Task 1:

Level 1

When I got the data I noted the three missing features.
I found them out through analysis of various plots and reasoning, both of which can be found in the notebook.

Feature 1 - age
Feature 2 - studytime
Feature 3 - Weekend Alcohol Consumption

Level 2

I found all the columns with null values. I replaced the ones containing categorical data with their mode. Categorical data meaning where order does not matter but frequency does.
For non categorical data such as columns that have numerical data or ranking data where the ordering matters I used the median to replace nan values.

Level 3(EDA):

I asked many questions between many columns results and analysis for which can be found in the notebook. I'll copy paste the results below:

Result:

1.) Higher your age higher the chance of a romantic relationship(you do go through a break up phase at 21 though)

2.) From the boxplot seems like if you are not in relationship you have a chance of getting slightly better grades.(i checked the average as well)

3.) Mothers with low level of education have children with higher possibility of being in a romantic relationship(result is approx same for fathers as well)

4.) If you have plans to pursue higher educaation chances are you are not getting in a relationship

5.) higher extra curriculurs = higher chance of being in a romantic relationship

6.)Hacing internet access again increases your chance of being in a romantic relationship(online dating)

7.)surprisingly people who have rated themselves as have better health on average drink more alcohol

8.)better health leads to less absences from class

9.)a better family relationship leads to beter health

10.) Higher absences leads to lower grade

11.) A better family rlationship leadas to better grades

12.) Higher your age more diversified your grades are. At a young age more students are focused at getting better grades but as your age increases the student pool diversifies as can be seen from the boxplot. Also the median has a slighly decreasing trend

Level 4:

This was a problem of binary classification. I converted all binary columns(yes/no) into 1 and 0. Further all categorical data was One Hot encoded columns i.e. each category in a column gets made into a separate column and if the row has the category the categories cell data is 1 otherwise 0.

I trained on 90% train data and 10% test data
Now we tested two kinds of classifiers:

- Tree Based-
  - Decision Tree
  - Random Forest(Used GridSearchCV to find optimum n_estimators as 150)
  - XGB (Used GridSearchCV to find optimum n_estimators as 50)
- Ones that work with Values-
  - KNN
  - Logistic Regression
  - SVC

For tree based classifiers I did not scale the numerical values.. However for the other I used StandardScaler in the pipeline

Results:

	Model	Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)
0	Logistic Regression	0.630769	0.672414	0.886364	0.764706	0.285714	0.095238	0.142857
1	SVC	0.600000	0.655172	0.863636	0.745098	0.142857	0.047619	0.071429
2	KNN	0.523077	0.632653	0.704545	0.666667	0.187500	0.142857	0.162162
3	Decision Tree	0.584615	0.717949	0.636364	0.674699	0.384615	0.476190	0.425532
4	Random Forest	0.615385	0.666667	0.863636	0.752475	0.250000	0.095238	0.137931
5	XGBoost	0.569231	0.681818	0.681818	0.681818	0.333333	0.333333	0.333333

All of the models tend to lean heavily toward predicting class 0 (students not in a relationship), which shows up in the consistently high precision, recall, and F1 scores for that class. On the flip side, they all struggle with class 1 (students who are in a relationship). Logistic Regression came out on top in terms of overall accuracy (about 63%) and gave the most balanced results, but even then, its F1 score for class 1 was only 0.14 — so it still misses a lot of actual positives. SVC and KNN did even worse, barely catching any of the class 1 cases (recalls of just 0.047 and 0.14), which probably means they're skewed by the class imbalance and just playing it safe by defaulting to class 0. The Decision Tree did a bit better at picking up class 1, with an F1 of 0.42 — not great, but relatively the best — though that came at the cost of lower performance on class 0 and overall accuracy. Random Forest, while accurate on class 0, basically ignored class 1 (F1 of 0.13), which is typical when ensembles like this are influenced by majority voting. XGBoost was a bit different — it had fairly balanced recall for both classes (about 0.68 each), but its F1 for class 1 was still low (0.33), suggesting it predicted more positives but a lot of them were wrong. The neural network, despite its flexibility, actually performed the worst on class 1 — with precision, recall, and F1 all well below acceptable levels. It mostly learned to predict class 0, likely due to the imbalance in the data and lack of sufficient examples for class 1 to learn meaningful patterns. Overall, class 1 performance is clearly a weak point for all the models.

I also tried running a neural network and tested various activation functions.. But it usually led to overfitting so it was no good.

Level 5:
I chose G3 and age as the two columns. I have plotted all the decision boundaries, SHAP summary tables and waterfall plots for all models in the notebook.

Bonus Task:
PLOT 1: This looks like a tree based classifier due to the rectangles. I think it is XGBoost because of the granularity of the rectangles
PLOT 2: This looks like overlap of many rectangles so i think an ensemble of many decision trees that is random forest
PLOT 3: I think it is support vector
PLOT 4: It is support vector in my opinion as it smooth curve and looks like the one i got above
PLOT 5: This is a KNN it was jagged

These plots could have used other classifiers as well but the classifier I mentioned also might have produced similar plots

Task 2:

Level 1:
I used Gemini 2.0 flash
I started with testing various methods one could apply. I was creating different agents and one was using the same agent but with different tools.

For level 1 I created different agents while experimenting. 1 was calculator and 1 was for general chatbot. It was slightly bad because I did not use classifier to route to these agents. I just used a simple classification that if the message only contained operators and numbers go with the calculator agent. I fixed this in later levels though

The visualizations and all are included in the code file.

Level 2:
This time I created three tools: one for calculator, one for weather, one for fashion trends.
For calculator same logic as above but this time the llm could convert say “Tell what is 2 times 3” to 2*3 and use this tool as I mentioned what arguments the tool took.

For the weather api I used openweathermap version 2.5 though even if it has been depreciated we can put the argument as city rather than coordinates. I made this tool and got data and returned some things about the weather.

Fashion Trends was interesting as there was no specific API I could use. I searched through integrations in the Langchan documentations and found a free search tool Tavily. I used it as a tool and gave specifications in the prompt, how to find fashion trends. This search could properly be used in Level 3

Level 3:

I designed a very nice prompt which made a very good judgement for when to use which tool. The prompt can be checked out in the final file. Used Chat prompt template for this and also included a place where I can give it chat history so that it has memory.

I created an agent node.. If the last message was a human's message it invokes it.. It also stores all the chats history(messages sent by agent) in a variable which can be passed in the prompt for memory.

I then built the final graph whose visualization can be found in the final notebook.

The chatbot works like a charm, remember previous conversation and gives accurate results using necessary tools(i checked it using verbose=True during development stage)

To stop conversing with the bot you can simply write exit and it stops.

Level 4:

Unfortunately, I could not do this task. I started a little late with the recruitment process(25th afternoon) as I was busy earlier so could not really do this. Task 2 seemed really fun though and I will keep experimenting with this. Sorry if the report seemed a little haphazard, I wrote it very quickly.