```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score


df = pd.read_csv('/content/student-mat.csv', delimiter=';')
df.head()
```

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

5 rows × 33 columns

```
print("Shape:", df.shape)
print(df.info())
print(df.describe())
df.head()
```

```
Shape: (395, 33)
<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
 7
 8   Mjob        395 non-null    object
 9   Fjob        395 non-null    object
 10  reason      395 non-null    object
 11  guardian    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
```

	age	Medu	Fedu	traveltime	studytime	failure
count	395.000000	395.000000	395.000000	395.000000	395.000000	395.00000
mean	16.696203	2.749367	2.521519	1.448101	2.035443	0.33417
std	1.276043	1.094735	1.088201	0.697505	0.839240	0.74365
min	15.000000	0.000000	0.000000	1.000000	1.000000	0.00000
25%	16.000000	2.000000	2.000000	1.000000	1.000000	0.00000
50%	17.000000	3.000000	2.000000	1.000000	2.000000	0.00000
75%	18.000000	4.000000	3.000000	2.000000	2.000000	0.00000
max	22.000000	4.000000	4.000000	4.000000	4.000000	3.00000

	famrel	freetime	goout	Dalc	Walc	healt
count	395.000000	395.000000	395.000000	395.000000	395.000000	395.00000
mean	3.944304	3.235443	3.108861	1.481013	2.291139	3.55443
std	0.896659	0.998862	1.113278	0.890741	1.287897	1.39030
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.00000
25%	4.000000	3.000000	2.000000	1.000000	1.000000	3.00000
50%	4.000000	3.000000	3.000000	1.000000	2.000000	4.00000
75%	5.000000	4.000000	4.000000	2.000000	3.000000	5.00000
max	5.000000	5.000000	5.000000	5.000000	5.000000	5.00000

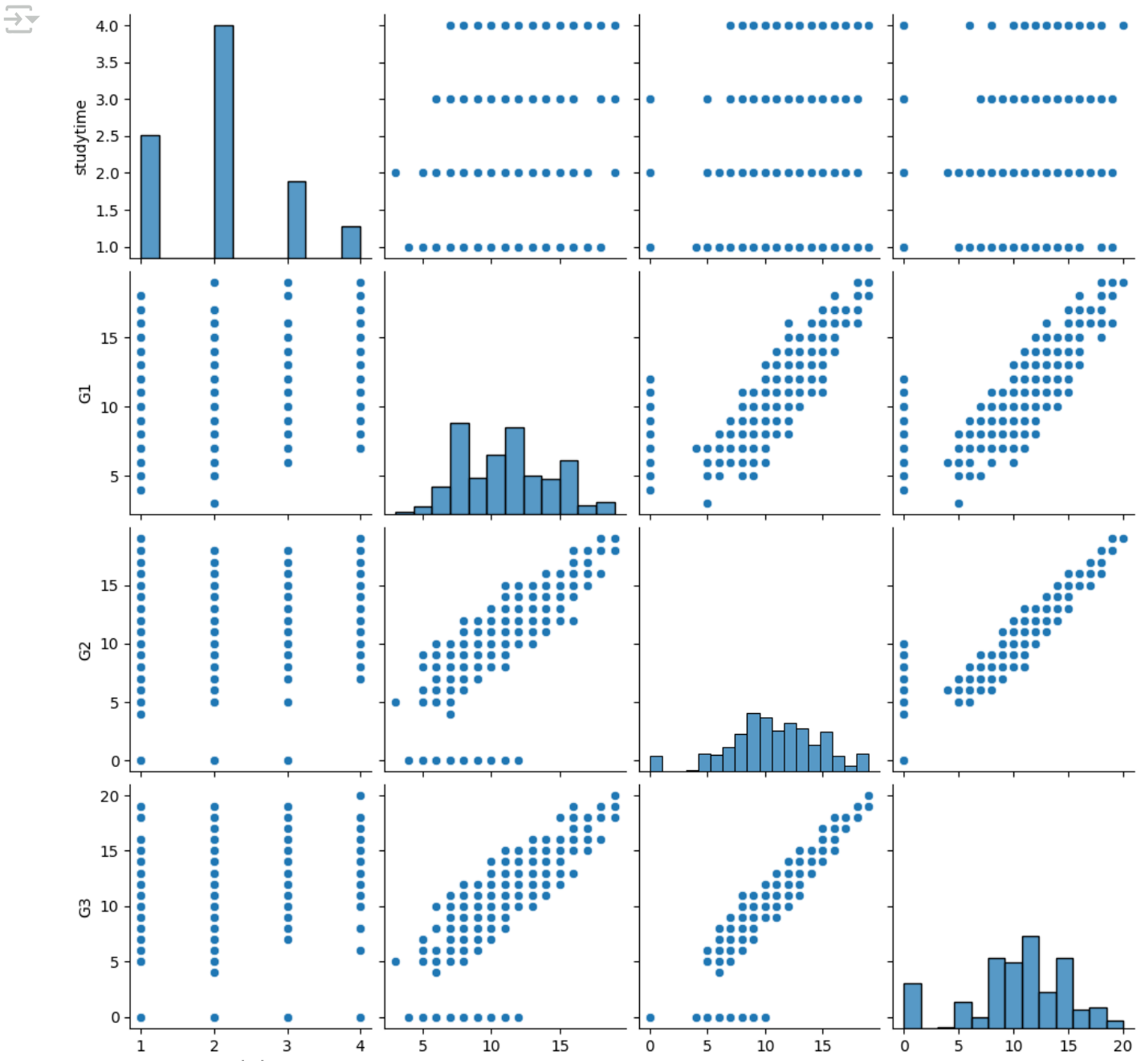
	absences	G1	G2	G3
count	395.000000	395.000000	395.000000	395.000000
mean	5.708861	10.908861	10.713924	10.415190
std	8.003096	3.319195	3.761505	4.581443
min	0.000000	3.000000	0.000000	0.000000
25%	0.000000	8.000000	9.000000	8.000000
50%	4.000000	11.000000	11.000000	11.000000
75%	8.000000	13.000000	13.000000	14.000000
max	75.000000	19.000000	19.000000	20.000000

