accuracy: 0.6115626983799577

FOREST DEPTH TESTING
depht: 1 -- accuracy: 0.6624176454310068
depht: 2 -- accuracy: 0.6199750748711955
depht: 3 -- accuracy: 0.5918347455113 78
depht: 4 -- accuracy: 0.5954969981487844
depht: 5 -- accuracy: 0.613579306275516
depht: 6 -- accuracy: 0.6316411637589887
```

```
FOREST SIZE TESTING
size: 0 -- accuracy: NaN
size: 10 -- accuracy: 0.59596878556275 08
size: 20 -- accuracy: 0.5977283207362167
size: 30 -- accuracy: 0.5942868255474957
size: 40 -- accuracy: 0.5921901090151988
size: 50 -- accuracy: 0.5930870185676718
size: 60 -- accuracy: 0.5929506414430784
size: 70 -- accuracy: 0.5908509234010015
size: 80 -- accuracy: 0.5896695535650317
size: 90 -- accuracy: 0.591559523021654
size: 100 -- accuracy: 0.5905664345294402

BOOTSTRAP SIZE TESTING
size: 10 -- accuracy: 0.5920437974733779
size: 20 -- accuracy: 0.5915116048054091
size: 30 -- accuracy: 0.5910071476239593
size: 40 -- accuracy: 0.5927582825988333
size: 50 -- accuracy: 0.5917886533257734
```

forest depth: 3, splits for every tree: 2, bootstrap size: 25, forest size: 50, number of variable: 3

Method to Build the Decision tre

1) Extract the attributes (X) and course values (Y) from the data

2) While stopping conditions are not met

2.1) Check if the best split is valid

2.2) Build recursively the left and right subtrees

2.3) Returns a decision node with the current split data

3) Returns leaf node with a prediction value

Method to find the best split based on variance reduction

1) Initialise a String[][][] array to assign the BestSplit

2) Loop over all features and get the unique values of the current feature

3) Loop over all unique feature values and get the current split

3.1) If children are not null

3.2)Extract labels (y, left_y, right_y)

3.3) Compute variance reduction

3.4)If variance reduction is higher, assign the current Split to the BestSplit

4) Return Bestplit