OPTIMIZING BRAND VISIBILITY IN GENERATIVE AI SEARCH: PASSWORD MANAGERS

November 30, 2025

Dziugas Balciunas, 3262030

Martyna Dygdoń, 3230121

Alessandro Ferraiolo, 3238219

Oliwia Piecuch, 3214430

Ayan Sultanov, 3257705

1. Introduction & Research Question

Our project analyses how brands can optimize their visibility in generative AI search, focusing on the global market for password managers. We selected this industry, represented by seven major providers (1Password, Bitwarden, LastPass, Dashlane, Keeper, NordPass, and RoboForm), because purchase decisions within it are driven primarily by price, trust, and perceived security, rather than by trend or impulse. Through a layer of objectivity and careful evaluation, users typically adopt a password manager only after careful consideration of security guarantees, thus any recommendation made by large language models (LLMs) has a high chance of influencing real adoption.

A second reason for concentrating on password managers is the clarity of user intent segmentation. The market can be organised around well-defined query types such as “best free password manager”, “best secure password manager” or “best password manager for business teams”. In our study, we explicitly work with three topics that mirror these intents: Price (“cheap”, “free”, “affordable”), Security (“zero-knowledge”, “encryption”, “privacy”) and Fit/Use (“for families”, “for teams”, “for Mac”, etc.) These elements can be valuable for structured query design and topic-specific analysis of LLM behaviour.

Lastly, the industry has a reasonable number of global players. With only around seven key brands, we can cover virtually the entire competitive set, reducing selection bias and allowing us to interpret differences in LLM recommendations as genuine differences in market signals rather than a byproduct of sampling.

Following this rationale, we formulated the following research question:

Which are the drivers increasing the probability of being suggested by ChatGPT or Gemini?

To answer it, we treat ChatGPT and Gemini as “gatekeepers” that either include or exclude a brand when responding to consumer queries. We construct a structured set of natural-language prompts across the three core topics (Price, Security and Fit/Use) and observe (per-query) whether each of

the seven brands is mentioned in the LLM response. We then interpret these outcomes through the lens of externally observable marketing and visibility signals at brand and topic level. This setup allows us to move beyond speculations about "AI favouring certain brands" and instead quantify which types of signals are most strongly associated with being recommended in generative AI search and how to act on this digital search migration from a managerial standpoint.

2. Data Description & Methodology

Data Collection Strategy

We constructed a dataset of **840 observations**, consisting of **60 unique queries** run across **2 LLMs (**ChatGPT and Gemini) for **7 Brands**. We designed 20 queries for each of the three topics: *Price*, *Security*, and *Fit/Use*. Since we intend to model the probability of a password brand being mentioned in an LLM response; the **Dependent Variable** of the model is a binary *mention* variable (1: if mentioned by an LLM, 0: otherwise). The **Independent Variables** are based on the count of brand presence across specific channels (Reddit, YouTube, LinkedIn, Listicles, Blogs) where data was collected using Google Search Operators two more independent variables are G2 review data (the most cited source by LLMs for tech products reviews) and Lighthouse SEO scores (measuring efficiency and speed of webpages).

Data collection process and dataset construction

In order to transform raw LLM outputs, brand-level metadata, and topic-specific online presence signals into a unified modeling dataset suitable for logistic regression with query fixed effects, we need to create a robust data pipeline. The crucial steps in the dataset construction include: defining entities, generating queries, executing automated API calls, extracting brand mentions, gathering brand-level attributes, and constructing topic–brand fit features. Below, we describe each step in detail.

1) Defining the Entities: Brands Table. We began by constructing a brands table consisting of seven different password-manager providers in our niche: 1Password, Bitwarden, LastPass,

Dashlane, Keeper, NordPass, and RoboForm, differing in target-user, price, security and popularity. For each brand, we also recorded its canonical domain (e.g., 1password.com), which was later used for Lighthouse SEO scoring and domain-level topic-hit queries.

2) Query Design and Queries Table. To elicit realistic LLM recommendations, we created a queries table with 60 unique natural-language prompts. The queries were evenly distributed and connected to the three topics: Price (e.g., "What's the most affordable password manager for everyday use?"), Security (e.g., "Which password manager is best for privacy-focused users?"), Fit/Use (e.g., "What's the best password manager for families sharing accounts?")

(The full list of queries is provided in the Dataset table.)

