Final Report

● Ungraded

1 Day, 19 Hours Late

Group

Tuneer Roy
Javier Farach
Julia Fremberg
...and 1 more
View or edit group

Total Points

- / 15 pts

Question 1

Report 15 pts

Question assigned to the following page: 1

Final Report

App: https://fun-facts-2c47073e0b4e.herokuapp.com/

GitHub: https://github.com/tuneerroy/fun-facts

Basic project information

Name of your project

FunFacts

Name of your teammates

Javier Farach, Tuneer Roy, Adam Thomson, Julia Fremberg

Give a one-sentence description of your project. Please use the name of the project in your description.

FunFacts is a web application that uses the crowd to generate interesting facts and fiction to then use in a game where crowd members compete against each other to determine who is the best at distinguishing fact from fiction.

Logo for your project.

What problem does it solve?

Question assigned to the following page: 1

Education! It helps educate people around the world and heightens opportunities for people to find subjects that interest them. It's similar to "Games With a Purpose" where humans are encouraged to post and learn about interesting content while treating it like a game.

What similar projects exist?

A similar project would be "The Wiki Game" where people race to find a target Wikipedia site from another random site and, while having fun, it also encourages people to learn random facts about the world. Other similar sites like Twitter and Reddit where people post content together that is interesting with some portions of the sites being educational.

What type of project is it?

A business idea that uses crowdsourcing.

What was the main focus of your team's effort?

Something in-between. We built and deployed a full-stack web application using Python, FastAPI, MongoDB, TypeScript, and React, but we also did a lot of data analysis on the surveys done through Amazon MTurk.

How does your project work? Describe each of the steps involved in your project. What parts are done by the crowd, and what parts will be done automatically?

First, there is the data collection stage. There are a few ways that we get the data: Amazon MTurk, our web app, and LLMs. The MTurk data is retrieved from the crowd, and the web app is also meant to get data from a crowd (from users who input fiction/facts).

Then there is the quality control stage. We use MTurk again and also manually filter some results to get a high-quality dataset of facts/fiction. Through the app, there is also a moderation system where moderators can approve/reject facts/fiction (and the majority vote is assumed as the correct answer). Again, MTurk and the moderators are work done by the crowd.

Next, there's the deployment stage where these pieces of information are incorporated into a game. This is completely automatic where each piece of information (depending on their ratings assigned by the moderators / MTurk QCers) is shown to a user during a game.

Finally, the user has access to a leaderboard page where they can see how they are faring against other players in terms of being able to distinguish fact from fiction (and human-generated fiction from GPT-generated fiction).

Provide a link to your final presentation video. Give the full URL to your YouTube video or your Google Drive video and make sure it is publicly available (OK to keep it unlisted).

Question assigned to the following page: 1

https://drive.google.com/file/d/12DfO8wqRZ8sLY5yeh4FKDTOa7szjGgH7/view?usp=sharing

Which two sections below did you pick for your in-depth analysis?

"What the crowd gives you" & "Project Analysis"

The Crowd

Who are the members of your crowd?

The members of our crowd are everyday people. This includes gig workers like those on MTurk, classmates, and the people who engage on our application/platform.

For your final project, did you simulate the crowd or run a real experiment?

- Simulated crowd
- Real crowd

We had a mix of a real crowd and a simulated crowd. The simulated crowd came from our classmates and also from our work in applying quality control to the data entries that we got from MTurk. We also had real MTurkers generate data, though, and also do quality control surveys on other MTurker responses.

If the crowd was simulated, how did you collect this set of data?

Classmates and LLMs.

If the crowd was simulated, how would you change things to use a real crowd?

Have more quality control questions since a random person will be more likely to try to game the system than a classmate.

If the crowd was real, how did you recruit participants?

We used Amazon MTurk to get participants.

How many unique participants did you have?

97 statements submitted by UPenn (13 contributors)

100 statements submitted by MTurkers (100 contributors)

For quality control: 124 unique workers with a total of 1300 responses to statements

Incentives

Question assigned to the following page: 1

What motivation does the crowd have for participating in your project?

For the users, they get to play a game that helps them learn and through game mechanics are rewarded for their improvement and success. The Turkers who provide quality control and some of the first statements to be classified are motivated through pay; Tukers get $0.06 for each statement they submit, and $0.01 for each statement that they classify as a fact or a myth.

How do you incentivize the crowd to participate? Please write 1-3 paragraphs giving the specifics of how you incentivize the crowd. If your crowd is simulated, then what would you need to do to incentivize a real crowd?

We incentivized the Turkers through financial means. They were paid $0.06 for each statement that they submitted. This is higher than the $0.01 they are given for the classification tasks, mainly because if they submitted a fact they must have also provided a source. This takes more time than simply clicking on a few radio buttons and moving sliders, so we wanted to pay them more for this task. For classification, they are simply presented with a statement and determine whether they think the statement is a fact or a fiction. Included in this is a slider that allows the Turker to quantify how confident they are in each of their responses. Because this is a simpler and quicker task, we felt that it was fair to pay them less for each statement they classify.

For the users of the app, the incentive is education and enjoyment. As you play the game more and become more proficient at identifying facts vs. myths, you receive a higher score and climb the leaderboard. This competition also acts as an incentive for the user, as they will strive to stay ahead of those behind them and catch those who have a higher score.

Did you perform any analysis comparing different incentives?

We did not perform an analysis comparing the different incentives.

If you compared different incentives, what analysis did you perform? If you have a graph analyzing incentives, include a graph here.

We did not compare different incentives for these different groups.

What the crowd gives you

What does the crowd provide for you?

The crowd performs a variety of tasks for our application. First, they provide the content. We used MTurk, our classmates, and ChatGPT to come up with a bunch of facts and fiction to serve as the source for our game. We also have crowds serving as moderators in two senses. First, for actually filtering the facts and fiction and ensuring their correctness, we have moderators both on the application and through MTurk to ensure quality control. We

Question assigned to the following page: 1

also envision a Reddit-like system where users of the game can upvote/downvote pieces of facts and fiction that they find interesting or might even just be wrong. These rankings detect the frequency distribution of the facts/fiction in the game. Finally, we have the crowd play the game and provide traffic to the website!

Is this something that could be automated? (answered below)

If it could be automated, say how. If it is difficult or impossible to automate, say why.

The pipeline is already automated on the website where a user can submit a fact and it will automatically show up on a moderator's TODO page for review. The rating system is also already automated. If we wanted to bring MTurk into the mix, that might not be as easy to moderate but would still likely be possible through MTurk API. We could also find some way of verifying facts/fiction through automatic checking via fact repositories/dumps or even through the help of LLMs in finding/verifying sources. The actual crowd playing the game should not be automated as that is the whole purpose of the application.

Did you train a machine learning component from what the crowd gave you?

No.

If you trained a machine learning component, describe what you did.

N/A. However, we probably could train some ML model to act as reviewers for pieces of facts/fiction or perhaps even a critic model that reviews and rates some of these statements. Even an actor-critic model might be interesting to train in where the actor generates facts/fiction and the critic responds.

Did you analyze the quality of the machine learning component? For instance, did you compare its quality against crowd workers using an n-fold cross-validation?

N/A

If you have a graph analyzing a machine learning component, include the graph here.

N/A

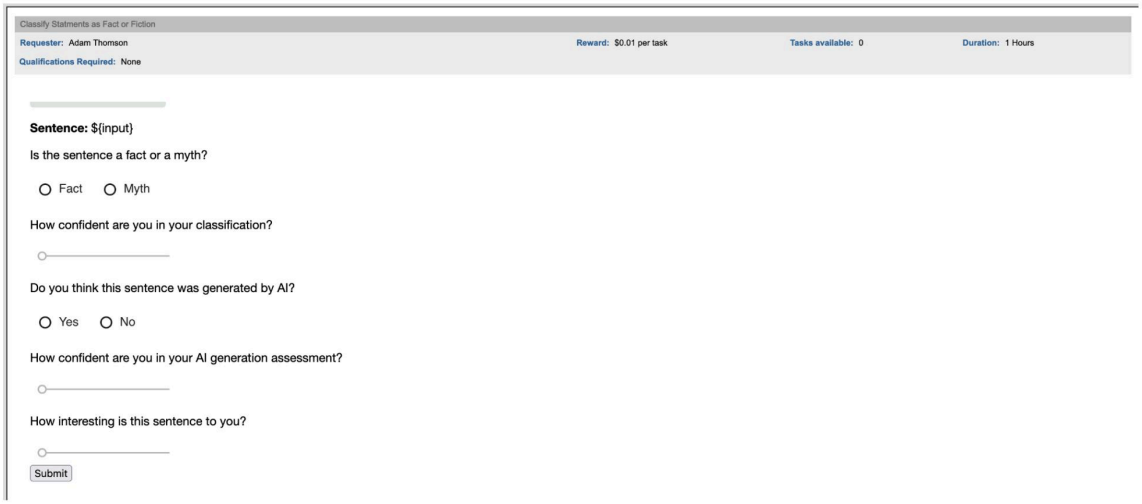
Did you create a user interface for the crowd workers? Answer yes even if it's something simple like an HTML form on MTurk.

Yes, our data collection for this project was run through MTurk. There were two forms: one to submit statements as fact or fiction, and one to classify statements as fact or myths, along with sliders to indicate the confidence of their classifications.

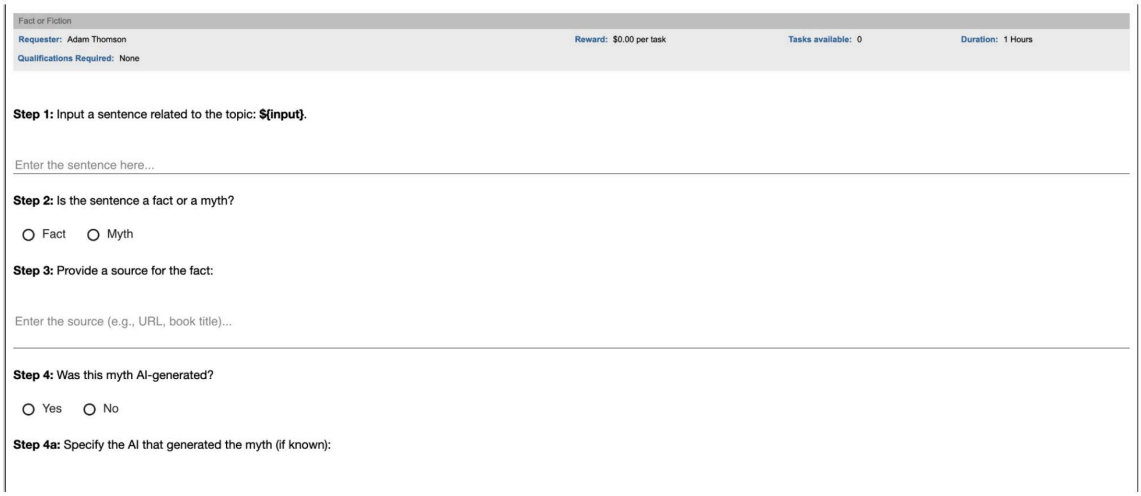
If yes, please include a screenshot of the crowd-facing user interface in your report. You can include multiple screenshots if you want.

Question assigned to the following page: 1

Below is a screenshot of the UI that a Turker saw for our classification HITs:



Here is a screenshot of the UI that Turkers saw when submitting facts and myths. (Here it is important to note that the question asking if the myth was AI generated was conditionally shown upon the answer being ‘Myth’ in the second question):



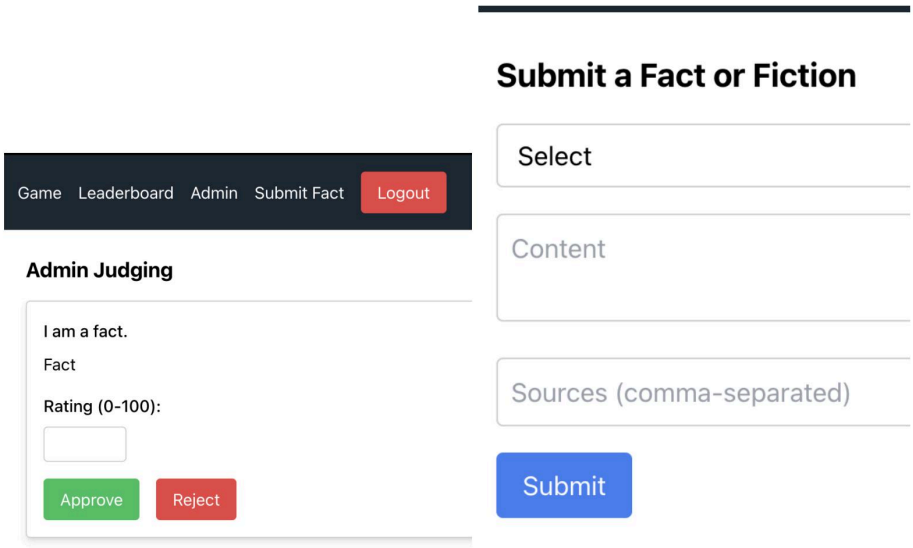
Additionally, here is the bottom of the page for submitting facts and myths that did not make it into the screenshot above:

Question assigned to the following page: 1

Describe your crowd-facing user interface. This can be a short caption for the screenshot. Alternatively, if you put a lot of effort into the interface design, you can give a longer explanation of what you did.

So we have a registration/login page where it's easy to make an account. Regardless of what URL you go to, if you're not logged in, you'll be redirected to this page. Once logged in, you come across a few tabs: Game, Leaderboard, Submit Fact(, and Admin if you're… an admin).

The "Game" tab, it's a basic form that shows a statement and the user has to guess whether or not it's a fact or fiction. If the user guesses it's fiction, they also have an option to guess whether or not it's AI-generated fiction. For the "Leaderboard" tab, it shows a list of the users alongside their scores in decreasing order for the scores. The "Submit Fact" page, it's a form that lets the user submit a new fact or piece of fiction. They also have the option to submit sources (it becomes required if the user is trying to submit a fact). For the "Admin" page, all the pieces of facts/fiction that the currently logged admin has not yet judged/reviewed show up where the admin can assign a rating and approve/reject The rating is required if the admin approves the statement.



Question assigned to the following page: 1

Ethics

Should my application exist at all?

Fact or Fiction is designed to analyze myths and facts for educational and entertainment purposes. While this goal is valuable, the nature of user-submitted content introduces ethical challenges, particularly when topics become highly politicized. Statements such as "Should you force people to be vaccinated to go to class?" or "Is climate change real?" reflect deeply divisive issues that risk escalating ideological tensions. Simplifying complex, nuanced topics into binary classifications of "myth" or "fact" can unintentionally misinform users or reinforce existing biases.

This risk is amplified by the potential for subtle bias in how statements are categorized or presented. Without clear context or supporting evidence, classifications might appear to endorse particular viewpoints, alienating users with differing perspectives. Additionally, controversial or misleading statements could gain traction on the platform, inadvertently propagating misinformation or validating fringe beliefs. These outcomes would undermine the application's credibility and harm its educational mission.

To address these concerns, the application must implement stringent content moderation to filter divisive or harmful submissions and emphasize context by providing clear explanations for classifications. Prioritizing educational value over contentious topics and offering a "complex" or "uncertain" label for nuanced issues can reduce misinterpretation. By focusing on transparency, balance, and responsible curation, the application can fulfill its purpose while minimizing the risks of misinformation and polarization.

Does this task potentially expose workers to harm (for example, content moderation)? What effect can it have on them?

The task of collecting and analyzing user-submitted myths and facts has the potential to expose workers to harm, particularly if the submissions include sensitive, offensive, or

Question assigned to the following page: 1

graphic content. While the project may not specifically solicit such material, the open-ended nature of user contributions creates a risk of encountering harmful or emotionally distressing topics. For example, statements about political controversies, violence, or discrimination could lead to stress, discomfort, or moral injury for workers reviewing them.

For crowd workers on platforms like MTurk, these effects can be compounded by the lack of institutional support or resources to manage emotional well-being. Unlike traditional employees, crowd workers may not have access to counseling, debriefing sessions, or workplace protections. Regular exposure to controversial or harmful content could lead to cumulative mental health effects, including burnout or vicarious trauma, particularly if moderation duties are involved.

Are you fairly compensating the workers for their time?

The compensation structure for your tasks paid workers $0.06 per fact or fiction submitted and $0.01 per sentence classified. While internal testing by your group indicated that submitting 10 facts took approximately 3 minutes (equating to $12/hour) and sentence classification took 1 minute (equating to $0.60/hour), the average times reported by MTurk workers were significantly longer. For example, workers took 12 minutes and 22 seconds on average to submit facts or fiction, reducing their effective earnings to $2.91/hour, which is well below ethical standards. This discrepancy suggests that many workers may have struggled with task requirements or left their browsers idle, inflating reported times.

The low rate for sentence classification, $0.01 per sentence, results in an hourly rate far below any acceptable standard, regardless of whether workers met your internal benchmarks or reported averages. While factors such as task unfamiliarity, unclear instructions, or distractions could explain the disparity in time, the current rates fail to ensure that workers earn a fair wage. Ethical guidelines for crowdwork suggest that workers should earn at least the equivalent of minimum wage for their efforts, which these tasks currently do not achieve.

Is your evaluation sound? Do the conclusions you reach stand up to scientific scrutiny?

A critical component of standing up to scientific scrutiny is the quality of the dataset. However, our dataset relies on user-submitted statements classified as fact or fiction. Sentiment comparisons do provide us with valuable insights, but we do not have rigorous quality control metrics – it is us filtering through. Therefore, this could weaken the scientific legitimacy of our conclusions. Our project’s reliance on user input from diverse sources, such as Penn students and MTurk workers, introduces potential biases. Differences in demographics, cultural backgrounds, or political leanings between groups could skew results. For example, MTurk workers may have different interpretations of "interesting" or "ambiguous" than students. Without accounting for these variations or providing clear

Question assigned to the following page: 1

validation of statement truthfulness, the findings might reflect the biases of the sample rather than universal truths.

Skills

Do your crowd workers need specialized skills?

No, our crowd workers do not need specific skills. There are two stages; the first where users submit statements that are either fact or fiction that the rest of the user base will vote on, and the second is when the user classifies the statement as a fact or a myth. Both of these stages do not require specific technical skills and can be completed by anyone who understands English and knows how to operate a computer or mobile phone properly.

What sort of skills do they need?

They need to have fluency in the English language to know what the statements are saying, and they need to know that AI/LLMs can be used to generate sentences of this nature. They also need to have the technical skills to use a computer and navigate the MTurk Worker platform correctly, and in the world where our web app is deployed to the public, they also need to know how to use the UI. If a user wants to submit a fact or a myth, they also need to know what is considered a sufficient source to back the validity of a fact that is submitted.

Do the skills of individual workers vary widely?

No, the skills of the individual workers do not vary widely. Most people who are fluent English speakers are well suited for this task.

If skills vary widely, what factors cause one person to be better than another?

Skills do not vary widely, for our app they are all qualified enough to be treated the same in their responses.

Did you analyze the skills of the crowd?

We did not analyze the skills of the crowd, only so far as to make sure that they understood English.

If you analyzed skills, what analysis did you perform? How did you analyze their skills? What questions did you investigate? Did you look at the quality of their results? Did you analyze the time it took individuals to complete the task? What conclusions did you reach?

We did not analyze the skills of the workers. If I were to analyze the skills of the workers, I would try and find their general level of education, with the idea that those who had been exposed to more schooling had more general knowledge and would be able to submit more

Question assigned to the following page: 1

interesting facts/myths and be better able to identify which ones are which. I would also think about trying to test the crowd on which ones can identify the AI-generated sentences more effectively. This could be done through a series of ‘gold standard’ questions for AI detection, and if the worker passes a specific threshold level then their answers about AI generation would be given more credence.

Do you have a graph analyzing skills? If you have a graph analyzing skills, include the graph here.

We do not have a graph analyzing the skills of our workers.

Quality Control

Is the quality of what the crowd gives you a concern?

Yes, the quality of the data provided by the crowd is a concern for our project, especially since we are aggregating data from diverse groups like Mturk, students, and, frankly, anyone. Variability in worker expertise, motivation, and the potential for random or systematic bias are intrinsic risks. The spread of misinformation could be a real issue. Also, there are a lot of differences in the types of statements we received. For example, the UPenn group exhibited stronger sentiment values – different demographic or contextual factors influenced the nature of their submissions. This divergence highlights the need to ensure data quality for meaningful analysis.

How do you ensure the quality of the crowd provides?

We do an initial filtering of statements submitted to make sure that they are not offensive. Then we use the wisdom of the crowds to ensure that the label (fact/myth) provided with the statement is the truth – we have workers input their confidence as to which is the correct answer. With this data, we can ensure the quality of the labeling through designated quality control tasks, worker rating iteration, weighted majority voting – based on confidence and worker ratings, and Controlled Survey Design. For instance, we calculate weighted confidence scores, where inputs from more reliable workers (based on past performance) are given greater weight in the final decision. Additionally, redundancy is built into the task design by having multiple workers evaluate the same statement, ensuring that any individual biases or errors are minimized.

If quality is a concern, then what did you do for quality control? If it is not a concern, then what about the design of your system obviates the need for explicit QC? This answer should be substantial (several paragraphs long).

Quality control was a fundamental aspect of your project. We make sure to do quality scoring for workers (i.e. workers were scored based on their consistency and accuracy in labeling tasks and confidence scores served as a proxy for reliability and were weighted

Question assigned to the following page: 1

accordingly). Most importantly, we use an iterative approach for recalibrating worker weights based on their performance over time. This loop ensured that low-quality workers were progressively deprioritized in the labeling process.

Did you analyze the quality of what you got back? For instance, did you compare the quality of results against a gold standard? Did you compare different QC strategies?

We tried several QC strategies and found the iterative worker weightings and majority weighted (by confidence and worker scores) vote to be the most effective.

What analysis did you perform on quality?

We looked at the initial statement submitters to see if their labeling matched the quality control group labeling. They seem to match up the vast majority of the time. There were only a few statements that didn't. These tended to be ambiguous and should be initially filtered out. Going forward, in the case that the statement submitter incorrectly labels it, we will subtract points from their overall quality score.

Aggregation

How do you aggregate the results from the crowd?

We had a few ways of aggregating the results. First, we performed all sorts of merges and data-science aggregation methods to analyze the results and answers from our MTurk surveys. For our application, we aggregate them in the shape of a leaderboard where users and their scores are displayed. We also envisioned a sort of Reddit platform where all facts/fiction appear on a scrollable page where users can upvote/downvote the statements.

Did you analyze the aggregated results?

Yes, we analyzed the results from MTurk.

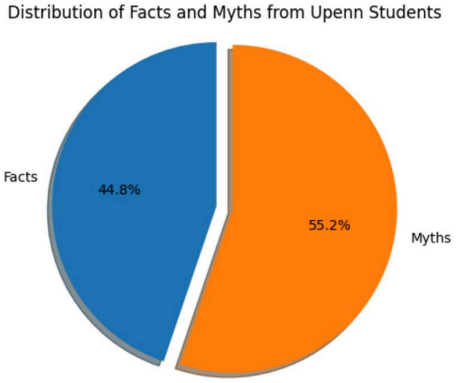
What analysis did you perform on the aggregated results? What questions did you investigate? Did you compare aggregated responses against individual responses? What conclusions did you reach?

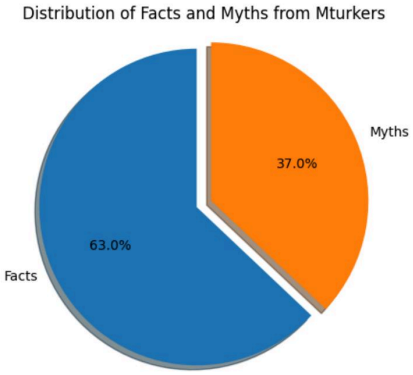
We did aggregation analysis in a Colab notebook. (https://colab.research.google.com/drive/1EC1F2jpYNnPUEE4jIh8guubRYgr1zbLi?usp=sharing). We initially collected statements through two Mturk surveys. One survey was just Upenn students and the other was open to the whole Mturk community. With this aggregate data, we did sentiment analysis on the statements. Interestingly enough, UPenn students submitted statements with stronger sentiment values. We then got quality control data, where Mturkers would decide if a given statement was a fact or a myth and give a confidence and interesting score. We looked at the distribution of results and confidence and interesting scores. Then, we took an iterative approach to rating the quality of the

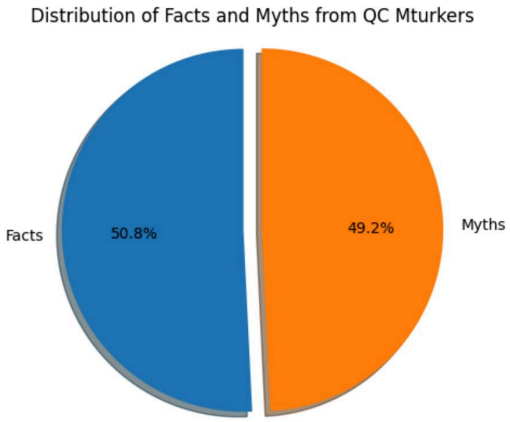
Question assigned to the following page: 1

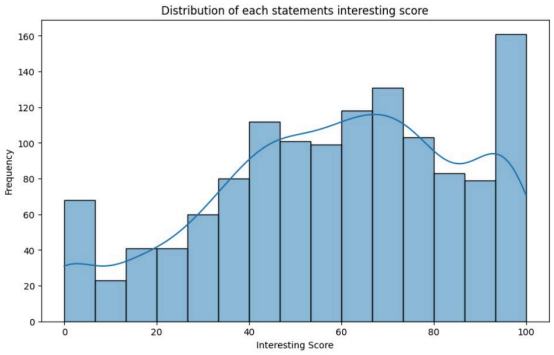
workers and labeling each statement by weighted majority vote – weighted by confidence and worker quality.

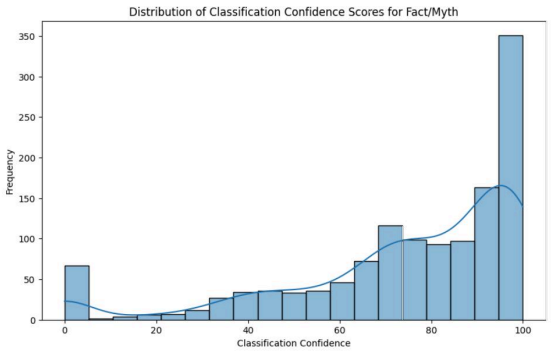
Do you have a graph analyzing the aggregated results? If you have a graph analyzing the aggregated results, include the graph here.



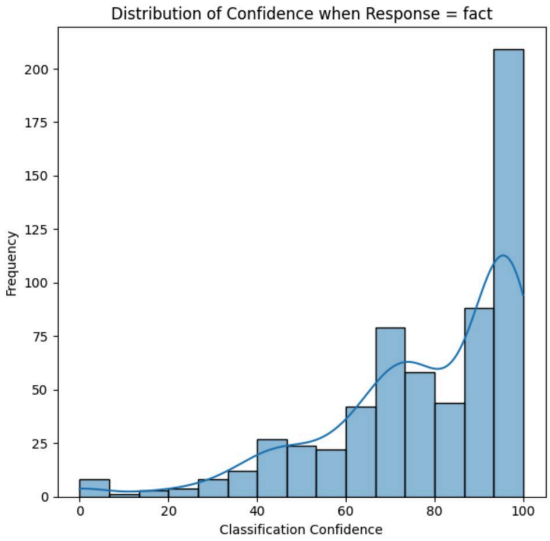


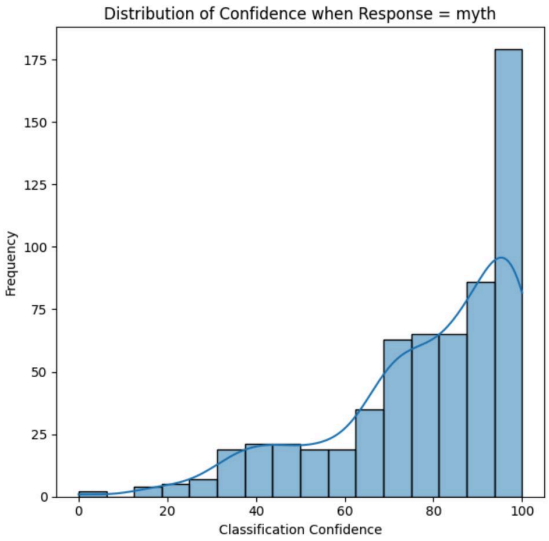


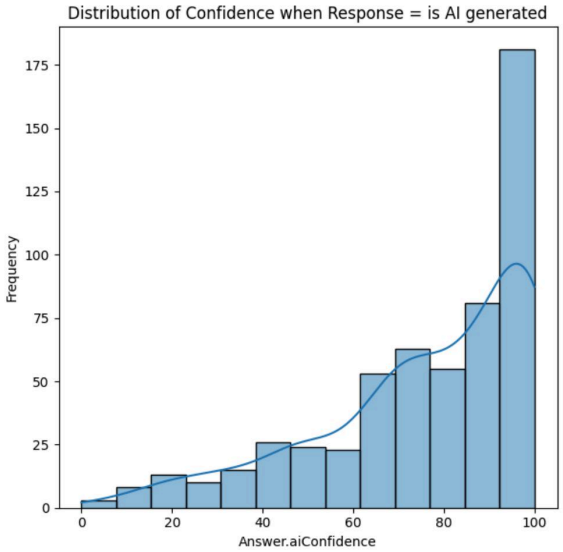


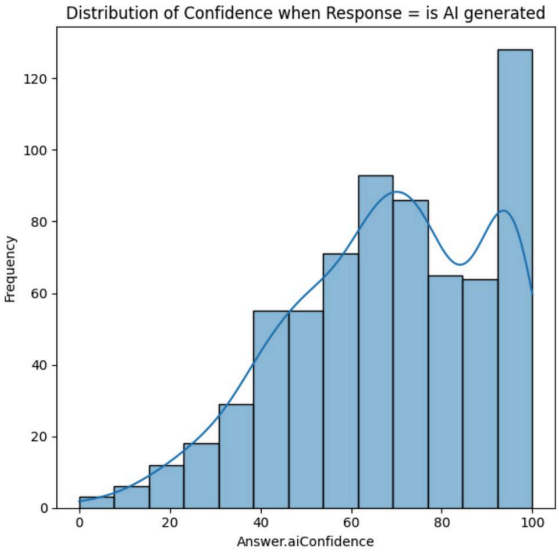


Question assigned to the following page: 1


Distribution of Confidence when Response = fact
Frequency
Classification Confidence
0
20
40
60
80
100
0
25
50
75
100
125
150
175
200


Distribution of Confidence when Response = myth
Frequency
Classification Confidence
0
20
40
60
80
100
0
25
50
75
100
125
150
175


Distribution of Confidence when Response = is AI generated
Frequency
Answer.aiConfidence
0
20
40
60
80
100
0
25
50
75
100
125
150
175

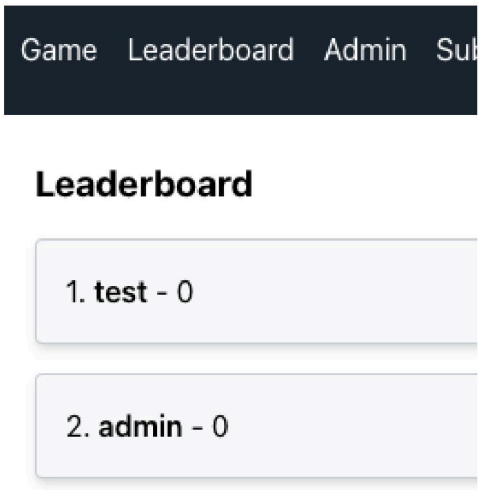

Distribution of Confidence when Response = is AI generated
Frequency
Answer.aiConfidence
0
20
40
60
80
100
0
20
40
60
80
100
120

Question assigned to the following page: 1

	sentence	fact_weighted_sum	myth_weighted_sum	is_fact	is_myth
0	99% of gamblers quit right before they win it ...	101	247	False	True
1	A cause of autism is from a vaccine.	96	322	False	True
2	A cloud weighs around a million tonnes	364	41	True	False
3	AI systems can think and feel just like humans.	65	394	False	True
4	About 70 million soldiers fought for the allie...	256	158	True	False
...	...	...	...	...	...
59	climate change is a figment of my imagination	85	334	False	True
60	from here Spacecraft can stop in space and hov...	108	312	False	True
61	mRNA vaccines helped slow down the COVID pandemic	368	48	True	False
62	quantum entanglement allows for instantaneous ...	83	270	False	True
63	there are laws of Quantum Physics	465	32	True	False

Did you create a user interface for the end users to see the aggregated results? If yes, please include a screenshot of the user interface for the end user in your final report. You can include multiple screenshots if you want.

On the Fun Facts website, users cannot see the aggregated data about how statements are labeled and the weighted votes. They can see the leaderboard of top QC contributors and players which have to do with worker quality (i.e. who is labeling statements that agree with the votes).



Scaling Up

Question assigned to the following page: 1

What is the scale of the problem that you are trying to solve?

The scale of the problem FunFacts seeks to address lies in accurately classifying myths and facts while understanding the nuances of user-submitted data, such as sentiment and ambiguity. Misinformation and the ability to distinguish fact from fiction are widespread issues with significant societal implications, ranging from everyday misconceptions to deeply polarizing topics like climate change and vaccination. The project aims to contribute by creating a dataset and analysis pipeline that sheds light on these dynamics and potentially aids in combating misinformation.

Would your project benefit if you could get contributions from thousands of people?

Currently, the project operates on a relatively small scale, with 197 data points contributed by 113 people. While valuable, this dataset is limited in scope and does not fully capture the complexity or diversity of global perspectives.

If it would benefit from a huge crowd, how would it benefit?

A larger dataset would allow the project to explore broader trends, uncover patterns in how myths and facts are perceived across different demographics, and improve the robustness of any machine learning models applied. Expanding the scale would also address questions about bias and representation in the data. With more contributors, the dataset could better reflect a wide range of cultural, social, and educational backgrounds, which is essential for ensuring the findings are generalizable.

What challenges would scaling to a large crowd introduce?

Scaling the FunFacts project to a large crowd introduces several challenges, particularly regarding data quality, management, and cost. One major issue is the increased likelihood of low-quality or malicious inputs. With more contributors, it becomes more challenging to ensure that submissions are meaningful and accurate, as some participants may submit irrelevant or nonsensical data to complete the task quickly. Effective moderation strategies, such as automated filters and manual reviews, become critical but add complexity and cost.

Another significant challenge is maintaining fair compensation at scale. With thousands of contributors, the total cost of paying workers can escalate quickly. Balancing ethical compensation with budgetary constraints requires careful planning, possibly including task optimization to reduce the time workers spend without compromising quality. Additionally, scaling up could exacerbate variability in worker performance, with some individuals being highly efficient and accurate while others struggle with the task, necessitating robust quality control mechanisms like consensus scoring or qualification tests.

Did you perform an analysis about how to scale up your project? For instance, a cost analysis?

Question assigned to the following page: 1

We conducted a cost analysis to evaluate the feasibility of scaling up the FunFacts project. Currently, the dataset includes 197 data points contributed by 113 participants, with an estimated cost of $0.10 per data point, totaling approximately $30, with fees taken into account. To project costs for larger datasets, we calculated that collecting 10,000 data points would cost around $1,000, and scaling further to 100,000 data points would increase the cost to $10,000. This proportional rise in costs highlights the financial challenge of scaling, especially as the number of contributors grows. To address the cost, we even explored task optimization strategies aimed at reducing costs while maintaining data quality. Tasks could be designed to require minimal time investment per contributor with an improved UI and automated validation tools could be employed to check for plausibility. While scaling up is feasible, it requires careful planning to manage costs and ensure data quality. Moderate expansion to 10,000 data points is achievable within the current budget, but larger datasets would necessitate task optimizations and robust quality control mechanisms to remain viable.

What analysis did you perform on the scaling up?

To analyze scaling up, we focused on three key areas: cost projections, quality control, and task optimization. We first conducted a cost analysis to estimate the financial resources required to expand our dataset from its current size of 197 data points to larger datasets. With a current cost of $0.10 per data point, we calculated that collecting 10,000 data points would cost $1,000, and scaling to 100,000 data points would require $10,000, highlighting the financial challenges of large-scale expansion. We also examined potential quality control issues that arise with a larger crowd, such as increased risks of low-quality or irrelevant submissions. To address this, we evaluated mechanisms like qualification tests, redundancy through consensus scoring, and automated validation systems to maintain data reliability. Additionally, we explored task optimization strategies to reduce costs, such as improving task instructions, simplifying the user interface, and streamlining workflows to reduce the time required per task. Our analysis concluded that moderate scaling (up to 10,000 data points) is achievable within the current budget, but larger expansions would require additional funding and strategic optimizations to balance cost, quality, and efficiency effectively.

What questions did you investigate? What conclusions did you reach?

To analyze scaling up, we investigated several key questions: (1) How would the cost of collecting additional data points increase as the dataset grows? (2) What quality control mechanisms would be necessary to maintain data reliability as the number of contributors increases? (3) How could task efficiency be improved to reduce costs without compromising quality? (4) Would scaling introduce new biases, and how could those biases be mitigated? To address these questions, we conducted a cost projection based on the current cost of $0.10 per data point, estimating that 10,000 data points would cost $1,000 and 100,000 data points would cost $10,000. We also considered implementing additional quality control

Question assigned to the following page: 1

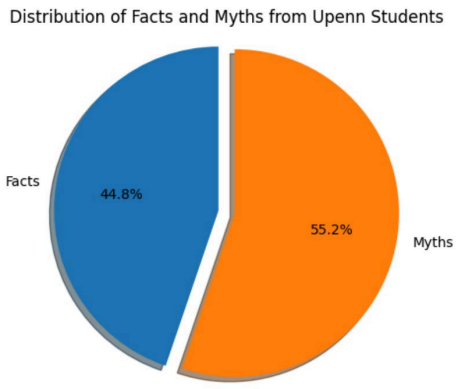
mechanisms, such as qualification tests for workers, automated checks for submission plausibility, and consensus-based scoring to manage the increase in low-quality or irrelevant data submissions. Task optimization strategies, such as simplifying the user interface and providing clearer instructions, were evaluated as methods to improve worker efficiency and reduce overall costs. Our analysis concluded that scaling to 10,000 data points is feasible within our current budget with minimal adjustments, but achieving larger datasets, such as 100,000 data points, would require significant investment in both financial resources and task design optimization. This suggests that while moderate scaling is achievable, extensive scaling would need strategic improvements to remain cost-effective and maintain data quality.

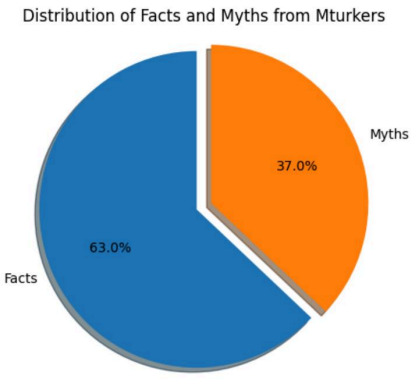
Project Analysis

Did your project work? How do you know? Analyze some results, and discuss some positive outcomes of your project.

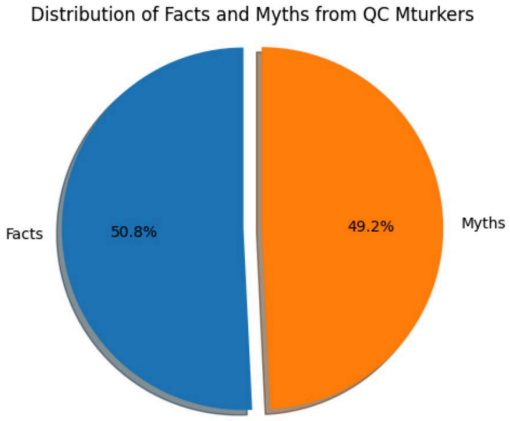
Our project worked as a proof of concept for what a full-fledged app of this nature would look like. Even though we did not have a substantial user base using our platform, we still demonstrated how to utilize crowdsourcing to help with quality control, in addition to collecting the data that serves as the launch pad for our project.

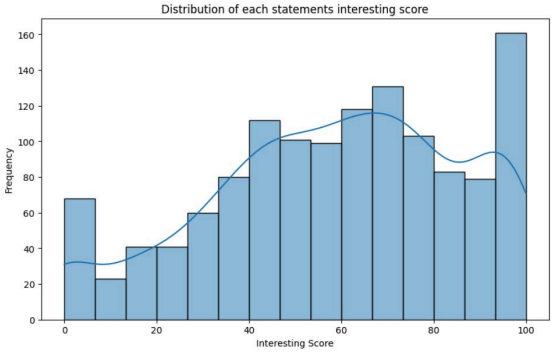
Do you have a graph analyzing your project? If you have a graph analyzing your project, include the graph here.

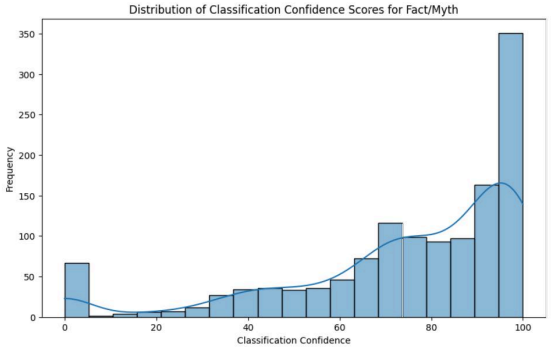


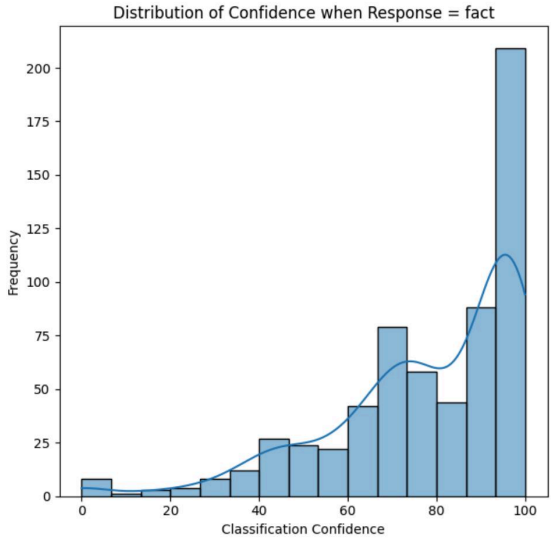


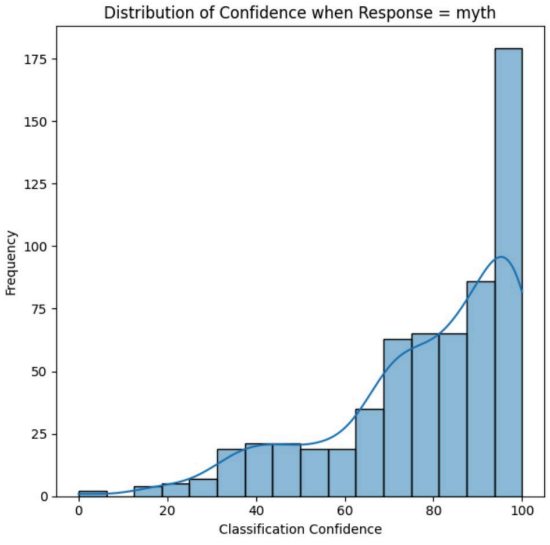
Question assigned to the following page: 1


Distribution of Facts and Myths from QC Mturkers
Facts
50.8%
49.2%
Myths

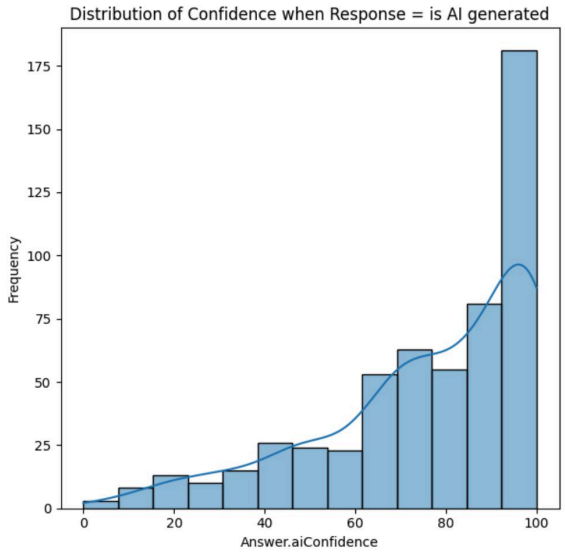

Distribution of each statements interesting score
Frequency
Interesting Score

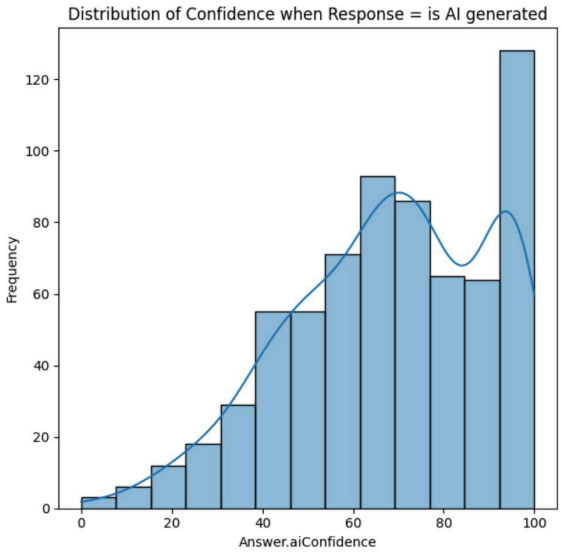

Distribution of Classification Confidence Scores for Fact/Myth
Frequency
Classification Confidence


Distribution of Confidence when Response = fact
Frequency
Classification Confidence


Distribution of Confidence when Response = myth
Frequency
Classification Confidence

Question assigned to the following page: 1





	sentence	fact_weighted_sum	myth_weighted_sum	is_fact	is_myth
0	99% of gamblers quit right before they win it ...	101	247	False	True
1	A cause of autism is from a vaccine.	96	322	False	True
2	A cloud weighs around a million tonnes	364	41	True	False
3	AI systems can think and feel just like humans.	65	394	False	True
4	About 70 million soldiers fought for the allie...	256	158	True	False
...	...	...	...	...	...
59	climate change is a figment of my imagination	85	334	False	True
60	from here Spacecraft can stop in space and hov...	108	312	False	True
61	mRNA vaccines helped slow down the COVID pandemic	368	48	True	False
62	quantum entanglement allows for instantaneous ...	83	270	False	True
63	there are laws of Quantum Physics	465	32	True	False

What were the biggest challenges that you had to deal with?

Creating a functioning web app was much more challenging than we anticipated. The functionality of creating a database to store the sentences, along with the process of getting data from MTurk into the web app was challenging. This involved mapping over the MTurkers as moderators and also ensuring that the data we accumulated from MTurk was piped in the correct format to the MongoDB database. There were other challenges with the web app that we discuss below in the technical challenges section.

Were there major changes between what you originally proposed and your final product?

Question assigned to the following page: 1

There were several changes between our original proposal and the final product, driven by practical constraints, user feedback, and a deeper understanding of the problem space as the project progressed. Initially, the scope of our proposal was broader and more ambitious, aiming to address multiple use cases or include advanced features such as real-time updates or complex machine learning models. However, due to time constraints, technical challenges, and resource limitations, we refined the scope to focus on a specific use case and simplified certain features. For example, instead of implementing a sophisticated deep learning model, we opted for a simpler algorithm that met user needs effectively. Additionally, feedback from early testing revealed that some planned features were unnecessary or overly complex, prompting us to prioritize usability and better align with user expectations. These adjustments allowed us to deliver a functional and reliable final product while staying within the project's constraints.

If so, what changed between your original plan and your final product?

