**SUPERHERO RECOMMENDATION BASED ON CHARACTER RELATIVITY**

**Group – A**

Abhishek Naithani - c0871411

Karishma Pramod Shirsath - c0871245

Kalla Tejaswi - c0852124

Venkata Sai Charan Yerapasetty - c0863224

**Instructor:** **Vahid Hadavi**

**Class: AML – 2203 (Advanced Python AI and ML Tools)**

**Date:** 29-07-2023

**ABSTRACT:**

The world of superheroes has captivated audiences for decades, spanning ages, cultures, and geographics. Superheroes, from the most popular characters like Superman and Spider-Man to less prominent characters, have become an integral part of popular culture. It can be difficult to keep up with the enormous variety of superheroes and their various abilities given the comic universe's ongoing expansion. The "Superhero API" can help in this situation. This API serves as a quantified and programmatically accessible data source for all superheroes. The data is scrapped from the API to obtain information about various superhero characters, such as their names, attributes, abilities, affiliations, and other relevant details. This report presents an unsupervised learning model on the superhero recommendation system. It covers some of the crucial data wrangling steps to format the data for analysis and visualization.

**MOTIVATION:**

Superheroes, the iconic figures of courage and strength, have become an integral part of modern storytelling and entertainment. Superheroes have a broad and unique universe that can be found in a variety of comic books, films, and animations. Superheroes, who first appeared in the early 20th century, have developed, and changed with time, reflecting evolving standards and expectations of society.

The attraction to superheroes is everlasting in this vibrant world. The history of these extraordinary characters keeps expanding with each new version and adaptation, indulging followers all around the world. Superheroes and their world, from their modest beginnings to their spectacular deeds, continue to be a beloved aspect of contemporary entertainment. A superhero recommendation system serves as a valuable tool for users to discover, explore, and enjoy the vast superhero universe. It will allow users to explore their interests, preferences, creative projects, and plots which would create a unique and engaging experience.

Through this report, we discuss the process of scrapping data from the ‘Superhero API’ using access tokens from Facebook and respective character IDs. Data cleaning, validating, transforming, and various visualization plots will be understood along with data analysis. K-means clustering and DBSCAN (Density-Based Spatial Clustering of Applications with Noise) machine learning models to develop a superhero recommendation system that will be evaluated on similarity metrics and clusters will be given an overview.

**Data Preprocessing:**

The ‘Request’ library of Python facilitates web scraping and API interaction to gather the character data. The character specifications from each HTTP request are initially stored in a list and the retrieved data from the API is stored in a ‘CSV’ file and converted to a data frame, a highly efficient Pandas tool of the Python library. This data frame is crucial for manipulating and integrating with other libraries like Scikit-learn, Matplotlib, Seaborn, etc. The dimension of the raw data is 759 rows and 28 columns. The data type of each attribute, the dispersion (descriptive statistics), and the presence of missing and duplicate values are checked before preparing the data for further analysis.

**Data Cleaning and Validating:**

Prior to handling the missing values, less relevant columns (‘biography. aliases',' biography. alter-egos',' biography. First appearance,' work. Occupation’, ‘image URL’, etc.) are removed. The final data has been brought down to 15 columns.

1. **Handling Missing Values:**

Considering the data characteristics and the impact of the missing values on the analysis and modeling, the **Imputation** method has been chosen to fill the missing values in the numerical columns (with the mean value of the respective column) and the categorical columns (with the mode value of the respective column). The idea behind choosing the mean and mode values is that they preserve the central tendency of the data, the pattern of the categorical features, and less bias in case of outliers in the data.

1. **Transforming Categorical Columns:**

The two nominal categorical columns (‘appearance. gender', 'biography. alignment') are transformed by **one-hot encoding** to machine-readable numeric format. One-hot encoding vectorizes the nominal data with a ‘get\_dummies()’ function to 0 or 1 value.

1. **Handling Outliers:**

An outlier is a data point that is abnormally high or low in comparison to the closest data point and the rest of the values in the dataset. A deeper statistical analysis of the data is necessary to categorize a data point as an outlier since some of the genuine extreme values might impact the result of the analysis. Understanding the spread of the data, sorting, and splitting the data points helps in efficiently determining each (commonly 4) quartile of the data. Calculating the Inter Quartile Range is a crucial step in finding out the outliers. Monitoring the outliers should be based on research and domain knowledge to avoid bias and distortion in the data.

We have observed that three columns (‘**powerstats. intelligence’, ‘powerstats. speed’, ‘powerstats. combat’**) have outliers. The three techniques to address outliers are explored.

* **Quantile based Flooring and Capping:**

It involves setting a lower bound (flooring) and an upper bound (capping) on the values of a variable based on certain quantiles of the data distribution. We have chosen the **5th and 95th percentiles** of the data points as the floor and cap values that replace the points that are below and above this threshold. This way the data is constrained to have designated minimum and maximum values that improve the stability of the analysis results.

* **Trimming:**

It is a process of removing (truncating) that are present below and above specified threshold quantiles. The defined lower and upper thresholds are the **30th and 95th percentiles** of the data respectively. This method is advisable when the outliers are rare and not relatively significant.

* **Log Transformation:**

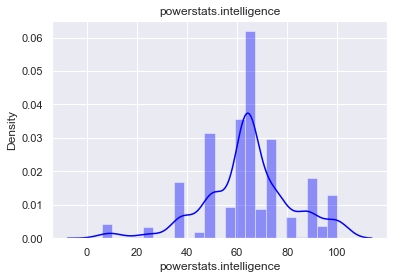
Data undergoes a mathematical process known as a log transformation to lessen the proportion of extreme values and improve the symmetry of the data distribution. It computes the logarithm of the data, typically in base 10 (log10) or the natural logarithm (ln). This method shows the best results especially when the data is skewed and does not contain zero or negative values for outliers.

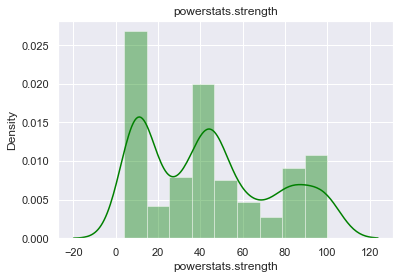
Through visualizing the boxplots of the above-mentioned methods, it is observed that ‘Trimming’ has given decent results and retained most of the data. While Log Transformation has shown some distortion in the distribution of the data. Therefore the data after trimming the outliers is used for clustering.

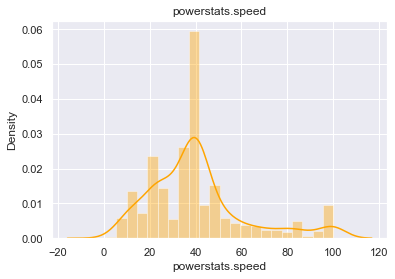
**DATA VISUALIZATION:**

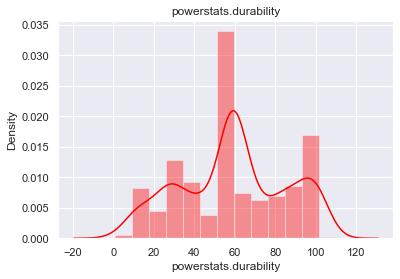
It is an interactive form of understanding Big Data and simplifies complex relationships, patterns, and trends to make informed business decisions.

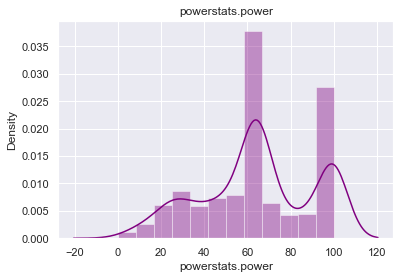
**Distribution of the numerical columns**: Explains the frequency intervals and distribution of the continuous and discrete numerical variables in the data.

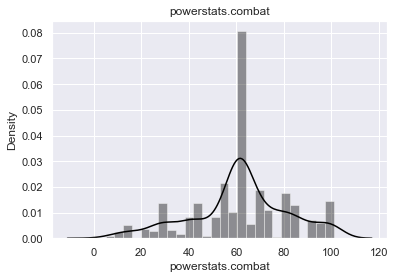












* It is inferred through the below plot that Among the characters, 69 percent are positive and can be categorized as superheroes, while the remaining 27 percent are villains, and the remaining 4 percent fall into the neutral group.
* 70 percent of the characters are male while 26 percent are female superheroes.

**Dimensionality Reduction with Principal Component Analysis (PCA):**

The principal component analysis is a technique for reducing the number of dimensions in large data sets by condensing a large collection of variables into a smaller set that retains most of the large set's information**.** PCA aims to maximize the variance of the projected data while minimizing the reconstruction error. We can get better clustering results by applying PCA as a preprocessing step before K-means clustering with high-dimensional data. By eliminating noise and pointless dimensions with **Simple Imputer**, PCA helps to concentrate on the data's most significant differences. This may result in faster computing times, better clustering performance, and more understandable results.

