Analysis of College Student Sleep Pattern

Evelyn Pan, Lois Liu, Samanyu Arora, YiduoWang, Zesheng Yin

1 Questions

1. Can we predict sleep quality based on lifestyle factors
2. Can we find the correlation between sleep quality and other features? Which feature has the most influential effect?
3. Can we visualize the sleep patterns over weekdays and weekends?

2 Limitations

The conclusion & limitation

Data Set

```
[9]: import pandas as pd
```

```
[11]: file_path = "C:/Users/Lois/Desktop/Stat 451/Homework/student_sleep_patterns.csv"
student_sleep_patterns = pd.read_csv(file_path)
student_sleep_patterns.head()
```

[11]:

	Student_ID	Age	Gender	University_Year	Sleep_Duration	Study_Hours	Screen_Time	Caffeine_Intake	Physical_Activity	Sleep_Quality	Weekday_Sleep_Start	Weel
0	1	24	Other	2nd Year	7.7	7.9	3.4	2	37	10	14.16	
1	2	21	Male	1st Year	6.3	6.0	1.9	5	74	2	8.73	
2	3	22	Male	4th Year	5.1	6.7	3.9	5	53	5	20.00	
3	4	24	Other	4th Year	6.3	8.6	2.8	4	55	9	19.82	
4	5	20	Male	4th Year	4.7	2.7	2.7	0	85	3	20.98	

```
[15]: student_sleep_patterns.describe()
```

[15]:

	Student_ID	Age	Sleep_Duration	Study_Hours	Screen_Time	Caffeine_Intake	Physical_Activity	Sleep_Quality	Weekday_Sleep_Start	Weekend_Sleep_Start
count	500.000000	500.00000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000
mean	250.500000	21.53600	6.472400	5.981600	2.525000	2.462000	62.342000	5.362000	11.166860	12.375860
std	144.481833	2.33315	1.485764	3.475725	0.859414	1.682325	35.191674	2.967249	5.972352	5.789611
min	1.000000	18.00000	4.000000	0.100000	1.000000	0.000000	0.000000	1.000000	1.080000	2.050000
25%	125.750000	20.00000	5.100000	2.900000	1.800000	1.000000	32.750000	3.000000	6.087500	7.297500

Categorical data:
Gender,
University_Year

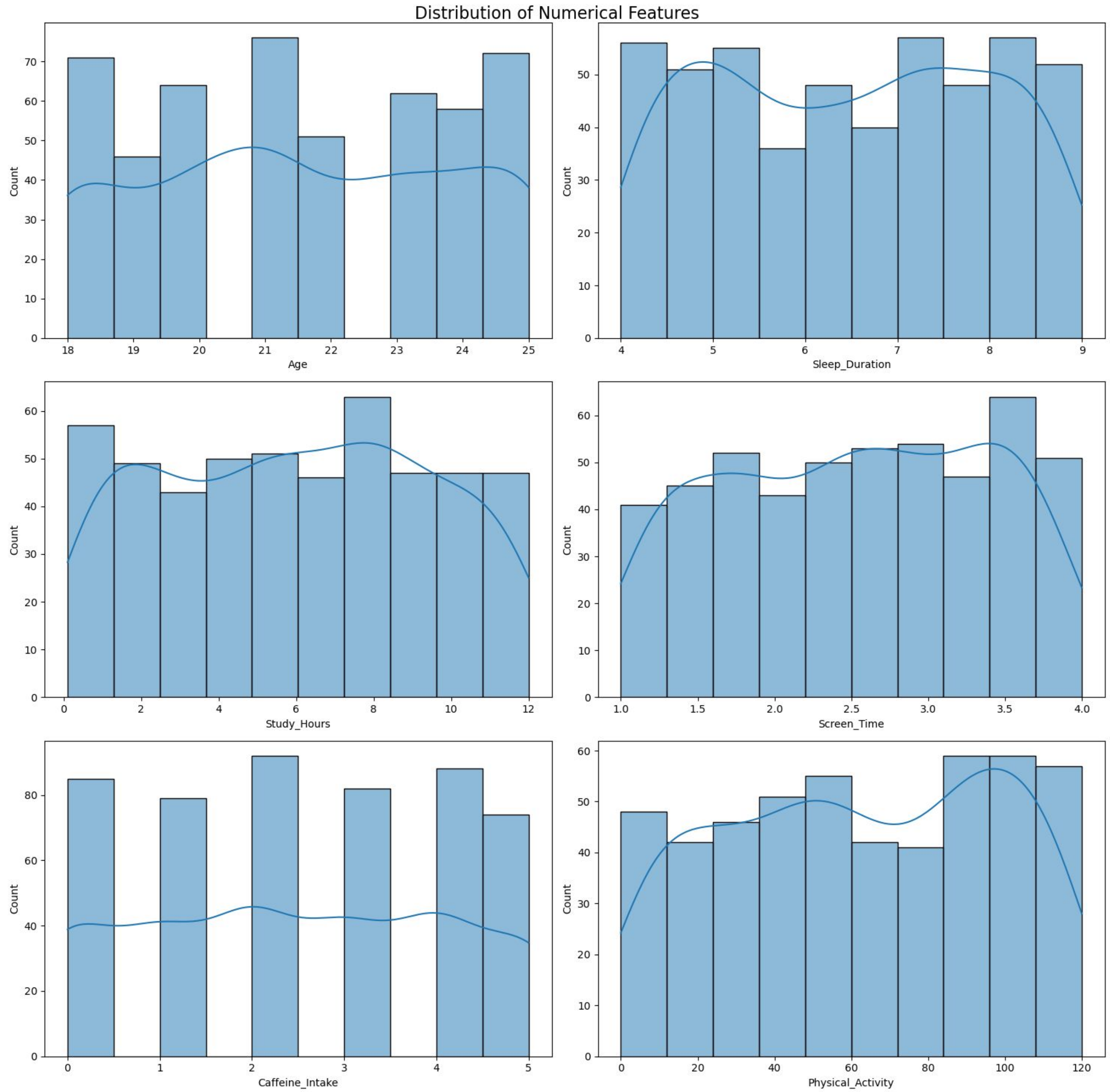
Prediction data:
Sleep_Duration,
Sleep_Quality

Lifestyle factors:
Study_Hours,
Screen_Time,
Physical_Activity.

```
Missing Values:
Student_ID           0
Age                  0
Gender               0
University_Year      0
Sleep_Duration       0
Study_Hours          0
Screen_Time          0
Caffeine_Intake      0
Physical_Activity    0
Sleep_Quality        0
dtype: int64

Data Types:
Student_ID             int64
Age                    int64
Gender                 int32
University_Year        int32
Sleep_Duration       float64
Study_Hours          float64
Screen_Time          float64
Caffeine_Intake        int64
Physical_Activity      int64
Sleep_Quality          int64
dtype: object
```

```
Descriptive Statistics:
       Student_ID        Age      Gender  University_Year  Sleep_Duration
count  500.000000  500.00000  500.000000       500.000000      500.000000
mean   250.500000   21.53600    0.964000         1.462000        6.472400
std    144.481833    2.33315    0.792439         1.094968        1.485764
min      1.000000   18.00000    0.000000         0.000000        4.000000
25%    125.750000   20.00000    0.000000         0.750000        5.100000
50%    250.500000   21.00000    1.000000         1.000000        6.500000
75%    375.250000   24.00000    2.000000         2.000000        7.800000
max    500.000000   25.00000    2.000000         3.000000        9.000000

       Study_Hours  Screen_Time  Caffeine_Intake  Physical_Activity  \
count   500.000000   500.000000       500.000000         500.000000
mean      5.981600     2.525000         2.462000          62.342000
std       3.475725     0.859414         1.682325          35.191674
min       0.100000     1.000000         0.000000           0.000000
25%       2.900000     1.800000         1.000000          32.750000
50%       6.050000     2.600000         2.000000          62.500000
75%       8.800000     3.300000         4.000000          93.250000
max      12.000000     4.000000         5.000000         120.000000

       Sleep_Quality
count     500.000000
mean        5.362000
std         2.967249
min         1.000000
25%         3.000000
50%         5.000000
75%         8.000000
max        10.000000
```

Plotting distribution of numerical features




```python
# Plotting distribution of numerical features
fig, axes = plt.subplots(3, 2, figsize=(15, 15))
sns.histplot(data['Age'], ax=axes[0, 0], kde=True)
sns.histplot(data['Sleep_Duration'], ax=axes[0, 1], kde=True)
sns.histplot(data['Study_Hours'], ax=axes[1, 0], kde=True)
sns.histplot(data['Screen_Time'], ax=axes[1, 1], kde=True)
sns.histplot(data['Caffeine_Intake'], ax=axes[2, 0], kde=True)
sns.histplot(data['Physical_Activity'], ax=axes[2, 1], kde=True)

fig.suptitle('Distribution of Numerical Features', fontsize=16)
plt.tight_layout()
plt.show()
```

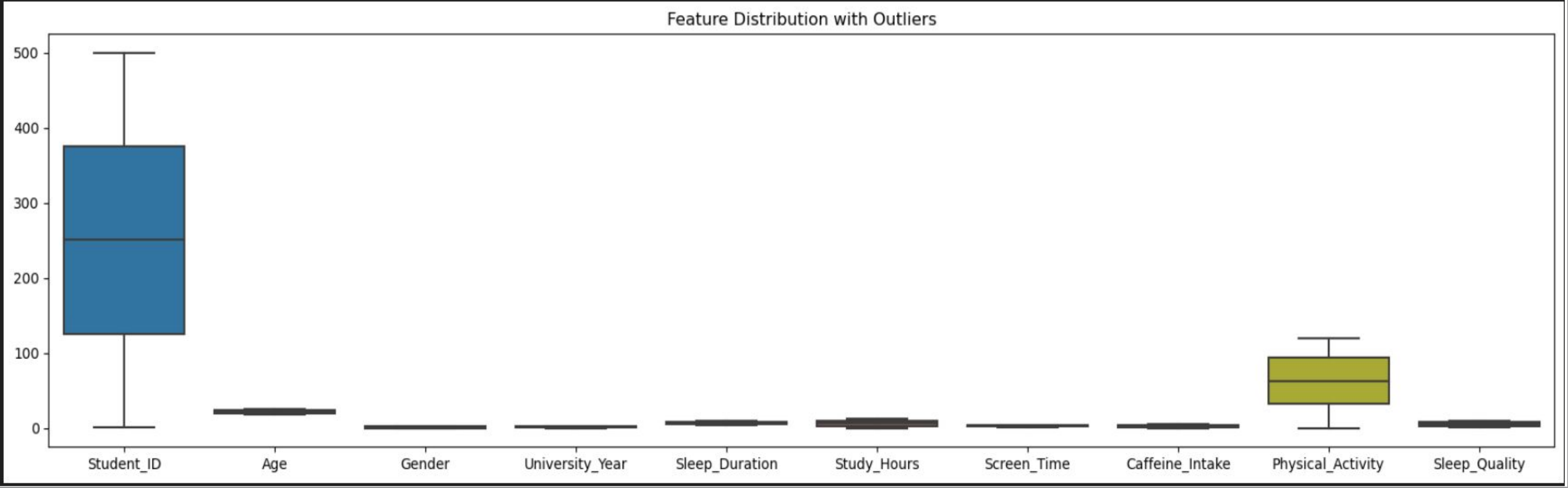
Pairplot of Key Variables

BoxPlot to identify outliers and for insights in order to create new features

```python
# boxplot to identify outliers
plt.figure(figsize=(20, 5))
sns.boxplot(data=data)
plt.title("Feature Distribution with Outliers")
plt.show()
```