We ended up utilizing Mechanical Turk far more extensively than originally anticipated, with the majority of our data ultimately coming from that platform. Initially, our plan was to rely on a custom-built database as the primary source of data for our project. However, the process of creating and curating this database turned out to be significantly more time-consuming and technically challenging than we had expected. Developing the database required not only collecting high-quality data but also cleaning, labeling, and structuring it in a way that was usable for our system, which exceeded the timeline we had allocated for this task. As a result, we turned to Mechanical Turk as a more efficient solution for rapidly gathering and processing data. The platform allowed us to crowdsource large quantities of labeled data in a relatively short period of time, enabling us to move forward with our project despite the delays. This pivot not only helped us stay on track with our development timeline but also underscored the importance of flexibility and adaptability in the face of unforeseen challenges. While this wasn't our original plan, leveraging Mechanical Turk became a crucial part of our workflow and allowed us to complete the project successfully.

What are some limitations of your product? If yours is an engineering-heavy project, what would you need to overcome to scale (cost/incentives/QC…)? If yours was a scientific study, what are some sources of error that may have been introduced by your method?

One of the main limitations of our product lies in its heavy reliance on data collected from Mechanical Turk, which introduces potential challenges related to data quality and consistency. While we implemented quality control measures, such as attention checks and redundancy in task assignments, there is still a risk that some responses may not fully meet the desired standards of accuracy or reliability. Additionally, the scalability of our product is constrained by the cost and time required to continue using Mechanical Turk for data collection and labeling at a larger scale. As the size of our dataset grows, the costs associated with maintaining high-quality data through crowdsourcing could become

Question assigned to the following page: 1

prohibitive without additional funding or more efficient data collection methods. Furthermore, our system is currently optimized for a specific use case and may not generalize well to other domains without significant modifications, such as retraining models on new datasets or developing domain-specific features. These limitations highlight areas where further development and optimization are needed to improve scalability and robustness.

Did your results deviate from what you would expect from previous work or what you learned in the class?

Yes, our results deviated from what we expected based on previous work and what we learned in class, particularly in terms of the submissions we received from participants. We were surprised to find that Penn students submitted significantly more fiction-based claims compared to fact-based ones, with their fiction submissions often displaying higher levels of emotional variability and creativity. This outcome was unexpected because previous research and classroom examples suggested that participants are generally more likely to rely on factual information, especially in environments where accuracy and reliability are emphasized. The greater frequency of fiction submissions from Penn students may indicate that they found fiction easier or more engaging to generate, or perhaps they perceived it as more entertaining for others to evaluate. Additionally, the higher emotional variability in their fiction claims—such as dramatic or sensational elements—might reflect a deliberate effort to make their submissions stand out or mislead voters. This pattern deviated from the more neutral, straightforward fact/fiction balance we anticipated, and it added an unexpected layer of complexity to interpreting the results and analyzing the voting trends.

If your results deviated, why might that be?

This may have occurred because participants found it more engaging or entertaining to create fictional claims, particularly ones with dramatic or exaggerated elements, which may have felt more creative or rewarding compared to submitting straightforward factual claims. Additionally, the context of the task may have implicitly encouraged creativity over accuracy—for example, if the instructions or incentives didn't explicitly emphasize a balance between fact and fiction, participants may have leaned toward fiction for its flexibility and subjective appeal. Another contributing factor could be the demographic or cultural traits of the participant pool; as Penn students are a highly creative and competitive group, they may have treated the submission process as an opportunity to showcase ingenuity rather than adhere strictly to an expected fact/fiction balance.

Technical Challenges

Did your project require a substantial technical component? Did it require substantial software engineering? Did you need to learn a new language or API?

Yes. We made a full-stack web application using Python, FastAPI, MongoDB, React, and TypeScript. It required writing RESTful API on the backend, setting up the database, and

Question assigned to the following page: 1

writing a frontend that communicated with the backend with all components having error management. Our group had some familiarity with a few of the technologies but were not experts. Specifically, we were completely new to a MongoDB library called Beanie that had some interesting model features and was fairly new to TailwindCSS.

If the project required a substantial technical component, describe the largest technical challenge you faced.

The largest technical challenge faced was deploying the web application on Heroku. It took multiple hours, mostly because DevOps is difficult to debug. Many of the errors/mistakes were tiny and the error messages provided were barely helpful. But after countless pushes to GitHub with slight code changes to the GitHub CLI (GitHub actions), we finally managed to get it working to the point that whenever we push to the repository, the website is redeployed with any changes.

How did you overcome this challenge? What new tools or skills were required? Feel free to nerd out a bit, to help us understand the amount of work that was required.

A lot of Google and use of ChatGPT. It was especially difficult because some of the resources were outdated and our issues were not well documented. Specifically, one of our libraries (the Beanie lib) had issues with the MongoDB bson library but the issues were only showing up when we tried deploying the app on Heroku.

Do you have any screenshots or flow diagrams to illustrate the technical component you described? If so, include the graph here.

Question assigned to the following page: 1

```yaml
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest
    env:
      CI: false
    steps:
      - name: Checkout repository
        uses: actions/checkout@v2

      - name: Install dependencies and build frontend
        run: |
          cd src/frontend
          npm install
          npm run build --omit=dev

      - name: Copy frontend build to backend
        run: |
          rm -rf src/backend/frontend
          cp -r src/frontend/build src/backend/frontend

      - name: Display contents of src/backend
        run: |
          ls src/backend

      - name: Deploy to Heroku
        uses: akhileshns/heroku-deploy@v3.13.15
        with:
          heroku_api_key: ${{ secrets.HEROKU_API_KEY }}
          heroku_app_name: "fun-facts"
          heroku_email: "liun0@seas.upenn.edu"
          appdir: "src/backend"
        env:
          MONGODB_URI: ${{ secrets.MONGODB_URI }}      You, 3 weeks ago • more secrets }:)
```

Heroku logs:

```
024-12-11T00:03:01.656879+00:00 app[web.1]: INFO:     Started parent process [2]
024-12-11T00:03:03.242243+00:00 app[web.1]: INFO:     Started server process [9]
024-12-11T00:03:03.242284+00:00 app[web.1]: INFO:     Waiting for application startup.
024-12-11T00:03:03.242311+00:00 app[web.1]: INFO:     Started server process [10]
024-12-11T00:03:03.242369+00:00 app[web.1]: INFO:     Waiting for application startup.
024-12-11T00:03:03.532744+00:00 app[web.1]: INFO:     Application startup complete.
024-12-11T00:03:03.538611+00:00 app[web.1]: INFO:     Application startup complete.
024-12-11T00:03:03.694425+00:00 heroku[web.1]: State changed from starting to up
024-12-11T00:03:08.000000+00:00 app[api]: Build succeeded
024-12-11T02:23:40.396004+00:00 app[web.1]: INFO:     165.123.227.155:0 - "GET /favicon.ico HTTP/1.1" 200 O

024-12-11T02:23:40.396539+00:00 heroku[router]: at=info method=GET path="/favicon.ico" host=fun-facts-2c470
3e0b4e.herokuapp.com request_id=63f72c09-1125-45fe-b236-0af57216add3 fwd="165.123.227.155" dyno=web.1 conne
t=1ms service=3ms status=200 bytes=798 protocol=https
```

Other info (optional)

Is there anything else you'd like to say about your project?

We're quite proud of the work that we did, and we're glad that we were able to take this class and show everything that we learned from it in our final project.

Question assigned to the following page: 1

If you have additional information about your project that didn’t fit into the above questions, put it here.