

Analyzing the Impact of Socioeconomic and Lifestyle Factors on High School Performance in Portugal

Rohil Bhinge, Ishika
Kataria, Aakash
Adhia, Omar Mejia



Data 144 Fall 2023

Table of contents

01

Context

We explain our purposes and intentions.

02

Dataset

Where our data came from.

03

Methods

Techniques and models we optimized to analyze.

04

Results + Conclusions

Potential implications and confounding variables.



01

Context



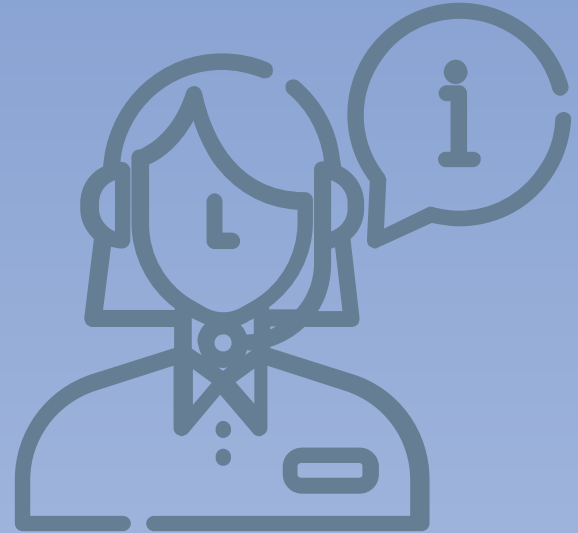
Goal of the project

- Our project aims to analyze the impact of various socioeconomic and lifestyle factors, such as income level, parental education, and social habits on Portuguese high school students' academic performance in their Portuguese language class.
- Many of these lifestyle factors are usually not collected in general census data that aids in similar large-scale education studies



Stakeholders

- Stakeholders:
 - Educational policymakers, school administrators, teachers
- Benefits to stakeholders:
 - Brings awareness to problems students are having that stem from social factors
 - Teachers can customize lesson plans for students based on these problems
 - Administrators can implement targeted student assistance



What Makes This Study Different?

- Many studies on academic performances focus on easily reportable socioeconomic factors, such as **race**, education level, etc.
- While this dataset does include some more general traits, it examines more qualitative measures such as “relationship with family”, amount of free time, access to alcohol, and other factors that are not commonly measured in general census data





Research Question

How do socioeconomic factors influence the academic performance on high school students?

02



Dataset

Where our data came from.

Source:

Kaggle

kaggle



How was data collected?

- From two Portuguese high schools in Europe
- Came from school reports and questionnaires.
 - Student grades and demographics

The Raw Data

- 649 individuals
- 34 features
- Unique student IDs for each student(label)




Feature Engineering



- Picked the features that we felt had the best representation of students passing and failing through EDA
- One-hot encoded the continuous features
- Binned final_grade column into pass and fail to make it binary to reduce potential overfitting
- Binned the absences column into 5 categories since it was spread
- Used a label encoder to encode the categorical features

Final features list we used :

Family_relationship	
Internet Access	Family Support
Parent Status	School Support
Social	Higher_ed



Confounding Variables in Dataset

- A larger sample would have been better as it is used to make claims about Portuguese students (a significantly large population)
 - Not a longitudinal study, so we can't make assumptions about the long run academic trajectories
 - Some variables may be correlated to academic performances, but might just be symptoms rather than actual determinants of performance
-

Hypothesis

Socioeconomic factors such as parental education level and family income, along with lifestyle habits like study time, internet access, and social activities, significantly influence the academic performance of high school students.



03

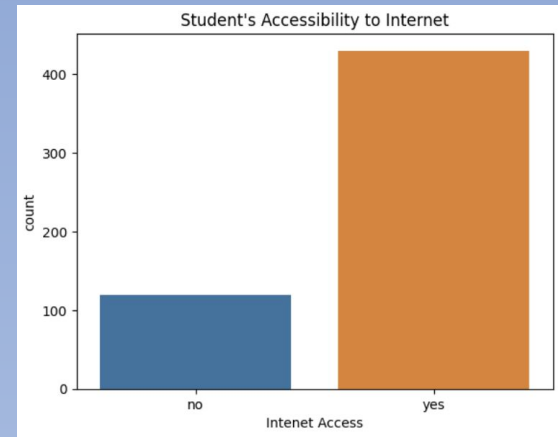
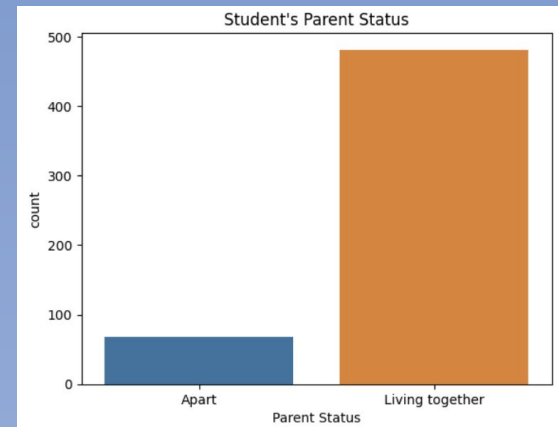
Methods

