

# The Use of LLMs to Annotate Data in Management Research: Warnings, Guidelines, and an Application to Organizational Communication

Natalie A. Carlson<sup>\*1</sup> and Vanessa Burbano<sup>2</sup>

<sup>1</sup>The Wharton School, University of Pennsylvania

<sup>2</sup>Columbia Business School, Columbia University

## Abstract

The emergence of large language models (LLMs) such as ChatGPT has opened new avenues for integrating artificial intelligence into the research process. One of the most promising early applications identified by social scientists is the use of LLMs for data annotation and text classification, a task traditionally performed either manually, with keyword methods, or through the development of customized machine learning tools. However, the benefits and opportunities of this particular use case of the technology remain poorly understood, and critically, the risks and potential downsides of the use of such technology in the research process have been largely unexplored. We examine the performance of ChatGPT in a research task for which we had previously developed a machine learning model at great effort and expense: identifying claims about sustainability in crowdfunding projects, based on the project text. We find that, with some prompt refinement, ChatGPT can easily match the performance of prior methods in annotating texts, at a great reduction in cost and time. However, our study also unearths that minor, seemingly inconsequential prompt variations can result in significantly different labels which, in turn, have implications for the robustness of downstream analyses and result interpretations. Through a sensitivity analysis of hypothetical downstream regressions, we demonstrate how prompt engineering can markedly alter result interpretations, and even enable the potential of unethical fishing for desired results. To guard against these risks, we develop a method we call Prompt Variance Estimation (PVE) to provide robustness to analyses that use LLM-generated labels, and provide instructions and code for its use.

**Keywords:** Artificial Intelligence, Research Methods, NLP, Classification, Crowdfunding

---

<sup>\*</sup>Please send correspondence to ncarlson@wharton.upenn.edu.

# 1 Introduction

The advent of large language models (LLMs) like ChatGPT presents a plethora of potential research applications, representing a paradigm shift across various fields (Eloundou et al., 2023). As articulated by Tu et al. (2023), LLMs “cultivate an extensive understanding of language, common sense, and reasoning, showcasing a semblance of intelligence.” This advancement opens up unprecedented avenues for research applications, with strategy research being no exception. Moreover, because the implications of organizational research often contain key practical insights for firm leaders and managers – whose positions inherently rely on language to shape their strategies – the integration of LLMs into organizational research is likely to impact decision-makers directly.

A particularly promising use case for LLMs in social science research is data annotation. Early experiments, such as those conducted by Gilardi et al. (2023), have demonstrated that ChatGPT substantially outperforms traditional methods like crowd workers on platforms such as mTurk in several annotation tasks. Similarly, Tornberg (2023) found that ChatGPT’s performance in annotating political tweets rivals, and in some cases, surpasses that of experts. Rathje et al. (2023) further illustrate ChatGPT’s capabilities in detecting complex psychological constructs like emotions, suggesting its potential superiority over other automated methods.

The prior methodological standard in this area has typically involved either human coding, often through a complex process of training research assistants, or automated methods ranging from dictionary- or keyword-based methods to custom-built supervised machine learning classifiers (see Miric et al., 2022). Each of these methods has its advantages and disadvantages for various applications; none are without their costs. In this study, we examine how ChatGPT, the most widely used LLM, performs on a research task for which we had previously built a machine learning model: classifying Kickstarter research campaigns along the dimension of product sustainability.

In our exploration, we tested various prompts with ChatGPT to classify the sustainability claims of products described in Kickstarter campaigns. These prompts ranged from simple, direct queries to more complex, nuanced requests. We compared the performance of ChatGPT against human-labeled data and our previously developed Gradient Boosting Machine (GBM) model. Our findings reveal that, with some prompt refinement, ChatGPT can easily – and with high reliability – match the performance of prior methods in annotating texts, achieving this at a significantly reduced cost and time. However, the study also highlights a critical potential downside: minor, seemingly inconsequential variations in prompts can lead to markedly different labels.

We explore how these variations in labels resulting from different prompts might affect hypothetical downstream analyses and conclusions. Specifically, we conduct a sensitivity analysis in which we examined how the estimated relationship between product sustainability claims and performance — in this context, measured as project fundraising success — differs with the varying methods used to measure the independent variable. Our analysis demonstrates that the magnitude, economic meaning, and even direction of the results can vary significantly based on the ChatGPT prompts used.

To guard against inaccuracies stemming from variation in prompts, we develop a method called Prompt Variance Estimation (PVE). In this method we turn to the LLM itself to develop dozens of variations of the initial prompt and estimate the subsequent regression with the data labels generated by each variation. The method provides a roughly normal distribution of coefficient estimates that can provide confidence about the underlying estimated relationship, as well as its sensitivity to minor prompt variations. This distribution allows researchers to effectively bound estimates made using LLM-generated variables. Moreover, the variation in prompts can feasibly be used to simulate the variety of perspectives and priors that might be brought to a data annotation task by human coders.

We discuss the implications of our findings, addressing both the opportunities and pitfalls

inherent in the use of LLMs for data annotation and classification in strategy research. Our paper contributes to the growing body of knowledge on the integration of AI and machine learning in management (e.g., Dell’Acqua et al., 2023; Girotra et al., 2023; Eloundou et al., 2023; Brynjolfsson et al., 2023). While a rich emergent literature focuses on the integration of such technologies within organizations (e.g., Otis et al., 2024; Boussioux et al., 2023), as well as recommendations for educators (Mollick and Mollick, 2023), we aim to provide specific insights and guidelines for researchers (Grimes et al., 2023; Kulkarni et al., 2023). We hope that this work aids in the responsible and rigorous adoption of these technologies in research methodology.

## **2 Data and Methods**

### **2.1 Sample and Objective**

As language models, LLMs excel in settings where the data is unstructured and predominantly text-based, and in which nuances and patterns might elude conventional analysis. This makes the digital, text-rich environment of online platforms an ideal context for investigation. We use as our setting the rewards-based crowdfunding platform Kickstarter. On crowdfunding platforms like Kickstarter, almost all pertinent information about a project is presented online, providing a comprehensive set of information for LLMs to analyze. Unlike traditional settings, where decision-makers might rely on a myriad of external and often intangible factors, Kickstarter’s platform encapsulates the project’s essence in its online presentation, minimizing the variables outside the LLM’s reach.

Crowdfunding has emerged as an increasingly prevalent method for raising capital for entrepreneurial organizations (Agrawal et al., 2014; Mollick, 2014; Sorenson et al., 2016; Yu et al., 2017), and rewards-based crowdfunding is an important type of crowdfunding platform. It includes such platforms as Kickstarter, Indiegogo, Rockethub, and GoFundMe. On rewards-based crowdfunding platforms, entrepreneurial organizations solicit funding in ex-

change for “rewards” (the reward is what, if anything, the funder receives in the exchange)<sup>1</sup>.

We use a database of all Kickstarter campaigns between the years 2009 and 2016, obtained from the CrowdBerkeley Database (managed by the Fung Institute at UC Berkeley). Over 4.2 billion dollars has been pledged through Kickstarter since its inception in 2009, and it is the “largest and most prominent crowdfunding platform in the world” (Mollick and Nanda, 2016, p. 1537). This database includes information about the campaign (for example, project category and whether it has a video) and its current status (success, number of backers, amount pledged), but does not contain the full description of the project from its campaign page. To augment this, we scraped the full campaign text from each project page URL and matched this text to the project metadata from the database. The full sample comprises 295,985 projects.

The objective of the methods described below is to classify each Kickstarter campaign text along the dimension of perceived social benefit. Examples of campaigns classified as offering socially beneficial products or services in the sample include a food supply company with a mission of serving food deserts, a retail store that prioritizes hiring homeless youth, or a drone imaging company used in search and rescue missions.

Given the size of the complete dataset, hand-labeling the projects was not computationally feasible. In the next two sections, we describe first the approach we took in 2017 – a custom-built supervised machine learning model – and then describe the approach to using LLMs to accomplish the same end.

---

<sup>1</sup>Pure donations can be made without receiving anything in exchange, donations can be made in exchange for input on product development, and donations can be made in exchange for the promise of a future product or service if the funding goal is met, for example. The backer of a rewards-based crowdfunding project can function as something akin to a consumer, an investor, and/or a donor, and can be considered an external stakeholder of the new venture.

## 2.2 Supervised GBM Model

First, a subset of projects was labeled by human coders to be used as training data. Second, the labeled subset was used to train a machine learning model, which was applied to the remainder of the data. The training set consisted of 2068 project texts, which were labeled by a combination of workers on Amazon Mechanical Turk and research assistants from two top US universities<sup>2</sup>. A label of social sustainability was assigned if coders identified the product/service as benefitting the broader society (1 if yes, 0 if no). Mechanical Turk workers labeled 1456 projects, with each text rated by three workers. A label of “yes” was assigned to each question if more than half of the coders answered in the affirmative for a given project. Research assistants trained by the authors labeled the remaining 612 projects using the same questions. Between three and four raters assessed each project. There was unanimous agreement 72 percent of the time on average, with less than five percent of projects having a 50-50 disagreement.

