Keshav Goyal(240101043)

Task-1:

We started analyzing the DataFrame after by first taking the raw dataframe and separating it by dropping the rows having nan values. We use the new dataframe for finding the correlation between all the columns by considering the Label Encoding. Now we start the different Levels given in the Problem Statement:

Level-I :

We find the correlation between the Feature columns and the other columns to find out what might be indicating. For each feature we made some plots like violin plots, heatmaps and histograms using Python library Plotly. We could also have used the Seaborn library for the same but since Plotly provides more user interactive graphs that was the reason we chose Plotly. We found the Correlation of Feature_1, Feature_2 and Feature_3:

- Feature_1 : It had strong negative correlation with many of the academic feature available to us and positive correlation with many features like guardian other than parents, etc. So we can deduce from there that it might be something that indicates that the Feature might be some index measuring the expected failure.
- Feature_2: It had strong correlation with the grades. So it is something like Academic Performance index.
- Feature_3: It had very strong correlation with the Dalc column indicating that on increasing the Alcohol consumption the Feature_3 increases indicates that Feature_3 is some index that measures the bad habits of a student. So we say it is Bad Habits index.

Level-II:

Now our target is to clean the dataset and make EDA. For imputation we have 2 options that either, we make a direct replacement of the na values by mode of the column or fill them according to the correlation between the other columns. Since all of the columns are classifiers so we hesitate to use any kind of line plot or bar plot.

- For the mode imputation, we can do this directly by importing the SimpleImputer from the sklearn library.
- For the other imputation we analyze the correlation matrix of the dataset and take the top 5 columns each time to impute the dataset. So we make a function that tells us the top 5 columns and make another function and return a dataframe.

But before the imputation we made some relationships of some columns with each other. The relationships were made in order to justify the correlations between them.

So we made a lot of violin plots and box plots inside of them.

Level-III:

Now it is time to make some questions ourselves for insights:

- Tell the relationship between Mjob and Fjob? The answer is it was kind of a linear relationship but there were only lines in the box plot with some outliers.
- Find the percentage of Students that are from GP school and are Female and are in romantic relationship.

Level-IV:

Now is the moment of truth, all the data cleaning and feature engineering was for this point for training the data. We split the data in 80:20 ratio. We train out data on the larger one and test on the smaller. But on testing the accuracy on the smaller, it was still less than the dumb model that predicted No every time. So we tried to use different models like the Logistic Regression model, Random-Forest Classifier and also the Logistic Regression Cross Validation model for reducing the potential multi collinearity between the columns. But still no good happened. Also we tried to hypertune the Random Forest but still no hope. So here we are with the maximum accuracy of 0.61 less than the Dumb model that always predicts a No.

Also we used the One Hot Encoder to encode the categorical columns I.e., all the columns.

Level-V:

We made a summary plot for the Logisitc Regression plot and found out that Grade 2 was the most responsible feature for the prediction. Then we created the force plot for a ‘Yes’ value and for a ‘No’ value

Task-2

Level-I:

We construct a Langgraph chatbot by making an API key for gemini and importing it to the system environment using the os library. We define a state for the graph and then we start building it through stategraph function in the node. Then we created a function that evaluated the string we input and used it as a toolnode and added it to the llm we imported. This finishes our Level-I

Level-II:

We construct a basic Chatbot and the add the following tools to the llm:

- Web Search: Create a Tavily API key and add it to the environment variable and put some basic queries like “What’s trending in fashion”.
- Weather: Make Open Weather Search API key and make a basic tool using that returning the weather of a city.

Now add both the tools to the main LLM.

Level-III: