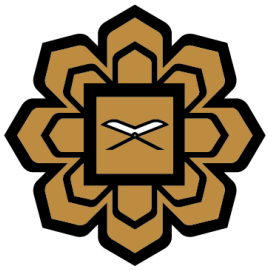


الجامعة الإسلامية العالمية ماليزيا
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA
يونيۏرسيتي إسلام انتارابغسا ملسيا
Garden of Knowledge and Virtue

Date: 06-Dec-2020

[ASSIGNMENT- REPORT]

MACHINE LEARNING

CODE: CSC 3304, SEC: 01

LECTURER: Dr. AMIR 'AATIEFF BIN AMIR HUSSIN

Name: Hasan Tanveer Mahmood

Matric No: 1725413

Table of Content

1. Title: Predict whether an animal is Predator.

2. Introduction:

Machine learning is basically a type of Artificial Intelligence that allows a system to learn from data rather than by programming directly. Although machine is not more intelligent than human, but machine can calculate data faster than human. Using some sort of calculation machine can detect or predict the results which is such difficult for human to think. In addition, Machine learning allows models before being deployed to train on data sets. Machine learning algorithm worked based on train data set, where developer create a model using some training data based on that this model can take some input and predict based on the training data. Sometime prediction can achieve 100 percent of accuracy.

In this paper I am going to develop a model using supervised learning which can predict the animal, whether is a predator or not. It will use some data from a dataset to make a prediction. At present, with the development of machine learning algorithm, more predictive methods such as Logistic Regression, Linier Regression, K-NN, Decision Tree, Random Forest etc, have been populated to make the prediction. So, it is difficult to classify the animals whether its venomous, predator or domestic. However, this model can help to predict the type of animal. So that it will be easier to distinguish the types. Creation of this kinds of model can play a significant role to zoo organization.

Moreover, in this model **Random Forest** machine algorithm will be applied. Dataset is taken from kaggle.

3. Methodology:

In this model **Random Forest Regression** machine algorithm will be applied. It is a supervised learning algorithm which using the approach of ensemble learning for classification and regression. The are no inter relation between these trees while construct the trees. It works by building a multitude of decision trees at training time and give output which is mode of classification and regression. It can combine the multiple prediction. I have decided to use **Random forest** method because compare to other algorithm Random Forest gives more accuracy in my dataset.

Decision trees are sensitive to work with my dataset. In my case, the resulting of decision tree can be quite different if the training data is altered and the predictions can also be slightly different in turn. In addition, it also costly to train computationally, bear a great risk of overfitting, and appear to find local optima because after splitting it cannot be undone. (Chakure 2019 P.4),

On the other hand, the major limitation of logistic regression is more than Random forest and logistic regression can only used to predict the discreate function. In my case it will give less accuracy.

To avoid these weaknesses, I am intended to use **Random Forest** which has the ability to integrate multiple decision trees into one model.

3.1 Dataset :

I have collected my dataset from kaggle.

3.2 Tools :

Jupyter Notebook (Anaconda), Spyder (Anaconda)

3.3 Machine Learning :

In this model I am using Random Forest Regression. It basically trained with the bagging method. The basic idea of this bagging method is that a combination of learning models to increase overall results.

