# CIS 9650 Group Project- Group X- Student Academic Performance

CUNY-Baruch College

CIS 9650 Professor Ali Koc

Group X

Xin Huang, Xinyue Chen, Tzuyi Li, Wanying Li, Ziling Wan, Szufan Chen

## Table of Contents

## 1. Description of the project

This project seeks to develop a regression model to predict the Mathematics grades of the students given specific values of features in the model. We hope to build a model which is highly accurate and gives us some insight into the features that impact on a student's academic performance.

## 2. Identifying the business problem

Education is one of the most popular topics for all people. Academic performance is essential in one’s personal development. It’s also used as a criterion to assess the education quality of educational institutions.

For our analysis, we will study the target variable: G1, G2, G3, and use the average grade as our target variable. We expect that this predicting analysis can figure out the factors that impact on a student's academic performance. It can help students achieve academic success and give the educators the direction to improve the quality of education.

We can use a correlation plot to see the correlation coefficient among all attributes and grade, so we can know which variables have more impacts to the grade performance. The attributes include the family situation (parents’ education, jobs, family support, guardian...), time allocation (study time, travel time, go out time, absence), alcohol consumption, and so on. Then, we can identify critical attributes to impact grades. We can also use the results to compare with the regression model and decision tree to see if there are difference between the two methods, check the reasons, and then refine our model.

## 3. Attributes

| | |
|---|---|
| school | student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira) |
| sex | student's sex (binary: 'F' - female or 'M' - male) |
| age | student's age (numeric: from 15 to 22) |
| address | student's home address type (binary: 'U' - urban or 'R' - rural) |

| | |
|---|---|
| famsize | family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3) |
| pstatus | parent's cohabitation status (binary: 'T' - living together or 'A' - apart) |
| medu | mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education) |
| fedu | father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education) |
| mjob | mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other') |
| fjob | father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other') |
| reason | reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other') |
| guardian | student's guardian (nominal: 'mother', 'father' or 'other') |
| traveltime | home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour) |
| studytime | weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours) |
| failures | number of past class failures (numeric: n if 1<=n<3, else 4) |
| schoolsup | extra educational support (binary: yes or no) |
| famsup | family educational support (binary: yes or no) |
| paid | extra paid classes within the course subject (Math or Portuguese) (binary: yes or no) |

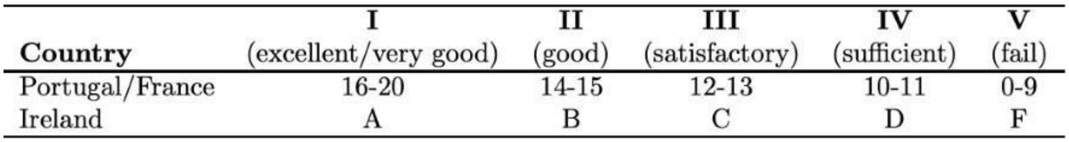
| | |
|---|---|
| activities | extra-curricular activities (binary: yes or no) |
| nursery | attended nursery school (binary: yes or no) |
| higher | wants to take higher education (binary: yes or no) |
| internet | Internet access at home (binary: yes or no) |
| romantic | with a romantic relationship (binary: yes or no) |
| famrel | quality of family relationships (numeric: from 1 - very bad to 5 - excellent) |
| freetime | free time after school (numeric: from 1 - very low to 5 - very high) |
| goout | going out with friends (numeric: from 1 - very low to 5 - very high) |
| dalc | workday alcohol consumption (numeric: from 1 - very low to 5 - very high) |
| walc | weekend alcohol consumption (numeric: from 1 - very low to 5 - very high) |
| health | current health status (numeric: from 1 - very bad to 5 - very good) |
| absences | number of school absences (numeric: from 0 to 93) |
| G1 | first period grade (numeric: from 0 to 20) |
| G2 | second period grade (numeric: from 0 to 20) |
| G3 | final grade (numeric: from 0 to 20, output target) |
| grade_pass | Average grade (Fail 0 - <10. Pass 1- >= 10) |

## Reason for Dropping and Adding Certain Columns

| | |
|---|---|
| School | The number of records of each school is imbalanced. And feature "school" does not contribute to our research. |

| | |
|---|---|
| Age | Student age does not affect our model and different age students are not comparable. |
| Mjob & Fjob | ·There are too many categories under “Mjob” & “Fjob” columns, this will make the model too loose to get a high accuracy. |
| Adding | |
| Grade_pass | Calculating mean of G1, G2 and G3. Take average grade as a final decision of pass/ fail. If less than 10, fail. If over 10, pass. |

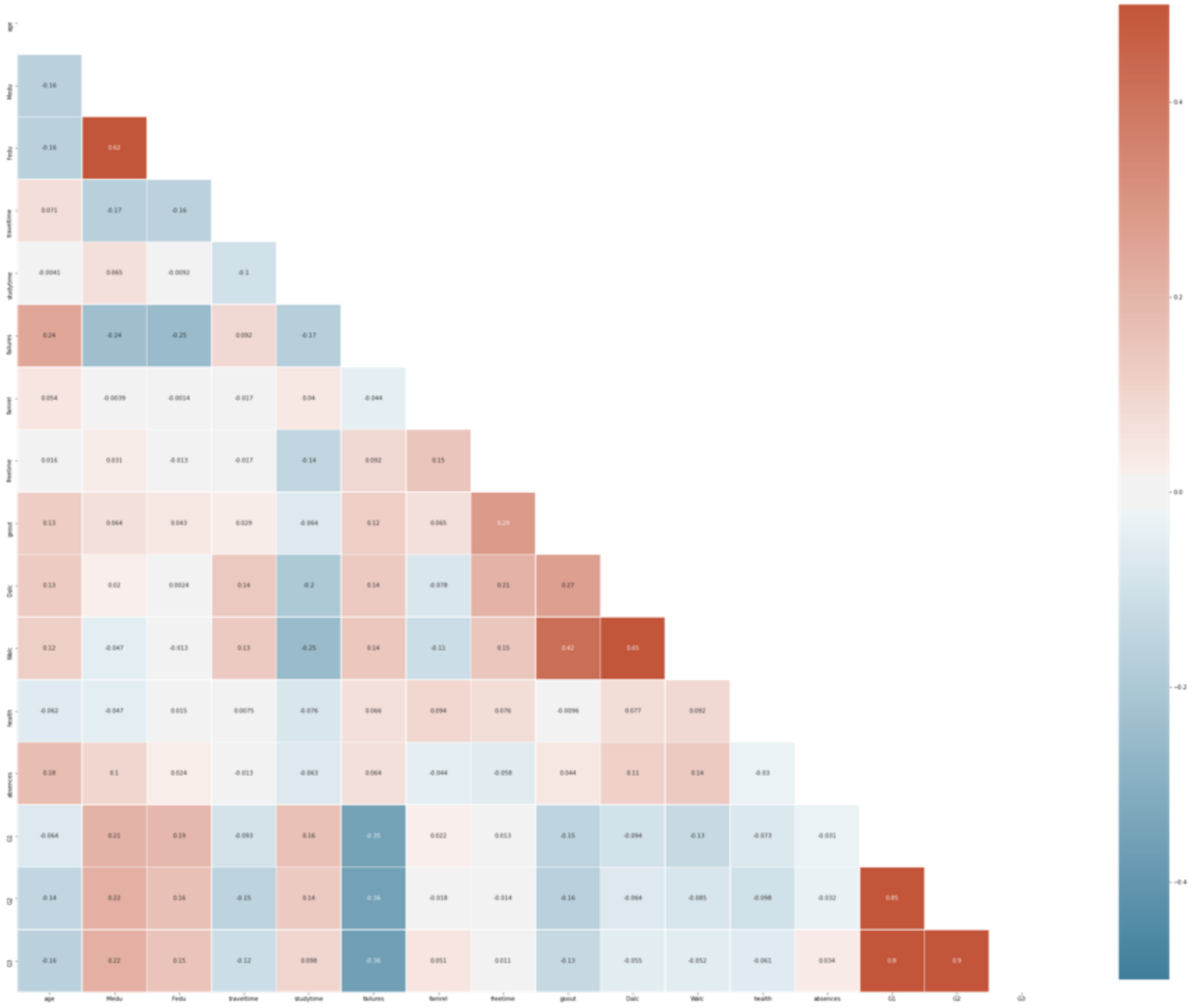
Table 2: The five-level classification system



| Country | I<br>(excellent/very good) | II<br>(good) | III<br>(satisfactory) | IV<br>(sufficient) | V<br>(fail) |
|---|---|---|---|---|---|
| Portugal/France | 16-20 | 14-15 | 12-13 | 10-11 | 0-9 |
| Ireland | A | B | C | D | F |

# 4. Correlation plot

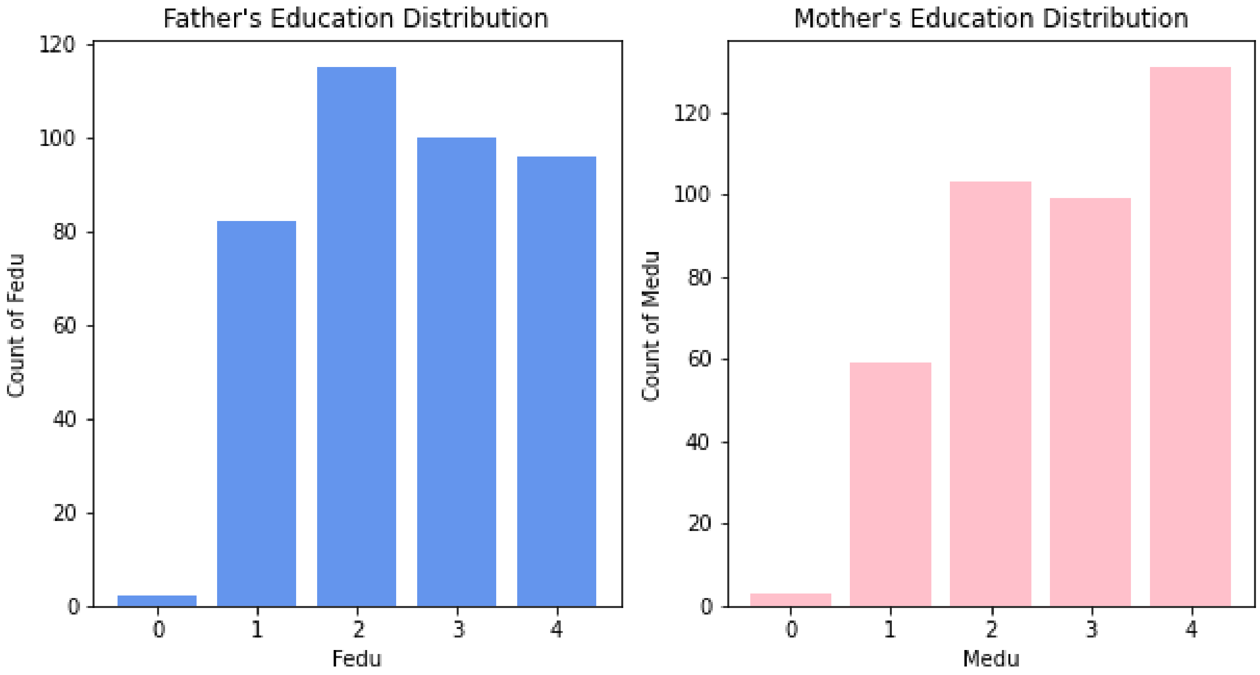
```
#https://seaborn.pydata.org/examples/many_pairwise_correlations.html

# Compute the correlation matrix
corr = df.corr()

# Generate a mask for the upper triangle
mask = np.triu(np.ones_like(corr, dtype=bool))

# Set up the matplotlib figure
f, ax = plt.subplots(figsize=(40, 40))

# Generate a custom diverging colormap
cmap = sns.diverging_palette(230, 20, as_cmap=True)

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.5, vmin=-.5,center=0,annot = True,
            square=True, linewidths=.5, cbar_kws={"shrink": .8})
```



## Fedu & Medu

In this correlation chart, we find the correlation efficient between "Fedu" and "Medu" was 0.62, and we can take a closer look at the relationship. From the following bar chart, we can see Father's education had a different distribution compared to mother's education. Most fathers' education was 2, while mothers tended to have higher education level. Therefore, we choose to keep both Fedu and Medu.

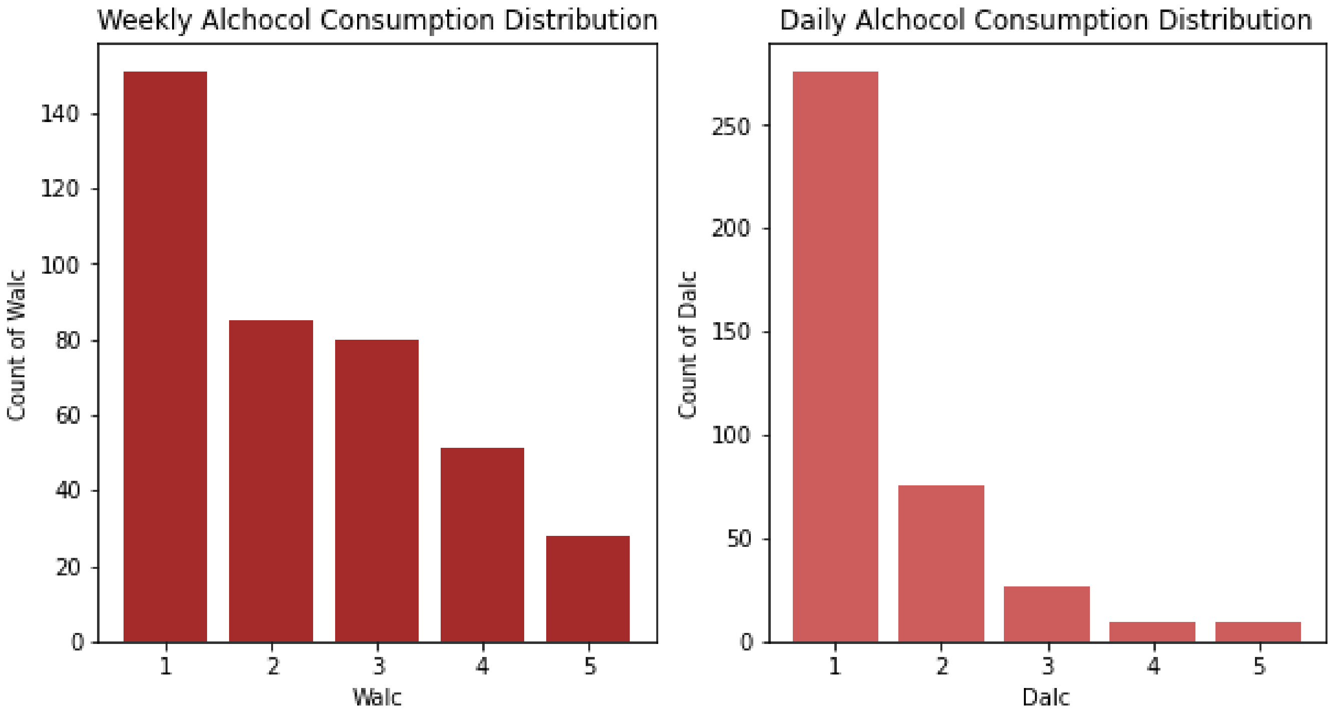
```
plt.rc('figure', figsize=(10, 5))

fig = plt.figure()

ax1 = fig.add_subplot(1, 2, 1)
Fedu = df['Fedu'].value_counts()
Fedu = pd.DataFrame(Fedu)
ax1.set_title("Father's Education Distribution")
plt.ylabel('Count of Fedu')
plt.xlabel('Fedu')
ax1.bar(Fedu.index, Fedu['Fedu'], color = 'cornflowerblue')

ax2 = fig.add_subplot(1, 2, 2)
Medu = df['Medu'].value_counts()
Medu = pd.DataFrame(Medu)
ax2.set_title("Mother's Education Distribution")
plt.ylabel('Count of Medu')
plt.xlabel('Medu')
ax2.bar(Medu.index, Medu['Medu'], color = 'pink')
```



## Walc & Dalc

Also, we find the correlation efficient between "Walc" and "Dalc" was 0.65. We used the bar chart to visualize their distribution and found although both Walc and Dalc had an increase pattern, the distribution seemed to be a little different. After level 2, the Dalc dropped dramatically than Walc. Therefore, we chose to keep both variables.

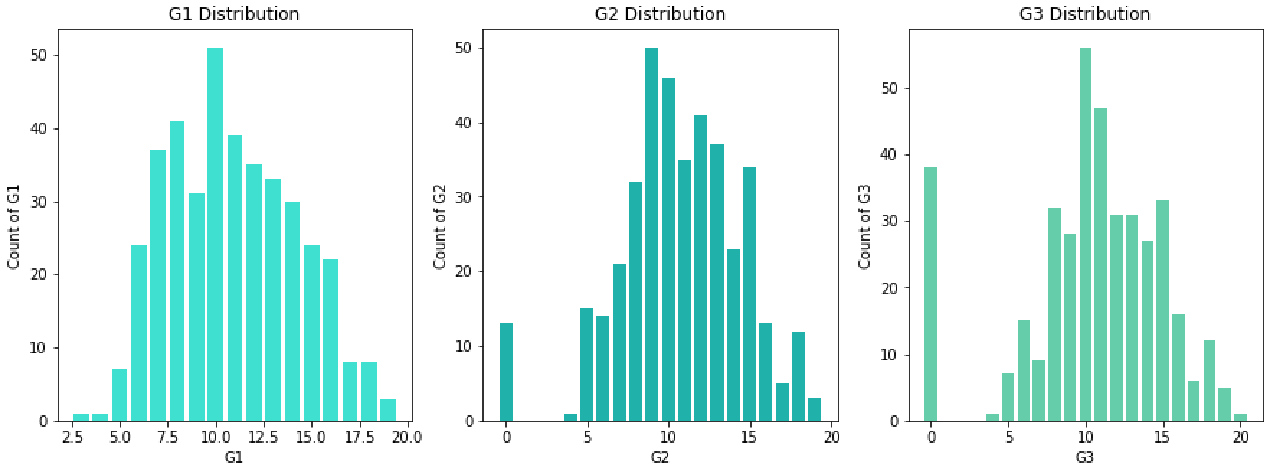
```
plt.rc('figure', figsize=(10, 5))

fig = plt.figure()

ax1 = fig.add_subplot(1, 2, 1)
Walc = df['Walc'].value_counts()
Walc = pd.DataFrame(Walc)
ax1.set_title("Weekly Alchocol Consumption Distribution")
plt.ylabel('Count of Walc')
plt.xlabel('Walc')
ax1.bar(Walc.index, Walc['Walc'], color = 'Brown')

ax2 = fig.add_subplot(1, 2, 2)
Dalc = df['Dalc'].value_counts()
Dalc = pd.DataFrame(Dalc)
ax2.set_title("Daily Alchocol Consumption Distribution")
plt.ylabel('Count of Dalc')
plt.xlabel('Dalc')
ax2.bar(Dalc.index, Dalc['Dalc'], color = 'indianred')
```



## G1, G2 and G3

The 3 grades seemed to have high correlation. We visualized their distribution to see it clearly. We found most students fall around 10 points in the three grades. However, more students got 0 in G2 and G3, while G1 had fewer students with 0 points. The 3 grades distributions were different, so we chose to keep all of them.

```
plt.rc('figure', figsize=(15, 5))

fig = plt.figure()

ax1 = fig.add_subplot(1, 3, 1)
G1 = df['G1'].value_counts()
G1 = pd.DataFrame(G1)
ax1.set_title("G1 Distribution")
plt.ylabel('Count of G1')
plt.xlabel('G1')
ax1.bar(G1.index, G1['G1'], color = 'turquoise')

ax2 = fig.add_subplot(1, 3, 2)
G2 = df['G2'].value_counts()
G2 = pd.DataFrame(G2)
ax2.set_title("G2 Distribution")
plt.ylabel('Count of G2')
plt.xlabel('G2')
ax2.bar(G2.index, G2['G2'], color = 'lightseagreen')

ax3 = fig.add_subplot(1, 3, 3)
G3 = df['G3'].value_counts()
G3 = pd.DataFrame(G3)
ax3.set_title("G3 Distribution")
plt.ylabel('Count of G3')
plt.xlabel('G3')
ax3.bar(G3.index, G3['G3'], color = 'mediumaquamarine')
```

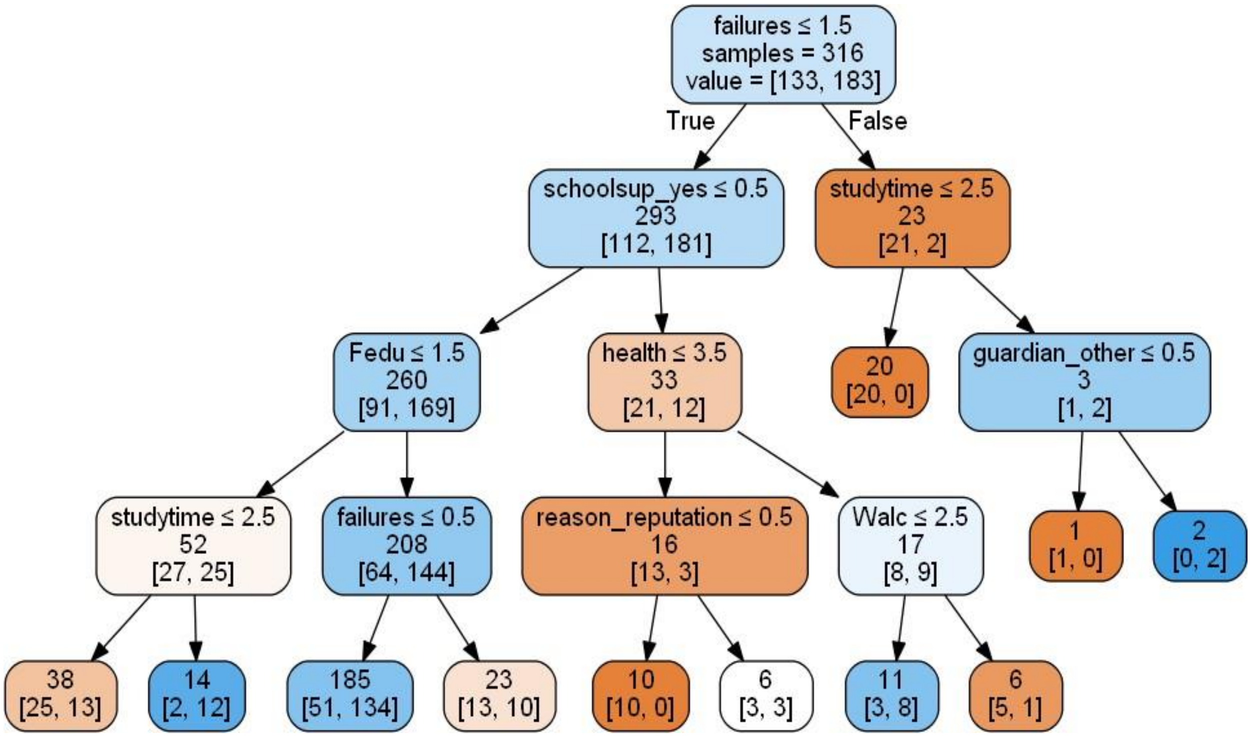


To sum up, we choose to keep Medu, Fedu, Walc, Dalc, G1, G2, and G3 in our analysis. As for grades, we choose to convert 3 grades into average grade to do further analysis.

# 5. Models

## (1) Decision Tree

```
fullClassTree = DecisionTreeClassifier(max_depth=4,random_state = 1)
fullClassTree.fit(train_X, train_y)
plotDecisionTree(fullClassTree, feature_names=train_X.columns)
```



```
prediction_train = fullClassTree.predict(train_X)#use the DT model to predict on the tr aining data
prediction_valid = fullClassTree.predict(valid_X)#use the DT model to predict on the validation data
# precision
print("precision on test is:",precision_score(valid_y,prediction_valid))
 # recall
print("recall on test is:",recall_score(valid_y,prediction_valid))
#f1
print("f1 on test is:",f1_score(valid_y,prediction_valid))
print("Logistic Regression:Accuracy on train is:",accuracy_score(train_y,prediction_tra in))
print("Logistic Regression:Accuracy on test is:",accuracy_score(valid_y,prediction_valid))
```

```
precision on test is: 0.7291666666666666
recall on test is: 0.7291666666666666
```

f1 on test is: 0.7291666666666665

Logistic Regression:Accuracy on train is: 0.7373417721518988
Logistic Regression:Accuracy on test is: 0.6708860759493671

```
importances = fullClassTree.feature_importances_
important_df = pd.DataFrame({'feature': train_X.columns, 'importance': importances})#,"std": std})
important_df = important_df.sort_values('importance',ascending=False)
 print(important_df)
```

```
        feature             importance
4       failures            0.393097
3       studytime           0.197139
21      schoolsup_yes       0.122222
1       Fedu                0.094747
9       Walc                0.062102
10      health              0.049043
18      reason_reputation   0.047716
20      guardian_other      0.033932
19      guardian_mother     0.000000
22      famsup_yes          0.000000
0       Medu                0.000000
17      reason_other        0.000000
24      activities_yes      0.000000
25      nursery_yes         0.000000
26      higher_yes          0.000000
27      internet_yes        0.000000
23      paid_yes            0.000000
14      famsize_LE3         0.000000
16      reason_home         0.000000
15      Pstatus_T           0.000000
13      address_U           0.000000
12      sex_M               0.000000
11      absences           0.000000
8        Dalc               0.000000
7        goout              0.000000
6        freetime           0.000000
5        famrel             0.000000
2       traveltime          0.000000
28      romantic_yes        0.000000
```

## (2) Logistic Regression

```
# partition data
df = pd.get_dummies(df, drop_first=True)
df.columns
predictors = ['Medu', 'Fedu', 'traveltime', 'studytime','failures', 'famrel',
        'freetime', 'goout', 'Dalc', 'Walc', 'health', 'absences', 'sex_M',
        'address_U', 'famsize_LE3', 'Pstatus_T','reason_home', 'reason_other',
        'reason_reputation', 'guardian_mother', 'guardian_other',
        'schoolsup_yes', 'famsup_yes', 'paid_yes', 'activities_yes',
        'nursery_yes', 'higher_yes', 'internet_yes', 'romantic_yes']
X=df[predictors]
y=df['grade_pass']
# partition data
train_X, valid_X, train_y, valid_y = train_test_split(X, y,test_size=0.2, random_state=1)

# fit a logistic regression (set penalty=l2 and C=1e42 to avoid regularization)
logit_reg = LogisticRegression(penalty="l2", C=1e42, solver='liblinear')
logit_reg.fit(train_X, train_y)

print('intercept ', logit_reg.intercept_[0])
print(pd.DataFrame({'coeff': sorted(abs(logit_reg.coef_[0]),reverse=True)}, index=X.columns))
print()
print('AIC', AIC_score(valid_y, logit_reg.predict(valid_X), df = len(train_X.columns) + 1))
```

```
intercept -0.1934938195086815
                    coeff
Medu                1.208985
Fedu                1.177700
traveltime          1.015240
studytime           0.916806
```

```
failures            0.752709
famrel              0.646225
freetime            0.545345
goout               0.477080
Dalc                0.457316
Walc                0.352142
health              0.344441
absences            0.304154
sex_M               0.294370
address_U           0.291227
famsize_LE3         0.270112
Pstatus_T           0.248134
reason_home         0.243812
reason_other        0.224576
reason_reputation   0.219315
guardian_mother     0.174067
guardian_other      0.173306
schoolsup_yes       0.167684
famsup_yes          0.047808
paid_yes            0.047651
activities_yes      0.024265
nursery_yes         0.020892
higher_yes          0.014198
internet_yes        0.012261
romantic_yes        0.000532

AIC 192.07216049893128
```

```
ssificationSummary(train_y, logit_reg.predict(train_X))
classificationSummary(valid_y, logit_reg.predict(valid_X))
```

```
Confusion Matrix (Accuracy 0.7468)

       Prediction
Actual 0  1
     0    77  56
     1    24 159
Confusion Matrix (Accuracy 0.6962)

       Prediction
Actual 0  1
     0   21 10
     1   14 34
```

```
clprediction_valid = logit_reg.predict(valid_X)
prediction_train = logit_reg.predict(train_X)
# precision
print("precision on test is:",precision_score(valid_y,prediction_valid))
# recall
print("recall on test is:",recall_score(valid_y,prediction_valid))
 #f1
print("f1 on test is:",f1_score(valid_y,prediction_valid))
print("Logistic Regression:Accuracy on train is:",accuracy_score(train_y,prediction_train))
print("Logistic Regression:Accuracy on testis:",accuracy_score(valid_y,prediction_valid))
```

```
precision on test is: 0.7727272727272727
recall on test is: 0.7083333333333334
f1 on test is: 0.7391304347826088
Logistic Regression:Accuracy on train is: 0.7468354430379747
Logistic Regression:Accuracy on test is: 0.6962025316455697
```

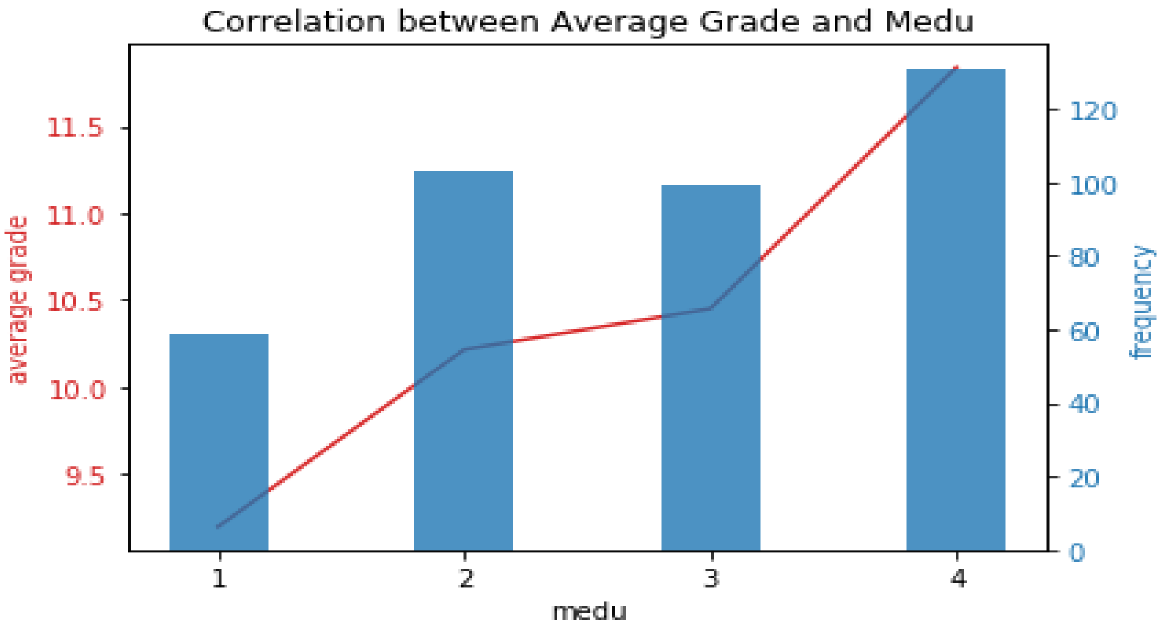
### (3) Comparison

By comparing Decision Tree and Logistic Regression, we figure out that Logistic Regression model has a higher accuracy. Therefore, we believe that the Medu, Fedu, Travel time and study time have significant relationship with students’ Grades.

## 6. Data Analysis

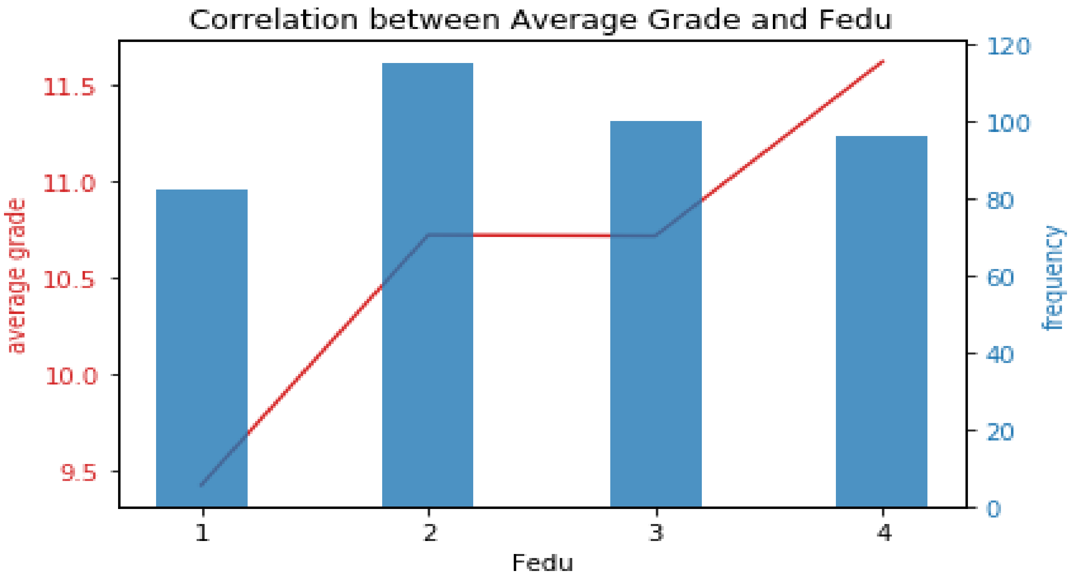
Correlation between Average Grade and Medu

```
# Compute the traveltime's counts and the average_grade based on the traveltime'sgroups

mean_traveltime = df.groupby('traveltime').mean()['average_grade'].values.tolist()
del mean_medu[0] # neglect the first element

count_traveltime = df.groupby('traveltime').count()['average_grade'].values.tolist()
del count_medu[0] # neglect the first element
x = ['1','2','3','4']

fig, ax1 = plt.subplots()

color = 'tab:red'
ax1.set_xlabel('traveltime')
ax1.set_ylabel('average grade', color=color)
ax1.plot(x, mean_traveltime, color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis

color = 'tab:blue'
ax2.set_ylabel('frequency', color=color) # we already handled the x-label with ax1
ax2.bar(x, count_fedu, 0.4, color=color, alpha = 0.8)
ax2.tick_params(axis='y', labelcolor=color)

# add title
plt.title('Correlation between Average Grade and travel')

plt.show()
```



