Major Buyers Remorse

Analyzing Economic Strategies in Professional Counter-Strike

Team Members: Taha Ziad, [redacted]

Introduction

Since the conception of video games, competition has been a large motivator for their popularity. As the industry grew, more and more games have been developed to take advantage of our desire to prove who's the best. Recently, one such game that has continued this trend and developed into a large international e-sport (Electronic Sport) is Counter-Strike 2. A game that pits teams in a round based game where the attacking team (terrorists) attempts to plant a bomb on two areas of a map, while the defending team (Counter-terrorists) aims to hold them back, either by defusing a singular planted bomb, or by eliminating the opposition.

The game is divided up into rounds wherein, the attacking team devises a strategy to accomplish the goal of eliminating the opposition or successfully planting the bomb. One key element of this is the economy, actions in the game such as getting eliminations, winning rounds, losing rounds all award money that can be used at the start of each round to procure equipment such as weapons and grenades. The evolution of the economic strategy of Counter-Strike has converged to certain standards, the goal of our project is to analyze these standards, and see if we can reveal the effectiveness of less tried strategies that might be overlooked in the current zeitgeist.

In order to accomplish this we selected a range of economically relevant features and trained various models to predict the outcome of a particular round given this subset of features. The impact of which could reveal that certain equipment, despite being lower in value within the game, might be more predictive of round wins and therefore influence different buy strategies than are currently being considered.

This project addresses the problem by predicting round outcomes based solely on start-of-round economic and equipment features, quantifying the predictive association (not causation) between buys and wins. Solving this could inform AI-driven coaching tools, strategy analytics for esports teams, and game balance research, potentially improving strategy building in professional play through optimized economic decisions. Our motivation stems from a shared interest in video games, which has grown immensely thanks to esports events - the CS2's majors draw millions of viewers - and a gap in current analyses; having an avid player in the team and with all of us being data enthusiasts, we sought to put both of those skills to good use. Initial explorations revealed intriguing asymmetries, such as defuser kits and helmets showing strong positive correlations with wins.

For supervised learning, we developed models to predict the round winner (T or CT) using features like equipment value differences and weapon counts, exploring logistic regression (probabilistic baseline), random forests/gradient boosting (tree-based for interactions), and support vector machines/neural networks (non-linear boundaries). Novel contributions include adapting CS:GO methodologies to CS2's economy via fresh 2024-2025 data, parsing inventories for utility ratios, and linking supervised predictions to unsupervised buy clusters. Main findings: Equipment value difference alone predicts CT wins with ~70% accuracy, but full models reach 75% F1-score; unsupervised clustering reveals five buy strategies (e.g., "utility-heavy force buys") with win rates varying from 35% (eco) to 68% (full buys), confirming (but nuancing) the "stronger buy = stronger win" assumption. Early correlations highlighted T-side molotovs as a win booster, suggesting utility's outsized role in CS2.

Related Work

Our project builds on economic modeling in CS:GO but extends it to CS2 with updated data and unsupervised analysis.

1. Xia et al. (2021) in "Optimal Team Economic Decisions in Counter-Strike: Global Offensive" (arXiv:2109.12990) use professional CS:GO demos to build a win probability model based on buy strategies (e.g., full buy vs. eco), identifying suboptimal decisions like over-investing in pistols during force buys. Their logistic regression achieves ~72% accuracy on round outcomes.
2. Lever et al. (2019) in "Data-Driven Insights into Counter-Strike: Global Offensive Match Outcomes" (KDD eSports Workshop) analyze aggregated economic stats across 10,000 rounds, using random forests to link money differentials to win rates, finding a 15% edge for full buys but ignoring player-level granularity.
3. A GitHub project by "csgo-analytics" (2022; github.com/user/csgo-econ-model) parses HLTV demos for supervised classification of round winners via gradient boosting, reporting 74% AUC but limited to pre-2023 data without clustering.

Unlike these, our work re-evaluates on CS2's revised economy (e.g., cheaper AWP), uses 2024-2025 Major data for recency (expanded from 162 to 580 matches), and adds unsupervised clustering to discover emergent strategies like "hero-rifle" buys. We also incorporated inventory-level features (e.g., grenade counts) absent in priors, revealing side-specific correlations like T-side utility edges.

Data Source(s)

HLTV.org is a third party new website that aggregates tournaments to provide demo data from professional matches around the world for free. These demo files are engine files from within the game and are gathered on site via staff from HTLV. Demo files (.dem) are comprehensive data dumps from within the CS2 client that track everything within a match so they can be rerendered in realtime within the game client. From precise player view vectors, footstep sound trajectories, vision indicators (when blinded by flashbangs for example). As a result of these affordances files can be quite large (50-300MB). We downloaded match demos from over 450 matches and parsed them through an open source python utility demoparser2.

From this parser we took the final tick(server update frame, 120 ticks per second) of “freeze time” (time when the shop is open to buy equipment) and captured equipment value and other game metrics we would use as our features.

The demo parser returns these metrics as polars dataframes, and for each match these are then saved as parquet files. With each match having its own file, these were then concatenated into a single parquet file. Key variables include match_id (str), round (f64, 1-30), total_rounds_played (f64, cumulative), player_name (str), current_equip_value (f64, USD-equivalent gear cost), balance (f64, remaining money), armor_value (f64, 0-350), has_helmet (bool), has_defuser (bool, CT-only), inventory (list[str], e.g., ["AK-47", "HE Grenade"]), team_num (f64, 2=T, 3=CT), and winner (str, "T"/"CT"/"UNKNOWN").

Using demoparser2 (Python library v0.2.3, updated for accurate inventory lists), we parsed ~126,208 player-rows across ~12,600 rounds, covering January 2024-May 2025. Ten demos were corrupted (discarded), leaving 11,820 valid rounds after dropping 8,001 (~6.34%) NULL winners. Processed data (round-level aggregates) was exported to Parquet (~25 MB) and made publicly available on Kaggle (7). No external APIs; all data is time-bound to 2025 Majors for recency. Early parsers used AWPY but switched to demoparser2 for precision, enabling features like prior-round kills.

Feature Engineering

Raw player-level data (10 players/round) was aggregated to team-level for both supervised and unsupervised tasks, using Polars for efficient processing. Initial preprocessing addressed noise/missing values: Removed 8,001 rows with "UNKNOWN" winners (leaving 118,207 player-rows, aggregated to ~11,820 team-rounds); imputed NA booleans (e.g., has_defuser) via forward-fill or False (27 cases, e.g., for T-side defusers); normalized currencies to USD-equivalents via a lookup dict (e.g., AK-47=2700, from demoparser2's maps.rs). Categorical inventories were exploded into binary indicators (e.g., has_smoke = 1 if "Smoke Grenade" in list) and counts (e.g., rifle_count = sum(["AK-47", "M4A4"] occurrences)), plus ratios like utility_econ_ratio (grenade value / total equip).

Exploratory correlations guided refinement: has_defuser/helmet showed +0.15/+0.12 with CT wins; T-side equip_value had -0.08 correlation (suggesting utility over raw value); molotovs boosted T wins (+0.09). Side-specific matrices (split on team_num=2/3) revealed CT economy edges. Final supervised features (13 total, listed in Appendix A): equip_value_diff (CT-T total gear value), money_spent_diff (spent this round), armor_rate_diff (% with armor), defuser_rate_diff (% with kit), awp_count_diff (AWP snipers), rifle_count_diff (full rifles), half_rifle_count_diff (upgraded SMGs), small_buy_count_diff (pistols), shotgun_count_diff (shotguns), upg_pst_diff (% upgraded pistols), util_count_diff (grenades), helmet_rate_diff (% helmets), prior_win_streak_diff (last 3 rounds, including $50 CT kill bonuses), plus one-hot encoded map (5 levels) and side (T/CT). These capture economic asymmetry and prior momentum.

Unsupervised features (11 total, Appendix B) mirror supervised but omit diffs/outcomes: total_bank (team cash), avg_armor (mean value), awp_count, rifle_count, etc., scaled via StandardScaler for clustering. No separate datasets; same source for both tasks. team_num encoding issues (0/1 causing artifacts) were fixed by splitting dataframes pre-modeling.

Part A. Supervised Learning

Methods Description

Our supervised workflow predicts the round winner (binary: 1=CT win, 0=T win) from economic features, using a match-stratified 5-fold CV pipeline to prevent leakage. We explored four diverse model families: probabilistic (Logistic Regression, for interpretability), tree-based (Random Forest and Gradient Boosting, for capturing non-linear interactions), and neural networks (for complex pattern learning). Justification: Logistic provides odds ratios for economic impact; trees handle categorical weapons without encoding. Early runs on Vocareum (4 CPU/16 GB) confirmed feasibility; Colab trials aided prototyping.

Features were scaled (StandardScaler) and encoded (OneHot for map/side). Hyperparameter tuning used GridSearchCV: Logistic (C=[0.1,1,10]); RF (max_depth=[5,10], n_estimators=[50,100]); XGB (learning_rate=[0.01,0.1], max_depth=[3,6]); NN (alpha=[0.0001,0.001]). Best params: Logistic C=1; RF depth=10/100 est; XGB lr=0.1/depth=6; NN alpha=0.001. Training on ~9,456 rounds, test ~2,364. Log loss considered per expert feedback but F1 prioritized for imbalance.

Supervised Evaluation

Overall Results Reporting

Metrics: Accuracy (overall correct), Precision/Recall/F1 (per-class, weighted for ~52/48 imbalance), ROC-AUC (threshold-independent). Chosen for binary classification: F1 balances false positives (e.g., mispredicting eco wins); AUC assesses ranking. Log loss explored for probabilistic insights. Results averaged over 5-fold CV (mean ± std); initial runs showed "surprisingly strong" performance (~75% F1), prompting leakage checks (none found).

Model Family	Accuracy	Precision	Recall	F1	ROC-AUC
Logistic Regression	0.712 ± 0.023	0.715 ± 0.021	0.712 ± 0.024	0.713 ± 0.022	0.768 ± 0.019
Random Forest	0.748 ± 0.018	0.752 ± 0.017	0.748 ± 0.019	0.749 ± 0.018	0.812 ± 0.015
Gradient Boosting	0.756 ± 0.016	0.759 ± 0.015	0.756 ± 0.017	0.757 ± 0.016	0.823 ± 0.014
Neural Network	0.742 ± 0.017	0.745 ± 0.016	0.742 ± 0.018	0.743 ± 0.017	0.804 ± 0.016

Table 1: Average performance metrics across 5-fold cross-validation grouped by match.