During the query creation process, we also followed strict **design principles**:

Queries were written in consumer-style natural language, not as templated prompts; Each query was topic-specific but never contained any brand name, which ensured unbiased model output; Phrasing was intentionally varied (e.g., "Which…", "What's the best…", "I need…, etc.) to induce query-level heterogeneity;The final queries table contains columns: *query_id, query_text, topic*.

3) Automated LLM Calls and Responses Table. Exactly one response per query was collected from each LLM model: gemini-2.5-flash and gpt-5-mini. We use these specific models because they are the default, lightweight options that most users encounter when opening the respective web interfaces, making them the most relevant for capturing real-world behavior.

To avoid manual data entry and ensure reproducibility, we implemented a Python API pipeline that automatically: (1) Read each query from *queries.csv*; (2) Sent it to both Gemini and ChatGPT via their official APIs; (3) Stored the resulting text in a structured responses table. See the script in *run_gemini_queries.py* and *run_chatgpt_queries.py* . The resulting dataset contains 60 rows. The *responses_gemini.csv* and *responses_chatgpt.csv* files include columns: *query_id, query_text, topic, response_text*. These tables serve as the foundation for supervised extraction of brand mentions.

4) Programmatic Mention Extraction Using Regex (Mentions Table). Next, we transformed free-text model outputs into a binary dependent variable:

$$\text{mention}_{bq} = \begin{cases} 1 & \text{if brand } b \text{ is referenced in the response to query } q, \\ 0 & \text{otherwise.} \end{cases}$$

To do this systematically, we wrote a Python script using robust regex patterns covering multiple name permutations for each brand, including: exact names (e.g., "1Password", "Bitwarden"); spacing variants (e.g., "1 Password", "Last Pass"); capitalization variants (e.g., "nordpass", "ROBOFORM"); hyphenation or common misspellings. We chose Python over Stata for this task because its mature text-processing libraries and broader ecosystem for scripting and automation make it better suited for large-scale regex matching and data cleaning.

Extract_mentions.py script produced a **mentions table** with one row per (query × brand × model) with the following columns: *query_id, brand, mention_gemini (0/1), mention_chatgpt (0/1).*

5) Brand-Level Attributes: G2, Lighthouse SEO. We manually collected brand-level metadata from: G2 (*avgrating_b_g2, reviewcount_b_g2*) the standard review source for tech products, known for community based, objective reviews and therefore cited very often by LLMs when choosing products in this category. Additionally, we collected Lighthouse SEO using PageSpeed Insights for each brand's homepage: *lighthouse_seo_b (0–100 score),* signaling, among different technical performance, how easily and quickly bots and crawlers (including those of LLMs) can read the content of the webpage.

6) Topic–Brand Fit Measures (Google Search Operators). To quantify how strongly each brand is associated with each topic across the web, we collected topic–brand hit counts using Google search operators, providing the number of indexed pages on google resulting from the input query. For each brand × topic pair (21 pairs total), we executed search queries in five categories:

a. Listicle hits: Articles of the form "Best X password managers". Example structure: *"Best cheap password manager" "BrandName"*

b. Reddit hits: To measure how frequently a brand is discussed in Reddit threads about password managers. Example structure: *"site:reddit.com "BrandName" "password manager" (price OR cheap OR affordable OR free)"*

c. YouTube hits: To measure how often a brand is featured in YouTube videos about password managers. Example structure: "*site:youtube.com "BrandName" "password manager" (security OR secure OR "zero-knowledge" OR privacy)*"

d. LinkedIn hits. To measure how frequently a brand shows up in LinkedIn content about password managers. Example structure: "*site:linkedin.com "BrandName" "password manager" (fit OR families OR family OR business OR teams OR Mac OR iOS)*"

e. Domain topic hits (brand-published content): To measure whether a brand produces content about a topic on its own website, by checking for brand- and topic-related pages on the brand's domain. Example structure: *"site:<brand-domain> "<BrandName>" "password manager" (<topic keywords>)"*

All queries used fixed OR keyword blocks per topic to ensure consistency. The outputs form a *topic_brand_hits.csv* dataset with columns: *brand, topic, listicle_topic_hits_bt, reddit_topic_hits_bt, youtube_topic_hits_bt, linkedin_topic_hits_bt, domain_topic_hits_bt.*

