Impact of Lifestyle Factors on Student Performance

August 24, 2025

Name:- Zunaira Asif

This project studies how lifestyle & demographic factors affect students' final Grades(G3) using Linear Regression.

The dataset includes age,family, background, study time,health, alcohol & consumption etc.

Notebook Covers:-

1-Data exploration & Cleaning

2-Encoding Categorical variables

3-Feature scaling

4-Train_Test_Split

5-Linear Regression Modeling

6-Model Training

7-Model Prediction

8-Model Evaluation (MSE & R2)

9- Visualization of actual & predicted grades(G3) & residuals

It's beginner friendly example ,well_explained demonstration showing which factor influence student's performance in python.

Each step is clearly explained for easy understanding , from data exploration and cleaning to modeling , evaluation and visualization!!!

Results::- MSE 0.24 | R2 0.78 → Model predicts students' Final Grades reasonably well!

[3]:

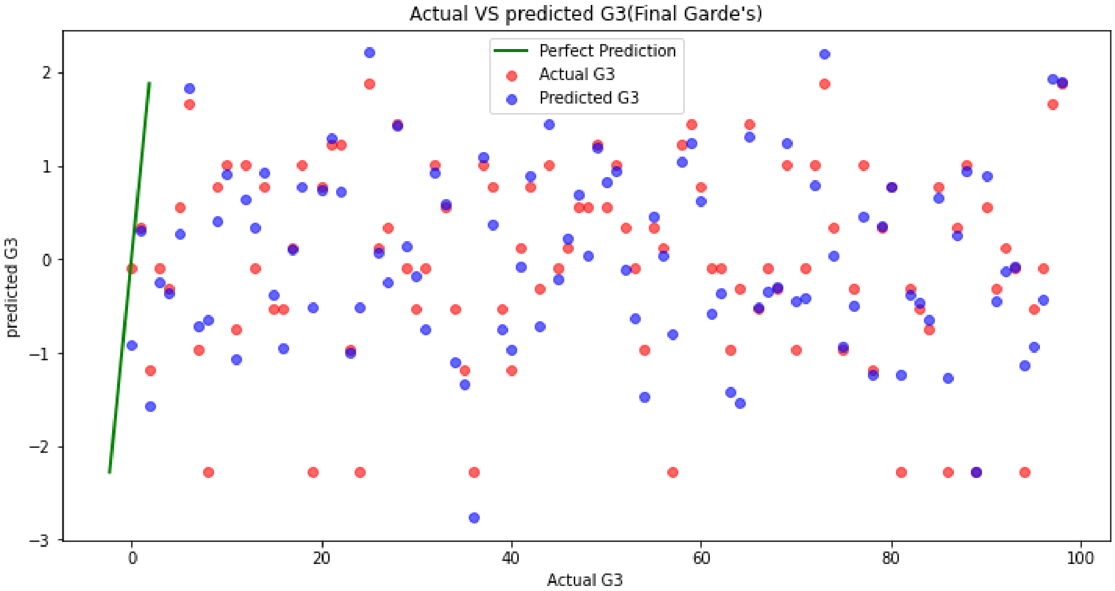
```
# Impact of Lifestyle Factors on Students' Performance model
import pandas as pd
import zipfile
# Load directly from zip( it's a compressed folder)into dataframe!
with zipfile.ZipFile("student.zip", "r") as z:
    # List all files inside the zip!
    print("Files in zip:", z.namelist())
    # Now we'll open the file we want (example: student-mat.csv)!
    with z.open("student-mat.csv") as f:
        # Now Dataframe!
        df = pd.read_csv(f, sep=";")
# It Shows first 5 rows by default!
print (df.head())
#Shape of data( how many rows and columns it has)
df.shape
# it shows columns names
df.columns
# General info (datatypes, missing values)
df.info()
# Summary statistics for numeric columns
df.describe()
# Check for missing values!
df.isnull().sum()
# We'll chk how many numeric & categorical columns!
cat_col=df.select_dtypes(include="object").columns
num_col=df.select_dtypes(exclude="object").columns
print ("Categorical Columns:",cat_col)
print ("Numerical Columns:",num_col)
# Check for missing values
print("\nMissing values:\n", df.isnull)
# Display the full DataFrame
from IPython.display import display
display(df)
#our data has no null values so,now we'll Encode categorical data!
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
#Label Encoding!
le = LabelEncoder()
df_label = df.copy()
for col in cat_col:
    df_label[col] = le.fit_transform(df_label[col])
#One Hot Encoding!
ohe=OneHotEncoder(sparse=False,drop='first')  # drop='first' avoids dummy variable trap
df_onehot_array = ohe.fit_transform(df[cat_col])
df_onehot=pd.DataFrame(df_onehot_array, columns=ohe.get_feature_names(cat_col))
# Combine with numerical columns
df_onehot = pd.concat([df[num_col].reset_index(drop=True), df_onehot], axis=1)
df_onehot.head(10)
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
#Only Scale numerical features
scaler_featured=scaler.fit_transform(df_onehot)
#Convert back to Dataframe!
df_scaled=pd.DataFrame(scaler_featured, columns=df_onehot.columns)
df_scaled.head(10)
# Now, we're going to split our data
from sklearn.model_selection import train_test_split
```

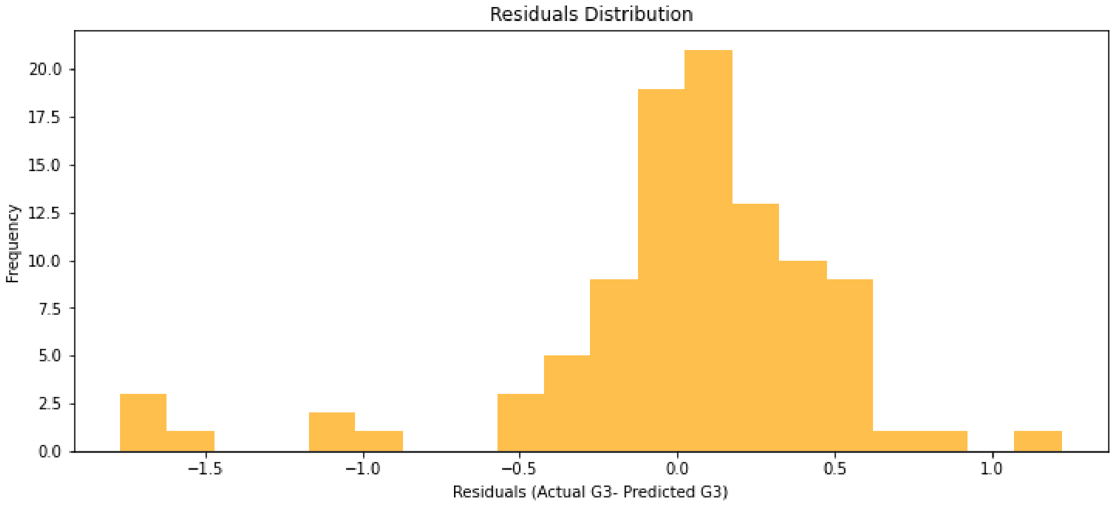
```
# Features(X)= all columns except target
X=df_scaled.drop('G3',axis=1)
#Target(Y or output, which we're going to predict)
y=df_scaled['G3']
#split into training and testing set,for training we'll use 75% of data & for␣
↪testing 25% of modeling!
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.
↪25,random_state=42)
#we'll check its shape
print("X_train",X_train.shape)
print ("X_test",X_test.shape)
print ("y_train",y_train.shape)
print ("y_test",y_test.shape)
#Now , it's time to select model, & we're going to use linear Regression!
from sklearn.linear_model import LinearRegression
model=LinearRegression()
#we're going to train our model
model.fit(X_train,y_train)
#Now, we're going to predicts on our data set
y_pred=model.predict(X_test)
# We'll going to evaluate the model
from sklearn.metrics import mean_squared_error,r2_score
mse=mean_squared_error(y_test,y_pred)
r2=r2_score(y_test,y_pred)
print("Mean Squared Error::",mse)
print("R2 Score::",r2)
# Now , we'll see visualization of prediction vs actual values using matpoltlib␣
↪Library!!!
import matplotlib.pyplot as plt
import numpy as np
# Create an index for x-axis
index = np.arange(len(y_test))
plt.figure(figsize=(12,6))
#plotting Actual Grades'!
plt.scatter(index,y_test,color='red', alpha=0.6, label='Actual G3')
# Plotting predicted Grades'!
plt.scatter(index,y_pred,color='blue', alpha=0.6, label='Predicted G3')
#plotting predicted line!
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='green',␣
↪linewidth=2, label='Perfect Prediction')
plt.xlabel("Actual G3")
plt.ylabel("predicted G3")
plt.title("Actual VS predicted G3(Final Garde's)")
plt.legend()
plt.show()
# Plotting Residuals(shows error for each prediction (R=ActualG3-predictedG3)␣
↪)separately!
```

```python
residuals = y_test - y_pred
plt.figure(figsize=(12,5))
plt.hist(residuals, bins=20, color='orange', alpha=0.7)
plt.xlabel("Residuals (Actual G3- Predicted G3)")
plt.ylabel("Frequency")
plt.title("Residuals Distribution")
plt.show()
```

```
Files in zip: ['student-mat.csv', 'student-por.csv', 'student-merge.R',
'student.txt']
  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  ...

  famrel freetime  goout  Dalc  Walc health absences  G1  G2  G3
0      4        3      4     1     1      3        6   5   6   6
1      5        3      3     1     1      3        4   5   5   6
2      4        3      2     2     3      3       10   7   8  10
3      3        2      2     1     1      5        2  15  14  15
4      4        3      2     1     2      5        4   6  10  10

[5 rows x 33 columns]
<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
dtypes: int64(16), object(17)
memory usage: 102.0+ KB
Categorical Columns: Index(['school', 'sex', 'address', 'famsize', 'Pstatus',
'Mjob', 'Fjob',
       'reason', 'guardian', 'schoolsup', 'famsup', 'paid', 'activities',
       'nursery', 'higher', 'internet', 'romantic'],
      dtype='object')
Numerical Columns: Index(['age', 'Medu', 'Fedu', 'traveltime', 'studytime',
'failures', 'famrel',
       'freetime', 'goout', 'Dalc', 'Walc', 'health', 'absences', 'G1', 'G2',
       'G3'],
      dtype='object')

Missing values:
