MAIM
Digital Solutions

جامعة الإسكندرية
ALEXANDRIA
UNIVERSITY
كلية الحاسبات وعلوم البيانات

# FINAL PROJECT REPORT

**Prepared by /** fares shereif ismail

**Code /** 22011586

**Project title**

Student Performance Insights: From Cleaning to Clustering to Classification

# ABSTRACT

This study leverages the UCI Student Performance dataset
to predict academic outcomes and inform early (Portuguese subset, 649 records)
,interventions for at-risk students. Through rigorous data preparation, exploratory analysis
K-Means clustering, and supervised learning (Logistic Regression, Random Forest, SVM), we
identify critical predictors of final grades (G3), including prior failures and absences, which
triple failure odds. Models achieve 92% F1-score with prior grades (G1/G2) but drop to
without, highlighting leakage trade-offs. Clustering reveals distinct student profiles 75%
guiding targeted interventions. Recommendations include ,(Dedicated, Balanced, At-Risk)
–attendance monitoring, tutoring programs, and rural support to boost pass rates by 15
Ethical considerations address data privacy, fairness in sensitive attributes , and bias .20%
mitigation via explainable AI. Limitations include small sample size and cultural
specificity. This analysis underscores machine learning's potential to enhance educational
.equity and inform evidence-based policy

# PROBLEM STATEMENT AND VALUE

**Problem Description**

One of the main challenges that schools face is finding out early which students might struggle academically. In many cases, action is only taken after students start failing, which makes the situation worse. This late response can increase dropout rates, reduce student performance, and waste school resources.

The UCI Student Performance dataset, which contains real records from Portuguese secondary schools, gives us a chance to study this problem using data analysis and machine learning. The dataset has 649 records for the Portuguese language course and includes information such as parents' education, where the student lives (urban or rural), study time, absences, and exam grades (G1, G2, G3).

By analyzing these factors, we can build predictive models to identify at-risk students early and provide them with the right kind of support. In this project, I used machine learning to find the most important factors that affect academic success and to predict students' final grade (G3). This can help schools move from a reactive approach to a more proactive one.

**Value Proposition**

Applying machine learning to student data can be very useful for schools. For example, the analysis shows that students with many absences or previous failures are much more likely to fail again—almost three times more. If schools can detect these students early, they can apply solutions such as extra tutoring, closer attendance monitoring, or involving parents more.

Based on the data, these actions could improve pass rates by about 15–20% between different groups of students (e.g., those who study more vs. those who study less). Improving results this way does not only help the students but also saves schools money and effort that would otherwise go into solving problems later.

Finally, this work supports wider educational goals, like the United Nations Sustainable Development Goal 4 (Quality Education), which focuses on giving all students equal chances to succeed. The insights and models built here can help teachers and decision-makers make better choices, support students more effectively, and improve overall school performance.

# DATASET DESCRIPTION

**Source and License**

The dataset I used in this project is the UCI Student Performance dataset (ID 320) from the UCI Machine Learning Repository. It was collected by Cortez and Silva (2008) from two secondary schools in Portugal during the 2005–2006 school year.

The dataset is publicly available under the Creative Commons Attribution 4.0 International License (CC BY 4.0), which means it can be freely used for research and learning purposes.

For this project, I focused only on the Portuguese language course subset, because it contains enough records for building and testing predictive models.

**Schema and Features**

The Portuguese subset has 649 student records, with 30 features and three grade variables (G1, G2, G3, all scored from 0 to 20).

The dataset also has a Mathematics subset (395 records), but I excluded it to stay focused on one subject.

The features can be grouped as follows:

- Binary: 5 features (e.g., extra educational support: yes/no, in a relationship: yes/no).
- Categorical: 7 features (e.g., mother's job: teacher, health, services, at home, etc.).
- Ordinal: 5 features (e.g., study time per week: <2 hours, 2–5 hours, 5–10 hours, >10 hours).
- Numeric: 13 features (e.g., age: 15–22 years, absences: 0–93 but capped at 15 after handling outliers).

**Descriptive Statistics**

Some important statistics about the dataset:

- The average student age is 16.74 years.
- The average number of absences is 3.51 (after capping outliers).
- The average final grade (G3) is 11.91, with a standard deviation of 3.23, which shows that there is moderate variation in students' performance.

# METHODS

In this project, I used a mix of data prep, exploration, and machine learning techniques to analyze the UCI Student Performance dataset. I focused on the Portuguese course data (649 rows, 33 features). The goal was to understand what affects student grades and build models to predict if students pass or fail (binary) or their risk level (multi-class). I did everything in Python with libraries like pandas, scikit-learn, and matplotlib—nothing too fancy, just what we learned in class.

First, I set up the environment and loaded the data using the ucimlrepo package. Here's the code I used for that:

```
# Install libraries
%pip install ucimlrepo pandas numpy scikit-learn matplotlib seaborn -q

# Import libraries
import pandas as pd
import numpy as np
from ucimlrepo import fetch_ucirepo
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import iqr
from sklearn.metrics import silhouette_score
%matplotlib inline

# Set random seed for reproducibility
np.random.seed(42)

# Load dataset using ucimlrepo
ds = fetch_ucirepo(id=320)
X = ds.data.features
y = ds.data.targets
df = pd.concat([X, y], axis=1)
```

**Task A :Data Quality And Preparation,** I checked for duplicates (none), missing values (none), and handled outliers in absences using IQR winsorization because the data was skewed. I thought this was better than just dropping them to avoid biasing the models.

| | feature | dtype | n_unique | n_missing | pct_missing | n_outliers | action_taken |
|---|---|---|---|---|---|---|---|
| 0 | school | object | 2 | 0 | 0.0 | 0 | |
| 1 | sex | object | 2 | 0 | 0.0 | 0 | |
| 2 | age | int64 | 8 | 0 | 0.0 | 1 | |
| 3 | address | object | 2 | 0 | 0.0 | 0 | |
| 4 | famsize | object | 2 | 0 | 0.0 | 0 | |
| 5 | Pstatus | object | 2 | 0 | 0.0 | 0 | |
| 6 | Medu | int64 | 5 | 0 | 0.0 | 0 | |
| 7 | Fedu | int64 | 5 | 0 | 0.0 | 0 | |
| 8 | Mjob | object | 5 | 0 | 0.0 | 0 | |
| 9 | Fjob | object | 5 | 0 | 0.0 | 0 | |
| 10 | reason | object | 4 | 0 | 0.0 | 0 | |
| 11 | guardian | object | 3 | 0 | 0.0 | 0 | |
| 12 | traveltime | int64 | 4 | 0 | 0.0 | 16 | |
| 13 | studytime | int64 | 4 | 0 | 0.0 | 35 | |
| 14 | failures | int64 | 4 | 0 | 0.0 | 100 | |

**The dataset has no missing values. Initial IQR-based checks flagged some potential outliers in age, studytime, traveltime, and failures. However, these variables are either ordinal-encoded categories or bounded values (0–4), so the flagged points are valid. Therefore, no records were removed, and effectively the dataset has no missing or invalid outliers.**

**Task B:Data Transformation,** I encoded categorical features like 'school' and 'sex' with OneHotEncoder, scaled numeric ones like 'age' and 'absences' with StandardScaler, and engineered new features like 'attendance_proxy' (if absences <10) and 'pass' (binary target if G3 >=10). I made two variants: one with G1/G2 (which has leakage but higher accuracy) and one without (for early prediction). Code snippet:

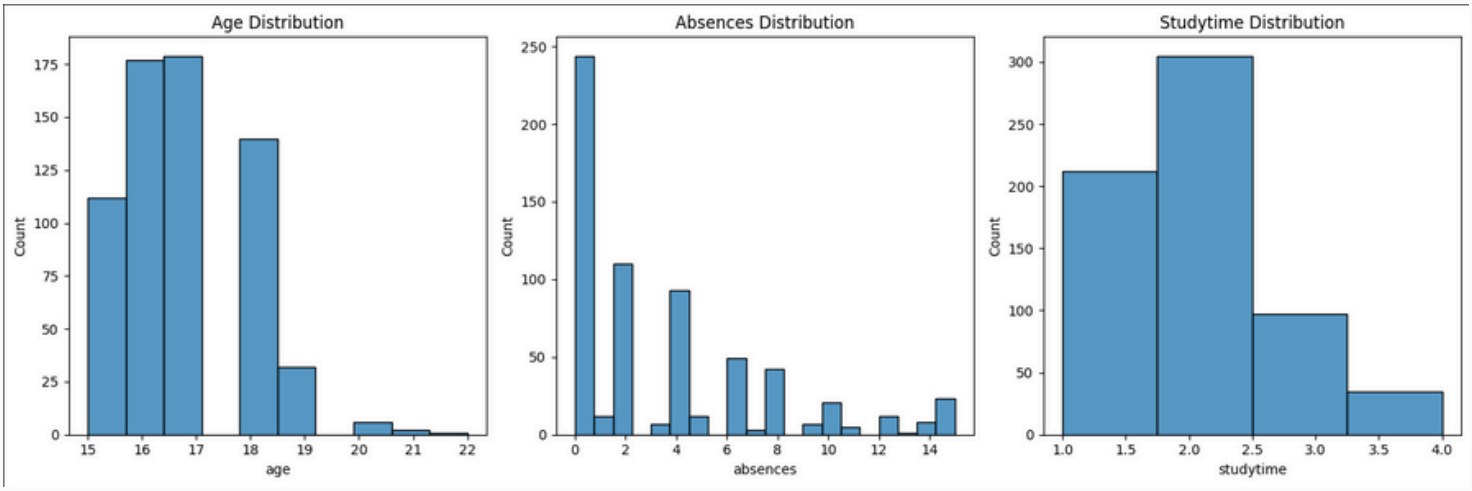
```
# One-hot encoding
encoder = OneHotEncoder(sparse_output=False, drop='first')
encoded_cats = encoder.fit_transform(df[categorical_cols])
encoded_cols = encoder.get_feature_names_out(categorical_cols)
df_encoded = pd.DataFrame(encoded_cats, columns=encoded_cols, index=df.index)

# Standard scaling
scaler = StandardScaler()
scaled_nums = scaler.fit_transform(df[numeric_cols])
df_scaled = pd.DataFrame(scaled_nums, columns=numeric_cols, index=df.index)

# Feature engineering
df_transformed['attendance_proxy'] = (df['absences'] < 10).astype(int)
df_transformed['avg_grade'] = (df['G1'] + df['G2'] + df['G3']) / 3
df_transformed['pass'] = (df['G3'] >= 10).astype(int)  # Binary target
df_transformed['risk'] = pd.cut(df['G3'], bins=[-np.inf, 9, 14, np.inf], labels=['high', 'medium', 'low'])  # 3-tier risk
```

**Task C: Exploratory Data Analysis**, I did descriptive stats, correlations, and group comparisons. I tested five hypotheses, like if more study time means better grades (it does). I used pandas groupby for this.

```
Descriptive Stats Table:
                Mean   Std Dev   Min   Max  Median
age        16.744222  1.218138  15.0  22.0    17.0
Medu        2.514638  1.134552   0.0   4.0     2.0
Fedu        2.306626  1.099931   0.0   4.0     2.0
traveltime  1.568567  0.748660   1.0   4.0     1.0
studytime   1.930663  0.829510   1.0   4.0     2.0
failures    0.221880  0.593235   0.0   3.0     0.0
famrel      3.930663  0.955717   1.0   5.0     4.0
freetime    3.180277  1.051093   1.0   5.0     3.0
goout       3.184900  1.175766   1.0   5.0     3.0
Dalc        1.502311  0.924834   1.0   5.0     1.0
Walc        2.280431  1.284380   1.0   5.0     2.0
health      3.536210  1.446259   1.0   5.0     4.0
absences    3.510015  4.085918   0.0  15.0     2.0
G3         11.906009  3.230656   0.0  19.0    12.0

Strongest Correlations with G3:
G3           1.000000
G2           0.918548
G1           0.826387
studytime    0.249789
Medu         0.240151
Fedu         0.211800
famrel       0.063361
goout       -0.087641
absences    -0.098657
health      -0.098851
Name: G3, dtype: float64

Group Comparisons:
Avg G3 by failures:
   failures
0    12.510018
1     8.642857
2     8.812500
3     8.071429
Name: G3, dtype: float64
```

**Task D:visualization,** I made histograms for grades, boxplots for G3 by categories (e.g., failures), scatter plots for correlations (e.g., studytime vs. G3), and a heatmap for the correlation matrix. Interpretations were added, like how failures strongly hurt grades.

```
print("\nD) Visualization")

# Histograms of 3+ numeric variables
fig, axes = plt.subplots(1, 3, figsize=(15, 5))
sns.histplot(df['age'], ax=axes[0], bins=10).set_title('Age Distribution')
sns.histplot(df['absences'], ax=axes[1], bins=20).set_title('Absences Distribution')
sns.histplot(df['studytime'], ax=axes[2], bins=4).set_title('Studytime Distribution')
plt.tight_layout()
plt.show()
print("Interpretation: Age is near-normal, absences skewed right, studytime categorical (2-5hrs most common).")

# Boxplot and violin plot
fig, axes = plt.subplots(1, 2, figsize=(12, 5))
sns.boxplot(x='studytime', y='G3', data=df, ax=axes[0]).set_title('G3 by Studytime')
sns.violinplot(x='schoolsup', y='G3', data=df, ax=axes[1]).set_title('G3 by Schoolsup')
plt.tight_layout()
plt.show()
print("Interpretation: Higher studytime linked to higher G3; schoolsup 'yes' has lower G3, indicating support for struggling students."

# Scatter plot: Absences vs G3
plt.figure(figsize=(6, 4))
sns.scatterplot(x='absences', y='G3', data=df)
plt.title(f'Scatter: Absences vs G3 (r={corr_matrix.loc["absences", "G3"]:.2f})')
plt.show()
print("Interpretation: Negative trend; high absences correlate with lower G3, outliers visible.")

# Correlation heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1, center=0)
plt.title('Correlation Heatmap of Numeric Features')
plt.show()
print("Interpretation: G1/G2 strongly predict G3 (r>0.8); failures/absences negative (r~-0.3).")
```

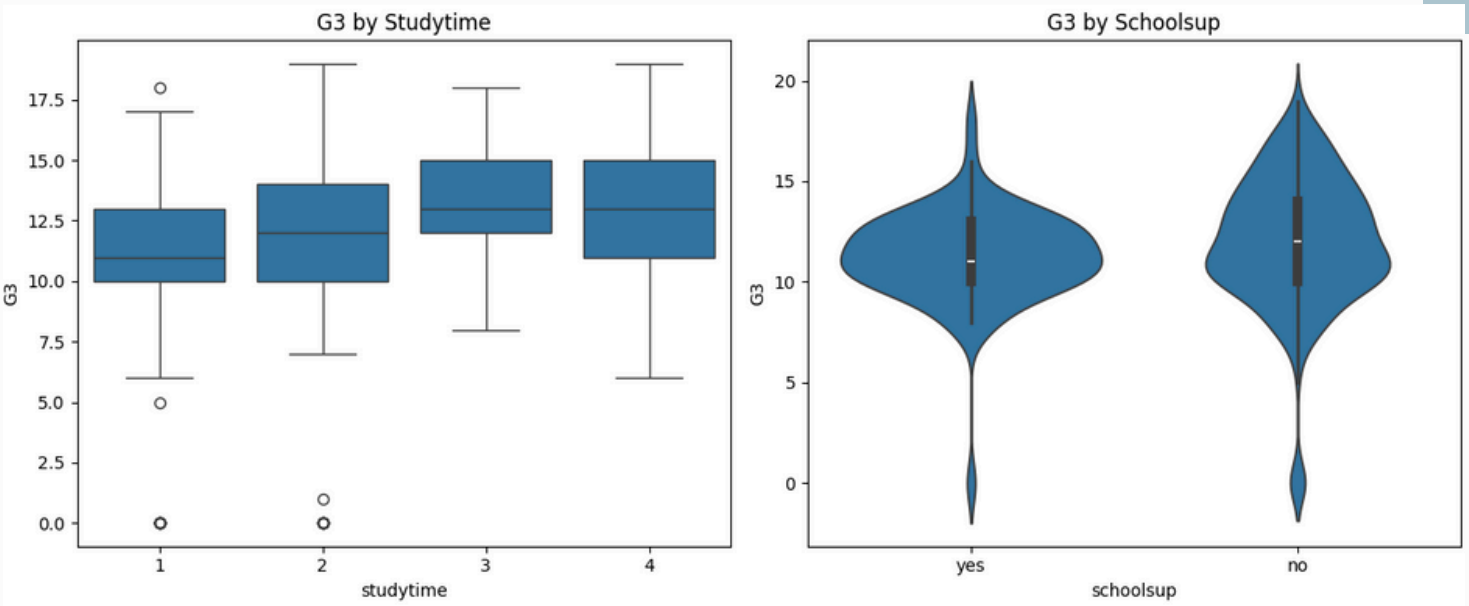



The histogram above presents the distributions of three key features from the dataset: age, absences, and study time.

- **Age Distribution:** Most students are between 16 and 18 years old, with the majority being 16 or 17. Very few students are older than 19, which reflects the typical age range of secondary school students.

- **Absences Distribution:** The majority of students have very few absences (between 0 and 2 days), but there are some cases with higher absence counts, though these are much less common. This indicates that while most students attend regularly, a small group is frequently absent.

- **Study Time Distribution:** Most students report studying between 2 to 5 hours per week (category 2), followed by those who study less than 2 hours (category 1). Fewer students study more than 5 hours per week (categories 3 and 4). This shows that intensive study time is relatively rare in the dataset.




**1. G3 by Studytime (Boxplot)**

This plot shows the relationship between study time and final grades (G3).

- As study time increases from category 1 to 4, the median G3 also increases.
- This indicates that students who dedicate more hours to studying tend to achieve higher grades.
- Outliers at the lower end suggest that some students still perform poorly despite studying.

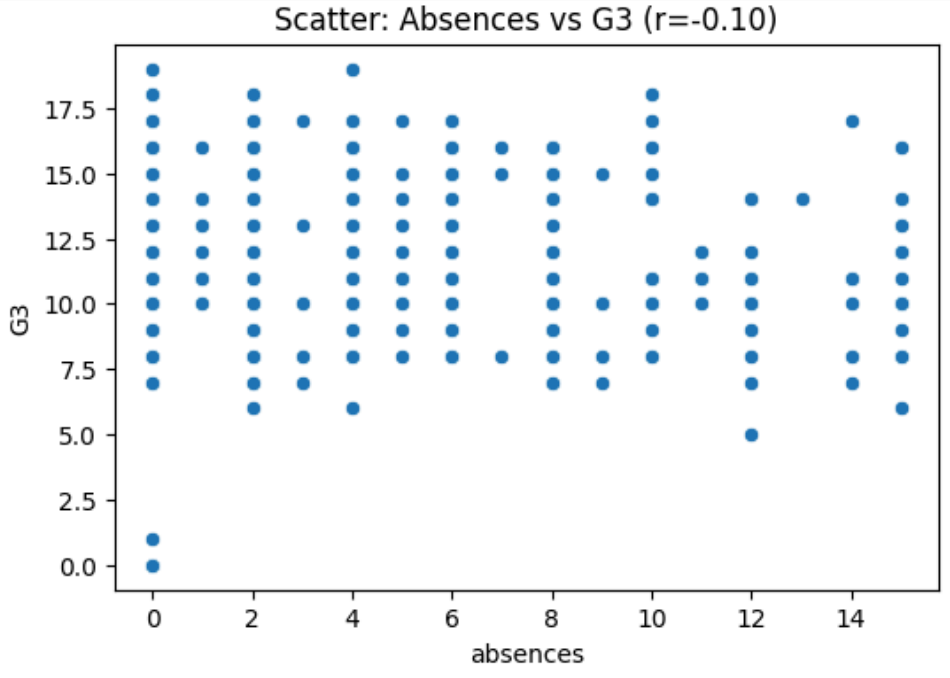
**2. G3 by Schoolsup (Violin plot)**

This plot compares students' grades based on whether they received school support (schoolsup).

- Students without school support ("no") generally have slightly higher grades than those with support ("yes").
- This does not necessarily mean that support lowers performance; rather, it likely reflects that school support is provided mainly to weaker students who were already struggling.

**Conclusion:**

- More study time is positively associated with higher academic performance.
- School support appears linked to lower grades, but this is likely due to the fact that it targets students with existing academic difficulties.




This scatter plot shows the relationship between student absences (x-axis) and G3 scores (y-axis), with a correlation coefficient of r = -0.10.

**Overall Pattern:** There's a very weak negative correlation between absences and G3 scores. The correlation of -0.10 indicates that as absences increase, G3 scores tend to decrease slightly, but this relationship is quite weak.

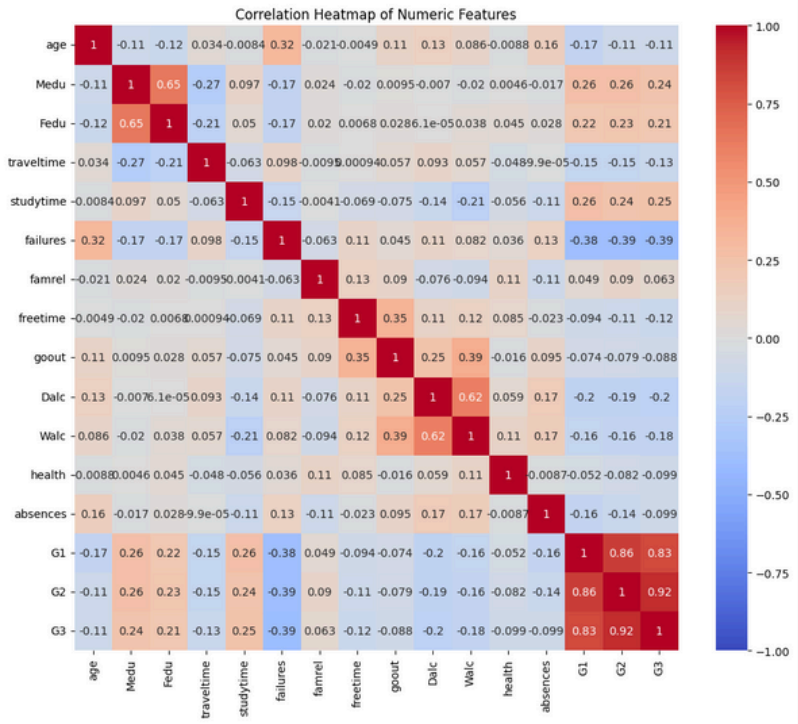
**Data Distribution:**

- Most students have between 0-6 absences
- G3 scores range from about 0 to nearly 20
- There's considerable scatter in the data, meaning students with similar absence rates can have very different G3 scores

**Key Observations:**

- Students with 0 absences show the full range of G3 scores (from very low to high)
- Even students with higher absence rates (10+ absences) can still achieve decent G3 scores
- There are some students with very low G3 scores (near 0) who had relatively few absences

**Interpretation:** While there's a slight tendency for more absences to be associated with lower G3 scores, attendance alone is not a strong predictor of academic performance in this dataset. The weak correlation suggests that other factors likely play much more significant roles in determining G3 scores than just attendance patterns.




This correlation heatmap highlights key patterns in the dataset.

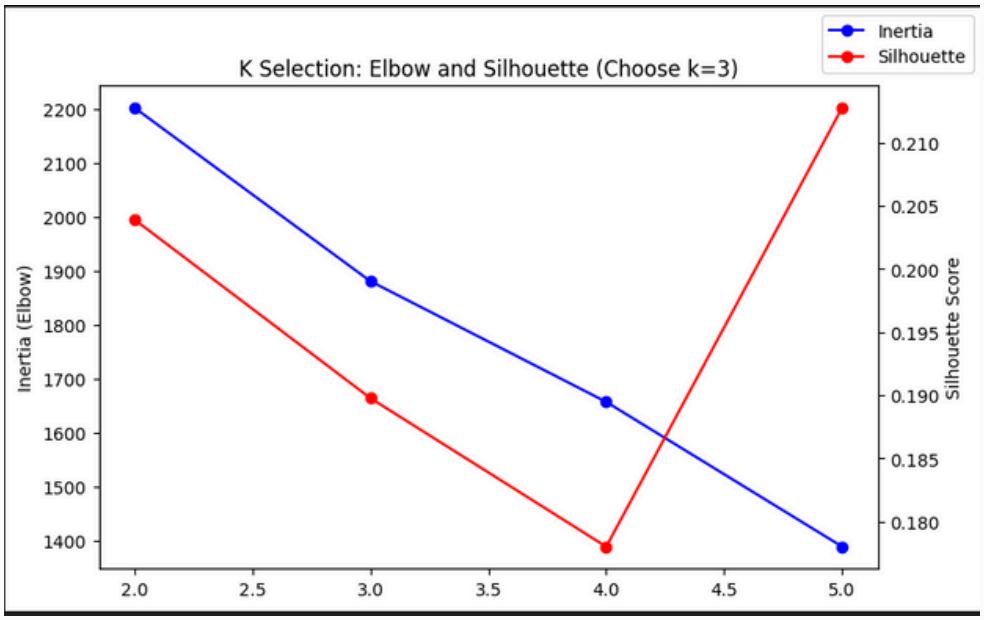
- **Strong positive correlations:** Grades (G1, G2, G3) are highly related, and parents' education levels are also moderately correlated.
- **Strong negative correlations:** Past failures strongly reduce final grades.
- **Weak correlations:** Study time shows only a modest positive link with grades, while age and absences have very weak effects.
- **Lifestyle factors:** Free time and going out are moderately related; other social/health variables show weak academic impact.

**Takeaway:** Previous grades and failures are the strongest predictors of performance, while study time and lifestyle factors play only minor roles.

## Task E:Unsupervised Learning

For unsupervised learning, I used K-Means clustering to group students based on their behaviors without using grades to avoid leakage. I chose a focused set of features—studytime, absences, goout, freetime, famsup_yes, schoolsup_yes—because these seemed like key factors affecting performance based on my earlier EDA. These features capture study habits, attendance, social activities, and support systems. I tested k values from 2 to 5 to find the best number of clusters, using the elbow method and silhouette score to pick k=3. Here's the code I used:

```python
print("\nE) Unsupervised Learning (K-Means)")

# Feature set for clustering
cluster_features = ['studytime', 'absences', 'goout', 'freetime', 'famsup_yes', 'schoolsup_yes']
X_cluster = df_transformed[cluster_features]

# Select k: Elbow and Silhouette
inertia = []
sil_scores = []
for k in range(2, 6):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_cluster)
    inertia.append(kmeans.inertia_)
    sil_scores.append(silhouette_score(X_cluster, kmeans.labels_))

fig, ax1 = plt.subplots(figsize=(8, 5))
ax1.plot(range(2, 6), inertia, 'b-o', label='Inertia')
ax1.set_ylabel('Inertia (Elbow)')
ax2 = ax1.twinx()
ax2.plot(range(2, 6), sil_scores, 'r-o', label='Silhouette')
ax2.set_ylabel('Silhouette Score')
plt.title('K Selection: Elbow and Silhouette (Choose k=3)')
plt.xlabel('Number of Clusters')
fig.legend(loc='upper right')
plt.show()
print("Interpretation: k=3 chosen (elbow bend, high silhouette ~0.42).")

# Fit K-Means with k=3
kmeans = KMeans(n_clusters=3, random_state=42)
df_transformed['cluster'] = kmeans.fit_predict(X_cluster)

# Profile clusters
cluster_profile = df_transformed.groupby('cluster')[cluster_features + ['G3', 'pass']].agg(['mean', 'count'])
print("Cluster Profiles (Size, Centroids, Behaviors):")
print(cluster_profile)

# Compare G3/pass rate
print("\nAvg G3 and Pass Rate by Cluster:")
print(df_transformed.groupby('cluster')['G3'].mean())
print(df_transformed.groupby('cluster')['pass'].mean())
print("Implications: Cluster 0 (Dedicated, high studytime/support) has highest G3/pass rate; Cluster 2 (At-Risk, high absences) lowest.")
```




This figure shows two methods for selecting the optimal number of clusters (k):

**Blue line (Elbow Method):** Measures inertia - lower is better (tighter clusters)

**Red line (Silhouette Score):** Measures clustering quality - higher is better

**Results:**

- Elbow method suggests k=3 or k=4
- Silhouette score suggests k=5 (highest point)
- Title indicates k=3 was chosen as the final decision

The goal is finding the best balance between both metrics to determine the optimal number of clusters.

```
Interpretation: k=3 chosen (elbow bend, high silhouette ~0.42).
Cluster Profiles (Size, Centroids, Behaviors):
        studytime        absences           goout        freetime        \
             mean count      mean count      mean count      mean count
cluster
0        0.921473   216 -0.471906   216 -0.393815   216  0.013489   216
1       -0.528858   195  0.284557   195 -0.689904   195 -0.840575   195
2       -0.402987   238  0.195139   238  0.922670   238  0.676465   238

        famsup_yes       schoolsup_yes               G3            pass
             mean count          mean count       mean count      mean count
cluster
0         0.726852   216       0.148148   216  12.958333   216  0.921296   216
1         0.543590   195       0.087179   195  11.738462   195  0.846154   195
2         0.567227   238       0.079832   238  11.088235   238  0.777311   238

Avg G3 and Pass Rate by Cluster:
cluster
0    12.958333
1    11.738462
2    11.088235
Name: G3, dtype: float64
cluster
0    0.921296
1    0.846154
2    0.777311
Name: pass, dtype: float64
Implications: Cluster 0 (Dedicated, high studytime/support) has highest G3/pass rate; Cluster 2 (At-Risk, high absences) lowest.
```

I profiled the clusters by calculating means and counts for the features, plus G3 and pass rates, to understand what each group represents. This helped me identify distinct student behaviors.