7) Construction of the Final Modeling Dataset. Finally, we merged all intermediate tables into a single wide-format modeling dataset. Final dataset (840 rows) contains: Query-level variables (*query_id, query_text, topic, model (Gemini / ChatGPT), response_text*), Brand-level variables (*brand, avgrating_b_g2, reviewcount_b_g2, lighthouse_seo_b, reviewindex_b)*, Topic–brand features (*listicle_topic_hits_bt, reddit_topic_hits_bt, youtube_topic_hits_bt, linkedin_topic_hits_bt, domain_topic_hits_bt*), Dependent variable (*mention (1/0)*)

In order to construct the dataset, we followed the following pipeline: (1) Expand queries to (query × brand × model) grid; (2) Merge in mentions; (3) Merge brand-level features; (4) Merge topic–brand hits; (5) Export *dataset.csv.* The pipeline was executed using *dataset_constructor.py* script. The

resulting dataset is rich, column-complete, and ready for estimation of our logit models with query fixed effects and clustered standard errors.

Note that we had to use python for this part as stata does not allow APIs of LLMs without installing external add-ons that allow the user to use python on stata. The rest of the analysis is conducted in the stata file “script.do”

Simplifications & Exclusions

During the study design, we removed two variables that we originally were planning to use in the Research Proposal. It is vital to understand why we removed them, as this defines the scope of our results:

The "Wikipedia" Dummy. Every brand in our sample (1Password, Bitwarden, etc.) has a Wikipedia page, so this variable shows no variation. Although Wikipedia is an important source of training data , we cannot statistically estimate its effect here because having a page is essentially a baseline requirement in this industry.

The "llms.txt" Dummy. Only one brand (Bitwarden) currently implements an llms.txt file (a file telling AI bots how to read the site). Including this variable causes **perfect separation** (perfectly predicting the outcome), which prevents model convergence. (it would essentially act as a "Bitwarden dummy"). This would bias the Z-scores and invalidate the model. As a result, **we are measuring marketing effort**, not experimental technical features. It is still worth noticing that Bitwarden was mentioned 100% of the time, therefore it is very interesting to conduct a study focusing especially on the llms.txt file.

Although we could not analyze the effect of these two important hypotheses, the benefits of selecting this specific industry (characterized by a controlled, limited, and well-defined environment from multiple perspectives, as previously explained) significantly outweigh this single limitation.

3. Summary statistics, model-free evidence

Feature Engineering & Diagnostics

Before running our model, we inspected the "granular" data. We found extremely high correlations (>0.90) between certain channels, a signal of **multicollinearity**. For example, brands popular on Reddit are almost invariably popular on LinkedIn. To avoid collinearity errors, we aggregated these into broader "Signal" categories, where all variables were converted to z-scores so that their effect sizes could be compared on the same scale.:

- Social Buzz (*z_social_buzz*): aggregation of Reddit, YouTube, and LinkedIn mentions.
- Articles (*z_articles*): aggregation of Listicles ("Top 10 lists") and blogs of the company on a certain topic.
- Review Index (*z_review_index*): interaction of Rating × Volume (since a 5-star rating with 1 review is noise).

	ln_red~t	ln_you~e	ln_lin~n	ln_lis~e	ln_dom~n	raw_ra~g	ln_rev~s
ln_reddit	**1.0000**						
ln_youtube	**0.2663***	**1.0000**					
ln_linkedin	**0.9100***	**0.3187***	**1.0000**				
ln_listicle	**0.4982***	**0.0173**	**0.5374***	**1.0000**			
ln_domain	**0.7106***	**0.1444***	**0.7941***	**0.3072***	**1.0000**		
raw_rating	**0.6690***	**0.3000***	**0.7254***	**0.1859***	**0.7101***	**1.0000**	
ln_reviews	**0.6303***	**0.3651***	**0.8021***	**0.2047***	**0.7496***	**0.9278***	**1.0000**

Table 1. Correlation matrix of key explanatory variables

4. Modeling & Estimation

We utilized a **Conditional Logistic Regression (Fixed Effects Logit)** grouped by query ID (source_j). This allows us to see which brand "wins" a specific query (e.g., "Best for Mac") while controlling for the difficulty of that query:

$$\text{logit}(\text{mention}_{ij}) = \beta_1 \text{z\_social\_buzz}_{ij} + \beta_2 \text{z\_articles}_{ij} + \beta_3 \text{z\_review\_index}_{ij} + \beta_4 \text{z\_seo}_j + \gamma_{\text{source}_j}$$

Driver	Odds Ratio	Significance	Interpretation
Social Buzz	**10.99**	*** (p<0.001)	The dominant driver. High buzz = massive visibility.
Articles/PR	**1.86**	*** (p<0.001)	Being in "Top 10" lists and having relevant blog posts for a certain topic nearly doubles mention odds.
SEO Score	**1.43**	** (p<0.05)	Significant, but weak compared to social signals.
Reviews	0.73	n.s. (p=0.125)	Not a differentiator in this top-tier market.
Gemini (Source)	**0.41**	*** (p<0.001)	Gemini is significantly "stingier" than ChatGPT.

Table 2. Regression output summary

	Delta-method dy/dx	std. err.	z	P>|z|	[95% conf.	interval]
z_social_buzz	.262656	.0128975	20.36	0.000	.2373774	.2879346
z_articles	.0677343	.0193188	3.51	0.000	.0298702	.1055984
z_review_index	-.0345357	.0219008	-1.58	0.115	-.0774606	.0083891
z_seo	.0393551	.0185369	2.12	0.034	.0030234	.0756868
source_id						
gemini	-.0962741	.0242422	-3.97	0.000	-.1437879	-.0487603

Table 3. Average Marginal Effects.

5. Results & Interpretation

Key Findings

- The coefficient for Gemini (0.41) is highly significant with negative AME. For the exact same query, Gemini is ~60% less likely to mention a brand than ChatGPT. Gemini acts as a stricter gatekeeper, compared to ChatGPT outputs, showing that it filters products that it wants to recommend. ChatGPT is more likely to recommend various password managers, whereas Gemini requires a product to be an indisputable leader in its market.
- The **Social Buzz** variable is the strongest predictor in the model, with an estimated odds ratio of about 11, and it appears to **drive most of the LLM's mentions**. This suggests that LLMs place greater weight on dynamic, community-generated sources (e.g., Reddit threads, YouTube discussions) than on more static content. In the password manager market, where trust is largely validated through user communities, the AI effectively reproduces this collective judgment. Moreover, the marginal effect of Social Buzz is sizable: increasing Social Buzz by one standard deviation is associated with a **26.3 percentage point increase** in the predicted probability of citation, holding other factors constant.
- The **Review Index is not statistically significant** in our model. This likely reflects the nature of this specific market: all seven brands already enjoy very high ratings (4.5+). Within such a

compressed range, small differences (e.g., 4.8 vs. 4.6) do not meaningfully influence the LLM's citation behavior. In other words, strong reviews are a baseline requirement to be considered at all, but once that threshold is met, **review scores no longer help a brand stand out** or secure more mentions relative to competitors.

- The model achieved a **Pseudo R^2 of 0.57**, far exceeding the standard 'excellent' threshold of 0.20–0.40. This high goodness-of-fit confirms that LLMs' recommendations are not random, but are highly predictable based on the Social Buzz and Content Authority signals we identified.

6. Validation & Robustness

To evaluate the reliability of our findings, we complement the baseline conditional logit specification with a set of internal consistency checks. First, we control for unobserved query-level heterogeneity by estimating a conditional logit with query-specific groups and clustering standard error. This ensures that our inferences are not driven by differences in baseline mention rates across individual prompts and that within-query correlations in the error term do not inflate statistical significance. Additionally, we include a source dummy (ChatGPT vs Gemini) to capture systematic differences in how many brands each model tends to mention, so that platform-level behaviour is not mistakenly attributed to brand-level characteristics.

Furthermore, before deciding on the pooled specification, we compared the behaviour of the two AI models. We estimated separate models by source and conducted a likelihood-ratio test on the equality of coefficients, which did not reject the hypothesis that the main drivers operate similarly across ChatGPT and Gemini. This result justifies the use of a pooled model with a source indicator: it allows us to retain a single, optimal set of coefficients for the marketing variables while still accounting for platform-specific differences in baseline brand-mention propensity.

Finally, we cross-check the regression results against model-free evidence. The box plots of standardized Social Buzz, Articles/PR, Review Index and SEO by mention status and source, show

visually that mentioned brands systematically score higher on community and editorial signals than non-mentioned brands on both platforms. The alignment between these descriptive patterns and the regression coefficients strengthens the interpretation that our main effects reflect genuine relationships in the data rather than features of a particular specification.

7. Managerial Implications

Our findings suggest that community signals are the main drivers of visibility in generative AI search for password managers. Social Buzz (the combined signal from Reddit, YouTube and LinkedIn) proves to be the strongest influence when it comes to being mentioned by ChatGPT or Gemini (odds ratio ≈ 11; +26.3 p.p. in mention probability for a one-standard-deviation increase). In comparison, the SEO score of the brand's homepage, while statistically significant, has a much smaller marginal effect (odds ratio ≈ 1.43; ≈ 3.9 p.p. per standard deviation). From a managerial perspective, this points towards a rebalancing of budgets: once basic technical SEO setup is ensured, incremental gains in generative AI visibility are more likely to come from fostering genuine discussion and advocacy in digital communities (e.g. product deep-dives on YouTube, technical threads on Reddit, thought-leadership on LinkedIn).

Structured third-party coverage remains relevant, but primarily as a supporting rather than leading variable. Our aggregated Articles/PR index almost doubles the odds of being cited by an LLM (odds ratio ≈ 1.86), suggesting that "Best X password manager" rankings and authoritative topical articles help LLMs treat a brand as a default option within a given use case. In practical terms, GEO programmes should align PR, content and community activities. Securing inclusion in reputable rankings and maintaining a topical blog presence are important, but their effect is maximised when combined with active social discussion that keeps the brand "alive" in the training and retrieval ecosystem.

Moreover, the model indicates that online review metrics are not a key differentiator within this set. Our Review Index, combining G2 rating and volume, is statistically insignificant in predicting LLM mentions, which is consistent with the fact that all seven brands have high ratings (≥ 4.5/5).

Reviews therefore provide a minimum quality and trust threshold to enter the consideration set, but do not act as a driver to secure additional mentions once that threshold is met. Managers should continue to monitor and maintain review quality, yet marginal investments aimed at moving from a rating of, for example, 4.6 to 4.8 are unlikely to materially affect generative AI visibility when compared with investments in social and content signals.

Finally, platform strategy appears to matter. Gemini's coefficient (odds ratio ≈ 0.41 relative to ChatGPT) shows that it is significantly more selective, mentioning fewer brands for the same query. In practice, ChatGPT behaves broadly by listing multiple acceptable options, while Gemini resembles a curated shortlist. For challenger brands such as NordPass or RoboForm, a staged strategy is recommended: first prioritise winning share of voice on ChatGPT (through Social Buzz and Articles) and then treat Gemini as a second-stage objective that requires category-leading levels of both community discussion and editorial coverage before the brand consistently appears in its answers.

8. Limitations & Future Work

This analysis has three main limitations. Initially, our social activity measures are proxies based on Google search operators (e.g. site:reddit.com), which capture the number of indexed pages mentioning a brand but not the sentiment, depth or recency of those conversations. Second, to avoid multicollinearity we aggregated Reddit, YouTube and LinkedIn into a single Social Buzz index, which improves model stability but conceals channel-specific effects. We can state that "social" matters, but not whether Reddit is more important than LinkedIn or YouTube in driving LLM mentions within this category. Third, there is a Bitwarden-specific anomaly: it is both open source and the only brand in our sample with an llms.txt file, and it is mentioned 100% of the time. Because this llms.txt dummy perfectly overlaps with the brand, our model cannot disentangle whether Bitwarden's outperformance is driven primarily by community popularity, technical readiness, or an interaction of both.

Future work could address these limitations along three directions. First, researchers could replace

search-operator proxies with native platform data and sentiment analysis (e.g., via Reddit and YouTube APIs), allowing visibility, recency, and tone of conversation to be modelled separately. Second, instead of aggregating channels, a larger cross-category sample could support channel-level models, clarifying which specific communities (developer forums vs consumer review platforms vs professional networks) are most predictive of LLM mentions across different industries. Lastly, the role of technical signals such as llms.txt would benefit from a dedicated experimental design, for example, by tracking a broader set of sites before and after deploying these types of files, in order to isolate whether these interventions shift GEO outcomes once community and PR signals are held constant.

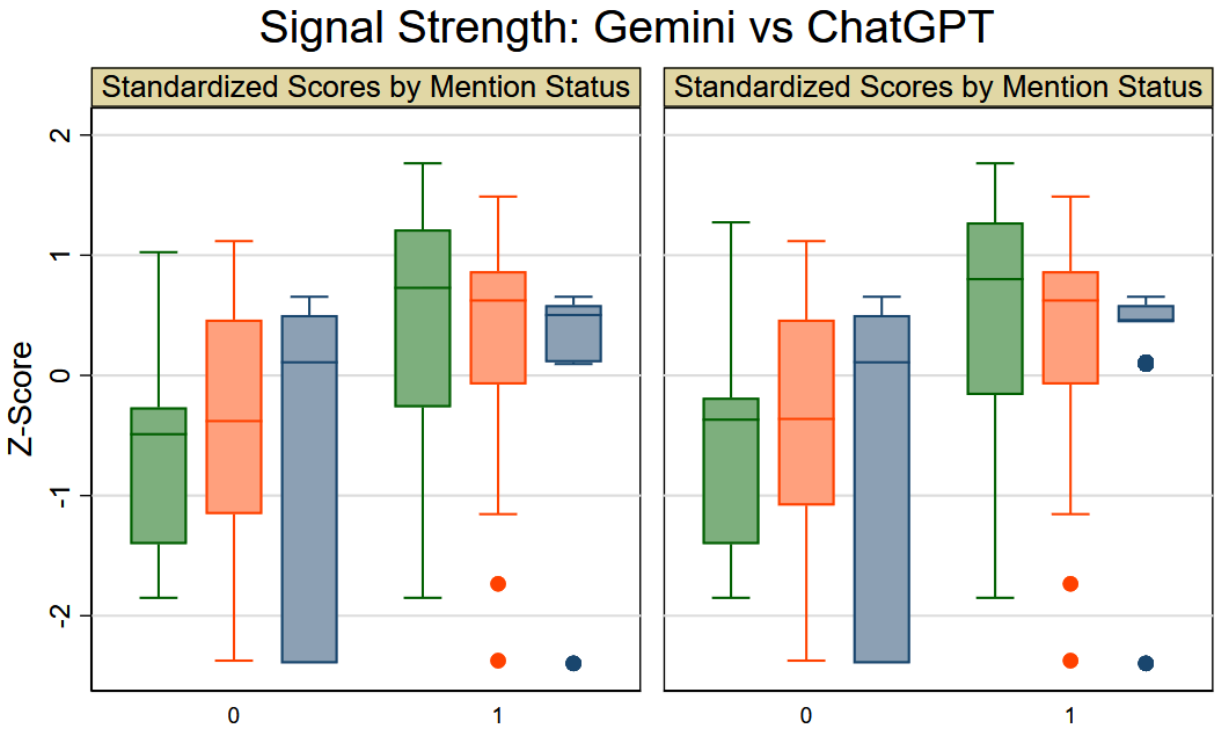

Figure 1. Standardized signal scores by mention status: Gemini vs ChatGPT

GREEN: Standardized values of ln_social_buzz
RED: Standardized values of ln_article
BLUE: Standardized values of review_index

