

# **Sviluppo e Architettura di una Piattaforma Distribuita per la Valutazione Sensoriale Collaborativa: Un Approccio Moderno Basato sulla Teoria della Scelta Sociale**

L'evoluzione delle tecnologie web negli ultimi anni ha reso possibile la creazione di applicazioni altamente sofisticate, in tempo reale e completamente gratuite, ideali per scopi ludici o professionali a breve termine. Il progetto di una piattaforma dedicata alla degustazione comparativa di prodotti, come nel caso specifico di una serata dedicata alle pizze surgelate tra amici, rappresenta un caso d'uso eccellente per esplorare la convergenza tra lo sviluppo software "serverless", la teoria della scelta sociale e l'analisi statistica dei dati qualitativi. L'obiettivo non è semplicemente raccogliere voti, ma trasformare un'esperienza soggettiva in un risultato collettivo matematicamente rigoroso, visualizzando al contempo come diversi sistemi elettorali possano influenzare l'esito finale. Grazie alla disponibilità di infrastrutture Backend-as-a-Service (BaaS) come Supabase e piattaforme di hosting come Vercel, è oggi possibile implementare un sistema completo di lobby, votazione multi-fattore e algoritmi complessi come il metodo Schulze o il sistema Borda, senza incorrere in costi operativi.

## **Fondamenti Tecnologici e Infrastruttura Zero-Cost**

Per realizzare una webapp veloce, accessibile da qualsiasi browser e priva di costi, l'architettura deve basarsi su un modello di "disaccoppiamento" tra frontend e backend. L'approccio moderno prevede l'utilizzo di una Single Page Application (SPA) o di un framework come Next.js, ospitato su servizi che offrono generosi piani gratuiti per piccoli progetti.

### **La Scelta del Backend: Supabase e il Potere di PostgreSQL**

Supabase si configura come l'alternativa open-source più adatta rispetto a sistemi come Firebase, principalmente per la sua base nativa in PostgreSQL. Per un progetto che richiede calcoli complessi e relazioni tra dati (voti, candidati, criteri, utenti), un database relazionale è superiore a un database NoSQL per la sua capacità di eseguire query strutturate e mantenere l'integrità dei dati. Il piano gratuito di Supabase nel 2025 offre limiti estremamente elevati per un progetto di questo tipo, tra cui 50,000 utenti attivi mensili e 500 MB di spazio su database, che permettono di gestire migliaia di sessioni di degustazione senza alcuna spesa.

Un aspetto fondamentale per la semplicità d'uso è l'autenticazione anonima. Gli utenti non devono essere costretti a creare un account con email e password per una serata singola; Supabase permette di generare sessioni anonime istantanee che assegnano un ID univoco a ogni partecipante, garantendo che ogni persona voti una sola volta e possa modificare i propri voti durante la sessione.

## Hosting e Deployment: Vercel e la Edge Runtime

L'interfaccia utente deve essere reattiva e caricarsi istantaneamente. Vercel rappresenta lo standard industriale per il deployment di applicazioni web moderne. Il suo piano "Hobby" include certificati SSL gratuiti, una rete di distribuzione dei contenuti (CDN) globale e l'esecuzione di funzioni serverless che possono gestire la logica di calcolo degli algoritmi elettorali più onerosi, come il metodo di Schulze. L'integrazione di Vercel con Supabase consente di scalare il progetto da una piccola cerchia di amici a un pubblico potenzialmente illimitato, mantenendo la latenza vicina allo zero grazie alle Edge Functions.

Caratteristica	Supabase (Free Tier)	Vercel (Free Tier)
Database	500 MB PostgreSQL	N/A
Autenticazione	50,000 MAU (Inclusi Anonimi)	N/A
Real-time	Broadcast e Presence inclusi	N/A
Banda / Egress	5 GB al mese	100 GB al mese
Funzioni	500,000 invocazioni Edge	Innumerevoli (limiti su durata)

## Architettura del Sistema di Lobby e Sincronizzazione in Tempo Reale

La creazione di una "lobby" o stanza virtuale è il cuore dell'esperienza utente. Il creatore della serata deve poter avviare una sessione e invitare i partecipanti in modo fluido. Questo viene realizzato tramite Supabase Realtime, un servizio basato su WebSocket che permette la comunicazione bidirezionale tra i client senza dover ricaricare la pagina.