[40]

Python



Linear Regression Model predicting sleep duraction based on the lifestyle factors

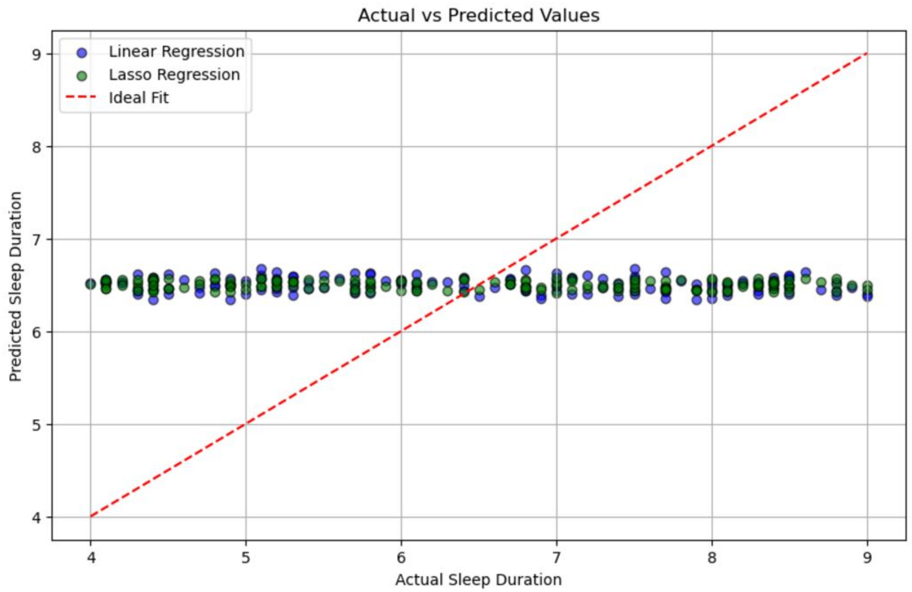
```
# Feature selection
X = data[['Study_Hours', 'Screen_Time', 'Physical_Activity']]
y = data['Sleep_Duration']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Training a Linear Regression Model
linear_model = LinearRegression()
linear_model.fit(X_train, y_train)
y_pred = linear_model.predict(X_test)

#results
print("Linear Regression R2 Score:", r2_score(y_test, y_pred))
print("Mean Squared Error:", mean_squared_error(y_test, y_pred))
```

```
Linear Regression R2 Score: -0.010234377553459462
Mean Squared Error: 2.2773686633604235
```

Regularization Method: Lasso Regression

```python
# Lasso Regression Model
lasso_model = Lasso(alpha=0.1, random_state=42)
lasso_model.fit(X_train, y_train)
y_pred_lasso = lasso_model.predict(X_test)

print("Lasso Regression R2 Score:", r2_score(y_test, y_pred_lasso))
print("Lasso Regression Mean Squared Error:", mean_squared_error(y_test, y_pred_lasso))

# Plotting
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred_lr, alpha=0.6, color='blue', label='Linear Regression', edgecolor='k')
plt.scatter(y_test, y_pred_lasso, alpha=0.6, color='green', label='Lasso Regression', edgecolor='k')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color='red', linestyle='--', label='Ideal Fit')

# Legend
plt.title('Actual vs Predicted Values')
plt.xlabel('Actual Sleep Duration')
plt.ylabel('Predicted Sleep Duration')
plt.legend()
plt.grid(True)
plt.show()
```

```
Lasso Regression R2 Score: -0.010989650960830044
Lasso Regression Mean Squared Error: 2.2790712741885963
```



kNN model predicting sleep duration based on lifestyle factors

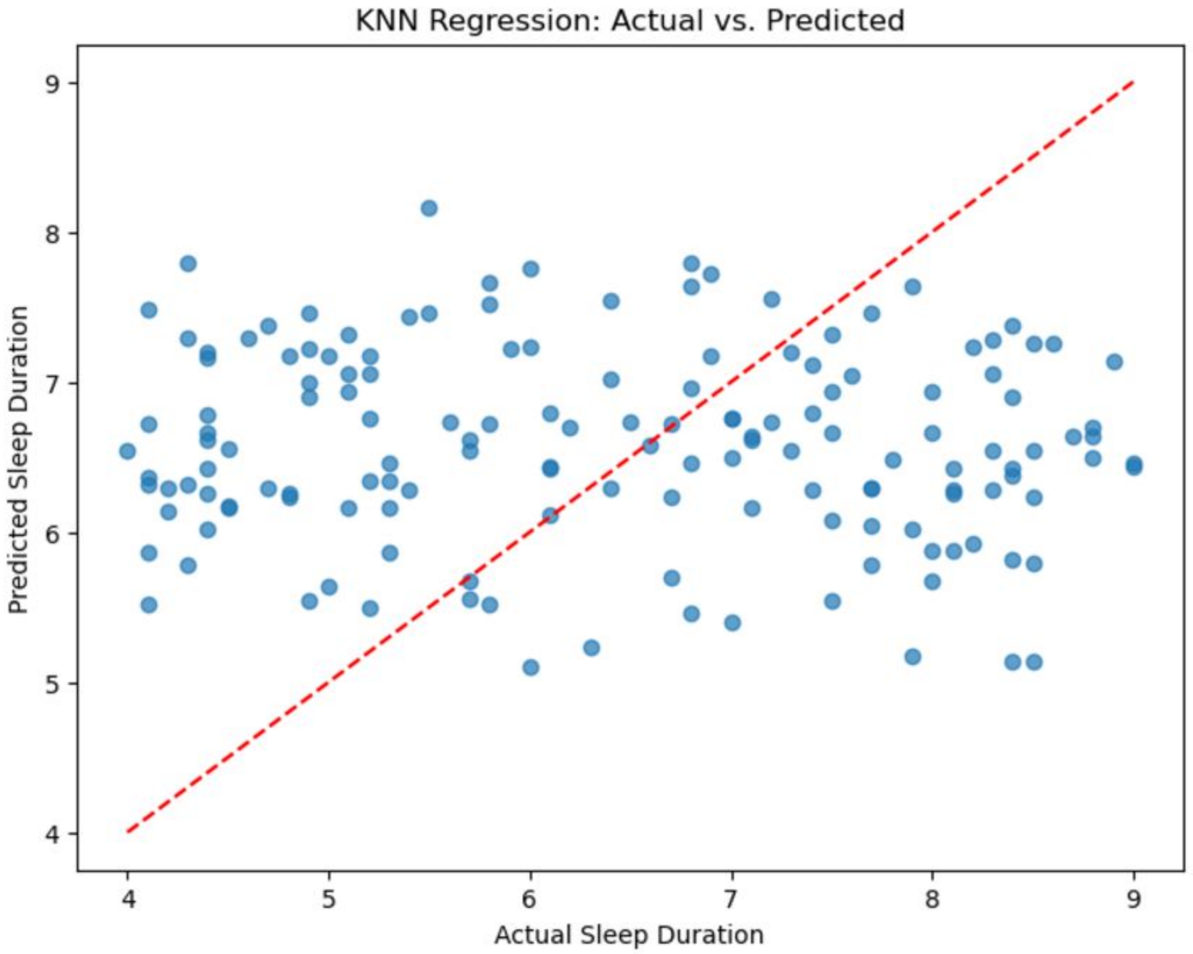
predict outcomes based on the similarity of features → better for a nonlinear relationship?

```
X = sleep_data[['Study_Hours', 'Screen_Time', 'Physical_Activity']]
y = sleep_data['Sleep_Duration']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# Train a KNN Regressor
knn_model = KNeighborsRegressor(n_neighbors=5)
knn_model.fit(X_train, y_train)
y_pred = knn_model.predict(X_test)
knn_r2 = r2_score(y_test, y_pred)
knn_mse = mean_squared_error(y_test, y_pred)

#results
print("KNN R-squared Score:", knn_r2)
print("Mean Squared Error:", knn_mse)
```

```
KNN R-squared Score: -0.25607092350432326
Mean Squared Error: 2.831557333333333
```

still a negative R-squared, which means the kNN model are still not effectivey capturing the relationship with Sleep_Duration

```python
# Scatter plot of actual vs. predicted values
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, alpha=0.7)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel("Actual Sleep Duration")
plt.ylabel("Predicted Sleep Duration")
plt.title("KNN Regression: Actual vs. Predicted")
plt.show()
```

