Leveraging Large Language Models for Organizational Decision-Making

Daniel G. Barstow

Department of Computer Science and Information Systems

Elmhurst University

Dr. Jim Kulich

18 July 2024

Executive Summary

This project, conducted as part of the 2024 Society for Industrial and Organizational Psychology (SIOP) machine learning competition (where it placed 8th out of 19 participating teams), explores the application of advanced machine learning techniques within the domain of industrial/organizational psychology. This study incorporates large language models (LLMs) to tackle four critical challenges: detecting empathy in emails, evaluating the clarity of personality inventory items, determining perceived fairness between two workplace policies, and generating plausible interview responses based on previous question-answer pairs. By merging the analytical capabilities of generative AI with a deep understanding of human psychology, this project aims to enhance decision-making processes and improve workplace dynamics. The findings underscore the potential of machine learning to offer innovative tools to foster a more empathetic, fair, and effective work environment.

Table of Contents

List of Tables and Figures

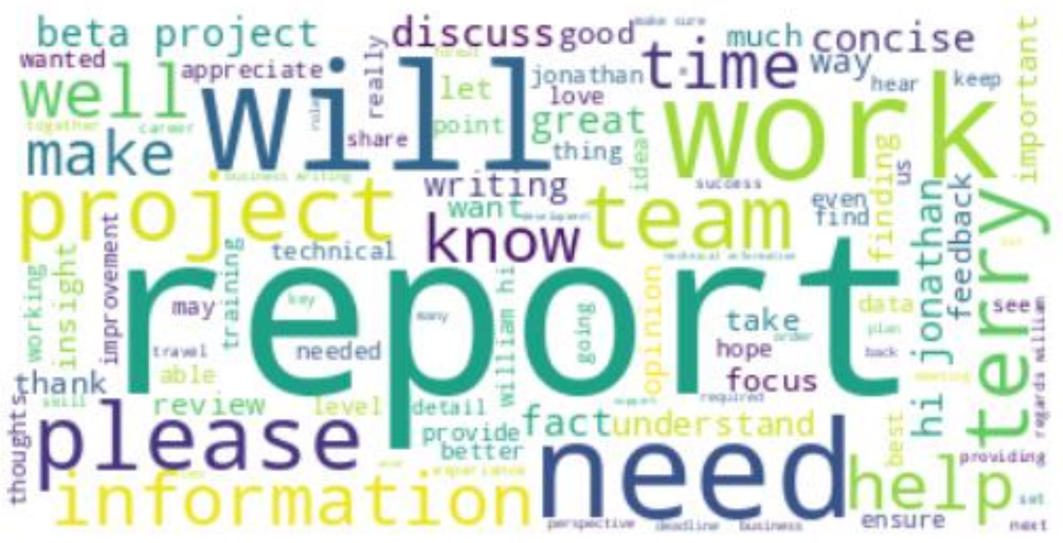


Figure 1 – A word cloud of the `text` column in empathy_train.csv.

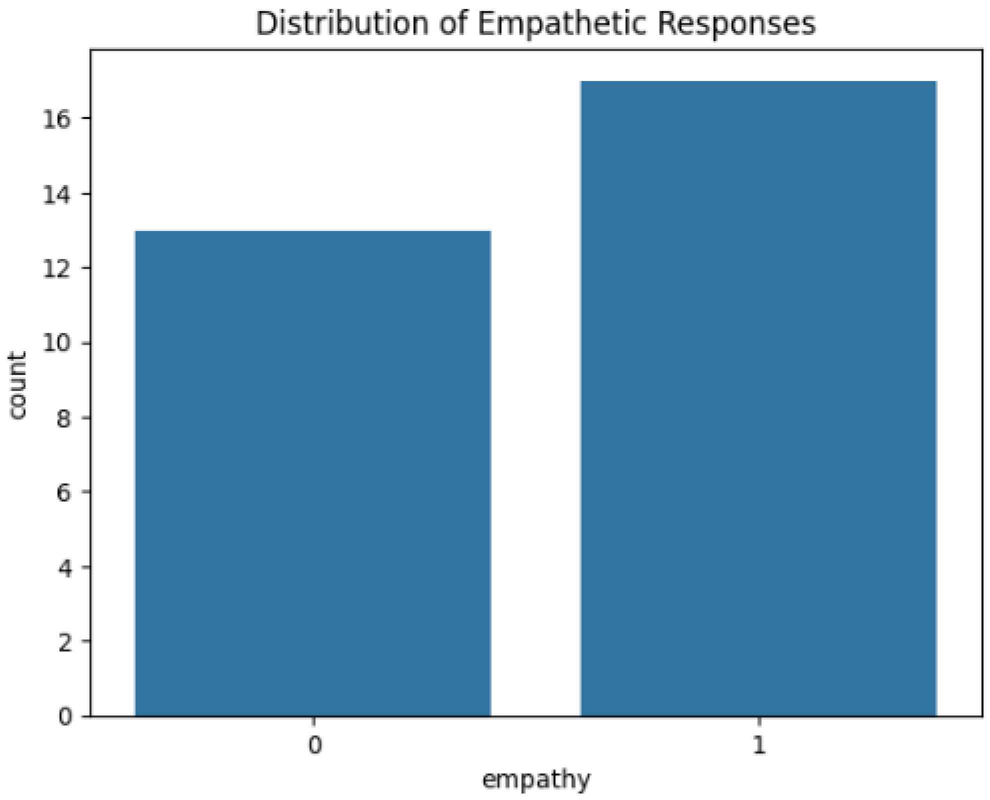


Figure 2 – The distribution of empathetic responses in empathy_train.csv.

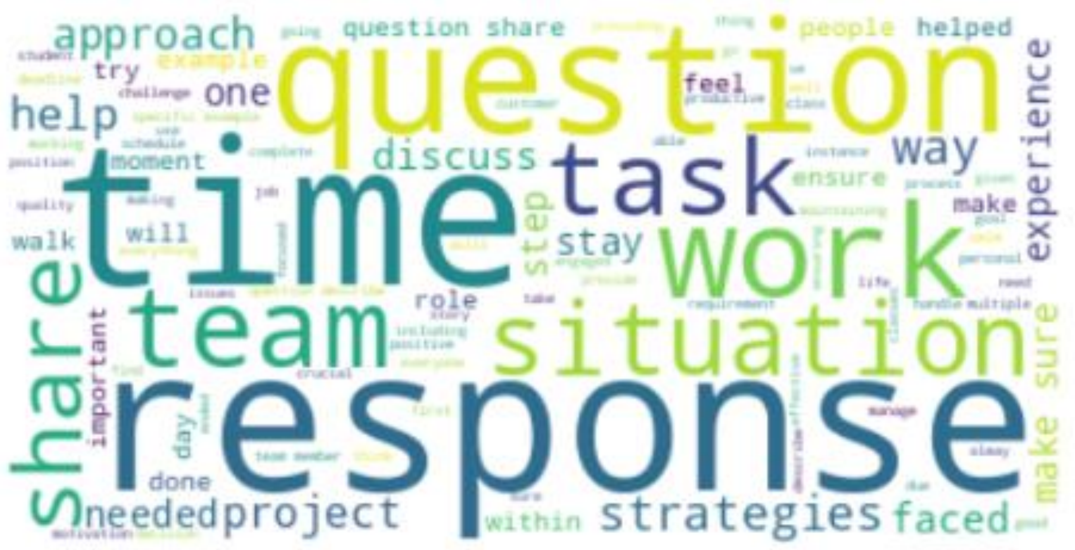


Figure 3 – A word cloud of the `questions_answers` column in interview_train.csv.

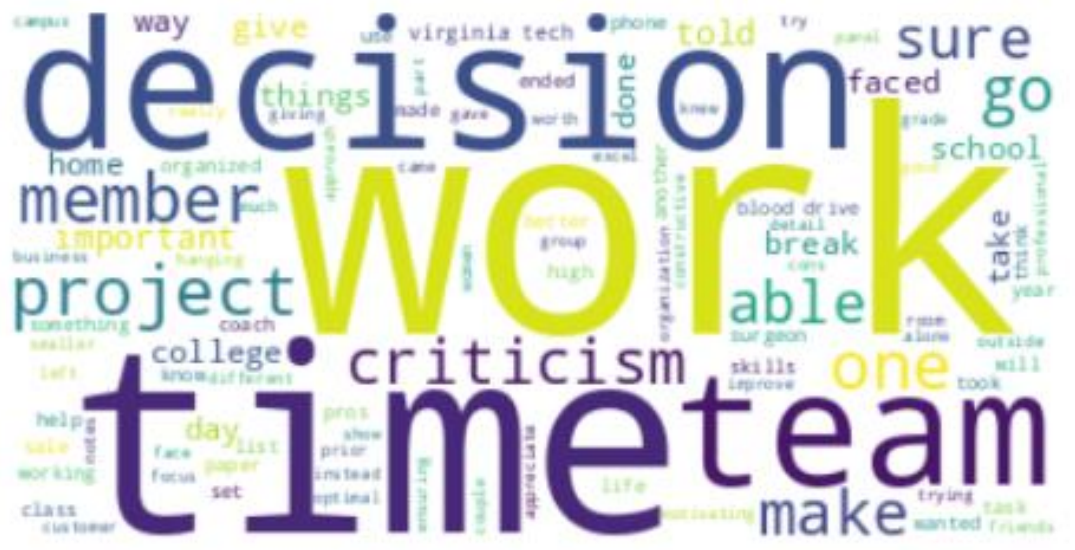


Figure 4 – A word cloud of the `last_response` column in interview_train.csv.

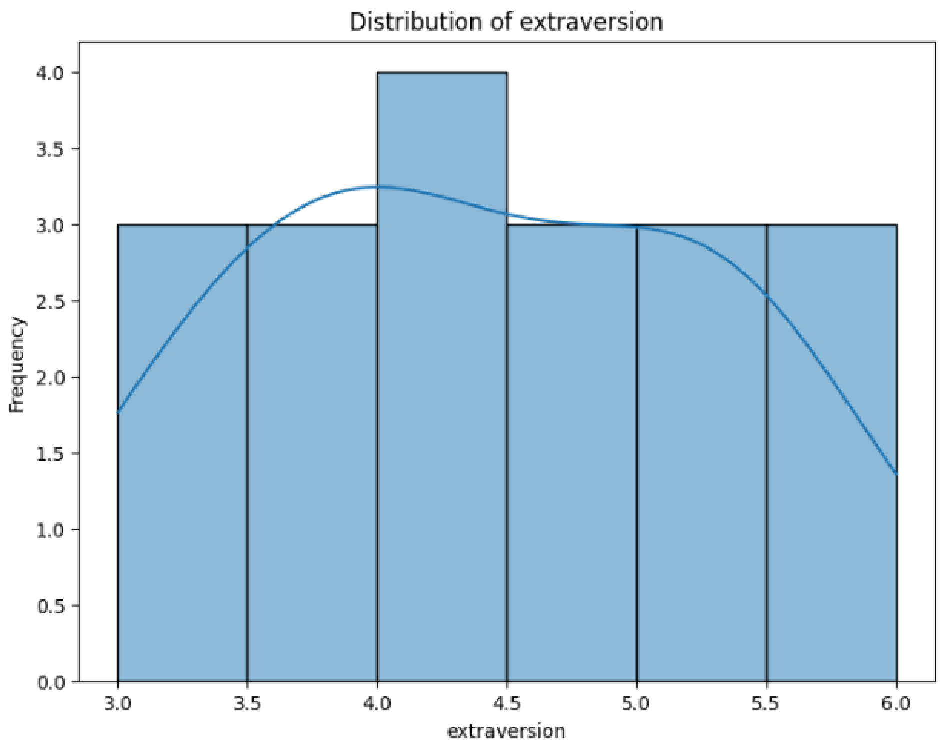


Figure 5 – The distribution of extraversion in personality_train.csv.

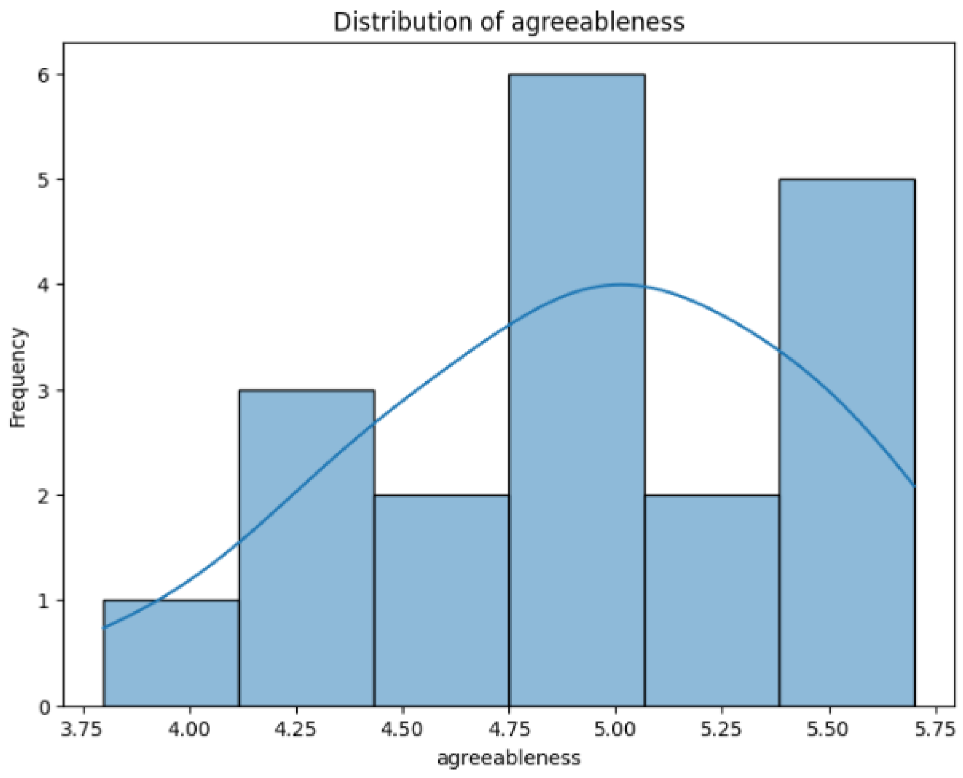


Figure 6 - The distribution of agreeableness in personality_train.csv.

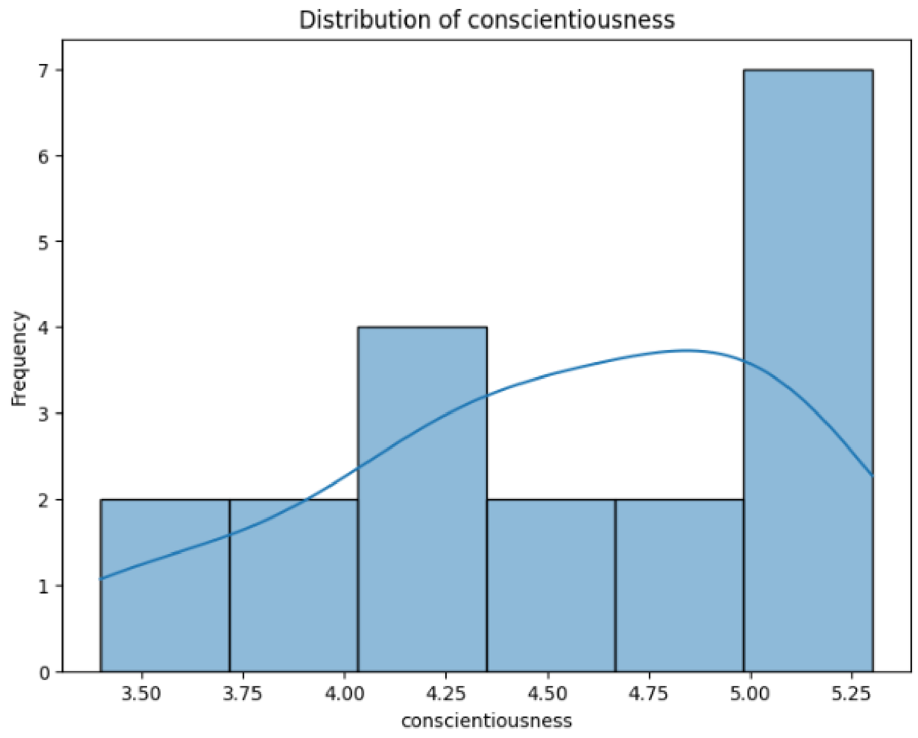


Figure 7 – The distribution of conscientiousness in personality_train.csv.

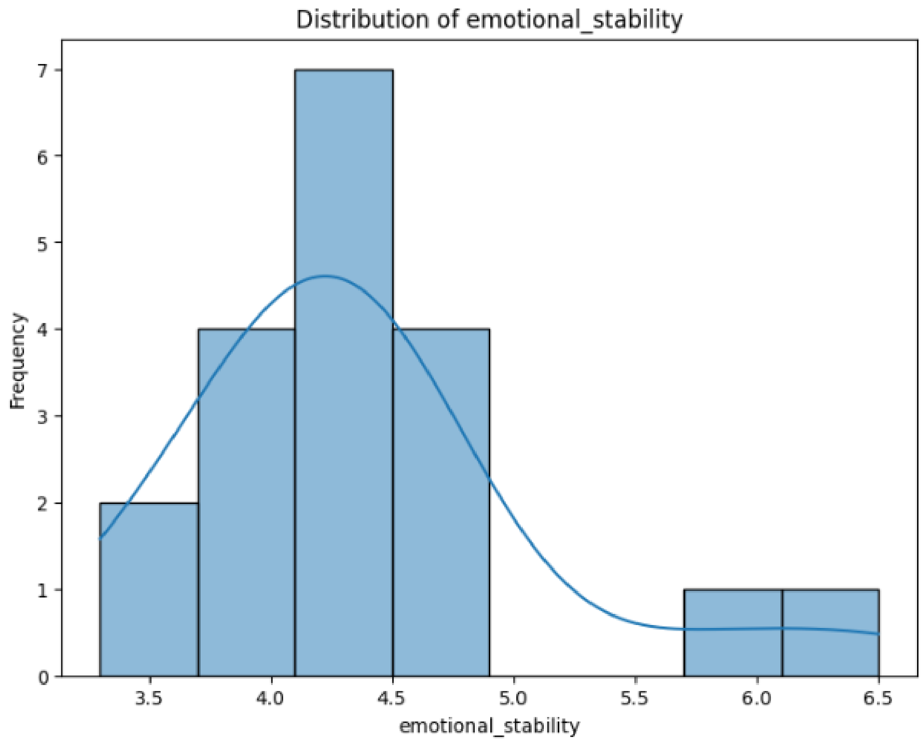


Figure 8 – The distribution of emotional_stability in personality_train.cv.

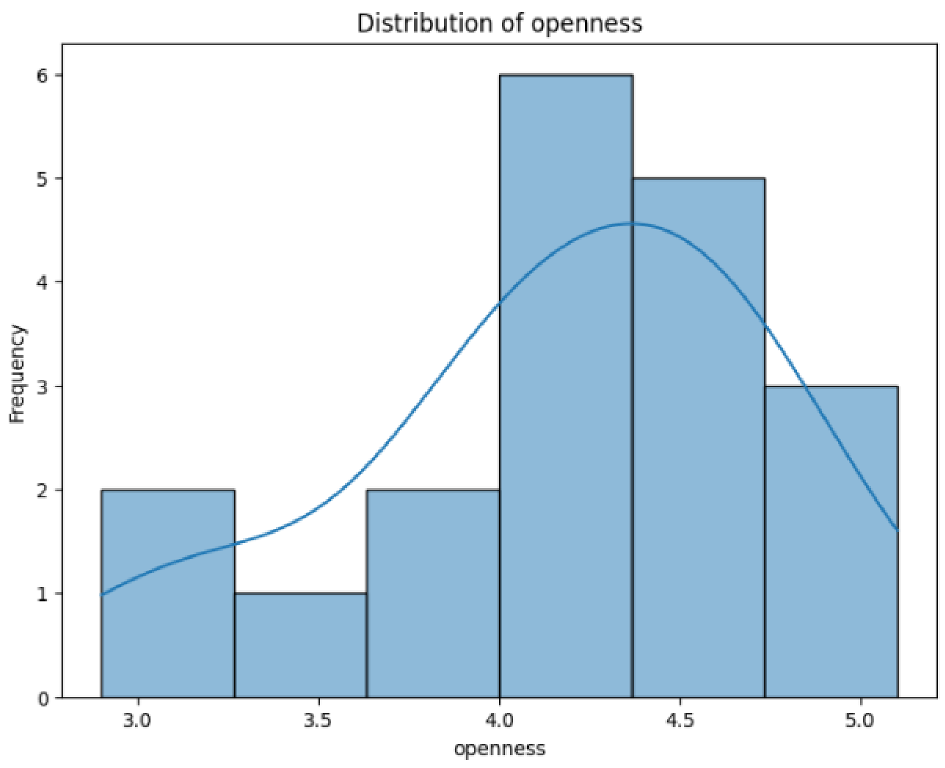


Figure 9 – The distribution of openness in personality_train.csv.

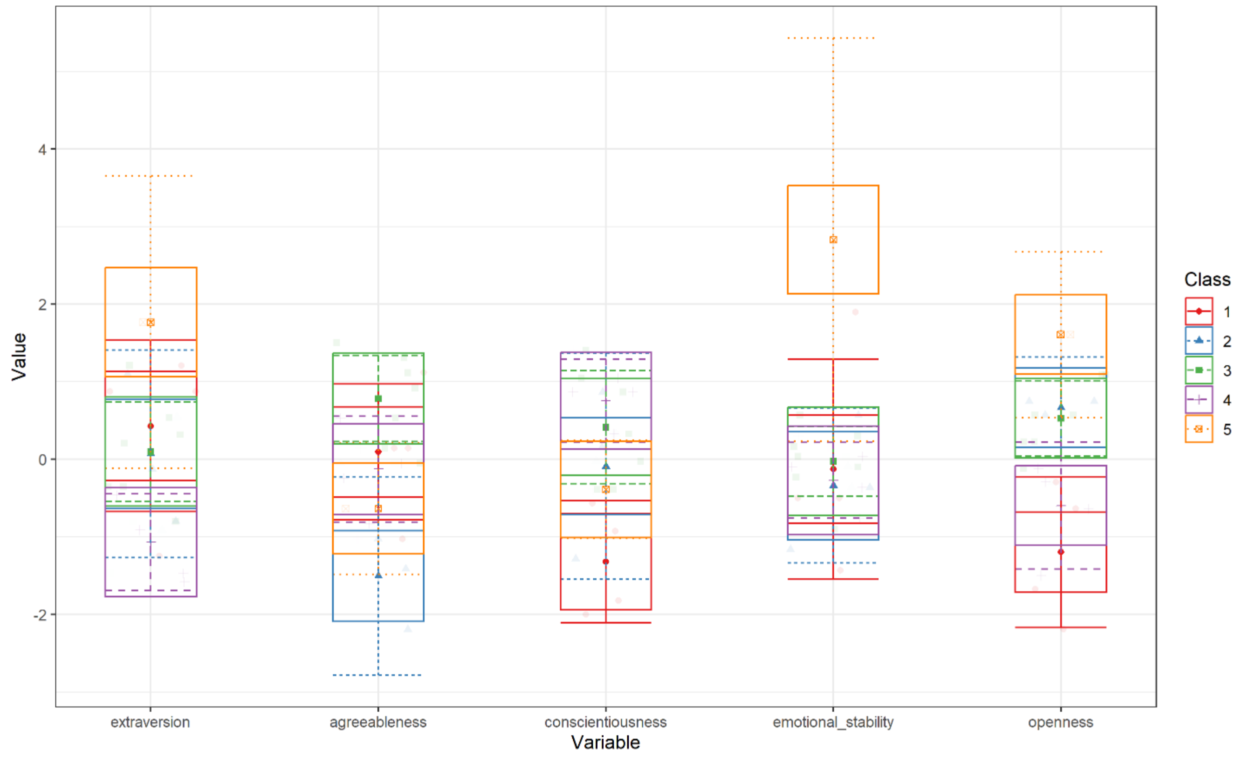


Figure 10 – Latent Profile Analysis (LPA) with 5 classes of the data in personality_train.csv.

