01-Decision Trees and Random Forests in Python

March 28, 2023

1 Decision Trees and Random Forests in Python

1.1 Import Libraries

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

1.2 Get the Data

[2]:
```
df = pd.read_csv('kyphosis.csv')
```

[3]:
```
df.head()
```

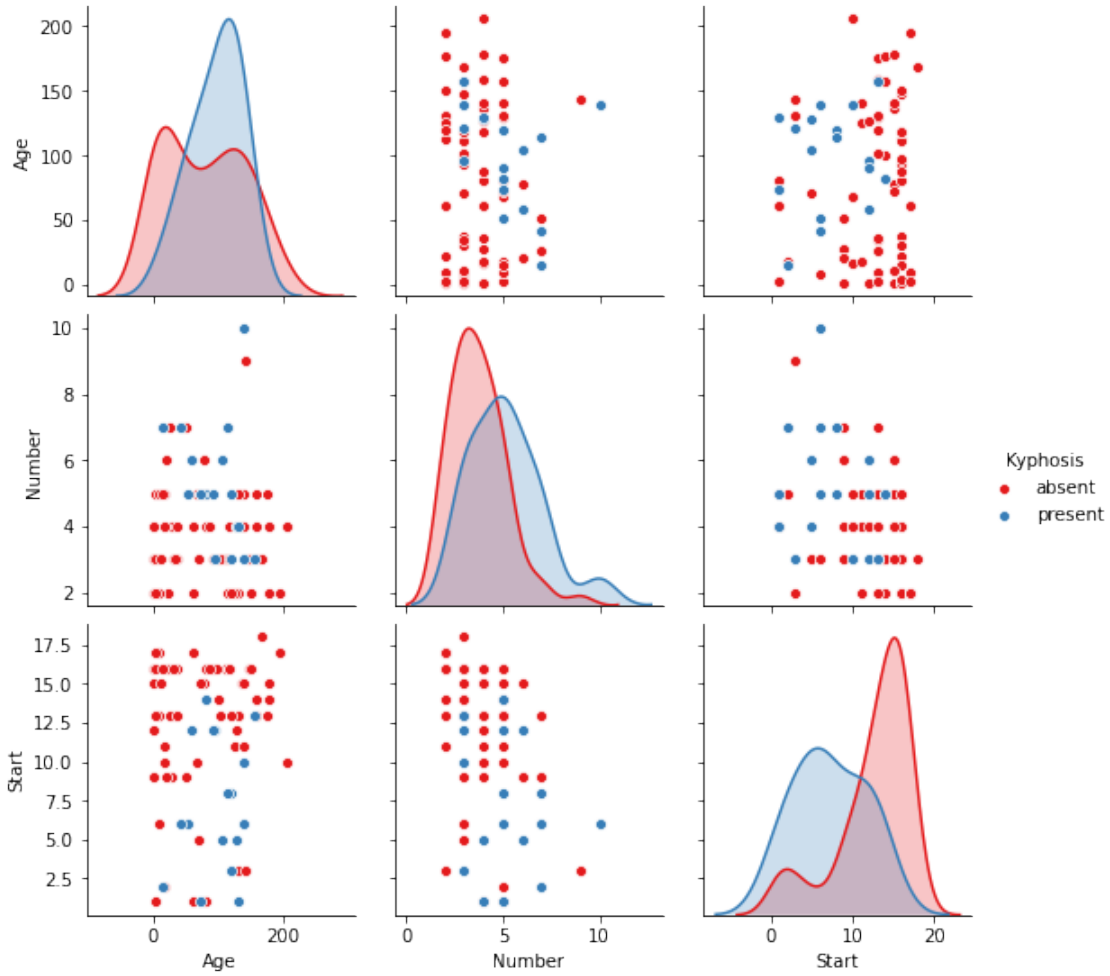
[3]:

	Kyphosis	Age	Number	Start
0	absent	71	3	5
1	absent	158	3	14
2	present	128	4	5
3	absent	2	5	1
4	absent	1	4	15

1.3 We'll just check out a simple pairplot for this small dataset.

[4]:
```
sns.pairplot(df,hue='Kyphosis',palette='Set1')
```

[4]:
```
<seaborn.axisgrid.PairGrid at 0x7f7eb12b4950>
```



1.4 Train Test Split

Let's split up the data into a training set and a test set!

[5]:
```
from sklearn.model_selection import train_test_split
```

[6]:
```
X = df.drop('Kyphosis',axis=1)
y = df['Kyphosis']
```

[7]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.
↪30,random_state=101)
```

1.5 Decision Trees

We'll start just by training a single decision tree.

[8]:
```
from sklearn.tree import DecisionTreeClassifier
```

[9]:
```
dtree = DecisionTreeClassifier()
```

[10]:
```
dtree.fit(X_train,y_train)
```

[10]:
```
DecisionTreeClassifier()
```

1.6 Prediction and Evaluation

Let's evaluate our decision tree.

[11]:
```
predictions = dtree.predict(X_test)
```

[12]:
```
from sklearn.metrics import classification_report,confusion_matrix
```

[13]:
```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

      absent       0.67      0.71      0.69        17
     present       0.29      0.25      0.27         8

    accuracy                           0.56        25
   macro avg       0.48      0.48      0.48        25
weighted avg       0.54      0.56      0.55        25
```

[14]:
```
print(confusion_matrix(y_test,predictions))
```

```
[[12  5]
 [ 6  2]]
```

1.7 Tree Visualization

Scikit learn actually has some built-in visualization capabilities for decision trees, you won't use this often and it requires you to install the pydot library, but here is an example of what it looks like and the code to execute this:

[16]:
```
#from IPython.display import Image
#from sklearn.externals.six import StringIO
#from sklearn.tree import export_graphviz
#import pydot

#features = list(df.columns[1:])
#features
```

[17]:
```
#dot_data = StringIO()
#export_graphviz(dtree,␣
 ↪out_file=dot_data,feature_names=features,filled=True,rounded=True)
```

```
#graph = pydot.graph_from_dot_data(dot_data.getvalue())
#Image(graph[0].create_png())
```

1.8 Random Forests

Now let's compare the decision tree model to a random forest.

[18]:
```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=100)
rfc.fit(X_train, y_train)
```

[18]:
```
RandomForestClassifier()
```

[19]:
```
rfc_pred = rfc.predict(X_test)
```

[20]:
```
print(confusion_matrix(y_test,rfc_pred))
```

```
[[17  0]
 [ 6  2]]
```

[21]:
```
print(classification_report(y_test,rfc_pred))
```

```
              precision    recall  f1-score   support

      absent       0.74      1.00      0.85        17
     present       1.00      0.25      0.40         8

    accuracy                           0.76        25
   macro avg       0.87      0.62      0.62        25
weighted avg       0.82      0.76      0.71        25
```

1) Explicar, estatisticamente, a diferença do modelo Decision Tree e Random Forest

2) Comparar os resultados apresentados pelos dois métodos

3) Explicar que tipo de dados são analisados por Random Forest

4) Qual é a diferença entre Random Forest e Knn, uma vez que ambos utilizam matriz de confusão?

[ ]:
```

```