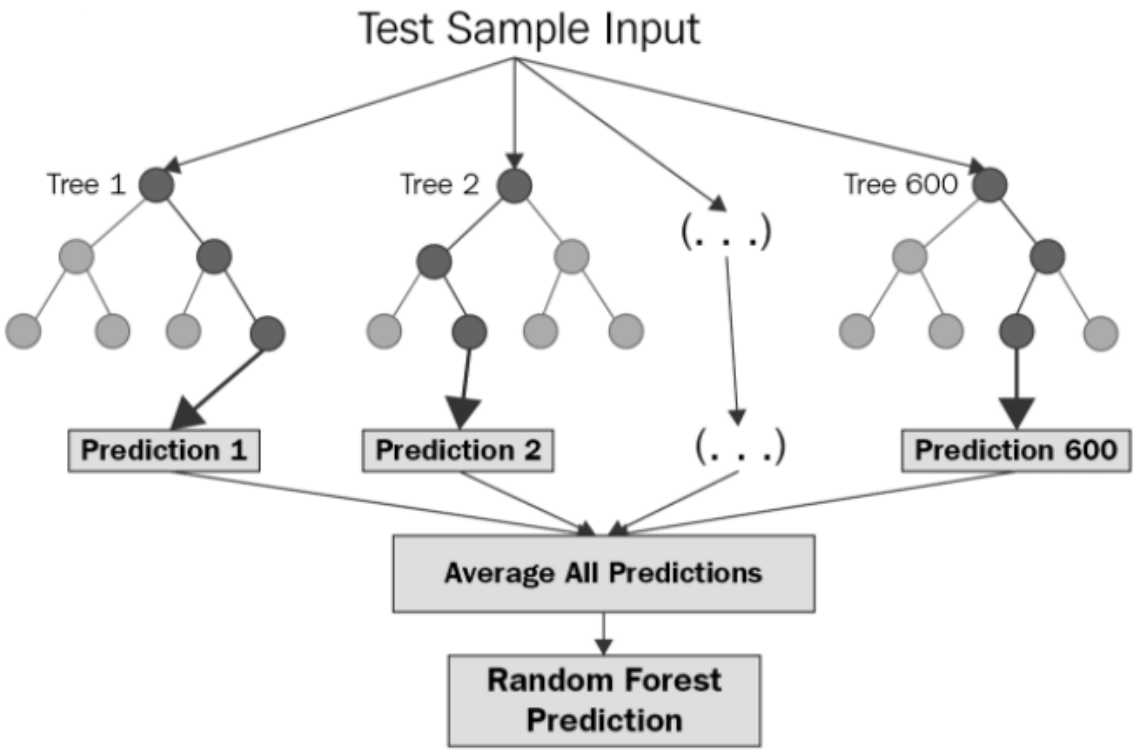


Figure 1: Image source

Formally, the terminologies of the Random Forest Regression are as follows:

1. The number of features which one is selected that can be split on at each node that is limited to some percentage of the whole.

2. Each tree is drawn by random sampling from the original dataset when splitting, adding the next element of randomness which is preventing the overfitting.

3.4 Feature Selection:

Feature selection is a process of eliminating unnecessary columns of the dataset. Feature selection is very important. To design my model, I have to decide which one my target variable and which one I can select to set my train data. However, I will not take all the feature (columns). I will choose only those features which has some relation with the target variable.

So, in given dataset I can see that feature **('Domestic', 'Backbone', 'Breathes'**) has some relation with target variable (**'Predator')**. For instance, form dataset we can simply identified that **which animals are domestic they are not predator.**

I have set my:

Feature columns = ['Domestic', 'Backbone', 'Breathes'] as my x variable

Target Column = ['Predator'] as my y variable.

3.5 Data Processing:

In given dataset I checked value whether there is missing value or not. But luckily, I do not have any missing values.

I have checked my missing values using **dataset.info().**

Then I have set my train and test data with the ratio of 75% and 25%.

3.6 Modeling:

1. import necessary library.
2. import dataset using pandas.
3. Set feature columns and target columns.
4. Split data and Set the ration 75:25 for training and testing.
5. Apply Random Forest regression Algorithm
6. Construct confusion matrix and calculate the accuracy.

This how I construct my model using random forest.

```
#Random Forest
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier

max_accuracy = 0


for x in range(105):
    rf = RandomForestClassifier(random_state=x)
    rf.fit(x_train,y_train)
    y_pred = rf.predict(x_test)
    current_accuracy = round(accuracy_score(y_pred,y_test)*100,2)
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x

rf = RandomForestClassifier(random_state=best_x)
rf.fit(x_train,y_train)

y_pred = rf.predict(x_test)
```

This is how I code for getting the confusion matrices

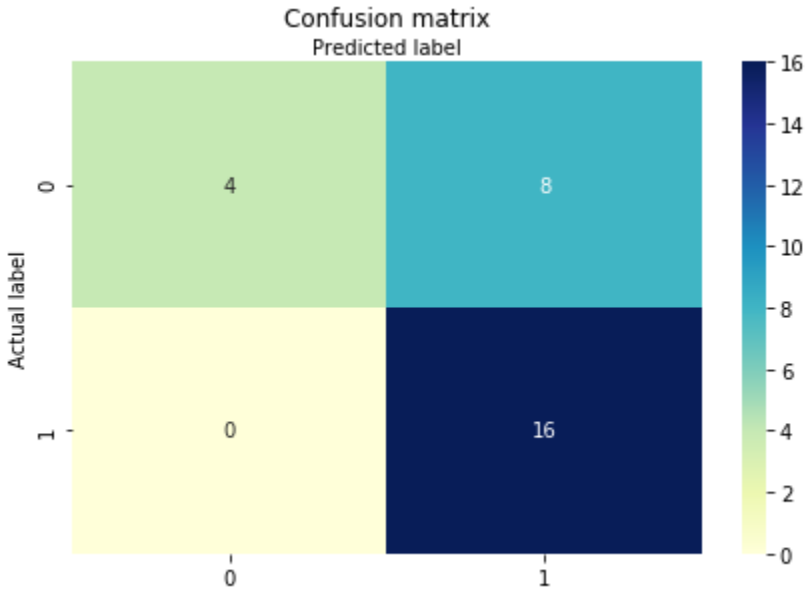
```
#For visualizing this library need to import
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
# This is the name of classes.
class_names = [0,1]
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
sns.heatmap(pd.DataFrame (cnf_matrix), annot=True, cmap="YlGnBu", fmt = 'g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

4. Results :

Figure bellow shows the confusion matrix that was produced by Random Forest Algorithm. From this confusion matrix we are able to conduct performance evaluation by calculation the accuracy, precision and recall for the predictive model

N = 28	Predicted 0	Predicted 1
Actual 0	True Positive	False Negative
Actual 1	False positive	True Negative

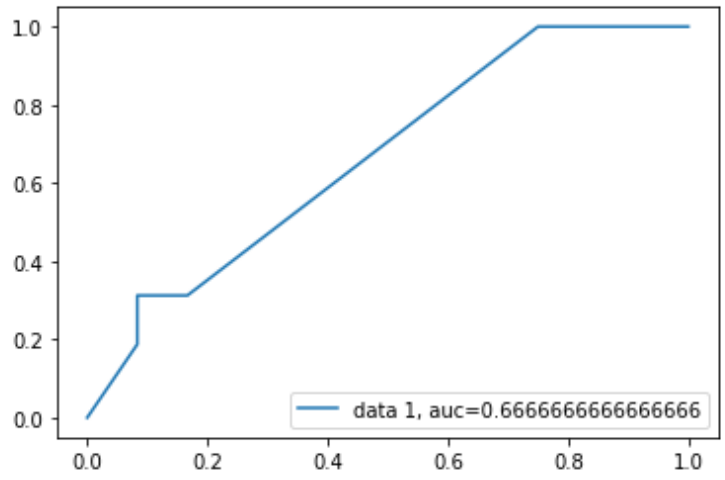


So, from this confusion matrix we can state that out of 28 animals, classifiers predict yes 24 times and no 4 times. In reality 16 animal in sample is the predator and 12 is not a predator.

To improve my algorithm performance, I tried with the logistic regression method but then I got the less accuracy than Random forest. In logistic regression I tuned the values and, parameter. But got the same result.

