

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokemon Using Multiple ML Algorithms

Become a certified GenAI professional: 10+ Real-world



Home > Classification

> A Beginners Guide to Machine Learning: Binary Classification of Leg...



Siddharth M

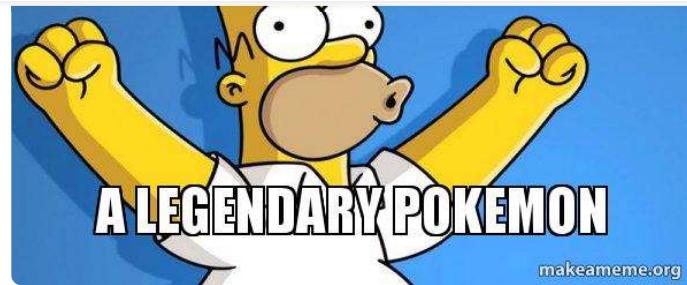
29 Aug, 2023 • 8 min read

Introduction

Machine Learning is widely used across different problems in real-world scenarios. Most machine learning algorithms are applied to large amounts of data. You may have found data classification to be a major problem when working with such datasets. Now, data classification can be either binary or multi-class classification. In this article, we will dive deep into binary classification. We will first understand binary classification and then apply different ML algorithms to see how accurately we can classify the target. What is the target, you ask? Well, how about catching some Pokemon?

For this tutorial, I will be using the Pokemon stats dataset. Here we have stats of all the Pokemon, and we will try to accurately classify if the Pokemon is legendary or not. For those of you who don't know, legendary Pokemon are the ones who are very rare and powerful. So let's go ahead and find out if selected features and the stats help us to classify them correctly.

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokémon Using Multiple ML Algorithms



Source: makeameme.org

This article was published as a part of the [Data Science Blogathon](#)

[Table of Contents](#)

What Is Classification?

In the Machine Learning world, classification refers to separating data into separate class labels. For a particular row in our dataset or values of features, we are interested in associating them with a particular target value. This is widely used in many applications, for example, to find if you got spam mail or not, to identify if an image is fake or not, etc. If there are more than two class labels, we call them multi-class classification.

What Is Binary Classification?

In binary classification, the output of the model is a binary decision: either the input belongs to the positive class, or it belongs to the negative class. This type of classification is widely used in various fields such as finance, healthcare, natural language processing, image recognition, and more.

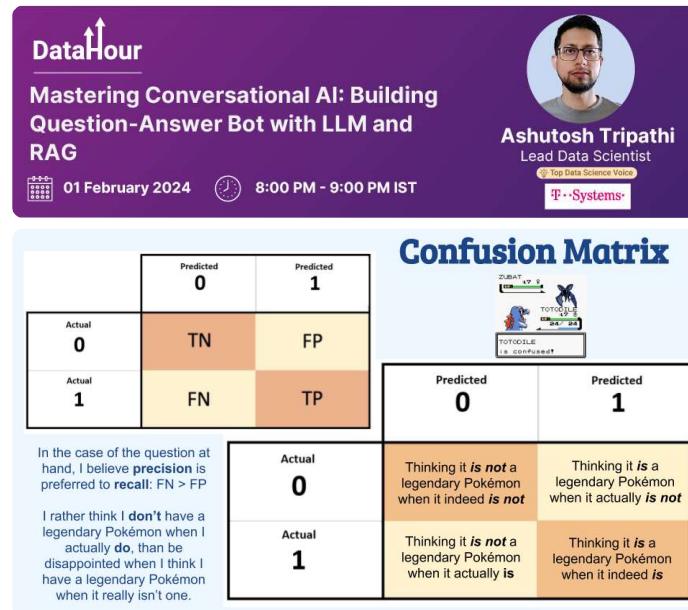
In our case, we deal with only two classes Legendary or Not Legendary Pokémon. So we have a binary classification problem. The question is, how do we

actually measure the accuracy? We have a confusion matrix

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using Analytics Vidhya, you agree to our [Privacy Policy](#) and [Terms of Use](#). [Accept](#)

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokémon Using Multiple ML Algorithms

Pokemon's confusion matrix.



Source: towardsdatascience.com

In the first diagonal of our confusion matrix, we can find the values that are perfectly predicted. Adding them and dividing them with all values inside the matrix gives us prediction accuracy.

Now let's dive deep into different machine learning algorithms. In this tutorial, I intend to show you the coding of each algorithm. The complete pre-processing and encoding with all codes are available in this [COLAB link](#).

Our Dataset

The dataset is available [here](#).

