# **Measures of Success:** Determining the most significant predictors of student achievement

*An analysis of student achievement data in secondary education of two Portuguese schools courtesy of UC Irvine Machine Learning Repository*

Seth Dalmacio, Rachel Brown, Talibah Timothy, Liliana Perez Diaz, Nedal Rashid

# Project Overview

*In this project we sought to answer,* ***what are the most important predictors that will determine whether a student passes or fails?***

Our Methodology:

1. Clean data
2. Use a classification algorithm (Logistic regression, K-Means Clustering, etc) to determine the variables that appear to have the most impact on student achievement
3. Create and train a model to predict student outcomes
4. Visualize findings
5. Evaluate model performance

**Dataset Used**: https://archive.ics.uci.edu/dataset/320/student+performance

**Tools Used**: Pandas, Matplotlib, Seaborn, Scikit learn

# Data cleaned to ensure consistency and usability

**Purpose:** Ensure the dataset is consistent, understandable, and ready for analysis by removing irrelevant or redundant data and handling missing values.

1

**Remove**

Remove unnecessary columns identity those that do not contribute to analysis

2

**Rename**

Make column names more consistent and descriptive

3

**Address inconsistencies**

Handle missing or inconsistent values; check for missing data and either fill in with suitable value or drop

4

**Scale**

Normalize and scale numerical features

5

**Export**

Export cleaned dataset for use in analysis

# Data cleaned to ensure consistency and usability

Before cleaning

| school;sex;age;address;famsize;Pstatus;Medu;Fedu;Mjob;Fjob;reason;guardian;traveltime;studytime;failures;schoolsup;famsup;paid;activities;nursery;h |
|---|
| GP;"F";18;"U";"GT3";"A";4;4;"at_home";"teacher";"course";"mother";2;2;0;"yes";"no";"no";"no";"yes";"yes";"no";"no";4;3;4;1;1;3;6;"5";"6";6 |
| GP;"F";17;"U";"GT3";"T";1;1;"at_home";"other";"course";"father";1;2;0;"no";"yes";"no";"no";"no";"yes";"yes";"no";5;3;3;1;1;3;4;"5";"5";6 |
| GP;"F";15;"U";"LE3";"T";1;1;"at_home";"other";"other";"mother";1;2;3;"yes";"no";"yes";"no";"yes";"yes";"yes";"no";4;3;2;2;3;3;10;"7";"8";10 |
| GP;"F";15;"U";"GT3";"T";4;2;"health";"services";"home";"mother";1;3;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"yes";3;2;2;1;1;5;2;"15";"14";15 |
| GP;"F";16;"U";"GT3";"T";3;3;"other";"other";"home";"father";1;2;0;"no";"yes";"yes";"no";"yes";"yes";"no";"no";4;3;2;1;2;5;4;"6";"10";10 |
| GP;"M";16;"U";"LE3";"T";4;3;"services";"other";"reputation";"mother";1;2;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";5;4;2;1;2;5;10;"15";"15";15 |

After cleaning

| school | sex | age | Parent_sta | Mother_Ed | Father_Ed | Mjob | Fjob | reason | traveltime | Study_Tim | failures | schoolsup | famsup | paid | activities | nursery | higher | internet |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GP | F | 18 | A | 4 | 4 | at_home | teacher | course | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| GP | F | 17 | T | 1 | 1 | at_home | other | course | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| GP | F | 15 | T | 1 | 1 | at_home | other | other | 1 | 2 | 3 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| GP | F | 15 | T | 4 | 2 | health | services | home | 1 | 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| GP | F | 16 | T | 3 | 3 | other | other | home | 1 | 2 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| GP | M | 16 | T | 4 | 3 | services | other | reputatior | 1 | 2 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| GP | M | 16 | T | 2 | 2 | other | other | home | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

