

```

In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set()
from IPython.display import display

In [2]: titanic = pd.read_csv('titanic.csv')

In [3]: all_data = [titanic]

In [4]: for dataset in all_data:
    dataset.loc[ dataset['Age'] <= 12, 'Age'] = 0,
    dataset.loc[(dataset['Age'] > 12) & (dataset['Age'] <= 20), 'Age'] = 1
    ,
    dataset.loc[ dataset['Age'] > 20, 'Age'] = 2

In [5]: sex_mapping = {"male": 0, "female": 1}
for dataset in all_data:
    dataset['Sex'] = dataset['Sex'].map(sex_mapping)

In [6]: for dataset in all_data:
    dataset.loc[ dataset['SiblingSpouse'] == 0, 'SiblingSpouse'] = 0,
    dataset.loc[ dataset['SiblingSpouse'] > 0, 'SiblingSpouse'] = 1,
    dataset.loc[ dataset['ParentChild'] == 0, 'ParentChild'] = 0,
    dataset.loc[ dataset['ParentChild'] > 0, 'ParentChild'] = 1,

In [7]: titanic.isnull().sum()

Out[7]: PassengerId      0
Survived      0
PassengerClass  0
Sex           0
Age           0
SiblingSpouse  1
ParentChild    0
dtype: int64

In [8]: # Deal with missing data on "SiblingSpouse"
titanic = titanic.fillna(titanic.mean())

In [9]: from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics

In [10]: target = titanic['Survived']
data = titanic.drop('Survived', axis=1)

In [11]: # Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(data, target, test_size=40/63, random_state=1) # 70% training and 30% test

In [12]: # Create Decision Tree classifier object
clf = DecisionTreeClassifier()

# Train Decision Tree Classifier
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)

```

## Evaluating Model

After the prediction , we make a confusion matrix based on the prediction result and target.

```

In [13]: # making a confusion metrix
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
confusion_matrix

Out[13]: array([[19,  3],
               [12,  6]])

```

The accuracy fo the decision tree model:

```

In [14]: print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

Accuracy: 0.625

```

Now we got a classification rate of 72.5%. We can improve this accuracy by tuning the parameters in the Decision Tree Algorithm.

## Visualization

```

In [15]: from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
from IPython.display import Image
import pydotplus

In [16]: dot_data = StringIO()
feature_cols = list(data.columns.values)
export_graphviz(clf, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True,feature_names = feature_cols,class_
_names=['0','1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('titanic_dt.png')
Image(graph.create_png())

```