```
logreg = LogisticRegression(penalty = 'l2', C = 1,random_state = 0)
```

```
import sklearn.metrics as metrics
from matplotlib import pyplot as plt
y_pred = logreg.predict_proba(x_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred)
auc = metrics.roc_auc_score(y_test, y_pred)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()
```



To Improve My Algorithm, I did these necessary steps :

1. Firstly, I choose more columns as feature data. Which is **'Toothed', 'Fins', 'Domestic', 'Backbone', 'Breathes', 'Class_type'** Then I got less accuracy. So that I reduce the feature columns into three.

2. Secondly, I tried with different ratio of train data and test data. For instance, I tried 30:70 .with this ratio I get less accuracy. Then I tried with 25:75 then I got more accuracy.

5. Discussion:

Using this confusion matrix, we can easily identify the results. From my dataset I got this result bellow.

Accuracy: (TP + TN)/ N = (4+16)/28 = 0 .7142
Precision: (TP/ Predict Yes) = 16/24 = 0.6667
Recall : TP/(TP+FP) = 4 / (4+0) = 1.0
Sensitivity : TP/(TP+FN) = 4/ (4+8) = 0.33
Specificity : TN/(TN+FP) = 16/(16+0) = 1.0

Which I got exactly from my code.

```
print("Acuracy : ", metrics.accuracy_score(y_test,y_pred))
print("Precision: ", metrics.precision_score(y_test, y_pred))
print("Recall: ", metrics.recall_score(y_test, y_pred))
```

```
Acuracy :  0.7142857142857143
Precision:  0.6666666666666666
Recall:  1.0
```

```
sensitivity1 = cnf_matrix[0,0]/(cnf_matrix[0,0]+cnf_matrix[0,1])
print('Sensitivity : ', sensitivity1 )

specificity1 = cnf_matrix[1,1]/(cnf_matrix[1,0]+cnf_matrix[1,1])
print('Specificity : ', specificity1)
```

```
Sensitivity :  0.3333333333333333
Specificity :  1.0
```

So, in overall, Accuracy I got is 71% which means how open is the classifier is correct. That is quite bad. For this dataset it can be acceptable. Because this data set does not have enough rows and columns to predict with the accuracy of 100%. For instance, it cannot be classified whether the animal is predator or not by using such columns milks, tooth, fins etc. Cause these columns can not help to predict whether the animal is predator or not.

And Precision I got is 66%. Which means when the model predicts yes how often its correct.

And Recall is 100% . Which means the ability of a classifier can identify all relevant instances.

And Sensitivity is 33%, Which has correctly identified the animal is predator.

And Specificity is 100% which has correctly identified the animal is not a predator.

But for any other prediction this model can show more than 97% accuracy.

6. Conclusion:

To sum up, we have learned how to work with supervised learning. How to predict accurately. In this project there also some challenges. For instance, A animal whether predator or not it can not be identified by its legs, fins, tooth, backbone etc. But there is one column which has the inter relation as the animal can be predator or not. In data set I found that which animals are domestics they are not predator and some other features such as backbone, breathes has little relation with the target variable predator.

However, this model is able to get 100% of accuracy on another dataset. And also, can predict better than this data. Random forest regression provides high accuracy than other regression algorithm.

Random Forest regression can use when the data has a non-linear trend moreover, extrapolation outside the training data is not important. And it is better to not use Random Forest when the dataset has time series form. Because, time series data require identification of a growing or decreasing trend that a Random Forest Regressor can to formulate.

Using this supervised learning knowledge, in future this model can be used with different purposes such as predicting **Coved 19**, **Heart disease, and any other predictable** model. To get more accuracy with this dataset and different regression algorithm can see here.