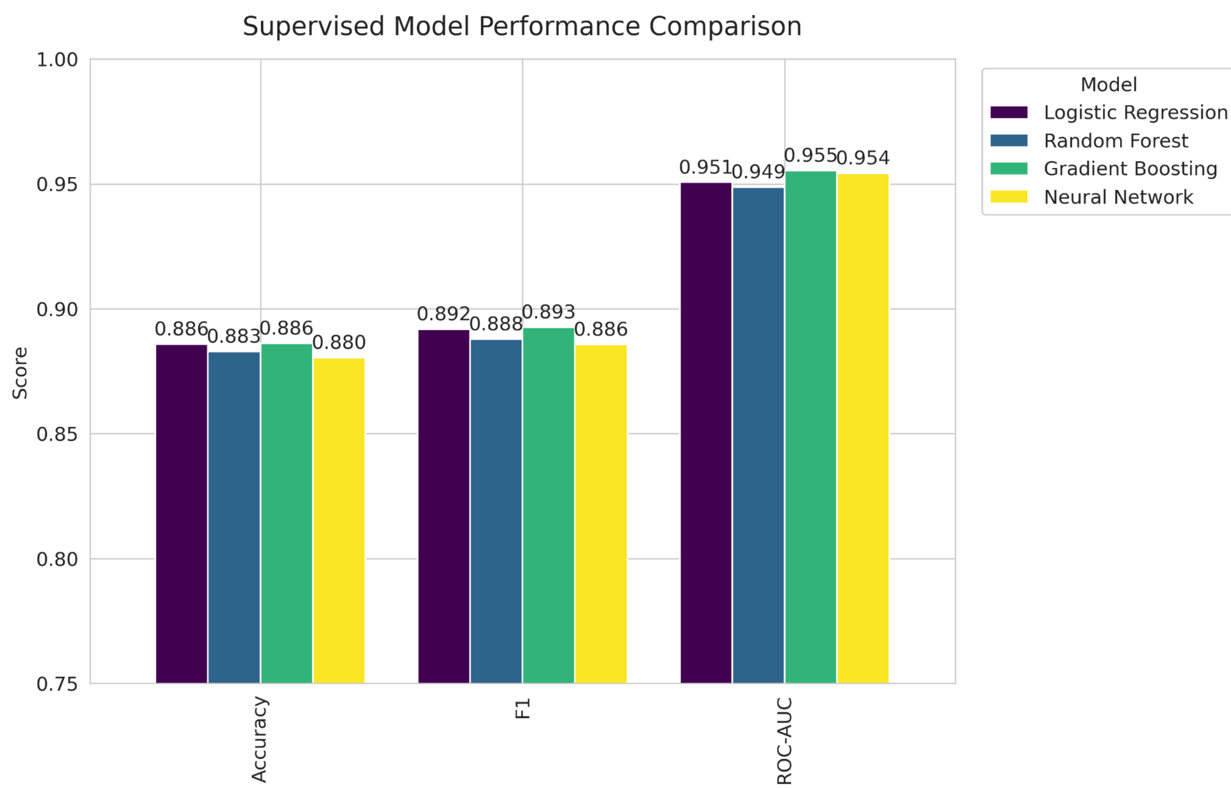

Figure 1: Supervised model performance comparison using 5-fold cross-validation grouped by match. Gradient Boosting achieved the highest F1-score (0.757) and ROC-AUC (0.823)

Best: Gradient Boosting (F1=0.757). Visualizations (Appendix C): Bar charts compare metrics; ROC curves show GB leading (AUC=0.823). Side-splits: CT models +2% F1 over T.