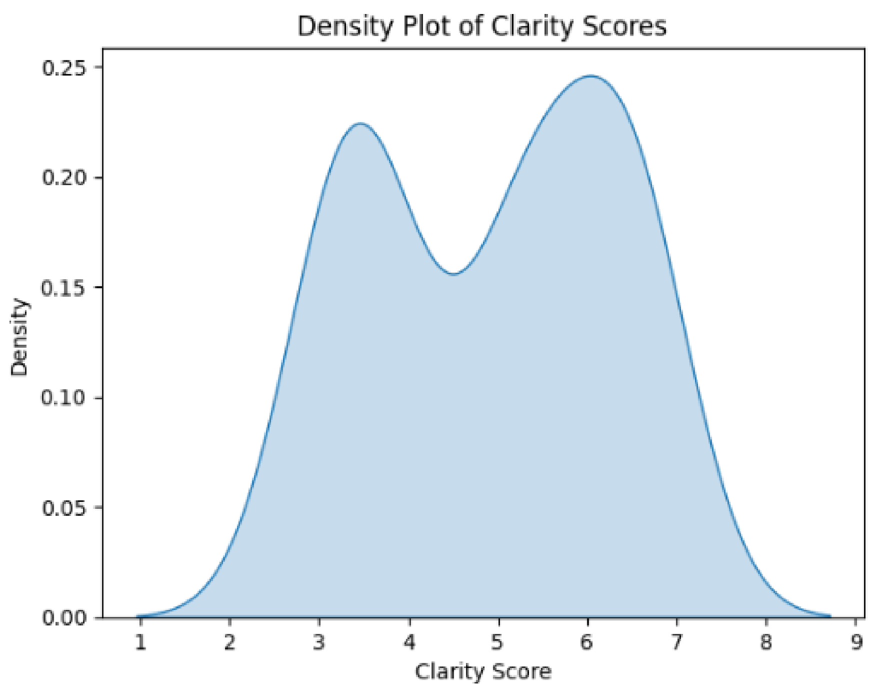
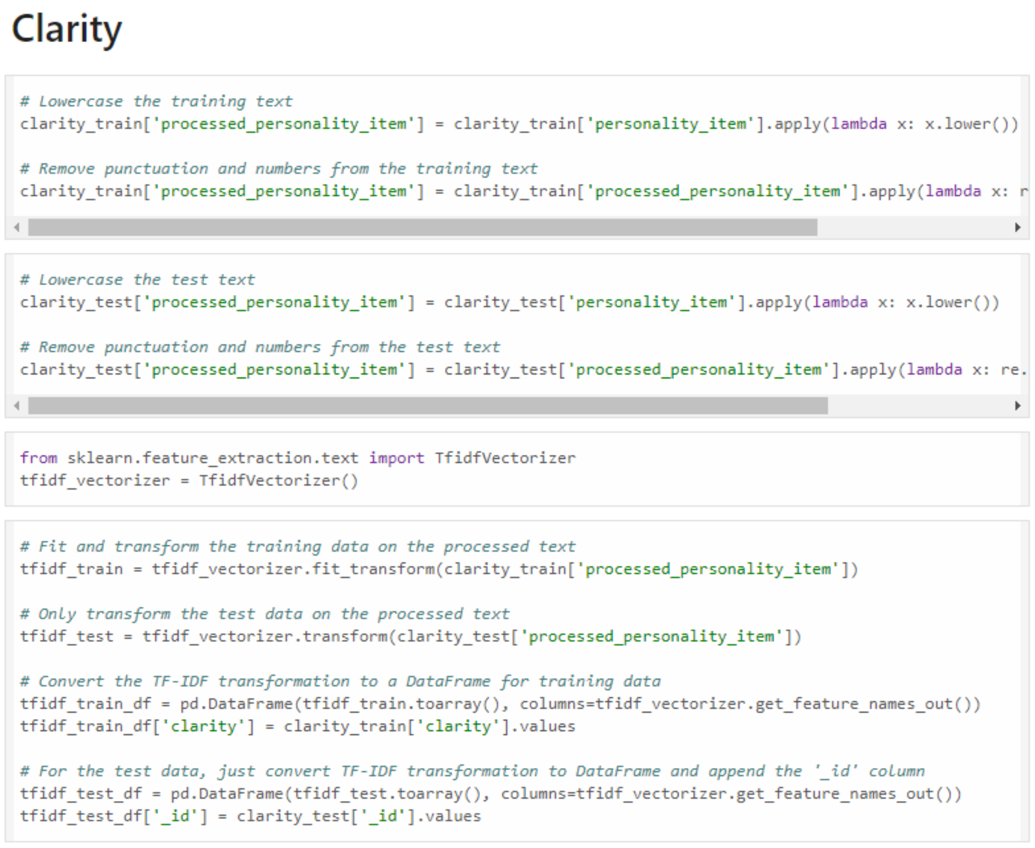


Figure 11 – The density plot of clarity scores in clarity_train.csv.



Clarity

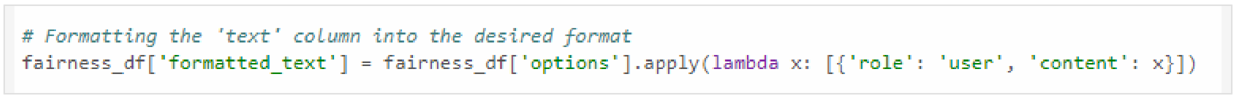
```
# Lowercase the training text
clarity_train['processed_personality_item'] = clarity_train['personality_item'].apply(lambda x: x.lower())

# Remove punctuation and numbers from the training text
clarity_train['processed_personality_item'] = clarity_train['processed_personality_item'].apply(lambda x: r
```

```
# Lowercase the test text
clarity_test['processed_personality_item'] = clarity_test['personality_item'].apply(lambda x: x.lower())

# Remove punctuation and numbers from the test text
clarity_test['processed_personality_item'] = clarity_test['processed_personality_item'].apply(lambda x: re.
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer()
```

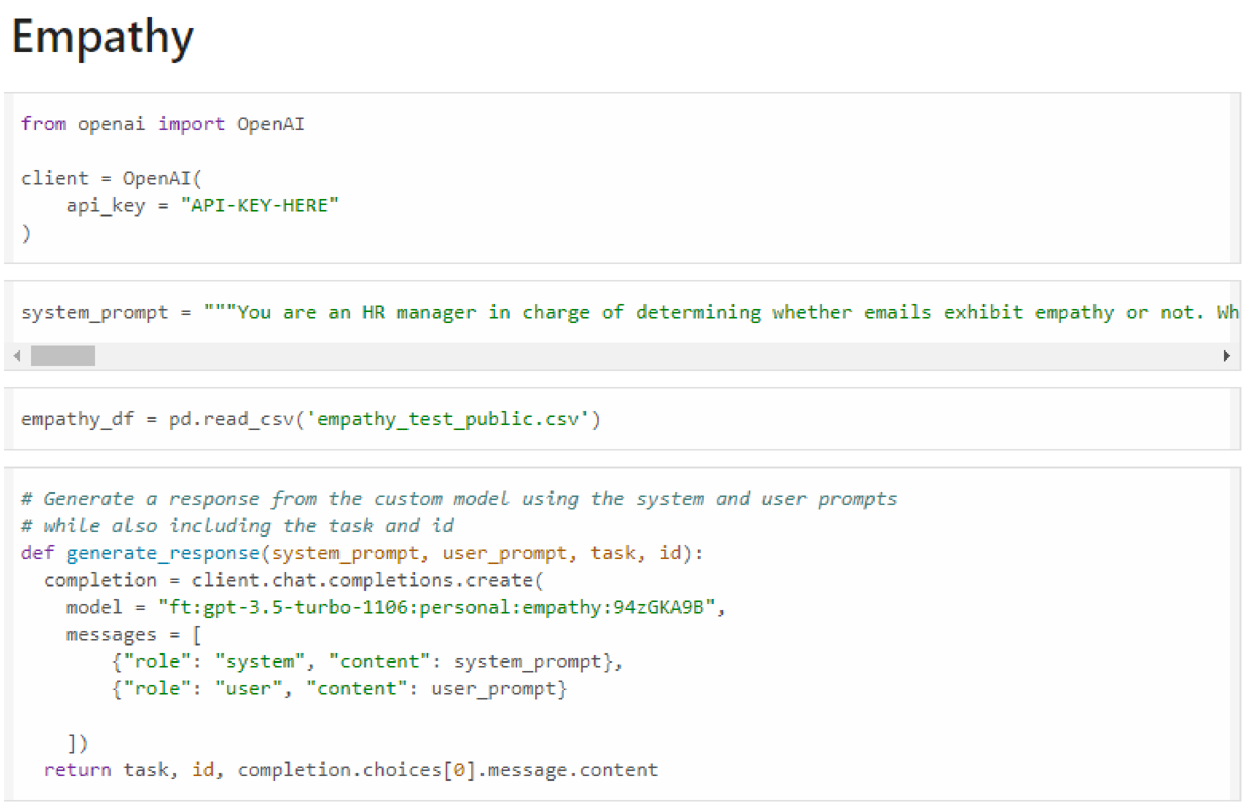
```
# Fit and transform the training data on the processed text
tfidf_train = tfidf_vectorizer.fit_transform(clarity_train['processed_personality_item'])

# Only transform the test data on the processed text
tfidf_test = tfidf_vectorizer.transform(clarity_test['processed_personality_item'])

# Convert the TF-IDF transformation to a DataFrame for training data
tfidf_train_df = pd.DataFrame(tfidf_train.toarray(), columns=tfidf_vectorizer.get_feature_names_out())
tfidf_train_df['clarity'] = clarity_train['clarity'].values

# For the test data, just convert TF-IDF transformation to DataFrame and append the '_id' column
tfidf_test_df = pd.DataFrame(tfidf_test.toarray(), columns=tfidf_vectorizer.get_feature_names_out())
tfidf_test_df['_id'] = clarity_test['_id'].values
```

Figure 12 – An example of using TD-IDF to vectorize the text in clarity_train.csv.



```python
# Formatting the 'text' column into the desired format
fairness_df['formatted_text'] = fairness_df['options'].apply(lambda x: [{'role': 'user', 'content': x}])
```

Figure 13 – An example of the JSONL formatting required to pass to the OpenAI API.



Empathy

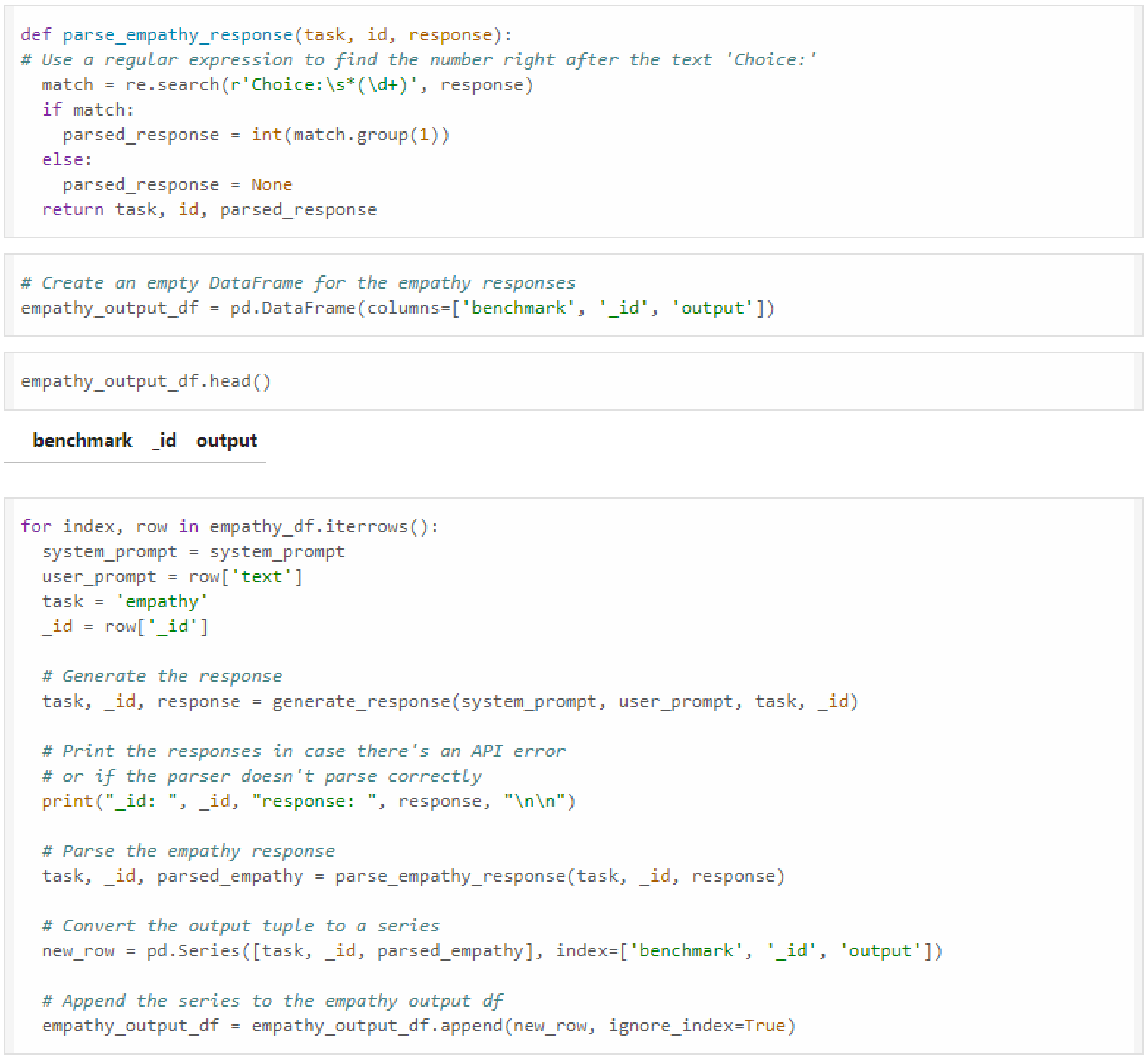
```python
from openai import OpenAI

client = OpenAI(
    api_key = "API-KEY-HERE"
)
```

```python
system_prompt = """You are an HR manager in charge of determining whether emails exhibit empathy or not. Wh
```

```python
empathy_df = pd.read_csv('empathy_test_public.csv')
```

```python
# Generate a response from the custom model using the system and user prompts
# while also including the task and id
def generate_response(system_prompt, user_prompt, task, id):
  completion = client.chat.completions.create(
    model = "ft:gpt-3.5-turbo-1106:personal:empathy:94zGKA9B",
    messages = [
        {"role": "system", "content": system_prompt},
        {"role": "user", "content": user_prompt}

    ])
  return task, id, completion.choices[0].message.content
```

Figure 14 – A custom function to generate responses from a fine-tuned version of GPT-3.5-Turbo via the OpenAI API.



```python
def parse_empathy_response(task, id, response):
# Use a regular expression to find the number right after the text 'Choice:'
  match = re.search(r'Choice:\s*(\d+)', response)
  if match:
    parsed_response = int(match.group(1))
  else:
    parsed_response = None
  return task, id, parsed_response
```

```python
# Create an empty DataFrame for the empathy responses
empathy_output_df = pd.DataFrame(columns=['benchmark', '_id', 'output'])
```

```python
empathy_output_df.head()
```

```
benchmark  _id  output
```

```python
for index, row in empathy_df.iterrows():
  system_prompt = system_prompt
  user_prompt = row['text']
  task = 'empathy'
  _id = row['_id']

  # Generate the response
  task, _id, response = generate_response(system_prompt, user_prompt, task, _id)

  # Print the responses in case there's an API error
  # or if the parser doesn't parse correctly
  print("_id: ", _id, "response: ", response, "\n\n")

  # Parse the empathy response
  task, _id, parsed_empathy = parse_empathy_response(task, _id, response)

  # Convert the output tuple to a series
  new_row = pd.Series([task, _id, parsed_empathy], index=['benchmark', '_id', 'output'])

  # Append the series to the empathy output df
  empathy_output_df = empathy_output_df.append(new_row, ignore_index=True)
```

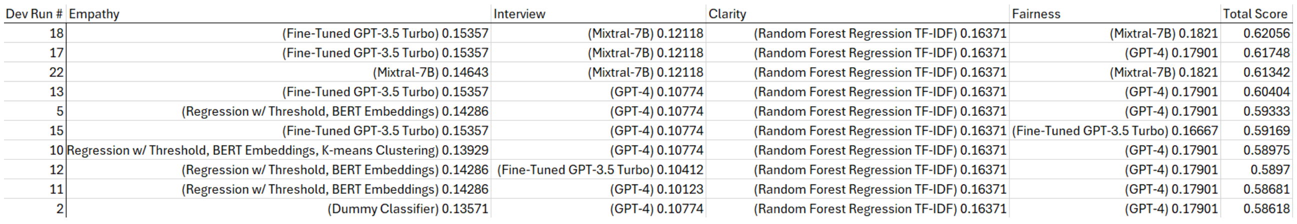
Figure 15 – The custom parser and logic to send and retrieve data from the OpenAI API, saving the responses into an output.csv file.

Task (GPT-4)	**Score**
Empathy	0.13571
Interview	0.10774
Clarity	0.05107
Fairness	0.17901
Final Score	0.47354

Table 1 – The results for the first attempt using only GPT-4.

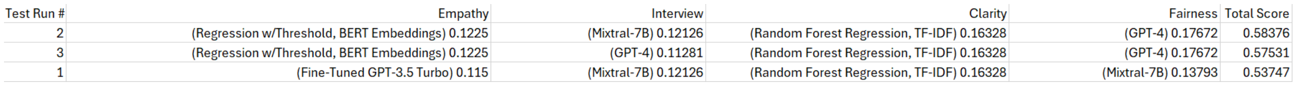
Task	**Score**
Empathy (Dummy)	0.13571
Interview (GPT-4)	0.10774
Clarity (Linear Regression)	0.16371
Fairness (GPT-4)	0.17901
Final Score	0.58618

Table 2 – The results for the second attempt using a mixture of Traditional ML and GPT-4.



Dev Run #	Empathy	Interview	Clarity	Fairness	Total Score
18	(Fine-Tuned GPT-3.5 Turbo) 0.15357	(Mixtral-7B) 0.12118	(Random Forest Regression TF-IDF) 0.16371	(Mixtral-7B) 0.1821	0.62056
17	(Fine-Tuned GPT-3.5 Turbo) 0.15357	(Mixtral-7B) 0.12118	(Random Forest Regression TF-IDF) 0.16371	(GPT-4) 0.17901	0.61748
22	(Mixtral-7B) 0.14643	(Mixtral-7B) 0.12118	(Random Forest Regression TF-IDF) 0.16371	(Mixtral-7B) 0.1821	0.61342
13	(Fine-Tuned GPT-3.5 Turbo) 0.15357	(GPT-4) 0.10774	(Random Forest Regression TF-IDF) 0.16371	(GPT-4) 0.17901	0.60404
5	(Regression w/ Threshold, BERT Embeddings) 0.14286	(GPT-4) 0.10774	(Random Forest Regression TF-IDF) 0.16371	(GPT-4) 0.17901	0.59333
15	(Fine-Tuned GPT-3.5 Turbo) 0.15357	(GPT-4) 0.10774	(Random Forest Regression TF-IDF) 0.16371	(Fine-Tuned GPT-3.5 Turbo) 0.16667	0.59169
10	Regression w/ Threshold, BERT Embeddings, K-means Clustering) 0.13929	(GPT-4) 0.10774	(Random Forest Regression TF-IDF) 0.16371	(GPT-4) 0.17901	0.58975
12	(Regression w/ Threshold, BERT Embeddings) 0.14286	(Fine-Tuned GPT-3.5 Turbo) 0.10412	(Random Forest Regression TF-IDF) 0.16371	(GPT-4) 0.17901	0.5897
11	(Regression w/ Threshold, BERT Embeddings) 0.14286	(GPT-4) 0.10123	(Random Forest Regression TF-IDF) 0.16371	(GPT-4) 0.17901	0.58681
2	(Dummy Classifier) 0.13571	(GPT-4) 0.10774	(Random Forest Regression TF-IDF) 0.16371	(GPT-4) 0.17901	0.58618

Table 3 – The top 10 performing model combinations on the development data.



Test Run #	Empathy	Interview	Clarity	Fairness	Total Score
2	(Regression w/Threshold, BERT Embeddings) 0.1225	(Mixtral-7B) 0.12126	(Random Forest Regression, TF-IDF) 0.16328	(GPT-4) 0.17672	0.58376
3	(Regression w/Threshold, BERT Embeddings) 0.1225	(GPT-4) 0.11281	(Random Forest Regression, TF-IDF) 0.16328	(GPT-4) 0.17672	0.57531
1	(Fine-Tuned GPT-3.5 Turbo) 0.115	(Mixtral-7B) 0.12126	(Random Forest Regression, TF-IDF) 0.16328	(Mixtral-7B) 0.13793	0.53747

Table 4 – The top 3 performing model combinations on the test data.

Business Understanding

In 2018, the Society for Industrial and Organizational Psychology began hosting an annual machine learning competition to determine how these techniques can be applied to industrial/organizational psychology. In 2024, the competition focused on, "prompting LLMs to effectively complete a set of diverse tasks presented in benchmark datasets" [1]. Five datasets were provided in both the training phase and development phase, and a test set was released two weeks before the competition ended. The training data was used to train models, the development data was used to refine the models, and the test data was used to score each team. The competition began on Feb 7th, 2024, and concluded on April 4th, 2024. The rules for the competition were as follows:

> "*You are allowed to use any LLM technique(s) during the entirety of the competition (e.g., base models, fine-tuned models, chaining). However, top-scoring teams will need to submit a reproducible solution that incorporates LLMs to the organizing committee for review at the end of the competition. If your solution cannot be reproduced or does not incorporate LLMs, then you will be disqualified*" [2].

The four assigned tasks according to the competition rules were:

> *"Predicting Empathy: Job candidates were asked to provide empathetic responses to a difficult workplace situation. Your task is to classify whether empathy was demonstrated or not in each simulated response.*
>
> *Generating Interview Responses: Job candidates responded to 5 common interview questions. You will be given the text of 4 question and response pairs. Your task is to generate a likely text response for the 5th question based on the previous responses.*
>
> *Rating Item Clarity: Respondents rated the clarity of personality test items using a 7-point scale from 1 = extremely unclear to 7 = extremely clear. Your task is to predict the average clarity rating for each item based on the responses.*
>
> *Identifying Fairness Perceptions: Respondents compared two organizational policies and voted on which was fairest. Your task is to identify which policy received the majority vote as the fairer option"* [2].

Data Understanding

There were 5 datasets provided to accomplish the 4 tasks—all containing text data, some with numerical data. Each of these datasets were small in size, ranging from 19-30 records in total, which proved to a be challenge. One idiosyncrasy was that the interview dataset was intended to be used in conjunction with the personality dataset for generating interview responses. An exploratory data analysis (EDA) was conducted on each of the datasets to determine what preprocessing steps were required. Due to the nature of this problem, a manual text analysis was also conducted with the aid of an LMM. Instead of relying on explicit features in the data, the text was analyzed by a human and an LMM, and higher-level heuristics were found. These heuristics became useful later for determining good system prompts to use with the LLM.

Empathy_train.csv

This dataset consisted of 30 records and 3 columns (_id, text, empathy). The `text` column contained emails in text format from a senior employee to a junior employee. The `empathy` column was binary—with `1` indicating that the email was empathetic. This data was slightly imbalanced, with 17 emails classified as empathetic, 13 classified as non-empathetic (shown in fig. 2.). The analysis of this data with the higher-level heuristics can be found in [4]. The goal of the analysis was to determine what aspects of the text responses were characteristically empathetic or not.

Unempathetic responses were characterized by a lack of personalized feedback, with a focus on urgent deadlines and additional oversight, and conditional support for career growth mixed with veiled critiques. Empathetic responses were characterized by an acknowledgement the recipient's efforts, circumstances, and contributions, with an inviting tone that offered personalized feedback, and emphasized the importance of the recipient's role, contributions, and the positive impact of their work on the team.

The analysis of this particular dataset was challenging due to it having significant overlap between empathetic and non-empathetic responses. This overlap could have been due to mislabeling of the data, or inconsistencies in the data collection process. One issue in particular was the "complement sandwich", where criticisms are bookended by complements. These challenges made it difficult to develop good models.

Interview_train.csv

This dataset consisted of 19 records and 3 columns (_id, questions_answers, last_answer). The `questions_answers` column consisted of 3 question-response pairs for a particular applicant, and a 4th question without a response. The `last_answer` column contained the response to the 4th unanswered question. The goal for this data was to generate a response based on the previous question-answer pairs that had a high cosine-similarity score with the 4th response.

One potential approach for working with this data was to analyze the questions to determine what kind of experiences are asked about in interviews. A common question is, "Tell me about a time where you had multiple projects and had to manage your time well." This question gets at the lived human experience of the applicant, which a machine does not have. The responses to these questions could be used to build a profile of the applicant, allowing the LLM to adopt the persona of the applicant to better answer the questions. The manual analysis of this data can be found in [5]. The analysis showed the most important experiences that the persona needed to have:

- Experience with continuous learning.
- Experience with decision making and problem solving.
- Experience with communication, team management, and handling adversity.

- Experience with leadership and strategic planning.
- Experiences where empathy and inclusivity played a key role in workplace interactions.

Personality_train.csv

This data consisted of 19 records and 6 columns (_id, extraversion, agreeableness, conscientiousness, emotional_stability, and openness). Each record in this dataset contains the values of a particular applicant's "Big 5 Traits", and the `_id` column corresponds to the `_id` column in the interview dataset. The distributions for each of the big 5 traits are shown in figures 5-9. Further, a latent profile analysis (LPA) was conducted with R to determine if there were underlying subgroups among the applicants based on their personality traits. Unfortunately, the LPA did not reveal any distinct or meaningful profiles within the applicant pool regardless of the number of classes chosen. One of the LPAs is shown in fig. 10. Ultimately, the personality data was not used due to the difficult nature of using continuous numerical values to generate text responses.

Fairness_train.csv

This dataset consisted of 24 records and 4 columns (_id, first_option, second_option, majority_vote). The option columns contained two different workplace policies and the majority vote column contained the majority vote by employees as to which policy they perceived as being fairer. The manual analysis of this dataset in [6] led to these heuristics:

Fairer workplace policies tended to require minimal effort on behalf of employees, prioritized in-person interactions over digital or anonymous methods, emphasized employee well-being, and employed proactive/preventative measures for dealing with workplace conflicts. Unfair workplace policies tended to require more time and effort from employees (especially outside of working hours), focused on impersonal interactions and an overreliance on technology. They also emphasized punitive measures and increased monitoring, and in some cases mandating role-playing exercises or workshops.

Clarity_train.csv

This dataset consisted of 30 records and 3 columns (_id, personality_item, clarity). The `personality_item` column consisted of prospective items from a psychological inventory. A group of I/O psychologists voted on the clarity of the item which produced a continuous clarity score between 1 and 7. The distribution of the clarity column was bimodal, having two peaks around 3 and 6.5 (shown in fig. 11). The manual analysis of this data in [7] led to the following heuristics about the clarity of the items:

Items deemed less clear exhibited the following qualities:

- Passive voice ("I am considered…" vs. "I consider myself….")
- Past tense ("I did not feel like eating… vs. "I do not feel like eating…")
- Normative statements ("…even though I should have been…", "…needs to…")

- Metaphorical language (“I look for something to hold onto” vs. “I seek stability”)
- Negations (“…not too high and not too low”)
- Conditional statements (“I am able to work hard to achieve results that I will only get at a time far in the future.”)
    - The “results” are contingent on “work[ing] hard”

There are a few psychological explanations for why these items were deemed as less clear. Passive voice is often considered indirect (i.e. less clear) than active voice. Metaphorical and idiomatic language, negative statements, and conditional statements all take more mental effort to parse than simple direct language. Additionally, the use of vague or ambiguous words or phrases serves to make items less clear such as “very few”, “tend to”, “financially well-off”, and “regular basis”, as there are many interpretations for what these statements might mean depending on the context. The length of the sentence was a factor, as longer sentences have more complex structure and require cognitive labor to parse.

Data Preparation

The analyses from the previous step led to the creation of system prompts for the LLMs. Two main techniques were used to prepare the text data for traditional machine learning methods: Term frequency-inverse document frequency (TF-IDF) and BERT word embeddings—an example is shown in fig. 12. Simple preprocessing steps were implemented such as consolidating the options columns in the fairness dataset into a single column and lowercasing all the text in each dataset. Little cleaning effort was required as the datasets were relatively clean at the outset. To interact with LLMs in a programmatic way through the OpenAI API, the data was formatted as JSONL with keys specifying who was speaking and the content of their message (shown in fig. 13). To actually call the OpenAI API, a custom function was written in Python, specifying the model to call, the data to send, and what to return to the function. An example of this is shown in fig. 14. Additionally, a custom parser was required to parse the responses from the OpenAI API and combine them into an output.csv file. The code for this is shown in fig. 15.

Modeling

A mixture of traditional machine learning methods was used in tandem with modern LLMs. The system prompts were derived from the exploratory analysis and consultation with subject matter experts in the field of industrial-organizational psychology. In most cases, the LLMs outperformed the traditional methods. Custom parsers were required to parse the responses, as they were in JSON format. The first attempts utilized GPT-4 for each task with the OpenAI API. The fairness and empathy system prompts are listed below as examples.

Fairness System Prompt:

"You are a psychological researcher in the field of industrial-organizational psychology. You will be given two workplace policies. Your goal is to determine which one employees voted as being more fair. When evaluating the fairness of the two policies, consider the following heuristics:

Characteristics of less fair policies are:

Requiring more time or effort from employees (especially outside of working hours). Creating unnecessary work for employees. Formality over flexibility. Excessive use of technology instead of personal interactions. Less personal solutions. Lack of personal support. Bringing in 3rd parties. Monitoring and surveillance. Punitive Measures. Focus on criticism. Unstructured conflict resolution. Mandatory role-playing or workshops. Limited opportunities for direct input. Restrictive communication channels.

Characteristics of more fair policies are:

Requiring minimal effort from employees. In-person interactions over digital or anonymous methods. Focus on individual growth and personalization. Empowering autonomy and speaking up about issues. Professional/career growth opportunities. Emphasis on employee well-being. Providing emotional and peer-based support. Preventative measures and proactive approaches. Training managers in conflict resolution. Holding managers accountable. Clear conflict resolution policies. Recognition of employee contributions. Offering opportunities for dialogue. Focus on building relationships. Involving people over technology. Working out issues without 3rd party intervention.

Instructions:

Read the two policies carefully. Consider the heuristics provided for less fair and more fair policies. Make a determination as to which policy is considered more fair and specifically respond with 'Choice: First' or 'Choice: Second'. Justify your answer briefly, highlighting which aspects of the policies influenced your decision."

Fairness Example Output:

"_id: 24 response: Choice: Second

The second policy appears to be more fair according to the heuristics provided. This policy exemplifies a proactive approach with a focus on training managers in conflict resolution, holding them accountable, and a clear resolution policy. On the contrary, the first policy promotes 3rd-party interventions, which according to the heuristics, is a characteristic of a less fair policy."

Empathy System Prompt:

"You are an HR manager in charge of determining whether emails exhibit empathy or not. When evaluating the empathy of an email, consider the following heuristics:

Characteristics of less empathetic emails:

- Lack of personalized, positive feedback and explicit guidance.
- Direct and implicit criticism without constructive direction
- Pressure, both implicit and explicit, without offering support.
- Prioritizing organizational needs over individuality and stakeholder feedback.
- Emphasizing efficiency and compliance over personal development and dialogue.
- Failure to recognize or invite collaborative communication and dialogue.
- Making unclarified assumptions and neglecting the emotional impact of messages.
- Using directive language, imposing solutions, and mandating training without consensus.
- Vagueness in feedback, goals, and the highlighting of public consequences of errors.
- Formal or informal communication that mismatches the seriousness of feedback.
- Neglecting to acknowledge efforts, emphasizing mistakes, and undermining achievements.
- Imposing urgent deadlines, additional oversight, and repetitive tasks without clear support.
- Conditional support for career growth and veiled critiques that imply inadequacy.
- Communication errors such as mixed messages or compliment sandwiches.

Instructions for Evaluation:

Read each email message attentively. Balance your evaluation based on the heuristics for both empathetic and less empathetic characteristics. If an email does not definitively show several empathetic traits or displays numerous characteristics of less empathetic communication, classify it as lacking empathy. Your determination should be made with consideration to the nuanced balance between these traits. Respond with 'Choice: 0' for emails that do not exhibit empathy and 'Choice: 1' for emails that do. Provide a brief justification for your choice, focusing on the key aspects of the message that influenced your decision."

Empathy Example Output:

_id: 198 response: Choice: 1

"Justification: Although the email does include language related to task or requirements like "submitting," "reports," and "meetings," it does so in an empathetic manner. Words like "feedback," "enthusiasm," "eagerness," and "assistance" indicate empathy towards Jonathan's feelings. The line "Let me know if you need any assistance on this" directly implies a direct, personal engagement, indicating empathy. The criticism in the email is also very constructive, focusing on Jonathan's development and future improvement."

Further attempts utilized an open-source LLM, Mixtral-7B, with the same system prompts. Interestingly, even though Mixtral-7B had far fewer parameters compared to GPT-4, it outperformed it in many cases on the development data. Additionally, the OpenAI Playground was used to fine-tune GPT-3.5 Turbo, the best model available for fine-tuning at the time. All that was required was that the training data be converted to a JSONL format. The model was connected to in a similar way to GPT-4, using an API key and specifying the fine-tuned model in Google Colab [9].

Each LLM performed better on different tasks, but it was generally found that Mixtral-7B outperformed GPT-4, while the fine-tuned GPT-3.5 Turbo could outperform Mixtral-7B on one or two tasks.

Development Results

1st Attempt (GPT-4 Only)

The first approach (using GPT-4 for all tasks) led to an overall accuracy of 0.47354. The top scoring team had a reported overall accuracy of 0.668 at the time. The results for each task are shown in table 1. The obvious area for improvement was the clarity task; the interview score was low due to the inherent difficulty of that task; the empathy score was deceptive, as the LLM ended up acting as a dummy classifier, predicting every response as empathetic; the fairness score was surprisingly high.

2nd Attempt (Traditional ML + GPT-4)

Multiple iterations of this method were used to determine where traditional ML methods could outperform the LLM, and vice versa. A linear regression was used on the clarity data, a quadratic discriminant analysis was used on the fairness data, and a variety of classification models were used on the empathy data along with and without sentiment analysis. However, the interview data still necessitated the use of an LLM for text generation.

The regression on the clarity data led to a significant improvement in accuracy over the LLM. The QDA on the fairness data did not lead to an improvement in the accuracy, so the LLM was favored in this case. None of the classification models created for the empathy data were able to achieve accuracy better than a dummy classifier. The results from the best iteration are shown in table 2. The overall score increased from the previous attempt from 0.47354 to 0.58618.

Subsequent Attempts

Over 20 iterations later, the team was able to improve the final score from 0.58618 to 0.62056 on the development data. Small increases were seen on the empathy, interview, and fairness tasks, while the Random Forest Regression with TF-IDF consistently outperformed all other methods on the clarity task. Mixtral-7B performed best on the interview and fairness tasks using the same system prompts as GPT-4. The fine-tuned GPT-3.5 Turbo outperformed Mixtral-7B and GPT-4 on the empathy task. Table 3 shows the top 10 model combinations for the development data overall.

Conclusion

The best combination of models for the development data was not necessarily the best combination for the test data, however. The empathy task was the downfall here, as the score saw a significant decrease compared to the development data. It turned out that a linear regression model with a threshold using BERT word embeddings was the best performing model for the empathy task—significantly outperforming a logistic regression using the same embeddings and slightly beating out the fine-tuned GPT-3.5 Turbo. Mixtral-7B had the best performance on the interview task. The random forest regression with TF-IDF continued to be the best model for the clarity task. GPT-4 outperformed Mixtral-7B on the fairness task on the test data, which it failed to do on the development data. The final score for the best run on the test data was 0.58376. This yielded the team 8th place out of 19.

Table 3 shows the best 3 runs on the test data, as participants in the competition were only allowed a maximum of 5 submissions for the test data. The Google Colab file containing all the Python code for this project can be found in [9].

Opportunities for Improvement

The team unanimously agreed that the empathy dataset was poorly labelled. The unfortunate truth is that real-world data is not always necessarily representative of reality. Multiple other teams shared the same sentiment. It's possible that there were more preprocessing steps that could've been implemented to make the class distinction clearer, however they would have needed to be done in a way that was future-proof. Since the text content of emails is highly variable, it would've been difficult to develop repeatable feature engineering steps for this dataset.

Another area for improvement was with the interview and personality datasets. The personality data was not used for the generation of interview question responses, but it could have been. It's not exactly clear how this would've been done though. When the team tried to incorporate the personality data into the interview responses, the LLM would come back with something like, "Because of my high openness score of 6, I have found it easy to include others in my work," which obviously doesn't make sense for this use case because job applicants do not refer to their big 5 traits when responding to interview questions.

Other teams utilized a variety of different techniques to accomplish these tasks, but the one thing that was common among all the top scorers was that they used multiple chains of LLMS, rather than just a single one. For example, the team scoring 2nd place used the following methods for each of the tasks:

Empathy: An ensemble method combining zero-shot chain-of-thought (COT), n-shot COT, and an ELO rater (adapted from chess), where the majority vote between these 3 LLMs yielded classification of empathetic or not.

Clarity: Another ensemble method. The first LLM used a meta prompt which broke down each item into binary answers to questions about the clarity of the item. The second model utilized this binary to classify the item as empathetic or not.

Interview: The LLM was instructed to generate the response based on the style, tone, and characteristics of the previous responses. Then, n in-context samples were generated to create more diverse responses to the same question for the same applicant. Then, the response with the highest cosine similarity with the previous responses was selected as the output.

Fairness: The LLM was given subsets containing n samples (where n is odd and is initially small), then makes its prediction. Then, n is increased to make more predictions while ensuring consistency between all fairness rankings.

There were far too many different methodologies used by the top-ranking teams to list, but a Github repository containing the top 4 teams' presentations at the annual SIOP conference can be found in [10].

Acknowledgements

I would like to express my deepest gratitude to all those who have supported and guided me throughout the course of this capstone project.

First and foremost, I would like to thank Dr. Jim Kulich for recommending me for this project. His advice and expertise have been instrumental in the success of this project and many others. I would like to thank Dr. Catrina Notari for allowing me to be a part of her team, her I/O psychology expertise, and her encouragement. I would like to thank Albert Cassamajor for his advice about software best-practices and programming in general. Finally, I'd like to thank Ian Lee for his contributions to this project, especially with the empathy data.

References

[1] Stark, A. (2024, January 23). SIOP is Once Again Hosting Machine Learning and Student Consulting Competitions. *Society for Industrial and Organizational Psychology*. https://www.siop.org/Annual-Conference/ArtMID/35503/ArticleID/9122/SIOP-is-Once%20Again%20Hosting-Machine-Learning-and-Student-Consulting-Competitions

[2] EvalAI: Evaluating state of the art in AI. (n.d.). EvalAI. https://eval.ai/web/challenges/challenge-page/2207/overview

[3] API Reference. OpenAI. https://platform.openai.com/docs/api-reference/introduction

[4] Empathy Analysis. Google Sheets. https://docs.google.com/document/d/1N0Ajx6gnlupG2reUS3Xuq0Lharq-rk8p8gbZeqWwyDs/edit?usp=sharing

[5] Interview Analysis + Heuristics. Google Sheets. https://docs.google.com/document/d/1iftGPrg8EzfJXt8VRy6IGKd9ZtRLLo7C8jenb1mujZ4/edit?usp=sharing

[6] Fairness Analysis. Google Sheets. https://docs.google.com/document/d/12ESanlRRF-0ds_-pYuvSitiHMJM1t8YZ6DoRWwmlQyM/edit?usp=sharing

[7] Clarity Analysis. Google Sheets. https://docs.google.com/document/d/13_es5_uaWAog9_F9zfc8KNt7Z4j6RVzLOjFdWSoek18/edit

[8] Initial OpenAI API Testing. Google Colab. https://colab.research.google.com/drive/1OBo7U7hGeRU3_c-dBlJ3ELDJ0Rq0Agjc?usp=sharing

[9] Final Colab File. Github. https://github.com/barstow2/siop-24-ml-competition/blob/main/SIOP%20ML%20Competition%202024.ipynb

[10] SIOP Machine Learning Competition. Github. https://github.com/izk8/2024_SIOP_Machine_Learning_Competition/tree/main