

**Department of Industrial and Systems Engineering**

**ISYE 670 Final Project Report**

**Prediction of Student Performance in Mathematics**

**Md Abu Naeem Khan Z1985093**

ISYE 670: Intro Data Analytics Engineers

Dr. Christine Nguyen

April 20, 2025

Contents

[**1.0 Introduction** 4](#_Toc196068923)

[**2.0 Explorative Data Analysis** 5](#_Toc196068924)

[**2.1 Cardinality and Missing Data** 6](#_Toc196068925)

[**2.2 Handling Data Types and Conversions:** 6](#_Toc196068926)

[**2.3 Handling Target Variable, G3** 9](#_Toc196068927)

[**2.4 Correlation:** 9](#_Toc196068928)

[**2.4 Other Visualizations** 10](#_Toc196068930)

[**3.0 Data Preprocessing and Feature Selection** 12](#_Toc196068931)

[**3.1 Feature Encoding:** 12](#_Toc196068932)

[**3.2 Feature Selection:** 13](#_Toc196068933)

[**4. Support Vector Machines (SVM)** 13](#_Toc196068934)

[**4.1 Data Preparation:** 13](#_Toc196068935)

[**4.2** **Models Constructed:** 13](#_Toc196068936)

[**4.3 Analysis and Results:** 14](#_Toc196068937)

[**5. Logistic Regression** 15](#_Toc196068938)

[**5.1 Data Preparation:** 15](#_Toc196068939)

[**5.2 Model Constructed:** 15](#_Toc196068940)

[**5.3 Analysis and Results:** 15](#_Toc196068941)

[**6. Random Forest** 16](#_Toc196068942)

[**6.1 Data Preparation:** 16](#_Toc196068943)

[**6.2 Model Constructed:** 16](#_Toc196068944)

[**6.3 Analysis and Results:** 16](#_Toc196068945)

[**7. Model Comparison** 17](#_Toc196068946)

[**7.1 McNemar’s Test Results:** 17](#_Toc196068947)

[**7.2 Proportional Z test** 18](#_Toc196068948)

[**8. Conclusion** 18](#_Toc196068949)

**List of Figures**

[Figure 1 Distribution of G3 9](#_Toc196057709)

[Figure 2 Heat Map – Correlation 10](#_Toc196057710)

[Figure 3 G3 by failures 11](#_Toc196057711)

[Figure 4 G3 by Schoolsup 11](#_Toc196057712)

[Figure 5 Pass Distribution for failures 12](#_Toc196057713)

[Figure 6 SVM Tuning Parameter 14](#_Toc196057714)

[Figure 7 Test Accuracy 17](#_Toc196057715)

**List of Tables**

[Table 1 Raw Data - Fields & Data Type 4](#_Toc196058167)

[Table 2 Data Preparation & Cleaning 6](#_Toc196058168)

[Table 3 Confusion matrix for the best SVM (RBF kernel, C=1) on the test set. 15](#_Toc196058169)

[Table 4 Confusion Matrix – Random Forest 16](#_Toc196058170)

[Table 5 Contingency Table – SVM vs Logistic Regression 17](#_Toc196058171)

[Table 6 Contingency Table – SVM vs Random Forest 18](#_Toc196058172)

# **1.0 Introduction**

Student Performance dataset compiled by the Cortez and Silva (2008), contains detailed records on student grades, as well as demographic, social, and school-related factors. It was created through the combination of official school reports and student-filled questionnaires and includes two subsets based on subject areas: Mathematics (mat) and Portuguese (por). I selected dataset for analysis is mathematics. The dataset contains 395 instances and 33 attributes, representing a diverse set of features including student grades, demographics, family background, social behavior, and academic support indicators. The goal of this project is to predict the student’s performance G3. The target attribute G3 has a strong correlation with attributes G2 and G1. This occurs because G3 is the final year grade (issued at the 3rd period), while G1 and G2 correspond to the 1st and 2nd period grades. In this project, I run three algorithm (SVM, Logistics Rgression, Random Forest).