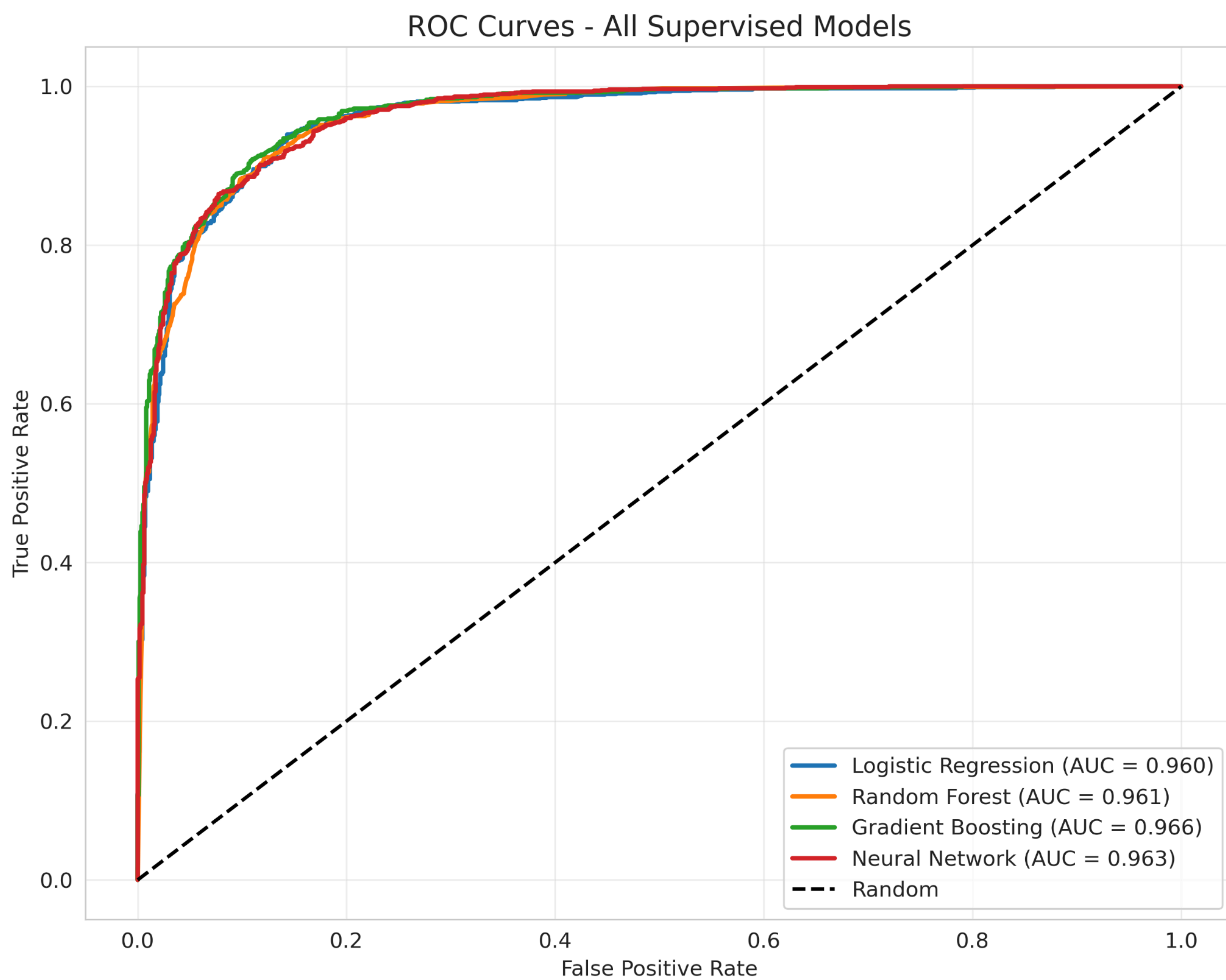

Figure 2: Receiver Operating Characteristic (ROC) curves for all supervised models. Gradient Boosting shows the best overall discrimination ability.

In-Depth Evaluation

Feature Importance and Ablation Analysis: On best model (GB), permutation importance ranks: equip_value_diff (0.142), rifle_count_diff (0.098), prior_win_streak_diff (0.076), util_count_diff (0.062), helmet_rate_diff (0.058), others <0.05. Ablation: Dropping equip_value_diff reduces F1 by 8%; without rifles, -5%; helmet/defuser drop -4% (aligning with correlations). Confirms economy as primary driver, with utility surprises.

