In [2]:
```
import pandas as pd
```

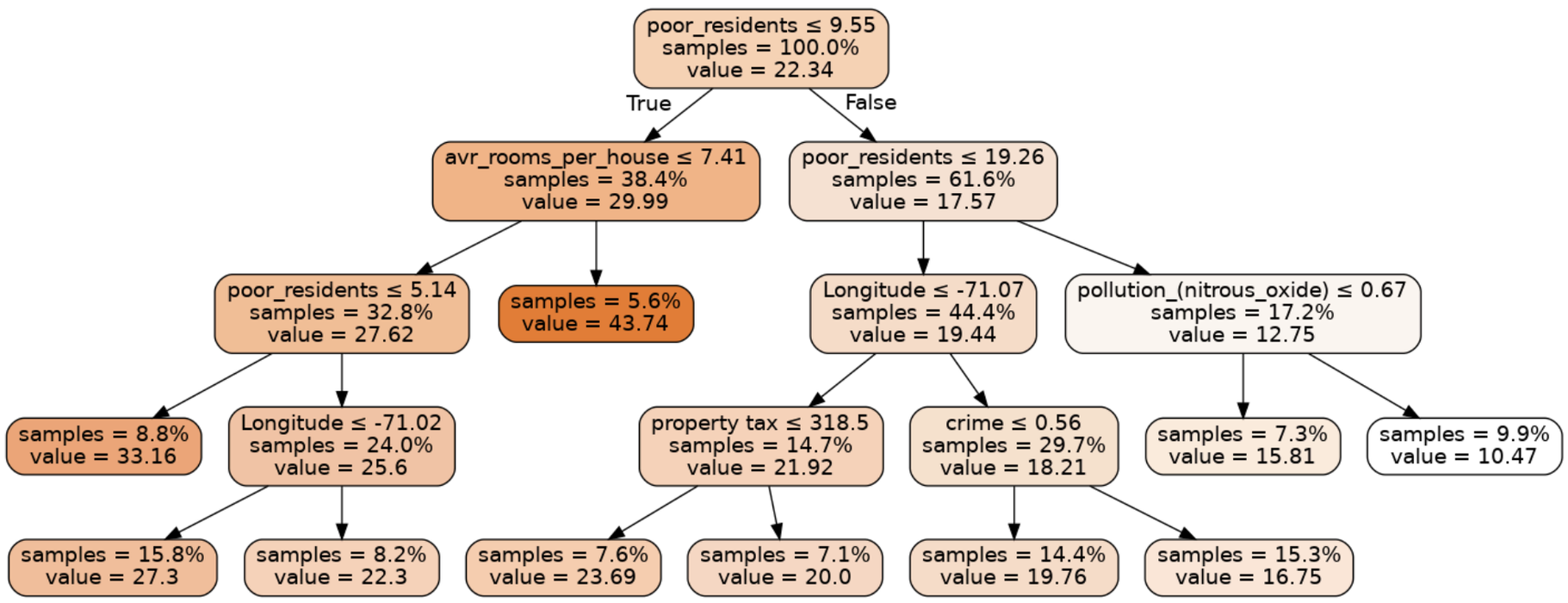
In [7]:
```
data = pd.read_csv("Boston_housing_modified.csv", header=0)
data.head()
```

Out[7]:

| | area_number | Longitude | crime | pollution_(nitrous_oxide) | avr_rooms_per_house | %houses_built_<1940 | distance_to_employment_offices | property tax | teacl student_r |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | -70.955 | 0.00632 | 0.538 | 6.575 | 65.2 | 4.0900 | 296 | 1 |
| **1** | 1 | -70.950 | 0.02731 | 0.469 | 6.421 | 78.9 | 4.9671 | 242 | 1 |
| **2** | 1 | -70.936 | 0.02729 | 0.469 | 7.185 | 61.1 | 4.9671 | 242 | 1 |
| **3** | 2 | -70.928 | 0.03237 | 0.458 | 6.998 | 45.8 | 6.0622 | 222 | 1 |
| **4** | 2 | -70.922 | 0.06905 | 0.458 | 7.147 | 54.2 | 6.0622 | 222 | 1 |

In [8]:
```
import pandas as pd
import sklearn
from sklearn.tree import DecisionTreeRegressor
from sklearn.tree import export_graphviz
from six import StringIO
from IPython.display import Image
import pydotplus

data = pd.read_csv("Boston_housing_modified.csv", header=0)
predictors = data.drop(['median_price'],axis=1) # Features
target = data['median_price'] # Target variable


from sklearn.model_selection import train_test_split
predictors_teach, predictors_test, target_teach, target_test = sklearn.model_selection.train_test_split(predictors,
target, test_size=0.3, random_state=1) # 70% training and 30% test
decision_tree = DecisionTreeRegressor(min_impurity_decrease=0.02,max_depth=4 ,min_samples_leaf=20)
decision_tree = decision_tree.fit(predictors_teach, target_teach)
dot_data = StringIO()

export_graphviz(decision_tree, out_file=dot_data,
                filled=True, rounded=True,impurity=False, proportion=True,precision=2,
                special_characters=True, feature_names = predictors.columns,class_names=['survived','died'])

graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

Out[8]:




In [9]:
```
feature_importances = pd.DataFrame({'predictor': predictors.columns,
                    'importance': decision_tree.feature_importances_}).\
                     sort_values('importance', ascending = False)


feature_importances.head()
```

Out[9]:

| | predictor | importance |
|---|---|---|
| **10** | poor_residents | 0.722157 |
| **4** | avr_rooms_per_house | 0.197858 |
| **1** | Longitude | 0.042606 |
| **3** | pollution_(nitrous_oxide) | 0.018924 |
| **2** | crime | 0.010577 |

In [10]:
```
prediction = decision_tree.predict(predictors_test)

from sklearn.metrics import r2_score


accuracy = r2_score(target_test, prediction)


print("r2_score:",accuracy)
```

```
r2_score: 0.7866802906225296
```

In [ ]:
```

```