## Correlation between Average Grade and Fedu

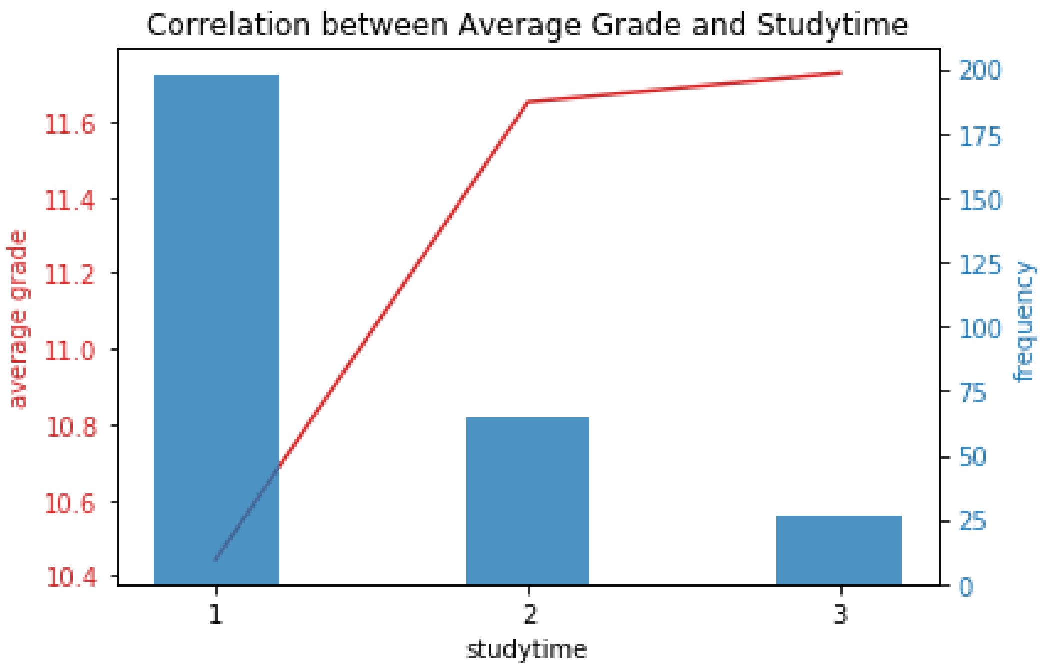
```
# Compute the Fedu's counts and the average_grade based on the Fedu's groups

mean_fedu = df.groupby('Fedu').mean()['average_grade'].values.tolist()
del mean_fedu[0] # neglect the first element

count_fedu = df.groupby('Fedu').count()['average_grade'].values.tolist()
del count_fedu[0] # neglect the first element
x = ['1','2','3','4']

fig, ax1 = plt.subplots()

color = 'tab:red'
ax1.set_xlabel('Fedu')
ax1.set_ylabel('average grade', color=color)
ax1.plot(x, mean_fedu, color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis

color = 'tab:blue'
ax2.set_ylabel('frequency', color=color) # we already han dled the x-label with ax1
ax2.bar(x, count_fedu, 0.4, color=color, alpha = 0.8)
ax2.tick_params(axis='y', labelcolor=color)

# add title
plt.title('Correlation between Average Grade and Fedu')

plt.show()
```



From the graph, the average grade also reflects the increase in parents' education. According to the research named Long-term Effects of Parents’ Education on Children’s Educational and Occupational Success: Mediation by Family Interactions, Child Aggression, and Teenage Aspirations, they did tests to examine the correlation between parent's educational levels and individuals’ educational and occupational success. "The results of this study suggest that the beneficial effects of parental educational level when the child is young are not limited to academic achievement throughout the school years, but have long-term implications for positive outcomes into middle adulthood (i.e., higher educational level, more prestigious occupations) (Eric F. Dubow, Paul Boxer, and L. Rowell Huesmann,2009). "

## Correlation between Average Grade and Study time

```
# Compute the Studytime's counts and the average_grade based on the Studytime's groups

mean_studytime = df.groupby('studytime').mean()['average_grade'].values.tolist()
del mean_studytime[0] # neglect the first element
 print(mean_studytime)

count_studytime = df.groupby('studytime').count()['average_grade'].values.tolist()
del count_studytime[0] # neglect the first element
 print(count_studytime)
[10.442760942760941, 11.65128205128205, 11.728395061728394]
[198, 65, 27]
x = ['1','2','3']

fig, ax1 = plt.subplots()

color = 'tab:red'
ax1.set_xlabel('studytime')
ax1.set_ylabel('average grade', color=color)
ax1.plot(x, mean_studytime, color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis

color = 'tab:blue'
ax2.set_ylabel('frequency', color=color) # we already handled the x-label with ax1
ax2.bar(x, count_studytime, 0.4, color=color, alpha = 0.8)
ax2.tick_params(axis='y', labelcolor=color)

# add title
plt.title('Correlation between Average Grade and Studytime')

plt.show()
```

