**Report**

**Hand Made Akinator Game in python with Decision tree**

Course: CSS 324 Introduction to Machine Learning

Teacher: Meraryslan Meraliyev

Made by: Beibitov Damir

**Final Project Report: "Hand-Made Akinator for Comic Book Characters"**

1. **Introduction**

The purpose of this project was to develop an interactive game similar to Akinator, specifically tailored for guessing comic book characters. This project was undertaken as part of the course "Introduction to Machine Learning". The core objective was to apply machine learning techniques in an engaging and user-friendly manner. The game leverages a decision tree classifier to infer the identity of a comic book character based on user-provided information. It demonstrates the practical application of machine learning concepts in creating an interactive experience.

1. **Methodology**
2. **Data Preparation**

The dataset, titled "comix\_characters\_info.csv", comprises various attributes of comic book characters from different publishers. Key attributes include Alignment, Gender, EyeColor, Race, HairColor, Publisher, SkinColor, Height, and Weight. The initial stage involved preprocessing the data, where missing values were handled, and categorical variables were encoded to facilitate machine learning model training.

1. **Model Selection and Training**

A decision tree classifier was chosen for this project due to its interpretability and efficiency in handling categorical data. The model was trained on the preprocessed dataset. Decision trees are particularly suitable for this type of application as they mimic human decision-making processes, making them ideal for a question-answer based game like Akinator.

1. **Implementation**

The game was implemented in Python and involves a series of yes/no questions posed to the user. These questions are based on the attributes of comic book characters, such as publisher, race, eye color, hair color, etc. The decision tree navigates through its branches based on the user’s responses, eventually converging on a specific character prediction.

1. **Results and Demonstration**

The Jupyter notebook includes a demonstration of the game, illustrating its functionality. In the demo, the model successfully guesses a comic book character after a sequence of questions answered by the user. This showcases the decision tree's ability to accurately navigate through its branches and make correct predictions based on limited information.

1. **Conclusion**

This project successfully demonstrates the application of a decision tree classifier in an interactive and user-friendly game. It highlights how machine learning can be utilized beyond traditional data analysis and into areas that engage end-users in a dynamic way. Future enhancements could include expanding the dataset, refining the model for greater accuracy, or adapting the game for different genres or themes.

The Jupyter notebook titled "hand\_made\_akinator.ipynb" centered around creating a game similar to Akinator, which guesses comic book characters based on user responses. Here's an overview of its content:

1. **Data Loading and Preparation:**
   * The notebook begins with importing necessary libraries and loading the data from "comix\_characters\_info.csv".
   * Data preprocessing steps are undertaken, such as handling missing values and encoding categorical variables.
2. **Model Training:**
   * A decision tree classifier is trained on the preprocessed data. The decision tree model is a suitable choice for this kind of application due to its interpretability and ability to handle categorical data effectively.
3. **Game Implementation:**
   * A game function is implemented, which uses the trained decision tree model to guess comic book characters based on the user's input.
   * The game iteratively asks the user yes/no questions about the character's attributes (such as publisher, race, eye color, hair color, etc.) and uses the decision tree's logic to narrow down to a specific character.
4. **Game Demonstration:**
   * The notebook includes an example of how the game is played, with a series of questions and user inputs leading to the model successfully guessing a comic book character.





There are additional characters:





The CSV file "comix\_characters\_info.csv" contains information about comic book characters, which was used to train the decision tree model in my project. Here's a brief overview of the data structure:

1. **Columns in the Dataset:**
   * **ID:** A unique identifier for each character.
   * **Name:** The name of the comic book character.
   * **Alignment:** The character's alignment (good, bad, etc.).
   * **Gender:** The gender of the character (Male, Female, etc.).
   * **EyeColor:** The eye color of the character.
   * **Race:** The race or species of the character.
   * **HairColor:** The hair color of the character.
   * **Publisher:** The publishing company of the comic (e.g., Marvel Comics, DC Comics).
   * **SkinColor:** The skin color of the character.
   * **Height:** The height of the character (in centimeters).
   * **Weight:** The weight of the character (in kilograms).
2. **Data Characteristics:**
   * The data includes a variety of categorical attributes (like EyeColor, HairColor, Race) and numerical attributes (Height, Weight).
   * Some fields contain missing or placeholder values (e.g., '-99' for Height and Weight, '-' for SkinColor).