Gotta Catch **'Em All**

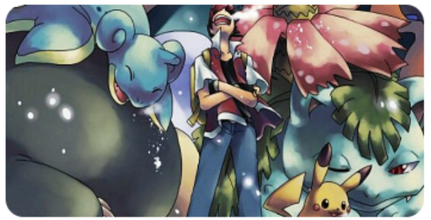
Fatih Bildirici – 23823406
Ankara University Artificial Intelligence PhD Programme
803400815020 - Machine Learning with Python Course

About Dataset

This dataset contains information on a total of 802 Pokémon from all seven generations. The data was scraped from the Serebii site and includes basic statistics of Pokémon, their performance against other species, height and weight, classification, egg steps, experience points, abilities, etc. The dataset is available on Kaggle.

The Complete Pokemon Dataset

Data on more than 800 Pokemon from all 7 Generations.



Data Card | Code (181) | Discussion (11) | Suggestions (0)

About Dataset

Context

This dataset contains information on all 802 Pokemon from all Seven Generations of Pokemon. The information contained in this dataset include Base Stats, Performance against Other Types, Height, Weight, Classification, Egg Steps, Experience Points, Abilities, etc. The information was scraped from http://serebii.net/

Usability
8.24

License
CC0: Public Domain

Expected update frequency
Not specified

Attributes of Dataset

Characteristics of the Dataset:
The data set contains a total of 41 columns. These columns and their contents are explained in detail below:

abilities: List of abilities that Pokémon can have (Text).
against_bug: Pokémon's damage rate against bug-type attacks (Decimal).
against_dark: Pokémon's damage rating against dark-type attacks (Decimal).
against_dragon: Pokémon's damage rating against dragon-type attacks (Decimal).
against_electric: Pokémon's damage rating against electric-type attacks (Decimal).
against_fairy: Pokémon's damage rating against fairy-type attacks (Decimal).
against_fight: Pokémon's damage rating against fight-type attacks (Decimal).
against_fire: Pokémon's damage rating against fire-type attacks (Decimal).
against_flying: Pokémon's damage rating against flying-type attacks (Decimal).
against_ghost: Pokémon's damage rating against ghost-type attacks (Decimal).
against_grass: Pokémon's damage rating against grass-type attacks (Decimal).
against_ground: Pokémon's damage rating against earth-type attacks (Decimal).
against_ice: Pokémon's damage rating against ice-type attacks (Decimal).
against_normal: Pokémon's damage rating against normal-type attacks (Decimal).
against_poison: Pokémon's damage rating against poison-type attacks (Decimal).
against_psychic: Pokémon's damage rating against psychic-type attacks (Decimal).
against_rock: Pokémon's damage rating against rock-type attacks (Decimal).
against_steel: Pokémon's damage rating against steel-type attacks (Decimal).
against_water: Pokémon's damage rating against water-type attacks (Decimal).
attack: Pokémon's basic attack power (Integer)

Attributes of Dataset

base_egg_steps: The number of steps required for the Pokémon's egg to hatch (Integer).
base_happiness: Pokémon's base happiness value (Integer).
base_total: Pokémon's total base statistic value (Integer).
capture_rate: Pokémon's capture rate (Text).
classification: The classification with which the Pokémon is identified in the Sun and Moon Pokedex (Text).
defense: Pokémon's basic defense (Integer).
experience_growth: Pokémon's experience point growth rate (Integer).
height_m: Pokémon's height (in meters) (Decimal).
hp: Pokémon's base health (Integer).
japanese_name: Pokémon's Japanese name (Text).
name: Pokémon's English name (Text).
percentage_male: The percentage of the Pokémon type that is male (If left blank, the Pokémon is genderless) (Decimal).
pokedex_number: Pokémon's entry number in the National Pokedex (Integer).
sp_attack: Pokémon's basic special attack power (Integer).
sp_defense: Pokémon's basic special defense (Integer).
speed: Pokémon's base speed (Integer).
type1: Pokémon's primary type (Text).
type2: Pokémon's secondary type (if any) (Text).
weight_kg: Pokémon's weight (in kilograms) (Decimal number).
generation: The generation in which the Pokémon was first introduced (Integer).
is_legendary: Value indicating whether the Pokémon is legendary (1: Legendary, 0: Not legendary) (Integer).

Data Cleaning and Scaling

As a result of these processes, the resulting data set has the following characteristics:

Filling Missing Values: Missing values are filled with the previous valid value.
Scaling of Numeric Columns: All numeric columns are standardized (mean 0, standard deviation 1).
Addition of Synthetic Data: The dataset was expanded with 500 synthetic samples to provide better balance and diversity.
Removal of Non-numeric Columns: Non-numeric columns were removed from the dataset, making the data analysis and modeling processes more efficient.
This dataset has been made more balanced and useful for various machine learning applications. It is expected to perform better especially in classification problems.

This dataset is enriched with original Pokémon data as well as synthetic data. Missing data were filled in and numeric columns were standardized. This makes the dataset more balanced and useful for various machine learning applications.

scale.py

Descriptive Statistics

Feature	Count	Mean	Std Dev	Min	25th Percentile	Median (50th Percentile)	75th Percentile	Max
against_bug	1301	0.08	1.419	-5.319	-0.831	0.006	0.908	5.032
against_dark	1301	-0.009	0.986	-2.82	-0.331	-0.13	0.141	6.721
against_dragon	1301	0.211	1.467	-6.871	0.088	0.088	0.285	7.101
against_electric	1301	-0.183	1.71	-9.416	-0.877	-0.113	1.025	5.361
against_fairy	1301	-0.008	1.02	-3.233	-0.406	-0.132	0.143	5.617
against_fight	1301	-0.017	0.997	-2.943	-0.789	-0.091	0.584	4.094
against_fire	1301	-0.022	0.989	-3.242	-0.919	-0.196	0.665	4.143
against_flying	1301	0.007	1.002	-3.139	-0.319	-0.319	0.725	4.647
against_ghost	1301	-0.418	2.76	-14.032	-1.766	0.027	0.027	10.783
against_grass	1301	0.03	1.499	-6.724	-0.677	-0.043	0.832	7.712
against_ground	1301	-0.019	1.493	-5.862	-0.81	-0.133	1.222	6.439
against_ice	1301	-0.236	1.436	-6.779	-0.964	-0.283	1.028	5.13
against_normal	1301	-0.288	1.491	-7.762	-1.28	0.425	0.425	3.892
against_poison	1301	0.007	0.972	-2.9	-0.696	0.045	0.176	5.509
against_psychic	1301	-0.062	1.438	-4.928	-1.021	-0.011	0.147	6.051
against_rock	1301	0.015	0.996	-3.922	-0.359	-0.359	0.834	3.947
against_steel	1301	0.293	1.492	-5.445	-0.967	0.033	1.174	6.277
against_water	1301	-0.021	1.012	-3.221	-0.921	-0.096	0.236	4.853
attack	1301	0.002	1.013	-2.954	-0.711	-0.078	0.689	3.529
base_egg_steps	1301	-0.375	3.392	-18.913	-0.511	-0.316	0.27	18.152
base_happiness	1301	0.012	0.991	-3.337	-0.079	0.237	0.237	3.811
base_total	1301	0.007	0.997	-2.973	-0.825	0.056	0.677	3.284
defense	1301	0.025	1.006	-3.601	-0.748	-0.098	0.668	5.105
experience_growth	1301	0.018	0.99	-3.321	-0.343	0.002	0.591	3.653
height_m	1301	0.005	1.592	-5.587	-0.714	-0.153	0.502	12.473
hp	1301	0.011	1.01	-3.376	-0.714	-0.052	0.57	7.005
percentage_male	1301	-0.004	2.729	-14.209	-0.21	-0.21	1.401	11.376
pokedex_number	1301	0.148	2.341	-9.564	-1.051	0.091	1.185	12.153
sp_attack	1301	0.144	1.428	-5.648	-0.814	-0.04	0.944	5.641
sp_defense	1301	-0.01	1.001	-3.083	-0.749	-0.048	0.648	5.697
speed	1301	-0.002	1.018	-2.994	-0.738	-0.046	0.716	3.934
weight_kg	1301	-0.007	1.006	-3.25	-0.492	-0.26	0.295	8.685
generation	1301	0.114	1.029	-3.837	-0.876	0.044	0.679	3.187
is_legendary	1301	-0.309	0.847	-0.309	-0.309	-0.309	0	3.232

Descriptive Statistics

Descriptive Statistics Interpretation
The provided descriptive statistics offer a detailed summary of the Pokémon dataset's various features. Here is a brief interpretation of the key insights:

Count: Each feature has 1301 entries, indicating no missing values after preprocessing.

Mean and Median:
Most features have means close to zero, suggesting that the dataset is centered around its mean values, a result of the standardization process.
Median values being close to zero also indicate a relatively balanced distribution around the mean for most features.

Standard Deviation (Std Dev):
Features like against_ghost (2.760) and percentage_male (2.729) show high variability, indicating significant differences across Pokémon.
Conversely, features like against_dark (0.986) and against_poison (0.972) exhibit lower variability.

Minimum and Maximum:
The wide range between minimum and maximum values in features such as against_ghost (from -14.032 to 10.783) and percentage_male (from -14.209 to 11.376) highlights the diversity and extreme values present in the dataset.
Other features like is_legendary have a more limited range (from -0.309 to 3.232), indicating less variability.

25th and 75th Percentiles:
The interquartile range (IQR) provides insights into the spread of the middle 50% of the data.
Features with a smaller IQR (e.g., against_dark and against_poison) suggest that most Pokémon have similar values in these characteristics.
Features with a larger IQR (e.g., against_ghost and base_egg_steps) indicate a broader spread in these traits.

Classification Analysis

Model	Reasons for Selection
Decision Tree	- Simplicity and Interpretability: Visualizes decision-making processes and is easy to understand.
	- Feature Selection: Automatically selects the most important features and ignores unnecessary ones.
	- Complex Decision Boundaries: Can learn complex decision boundaries in the dataset.
Logistic Regression	- Speed and Efficiency: Works quickly and efficiently on large datasets.
	- Binary and Multiclass Classification: Can be used for both binary and multiclass classification problems.
	- Understanding Feature Importance: Model coefficients help understand the impact of features on the target variable.
Random Forest	- Generalization Ability: Combines multiple decision trees to reduce overfitting risk and improve generalization ability.
	- Versatility: Can be used for both classification and regression problems.
	- Determining Feature Importance: Helps identify which features are more critical.
Support Vector Machine (SVM)	- Complex Decision Boundaries: Can learn complex decision boundaries in high-dimensional datasets.
	- Maximum Margin: Finds the widest margin between classes, generally providing better generalization performance.
	- Kernel Trick: Effective in non-linear datasets.

Analysis for Classification
We removed the **is_legendary** column from the dataset and split it into independent variables (features) and dependent variable (target).

We split the dataset into training and test sets. This was done in order to use 70% of the data as training set and 30% as test set.

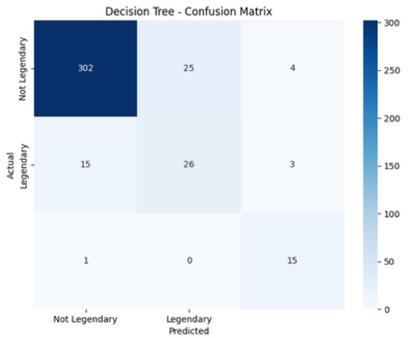
We defined four different classification models that we will use: Decision Tree, Logistic Regression, Random Forest and SVM. These models aim to solve the classification problem using different algorithms.

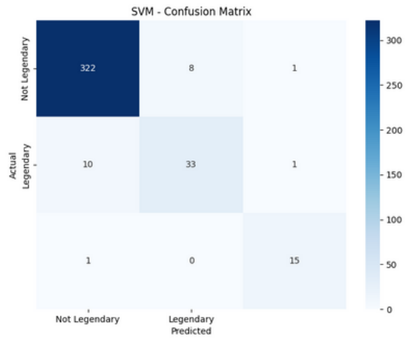
To evaluate the performance of each model, we created an empty DataFrame, which will contain the accuracy, precision, recall, F1 score and AUC scores of the models.

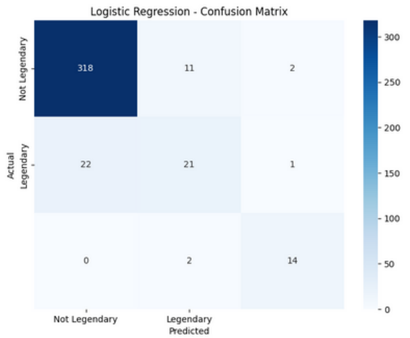
We trained each model on the training data and made predictions on the test data. Using these predictions, we calculated various evaluation metrics and added them to the DataFrame.

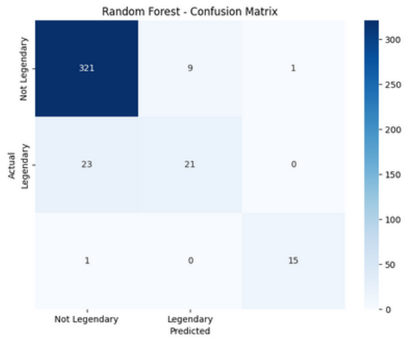
Classification Analysis

Confusion Matrixes

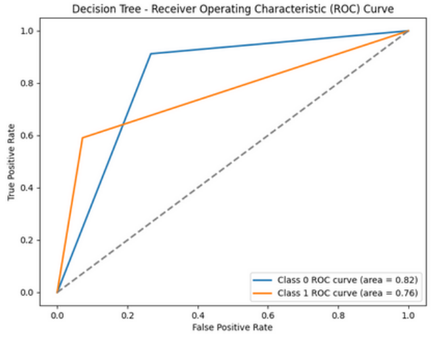


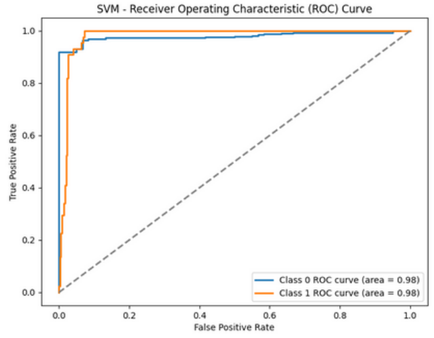


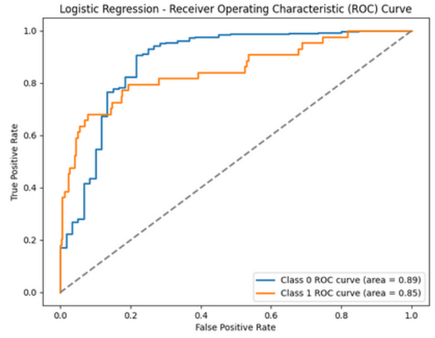


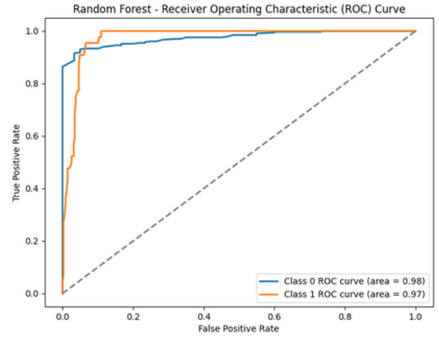


ROC curve (Receiver Operating Characteristic curve)



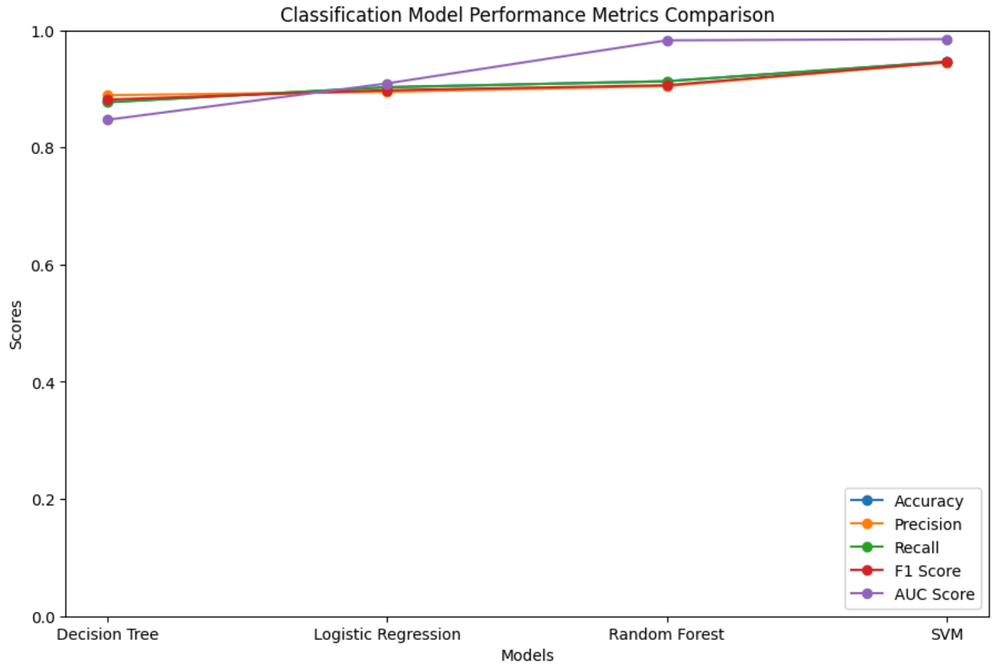






Classification Analysis

Comparison of Model's Performance Metrics



Overall Rating:

Best Model: SVM shows the best performance in all metrics. It stands out especially with its high accuracy and AUC score.

Random Forest: It is the best performing model after SVM. Both accuracy and AUC score are quite high.

Logistic Regression: It performs close to the random forest model and stands out especially with its AUC score.

Decision Tree: Underperforms compared to the other models, but still has an acceptable performance.

Regression Analysis

Model	Reason for Selection
Linear Regression	Simplicity and Interpretability: Linear Regression is an easy to apply and understand model. It provides a good starting point for identifying linear relationships. Baseline Performance Determination: Provides a reference point to compare the performance of other models.
Ridge Regression	Avoiding Overfitting: Ridge Regression avoids overfitting by penalizing large coefficients with the L2 penalty term. Balanced Performance: It provides more balanced and reliable results, especially in cases of multicollinearity.
Random Forest Regressor	Capturing Complex Relationships: Captures complex and non-linear relationships by creating and combining multiple decision trees. Strong Performance: Provides more accurate and stable forecasts by better modeling the interactions between variables in the dataset.

We evaluate the performance of each model with the following metrics:

Mean Squared Error (MSE): Measures the mean squared difference between actual and predicted values.

Mean Absolute Error (MAE): Measures the mean absolute difference between actual and predicted values.

R^2 Score: Shows how much of the variance of the dependent variable can be explained by the independent variables.

In the data preparation phase, we identified the features (independent variables) and the target variable (dependent variable) to use in the regression analysis.

x_reg: Properties, all columns except the target variable.
y_reg: Column 'attack', the target variable.

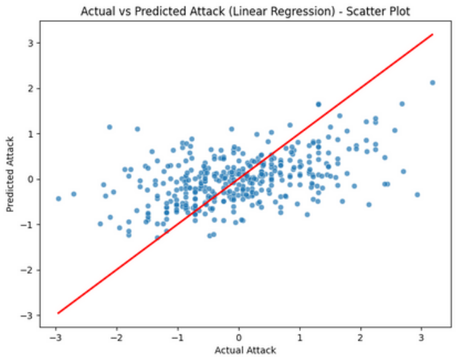
We split our dataset into training and test sets. The training set is used to learn the model and the test set is used to evaluate the performance of the model. We divided the data into 70% training set and 30% test set.

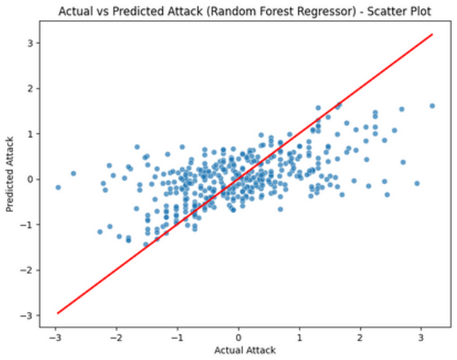
Classification Metrics:

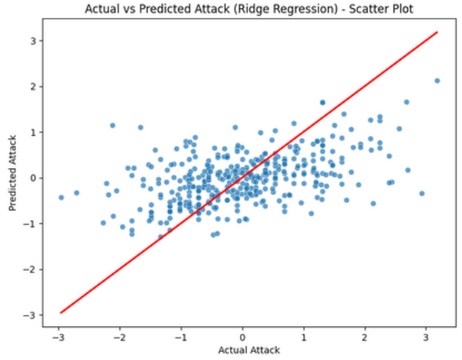
	Model	Accuracy	Precision	Recall	F1 Score	AUC Score
0	Decision Tree	0.877238	0.889223	0.877238	0.881752	0.847236
1	Logistic Regression	0.902813	0.894975	0.902813	0.897705	0.909320
2	Random Forest	0.913043	0.904793	0.913043	0.906202	0.982773
3	SVM	0.946292	0.945264	0.946292	0.945627	0.984966

