# Superhero Attributes and Power Classification

A Data Mining Approach to Character Analysis

DSCI 4411 - Fundamentals of Data Mining
The American University in Cairo
Fall 2025

December 11, 2025

**Abstract**

This project applies classification and clustering techniques to analyze a dataset of 1,200 superhero and villain characters. We investigate whether machine learning models can distinguish between heroes and villains based on their attributes (powers, physical traits, behavioral metrics), and whether natural character archetypes emerge through unsupervised clustering. After testing 19 classification algorithms with extensive hyperparameter tuning, we find that all models plateau at approximately 65% accuracy, suggesting that the available features lack sufficient signal to reliably predict moral alignment. Clustering analysis reveals that characters naturally group by power level rather than hero/villain status, providing insights into the structure of superhero universes.

# Contents

# 1 Introduction

## 1.1 Problem Statement

The superhero genre has become one of the most prominent forms of modern storytelling, spanning comic books, films, television, and video games. Characters in these narratives are typically classified as either *heroes* (protagonists who protect society) or *villains* (antagonists who threaten it). This project addresses two fundamental questions:

1. **Classification Problem**: Can we predict whether a character is a hero or villain based solely on their measurable attributes—such as their superpowers, physical characteristics, and behavioral patterns?
2. **Clustering Problem**: Do natural groupings or "archetypes" exist among superhero characters that transcend the simple hero/villain binary?

## 1.2 Motivation

Understanding the patterns that differentiate heroes from villains has practical applications in:

- **Content Recommendation Systems**: Suggesting similar characters to readers/viewers based on attribute profiles
- **Character Design**: Informing writers and game designers about which attribute combinations are associated with heroic or villainous characters
- **Narrative Analysis**: Quantifying trends across fictional universes to understand storytelling conventions
- **Educational Value**: Demonstrating classification and clustering techniques on an engaging, accessible dataset

## 1.3 Dataset Overview

We utilize the **Kaggle Super-Heros Dataset**[1], which contains 1,200 character records with 17 features. The target variable is `is_good`, a binary indicator where 1 represents a hero and 0 represents a villain. The dataset exhibits a moderate class imbalance: 65% heroes (780 characters) and 35% villains (420 characters).

## 1.4 Report Structure

The remainder of this report is organized as follows: Section 2 describes the methodology, including data preprocessing, feature engineering, classification algorithms, and clustering approaches. Section 3 presents the experimental results with detailed analysis. Section 4 summarizes our findings and discusses limitations and future work.

[1] `https://www.kaggle.com/datasets/kenil1719/super-heros`

# 2 Methods

## 2.1 Data Description

The dataset contains three categories of features:

### 2.1.1 Physical Attributes (4 features)

Table 1: Physical attribute features

| Feature | Description | Range |
|---|---|---|
| `height_cm` | Height in centimeters | 150–250 |
| `weight_kg` | Weight in kilograms | 45–128 |
| `age` | Character age in years | 18–100+ |
| `years_active` | Years operating as hero/villain | 1–50 |

### 2.1.2 Behavioral Metrics (4 features)

Table 2: Behavioral metric features

| Feature | Description | Range |
|---|---|---|
| `power_level` | Overall power rating | 0–100 |
| `public_approval_rating` | Public perception score | 0–100 |
| `training_hours_per_week` | Weekly training intensity | 0–60 |
| `civilian_casualties_past_year` | Collateral damage count | 0–10 |

### 2.1.3 Superpower Flags (8 binary features)

Eight binary indicators representing the presence or absence of specific superpowers: `super_strength`, `flight`, `energy_projection`, `telepathy`, `healing_factor`, `shape_shifting`, `invisibility`, and `telekinesis`. Each power is present in approximately 30% of characters, with no significant difference in prevalence between heroes and villains.

## 2.2 Exploratory Data Analysis

Initial exploration revealed several important characteristics:

1. **Class Balance**: The target variable shows a 65/35 split (heroes/villains), representing moderate imbalance that does not necessitate resampling techniques.
2. **Feature-Target Correlation**: Correlation analysis revealed that no individual feature has a strong linear relationship with the target variable. The highest absolute correlation with `is_good` is approximately 0.15, indicating weak predictive signal in individual features.
3. **Power Distribution**: Superpowers are distributed nearly equally between heroes and villains, suggesting that the type of power a character possesses does not determine their moral alignment.

4. **No Missing Values**: The dataset is complete with no missing values requiring imputation.

## 2.3 Feature Engineering

To enhance the predictive power of our models, we engineered six additional features that capture relationships between existing attributes:

Table 3: Engineered features and their rationale

| Feature | Formula | Rationale |
|---|---|---|
| `total_powers` | $\sum_{i=1}^{8} \text{power}_i$ | Character versatility |
| `power_efficiency` | $\frac{\text{power\_level}}{\text{years\_active}+1}$ | Talent vs. experience |
| `training_intensity` | $\frac{\text{training\_hours}}{\text{age}}$ | Relative dedication |
| `approval_power_ratio` | $\frac{\text{approval}}{\text{power\_level}+1}$ | Public trust relative to power |
| `bmi` | $\frac{\text{weight}}{(\text{height}/100)^2}$ | Physical archetype |
| `experience_score` | $\text{years\_active} \times \text{training\_hours}$ | Lifetime mastery |

The `training_intensity` feature proved particularly valuable, ranking as the second most important predictor in our tuned Random Forest model.

## 2.4 Data Preprocessing

### 2.4.1 Train-Test Split

The dataset was split into training (80%, $n = 960$) and testing (20%, $n = 240$) sets using stratified sampling to preserve the class distribution.

### 2.4.2 Feature Scaling

For algorithms sensitive to feature magnitude (SVM, KNN, Neural Networks), we applied `StandardScaler` to transform features to zero mean and unit variance:

$$z = \frac{x - \mu}{\sigma}$$

Tree-based algorithms (Random Forest, Gradient Boosting) were trained on unscaled data, as they are invariant to monotonic transformations.

## 2.5 Classification Methodology

We evaluated 19 classification algorithms spanning six categories:

### 2.5.1 Linear Models

- Logistic Regression (with L2 regularization)
- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)

#### 2.5.2 Tree-Based Models

- Decision Tree
- Random Forest (200 estimators)
- Extra Trees (200 estimators)
- Gradient Boosting
- Histogram-based Gradient Boosting
- AdaBoost
- XGBoost

#### 2.5.3 Support Vector Machines

- SVM with Linear kernel
- SVM with Radial Basis Function (RBF) kernel
- SVM with Polynomial kernel (degree 3)

#### 2.5.4 Instance-Based Methods

- K-Nearest Neighbors ($k = 5$)
- K-Nearest Neighbors ($k = 10$)

#### 2.5.5 Probabilistic Models

- Gaussian Naive Bayes

#### 2.5.6 Neural Networks

- Multi-Layer Perceptron with architectures: (50), (100, 50), (100, 100, 50)

### 2.6 Hyperparameter Tuning

For the top-performing models, we conducted exhaustive grid search with 5-fold stratified cross-validation:

**Random Forest:**

```
n_estimators: [100, 200, 300]
max_depth: [5, 10, 15, None]
min_samples_split: [2, 5, 10]
min_samples_leaf: [1, 2, 4]
```

**Gradient Boosting:**

```
n_estimators: [100, 200]
learning_rate: [0.01, 0.1, 0.2]
max_depth: [3, 5, 7]
```

**SVM:**

```
C: [0.1, 1, 10, 100]
gamma: ['scale', 'auto', 0.1, 0.01]
kernel: ['rbf', 'poly']
```

## 2.7 Ensemble Methods

We implemented two ensemble strategies:

1. **Voting Classifier**: Hard voting combination of tuned Random Forest, Gradient Boosting, and Logistic Regression
2. **Stacking Classifier**: Random Forest, Gradient Boosting, and KNN as base learners with Logistic Regression as the meta-learner

## 2.8 Clustering Methodology

For unsupervised analysis, we selected seven behavioral features for clustering, deliberately excluding the target variable `is_good` to avoid data leakage:

- `power_level`
- `civilian_casualties_past_year`
- `training_hours_per_week`
- `years_active`
- `public_approval_rating`
- `total_powers` (engineered)
- `power_efficiency` (engineered)

### 2.8.1 Clustering Performed in High-Dimensional Space

An important methodological note: **clustering was performed on the full 7-dimensional feature space**, not on reduced PCA components. Principal Component Analysis was applied **only for visualization purposes** after clustering was complete. This ensures that the clustering algorithm had access to all available information when forming groups.

### 2.8.2 Algorithms Tested

1. **K-Means**: Tested $k \in \{2, 3, ..., 9\}$ with 20 random initializations
2. **DBSCAN**: Grid search over $\epsilon \in \{0.5, 1.0, 1.5, 2.0\}$ and min_samples $\in \{3, 5, 10\}$
3. **Agglomerative Clustering**: Tested $n \in \{2, 3, 4, 5\}$ with linkage methods: ward, complete, average

#### 2.8.3 Evaluation Metrics

- **Silhouette Score**: Measures how similar objects are to their own cluster compared to other clusters. Range: $[-1, 1]$, higher is better.
- **Elbow Method**: Visual inspection of inertia (within-cluster sum of squares) to identify the optimal number of clusters.

# 3 Results

## 3.1 Classification Results

### 3.1.1 Model Comparison

Table 4 presents the performance of all 19 classification algorithms, sorted by test accuracy.

Table 4: Classification model performance (sorted by test accuracy)

| Model | CV Accuracy | Test Accuracy | F1 Score |
|---|---|---|---|
| LDA | 63.9% | **65.0%** | 0.778 |
| SVM (Linear) | 65.0% | **65.0%** | 0.788 |
| Logistic Regression | 63.8% | 64.6% | 0.776 |
| AdaBoost | 63.5% | 64.6% | 0.768 |
| Random Forest | 62.6% | 64.2% | 0.768 |
| Gradient Boosting | 62.9% | 63.3% | 0.766 |
| Extra Trees | 61.5% | 62.9% | 0.758 |
| SVM (RBF) | 64.8% | 62.5% | 0.752 |
| MLP (100, 50) | 62.1% | 62.1% | 0.748 |
| KNN (k=5) | 57.7% | 60.8% | 0.737 |
| Decision Tree | 55.2% | 58.8% | 0.715 |
| XGBoost | 58.5% | 58.3% | 0.708 |

**Key Observation**: All models cluster within a narrow accuracy range of 58–65%, with simple linear models (LDA, Logistic Regression) performing comparably to complex ensemble methods. This suggests that the problem is approximately linearly separable but contains insufficient signal for higher accuracy.

### 3.1.2 Hyperparameter Tuning Results

Table 5: Hyperparameter tuning results for top models

| Model | Best Parameters | CV Score | Test Score |
|---|---|---|---|
| Random Forest | max_depth=15, n_estimators=200 | 65.7% | 63.3% |
| Gradient Boosting | learning_rate=0.01, max_depth=3 | 65.0% | **65.0%** |
| SVM | C=1, kernel='poly' | 65.0% | 62.1% |

### 3.1.3 Ensemble Performance

Table 6: Ensemble method performance

| Ensemble | CV Accuracy | Test Accuracy |
|---|---|---|
| Voting (RF + GB + LR) | 64.1% | 63.8% |
| Stacking (RF + GB + KNN → LR) | 62.3% | 63.3% |

Notably, ensemble methods did **not** outperform the best individual tuned models, indicating that we have reached the performance ceiling for this dataset.

### 3.1.4 Feature Importance Analysis

Figure 1 shows the feature importance scores from the tuned Random Forest model.

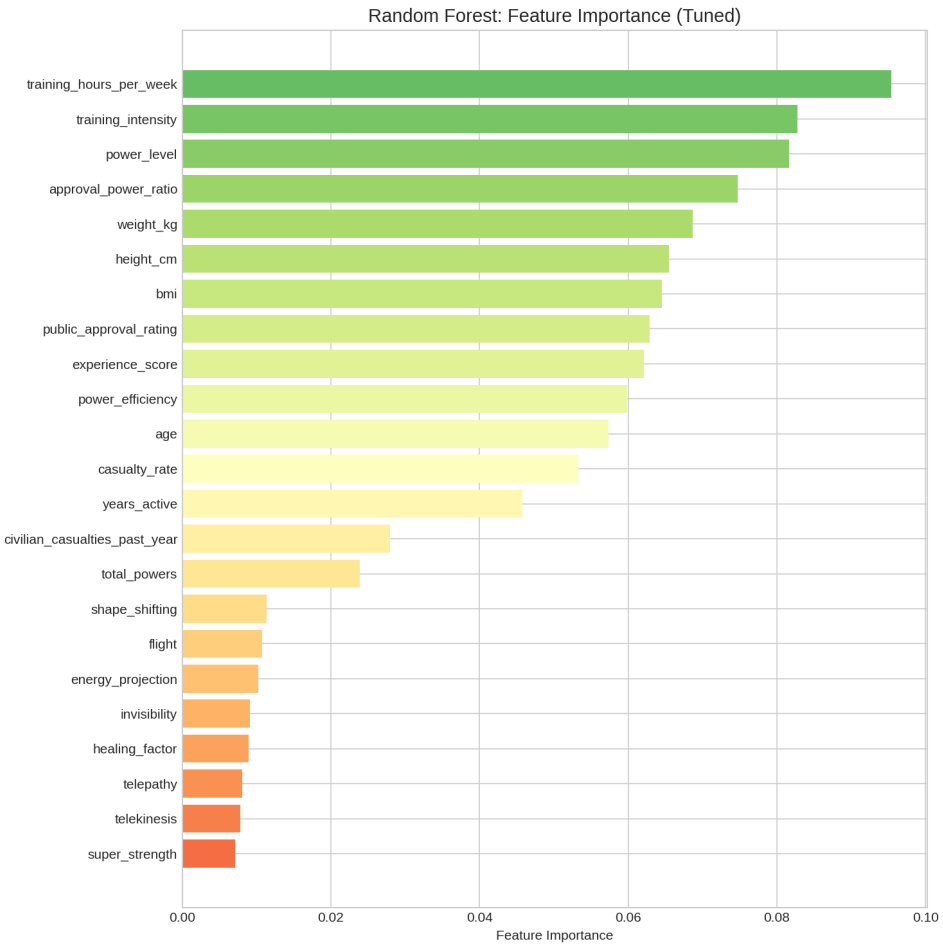


Figure 1: Feature importance from tuned Random Forest classifier

The top three predictive features are:

1. `power_level` — Overall power rating
2. `training_intensity` — Engineered feature (training hours / age)

3. `training_hours_per_week` — Raw training dedication

The presence of an engineered feature (`training_intensity`) in the top three validates our feature engineering approach.

## 3.2 Clustering Results

### 3.2.1 Optimal Number of Clusters

Figure 2 shows the combined elbow method and silhouette analysis for K-Means clustering.

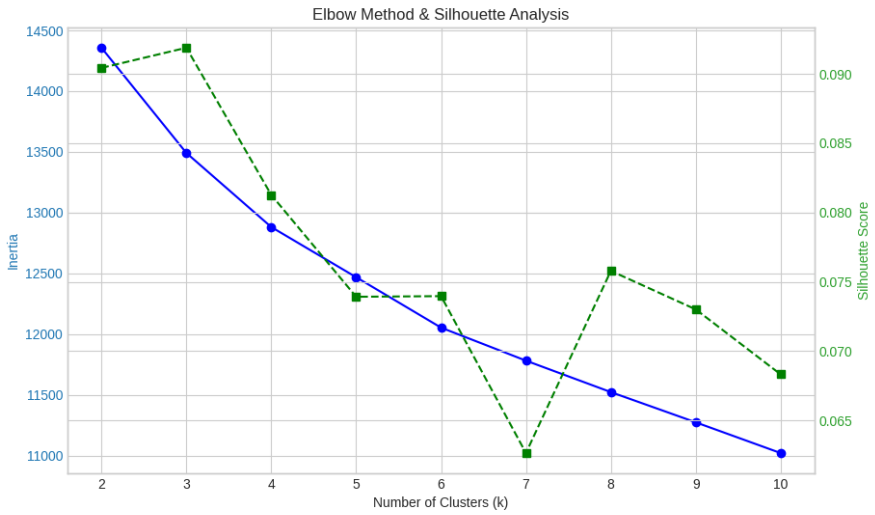


Figure 2: Elbow method (inertia) and silhouette score analysis

The silhouette score peaks at $k = 2$ with a value of 0.167, indicating that the data naturally forms two distinct groups.

### 3.2.2 Algorithm Comparison

Table 7: Clustering algorithm performance

| Algorithm | Best Configuration | Silhouette Score |
|---|---|---|
| K-Means | $k = 2$ | **0.167** |
| Agglomerative | $n = 2$, linkage=ward | 0.154 |
| DBSCAN | Various | Poor (excessive noise) |

K-Means produced the best results, which is expected given the relatively uniform density of the data (DBSCAN struggles with such distributions).

### 3.2.3 Cluster Interpretation

Analysis of the two clusters reveals that they correspond to **power levels** rather than moral alignment:

Table 8: Cluster profiles (mean values)

| Cluster | Size | Power Level | Casualties | Training | % Heroes |
|---|---|---|---|---|---|
| Cluster 0 | ∼600 | High (60+) | Higher | Moderate | 64% |
| Cluster 1 | ∼600 | Low-Mid (<60) | Lower | Variable | 66% |

**Critical Finding**: Both clusters contain nearly equal proportions of heroes and villains (64–66%), confirming that the unsupervised algorithm does **not** discover hero/villain groupings. Instead, it identifies:

- **Cluster 0**: High-power characters (both heroes like Superman and villains like Darkseid)
- **Cluster 1**: Lower-power characters (both heroes like Hawkeye and villains like Crossbones)

### 3.2.4 PCA Visualization

Figure 3 shows the clustering results projected onto two principal components.