For the classification problem, we have used this dataset with a Legendary column that tells us if the Pokémon is legendary or not with True or False. We use Label encoding to encode True as 1 and False as 0 before jumping into the next steps. Columns #, Name, Type 1, and Type 2 are not required and are removed from the dataset as part of the preprocessing step. The final dataset used for ML algorithms looks like this:

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokemon Using Multiple ML Algorithms

4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False
---	---	------------	------	-----	-----	----	----	----	----	----	----	---	-------

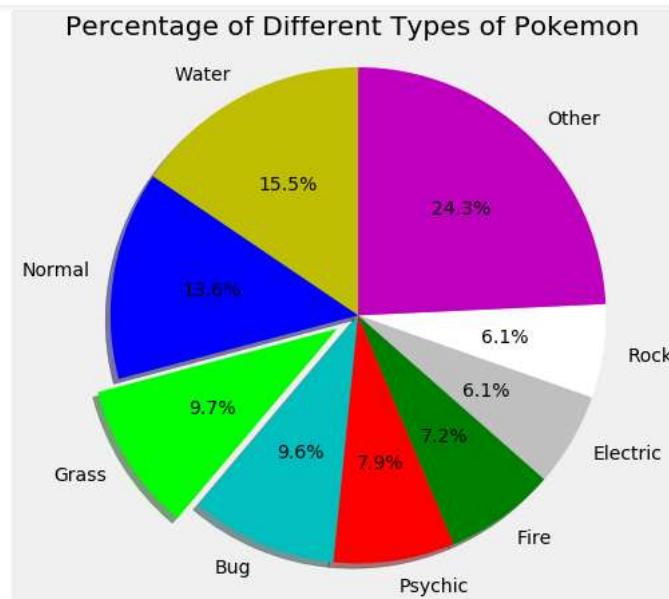
As you can see here, various features are listed across 8 columns. The last column, 'Legendary', stands as the target variable to be predicted.

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokémon Using Multiple ML Algorithms

 main.py

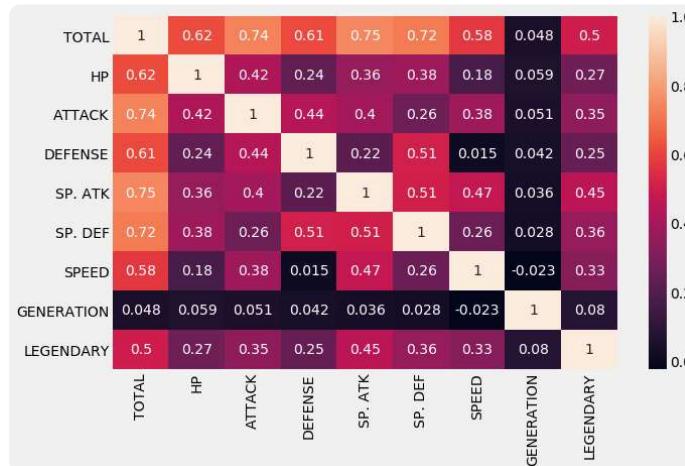
```
1 import pandas as pd
2
3 df = pd.read_csv('Pokemon.csv')
4 df.head()
5
6 df.drop(columns=['#', 'Name', 'Type 1', 'Type
7 2'], inplace=True)
8 print(df)
```

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokémon Using Multiple ML Algorithms



As we can see here, Water Pokémons are ubiquitous compared to all the rest. Rock and electric Pokémons are found less.

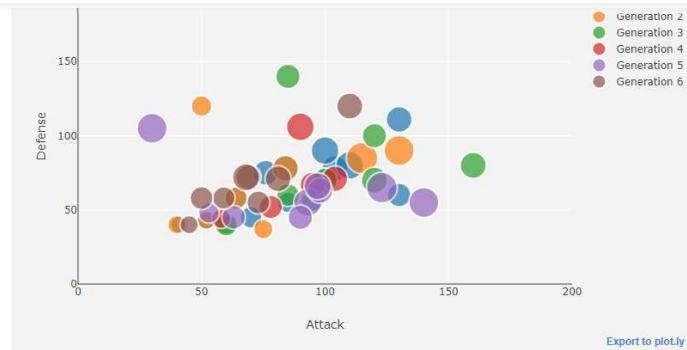
How Correlated Are Each Pokémon Features?



The heatmap above shows that there is not much correlation between the attributes of the Pokémons. The highest we can see is the correlation between Special Attack and the Total.

Attack vs. Defence for Fire Pokémons in Each Generation

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokémon Using Multiple ML Algorithms



Generation 5 tends to have a lower defense. One of the 5th gen Pokémon is the best attacker.

Machine Learning Algorithms

Let us use some of the most common machine learning algorithms in our binary classification and see how they work.



