**🧾 🎯 Project Title: GENETIC ALGORITHMS  
📅 Project Timeline:** February 2020 – April 2020  
🎥 YouTube Demo: <https://youtu.be/AtRJN3bizy0>   
📦 GitHub Source Code: <https://github.com/IvanSicaja/2020.06.03_GitHub_Genetic-Algorithms>  
----------------------------------------------------------------------------------------------------------------

🏷️ My Personal Profiles: ⬇︎  
🎥 Video Portfolio: To be added  
📦 GitHub Profile: <https://github.com/IvanSicaja>  
🔗 LinkedIn: <https://www.linkedin.com/in/ivan-si%C4%8Daja-832682222>  
🎥 YouTube: <https://www.youtube.com/@ivan_sicaja>  
----------------------------------------------------------------------------------------------------------------

### 📚🔍 Project description: ⬇︎⬇︎⬇︎

### 💡 App Purpose

The **genetic algorithms** project demonstrates how evolutionary principles such as selection, crossover, and mutation can be applied to problem-solving.

The project was built to **explore, understand, and demonstrate the principle of genetic algorithms** in a simple, hands-on manner. It shows how randomness, reproduction, and mutation can efficiently find solutions in a search space that would otherwise be too large to explore exhaustively.

### 🧠 How It Works

In this **Python** implementation, the goal was to generate a target sequence: **"Split"** (the name of a Croatian city). The algorithm starts with a **random population** of character sequences, evaluates their similarity to the target using the **fuzzywuzzy library**, and iteratively improves them over successive **generations**.  
A defined percentage of the **best matches (20%)** is selected for **crossover** and **mutation**, producing new generations until the **solution** is found. For the sequence **"Split"**, the program typically **converges** within minutes, though runtime increases **exponentially** with longer target sequences.

WORKFLOW :

* **Random sequences** of letters are generated as the **initial population**.
* **Fitness** is evaluated using the **fuzzywuzzy library** to measure similarity to the target string.
* The top **20%** of sequences are selected for **reproduction**.
* **Crossover** combines parents to produce children, while **mutation** introduces random changes.
* This process repeats over **generations** until a sequence reaches **≥90% similarity** with the target.

### ⚠️ Note

None.

### 🔧 Tech Stack

**Python, Genetic algorithms, fuzzywuzzy, Random module, String module, Evolutionary algorithms, Selection, Crossover, Mutation, VS Code**

### 📸 Project Snapshot

Not available.

### 🎥 Video Demonstration

Not available.

### 📣 Hashtags Section

**# #GeneticAlgorithms #Python #EvolutionaryComputing #Optimization #AlgorithmDesign #MachineLearning #SearchAlgorithms #Crossover #Mutation #AI**