decision tree-practice

March 27, 2020

the motive of this assignment is to apply every parameter and also plot roc_auc_curve of train and test data and also plot 3d plot n_estimator,max_depth and roc_curve

[1]:

```
import pyforest
from sklearn.externals.six import StringIO
from sklearn.tree import export_graphviz
import pydot
import pydotplus
```

```
/home/sushil/anaconda3/lib/python3.7/site-packages/sklearn/externals/six.py:31:
FutureWarning: The module is deprecated in version 0.21 and will be removed in
version 0.23 since we've dropped support for Python 2.7. Please rely on the
official version of six (https://pypi.org/project/six/).
  "(https://pypi.org/project/six/).", FutureWarning)
```

[2]:

```
lazy_imports()
```

[2]:

```
['import dash',
 'from sklearn.model_selection import train_test_split',
 'import plotly.graph_objs as go',
 'from sklearn.ensemble import RandomForestRegressor',
 'from sklearn import svm',
 'import bokeh',
 'import gensim',
 'from openpyxl import load_workbook',
 'from sklearn.feature_extraction.text import TfidfVectorizer',
 'from sklearn.manifold import TSNE',
 'import glob',
 'import sys',
 'from sklearn.ensemble import GradientBoostingRegressor',
 'import plotly.express as px',
 'import pandas as pd',
 'import sklearn',
 'import statistics',
 'import pickle',
 'import spacy',
 'from dask import dataframe as dd',
 'import re',
 'import seaborn as sns',
 'import matplotlib as mpl',
 'import numpy as np',
 'import altair as alt',
 'import keras',
 'import os',
 'import pydot',
 'from pathlib import Path',
 'import nltk',
 'from sklearn.ensemble import GradientBoostingClassifier',
 'import matplotlib.pyplot as plt',
 'from sklearn.preprocessing import OneHotEncoder',
 'import tensorflow as tf',
 'from pyspark import SparkContext',
 'import datetime as dt',
 'import tqdm',
 'import plotly as py',
 'from sklearn.ensemble import RandomForestClassifier']
```

[3]:

```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import DecisionTreeRegressor
```

[4]:

```python
datasets=pd.read_csv("/home/sushil/Downloads/py-master/ML/9_decision_tree/
 ↪Exercise/titanic.csv")
```

```
<IPython.core.display.Javascript object>
```

[5]:

```python
datasets.Sex=datasets.Sex.map({"male":1,"female":0})
datasets
```

[5]:

```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           …         …     …
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                  Name  Sex   Age  SibSp  \
0                              Braund, Mr. Owen Harris    1  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…    0  38.0      1
2                                Heikkinen, Miss. Laina    0  26.0      0
3          Futrelle, Mrs. Jacques Heath (Lily May Peel)    0  35.0      1
4                              Allen, Mr. William Henry    1  35.0      0
..                                                 …  …   …    …
886                               Montvila, Rev. Juozas    1  27.0      0
887                        Graham, Miss. Margaret Edith    0  19.0      0
888            Johnston, Miss. Catherine Helen "Carrie"    0   NaN      1
889                               Behr, Mr. Karl Howell    1  26.0      0
890                                 Dooley, Mr. Patrick    1  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     …               …      …   …      …
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

[6]:

```python
Y=datasets.Survived
Y
```

[6]:

```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

[7]:

```python
#datasets.drop(["Pclass","Sex","Age","SibSp","Parch","Fare","Embarked"], axis=1, inplace=True)
datasets.drop(["PassengerId","Survived","Name","Ticket","Cabin","Embarked"], axis=1, inplace=True)
```

[8]:
```
X=datasets
X
```

[8]:
```
     Pclass  Sex   Age  SibSp  Parch     Fare
0         3    1  22.0      1      0   7.2500
1         1    0  38.0      1      0  71.2833
2         3    0  26.0      0      0   7.9250
3         1    0  35.0      1      0  53.1000
4         3    1  35.0      0      0   8.0500
..      …   …    …     …     …      …
886       2    1  27.0      0      0  13.0000
887       1    0  19.0      0      0  30.0000
888       3    0   NaN      1      2  23.4500
889       1    1  26.0      0      0  30.0000
890       3    1  32.0      0      0   7.7500

[891 rows x 6 columns]
```

[9]:
```
Y.value_counts()
```

[9]:
```
0    549
1    342
Name: Survived, dtype: int64
```

[10]:
```
#df.Age = datasets.Age.fillna()
datasets.Age= datasets.Age.interpolate()
```

[11]:
```
datasets.Fare.value_counts(dropna=False)
```

[11]:
```
8.0500     43
13.0000    42
7.8958     38
7.7500     34
26.0000    31
           ..
8.4583      1
9.8375      1
8.3625      1
14.1083     1
17.4000     1
Name: Fare, Length: 248, dtype: int64
```

[12]:
```
datasets.Fare= datasets.Fare.interpolate()
```

[13]:
```
X=datasets
```

[14]:
```
X
```

[14]:

```
     Pclass  Sex   Age  SibSp  Parch     Fare
0         3    1  22.0      1      0   7.2500
1         1    0  38.0      1      0  71.2833
2         3    0  26.0      0      0   7.9250
3         1    0  35.0      1      0  53.1000
4         3    1  35.0      0      0   8.0500
..      …   …    …     …      …       …
886       2    1  27.0      0      0  13.0000
887       1    0  19.0      0      0  30.0000
888       3    0  22.5      1      2  23.4500
889       1    1  26.0      0      0  30.0000
890       3    1  32.0      0      0   7.7500

[891 rows x 6 columns]
```

[15]:

```python
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.3)
print(x_train.shape)
print(x_test.shape)
print(len(y_train))
print(len(y_test))
```

```
<IPython.core.display.Javascript object>


(623, 6)
(268, 6)
623
268
```

[16]:

```python
dt=DecisionTreeClassifier()
parameters={"max_depth":list(np.arange(10,100,3)),"min_samples_leaf":list(np.
 ↪arange(7,20))}
```

```
<IPython.core.display.Javascript object>


<IPython.core.display.Javascript object>
```

[17]:

```python
list(np.arange(7,20))
```

```
<IPython.core.display.Javascript object>
```

[17]:

```
[7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

[18]:

```python
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report,roc_auc_score,roc_curve
```

```
[19]: %%time
      clf=GridSearchCV(dt,param_grid=parameters,scoring="roc_auc",cv=4,return_train_score=True)
      clf.fit(x_train,y_train)
```

```
CPU times: user 13 s, sys: 30.8 ms, total: 13.1 s
Wall time: 13.2 s
```

```
[19]: GridSearchCV(cv=4, error_score=nan,
                   estimator=DecisionTreeClassifier(ccp_alpha=0.0, 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='deprecated',
                                                    random_state=None,
                                                    splitter='best'),
                   iid='deprecated', n_jobs=None,
                   param_grid={'max_depth': [10, 13, 16, 19, 22, 25, 28, 31, 34, 37,
                                             40, 43, 46, 49, 52, 55, 58, 61, 64, 67,
                                             70, 73, 76, 79, 82, 85, 88, 91, 94, 97],
                               'min_samples_leaf': [7, 8, 9, 10, 11, 12, 13, 14, 15,
                                                    16, 17, 18, 19]},
                   pre_dispatch='2*n_jobs', refit=True, return_train_score=True,
                   scoring='roc_auc', verbose=0)
```

```
[20]: print(clf.best_params_)
      print(clf.best_score_)
```

```
{'max_depth': 37, 'min_samples_leaf': 18}
0.8558244320856874
```

```
[21]: import sklearn.metrics as metrics
      y_train_pred=clf.predict_proba(x_train)
      fpr_tr,tpr_tr,threshold=roc_curve(y_train,y_train_pred[:,1])
      roc_auc_tr=metrics.auc(fpr_tr,tpr_tr)
      y_pred=clf.predict_proba(x_test)
      fpr,tpr,thrshold=roc_curve(y_test,y_pred[:,1])
      roc_auc=metrics.auc(fpr,tpr)
```

```
[22]: plt.plot(fpr,tpr,label="AUC = %0.2f "%roc_auc)
      plt.plot(fpr_tr,tpr_tr,label="AUC_Tr= %0.2f"%roc_auc_tr)
      plt.title("Roc AUC CURVE")
      plt.legend(loc = 'lower right')
      plt.plot([0,1],[0,1],"r--")
      #plt.xlim([0,1])
      #plt.ylim([0,1])
      #plt.xlabel("FPR")
      plt.ylabel("TPR")
      plt.show()
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

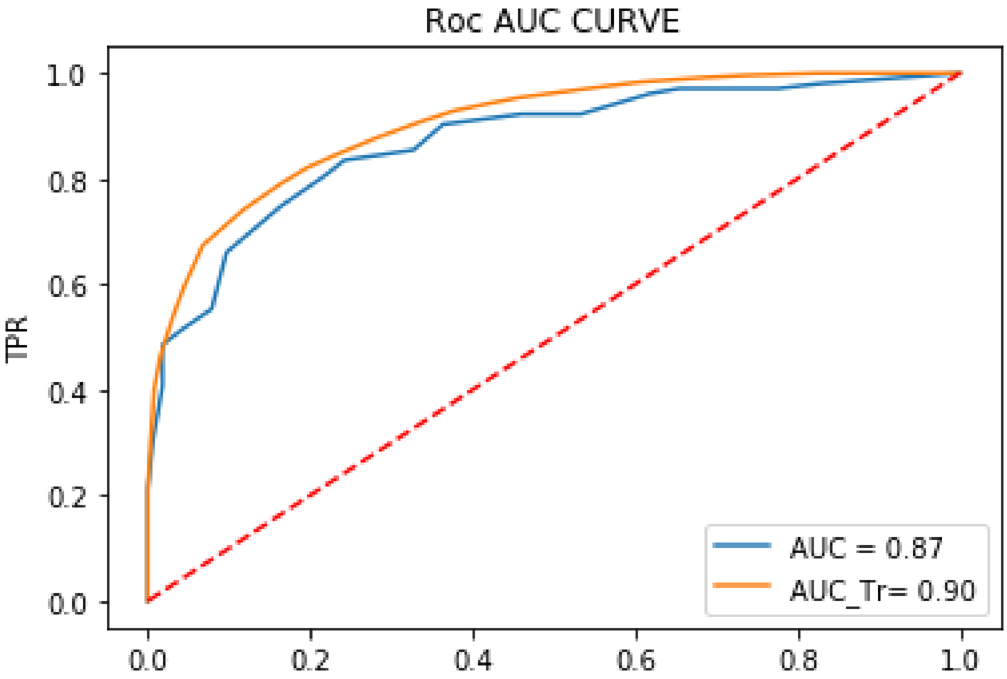
```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```



[23]:

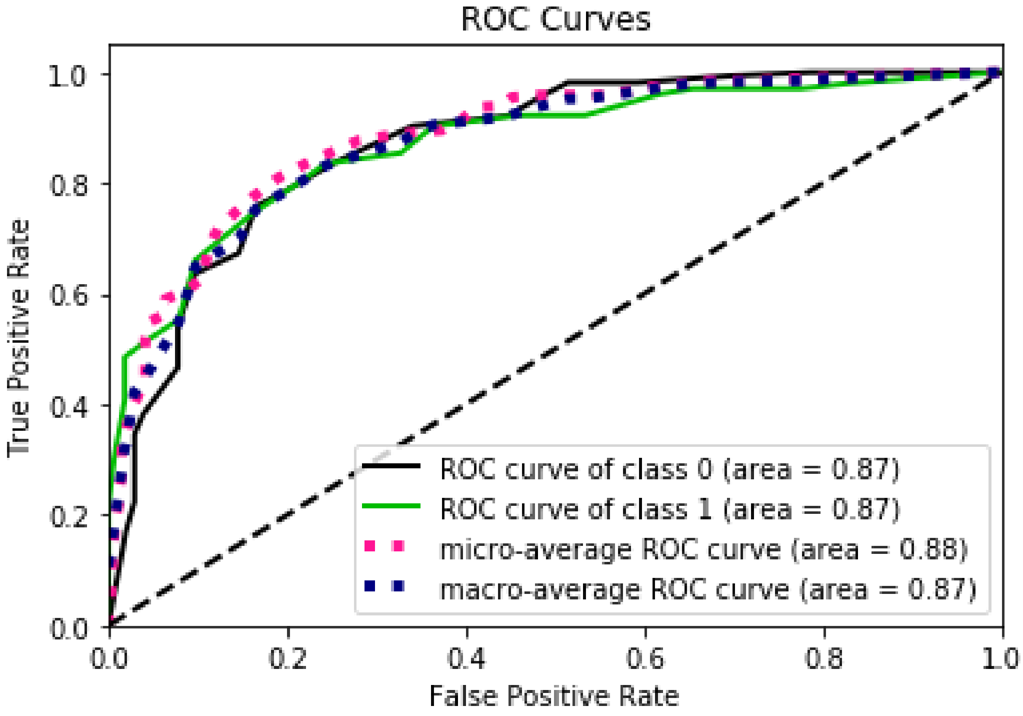
```
import scikitplot as skplt
```

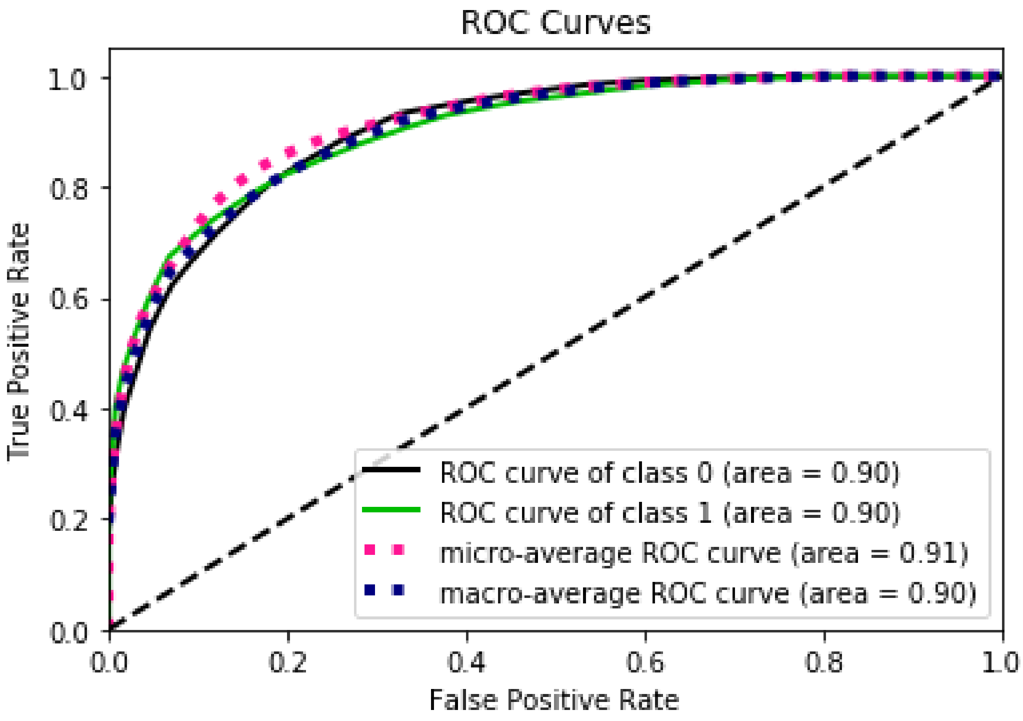
[24]:

```
skplt.metrics.plot_roc_curve(y_test, y_pred)
skplt.metrics.plot_roc_curve(y_train,y_train_pred)
plt.show()
```

```
/home/sushil/anaconda3/lib/python3.7/site-
packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_roc_curve
is deprecated; This will be removed in v0.5.0. Please use
scikitplot.metrics.plot_roc instead.
  warnings.warn(msg, category=FutureWarning)
/home/sushil/anaconda3/lib/python3.7/site-
packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_roc_curve
is deprecated; This will be removed in v0.5.0. Please use
scikitplot.metrics.plot_roc instead.
  warnings.warn(msg, category=FutureWarning)
```

```
<IPython.core.display.Javascript object>
```





[25]:
```
from sklearn.metrics import accuracy_score
y_predict=clf.predict(x_test)
accuracy=accuracy_score(y_test,y_predict)
print("the accuracy",accuracy)
```

```
the accuracy 0.8097014925373134
```

[26]:
```
#optimized_GBM.best_estimator_.feature_importances_
feature_importance=clf.best_estimator_.feature_importances_
print("feature importance:-",feature_importance)
```

```
feature importance:- [0.18146957 0.58959398 0.13138654 0.         0.
0.09754991]
```

[27]:
```
from IPython.display import Image
```

[28]:
```
from sklearn.externals.six import StringIO
from sklearn.tree import export_graphviz
import pydot
import pydotplus
```

[29]:
```
feature_name=list(X.columns)
feature_name
```

```
[29]: ['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare']
```

```
[30]: dot_data=StringIO()
```

```
[31]: export_graphviz(clf.
       ↪best_estimator_,feature_names=feature_name,filled=True,rounded=True)
```

```
[31]: 'digraph Tree {\nnode [shape=box, style="filled, rounded", color="black",
      fontname=helvetica] ;\nedge [fontname=helvetica] ;\n0 [label="Sex <= 0.5\\ngini
      = 0.473\\nsamples = 623\\nvalue = [384, 239]", fillcolor="#f5cfb4"] ;\n1
      [label="Pclass <= 2.5\\ngini = 0.382\\nsamples = 218\\nvalue = [56, 162]",
      fillcolor="#7dbfee"] ;\n0 -> 1 [labeldistance=2.5, labelangle=45,
      headlabel="True"] ;\n2 [label="Fare <= 13.25\\ngini = 0.097\\nsamples =
      117\\nvalue = [6, 111]", fillcolor="#44a2e6"] ;\n1 -> 2 ;\n3 [label="gini =
      0.278\\nsamples = 18\\nvalue = [3, 15]", fillcolor="#61b1ea"] ;\n2 -> 3 ;\n4
      [label="Fare <= 28.856\\ngini = 0.059\\nsamples = 99\\nvalue = [3, 96]",
      fillcolor="#3fa0e6"] ;\n2 -> 4 ;\n5 [label="gini = 0.121\\nsamples = 31\\nvalue
      = [2, 29]", fillcolor="#47a4e7"] ;\n4 -> 5 ;\n6 [label="Fare <= 116.638\\ngini =
      0.029\\nsamples = 68\\nvalue = [1, 67]", fillcolor="#3c9ee5"] ;\n4 -> 6 ;\n7
      [label="gini = 0.0\\nsamples = 50\\nvalue = [0, 50]", fillcolor="#399de5"] ;\n6
      -> 7 ;\n8 [label="gini = 0.105\\nsamples = 18\\nvalue = [1, 17]",
      fillcolor="#45a3e7"] ;\n6 -> 8 ;\n9 [label="Fare <= 23.7\\ngini = 0.5\\nsamples
      = 101\\nvalue = [50, 51]", fillcolor="#fbfdfe"] ;\n1 -> 9 ;\n10 [label="Fare <=
      15.373\\ngini = 0.481\\nsamples = 82\\nvalue = [33, 49]", fillcolor="#bedff7"]
      ;\n9 -> 10 ;\n11 [label="Fare <= 7.888\\ngini = 0.498\\nsamples = 58\\nvalue =
      [27, 31]", fillcolor="#e5f2fc"] ;\n10 -> 11 ;\n12 [label="gini = 0.436\\nsamples
      = 28\\nvalue = [9, 19]", fillcolor="#97cbf1"] ;\n11 -> 12 ;\n13 [label="gini =
      0.48\\nsamples = 30\\nvalue = [18, 12]", fillcolor="#f6d5bd"] ;\n11 -> 13 ;\n14
      [label="gini = 0.375\\nsamples = 24\\nvalue = [6, 18]", fillcolor="#7bbeee"]
      ;\n10 -> 14 ;\n15 [label="gini = 0.188\\nsamples = 19\\nvalue = [17, 2]",
      fillcolor="#e89050"] ;\n9 -> 15 ;\n16 [label="Age <= 6.973\\ngini =
      0.308\\nsamples = 405\\nvalue = [328, 77]", fillcolor="#eb9f67"] ;\n0 -> 16
      [labeldistance=2.5, labelangle=-45, headlabel="False"] ;\n17 [label="gini =
      0.401\\nsamples = 18\\nvalue = [5, 13]", fillcolor="#85c3ef"] ;\n16 -> 17 ;\n18
      [label="Pclass <= 1.5\\ngini = 0.276\\nsamples = 387\\nvalue = [323, 64]",
      fillcolor="#ea9a60"] ;\n16 -> 18 ;\n19 [label="Age <= 36.5\\ngini =
      0.436\\nsamples = 81\\nvalue = [55, 26]", fillcolor="#f1bd97"] ;\n18 -> 19 ;\n20
      [label="gini = 0.496\\nsamples = 35\\nvalue = [19, 16]", fillcolor="#fbebe0"]
      ;\n19 -> 20 ;\n21 [label="Age <= 50.5\\ngini = 0.34\\nsamples = 46\\nvalue =
      [36, 10]", fillcolor="#eca470"] ;\n19 -> 21 ;\n22 [label="gini = 0.408\\nsamples
      = 28\\nvalue = [20, 8]", fillcolor="#efb388"] ;\n21 -> 22 ;\n23 [label="gini =
      0.198\\nsamples = 18\\nvalue = [16, 2]", fillcolor="#e89152"] ;\n21 -> 23 ;\n24
      [label="Age <= 32.25\\ngini = 0.218\\nsamples = 306\\nvalue = [268, 38]",
      fillcolor="#e99355"] ;\n18 -> 24 ;\n25 [label="Age <= 30.75\\ngini =
      0.282\\nsamples = 200\\nvalue = [166, 34]", fillcolor="#ea9b62"] ;\n24 -> 25
      ;\n26 [label="Age <= 16.75\\ngini = 0.254\\nsamples = 181\\nvalue = [154, 27]",
      fillcolor="#ea975c"] ;\n25 -> 26 ;\n27 [label="gini = 0.384\\nsamples =
      27\\nvalue = [20, 7]", fillcolor="#eead7e"] ;\n26 -> 27 ;\n28 [label="Pclass <=
      2.5\\ngini = 0.226\\nsamples = 154\\nvalue = [134, 20]", fillcolor="#e99457"]
      ;\n26 -> 28 ;\n29 [label="gini = 0.117\\nsamples = 32\\nvalue = [30, 2]",
      fillcolor="#e78946"] ;\n28 -> 29 ;\n30 [label="Fare <= 9.492\\ngini =
      0.252\\nsamples = 122\\nvalue = [104, 18]", fillcolor="#ea975b"] ;\n28 -> 30
      ;\n31 [label="Fare <= 7.812\\ngini = 0.208\\nsamples = 102\\nvalue = [90, 12]",
      fillcolor="#e89253"] ;\n30 -> 31 ;\n32 [label="Age <= 24.5\\ngini =
      0.278\\nsamples = 48\\nvalue = [40, 8]", fillcolor="#ea9a61"] ;\n31 -> 32 ;\n33
      [label="gini = 0.198\\nsamples = 27\\nvalue = [24, 3]", fillcolor="#e89152"]
      ;\n32 -> 33 ;\n34 [label="gini = 0.363\\nsamples = 21\\nvalue = [16, 5]",
      fillcolor="#eda877"] ;\n32 -> 34 ;\n35 [label="Age <= 21.5\\ngini =
      0.137\\nsamples = 54\\nvalue = [50, 4]", fillcolor="#e78b49"] ;\n31 -> 35 ;\n36
      [label="gini = 0.266\\nsamples = 19\\nvalue = [16, 3]", fillcolor="#ea995e"]
      ;\n35 -> 36 ;\n37 [label="gini = 0.056\\nsamples = 35\\nvalue = [34, 1]",
      fillcolor="#e6853f"] ;\n35 -> 37 ;\n38 [label="gini = 0.42\\nsamples =
      20\\nvalue = [14, 6]", fillcolor="#f0b78e"] ;\n30 -> 38 ;\n39 [label="gini =
      0.465\\nsamples = 19\\nvalue = [12, 7]", fillcolor="#f4caac"] ;\n25 -> 39 ;\n40
      [label="Fare <= 7.91\\ngini = 0.073\\nsamples = 106\\nvalue = [102, 4]",
      fillcolor="#e68641"] ;\n24 -> 40 ;\n41 [label="gini = 0.0\\nsamples = 37\\nvalue
      = [37, 0]", fillcolor="#e58139"] ;\n40 -> 41 ;\n42 [label="Fare <= 8.352\\ngini
      = 0.109\\nsamples = 69\\nvalue = [65, 4]", fillcolor="#e78945"] ;\n40 -> 42
      ;\n43 [label="gini = 0.278\\nsamples = 18\\nvalue = [15, 3]",
      fillcolor="#ea9a61"] ;\n42 -> 43 ;\n44 [label="Age <= 36.25\\ngini =
      0.038\\nsamples = 51\\nvalue = [50, 1]", fillcolor="#e6843d"] ;\n42 -> 44 ;\n45
      [label="gini = 0.091\\nsamples = 21\\nvalue = [20, 1]", fillcolor="#e68743"]
      ;\n44 -> 45 ;\n46 [label="gini = 0.0\\nsamples = 30\\nvalue = [30, 0]",
      fillcolor="#e58139"] ;\n44 -> 46 ;\n}'
```

[32]:
```
"""graph=pydot.graph_from_dot_data(dot_data.getvalue())
Image(graph[0].create_png())"""
```

[32]:
```
'graph=pydot.graph_from_dot_data(dot_data.getvalue())\nImage(graph[0].create_png
())'
```

[33]:
```
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
```

