Imam Mohammed Ibn Saud University
College of Computer Science and Information Sciences
Computer Science Department
Third Semester 1446 H – 2025 G

Term Project
Design, Implement, and Evaluate SVM, Decision tree and KNN

By

Abdulaziz AlOwain	442019744
Saud Alhafith	444000578
Salman Alshawmar	444008888
Salman Alzamil	443017667

Instructor
Dr. Umran AlQureshi

May 2025

Contents

1 Introduction

In fulfillment of the requirements of the course CS364 at Imam Mohammed Ibn Saud University, this document attempts to predict the final grades of secondary education students from a set of data points[1] using Decision Trees , Support Vector Machines , and k-Nearest Neighbors we extensively discuss the data's format and our design decisions in it in Section 2. This work was done collaboratively by a team of four students, and our project timeline was organized using a Gantt Chart.

2 Dataset

2.1 Dataset Meta-Information

The dataset used is taken as-is provided through the assignment requirements, it is found online, for the purposes of predicting student performance based on prior performance and other information, it was gathered using school reports and questionnaires. It provides two different school subjects' grades; Portuguese, and Mathematics. [1]

The provided data is dense[1], with some intersection between the students participating in the Mathematics dataset and the Portuguese dataset. The Portuguese dataset has 649 and the Mathematics dataset has 395, with 382 students being in both datasets.

However, only the Mathematics dataset has been used; while 97% of the Mathematics participant students also pariticpated in the Portuguese dataset, no data or predictive factors have been taken from the Portuguese dataset.

2.2 Dataset Split

As required by the assignment description, a randomized split in the dataset for training, testing, and validation is required to evaluate the model and in ensuring that the model does not overfit on the provided training dataset. We discuss strategies used to avoid overfitting in Section 5. We opted for a split of 60-20-20, with 60% of the dataset being randomly allocated to training the model, 20% of the dataset is split for the purpose of validating our assumptions in creating the model, such assumptions as the maximum allowed depth of the Decision Tree or the classification criterion, finally, 20% of the dataset was allocated to test our results, testing the model on data it has not seen while training creates an objective measure on the performance of our model.

In splitting the data, to maintain reproducibility, we used a qualified random seed that will generate the same split across runs, however, the qualification of a certain seed to split the data is unnecessary, and was put only for the development process of the program.

2.3 Overrepresented Entries

It is evident that certain features by their nature are better predictors than others, certain features dominated the Decision Trees splits, making them disproportionately influential. Previous grades drastically surfaced more than other features as primary predictors. Such an outcome is not unexpected—previous academic performance is the strongest signal for future grades, nearly rendering other variables irrelevant. [2]

[1] As in "not sparse", providing all data points, and having no missing values.

Previous failures, time spent studying, absences, along with the mothers' and fathers' education also showed strong predictive weight. Higher parental education levels consistently correlate with better student outcomes, a relationship well-documented in education research[3], [4].

Meanwhile, features like school, address, the mothers' or the fathers' jobs, and most lifestyle variables were mostly noise—used rarely, if at all, by the model. As such, the model clung to prior grades and parental background, and underrepresented everything else.

2.4 Data Dictionary

Table 1: A data dictionary for all the available data points, with the output target being highlighted in gray.

Name	Data Type	Description
school	`bool`	Student's school—Gabriel Pereira being `True` and Mousinho da Silveira being `False`.
sex	`bool`	Student's sex—Female being `True` and Male being `False`.
age	`int`	Student's age—ranged from 15 to 22.
address	`bool`	Student's home address type—Urban being `True` and Rural being `False`.
famsize	`bool`	Family size—Less than or equal to three ($3 \geq$) being `True` and Greater than three ($3 <$) being `False`.
Pstatus	`bool`	Parent's cohabitation status—Living together being `True` and Apart being `False`.
Medu	`int`	Mother's education level—From 0 (none) to 4 (higher education).
Fedu	`int`	Father's education level—From 0 (none) to 4 (higher education).
Mjob	`str`	Mother's job—One of: "teacher", "health", "services", "at_home", or "other".
Fjob	`str`	Father's job—Same categories as mother's job.
reason	`str`	Reason for choosing school—One of: "home", "reputation", "course", or "other"
guardian	`str`	Student's guardian—One of: "mother", "father", or "other"
traveltime	`int`	Travel time to school—From 1 (under 15 min) to 4 (over 1 hour).
studytime	`int`	Weekly study time—From 1 (under 2 hours) to 4 (over 10 hours).
failures	`int`	Number of past class failures—From 0 to 3, 4 if more.
schoolsup	`bool`	Extra educational support—`True` if yes, `False` if no.
famsup	`bool`	Family educational support—`True` if yes, `False` if no.
paid	`bool`	Extra paid classes—`True` if yes, `False` if no.
activities	`bool`	Extra-curricular activities—`True` if yes, `False` if no.
nursery	`bool`	Attended nursery school—`True` if yes, `False` if no.
higher	`bool`	Wants higher education—`True` if yes, `False` if no.
internet	`bool`	Internet access at home—`True` if yes, `False` if no.
romantic	`bool`	In a romantic relationship—`True` if yes, `False` if no.
famrel	`int`	Quality of family relationships—From 1 (very bad) to 5 (excellent).
freetime	`int`	Free time after school—From 1 (very low) to 5 (very high).
goout	`int`	Going out with friends—From 1 (very low) to 5 (very high).
Dalc	`int`	Workday alcohol use—From 1 (very low) to 5 (very high).

Name	Data Type	Description
Walc	int	Weekend alcohol use—From 1 (very low) to 5 (very high).
health	int	Current health status—From 1 (very bad) to 5 (very good).
absences	int	Number of school absences—From 0 to 93.
G1	int	First period grade—From 0 to 20.
G2	int	Second period grade—From 0 to 20.
G3	**int**	**Final grade—From 0 to 20. This is the output target.**

3 Results

The models are trained such that they predict whether a student gets a grade >10 (pass), or <=10 (fail). A failing prediction is when the predicted category does not match the actual one. A passing prediction is when the model correctly matches the actual category[2].

3.1 Predictive Ability

3.1.1 Decision Tree

Table 2: The model trained without a maximum depth, using Gini Index as the splitting criterion.

	Precision	Recall	F_1-score	Support
Fail	0.93	0.90	0.92	42
Pass	0.89	0.92	0.91	37
Accuracy			0.91	79
Macro Average	0.91	0.91	0.91	79
Weighted Average	0.91	0.91	0.91	79

Table 3: Results of the model trained with a maximum depth of 3, using Gini Index for splitting.

	Precision	Recall	F_1-score	Support
Fail	0.97	0.93	0.95	42
Pass	0.92	0.97	0.95	37
Accuracy			0.95	79
Macro Average	0.95	0.95	0.95	79
Weighted Average	0.95	0.95	0.95	79

When comparing Table 2 with Table 3, results are better with limited depth due to avoiding overfitting as discussed in Section 5.

3.1.2 k-Nearest Neighbor

k-Nearest Neighbor was chosen for its simplicity and locality-based decision-making. We scaled inputs and tuned $k = 21$, $p = 1$, and `weights = distance` via cross-validation. The final model improved both precision and recall on test data.

Table 4: The tuned k-Nearest Neighbor model, best k from cross-validation, evaluated on the test set.

	Precision	Recall	F_1-score	Support
Fail	0.85	0.81	0.83	42
Pass	0.79	0.84	0.82	37
Accuracy			0.82	79
Macro Average	0.82	0.82	0.82	79
Weighted Average	0.82	0.82	0.82	79

[2]That is, the predicted grade category equals the actual student's grade category.

3.1.3 Support Vector Machine

Support Vector Machine was selected for its strong generalization in binary classification. We scaled features and tuned `C = 0.1`, `kernel = linear`, and `gamma = scale`. It achieved high test accuracy and balanced class performance.

Table 5: Tuned Support Vector Machine with linear kernel (C = 0.1), selected via cross-validation.

	Precision	Recall	F_1-score	Support
Fail	0.95	0.90	0.93	42
Pass	0.90	0.95	0.92	37
Accuracy			0.92	79
Macro Average	0.92	0.93	0.92	79
Weighted Average	0.93	0.92	0.92	79

3.2 Models

3.2.1 Decision Tree

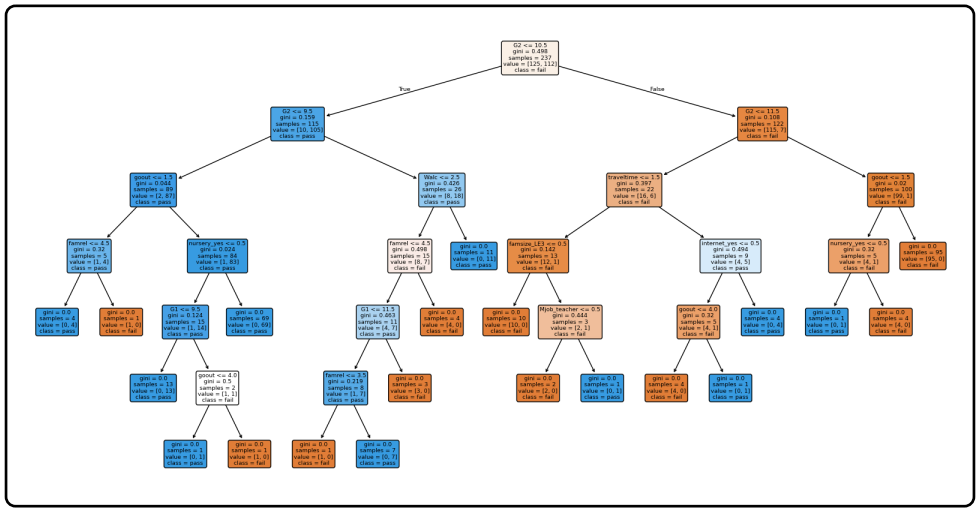

Figure 1: The shown figure shows the generated Decision Tree without *a limit on the maximum depth, it has been similarly classified using the Gini Index .*

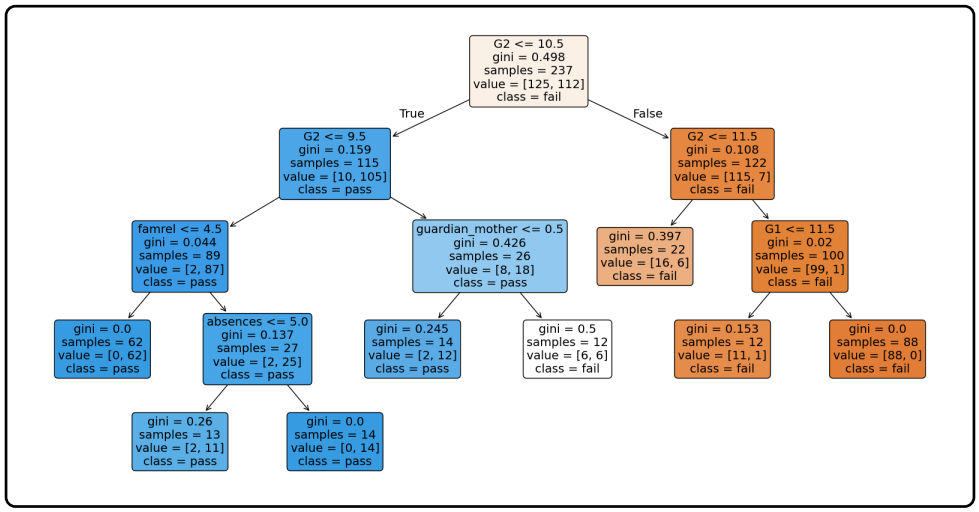

Figure 2: This shows the generated Decision Tree when being generated without a limitation on its maximum depth and using the Gini Index , the Decision Tree has been pruned, as the complete Decision Tree would be illegible.

3.2.2 k-Nearest Neighbor

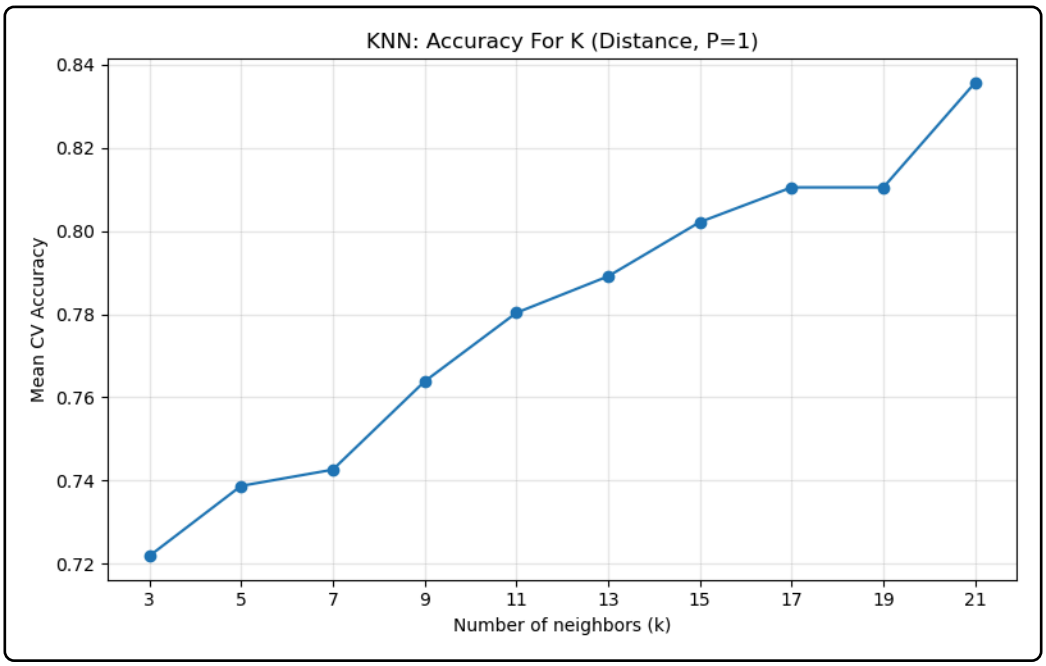

Figure 3: This figure shows the relationship between the number of neighbors (k) and the cross-validated accuracy of the k-Nearest Neighbor model. The best value of (k) was selected based on this curve.

3.2.3 Support Vector Machine

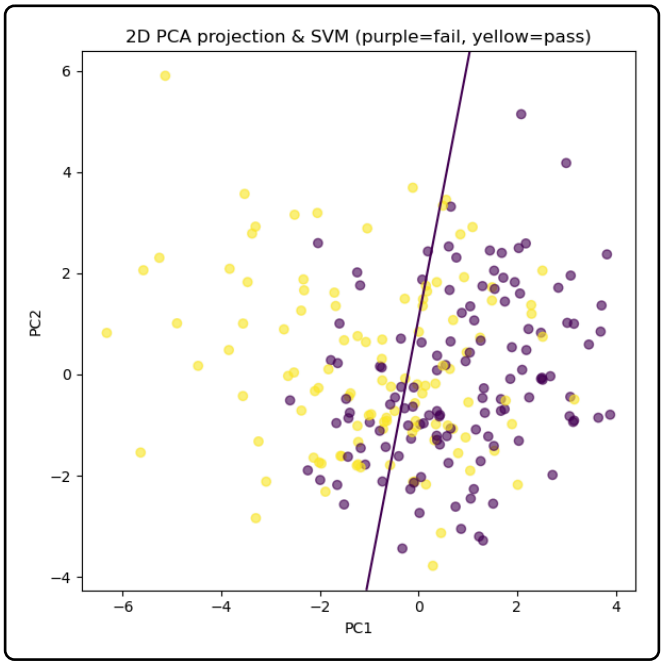

Figure 4: A 2D PCA projection of the dataset with the decision boundary learned by the Support Vector Machine model. The classes are colored (blue = fail, orange = pass), and the separating hyperplane demonstrates how the model distinguishes between them in reduced space.

4 Visualization

4.1 Confusion Matrices

4.1.1 Decision Tree Model

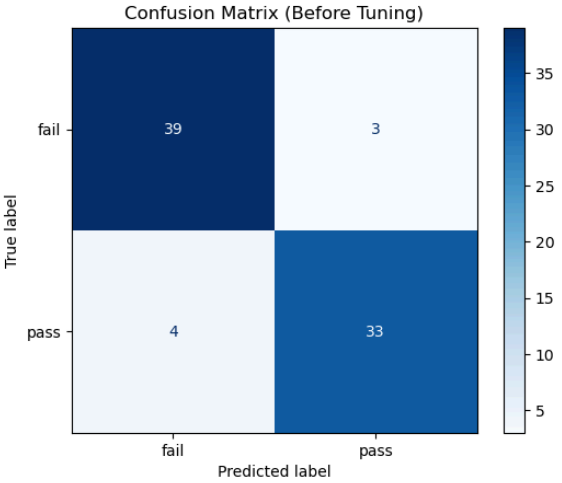

Figure 5: Confusion matrix of the Decision Tree model trained without a maximum depth, using the Gini Index criterion.

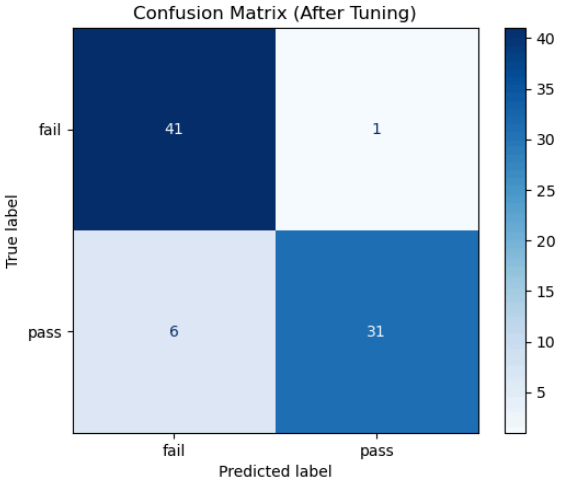

Figure 6: Confusion matrix of the Decision Tree model trained with a maximum depth of 3, also using the Gini Index criterion.

Although the model without a depth limit (Figure 5) can capture more complex patterns, it also overfits, leading to lower generalization on unseen data compared to the pruned version (Figure 6). This supports the concept of overfitting , discussed in Section 5.

4.1.2 k-Nearest Neighbor Model

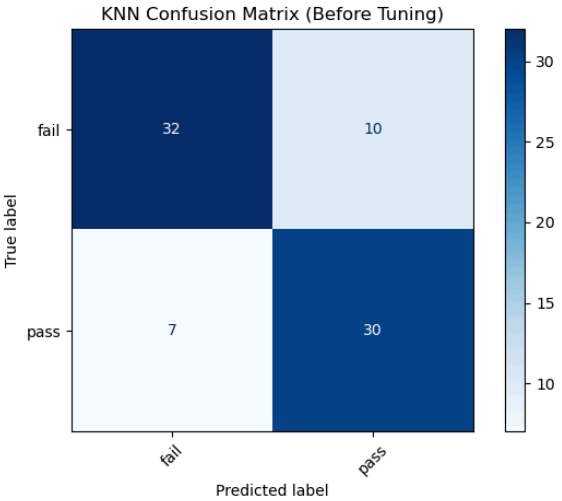

Figure 7: Confusion matrix of the k-Nearest Neighbor model before hyperparameter tuning.

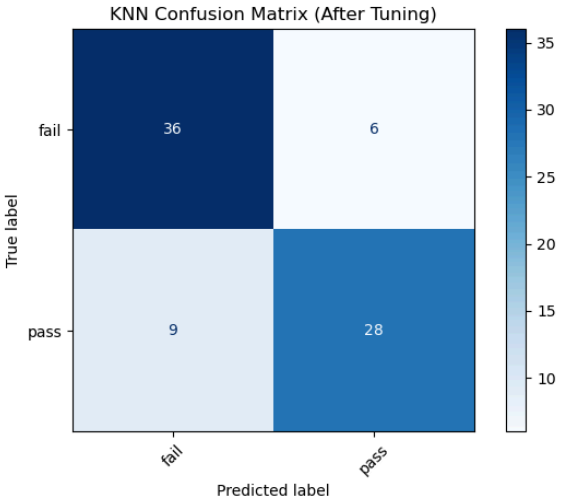

Figure 8: Confusion matrix of the final tuned k-Nearest Neighbor model evaluated on the test set.

Before tuning (Figure 7), the model had a slight imbalance in correctly classifying "fail" vs. "pass" cases. After tuning the value of (k) and using distance-based weighting,

the model's performance improved noticeably in both classes (Figure 8), aligning with the results shown in Table 4.

4.1.3 Support Vector Machine Model

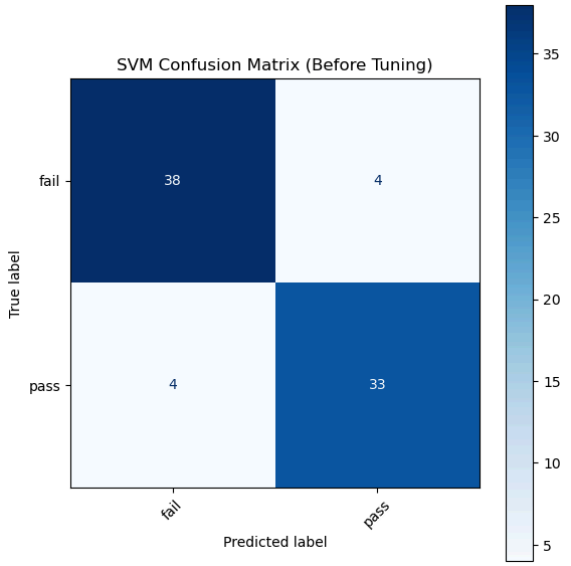

Figure 9: Confusion matrix of the initial Support Vector Machine model before tuning, evaluated on the test set.

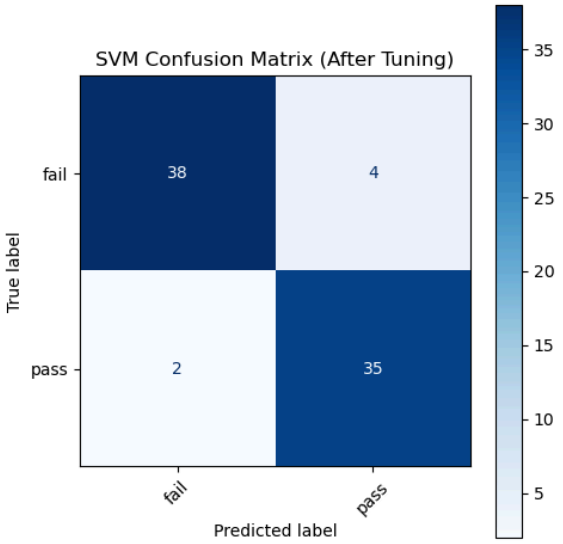

Figure 10: Confusion matrix of the final tuned Support Vector Machine model evaluated on the test set.

The tuned Support Vector Machine model achieved a well-balanced prediction rate across both classes, as seen in Figure 9. Its linear kernel with (C = 0.1) provided strong generalization, confirming its robust performance as shown in Table 5.

5 Avoiding Overfitting

Typically, avoiding overfitting is by reducing the models' ability to represent characteristics of the data that are not intrinsic patterns in the data, such patterns are noise.

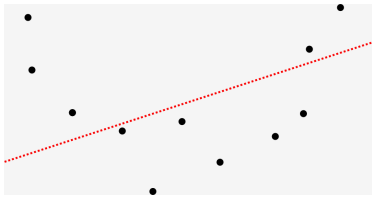

Figure 11: This shows an example of underfitting , where the data does not accurately represent the underlying data characteristics.

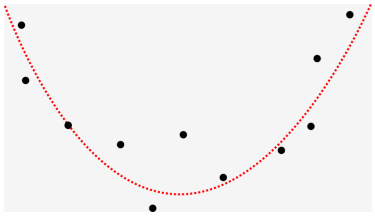

Figure 12: A properly fit model, depicting the underlying relationship while still being predictive of new data.

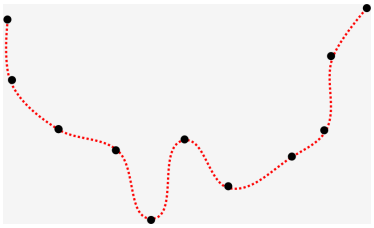

Figure 13: An overfitted model depicts the data too closely, it fails to predict the data's pattern instead predicting the functions' form; capturing the noise.

5.1 Decision Tree – Limiting Depth and Leaf Size

To reduce overfitting , we limited the maximum depth of the Decision Tree model and enforced a minimum number of samples per leaf. Without these constraints, the model generated branches with overly specific rules tied to individual students rather than general patterns. As shown in Table 2 vs. Table 3, these changes improved accuracy and class balance.

