2024 SIOP Machine Learning Competition

Team Hungry Llama

Jennifer Gibson, PhD, PStat®

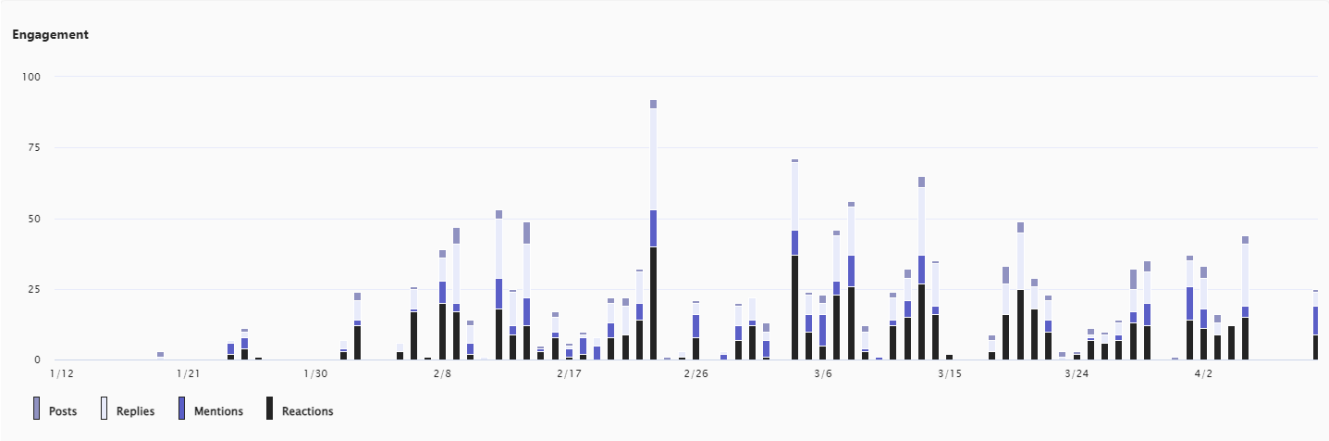
April 2024

ForsMarsh

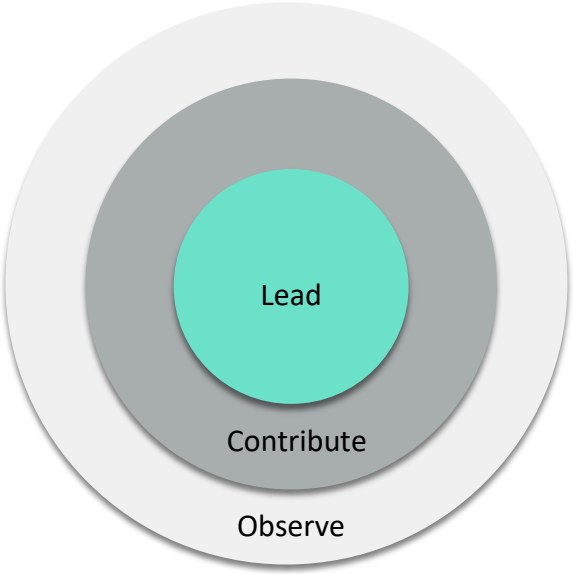
Team Processes

- Recruitment
- Specialties
- All levels welcome
- Divide tasks but share ideas and code
- MS Teams
- Weekly huddle
- Naming

