

An Analysis of Students' Academic Performance Using Big Data Technologies

Abstract

This paper looks into the different factors impacting students' academic performance, measured by class grade. Factors reviewed in this paper include parental involvement, study habits, extracurricular involvement, and academic performance measured by GPA. The data was analysed using both Tableau for data visualisations and Logistic Regression for model prediction. The results proved that students' class grade increases as music activities, parental support, average weekly study time, tutoring participation, extracurricular participation, or volunteering participation increases. This paper also found that logistic regression can be used to predict students' class grade based on age, gender, ethnicity, parental education, study time weekly, absences, tutoring, parental support, extracurricular, sports, music, volunteering, and GPA. The evaluated logistic regression model was a good fit but may be improved in future work.

Introduction

Academic performance refers to the measurement of an individual's achievement across different academic courses (Ballotpedia, n.d). Academic performance can be influenced by many factors such as gender, study habits, socioeconomic status, ethnicity, or extracurricular activities (Al-Tameemi et al, 2023). Academic performance is important because it is a key factor when determining the success of an individual in the workforce (Mappadang, 2022). Thus, making it a priority for universities and schools to track individuals' academic performance to be able to assist individuals with low academic performance promptly. Additionally, poor academic performance is a worldwide issue so understanding the factors that contribute to academic performance can help planners and teachers at educational institutions to make better programs and support plans for students to ensure workforce and academic success (Al-Tameemi et al, 2023).

The aim of this project is to explore and learn more about the different factors that can influence student grades. In the dataset retrieved from Kaggle, different factors that will be studied include parental involvement, study habits, extracurricular involvement, and academic performance. This area is important to study as it can help identify activities and social relationships that can help improve students' academic outcomes. This study can also help increase funding for certain activities while providing insights to students, parents, researchers, and teachers regarding the determinants that can improve academic success and the determinants that can have a negative effect on academic success.

In order to analyse the effects of different factors affecting student grades, a machine learning analysis will be done using the logistic regression model with the PySpark library on Google Colab. Evaluation metrics that will be used include accuracy, precision, recall, and F1 scoring. Additionally, data visualisation will be done using Tableau to visualise the relations between the different factors and the outcome variable and provide additional insights on any trends or patterns that may be evident in the data.

Related Works

In the literature it has been found that there are many different variables that can impact academic performance. The determinants that can impact academic performance range from physical activity, academic interest, gender, socioeconomic status, ethnicity, classroom climate, mental health, social factors, demographic factors, to personal factors (Al-Tameemi et al, 2023). Some of the reviews of the literature are provided below.

Mappadang and colleagues studied the academic performance of 872 undergraduate students in Indonesia and how different variables can impact academic performance (Mappadang, 2022). The variables that were assessed included academic interest, learning attitude, and learning quality (Mappadang, 2022). The results of the study found that academic interest can determine academic performance and students with high academic interest have a higher chance of better academic performance compared to students with little academic interest (Mappadang, 2022). Learning attitude and quality, however, did not show any contribution to academic performance (Mappadang, 2022). The results of this study helped the student's university in adding new learning activities to help promote academic interest of the students to continue providing high academic performance (Mappadang, 2022).

James and colleagues studied the effect of physical activity on academic performance (James et al, 2023). In this study, 6788 students were included, and physical activity frequency, intensity, and type were studied and assessed on how they impact academic performance (James et al, 2023). The results showed that doing physical activity for 90 minutes or more at moderate to vigorous intensity per week can improve academic performance (James et al, 2023). It was also found that the best duration to do physical activity was between 30 to 60 minutes per session doing any type of physical activity to improve academic performance (James et al, 2023). An advantage of this study is that they used participants from different diversities so the results can be generalised to different populations (James et al, 2023). The findings of this study can help teachers, policymakers, parents, and healthcare practitioners in creating physical activity programs or guidelines for students to follow to help in improving students' academic performance (James et al, 2023).

Al-Tameemi and colleagues studied the reasons for students' academic underperformance (Al-Tameemi et al, 2023). This study is a systematic literature review done on 50 studies including both quantitative and qualitative studies (Al-Tameemi et al, 2023). The results found that the main categories that impact students' academic performance include academic, personal, social, and demographic determinants (Al-Tameemi et al, 2023). From these categories, factors that were found to cause poor academic performance include procrastination, low self-esteem, poor mental health, learning disabilities, family challenges, and lack of support (Al-Tameemi et al, 2023). This study also suggests that additional mental health awareness and support programs, financial support systems, peer mentoring programs, extracurricular activities, personal tutoring, and career counsellors can help address underperformance (Al-Tameemi et al, 2023). An advantage of this study is that it increases awareness on the factors that can have

negative effects on students' academic performance and provides ways to address these factors (Al-Tameemi et al, 2023).

Dataset

The student's performance dataset was used. It can be found at <https://www.kaggle.com/datasets/rabieelkharoua/students-performance-dataset?resource=download>. This dataset has 15 columns with 2392 rows. The dataset has 14 features which include student ID, age, gender, ethnicity, parental education, weekly study time, absences per year, tutoring, parental support, extracurricular, sports, music, volunteering, and grade point average (GPA). The outcome variable is class grade. This dataset includes categorical, integer, float, and boolean data types which can be seen in table 1 along with their descriptors.

Table 1. Student Performance Dataset Summary (Kharoua, 2024)

Data Name	Data Type	Description
Student ID	Integer	Identifier assigned to each student from 1001 to 3392
Age	Integer	Age of the students with range between 15 and 18 years
Gender	Boolean	Students gender, 0 for male and 1 for female
Ethnicity	Categorical	Students ethnicity 0 - caucasian 1 - african american 2 - asian 3 - other
Parental Education	Categorical	Parents education level 0 - none 1 - high school 2 - some college 3 - bachelor's 4 - higher
Study Time Weekly	Float	Weekly study time in hours from 0 to 20
Absences	Integer	Total absences during the school year from 0 to 30
Tutoring	Boolean	If the student receives tutoring or not 0 - no 1 - yes

Parental Support	Categorical	The level of parental support the student receives 0 - none 1 - low 2 - moderate 3 - high 4 - very high
Extracurricular	Boolean	If the student participates in extracurricular activities or not 0 - no 1 - yes
Sports	Boolean	If the student participates in sports or not 0 - no 1 - yes
Music	Boolean	If the student participates in music activities or not 0 - no 1 - yes
Volunteering	Boolean	If the student participates in volunteering or not 0 - no 1 - yes
GPA	Float	The students grade point average from 2.0 to 4.0
Class Grade (Target variable)	Categorical	The students grade based on the GPA mark 0 - 'A' (GPA ≥ 3.5) 1 - 'B' ($3.0 \leq \text{GPA} < 3.5$) 2 - 'C' ($2.5 \leq \text{GPA} < 3.0$) 3 - 'D' ($2.0 \leq \text{GPA} < 2.5$) 4 - 'F' (GPA < 2.0)

Data Processing for Pyspark Analysis

All the categorical data types in the dataset are already encoded to numerical values so no encoding was done in the preprocessing stage. In the preprocessing stage, a vector was created to combine all the features except for student ID as student ID would not be used in the

machine learning model. This vector was used as the features along with class grade as the outcome in the machine learning model. There are no missing values in the dataset.

Data Processing for Tableau Visualisation

In order to do visualisation with Tableau, the categorical data types and boolean data types were encoded from numerical to string. This was done in Excel using the find and replace dialogue. The data that was converted to string included gender, ethnicity, parental education, tutoring, parental support, extracurricular, sports, music, volunteering and class grade. A screenshot of the raw data in excel can be seen in figure 1 and a screenshot of the processed data in Excel can be seen in figure 2.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Student_performance_data_														
2	StudentID	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular	Sports	Music	Volunteering	GPA	GradeClass
3	1001	17	1	0	2	19.83372281	7	1	2	0	0	1	0	2.929195592	2
4	1002	18	0	0	1	15.40875606	0	0	1	0	0	0	0	3.042914833	1
5	1003	15	0	2	3	4.210569769	26	0	2	0	0	0	0	0.112602254	4
6	1004	17	1	0	3	10.02882947	14	0	3	1	0	0	0	2.05421814	3
7	1005	17	1	0	2	4.672495273	17	1	3	0	0	0	0	1.288061182	4
8	1006	18	0	0	1	8.191218545	0	0	1	1	0	0	0	3.084183614	1
9	1007	15	0	1	1	15.60168047	10	0	3	0	1	0	0	2.748237415	2
10	1008	15	1	1	4	15.42449631	22	1	1	1	0	0	0	1.360142712	4
11	1009	17	0	0	0	4.562007558	1	0	2	0	1	0	1	2.89681919	2
12	1010	16	1	0	1	18.44446636	0	0	3	1	0	0	0	3.57347421	0
13	1011	17	0	0	1	11.85136366	11	0	1	0	0	0	0	2.147171625	3
14	1012	17	0	0	1	7.598485819	15	0	2	0	0	0	1	1.559594519	4
15	1013	17	0	1	1	10.03871162	21	0	3	1	0	0	0	1.520077815	4
16	1014	17	0	1	2	12.10142507	21	0	4	0	1	0	0	1.751580958	4
17	1015	18	1	0	1	11.19781064	9	1	2	0	0	0	0	2.396788117	3
18	1016	15	0	0	2	9.728100711	17	1	0	0	1	0	0	1.341520717	4
19	1017	18	0	3	1	10.09865608	14	0	2	1	1	0	0	2.232175278	3
20	1018	18	1	0	0	3.528238209	16	1	2	0	0	0	0	1.384404176	4
21	1019	18	0	1	3	16.25465809	29	0	2	1	0	0	1	0.469553323	4
22	1020	17	0	0	1	10.8352064	9	0	2	0	0	1	0	2.395784095	3
23	1021	16	1	0	3	2.621597234	2	0	3	0	0	0	1	2.7784113	2
24	1022	15	0	0	2	15.32314203	25	0	1	1	0	0	0	0.346894037	4
25	1023	16	1	1	0	18.64887957	29	1	1	0	0	0	0	0.312546231	4
26	1024	18	1	3	4	18.94613798	20	0	2	1	0	0	0	1.770131877	4
27	1025	18	1	0	1	7.380354648	15	0	2	0	0	0	0	1.505155622	4
28	1026	16	1	0	3	2.710337471	5	0	4	0	0	1	0	2.977851918	2
29	1027	16	0	0	1	10.36799253	2	0	2	0	1	0	0	2.948717672	2
30	1028	16	1	0	3	2.252184587	8	0	3	0	0	1	0	2.14520472	3
31	1029	18	0	0	0	18.67974837	10	0	3	1	0	0	0	2.854803929	2
32	1030	18	0	0	2	3.671592547	20	0	3	1	0	0	0	1.519441726	4

