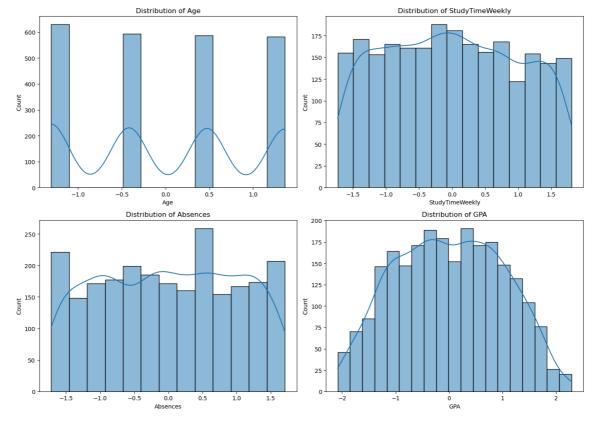
```
import pandas as pd
   In [7]:
                              import seaborn as sns
                             import numpy as np
                             import matplotlib.pyplot as plt
                             from sklearn.model_selection import train_test_split
                             from sklearn.preprocessing import StandardScaler
                             from sklearn.metrics import accuracy_score, classification_report, confusion
                             # Import the dataset
   In [8]:
   In [9]: data = pd.read_csv("E:/Hello Tech DS Project/Data sets/Student Performance of the control of the c
In [10]:
                             # Display first 10 rows
                            data.head()
In [11]:
Out[11]:
                                        StudentID Age Gender Ethnicity ParentalEducation StudyTimeWeekly Absences Tutorii
                                0
                                                    1001
                                                                                                                           0
                                                                                                                                                                                                      19.833723
                                                                                                                                                                                                                                                      7
                                                                        17
                                                    1002
                                                                        18
                                                                                                0
                                                                                                                           0
                                                                                                                                                                           1
                                                                                                                                                                                                      15.408756
                                                                                                                                                                                                                                                      0
                                1
                                2
                                                    1003
                                                                        15
                                                                                                0
                                                                                                                           2
                                                                                                                                                                           3
                                                                                                                                                                                                        4.210570
                                                                                                                                                                                                                                                   26
                                3
                                                    1004
                                                                        17
                                                                                                 1
                                                                                                                           0
                                                                                                                                                                                                      10.028829
                                                                                                                                                                                                                                                    14
                                4
                                                    1005
                                                                        17
                                                                                                 1
                                                                                                                           0
                                                                                                                                                                                                        4.672495
                                                                                                                                                                                                                                                    17
                             # Display last 10 rows
                            data.tail(5)
In [13]:
Out[13]:
                                                StudentID Age
                                                                                          Gender Ethnicity ParentalEducation StudyTimeWeekly Absences Tul
                                2387
                                                            3388
                                                                                18
                                                                                                         1
                                                                                                                                  0
                                                                                                                                                                                   3
                                                                                                                                                                                                              10.680555
                                                                                                                                                                                                                                                               2
                                2388
                                                            3389
                                                                                                         0
                                                                                                                                  0
                                                                                                                                                                                                                7.583217
                                                                                17
                                                                                                                                                                                   1
                                                                                                                                                                                                                                                              4
                                2389
                                                            3390
                                                                                16
                                                                                                                                  0
                                                                                                                                                                                   2
                                                                                                                                                                                                                6.805500
                                                                                                                                                                                                                                                            20
                                2390
                                                            3391
                                                                                16
                                                                                                                                   1
                                                                                                                                                                                   0
                                                                                                                                                                                                              12.416653
                                                                                                                                                                                                                                                            17
                                2391
                                                            3392
                                                                                                                                   0
                                                                                                                                                                                   2
                                                                                                                                                                                                              17.819907
                                                                                16
                                                                                                                                                                                                                                                            13
                             # Finding the Total number of rows and columns
In [14]:
In [15]: data.shape
Out[15]: (2392, 15)
```