MS Teams activity over two months



Levels of Involvement



Hungry Llama

Shane Halder
Task 1 Lead 1·2·3

Jen Gibson
Task 2 Lead 2

Blake Hoffman
Task 3 Lead 3

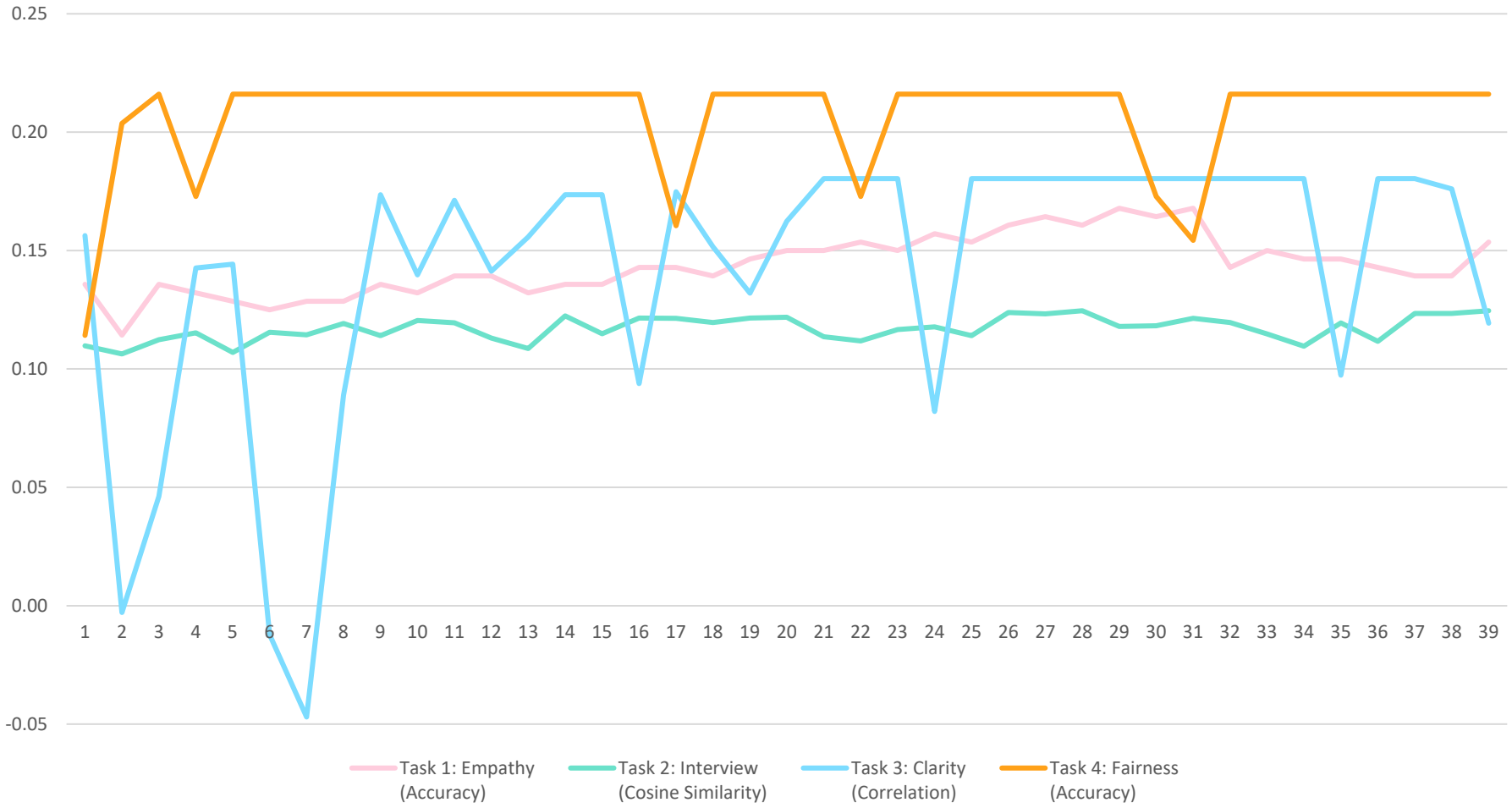
Joe Luchman
Task 1 Lead 3·4

Selena Tran
1

Nick McCann
3

Hannah Johnson
1

Learning Journey: Solution Quality Over 40 Submissions



Task 1: Empathy

Evolution of the Empathy Solution

- Initially tried machine-learning algorithms to model Basic Empathy Scale (BES) emotions
  - Fear, Sadness, Anger, and Happiness
  - Python packages such as NRCLex, VADER, TextBlob, and Roberta GoEmotions
- Tried using cosine similarity among training cases
- Tried an ensemble of ML, Mistral 7B zero-shot, and cosine similarity
- Needed to break empathy down based on how it was rated in the data vs the LLMs interpretation of what empathy is; came up with multiple prompts rating empathy in different ways
- ChatGPT 3.5 Turbo but did not perform as well as Mistral

Approach

1. Programmed in Python
2. Uses the following LLMs via the Hugging Face API:
    a) Mistral-7B-Instruct-v0.2
    b) Mixtral-8x7B-Instruct-v0.1
3. Quantify empathy in 9 metrics:
    a) 8 different LLM prompts (empathy few-shot, sentiment zero-shot, negativity zero-shot, approval zero-shot, appreciation zero-shot, support zero-shot, rapport building zero-shot, empathetic phrases zero-shot)
    b) 1 harsh words regular expression pattern matching
4. Predictions via weighted metric sums and cutoffs

Predicting Empathy

LLM Prompt Template & Parameters:

```
prompt_template = {
    "inputs": f"<s>[INST] {prompt}[/INST]",
    "parameters": {
        "do_sample": False,
        "max_new_tokens": 2000,
        "top_p": 0.9,
        "top_k": 1,
        "repetition_penalty ": 1,
        "presence_penalty": 0,
        "frequency_penalty": 0,
        "return_full_text": False
    }
}
```

Predicting Empathy

LLM Prompts:

Empathy

```
empathy_prompt = f"""You are an email message categorization bot. Your task is to categorize the email message after <<<>>> into one of the following predefined categories:

A
B
C

You will only respond with a JSON object with the category A or B, the probability of being category A, and confidence. Do not provide explanations or notes. If you're not sure use category C.

###
Here are some examples, but ignore the order:

{empathetic_examples}{unempathetic_examples}###
"""

empathy_prompt += """<<<
Email Message: {0}
>>> """
```

Sentiment

Negativity

Approval

Appreciate

Support

Rapport

Phrases

```
sentiment_prompt = """Rate, from 0 to 1, the overall sentiment towards the recipient in the following email message. You will only respond with a JSON object with the sentiment rating, and confidence. Do not provide explanations or notes.

{0}
"""

negativity_prompt = """Rate, from 0 to 1, the sender's overall negativity towards the recipient in the following email message. You will only respond with a JSON object with the negativity rating, and confidence. Do not provide explanations or notes.

{0}
"""

approval_prompt = """Rate, from 0 to 1, the sender's overall approval for the recipient in the following email message. You will only respond with a JSON object with the approval rating, and confidence. Do not provide explanations or notes.

{0}
"""

appreciation_prompt = """Rate, from 0 to 1, the sender's appreciation towards the recipient's contributions to the project in the following email message. You will only respond with a JSON object with the appreciation rating, and confidence. Do not provi

{0}
"""

support_prompt = """Rate, from 0 to 1, how well the sender offers support in terms of learning resources, mentorship or coaching for the recipient in the following email message. You will only respond with a JSON object with the support rating, and confi

{0}
"""

rapport_prompt = """Rate, from 0 to 1, how well the sender builds rapport through shared experiences with the recipient in the following email message. You will only respond with a JSON object with the rapport rating, and confidence. Do not provide expla

{0}
"""

phrase_prompt = """Rate, from 0 to 1, the number of empathetic phrases in the following email message. You will only respond with a JSON object with the phrase rating, and confidence. Do not provide explanations or notes.

{0}
"""
```

Task 2: Interview

Evolution of the Interview Solution

- Python, Google Colab, Mistral 7b HF API, Mixtral8x7b HF API, ChatGPT (3.5 and 4) API, and ChatGPT Plus
- Develop an idea of the job candidate's way of responding and use that to guess what they would say in response to Q4
- Personality and emotion
- Added Big 5 dimensions in groups
- Tone and emotion from A1-A3
- Varied combinations and thresholds, terminology
- Use basic characteristics of A1-A3: Number of words or sentences
- Encode Flesch reading level

Approach

- ChatGPT 4
- Restate the question
- Big 5 (Example: Be more/less extraverted than average)
- Basic characteristics of A1-A3
  - Average sentences across answers
  - Number of words or sentences
- Average reading level across answers
  - Example: Write at the reading level of a middle school/high school/college graduate and professional audience
- Varying the prompt

Variants That Did Not Improve Prediction

- Instead of “Be similar in tone”
  - Textblob polarity
  - Vader polarity
- Emotion scores from NLP packages (e.g., fear, joy, surprise)
- Mistral
- Mimic basic characteristics of most recent answer (A3)

Example Prompt

[Questions Q1-Q4 and Answers A1-A3]

Generate a response for the provided question following these rules:

- Start with a restatement of the question
- Have a similar writing style
- Write at least [3] sentences
- Be similar in tone
- Be [more extraverted] than average
- Be [less conscientious] than average
- Be [less open to new experiences] than average
- Be [less neurotic] than average
- Be [less open to new experiences] than average
- Write at the reading level of [a college graduate and professional audience]

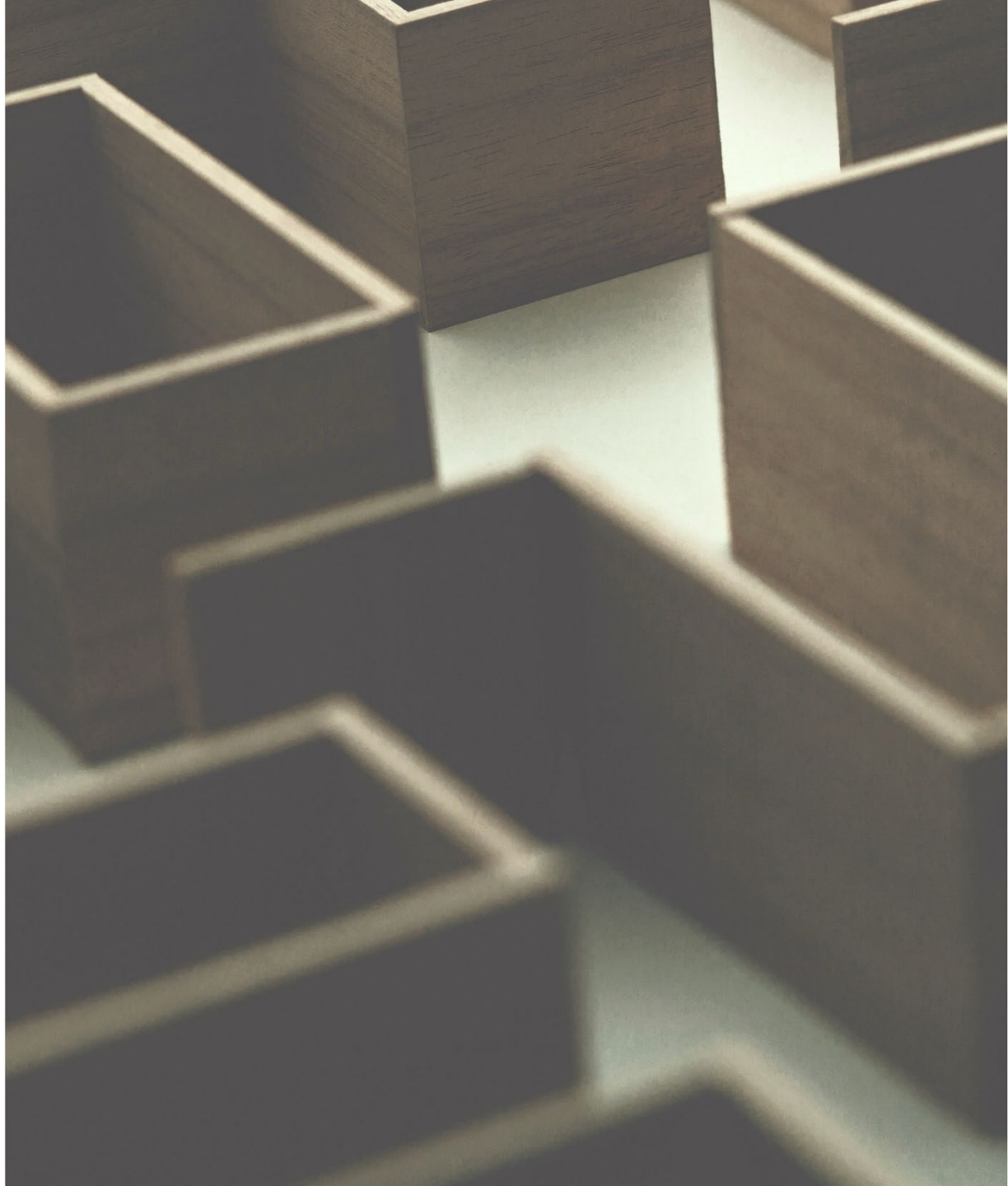
Task 3: Clarity

Evolution of the Clarity Solution

- Initially tried a single zero-shot and few shot rating by Mixtral
- Tried variants of random forests, gradient boosters, and model averaged regressions
- Used ratings from Llama 2.0, individual ratings from Mixtral 8x7B, used ratings from OpenChat 3.5
- Tried using all NLP features and all 1,024 embedding features
- Best solution up until 2 weeks prior to test data release was randomly guessing between values of 5.5 and 6.5 but using a value of 3 for cases with any phrase with ‘am’ , a comma, or negation
- Python (NLP encoding) and R (everything else)

Approach

1. Process the data: Add NLP features
2. Generate personality ratings for each phrase (zero-shot)
3. Generate clarity ratings directly (few-shot)
4. Generate final predictions of clarity



Process the Data

Nineteen Python *nltk*- and *vader*-based features including:

1. Does phrase contain word “am”, a comma, include negation (i.e., “not”).
2. Flesch reading ease and difficult word count.
3. Phrase sentiment and subjectivity
4. Parts of speech including adjective, pronoun, past tense verbs, etc.

Word embeddings obtained from user *intfloat*’s ‘multilingual-e5-large’ Hugging Face model called with *httr2* in R.

Generate a single average embedding score

All 1,024 features

averaged within row to obtain a **single average embedding score**.

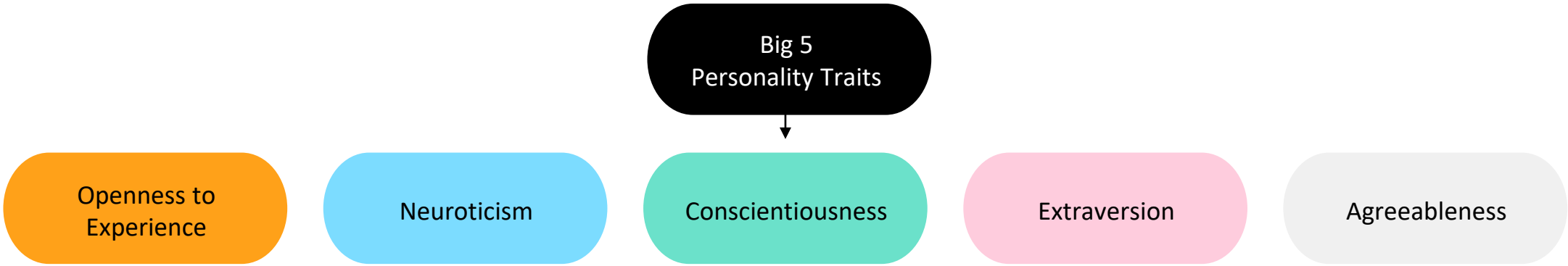
Generate Personality Ratings

Zero-shot classification from Facebook's Bart model on Hugging Face called with *httr2* in R.

Each phrase was rated on the probability it belonged to one of the Big 5 personality traits.

The proportions were logit-transformed.

The standard deviation of the logit-transformed probabilities was used as a metric of classification uncertainty



Generate Intermediate Clarity Ratings

MistralAI's Mixtral 7x8B model on Hugging Face called with *httr2* in R.

Ten different direct ratings of clarity with 20 different training examples randomly selected from training data.

Combined the 20 training examples in the following way:

```
"<s> [INST] -- context -- Learn from the examples. Respond with a number between 1
and 7. Do not explain your reasoning. -- examples -- Am considered well-off
financially. [/INST] 3 [INST] Do not exercise ... </s> [INST] Rarely overindulge.
[/INST]
```

Note the use of Mixtral's special tags to identify past conversation (i.e., `<s>` and `</s>`) and instructions (i.e., `[INST]` and `[/INST]`).

First mention of a digit was used as the rendered clarity rating.

Generate Final Predictions of Clarity

R's *ranger* package with `extratrees` method/extremely randomized forests.

The average direct rating, 6 personality ratings, and 20 NLP features used. Trained on training data and applied to test data.

Top 5 training features:

Did phrase have 'am' in it?	**Average Mixtral rating for clarity**	**Did phrase have negation?**	**Difficult word count**	**Pronoun count**
01	02	03	04	05

Task 4: Fairness

Evolution of the Fairness Solution

- Initially tried a single zero-shot rating by Mistral 7B – which was nearly as good as the final method
- Final method devised in 3rd dev submission (of 40 total!)
- Used a variation on the prompts to include special Mixtral tags discussed in Task 3
- Used method like the Mixtral-based method in Task 3 where Mixtral received fewer training examples but had 10 different ratings and were combined in an extremely randomized forest
- R only

Approach: Few-Shot Learning with Mixtral

1. Select training observations
2. Develop prompt
3. Submit prompt to Mixtral 7x8B and parse

Select Training Observations

Randomly remove 1 of the 24 rows from training sample.

```
prompts_selected <-
  fair_train_df |>
    dplyr::slice_sample(n = 23)
```

Twenty-three chosen to match training approach where all but focal row is used.

Hence, 23 rows.

Sampling seed set with assistance from RANDOM.org's random number generator.

Develop Prompt: Offer Context

Combine and label all 23 few shot instances like:

```
Question {x}: Which option is fairer?

Option 1 = Conflict Resolution Workshops:
We ...

Option 2 = Conflict Resolution Workbooks:
Resources ...

Answer {x}: Option 1

...
```

Where `{x}` is a number between 1 and 23.