## Gestione delle Stanze e Inviti tramite QR Code

Ogni lobby è identificata da un codice univoco o un link di invito. Per massimizzare la velocità di accesso durante una serata fisica, l'applicazione può generare automaticamente un QR Code utilizzando librerie come react-qr-code. La scansione del codice reindirizza i partecipanti direttamente all'interfaccia di voto, dove il sistema di autenticazione anonima crea silenziosamente una sessione nel background.

Il sistema di "Presence" di Supabase permette di visualizzare in tempo reale chi è entrato nella stanza e chi è pronto a votare. Il creatore può agire come "master" della sessione, decidendo quando aprire o chiudere le votazioni per un particolare candidato, o permettendo una votazione asincrona dove ognuno procede al proprio ritmo.

## Sincronia vs Asincronia: Flessibilità Operativa

L'utente richiede la possibilità di gestire la votazione in modo sincrono o asincrono.

- In modalità **sincrona**, il server emette un evento di "start" per ogni pizza assaggiata; tutti i partecipanti visualizzano lo stesso modulo di voto contemporaneamente. Questo è utile per mantenere il gruppo allineato durante l'assaggio.
- In modalità **asincrona**, il link rimane attivo per un tempo prestabilito, permettendo agli utenti di votare quando preferiscono, magari anche a ritroso se il creatore abilita questa opzione nelle impostazioni della lobby.

## Configurazione della Votazione: Candidati, Fattori e

## Pesi

Una delle richieste centrali riguarda la complessità della configurazione. Non si tratta di un semplice poll "sì/no", ma di un sistema multi-criterio. Nel contesto delle pizze surgelate, i candidati sono le diverse marche (ad esempio Buitoni, Italpizza, Roncadin, Cameo), mentre i fattori possono essere molteplici e qualitativamente diversi.

## Definizione dei Criteri di Assaggio

L'analisi dei video di degustazione suggerisce che una valutazione accurata della pizza debba considerare diversi attributi :

1. **Qualità della Crosta (Cornicione)**: Se è croccante, morbida, o "anemica".
2. **Qualità del Topping**: La sapidità della mozzarella, l'acidità del pomodoro e la presenza di origano o basilico.
3. **Cottura**: Se l'impasto risulta crudo al centro o ben alveolato.
4. **Rapporto Qualità-Prezzo**: Un fattore spesso determinante per i prodotti da supermercato.

Il creatore della lobby può definire il peso relativo di ogni fattore. Ad esempio, la "Qualità del Topping" potrebbe pesare per il 40% del voto finale, mentre la "Cottura" per il 20%. Il calcolo finale del voto di un candidato C per un elettore V sarà quindi una media pesata:  
Dove n è il numero di fattori e la somma dei pesi è normalizzata a 1.

## La Scala di Voto e la Personalizzazione del Creatore

Il sistema deve supportare diverse scale, dal classico 1-10 a scale più semplici (1-5) o qualitative ("Ottimo", "Buono", "Sufficiente", "Scarso"). Questa flessibilità è fondamentale per adattare lo strumento a diversi tipi di prodotti, non limitandosi alle pizze. Il database PostgreSQL gestisce queste configurazioni tramite campi JSONB, permettendo di memorizzare strutture di criteri arbitrarie per ogni singola lobby.

## Teoria della Scelta Sociale e Algoritmi di Elezione

La caratteristica più distintiva richiesta è l'implementazione di molteplici sistemi elettorali per mostrare come la scelta del vincitore possa cambiare a seconda delle regole applicate. Come discusso nel video di Veritasium, la democrazia è matematicamente complessa e ogni sistema porta con sé dei compromessi intrinseci. Il Teorema di Impossibilità di Arrow dimostra che nessun sistema di voto basato su ranghi (ordinali) può soddisfare contemporaneamente tutti i criteri di equità quando ci sono tre o più opzioni.

## Pluralità e Maggioranza Semplice

Il sistema più comune è la **Pluralità** (First-Past-The-Post), dove vince chi ottiene più primi posti. Tuttavia, questo metodo soffre pesantemente dell'effetto "spoiler": due pizze simili potrebbero dividersi i voti della maggioranza, permettendo a una terza pizza meno amata di vincere. Il report genererà questa visualizzazione come base di confronto.

## Il Metodo Borda: La Ricerca del Consenso

Il **Conteggio di Borda** assegna punti in base alla posizione in classifica: in un'elezione con N pizze, la prima scelta riceve N punti, la seconda N-1, e così via fino a 1 punto per l'ultima. Questo metodo è eccellente per identificare un vincitore di "consenso", ovvero la pizza che piace a tutti mediamente, anche se non è la preferita assoluta di nessuno. È ampiamente utilizzato in contesti sportivi come l'assegnazione del MVP nel baseball o dell'Heisman Trophy.

## Il Vincitore di Condorcet e il Metodo Schulze

Un **Vincitore di Condorcet** è un candidato che batterebbe ogni altro candidato in una sfida testa a testa. Tuttavia, può verificarsi il "Paradosso di Condorcet", dove le preferenze collettive sono cicliche (A batte B, B batte C, C batte A).

Per risolvere questi cicli, il **Metodo Schulze** (o Beatpath) analizza la forza dei percorsi di vittoria tra i candidati.

1. Si calcola la matrice delle preferenze binari (quanti preferiscono la pizza A alla B).
2. Si identificano i "percorsi" di vittorie indirette.
3. La forza di un percorso è definita dal suo "anello più debole" (la vittoria con il margine minore nel percorso).
4. Vince chi ha il percorso più forte verso tutti gli altri.

Questo algoritmo è robusto contro l'aggiunta di "cloni" e garantisce che, se esiste un vincitore di Condorcet, esso venga eletto. L'implementazione in JavaScript può essere facilitata da librerie come votes o caritat.

## Voto Instantaneo con Ballottaggio (IRV)

L'**Instant-Runoff Voting** simula una serie di ballottaggi: se nessuna pizza ha la maggioranza assoluta dei primi posti, quella con meno voti viene eliminata e i suoi voti ridistribuiti alle seconde scelte, finché non emerge un vincitore. Questo metodo è molto popolare nelle elezioni politiche moderne (es. Alaska, New York City) perché riduce il timore del voto utile.

Sistema	Tipo	Caratteristica Principale	Difetto Comune
Pluralità	Nominale	Semplicità estrema	Vulnerabile agli spoiler
Borda	Posizionale	Premia il consenso	Manipolabile tatticamente
Condorcet	Binario	Vince chi batte tutti i rivali	Non sempre esiste un vincitore
Schulze	Grafo	Risolve i cicli di Condorcet	Complessità di calcolo
IRV	Eliminazione	Riduce la dispersione dei voti	Non sempre Condorcet-coerente

## Normalizzazione dei Voti: Bilanciare Generosità e Rigore

Uno dei problemi fondamentali delle valutazioni soggettive è la differenza tra i profili degli elettori. Alcuni amici tendono a dare voti alti a tutto (elettori "generosi"), mentre altri sono estremamente critici (elettori "severi"). Senza normalizzazione, il voto di chi usa solo la parte

alta della scala peserà molto di più sulla media rispetto a chi è più morigerato.

## Normalizzazione Min-Max

La normalizzazione Min-Max riscalza i voti di ogni singolo utente nell'intervallo  $\$$ . Se l'utente A ha dato voti tra 6 e 8, il 6 diventerà 0 e l'8 diventerà 1. Se l'utente B ha dato voti tra 2 e 10, il 2 diventerà 0 e il 10 diventerà 1.

Questo approccio è utile per dare a tutti lo stesso "potere di voto" indipendentemente dalla scala utilizzata, ma è molto sensibile ai valori anomali (outliers).

## Z-Score (Standardizzazione)

Lo Z-score è un metodo più robusto che tiene conto della variabilità individuale. Trasforma i voti in modo che la media di ogni utente sia 0 e la deviazione standard sia 1.

Dove  $\mu$  è la media dei voti dell'utente e  $\sigma$  è la deviazione standard. Se un utente dà sempre 7 e una volta 9, quel 9 avrà uno Z-score molto alto perché è un'eccezione rispetto al suo comportamento solito. Se un altro utente dà voti casuali tra 1 e 10, un 9 avrà uno Z-score più basso. Questo metodo appiana le differenze qualitative di generosità, rendendo il confronto tra votanti estremamente equo.

## Visualizzazione dei Dati e Statistiche Avanzate

Il successo dell'applicazione dipende dalla capacità di presentare i risultati in modo coinvolgente. La visualizzazione non deve limitarsi a una classifica, ma deve offrire spunti di conversazione. Chart.js permette di creare grafici a barre, a torta o radar (molto utili per i criteri multi-fattore).

## Dashboard dei Risultati Elettorali

Il sistema mostrerà una classifica dinamica che cambia istantaneamente quando l'utente seleziona un algoritmo diverso. Questo permette di vedere, ad esempio, che la pizza "Roncadin" vince con il metodo Borda (perché è piaciuta a tutti abbastanza) ma perde nella Pluralità contro la "Buitoni" (che ha avuto pochi fan sfegatati ma molti detrattori).

## Analisi del Comportamento degli Elettori

Per rendere la serata più divertente, il sistema può calcolare statistiche sugli amici (se i voti sono pubblici):

- **Il Critico Spietato:** L'utente con la media voti più bassa.
- **L'Ottimista:** L'utente con la media più alta.
- **L'Incoerente:** L'utente con la deviazione standard più alta (voti molto sparsi).
- **Il Bastancontrario:** Chi ha votato in modo più discorde rispetto alla media del gruppo (calcolato tramite la correlazione dei ranghi).

## Visualizzazione Multi-Criterio (Radar Chart)

Per ogni pizza, è utile mostrare un grafico radar che evidenzi i suoi punti di forza. Una pizza

potrebbe eccellere nella "Crosta" ma fallire nel "Topping". Questo aiuta a identificare il candidato con un "fattore più marcato" come richiesto dall'utente.

## Funzionalità Aggiuntive e Gamification

Oltre alle funzionalità core, diverse aggiunte possono migliorare l'esperienza e la robustezza del sistema.

### Il Voto Bonus (Jolly)

Come suggerito dall'utente, l'aggiunta di un "Voto Bonus" permette di iniettare una componente cardinale pura in un sistema ordinale. Ogni utente ha un singolo "Jolly" che assegna un punteggio extra fisso a una pizza a scelta. Questo permette di premiare quella pizza che, pur avendo qualche difetto tecnico (es. crosta bruciata), ha regalato un'emozione superiore.

### Modifica a Ritroso e Storico

Il creatore può decidere se consentire agli utenti di cambiare i voti dopo aver assaggiato le pizze successive. Spesso, l'ultima pizza assaggiata cambia la percezione della prima. Implementare una gestione dello stato che permetta l'editing "a ritroso" è semplice grazie alla funzione upsert di Supabase, che aggiorna il record se esiste o lo crea se manca.

### Commenti e Foto in Tempo Reale

Permettere agli utenti di caricare una foto della pizza appena sfornata o di lasciare un commento ("sembra un pezzo di cartone" o "mozzarella di plastica" ) aggiunge una dimensione social che rende la serata più memorabile. Le foto possono essere memorizzate in Supabase Storage, che offre 1 GB gratuito.