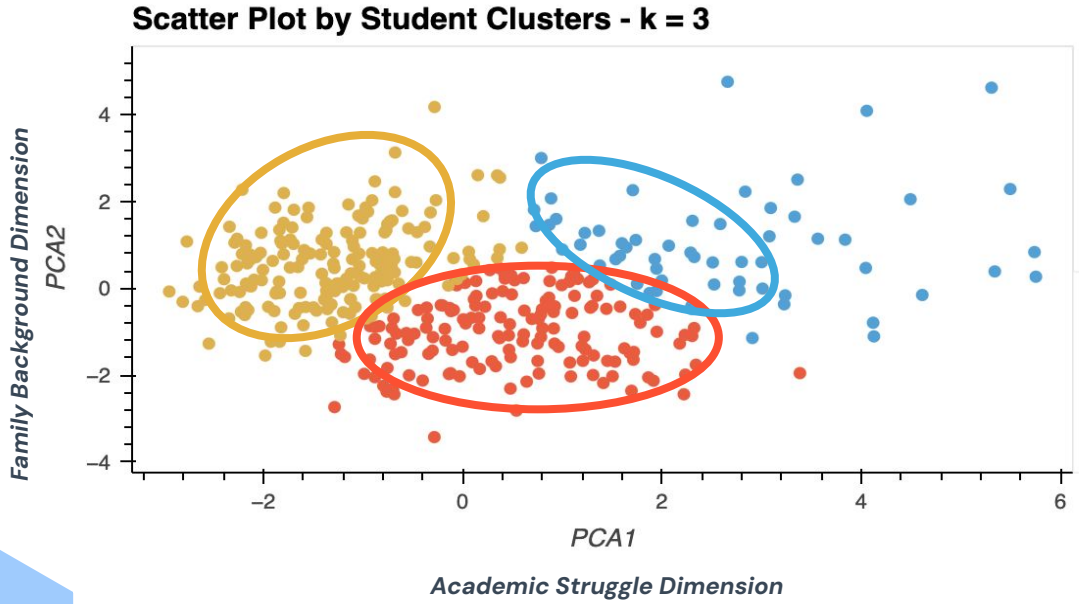
# Data cleaned to ensure consistency

```
[1]: import pandas as pd
     from sklearn.preprocessing import MinMaxScaler

[2]: # Load the dataset
     data = pd.read_csv('Resources/student-mat.csv', delimiter=';')

[3]: # Remove unnecessary columns
     columns_to_remove = ["G1", "G2", "Walc", "address", "famrel", "Dalc", "guardian", "famsize"]
     data.drop(columns=[col for col in columns_to_remove if col in data.columns], errors='ignore', inplace=True)

[4]: # Rename columns
     data.rename(columns={"G3": "final_grade", "studytime": "Study_Time_Hours", "Fedu": "Father_Edu", "Medu": "Mother_Edu", "Pstatus"

[5]: # Convert 'yes'/'no' to 0's and 1's
     binary_columns = ['schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'romantic']
     for col in binary_columns:
         if col in data.columns:
             data[col] = data[col].map({'yes': 1, 'no': 0})

[6]: # Handle missing values: Fill with mean for 'absences'
     if 'absences' in data.columns:
         data['absences'].fillna(data['absences'].mean(), inplace=True)
```

- Clean Data
- Rename columns
- Remove columns
- Prepare data for for rest of the steps

**And more**

# Clusters only explain ~20% of variance

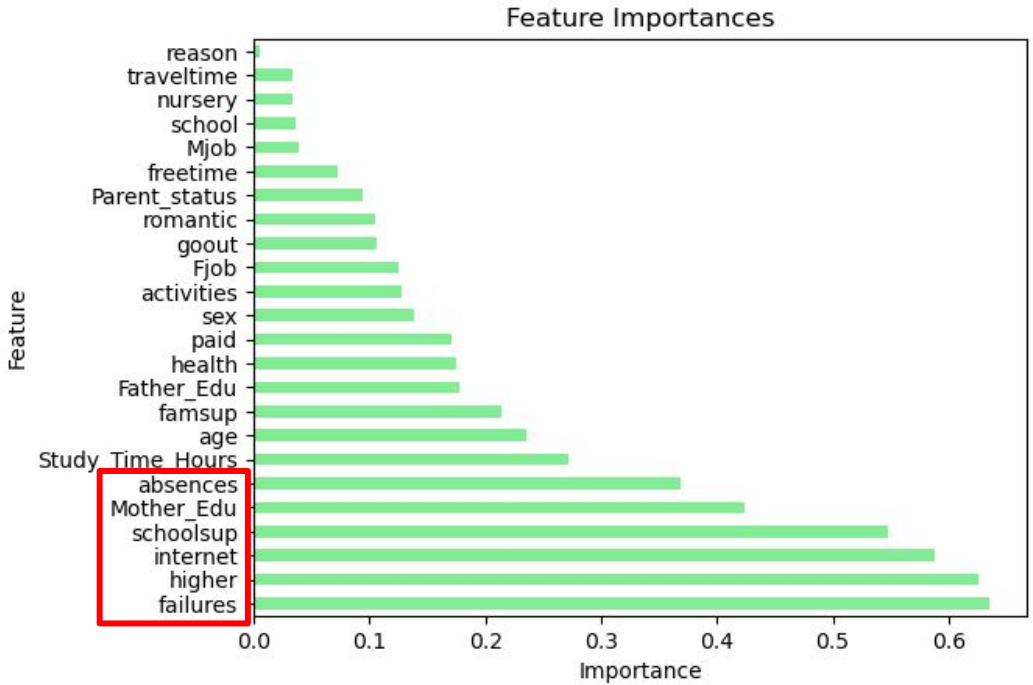



Variance
- PCA 1: 13.10%
- PCA 2: 7.36%

Highest homogeneity with students in cluster 2

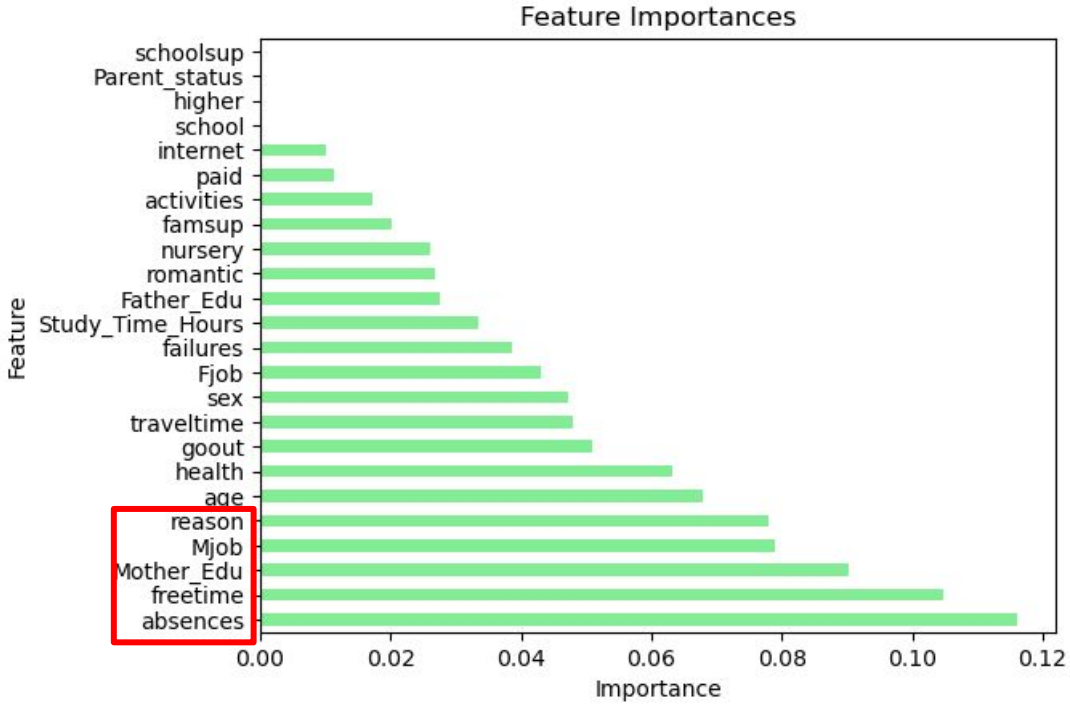
Clusters overlap

# Logistic regression shows past failures, desire for higher ed, internet access may be predictors of success




| | Feature | Coefficient |
|---|---|---|
| 11 | failures | 0.637093 |
| 17 | higher | 0.625623 |
| 18 | internet | 0.586423 |
| 12 | schoolsup | 0.546977 |
| 4 | Mother_Edu | 0.424645 |
| 23 | absences | 0.368176 |
| 10 | Study_Time_Hours | 0.272150 |
| 2 | age | 0.235348 |
| 13 | famsup | 0.214005 |
| 5 | Father_Edu | 0.177268 |
| 22 | health | 0.175581 |
| 14 | paid | 0.170889 |
| 1 | sex | 0.138447 |
| 15 | activities | 0.127860 |
| 7 | Fjob | 0.125328 |
| 21 | goout | 0.106399 |
| 19 | romantic | 0.104248 |
| 3 | Parent_status | 0.093528 |
| 20 | freetime | 0.073369 |
| 6 | Mjob | 0.039472 |
| 0 | school | 0.035761 |
| 16 | nursery | 0.033901 |
| 9 | traveltime | 0.033049 |
| 8 | reason | 0.005652 |

# Decision Trees reveal absences, free time, and mother's education level to be potential predictors of success




| | Feature | Importance |
|---|---|---|
| 23 | absences | 0.116200 |
| 20 | freetime | 0.104882 |
| 4 | Mother_Edu | 0.090366 |
| 6 | Mjob | 0.079067 |
| 8 | reason | 0.077947 |
| 2 | age | 0.067986 |
| 22 | health | 0.063255 |
| 21 | goout | 0.050796 |
| 9 | traveltime | 0.048039 |
| 1 | sex | 0.047128 |
| 7 | Fjob | 0.043158 |
| 11 | failures | 0.038519 |
| 10 | Study_Time_Hours | 0.033431 |
| 5 | Father_Edu | 0.027494 |
| 19 | romantic | 0.026718 |
| 16 | nursery | 0.026166 |
| 13 | famsup | 0.020151 |
| 15 | activities | 0.017273 |
| 14 | paid | 0.011335 |
| 18 | internet | 0.010087 |
| 0 | school | 0.000000 |
| 17 | higher | 0.000000 |
| 3 | Parent_status | 0.000000 |
| 12 | schoolsup | 0.000000 |