Figure 1. Pre-processed Tableau data in Excel

Student_performance_data_															
StudentID	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular	Sports	Music	Volunteering	GPA	GradeClass	
1001	17	Female	Caucasian	Some College	19.83372281	7	Yes	Moderate	No	No	Yes	No	2.929195592	C	
1002	18	Male	Caucasian	High School	15.40875606	0	No	Low	No	No	No	No	3.042914833	B	
1003	15	Male	Asian	Bachelor's	4.210569769	26	No	Moderate	No	No	No	No	0.112602254	F	
1004	17	Female	Caucasian	Bachelor's	10.02882947	14	No	High	Yes	No	No	No	2.05421814	D	
1005	17	Female	Caucasian	Some College	4.672495273	17	Yes	High	No	No	No	No	1.288061182	F	
1006	18	Male	Caucasian	High School	8.191218545	0	No	Low	Yes	No	No	No	3.084183614	B	
1007	15	Male	African Am	High School	15.60168047	10	No	High	No	Yes	No	No	2.748237415	C	
1008	15	Female	African Am	Higher	15.42449631	22	Yes	Low	Yes	No	No	No	1.360142712	F	
1009	17	Male	Caucasian	None	4.562007558	1	No	Moderate	No	Yes	No	Yes	2.89681919	C	
1010	16	Female	Caucasian	High School	18.44446636	0	No	High	Yes	No	No	No	3.57347421	A	
1011	17	Male	Caucasian	High School	11.85136366	11	No	Low	No	No	No	No	2.147171625	D	
1012	17	Male	Caucasian	High School	7.598485819	15	No	Moderate	No	No	No	Yes	1.559594519	F	
1013	17	Male	African Am	High School	10.03871162	21	No	High	Yes	No	No	No	1.520077815	F	
1014	17	Male	African Am	Some College	12.10142507	21	No	Very High	No	Yes	No	No	1.751580958	F	
1015	18	Female	Caucasian	High School	11.19781064	9	Yes	Moderate	No	No	No	No	2.396788117	D	
1016	15	Male	Caucasian	Some College	9.728100711	17	Yes	None	No	Yes	No	No	1.341520717	F	
1017	18	Male	Other	High School	10.09865608	14	No	Moderate	Yes	Yes	No	No	2.232175278	D	
1018	18	Female	Caucasian	None	3.528238209	16	Yes	Moderate	No	No	No	No	1.384404176	F	
1019	18	Male	African Am	Bachelor's	16.25465809	29	No	Moderate	Yes	No	No	Yes	0.469553323	F	
1020	17	Male	Caucasian	High School	10.8352064	9	No	Moderate	No	No	Yes	No	2.395784095	D	
1021	16	Female	Caucasian	Bachelor's	2.621597234	2	No	High	No	No	No	Yes	2.7784113	C	
1022	15	Male	Caucasian	Some College	15.32314203	25	No	Low	Yes	No	No	No	0.346894037	F	
1023	16	Female	African Am	None	18.64887957	29	Yes	Low	No	No	No	No	0.312546231	F	
1024	18	Female	Other	Higher	18.94613798	20	No	Moderate	Yes	No	No	No	1.770131877	F	
1025	18	Female	Caucasian	High School	7.380354648	15	No	Moderate	No	No	No	No	1.505155622	F	
1026	16	Female	Caucasian	Bachelor's	2.710337471	5	No	Very High	No	No	Yes	No	2.977851918	C	
1027	16	Male	Caucasian	High School	10.36799253	2	No	Moderate	No	Yes	No	No	2.948717672	C	
1028	16	Female	Caucasian	Bachelor's	2.252184587	8	No	High	No	No	Yes	No	2.14520472	D	
1029	18	Male	Caucasian	None	18.67974837	10	No	High	Yes	No	No	No	2.854803929	C	
1030	18	Male	Caucasian	Some College	3.671592547	20	No	High	Yes	No	No	No	1.519441726	F	

Sheet 1 - Student_performance_d

Figure 2. Processed Tableau data in Excel

Methodology

Logistic regression

Logistic regression will be the machine learning method used for analysing the dataset and making predictions for students' class grade using the Pyspark library on Google Colab. Logistic regression is a supervised learning method that works to determine classifications between different categories (IBM, 2021). Multinomial regression will be used as it is the preferred method to use when the predicted outcome can be chosen from three or more classes (IBM, 2021).

Tableau

Tableau is a data visualisation software that was founded in 2003 (Tableau, 2024). This software allows researchers to explore, understand and manage data, and share insights to others (Tableau, 2024). Tableau desktop 2024.1 was installed following the instructions provided in section 2 tutorial 1 pdf from Aula 7153CEM big data analytics and data visualisation. Tableau desktop 2024.1 will be used for the data visualisation aspect of this assignment to view the relationships between the different data columns. A screenshot of Tableau desktop installation can be found in figure 3.

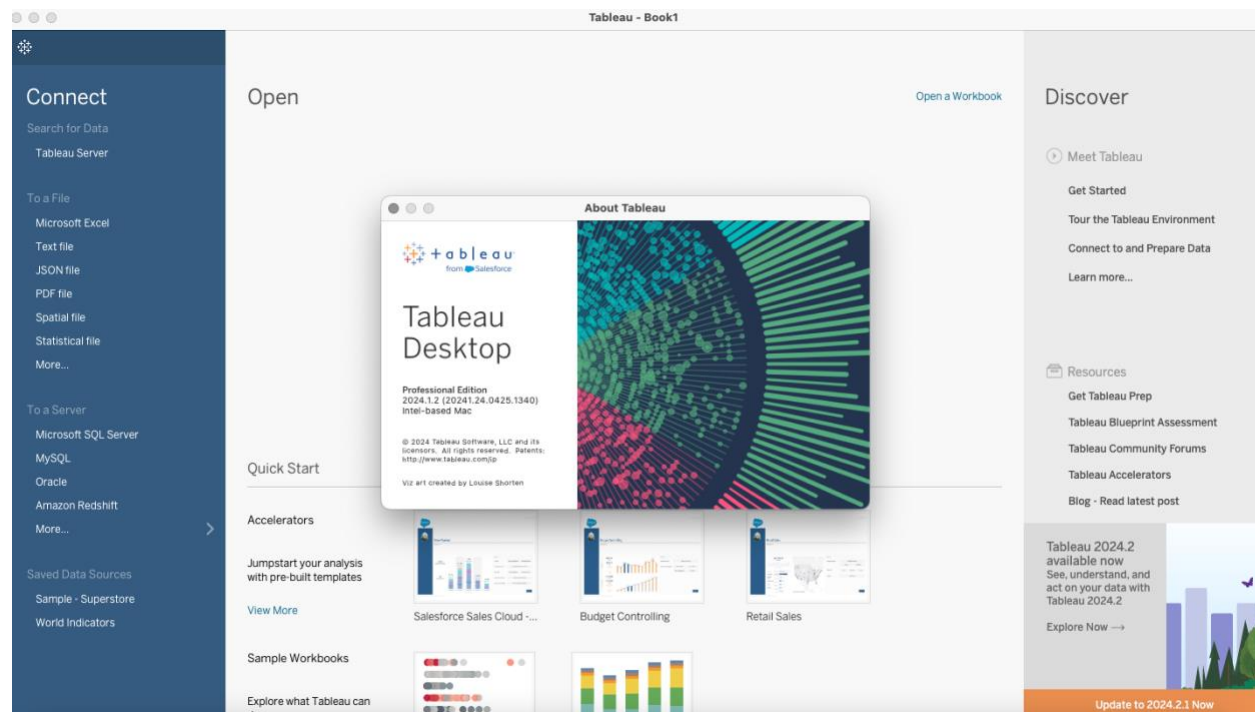
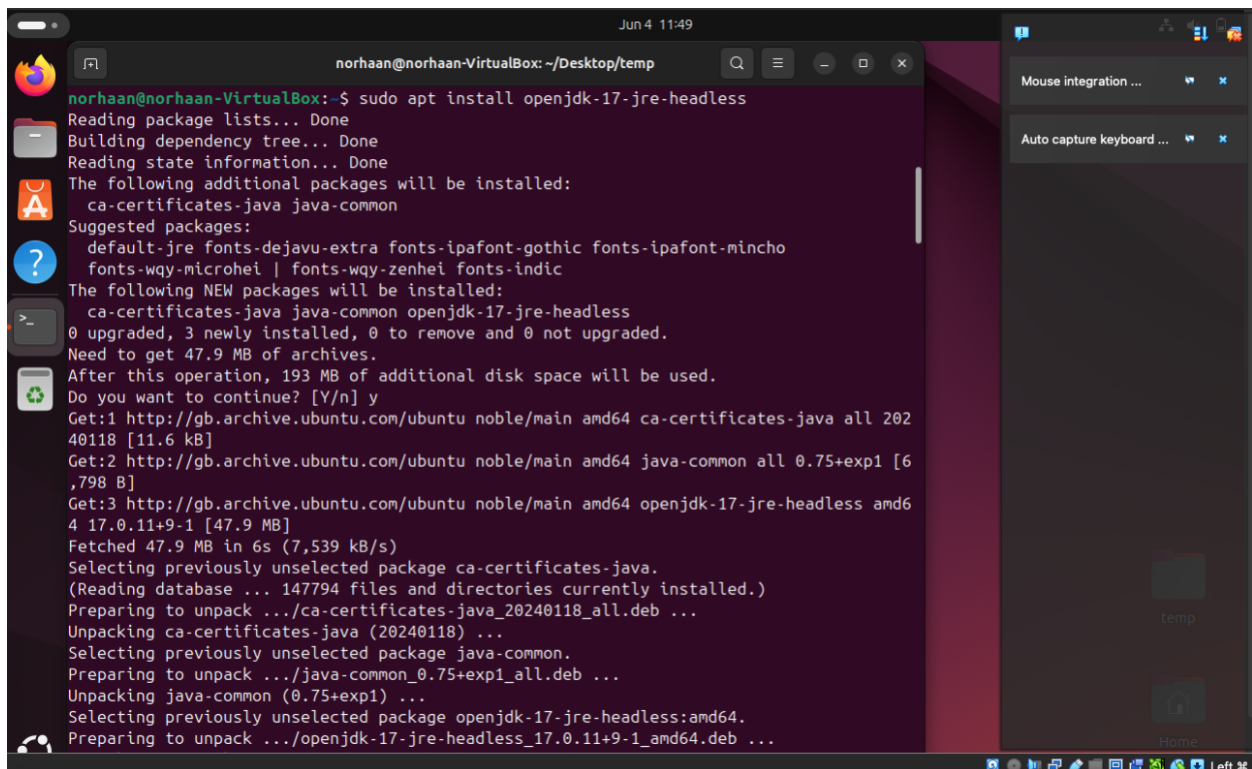


Figure 3. Tableau version

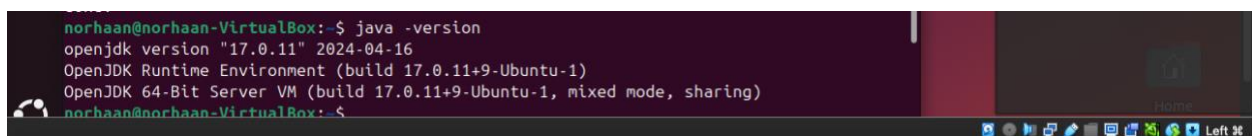
Software Installations

Hadoop 3.3.6 and Spark 3.5.1 were installed on Ubuntu on a virtual machine as requested in the coursework. Java version 17.0.11 was installed as well. Java and Hadoop were installed following the instructions provided in section 4 lab 4 Hadoop local mode PowerPoint slides from Aula 7153CEM Big Data Analytics and Data Visualisation. Spark was installed following the instructions provided in section 5 Spark local mode PowerPoint slides from Aula 7153CEM Big Data Analytics and Data Visualisation. Screenshots of the download and installation of java can be found in figures 4 and 5. Screenshots of the download and installation of Hadoop can be found in figures 6 and 7. Screenshots of the download and installation of Spark can be found in figures 8 and 9.



```
norhaan@norhaan-VirtualBox: ~/Desktop/temp
norhaan@norhaan-VirtualBox:~$ sudo apt install openjdk-17-jre-headless
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java java-common
Suggested packages:
  default-jre fonts-dejavu-extra fonts-ipafont-gothic fonts-ipafont-mincho
  fonts-wqy-microhei | fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
  ca-certificates-java java-common openjdk-17-jre-headless
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 47.9 MB of archives.
After this operation, 193 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://gb.archive.ubuntu.com/ubuntu noble/main amd64 ca-certificates-java all 202
40118 [11.6 kB]
Get:2 http://gb.archive.ubuntu.com/ubuntu noble/main amd64 java-common all 0.75+exp1 [6
,798 B]
Get:3 http://gb.archive.ubuntu.com/ubuntu noble/main amd64 openjdk-17-jre-headless amd6
4 17.0.11+9-1 [47.9 MB]
Fetched 47.9 MB in 6s (7,539 kB/s)
Selecting previously unselected package ca-certificates-java.
(Reading database ... 147794 files and directories currently installed.)
Preparing to unpack .../ca-certificates-java_20240118_all.deb ...
Unpacking ca-certificates-java (20240118) ...
Selecting previously unselected package java-common.
Preparing to unpack .../java-common_0.75+exp1_all.deb ...
Unpacking java-common (0.75+exp1) ...
Selecting previously unselected package openjdk-17-jre-headless:amd64.
Preparing to unpack .../openjdk-17-jre-headless_17.0.11+9-1_amd64.deb ...
```

Figure 4. Java download



```
norhaan@norhaan-VirtualBox:~$ java -version
openjdk version "17.0.11" 2024-04-16
OpenJDK Runtime Environment (build 17.0.11+9-Ubuntu-1)
OpenJDK 64-Bit Server VM (build 17.0.11+9-Ubuntu-1, mixed mode, sharing)
norhaan@norhaan-VirtualBox:~$
```

Figure 5. Java installation

```
norhaan@norhaan-VirtualBox:~/Desktop/temp$ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz
--2024-06-04 11:53:36-- https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)[151.101.2.132]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 730107476 (696M) [application/x-gzip]
Saving to: 'hadoop-3.3.6.tar.gz'

hadoop-3.3.6.tar.gz 100%[=====] 696.28M 11.3MB/s in 59s

2024-06-04 11:54:40 (11.7 MB/s) - 'hadoop-3.3.6.tar.gz' saved [730107476/730107476]
```

Figure 6. Hadoop download

```
norhaan@norhaan-VirtualBox: ~
jdi
norhaan@norhaan-VirtualBox:~$ hadoop
Usage: hadoop [OPTIONS] SUBCOMMAND [SUBCOMMAND OPTIONS]
or hadoop [OPTIONS] CLASSNAME [CLASSNAME OPTIONS]
where CLASSNAME is a user-provided Java class

OPTIONS is none or any of:

buildpaths          attempt to add class files from build tree
--config dir         Hadoop config directory
--debug             turn on shell script debug mode
--help             usage information
hostnames list[,of,host,names] hosts to use in worker mode
hosts filename      list of hosts to use in worker mode
loglevel level      set the log4j level for this command
workers            turn on worker mode

SUBCOMMAND is one of:

Admin Commands:

daemonlog          get/set the log level for each daemon

Client Commands:

archive            create a Hadoop archive
checknative        check native Hadoop and compression libraries availability
classpath          prints the class path needed to get the Hadoop jar and the
                  required libraries
confcheck          validate configuration XML files
```

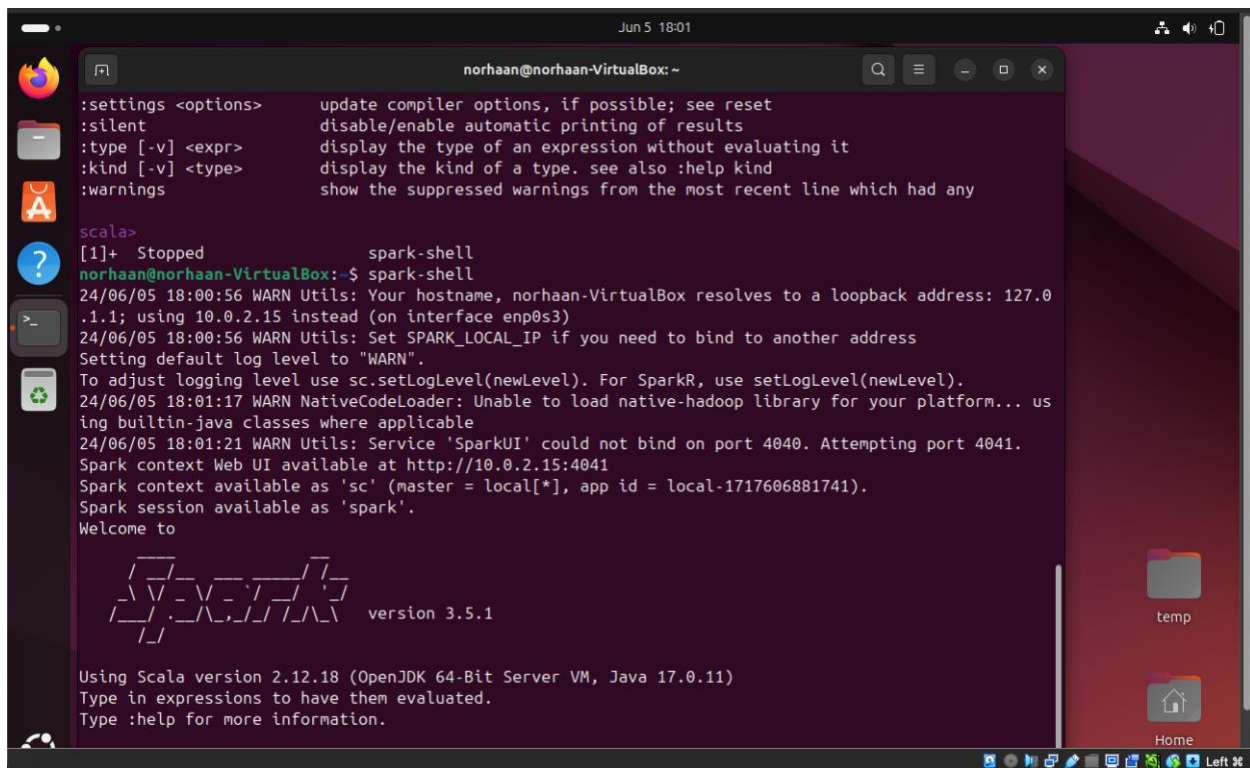
Figure 7. Hadoop installation

```
norhaan@norhaan-VirtualBox:~/Desktop/temp$ wget https://dlcdn.apache.org/spark/spark-3.5.1/spark-3.5.1-bin-hadoop3.tgz
--2024-06-05 16:06:17-- https://dlcdn.apache.org/spark/spark-3.5.1/spark-3.5.1-bin-hadoop3.tgz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)[151.101.2.132]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 400446614 (382M) [application/x-gzip]
Saving to: 'spark-3.5.1-bin-hadoop3.tgz'

spark-3.5.1-bin-hadoo 100%[=====] 381.90M 3.00MB/s in 1m 59s

2024-06-05 16:08:20 (3.20 MB/s) - 'spark-3.5.1-bin-hadoop3.tgz' saved [400446614/400446614]
```

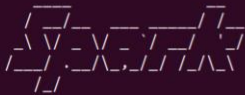
Figure 8. Spark download



The screenshot shows a terminal window titled "norhaan@norhaan-VirtualBox: ~" with a dark purple background. The terminal displays the following content:

```
:settings <options>    update compiler options, if possible; see reset
:silent                disable/enable automatic printing of results
:type [-v] <expr>      display the type of an expression without evaluating it
:kind [-v] <type>      display the kind of a type. see also :help kind
:warnings              show the suppressed warnings from the most recent line which had any
```

scala>
[1]+ Stopped spark-shell
norhaan@norhaan-VirtualBox:~\$ spark-shell
24/06/05 18:00:56 WARN Utils: Your hostname, norhaan-VirtualBox resolves to a loopback address: 127.0.0.1; using 10.0.2.15 instead (on interface enp0s3)
24/06/05 18:00:56 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/06/05 18:01:17 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
24/06/05 18:01:21 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Spark context Web UI available at http://10.0.2.15:4041
Spark context available as 'sc' (master = local[*], app id = local-1717606881741).
Spark session available as 'spark'.
Welcome to

 version 3.5.1

Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 17.0.11)
Type in expressions to have them evaluated.
Type :help for more information.

The terminal window is part of a desktop environment with a taskbar at the bottom showing various application icons and system status indicators. The desktop background is a dark purple gradient with a folder icon labeled "temp" and a home icon labeled "Home".

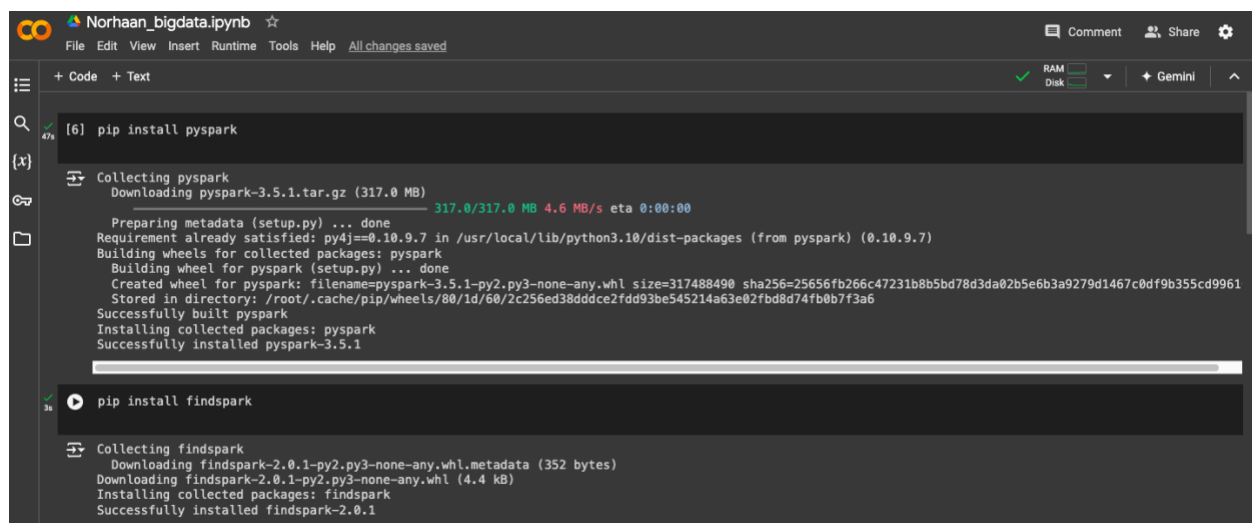
Figure 9. Spark installation

Experimental

Logistic Regression Analysis

In order to perform logistic regression on the dataset using Pyspark on Google Colab, the dataset needed to be downloaded from the Kaggle site and converted to a csv file to be opened in Google Colab. Pyspark and findspark were installed onto Google Colab and a new Spark session was created as shown in figure 10 and 11. After this, the dataset was opened and viewed to ensure it was converted correctly as shown in figure 12.

The vectorassembler function was then imported to combine the selected features into one vector to be used in the machine learning algorithm. Once this was done, a new table was created with the features in one column and the outcome variable which is class grade in the other column. This was named finalised_data and is shown in figure 13. After this, the data was split into the test and train set with 80% of the data going into the training set and 20% of the data going into the test set. Logistic regression was done, and evaluations were carried out using the multiclassclassificationevaluator function from Pyspark. Evaluation metrics that were calculated include accuracy, precision, recall, and f1 scoring which can be seen in figure 14 and table 2. Additionally, all the coding input can be found in the appendix.



```
Norhaan_bigdata.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[6] pip install pyspark

Collecting pyspark
  Downloading pyspark-3.5.1.tar.gz (317.0 MB)
    317.0/317.0 MB 4.6 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.5.1-py2.py3-none-any.whl size=317488490 sha256=25656fb266c47231b8b5bd78d3da02b5e6b3a9279d1467c0df9b355cd9961
  Stored in directory: /root/.cache/pip/wheels/80/1d/60/2c256ed38ddc2fdd93be545214a63e02fbd8d74fb0b7f3a6
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.5.1

[36] pip install findspark

Collecting findspark
  Downloading findspark-2.0.1-py2.py3-none-any.whl.metadata (352 bytes)
  Downloading findspark-2.0.1-py2.py3-none-any.whl (4.4 kB)
Installing collected packages: findspark
Successfully installed findspark-2.0.1
```

Figure 10. Pyspark and findspark installations

```
Norhaan_bigdata.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[9] import findspark
    findspark.init()

[10] from pyspark.sql import SparkSession
    from pyspark.ml.feature import VectorAssembler
    from pyspark.ml.classification import LogisticRegression

[12] spark = SparkSession.builder.appName("student performance").getOrCreate()
```

Figure 11. Creating a new spark session

```
Norhaan_bigdata.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

dataset = spark.read.csv("/content/Student_performance_data_2.csv", inferSchema = True, header = True)
print('There are ',len(dataset.columns),'columns in this dataset')
print('There are ',dataset.count(), 'rows in this dataset')

There are 15 columns in this dataset
There are 2392 rows in this dataset

[ ] dataset.show()

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|StudentID|Age|Gender|Ethnicity|ParentalEducation|StudyTimeWeekly|Absences|Tutoring|ParentalSupport|Extracurricular|Sports|Music|Volunteering|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|1001|17|1|0|2|19.8337228078547|7|1|2|0|0|1|0|2.9291955916|
|1002|18|0|0|1|15.4087560558467|0|0|1|0|0|0|0|3.0429148334|
|1003|15|0|2|3|4.21056976881226|26|0|2|0|0|0|0|0.11260225446|
|1004|17|1|0|3|10.0288294739582|14|0|3|1|0|0|0|2.0542181397|
|1005|17|1|0|2|4.67249527297133|17|1|3|0|0|0|0|1.2880611817|
|1006|18|0|0|1|8.19121854525019|0|0|1|1|0|0|0|3.0841836144|
|1007|15|0|1|1|15.6016804746993|10|0|3|0|1|0|0|2.7482374148|
|1008|15|1|1|4|15.4244963058081|22|1|1|1|0|0|0|1.3601427123|
|1009|17|0|0|0|4.5620075580477|1|0|2|0|1|0|1|2.8968191895|
|1010|16|1|0|1|18.4444663630972|0|0|3|1|0|0|0|3.5734742103|
|1011|17|0|0|1|11.8513636552965|11|0|1|0|0|0|0|2.1471716250|
|1012|17|0|0|1|7.59848581924029|15|0|2|0|0|0|1|1.5595945190|
|1013|17|0|1|1|10.0387116156172|21|0|3|1|0|0|0|1.5200778148|
|1014|17|0|1|2|12.1014250687549|21|0|4|0|1|0|0|1.7515809583|
|1015|18|1|0|1|11.1978106369157|9|1|2|0|0|0|0|2.306788117|
|1016|15|0|0|2|9.72810071072356|17|1|0|0|1|0|0|1.3415207165|
|1017|18|0|3|1|10.098656081788|14|0|2|1|1|0|0|2.2321752777|
|1018|18|1|0|0|3.52823820855772|16|1|2|0|0|0|0|1.3844041756|
|1019|18|0|1|3|16.2546580860936|29|0|2|1|0|0|1|0.4695533233|
|1020|17|0|0|1|10.8352063988203|9|0|2|0|0|1|0|2.395784094|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

only showing top 20 rows
```

Figure 12. Viewing the dataset

The screenshot shows a Jupyter Notebook interface with the following code and output:

```
[17] assembler = VectorAssembler(inputCols = ['Age', 'Gender', 'Ethnicity', 'ParentalEducation', 'StudyTimeWeekly', 'Absences', 'Tutoring', 'ParentalSupport', '
output = assembler.transform(dataset)

finalised_data = output.select('features', 'GradeClass')
finalised_data.show()
```

The output displays a table with two columns: 'features' and 'GradeClass'. The 'features' column contains a list of numerical values for each row, and the 'GradeClass' column contains a single numerical value. The table shows the top 20 rows of data.

features	GradeClass
[17.0, 1.0, 0.0, 2.0, ...]	2
[13.0, 3.4, 7.12, ...]	1
[13.0, 2.3, 4.5, 7.0, ...]	4
[17.0, 1.0, 0.0, 3.0, ...]	3
[17.0, 1.0, 0.0, 2.0, ...]	4
[13.0, 3.4, 7.8, 12.0, ...]	1
[15.0, 0.0, 1.0, 1.0, ...]	2
[15.0, 1.0, 1.0, 4.0, ...]	4
[13.0, 0.4, 5.7, 9.11, ...]	2
[13.0, 0.1, 3.4, 7.8, ...]	0
[13.0, 0.3, 4.5, 7.12, ...]	3
[13.0, 0.3, 4.5, 7.11, ...]	4
[17.0, 0.0, 1.0, 1.0, ...]	4
[17.0, 0.0, 1.0, 2.0, ...]	4
[18.0, 1.0, 0.0, 1.0, ...]	3
[13.0, 0.3, 4.5, 6.9, ...]	4
[18.0, 0.0, 3.0, 1.0, ...]	3
[13.0, 0.1, 4.5, 6.7, ...]	4
[18.0, 0.0, 1.0, 3.0, ...]	4
[13.0, 0.3, 4.5, 7.10, ...]	3

Figure 13. Viewing the data to be input into the machine learning algorithm

The screenshot shows a Jupyter Notebook interface with the following code and output:

```
[20] train, test = finalised_data.randomSplit([0.8, 0.2])

[23] log_reg = LogisticRegression(featuresCol = "features", labelCol="GradeClass")
log_regmodel = log_reg.fit(train)
predictions = log_regmodel.transform(test)

[25] from pyspark.ml.evaluation import MulticlassClassificationEvaluator
multi_eval = MulticlassClassificationEvaluator(labelCol="GradeClass", predictionCol="prediction")
accuracy = multi_eval.evaluate(predictions, {multi_eval.metricName: "accuracy"})
precision = multi_eval.evaluate(predictions, {multi_eval.metricName: "weightedPrecision"})
recall = multi_eval.evaluate(predictions, {multi_eval.metricName: "weightedRecall"})
f1 = multi_eval.evaluate(predictions, {multi_eval.metricName: "f1"})

print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1: {f1:.4f}")
```

The output displays the evaluation metrics for the logistic regression model:

```
Accuracy: 0.8090
Precision: 0.7911
Recall: 0.8090
F1: 0.7959
```

Figure 14. Logistic regression evaluation metrics results

Table 2. Evaluation metrics

Algorithm	Accuracy	Precision	Recall	F1
Logistic regression	0.809	0.791	0.809	0.795

Tableau Visualisation

In order to do visualisation with Tableau, the categorical data types and boolean data types were first encoded from numerical to string data types. This was done in Excel using the find and replace dialogue. The data that was converted to string included gender, ethnicity, parental education, tutoring, parental support, extracurricular, sports, music, volunteering and class grade. After this was complete, the Excel file was opened in Tableau to do visualisation. The graph outputs can be found in figures 15 to 28.

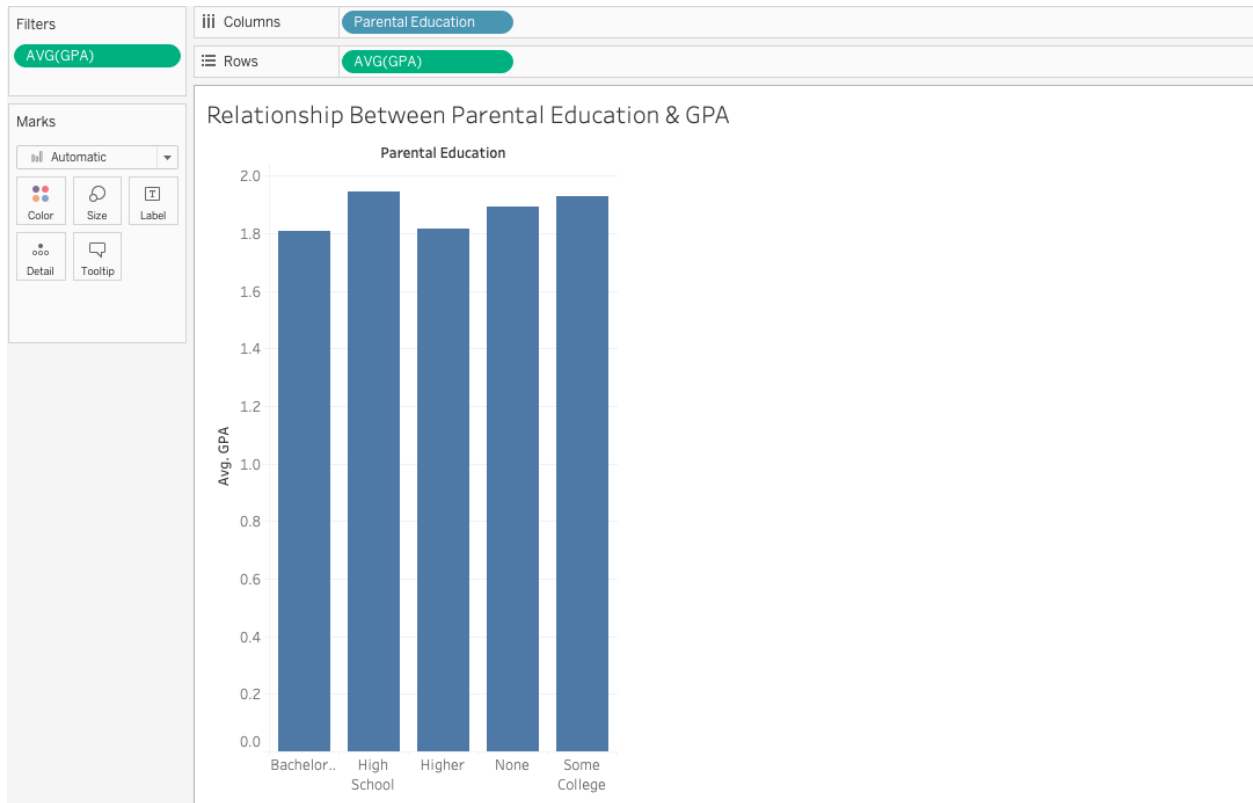


Figure 15. The relationship between parental education and GPA



Figure 16. The relationship between gender, age, ethnicity, and average weekly study time

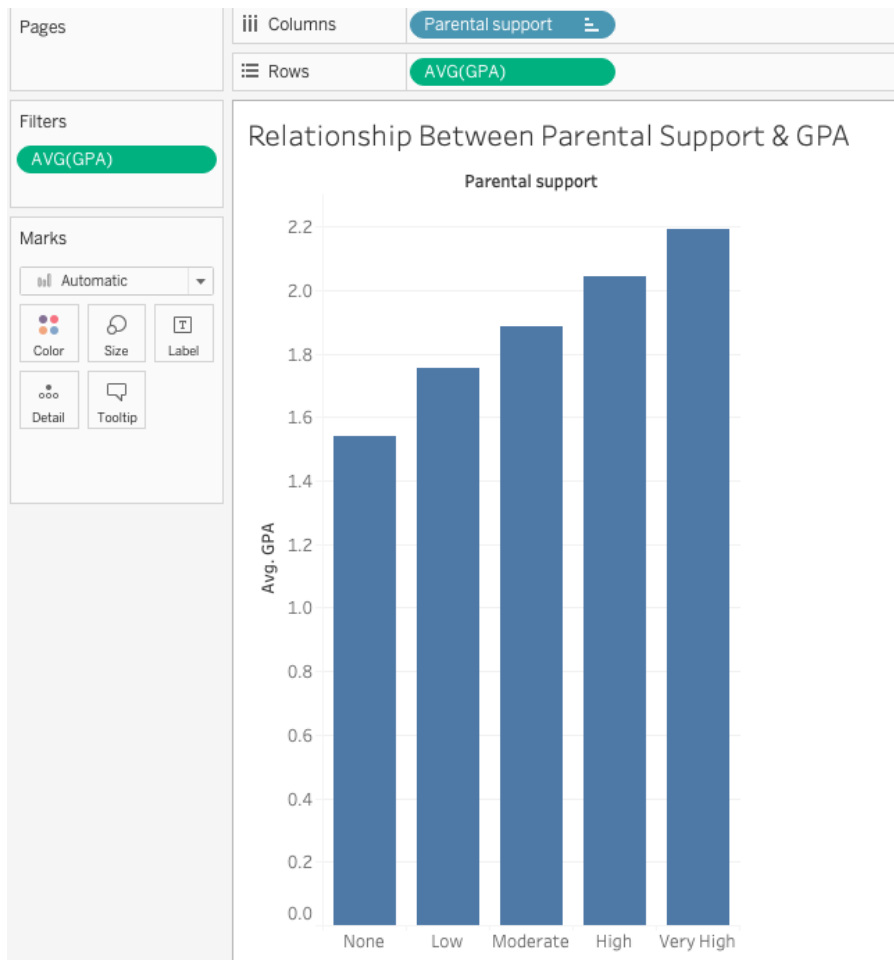


Figure 17. The relationship between parental support and GPA

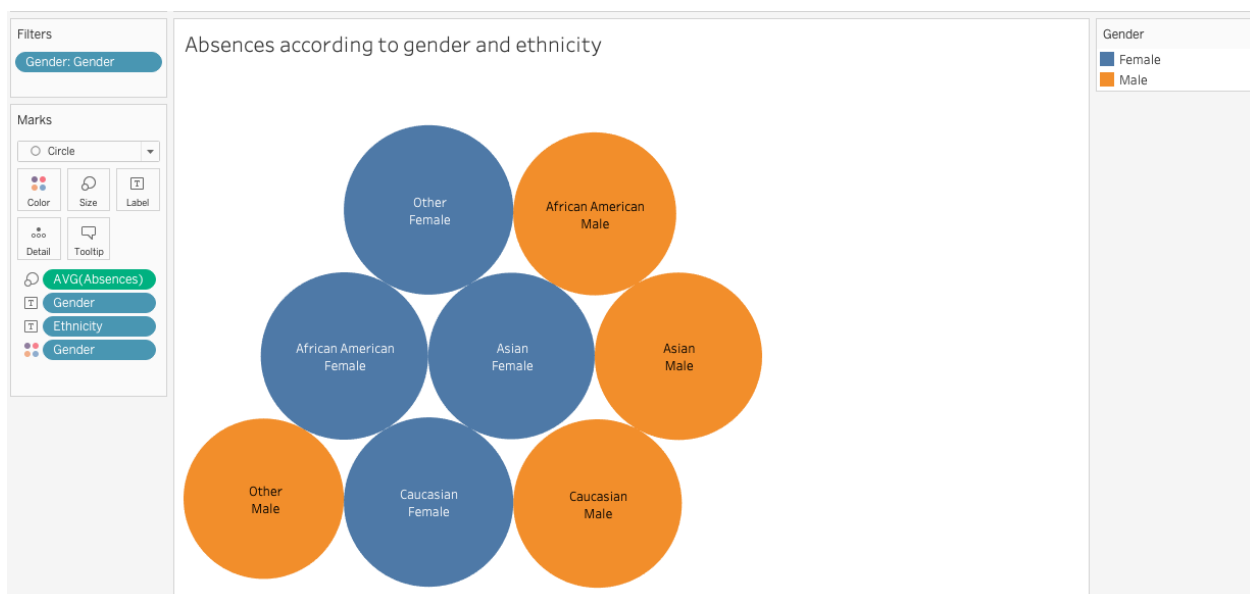


Figure 18. The relationship between absences per year, gender, and ethnicity

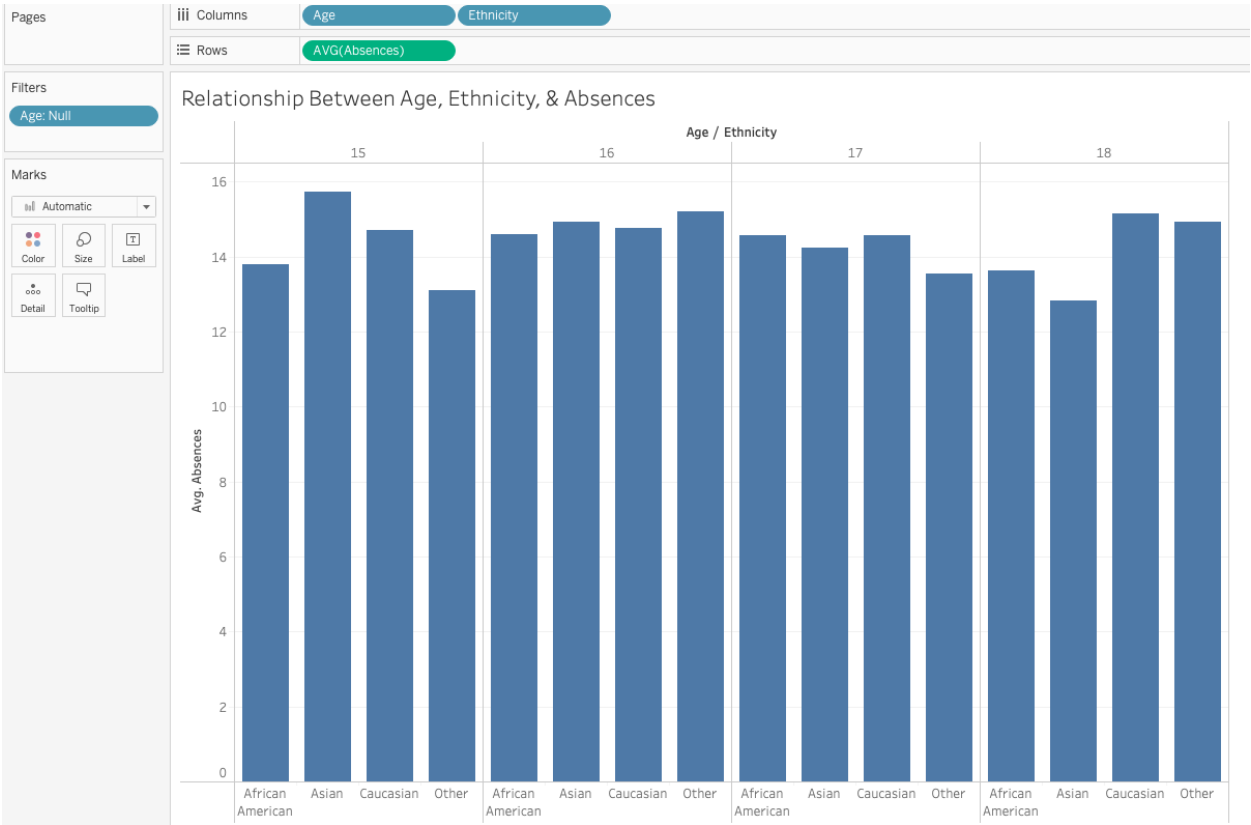


Figure 19. The relationship between age, ethnicity, and absences per year

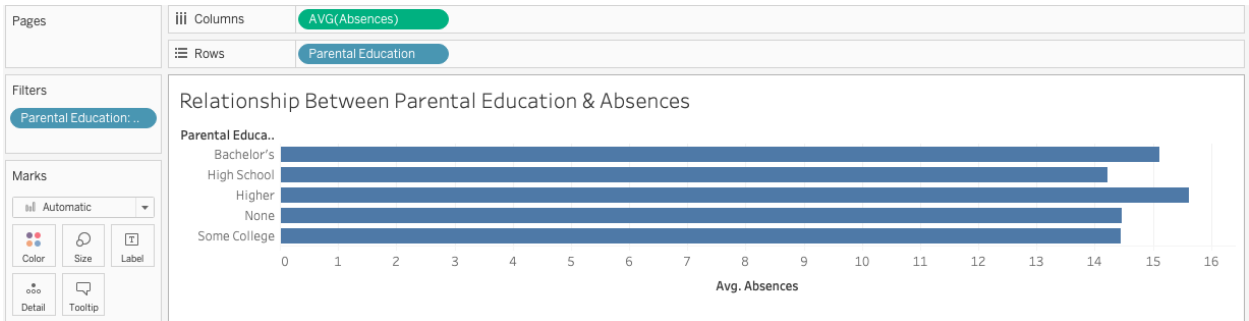


Figure 20. The relationship between parental education and absences per year

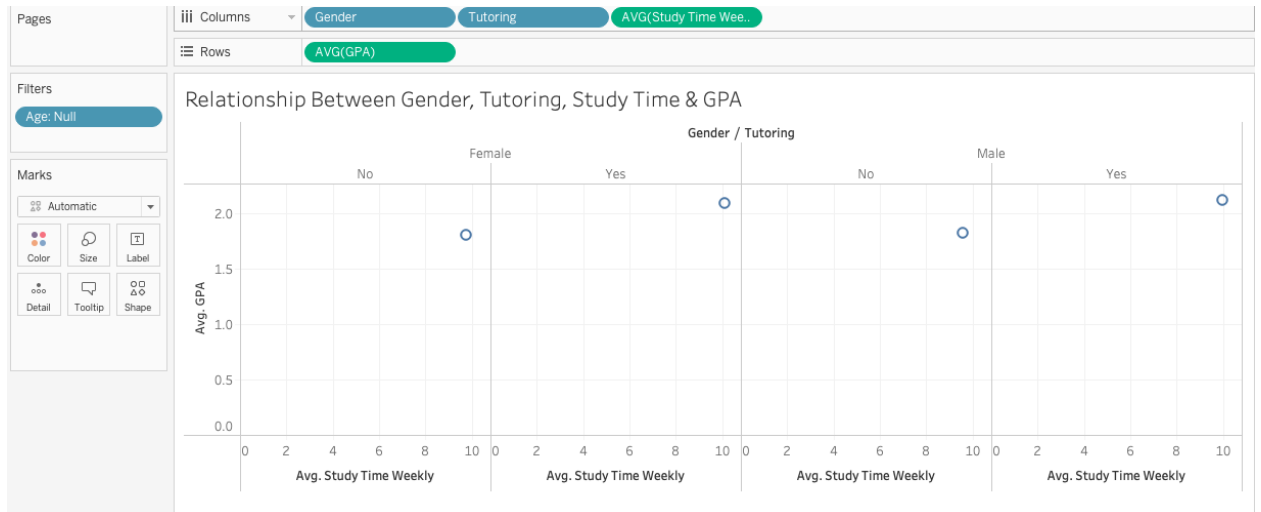


Figure 21. The relationship between gender, tutoring, average weekly study time and GPA

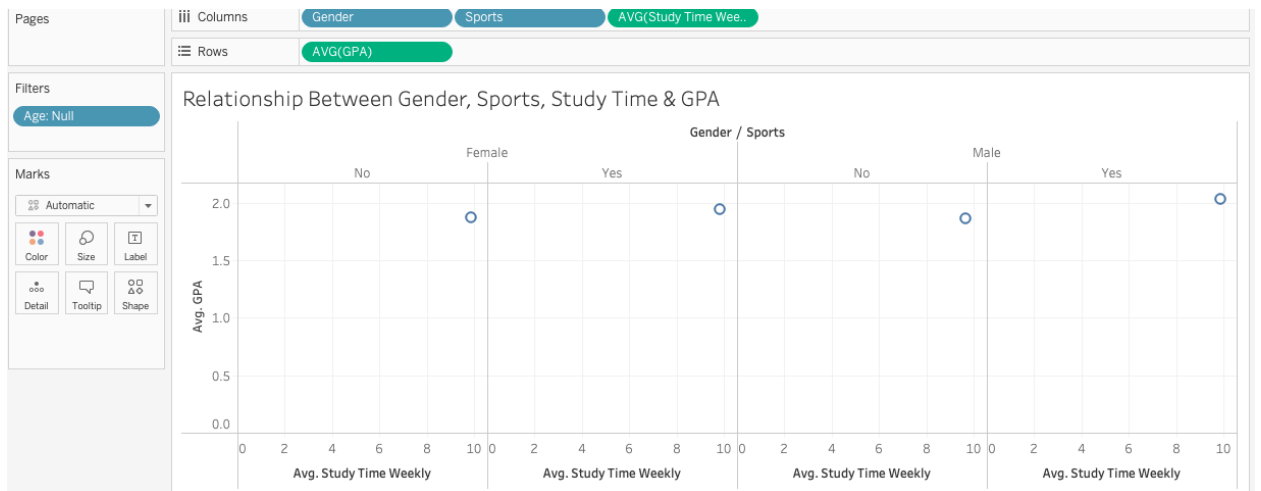


Figure 22. The relationship between gender, sports, weekly study time, and GPA



Figure 23. The relationship between gender and GPA

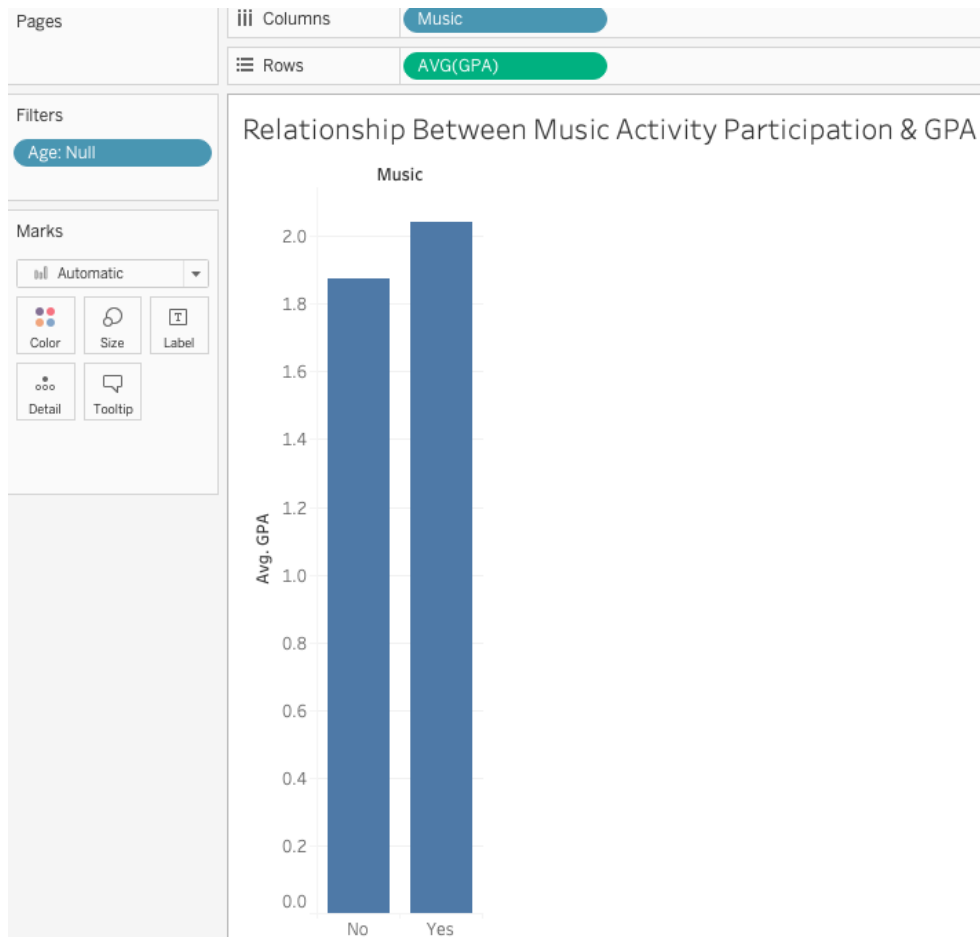


Figure 24. The relationship between music activity participation and GPA

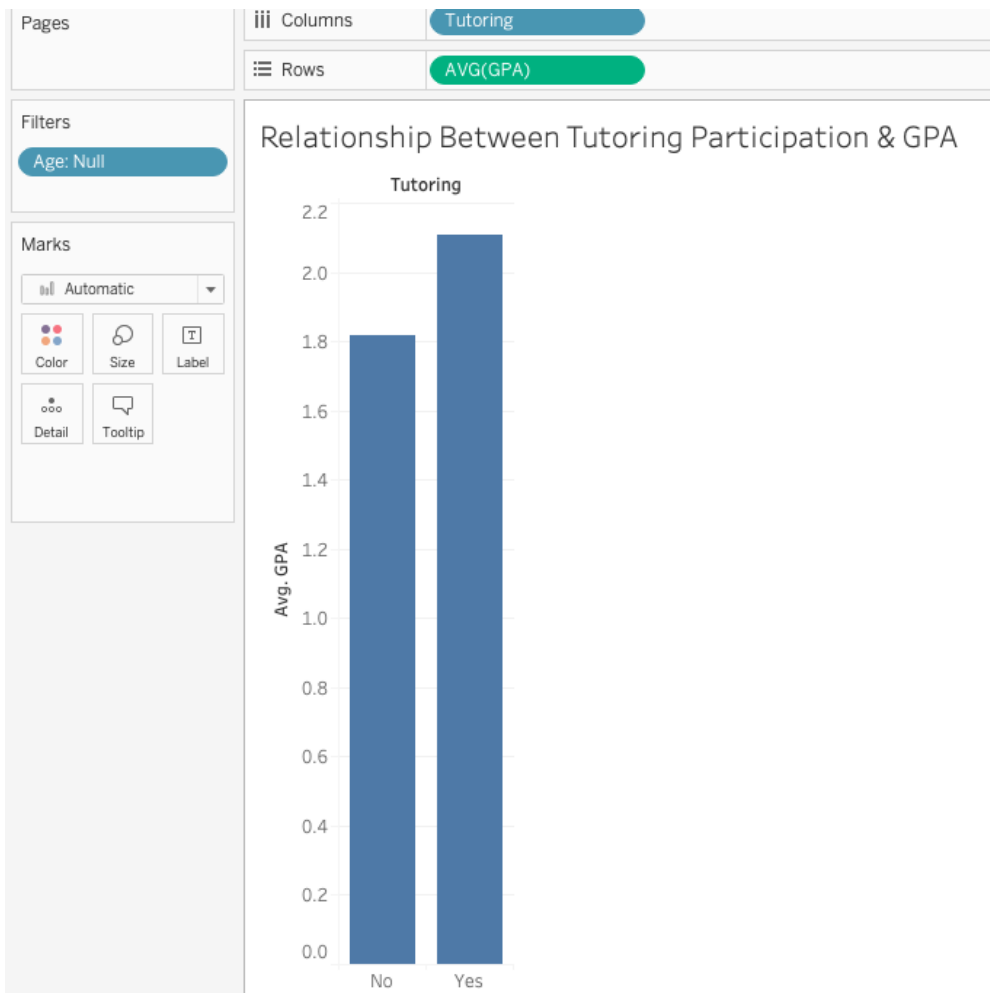


Figure 25. The relationship between tutoring participation and GPA

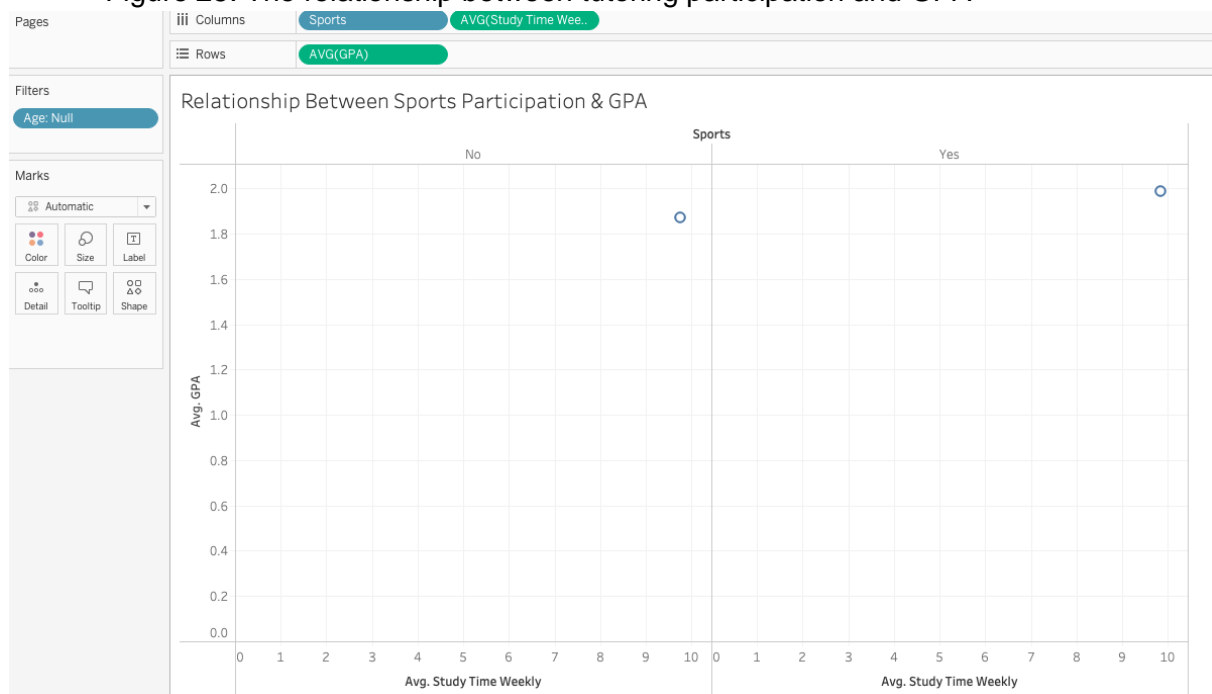


Figure 26. The relationship between sports participation and GPA



Figure 27. The relationship between extracurricular activity participation, weekly study time, and GPA

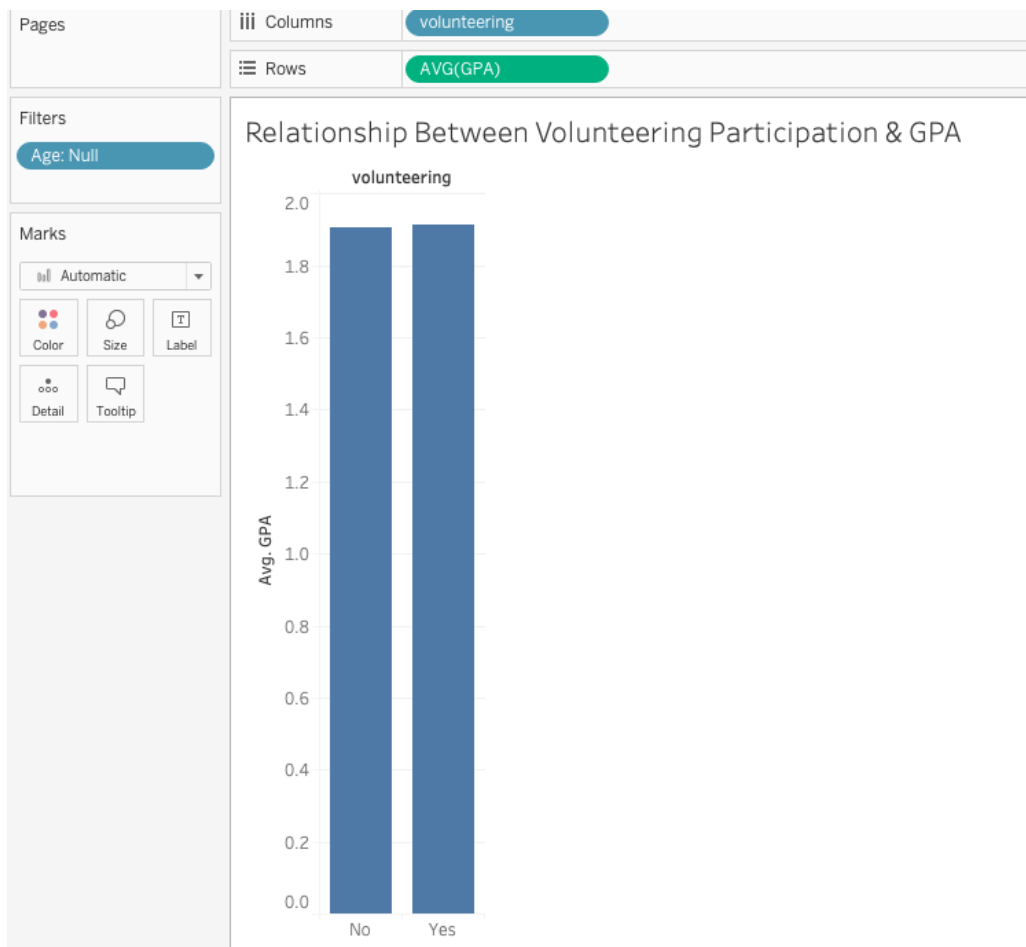


Figure 28. The relationship between volunteering participation and GPA

Result Discussion

Logistic Regression Findings

The results of the evaluation metrics of the logistic regression model when applied to the dataset gave a good accuracy of 80%, precision of 79%, recall of 80% and F1 score of 79%. These evaluation metrics suggest that the model is a good fit however there is still room for improvement.

Tableau Findings

The data visualisation done with Tableau provides many insights on the data. In figure 15, when investigating the relationship between parental education and average GPA, students with parents with high school education have the highest average GPA of 1.94. It is also seen in figure 15 that students with parents with bachelor's education have the lowest average GPA of 1.80. It can be seen in figure 16 that when gender, ethnicity, and average weekly study time are compared, females with other ethnicity and age 18 years have the highest average weekly study time of 11.86 hours. Males with Asian ethnicity and age 17 years have the lowest average weekly study time of 8.56 hours. It can be seen in figure 17 that when parental support is compared to average GPA, students with high parental support have the highest average GPA of 2.19 while students with no parental support have the lowest average GPA of 1.54. This graph shows a clear trend that as parental support increases, average GPA increases. In figure 18 when comparing average absences per year, gender and ethnicity, the results show that Caucasian females have the highest average absences per year of 14.89 days. It is also shown that males of other ethnicity have the least number of average absences per year of 13.32 days. In figure 19 when investigating the relationship between age, ethnicity and absences, overall students with Asian ethnicity and age of 15 years have the highest average absences of 15.72 and students with Asian ethnicity and age of 18 years have the lowest average absences of 12.82. It can also be seen that absences for students with Asian ethnicity decrease as age increases. This trend is not seen among the other ethnicities. Specifically for different age groups for 15-year-old students, Asian ethnicity have the highest average absences, for 16-year-olds, other ethnicity has the highest average absences, for 17-year-olds Caucasians have the highest average absences and for 18-year-olds, Caucasians have the highest average absences. For 15-year-olds, other ethnicity has the lowest average absences, for 16-year-olds African Americans have the lowest average absences, for 17-year-olds other ethnicity have the lowest average absences and for 18-year-olds Asians have the lowest average absences. In figure 20 students with high parental education have the highest average absences of 15.67 and students with high school parental education have the lowest average absences of 14.22. In figure 21 when investigating the relationship between Gender, tutoring, average weekly study time, and GPA male students that participate in tutoring have the highest average GPA however females with tutoring have the highest study time. Also, females with no tutoring have the lowest average GPA while males with no tutoring have the lowest study time. It can be seen in figure 22 that when comparing gender, sports, study time and average GPA, males who do sports have the highest average GPA and study time while males that do not do sports have the lowest

average GPA and study time. In figure 23 when comparing gender and average GPA male students have higher average GPA than female students. In figure 24 when investigating the relationship between music activity participation and average GPA students that participate in music activities have higher average GPA than students who don't. This is the same as in figure 25 with students who participate in tutoring. In figure 26 when investigating the relationship between sports participation, average weekly study time, and average GPA students who participate in sports have higher average weekly study time and average GPA compared to students who don't participate in sports. This trend is also seen with extracurricular activity participation in figure 27. Lastly, in figure 28 when comparing the relationship between volunteering participation and average GPA students who participate in volunteering have higher average GPA compared to students who don't do volunteering. In conclusion, after viewing and analysing all these figures it can be determined that average GPA and class grade increases as music activities, parental support, average weekly study time, tutoring participation, extracurricular participation, or volunteering increases.

Conclusion and Future Works

In conclusion, the results of this paper found that students' class grade increases as music activities, parental support, average weekly study time, tutoring participation, extracurricular participation, or volunteering participation increases. This paper also found that logistic regression can be used to predict students' class grade based on age, gender, ethnicity, parental education, study time weekly, absences, tutoring, parental support, extracurricular, sports, music, volunteering, and GPA. The evaluated logistic regression model was a good fit but may be improved in future work if feature selection is done on the data, normalisation, removing outliers, or applying principal component analysis (PCA). The dataset used has many instances so future works can aim to continue to do research on large datasets to help with result generalisation. Future works can also focus on how to address the factors that impact students' class grade to enhance student performance as this is not covered much in the literature.

Social and Ethical Considerations

A social consideration of this project is that investigating the effects of ethnicity, parental education, or gender on class grade can reinforce stereotypes in society. If it is determined that certain ethnicities or certain students with a specific parental education perform better than others, different expectations may be placed on these students to perform better. This can increase stress on these students as well as increase bias. Another social consideration of this project is that when specific factors such as sports, volunteering or music are found to increase academic performance, schools may spend more to have these activities easily available to students to boost academic performance. This may cause gaps among different populations in society if some schools in a certain area implement these activities and other schools in the same area are not able to due to financial or other reasons. Ethical considerations this project may have are privacy concerns if the students providing data for research can be identified and how the data would be protected.



References

Al-Tameemi, R. A. N., Johnson, C., Gitay, R., Abdel-Salam, A. S. G., Al Hazaa, K., BenSaid, A., & Romanowski, M. H. (2023). Determinants of poor academic performance among undergraduate students—A systematic literature review. *International Journal of Educational Research Open*, 4, 100232.

Ballotpedia. (n.d.). *Academic performance*. https://ballotpedia.org/Academic_performance

IBM. (2021, August 16). *What is logistic regression?* <https://www.ibm.com/topics/logistic-regression>

James, J., Pringle, A., Mourton, S., & Roscoe, C. M. (2023). The effects of physical activity on academic performance in school-aged children: A systematic review. *Children*, 10(6), 1019.

Kharoua, R. E. (2024, June 12).  *students performance dataset*  Kaggle. <https://www.kaggle.com/datasets/rabieelkharoua/students-performance-dataset?resource=download>

Mappadang, A., Khusaini, K., Sinaga, M., & Elizabeth, E. (2022). Academic interest determines the academic performance of undergraduate accounting students: Multinomial logit evidence. *Cogent Business & Management*, 9(1), 2101326.

Tableau. (2024). *What is tableau?* <https://www.tableau.com/why-tableau/what-is-tableau>

Appendix

Code

```
pip install pyspark
pip install findspark
import findspark
findspark.init()
from pyspark.sql import SparkSession
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.classification import LogisticRegression
spark = SparkSession.builder.appName("student performance").getOrCreate()
dataset = spark.read.csv("/content/Student_performance_data _ 2.csv", inferSchema = True,
header = True)
print('There are ',len(dataset.columns),'columns in this dataset')
print('There are ',dataset.count(), 'rows in this dataset')
dataset.show()
assembler = VectorAssembler(inputCols = ['Age', 'Gender', 'Ethnicity', 'ParentalEducation',
'StudyTimeWeekly', 'Absences', 'Tutoring', 'ParentalSupport',
'Extracurricular','Sports','Music','Volunteering','GPA'], outputCol='features')
output = assembler.transform(dataset)
finalised_data = output.select('features', 'GradeClass')
finalised_data.show()
train, test = finalised_data.randomSplit([0.8, 0.2])
log_reg = LogisticRegression(featuresCol = "features", labelCol="GradeClass")
log_regmodel = log_reg.fit(train)
predictions = log_regmodel.transform(test)
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
multi_eval = MulticlassClassificationEvaluator(labelCol="GradeClass",
predictionCol="prediction")
accuracy = multi_eval.evaluate(predictions, {multi_eval.metricName: "accuracy"})
precision = multi_eval.evaluate(predictions, {multi_eval.metricName: "weightedPrecision"})
recall = multi_eval.evaluate(predictions, {multi_eval.metricName: "weightedRecall"})
f1 = multi_eval.evaluate(predictions, {multi_eval.metricName: "f1"})
print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1: {f1:.4f}")
```

