Coding Week: Machine Learning

Task 1: The CampusePulse Initiative

"Data is the heartbeat of understanding, and in The CampusPulse Initiative, I listened closely to uncover the hidden associations of student life, predicting relationship status."

Task 1 is divided and solved into several levels as a step-by-step procedure to get insights from the data and create a model which can accurately predict the relationship status.

Let us dive into the code and understand the thoughts and the intuition behind the process

Level 0: Dataset Information and Getting Tools Ready

- Loaded the dataset (Dataset.csv) and libraries (pandas, numpy, matplotlib, seaborn, sklearn, shap).
- Set random seed (np.random.seed(0)) for reproducibility.
- Displayed initial rows and used info() to summarize columns and data types.

Level 1: Variable Identification Protocol

- Firstly, I got a statistical summary of all the features in the dataset. Feature_1 had more range than both Feature_2 and Feature_3
- I also found out the distribution of the hidden features by Histogram.
- Since, we are working with student data Feature_1 would be **age** because it followed a right tail distribution with maximum number of values in the range 15-19.
- Feature_2 and Feature_3 are categorical values which can encode a number of possible features - stress **levels, screen time, study time, GPA, etc**.
- Since Feature_2 is negatively correlated with average failures and positively correlated with average grade, It can be a positive attribute maybe **study time/GPA.**
- Since, grades increase linearly with Feature_2 but then decrease slightly it cannot be GPA and thus it is **studytime.**
- Similarly, grades decrease linearly with Feature_2 and increase with failures, number of absences. It is a negative attribute and can be **stress levels.**
- These features can be further checked in **Level 3**

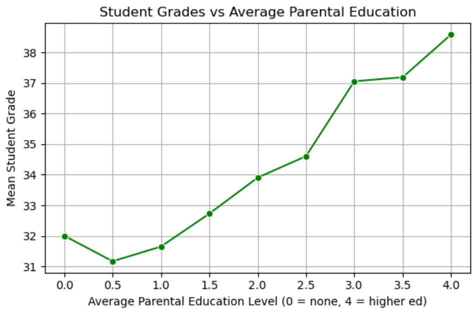
Level 2: Data Integrity Audit

- I first checked the number of missing rows in categorical columns using value_counts()
- I filled the “famsize” with mode since it represents the data correctly while for “higher” I maintained the proportion in the data by the target column and filled accordingly.
- Further, the categorical values which were encoded are filled by the mode of the data.
- While the continuous values are filled with median/mean.

Level 3: Exploratory Insight Report

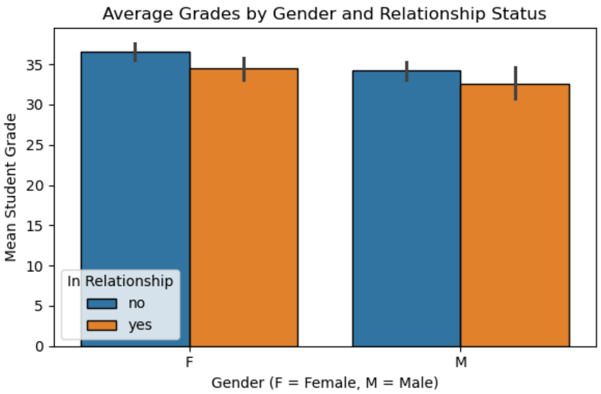
Answered these 5 questions based on the dataset using seaborn and matplotlib

1. Does **education of parents** affect the **grades** of the student?



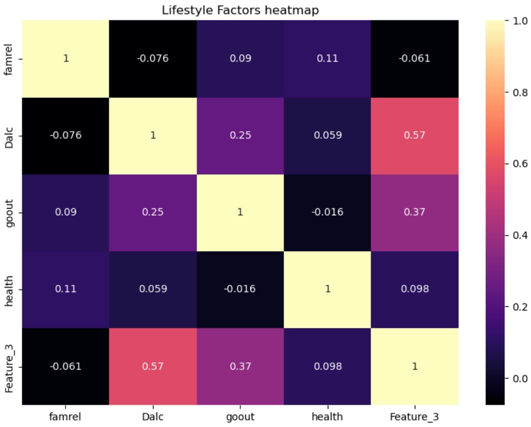
This plot clearly shows that higher average parental education is associated with better student grades. This trend is likely due to several factors like better academic support at home, higher expectations, and access to resources. The strongest gains appear after an average parental education of 2.5, pointing to the fact that educational performance increases substantially when both parents are well-educated.

2. How does **relationship status** and **sex** affect the grades of students?



- The chart highlights a consistent pattern: students who are not in a romantic relationship tend to perform better academically. The reason for this can be more time is spent to maintain the relation and thus less time is available for academic activities.
- Female students outperform male students regardless of their relationship status.
- The negative impact of relation on grades affects both the gender nearly the same.

3. How **are famrel, Dalc, goout, health, Feature_3(stress level)** associated with each other?



- Dalc vs Feature_3 (Stress level) -> 0.57

This is a strong relationship. Students who drink more alcohol on weekdays also tend to report high amount of stress.

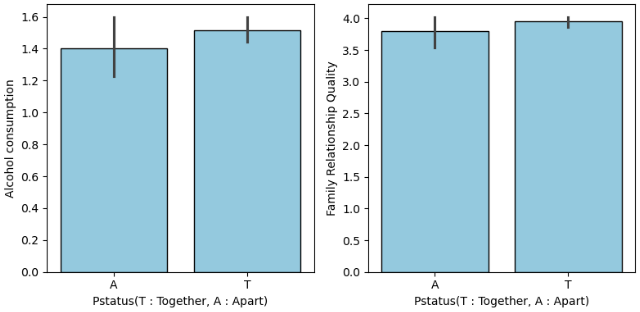
- Dalc vs goout -> 0.25

Moderate correlation. Students who socialize more often tend to drink more during the week — this is expected as a *party culture*.

- goout vs Feature_3 -> 0.37

Socially active students are also more likely to report high stress levels, reinforcing the idea of *peer pressure and fitting into the society mindset*.

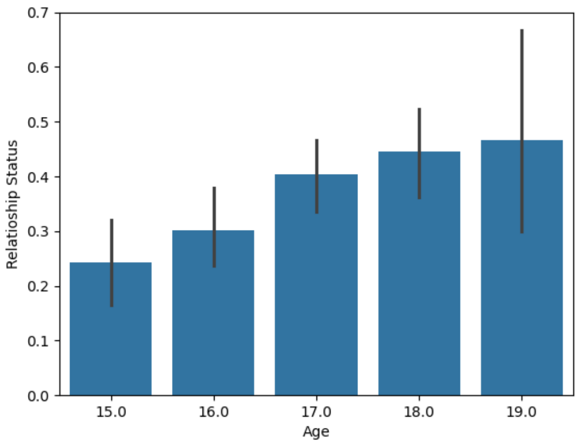
4. How does Parental cohabitation status affect student's Alcohol consumption and Family relationship quality?



- Alcohol consumption of students whose parent live together is greater than the alcohol consumption of students whose parents live apart. The reason for this is that student living with a single parent are more protected and often have less freedom.

- Students with both parents together report slightly higher family relationship quality than those with parents apart. Students with both parents have greater emotional support and family bonding.

5. Relation between Age and Relationship Status?



The bar chart indicates a strong positive relationship between age and possibility of being in a romantic relationship among students aged 15 to 19. The trend suggests that emotional and social maturity with age plays a key role in relationship formation. Students with increasing age also have greater freedom.

Level 4: Relationship Prediction Model

- I dropped a few columns which I created for **EDA** purpose.
- Mapped the categorical columns with map() function.
- Scaled the data for KNN and Logistic Regression, then split the data into testing and training data. Also used "stratify=y" to maintain equal ratio in testing and training data.
- I implemented different models and compared their "f1-scores" to get the best model
- **Logistic Regression**

```
                     precision        recall      f1-score      support
0                         0.68          0.93          0.79          123
1                         0.70          0.26          0.38           72
accuracy                                              0.69          195
macro avg                 0.69          0.69          0.59          195
weighted avg              0.69          0.69          0.64          195
```

- **Random Forest**

```
                     precision        recall      f1-score       support

0                         0.65          0.75          0.79          123
1                         0.70          0.26          0.38           72
accuracy                                              0.69          195
macro avg                 0.69          0.69          0.59          195
weighted avg              0.69          0.69          0.54          195
```

- **KNN**

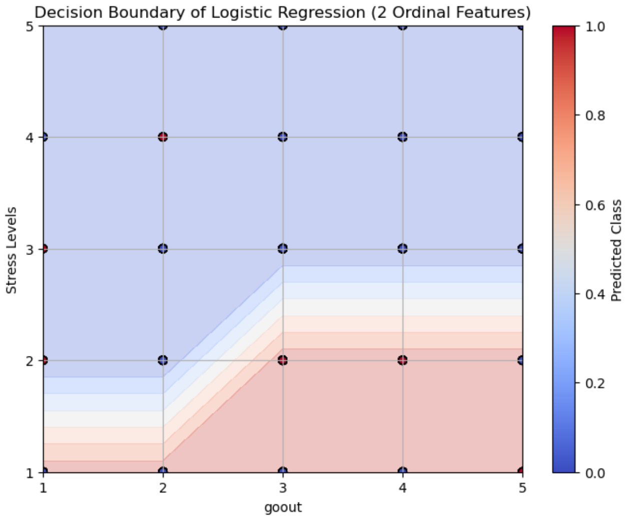
```
              precision      recall      f1-score      support
0                0.65         0.75         0.79          123
1                0.70         0.26         0.38           72
accuracy                                   0.69          195
macro avg        0.69         0.69         0.59          195
weighted avg     0.69         0.69         0.54          195
```

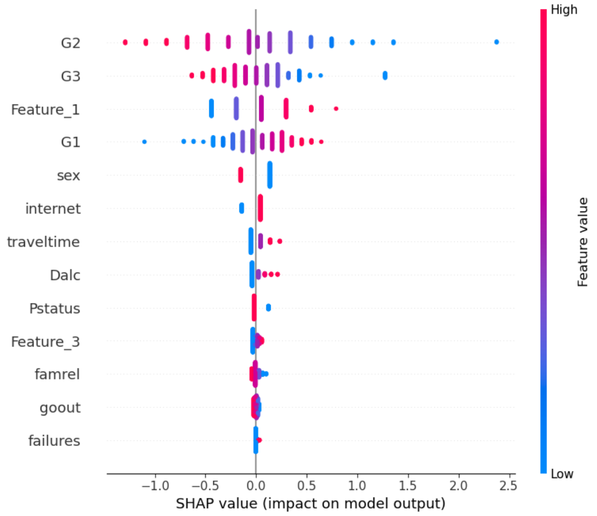
- **XGBoost**

```
              precision      recall      f1-score      support
0                   0.65         0.75         0.79          123
1                   0.70         0.26         0.38           72
accuracy                                      0.69          195
macro avg           0.69         0.69         0.59          195
weighted avg        0.69         0.69         0.54          195
```

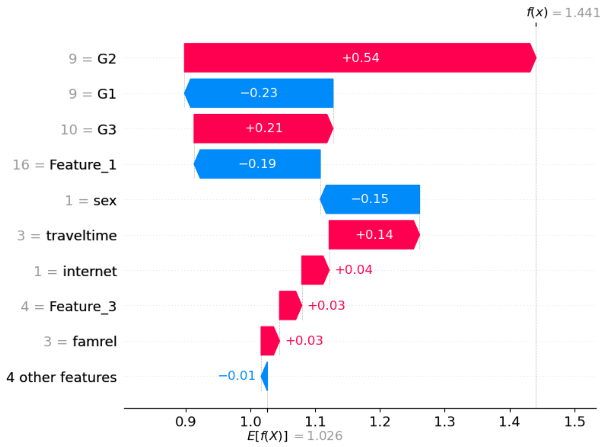
Level 5: Model Reasoning & Interpretation(Logistic Regression)

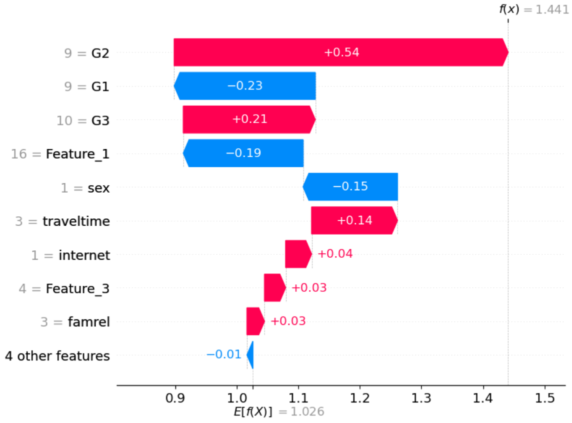
- Plot the decision boundary using meshgrid() . I use this resource for the same.





- Then using shap, plot a beeswarm plot. The beeswarm plot will highlight the important features which will shape the model prediction.
- The important features for the case of Logistic Regression are – G2, G3, age, G1, sex, internet, traveltime and so on.
- The color bar shows the value of the feature while the x-axis shows the impact of the features value on the prediction.
- Plot waterfall plots for two labels, one predicted yes and one predicted no.





SHAP Value Interpretation Report

Model: Logistic Regression

Analyze the predictions made by Logistic Regression model using **SHAP**, to understand how each feature contributes to the model's decisions.

SHAP Beeswarm Plot — Global Feature Importance

Rank	Feature	Interpretation
1	G2	More G2 -> NO
2	G3	Higher G3 -> NO
3	Feature_1	Higher Age -> YES
4	sex	male -> NO
5	G1	Higher G1 -> YES

SHAP Waterfall Plot — Local Explanation (YES)

`f(x) = 3.502` -> **Yes**

Feature	Value	SHAP Contribution
`G3`	15	+1.27
`G2`	14	+1.15
`Feature_1`	18	+0.30
`traveltime`	2	+0.04
`famrel`	5	+0.04
`internet`	1	+0.04
`Dalc`	1	-0.04
`G1`	8	-0.33

Interpretation:
The student is predicted **Yes** largely because he has good grades and has a good age to be in a relationship.

SHAP Waterfall Plot — Local Explanation (NO)

`f(x) = 1.441` -> **No**

Feature	Value	SHAP Contribution
`G1`	9	-0.23

Feature	Value	SHAP Contribution
Feature_1	16	-0.19
sex	Male	-0.15
G2	9	+0.54
G3	10	+0.21
traveltime	3	-0.14

Interpretation:
This student is predicted **No**, mainly beacause of G1, age and sex. Although his G3 and G2 have positive contributions, they are outweighed by negative factors.

Conclusion

- **Global SHAP** analysis shows that activities, sex, age and attendance drive most predictions, which is trivial.
- **Local SHAP** helps to understand why student was predicted "Yes" or "No".

Bonus Level: Decision Boundary

The Bonus Task is attached in the notebook itself in notebook section “Bonus Level”.

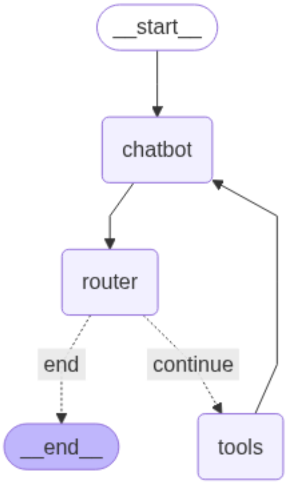
Task 2: The Rise Of Weathermind

""WeatherMind: Where a symphony of AI agents transforms your questions into a whirlwind of witty, weather-savvy, and stylishly informed answers!"

Level 1: Core Activation

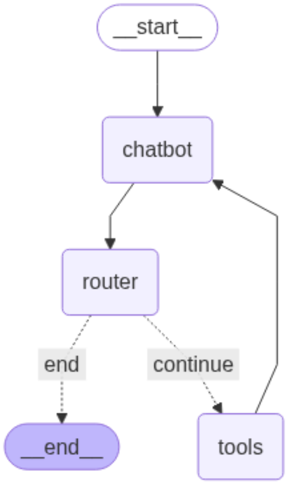
- The first step was to get the "gemini api”. I first learned about the state and graphs and created some graphs for practice while following freecodecamp tutorial.

- Then read the langgraph documentation to create my tool and integrate it with llm.
- Here is my neural network for Level 1



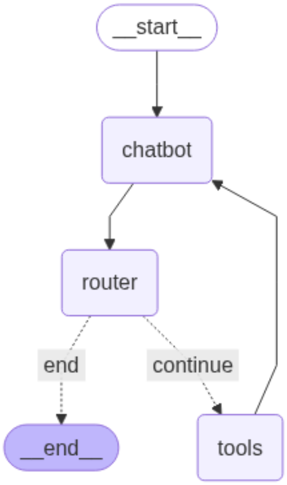
Level 2: Senses of the World

- I just made two new tools *weather_extractor_tool* and *fashion_recommender_tool* in this level*.*
- The *weather_extractor_tool* takes input a location which is fed by the llm after parsing the user query.
- The geocoding api requests the latitude and longitude of the location and then again call the open map api to fetch the current temperature and short summary of weather description.
- The tool then feeds the data to the llm which displays it to the user by making it witty.
- Similarly the *fashion_recommender_tool* takes the location from the llm, uses the tavily search at the backend and then return top three result. The llm then parses the result.
- The neural network is same as the previous level.



Level 3: Judgement and Memory

- The agent already had a router logic so no need to implement it.
- To equip the agent with memory, I used "MemorySaver()" and initialize the agent with "checkpointer=memory". The neural network was same as previous.

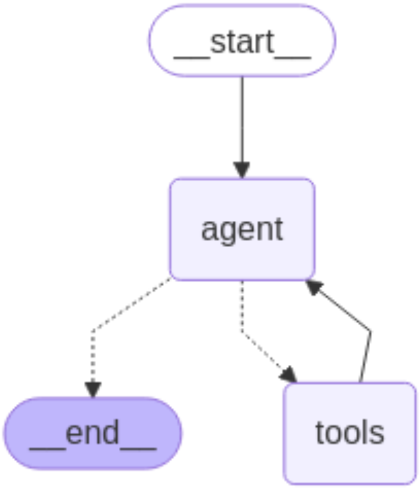


- I also incorporated a conversation type input and output to use conversation history.

Level 4: The Architect's Trial – Multi-Agent Evolution

The level 4 was the hardest level in all the levels of task 2. I went through a ton of **medium article.** The links I have provided in the resources section of task 2.

- I changed the state definition a bit to incorporate smart agent calling by the *decider_agent.*
- Then, I created four different agents with their own tools and a brief prompt.
    - weather_agent - handles weather query
    - fashion_agent – handles fashion query
    - calculator – handles calculations
    - researcher_agent – handles research queries on any topic which are not in the domain of these three agents and give a brief answer.
- Create function to call these agents which will be called by the *decision_maker agent* wrapped into the tools*.* Finally, *decision_maker agent* was created to call the different functions which would invoke the respective agents using tools.
- All the agents were given their respective memories so they can conversation data.
- Here is the neural network :



- This neutral network is same for all the agents where the tools are as follows:
- *decision_maker agent -> weather_tool ->call_weather_agent->weather_extractor_tool*

  ->calculator_tool->call_calculation_agent->calculator

  ->fashion_tool->call_fashion_agent->fashion_recommender_tool

  ->researcher_tool->researcher_agent->researcher_tool
- The *decision_maker agent* can also use **two or more agents** to give an answer to the query.
- The full langraph model at each stage is fully scalable and very easy to maintain.

Conclusion

The Coding Week tasks provided a rich learning experience in machine learning and AI system design. Task 1 honed skills in data analysis, model building, and interpretation. Since, I was already well educated about the tools used in Task 1 it gave me a lot of practice and experience while Task 2 demonstrated the creation of an interactive, multi-agent system. Together, they showcase a versatile skill set in AI and data science.

Thank you,
Manthankumar Bagade