

DATA ANALYTICS

Deepika Karanji- PES1201700103

Dheemanth GR- PES1201700229

Raunak Sengupta- PES1201700072

Navneet Raju- PES1201701545

ASSIGNMENT 4 - Decision Tree Implementation

Problem statement - Use Decision Tree to Classify if whether the person survived or died in Titanic.

```
DataAnalyticsAssignments/DecisionTree-2.ipynb at master · raunaks42/DataAnalyticsAssignments - Mozilla Firefox
DA Assignment Writeup X +
v/raunaks42/DataAnalyticsAssignments/blob/master/Assignment 8/DecisionTree-2.ipynb
313 lines (313 sloc) 8.41 KB
<> Raw Blame History

In [6]: # Imports needed for the script
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Loading the data
train_url = '/home/dpk/DATA ANALYTICS PYTHON2/titanic/train.csv'
test_url = '/home/dpk/DATA ANALYTICS PYTHON2/titanic/test.csv'

train = pd.read_csv(train_url)
test = pd.read_csv(test_url)

df = pd.read_csv(train_url, index_col = 'PassengerId')
# Store our test passenger IDs for easy access

print(df.head())

   Survived  Pclass \
PassengerId
1          0       3
2          1       1
3          1       3
4          1       1
5          0       3

   PassengerId  Name                       Sex  Age  \
1              Braund, Mr. Owen Harris    male  22.0
2      Cumings, Mrs. John Bradley (Florence Briggs Th... female  38.0
3      Heikkinen, Miss. Laina              female  26.0
4      Futrelle, Mrs. Jacques Heath (Lily May Peel)    female  35.0
5      Allen, Mr. William Henry              male  35.0

   PassengerId  SibSp  Parch  Ticket            Fare Cabin Embarked
1              1      0      A/5 21171    7.2500   NaN      S
2              2      0      PC 17599   71.2833   C85      C
3              3      0  STON/O2. 3101282    7.9250   NaN      S
4              4      1      113803   53.1000  C123      S
5              5      0      373450    8.0500   NaN      S

In [7]: df = df[['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Survived']]

In [8]: df['Sex'] = df['Sex'].map({'male': 0, 'female': 1})
# We need to convert 'Sex' into an integer value of 0 or 1.

In [9]: # Drop rows with missing vals
df = df.dropna()
X = df.drop('Survived', axis=1)
y = df['Survived']
```

```
In [9]: # Drop rows with missing vals
df = df.dropna()
X = df.drop('Survived', axis=1)
y = df['Survived']
```

```
In [11]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = \
train_test_split(X, y, random_state=1)
```

```
In [12]: from sklearn import tree

model = tree.DecisionTreeClassifier()
```

```
In [13]: model
```

```
Out[13]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                                max_features=None, max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, presort=False,
                                random_state=None, splitter='best')
```

```
In [14]: model.fit(X_train, y_train)
```

```
Out[14]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                                max_features=None, max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, presort=False,
                                random_state=None, splitter='best')
```

```
In [15]: # Then we score the predicted output from model on
# our test data against our ground truth test data.
y_predict = model.predict(X_test)

from sklearn.metrics import accuracy_score

accuracy_score(y_test, y_predict)
```

```
Out[15]: 0.8212290502793296
```

```
In [16]: from sklearn.metrics import confusion_matrix

pd.DataFrame(
    confusion_matrix(y_test, y_predict),
    columns=['Predicted Not Survival', 'Predicted Survival'],
    index=['True Not Survival', 'True Survival']
)
```

```
Out[16]:
```

	Predicted Not Survival	Predicted Survival
True Not Survival	96	16
True Survival	16	51

```
In [ ]: tree.export_graphviz(model.tree_, out_file='tree.dot', feature_names=X.columns)
```