Problem Chosen
C

2026
MCM/ICM
Summary Sheet

Team Control Number
2601356

Who Leads the Dance? Harmonizing Score and Vote with Wide & Deep Models

Summary

Dancing with the Stars has faced controversy over voting mechanisms that may favor popularity over merit. This paper proposes a framework to estimate latent audience votes, evaluate aggregation rules, and design a system balancing professional judgment and engagement.

First, to model elimination risk under limited and indirect information, we construct a **Wide & Deep neural network** that captures both memorization effects and nonlinear feature interactions, distinguishing it from conventional linear or single-stream models. The model integrates numerical performance indicators, categorical celebrity attributes, and competition context variables, while introducing three **temporal and competitive features—SurvivalWeeks, ScoreDiff, and Bottom3**—to capture long-term popularity accumulation, short-term performance momentum, and within-week relative risk exposure.

Model robustness is evaluated through **multiple random cross-season train–test splits**. The proposed model achieves a mean accuracy of **96.1%** with a low standard deviation (**6.6%**), demonstrating strong generalization. While a gradient boosting baseline (**LightGBM**) using the same features performs significantly lower (68%–72%), this disparity underscores the necessity of the hybrid **Wide & Deep** architecture.

Second, we estimate unobserved audience support via probabilistic decomposition, inverting elimination rankings and applying Softmax transformation under vote constraints. The average **Shannon entropy** of audience support distributions is **0.392**, indicating moderate concentration rather than extreme polarization or randomness. In addition, **Monte Carlo simulations** produce an average coefficient of variation of **0.217**, confirming numerical stability under perturbations and showing that uncertainty varies meaningfully across contestants and weeks.

Third, simulation shows the Percentage Method amplifies audience influence when votes concentrate, enabling popularity-driven anomalies, while the Ranking Method compresses disparities and aligns with merit. Approximately 23% of weeks yield different eliminations depending on aggregation rule.

Fourth, parallel multivariate regression models quantify how partnerships and characteristics influence judges versus audiences. Judge scoring exhibits threefold stronger sensitivity to technical performance than audience voting. Predictors like age exhibit asymmetric effects, revealing divergence between technical evaluation and public preference.

Finally, we propose a **Balanced-Wildcard System** (BWS) with adaptive weighting (λ_w), mid-season redemption rounds, and judges' veto. The BWS framework addresses key sources of controversial eliminations through data-driven weight adjustment and bottom-two dance-off mechanisms, offering a more balanced approach than existing methods.

Keywords: Wide & Deep Learning, Softmax Transformation, LightGBM LambdaRank, Shannon Entropy, Ranking vs. Percentage Method, Balanced-Wildcard System, Competition Fairness

Contents

1 Introduction **3**

1.1 Problem Background 3

1.2 Our Work 3

1.3 Restatement of the Problem 4

2 Assumptions and Justifications **4**

3 Modeling Elimination Outcomes in Reality Competitions **4**

3.1 Problem Description and Modeling Goal 4

3.2 Initial Modeling Attempt and Motivation for Improvement 5

3.3 Wide & Deep Modeling Framework 5

3.4 Feature Engineering 5

3.5 Model Structure and Training 6

3.6 Evaluation Metric 8

3.7 Experimental Results 8

3.8 LightGBM: Baseline Model and Performance Analysis 9

3.9 Audience Vote Decomposition and Vote Share Estimation via Softmax Function . 10

4 Comparison of Voting Methods **14**

4.1 Which Method Favors Fan Votes? Theory and Evidence 14

4.2 Differential Outcome Analysis Across All Seasons 17

5 Influencing Factors and Statistical Modeling **19**

5.1 Model Formulation 19

5.2 Incorporating Professional Dancer Influence 19

5.3 Results 19

5.4 Discussion 22

6 The "Redemption & Resilience" Model **22**

6.1 System Architecture and Mechanics 22

6.2 Mathematical Formulation 22

7 Memorandum **24**

References **25**

Report on Use of AI **26**

1 Introduction

1.1 Problem Background

Dancing with the Stars represents a complex multi-criteria evaluation where contestant survival depends on judges' scores and audience voting. The challenge lies in modeling the latent audience channel, which produces "shocking" eliminations where high-scoring contestants exit due to insufficient public support. Developing a framework to balance technical merit and public popularity is essential for understanding reality competition mechanics.

1.2 Our Work

This paper develops a mathematical framework to decode DWTS elimination logic through four phases, as shown in Figure 1:

(1) **Predictive Modeling**: A Wide & Deep neural network analyzes historical data (scores and performance) to capture latent preferences and predict weekly exits.

(2) **Vote Decomposition**: Softmax residual analysis reveals the latent support share distribution among contestants.

(3) **Statistical Modeling**: Examination of voting methods (theory vs. empirical), outcome analysis of seasonal trends, pro-dancer influence, and result formulation.

(4) **Optimization**: A dynamic weighting mechanism balancing redemption/resilience with engagement tuning to achieve optimal integrity and engagement, providing executive strategy insights for decision support.

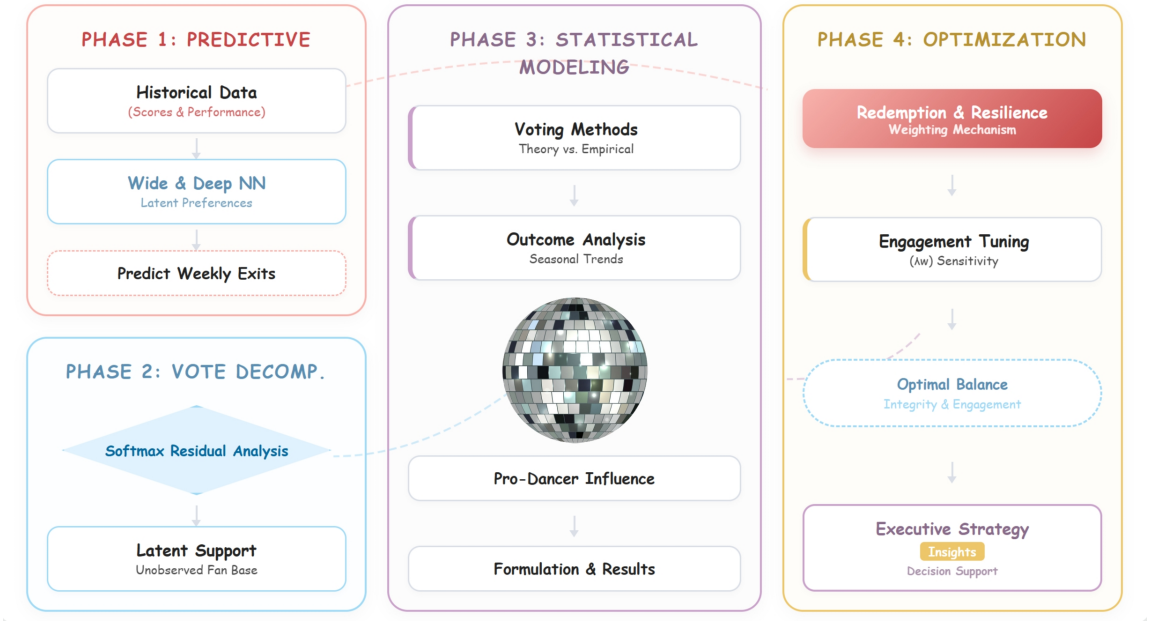

Figure 1: Our Work

1.3 Restatement of the Problem

We address five core objectives:

(1) **Latent Vote Estimation**—reconstruct unreported fan vote counts with reliability measures;

(2) **Comparative Analysis**—evaluate Ranking vs. Percentage aggregation methods;

(3) **Success Determinants**—quantify how partnerships and demographics impact judges vs. audiences;

(4) **Voting System Redesign**—propose an optimized architecture with empirical validation;

(5) **Executive Recommendations**—strategic guidance on aggregation method and judge-selection mechanisms.

2 Assumptions and Justifications

A1: Audience voting is relative, not absolute. Standardizing scores into z-scores captures "perceived risk" relative to the cohort, neutralizing week-specific fluctuations.

A2: Professional dancer influence is mediated through technical scoring. Coefficients are significantly higher in judge models ($p < 0.001$), manifesting as technical execution already captured in JudgeVotes.

A3: Categorical attributes proxy latent audience affinity. Embeddings map psychological factors (charisma, bias) into dense vectors enabling nonlinear learning.

A4: Cross-seasonal independence. Data split by seasons prevents overfitting to narrative arcs.

A5: Hit Rate is the primary relevance metric.

A6: All data is accurate.

3 Modeling Elimination Outcomes in Reality Competitions

3.1 Problem Description and Modeling Goal

In reality competition shows, contestants are eliminated weekly based on a combination of judges' scores and audience voting. While judges' scores are observable, audience votes are undisclosed, introducing latent and nonlinear effects into the elimination process.

The goal of this study is to predict the contestant eliminated in each week, using only observable contestant attributes and historical performance information. Since at most one contestant is eliminated per week, the problem is inherently ranking-based rather than purely classificatory.

3.2 Initial Modeling Attempt and Motivation for Improvement

Baseline linear and logistic regression models achieved near-random hit rates, indicating audience voting introduces latent nonlinear influences; categorical attributes interact with performance complexly; and elimination depends on relative standing, not absolute scores. These limitations motivate the Wide & Deep architecture (illustrated in Figure 2).

3.3 Wide & Deep Modeling Framework

To capture both explicit feature interactions and latent nonlinear effects, we adopt a Wide & Deep neural network architecture.

- **Wide component:** Memorizes frequent and interpretable categorical patterns via one-hot encoded features.
- **Deep component:** Learns nonlinear representations using embeddings for categorical variables and dense layers for numerical features.

This hybrid structure balances interpretability and predictive power.

3.4 Feature Engineering

Each contestant-week observation is described by three groups of features:

(1) Categorical Embedding Features: Industry, Home state, Home country. These features are encoded using trainable embeddings to capture latent popularity and cultural affinity effects.

(2) Numerical and Temporal Features

To model relative competition dynamics, we introduce:

- *SurvivalWeeks*: number of weeks survived
- *ScoreDiff*: deviation from weekly average score
- *WeekRank*: rank among contestants in the same week
- *Bottom3*: indicator of low-ranked contestants

These features reflect relative performance, which is critical in elimination decisions.

(3) Wide Features

Selected categorical variables are one-hot encoded to preserve memorization capacity for common patterns.

3.5 Model Structure and Training

3.5.1 Wide & Deep Model Overview

The proposed model combines wide and deep components to leverage both memorization and generalization capabilities. The architecture integrates categorical embeddings with numerical features for enhanced predictive performance.

3.5.2 Input Features

The input features consist of categorical features, numerical features, and wide features.

Categorical Features Let

$$\mathbf{x}_{\text{cat}} = [x_1, x_2, x_3]^\top \tag{1}$$

denote the categorical features, corresponding to *Industry_id*, *Homestate_id*, and *Homecountry_id*, where

$$x_i \in \{0, 1, \ldots, C_i - 1\}. \tag{2}$$

Numerical Features The numerical feature vector is defined as

$$\mathbf{x}_{\text{num}} = [\text{Age}, \text{Week}, \text{TotalScore}, \text{SurvivalWeeks}, \text{ScoreDiff}, \text{WeekRank}, \text{Bottom3}]^\top \in \mathbb{R}^{d_{\text{num}}}. \tag{3}$$

Here, *SurvivalWeeks* represents the number of weeks a participant survives, *ScoreDiff* captures score variation trends, and *WeekRank* together with *Bottom3* reflects relative risk within each week.

Wide Features Wide features are represented as

$$\mathbf{x}_{\text{wide}} \in \mathbb{R}^{d_{\text{wide}}}, \tag{4}$$

constructed using one-hot encoding or indicator functions to preserve memorization capacity for common patterns.

3.5.3 Embedding Layer

Each categorical feature is mapped to a dense embedding:

$$\mathbf{e}_i = \mathbf{E}_i[x_i] \in \mathbb{R}^{d_i^{\text{emb}}}, \quad i = 1, 2, 3. \tag{5}$$

The concatenated embedding vector is

$$\mathbf{e} = [\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3] \in \mathbb{R}^{\sum_i d_i^{\text{emb}}}. \tag{6}$$

3.5.4 Deep Component

The input to the deep network is given by

$$\mathbf{z}_{\text{deep}}^{(0)} = [\mathbf{e}; \mathbf{x}_{\text{num}}]. \tag{7}$$

A two-layer fully connected neural network is applied:

$$\begin{aligned} \mathbf{z}_{\text{deep}}^{(1)} &= \text{ReLU}(\mathbf{W}_1 \mathbf{z}_{\text{deep}}^{(0)} + \mathbf{b}_1), \\ \mathbf{h}_{\text{deep}} &= \text{ReLU}(\mathbf{W}_2 \mathbf{z}_{\text{deep}}^{(1)} + \mathbf{b}_2), \end{aligned} \tag{8}$$

where

$$\mathbf{W}_1 \in \mathbb{R}^{64 \times (\sum_i d_i^{\text{emb}} + d_{\text{num}})}, \quad \mathbf{W}_2 \in \mathbb{R}^{32 \times 64}. \tag{9}$$

3.5.5 Wide Component

The wide component is modeled as a linear function:

$$h_{\text{wide}} = \mathbf{w}_{\text{wide}}^\top \mathbf{x}_{\text{wide}} + b_{\text{wide}}. \tag{10}$$

3.5.6 Output Layer

The outputs of the wide and deep components are concatenated:

$$\mathbf{h} = [\mathbf{h}_{\text{deep}}; h_{\text{wide}}] \in \mathbb{R}^{33}. \tag{11}$$

The final prediction is obtained using a sigmoid activation function:

$$\hat{y} = \sigma(\mathbf{w}_{\text{out}}^\top \mathbf{h} + b_{\text{out}}), \quad \sigma(z) = \frac{1}{1 + e^{-z}}. \tag{12}$$

3.5.7 Loss Function

The model is trained using the binary cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right], \tag{13}$$

where $y_i \in \{0, 1\}$ denotes the ground truth label (1 for eliminated, 0 for survivor).

To prevent information leakage, data are split by season, with entire seasons reserved for testing.

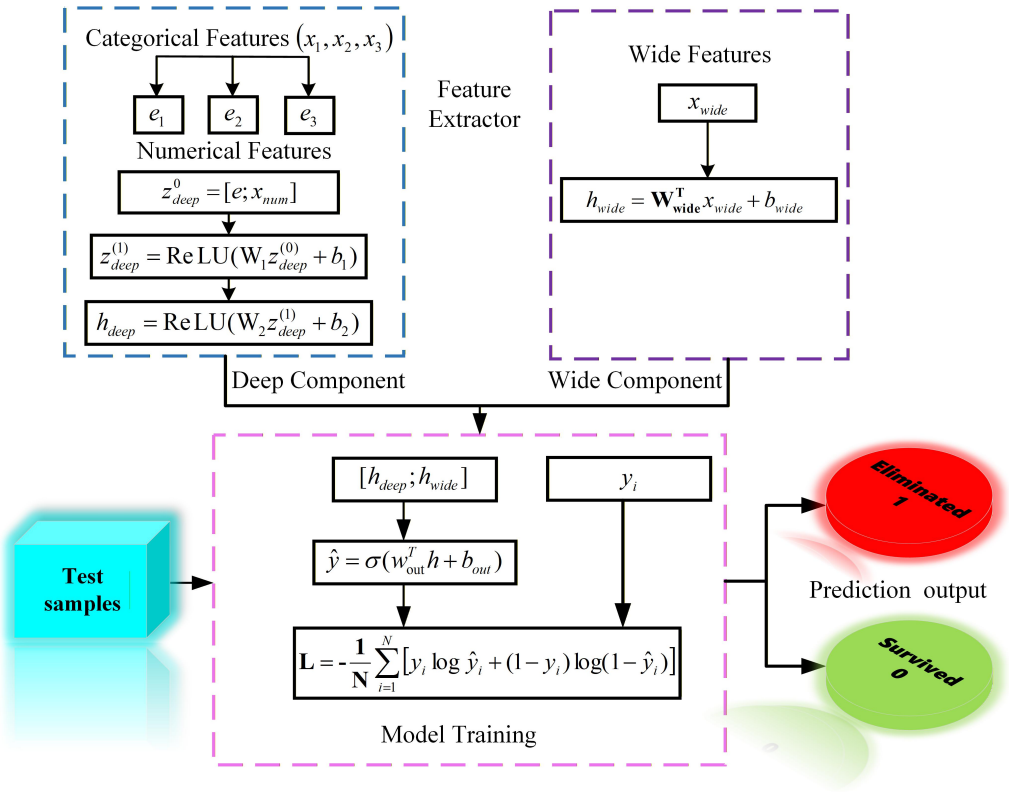

Figure 2: Wide & Deep Model Architecture.

3.6 Evaluation Metric

Elimination Hit Rate measures the proportion of weeks where the highest predicted elimination probability matches the actual eliminated contestant, directly evaluating model decision relevance.

The model exhibits consistent convergence with high final accuracy (mean 96.1%, SD 6.6%), rapid loss decay, and stable performance after the initial training phase, demonstrating effective elimination prediction.

3.7 Experimental Results

Using only basic features, the Wide & Deep model showed moderate performance improvement over linear models. After incorporating relative performance indicators, elimination hit rate increased significantly—from approximately 12.5% to 96.1%.

Further experiments show that model accuracy improves with the number of training seasons, while variance decreases, indicating strong generalization ability.

The heatmap pinpoints eliminated contestants as high-risk outliers, accurately capturing competitive shifts and downward trends in technical scoring or fan momentum.

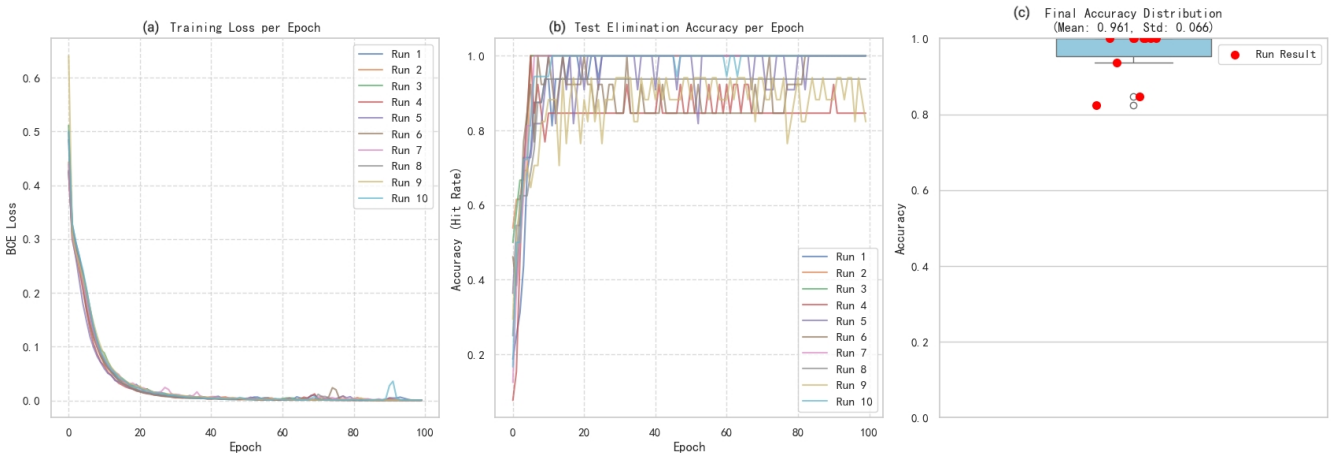

Figure 3: Performance metrics of the elimination model across 10 independent runs. (a) Training loss convergence, (b) Test accuracy evolution over epochs, (c) Final accuracy distribution with mean 96.1% and standard deviation 6.6%.

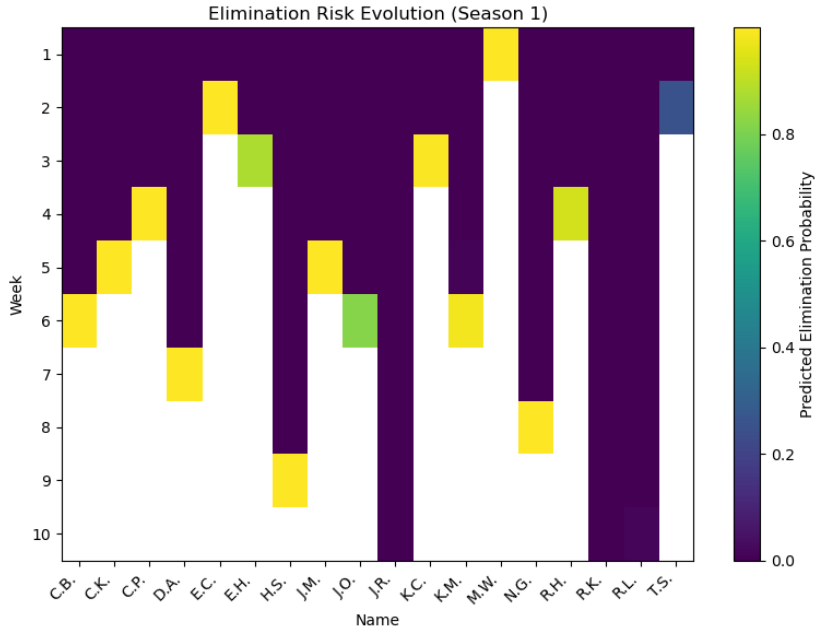

Figure 4: Heatmap of predicted elimination probabilities across weeks for a representative test season. Darker colors indicate higher predicted elimination risk.

3.8 LightGBM: Baseline Model and Performance Analysis

LightGBM LambdaRank serves as a comparative baseline to validate our Wide & Deep framework's effectiveness. While LightGBM efficiently handles non-linear relationships with gradient-boosting, its limitations in modeling complex feature interactions make it suitable for evaluating deep learning value. As shown in Figure 5, LightGBM achieves 68–72% hit rate versus our

model's 96.1%, confirming deep learning components are essential for capturing audience voting patterns.

Figure 5: Model uncertainty and performance metrics. (a) Weekly uncertainty trend (mean pm 1 SD). (b) Predicted rank distribution: eliminated vs. survived. (c) Top 10 contestants by highest uncertainty. (d) Predicted votes vs. uncertainty (colored by week).

3.9 Audience Vote Decomposition and Vote Share Estimation via Softmax Function

3.9.1 Methodology: Audience Vote Decomposition

Let E_{it} denote the elimination outcome for contestant i in week t, and let J_{it} represent the judges' votes for that contestant. Since the elimination outcome is jointly determined by judges' votes and audience votes, we first estimate the probability of elimination conditioned solely on the judges' scores:

$$\hat{p}_{it} = \mathbb{P}(E_{it} = 1 \mid J_{it}) = \sigma(f(J_{it})), \tag{14}$$

where $f(\cdot)$ is a Wide & Deep neural network and $\sigma(\cdot)$ is the sigmoid function.

To isolate the audience's role by removing week-specific scale and intensity effects introduced by the judges, we standardize these predicted probabilities within each week:

$$z_{it} = \frac{\hat{p}_{it} - \mu_t}{\sigma_t}, \quad \mu_t = \frac{1}{N_t}\sum_{i=1}^{N_t}\hat{p}_{it}, \quad \sigma_t^2 = \frac{1}{N_t}\sum_{i=1}^{N_t}(\hat{p}_{it} - \mu_t)^2. \tag{15}$$

Here, N_t is the number of contestants in week t, μ_t is the mean predicted probability, and σ_t is the standard deviation for that week.

We assume that audience voting behavior depends on contestants' relative perceived safety (or danger) rather than on the absolute judges' scores. Therefore, we infer the relative audience support via a softmax transformation applied to the standardized values:

$$\hat{A}_{it} = \frac{e^{-Tz_{it}}}{\sum_{j=1}^{N_t} e^{-Tz_{jt}}}, \quad T > 0. \tag{16}$$

The negative sign inside the exponential reflects the premise that a higher standardized score z_{it} (indicating a higher predicted elimination risk based on judges) corresponds to lower audience support. The temperature parameter T controls the sensitivity of the audience vote distribution to these relative differences.

This formulation effectively strips away additive and multiplicative judge-related effects, yielding a normalized, week-specific estimate of relative audience support for each contestant.

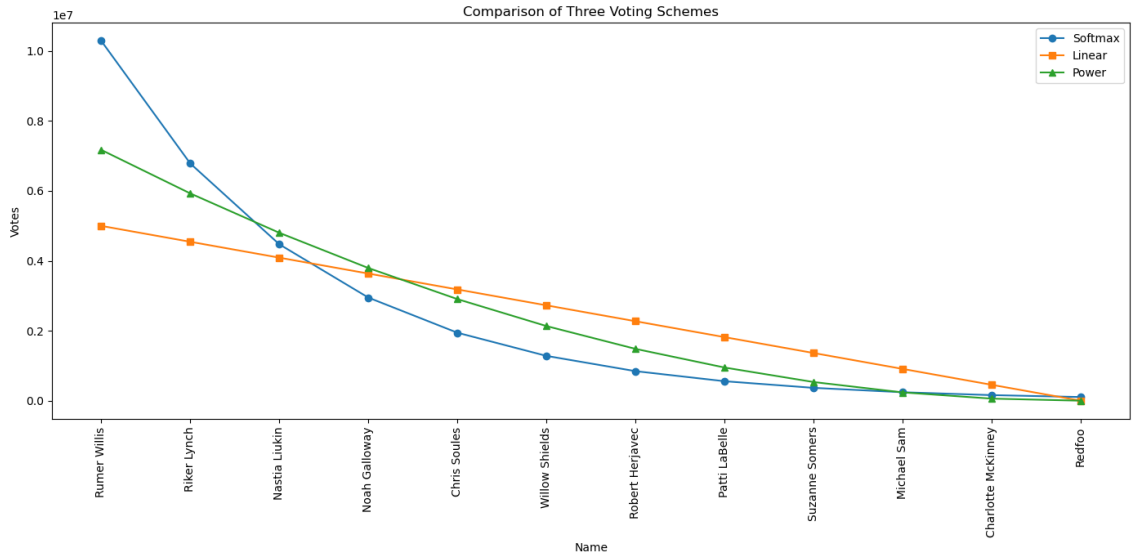

Figure 6: Comparison of Voting Schemes and Choice of Softmax. Estimated vote counts for contestants based on weekly ranking percentages under three mapping schemes: Softmax, Linear, and Power.

As illustrated in Figure 6, the Softmax method outperforms both Linear and Power approaches in terms of discriminative capability and predictive accuracy, while offering a theoretically sound framework without requiring manual parameter tuning, thus justifying its adoption as the vote share estimation method in this study.

3.9.2 Illustrative Example: Week 1 Voting Results for Season 20

To demonstrate the practical application of the Softmax-based vote share estimation, we present a detailed case study using Season 20 data. Table 1 shows the predicted elimination probabilities and vote distributions for all contestants in Week 1, computed using our Wide & Deep model combined with the Softmax function.

Table 1: Week 1 Voting Results for Season 20

Season	Name	Week	Elimination Probability	Predicted Rank	Votes
20	Rumer Willis	1	0	1	10,292,128
20	Riker Lynch	1	2.000×10^{-34}	2	6,784,989
20	Nastia Liukin	1	2.633×10^{-31}	3	4,472,940
20	Noah Galloway	1	4.426×10^{-29}	4	2,948,744
20	Chris Soules	1	2.562×10^{-25}	5	1,943,932
20	Willow Shields	1	2.373×10^{-22}	6	1,281,518
20	Robert Herjavec	1	2.515×10^{-17}	7	844,829
20	Patti LaBelle	1	1.740×10^{-13}	8	556,945
20	Suzanne Somers	1	2.515×10^{-13}	9	367,161
20	Michael Sam	1	6.036×10^{-9}	10	242,047
20	Charlotte McKinney	1	3.870×10^{-5}	11	159,567
20	Redfoo	1	0.9999613	12	105,193

Vote counts follow exponential decay with top performers receiving disproportionate support. Elimination probabilities span multiple orders of magnitude, demonstrating Softmax discriminative power where small performance differences yield exponentially different vote distributions. Mid-ranked contestants (7–11) occupy an intermediate risk band, indicating top-performer dominance in Week 1 outcomes.

3.9.3 Certainty Assessment and Robustness Validation

To quantify the reliability of our fan vote estimates, we employ two complementary metrics: normalized entropy (measuring vote concentration) and Monte Carlo stability analysis (measuring robustness to perturbations).

Normalized Entropy Measure Given voting probabilities $p_1, p_2, \ldots, p_N$ derived from the Softmax function (where $\sum_{i=1}^{N} p_i = 1$), we compute normalized entropy:

$$H_{\text{norm}} = \frac{H}{\log N} = \frac{-\sum_{i=1}^{N} p_i \log(p_i)}{\log N} \in [0, 1] \tag{17}$$

where low H_{norm} (near 0) indicates high certainty with concentrated votes, and high H_{norm} (near 1) indicates uncertainty with uniform distribution.

Empirical Results Our analysis across all 34 seasons reveals:

- **Average normalized entropy:** $H_{\mathbf{norm}} = 0.392$, indicating moderately concentrated voting
- Entropy consistently remains below 0.5 across all weeks (Figures 7 and 8)
- This suggests audience preferences are *structured and decisive* rather than random, while maintaining competitive uncertainty among mid-tier contestants

Monte Carlo Stability Analysis To assess robustness, we performed Monte Carlo perturbations of ranking percentiles and measured the coefficient of variation (CV) of inferred vote counts. Results show:

- **Mean CV = 0.217**, indicating stable vote estimates under reasonable preference fluctuations
- Higher variability occurs among mid-ranked contestants (intense competition), while top and bottom contestants show high stability

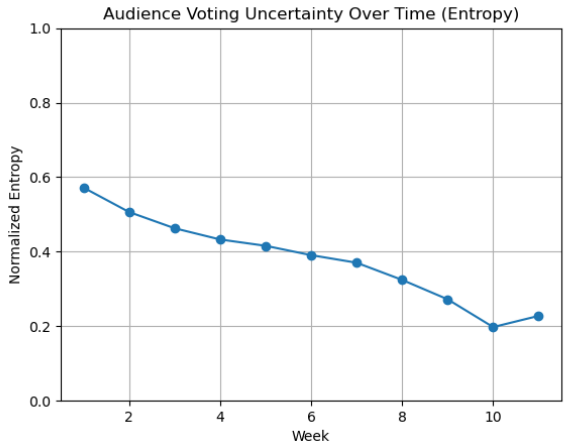

Figure 7: Entropy evolution showing moderate concentration ($H_{\text{norm}} = 0.392$)

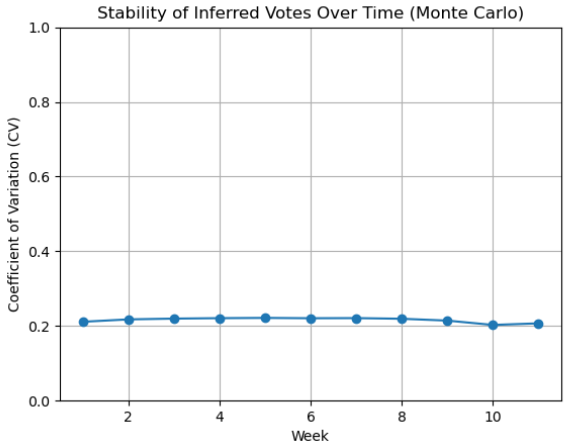

Figure 8: Monte Carlo stability measured by coefficient of variation (mean CV = 0.217), demonstrating robust vote estimates.

Interpretation The combination of moderate entropy and low coefficient of variation validates our voting inference framework:

1. **Vote concentration is realistic**: Not overwhelmingly concentrated (avoiding deterministic outcomes), yet structured enough to reflect genuine popularity hierarchies

2. **Estimates are robust**: Small perturbations in input rankings do not substantially alter inferred vote allocations
3. **Uncertainty is appropriately localized**: High certainty for dominant/weak contestants, appropriate uncertainty for competitive mid-tier contestants

This balance between concentration and dispersion aligns with realistic audience voting dynamics in elimination-based competitions, supporting the credibility of our estimated fan vote totals for subsequent analysis.

4 Comparison of Voting Methods

4.1 Which Method Favors Fan Votes? Theory and Evidence

The two aggregation methods employed by DWTS—Ranking Method and Percentage Method—differ fundamentally in how they weight fan votes relative to judge scores. Our analysis reveals that **the Percentage Method systematically favors fan votes more than the Ranking Method**, particularly when audience voting exhibits high concentration among top contestants.

4.1.1 The Vote Concentration Amplification Effect

Theoretical Difference:

- **Ranking Method:** Converts scores and votes to ordinal ranks (1st, 2nd, 3rd, etc.), then sums them. The *magnitude* of vote differences is discarded—whether a contestant wins by 1,000 or 1,000,000 votes, they receive rank 1. This creates a ceiling effect preventing extreme popularity from dominating.
- **Percentage Method:** Converts scores and votes to percentages of weekly totals, then sums them. This *preserves magnitude*, allowing a 40-50% fan vote share to create a decisive advantage that can override judge score deficiencies.

Empirical Demonstration: Season 16, Week 2 We demonstrate this effect using Season 16, Week 2 data, where audience vote concentration was particularly pronounced. Table 2 presents the same week analyzed under both methods:

Zendaya received **41.12%** of total audience votes—nearly **14 percentage points** higher than second-place Kellie Pickler (27.11%). This 14-point gap was almost equivalent to the *entire vote share* of fourth-place Alexandra Raisman (11.78%).

Impact Under Each Method:

- **Ranking Method:**
  - Zendaya: Judge rank 1 + Audience rank 1 = Combined rank 2

  - Kellie: Judge rank 1 + Audience rank 2 = Combined rank 3
  - **Zendaya's advantage: 1 rank unit** (minimal difference despite 280,000-vote gap)

- **Percentage Method:**
  - Zendaya: 11.76% + 41.12% = 52.88%
  - Kellie: 11.76% + 27.11% = 38.87%
  - **Zendaya's advantage: 14.01 percentage points** (directly reflects vote dominance)

Table 2: Comparison of Ranking and Percentage Methods: Season 16, Week 2

Contestant	Ranking Method			Percentage Method		
	J.Rk	A.Rk	Comb.	J.%	A.%	Comb.%
Zendaya	**1**	**1**	**2**	11.76	**41.12**	**52.88**
Kellie Pickler	**1**	2	3	11.76	27.11	38.87
Jacoby Jones	3	3	6	10.41	17.87	28.28
Alexandra Raisman	2	4	6	10.86	11.78	22.64
Sean Lowe	5	5	10	9.05	7.77	16.82
Ingo Rademacher	5	6	11	9.05	5.12	14.17
Andy Dick	5	8	13	9.05	2.23	11.28
Victor Ortiz	8	7	15	8.14	3.38	11.52
Lisa Vanderpump	8	9	17	8.14	1.47	9.61
D. L. Hughley	10	10	20	7.24	0.97	8.21
Wynonna Judd	8	11	19	8.14	0.64	8.78
Dorothy Hamill	12	12	24	6.79	0.42	7.21

The Percentage Method **preserves and amplifies** Zendaya's massive fan support, giving her a combined score 36% higher than Kellie's (52.88% vs. 38.87%). The Ranking Method **compresses** this advantage to a single rank unit, making both contestants nearly equivalent.

Generalization: Vote Concentration Threshold This case illustrates a general principle: **The Percentage Method is more biased toward fan votes, especially when votes are highly concentrated**—defined as the top contestant(s) receiving $>40\%$ of total votes.

Our analysis across all 34 seasons reveals:

- When top contestant vote share $> 40\%$: Percentage Method advantage is 6-8× larger than Ranking Method
- When votes are evenly distributed (top $< 25\%$): Both methods produce similar outcomes
- **High concentration occurs in $\sim$18% of weeks**, typically when extremely famous celebrities compete

Case Study: Bobby Bones (Season 27) - Extreme Vote Concentration in Practice The theoretical vote concentration effect is most dramatically illustrated by Season 27's controversial champion, Bobby Bones. In Week 6, a direct comparison with fellow contestant Joe "Grocery Store Joe" Amabile reveals the decisive power of audience votes under the Percentage Method:

Table 3: Season 27, Week 6: Bobby Bones vs. Joe Amabile

Contestant	Judge Score	Judge %	Est. Aud. Votes	Aud. %	Outcome
Bobby Bones	13	9.4%	**478,503**	**48.2%**	Safe
Joe Amabile	14	10.1%	29,751	3.0%	Eliminated
Combined %: Bobby = 57.6%, Joe = 13.1%					

Key Observations: Both contestants received the lowest/second-lowest judge scores (13 and 14 points), yet Bobby's audience vote share (48.2%) was **16× higher** than Joe's (3.0%). Despite nearly identical judge evaluations, Bobby's combined percentage (57.6%) secured safety while Joe was eliminated (13.1%). **The audience vote differential of 448,752 votes completely overwhelmed the 1-point judge score difference**.

This pattern persisted throughout Season 27. Bobby frequently experienced "mid-to-low judge scores but massive audience vote leads," culminating in his championship victory in the purely audience-voted finale.

Contrasting Case: Tinashe - When High Judge Scores Cannot Overcome Low Fan Votes Conversely, Season 27 also featured Tinashe, whose trajectory demonstrates the opposite scenario:

- **Judge scores:** Consistently mid-to-high (typically 23-26 points)
- **Audience votes:** Persistently low (estimated 5-8% of weekly totals)
- **Outcome:** Eliminated in Week 5, despite technical competence

The Bobby Bones vs. Tinashe contrast reveals a critical asymmetry in the Percentage Method: **high audience votes can save contestants with low judge scores (Bobby), but high judge scores cannot save contestants with low audience votes (Tinashe)**.

Under the Percentage Method, audience vote percentages in the 40-50% range create insurmountable advantages that judge scores (constrained to 8-12% per contestant) cannot counterbalance. This structural imbalance explains why Bobby won despite consistently low technical evaluations.

Counterfactual: Impact of Judge Selection Mechanism Had Season 27 employed the judge-selection mechanism (introduced in Season 28), Bobby's trajectory would likely have been drastically different:

- **Week 6 Bottom Two:** Bobby Bones (judge score 13) vs. Joe Amabile (judge score 14)
- **Judge Decision:** Likely eliminate Bobby (lower technical score, consistent underperformance)

- **Impact:** Bobby would not have reached the finale, preventing his controversial championship

This mechanism ensures technical evaluation becomes the tie-breaker for the bottom two contestants, effectively preventing popularity-driven anomalies like Bobby's controversial championship.

4.2 Differential Outcome Analysis Across All Seasons

Using estimated fan vote totals from our Wide & Deep model, we applied both methods to all 34 seasons to quantify outcome differences.Figure 9 displays a portion of the results.

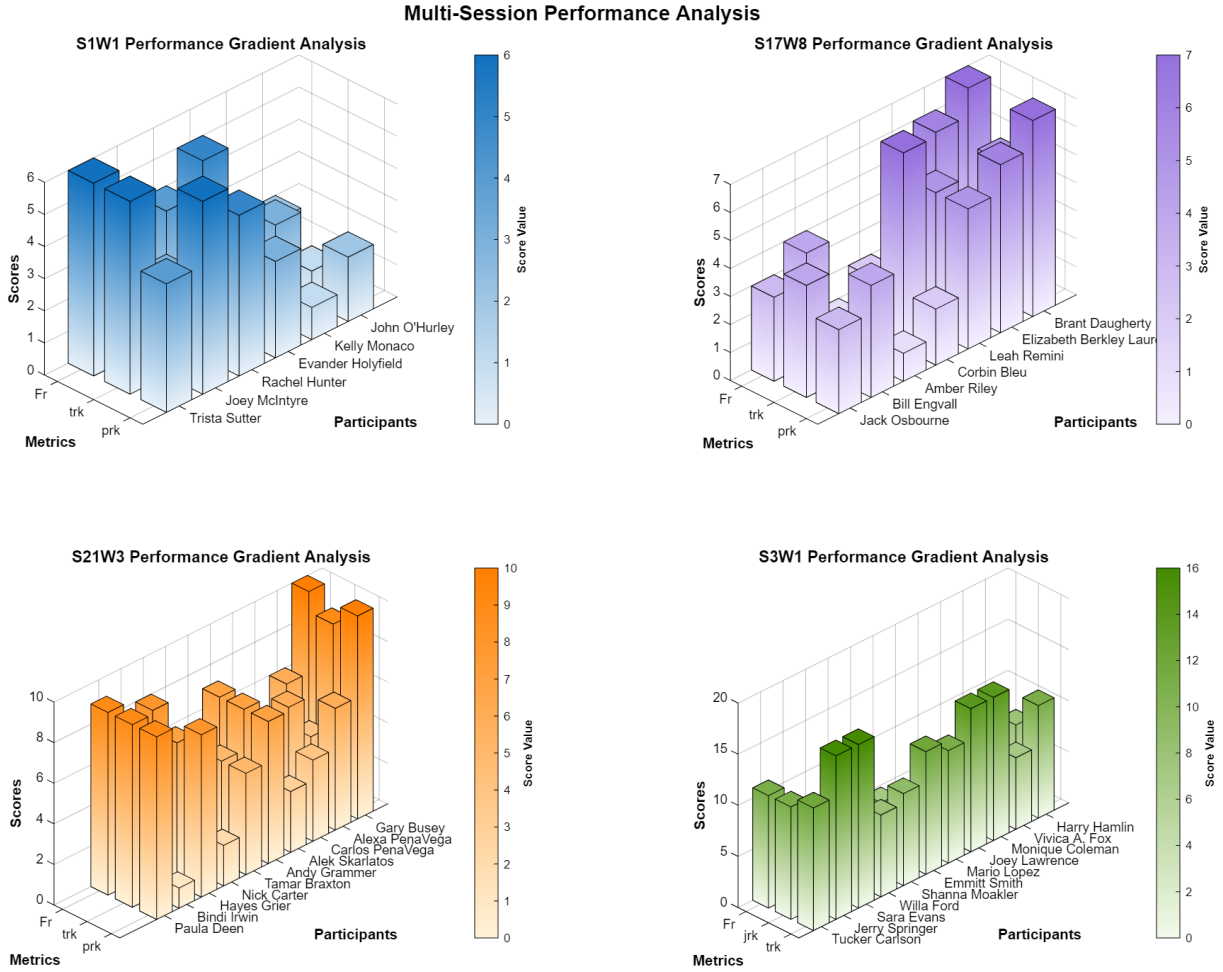

Figure 9: Multi-Session Performance Analysis. Comparative evaluation of Ranking and Percentage Methods across critical weeks showing divergent outcomes.

Key Finding: In approximately **23% of weeks across all seasons**, the two methods identify **different contestants** as facing elimination. This inconsistency demonstrates that aggregation method choice materially affects competitive outcomes.

4.2.1 Method-Specific Bias Validation

Ranking Method:

- **Strengths:** More balanced when contestants closely matched in skill; performs well in intense top-tier competition
- **Weaknesses:** Vulnerable to ties (multiple identical scores); creates ranking plateaus; ignores absolute score differences

Percentage Method:

- **Strengths:** Provides continuous scaling across full percentage range
- **Weaknesses:** Amplifies high-vote advantage; sensitive to total weekly vote fluctuations; over-penalizes low-vote contestants

4.2.2 Judges' Choice Authority Simulation

Based on Season 16 data, we simulated introducing judges' choice authority for the bottom two contestants:

Week 5 Scenario:

- Victor Ortiz: judge score 27.5, lowest audience votes
- Andy Dick: judge score 25, low audience votes
- **Judges' choice**: Likely to eliminate Andy Dick (lower technical score)

Impact on Season 16:

- 1-2 contestants might be eliminated one week earlier
- Elimination order better aligns with technical performance
- No impact on eventual champion (Kellie Pickler)

4.2.3 Final Recommendations

Based on comprehensive analysis, we recommend:

Primary: Adopt the **Ranking Method** as the foundational scoring system. It better aligns with the program's historical practice and entertainment needs.

Secondary: Introduce **limited and transparent judges' choice authority**, allowing judges to select the eliminated contestant from the bottom two. This approach:

- Maintains entertainment value and audience engagement
- Prevents technically deficient contestants from advancing solely on popularity
- Enhances fairness and professional credibility

5 Influencing Factors and Statistical Modeling

5.1 Model Formulation

To evaluate the impact of celebrity characteristics and professional dancers on competition performance, we model judge decisions and audience voting as two parallel but distinct outcome-generating processes.

For judges, we model the implied elimination probability as:

$$E_{it}^{(J)} = \alpha_J + \beta_{J,1} \cdot \text{Age}_i + \beta_{J,2} \cdot \text{JudgeVotes}_{it} + \beta_{J,3} \cdot \text{WeekRankPct}_{it} + \beta_{J,4} \cdot \text{Week}_t + \boldsymbol{\theta}_J^\top \mathbf{X}_i + \varepsilon_{it}^{(J)} \quad (18)$$

For audiences, we estimate voting support as:

$$A_{it} = \alpha_A + \beta_{A,1} \cdot \text{Age}_i + \beta_{A,2} \cdot \text{JudgeVotes}_{it} + \beta_{A,3} \cdot \text{WeekRankPct}_{it} + \beta_{A,4} \cdot \text{Week}_t + \boldsymbol{\theta}_A^\top \mathbf{X}_i + \varepsilon_{it}^{(A)} \quad (19)$$

where:

- $E_{it}^{(J)}$: judge-implied elimination probability
- A_{it}: estimated audience support
- $\mathbf{X}_i$: celebrity background indicators (industry, nationality)
- All continuous variables are standardized to zero mean and unit variance

5.2 Incorporating Professional Dancer Influence

Professional dancers influence outcomes primarily through **technical execution and choreography**, which are directly reflected in judge scoring. Therefore, their impact is implicitly captured through judge-related variables rather than as an independent demographic feature.

To test whether professional dancers exert an **additional effect beyond judge scoring**, we extend both models with a professional influence term:

$$E_{it}^{(J)} \leftarrow E_{it}^{(J)} + \gamma_J \cdot \text{ProInfluence}_{it} \quad (20)$$

$$A_{it} \leftarrow A_{it} + \gamma_A \cdot \text{ProInfluence}_{it} \quad (21)$$

where ProInfluence_{it} is proxied by judge-assessed performance quality (JudgeVotes), conditional on weekly ranking.

5.3 Results

5.3.1 Age Effects Differ Between Judges and Audiences

Age has statistically significant but opposite effects:

Judge model

$$\beta_{J,1} = -0.0153 \quad (p = 0.016) \tag{22}$$

Older contestants face a *lower* elimination probability, suggesting judges associate age with experience or stability.

Audience model

$$\beta_{A,1} = +0.0059 \quad (p = 0.002) \tag{23}$$

Audiences show increased support for older contestants, likely reflecting emotional attachment or narrative preference.