# Model correctly classified around 71% of cases, may be some bias toward positive predictions

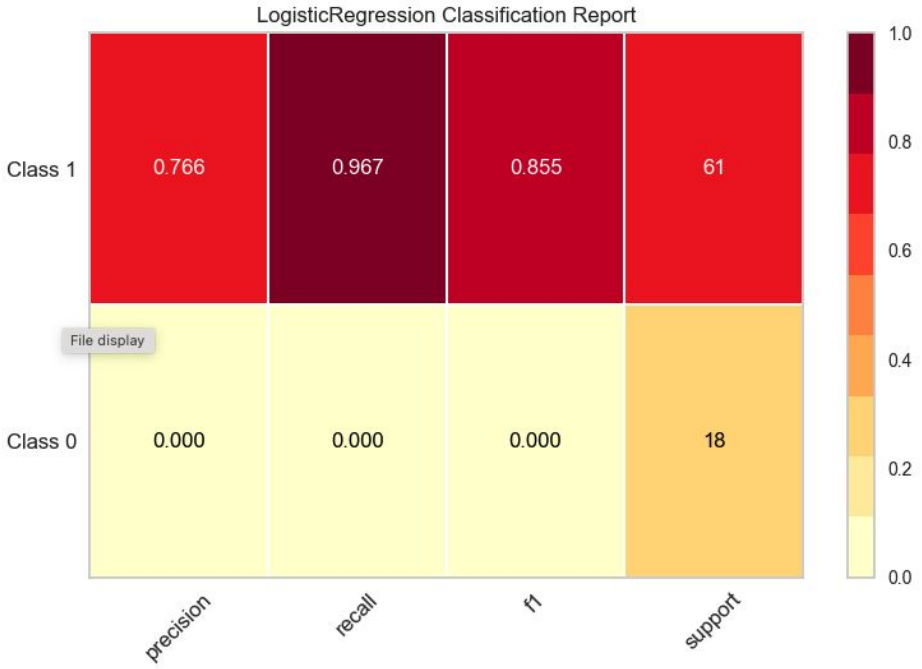
Confusion Matrix

| | Predicted 0 | Predicted 1 |
|---|---|---|
| **Actual 0** | 5 | 16 |
| **Actual 1** | 13 | 65 |

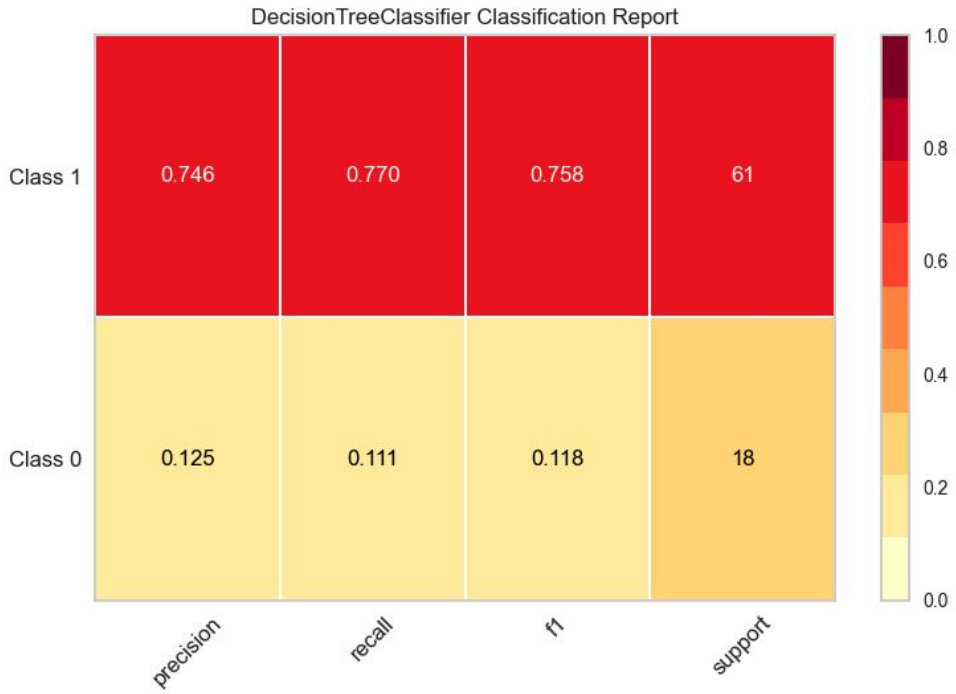
Accuracy Score : 0.7070707070707071
Classification Report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.28 | 0.24 | 0.26 | 21 |
| 1 | 0.80 | 0.83 | 0.82 | 78 |
| accuracy | | | 0.71 | 99 |
| macro avg | 0.54 | 0.54 | 0.54 | 99 |
| weighted avg | 0.69 | 0.71 | 0.70 | 99 |

# Model shows strong bias toward predicting positive cases but struggles significantly with negative cases suggesting potential imbalance issues in the training data



# Decision Tree shows slightly more balanced performance between classes, though still poor for Class 0



# Thank you!

Questions?

# Resources

## Data

- https://archive.ics.uci.edu/dataset/320/student+performance