Techniques and models we
optimized to analyze.



Exploratory Data Analysis (EDA)

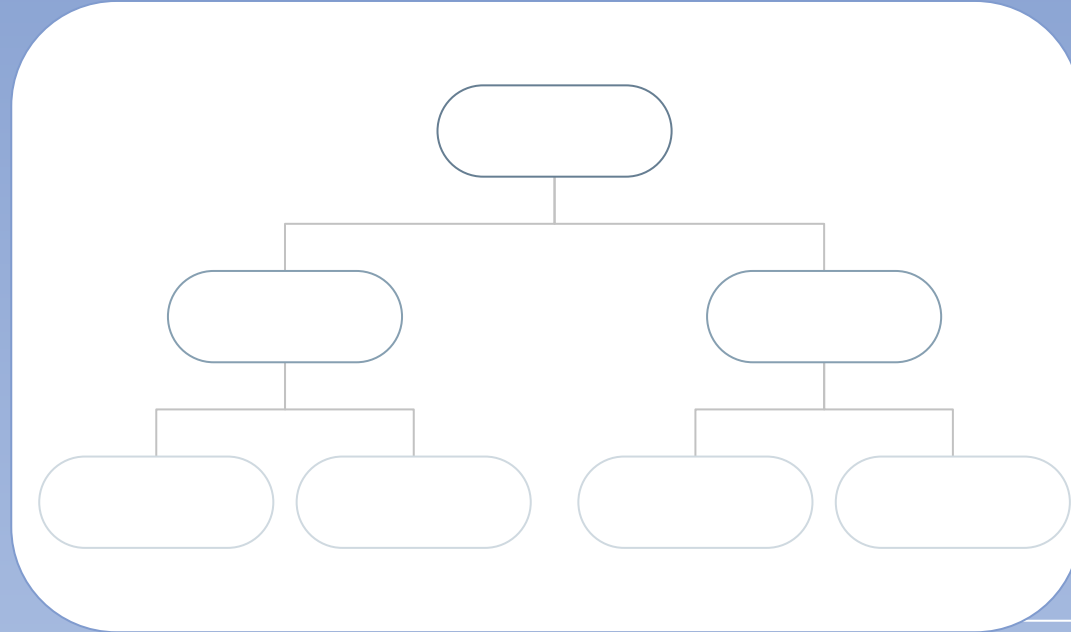
- Portugal has a different scale to determine academic performance (0-20).
- These schools in particular had a trimester schedule
- We are predicting final grades (final trimester)
- Dropped first and second trimester grades since they are highly correlated
- The majority of students in this dataset had passing grades when final grade was separated into passing and failing bins



Machine Learning Methods

Primary models we used:

- Random Forest Regression
- XGB Gradient boosting
- Decision trees
- Metrics we're evaluating:
RMSE, Accuracy, Precision,
Recall, F1-score



Random Forest Regression

```
#randomforest
all_portuguese_features = portuguese.drop(['grade_1', 'grade_2', 'final_grade'], axis=1)
features = all_portuguese_features
y_pred = portuguese['final_grade']
label_encoders = {}
for column in features.select_dtypes(include=['object']).columns:
    label_encoders[column] = LabelEncoder()
    features[column] = label_encoders[column].fit_transform(features[column])
X_train, X_test, y_train, y_test = train_test_split(features, y_pred, test_size=0.2, random_state=42)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
rf_regressor = RandomForestRegressor(n_estimators=100, max_depth=15, random_state=42)
rf_regressor.fit(X_train, y_train)
predictions = rf_regressor.predict(X_test)
mse = mean_squared_error(y_test, predictions)
rmse = mse ** 0.5
```