We then used these labels to train a Gradient Boosting Machine (GBM) model (Friedman, 2001) to label the remaining projects. GBM is an ensemble method involving a collection of decision trees. GBMs derive their predictive power from the insight that a large ensemble of weak learners – each learner being a short decision tree – can be accurate in the aggregate. As a boosting model, GBM begins with one decision tree and adds more trees one at a time, adjusting the weights on each tree at every iteration of the training process. To create the model training data, the project descriptions were transformed into a document-term matrix, with each column representing a unique term and each row containing the number of times the term appeared in each text. Stopwords (common but semantically insignificant words like articles and pronouns) and words that did not appear at least 20 times or in at least 20 project descriptions of the training set were removed, resulting in a vocabulary of

---

<sup>2</sup>The combination of these labeling sources was intended to balance speed – the use of workers on Amazon Mechanical Turk enabled many training projects to be completed in a short amount of time – with quality in assembling the training set.

2912 terms.

The models each contain a weighted collection of 150 decision trees. When a new data example is passed through the model, the text is converted to a 1 x 2912 vector of counts for each term in the model vocabulary. This vector is then applied to the decision trees, and the weighted combination of each tree’s vote represents a probability estimate that the project should have a positive label for social sustainability.

To prevent the GBM models from overfitting to the training data, we employed cross-validation with three folds. In this process, the training data is divided into three equal parts, and each part in turn is held out as a validation set for a model trained on the remaining two parts. The final selected model maximizes the average performance over the held-out samples, searching over a grid of different tree depths and tree counts. This process helps to ensure that the model does not become overly complex and therefore overfit to the training set. The authors hand-labeled a test set of 250 examples to assess how the final GBM model would perform on unseen data, using our judgments as a standard for comparison. The final models generated a probability estimate that the project should have a positive label. For each model, if the probability was over 50 percent, the project was given a label of one; otherwise, it was given a label of zero.

### **2.3 ChatGPT API**

ChatGPT, developed by OpenAI, is an advanced tool utilizing deep learning and natural language processing to simulate conversational exchanges. Part of the Generative Pre-trained Transformer (GPT) series, this model is a product of significant strides in AI technology (Vaswani et al., 2017; Radford et al., 2019). ChatGPT is part of a broader family of large language models that have transformed the landscape of computational linguistics, offering capabilities that range from writing assistance to complex problem-solving. These models share a common foundation in that they are trained on extensive text corpora and can

generate language-based outputs with varying degrees of sophistication.

These AI systems employ a transformer architecture that effectively manages long-range dependencies in text (Vaswani et al., 2017). As a result, ChatGPT can generate coherent and contextually relevant responses, tailored for a range of tasks from information retrieval to conversation simulation (Brown et al., 2020). Notably, ChatGPT’s knowledge is static, limited to the time of its last training<sup>3</sup>. It does not update in real-time, which means its output should be critically evaluated, especially when used for current information (Bender et al., 2021).

The mechanics of the current application of ChatGPT are fairly straightforward. We use a paid OpenAI account to call the ChatGPT API through the coding language R, borrowing the language of the function used in Rathje et al. (2023). We use the “ChatGPT-3.5-turbo” to maximize speed, with temperature set to zero to minimize randomness in output. Code snippets in Figure 1 (a) show the function used to call the API. We then looped over each item in the data set, using the API to label the main product or service offered in each project as having a societal benefit or not with a zero/one label. The labels are appended to the data as a new column. We repeat this process three times with each prompt to access intercoder reliability.

[INSERT FIGURE 1 ABOUT HERE]

As opposed to the extensive training required for older supervised methods (as in the GBM model above), new generations of LLMs are “few shot learners” (Brown et al., 2020), meaning they can learn from only a few examples, or perhaps no examples (“zero-shot learning,” as described in Tornberg, 2023).

The art of effective prompting, however, is subtle. Initially, the state of the art for how to best make use of these tools (“prompt engineering”) proliferated through trial and error and

---

<sup>3</sup>As of this writing, ChatGPT’s knowledge is current to January 2022.



information sharing through informal sources like blogs. More recently, a growing stream of research has emerged to study the process of prompt engineering for particular desired outcomes (e.g., Meincke et al., 2024). Experts in this domain emphasize that the need for mastery in prompting will decrease over time as LLMs become ever more sophisticated<sup>4</sup>. In the meantime, though, researchers have documented many curious facts, including the fact that prompt performance can sometimes be affected by such innocuous factors as formatting changes (Sclar et al., 2023), and that when LLMs are asked to optimize their own prompts for given outcomes, the resulting text is often bizarre to human eyes. As summarized in Battle and Gollapudi (2024): “[T]he only real trend may be no trend. What’s best for any given model, dataset, and prompting strategy is likely to be specific to the particular combination at hand.”

Nevertheless, a few rules of best practice have emerged, as described by Mollick and Mollick (2023):

1. Tell the AI who it is: Clearly defining the role or identity for the AI can steer its responses to be more in line with the desired outcome.
2. Explicit instructions: It is important to provide unambiguous instructions to ensure the AI’s responses meet the user’s objectives.
3. Inclusion of examples and structured steps: Supplying examples and breaking down tasks into steps can help the AI generate more accurate and relevant responses.

To that end, we test several different prompts for the same task, integrating some of these rules of thumb. The text of each of the prompt preambles can be viewed in Table 1.

[INSERT TABLE 1 ABOUT HERE]

The first prompt is the most basic, asking the LLM only to evaluate whether the product or service described benefits society and answer in a yes or no fashion. The second two prompts test providing the LLM with an identity: first a student research assistant, the second an

---

<sup>4</sup>For a recent discussion, see <https://www.oneusefulting.org/p/working-with-ai-two-paths-to-prompting>.

expert researcher. Aside from the identity statement, the core of the prompt remains the same. The final prompt amends the basic text with examples aimed at reducing false positive responses.

### 3 Results

#### 3.1 Methods Comparison

Table 2 displays how each of the models performed on a set of 250 Kickstarter campaigns, previously designated as a test set with hand labels agreed upon by the authors. For each of the methods, we report overall accuracy, the proportion of false positives (i.e., projects labeled as one when the ground-truth label was zero), and the proportion of false negatives (i.e., projects labeled as zero when the ground-truth label was one). Because LLMs like ChatGPT are non-deterministic (meaning that the same input can generate different responses upon repeat submission, as noted in Reiss, 2023), we ran each prompt three times on the test set to assess intercoder reliability, reported as Fleiss’s Kappa. The accuracy metrics therefore represent the consensus of the three labels, though in practice we found the results were virtually identical for each repetition<sup>5</sup>. Finally, we report the overall cost in USD of using each method to label the test set.

[INSERT TABLE 2 ABOUT HERE]

Our baseline method, the GBM classifier, accurately classified 92 percent of the test set, with a balance of 3.6 percent false positives and 4.4 percent false negatives. Notably, this method was carefully developed over the course of months for this precise task, requiring over 2500 US dollars in labeling and research assistant costs and many researcher hours. The most basic ChatGPT prompt fell short of this performance, with an overall accuracy of 84.8 percent. The bulk of these misclassifications came in the form of false positives (12.8

---

<sup>5</sup>As noted above, we use a temperature setting of zero to minimize randomness in output; with a higher temperature, reliability would be lower.

percent).

Curiously, simply amending the basic prompt by adding an identity statement (e.g., “You are a student research assistant working on a research project on corporate sustainability”) improved overall accuracy by two to three percent. The student identity and expert researcher identity conditions performed similarly, with overall accuracies of 87.2 percent and 86.8 percent, respectively. These too had relatively high rates of false positives, each at around 10 percent.

Finally, adding three sentences to the prompt aimed at providing examples to reduce false positives proved remarkably effective. With this prompt, false positives dropped to 0.4 percent and overall accuracy rose to 93.2 percent, outperforming the GBM classifier. This improvement came at the slight expense of false negatives, which rose to 6.0 percent.

We found the reliability upon repeat runs of the same prompt to be extremely high, consistent with prior work that has found ChatGPT’s reliability to far exceed expert human coders (Tornberg 2023). Indeed, on several of the prompts, the generated results were identical for all 250 instances each of the three times the prompts were repeated.

Finally, we report the cost of each data labeling exercise, using OpenAI’s token-based pricing system. In each case, labeling costs did not exceed 1.50 US dollars, a stark contrast from the cost of acquiring human labels.

### **3.2 Examining False Positives and False Negatives**

A curious and notable feature of the above exercise is the extent to which slight changes in the preamble of the prompt text shifted certain labels in a distinct and replicable manner. To probe this further, we examine specific instances of false positive and false negative labels under the different prompts.

We found that the false positives identified by the basic ChatGPT prompt were not neces-

sarily random, but clustered on projects for which a plausible but vague argument for social benefit could be made. Much of these seemed to focus on ventures with a goal related to educating the public, for instance about a historical topic. Some examples can be seen below.

“I want to create a history buff’s travel guide for the iPhone. The app will put historic data at the traveler’s fingertips in a nicely organized manner. An attractive interface will make it easy to locate and visit historic landmarks such as Victorian or antebellum homes, historic courthouses, light houses, one-room schools, covered bridges, Civil War battle fields, and much more.”

“Romania often looks upon its dictatorial past involving Nicolae and Elena Ceausescu (Romania’s last dictator) as a grim and dark part of its history. Kollectivul TotNoi is a project which tries to put a spin on this, to look at it through a humorous spectrum and exorcise the demons generated by this controversial topic. The porcelain dictator heads (in the form of salt and pepper shakers or penny banks) are practical elements of design which literally tell you how to destroy the myth and perhaps exorcize [*sic*] your own demons with the two figures.”

In both of these examples, adding either of the identity statements (“student research assistant” or “expert researcher”) reliably changed the label produced from one to zero. A possible explanation for this effect is the fact that both identity statements indicated that the project was about corporate sustainability, contextualizing the question about societal benefit and perhaps shifting the focus to constructs more closely associated with sustainability (e.g., environment, conservation) rather than abstract social benefit. Ultimately, though, this is pure conjecture; we discuss more the difficulties of explainability and interpretability below.

The two “identity” prompts are nearly identical except in that in one prompt, the LLM is told that it is a student research assistant, and in the other, an expert researcher. This small change consistently shifts the labels given on only several projects. Below are two excerpts from projects that the “student” identity incorrectly classifies as having a social benefit (false positive) and which the “expert” identity correctly classifies as negative.

“My new graphic novel, A GIRL CALLED ANA TEACHES KITTENS HOW TO DRAW, is meant to provide inspiration for the next generation of young sequential artists and storytellers. It is the first in a series of books aimed especially at young girls, but hopefully entertaining for readers of all ages, girls or boys.”

“Bye Bye Bubba is a transitional pacifier system consisting of three BPA free pacifiers designed to aid parents in weaning children off their pacifiers.”

There is no reliable way to understand precisely why the different identity changes these labels, but one possibility is that the “student” identity generates an underlying embedding that is closer to words about children or young people, causing the LLM to interpret text related to youth differently.

The best-performing ChatGPT prompt that we tested includes example text aimed to reduce false positives. The resulting performance improvement is accompanied by a slight increase in false negatives. In examining these, we find a preponderance of the term “community,” which corresponds directly to the text in the prompt (“Just because a project uses the word ‘community’ doesn’t mean it benefits the broader society”). Below is one excerpt from a project that this prompt incorrectly classified as negative:

“To create 300,000 ‘Impact **Communities**’ around the world with one mission: Participants from our online **community**, along with backers, individuals, & organizations will receive an online and/or live education on how to create their own Impact **Community** (online or offline) to support the ‘greater good’ of this world: We film, follow, and grow 11 of them. This project is designed to destroy the ‘I can’t’ belief system by using technology, film, and engaging and empowering education to create ‘mini-movements’ around the globe.”

While this addition to the prompt is highly effective in reducing false positives and boosting performance overall, it may cause the LLM to overindex on discounting the specific term “community” – though this oversight could likely be ameliorated with additional massaging of the prompt.

### 3.3 Explainability and Interpretability

Interpretability in machine learning, as outlined by Rudin (2019), focuses on the design of models that are inherently interpretable. This means creating models whose internal mechanics and decision-making processes are transparent and understandable to humans from the outset. The goal is to build models where the rationale behind each decision or prediction is clear without the need for additional explanation, as in, for example, a logistic regression.

Explainability, in contrast, deals with providing post hoc explanations for existing black box models. These are models whose internal workings are either too complex for straightforward human comprehension or are proprietary and not openly disclosed. Explainable ML seeks to shed light on how these black box models arrive at their conclusions, often through additional tools or techniques that interpret the model’s outputs and decision-making processes (Marcinkevičs and Vogt, 2020).

LLMs, by this standard, are clearly not interpretable. The underlying mechanics of the generative transformer are so complex as to mystify even its creators, to a certain degree (Hassenfeld, 2023). But are they explainable?

One of the curious features of LLMs is that, asked to explain their “reasoning,” they can successfully generate a block of text that resembles a very plausible justification. These explanations, while compelling, are essentially a mirage. As put by Mollick (2023): “AI doesn’t explain itself, it only makes you think it does. If you ask it to explain why it wrote something, it will give you a plausible answer that is completely made up. When you ask it for its thought process, is not interrogating its own actions, it is just generating text that sounds like it is doing so.”

[INSERT FIGURE 2 ABOUT HERE]

Figure 2 displays an example of how a LLM typically responds when asked to interrogate its reasoning. In this prompt, we asked ChatGPT to explain why its “student researcher” incarnation gave the graphic novel project described above a positive rating, while the “expert researcher” identity gave the same project a negative rating. ChatGPT provides a seductively compelling explanation: the student identity might “emphasize the educational and inspirational aspects of the project, seeing it as a positive contribution to society,” while the expert role might be more likely to “have been more narrowly focused on projects that directly align with sustainability goals, such as environmental protection, sustainable business practices, or social equity in a more traditional sense.”

This rationale underscores the strange paradox of LLMs: when encountering such reasonable-sounding, human-like text, it can be difficult to remember that the explanation is a product of a probabilistic generative process. We cannot interpret this as a strictly “true” explanation, or, crucially, one that will necessarily replicate or extend to analogous situations.

### 3.4 Downstream Consequences of Model Selection

Finally, we consider the downstream consequences of the subtle differences in prompts chosen, *if* the labels generated by the LLM were to be used as independent variables in a subsequent analysis. We compare these results to the labels generated by the original GBM model, as well as the agreed-upon human labels, the latter being the closest thing we have as “ground truth.”

We conduct a simple analysis: what is the relationship between offering a socially beneficial product and the likelihood of project success? In this context, the most relevant measure of project success is a binary measure capturing whether or not the project met its funding goal. Repeating the analysis with each set of labels generated by the different models, we conduct a logistic regression with fixed effects for project category and year, controlling for the logged magnitude of the funding goal. The coefficients on the relevant variable, *Socially*

*Beneficial Product*, can be seen plotted in Figure 3, while Table 3 reports the regression results.

[INSERT FIGURE 3 ABOUT HERE]

[INSERT TABLE 3 ABOUT HERE]

The human labels show a positive association between touting a socially beneficial product and project success, with the coefficient of 1.645 suggesting the likelihood of success is approximately greater for these projects than for those without ( $p = 0.036$ ). The labels produced by the GBM result in a similar coefficient with a value of 1.438 ( $p = 0.023$ ). This finding is consistent with prior work examining the relationship between sustainability orientation and project success in a crowdfunding context (e.g., Calic and Mosakowski, 2016).

By contrast, the labels produced by the differing ChatGPT prompts produce downstream coefficients that are notably dissimilar from the baseline model and from one another. Only one of the prompts – the “student research assistant” identity – produces a result that is consistent with the baseline finding in direction ( $\beta = 1.051$ ,  $p = 0.097$ ). The others are not significantly different from zero, with the point estimate on the final prompt being negative. This is particularly striking given how essentially similar the prompts were to one another. It is also surprising given that the final prompt had the highest overall accuracy; the discrepancy arises from the fact that the “false negatives” in the final prompt were disproportionally drawn from successful projects, while the projects it correctly labeled as socially beneficial were primarily unsuccessful.

#### 4 Recommended Method: Prompt Variance Estimation

The variability in results stemming from small differences in prompts not only challenges the reliability of the findings, but also raises concerns about the potential for manipulation in the pursuit of desired outcomes. In theory, less scrupulous researchers could potentially iterate



on plausible-sounding prompts until they find one that generates the preferred downstream results. To address these concerns, we introduce a novel method we call Prompt Variance Estimation (PVE).

PVE is designed as a systematic approach to capture and analyze the variance introduced by different prompt formulations. By employing a broad array of prompts and conducting downstream analyses for each, the method aims to generate a distribution of estimations, offering a more comprehensive understanding of the results’ sensitivity to prompt variations. This type of approach may be crucial in mitigating the risks associated with the non-deterministic nature of LLMs and ensuring the integrity of research findings.

Because the generation of dozens of diverse prompt variations is in itself arduous, we again turned to the capabilities of ChatGPT. Engaging the model in the chat interface, we tasked it with producing 100 variations of the Kickstarter prompt, guiding it with specific examples to ensure relevance and variety. This conversation is excerpted in Figure 4. Once satisfied with the relevant prompt examples, we asked the model to export the list of prompts to a CSV file for easy use.

[INSERT FIGURE 4 ABOUT HERE]

With the diverse set of 100 prompts in hand, we proceeded to label the projects using each prompt, run the regression analysis, and record the resulting coefficients. For our sample of 250 observations, this entire process took approximately 6 hours and incurred a total cost of 6.65 USD. It’s worth noting that for larger datasets, an optional step of bootstrapping could be introduced at this stage. By randomly selecting subsets of the data, such as 500 observations at a time, the process becomes more manageable and computationally efficient. Code to replicate this process can be found in the supplementary materials, and is excerpted in Figure 5.

[INSERT FIGURE 5 ABOUT HERE]

Finally, we plot the resulting distribution of coefficients obtained from this exercise in a histogram, as shown in Figure 6. This distribution allows for easy visualization of the variance in results, as well as the estimation of confidence intervals and other statistical measures if desired. In this case, the mean of the distribution lies at 0.78, with an estimated standard error of 0.42 and 97 percent of the estimates lying above zero. While a 95 percent confidence interval would not exclude zero, the result is not inconsistent with an interpretation of a modestly positive relationship between product sustainability and project success, as found through other methods and in past work.

[INSERT FIGURE 6 ABOUT HERE]

An optional variation on PVE could incorporate expert human labels on a subset of observations, as a way of adjusting for the varying accuracy of specific prompts relative to a “ground truth” measure, as in the method of design-based supervised learning (DSL) offered by Egami et al. (2024). Using this method, a subset of the data could be randomly selected to be labeled by the researchers or other experts. Estimates from different prompt variations would then be weighted according to their accuracy relative to the expert labels, with more accurate prompt variations receiving higher weight. Implementing this method using the expert human labels for the use case above, we see a slight rightward shift in the weighted distribution of estimates, with a weighted mean coefficient of 0.81<sup>6</sup>.

[INSERT FIGURE 7 ABOUT HERE]

When might it be recommended to implement PVE? We offer two rules of thumb, summarized in Figure 7. The first concerns the key variable’s centrality to the analysis. Independent and dependent variables generated with LLM assistance should generally be subjected to more scrutiny compared to control variables. Note, for example, that the coefficient on the size of the funding goal in Table 3 does not meaningfully change in direction and magnitude

---

<sup>6</sup>Please see the appendix for sample code implementing this method, as well as code implementation in Python.

across models. The second factor relates to the task complexity and subjectivity. The more complex or subjective the task, the greater the need for scrutiny. A general rule of thumb is that tasks with higher intercoder reliability among humans are likely to be less concerning (e.g., identifying all references to a given entity in a document would be considered a less subjective task than identifying emotions like “disgust” from text).

[INSERT FIGURE 8 ABOUT HERE]

## 5 Discussion

The results of this exercise demonstrate why we might expect researchers increasingly to turn to LLMs for data annotation tasks. They have several advantages over human assistants in these functions: they have significantly higher intercoder reliability than humans; make far fewer errors due to inattention, misreading, or typos; and they can be “trained,” through prompt engineering and trial and error, to provide the researcher with the desired annotations more efficiently than perhaps even the best undergraduate assistant. The GBM classifier set a high benchmark with an accuracy of 92 percent, balanced by false positives and negatives. In comparison, the basic ChatGPT prompt showed lower accuracy (84.8 percent), primarily due to a higher rate of false positives. The addition of identity statements and specific examples to the prompts led to incremental improvements in accuracy, with the best-performing prompt surpassing the GBM classifier.

Perusing recent editions of *Strategic Management Journal*, we found several recent articles in which there were clear potential applications for LLMs. These included such varying uses as reading press releases for acqui-hire classification (Boyacıoğlu et al., 2023), coding engagement metrics in nascent market studies (Harmon et al., 2023), and identifying stigmatizing labels in news media (Sadri et al., 2023). Moreover, the cost and time efficiency of LLMs are unparalleled. The ability to process large volumes of data at a fraction of the cost and time required for human coders is a significant advantage, particularly in large-scale research

projects.

However, problems are more likely to arise when the output generated by LLMs like ChatGPT are used in downstream analyses, precisely because the output is so sensitive to the prompts used — and these prompts can be infinitely and nearly costlessly iterated upon. This means that in the hands of less scrupulously researchers any form of prompt “massaging” may be possible until the desired result is achieved; a problem not limited to these novel methods but exacerbated by the ease and low cost of iteration. Moreover, the search dimensions of prompt engineering are effectively boundless, unlike the more constrained gridline search of parameter tuning in other models.

Additionally, our exploration into the explainability and interpretability of LLMs like ChatGPT highlighted a critical challenge. While these models can generate plausible-sounding justifications for their outputs, these explanations are not necessarily reflective of the models’ actual decision-making processes. The inherent complexity and subjectivity of some tasks may increase the risk of error. The more complex or subjective the task, the greater the potential for discrepancies between LLM outputs and human judgment.

We suggest Prompt Variance Estimation – using the LLM itself to generate minor prompt variations for the purposes of sensitivity analysis – as a way of bounding the variance that might come from minor differences in prompts. Variations on this method can both provide reassurance to researchers that their findings do not hinge on the specifics of one particular prompt, as well as guiding peer reviewers in their interpretation of results hinging on LLM-created annotations. The speed and cost benefits of this technology signify a great opportunity, in that it is feasible for reviewers to request additional prompt variations at minimal cost to authors. Academic readers should keep these considerations in mind in order to minimize the risk and maximize the benefit associated with the use of LLMs in research.

In cases when it is not feasible to apply PVE, at minimum, a full accounting of the prompt

engineering process should be provided by the researcher. We suggest a potential roadmap for this process: begin with a subset of data, hand-labeling these observations. Use these labels as a benchmark to develop and refine the initial prompt. Then, iterate by continuously testing and adjusting the prompt based on a sample of results, aiming to align the LLM’s output as closely as possible with the benchmark labels. In the methodology or appendix, provide a detailed account of the prompt development process, including the initial versions of the prompt, the rationale behind each modification, and the impact of these changes on the LLM’s performance. Finally, consider creating a replication package that includes the final prompt, along with examples of iterations and modifications made during the development process. This can serve as a valuable resource for other researchers and help in fostering transparency and reproducibility in LLM-based research.

We will also offer one prospective and unproven future use of PVE: the method could also potentially be used as a supplementary analysis in projects that use human coding, in which the AI-generated prompts could be used to simulate a variety of different perspectives that human coders might bring to a project. Notably, the ChatGPT-generated identities suggested in the prompt variations appear to simulate a variety of plausible human perspectives (e.g., “You are a sociology student examining the social impact of entrepreneurial endeavors,” “You are a volunteer evaluating projects for a non-profit organization”) and objectives or biases (e.g., “Focus particularly on environmental impacts, like sustainability or eco-friendly materials,” “Reflect on aspects like community engagement or addressing social issues,” “Assess the inclusivity of the project, considering diverse communities and populations”). A guided process could similarly generate a desired varieties of perspectives for any human-coded project.

The specifics of how PVE may be used in this way will likely depend on the particular demands of the project. For one example, consider a case in which academic reviewers are concerned about the potential bias introduced by using only undergraduate research assis-

tants in coding a variable of interest. PVE might then be used to simulate the perspective of human coders at various ages and education levels, to examine the sensitivity of results to a hypothetical set of older coders, or those without a college education. Carefully designed experiments may validate or invalidate this use of the method, and we suggest it as a direction for additional research.

## **6 Conclusion**

This study underscores the potential of Large Language Models (LLMs) like ChatGPT in the realm of strategic management research, particularly in the context of data annotation and classification. Our findings reveal that ChatGPT can match or even surpass traditional methods such as human coding and custom-built machine learning models in terms of efficiency, cost-effectiveness, and accuracy. However, a critical insight from our research is the sensitivity of LLM outcomes to prompt engineering. Minor variations in prompts can lead to significant differences in results, highlighting the need for meticulous prompt design and a deeper understanding of LLM mechanics.

As we move forward, future research could build on this foundation by exploring more nuanced aspects of LLM application in strategic management, leveraging the inevitable future generations of AI technologies. Additionally, there is vast potential for investigating the ethical implications and biases inherent in LLMs, ensuring their responsible and equitable use in research. By advancing our understanding and application of these advanced computational models, future research can deepen our insights into strategic management, leading to more accurate and informed analyses.

## References

- Agrawal, A., C. Catalini, and A. Goldfarb (2014). Some Simple Economics of Crowdfunding. *Innovation Policy and the Economy* 14(June), 63–97.
- Battle, R. and T. Gollapudi (2024). The Unreasonable Effectiveness of Eccentric Automatic Prompts.
- Bender, E. M., T. Gebru, A. McMillan-Major, and S. Shmitchell (2021). On the dangers of stochastic parrots: Can language models be too big? *FAccT 2021 - Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623.
- Boussioux, L., J. N. Lane, M. Zhang, V. Jacimovic, and K. R. Lakhani (2023). The Crowdless Future? How Generative AI Is Shaping the Future of Human Crowdsourcing. *SSRN Electronic Journal*.
- Boyacıoğlu, B., M. N. Özdemir, and S. Karim (2023). Acqui-hires: Redeployment and retention of human capital post-acquisition. *Strategic Management Journal*.
- Brown, T. B., G. Krueger, B. Mann, A. Askell, A. Herbert-voss, C. Winter, D. M. Ziegler, A. Radford, and S. McCandlish (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems* 33.
- Brynjolfsson, E., D. Li, and L. R. Raymond (2023). Generative AI at Work. *NBER Working Paper*.
- Calic, G. and E. Mosakowski (2016). Kicking Off Social Entrepreneurship: How A Sustainability Orientation Influences Crowdfunding Success. *Journal of Management Studies* 53(5), 738–767.
- Dell’Acqua, F., E. McFowland, E. Mollick, H. Lifshitz-Assaf, and K. C. Kellogg (2023). Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality. *Working Paper*.
- Egami, N., M. Hinck, B. M. Stewart, and H. Wei (2024). Using Large Language Model Annotations for the Social Sciences : A General Framework of Using Predicted Variables in Statistical Analyses.
- Eloundou, T., S. Manning, P. Mishkin, and D. Rock (2023). GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models. *Working Paper*.
- Friedman, J. H. (2001). Greedy Function Approximation a Gradient Boosting Machine. *Annals of Statistics*, 1189–1232.
- Gilardi, F., M. Alizadeh, and M. Kubli (2023). ChatGPT Outperforms Crowd-Workers for Text-Annotation Tasks. *arXiv Working Paper*.