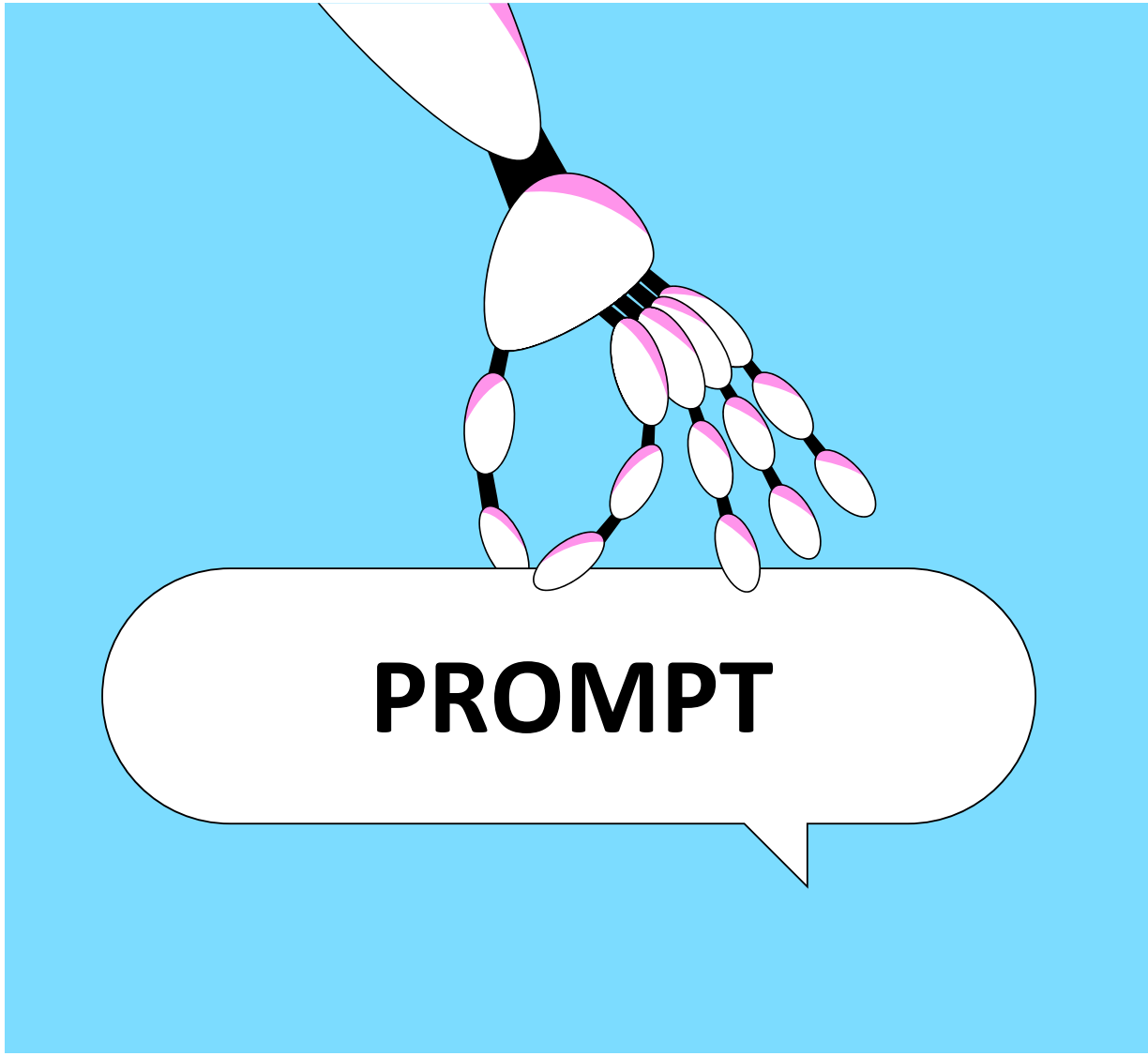
Include an incomplete from the test data at the end like:

```
Question 24: Question {x}: Which option
is fairer?

Option 1 = Employees can request a transfer
...

Option 2 = Arbitration: We offer a neutral
...

Answer 24:
```

Develop Prompt: Question for Mixtral

Mixtral is asked to:

```
Answer Question 24 below given the responses
to Questions 1 through 23 above. Only
respond with 'Option 1' or 'Option 2' and do
not explain your reasoning. The order of
'Option 1' and 'Option 2' is random. Do not
use the order that the options appear when
judging which is fairer. The order of
Questions 1 to 23 is random. Do not use the
order in which the questions to judge the
fairness of the options.
```



Submit Prompt

Mixtral 7x8B on HuggingFace called using *httr2*

Parameters to Mixtral were identical to those in ‘clarity’ task

First response of ‘Option 1’ or ‘Option 2’ was used as the rendered fairness judgment even if more than one was reported back by Mixtral

Final Five

Select top 3 solutions for each of the tasks

Fourth submission: Submit best of (highest score)

Fifth submission: Submit best of with one last change

Acknowledgements

Thank you.