Alcohol Consumption and Students’ School Performance

Group - Sober Students

Kennedy Tranel, Jaeyeon Won, Ashley Sackpraseuth

Stage 1: Ask a Question

Stage 1: Ask a Question

- **What is the dataset?**
  - Student Alcohol Consumption
  - More details explained in Stage 2

Stage 1: Ask a Question

- **Research Questions**
  - How is Student Weekday Alcohol Consumption related to Students' Overall Math Grades?
  - What is the predicted Students' Overall Math Grades based on their weekday alcohol consumption?

Stage 1: Ask a Question

- **Benefits of this Project**
  - Identify students' alcohol consumption trends
  - Identify the relationship between Student Weekday Alcohol Consumption and Students' School Performance (Overall Math Grades)
  - Predict Students' Overall Math Grades based on Student Weekday Alcohol Consumption using linear regression model

Stage 2: Get the Data

Stage 2: Get the Data

- **Dataset Description**
  - **What is our data?**
    - Student Alcohol Consumption
  - **What is the source of our data?**
    - Kaggle https://www.kaggle.com/uciml/student-alcohol-consumption
    - Only used *student-mat* dataset
  - **No need to obtain more related dataset**

Stage 2: Get the Data

- **Dataset Description**
  - **What does our data provide?**

- The *Student Alcohol Consumption* dataset contains 395 observations with 33 variables about students in secondary school
- 28 social, demographic, and study information variables: "*age*", "*sex*", "*studytime*", and etc.
- 2 variables about alcohol consumption on the scale of 1 (very low) to 5 (very high): "*Dalc*" (workday alcohol consumption), "*Walc*" (weekend alcohol consumption)
- 3 variables about students' grade on the scale of 0 to 20: "*G1*" (1st period grade), "*G2*" (2nd period grade), "*G3*" (final grade)

Stage 2: Get the Data

- **Getting Started**
    - Importing all necessary functions

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import train_test_split, KFold, cross_val_score, cross_val_predict
from sklearn import metrics
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

Stage 2: Get the Data

- **Getting Started**
  - Reading in our dataset ("*student-mat*")

```
alcohol_consumption = pd.read_csv('student-mat.csv')
```

  - Information of our dataset

```
alcohol_consumption.head()
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6	5	6	6
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4	5	5	6
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	10	7	8	10
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	2	15	14	15
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	4	6	10	10

```
alcohol_consumption.info()

<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
```

Stage 2: Get the Data

- **Data Cleaning**
  - Checked if there is any missing value (no missing values found)

```
#Checking for missing values
alcohol_consumption.isnull().values.any() #We don't have any missing values
```

```
False
```

  - Dropped duplicates and reset index

```
#Dropping duplicates and resetting index
alcohol_consumption.drop_duplicates()
alcohol_consumption = alcohol_consumption.reset_index(drop=True)
```

Stage 2: Get the Data

- **Data Preparation**

  - Created "***overall_grade***", which is the mean of *"G1", "G2",* and *"G3"* (on the scale of 0 to 20)

```
#Creating "overall_grade" variable, which is the mean of G1, G2, and G3
alcohol_consumption["overall_grade"] = (alcohol_consumption["G1"] + alcohol_consumption["G2"] + alcohol_consumption["G3"])/3
alcohol_consumption.head()
```

school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3	overall_grade
GP	F	18	U	GT3	A	4	4	at_home	teacher	...	3	4	1	1	3	6	5	6	6	5.666667
GP	F	17	U	GT3	T	1	1	at_home	other	...	3	3	1	1	3	4	5	5	6	5.333333
GP	F	15	U	LE3	T	1	1	at_home	other	...	3	2	2	3	3	10	7	8	10	8.333333
GP	F	15	U	GT3	T	4	2	health	services	...	2	2	1	1	5	2	15	14	15	14.666667
GP	F	16	U	GT3	T	3	3	other	other	...	3	2	1	2	5	4	6	10	10	8.666667

Stage 2: Get the Data

- **Data Preparation**
  - Created dummy variables for “*Dalc*”

```
#Making dummy variables for "Dalc"
alcohol_consumption = pd.get_dummies(alcohol_consumption, columns = ["Dalc"])
alcohol_consumption.head()
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	absences	G1	G2	G3	overall_grade	Dalc_1	Dalc_2	Dalc_3	Dalc_4
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	6	5	6	6	5.666667	1	0	0	0
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	4	5	5	6	5.333333	1	0	0	0
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	10	7	8	10	8.333333	0	1	0	0
3	GP	F	15	U	GT3	T	4	2	health	services	...	2	15	14	15	14.666667	1	0	0	0
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	6	10	10	8.666667	1	0	0	0

Stage 2: Get the Data

- **Data Preparation**
  - Dropped all variables other than *“sex”, “age”, “overall_grade”, “Dalc”,* and dummy variables for *“Dalc”* (*“Dalc_2”, “Dalc_3”, “Dalc_4”, “Dalc_5”*)

```
#Selecting the variables needed
alcohol_consumption = alcohol_consumption[["sex", "age", "overall_grade", "Dalc_2", "Dalc_3", "Dalc_4", "Dalc_5"]]
alcohol_consumption2 = pd.read_csv('student-mat.csv')
alcohol_consumption["Dalc"] = alcohol_consumption2["Dalc"]
alcohol_consumption.head()
```

	sex	age	overall_grade	Dalc_2	Dalc_3	Dalc_4	Dalc_5	Dalc
0	F	18	5.666667	0	0	0	0	1
1	F	17	5.333333	0	0	0	0	1
2	F	15	8.333333	1	0	0	0	2
3	F	15	14.666667	0	0	0	0	1
4	F	16	8.666667	0	0	0	0	1

Stage 2: Get the Data

- **Dataset Description**
  - **What are our Explanatory and Response Variables?**
    - Explanatory Variable: “*Dalc*”
    - Response Variable: “*overall_grade*”

Stage 3: Explore the Data

Stage 3: Explore the Data:

- **Hypothesis**
  - Weekday alcohol consumption is negatively correlated with students’ overall math grade
  - The predicted overall math grade will be lower as the weekday alcohol consumption goes from “very low” to “very high”

Stage 3: Explore the Data:

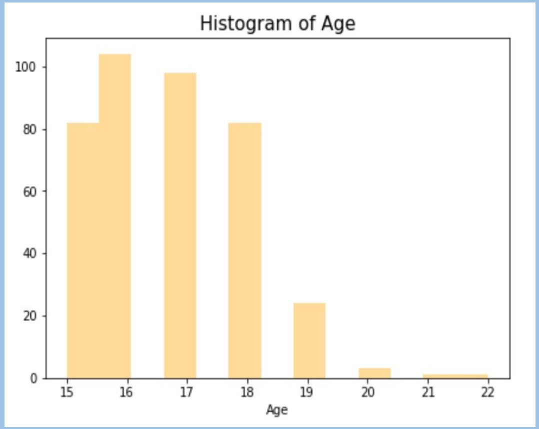
- **Histograms of Quantitative Variables**
  - **Code**

```
#Histograms of Quantitative Variables
#Age
plt.figure(figsize = (7,5))
histogram1 = sns.distplot(alcohol_consumption["age"], kde=False, color = "orange")
histogram1.set_title("Histogram of Age", fontsize = 15)
histogram1.set_xlabel("Age", fontsize = 10)

#overall_grade
plt.figure(figsize = (7,5))
histogram3 = sns.distplot(alcohol_consumption["overall_grade"], kde=False, color = "blue")
histogram3.set_title("Histogram of Overall Grade", fontsize = 15)
histogram3.set_xlabel("Overall Grade", fontsize = 10)
```

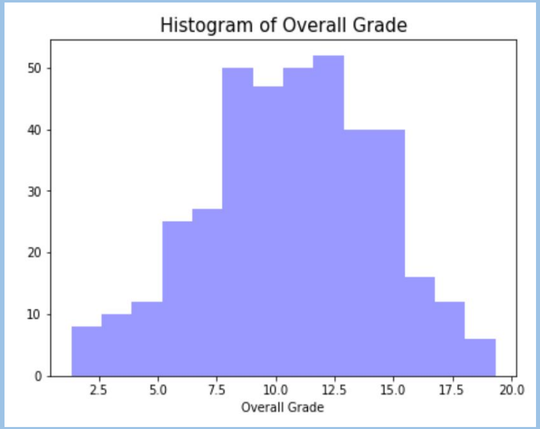
- Used sns.distplot
- Set KDE = False because we did not want to plot a gaussian kernel density estimate
- Set the title and x label manually

Stage 3: Explore the Data:

- **Histograms of Quantitative Variables**
  - **Visualization**



- Skewed right
- Unimodal
- Outliers in age group 20 - 22



- Symmetric
- Unimodal
- No outliers

Stage 3: Explore the Data:

- **Correlation of Quantitative Variables**
  - **Code**

- Correlations

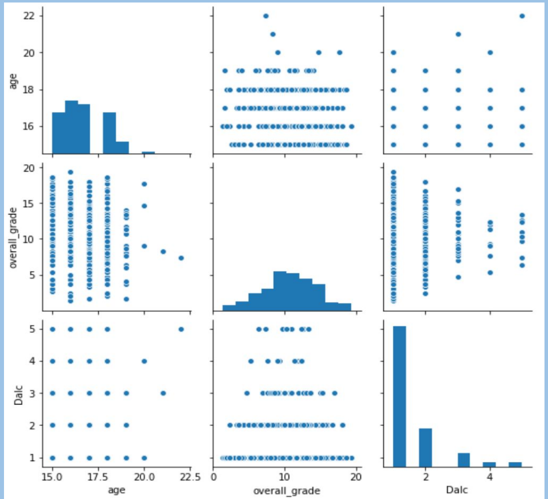
```
#Correlations of Quantitative Variables + Weekday alcohol consumption
alcohol_consumption3 = alcohol_consumption[["age", "overall_grade", "Dalc"]]
corr = alcohol_consumption3.corr()
corr.style.background_gradient(cmap = 'coolwarm').set_precision(2)
```

- sns.pairplot

```
#Relationships amongst Quantitative Variables by Dalc
sns.pairplot(alcohol_consumption, vars = ['age', 'overall_grade', 'Dalc'], diag_kind = 'hist')
```

Stage 3: Explore the Data:

- **Correlation of Quantitative Variables**
  - **Visualization**



	age	overall_grade	Dalc
age	1.00	-0.13	0.13
overall_grade	-0.13	1.00	-0.07
Dalc	0.13	-0.07	1.00

- Age and Overall Grade are negatively and very weakly correlated
- Age and Weekday Alcohol Consumption are positively and weakly correlated
- Overall Grade and Weekday Alcohol Consumption are negatively and weakly correlated

Stage 3: Explore the Data:

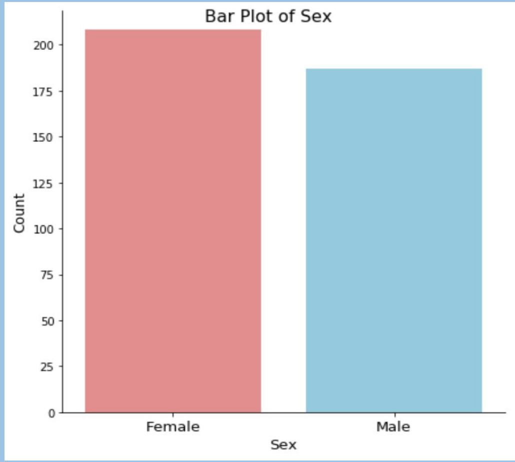
- **Bar Graphs of Categorical Variables**
  - **Code**

