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]:
```
train = pd.read_csv('titanic-train.csv')
test = pd.read_csv('titanic-test.csv')
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.head(40)
```

Out[7]:

	PassengerId	Survived	PassengerClass	Sex	Age	SiblingSpouse	ParentChild
0	1	0	3	0	2	1.0	0
1	2	1	1	1	2	1.0	0
2	3	1	3	1	2	0.0	0
3	4	1	1	1	2	1.0	0
4	5	0	3	0	2	0.0	0
5	6	0	1	0	2	0.0	0
6	7	0	3	0	2	0.0	0
7	8	1	3	0	0	1.0	1
8	9	1	2	1	1	1.0	0
9	10	1	3	1	0	1.0	1
10	11	1	1	1	2	0.0	0
11	12	0	3	0	1	0.0	0
12	13	0	3	0	2	1.0	1
13	14	0	3	1	1	0.0	0
14	15	1	2	1	2	0.0	0
15	16	0	3	0	0	1.0	1
16	17	0	3	1	0	0.0	1
17	18	0	2	0	2	0.0	0
18	19	1	2	0	2	0.0	0
19	20	1	3	1	1	0.0	0
20	21	1	2	0	2	0.0	0
21	22	0	3	1	0	1.0	1
22	23	1	3	1	2	1.0	0
23	24	0	1	0	1	0.0	1
24	25	0	1	0	2	0.0	0
25	26	0	2	0	2	0.0	0
26	27	0	1	0	2	1.0	0
27	28	0	1	0	2	1.0	0
28	29	0	3	0	2	0.0	0
29	30	0	3	0	1	1.0	0
30	31	1	3	1	1	1.0	0
31	32	0	3	1	2	1.0	0
32	33	0	2	1	2	1.0	0
33	34	1	2	1	0	1.0	1
34	35	1	3	1	1	0.0	0
35	36	0	3	1	1	1.0	0
36	37	0	3	0	0	1.0	1
37	38	0	3	0	2	0.0	0
38	39	1	1	1	2	1.0	0
39	40	1	2	1	2	1.0	0

In [8]:
```
titanic.tail(23)
```

Out[8]:

	PassengerId	Survived	PassengerClass	Sex	Age	SiblingSpouse	ParentChild
40	41	0	1	0	2	0.0	1
41	42	1	2	1	2	0.0	0
42	43	0	3	0	2	0.0	0
43	44	1	2	1	0	1.0	1
44	45	0	3	0	0	1.0	1
45	46	0	3	0	2	0.0	0
46	47	0	1	0	2	1.0	1
47	48	0	3	0	0	1.0	1
48	49	1	2	1	2	0.0	0
49	50	0	3	0	1	0.0	0
50	51	1	3	1	1	1.0	1
51	52	0	3	0	2	1.0	0
52	53	0	2	0	2	0.0	0
53	54	0	3	1	1	1.0	1
54	55	0	2	0	2	0.0	0
55	56	0	3	0	2	1.0	0
56	57	1	3	0	2	0.0	0
57	58	0	3	0	2	0.0	0
58	59	1	2	0	0	0.0	1
59	60	1	3	1	2	0.0	0
60	61	0	3	0	2	NaN	0
61	62	1	3	0	2	0.0	0
62	63	0	1	0	2	0.0	0

In [9]:
```
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Cla
ssifier
from sklearn.model_selection import train_test_split # Import train_test_s
plit function
from sklearn import metrics
```

In [10]:
```
titanic.replace(np.NaN, 0)
```

Out[10]:

	PassengerId	Survived	PassengerClass	Sex	Age	SiblingSpouse	ParentChild
0	1	0	3	0	2	1.0	0
1	2	1	1	1	2	1.0	0
2	3	1	3	1	2	0.0	0
3	4	1	1	1	2	1.0	0
4	5	0	3	0	2	0.0	0
5	6	0	1	0	2	0.0	0
6	7	0	3	0	2	0.0	0
7	8	1	3	1	2	0.0	1
8	9	1	2	1	1	1.0	0
9	10	1	3	1	0	1.0	1
10	11	1	1	1	2	0.0	0
11	12	0	3	0	1	0.0	0
12	13	0	3	0	2	1.0	1
13	14	0	3	1	1	0.0	0
14	15	1	2	1	2	0.0	0
15	16	0	3	0	0	1.0	1
16	17	0	3	1	0	0.0	1
17	18	0	2	0	2	0.0	0
18	19	1	2	0	2	0.0	0
19	20	1	3	1	1	0.0	0
20	21	1	2	0	2	0.0	0
21	22	0	3	1	0	1.0	1
22	23	1	3	1	2	1.0	0
23	24	0	1	0	1	0.0	1
24	25	0	1	0	2	0.0	0
25	26	0	2	0	2	0.0	0
26	27	0	1	0	2	1.0	0
27	28	0	1	0	2	1.0	0
28	29	0	3	0	2	0.0	0
29	30	0	3	0	1	1.0	0
...	...	...	...	...	...	...	...
34	35	1	3	1	1	0.0	0
35	36	0	3	1	1	1.0	0
36	37	0	3	0	0	1.0	1
37	38	0	3	0	2	0.0	0
38	39	1	1	1	2	1.0	0
39	40	1	2	1	2	1.0	0
40	41	0	1	0	2	0.0	1
41	42	1	2	1	2	0.0	0
42	43	0	3	0	2	0.0	0
43	44	1	2	1	0	1.0	1
44	45	0	3	0	0	1.0	1
45	46	0	3	0	2	0.0	0
46	47	0	1	0	2	1.0	1
47	48	0	3	0	0	1.0	1
48	49	1	2	1	2	0.0	0
49	50	0	3	0	1	0.0	0
50	51	1	3	1	1	1.0	1
51	52	0	3	0	2	1.0	0
52	53	0	2	0	2	0.0	0
53	54	0	3	1	1	1.0	1
54	55	0	2	0	2	0.0	0
55	56	0	3	0	2	1.0	0
56	57	1	3	0	2	0.0	0
57	58	0	3	0	2	0.0	0
58	59	1	2	0	0	0.0	1
59	60	1	3	1	2	0.0	0
60	61	0	3	0	2	0.0	0
61	62	1	3	0	2	0.0	0
62	63	0	1	0	2	0.0	0

63 rows × 7 columns

In [11]:
```
titanic = titanic.fillna(titanic.mean())
```

In [12]:
```
target = titanic['Survived']
data = titanic.drop('Survived', axis=1)
```

In [13]:
```
# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(data, target, test_siz
e=0.5, random_state=1) # 70% training and 30% test
```

In [14]:
```
# 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 [15]:
```
# making a confusion matrix
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
confusion_matrix
```

Out[15]:
```
array([[20,  2],
       [ 9,  9]])
```

The accuracy to the decision tree model:

In [16]:
```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.725
```

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 [17]:
```
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
from IPython.display import Image
import pydotplus
```

In [18]:
```
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())
```

Out[18]: