Task 1

Q1.1

First, we preprocess the dataset by computing answer complexity and size from explanations and encoding professions. We split the student dataset into training and holdout subsets, where a Random Forest classifier is trained on the student data. The initial precision and recall scores are calculated on the student holdout set. Then, we incrementally sample non-student data and add it to the holdout set, measuring how the classifier’s precision and recall degrade. Thresholds are identified for when the drop in metrics reaches 5% and 10% of the initial values, and we compute the average non-student sample size causing these drops.

```
Accuracy score:  0.8065843621399177 Initial: Precision: 1.0 Recall:
0.04081632653061224
```

```
Average Number of Non-Students Added: Average for 5% drop: 96.0 Average for 10%
drop: 106.0
```

Q1.2

```
old_model_precision = 0.9711286089238845
```

```
old_model_recall = 0.8809523809523809
```

```
Minimum Non-Student Size to have same outcome with old model: 0
```

However, the result indicates that even without adding any non-students, the current precision and recall are already outside the acceptable range compared to the old model. This is because the behavior of the current model differs significantly, due to differences in training data.

Task 2

Q2.1

For readability, the Flesch Reading Ease Score is used as a key metric. This score considers the average sentence length and the average number of syllables per word to determine how easy a text is to read. The score ranges from 0 to 100, with higher scores indicating easier readability. For this explanation, a threshold of 40 is chosen, reflecting the assumption that readers will have a certain level of computer knowledge and familiarity with technical terms.

Score Range	Readability Level	Intended Audience
90-100	Very Easy	5-th grade or younger
80-89	Easy	6th grade level
70-79	Fairly Easy	7th grade level
60-69	Standard	8 to 9th grade level
50-59	Fairly Difficult	High school students
30-49	Difficult	College students
0-29	Very Difficult	Advanced Readers

We aimed at threshold of 40 since this is an explanation of a bug inside a computer program, fairly the person who will read this will be having a certain level of computer education and will be familiar with the argon.

Additionally, to ensure that the consolidated bug description aligns closely with the ground truth, we use cosine similarity as a metric to measure the semantic similarity between the two texts. The text is first converted into vector representations using SentenceTransformer embeddings, which capture the meaning of the sentences. Cosine similarity then calculates the angle between these vectors, with a value close to 1 indicating high similarity. A threshold of **0.7** ensures that the merged output remains relevant and aligned with the ground truth.

```
FailingMethod HIT01_8 required 2 explanations to pass the threshold

FailingMethod HIT02_24 required 2 explanations to pass the threshold

FailingMethod HIT03_6 required 4 explanations to pass the threshold

FailingMethod HIT04_7 required 93 explanations to pass the threshold

FailingMethod HIT05_35 required 2 explanations to pass the threshold

FailingMethod HIT06_51 required 22 explanations to pass the threshold

FailingMethod HIT07_33 required 7 explanations to pass the threshold

FailingMethod HIT08_54 required 36 explanations to pass the threshold
```

Task 3

In the code base, we aim to select a diverse set of high-quality bug explanations by evaluating them based on readability, semantic similarity to expert-written ground truth, and diversity across worker demographics and linguistic styles. we proceed on processing a dataset of bug explanations and ground truth references, computes readability scores using the Flesch Reading Ease metric, and measures semantic similarity using Sentence-BERT embeddings with cosine similarity. We ensured that explanations that fall below predefined readability and similarity thresholds are filtered out, to make sure only clear and relevant explanations are considered in our solution.. We then used Entropy to assess diversity across attributes like profession, gender, country, and programming language, and a final subset is selected by grouping explanations based on these attributes while prioritizing those with the highest semantic similarity.

Our results include a refined dataset of high-quality explanations, diversity metrics indicating the distribution of worker attributes, and a final diverse subset that balances readability, accuracy, and representation. Maximum readability and similarity scores are reported to highlight upper performance limits, and an additional trade-off analysis examines the relationship between maintaining high semantic similarity and preserving diversity. Finally, we saved the selected explanations to a CSV file for further use, ensuring a well-rounded and diverse set of explanations is available for analysis or deployment.
Given the explanation above,

3.1

We measured diversity using Shannon entropy, which quantifies how evenly different categories are represented in features like profession, gender, country, and programming language. It works by calculating the probability of each category and using a formula to determine how spread out or concentrated they are. We found that higher entropy means greater diversity, while lower entropy indicates dominance by a few categories. For example, high entropy in Worker.country means explanations come from many countries, while low entropy in Worker.gender suggests one gender is overrepresented. Entropy is used because it is a reliable way to measure variation in categorical data.

Below are the results 3.1

- Worker.profession: 1.379
- Worker.gender: 0.464
- Worker.country: 1.565
- Worker.programmingLanguage: 3.395

3.2

The maximum readability and semantic similarity are determined by identifying the highest Flesch Reading Ease score and cosine similarity in the dataset. The most readable explanation has the highest readability score, while the explanation most aligned with the ground truth has the highest similarity score. For example, a max readability of 95.20 indicates very easy readability, and a max similarity of 0.985 shows near-perfect alignment with the ground truth.

Below are the results 3.2

- Max readability in dataset: 206.84
- Max semantic similarity in dataset: 0.856

3.3

To find the maximum diversity for explanations with near-maximum semantic similarity (compromising readability), we filtered the dataset to include explanations with similarity scores within 95% of the highest value. Shannon entropy is then computed for diversity features like profession, gender, country, and programming language. The results show how diverse the most semantically similar explanations are, with some attributes exhibiting high entropy (e.g., country) while others remain less diverse (e.g., gender). This trade-off analysis helps understand the balance between maintaining high similarity, diversity, and readability.

Below are the results 3.3

- Worker.profession: 0.662
- Worker.gender: -0.000
- Worker.country: 0.900
- Worker.programmingLanguage: 1.494