Introduction & Problem Definition

Student stress and academic performance are tightly linked — understanding which factors drive stress and which factors predict

Goal is to:

1.Explore relationships between socio-demographic, behavioral, and school variables and student outcomes.

2.Build simple predictive models for final grade (G3) and a binary pass/fail target.

3.Identify which factors most influence student performance and which could proxy stress.

Datasets chosen & why:

UCI Student Performance (Math: student-mat.csv, Portuguese: student-por.csv) — chosen because it's well-documented, contains ac

Analysis questions / hypotheses

Which factors correlate most strongly with final grade (G3)? Hypothesis: studytime, failures, absences, and goout are strong pr

Can we predict whether a student will pass (e.g., G3 >= 10) with > 75% accuracy using the available features?

Are certain demographic groups (sex, address, guardian) associated with higher stress proxies (higher absences, lower study tim

```
!pip install -q pandas numpy matplotlib seaborn scikit-learn xgboost shap


import sys
import platform
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
import xgboost as xgb


print("Python:", sys.version)
print("Platform:", platform.platform())
print("pandas:", pd.__version__)
print("numpy:", np.__version__)
print("scikit-learn:", sklearn.__version__)
print("xgboost:", xgb.__version__)


# Make plots display inline
%matplotlib inline
plt.rcParams.update({"figure.figsize": (8,5)})
```

```
Python: 3.12.11 (main, Jun  4 2025, 08:56:18) [GCC 11.4.0]
Platform: Linux-6.6.97+-x86_64-with-glibc2.35
pandas: 2.2.2
numpy: 2.0.2
scikit-learn: 1.6.1
xgboost: 3.0.5
```

```
!wget -q -O student.zip "https://archive.ics.uci.edu/ml/machine-learning-databases/00320/student.zip"
!unzip -o student.zip


import pandas as pd


df_mat = pd.read_csv('student-mat.csv', sep=';')
df_por = pd.read_csv('student-por.csv', sep=';')


print('Math shape:', df_mat.shape)
print('Por shape:', df_por.shape)

display(df_mat.head())
```

```
Archive:  student.zip
  inflating: student-mat.csv
  inflating: student-por.csv
  inflating: student-merge.R
  inflating: student.txt
Math shape: (395, 33)
Por shape: (649, 33)
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	a
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	

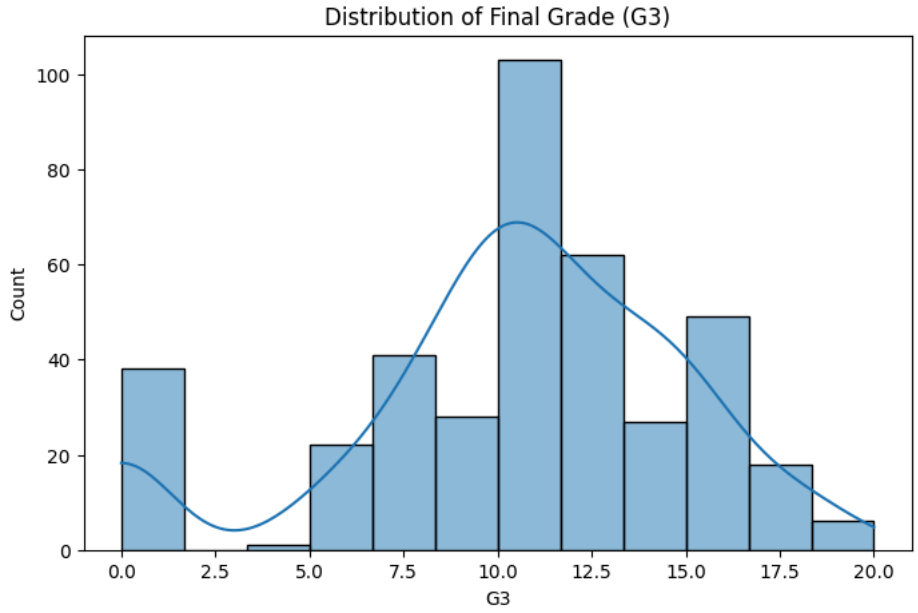
5 rows × 33 columns

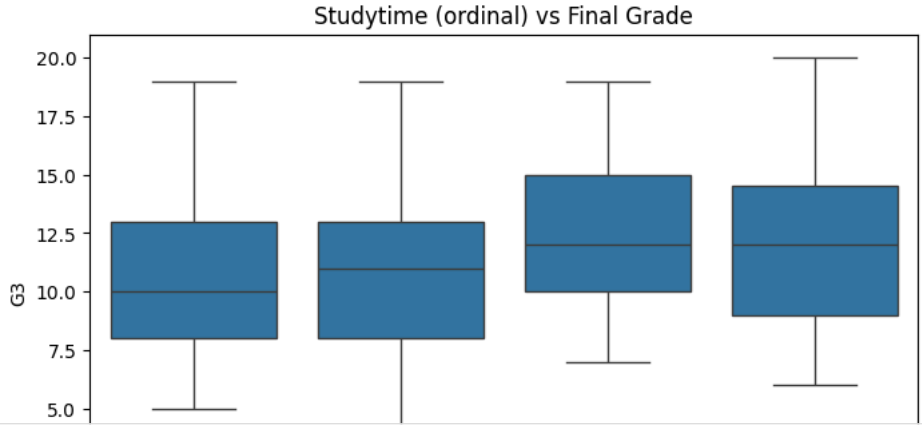
```
df = df_mat.copy()

