Poképredictor:Gotta predict em all!

Name:	**Satyajit Das**
Registration No./Roll No.:	19281
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	February 02, 2022
Date of Submission:	April 24,2022

Introduction

The objective of this project is to predict the winner based on characteristics of the two pokemons.The data set contains three files the training data,test and Pokémon data. This Pokémon data set includes about 800 Pokémon, including their number, name, first and second type, and basic stats: HP, Attack, Defense, Special Attack, Special Defense, and Speed. Based on HP, Attack, Speed, Defense, Special Attack, Special Defense and other features to predict which Pokémon will win the battle. At the same time, every Pokémon have different classification type. Each Pokémon has a different type. Here I am using Naive Bayes, SVM, Random Forest, and KNN to do the classification. The best way for to compare the accuracy of each method is to perform cross-validation tests on each algorithm one by one, compare the result, and then adjust the parameters to ensure that each algorithm achieves the best solution. With those certain types, I prepared a machine learning model for final project.

Methods

Classifiers

Naive Bayes: A naive Bayes classifier is an algorithm that uses Bayes' theorem to classify objects. Naive Bayes classifiers assume strong, or naive, independence between attributes of data points. Popular uses of naive Bayes classifiers include spam filters, text analysis and medical diagnosis. These classifiers are widely used for machine learning because they are simple to implement.

Support Vector Machine (SVM): A support vector machine (SVM) is machine learning algorithm that analyzes data for classification and regression analysis. SVM is a supervised learning method that looks at data and sorts it into one of two categories. An SVM outputs a map of the sorted data with the margins between the two as far apart as possible. SVMs are used in text categorization, image classification, handwriting recognition and in the sciences.

Random Forest: A random forest is a data construct applied to machine learning that develops large numbers of random decision trees analyzing sets of variables. This type of algorithm helps to enhance the ways that technologies analyze complex data.

KNN: A k-nearest-neighbor algorithm, often abbreviated K-NN, is an approach to data classification that estimates how likely a data point is to be a member of one group or the other depending on what group the data points nearest to it are in. The k-nearest-neighbor is an example of a "lazy learner" algorithm, meaning that it does not build a model using the training set until a query of the data set is performed. Other than these Classifiers I also tried some other Classifiers like Logistic Regression and Decision Tree. In the given dataset, there are 18 characteristics. How to select the features that

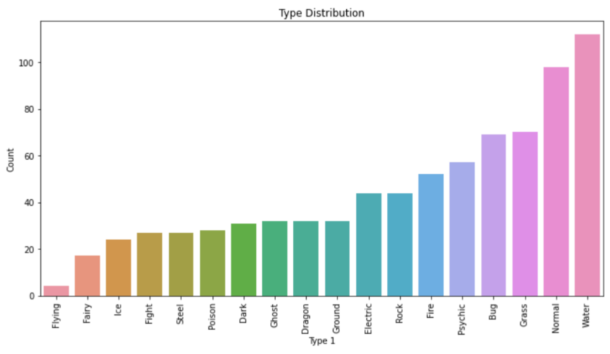


Figure 1: Distribution

have the greatest effect on the results in order to reduce the number of features when building a model is a problem that was a challenge. The idea of assessing the importance of features using these four methods is actually very simple. It is to look at how much each feature contributes, then take an average, and finally make up the contribution between features and to proper parameter tuning to get good accuracy and select the model with highest accuracy. In this project, I did the data cleaning first and then some data visualization. This plot shows the distribution of the type. It's clearly that type water has the most Pokémon in the dataset and normal ranks 2. Flying Pokémon is the rarest type of Pokémon, After completion of the classification of types, used two datasets which has the combats results of the previous matches between Pokémon's. The data engineering part was the most challenging because it was difficult to think what could be done with the type and thought of dropping the feature, but came across a technique of giving weights. So that I did the similar thing and started giving weights by starting everything at 1 and if I found that a particular type of Pokémon is more effective on the other it would multiply the factor by 2 otherwise divide it by 2. By doing this I got weights for every combat based.After creating these function then I created different models for prediction and ran different classifiers , found their accuracy and then jumped to parameter tuning of that classifier whose accuracy was highest .For this I used GridSearchCV where along with Grid Search, cross-validation is also performed. Cross-Validation is used while training the model. Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. Link to github: Pokemon-Fight-Winner-Prediction