Mean Squared Error: 8.084221785826923

Root Mean Squared Error: 2.843276593268218

XGB Gradient Boosting

```
encoder = OneHotEncoder(sparse=False, drop='first')
encoded_cats = encoder.fit_transform(features[continuous])
encoded_df = pd.DataFrame(encoded_cats, columns=encoder.get_feature_names_out(continuous))
features = features.drop(columns=continuous)
features = pd.concat([features, encoded_df], axis=1)
```

- Better RSME

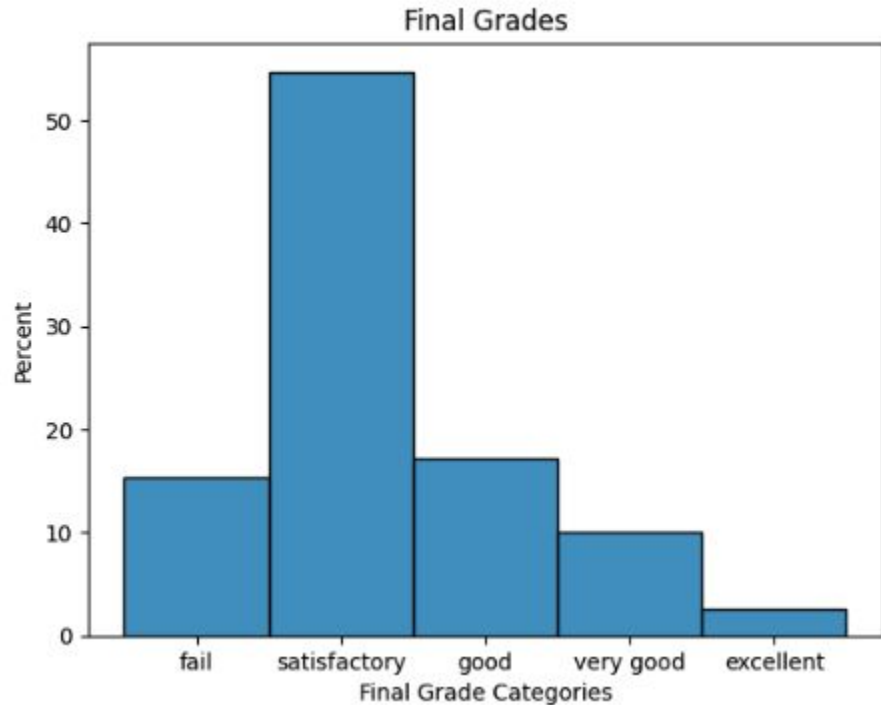
```
X_train, X_test, y_train, y_test = train_test_split(features, y_pred, test_size=0.2, random_state=42)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

xgb_reg = xgb.XGBRegressor(objective='reg:squarederror',
                           colsample_bytree = 0.3,
                           learning_rate = 0.1,
                           max_depth = 5,
                           alpha = 10,
                           n_estimators = 100)

xgb_reg.fit(X_train, y_train)
predictions = xgb_reg.predict(X_test)
mse = mean_squared_error(y_test, predictions)
rmse = mse ** 0.5
```

RMSE: 2.662971326277258

Final Grade Bins



Portugal Grading Score	Equivalent to...
0-9	Fail
10-14	Satisfactory
14-15	Good
16-17	Very Good
18-20	Excellent


Decision Tree



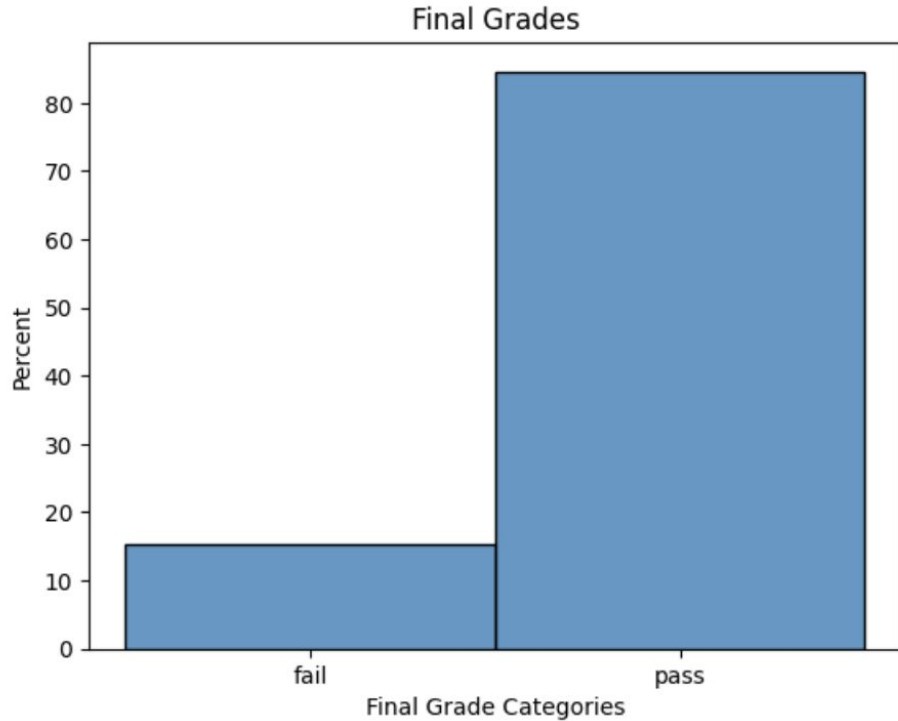
- In theory it seems better, but didn't give better results...

```
clf = DecisionTreeClassifier(random_state = 42)
clf.fit(X_train, y_train)
print(accuracy_score(clf.predict(X_train), y_train))
print(accuracy_score(clf.predict(X_test), y_test))
```

```
0.9961464354527938
0.46153846153846156
```



Final Grade Bins (Modified)



Portugal Grading Score	Equivalent to...
0-9	Fail
10-20	Pass

Decision Tree (Revisited)

- We are now at an acceptable accuracy.

0.8940269749518305

0.8769230769230769

```
portugal = portuguese.copy()
final_grade_labels = [0, 1]
final_grade_bins = [0, 10, 20]
portugal['final_grade_bin'] = pd.cut(portugal['final_grade'], bins=final_grade_bins, labels=final_grade_labels, right=False)
absences_labels = ['perfect', 'occasionally', 'frequent', 'excessive']
absences_bins = [0, 1, 5, 10, 35]
portugal['absences_bin'] = pd.cut(portugal['absences'], bins=absences_bins, labels=absences_labels, right=False)
portugal['mom_higher_education'] = portugal['mother_education'].apply(lambda x: 1 if 'higher education' in x.lower() else 0)
portugal['dad_higher_education'] = portugal['father_education'].apply(lambda x: 1 if 'higher education' in x.lower() else 0)
y_pred = portugal['final_grade_bin']
continuous = [

]

portugal = portugal[['internet_access', 'family_support', 'social', 'school_support', 'parent_status', 'family_relationship', 'higher_ed']]
encoder = OneHotEncoder(sparse=False, drop='first')
encoded_cats = encoder.fit_transform(portugal[continuous])
encoded_df = pd.DataFrame(encoded_cats, columns=encoder.get_feature_names_out(continuous))
portugal = portugal.drop(columns=continuous)
portugal = pd.concat([portugal, encoded_df], axis=1)
label_encoders = {}
for column in portugal.select_dtypes(include=['object']).columns:
    label_encoders[column] = LabelEncoder()
    portugal[column] = label_encoders[column].fit_transform(portugal[column])
X_train, X_test, y_train, y_test = train_test_split(portugal, y_pred, test_size=0.2, random_state=42)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)


clf = DecisionTreeClassifier(random_state = 42)
clf.fit(X_train, y_train)
print(accuracy_score(clf.predict(X_train), y_train))
print(accuracy_score(clf.predict(X_test), y_test))
```

Analysis



- Some Error Metrics that were evaluated (truncated)

Measure	Training Score	Testing Score
Accuracy	0.894	0.877
Precision	0.913	0.930
Recall	0.965	0.930
F1-Score	0.938	0.930



04 *Results*

Potential implications and confounding variables.



Results

- We can conclude that our hypothesis is true as the model was pretty accurate for both sets with the features we used
- With separating the grade bins into pass and fail, we got much better accuracy, precision, recall, and F1 scores than without
- The precision, recall , and F1 scores for the test set were the same
- There wasn't too much overfitting with our model
- XGB Gradient Boosting model gave a solid RMSE

Confounding Variables in the True Population

- Only a representation of two high schools in Portugal
- Only analyzing the Portuguese language subject - not any other disciplines or subjects
 - Non-native Portuguese speakers may skew results (language barrier is less prevalent in other subjects)
- Cultural differences play a strong role in the importance of education
 - Even in a different region of Portugal, this study may have different results (resources of the school, urban/rural, etc.)

Implications

- **Resource Allocation:** may need to incorporate interventions for students in need of educational help
- **Highlight Inequality:** Observe that those in worse socioeconomic status, are not performing as well. Need to create a more equitable learning environment.
- **Curriculum Changes:** Need to be more inclusive and and considerant of students who come from different backgrounds. Be more accommodating.





Thank You