Source: miro.medium.com

Logistic Regression

Logistic regression is widely used for binary classification. It uses the logit function for the outcome. It generates a probability in the output and classifies the data into 0 or 1 using the sigmoid activation function. The sigmoid function is given as:

$$Y = 1 / (1 + e^{-z})$$

```
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
```

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using Analytics Vidhya, you agree to our [Privacy Policy](#) and [Terms of Use](#). [Accept](#)

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokemon Using Multiple ML Algorithms

The accuracy score achieved using Logistic Regression is:

93.12 %

Gaussian Naive Bayes

Naive Bayes is a probabilistic algorithm that makes use of the Bayes Theorem. We can give it as,

If B is true, the probability of A is equal to the probability of B. If A is true, multiplied by the probability of A is true, divided by the probability of B is true.

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(X_train,Y_train)
Y_pred_nb = nb.predict(X_test)
score_nb = round(accuracy_score(Y_pred_nb,Y_test)*100,2)
print("The accuracy score we have achieved using Naive Bayes is
```



The accuracy score achieved using Naive Bayes is: 91.88 %

Support Vector Machines

SVM are supervised ML algorithms that are used to solve classification problems. We draw a hyperplane trying to separate two different classes here. With more complex data, we can produce better results using the SVM algorithm. The algorithm can train large datasets but tends to be slower in nature.

```
from sklearn import svm
sv = svm.SVC(kernel='linear')
sv.fit(X_train, Y_train)
Y_pred_svm = sv.predict(X_test)
score_svm = round(accuracy_score(Y_pred_svm,Y_test)*100,2)
print("The accuracy score we have achieved using Linear SVM is: '
```



The accuracy score achieved using Linear SVM is: 94.38 %

K-Nearest Neighbours

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokémon Using Multiple ML Algorithms

in KNN, and the neighbor near the points wins the point. K here denotes the number of neighbors that are available in our model.

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train,Y_train)
Y_pred_knn=knn.predict(X_test)
score_knn = round(accuracy_score(Y_pred_knn,Y_test)*100,2)
print("The accuracy score we have achieved using KNN is: "+str(score_knn))
```

The accuracy score achieved using KNN is: 96.25 %

Decision Tree

Decision trees are similar to a question-answering system, which decides which child to place the points based on a specific condition. It basically acts as a flow chart, splitting the data points into two categories at a time, from “trunk” to “branches,” then “leaves,” where the data within each category is split based on similarity.

```
from sklearn.tree import DecisionTreeClassifier
max_accuracy = 0
for x in range(200):
    dt = DecisionTreeClassifier(random_state=x)
    dt.fit(X_train,Y_train)
    Y_pred_dt = dt.predict(X_test)
    current_accuracy = round(accuracy_score(Y_pred_dt,Y_test)*100,2)
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x
#print(max_accuracy)
#print(best_x)
dt = DecisionTreeClassifier(random_state=best_x)
dt.fit(X_train,Y_train)
Y_pred_dt = dt.predict(X_test)
score_dt = round(accuracy_score(Y_pred_dt,Y_test)*100,2)
print("The accuracy score we have achieved using Decision Tree is: "+str(score_dt))
```

The accuracy score achieved using the Decision Tree is:

96.25 %

Random Forest

Random forest expands the decision tree and mainly fixes the decision tree's drawback of unnecessarily forcing data points into a somewhat incorrect category. It works by

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokemon Using Multiple ML Algorithms

the nearest tree on the data scale.

```
from sklearn.ensemble import RandomForestClassifier
max_accuracy = 0

for x in range(2000):
    rf = RandomForestClassifier(random_state=x)
    rf.fit(X_train,Y_train)
    Y_pred_rf = rf.predict(X_test)
    current_accuracy = round(accuracy_score(Y_pred_rf,Y_test)*1
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x
    #print(max_accuracy)
    #print(best_x)
    rf = RandomForestClassifier(random_state=best_x)
    rf.fit(X_train,Y_train)
    Y_pred_rf = rf.predict(X_test)
    score_rf = round(accuracy_score(Y_pred_rf,Y_test)*100,2)
    print("The accuracy score we have achieved using Decision Tree
```

The accuracy score achieved using the Decision Tree is:

98.12 %

XG-Boost

XGBoost is mainly an implementation of gradient-boosted decision trees used for speeding upscaling the performance in classification.

```
import xgboost as xgb
xgb_model = xgb.XGBClassifier(objective="binary:logistic", rand
xgb_model.fit(X_train, Y_train)
Y_pred_xgb = xgb_model.predict(X_test)
score_xgb = round(accuracy_score(Y_pred_xgb,Y_test)*100,2)
print("The accuracy score we have achieved using XGBoost is: "+
```

The accuracy score achieved using XGBoost is: 96.88 %

Neural Network

Neural networks are networks that mimic the human brain. Here we have constructed a neural network with 32 layered hidden layers. Since we have 8 features, we take in as input dimensions. In the last layer, we use sigmoid as it is a binary classification problem. In between, we use