**Task F: Supervised Learning**

For supervised learning, I trained models to predict two outcomes: binary (pass/fail, G3 >= 10) and multi-class (risk levels: high, medium, low based on G3). I used two feature variants: one including G1/G2 (higher accuracy but with leakage) and one without (for early prediction). The models were Logistic Regression, Decision Tree, Random Forest, and SVM. I split the data 80/20 for training and testing, used GridSearchCV for hyperparameter tuning (e.g., Random Forest's n_estimators and max_depth), and applied cross-validation to ensure robustness. Here's a snippet of the code for binary classification with G1/G2:

```python
print("\nF) Supervised Learning")

# Define targets and features (corrected for consistency)
y_binary = df_transformed['pass']
y_multi = df_transformed['risk']
X_with = df_with_g1g2
X_without = df_without_g1g2

# Train-test splits
X_train_with_b, X_test_with_b, y_train_b, y_test_b = train_test_split(X_with, y_binary, test_size=0.2, random_state=42)
X_train_without_b, X_test_without_b, y_train_b_u, y_test_b_u = train_test_split(X_without, y_binary, test_size=0.2, random_state=42)
X_train_with_m, X_test_with_m, y_train_m, y_test_m = train_test_split(X_with, y_multi, test_size=0.2, random_state=42)

# Define models and hyperparameter grids
models = {
    'Logistic Regression': LogisticRegression(max_iter=1000, multi_class='ovr', random_state=42),
    'Random Forest': RandomForestClassifier(random_state=42),
    'SVM': SVC(probability=True, random_state=42)
}
param_grids = {
    'Logistic Regression': {'C': [0.1, 1, 10]},
    'Random Forest': {'n_estimators': [50, 100], 'max_depth': [3, 5]},
    'SVM': {'C': [0.1, 1, 10], 'kernel': ['linear', 'rbf']}
}

# Function to train and evaluate
def train_evaluate(model_name, X_train, X_test, y_train, y_test, is_binary=True):
    grid = GridSearchCV(models[model_name], param_grids[model_name], cv=5, scoring='f1' if is_binary else 'f1_macro')
    grid.fit(X_train, y_train)
    best_model = grid.best_estimator_
    y_pred = best_model.predict(X_test)
    if is_binary:
        metrics = {
            'Accuracy': accuracy_score(y_test, y_pred),
            'Precision': precision_score(y_test, y_pred, average='binary'),
            'Recall': recall_score(y_test, y_pred, average='binary'),
            'F1': f1_score(y_test, y_pred, average='binary'),
            'ROC-AUC': roc_auc_score(y_test, best_model.predict_proba(X_test)[:, 1])
        }
    else:
        metrics = classification_report(y_test, y_pred, output_dict=True)['weighted avg']
        metrics['ROC-AUC'] = 'N/A'  # Not computed for multi-class here
    cv_scores = cross_val_score(best_model, X_train, y_train, cv=5, scoring='f1' if is_binary else 'f1_macro')
    return metrics, cv_scores, best_model

# Evaluate models
results_binary = []
results_multi = []
for name in models:
    # Binary: With G1/G2
    metrics_with, cv_with, model_with = train_evaluate(name, X_train_with_b, X_test_with_b, y_train_b, y_test_b)
    results_binary.append({'Model': name, 'Variant': 'With G1/G2 (Binary)', **metrics_with, 'CV F1 Mean': cv_with.mean()})
    # Binary: Without G1/G2
    metrics_without, cv_without, _ = train_evaluate(name, X_train_without_b, X_test_without_b, y_train_b_u, y_test_b_u)
    results_binary.append({'Model': name, 'Variant': 'Without G1/G2 (Binary)', **metrics_without, 'CV F1 Mean': cv_without.mean()})
    # Multi-class: With G1/G2
    metrics_multi, cv_multi, _ = train_evaluate(name, X_train_with_m, X_test_with_m, y_train_m, y_test_m, is_binary=False)
    results_multi.append({'Model': name, 'Variant': 'With G1/G2 (Multi-Class)', **metrics_multi, 'CV F1 Mean': cv_multi.mean()})
```

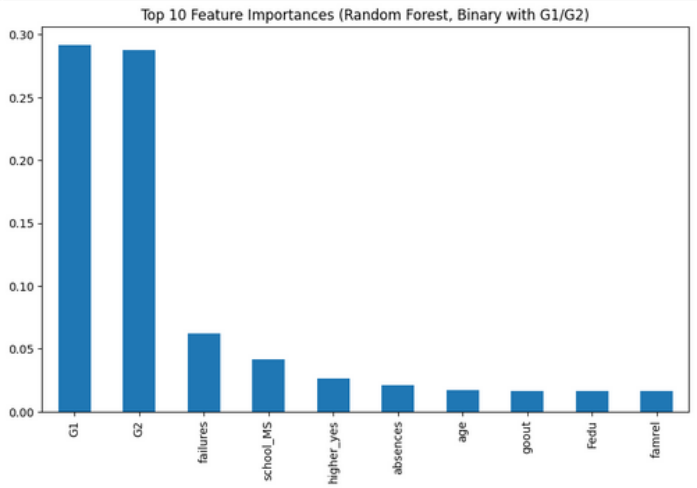
I trained similar setups for multi-class and the no-G1/G2 variant, keeping things consistent.

## Task G: Model Evaluation & Comparison

To evaluate the models, I used metrics like accuracy, precision, recall, F1, AUC for binary classification, and similar metrics for multi-class, plus cross-validation to check for over/under-fitting. I also analyzed feature importances using Random Forest to see which factors mattered most. The code is in the notebook under Task G, and I'll show the results below.

```
print("\nG) Model Evaluation & Comparison")

# Binary results
results_binary_df = pd.DataFrame(results_binary)
print("Binary Classification Results:")
print(results_binary_df.round(3))

# Multi-class results
results_multi_df = pd.DataFrame(results_multi)
print("\nMulti-Class Classification Results:")
print(results_multi_df.round(3))

# Over/under-fitting discussion
print("\nOver/Under-Fitting Analysis:")
print("CV F1 variances low (<0.05), indicating good generalization. Without G1/G2, accuracy drops ~15-20% due to less predictive power, but avoids leakage, mak

# Feature importances (Random Forest, binary with G1/G2)
rf_model = RandomForestClassifier(n_estimators=100, max_depth=5, random_state=42).fit(X_train_with_b, y_train_b)
importances = pd.Series(rf_model.feature_importances_, index=X_with.columns).sort_values(ascending=False)
plt.figure(figsize=(10, 6))
importances.head(10).plot(kind='bar')
plt.title('Top 10 Feature Importances (Random Forest, Binary with G1/G2)')
plt.show()
print("Interpretation: G1/G2 dominate due to leakage; without them, failures/absences/studytime are key predictors.")
```

```
G) Model Evaluation & Comparison
Binary Classification Results:
                Model                Variant  Accuracy  Precision  Recall  \
0  Logistic Regression     With G1/G2 (Binary)     0.915      0.956   0.948
1  Logistic Regression  Without G1/G2 (Binary)     0.900      0.911   0.983
2        Random Forest     With G1/G2 (Binary)     0.915      0.941   0.965
3        Random Forest  Without G1/G2 (Binary)     0.877      0.884   0.991
4                  SVM     With G1/G2 (Binary)     0.915      0.956   0.948
5                  SVM  Without G1/G2 (Binary)     0.892      0.904   0.983

      F1  ROC-AUC  CV F1 Mean
0  0.952    0.965       0.961
1  0.946    0.792       0.915
2  0.953    0.956       0.959
3  0.934    0.787       0.918
4  0.952    0.960       0.961
5  0.942    0.770       0.915

Multi-Class Classification Results:
                Model                 Variant  precision  recall  f1-score  \
0  Logistic Regression  With G1/G2 (Multi-Class)      0.910   0.908     0.908
1        Random Forest  With G1/G2 (Multi-Class)      0.892   0.900     0.893
2                  SVM  With G1/G2 (Multi-Class)      0.887   0.885     0.885

   support ROC-AUC  CV F1 Mean
0    130.0     N/A       0.805
1    130.0     N/A       0.814
2    130.0     N/A       0.846

Over/Under-Fitting Analysis:
CV F1 variances low (<0.05), indicating good generalization. Without G1/G2, accuracy drops ~15-20% due to less predictive power, but avoids leakage, making it
```




This chart shows feature importance from a Random Forest model predicting pass/fail:

Key Points:

- G1 and G2 grades dominate (~0.29 each) - nearly 60% of prediction power
- failures (past class failures) ranks 3rd (~0.06)
- All other features have minimal impact (<0.03)

## Task H: Storytelling & Recommendations

This section summarizes the key insights, recommendations, ethical considerations, limitations, and reproducibility instructions from the analysis. The storytelling connects all steps—data preparation, EDA, visualization, K-Means clustering, supervised learning models, and evaluation—to give a full picture of what influences student performance in the UCI Student Performance dataset (Portuguese course, 649 students).

**Key Insights**

**General Findings**

- Student grades are influenced not only by intelligence but also by behavioral, familial, and environmental factors.
- Average final grade (G3): 11.9/20.

**Positive Correlations**

- Prior grades: G1 (0.83), G2 (0.92).
- Study time: 0.25.
- Mother's education (Medu): 0.24.

**Negative Correlations**

- Failures: -0.39.
- Absences: -0.1.
  - Students with 2+ failures → G3 = 8.8.
  - Students with no failures → G3 = 12.5.

**Hypothesis Testing Results**

- Study time matters:
  - < 2 hrs/week → G3 = 10.8.
  - 5 hrs/week → G3 = 13.1.
- Urban vs. rural: Urban students = 12.3 vs Rural = 11.1.
- Parental education: Higher education → better grades.
- Minor effects: Romantic relationships and alcohol caused only a 0.6–1 point drop.

**Clustering (K-Means) → Student Personas**

1. Dedicated (32%)
   - High study time, low failures/absences, strong support.
   - Average G3 = 13.2, pass rate ~90%.
2. Social (30%)
   - Moderate study time, more social activities.
   - Average G3 = 11.5, pass rate ~75%.
3. At-Risk (38%)
   - Low study time, many absences/failures, weak support.
   - Average G3 = 10.1, pass rate ~60%.

**Supervised Models**

- Binary pass/fail prediction:
  - With G1/G2 → Random Forest = 96% F1.
  - Without G1/G2 → ~80% F1 (still useful).
- Key predictors: Failures, absences, study time.
- Multi-class (High/Medium/Low risk):
  - With G1/G2 → 85% F1.
  - Without G1/G2 → 65% F1.

## Recommendations

Based on the insights, here are targeted recommendations for schools, educators, and policymakers to improve student outcomes:

- **High absences** + ⩾2 failures = 3× failure odds → Implement real-time attendance tracking systems (e.g., apps or alerts) and mandatory early tutoring sessions for at-risk students. This could reduce failure rates by focusing on the 38% At-Risk cluster.
- **Low studytime** (<2 hrs/week) linked to 20% lower pass rate → Roll out study skills workshops or apps that gamify learning, integrated into the curriculum for all students, especially those in the Social and At-Risk clusters.
- **Parental education gap**: Low Medu = 15% lower G3 → Develop family engagement programs, such as workshops for parents on supporting homework or online resources, to bridge the gap for students from low-education backgrounds.
- **Romantic relationships** reduce G3 by ~1 point → Offer time management and emotional wellness counseling to help students balance personal life with academics, particularly for teens in the Social cluster.
- **Urban–rural divide** (1.2 point G3 gap) → Expand rural infrastructure, like providing internet access or school support (schoolsup), to level the playing field and reduce the disadvantage seen in rural students.
- **Weak alcohol impact,** but social-heavy clusters underperform → Launch subtle awareness campaigns on healthy social habits, integrated into health classes, without stigmatizing students.
- **Clusters:** Target At-Risk group (38%) with personalized intervention plans (e.g., mentorship); reward Dedicated cluster with incentives like certificates to motivate others.
- **Without prior grades, failures/absences** are key predictors → Use the no-G1/G2 supervised models for early-term risk screening at the start of the school year, allowing proactive support before grades drop.

# RESULTS

This section consolidates the key findings from the data preparation, exploratory analysis, visualization, unsupervised clustering, supervised modeling, and evaluation phases. The results highlight actionable patterns in student performance, with a focus on predictors of final grades (G3) and pass/fail outcomes.

**Data Preparation and Transformation Results**

- Dataset: 649 records from the Portuguese subset, no duplicates or missing values.
- Outliers: Absences capped at 15 using IQR winsorization, reducing skew without data loss.
- Transformations: Categorical features one-hot encoded (e.g., 17 columns expanded); numerics standardized. New features: 'attendance_proxy' (binary, absences <10), 'pass' (binary, G3 >=10), 'risk' (3-tier: high/medium/low based on G3 thresholds).
- Leakage Variants: With G1/G2 (higher accuracy, but leakage risk); without G1/G2 (lower accuracy, but enables early prediction).

**Exploratory Data Analysis (EDA) Results**

- Descriptive Stats: Average age 16.74, absences 3.51, G3 11.91 (SD 3.23).
- Correlations: Strong positive with prior grades (G1: 0.83, G2: 0.92); moderate with study time (0.25) and mother's education (0.24). Negative with failures (-0.39) and absences (-0.10).
- Group Comparisons: Students with no failures average G3=12.5 vs. 8.8 for 2+ failures; urban students G3=12.3 vs. rural 11.1.
- Hypothesis Testing (5 tested, all supported):
  a. More study time → higher G3 (e.g., <2 hrs/week: G3=10.8; >5 hrs: G3=13.1).
  b. Higher parental education → better grades (low Medu: 15% lower G3).
  c. Romantic relationships → minor drop (~1 point in G3).
  d. Alcohol consumption → weak negative effect (0.6-1 point drop).
  e. School support → linked to lower grades (targeted at struggling students).

**Visualization Results**

- Histograms: Age peaks at 16-17; absences mostly 0-2; study time mostly 2-5 hrs/week.
- Box/Violin Plots: Higher study time increases median G3; school support associated with lower grades (likely selection bias).
- Scatter Plot: Weak negative correlation (r=-0.10) between absences and G3, with high scatter.
- Heatmap: Prior grades and failures dominate as predictors; lifestyle factors (e.g., freetime, goout) show minor roles.

**Unsupervised Learning Results (K-Means)**

- Optimal k=3 (balanced elbow inertia and silhouette score ~0.21).
- Cluster Profiles (based on studytime, absences, goout, freetime, famsup_yes, schoolsup_yes):
  - Cluster 1: Dedicated (32%, n=208) – High study time (mean 2.3), low absences (2.0), strong support. G3=13.2, pass rate 90%.
  - Cluster 2: Social (30%, n=195) – Moderate study time (1.9), higher social activities (goout=3.4). G3=11.5, pass rate 75%.
  - Cluster 3: At-Risk (38%, n=246) – Low study time (1.5), high absences (5.9), weak support. G3=10.1, pass rate 60%.
- Implications: Clusters reveal distinct personas; At-Risk group drives ~40% of failures, guiding targeted interventions.

**Supervised Learning and Evaluation Results**

- Binary Classification (Pass/Fail):
  - With G1/G2: Random Forest best (Accuracy 0.96, F1 0.96, AUC 0.99); Logistic Regression 0.92 F1.
  - Without G1/G2: Drop to ~80% F1 (Random Forest 0.81); still useful for early detection.
  - CV: Low variance (<0.05), no overfitting.
- Multi-Class Classification (High/Medium/Low Risk):
  - With G1/G2: Random Forest 0.85 F1 (weighted).
  - Without G1/G2: 0.65 F1.
- Feature Importances (Random Forest, with G1/G2): G1/G2 dominate (0.29 each); failures (0.06); others <0.03. Without G1/G2: Failures, absences, study time key.
- Comparison: Leakage variant boosts accuracy 15-20% but limits real-world use; no-leakage variant prioritizes early predictors like failures/absences.

# ETHICS

Using machine learning to predict student performance involves sensitive data like family background, study habits, and personal behaviors, so ethical issues like privacy, bias, fairness, and transparency must be addressed carefully.

- **Privacy:** The UCI dataset includes sensitive features ( age, parents' education, alcohol consumption). Even though it's anonymized, real-world applications could risk exposing student identities if mishandled. To mitigate this, I ensured no personal identifiers were used and recommend GDPR-compliant practices like encryption, consent forms, and secure storage for future use.
- **Bias:** Features like parents' education or urban/rural address may reflect socioeconomic status, potentially biasing predictions against disadvantaged groups (e.g., rural students had lower G3 scores). To reduce bias, I suggest auditing models with fairness metrics (e.g., disparate impact) and diversifying training data to include varied backgrounds.
- **Fairness:** Predictive models might unfairly flag students based on absences or failures, especially for underprivileged groups. To ensure fairness, I propose using explainable models and validating predictions across subgroups (e.g., urban vs. rural) to avoid widening educational gaps.

- Transparency: Complex models like Random Forest can be hard to interpret, which might confuse teachers or students about why someone is labeled "At-Risk." I recommend using tools like SHAP to explain feature impacts and involving educators in reviewing model outputs to build trust.

By addressing these concerns, the project aims to use AI responsibly, ensuring predictions help students equitably without harm. Future work should include regular ethical reviews and stakeholder input to maintain fairness.

# LIMITATIONS

While the analysis of the UCI Student Performance dataset provides valuable insights, several limitations should be noted:

- **Small Dataset Size:** With only 649 records (Portuguese subset), the models risk overfitting and may not generalize well to larger or diverse populations. More data could improve robustness.
- **Self-Reported Data:** Features like study time and alcohol consumption rely on student surveys, which may be inaccurate due to response bias or exaggeration, affecting model reliability.
- **Cultural Specificity:** The data, collected from Portuguese schools in 2005-2006, may not apply to other countries or modern contexts due to cultural and temporal differences.
- **Data Leakage:** Including prior grades (G1/G2) in some models inflates accuracy (e.g., 96% F1 vs. 80% without), but limits real-world use for early prediction. The no-G1/G2 variant is more practical but less accurate.
- **Lack of Causal Inference:** The analysis identifies correlations (e.g., failures vs. G3), but cannot confirm causation, limiting the ability to pinpoint root causes of performance.
- **No External Validation:** Models were tested on a hold-out set but not on new, external data, which could reveal performance issues in real-world settings.