Analysis Arena #001

Participant name: T S Akhilesh

Problem statement: Explore the Pokémon dataset to uncover interesting insights and solve the following optimization problem

Dataset used: The dataset used is the Pokémon dataset, which includes various attributes for each Pokémon, such as their name, types, base stats, abilities, and more.

Data Exploration: .head(), .shape, .info(), and .isnull().sum() methods are used to get an overview of the dataset, its structure, and missing values.

Handling Missing Data

- Numerical columns ('height_m', 'weight_kg', 'percentage_male') are filled with mean values.
- Categorical column ('type2') is filled with 'None'.

Exploratory Data Analysis (EDA)

Various EDA tasks are performed:

- Counting unique Pokémon names.
- Distribution of legendary vs. normal Pokémon.
- Pokémon count by generation and type.
- Top Pokémon classifications and abilities.

Insights drawn from the analysis:

- There are **801** unique pokemons in the dataset
- Legendary pokemons consist of **8.7%** of the total pokemons
- **Dragon pokemon** has the highest number of pokemons in classification
- **Basic_egg_steps, base_total, percentage_male** are columns more related to is_legendary attribute
- **Water** has the highest number of pokemons
- **Generation 7** has the highest value of **attack**
- **Generation 3** has the highest **capture rate** in pokemons

Optimization Process:

- Calculates and ranks Pokémon by total base stats and number of abilities.
- Visualizes type coverage of Pokémon.

Synergy and Team Optimization:

- Defines functions to calculate synergy scores among Pokémon teams and optimize a team for battle effectiveness.

Team optimisation:

- The best team has been evaluated using the scores of each pokemon like attack, defense, special attack, special defense, etc
- **'Kyogre', 'Stoutland', 'Mamoswine', 'Sharpedo', 'Necrozma', 'Kingdra'** is the best team with a score of **4185.139383273339**

Predictive Modeling:

- Prepare data for machine learning:
    - Encodes categorical features using LabelEncoder.
    - Splits data into training and testing sets.
- Train a **RandomForestClassifie**r model and evaluates its performance using metrics like accuracy, precision, recall, and F1-score.
- **Accuracy: 0.99, Precision: 1.00, Recall: 0.94, F1 Score: 0.97** are the best results obtained