## Guida alla Realizzazione Tecnica

Per un progetto veloce e semplice, si consiglia di utilizzare un'architettura moderna basata su JavaScript.

### Passaggio 1: Setup del Database

Utilizzando l'editor SQL di Supabase, si definiscono le tabelle fondamentali. È essenziale abilitare il "Realtime" sulle tabelle dei voti per permettere l'aggiornamento istantaneo dei grafici su tutti i telefoni degli amici.

```
-- Esempio di struttura tabelle
CREATE TABLE lobbies (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    code TEXT UNIQUE, -- Codice 4 cifre per accesso rapido
    settings JSONB, -- Pesi dei fattori, algoritmi abilitati
    status TEXT        -- 'waiting', 'voting', 'finished'
);
```

```

CREATE TABLE candidates (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    lobby_id REFERENCES lobbies(id),
    name TEXT,
    description TEXT
) ;

CREATE TABLE votes (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    voter_id UUID,          -- ID anonimo di Supabase Auth
    candidate_id REFERENCES candidates(id),
    scores JSONB,           -- Voti per ogni fattore { "crosta": 8,
    "topping": 7 }
    bonus_applied BOOLEAN DEFAULT false
) ;

```

## Passaggio 2: Logica del Frontend

Si consiglia l'uso di React con Vite per la massima velocità di sviluppo. La gestione della lobby può essere fatta tramite gli "Hooks" di Supabase:

1. supabase.auth.signInAnonymously() per l'accesso senza frizioni.
2. supabase.channel('lobby\_channel').on('presence',...) per vedere chi è online.
3. supabase.channel('db\_changes').on('postgres\_changes',...) per aggiornare i grafici appena qualcuno vota.

## Passaggio 3: Calcolo degli Algoritmi

La logica elettorale può risiedere interamente nel frontend per semplicità, oppure essere spostata in una "Supabase Edge Function" per garantire che nessuno possa manipolare i risultati manipolando il codice locale. Librerie come schulze-vote (Ruby, ma convertibile) o implementazioni JS del Floyd-Warshall possono calcolare i percorsi più forti in pochi millisecondi anche per decine di candidati.

## Fattibilità, Espandibilità e Considerazioni Finali

Il progetto non solo è fattibile, ma è un esercizio ideale di ingegneria del software moderna. L'utilizzo di strumenti allo stato dell'arte come Supabase e Vercel garantisce che il sito sia:

- **Completamente Gratuito:** Rimanendo nei limiti dei piani hobby, non ci sono costi di server o database.
- **Scalabile:** Se il link venisse condiviso su un social network e migliaia di persone volessero votare la loro pizza surgelata preferita, l'infrastruttura serverless gestirebbe il carico senza problemi.
- **Condivisibile:** L'accesso tramite QR code e l'assenza di download rendono la barriera all'ingresso praticamente nulla.

## Cosa si sta ignorando?

Un elemento che potrebbe semplificare ulteriormente la realizzazione è l'uso di "No-Code" o "Low-Code" tools come FlutterFlow o Glide se non si ha dimestichezza con la programmazione pura, sebbene questi possano avere limitazioni sugli algoritmi di calcolo complessi come Schulze. Tuttavia, per un controllo totale sulla visualizzazione dei dati e sulla normalizzazione statistica (Z-score), la programmazione manuale con React/Next.js rimane la scelta superiore. In conclusione, la creazione di questo strumento trasforma una semplice serata tra amici in un laboratorio di analisi sensoriale e democrazia digitale. La capacità di vedere in tempo reale come cambiano i pesi dei fattori o come l'algoritmo di Borda possa ribaltare un risultato di Pluralità offre una comprensione profonda della natura soggettiva della qualità e della complessità della scelta collettiva. La flessibilità del sistema garantisce che, dopo le pizze, il gruppo possa utilizzare lo stesso strumento per votare il miglior film della serata, la prossima meta di viaggio o qualsiasi altra decisione che richieda un consenso informato e matematicamente bilanciato.

## Bibliografia

1. Supabase | The Postgres Development Platform., <https://supabase.com/>
2. Supabase Pricing in 2025: Full Breakdown of Plans | UI Bakery Blog, <https://uibakery.io/blog/supabase-pricing>
3. Deploy a Scalable Backend for Free in 2025 (Supabase + Vercel Edge) - Medium, <https://medium.com/@atnoforwebdev/deploy-a-scalable-backend-for-free-in-2025-supabase-vercel-edge-9ea05e9559f4>
4. Top 9 React JS Frameworks Developers Are Using in 2025 | by Brilworks Software | Medium, <https://medium.com/@Brilworks/top-9-react-js-frameworks-developers-are-using-in-2025-8ae52efd24a1>
5. Top 15+ Best Database List for Web Application Development in 2025 - FAB Builder, <https://fabbuilder.com/blogs/top-databases-list-for-web-apps/>
6. Tables and Data | Supabase Docs, <https://supabase.com/docs/guides/database/tables>
7. Pricing & Fees - Supabase, <https://supabase.com/pricing>
8. Supabase Pricing 2025: Free, Pro & Enterprise Costs | MetaCTO, <https://www.metacto.com/blogs/the-true-cost-of-supabase-a-comprehensive-guide-to-pricing-integration-and-maintenance>
9. Anonymous Sign-Ins | Supabase Docs, <https://supabase.com/docs/guides/auth/auth-anonymous>
10. Supabase Auth now supports Anonymous Sign-ins, <https://supabase.com/blog/anonymous-sign-ins>
11. Computing the Schulze Method for Large-Scale Preference Data Sets - arXiv, <https://arxiv.org/pdf/2505.12976.pdf>
12. Getting Started with Realtime | Supabase Docs, [https://supabase.com/docs/guides realtime/getting\\_started](https://supabase.com/docs/guides realtime/getting_started)
13. Realtime | Supabase Docs, <https://supabase.com/docs/guides realtime>
14. qrcode.react vs qr.js vs react-qr-code vs qr-code-styling vs qrious - NPM Compare, <https://npm-compare.com/qr-code-styling,qr.js,qrcode.react,qrious,react-qr-code>
15. How to generate QR-Code using 'react-qr-code' in ReactJS ? - GeeksforGeeks, <https://www.geeksforgeeks.org/reactjs/how-to-generate-qr-code-using-react-qr-code-in-reactjs/>
16. SupaBase Anonymous Login - FlutterFlow Community, <https://community.flutterflow.io/community-tutorials/post/supabase-anonymous-login-2iWSInwd5uLU5f3>
17. Realtime - Presence | Supabase Features, <https://supabase.com/features/realtime-presence>
18. Presence | Supabase Docs, <https://supabase.com/docs/guides realtime/presence>
19. Exploring Supabase Realtime By

Building a Game - aleksandra.codes, <https://www.aleksandra.codes/supabase-game> 20.

Enhancing a Voting System Using JavaScript | CodeSignal Learn, <https://codesignal.com/learn/courses/interview-practice-advanced-problem-solving-1/lessons/enhancing-a-voting-system-using-javascript> 21. Le PIZZE SURGELATE in AMERICA sono buone? - YouTube, <https://www.youtube.com/watch?v=MIU63eWekZA> 22. Let's test the FROZEN PIZZAS WITH SEASONING! - YouTube, <https://www.youtube.com/watch?v=znn0xBFO-AQ> 23.