#### **DATA PREPROCESSING**

```
In [16]:
          # Checking null values
In [17]:
         data.isnull().sum()
Out[17]: StudentID
                                 0
          Age
                                 0
          Gender
                                 0
          Ethnicity
                                 0
          ParentalEducation
                                 0
          StudyTimeWeekly
                                 0
          Absences
                                 0
          Tutoring
          ParentalSupport
                                 0
          Extracurricular
                                 0
          Sports
                                 0
          Music
                                 0
          Volunteering
                                 0
          GPA
                                 0
          GradeClass
                                 0
          dtype: int64
In [18]:
          # Removing empty rows
In [19]:
          data.dropna(inplace=True)
In [20]:
          # After removing the empty rows
In [21]:
         data.shape
Out[21]: (2392, 15)
In [22]:
          # Feature Scaling
In [23]:
          numerical_features = ['Age','StudyTimeWeekly','Absences','GPA']
          scaler = StandardScaler()
In [24]:
          data[numerical_features] = scaler.fit_transform(data[numerical_features])
          data.head()
Out[24]:
             StudentID
                            Age Gender
                                        Ethnicity ParentalEducation StudyTimeWeekly
                                                                                  Absences 1
           0
                  1001
                        0.472919
                                                               2
                                                                          1.780336
                                                                                   -0.890822
           1
                  1002
                        1.362944
                                      0
                                               0
                                                               1
                                                                          0.997376
                                                                                  -1.717694
           2
                  1003 -1.307132
                                               2
                                                               3
                                                                         -0.984045
                                                                                   1.353542
           3
                  1004
                       0.472919
                                               0
                                                               3
                                                                          0.045445
                                                                                  -0.063951
                  1005
                       0.472919
                                      1
                                               0
                                                               2
                                                                         -0.902311
                                                                                   0.290422
```

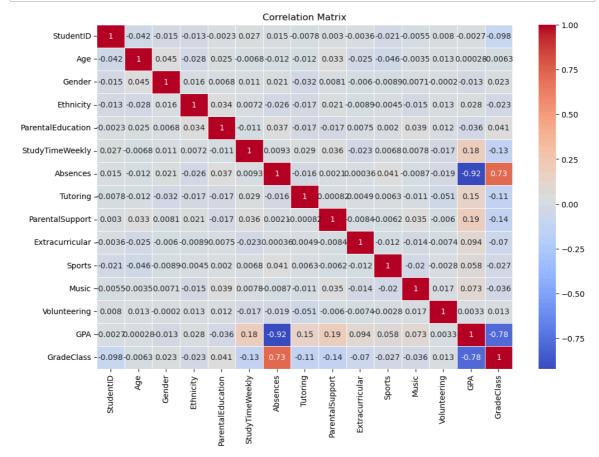
# **EXPLORATORY DATA ANALYSIS (EDA)**

```
In [25]: plt.figure(figsize=(14,10))
    for i,feature in enumerate(numerical_features,1):
        plt.subplot(2,2,i)
        sns.histplot(data[feature],kde=True)
        plt.title(f'Distribution of {feature}')
    plt.tight_layout()
    plt.show()
```