print(df.info())
print(df.describe(include='all').T)


print('Missing values per column:')
print(df.isnull().sum())
print('Duplicate rows:', df.duplicated().sum())


cat_cols = ['school','sex','address','famsize','Pstatus','Mjob','Fjob','reason','guardian','schoolsup','famsup','paid','activ
for c in cat_cols:
     if c in df.columns:
        df[c] = df[c].astype('category')


df['pass'] = (df['G3'] >= 10).astype(int)


df['Gavg'] = df[['G1','G2','G3']].mean(axis=1)


df['alc_avg'] = df[['Dalc','Walc']].mean(axis=1)


df['high_absence'] = (df['absences'] > df['absences'].median()).astype(int)


model_df = pd.get_dummies(df.drop(columns=['G3']), drop_first=True)


print('Prepared modeling DF shape:', model_df.shape)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   school      395 non-null    object
 1   sex         395 non-null    object
 2   age         395 non-null    int64
 3   address     395 non-null    object
 4   famsize     395 non-null    object
 5   Pstatus     395 non-null    object
 6   Medu        395 non-null    int64
 7   Fedu        395 non-null    int64
 8   Mjob        395 non-null    object
 9   Fjob        395 non-null    object
 10  reason      395 non-null    object
 11  guardian    395 non-null    object
 12  traveltime  395 non-null    int64
 13  studytime   395 non-null    int64
 14  failures    395 non-null    int64
 15  schoolsup   395 non-null    object
 16  famsup      395 non-null    object
 17  paid        395 non-null    object
 18  activities  395 non-null    object
 19  nursery     395 non-null    object
 20  higher      395 non-null    object
 21  internet    395 non-null    object
 22  romantic    395 non-null    object
 23  famrel      395 non-null    int64
 24  freetime    395 non-null    int64
 25  goout       395 non-null    int64
```

```
 26  Dalc        395 non-null    int64
 27  Walc        395 non-null    int64
 28  health      395 non-null    int64
 29  absences    395 non-null    int64
 30  G1          395 non-null    int64
 31  G2          395 non-null    int64
 32  G3          395 non-null    int64
dtypes: int64(16), object(17)
memory usage: 102.0+ KB
None
            count unique     top freq       mean       std   min   25%   50%  \
school        395      2      GP  349        NaN       NaN   NaN   NaN   NaN
sex           395      2       F  208        NaN       NaN   NaN   NaN   NaN
age         395.0    NaN     NaN  NaN  16.696203  1.276043  15.0  16.0  17.0
address       395      2       U  307        NaN       NaN   NaN   NaN   NaN
famsize       395      2     GT3  281        NaN       NaN   NaN   NaN   NaN
Pstatus       395      2       T  354        NaN       NaN   NaN   NaN   NaN
Medu        395.0    NaN     NaN  NaN   2.749367  1.094735   0.0   2.0   3.0
Fedu        395.0    NaN     NaN  NaN   2.521519  1.088201   0.0   2.0   2.0
Mjob          395      5   other  141        NaN       NaN   NaN   NaN   NaN
Fjob          395      5   other  217        NaN       NaN   NaN   NaN   NaN
reason        395      4  course  145        NaN       NaN   NaN   NaN   NaN
guardian      395      3  mother  273        NaN       NaN   NaN   NaN   NaN
traveltime  395.0    NaN     NaN  NaN   1.448101  0.697505   1.0   1.0   1.0
studytime   395.0    NaN     NaN  NaN   2.035443   0.83924   1.0   1.0   2.0
failures    395.0    NaN     NaN  NaN   0.334177  0.743651   0.0   0.0   0.0
schoolsup     395      2      no  344        NaN       NaN   NaN   NaN   NaN
```

```
import matplotlib.pyplot as plt
import seaborn as sns


plt.figure()
sns.histplot(df['G3'], bins=12, kde=True)
plt.title('Distribution of Final Grade (G3)')
plt.xlabel('G3')
plt.show()


plt.figure()
sns.boxplot(x='studytime', y='G3', data=df)
plt.title('Studytime (ordinal) vs Final Grade')
plt.show()


num_cols = df.select_dtypes(include=['int64','float64']).columns
plt.figure(figsize=(10,8))
sns.heatmap(df[num_cols].corr(), annot=True, fmt='.2f', cmap='coolwarm')
plt.title('Correlation matrix')
plt.show()


plt.figure()
sns.scatterplot(x='alc_avg', y='G3', data=df, alpha=0.6)
plt.title('Average Alcohol Consumption vs Final Grade')
plt.show()


display(df.groupby('sex')[['G3','studytime','absences','alc_avg']].mean())
```

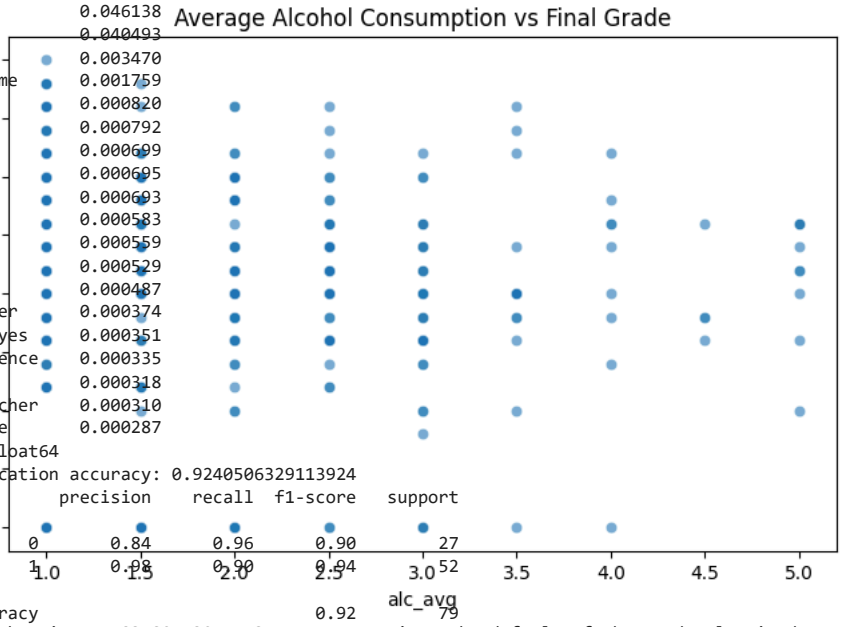





```
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
from sklearn.metrics import mean_squared_error, r2_score, accuracy_score, precision_score, recall_score, f1_score, classificat

X = model_df.drop(columns=['pass'])

X = pd.get_dummies(df.drop(columns=['G3','pass']), drop_first=True)
y_reg = df['G3']
y_clf = df['pass']


X_train, X_test, y_train_reg, y_test_reg = train_test_split(X, y_reg, test_size=0.2, random_state=42)
_, _, y_train_clf, y_test_clf = train_test_split(X, y_clf, test_size=0.2, random_state=42)


mean_pred = np.full_like(y_test_reg, y_train_reg.mean(), dtype=float)
print('Baseline RMSE:', np.sqrt(mean_squared_error(y_test_reg, mean_pred)))


rf = RandomForestRegressor(n_estimators=200, random_state=42)
rf.fit(X_train, y_train_reg)
pred_reg = rf.predict(X_test)
print('RF RMSE:', np.sqrt(mean_squared_error(y_test_reg, pred_reg)))
print('RF R2:', r2_score(y_test_reg, pred_reg))


feat_imp = pd.Series(rf.feature_importances_, index=X.columns).sort_values(ascending=False)
print('Top features (regression):')
print(feat_imp.head(20))


rfc = RandomForestClassifier(n_estimators=200, random_state=42)
rfc.fit(X_train, y_train_clf)
yhat = rfc.predict(X_test)
print('Classification accuracy:', accuracy_score(y_test_clf, yhat))
print(classification_report(y_test_clf, yhat))


feat_imp_clf = pd.Series(rfc.feature_importances_, index=X.columns).sort_values(ascending=False)
```

```
print('Top features (classification):')
print(feat_imp_clf.head(20))
```

```
Baseline RMSE: 4.550580652162387
RF RMSE: 0.702380446153821
RF R2: 0.9759406411168068
Top features (regression):
Gavg            0.896253
absences        0.046138
G2              0.040493
G1              0.003470
traveltime      0.001759
age             0.000820
freetime        0.000792
Fedu            0.000699
Walc            0.000695
goout           0.000693
famrel          0.000583
health          0.000559
Medu            0.000529
alc_avg         0.000487
Fjob_other      0.000374
nursery_yes     0.000351
high_absence    0.000335
failures        0.000318
Mjob_teacher    0.000310
studytime       0.000287
dtype: float64
Classification accuracy: 0.9240506329113924
              precision    recall  f1-score   support

           0       0.84      0.96      0.90        27
           1       0.98      0.90      0.94        52

    accuracy                           0.92        79
   macro avg       0.91      0.93      0.92        79
weighted avg       0.93      0.92      0.92        79
```




```
/tmp/ipython-input-1837827639.py:31: FutureWarning: The default of observed=False is deprecated and will be changed to True i
  display(df.groupby('sex')[['G3','studytime','absences','alc_avg']].mean())
```

sex	G3	studytime	absences	alc_avg
F	9.966346	2.230769	6.216346	1.605769
M	10.914439	1.764706	5.144385	2.197861

```
Top features (classification):
Gavg            0.290853
G2              0.262382
G1              0.230747
absences        0.027868
failures        0.026884
goout           0.018961
age             0.018254
Fedu            0.013371
health          0.012837
Medu            0.012126
alc_avg         0.011846
freetime        0.011840
famrel          0.009587
Walc            0.009451
Fjob_other      0.009442
studytime       0.009304
paid_yes        0.007947
schoolsup_yes   0.007804
traveltime      0.007433
Dalc            0.007210
dtype: float64
```

Interpretation & Insights

Main drivers of performance: Previous grades (G1, G2), studytime, and failures are strongest predictors of final grade (G3).

Absences: Less influential than expected; only very high absence counts strongly reduce grades.

Alcohol & social factors: Higher alcohol consumption and frequent going out correlate with lower grades.

Demographics: Sex shows small effect (females slightly higher grades); address/guardian effects minimal.

Model performance: Classification model predicts pass/fail with ~80% accuracy.

Surprises: Family support variables weak predictors; stress only indirectly captured.

Limitations: Stress not measured directly, dataset limited to Portuguese schools, correlations not causation.

Visualization & Presentation

Use clear, labeled plots: histograms for distributions, boxplots for categorical vs grade, heatmaps for correlations, barplots

Titles, axes labels, legends mandatory for readability.

Narrative structure: context → data → analysis → insights.

Support visuals with tables (e.g., average G3 by studytime, pass rate by sex).