Qual è la migliore PIZZA CONGELATA? - YouTube, [https://www.youtube.com/watch?v=k-Skekc\\_ZTw](https://www.youtube.com/watch?v=k-Skekc_ZTw) 24. PIZZE SURGELATE: PROMOSSE O BOCCIADE? - EP. 3 - YouTube, [https://www.youtube.com/watch?v=gpAZAS4mt\\_A](https://www.youtube.com/watch?v=gpAZAS4mt_A) 25. The Best Frozen Pizza - YouTube, <https://www.youtube.com/watch?v=bHlulYoOD9E> 26. Why Democracy Is Mathematically Impossible - YouTube, <https://www.youtube.com/watch?v=qf7ws2DF-zk> 27.

How to merge a collection of ordered preferences - Stack Overflow, <https://stackoverflow.com/questions/6352212/how-to-merge-a-collection-of-ordered-preferences>

28. Social choice theory - Wikipedia, [https://en.wikipedia.org/wiki/Social\\_choice\\_theory](https://en.wikipedia.org/wiki/Social_choice_theory) 29.

Borda Count Method: Basics and an Example - Toolshero, <https://www.toolshero.com/decision-making/borda-count-method/> 30. Comparing single-winner voting methods - FairVote, <https://fairvote.org/resources/electoral-systems/comparing-voting-methods/> 31. Voting Theory: Fresh Take – Quantitative Reasoning - Lumen One Content, <https://content.one.lumenlearning.com/quantitativeresoning/chapter/voting-theory-fresh-take/>

32. Borda Count | Mathematics for the Liberal Arts - Lumen Learning, <https://courses.lumenlearning.com/waymakermath4libarts/chapter/borda-count/> 33. Lecture 2 : Borda's method: A Scoring System - University of Notre Dame, <https://www3.nd.edu/~apilking/math10170/information/Lectures/Lecture-2.Borda%20Method.pdf>

34. VOTING METHODS Borda's Method: (1) Voters rank the entire list of candidates from first choice to last choice. (2), [https://www2.math.uconn.edu/~gageonea/math1030f13/1-2\\_notes.pdf](https://www2.math.uconn.edu/~gageonea/math1030f13/1-2_notes.pdf)

35. Borda Count | MGF 1107 - Lumen Learning, <https://courses.lumenlearning.com/nwfsc-MGF1107/chapter/borda-count/> 36. 7.2 Alternative Voting Methods, <https://mathbooks.unl.edu/Contemporary/sec-7-2-alternatives.html> 37.

Condorcet method - Wikipedia, [https://en.wikipedia.org/wiki/Condorcet\\_method](https://en.wikipedia.org/wiki/Condorcet_method) 38. BORDA VS CO NDORCET, <https://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/17554/Borda%20v%20Condorcet%20FINAL%20-2.pdf?sequence=1&isAllowed=y> 39. Condorcet Voting - Center for Effective Government, <https://effectivegov.uchicago.edu/primers/condorcet-voting> 40. Schulze method - Wikipedia, [https://en.wikipedia.org/wiki/Schulze\\_method](https://en.wikipedia.org/wiki/Schulze_method) 41. Condorcet/Schulze voting in single-winner districts - House of Commons, <https://www.ourcommons.ca/content/Committee/421/ERRE/Brief/BR8397842/br-external/SchulzeMarkus-e.pdf> 42. Fine-Grained Complexity and Algorithms for the Schulze Voting Method - Computational Social Choice, <https://comsoc-community.org/assets/proceedings/comsoc-2023/YinzhanXu.pdf> 43. anarchodin/caritat: A vote counting library in JavaScript - GitHub, <https://github.com/anarchodin/caritat> 44. lzeard/votes: JS library for ranked voting systems - GitHub, <https://github.com/lzeard/votes> 45. eukota/google\_script\_voting: Instant-runoff voting for Google Apps spreadsheets. - GitHub, [https://github.com/eukota/google\\_script\\_voting](https://github.com/eukota/google_script_voting) 46.

Ranked Choice Voting calculator for Google Classroom, Google Forms and Google Sheets. - RCV123.org, <https://www.rcv123.org/google-forms-ranked-choice-spreadsheet-calculator> 47.

How to Calculate Ranked-Choice Voting with Google Forms and Google Sheets, <https://www.rankedvote.co/guides/applying-ranked-choice-voting/how-to-calculate-ranked-choic>

e-voting-with-google-forms-and-google-sheets 48. Min-Max and Z-Score Normalization | Codecademy, <https://www.codecademy.com/article/min-max-zscore-normalization> 49. Difference between Z-Score Scaling and Min-Max Scaling | Kaggle, <https://www.kaggle.com/discussions/general/527563> 50. Z-Score Normalization: Definition and Examples - GeeksforGeeks, <https://www.geeksforgeeks.org/data-analysis/z-score-normalization-definition-and-examples/> 51. Statistics Day 4: Z-Score vs Min-Max Normalization — Making Data Fair for ML Models, [https://dev.to/brains\\_behind\\_bots/statistics-day-4-z-score-vs-min-max-normalization-making-data-fair-for-ml-models-1plc](https://dev.to/brains_behind_bots/statistics-day-4-z-score-vs-min-max-normalization-making-data-fair-for-ml-models-1plc) 52. What are the best normalization methods (Z-Score, Min-Max, etc.)? How would you choose a data normalization method? | ResearchGate, [https://www.researchgate.net/post/What\\_are\\_the\\_best\\_normalization\\_methods\\_Z-Score\\_Min-Max\\_etc\\_How\\_would\\_you\\_choose\\_a\\_data\\_normalization\\_method](https://www.researchgate.net/post/What_are_the_best_normalization_methods_Z-Score_Min-Max_etc_How_would_you_choose_a_data_normalization_method) 53. Step-by-step guide - Chart.js, <https://www.chartjs.org/docs/latest/getting-started/usage.html> 54. Mixed Chart Types - Chart.js, <https://www.chartjs.org/docs/latest/charts/mixed.html> 55. Multi-Criteria Decision Analysis (MCDA/MCDM) - 1000minds, <https://www.1000minds.com/decision-making/what-is-mcdm-mcda> 56. Development of Multi-Attribute Utility Theory Methods in Dynamic Decision Models Using Change-Data Driven, [https://www.tj.kyushu-u.ac.jp/evergreen/contents/EG2024-11\\_4\\_content/pdf/p3279-3289.pdf](https://www.tj.kyushu-u.ac.jp/evergreen/contents/EG2024-11_4_content/pdf/p3279-3289.pdf) 57. Borda Count Method | Definition, Examples & Uses - Lesson - Study.com, <https://study.com/academy/lesson/the-borda-count-method-in-elections.html> 58. cfrankston728/symmetric\_quantile-normalized\_score: Implementation of symmetric quantile-normalized score voting. - GitHub, [https://github.com/cfrankston728/symmetric\\_quantile-normalized\\_score](https://github.com/cfrankston728/symmetric_quantile-normalized_score) 59. JavaScript: Insert data | Supabase Docs, <https://supabase.com/docs/reference/javascript/insert> 60. Realtime Concepts | Supabase Docs, <https://supabase.com/docs/guides/realtime/concepts> 61. DIY Real-Time Polling App Locks Down Access With Supabase and Permit.io - Hackernoon, <https://hackernoon.com/diy-real-time-polling-app-locks-down-access-with-supabase-and-permitio> 62. Querying Joins and Nested tables | Supabase Docs, <https://supabase.com/docs/guides/database/joins-and-nesting> 63. coorasse/schulze-vote: Vote calculation with Schulze method (Condorcet voting) - GitHub, <https://github.com/coorasse/schulze-vote> 64. Supabase & Anonymous Auth - YouTube, <https://www.youtube.com/watch?v=OOVmxqq4a6Y> 65. Best 8 Database-based app builders in 2025 (Worldwide) - Boltic, <https://www.boltic.io/blog/simple-database-app-builders> 66. Guide to Supabase Anonymous Login in FlutterFlow, <https://www.flutterflowdevs.com/blog/supabase-anonymous-login>