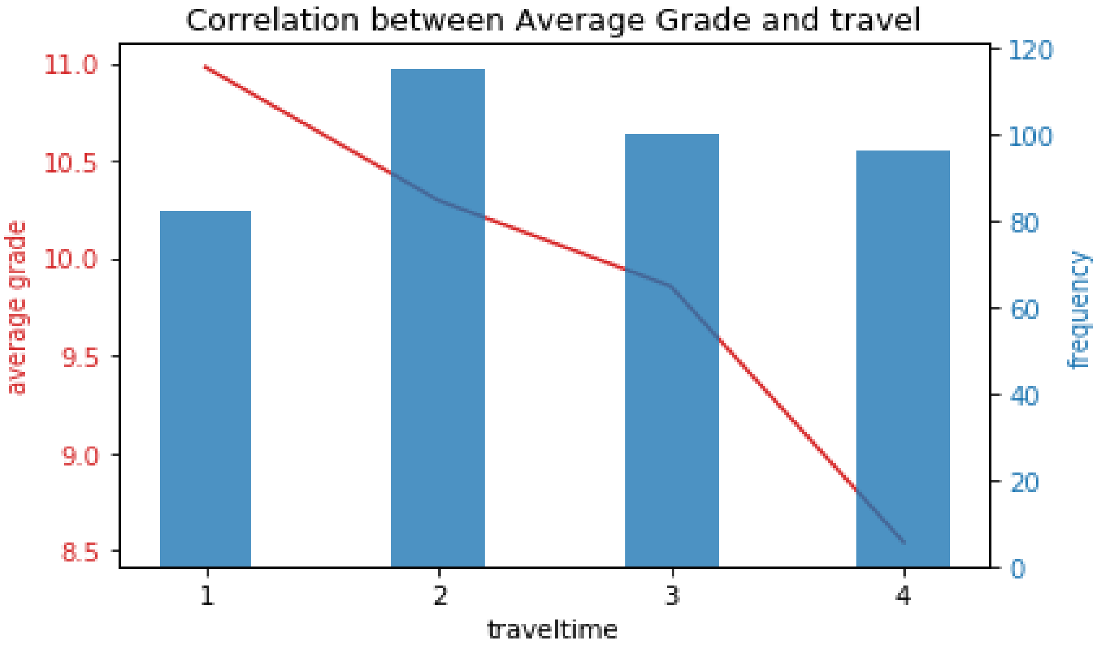


From the graph, when the study time is within 5 hours, we find that students' academic performance increases dramatically with the increase of their study time; when the study time is between 5 hours and 10 hours, the academic performance increases slightly and tends to be flat.

A result of the hypothesis from the study named Length of Study-Time Behaviour and Academic Achievement of Social Studies Education Students in the University of Uyo, which shows that "There is a significant difference between the long and short study time behaviour students' academic performance. Students who study for long hours tend to perform better than those who study for short study time.(Ukpong, D. E., & George, I. N., 2013). "

Correlation between Average Grade and Travel

```python
# Compute the traveltime's counts and the average_grade based on the traveltime's groups

mean_traveltime = df.groupby('traveltime').mean()['average_grade'].values.tolist()
del mean_medu[0] # neglect the first element

count_traveltime = df.groupby('traveltime').count()['average_grade'].values.tolist()

del count_medu[0] # neglect the first element
x = ['1','2','3','4']

fig, ax1 = plt.subplots()

color = 'tab:red'
ax1.set_xlabel('traveltime')
ax1.set_ylabel('average grade', color=color)
ax1.plot(x, mean_traveltime, color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis

color = 'tab:blue'
ax2.set_ylabel('frequency', color=color) # we already handled the x-label with ax1
ax2.bar(x, count_fedu, 0.4, color=color, alpha = 0.8)
ax2.tick_params(axis='y', labelcolor=color)

# add title
plt.title('Correlation between Average Grade and travel')

plt.show()
```



As the graph shown above, traffic time greatly affects academic performance, and the average grade decreases with the increase of the traffic time.

When ask why travel time will affect the academic performance, a thesis titled Associations Between Travel Behavior and the Academic Performance of University Students indicates that "travel time may shorten study time, and study time has been identified as positively contributing to academic performance. Considering that there is limited research examining travel behavior and academic achievement of university students, this field is worthwhile for further study"(WU, Q, 2014).

# 7. Summary

In our project, we analyzed different attributes on the dataset to figure out their impacts on student academic performance. In our analysis, Parents' education levels will have a long-term impact on students' academic performance. Students who study for longer hours tend to perform better. The average grade decreases with the increase of the traffic time.

According to the results, we have some recommendations for students, parents, and educational institutions to improve academic performance.

For students, it's important to spend more time on their study.

For educational institutions, they can provide commute bus to save the travel time for students, which can give students more time to take a rest and focus more on their study. They can also provide after-class tutoring for the students who have challenges on their study.

For parents, although they cannot change their education status immediately, they can choose nearer schools to save their children's travel time. If they're willing to have an advance education and study with their children, this would motivate students. By doing so, the parents would also broaden their horizons and help their children to perform better.

# 8. References

Table 2: The five-level classification system from

Paulo Cortez and Alice Silva. Using Data Mining to Predict Secondary School Student Performance

WU, Q. (2014). Associations Between Travel Behavior and the Academic Performance of University Students. Retrieved 28 November 2020, from https://tigerprints.clemson.edu/all_theses/2063/

Parental Involvement is Key to Student Success. Retrieved from https://www.publicschoolreview.com/blog/parental-involvement-is-key-to-student-success

LR;, D. (n.d.). Long-term Effects of Parents' Education on Children's Educational and Occupational Success: Mediation by Family Interactions, Child Aggression, and Teenage Aspirations. Retrieved November 25, 2020, from

https://pubmed.ncbi.nlm.nih.gov/20390050/

Ukpong, D. E., & George, I. N. (2013). Length of Study-Time Behaviour and Academic Achievement of Social Studies Education Students in the University of Uyo. International Education Studies, 6(3). doi:10.5539/ies.v6n3 p172

Orlando J. Olivares. An Analysis of the Study Time-Grade Association. Retrieved from https://radicalpedagogy.icaap.org/content/issue4_1/06_Olivares.html