Experimental Analysis

But unfortunately I could not increase the accuracy after applying hyperparameter tuning. So I stick with the simple model without any parameter tuning since it had higher accuracy. Since it was difficult to classify type and accuracy appropriately I moved forward with predicting the winner taking into account all the features including type. Results Obtained Using various classifiers. After parsing everything through different Classifiers I found that the random forest is the best classifier that gives the prediction accuracy of 0.95. Followed by Naive Bayes , Support Vector Machine and KNN.Though Decision tree classifier give good results but highest is achieved using Random Forest.Random forests are often the best method to solve for many classification problems. It is fast and adjustable to train. It doesn't need to worry about adjusting a lot of parameters like a support vector machine. It can handle irrelevant features, but it will ignore the correlation of data.Based on that model the test dataset was predicted and classification report was created.

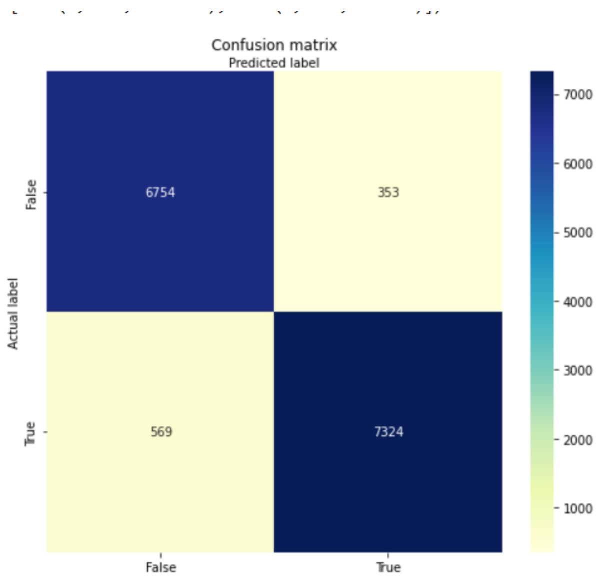
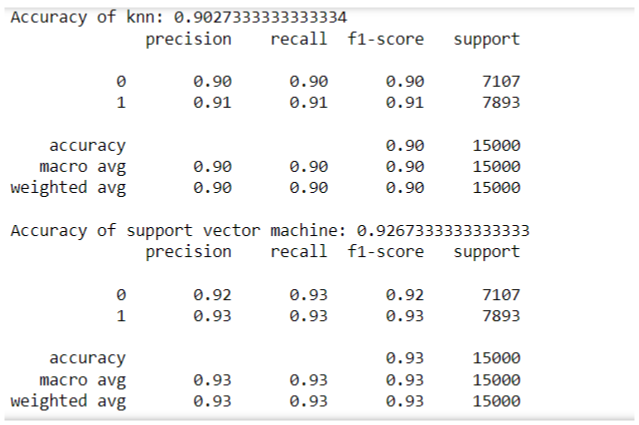
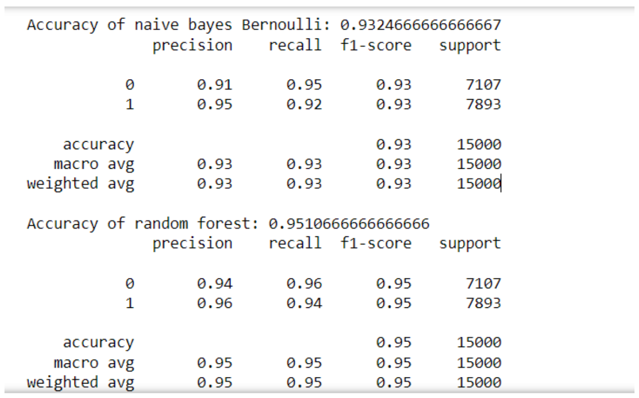


