```
"""
مشروع التنبؤ بالمعدل التراكمي (GPA) للطلاب

🔹 الهدف: بناء نموذج تعلم آلي لتوقع معدل GPA للطلاب بناءً على عوامل مختلفة.
🔹 الأدوات المستخدمة:
   - تحليل البيانات باستخدام Pandas
   - التصوير البياني باستخدام Seaborn و Matplotlib
   - بناء نماذج تعلم آلي باستخدام Scikit-learn
🔹 خطوات المشروع:
   1️⃣ تحميل البيانات ومعالجتها
   2️⃣ تحليل البيانات بصريًا
   3️⃣ تقسيم البيانات إلى تدريب واختبار
   4️⃣ تجربة أكثر من نموذج (Linear Regression, Decision Tree, Random Forest)
   5️⃣ مقارنة أداء النماذج واختيار الأفضل
   6️⃣ تنفيذ نموذج نهائي وتجربة التوقعات على بيانات جديدة
"""

import pandas as pd
from google.colab import files

uploaded = files.upload()  # رفع الملف
filename = list(uploaded.keys())[0]  # الحصول على اسم الملف المرفوع

df = pd.read_csv(filename)  # قراءة الملف مباشرة

# حذف الأعمدة غير المهمة
df.drop(columns=["StudentID", "GradeClass"], inplace=True)

# التحقق من البيانات المفقودة
missing_values = df.isnull().sum()
print("\n🔍 القيم المفقودة في كل عمود:\n", missing_values)
```

```python
# تعويض القيم المفقودة بالمتوسط
df.fillna(df.mean(), inplace=True)

# عرض أول 5 صفوف للتأكد من صحة البيانات بعد المعالجة
print("\n📊 نظرة سريعة على البيانات بعد المعالجة:\n", df.head())

import matplotlib.pyplot as plt
import seaborn as sns

# تحليل العلاقة بين ساعات الدراسة و GPA
plt.figure(figsize=(8,6))
sns.regplot(x=df['StudyTimeWeekly'], y=df['GPA'], scatter_kws={"color": "blue"}, line_kws={"color": "red"})
plt.xlabel('Number of study hours per week', fontsize=12)
plt.ylabel(' Cumulative (GPA)', fontsize=12)
plt.title('📚 The relationship between the number of hours and the GPA', fontsize=14, fontweight='bold')
plt.show()

from sklearn.model_selection import train_test_split

# تحديد المدخلات (X) والمخرجات (y)
X = df.drop(columns=["GPA"])  # جميع الميزات ما عدا GPA
y = df["GPA"]  # القيمة التي نريد التنبؤ بها

# تقسيم البيانات إلى 80% تدريب و 20% اختبار
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# التحقق من حجم البيانات بعد التقسيم
print("\n📌 عدد العينات في بيانات التدريب:", X_train.shape[0])
print("📌 عدد العينات في بيانات الاختبار:", X_test.shape[0])
```

```python
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# تجربة عدة نماذج
models = {
    "Linear Regression": LinearRegression(),
    "Decision Tree": DecisionTreeRegressor(),
    "Random Forest": RandomForestRegressor()
}

# تخزين النتائج
results = {}

for name, model in models.items():
    model.fit(X_train, y_train)  # تدريب النموذج
    y_pred = model.predict(X_test)  # التنبؤ بالنتائج

    # حساب الأخطاء ودقة النموذج
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    results[name] = {"MSE": mse, "R² Score": r2}
    print(f" 🔹 {name} - MSE: {mse:.4f}, R² Score: {r2:.4f}")

# اختيار النموذج الأفضل بناءً على R² Score
best_model = max(results, key=lambda k: results[k]["R² Score"])
print(f"\n🏆 النموذج الأفضل هو: {best_model} بدقة R² Score = {results[best_model]['R² Score']:.4f}")
```

```python
if best_model == "Random Forest":
    # تدريب النموذج الأفضل
    best_model_instance = RandomForestRegressor()
    best_model_instance.fit(X_train, y_train)

    # حساب أهمية كل ميزة
    importances = best_model_instance.feature_importances_
    features = X.columns

    # رسم أهمية الميزات
    plt.figure(figsize=(10,6))
    sns.barplot(x=importances, y=features, palette="coolwarm")
    plt.xlabel("مدى التأثير على التوقع")
    plt.ylabel("الميزة")
    plt.title("🔍 تحليل أهمية الميزات في التنبؤ بـ GPA")
    plt.show()

# بيانات افتراضية لطالب جديد
new_student = [[18, 1, 2, 15, 3, 1, 2, 1, 0, 1, 0, 0]]

# التوقع باستخدام النموذج الأفضل
if best_model == "Linear Regression":
    final_model = LinearRegression()
elif best_model == "Decision Tree":
    final_model = DecisionTreeRegressor()
else:
    final_model = RandomForestRegressor()

# تدريب النموذج النهائي
final_model.fit(X_train, y_train)
```

```python
# التنبؤ بدرجة الطالب الجديد
predicted_gpa = final_model.predict(new_student)
print(f"\n🎯 التوقع للمعدل التراكمي (GPA) للطالب الجديد: {predicted_gpa[0]:.2f}")
```

Choose Files Student_pe...ata _(1).csv

- **Student_performance_data _(1).csv**(text/csv) - 166901 bytes, last modified: 6/12/2024 - 100% done

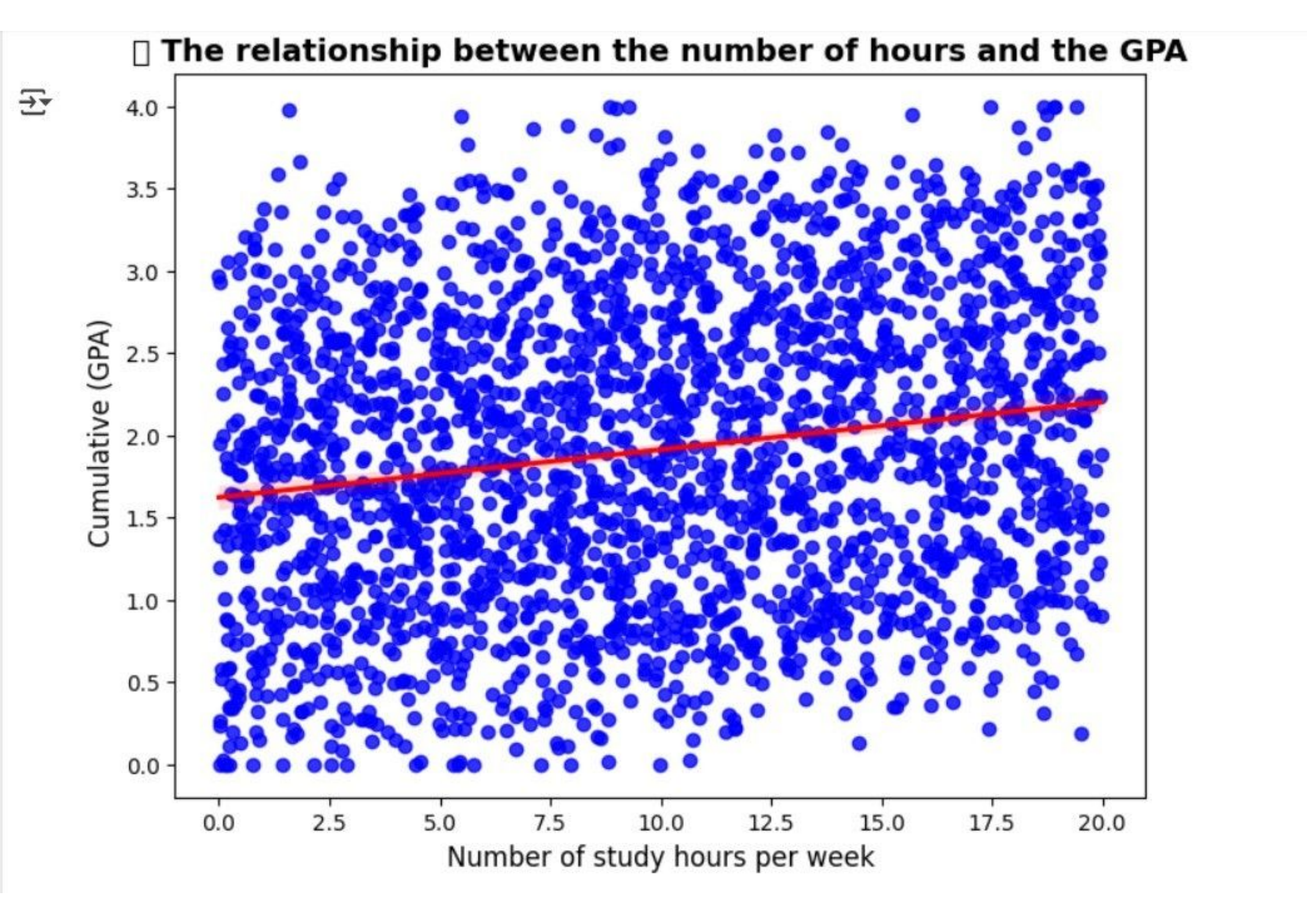
```
Saving Student_performance_data _(1).csv to Student_performance_data _(1) (2).csv

🔍 القيم المفقودة في كل عمود:
 Age                 0
Gender               0
Ethnicity            0
ParentalEducation    0
StudyTimeWeekly      0
Absences             0
Tutoring             0
ParentalSupport      0
Extracurricular      0
Sports               0
Music                0
Volunteering         0
GPA                  0
dtype: int64

📊 نظرة سريعة على البيانات بعد المعالجة:
   Age  Gender  Ethnicity  ParentalEducation  StudyTimeWeekly  Absences  \
0   17       1          0                  2        19.833723         7
1   18       0          0                  1        15.408756         0
2   15       0          2                  3         4.210570        26
3   17       1          0                  3        10.028829        14
4   17       1          0                  2         4.672495        17
```

```
   Tutoring  ParentalSupport  Extracurricular  Sports  Music  Volunteering  \
0         1                2                0       0      1             0
1         0                1                0       0      0             0
2         0                2                0       0      0             0
3         0                3                1       0      0             0
4         1                3                0       0      0             0

        GPA
0  2.929196
1  3.042915
2  0.112602
3  2.054218
4  1.288061
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 128218 (\N{BOOKS}) missing from font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```


The relationship between the number of hours and the GPA
Cumulative (GPA)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Number of study hours per week

Number of study hours per week

📌 عدد العينات في بيانات التدريب: 1913
📌 عدد العينات في بيانات الاختبار: 479
🔹 Linear Regression - MSE: 0.0387, R² Score: 0.9532
🔹 Decision Tree - MSE: 0.1051, R² Score: 0.8729
🔹 Random Forest - MSE: 0.0595, R² Score: 0.9281

🏆 النموذج الأفضل هو: Linear Regression بدقة R² Score = 0.9532

🎯 التوقع للمعدل التراكمي (GPA) للطالب الجديد: 3.36
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(