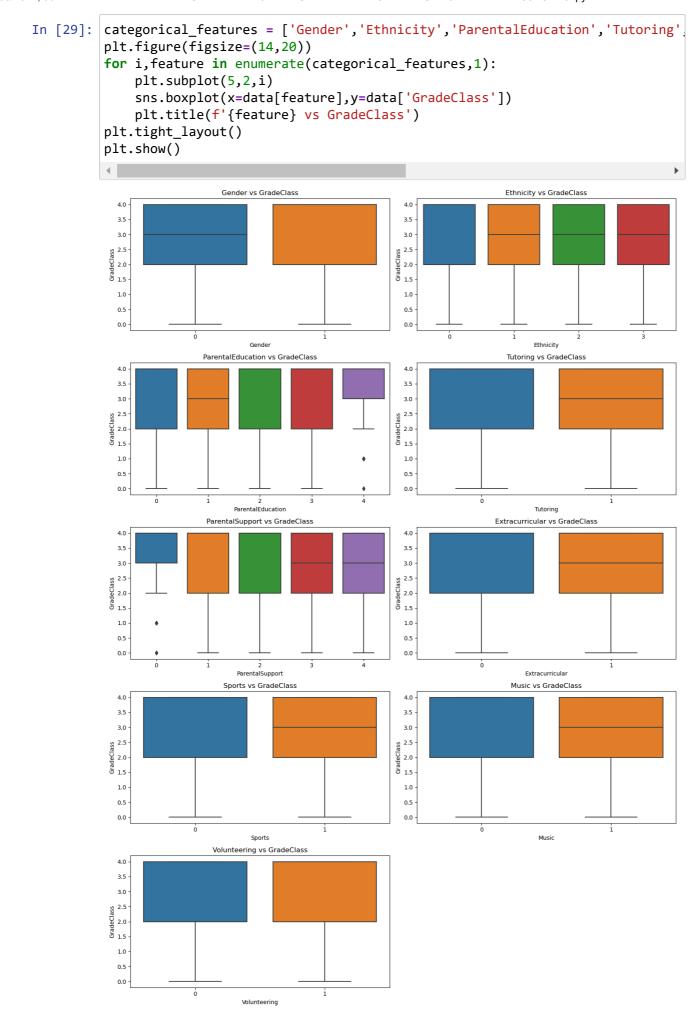


In [26]: # Correction Matrix

```
In [27]: plt.figure(figsize=(12,8))
    corr_matrix = data.corr()
    sns.heatmap(corr_matrix,annot = True, cmap = 'coolwarm',linewidths=0.5)
    plt.title('Correlation Matrix')
    plt.show()
```



In [28]: # Box plots for Categorical Features



#### MODEL SELECTION & TRAINING

```
In [30]: # Split the data
In [31]: x=data.drop(['StudentID','GradeClass'],axis=1)
y=data['GradeClass']
In [32]: # Splitting the data into training and testing sets
In [33]: x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2,randor)
```

## RANDOM FOREST CLASSIFIER

```
In [34]: from sklearn.ensemble import RandomForestClassifier
    RFC = RandomForestClassifier(random_state=42)
    RFC.fit(x_train,y_train)

Out[34]:    RandomForestClassifier
    RandomForestClassifier(random_state=42)
```

#### LOGISTIC REGRESSION

```
In [35]: from sklearn.linear_model import LogisticRegression
    LR = LogisticRegression(random_state=42, max_iter=1000)
    LR.fit(x_train,y_train)

Out[35]:    LogisticRegression
    LogisticRegression(max_iter=1000, random_state=42)
```

# **SUPPORT VECTOR MACHINE (SVM)**

# K-NEAREST NEIGBOUR (KNN)

## **DECISION TREE CLASSIFIER**

## **Model Evaluation**

```
In [ ]: 1. Random Forest Classifier
```

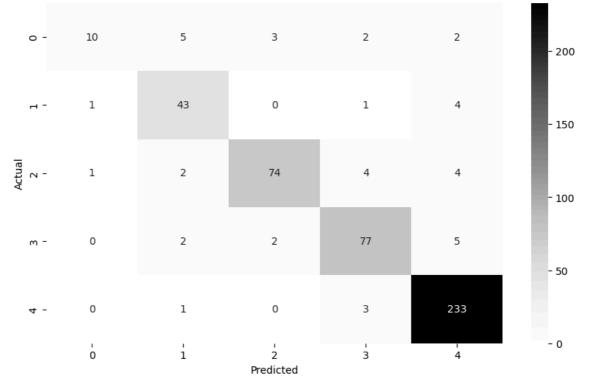
```
In [49]:
         # Make predictions
         y_pred = RFC.predict(x_test)
         #Accuracy
         accuracy = accuracy_score(y_test, y_pred)
         print(f'Accuracy: {accuracy:.2f}')
         #Classification Report
         print("Classification Report")
         print(classification_report(y_test, y_pred))
         #Confusion Matrix
         cm = confusion_matrix(y_test, y_pred)
         plt.figure(figsize=(10,6))
         sns.heatmap(cm,annot=True, fmt='d', cmap='Greys')
         plt.xlabel("Predicted")
         plt.ylabel("Actual")
         plt.title("Confusion Matrix")
         plt.show()
```

Accuracy: 0.91

Classification Report

			cpo. c	CIGSSI, ICGCIO
support	f1-score	recall	precision	
22	0.59	0.45	0.83	0.0
49	0.84	0.88	0.81	1.0
85	0.90	0.87	0.94	2.0
86	0.89	0.90	0.89	3.0
237	0.96	0.98	0.94	4.0
479	0.91			accuracy
479	0.84	0.82	0.88	macro avg
479	0.91	0.91	0.91	weighted avg