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	..

0	GP	F	18	U	GT3	A	4	4	at_home	teacher	..
1	GP	F	17	U	GT3	T	1	1	at_home	other	..
2	GP	F	15	U	LE3	T	1	1	at_home	other	..
3	GP	F	15	U	GT3	T	4	2	health	services	..
4	GP	F	16	U	GT3	T	3	3	other	other	..

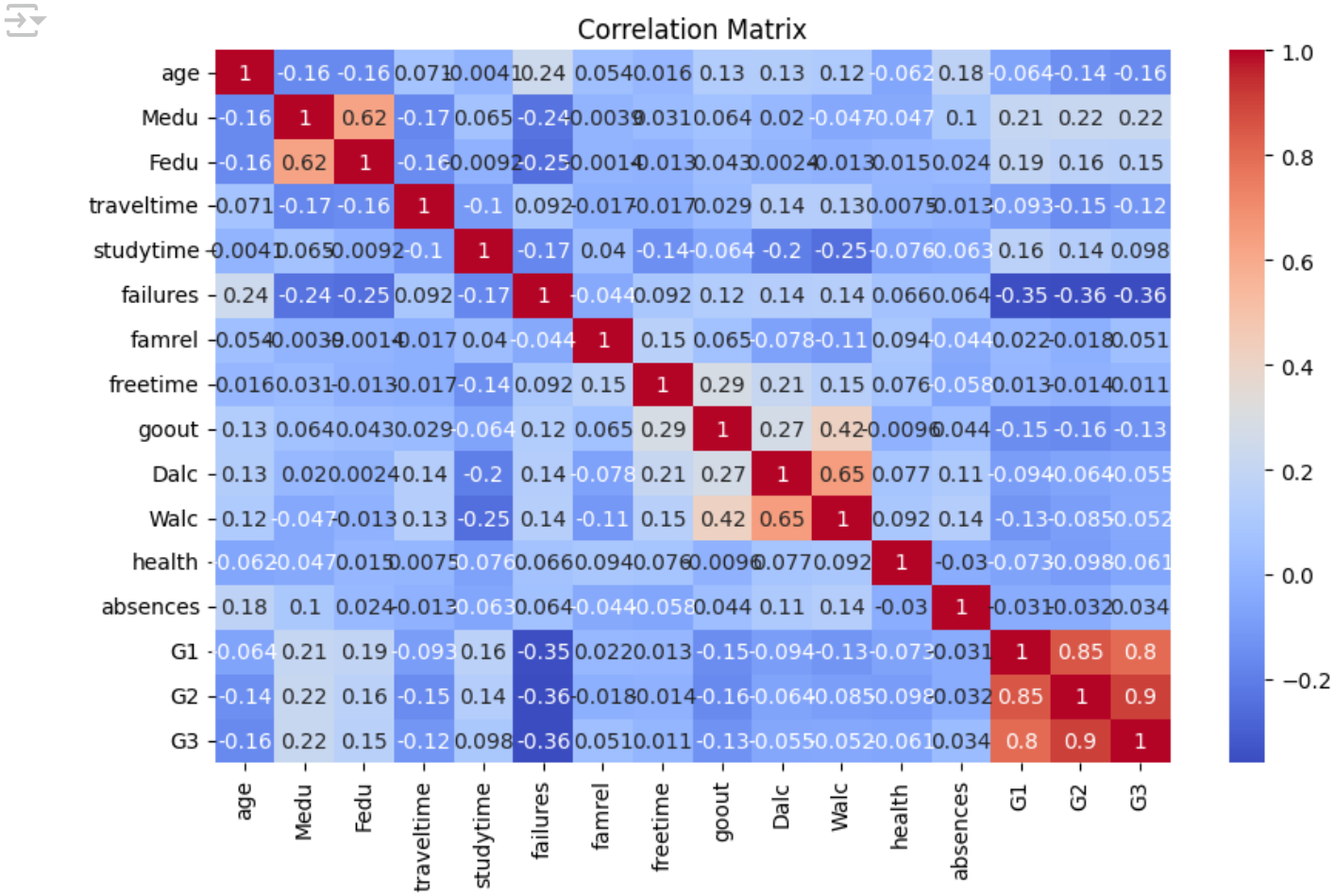
5 rows × 33 columns

```
sns.pairplot(df[['studytime', 'G1', 'G2', 'G3']])
plt.show()
```



	studytime	G1	G2	G3

```
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()
```




```
features = ['studytime', 'failures', 'absences', 'G1', 'G2']
target = 'G3'

X = df[features]
y = df[target]


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
```

```
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
y_pred_lr = lr_model.predict(X_test)
```

```
print("R² Score:", r2_score(y_test, y_pred_lr))
print("MSE:", mean_squared_error(y_test, y_pred_lr))
```

```
R² Score: 0.7821754247320557
MSE: 4.466503212015601
```

```
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
```