Regression Analysis

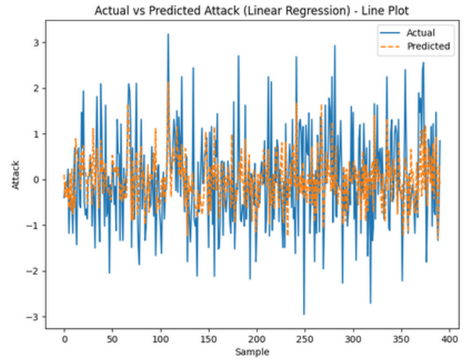
Scatter Plot of Regression Models

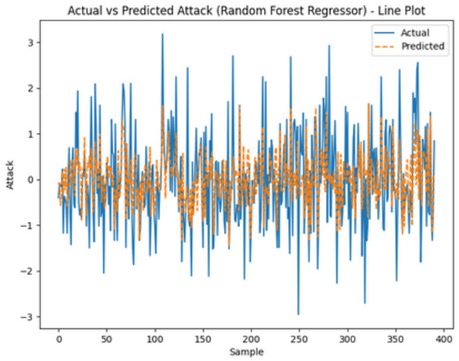


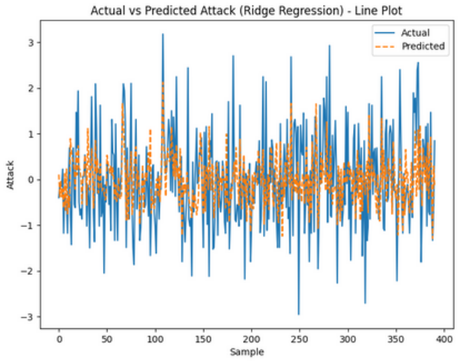




Bar Plot of Regression Models

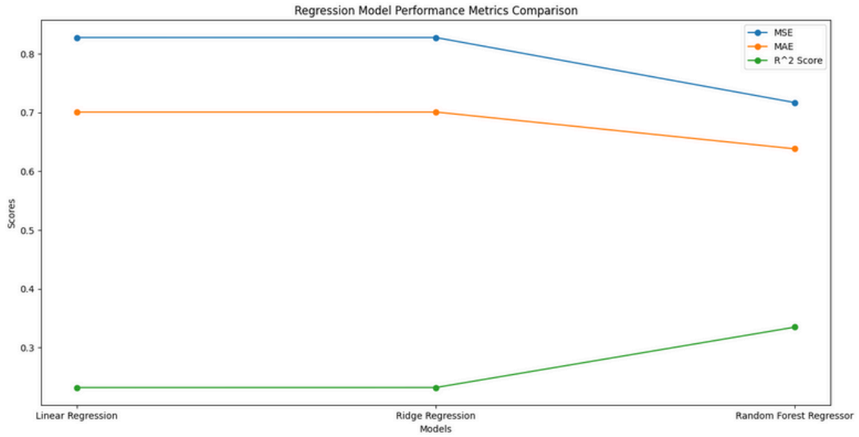


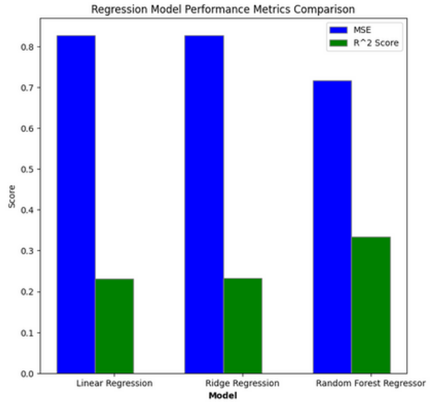




Regression Analysis

Comparison of Model's Performance Metrics





	Model	MSE	MAE	R^2 Score
0	Linear Regression	0.827447	0.700618	0.231863
1	Ridge Regression	0.827390	0.700643	0.231916
2	Random Forest Regressor	0.716720	0.638047	0.334653

Overall Rating:

Best Model: The random forest regression model performs best with lower error values (MSE and MAE) and higher R^2 score. This model is better able to capture the relationship between the independent variables and the dependent variable.

Linear and Ridge Regression: Both models perform similarly and are weak in explaining the dataset. The regularization effect of Ridge regression does not provide a significant improvement in this data set.

In conclusion, the Random Forest Regressor seems to be the best option for your regression problem. This model is better able to model the data with lower error rates and higher R^2 scores. Linear and Ridge regression models with their simpler structures can be useful in some cases, but their performance on this dataset is poor.

Regression Analysis

Comparison of Model’s Performance Metrics

General Conclusion and Comments

In this study, we compared and evaluated the performance of different models in classification and regression analyses. In classification analyses, the SVM model showed the highest performance, while in regression analyses, the Random Forest Regressor gave the best results.

Results:

For classification, the SVM model is the most successful model in accurately classifying legendary Pokemon.
For regression, the Random Forest Regressor model was the most successful in predicting the attack power of Pokemon.
These results provide important insights in understanding the relationships in the dataset and model performances. The obtained models can serve as a reference for future prediction and classification processes on similar datasets.

Evaluation in terms of Machine Learning

This study showed how machine learning models are applied and evaluated for different data types and problems. Classification and regression analyses provided a good basis for comparing the performance of different models.

Model Selection: Model selection was based on the characteristics of the data set and the target variable. A wide range of models were evaluated, from simple models (Linear Regression, Logistic Regression) to more complex models (Random Forest, SVM).
Performance Evaluation: The performance of the models was evaluated with metrics such as accuracy, precision, sensitivity, F1 score, AUC score, MSE, MAE and R^2 score. These metrics demonstrated the generalization capabilities and predictive accuracy of the models.
Avoiding Overfitting: Ridge Regression and Random Forest models provided more balanced and reliable results with their overfitting prevention effects.