Figure 2: Confusion Matrix



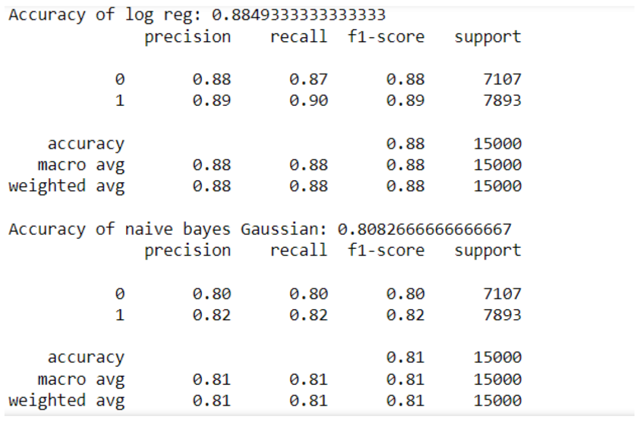
```
Accuracy of knn: 0.9027333333333334
              precision    recall  f1-score   support

           0       0.90      0.90      0.90      7107
           1       0.91      0.91      0.91      7893

    accuracy                           0.90     15000
   macro avg       0.90      0.90      0.90     15000
weighted avg       0.90      0.90      0.90     15000

Accuracy of support vector machine: 0.9267333333333333
              precision    recall  f1-score   support

           0       0.92      0.93      0.92      7107
           1       0.93      0.93      0.93      7893

    accuracy                           0.93     15000
   macro avg       0.93      0.93      0.93     15000
weighted avg       0.93      0.93      0.93     15000
```

Figure 3: support vector machine Classification Report



```
Accuracy of naive bayes Bernoulli: 0.9324666666666667
              precision    recall  f1-score   support

           0       0.91      0.95      0.93      7107
           1       0.95      0.92      0.93      7893

    accuracy                           0.93     15000
   macro avg       0.93      0.93      0.93     15000
weighted avg       0.93      0.93      0.93     15000

Accuracy of random forest: 0.9510666666666666
              precision    recall  f1-score   support

           0       0.94      0.96      0.95      7107
           1       0.96      0.94      0.95      7893

    accuracy                           0.95     15000
   macro avg       0.95      0.95      0.95     15000
weighted avg       0.95      0.95      0.95     15000
```

Figure 4: Random Forest Classification Report



```
Accuracy of log reg: 0.8849333333333333
              precision    recall  f1-score   support

           0       0.88      0.87      0.88      7107
           1       0.89      0.90      0.89      7893

    accuracy                           0.88     15000
   macro avg       0.88      0.88      0.88     15000
weighted avg       0.88      0.88      0.88     15000

Accuracy of naive bayes Gaussian: 0.8082666666666667
              precision    recall  f1-score   support

           0       0.80      0.80      0.80      7107
           1       0.82      0.82      0.82      7893

    accuracy                           0.81     15000
   macro avg       0.81      0.81      0.81     15000
weighted avg       0.81      0.81      0.81     15000
```

Figure 5: Classification Report Logistic of Regression and Naive Bayes

Discussions:

Being a big fan of Pokémon and watching the combats of the different Pokémon's on television, I was curious to know if I could determine the winner of a combat just by looking at the type and different features of the Pokémon.In this project I was only able to predict who the winner could be but failed to predict the type of Pokémon. so that I decided to predict who will be the winner in a Pokémon's battle. It was my first time to do a bit of data engineering and use Machine Learning Algorithms and I did encounter with a lot of difficulties but was able to cope with them and get to the result. Further improvement can be done by using different approach for feature engineering by using win ratio, etc.This Machine Learning model will be helpful for those gaming enthusiast who enjoy games like Pokémon GO which is an interactive, mobile game that uses your phone's GPS data and clock to show Pokémon hidden near your current, physical location. The Pokémon which appear on-screen in the app can be captured. Further this dataset can be solved more efficiently using neural network and reinforcement learning where it will be possible to predict the complete move set and predict winner.Pokemon can also extend to larger applications like robotics. In addition, neural network can help video game developers test new Pokemon and features.since the dataset had lots of missing data it was convenient to use simple classifier like Random forest which indeed did well.

References

[1] Devi Acharya, Rehaf Aljammaz, Beth Oliver, and Mirek Stolee. Gotta generate'em all! pokemon (with deep learning). 2019.

[2] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[3] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.

[4] Tzu-Tsung Wong. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, 48(9):2839–2846, 2015.

[4] [1] [2] [3]