Data Science East africa

Data Science Hacks, Tips and Tricks

Day 20/20

1. Visualization of Trees.

While working with any Machine Learning model, we only know which model will be suitable for datasets to solve a particular problem and based on datasets, we set the parameters of the model with some Mathematics/Logical reasoning. The model works completely fine and severs the purpose. But what inside that black-box no one cares. It sounds less appealing, right? Well now for any tree model it's not. Let's see how can we visualize the black-box.

We will import some necessary libraries to setup up our selves for the task!

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier,plot_tree
from sklearn.datasets import load_wine
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

We will use the famous wine data set for this purpose. Loading datasets in Pandas Data-frame and separating into predictor and response variables.

```
wine=load_wine()
df = pd.DataFrame(data= np.c_[wine['data'], wine['target']],
                  columns= wine['feature_names'] +
['target'])
X = df.drop('target',axis=1)
y = df["target"]
features = df.columns[:-1].values.tolist()
```

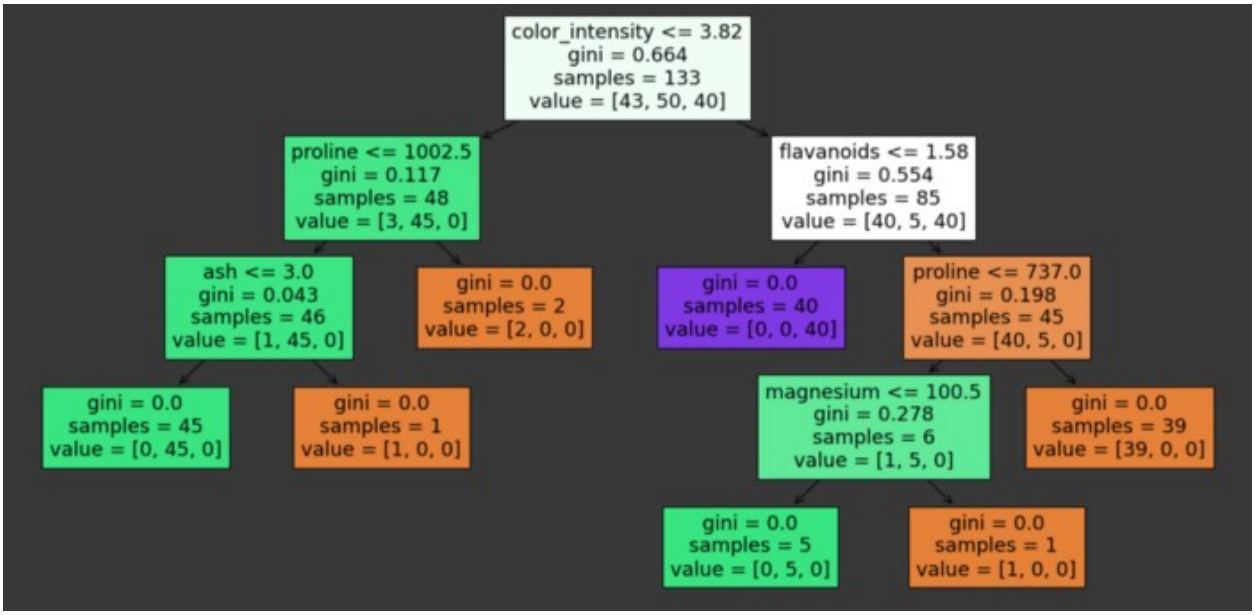
Splitting data into training and testing sets, selecting the appropriate Model and fitting the model.

```
X_train, X_test, y_train, y_test = train_test_split(X,
y,random_state = 2020)

model = DecisionTreeClassifier()distinguish between two things

model.fit(X_train, y_train)
```

We are going to use the library from sklearn.(import sklearn.tree.plot_tree)

```
plt.figure(figsize = (20, 10))
plot_tree(model, feature_names = features, filled = True)
```

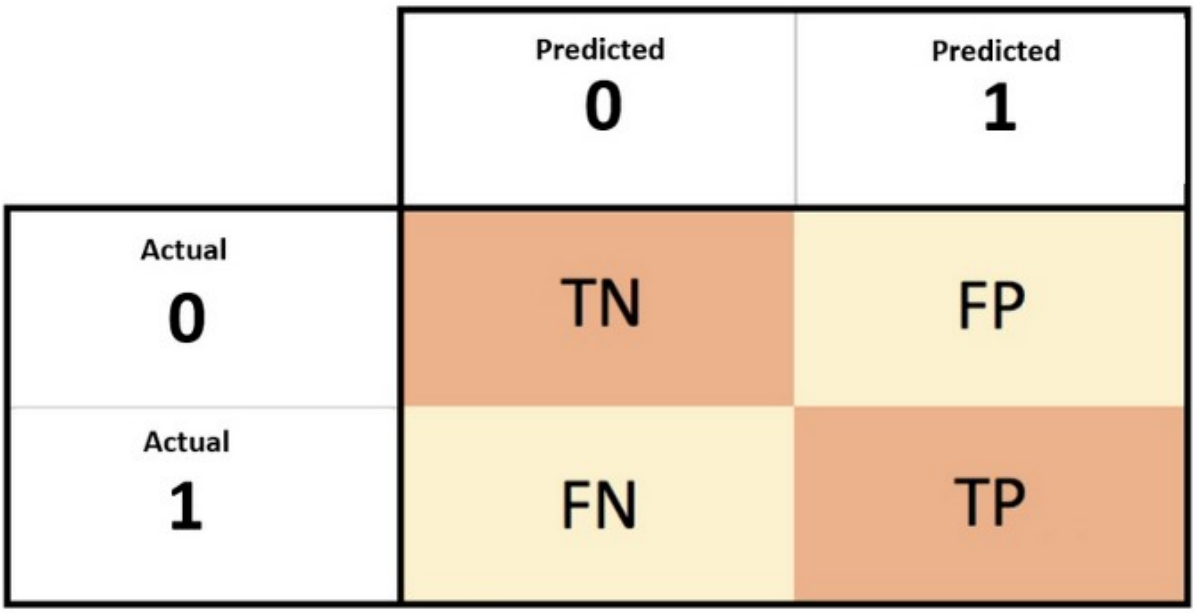
Output:



It tells us about samples(number of tuples), feature in each node, the number of nodes, criterion on which we made tree (here, by default, we have used Gini Index).

Refer to the documentation for more details and examples.

2.Plot confusion matrix.



Confusion Matrix, the name itself contains confusion in it. so why is that? answer to it seems confusing while understanding for the first time but once you have understood it will surely help make a fruitful decision about your model.

Note:- The representation of the confusion matrix can be different in any other source.

Importing some necessary Libraries.

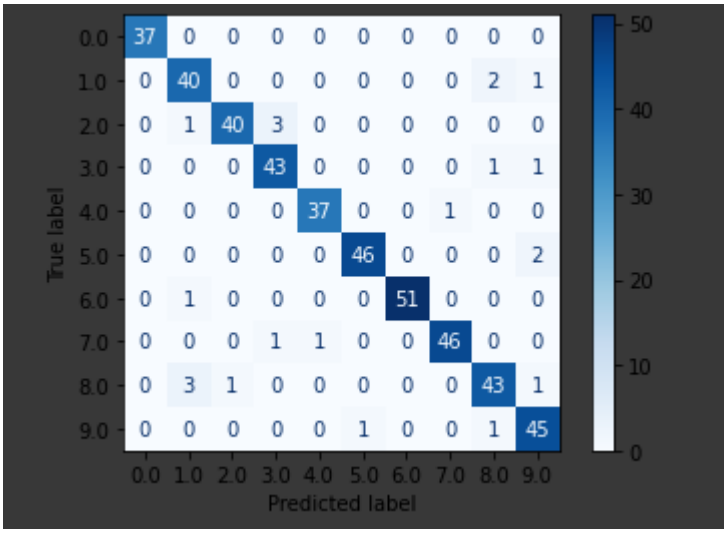
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import plot_confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import load_digits
```

We will use the Digit data set for this purpose. Loading datasets in Pandas Data-frame and separating into predictor and response variables.

```
db=load_digits()
df = pd.DataFrame(data= np.c_[db['data'], db['target']])
df=df.rename(columns={64:'target'})
X = df.drop('target',axis=1)
y = df['target']
```

Splitting data into training and testing sets, selecting the appropriate Model and fitting the model.

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
random_state = 0)
model = LogisticRegression(random_state = 2020)
model.fit(X_train, y_train)
disp = plot_confusion_matrix(model, X_test, y_test, cmap =
"Blues")
```



3. Plotting ROC curves in a single plot.

ROC (Receiver Operating Characteristic) Curve tells us about how good the model is in segregating classes into two or more. Better models can accurately segregate between the two or more. Whereas, a poor model will have difficulties in segregating between the two.

-Format your Code using Black.

Just imagine living in a magical house that automatically cleans itself when you wake up. The bedsheet is folded, the dishes are done and you can enjoy your morning coffee. Doesnt that sound awesome?

That is what black can do with your code! Black calls itself the uncompromising code formatter which I believe is because it has made life simpler for me as well my colleagues reading my code.

Black is an automatic code formatter for Python, therefore you just write code in your style and then black formats it into a consistently formatted code. This really helps in focusing on the content rather than the structure. Also, it makes code review faster. You can check out this cool playground which showcases the power of black.

Follow these steps to format your code automatically:

1. Save your Python file with .py extension.
2. Go to the terminal and type: black [filename.py]
3. Congrats! Your file is now formatted

- Python Generators or List Comprehensions.

The Python generator yields one item at a time and generates them only when in demand. Generators, hence, are much more memory efficient. Have you used it in your daily programming cycle yet? If not, then I'll try to give you reasons to include it in your practice through this code where I have compared the memory usage and time consumption of list comprehension and Python generators!

- Lazy import data science libraries using Pyforest.

Have you been in a situation where you feel youre spending way too much time thinking of which libraries to import at the start of any data science project? It happens to most of us!

I have some awesome news for you PyForest is the solution for all your library importing woes!

PyForest imports all the popular data science libraries to your work environment ONLY when you need them. This is known as lazy import in Python. Dont worry, it doesnt import all the list of libraries at once. It wont import the libraries or functions of libraries you havent imported.

Regards,

Data Science East Africa Team.