- Girotra, K., L. Meincke, C. Terwiesch, and K. T. Ulrich (2023). Ideas are Dimes a Dozen: Large Language Models for Idea Generation in Innovation. *Working Paper*.
- Grimes, M., G. Von Krogh, S. Feuerriegel, F. Rink, and M. Gruber (2023). From Scarcity To Abundance: Scholars and Scholarship in an Age of Generative Artificial Intelligence. *Academy of Management Journal* 66(6), 1617–1624.
- Harmon, D., E. Rhee, and Y. H. Cho (2023). Building a bridge to the future: Prospective legitimization in nascent markets. *Strategic Management Journal* 44(11), 2597–2633.
- Hassenfeld, N. (2023). Even the scientists who build AI can’t tell you how it works. *Vox*.
- Kulkarni, M., S. Matere, E. Vaara, E. van den Broek, S. Pachidi, V. L. Glaser, J. Gehman, G. Petriglieri, D. Lindebaum, L. D. Cameron, H. A. Rahman, G. Islam, and M. Greenwood (2023). The future of research in an artificial intelligence-driven world. *Journal of Management Inquiry*.
- Marcinkevičs, R. and J. E. Vogt (2020). Interpretability and Explainability: A Machine Learning Zoo Mini-tour.
- Meincke, L., E. Mollick, and C. Terwiesch (2024). Prompting Diverse Ideas: Increasing AI Idea Variance. *Working Paper*.
- Miric, M., N. Jia, and K. G. Huang (2023). Using supervised machine learning for large-scale classification in management research: The case for identifying artificial intelligence patents. *Strategic Management Journal* 44(2), 491–519.
- Mollick, E. (2014). The dynamics of crowdfunding: An exploratory study. *Journal of Business Venturing* 29, 1.
- Mollick, E. (2023). How to Use AI to Do Stuff: An Opinionated Guide. *Blog Post* (oneusefulthing.org).
- Mollick, E. and R. Nanda (2016). Wisdom or madness? Comparing crowds with expert evaluation in funding the arts. *Management Science* 62(6), 1533–1553.
- Mollick, E. R. and L. Mollick (2023). Assigning AI: Seven Approaches for Students, with Prompts. *Working Paper*.
- Otis, N., R. P. Clarke, S. Delecourt, D. Holtz, and R. Koning (2024). The Uneven Impact of Generative AI on Entrepreneurial Performance. *SSRN Electronic Journal*.
- Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever (2019). Language Models are Unsupervised Multitask Learners. *OpenAI Blog* 1(8).
- Rathje, S., D.-M. Mirea, I. Sucholutsky, R. Marjeh, C. Robertson, and J. Van Bavel (2023). GPT is an Effective Tool for Multilingual Psychological Text Analysis. *PsyArXiv*



*Preprints.*

- Reiss, M. V. (2023). Testing the Reliability of ChatGPT for Text Annotation and Classification: A Cautionary Remark.
- Rudin, C. (2019). Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nature Machine Intelligence* 1, 206.
- Sadri, M., A. Piazza, K. Phung, and W. Helms (2023). The disparate economic outcomes of stigma: Evidence from the arms industry. *Strategic Management Journal* 44(10), 2489–2533.
- Sciar, M., Y. Choi, Y. Tsvetkov, and A. Suhr (2023). Quantifying Language Models’ Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting.
- Sorenson, O., V. Assenova, G.-C. Li, J. Boada, and L. Fleming (2016). Expand innovation finance via crowdfunding. *Science* 354(6319), 1526–1528.
- Tornberg, P. (2023). ChatGPT-4 Outperforms Experts and Crowd Workers in Annotating Political Twitter Messages with Zero-Shot Learning. *arXiv Working Paper*.
- Tu, X., J. Zou, W. J. Su, and L. Zhang (2023). What Should Data Science Education Do with Large Language Models. *arXiv Working Paper*.
- Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 5999–6009.
- Yu, S., S. Johnson, C. Lai, A. Cricelli, and L. Fleming (2017). Crowdfunding and regional entrepreneurial investment: an application of the CrowdBerkeley database. *Research Policy* 46(10), 1723–1737.

## 7 Figures and Tables

Figure 1: Code Snippet: Calling ChatGPT API in R

```
#The "hey_chatGPT" function will help you access the API and prompt GPT
hey_chatGPT <- function(answer_my_question) {
  chat_GPT_answer <- POST(
    url = "https://api.openai.com/v1/chat/completions",
    add_headers(Authorization = paste("Bearer", my_API)),
    content_type_json(),
    encode = "json",
    body = list(
      model = "gpt-3.5-turbo",
      temperature = 0,
      messages = list(
        list(
          role = "user",
          content = answer_my_question
        )
      )
    )
  )
  str_trim(content(chat_GPT_answer)$choices[[1]]$message$content)
}
```

a) Function to call chatGPT API in R

```
for (i in 1:3){
  # Create a "gpt" column
  data$basic_social <- NA

  # Run a loop over your dataset and prompt ChatGPT
  for (j in 1:nrow(data)) {
    question <- "Please read the text of the following Kickstarter campaign and consider the following question.
    Does the main product or service offered benefit the broader society (beyond its consumers)?
    Answer only with a number: 1 for yes, 0 for no. Here is the text:"
    text <- data[j,2]
    concat <- paste(question, text)
    result <- hey_chatGPT(concat)
    data$basic_social[j] <- ifelse(length(result)>0,result,NA)
  }

  # rename gpt column
  data[[as.symbol(paste0('basic_social', i))]] <- data$basic_social
  data[[as.symbol(paste0('basic_social', i))]] <- as.numeric(data[[as.symbol(paste0('basic_social', i))]])
}
}
```

b) Data labeling loop

Note: Code inspired by Rathje et al. (2023).

**Table 1:** ChatGPT Prompt Preambles

Basic Prompt	<i>Please read the text of the following Kickstarter campaign and consider the following question. Does the main product or service offered benefit the broader society (beyond its consumers)? Answer only with a number: 1 for yes, 0 for no. Here is the text:</i>
Basic Prompt + Identity (Student RA)	<b><i>You are a student research assistant working on a research project on corporate sustainability. Please read the text of the following Kickstarter campaign and consider the following question. Does the main product or service offered benefit the broader society (beyond its consumers)? Answer only with a number: 1 for yes, 0 for no. Here is the text:</i></b>
Basic Prompt + Identity (Expert Researcher)	<b><i>You are an expert researcher on corporate sustainability working on a research project. Please read the text of the following Kickstarter campaign and consider the following question. Does the main product or service offered benefit the broader society (beyond its consumers)? Answer only with a number: 1 for yes, 0 for no. Here is the text:</i></b>
Basic Prompt + Examples (False Positive Reduction)	<i>Please read the text of the following Kickstarter campaign and consider the following question. Does the main product or service offered benefit the broader society (beyond its consumers)? <b>In answering, please consider that Kickstarter campaigns often make spurious claims of social benefit. For example, just because a project uses the word ‘community’ doesn’t mean it benefits the broader society. In this case, we are looking for projects that have a concrete benefit, such as donating proceeds to the homeless.</b> Answer only with a number: 1 for yes, 0 for no. Here is the text:</i>

**Table 2:** Model Comparison

	GBM Classifier	ChatGPT: Basic Prompt	ChatGPT: Basic Prompt + Identity (Student RA)	ChatGPT: Basic Prompt + Identity (Expert Researcher)	ChatGPT: Basic Prompt + Examples (False Positive Reduction)
Overall Accuracy	0.920	0.852	0.872	0.872	0.936
False Positives	0.036	0.128	0.100	0.096	0.004
False Negatives	0.044	0.020	0.028	0.032	0.060
Intercoder Reliability (Fleiss’s Kappa)	-	1.000	0.974	0.974	1.000
Cost (USD)	-	0.31	0.42	0.43	1.26

*Notes:* Accuracy metrics computed on a sample of 250 observations and compared to manual labels previously assigned by the authors. Costs represent total cost in USD of labeling all projects one time.

**Figure 2:** Interrogating ChatGPT's Reasoning

**You**

In a previous exercise, I asked ChatGPT to rate a series of Kickstarter projects as either having a benefit to the broader society, or not. In one prompt, I told ChatGPT that it was a "student research assistant working on a research project on corporate sustainability." In another prompt, I told ChatGPT it was an "expert researcher on corporate sustainability working on a research project." Other than this identity difference, the prompt was the same. I am curious why ChatGPT rated certain projects as socially beneficial when it was told it was a student RA, and not as socially beneficial when it was told it was an expert researcher. Can you help me understand the reasoning? Here is an excerpt from one of the projects: "My new graphic novel, A GIRL CALLED ANA TEACHES KITTENS HOW TO DRAW, is meant to provide inspiration for the next generation of young sequential artists and storytellers. It is the first in a series of books aimed especially at young girls, but hopefully entertaining for readers of all ages, girls or boys." Why did the "student RA" version of ChatGPT rate this project as socially beneficial while the "expert researcher" version did not? Please explain the reasoning.



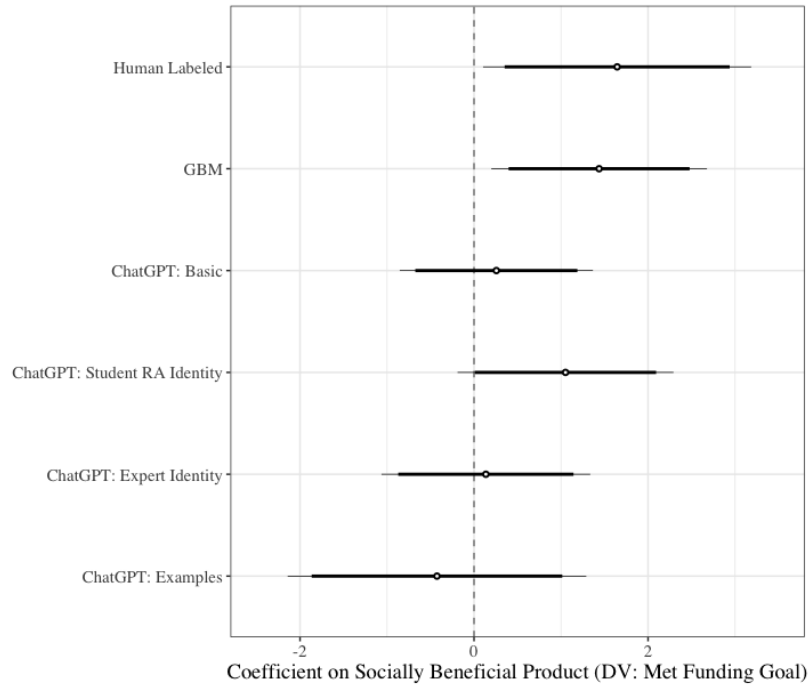
**ChatGPT**

The difference in the responses you observed when ChatGPT was role-playing as a "student research assistant" versus an "expert researcher" in evaluating the social benefit of a Kickstarter project like the graphic novel "A GIRL CALLED ANA TEACHES KITTENS HOW TO DRAW" could be attributed to the perceived scope of expertise and perspective associated with each role.

1. **Student Research Assistant Perspective:** As a student research assistant, ChatGPT might have adopted a broader, more inclusive view of what constitutes social benefit. Students, especially in academic settings, are often encouraged to consider a wide range of perspectives and to value diverse types of contributions. In this context, a graphic novel aimed at inspiring young artists, especially girls, could be seen as socially beneficial because it promotes creativity, inclusivity, and potentially addresses gender disparities in certain fields (like sequential art and storytelling). The student role might emphasize the educational and inspirational aspects of the project, seeing it as a positive contribution to society.
2. **Expert Researcher Perspective:** When role-playing as an expert researcher in corporate sustainability, ChatGPT's assessment criteria might have been more narrowly focused on projects that directly align with sustainability goals, such as environmental protection, sustainable business practices, or social equity in a more traditional sense. An expert in this field might prioritize projects that have a clear, direct impact on sustainability issues. Therefore, a graphic novel, while culturally and educationally valuable, might not be seen as directly contributing to corporate sustainability goals. The expert researcher role would likely apply a more specialized lens, possibly overlooking broader cultural or educational benefits in favor of projects with more explicit ties to sustainability.

It's important to note that these responses are based on the roles ChatGPT was asked to assume and the associated perspectives it was directed to adopt. The model itself does not have personal beliefs or knowledge beyond its training data, but it can simulate different viewpoints based on the context provided in the prompt.



**Figure 3:** Downstream Sensitivity to Different Labeling Models

*Notes:* Coefficients from logistic regression with binary indicator for Kickstarter success (1=met funding goal, 0=did not meet funding goal) as dependent variable, controlling for year, logged funding goal, and category. Full regression results in Table 3.

**Table 3:** Downstream Sensitivity to Different Labeling Models

	<i>Dependent variable:</i>					
	Project Successfully Met Goal					
	<i>ChatGPT Labeled</i>					
	Human Labeled	GBM	<i>Basic</i>	<i>Student</i>	<i>Expert</i>	<i>Example</i>
	(1)	(2)	(3)	(4)	(5)	(6)
Socially Beneficial Product	1.645 (0.786) p = 0.037	1.438 (0.632) p = 0.023	0.256 (0.566) p = 0.651	1.051 (0.633) p = 0.097	0.136 (0.612) p = 0.825	-0.426 (0.876) p = 0.627
Ln (Project Goal)	-0.757 (0.201) p = 0.0002	-0.702 (0.194) p = 0.0003	-0.656 (0.187) p = 0.0005	-0.700 (0.189) p = 0.0003	-0.653 (0.189) p = 0.001	-0.631 (0.186) p = 0.001
Category FE	Yes	Yes	Yes	Yes	Yes	Yes
Year FE	Yes	Yes	Yes	Yes	Yes	Yes
Observations	250	250	250	250	250	250
Log Likelihood	-84.597	-84.202	-86.882	-85.540	-86.960	-86.864
Akaike Inf. Crit.	353.194	352.404	357.763	355.079	357.920	357.729

*Note:* Logistic regressions with standard errors in parentheses.