5.2 k-Nearest Neighbor – Tuning (k)

We observed overfitting in k-Nearest Neighbor when using small values of (k). By evaluating performance across multiple (k) values using cross-validation, we selected (k = 21), which offered a better generalization to unseen data (see Figure 8).

5.3 Support Vector Machine – Controlling Margin with (C)

Overfitting in Support Vector Machine was managed by tuning the regularization parameter (C). A lower value (C = 0.1) allowed for a softer margin and reduced sensitivity to outliers, which improved test performance compared to higher values.

6 Evaluation & Results Discussion

We evaluated all models using accuracy, precision, recall, and F1-score on a held-out test set. The tuned Support Vector Machine model achieved the highest performance, with an accuracy of 92% and balanced metrics across both classes (Table 5). k-Nearest Neighbor followed with 82% accuracy after tuning (k = 21) and using distance weighting (Table 4). Decision Tree also showed strong results, but only when depth and leaf constraints were applied (Table 3); otherwise, its performance dropped due to overfitting (Table 2).

Overall, tuning had a significant impact on generalization across all models, especially for Support Vector Machine and k-Nearest Neighbor .

7 Conclusion

In this project, we explored three classification algorithms— Decision Tree , k-Nearest Neighbor , and Support Vector Machine —to predict student performance. After careful tuning, the Support Vector Machine model performed best, followed by k-Nearest Neighbor and the constrained Decision Tree . Overfitting was a key challenge, particularly in decision trees and low-(k) neighbors, which we addressed using depth limits and cross-validation. Future work could explore feature selection or ensemble methods to improve robustness further.

8 Glossary

Accuracy: Ratio of correct predictions to total predictions; misleading if classes are imbalanced. 5, 6

Decision Tree: A flowchart-like model that splits data into branches based on feature values, forming a tree structure where each leaf represents a prediction. 2, 3, 5, 6, 8, 10

F_1-score: The harmonic mean of precision and recall. Useful when you need a balance between false positives and false negatives, especially with imbalanced datasets. Punishes extreme differences between precision and recall. 5, 6

Gini Index: A metric used to evaluate splits in classification trees. It measures how often a randomly chosen element would be incorrectly labeled. 5, 6, 8

k-Nearest Neighbor: A simple, instance-based algorithm that classifies data points based on the majority label of their k closest neighbors in the training set. No training phase; all computation is deferred to prediction time. 2, 5, 7, 8, 10

Overfit: When a model learns noise and details from training data too well, harming its ability to generalize to new data. Classic symptoms: very low training error, high validation error. See Section 5. 3, 5, 8, 9, 10

Precision: Ratio of true positives to all predicted positives; measures prediction exactness. 5, 6

Recall: Ratio of true positives to all actual positives; measures prediction completeness. 5, 6

Support: The number of actual occurrences of each class in the dataset. Important for interpreting precision, recall, and F1 — low support means those metrics can be unstable. 5, 6

Support Vector Machine: A supervised learning algorithm that finds the optimal hyperplane to separate classes in the feature space. Effective in high-dimensional spaces and works well with clear margin separation. 2, 6, 7, 9, 10

Underfit: When a model is too simple to capture the underlying pattern of the data. It performs poorly on both training and validation sets, suggesting it needs more complexity or features. See Section 5. 9

References

[1] P. Cortez, “Student Performance.” Accessed: Apr. 23, 2025. [Online]. Available: https://archive.ics.uci.edu/dataset/320/student+performance

[2] E. M. Allensworth and J. Q. Easton, "What Matters for Staying On-Track and Graduating in Chicago Public High Schools: A Close Look at Course Grades, Failures, and Attendance in the Freshman Year. Research Report.," *Consortium on Chicago School Research*, 2007.

[3] E. F. Dubow, P. Boxer, and L. R. Huesmann, "Long-term effects of parents' education on children's educational and occupational success: Mediation by family interactions, child aggression, and teenage aspirations," *Merrill-Palmer Quarterly*, vol. 55, no. 3, pp. 224–249, 2009.

[4] P. E. Davis-Kean, "The influence of parent education and family income on child achievement: the indirect role of parental expectations and the home environment.," *Journal of family psychology*, vol. 19, no. 2, p. 294, 2005.