PROJECT 5, Naive Bayes Classifier
CS5830, Introduction to Data Science

Kian Arnold
Shylee Oler

Github Link
https://github.com/SnartBlast/cs5830_project5

Slides Link
https://docs.google.com/presentation/d/1NurI49elmjgP3n95lTRDHJTOzRu7VM9FpLn01eVhlso/edit?usp=sharing

INTRODUCTION

For this project, we analyzed data involved in the fictional world of Pokemon. The intent behind analyzing this data was to determine important relationships between elements of the data set. Our analyses are based on the data set which was created by our group using the Pokemon API. The created data set includes columns of Pokemon names, evolution levels, as well as several stats relating to Pokemon performance and viability in Pokemon combat. The results of this data may be valuable to professional gamers, video game developers, and betters alike and provide valuable insight into choosing the best early-level Pokemon, determining the best area for catching Pokemon, and the best move sets for a variety of Pokemon types.

DATA SET

For our analysis, we used the data set present in the Github repo titled, ‘pokemon.csv’. This data set was created previously by Kian Arnold and her previous group for project 3. This data set includes several items of interest in Pokemon such as type, name, evolution level, attack stat, defense stat, and more. The purpose of using this data set was to create a Bayes Classifier that would determine a Pokemon’s evolution level.

ANALYSIS TECHNIQUE

To begin with an analysis, the data needed to be cleaned and organized. We began cleaning the data by firstly removing the data set column, 'Type_2'. This column contained categorical variables of the Pokemon's second type. This data was previously important but would complete the data frame if it were attempted to create one hot encoding with it. Therefore, this column was removed. Additionally, to create a value for the Bayes classifier, third evolution Pokemon were removed from the data set, leaving just two options for Pokemon evolution level. The remaining Pokemon were denoted with a '0', if they were evolution level 1, and a '1' if they were evolution level 2. The final cleaning and organizing attempt on this data set was to create a one-hot encoding of the primary Pokemon type. There are 15 Pokemon types and therefore, 15 columns for hot encoding were created which helped to identify Pokemon types.

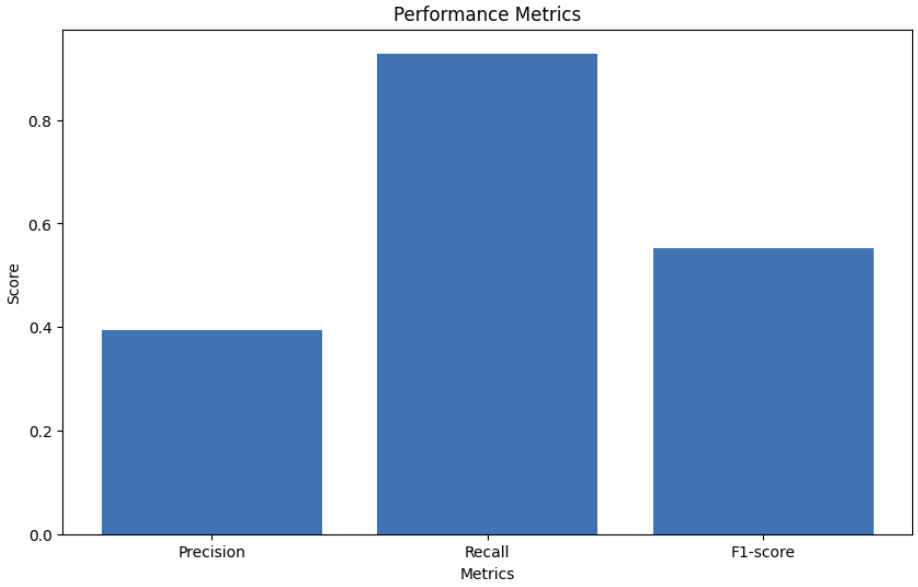
RESULTS

The goal of this project was to create an effective Bayes classifier to determine if Pokemon were evolution level 1 or 2. As previously stated, plenty of data cleaning and organizing went into creating our new data frame. After the data was cleaned and organized, a Gaussian Naive Bayes classifier was created and NaN values were removed from the data frame. We then performed a 10-fold cross-validation of precision, recall, f1, and support scores and kept the scores when their precision and recall were higher than the previous highest precision and recall scores. The best score that we received is shown below.

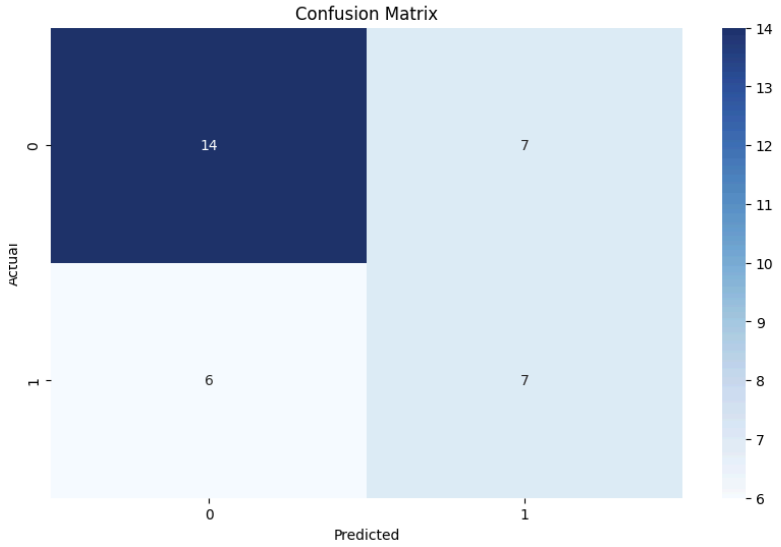
```
'Best Precision -> 0.5625'
'Best Recall    -> 0.9473684210526315'
'Best F1        -> 0.7058823529411765'
```

These scores were surprising to us as the precision is relatively low and the recall is relatively high. Knowing that precision is a measure of the accuracy of our Bayes classifier to determine real positives in our data set, it seems that our data set was not conducive to determining first and second-level Pokemon. Additionally, because recall is a measure of our Bayes classifier to determine predictive positives, it seems that our data excelled in predicting these values. With a high score F1 of 0.706, it seems that our data set is a poor selection for maximizing Bayes classifier predictions.

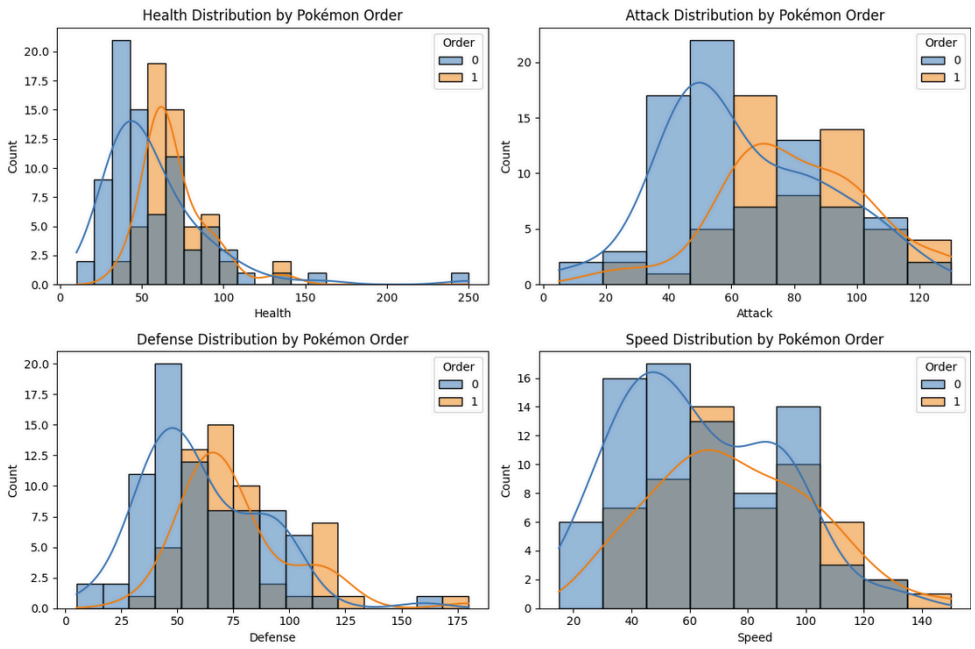
This graph helps us evaluate the overall performance of the Naive Bayes classifier by visualizing key metrics such as precision, recall, and F1-score.



The confusion matrix heatmap offers a detailed view of the classifier's performance by illustrating the distribution of true positive, false positive, true negative, and false negative predictions. This visualization helps in identifying any patterns of misclassification and assessing the classifier's accuracy and reliability.



These distribution plots provide valuable insights into the relationship between Pokémon attributes (e.g., health, attack, defense, speed) and their respective orders in the Pokédex. By visualizing attribute distributions, this graph aids in understanding the variation and significance of attributes in the Pokédex.



TECHNICAL

In the initial stage of the analysis, the dataset was preprocessed to suit the requirements of the Naive Bayes classifier. This involved removing third-evolution Pokémon entries and adjusting the index accordingly. Additionally, the categorical variable 'Type_1' was subjected to one-hot encoding to facilitate its integration into the classifier. Missing values were also handled appropriately before proceeding with the analysis.

Given that the dataset consists of quantitative variables such as health, attack, defense, and speed, the Gaussian Naive Bayes classifier is suitable due to its assumption of feature independence and its capability to handle continuous data. The Gaussian Naive Bayes classifier was chosen for its compatibility with the dataset's attributes and its simplicity. In the analysis process, the model was trained and evaluated iteratively using train-test splits. Precision, recall, and F1-score were calculated for each iteration, with the best precision and recall values recorded across multiple runs. Despite achieving a high recall value, the classifier's precision remained relatively modest, indicating potential areas for improvement. Alternative approaches could involve experimenting with different feature sets or exploring other classification algorithms to further enhance the model's performance.