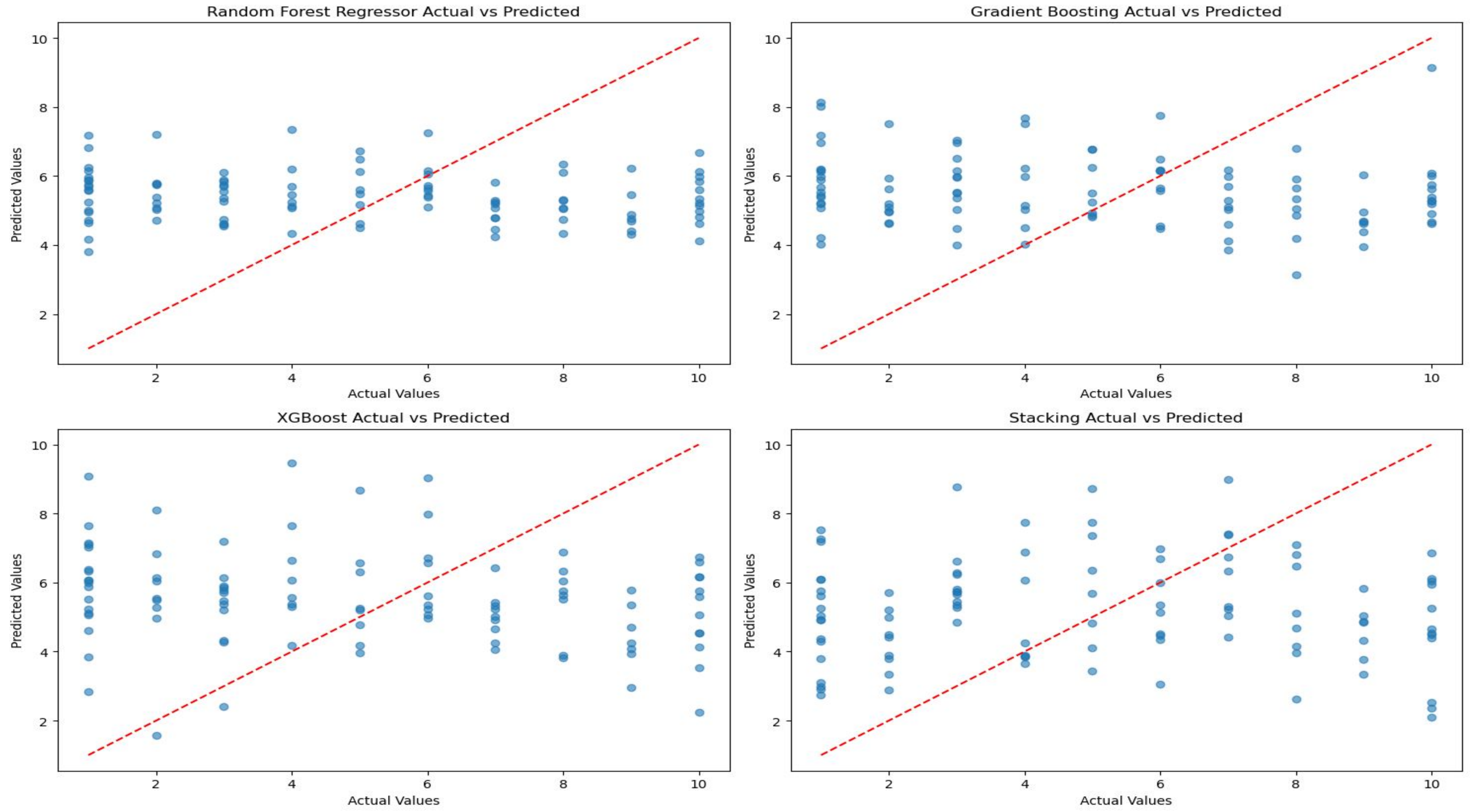


The red line represents the perfect prediction and points lying on or close to the line indicate accurate prediction. Since our graphs showed that most points are scattered away from red line and very discrete, indicating that the kNN model is struggling to make accurate predictions.

```python
# Random Forest Regressor
rf_model = RandomForestRegressor(n_estimators=200, max_depth=10, random_state=42)
rf_model.fit(X_train_reg_scaled, y_train_reg)
rf_pred = rf_model.predict(X_test_reg_scaled)

# Gradient Boosting Regressor
gb_model = GradientBoostingRegressor(n_estimators=200, learning_rate=0.05, random_state=42)
gb_model.fit(X_train_reg_scaled, y_train_reg)
gb_pred = gb_model.predict(X_test_reg_scaled)

# XGBoost Regressor
xgb_model = XGBRegressor(n_estimators=200, learning_rate=0.05, max_depth=6, random_state=42)
xgb_model.fit(X_train_reg_scaled, y_train_reg)
xgb_pred = xgb_model.predict(X_test_reg_scaled)

# Stacking Regressor
stacking_model = StackingRegressor(
    estimators=[('rf', rf_model), ('gb', gb_model)],
    final_estimator=xgb_model
)
stacking_model.fit(X_train_reg_scaled, y_train_reg)
stack_pred = stacking_model.predict(X_test_reg_scaled)

# Evaluation for regression models
models = ['Random Forest Regressor', 'Gradient Boosting', 'XGBoost', 'Stacking']
predictions = [rf_pred, gb_pred, xgb_pred, stack_pred]

for model, pred in zip(models, predictions):
    print(f"{model} R^2 Score:", r2_score(y_test_reg, pred))
    print(f"{model} Mean Squared Error:", mean_squared_error(y_test_reg, pred))
```