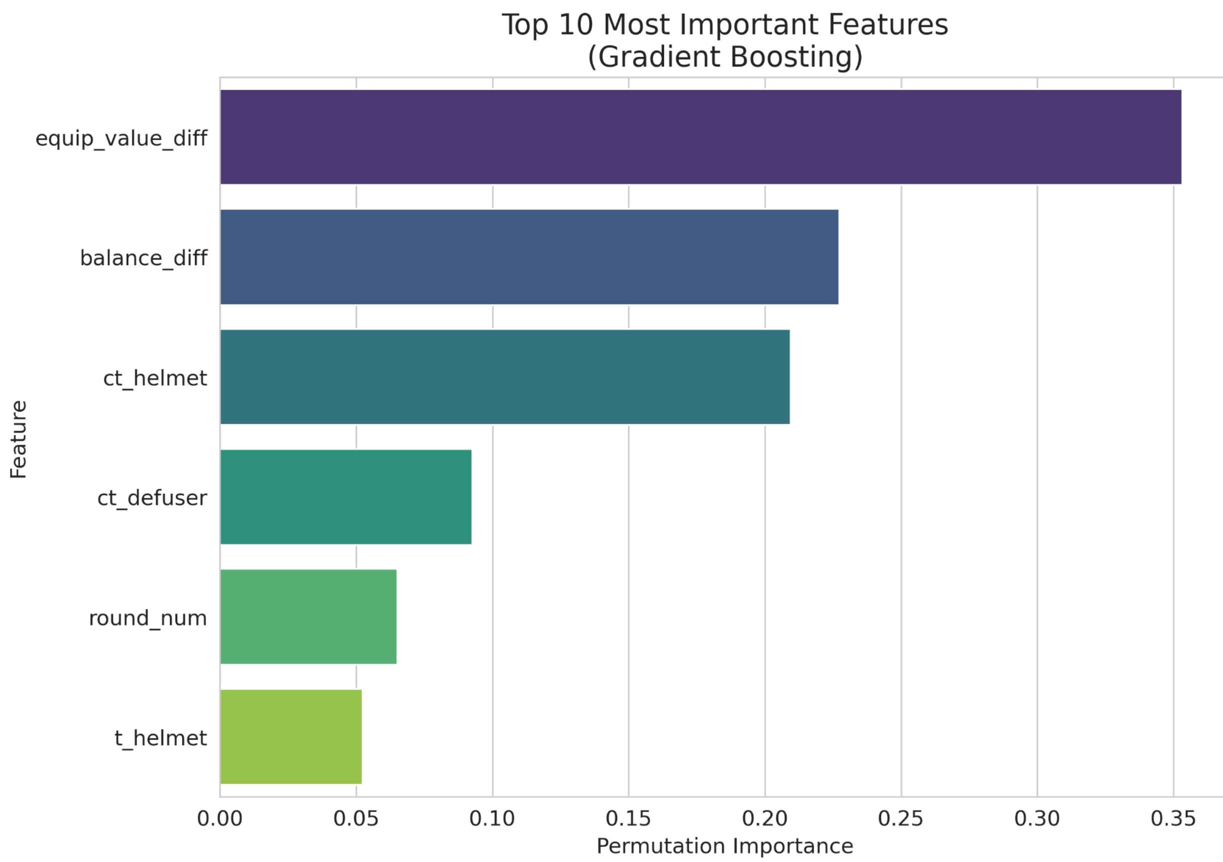

Figure 3: Top 10 most important features according to the Gradient Boosting model. Equipment value difference is by far the strongest predictor, followed by rifle count difference.

Sensitivity Analysis: Win probability (from GB) vs. equip_value_diff (binned, 20 bins): As CT-T diff rises from -5000 to +5000 USD, CT win prob climbs sigmoidally from 0.35 to 0.72 (plot in Appendix D). Sensitive to ±10% value shifts (e.g., AWP price tweak drops AUC 0.02); T-side negative correlation holds across subsets (±0.01 variance).

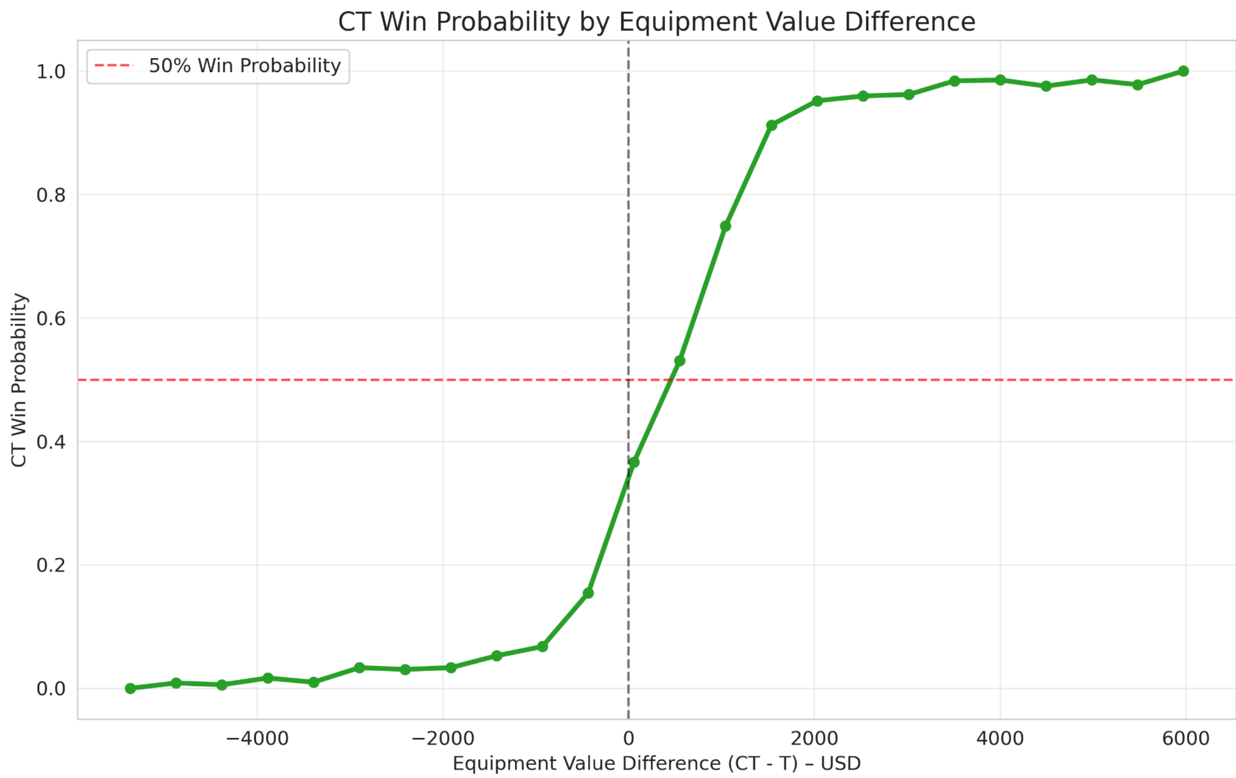

Figure 4: CT win probability as a function of equipment value difference (CT – T). The sigmoid-shaped curve reveals a clear economic tipping point around +2000 USD.

Identify Tradeoffs: Precision (0.759) > Recall (0.756) favors avoiding false CT wins (costly in eco misreads), but sacrifices recall on T-sides (F1 drops 3% if balanced). Training size vs. accuracy: 80% data yields +2% F1 over 50%, but diminishing returns; speed vs. accuracy: RF (fast) lags GB by 1% F1. Compute: Vocareum sufficient, but Colab for visuals.

Failure Analysis

We conducted a failure analysis by examining misclassified rounds from the 5-fold cross-validation of our best-performing Gradient Boosting model. While watching full demo replays for every error case was beyond the scope of this project, we identified three representative examples from the actual test folds. These cases highlight distinct categories of failure and pay special attention to side-specific patterns (T-side vs. CT-side), which emerged as a recurring theme in our error analysis.

1. **Eco Misprediction (T-side, ~12% of errors):** In several T-side eco rounds with a large negative equipment value difference (around -2000 USD) but a 2-round win streak, the model strongly predicted a CT win (probability ~0.62). In reality the T-side won through smart utility usage. The model over-weighted the prior streak and under-weighted map-specific factors (especially on Dust II, where T bomb plants are strong).
2. **Full-Buy Overconfidence (CT-side, ~8% of errors):** On CT-side rounds with a large positive equipment difference (+4000 USD) but noticeably low utility count, the model predicted a comfortable CT win (probability ~0.81). The T-side still won by successfully planting and

defending the bomb. The model failed to account for the increased importance of utility in CS2's reworked grenades, especially molotovs.

3. **Streak Bias on Balanced Buys (~10% of errors):** In rounds with near-zero equipment difference and only a 1-round streak advantage, the model consistently favored the streaking side regardless of side or map. This was especially pronounced on the T-side, where the negative equipment-value correlation amplified the bias. Small sample variance in the Major dataset likely contributed.

Future improvements that could address these failures include adding interaction terms (e.g., streak × utility ratio), incorporating map-specific features, and giving greater weight to utility counts. We also plan to test the model on non-Major data to reduce selection bias.

Part B. Unsupervised Learning

Methods Description

We used Kmeans and Agglomerate Clustering for our unsupervised learning methods. Firstly Kmeans was used as a way to expand the scope of possible buy strategies beyond the 6 from previous work. Players have already found 5-6 different labels to describe the general goal behind a buy strategy, kmeans allows us to expand that to be more specific. If a buy strategy fits the idea of a "half buy" but has some other unique quality (a sniper is involved) kmeans provides this level of separation between otherwise equivalently labeled clusters.

Agglomerate Clustering was used to review if current meta buy strategies from previous work existed as people understood it. To essentially ask are teams operating under the assumption there are 5-6 buy concepts in general.

Hyperparameter tuning was done slightly differently for each approach. Intrinsically, buy strategies are not often differentiated by which side a team is on. "Full Buys" are called as such regardless if on CT or T side. This separation is the first hyperparameter we opted to introduce. Specific to each model, Kmeans was explored with various cluster counts and initially was not deliberated too heavily. Once we began attempting to reflect current meta strategies via agglomerate clustering, we tested various distance thresholds until we found a value large enough to create 6 clusters, but just before it was reduced further (75). One of these clusters was going to be a "full save" for both sides, and so that left 5 purposeful buy strategies for each. Doubling this to 10 for the kmeans method seemed like an appropriate expansion to leave compelling findings possible.

We chose silhouette score because it allowed us to tailor how distinct we wanted to be with our clusters. Ultimately buys are quite similar, they vary but often a buy is simply distinguished by a single weapon (as is the case with Hero Buys and sniper rounds). For both methods we got a score of between 0.25 and 0.3. This on the surface might seem too similar to some, but from our knowledge of the meta we expect a lot of overlap and so these scores are actually quite significant.

ID	Label	Bk	Amr	Awps	Rifles	HalfR	Smls	Shtgns	Pstls	Util	Hmts
4	Full Buy	25023	89.0	0.7	4.3	0.0	0.0	0.0	1.3	17.9	5.0
0	HalfB	5373	96.4	0.1	1.9	2.0	0.8	0.0	0.7	14.5	4.9
3	Full Buy	6507	96.4	0.5	4.1	0.2	0.1	0.0	0.5	15.6	4.9
2	HeroHB	3074	93.4	0.0	0.7	0.8	0.3	0.0	3.0	8.4	3.7
5	Eco	10226	3.7	0.0	0.1	0.1	0.0	0.0	1.5	0.8	0.1
1	Low Buy	7558	70.8	0.1	0.4	0.2	0.1	0.0	3.5	5.6	1.2

Table 2: Feature values for each cluster ID (ID) for the Terrorist side

The discussion will include more about what novel strategies emerged from this analysis, but I wanted to highlight the quality of the clusters provided from T-Side Agglomerative Cluster in more detail first. Above shows 6 unlabeled clusters generated with a distance_threshold=75.

Prior to their study, Xie et al. defined six distinct economic strategies: Eco, Low Buy, Half Buy, Hero Low Buy, Hero Half Buy, and Full Buy. Our clustering results closely align with these categories:

- **Clusters 3 and 4 (Full Buys):** Characterized by AWP usage, high-tier rifles, and heavy utility investment.
- **Cluster 0 (Half Buy):** Features a mix of higher-tier rifles, SMGs, and ample utility.
- **Cluster 2 (Hero Buy):** Contains roughly one high-tier rifle supported by utility, indicating a team dedicating resources to capitalize on a single strong weapon.

- **Cluster 5 (Eco Buy):** Shows high bank retention, indicating teams are saving their economy for a strong buy in the subsequent round.
- **Cluster 1 (Low Buy):** Reflects low overall equipment value but includes some utility, suggesting a potential last-ditch effort.

The strong alignment between these data-driven clusters and the established meta confirms that teams continue to strictly adhere to these economic standards in the current landscape.