The **Scikit-learn** library of Python is used to create instances of the principal class for the input data, to fit-transform it to a new lower-dimensional data. The data is standardized using **StandardScaler** to a zero mean and a unit-variance feature space. This process results in components that are linear combinations of the original features, ranked in order of variance captured, and uncorrelated. The dominant features and important relationships among the selected power stats are visualized through Python Seaborn’s scatterplot and used in further modeling.

**Unsupervised Machine Learning Modeling:**

Unsupervised learning involves the computer learning on its own while using unlabelled data to look for hidden patterns and categories. Clustering is an unsupervised technique that finds possible groups and associations in the feature space of the fitted data. We have taken two of various clustering algorithms for the recommendation system.

1. **K-Means Clustering:**

K-means clustering is a well-known unsupervised machine learning approach that divides a dataset into K different clusters based on data similarity trends. The algorithm attempts to limit the differences between data points from distinct clusters to a minimum while clustering together data points that are similar to one another. We have initialized the value of ‘k’ for the number of clusters chosen and iterated to get the optimum ‘k’ value.

1. **Density-Based Spatial Clustering of Applications with Noise (DBSCAN)**

A prominent density-based clustering approach in machine learning is called DBSCAN (Density-Based Spatial Clustering of Applications with Noise). In contrast to conventional clustering algorithms like K-Means, DBSCAN does not demand that the user predetermine the number of clusters. Instead, it uses the density of data points in the feature space to automatically identify clusters. The DBSCAN algorithm's fundamental principle is to combine data points in high-density regions while treating points in low-density regions as outliers or noise. The hyper-parameters of this algorithm are the Epsilon distance between the data points and the Minimum Density points.

* **SILHOUETTE ANALYSIS:**

We have used a similarity metric called ‘The silhouette score’ to quantify the clustering results. The Silhouette coefficients range from -1 to 1, where ‘+1’ indicates a finely clustered object, ‘0’ is close to the decision boundary of two different clusters, and ‘-1’ is a wrongly clustered object. It is an efficient metric to assess the model performance especially when the input data does not have true labels.

* **Superhero Recommendation system:**

The designed recommendation function provides suggestions for superhero characters that are similar to the input superhero character. It is based on three arguments:

* **Name:** It is the name of the entity for which the recommendations are to be generated.
* **Tolerance:** A number indicating the degree of resemblance necessary for a product to be taken into consideration as a suggestion. The similarity score of the chosen recommendation should be greater than or equal to the tolerance value.
* **Similarity metric:** A popular similarity metric in recommendation systems is **cosine similarity**. It calculates a similarity score between -1 and 1, with 1 denoting perfect resemblance, by measuring the cosine of the angle between two non-zero feature vectors or matrices.

For example, the function is given an input of **‘Iron Man’** and to identify characters that are comparable to Iron Man it uses the superhero data retrieved from the Superhero API. 'Iron Man' and other superhero characters' similarity scores are calculated using that similarity metric it filters the characters whose similarity scores fall within the specified range based on the tolerance level. The method then produces a list of superheroes that are comparable to ‘Iron Man’.

**CONCLUSION:**

**References:**

[1] “Superhero api documentation,” SuperHero API, https://www.superheroapi.com/ (accessed Jul. 29, 2023).

[1] “Selecting the number of clusters with silhouette analysis on kmeans clustering,” scikit, https://scikit-learn.org/stable/auto\_examples/cluster/plot\_kmeans\_silhouette\_analysis.html (accessed Jul. 29, 2023).

[1] S. Mishra, “Unsupervised learning and data clustering,” Medium, https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a (accessed Jul. 29, 2023).

[1] D. Singh, “Deepika Singh,” Pluralsight, https://www.pluralsight.com/guides/cleaning-up-data-from-outliers (accessed Jul. 29, 2023).

[1] “Sklearn.preprocessing.onehotencoder,” scikit, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html (accessed Jul. 29, 2023).

[1] S. Y&#305;ld&#305;r&#305;m, “DBSCAN clustering - explained,” Medium, https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556 (accessed Jul. 29, 2023).

[1] “A step-by-step explanation of principal component analysis (PCA),” Built In, https://builtin.com/data-science/step-step-explanation-principal-component-analysis (accessed Jul. 29, 2023).