Table 1 Raw Data - Fields & Data Type

|  |  |  |
| --- | --- | --- |
| S. No. | Field | Data Type |
| 1 | School | object |
| 2 | Sex | object |
| 3 | Age | Int64 |
| 4 | Address | object |
| 5 | Famsize | object |
| 6 | Pstatus | object |
| 7 | Medu | int64 |
| 8 | Fedu | Int64 |
| 9 | Mjob | object |
| 10 | Fjob | object |
| 11 | Reason | object |
| 12 | Guardian | object |
| 13 | Traveltime | int64 |
| 14 | Studytime | Int64 |
| 15 | Failures | int64 |
| 16 | Schoolsup | object |
| 17 | Famsup | object |
| 18 | Paid | object |
| 19 | Activities | object |
| 20 | Nursery | object |
| 21 | Higher | object |
| 22 | Internet | object |
| 23 | Romantic | object |
| 24 | Famrel | int64 |
| 25 | Freetime | Int64 |
| 26 | Gout | int64 |
| 27 | Dalc | int64 |
| 28 | Walc | Int64 |
| 29 | Health | int64 |
| 30 | Absences | int64 |
| 31 | G1 | Int64 |
| 32 | G2 | int64 |
| 33 | G3 | int64 |

# **2.0 Explorative Data Analysis**

Data cleaning and preparation is the most pivotal and primary step in any data analysis. Data cleaning helps to remove inaccurate information, which may lead to incorrect analysis. The Student Performance (Mathematics) dataset initially appeared as a single unstructured column because the delimiter (;) was not parsed correctly. This was resolved by specifying sep=';' while reading the CSV with pandas, which properly split the columns and restored the table format.

## **2.1 Cardinality and Missing Data:**The dataset has no missing rows, meaning every record is complete and ready for modeling. However, 17 columns contain categorical (non-numeric) data that need to be encoded before feeding them into machine learning models.

## **2.2** **Handling Data Types and Conversions:**

Table 2 Data Preparation & Cleaning

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable | Description | Expected Cardinality (Assumed) | Actual Cardinality | Data Type | Missing Values |
| School | student's school | Binary | 2 | object | 0 |
| Sex | student's sex | Binary | 2 | object | 0 |
| Age | student's age | Medium | 8 | int64 | 0 |
| address | student's home address type | Binary | 2 | object | 0 |
| famsize | family size | Binary | 2 | object | 0 |
| Pstatus |  | Binary | 2 | object | 0 |
| Medu | mother's education | Low | 5 | int64 | 0 |
| Fedu | father's education | Low | 5 | int64 | 0 |
| Mjob | mother's job | Low | 5 | object | 0 |
| Fjob | father's job | Low | 5 | object | 0 |
| Reason | reason to choose this school | Low | 4 | object | 0 |
| guardian | student's guardian | Low | 3 | object | 0 |
| traveltime | home to school travel time | Low | 4 | int64 | 0 |
| studytime | weekly study time | Low | 4 | int64 | 0 |
| failures | number of past class failures | Low | 4 | int64 | 0 |
| schoolsup | extra educational support (binary: yes or no) | Binary | 2 | object | 0 |
| Famsup | family educational support | Binary | 2 | object | 0 |
| Paid | extra paid classes within the course subject | Binary | 2 | object | 0 |
| activities | extra-curricular activities | Binary | 2 | object | 0 |
| nursery | attended nursery school | Binary | 2 | object | 0 |
| Higher | wants to take higher education | Binary | 2 | object | 0 |
| internet | internet access at home | Binary | 2 | object | 0 |
| romantic | with a romantic relationship | Binary | 2 | object | 0 |
| Famrel | quality of family relationships | Low | 5 | int64 | 0 |
| freetime | free time after school | Low | 5 | int64 | 0 |
| Gout | going out with friends | Low | 5 | int64 | 0 |
| Dalc | workday alcohol | Low | 5 | int64 | 0 |
| Walc | weekend alcohol consumption | Low | 5 | int64 | 0 |
| Health | current health status | Low | 5 | int64 | 0 |
| absences | number of school absences | Many | 34 | int64 | 0 |
| G1 | nan | Many | 17 | int64 | 0 |
| G2 | nan | Many | 17 | int64 | 0 |
| G3 | nan | Many | 18 | int64 | 0 |

## **2.3 Handling Target Variable, G3:** The variable G3 represents the final student grade. The algorithm transforms the G3 variable into a two-class output where results above 10 are classified as "Pass" while the rest become "Fail". It enables labeling of student outcomes for applying supervised classification algorithms during prediction.