ReLU as the activation function

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using Analytics Vidhya, you agree to our [Privacy Policy](#) and [Terms of Use](#). [Accept](#)

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokemon Using Multiple ML Algorithms

```

model = Sequential()
model.add(Dense(32,activation='relu',input_dim=8))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
model.fit(X_train,Y_train,epochs=100, callbacks = tf.keras.callbacks.EarlyStopping(monitor='val_accuracy', patience=10))
Y_pred_nn = model.predict(X_test)
rounded = [round(x[0]) for x in Y_pred_nn]
Y_pred_nn = rounded
score_nn = round(accuracy_score(Y_pred_nn,Y_test)*100,2)
print("The accuracy score we have achieved using Neural Network")
    
```



The accuracy score achieved using the Neural Network is:

90.62 %

The Results

After running our Pokemon features on various ML algorithms, we found that the XG-boost algorithm works well in our case with 96.88% accuracy, followed by Random forest with 96.12%. Neural Networks aren't that great for this current problem. But we may get better results by playing with different hidden layers or using some complex models.



Source: heavy.com

Conclusion

In this article, we saw how each element in a dataset can be classified into one of two categories. We did the exploratory analysis of the data and then applied various machine learning algorithms to predict the classification. We saw how logistic regression, Naive Bayes, support vector machine, decision trees, K-nearest neighbor,

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokemon Using Multiple ML Algorithms

If this article has sparked your interest in the topic of data classification and you wish to learn about it in-depth, then our [BlackBelt program](#) is here to help you. This advanced program empowers you to master binary classification, enhancing your proficiency in data classification and analytics.



Source: www.usgamer.net

Frequently Asked Questions

Q1. What is binary classification with an example?

Ans. Binary classification is a machine learning concept that categorizes data instances into one of two possible classes or categories. For example, in credit card fraud detection, a transaction can be either classified as fraud or non-fraud (legitimate). Since only 2 possible classes exist, it is a binary classification problem.

Q2. What is binary classification in logistic regression?

Ans. Binary classification in logistic regression is a specific application of the logistic regression algorithm to solve problems where the goal is to classify data into one of two possible classes.

Q3. What is binary classification number of classes?

Ans. The number of classes in binary classification is two.

Q4. How do you calculate binary classification?

Ans. The process of calculating binary classification

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokemon Using Multiple ML Algorithms

The media shown in this article are not owned by Analytics Vidhya and are used at the Author's discretion.

[binary classification](#)
[blogathon](#)
[Data classification](#)
[machine learning](#)


Siddharth M

29 Aug 2023

[Classification](#)
[Intermediate](#)
[Machine Learning](#)
[Project](#)
[Python](#)

Responses From Readers

What are your thoughts?...

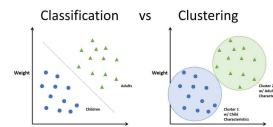
[Submit reply](#)

Related Articles

REGRESSION V/S CLASSIFICATION

Regression vs Classification in Machine Learning Explained!

Regression vs Classification: Difference between classification and



Classification vs. Clustering- Which One is Right for Your Data?

This article brings the best Classification vs. Clustering debate so that

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokemon Using Multiple ML Algorithms



[avcontentteam](#)
27 Jun, 2023 • 7 min
read



[avcontentteam](#)
27 Jun, 2023 • 9 min
read



A Complete guide to Understand Classification in Machine Learning

Classification is a task of ML which assigns a label value to a specific class

blogathon

classification



[Arnab Mondal](#)
05 Apr, 2023 • 8 min
read

Scribe, Shine, Succeed →

Write, captivate, and earn accolades and rewards for your work

- ✓ Reach a Global Audience
- ✓ Get Expert Feedback
- ✓ Build Your Brand & Audience
- ✓ Cash In on Your Knowledge
- ✓ Join a Thriving Community
- ✓ Level Up Your Data Science Game

A Beginners Guide to Machine Learning: Binary Classification of Legendary Pokémon Using Multiple ML Algorithms

Company	Discover
About Us	Blogs
Contact Us	Expert session
Careers	Podcasts
	Comprehensive Guides
Learn	Engage
Free courses	Community
Learning path	Hackathons
BlackBelt program	Events
Gen AI	Daily challenges
Contribute	Enterprise
Contribute & win	Our offerings
Become a speaker	Case studies
Become a mentor	Industry report
Become an instructor	quexto.ai

Download App



GET IT ON
Google Play



Download on the
App Store

[Terms & conditions](#) • [Refund Policy](#) • [Privacy Policy](#) •
[Cookies Policy](#) © Analytics Vidhya 2023. All rights reserved.