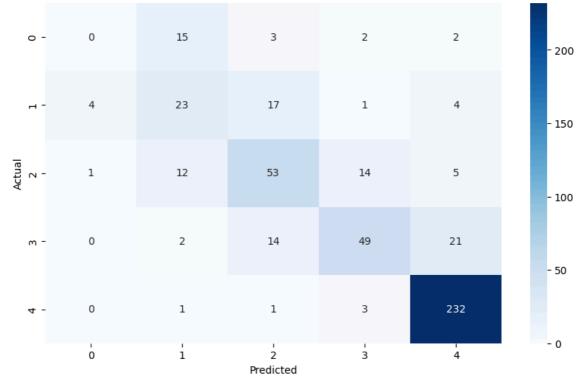
In [ ]: 2. Logistic Regression

```
In [50]:
         # Make prediction
         y_pred=LR.predict(x_test)
         #Accuracy
         accuracy = accuracy_score(y_test, y_pred)
         print(f'Logistic Regression Accuracy: {accuracy:.2f}')
         #Classification Report
         print("Logistic Regression Classification Report")
         print(classification_report(y_test, y_pred))
         #Confusion Matrix
         cm = confusion_matrix(y_test, y_pred)
         plt.figure(figsize=(10,6))
         sns.heatmap(cm,annot=True, fmt='d', cmap='Blues')
         plt.xlabel("Predicted")
         plt.ylabel("Actual")
         plt.title("Logistic Regtression Confusion Matrix")
         plt.show()
```

Logistic Regression Accuracy: 0.75 Logistic Regression Classification Report

U	U				
		precision	recall	f1-score	support
0.	0	0.00	0.00	0.00	22
1.	0	0.43	0.47	0.45	49
2.	0	0.60	0.62	0.61	85
3.	0	0.71	0.57	0.63	86
4.	0	0.88	0.98	0.93	237
accurac	у			0.75	479
macro av	/g	0.53	0.53	0.52	479
weighted av	/g	0.71	0.75	0.73	479





In [ ]: 3. Support Vector Machine(SVM)

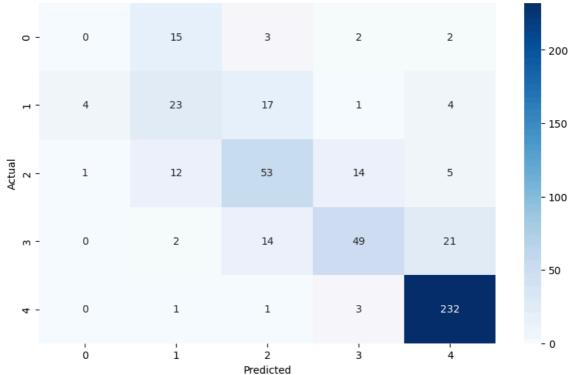
```
In [59]:
         # Make prediction
         y_pred=LR.predict(x_test)
         #Accuracy
         accuracy = accuracy_score(y_test, y_pred)
         print(f'SVM: {accuracy:.2f}')
         #Classification Report
         print("SVM Classification Report")
         print(classification_report(y_test, y_pred))
         #Confusion Matrix
         cm = confusion_matrix(y_test, y_pred)
         plt.figure(figsize=(10,6))
         sns.heatmap(cm,annot=True, fmt='d', cmap='Blues')
         plt.xlabel("Predicted")
         plt.ylabel("Actual")
         plt.title("SVM Confusion Matrix")
         plt.show()
```

SVM: 0.75

SVM Classification Report

support	f1-score	recall	precision	
22	0.00	0.00	0.00	0.0
49	0.45	0.47	0.43	1.0
85	0.61	0.62	0.60	2.0
86	0.63	0.57	0.71	3.0
237	0.93	0.98	0.88	4.0
470	0.75			
479	0.75			accuracy
479	0.52	0.53	0.53	macro avg
479	0.73	0.75	0.71	weighted avg





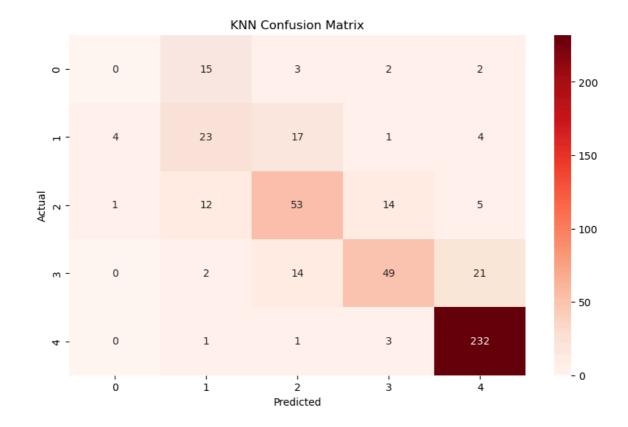
In [ ]: 4. k-Nearest Neighbors

```
In [62]:
         # Make prediction
         y_pred=LR.predict(x_test)
         #Accuracy
         accuracy = accuracy_score(y_test, y_pred)
         print(f'KNN Accuracy: {accuracy:.2f}')
         #Classification Report
         print("KNN Classification Report")
         print(classification_report(y_test, y_pred))
         #Confusion Matrix
         cm = confusion_matrix(y_test, y_pred)
         plt.figure(figsize=(10,6))
         sns.heatmap(cm,annot=True, fmt='d', cmap='Reds')
         plt.xlabel("Predicted")
         plt.ylabel("Actual")
         plt.title("KNN Confusion Matrix")
         plt.show()
```

KNN Accuracy: 0.75

KNN Classification Report

support	f1-score	recall	precision	
22	0.00	0.00	0.00	0.0
49	0.45	0.47	0.43	1.0
85	0.61	0.62	0.60	2.0
86	0.63	0.57	0.71	3.0
237	0.93	0.98	0.88	4.0
479	0.75			accuracy
479	0.52	0.53	0.53	macro avg
479	0.73	0.75	0.71	weighted avg

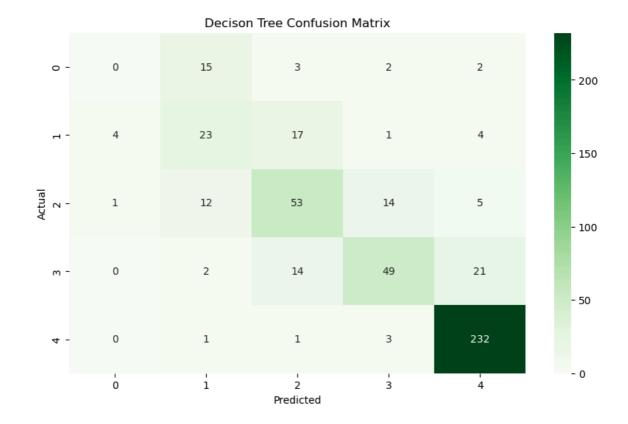


In [ ]: 5. Decision Tree Classifier

```
In [63]:
         # Make prediction
         y_pred=LR.predict(x_test)
         #Accuracy
         accuracy = accuracy_score(y_test, y_pred)
         print(f'Decison Tree Accuracy : {accuracy:.2f}')
         #Classification Report
         print("Decison Tree Classification Report")
         print(classification_report(y_test, y_pred))
         #Confusion Matrix
         cm = confusion_matrix(y_test, y_pred)
         plt.figure(figsize=(10,6))
         sns.heatmap(cm,annot=True, fmt='d', cmap='Greens')
         plt.xlabel("Predicted")
         plt.ylabel("Actual")
         plt.title("Decison Tree Confusion Matrix")
         plt.show()
```

Decison Tree Accuracy : 0.75
Decison Tree Classification Report

	precision	recall	f1-score	support
0.0	0.00	0.00	0.00	22
1.0	0.43	0.47	0.45	49
2.0	0.60	0.62	0.61	85
3.0	0.71	0.57	0.63	86
4.0	0.88	0.98	0.93	237
accuracy			0.75	479
macro avg	0.53	0.53	0.52	479
weighted avg	0.71	0.75	0.73	479



STUDENT PERFORMANCE ANALYSIS - CLASSIFICATION BASED PROJECT - Jupyter No.	oteboo
---	--------

In [ ]:

07/08/2024, 06:21