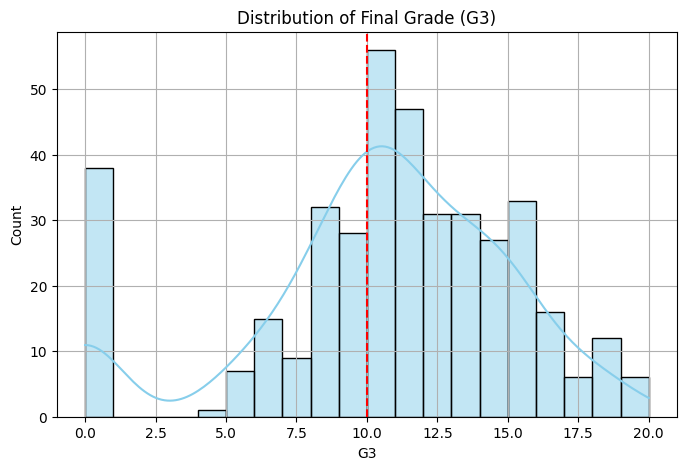


Figure 1 Distribution of G3

Distribution of final grades (G3) for students in the Math class. A red dashed line marks the passing

threshold at 10 points. The histogram shows that student grades range from 0 to 20, with notable peaks around scores 8–10 and 12–15. We observe a sizable number of students scoring just below 10 as well as a large group scoring well above 10. The majority of students scored above the threshold (to the right of the red line), consistent with approximately 67% of students passing the course. The distribution is slightly right-skewed, with a mean final grade of about 10.4.

**2.4 Correlation:** The heatmap illusatrates G1 and G2 academic performance serves as strong

predictors of G3 final results based on a 0.90 correlation value. Students who perform well in initial terms will most likely sustain their academic achievement until the course concludes. academic credentials emerge as the best predictor of academic achievement over lifestyle and behavioral elements thus defining our features selection. students with a history of past failures tend to score lower in the final grade, showing a moderate negative relationship. Interestingly, features like absences, alcohol consumption (Dalc and Walc), and free time show little to no direct correlation with G3, implying that these factors alone don’t strongly affect final grades. However, Dalc and Walc are closely related to each other, meaning students who drink during the week also tend to drink on weekends.