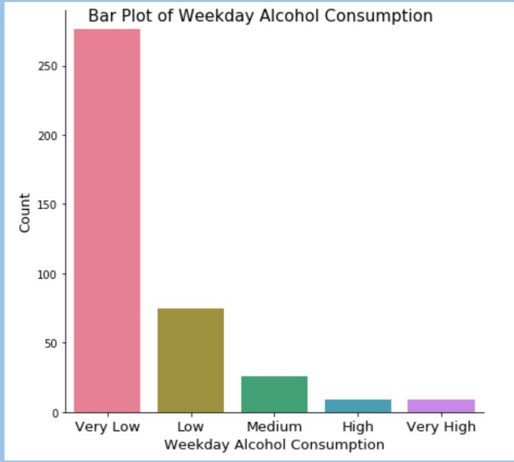
```
#Barplots of Cateogrical Variables
#Sex
my_pal = {"F" : "lightcoral", "M": "skyblue"}
plt.figure(figsize = (7,5))
bar1 = sns.catplot(data = alcohol_consumption, x = "sex", kind = "count", height = 6, palette = my_pal)
xlabels1 = ["Female", "Male"]
bar1.set_xticklabels(xlabels1, fontsize = 13)
bar1.fig.suptitle("Bar Plot of Sex", fontsize = 15)
plt.xlabel("Sex", fontsize = 13)
plt.ylabel("Count", fontsize = 13)

#Dalc
plt.figure(figsize = (7,5))
bar2 = sns.catplot(data = alcohol_consumption, x = "Dalc", kind = "count", height = 6, palette = "husl")
xlabels2 = ["Very Low", "Low", "Medium", "High", "Very High"]
bar2.set_xticklabels(xlabels2, fontsize = 13)
bar2.fig.suptitle("Bar Plot of Weekday Alcohol Consumption", fontsize = 15)
plt.xlabel("Weekday Alcohol Consumption", fontsize = 13)
plt.ylabel("Count", fontsize = 13)
```

- Used sns.catplot
- Set the names of the levels of the variable manually
- Set the title, y label, and x label manually

Stage 3: Explore the Data:

- **Bar Graphs of Categorical Variables**
  - **Visualization**





- Not much difference between the number of female students and male students

- Decreasing trend of weekday alcohol consumption from “very low” to “very high”

Stage 3: Explore the Data:

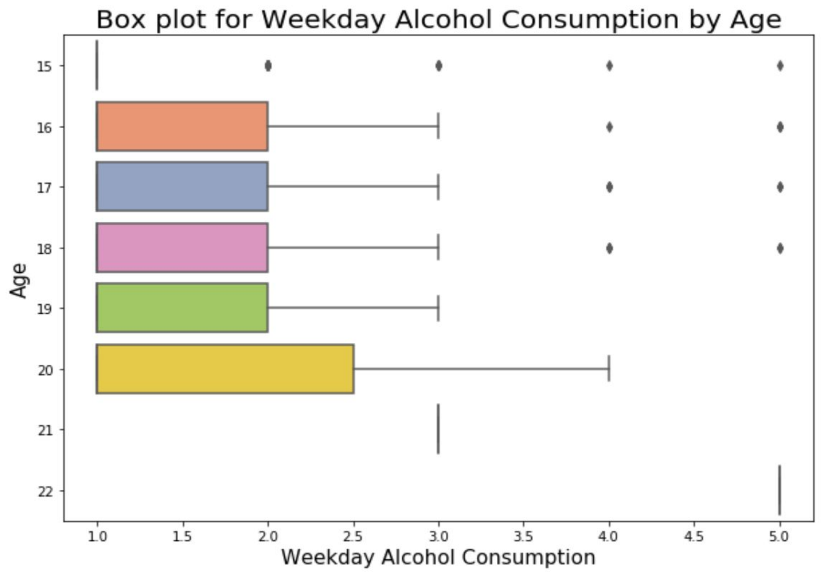
- **Students' Alcohol Consumption Trends**
  - **Code**

```
#Boxplots
#Boxplot of "Dalc" by "age"
plt.figure(figsize=(10,7))
boxplot1 = sns.boxplot(data = alcohol_consumption, y = "age", x = "Dalc", orient = "h", palette = "Set2")
boxplot1.set_title("Box plot for Weekday Alcohol Consumption by Age", fontsize = 20)
boxplot1.set_xlabel("Weekday Alcohol Consumption", fontsize = 15)
boxplot1.set_ylabel("Age", fontsize = 15)

#Boxplot of "Dalc" by "sex"
my_pal = {"F" : "lightcoral", "M": "skyblue"}
plt.figure(figsize=(10,7))
boxplot2 = sns.boxplot(data = alcohol_consumption, y = "sex", x = "Dalc", palette = my_pal)
boxplot2.set_title("Box plot for Weekday Alcohol Consumption by Sex", fontsize = 20)
boxplot2.set_xlabel("Weekday Alcohol Consumption", fontsize = 15)
boxplot2.set_ylabel("Sex", fontsize = 15)
```

- Used sns.boxplot
- Orient = “h” to visualize the plots horizontally
- Set the title, y label, and x label manually

Stage 3: Explore the Data:

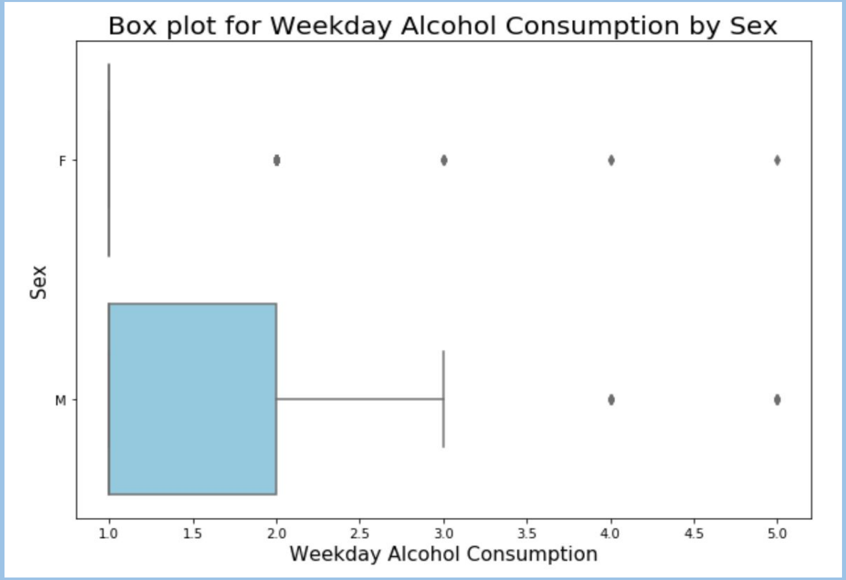
- **Students' Alcohol Consumption Trends by Age**
  - **Visualization**



- Weekday alcohol consumption is largely spread out from Very Low to Very High for age group 15
- Age groups 16 ~ 19 have very similar pattern on weekday alcohol consumption
- Age group 21 and 22 are lacking of the number of observations

Stage 3: Explore the Data:

- **Students' Alcohol Consumption Trends by Sex**
  - **Visualization**



- 50% of male students consume "very low" or "low" alcohol on weekday
- Female students' weekday alcohol consumption is largely spread out from "very low" to "very high"

Stage 3: Explore the Data:

- **Problems to our Dataset**
  - Potentially want more observations
- **Improvements to our Dataset**
  - Could obtain more observations and more updated datasets in the future
    - to see clearer trends of weekday alcohol consumption by students' demographics
    - to predict students' school performance more accurately

Stage 4: Model the Data

Stage 4: Model the Data

- **Machine Learning Model**
  - **Why Linear Regression Model?**
    - Want to understand the relationship between "*weekday alcohol consumption*" and a quantitative variable ("*overall_grade*")
    - Want to predict the value of a quantitative variable ("*overall_grade*")

Stage 4: Model the Data

- **Modeling the Data**

```
#Regression model

#To predict "overall_grade"
X1 = alcohol_consumption[['Dalc_2', 'Dalc_3', 'Dalc_4', 'Dalc_5']]
y1 = alcohol_consumption['overall_grade']

#Split train and test sets
X1_train, X1_test, y1_train, y1_test = train_test_split(X1, y1, test_size = 0.2, random_state = 100)

#Model initialization
model1 = LinearRegression()
model1.fit(X1_train, y1_train)

#Predict
y1_pred = model1.predict(X1_test)

#Comparing Actual values and predicted values
df1 = pd.DataFrame({'Actual': y1_test, 'Predicted': y1_pred})
print(df1.head(10))

#Model evaluation
mae1 = round(mean_absolute_error(y1_test, y1_pred), 4)
rmse1 = round(np.sqrt(mean_squared_error(y1_test, y1_pred)), 4
r21 = round(r2_score(y1_test, y1_pred), 4)

#Printing values
print('The coefficients are {}'.format(model1.coef_, 4))
print('The intercept is {}'.format(model1.intercept_))
print('Mean absolute error (MAE) of the model is {}'.format(mae1))
print('Root mean squared error (RMSE) of the model is {}'.format(rmse1))
print('R-squared score is {}'.format(r21))
```

- **Model**: Linear regression model
- **Train/Test Split**: 80% training, 20% testing
- **Random State**: 100

	Actual	Predicted
188	8.000000	10.859729
365	10.000000	9.748538
190	12.000000	10.859729
353	8.000000	10.608696
166	10.000000	9.748538
75	9.333333	9.748538
231	11.000000	10.859729
341	6.666667	9.748538
380	14.333333	10.859729
267	11.000000	9.748538

Stage 4: Model the Data

- **Fitted Model**

Predicted overall math grade = 10.8597 -1.1112 * (Low) -0.2510 * (Medium) -1.2407 * (High) -0.4431 * (Very High)

Interpretations

- If a student has a very low weekday alcohol consumption, the predicted overall math grade is 10.8597
- If a student has a low weekday alcohol consumption, the predicted overall math grade is 9.7485
- If a student has a medium weekday alcohol consumption, the predicted overall math grade is 10.6087
- If a student has a high weekday alcohol consumption, the predicted overall math grade is 9.6190
- If a student has a very high weekday alcohol consumption, the predicted overall math grade is 10.4166

Stage 4: Model the Data

- **Accuracy Measurement**
  - **Mean Absolute Error (MAE)**: 2.5355
    - The average magnitude of the errors in a set of predictions without considering their direction is 2.5355
  - **Root Mean Squared Error (RMSE)**: 3.1679
    - The square root of the average of squared differences between the prediction and the actual observation is 3.1679
  - **R - squared**: 0.0013
    - 0.1% of the variation in the response variable ("*overall_grade*") is explained by the regression model with the explanatory variable ("*Dalc*")
    - Underfitting

Stage 4: Model the Data

- **6 - Fold Cross - Validation**
    - Can’t use accuracy score as our evaluation because we are doing a regression problem
    - Used RMSE for cross-validated score

```
#Regression model - Perform 6 fold cross validation
scores1 = cross_val_score(model1, X1, y1, cv = 6, scoring = 'neg_root_mean_squared_error')
print('Cross-validated scores:', scores1)

rmse_scores1 = - scores1
print('Cross-validated root mean squared error scores:', rmse_scores1)

print('Final Cross-validation RMSE score:', round(rmse_scores1.mean(), 4), '(', round(rmse_scores1.std(), 4), ')')
```

- Used 6-fold cross validation
- Calculated *negative RMSE* for cross validation score first because Python does not provide RMSE for scoring
- Calculated *RMSE* by multiplying -1 to the *negative RMSE*
- Found the final cross validation RMSE score by averaging the scores from 6 folds and its standard deviation

Stage 4: Model the Data

- **6 - Fold Cross - Validation**
  - **Cross - Validated Root Mean Squared Error Scores**:
    - [3.5017, 3.8215, 4.2362, 3.6730, 3.3116, 3.7171]
  - **Final Cross - Validated RMSE Score**:
    - 3.7102 (sd = 0.2886)
    - Final cross - validated RMSE score is slightly higher than the RMSE of the independent test set

Stage 4: Model the Data

- ## Model Evaluation
  - ### Decision Tree Regressor

```
#DecisionTree Regressor

#Regression model

#To predict "overall_grade"
X2 = alcohol_consumption[['Dalc_2', 'Dalc_3', 'Dalc_4', 'Dalc_5']]
y2 = alcohol_consumption['overall_grade']

#Split train and test sets
X2_train, X2_test, y2_train, y2_test = train_test_split(X2, y2, test_size = 0.2, random_state = 100)

#Model initialization
model2 = DecisionTreeRegressor()
model2.fit(X2_train, y2_train)

#Predict
y2_pred = model2.predict(X2_test)

#Model evaluation
mae2 = mean_absolute_error(y2_test, y2_pred)
rmse2 = np.sqrt(mean_squared_error(y2_test, y2_pred))
r22 = r2_score(y2_test, y2_pred)

#Printing values
print('Mean absolute error of the model is {}:'.format(mae2))
print('Root mean squared error of the model is {}.'.format(rmse2))
print('R-squared score is {}.'.format(r22))
```

- **Model**: Decision Tree Regressor
- **Train/Test Split**: 80% training, 20% testing
- **Random State**: 100

Stage 4: Model the Data

- **Model Evaluation**
  - **Decision Tree Regressor**
    - Mean Absolute Error (MAE): 2.5355
    - Root Mean Squared Error (RMSE): 3.1679
    - R - squared: 0.0013

- Similar values of MAE, RMSE, and R-squared
- Very similar to Linear Regression Model

Stage 4: Model the Data

- **Model Evaluation**
  - **KNN**

```
#KNN

#To predict "overall_grade"
X3 = alcohol_consumption[['Dalc_2', 'Dalc_3', 'Dalc_4', 'Dalc_5']]
y3 = alcohol_consumption['overall_grade']

#Split train and test sets
X3_train, X3_test, y3_train, y3_test = train_test_split(X3, y3, test_size = 0.2, random_state = 100)

#Model initialization
model3 = KNeighborsRegressor(n_neighbors = 1)
model3.fit(X3_train, y3_train)

#Predict
y3_pred = model3.predict(X3_test)

#Model evaluation
mae3 = mean_absolute_error(y3_test, y3_pred)
rmse3 = np.sqrt(mean_squared_error(y3_test, y3_pred))
r23 = r2_score(y3_test, y3_pred)

#Printing values
print('Mean absolute error of the model is {}:'.format(mae3))
print('Root mean squared error of the model is {}.'.format(rmse3))
print('R-squared score is {}.'.format(r23))
```

- **Model**: Decision Tree Regressor
- **Train/Test Split**: 80% training, 20% testing
- **Random State**: 100

Stage 4: Model the Data

- **Model Evaluation**
  - **KNN**
    - Mean Absolute Error (MAE): 3.6878
    - Root Mean Squared Error (RMSE): 4.7102
    - R - squared: -1.2077

→

- Higher MAE and RMSE
- Lower R-squared
- Worse than Linear Regression Model

Stage 5: Communicate the Data

Stage 5: Communicate the Data

- **Conclusions**
  - Students’ weekday alcohol consumption and overall math grades are negatively correlated as we hypothesized (but very weak)
  - All three machine learning models tested (Linear Regression, Decision Tree Regressor, and KNN) are underfitted
  - Comparing the three models, of those three models, Linear Regression and Decision Tree Regressor have slightly better accuracy (MAE, RMSE, R-squared)
  - Weekday alcohol consumption is not a significant variable in predicting students’ school performance (overall math grade)

Stage 5: Communicate the Data

- **Potential Implications of this Project**
  - After presenting these findings to the school district, we would suggest focusing improvement efforts towards more student involvement
  - After school programs such as tutoring or sports clubs
    - Assuming the school is a drug-free zone, an after school program would allow the students to use the time they would've been drinking, to get extra help instead
  - Raise awareness about the issue
    - Alcohol education program; inform the students about long term effects of their actions.
    - Provide more resources to reach out for help. I.e. a counselor
  - Motivational speakers
    - Students are more than what they're told

Stage 5: Communicate the Data

- **Direct Benefit**
  - Student performance will increase, school district looks good
  - School district will be eligible for more funding
  - Continue to invest in students