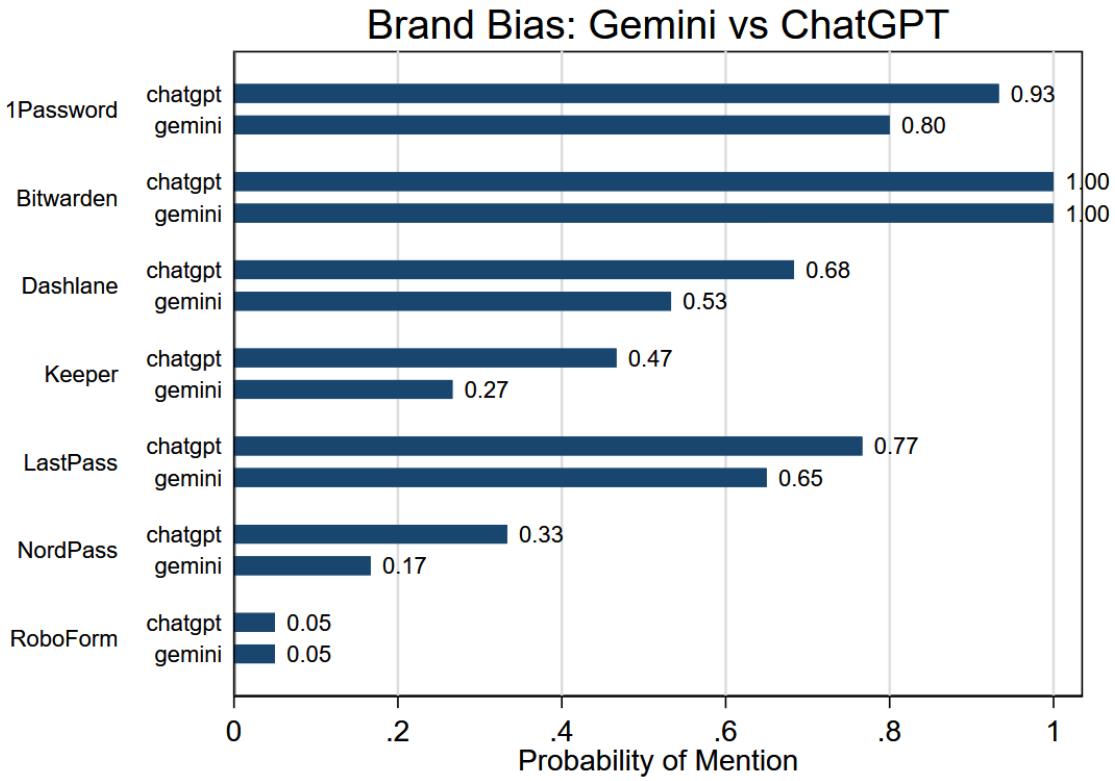

Figure 2. Brand-level mention rates by model: Gemini vs ChatGPT

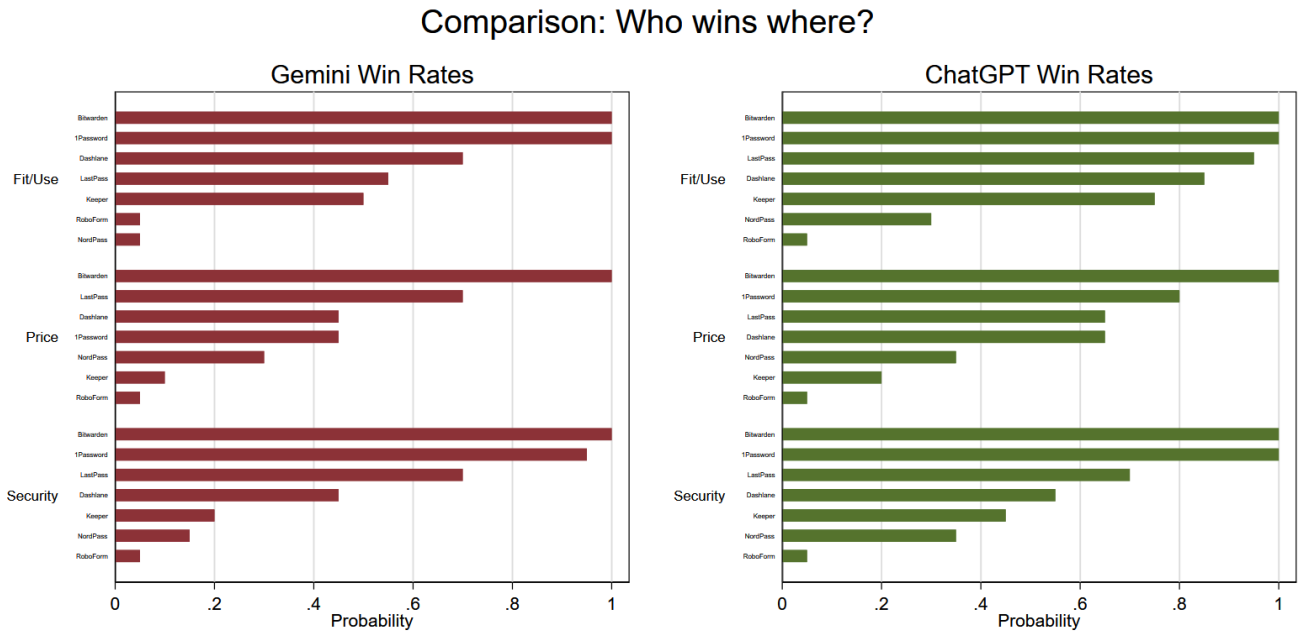

Figure 3. Topic-Specific brand mention probabilities: Gemini vs ChatGPT