 <bound method DataFrame.isnull of     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
..     …  ..  …     …     …     …   …   …       …       …
390     MS   M   20       U     LE3       A     2     2  services  services
391     MS   M   17       U     LE3       T     3     1  services  services
392     MS   M   21       R     GT3       T     1     1     other     other
393     MS   M   18       R     LE3       T     3     2  services     other
394     MS   M   19       U     LE3       T     1     1     other   at_home

     … famrel freetime  goout  Dalc  Walc health absences  G1  G2  G3
0    …      4        3      4     1     1      3        6   5   6   6
1    …      5        3      3     1     1      3        4   5   5   6
2    …      4        3      2     2     3      3       10   7   8  10
```

```
3    …      3       2      2     1     1      5        2  15  14  15
4    …      4       3      2     1     2      5        4   6  10  10
..   …    …      …     …    …   …     …       …  ..  ..  ..
390  …      5       5      4     4     5      4       11   9   9   9
391  …      2       4      5     3     4      2        3  14  16  16
392  …      5       5      3     3     3      3        3  10   8   7
393  …      4       4      1     3     4      5        0  11  12  10
394  …      3       2      3     3     3      5        5   8   9   9

[395 rows x 33 columns]>
    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
..     ...  ..  ...     ...     ...     ...   ...   ...       ...       ...
390     MS   M   20       U     LE3       A     2     2  services  services
391     MS   M   17       U     LE3       T     3     1  services  services
392     MS   M   21       R     GT3       T     1     1     other     other
393     MS   M   18       R     LE3       T     3     2  services     other
394     MS   M   19       U     LE3       T     1     1     other   at_home

     ... famrel freetime  goout  Dalc  Walc health absences  G1  G2  G3
0    ...      4        3      4     1     1      3        6   5   6   6
1    ...      5        3      3     1     1      3        4   5   5   6
2    ...      4        3      2     2     3      3       10   7   8  10
3    ...      3        2      2     1     1      5        2  15  14  15
4    ...      4        3      2     1     2      5        4   6  10  10
..   ...    ...      ...    ...   ...   ...    ...      ...  ..  ..  ..
390  ...      5        5      4     4     5      4       11   9   9   9
391  ...      2        4      5     3     4      2        3  14  16  16
392  ...      5        5      3     3     3      3        3  10   8   7
393  ...      4        4      1     3     4      5        0  11  12  10
394  ...      3        2      3     3     3      5        5   8   9   9

[395 rows x 33 columns]


X_train (296, 41)
X_test (99, 41)
y_train (296,)
y_test (99,)
Mean Squared Error:: 0.24085542698623674
R2 Score:: 0.7811139641406579
```


Actual VS predicted G3(Final Garde's)
Perfect Prediction
Actual G3
Predicted G3
predicted G3
2
1
0
−1
−2
−3
0
20
40
60
80
100
Actual G3


Residuals Distribution
Frequency
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
Residuals (Actual G3- Predicted G3)

```
[ ]:
```