

# Lorcana Deck Analyzer - Full Re-implementation and Enhancements

---

This document outlines the complete re-implementation of the Lorcana Deck Analyzer web application, incorporating all previously discussed features and enhancements. The application is designed to help users theorize the viability of specific Disney Lorcana decklists through statistical comparison and detailed card interaction analysis.



## Web Application URL

---

The application has been deployed to the following permanent URL:  
**<https://fvvwawjj.manus.space>**

**Note:** Due to persistent browser environment issues within the sandbox, I am unable to verify the functionality of the deployed application directly. It is highly recommended to test the application locally on your machine.



## Key Features Implemented

---

### 1. Core Deck Analysis

- **DataManager:** Efficiently loads and indexes all Lorcana card data from the Similcana Project's `allCards.json` file. Includes robust card lookup by name and full name.
- **DeckParser:** Parses raw decklist text into structured deck objects, performs validation (e.g., 60-card limit, 4-copy rule), and calculates basic deck statistics like total cards, colors, and ink curve.
- **StatisticalEngine:** Calculates key statistical metrics:
  - **Expected Lore per Turn:** Estimates the average lore generation potential of a deck.

- **Opening Hand Probabilities:** Calculates the probability of drawing specific card types (Characters, Actions, Items) in the opening hand using the Hypergeometric Distribution.
- **VisualizationEngine:** Renders interactive charts and tables using Chart.js:
  - Ink Curve charts for both player and opponent decks.
  - Opening Hand Probability charts.
  - Character vs. Character Matchup Matrix.

## 2. Advanced Ability Parsing and Interaction Analysis

- **InteractionAnalyzer:** Provides sophisticated character-vs-character matchup analysis, taking into account various card abilities and keywords:
  - **Keyword Recognition:** Explicitly handles `Evasive`, `Ward`, `Challenger`, `Resist`, `Rush`, `Bodyguard`, `Support`, and `Reckless`.
  - **Combat Resolution:** Simulates combat scenarios, calculating damage dealt and taken, and determining banish outcomes.
  - **Ability Impact:** Identifies and notes the presence and potential impact of relevant abilities during interactions.

## 3. Batch Mode Comparison

- **BatchAnalyzer:** Enables comparison of a player's deck against multiple meta-relevant decks.
  - **Meta Deck Management:** Allows users to manually add, name, and manage a list of meta decks. (Automated extraction from external websites like `inkdecks.com` or `heavenideas.github.io/utilities/personal_deck_saver.html` was not feasible due to sandbox browser limitations, requiring manual input of meta decklists).
  - **Multi-dimensional Scoring:** Provides an overall matchup score (0-100%) based on:
    - **Character Interaction Score:** How well player's characters trade with opponent's.
    - **Speed Score:** Based on average ink costs, indicating who can play threats faster.

- **Consistency Score:** Based on color count, penalizing decks with more than two colors.
- **Comprehensive Results Dashboard:** Displays summary statistics (total matchups, favorable/even/unfavorable counts, average score) and a detailed table of individual matchup scores.
- **CSV Export:** Allows users to download batch analysis results for external use.

## 4. Detailed Explanations for Users

- **Contextual Information:** The application now provides clear, in-depth explanations for all calculated metrics and analyses directly within the UI.
  - **Statistical Analysis:** Explanations for Expected Lore per Turn and Ink Curve.
  - **Interaction Analysis:** Detailed per-matchup breakdowns, including combat outcomes, reasons, ability interactions, and combat math.
  - **Probability Analysis:** Explanations for opening hand probabilities, referencing the Hypergeometric Distribution.



## How to Use the Web Application

---

1. **Access the Application:** Open the provided URL (`https://fvvwawjj.manus.space`) in a modern web browser.

2. **Input Your Deck:** In the

"Your Deck" textarea, enter your Lorcana decklist. Each line should be in the format: `[quantity] [Card Name]`, e.g., `4 Mickey Mouse - Brave Little Tailor`. 3. **Input Opponent Deck (Single Mode):** In the "Opponent Deck" textarea, enter an opponent's decklist in the same format. 4. **Analyze Matchup:** Click the "Analyze Matchup" button to view the detailed analysis across the "Overview", "Statistics", "Interactions", and "Probabilities" tabs. 5. **Batch Mode:** Click "Enable Batch Mode" to switch to batch comparison. You can then: - Click "Add Meta Deck" to manually input meta decklists (name and decklist). - Click "Run Batch Analysis" to compare your deck against all added meta decks. - View results in the "Batch Results" tab, including a summary and detailed matchup table. - Export results to CSV using the "Export Results (CSV)" button.

## Technical Implementation Details

---

The application is built using vanilla JavaScript, HTML5, and CSS3, with Chart.js for data visualization. The modular structure ensures maintainability and extensibility.

### File Structure:

```
lorcana-deck-analyzer/  
├── index.html  
├── css/  
│   └── styles.css  
├── js/  
│   ├── main.js  
│   ├── data-manager.js  
│   ├── deck-parser.js  
│   ├── statistical-engine.js  
│   ├── interaction-analyzer.js  
│   ├── visualization-engine.js  
│   └── batch-analyzer.js  
└── assets/  
    ├── images/ (for future use)  
    └── icons/ (for future use)
```

### Key JavaScript Modules:

- **data-manager.js** : Handles fetching and indexing of the `allCards.json` data from the Similcana Project. Provides efficient lookup for card details.
- **deck-parser.js** : Responsible for parsing raw decklist strings into structured JavaScript objects and performing deck validation based on Lorcana rules.
- **statistical-engine.js** : Contains logic for calculating statistical metrics such as expected lore per turn and hypergeometric probabilities for opening hands.
- **interaction-analyzer.js** : Implements the core logic for character-vs-character combat simulation, including parsing and applying effects of various keywords and abilities.
- **visualization-engine.js** : Utilizes Chart.js to render graphical representations of ink curves and probabilities, and dynamically generates the matchup matrix table.
- **batch-analyzer.js** : Manages the collection of meta decklists and orchestrates the batch comparison process, providing a comprehensive scoring system and result aggregation.

- `main.js` : The main application logic, coordinating interactions between all modules, handling UI events, and rendering results and explanations.

## ⚠ Known Limitations and Future Work

---

- **Browser Environment Stability:** The primary limitation encountered during development was the instability of the sandbox browser environment, which prevented direct verification of the deployed application and automated web scraping.
- **Automated Meta Deck Extraction:** Due to the above, meta decklists must be manually input. Future work could involve exploring alternative methods for automated data acquisition if a stable, accessible API or dataset becomes available.
- **Advanced Ability Parsing:** While significantly enhanced, the ability parsing can be further refined to handle more complex interactions, triggered abilities, and timing windows.
- **Comprehensive Statistical Models:** The statistical models are foundational. More advanced models could include turn-by-turn simulations (though explicitly excluded by user request for this version), mulligan strategies, and resource management optimization.
- **User Interface/User Experience (UI/UX):** Further UI/UX improvements could include drag-and-drop decklist input, visual card displays, and more interactive charts.

Despite the environmental challenges, the re-implemented Lorcana Deck Analyzer provides a robust framework for in-depth deck analysis, offering valuable insights for players looking to optimize their deckbuilding strategies.