

Fatih Bildirici – 23823406

Ankara University Artificial Intelligence PhD Programme

803400815020 - Machine Learning with Python Course

Abstract

This study aims to perform classification and regression analysis on the Pokémon dataset using various machine-learning techniques. First, the necessary data cleaning and preprocessing steps were performed on the dataset, and then classification and regression analyses were performed. The study aims to compare the performance of different machine learning models and to reveal which models give more effective results. In addition, the overall performance and generalization capabilities of the models are evaluated by adding synthetic data to the dataset.

Introduction

Machine learning is a powerful tool for making meaningful conclusions and predictions on large data sets. In this study, classification and regression analyses were performed with various machine-learning models using the Pokémon dataset. Preprocessing steps such as data cleaning, scaling, and synthetic data addition were applied to the dataset.

Data Set and Cleaning Processes

Data Loading and Analysis: First, the Pokémon dataset was loaded, and the first few rows and basic information were analyzed. The data type and missing value status of each column in the dataset were analyzed.

Processing Missing Values: In case of missing values in the dataset, these missing values were filled with appropriate methods. This step is critical to maintain the integrity of the dataset and ensure that the models are trained correctly.

Processing Categorical Values: The categorical values in the dataset were appropriately transformed to enable the models to better understand the data. This is usually done using techniques such as one-hot encoding or label encoding.

Scaling of Numeric Columns: The numeric columns in the dataset were scaled for better performance of the models. Using methods such as Min-Max Scaler or Standard Scaler, the numerical values in the dataset were scaled to a certain range.

Adding Synthetic Data

Synthetic Data Generation: To obtain a more balanced and diverse data set, synthetic data was created using the `make_classification` function. This synthetic data was added to the original data set to train and evaluate the models on a larger data set.

Merging the Data Sets: The synthetic data set was merged with the original data set. This step was performed to ensure that the models produced more general and reliable results.

This study covers machine learning analyses performed on the Pokémon dataset and the results of these analyses. The findings of the study provide valuable information to compare the performance of different machine learning models and to determine which models are more effective on specific datasets. The application of machine learning methods and interpretation of the results is an important reference for researchers working in the fields of data science and machine learning.

Keywords: Machine learning, classification, regression, data cleaning, synthetic data, Pokémon dataset

1. Scaling and Cleaning the Raw Dataset to

Data Cleaning and Preprocessing Steps

Data Upload and Review: The original Pokémon dataset was loaded and analyzed. The structure of the dataset, column names, and data types were determined.

Processing Missing Values: Missing values were filled with the forward fill method.

Removing Non-numeric Columns: Columns containing text data were removed to work with numeric data only.

Scaling of Numeric Columns: Numeric columns were scaled using StandardScaler.

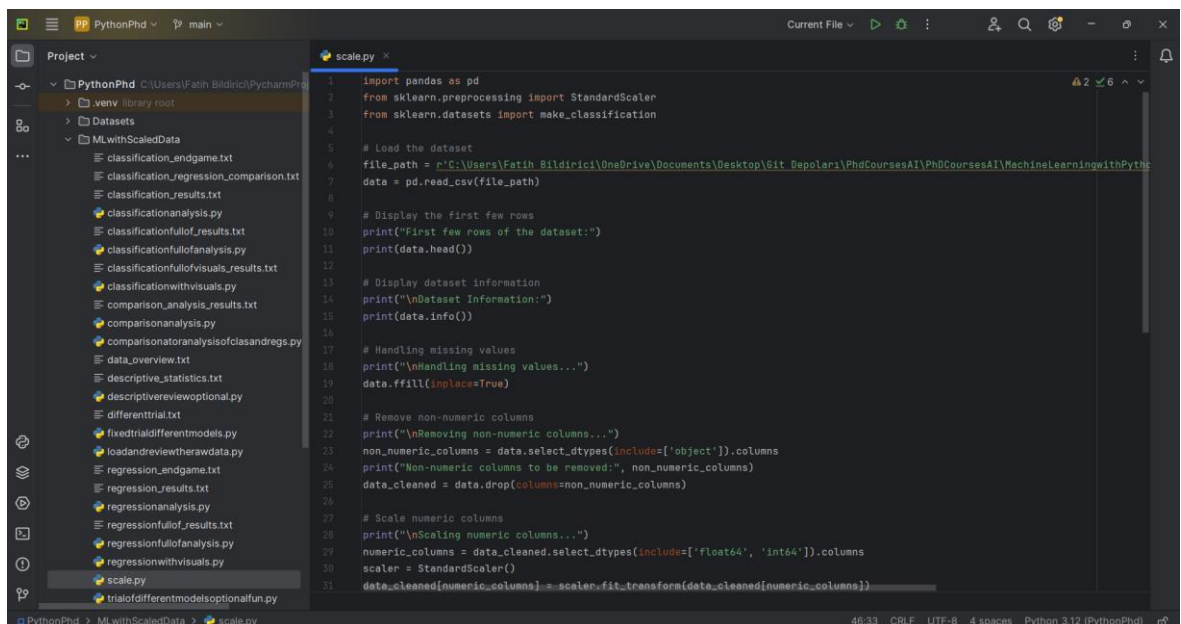
Adding Synthetic Data

Synthetic Data Creation: A synthetic dataset of 500 samples was created with the `make_classification` function and classified with the 'is_legendary' tag.

Merging the Data Sets: The original and synthetic datasets were merged, which allowed the models to be trained on a larger and more balanced dataset.

These operations on the dataset enabled machine learning models to make more efficient and accurate predictions. The addition of synthetic data improved the generalization capabilities of the models by increasing the balance of the dataset. The data cleaning and preprocessing steps positively affected the performance of the models by reducing the noise in the dataset.

As a result, these data cleaning and synthetic data additions made the Pokémon dataset more suitable for machine learning models and contributed to more reliable and accurate predictions. This process once again emphasizes the importance of data cleaning and preprocessing in data science and machine learning projects.



```
1 import pandas as pd
2 from sklearn.preprocessing import StandardScaler
3 from sklearn.datasets import make_classification
4
5 # Load the dataset
6 file_path = r'C:\Users\Fatih Bıldırıcı\OneDrive\Documents\Desktop\Git_Depoları\PhdCoursesAI\PhdCoursesAI\MachineLearningWithPython\
7 data = pd.read_csv(file_path)
8
9 # Display the first few rows
10 print("First few rows of the dataset:")
11 print(data.head())
12
13 # Display dataset information
14 print("\nDataset Information:")
15 print(data.info())
16
17 # Handling missing values
18 print("\nHandling missing values...")
19 data.ffill(inplace=True)
20
21 # Remove non-numeric columns
22 print("\nRemoving non-numeric columns...")
23 non_numeric_columns = data.select_dtypes(include=['object']).columns
24 print("Non-numeric columns to be removed:", non_numeric_columns)
25 data_cleaned = data.drop(columns=non_numeric_columns)
26
27 # Scale numeric columns
28 print("\nScaling numeric columns...")
29 numeric_columns = data_cleaned.select_dtypes(include=['float64', 'int64']).columns
30 scaler = StandardScaler()
31 data_cleaned[numeric_columns] = scaler.fit_transform(data_cleaned[numeric_columns])
```

Figure 1: Scale.py

2. Descriptive Statistics

Descriptive statistics provide basic statistical summaries of the Pokémon dataset. These summaries provide important insights into the overall structure of the dataset and help us understand the data before we begin data analysis. Here is a brief overview of these summaries:

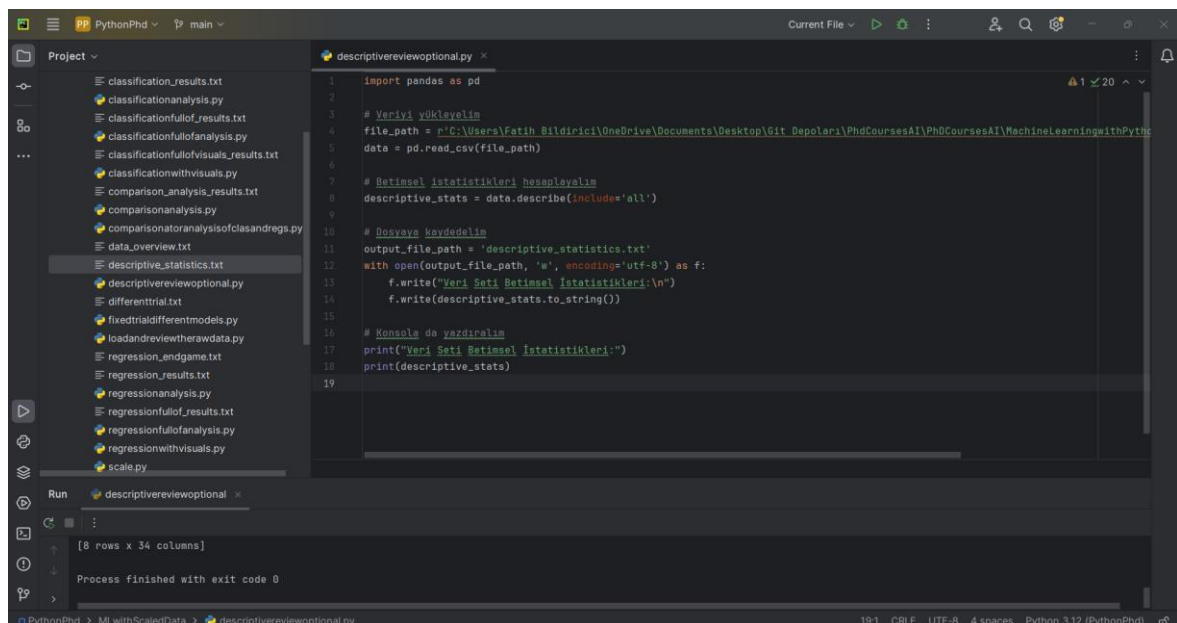
Mean: Shows the average value for each attribute. For example, the mean value of attack is approximately -0.021.

Standard Deviation (Std): Shows the spread of the data. A high standard deviation indicates that the data spans a wider range.

Minimum and Maximum (Min-Max): Shows the minimum and maximum values for each feature. This helps us identify outliers in the data set.

Quartiles (25%, 50%, 75%): Indicates the percentiles of the data. For example, the 25%, 50%, and 75% quartile values of attack are -0.711, -0.078 and 0.689 respectively.

These statistics help us understand the distribution of the dataset and the general behavior of the features. Considering that the features of the data are scaled, and non-numeric columns are removed, these descriptive statistics provide a better understanding of training and evaluation of machine learning models. The basic statistical values of the data, such as the mean, minimum and maximum values, and standard deviation, help us to identify the overall structure of the data and potential analysis challenges.



```
1 import pandas as pd
2
3 # Veriyi yukleyelim
4 file_path = r"C:\Users\Fatih.Bildirici\OneDrive\Documents\Desktop\Git_Depolari\PhdCoursesAI\PhdCoursesAI\MachineLearningWithPython\pokemon_data.csv"
5 data = pd.read_csv(file_path)
6
7 # Betimsel istatistikleri hesaplayalım
8 descriptive_stats = data.describe(include='all')
9
10 # Dosyaya kaydedelim
11 output_file_path = 'descriptive_statistics.txt'
12 with open(output_file_path, 'w', encoding='utf-8') as f:
13     f.write("Yeni Sati Betimsel İstatistikleri\n")
14     f.write(descriptive_stats.to_string())
15
16 # Konsola da yorduralım
17 print("Yeni Sati Betimsel İstatistikleri:")
18 print(descriptive_stats)
19
```

Run descriptiveviewoptional.py

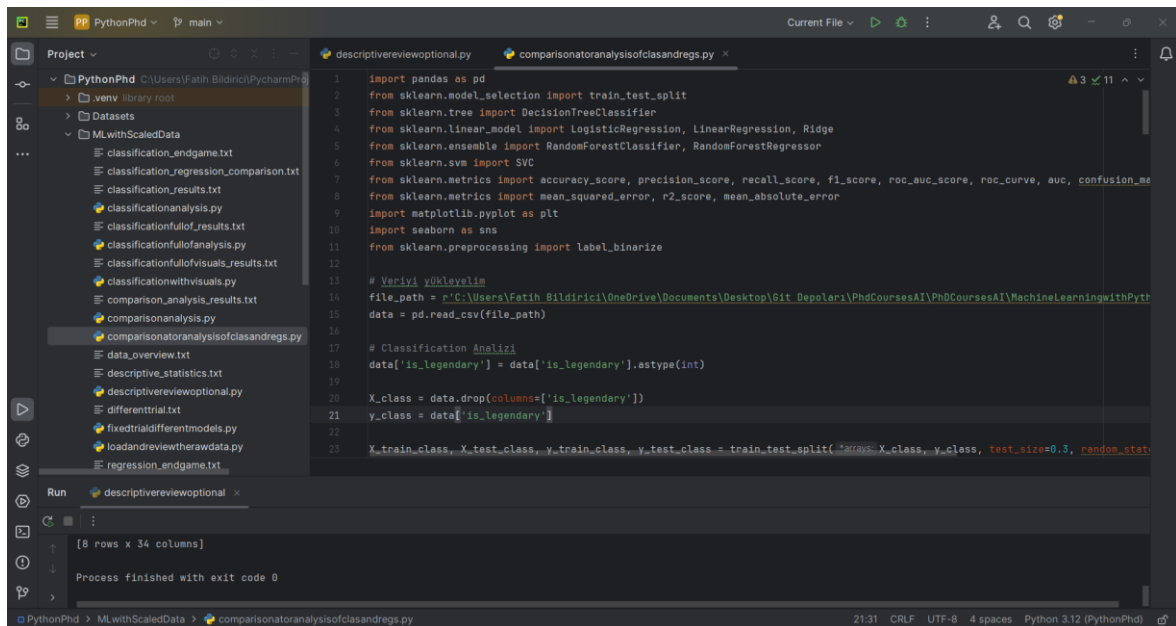
[8 rows x 34 columns]

Process finished with exit code 0

Figure 2: Descriptive Statistics

3. Detailed Analysis

In this study, classification and regression analyses were performed on the Pokémon dataset. First, the dataset was loaded and analyzed. Missing values were filled by forward filling and non-numeric columns were removed to work with numeric data only. The dataset was scaled using StandardScaler and a synthetic dataset of 500 samples was added with the `make_classification` function to make the dataset more balanced and diverse. In the classification analysis, Decision Tree, Logistic Regression, Random Forest, and SVM models were used to compare their performance. For each model, metrics such as accuracy, precision, recall, F1 score, and AUC score were calculated, and the results were visualized with a confusion matrix and ROC curves. In regression analysis, Linear Regression, Ridge Regression, and Random Forest regression models were used. The performance of these models was evaluated with metrics such as MSE, MAE, and R^2 score, and scatter plot and line plot visualizations of actual and predicted values were performed. Finally, the performance metrics of both classification and regression models were compared and the results were analyzed and presented graphically. This study provided a better understanding of the dataset and a comparison of the performance of machine learning models.



```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.tree import DecisionTreeClassifier
4 from sklearn.linear_model import LogisticRegression, LinearRegression, Ridge
5 from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
6 from sklearn.svm import SVC
7 from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, roc_curve, auc, confusion_matrix
8 from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
9 import matplotlib.pyplot as plt
10 import seaborn as sns
11 from sklearn.preprocessing import LabelBinarizer
12
13 # Veriyi yükleyelim
14 file_path = r'C:\Users\Fatih Bildirici\OneDrive\Documents\Desktop\Git_Depolar\PhdCoursesAI\PhdCoursesAI\MachineLearningWithPyT
15 data = pd.read_csv(file_path)
16
17 # Classification Analizi
18 data['is_legendary'] = data['is_legendary'].astype(int)
19
20 X_class = data.drop(columns=['is_legendary'])
21 y_class = data['is_legendary']
22
23 X_train_class, X_test_class, y_train_class, y_test_class = train_test_split(X_class, y_class, test_size=0.3, random_state=42)
```

Figure 3: Detailed and Merged Code

4. Comparison Analysis for Classification and Regression

Classification Analysis Results

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

Table 1: Classification Analysis Result

Accuracy: Shows the accuracy of the models. SVM has the highest accuracy rate.

Precision: Shows the true positive rates of the models. SVM also shows the highest performance here.

Recall: It shows the true positive rates. SVM is also ahead in this metric.

F1 Score: It is the harmonic average of Precision and Recall. SVM again shows the best performance.

AUC Score: This is the area under the ROC curve and SVM has the highest score, indicating that its classification performance is quite good.

Regression Analysis Results

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

Table 2: Regression Analysis Results

MSE (Mean Squared Error): Shows the mean squared error of the prediction errors of the models. Random Forest Regressor has the lowest MSE value.

MAE (Mean Absolute Error): Shows the average of the absolute values of the prediction errors of the models. Random Forest Regressor has the lowest error rate in this metric as well.

R^2 score: This shows the explained variance ratio of the models. Random Forest Regressor has the highest R^2 score, indicating that the model best explains the data.

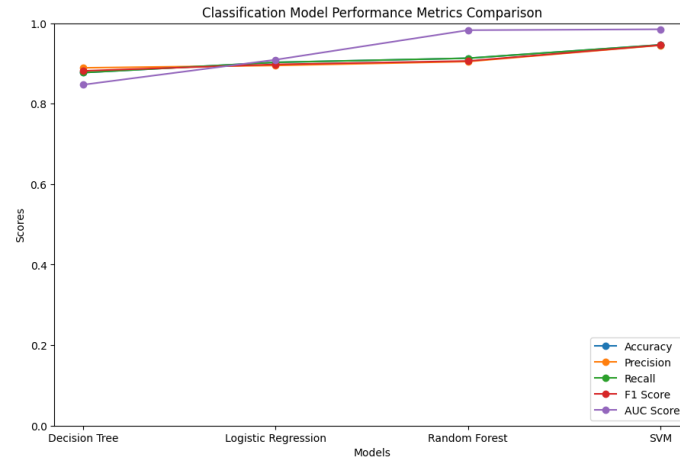


Figure 4: Classification Performance Metrics Comparison

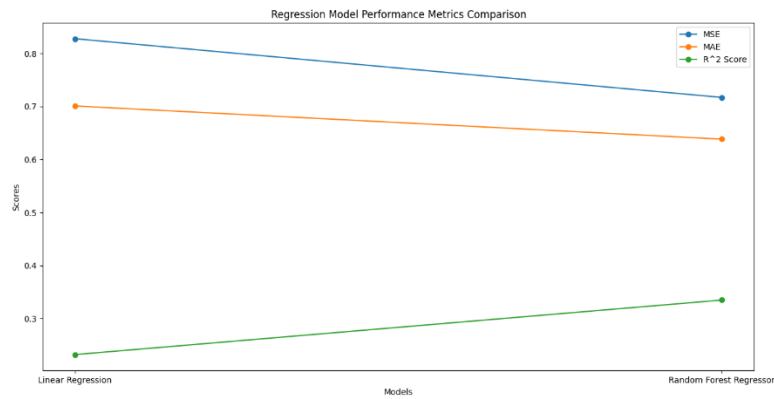


Figure 5: Regression Performance Metrics Comparison

General Evaluation

Classification Analysis: In the classification analysis, the SVM model showed the highest performance in all metrics. Logistic Regression and Random Forest models also performed well but lagged behind SVM. Decision Tree showed lower performance compared to the other models.

Regression Analysis: In regression analysis, the Random Forest Regressor model performed the best in all metrics. Linear Regression and Ridge Regression models performed similarly but lagged Random Forest Regressor. Ridge Regression performed slightly better than the Linear Regression model.

Conclusion: These analyses on the dataset provided a comparative evaluation of the performance of different machine learning models. The best classification model was SVM, while the best regression model was Random Forest Regressor. These results helped us to choose the best models for the dataset and the problem.