Random Forest, Gradient Boositng, XGBoost, Ensemble regressors

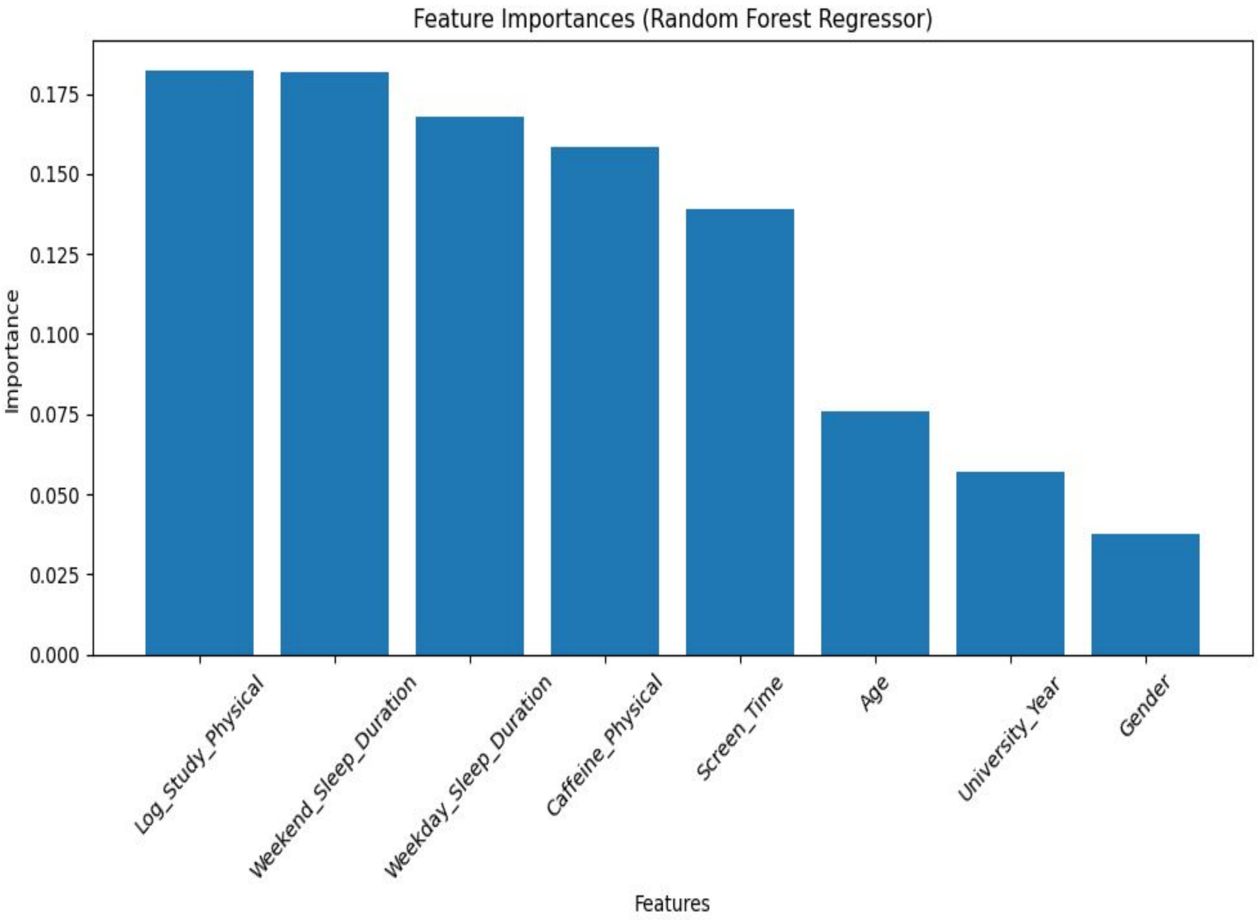
```
Random Forest Regressor R^2 Score: -0.13452194278199614
Random Forest Regressor Mean Squared Error: 10.816078393706439
Gradient Boosting R^2 Score: -0.258380725419894
Gradient Boosting Mean Squared Error: 11.9968984838631
XGBoost R^2 Score: -0.41392195224761963
XGBoost Mean Squared Error: 13.479768382562083
Stacking R^2 Score: -0.25733721256256104
Stacking Mean Squared Error: 11.98695090218868
```

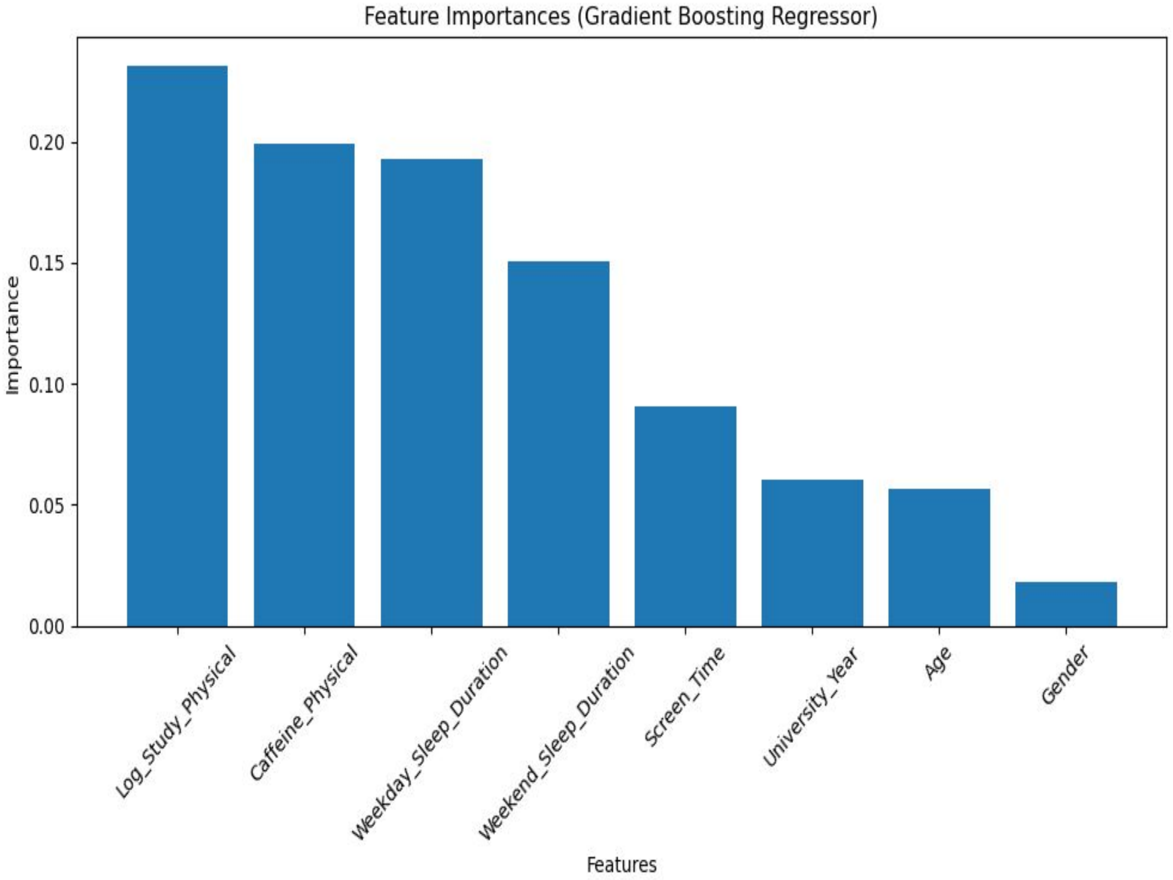


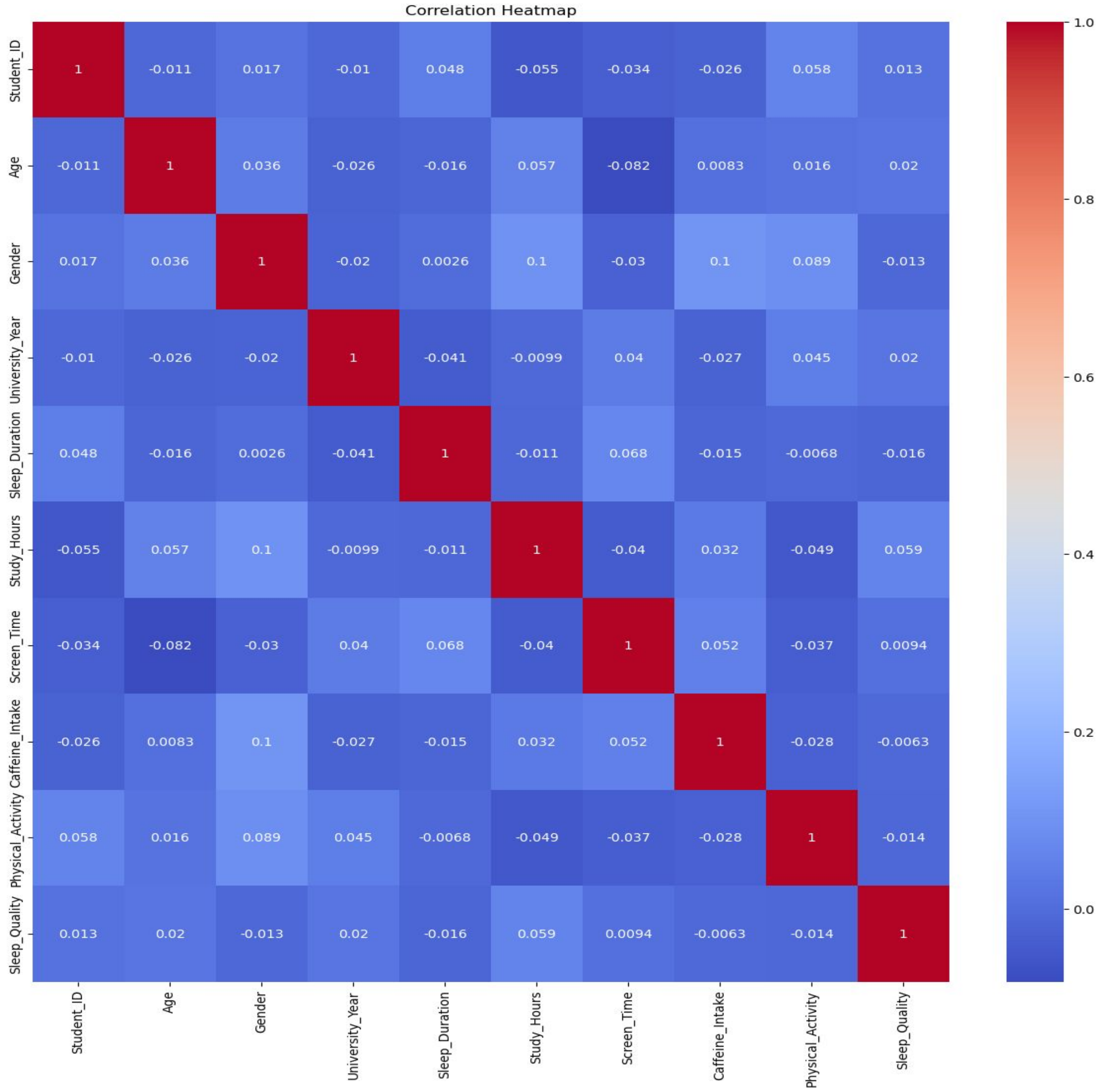
Random Forest Regressor Actual vs Predicted
Gradient Boosting Actual vs Predicted
XGBoost Actual vs Predicted
Stacking Actual vs Predicted
Predicted Values
Actual Values
2
4
6
8
10

Feature Comparison for two models

From these graphs we can see that the top 3 features study_physical, weekend sleep duration and caffiene_physical have the highest importance, so in order to provide insights regarding improving sleep cycle we should focus on these factors







Correlation Matrix

Based on the previous models insights we can feature engineer and use random forest classifier for our Final prediction