**Figure 4: PVE: Prompt Variation Process**

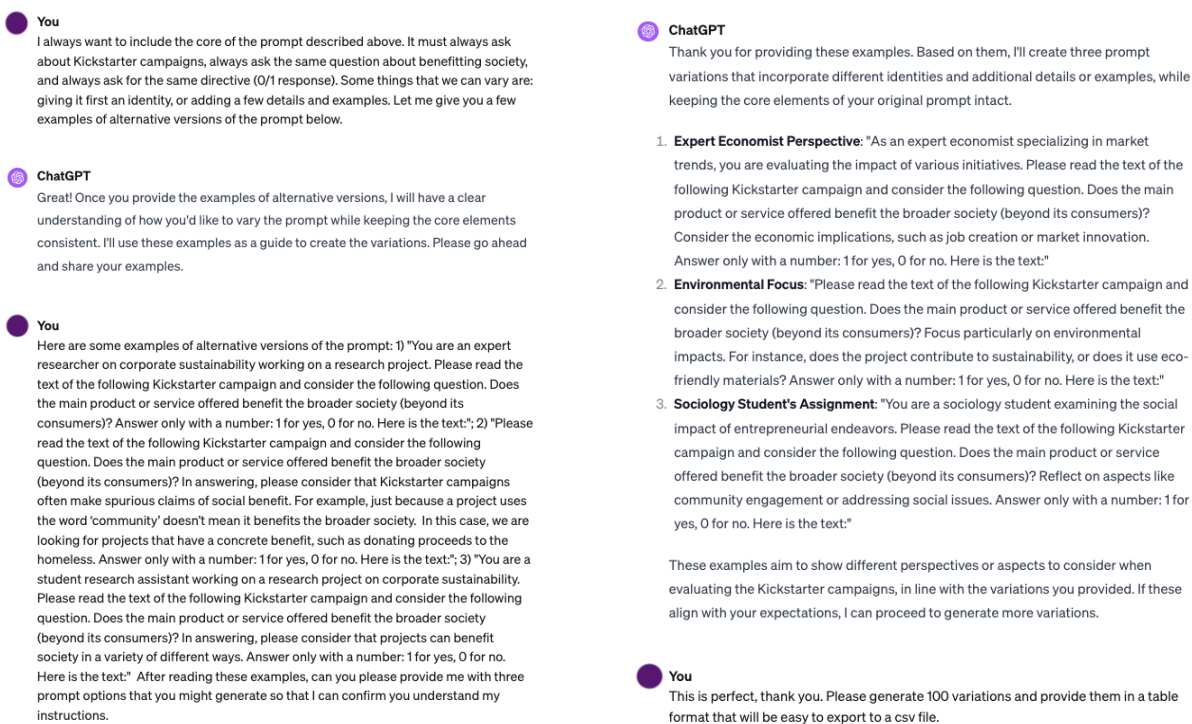


Figure 5: PVE Implementation: Code Excerpt

```
##### PROMPT VARIANCE ESTIMATION (PVE) -- WITHOUT WEIGHTING

# Read in dataset
data <- read.csv("DIRECT TO FILE")

# Read in prompts generated by ChatGPT: csvfile with three columns ("prompt", "coefficient", "pvalue")
prompts <- read.csv("DIRECT TO FILE")

for (i in 1:nrow(prompts)) {

  # Create a "gpt" column
  data$gpt_label <- NA

  # OPTIONAL: if data is large, can select a random subsample here
  # data <- data[sample(nrow(data), 500), ]

  # Run a loop over your dataset and prompt ChatGPT
  for (j in 1:nrow(data)) {
    question <- prompts$prompt[i]
    text <- data$text[j]
    concat <- paste(question, text)
    result <- hey_chatGPT(concat)
    data$gpt_label[j] <- ifelse(length(result)>0,result,NA)
  }

  # rename gpt column
  data$gpt_label <- as.numeric(data$gpt_label)

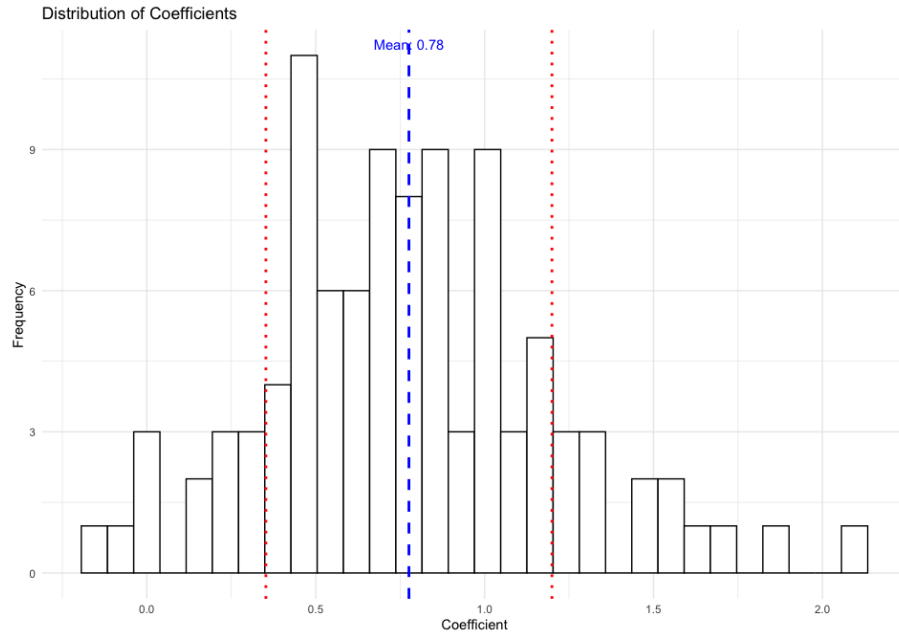
  ## REGRESSION (model specified as desired, ChatGPT-generated variable should be specified first)
  regression <- glm(success ~ gpt_label + lgoal + subcategory + year, data=data, na.action=na.omit, family="binomial")
  prompts$coefficient[i] <- summary(regression)$coefficients[2,1]
  prompts$pvalue[i] <- summary(regression)$coefficients[2,4]
}

## PLOT RESULTING DISTRIBUTION OF COEFFICIENTS
ggplot(prompts, aes(x=coefficient)) + geom_histogram(color="black", fill="white")

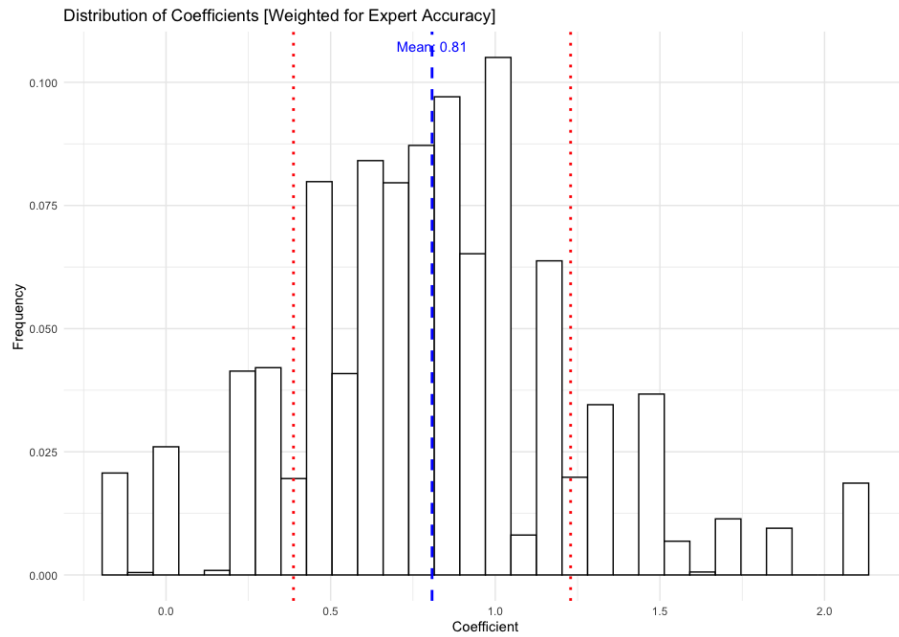
# Calculate summary statistics for coefficients
coef_mean <- mean(prompts$coefficient, na.rm = TRUE)
coef_sd <- sd(prompts$coefficient, na.rm = TRUE)

# Print summary statistics
cat("Mean of coefficients:", coef_mean, "\n")
cat("Standard error of coefficients:", coef_sd, "\n")

# PLOT RESULTING DISTRIBUTION OF COEFFICIENTS WITH MEAN AND ±1 SE
ggplot(prompts, aes(x = coefficient)) +
  geom_histogram(color = "black", fill = "white", bins = 30) +
  geom_vline(aes(xintercept = coef_mean), color = "blue", linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = coef_mean - coef_sd), color = "red", linetype = "dotted", size = 1) +
  geom_vline(aes(xintercept = coef_mean + coef_sd), color = "red", linetype = "dotted", size = 1) +
  labs(title = "Distribution of Coefficients",
       x = "Coefficient",
       y = "Frequency") +
  theme_minimal() +
  annotate("text", x = coef_mean, y = Inf, label = paste("Mean:", round(coef_mean, 2)), vjust = 2, color = "blue")
```

**Figure 6: PVE: Resulting Coefficient Distribution**

Notes: Values represent the point estimates of the estimated coefficient for the effect of *Socially Beneficial Product* on *Success*, over 100 different LLM prompt variations used to generate the independent variable. Dotted lines represent the mean and standard deviation of the distribution of coefficients.

**Figure 7: PVE: Resulting Coefficient Distribution, Weighted by Accuracy**

Notes: Values represent the point estimates of the estimated coefficient for the effect of *Socially Beneficial Product* on *Success*, over 100 different LLM prompt variations used to generate the independent variable. Prompt variations with higher accuracy relative to human accuracy labels receive higher weights. Dotted lines represent the mean and standard deviation of the distribution of coefficients.



Figure 8: Summary of PVE Recommendations

MEASURE’S CENTRALITY TO ANALYSIS			
		Low	High
MEASURE’S SUBJECTIVITY	Low	<p><b>Likely methods:</b> dictionary/keyword, LLMs</p> <p><b>Need for PVE or other sensitivity analysis if using LLMs:</b> Low (unless measure highly correlated with error term)</p>	<p><b>Likely methods:</b> dictionary/keyword, supervised ML, human labeling, LLMs</p> <p><b>Need for PVE or other sensitivity analysis if using LLMs:</b> High</p>
	High	<p><b>Likely methods:</b> human labeling, LLMs</p> <p><b>Need for PVE or other sensitivity analysis if using LLMs:</b> Medium (may be especially useful if measure is highly correlated with error term)</p>	<p><b>Likely methods:</b> supervised ML, human labeling, LLMs</p> <p><b>Need for PVE or other sensitivity analysis if using LLMs:</b> Very high</p>

## Appendix

**Figure 9:** Expert-Weighted PVE: Sample Code Implementation in R

```
##### PROMPT VARIANCE ESTIMATION (PVE) -- WITH WEIGHTING

# Read in dataset
data <- read.csv("DIRECT TO FILE")

# Read in prompts generated by ChatGPT: csvfile with three columns ("prompt", "coefficient", "pvalue")
prompts <- read.csv("DIRECT TO FILE")

# Initialize list to store weights
weights <- numeric(nrow(prompts))

# Filter data to include only rows with human labels
human_labeled_data <- data[!is.na(data$HUMAN), ]

for (i in 1:nrow(prompts)) {

  # Create a "gpt" column
  data$gpt_label <- NA

  # OPTIONAL: if data is large, can select a random subsample here
  data <- if (nrow(data) > 500) data[sample(nrow(data), 500), ] else data

  # Run a loop over your dataset and prompt ChatGPT
  for (j in 1:nrow(data)) {
    question <- prompts$prompt[i]
    text <- data$text[j]
    concat <- paste(question, text)
    result <- hey_chatGPT(concat)
    data$gpt_label[j] <- ifelse(length(result) > 0, result, NA)
  }

  # Convert gpt_label column to numeric
  data$gpt_label <- as.numeric(data$gpt_label)

  # Filter gpt_label to match the human-labeled subset
  human_labeled_data$gpt_label <- data$gpt_label[!is.na(data$HUMAN)]

  # Calculate accuracy
  correct_predictions <- sum(human_labeled_data$gpt_label == human_labeled_data$HUMAN, na.rm = TRUE)
  total_predictions <- sum(!is.na(human_labeled_data$gpt_label))
  accuracy <- correct_predictions / total_predictions
  weights[i] <- accuracy

  # REGRESSION (model specified as desired, ChatGPT-generated variable should be specified first)
  regression <- glm(success ~ gpt_label + lgoal + subcategory + year, data = data, na.action = na.omit, family = "binomial")
  prompts$coefficient[i] <- summary(regression)$coefficients[2, 1]
  prompts$pvalue[i] <- summary(regression)$coefficients[2, 4]
}

# Normalize weights to sum to 1
weights <- weights / sum(weights, na.rm = TRUE)

# Calculate weighted mean and standard error of coefficients
weighted_coef_mean <- sum(prompts$coefficient * weights, na.rm = TRUE)
weighted_coef_se <- sqrt(sum((prompts$coefficient - weighted_coef_mean)^2 * weights, na.rm = TRUE))

# Print summary statistics
cat("Weighted mean of coefficients:", weighted_coef_mean, "\n")
cat("Weighted standard error of coefficients:", weighted_coef_se, "\n")

# Optionally, return summary statistics as a dataframe
weighted_summary_stats <- data.frame(
  weighted_coef_mean = weighted_coef_mean,
  weighted_coef_se = weighted_coef_se
)

# Print the summary statistics dataframe
print(weighted_summary_stats)

# PLOT RESULTING DISTRIBUTION OF COEFFICIENTS WITH WEIGHTING
ggplot(prompts, aes(x = coefficient)) +
  stat_bin(aes(weight = weights), bins = 30, color = "black", fill = "white") +
  geom_vline(aes(xintercept = weighted_coef_mean), color = "blue", linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = weighted_coef_mean - weighted_coef_se), color = "red", linetype = "dotted", size = 1) +
  geom_vline(aes(xintercept = weighted_coef_mean + weighted_coef_se), color = "red", linetype = "dotted", size = 1) +
  labs(title = "Distribution of Coefficients",
       x = "Coefficient",
       y = "Frequency") +
  theme_minimal() +
  annotate("text", x = weighted_coef_mean, y = Inf, label = paste("Mean:", round(weighted_coef_mean, 2)), vjust = 2, color = "blue")
```

Figure 10: PVE: Sample Code Implementation in Python

```

## PROMPT VARIANCE ESTIMATION (PVE) -- without expert weighting

import requests
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt

# Function to call ChatGPT API
def hey_chatGPT(answer_my_question, api_key):
    headers = {
        'Authorization': f'Bearer {api_key}',
        'Content-Type': 'application/json'
    }
    body = {
        'model': 'gpt-3.5-turbo',
        'temperature': 0,
        'messages': [{'role': 'user', 'content': answer_my_question}]
    }
    response = requests.post('https://api.openai.com/v1/chat/completions', headers=headers, json=body)
    response_json = response.json()
    return response_json['choices'][0]['message']['content'].strip()

# Load dataset
data = pd.read_csv("DIRECT TO FILE")

# OPTIONAL: if data is large, select a random subsample here
data = data.sample(n=500) if len(data) > 500 else data

# Load prompts
prompts = pd.read_csv("DIRECT TO FILE")

# Initialize columns for coefficients and p-values
prompts['coefficient'] = np.nan
prompts['pvalue'] = np.nan

# Your API key
my_API = 'your_api_key_here'

for i in range(len(prompts)):

    # Create a 'gpt_label' column
    data['gpt_label'] = np.nan

    # Run a loop over your dataset and prompt ChatGPT
    for j in range(len(data)):
        question = prompts.loc[i, 'prompt']
        text = data.iloc[j, 'text']
        concat = f'{question} {text}'
        result = hey_chatGPT(concat, my_API)
        data.at[j, 'gpt_label'] = result if result else np.nan

    # Convert gpt_label column to numeric
    data['gpt_label'] = pd.to_numeric(data['gpt_label'], errors='coerce')

    # REGRESSION (model specified as desired, ChatGPT-generated variable should be specified first)
    X = data[['gpt_label', 'lgoal', 'subcategory', 'year']].dropna()
    y = data['success'].loc[X.index]
    X = sm.add_constant(X)
    regression = sm.Logit(y, X).fit(dis=0)
    prompts.at[i, 'coefficient'] = regression.params[1]
    prompts.at[i, 'pvalue'] = regression.pvalues[1]

# Calculate summary statistics for coefficients
coef_mean = prompts['coefficient'].mean()
coef_se = prompts['coefficient'].std()

# Print summary statistics
print("Mean of coefficients:", coef_mean)
print("Standard error of coefficients:", coef_se)

# Optionally, return summary statistics as a dataframe
summary_stats = pd.DataFrame({
    'coef_mean': [coef_mean],
    'coef_se': [coef_se]
})

print(summary_stats)

# PLOT RESULTING DISTRIBUTION OF COEFFICIENTS WITH MEAN AND ±1 SE
plt.hist(prompts['coefficient'].dropna(), bins=30, color='white', edgecolor='black')
plt.axvline(coef_mean, color='blue', linestyle='dashed', linewidth=1)
plt.axvline(coef_mean - coef_se, color='red', linestyle='dotted', linewidth=1)
plt.axvline(coef_mean + coef_se, color='red', linestyle='dotted', linewidth=1)
plt.xlabel('Coefficient')
plt.ylabel('Frequency')
plt.title('Distribution of Coefficients')
plt.annotate(f'Mean: {coef_mean:.2f}', xy=(coef_mean, plt.ylim()[1]), xytext=(coef_mean, plt.ylim()[1]*0.9),
    arrowprops=dict(facecolor='blue', shrink=0.05), color='blue')
plt.show()

```

Figure 11: Expert-Weighted PVE: Sample Code Implementation in Python

```

for i in range(len(prompts)):

    # Create a 'gpt_label' column
    data['gpt_label'] = np.nan

    # Run a loop over your dataset and prompt ChatGPT
    for j in range(len(data)):
        question = prompts.loc[i, 'prompt']
        text = data.iloc[j, 'text']
        concat = f'{question} {text}'
        result = hey_chatGPT(concat, my_API)
        data.at[j, 'gpt_label'] = result if result else np.nan

    # Convert gpt_label column to numeric
    data['gpt_label'] = pd.to_numeric(data['gpt_label'], errors='coerce')

    # Filter gpt_label to match the human-labeled subset
    human_labeled_data['gpt_label'] = data['gpt_label'][~data['HUMAN'].isna()]

    # Calculate accuracy
    correct_predictions = (human_labeled_data['gpt_label'] == human_labeled_data['HUMAN']).sum()
    total_predictions = human_labeled_data['gpt_label'].notna().sum()
    accuracy = correct_predictions / total_predictions if total_predictions > 0 else 0
    weights.append(accuracy)

    # REGRESSION (model specified as desired, ChatGPT-generated variable should be specified first)
    X = data[['gpt_label', 'lgoal', 'subcategory', 'year']].dropna()
    y = data['success'].loc[X.index]
    X = sm.add_constant(X)
    regression = sm.Logit(y, X).fit(disps=0)
    prompts.at[i, 'coefficient'] = regression.params[1]
    prompts.at[i, 'pvalue'] = regression.pvalues[1]

# Normalize weights to sum to 1
weights = np.array(weights) / np.sum(weights)

# Calculate weighted mean and standard error of coefficients
weighted_coef_mean = np.sum(prompts['coefficient'] * weights)
weighted_coef_se = np.sqrt(np.sum((prompts['coefficient'] - weighted_coef_mean) ** 2 * weights))

# Print summary statistics
print("Weighted mean of coefficients:", weighted_coef_mean)
print("Weighted standard error of coefficients:", weighted_coef_se)

# Optionally, return summary statistics as a dataframe
weighted_summary_stats = pd.DataFrame({
    'weighted_coef_mean': [weighted_coef_mean],
    'weighted_coef_se': [weighted_coef_se]
})

print(weighted_summary_stats)

# PLOT RESULTING DISTRIBUTION OF COEFFICIENTS WITH WEIGHTING
plt.hist(prompts['coefficient'].dropna(), bins=30, weights=weights, color='white', edgecolor='black')
plt.axvline(weighted_coef_mean, color='blue', linestyle='dashed', linewidth=1)
plt.axvline(weighted_coef_mean - weighted_coef_se, color='red', linestyle='dotted', linewidth=1)
plt.axvline(weighted_coef_mean + weighted_coef_se, color='red', linestyle='dotted', linewidth=1)
plt.xlabel('Coefficient')
plt.ylabel('Frequency')
plt.title('Distribution of Coefficients')
plt.annotate(f'Mean: {weighted_coef_mean:.2f}', xy=(weighted_coef_mean, max(weights)),
            xytext=(weighted_coef_mean, max(weights)*0.9),
            arrowprops=dict(facecolor='blue', shrink=0.05), color='blue')
plt.show()

```

Code up through defining API key (my\_API) should be the same as in unweighted version, seen in Appendix Figure 10.