Decision Tree

July 19, 2021

[1]:
```
import numpy as np
import pandas as pd
```

[2]:
```
data = pd.read_csv('../datasets/titanic_train.csv')
data.head()
```

[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th…	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

[3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
```

```
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

[5]:
```
columns_to_drop = ['PassengerId', 'Name', 'Ticket', 'Cabin', 'Embarked']

data_clean = data.drop(columns_to_drop, axis=1)
```

[6]:
```
data_clean.head()
```

[6]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
0	0	3	male	22.0	1	0	7.2500
1	1	1	female	38.0	1	0	71.2833
2	1	3	female	26.0	0	0	7.9250
3	1	1	female	35.0	1	0	53.1000
4	0	3	male	35.0	0	0	8.0500

[8]:
```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

data_clean['Sex'] = le.fit_transform(data_clean['Sex'])
```

[11]:
```
data_clean.head()
```

[11]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
0	0	3	1	22.0	1	0	7.2500
1	1	1	0	38.0	1	0	71.2833
2	1	3	0	26.0	0	0	7.9250
3	1	1	0	35.0	1	0	53.1000
4	0	3	1	35.0	0	0	8.0500

[12]:
```
data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    int64
 1   Pclass    891 non-null    int64
 2   Sex       891 non-null    int64
```

```
 3   Age         714 non-null    float64
 4   SibSp       891 non-null    int64
 5   Parch       891 non-null    int64
 6   Fare        891 non-null    float64
dtypes: float64(2), int64(5)
memory usage: 48.9 KB
```

[13]:
```
data_clean = data_clean.fillna(data_clean['Age'].mean())
```

[14]:
```
data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    int64
 1   Pclass    891 non-null    int64
 2   Sex       891 non-null    int64
 3   Age       891 non-null    float64
 4   SibSp     891 non-null    int64
 5   Parch     891 non-null    int64
 6   Fare      891 non-null    float64
dtypes: float64(2), int64(5)
memory usage: 48.9 KB
```

[15]:
```
input_cols = ['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare']
output_cols = ['Survived']

X = data_clean[input_cols]
Y = data_clean[output_cols]
```

[37]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
```

[28]:
```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.30,␣
 ↪random_state=42)
sk_tree = DecisionTreeClassifier(criterion='entropy')
sk_tree.fit(X_train, y_train)
```

[28]:
```
DecisionTreeClassifier(criterion='entropy')
```

[29]:
```
sk_tree.score(X_test, y_test)
```

[29]:
```
0.7574626865671642
```

[31]:
```
from sklearn.model_selection import GridSearchCV
```

[38]:
```
parameters = {
    'criterion':('entropy','gini'),
    'max_depth': [1,3,5,7,9]
}
```

[40]:
```
clf = GridSearchCV(sk_tree, parameters)
```

[41]:
```
clf.fit(X_train, y_train)
```

[41]:
```
GridSearchCV(estimator=DecisionTreeClassifier(criterion='entropy'),
             param_grid={'criterion': ('entropy', 'gini'),
                         'max_depth': [1, 3, 5, 7, 9]})
```

[46]:
```
clf.score(X_test, y_test)
```

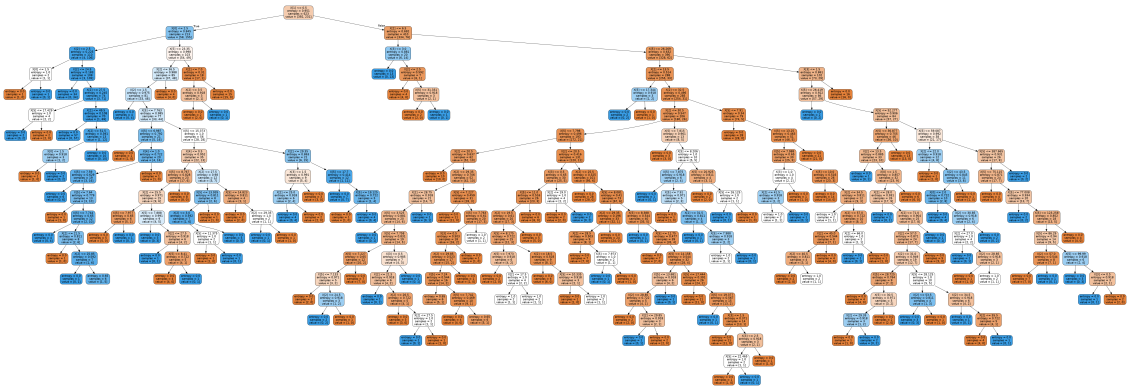
[46]:
```
0.8171641791044776
```

[53]:
```
import pydotplus
from six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
```

[54]:
```
dot_data = StringIO()
export_graphviz(sk_tree, out_file=dot_data, filled=True, rounded=True)
```

[56]:
```
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

[56]:



[ ]: