1 Data Exploration

1.1 Feature Types

Numerical features: "How likely are you to use this model for academic tasks?", "Based on your experience, how often has this model given you a response that felt suboptimal?", How often do you expect this model to provide responses with references or supporting evidence?, How often do you verify this model's responses?

Categorical features: Which types of tasks do you feel this model handles best?, For which types of tasks do you feel this model tends to give suboptimal responses?, label.

Text features: "In your own words, what kinds of tasks would you use this model for?", "Think of one task where this model gave you a suboptimal response. What did the response look like, and why did you find it suboptimal?", "When you verify a response from this model, how do you usually go about it?".

1.2 Distributions and Class Balance

- The labels ChatGPT, Claude, and Gemini each appear 275 times, as desired. This perfect class balance ensures the model will train from a fairly-weighted dataset.
- The distribution of response for each of the four numerical features is shown in Figure 1. As expected, the mean and median are generally close to the neutral answer 3, although the median is 4 for the first and last questions.

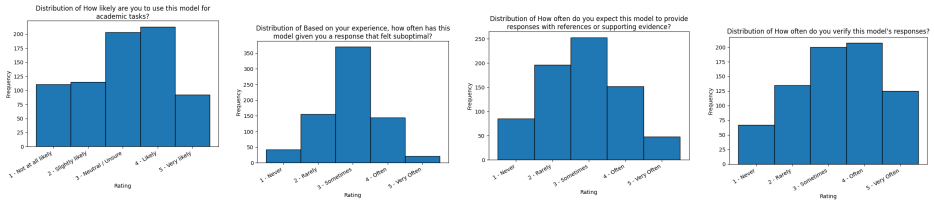

Figure 1: Distributions of numerical features.

1.3 Issues in Data and Proprocessing Transformations

- The dataset has missing values, such as in lines 9-10 or training_data_clean.csv.
- A nonsensical feature "#NAME?" appears 4 times.
- Missing or nonsensical values were handled by dropping rows.
- For features answered on a scale from 1 to 5, the first character of the string response was converted to an integer, to treat the feature as numerical.
- Multi-select survey responses were converted into binary features.

1.4 Data Splitting

- The last 30% of the rows were reserved as a test set, and ignored during data exploration. The first 70% will be split into training and validation sets.

2 Methodology

2.1 Model Families

Our dataset is relatively small, so we will focus on relatively simple, computationally-efficient models.

1. *k-Nearest Neighbours:* kNN naturally handles numerical ratings, binary features derived from multi-select responses, and structured features such as text. Its dependence on feature similarity makes it an intuitive choice for this dataset, where patterns of task preferences and ratings indicate model usage. The fact that kNN does not assume a linear relationship between features and labels is also desirable, as we have little reason to expect a linear relationship based on our data exploration, for example the distributions of numerical features.
2. *Decision Trees:* Decision trees are suitable for mixed feature types and capturing relationships between numerical features and binary categorical features, which form an important part of our dataset. The numerical features exhibit relatively low entropy distributions, with noticeable concentration near rating 3-4. This means sampled values are more predictable, and information gain from numerical feature splits can be quite high. We can extend this to random forests, using ensemble averaging to avoid overfitting.
3. *(Multinomial) Logistic Regression:* Logistic regression is a computationally efficient classification model, which can be adapted in many different ways using techniques such as feature mapping to capture nonlinear relationships between features, and regularization to prevent overfitting on our relatively small dataset.

2.2 Optimization Techniques

1. *k-Nearest Neighbours:* kNN is non-parametric.
2. *Decision Trees:* Decision trees are also non-parametric. In learning the decision tree, splits are chosen to maximize information gain.
3. *(Multinomial) Logistic Regression:* Gradient-based optimization is the natural choice for logistic regression. Stochastic gradient descent or mini-batch gradient descent would both be appropriate given our small dataset. Alternatively, other gradient-based optimization techniques such as limited-memory BFGS could be applied.
   If using SGD, decay or adaptive learning rate could help speed up convergence. For regularization, L2 regularization is a common choice although L1 could be worth trying as well.
   As convergence from gradient descent may not be feasible, we can try stopping after a fixed number of epochs / iterations or once the gradient is sufficiently small.

2.3 Validation Method

Recall that the first 70% of the dataset remains after excluding the test set. We can designate the first 50% as the training set and the middle 20% as the validation set. As with

the test set, it would not be appropriate to randomly select the validation set among the first 70% (or similarly, to use cross validation), because there are three data points per user answer and spreading a single user's answers between training set and validation set may cause data leakage.

Alternatively, this could be resolved by clustering data points from the same student, then applying cross-validation to the clustered data.

2.4 Hyperparameters

1. *k-Nearest Neighbours*
   (a) k, the number of nearest neighbours. Based on the heuristic $k < \sqrt{N} = \sqrt{275} \approx 17$, we can vary k in the range $k = 1$ to $k = 20$.
   (b) The distance metric. Canonical choices include Euclidean metric, Manhattan metric, cosine similarity, uniform metric, L^p metric. Different metrics will slightly change how the features interact, which can be important for capturing particular relationships between the numerical and categorical features.
2. *Decision Trees*
   (a) The maximum depth of a decision tree. A tree that is too deep will likely overfit the training data, while a tree that is too shallow cannot classify the data sufficiently precisely. An appropriate range for tree depth would be 3 to 20.
   (b) Minimum samples per leaf. A leaf with too few samples likely overfits the training data, while a leaf with too many samples likely means the tree did not learn more complex patterns in the data. A reasonable range would be 2 to 50.
   (c) Similarly, minimum samples to split a node.
   (d) Criterion. The information gain of a split may be computed using Gini impurity or entropy. By producing different splits, different criteria might better represent the data.
3. *(Multinomial) Logistic Regression*
   (a) Regularization method (L2, L1) and coefficient (0.01 to 10). This allows us to vary the degree to which large weights are penalized, because while we want the model to consider a combination of all the features, it is plausible that certain features are more predictive than others.
   (b) Maximum number of epochs / iterations, if early stopping. This represents an efficiency-accuracy tradeoff, as stopping earlier would be more computationally feasible but often less accurate. 100-5000 iterations should be suffice to capture both underfitting and overfitting models.
   (c) Various feature mappings. Different feature mapping will cause the logistic regression model to capture different relationships between the data. We can apply no feature mapping (assume linear relationship), or polynomial feature mapping of various degrees.
   (d) We can even do feature engineering with Term Frequency–Inverse Document Frequency to represent the text-structured data in a linear way.

2.5 Evaluation Metrics

- *Accuracy:* the number of correct predictions divided by the number of predictions. This is the standard evaluation metric for machine learning models.
- *Precision:* the number of correct positive predictions divided by the number of positive predictions. By focusing on positive outputs, precision provides insight into the reliability of the model's predictions with respect to each label. This can help with model interpretability, and thus it can inform hyperparameter tuning in ways that may not be apparent from accuracy. A model with many false positives would likely have a low precision, although the accuracy could still be relatively high.
- *Recall:* the number of correct positive predictions divided by the number of positive ground truth labels. Similarly to precision, this can measure the model's ability to identify a particular label. It is slightly different in that it compares the number of true positives to the number of ground truth positives, rather than the positive predictions. As a result, it can provide insight into the model's overall class coverage, ensuring that no class is disproportionately neglected in the model's predictions. A model with many false negatives would likely have a low recall, but it could still have a relatively high accuracy or precision.
- *F1:* $2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$, the harmonic mean of precision and recall. This provides a more balanced account of the previous two insights. Thus it provides insight into both false positives and false negatives.