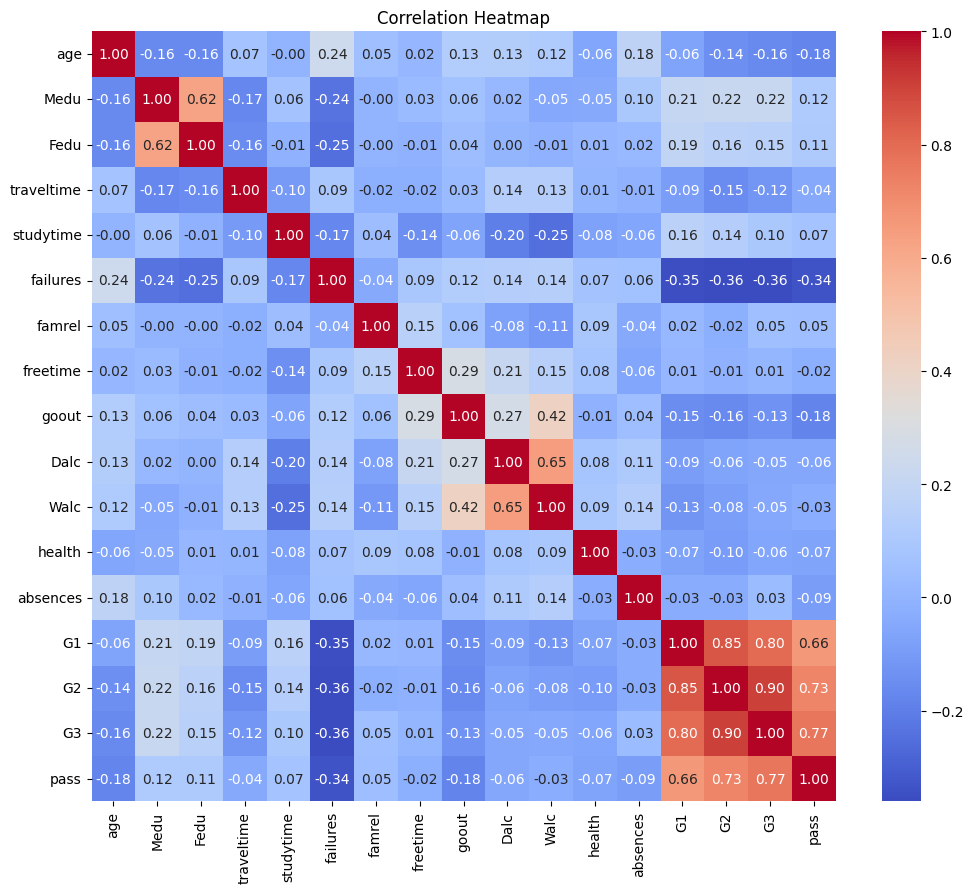


Figure 2 Heat Map – Correlation

## Students who experienced numerous failures in the past received lower final grades according to the blue G3 cell value of -0.36. Most features show weak linear relationships with G3 because their cells in the G3 column appear light in color. The relationship between Dalc and Walc shows heavy red coloring between their cells because students who often drink tend to consume alcohol daily. However, G3 cells display weak coloring indicating alcohol use occurs frequently among students yet leads to minimal impact on their grades.

**2.4 Other Visualizations:** I created boxplots and countplots to further investigate how categorical features relate to students’ final grades (G3) and pass/fail outcomes.

A graph of blue boxes

AI-generated content may be incorrect.

Figure 3 G3 by failures

A graph of blue and black lines

AI-generated content may be incorrect.

Figure 4 G3 by Schoolsup

A graph with blue and orange squares

AI-generated content may be incorrect.

Figure 5 Pass Distribution for failures

These visualizations (more tried in python) complement the heatmap by highlighting group-level trends and non-linear associations that raw correlation coefficients might overlook. For instance, the boxplot of G3 versus past class failures reveals a clear downward trend in performance as the number of failures increases—students with no prior failures had the highest median G3 scores, while those with three or more failures consistently scored lower.

# **3.0 Data Preprocessing and Feature Selection**

I performed several data preprocessing steps. These included encoding categorical variables into a numerical format suitable for machine learning algorithms, scaling features where appropriate, and reducing the feature set to the most relevant attributes via feature selection.

**3.1 Feature Encoding:** To prepare the data for modeling, I first handled feature encoding. The dataset includes a mix of numerical and categorical variables — for example, grades, age, and number of absences are numeric, while attributes like school, family job, and support services are categorical. I used one-hot encoding to convert the categorical variables into binary columns so they could be interpreted by machine learning models. For instance, multi-category features like mother’s job (Mjob) were broken into separate columns such as Mjob\_health or Mjob\_services. To prevent multicollinearity, one category was dropped as a reference during encoding. For binary yes/no features like schoolsup (extra support), I mapped 'yes' to 1 and 'no' to 0. Finally, I standardized all numeric columns using z-score normalization to ensure that features like grades and age were on the same scale — a critical step for algorithms like SVM that are sensitive to feature magnitude.

**3.2 Feature Selection:** I used Recursive Feature Elimination with Cross-Validation (RFECV), but temporarily excluded G1 and G2 to discover what other features contribute independently. I split the remaining features into numeric and categorical sets, applied a preprocessing pipeline, and then ran RFECV using Logistic Regression as the estimator. I assume chose a Logistic Regression model (with L2 regularization) as the estimator for RFECV, given its speed and linear nature which provides a straightforward importance ranking. This model iteratively removed the least important features and evaluated accuracy at each step using 5-fold cross-validation. The process selected four features: failures, Mjob\_health, Mjob\_services, and schoolsup\_yes. These captured elements of academic history, parental background, and institutional support. After identifying these, I scaled G1 and G2 separately and merged them back with the selected features, resulting in a concise and informative final feature set of six features for modeling.

For all models, I used the same preprocessed dataset, numeric features were standardized, and categorical ones were one-hot encoded to make machine-learning. Although Random Forest doesn’t technically require encoding or scaling (since trees can handle raw and categorical values directly), I kept the same format across models to maintain consistency and fairness in comparison. Also I used same amount of feature for every modeling

# **4. Support Vector Machines (SVM)**

**4.1 Data Preparation:** I split the dataset into 80% training and 20% testing, using stratified

sampling to maintain the same pass/fail ratio in both sets.

* 1. **Models Constructed:** I explored several SVM models with different kernel functions and

regularization strengths. In particular, I experimented with both linear and non-linear kernels to assess if non-linear relationships in the data would improve performance. The following SVM variants were trained using the preprocessed training data:

1. Tested with regularization parameter C ∈ {0.15,0.25, 1, 2}. A lower C imposes stronger regularization (preferring a larger margin at the cost of some misclassifications), whereas a higher C tries to fit the training data more exactly.
2. Tested with the same C values {0.15,.25, 1, 2}. The RBF kernel can capture non-linear interactions between features; we used the default kernel width (γ set to “scale” in sklearn).
3. I also tried a polynomial kernel (degree 3) for C ∈ {0.15, .25, 1, 2}. This was to see if a polynomial decision boundary might fit the data better than linear.

A graph with lines and numbers

AI-generated content may be incorrect.

Figure 6 SVM Tuning Parameter

Each model was evaluated via 5-fold cross-validation on the training set. I tracked the mean cross-validation accuracy for each combination of kernel and C to select the best SVM model. The best-performing SVM in cross-validation was the RBF kernel with C = 1, yielding approximately 91% accuracy on the training folds. This suggests that a non-linear decision boundary slightly outperformed a linear one for this problem. I therefore selected the RBF SVM (C=1) as the final model to evaluate on the test set.

## **4.3 Analysis and Results:**

To better understand the SVM’s performance, I present the confusion matrix , and the positive class is “Pass” (final grade ≥ 10), and the negative class is “Fail.”

Table 3 Confusion matrix for the best SVM (RBF kernel, C=1) on the test set.

| **Actual \ Predicted** | **Fail (Pred 0)** | **Pass (Pred 1)** |
| --- | --- | --- |
| **Fail (Actual 0)** | 43 (TN) | 6 (FP) |
| **Pass (Actual 1)** | 2 (FN) | 68 (TP) |

From the confusion matrix, I observe that out of 49 students who actually failed, 43 were correctly identified by the SVM (true negatives) while 6 were misclassified as “pass” (false positives). Similarly, out of 70 students who actually passed, 68 were correctly predicted (true positives) and only 2 were predicted to fail (false negatives). The SVM thus has an excellent recall for the “pass” class (97% of passing students were identified) and a high specificity for the “fail” class (88% of failing students were identified). The few mistakes made by the SVM tended to be labeling a failing student as passing, which is a more acceptable error in this context than the reverse (since we prefer to minimize students incorrectly flagged as failing). Overall, the SVM’s precision for predicting “pass” is 68/(68+6) ≈ 91%, and for “fail” is 43/(43+2) ≈ 95%. These metrics reinforce the model’s strong performance.

# **5. Logistic Regression**

**5.1 Data Preparation:** The data preparation for Logistic Regression was kept consistent with the SVM and Random Forest models to ensure fair comparison. The logistic model assumes no multicollinearity, and since feature selection had already removed highly correlated predictors, this assumption held.

**5.2 Model Constructed:** I trained a Logistic Regression model using sklearn with default L2 regularization and C=1.0. I did not manually tune the regularization here but ensured the baseline setup followed standard practice. Since Logistic Regression is linear and interpretable, it served as a strong reference for evaluating the added value of more complex models like SVM and Random Forest. The model was fitted on the training data and evaluated on the test data using accuracy.