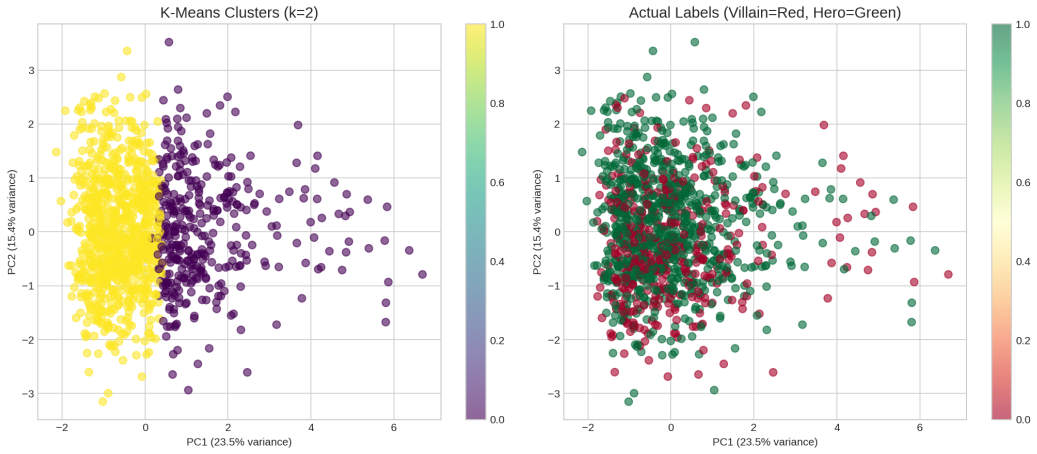


Figure 3: PCA visualization: K-Means clusters (left) vs. actual hero/villain labels (right)

The left panel shows the clusters found by K-Means; the right panel shows the ground truth labels. The visual comparison confirms that cluster boundaries do not align with hero/villain categories.

# 4 Conclusions

## 4.1 Summary of Findings

This project investigated whether machine learning can distinguish heroes from villains and identify character archetypes in a superhero dataset. Our key findings are:

1. **Classification Performance Ceiling**: Despite testing 19 algorithms with extensive hyperparameter tuning and feature engineering, classification accuracy plateaus at approximately 65%. Simple linear models (LDA, Logistic Regression) perform as well as complex ensembles, suggesting the problem is linearly separable but lacks sufficient signal.

2. **Feature Engineering Value**: Engineered features, particularly `training_intensity`, ranked among the top predictors, demonstrating that domain-informed feature creation can extract additional signal from raw data.

3. **Superpowers Do Not Determine Morality**: The eight superpower flags are distributed equally between heroes and villains. Having flight or super strength does not predict whether a character is good or evil.

4. **Natural Clusters Are Power-Based**: Unsupervised clustering reveals two groups based on power level, not moral alignment. Both high-power and low-power clusters contain similar proportions of heroes and villains.

5. **Behavioral Metrics Are Most Predictive**: Features like `power_level` and `training_hours` are the strongest predictors, suggesting that dedication and capability—not specific powers—differentiate characters.

## 4.2 Limitations

Several limitations affect the generalizability of our results:

1. **Potential Synthetic Data**: The dataset's uniform distributions and weak correlations suggest it may be synthetically generated, which would explain the limited predictive signal.

2. **Missing Narrative Features**: Real hero/villain distinctions depend on factors not captured in numerical attributes:

   - Origin stories (e.g., "bitten by radioactive spider" vs. "fell into chemical vat")
   - Motivations and intentions
   - Team affiliations (Avengers vs. Hydra)
   - Specific narrative events

3. **Binary Label Oversimplification**: The binary `is_good` label ignores the moral complexity of characters like anti-heroes (e.g., Punisher, Deadpool).

## 4.3 Future Work

Based on our findings, we recommend several directions for future research:

1. **Richer Data Acquisition**: Incorporate text-based features (character descriptions, origin stories) using natural language processing techniques.

2. **Multi-Class Classification**: Extend beyond binary labels to predict alignment spectra (e.g., Lawful Good, Chaotic Neutral, Chaotic Evil).

3. **Graph-Based Analysis**: Model character relationships and interactions as a network to capture affiliation patterns.

4. **Cross-Universe Analysis**: Compare patterns across different fictional universes (Marvel, DC, independent publishers).

### 4.4 Reproducibility

All code and data are available at: `https://github.com/elbarbary/superhero-classification`

The analysis was conducted using Python 3.10 with the following key libraries: pandas, scikit-learn, matplotlib, seaborn, and XGBoost.

## References

[1] Kenil Shah, "Super-Heros Dataset," Kaggle, 2023. [Online]. Available: `https://www.kaggle.com/datasets/kenil1719/super-heros`

[2] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[3] P. J. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," Computational and Applied Mathematics, vol. 20, pp. 53–65, 1987.