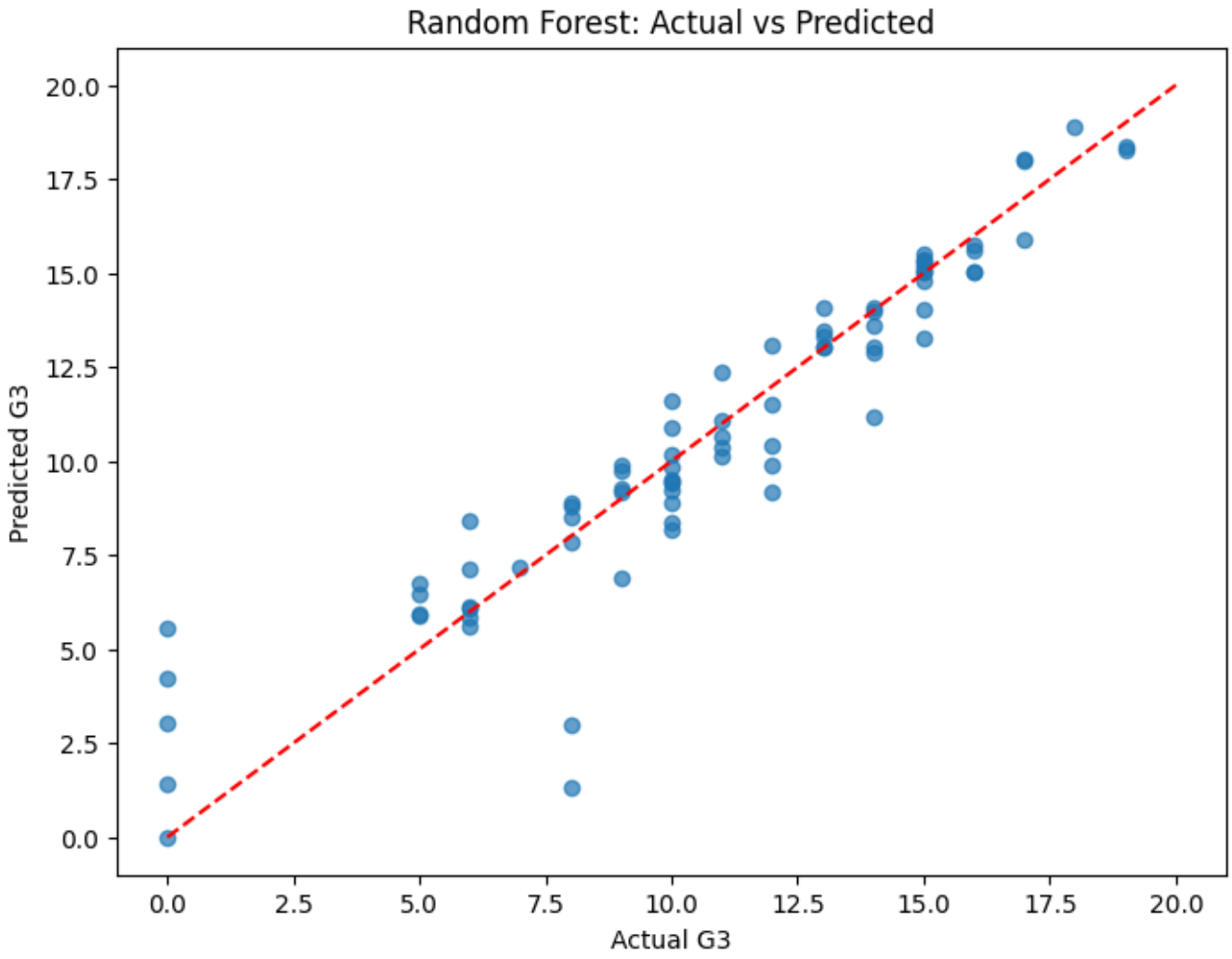
```
▾ RandomForestRegressor
RandomForestRegressor(random_state=42)
```

```
y_pred_rf = rf_model.predict(X_test)
```

```
print("R² Score:", r2_score(y_test, y_pred_rf))
print("MSE:", mean_squared_error(y_test, y_pred_rf))
```

```
R² Score: 0.8725239455080609
MSE: 2.61390252290377
```

```
plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred_rf, alpha=0.7)
plt.xlabel("Actual G3")
plt.ylabel("Predicted G3")
plt.title("Random Forest: Actual vs Predicted")
plt.plot([0, 20], [0, 20], 'r--')
plt.show()
```




```
results = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred_rf})
results.to_csv('prediction_results.csv', index=False)
```

```
# For Linear Regression
coef_df = pd.DataFrame({
    'Feature': X.columns,
    'Coefficient': lr_model.coef_
})
print(coef_df)

# For Random Forest
importances = pd.DataFrame({
    'Feature': X.columns,
    'Importance': rf_model.feature_importances_
}).sort_values(by='Importance', ascending=False)
print(importances)
```

```
      Feature  Coefficient
0  studytime    -0.071231
1   failures    -0.455813
2   absences     0.039245
3         G1     0.144463
4         G2     0.979615
     Feature  Importance
4         G2    0.813853
2   absences    0.119140
3         G1    0.031818
0  studytime    0.020381
1   failures    0.014807
```

Start coding or generate with AI.