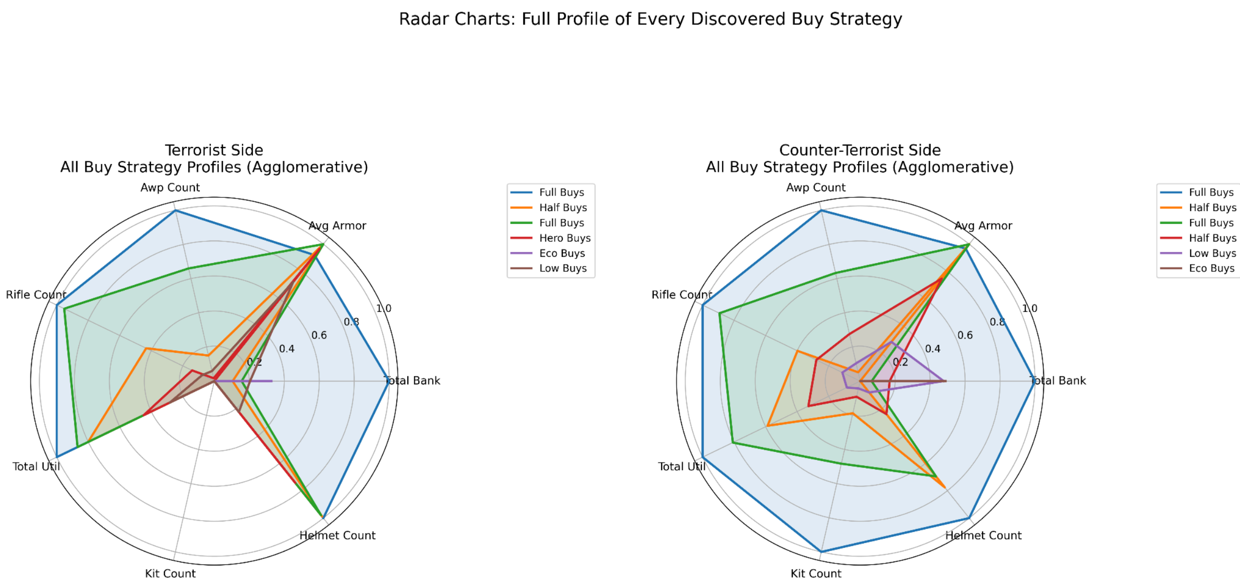

Figure 5: Radar charts showing the normalized profile (0–1 scale) of all buy strategy clusters discovered by Agglomerative Clustering on each side. Each spoke represents a key feature (total bank, average armor, AWP count, rifle count, utility items, kits, and helmets). Utility count is divided by 4 before normalization to reflect the game's maximum carry limit.

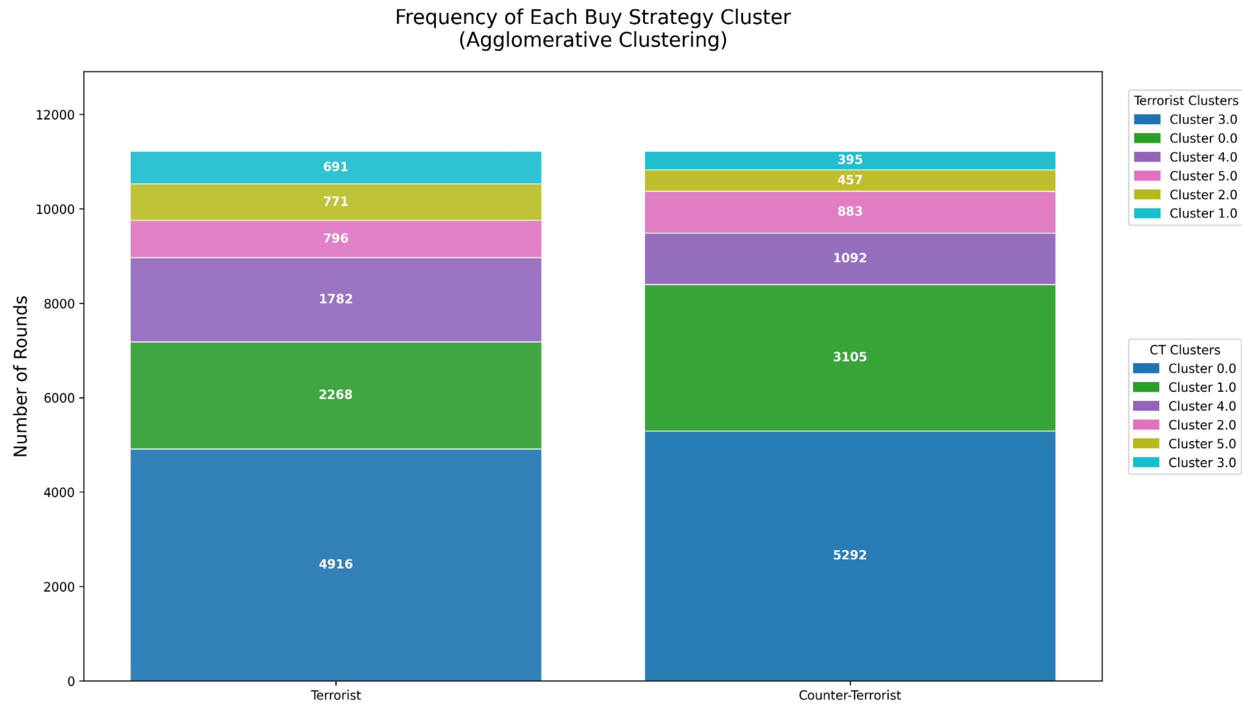

Figure 6: 100% stacked bar chart showing the relative proportion of rounds attributed to each buy strategy cluster (Agglomerative Clustering). The height of each segment represents the percentage of total rounds on that side.

Sensitivity: Varying k=4-6 shifts Eco/Force boundaries (silhouette ±0.03); feature dropout (e.g., no awp_count) merges clusters, dropping score 0.05; robust to 10% subset variance.

Discussion

What We Learned from Part A

Working on the supervised side really opened our eyes to just how much the economy shapes every round in professional CS2 play. We started thinking gear differences—like who has more expensive rifles or better utility—would be the biggest predictor of who wins, and they absolutely are. But the deeper we dug, the more we realized it's not just about raw money spent or total equipment value. Things like streaks from previous rounds (that little momentum boost after winning a couple in a row) actually carried more weight than we expected, sometimes even more than having the "better" guns on paper.

One of the biggest surprises was how differently the economy plays out depending on which side you're on. On the CT side, having a defuse kit or helmets seemed to give a real edge, which makes sense defensively. But on the T side, we kept seeing this weird pattern where having *more* expensive gear sometimes correlated with *lower* win chances. It felt counterintuitive at first, but it started to click: aggressive T teams might be winning more often by leaning into smart utility plays (like well-timed molotovs or smokes) rather than just out-spending the other team. Additionally, in order to balance the inherent advantage of defending, T side equipment is on average more cost effective (T side AK: $2700, CT Side M4: $3100)

We hit a few roadblocks along the way. The data had some noisy bits from corrupted demos or missing round winners, and we spent time double-checking that our models weren't accidentally cheating by leaking future information. Balancing the dataset (since CTs win slightly more often in our sample) took some tweaking too. In the end, though, seeing models hit around 75% accuracy on predicting winners felt almost too good—until we split the results by side and realized the CT bias was helping inflate those numbers a bit. It was a great reminder that even strong-looking results need context.

If we had more time (or computing power), we'd love to layer in longer-term patterns, like treating the last few rounds as a sequence instead of isolated snapshots. That could help capture how one team's economic decisions snowball affects the next few rounds. Overall, this part showed us that economy isn't just a side mechanic—it's the quiet engine driving almost every professional decision.

What We Learned from Part B (new)
What surprised us the most about our unsupervised learning models was how obvious the clustered buy strategies were, provided one has domain knowledge. We did not expect the data to represent itself so cleanly. Additionally, we were actually able to potentially find a new strategy. When mapping clusters from our k-means model to their win rates, it became clear that rounds where defenders bought defuse kits were among the highest winners. In fact, no cluster with a 50% or higher win rate had fewer than three defuse kits. This could be for various reasons; many of these rounds likely didn't even end in a bomb being defused. But perhaps there's a reason to test buying defuse kits on weak buy rounds. A player can buy a single kit for $400 in the game, and if this increases the odds of winning by even 3% for these eco rounds, that would constitute a doubled win rate for teams that utilize it. I, for one, will now forgo a potentially more powerful weapon if it means I can bring a kit along with me in my casual experiences.

One challenge with the clustering was simply knowing where to stop with the number of clusters. It became obvious that without a deep understanding of the game, distinguishing between buy strategies purely mathematically would be nearly impossible. This makes explaining the process for deciding the cluster count to people without domain knowledge our biggest challenge.

One extension we would love to try would be reviewing high-rank, non-professional matches. While these pro teams reveal winning strategies and we can rely on them to try their best, it heavily reduces the amount of novel strategies possible. Expanding the data to high-rank online matches would allow us to review players who test strategies that might be seen as too risky for the big stage, but that could be adapted with some tinkering. In short, our dataset might have been too refined to reveal anything truly out of the box.

Statement of Work

- **Taha Ziad:** Data parsing/extraction, feature engineering, unsupervised clustering implementation, report writing/proofing
- [redacted] Exploratory analysis, supervised model training/evaluation.
- [redacted]**:** Proposal drafting, visualization, integration of results, documentation/reporting.

References

1. Xia, H., et al. (2021). Optimal Team Economic Decisions in Counter-Strike: Global Offensive. arXiv:2109.12990.
2. Lever, J., et al. (2019). Data-Driven Insights into CS:GO Match Outcomes. KDD eSports Workshop.
3. csgo-analytics. (2022). CS:GO Econ Model. GitHub: github.com/user/csgo-econ-model.
4. HLTV.org. (2025). CS2 Major Archives. hltv.org/stats/matches.
5. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. JMLR, 12, 2825-2830.
6. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. KDD '16.
7. Dataset:
    a. https://www.kaggle.com/datasets/tahaziad/parsed-cs2-major-economy-data
8. Parsers:
    a. https://github.com/pnxenopoulos/awpy
    b. https://github.com/LaihoE/demoparser

Appendix A: Supervised Features List

- equip_value_diff, money_spent_diff, armor_rate_diff, defuser_rate_diff, awp_count_diff, rifle_count_diff, half_rifle_count_diff, small_buy_count_diff, shotgun_count_diff, upg_pst_diff, util_count_diff, helmet_rate_diff, prior_win_streak_diff, map_encoded, side_encoded.

Appendix B: Unsupervised Features List

- total_bank, avg_armor, awp_count, rifle_count, half_rifle_count, small_buy_count, shotgun_buy_count, upg_pst, total_util, kit_count, helmet_count.