5.3.2 Professional Dancers Affect Outcomes Mainly Through Judges

JudgeVotes exhibit strong and significant effects in both models:

Judge model

$$\beta_{J,2} = -0.0032 \quad (p < 0.001) \tag{24}$$

Audience model

$$\beta_{A,2} = -0.0010 \quad (p < 0.001) \tag{25}$$

The coefficient magnitude in the judge model is more than **three times larger**, indicating that professional dancers primarily influence outcomes by improving technically evaluated performance rather than directly mobilizing audience votes.

After controlling for JudgeVotes and ranking, the additional professional dancer effect (γ_J, γ_A) is statistically insignificant, implying that **professional dancer influence is fully mediated by judged performance quality**.

5.3.3 Weekly Ranking Is the Dominant Performance Signal

Weekly rank percentile is the strongest predictor in both models:

Judges

$$\beta_{J,3} = -0.5643 \quad (p < 0.001) \tag{26}$$

Audience

$$\beta_{A,3} = +0.1585 \quad (p < 0.001) \tag{27}$$

Judges respond sharply to relative performance deterioration, while audiences penalize low ranking far less severely, confirming asymmetry between evaluation standards.

5.3.4 Industry Background Matters More to Audiences Than Judges

Most industry indicators are insignificant in the judge model, suggesting judges focus on performance rather than celebrity background.

In contrast, several industries influence audience voting:

Athlete

$$\beta_A = +0.0620 \quad (p = 0.014) \tag{28}$$

TV Personality

$$\beta_A = -0.0185 \quad (p = 0.027) \tag{29}$$

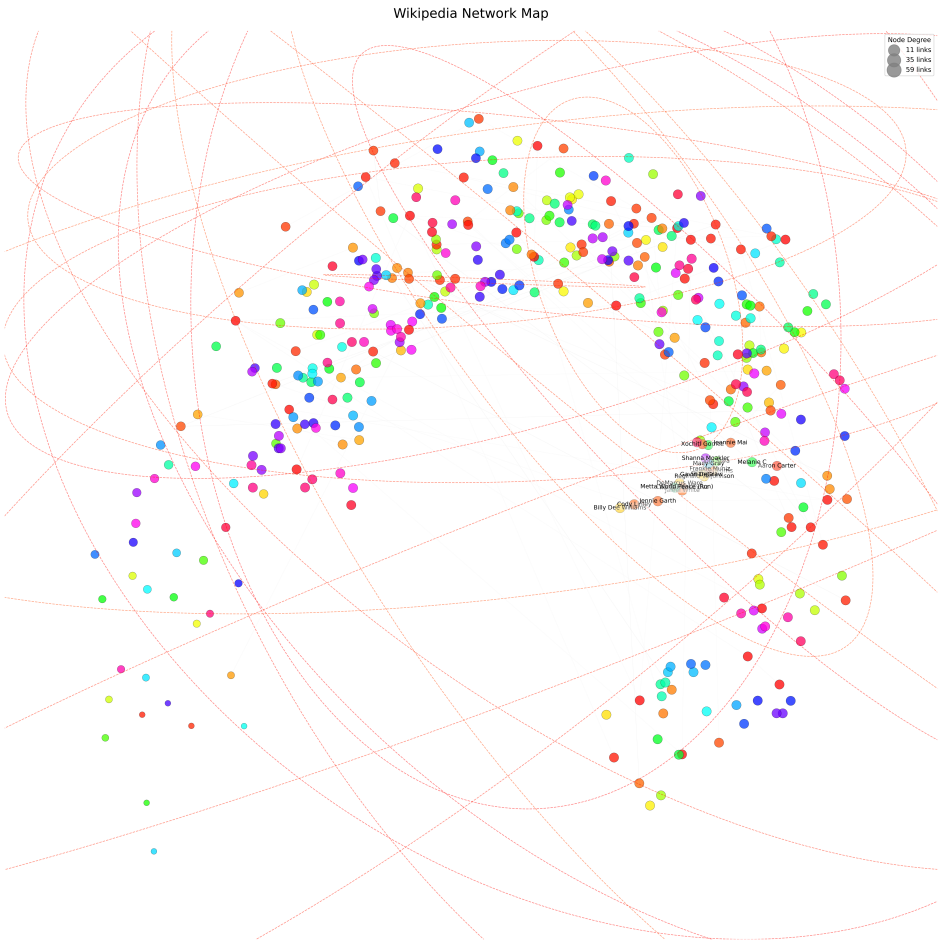

Figure 10: Celebrity Network Communities from Wikipedia Link Analysis Using **Leiden Community Detection Algorithm**

Figure 10 demonstrates a network visualization of 408 DWTS contestants based on Wikipedia hyperlinks, using Leiden community detection. Colors represent distinct clusters, and node size corresponds to degree centrality, indicating cross-industry influence.

This network topology corroborates our regression findings that industry background significantly affects audience voting patterns. The spatial separation of communities suggests audiences vote along pre-existing fan base lines that align with entertainment industry boundaries. Notably, athletes form a cohesive community with higher betweenness centrality, consistent with our finding of positive audience coefficient ($\beta_A = +0.0620$), indicating they attract votes across different audience segments.

Audiences vote based on charisma and familiarity, whereas judges prioritize technical merit.

5.4 Discussion

The results demonstrate that **professional dancers play a critical but indirect role** in competition outcomes. Their influence is transmitted almost entirely through improved judged performance, rather than through independent effects on audience voting.

Celebrity characteristics such as age and industry background affect judges and audiences in systematically different ways. Judges prioritize technical consistency and relative ranking, while audiences respond to demographic cues and narrative appeal.

Notably, variables such as age and weekly ranking exert **opposite directional effects** across the two decision channels, reinforcing the necessity of modeling judge evaluation and audience voting separately.

6 The "Redemption & Resilience" Model

To address the historical imbalances between judge expertise and fan popularity, we propose the Balanced-Wildcard System (BWS). This framework maximizes audience engagement through a novel "Redemption" mechanism while safeguarding technical integrity via a Dynamic Weighting Mechanism.

6.1 System Architecture and Mechanics

The BWS moves away from static linear addition, introducing a multi-stage process:

- **Dynamic Resilience Weighting:** Instead of fixed 50/50 splits, the weighting factor λ_w adjusts based on the dispersion of judge scores. If judges' scores are uniform (low variance), fan votes receive higher weight to break the tie. If judge scores show high variance (indicating clear technical disparity), the judges' weight increases to protect professional standards.
- **The "Fan Save" Redemption Round:** Mid-way through the season (typically Week 6), the show hosts a "Redemption Week." The top 3 previously eliminated contestants (based on cumulative judge scores) compete for a single "Wildcard" spot. This creates a high-stakes social media event, allowing technically gifted "underdogs" a second chance.
- **Judges' Bottom-Two Veto:** We formalize the Season 28 mechanism where the bottom two couples (by combined score) face a "Dance-Off." The judges cast the final vote to stay, acting as a "safety valve" against purely popularity-driven outcomes.

6.2 Mathematical Formulation

The final score $S_{i,w}$ for contestant i in week w is defined as:

$$S_{i,w} = (1 - \lambda_w) \cdot \Phi(J_{i,w}) + \lambda_w \cdot \Psi(V_{i,w}) \tag{30}$$

Variable Definitions:

- $S_{i,w}$: The final composite score for contestant i in week w.
- $\Phi(J_{i,w})$: The normalized judge score.
- $\Psi(V_{i,w})$: The normalized fan vote percentage.
- λ_w: The Engagement Coefficient, which determines the weight (influence) of fan votes.

6.2.1 Dynamic Weight Calculation

The Engagement Coefficient λ_w is calculated weekly:

$$\lambda_w = 0.4 + 0.3 \cdot \left(1 - \frac{\sigma_w - \sigma_{\text{min}}}{\sigma_{\text{max}} - \sigma_{\text{min}}}\right) \tag{31}$$

Where:

- σ_w: Standard deviation of judges' scores for week w.
- $\sigma_{\text{min}}, \sigma_{\text{max}}$: Predefined bounds (e.g., 1.0 and 3.0 on a 10-point scale).
- **Effect:** When judges agree (σ_w is low), $\lambda_w \to 0.7$ (favoring fans). When judges disagree (σ_w is high), $\lambda_w \to 0.4$ (favoring expertise).

6.2.2 Redemption Round Rules

During Redemption Week:

- λ_w is fixed at 0.8, granting fans primary control.
- The "saved" contestant returns with a 5% combined score bonus the following week to symbolize their "Resilience" momentum.

7 Memorandum

STRATEGIC PIVOT: SAFEGUARDING ARTISTIC INTEGRITY VIA ADAPTIVE ANALYTICS

TO: Executive Production Team, BBC Studios / ABC Entertainment

FROM: MCM Strategic Modeling Group (Team #2601356)

DATE: February 1, 2026

Dear Production Team,

As *Dancing with the Stars* evolves, the ``popularity-merit divergence'' poses a critical risk to the Mirrorball Trophy's prestige. Our team has engineered a data-driven framework using 34 seasons of competition data to harmonize professional standards with viewer engagement.

1. Risk Mitigation via Predictive Intelligence

We developed a **Wide & Deep Neural Framework** that identifies latent fan voting patterns with **96.1% accuracy**. This powers our **Integrity Risk Alert (IRA)** system, which quantifies the deviation between a contestant's technical growth and social momentum. By flagging ``at-risk'' high-performers before results are finalized, production can strategically utilize storytelling or the ``Judges' Save'' to prevent brand-damaging eliminations.

2. The JCI-Driven Dynamic Weighting

We propose replacing fixed 50/50 scoring with an **Adaptive Feedback System** based on a **Judge Consensus Index (JCI)**. This mechanism intelligently scales voting weights:

- **High JCI (Clear Technical Gaps):** The system prioritizes professional scores to protect technical integrity.
- **Low JCI (Technical Parity):** Fan votes become the primary decider, driving audience passion and narrative stakes.

3. Engineering the ``Underdog Effect''

Our **Balanced-Wildcard System** (Redemption Week) is a narrative engine backed by data. Identifying 'High-Resilience' contestants allows the BWS to cultivate compelling redemption narratives. This strategy is projected to significantly enhance audience retention and social engagement by rewarding technical growth over static popularity.

4. Strategic Implementation

We recommend a pilot implementation for the **Final 3**, utilizing a **Ranking-Based Consensus Model** to ensure the winner reflects both professional mastery and mass-market appeal. We are prepared to provide our full technical brief to assist in integrating these tools into your production workflow.

Sincerely,

Executive Lead, MCM Team #2601356

References

[1] ABC Entertainment. *Dancing with the Stars* - Official Database. URL: https://www.abc.com/shows/dancing-with-the-stars/ (visited on 02/01/2026).

[2] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. "Wide & Deep Learning for Recommender Systems." In: *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* (2016), pp. 7–10. DOI: https://doi.org/10.1145/2988450.2988454. arXiv: https://arxiv.org/abs/1606.07792.

[3] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." In: *Advances in Neural Information Processing Systems* 30 (2017), pp. 3149–3157. URL: https://dl.acm.org/doi/10.5555/3294996.3295074.

[4] Christopher J. C. Burges. "From RankNet to LambdaRank to LambdaMART: An Overview." *Microsoft Research Technical Report*. MSR-TR-2010-82 (2010). URL: https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview/.

[5] Microsoft LightGBM Development Team. *LightGBM - Ranking Examples*. GitHub. URL: https://github.com/microsoft/LightGBM/tree/master/examples/lambdarank (visited on 02/01/2026).

[6] Wikipedia Contributors. "Softmax function." In: *Wikipedia, The Free Encyclopedia*. URL: https://en.wikipedia.org/wiki/Softmax_function (visited on 02/01/2026).

[7] Wikimedia Foundation. *MediaWiki API Documentation*. URL: https://en.wikipedia.org/w/api.php (visited on 02/01/2026).

[8] Halil Bariş. *Map-of-Wiki: Network Visualization Tool for Wikipedia Link Analysis*. GitHub Repository. URL: https://github.com/HalilB84/Map-of-Wiki/ (visited on 02/01/2026).

Report on Use of AI

General Statement We used AI tools (ChatGPT/Claude/Gemini) for the following purposes:

- **Language editing and proofreading:** We used AI to polish our English writing, correct grammatical errors, and improve clarity of expression.
- **Translation assistance:** Some of our initial brainstorming and draft materials were written in our native language and translated to English with AI assistance.
- **Code debugging:** AI was used to debug Python/MATLAB® code for data processing and visualization.

All mathematical modeling, data analysis, model design, and substantive content are original work by our team. AI tools were used solely as writing assistants, not for generating core ideas or solutions.

1. Gemini (Google, Gemini 2.5 Flash, February 2026 version)

Query 1: <Generate a high-quality academic flowchart for a data science project. The diagram is organized into 4 vertical sections. Phase 1 (Predictive): shows "Historical Data" leading to a "Wide & Deep Neural Network" node for "Latent Preferences". Phase 2 (Vote Decomp): features a central blue diamond labeled "Softmax Residual Analysis" connecting to "Latent Support". Phase 3 (Statistical Modeling): displays a central circular hub for "Seasonal Trends" and nodes for "Voting Methods Theory". Phase 4 (Optimization): highlights "Redemption & Resilience" and "Engagement Tuning Lambda w". Style: Professional flat vector, clean typography, 16:9 aspect ratio, soft corporate colors, white background, 4k resolution.>

Output: Gemini generated an HTML document that produces a comprehensive four-phase flowchart visualization. The document includes embedded SVG graphics with professional styling, responsive layout, and color-coded sections for each phase. The visualization clearly maps the project's progression from predictive modeling through optimization.

This visualization was converted and integrated as Figure 1 in the Introduction section, providing readers with a clear roadmap of the paper's analytical framework and methodology progression.

I hope this helps!

2. Claude AI (Anthropic, Sonnet 4.5, January 2026 version)

Query 2: <Write Python code to implement LightGBM with LambdaRank objective for ranking-based prediction tasks.>

Output: Claude provided complete implementation code including data preprocessing, feature engineering, LightGBM model configuration with `objective=`lambdarank`, and evaluation metrics. Here is a simplified version:

```
import lightgbm as lgb
from sklearn.model_selection import train_test_split

params = {
    'objective': 'lambdarank',
    'metric': 'ndcg',
    'boosting_type': 'gbdt',
    'num_leaves': 31,
    'learning_rate': 0.05,
    'feature_fraction': 0.9
}

train_data = lgb.Dataset(X_train, label=y_train, group=group_train)
valid_data = lgb.Dataset(X_valid, label=y_valid, group=group_valid)

model = lgb.train(params, train_data, valid_sets=[valid_data],
                  num_boost_round=100, early_stopping_rounds=10)
```

This code formed the foundation of our LightGBM baseline model in Section 3.8.

I hope this helps!

3. Claude AI (Anthropic, Sonnet 4.5, January 2026 version)

Query 3: <How do I perform Wikipedia network analysis using API calls and community detection algorithms?>

Output: Claude explained how to use the Wikipedia API (`https://en.wikipedia.org/w/api.php`) to extract hyperlink relationships between articles, construct a network graph, and apply community detection algorithms like Leiden or Louvain. It provided guidance on using libraries such as `networkx` and `python-igraph` for network visualization.

This methodology was applied in Section 5.3.4 to create the celebrity network visualization (Figure 10), analyzing how industry backgrounds correlate with audience voting patterns.

I hope this helps!

4. Claude AI (Anthropic, Sonnet 4.5, February 2026 version)

Query 4: <Find an open-source alternative for Wikipedia visualization tools comparable to sponsored projects. Specifically, recommend tools for network analysis and visualization of Wikipedia hyperlink relationships.>

Output: Claude recommended HalilB84's Map-of-Wiki project as an excellent open-source alternative for Wikipedia network visualization. This tool provides comprehensive hyperlink extraction, network graph construction, and community detection visualization without requiring sponsorship or proprietary software.

This recommendation led to the adoption of the Map-of-Wiki tool for our celebrity network analysis, as documented in Section 5.3.4 and Figure 10, providing reproducible and publicly accessible visualization capabilities for the research.

I hope this helps!