**5.3 Analysis and Results:** On the test data, Logistic Regression achieved the highest accuracy among the three models at 93.67%. This suggests that even a relatively simple linear model is capable of capturing the relationship between selected features and final student performance.

Table 4 Confusion Matrix – Logistic Regression

|  |  |  |
| --- | --- | --- |
| Actual \ Predicted | Fail (0) | Pass (1) |
| Fail (0) | 45 (TN) | 4 (FP) |
| Pass (1) | 3 (FN) | 67 (TP) |

Out of 49 failing students, 45 were correctly identified, and 4 were misclassified as pass. Among 70 passing students, 67 were correctly predicted, with 3 misclassifications. Logistic Regression showed strong overall performance with a low false negative rate, making it a reliable model.

# **6. Random Forest**

**6.1 Data Preparation:** Random Forest does not require feature scaling or encoding to the extent that models like SVM or Logistic Regression do. However, for consistency, the same preprocessed dataset (standardized numerics and one-hot encoded categoricals) was used. Tree-based models like Random Forest can handle unscaled data and collinearity better, but applying the same format helped us fairly compare all models.

**6.2 Model Constructed:** The Random Forest was trained using 100 trees (n\_estimators=100) and a maximum depth of 10. This configuration was selected to balance model complexity with generalization. The model was trained on the same training data and evaluated using test accuracy.

**6.3 Analysis and Results:** The Random Forest classifier achieved a test accuracy of 88%, slightly lower than SVM and Logistic Regression. The confusion matrix is provided below.

Table 4 Confusion Matrix – Random Forest

|  |  |  |
| --- | --- | --- |
| Actual \ Predicted | Fail (0) | Pass (1) |
| Fail (0) | 43 (TN) | 6 (FP) |
| Pass (1) | 5 (FN) | 65 (TP) |

Out of 49 failing students, 43 were correctly predicted, while 6 were misclassified. Among 70 passing students, 65 were correctly classified, with 5 false negatives. While Random Forest had solid recall and precision, it made more errors than Logistic Regression and SVM.

# **7. Model Comparison**

**7.1 McNemar’s Test Results:** To test if the observed differences in model accuracy were statistically significant, I used McNemar’s test.

A graph showing different colored squares

AI-generated content may be incorrect.

Figure 7 Test Accuracy

Table 5 Contingency Table – SVM vs Logistic Regression

|  |  |  |
| --- | --- | --- |
|  | Correct (SVM) | Incorrect (SVM) |
| Correct (LR) | 72 | 0 |
| Incorrect (LR) | 2 | 5 |

McNemar’s Statistic = 0.5

p-value = 0.4795, Not statistically significant

Table 6 Contingency Table – SVM vs Random Forest

|  |  |  |
| --- | --- | --- |
|  | Correct (SVM) | Incorrect (SVM) |
| Correct (RF) | 68 | 4 |
| Incorrect (RF) | 2 | 5 |

McNemar’s Statistic = 0.0

p-value = 0.68 ,Not statistically significant

Despite Logistic Regression achieving the highest raw accuracy, McNemar’s test shows no significant difference compared to SVM or Random Forest at a 5% level.

**7.2 Proportional Z test**: The proportion Z-test is used to compare how well two models perform by checking if the difference in their accuracy rates is statistically meaningful. In this case, we compared SVM with Logistic Regression and Random Forest. The Z-statistics were close to zero and the p-values were above 0.05 in both comparisons (0.548 and 0.598), which tells us there’s no significant difference in performance.

# **8. Conclusion**

Among the three models tested, Logistic Regression performed the best in terms of accuracy, correctly classifying 93% of students. While the SVM came close at 91%, and Random Forest followed at 88%, statistical testing showed that these differences were not significant. This means that all three models performed comparably on this dataset. Logistic Regression’s simplicity, and high accuracy make it a strong candidate. However, the SVM still proved robust, especially in minimizing false negatives. The Random Forest offered good performance and valuable insight into feature importance but did not outperform the others.