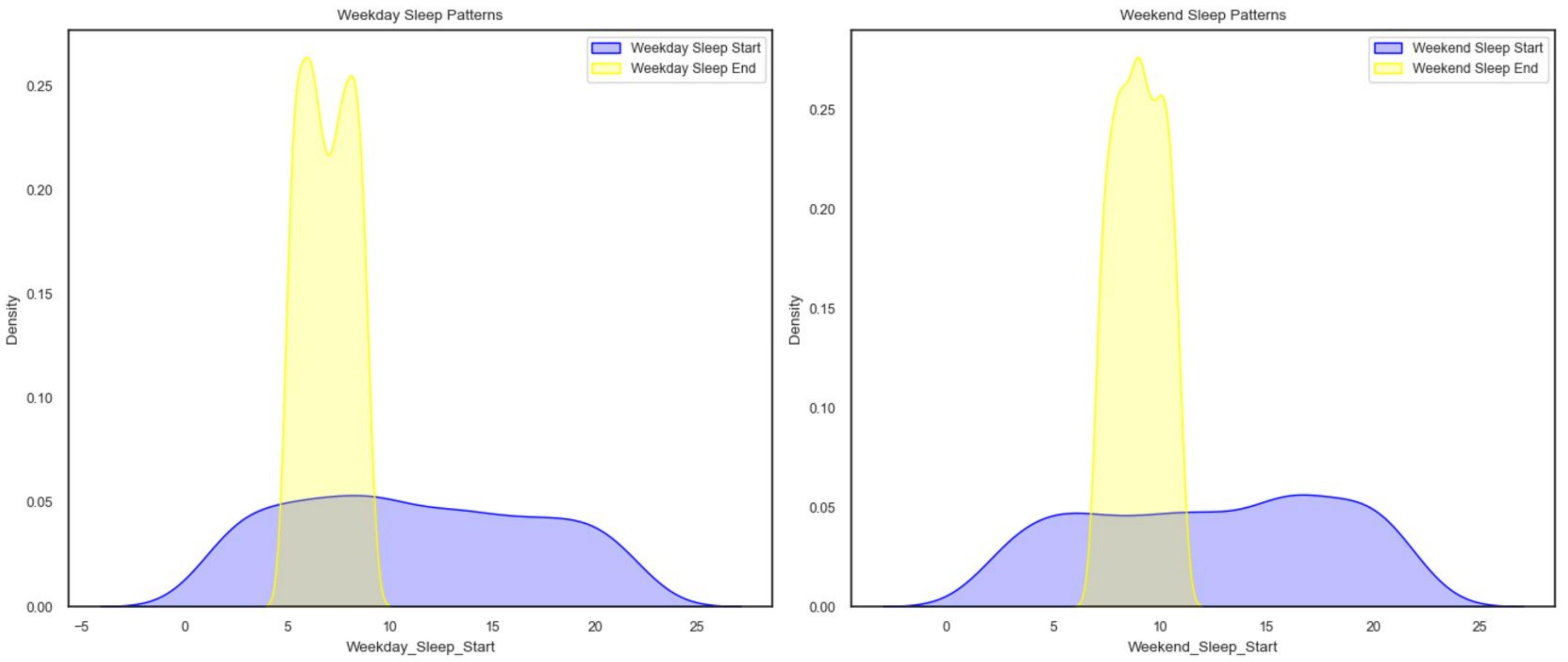
```
# Splitting data into features and target for RandomForestClassifier
x = data.drop('Sleep_Quality', axis=1)
y = data['Sleep_Quality']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

# Feature scaling (for RandomForestClassifier)
scaler = StandardScaler()
X_train = scaler.fit_transform(x_train)
X_test = scaler.transform(x_test)

# Model training (RandomForestClassifier)
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Predictions and evaluation (RandomForestClassifier)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.08

Classification Report:
              precision    recall  f1-score   support

           1       0.14      0.17      0.15        18
           2       0.14      0.22      0.17         9
           3       0.14      0.17      0.15        12
           4       0.00      0.00      0.00         8
           5       0.00      0.00      0.00         8
           6       0.00      0.00      0.00         9
           7       0.17      0.11      0.13         9
           8       0.00      0.00      0.00         8
           9       0.00      0.00      0.00         7
          10       0.00      0.00      0.00        12

    accuracy                           0.08       100
   macro avg       0.06      0.07      0.06       100
weighted avg       0.07      0.08      0.07       100
```

Analysis and Visualization of sleep pattern difference in weekdays and weekends.

```
fig, axes = plt.subplots(1, 2, figsize=(14, 6))
# Weekday sleep patterns
sns.kdeplot(data['Weekday_Sleep_Start'], ax=axes[0], shade=True, label='Weekday Sleep Start', color='blue')
sns.kdeplot(data['Weekday_Sleep_End'], ax=axes[0], shade=True, label='Weekday Sleep End', color='yellow')
axes[0].set_title('Weekday Sleep Patterns')
axes[0].legend()
# Weekend sleep patterns
sns.kdeplot(data['Weekend_Sleep_Start'], ax=axes[1], shade=True, label='Weekend Sleep Start', color='blue')
sns.kdeplot(data['Weekend_Sleep_End'], ax=axes[1], shade=True, label='Weekend Sleep End', color='yellow')
axes[1].set_title('Weekend Sleep Patterns')
axes[1].legend()
plt.tight_layout()
plt.show()
```

Use Kernel Density Estimation plots to visualize the distribution.
KDE plots show the density of values to identify differences.



The distribution for weekday_sleep_start is spread over a slightly broader range, suggesting more variability in when students go to sleep on weekdays than weekends.
The distribution for weekday_sleep_end has two sharper peak than weekend, showing that most students wake up around the same time
The gap between weekend_sleep_start and weekend_sleep_end is wider, indicating longer sleep duration fpr more students on weekends than weekdays.
Overall, students tend to maintain a stricter sleep scheduleand sleep duration appears shorter on weekdays

Limitation of Our Project

- The selected features (lifestyle factors) have a weak or no significant correlation with the target variable (Sleep_Qaulity and Sleep_Duration), making it difficult for any model to perform well since the R-squared is very low and even negative.

- theis happens because of several reasons:
  - the dataset is still consider small (only 500)
  - most of college students have the age range between 18-24, who are young and healthy without that many sleeping disorders. The study hours/caffeine intake/physical activity/screen time hours will not affect their sleep that much
  - the data range is relatively small and most of them are rounded to hours in whole number. There is not that many apparent correlation between each features in hour scale.
  - dataset lacks of other crucial features such as “stress_levels”, “envrionments”,”dietary habits” which are more likely to influence college students sleep patterns.