```
^
Expected {'graph' | 'digraph'}  (at char 0), (line:1, col:1)
```

[34]:
```
"""graph[0].write_png("dtree2.png")
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-48-b404494ad3df> in <module>
----> 1 graph[0].write_png("dtree2.png")

TypeError: 'NoneType' object is not subscriptable"""
```

[34]:

```
'graph[0].write_png("dtree2.png")\n---------------------------------------------------
------------------------------\nTypeError
Traceback (most recent call last)\n<ipython-input-48-b404494ad3df> in
<module>\n----> 1 graph[0].write_png("dtree2.png")\n\nTypeError: \'NoneType\'
object is not subscriptable'
```

[35]:

```
from sklearn import tree
dotfile =StringIO()
tree.export_graphviz(clf.
 ↪best_estimator_,feature_names=feature_name,filled=True,rounded=True,␣
 ↪out_file=dotfile)
graph=pydotplus.graph_from_dot_data(dotfile.getvalue())
graph.write_png("dtree.png")
```

[35]:

```
True
```

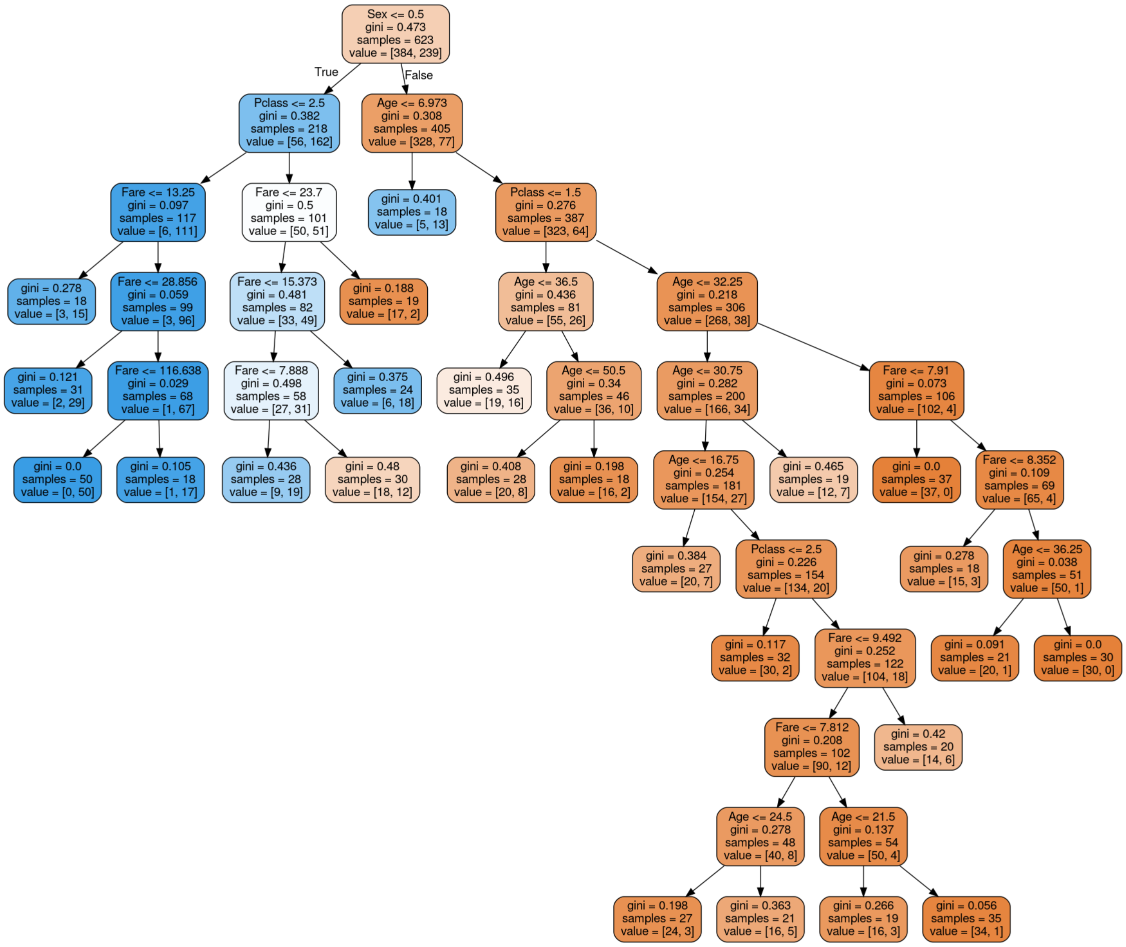
[36]:

```
(graph,)=pydot.graph_from_dot_data(dotfile.getvalue())
```

[37]:

```
Image(graph.create_png())
```

[37]:


Sex <= 0.5 gini = 0.473 samples = 623 value = [384, 239]
True
False
Pclass <= 2.5 gini = 0.382 samples = 218 value = [56, 162]
Age <= 6.973 gini = 0.308 samples = 405 value = [328, 77]
Fare <= 13.25 gini = 0.097 samples = 117 value = [6, 111]
Fare <= 23.7 gini = 0.5 samples = 101 value = [50, 51]
gini = 0.401 samples = 18 value = [5, 13]
Pclass <= 1.5 gini = 0.276 samples = 387 value = [323, 64]
gini = 0.278 samples = 18 value = [3, 15]
Fare <= 28.856 gini = 0.059 samples = 99 value = [3, 96]
Fare <= 15.373 gini = 0.481 samples = 82 value = [33, 49]
gini = 0.188 samples = 19 value = [17, 2]
Age <= 36.5 gini = 0.436 samples = 81 value = [55, 26]
Age <= 32.25 gini = 0.218 samples = 306 value = [268, 38]
gini = 0.121 samples = 31 value = [2, 29]
Fare <= 116.638 gini = 0.029 samples = 68 value = [1, 67]
Fare <= 7.888 gini = 0.498 samples = 58 value = [27, 31]
gini = 0.375 samples = 24 value = [6, 18]
gini = 0.496 samples = 35 value = [19, 16]
Age <= 50.5 gini = 0.34 samples = 46 value = [36, 10]
Age <= 30.75 gini = 0.282 samples = 200 value = [166, 34]
Fare <= 7.91 gini = 0.073 samples = 106 value = [102, 4]
gini = 0.0 samples = 50 value = [0, 50]
gini = 0.105 samples = 18 value = [1, 17]
gini = 0.436 samples = 28 value = [9, 19]
gini = 0.48 samples = 30 value = [18, 12]
gini = 0.408 samples = 28 value = [20, 8]
gini = 0.198 samples = 18 value = [16, 2]
Age <= 16.75 gini = 0.254 samples = 181 value = [154, 27]
gini = 0.465 samples = 19 value = [12, 7]
gini = 0.0 samples = 37 value = [37, 0]
Fare <= 8.352 gini = 0.109 samples = 69 value = [65, 4]
gini = 0.384 samples = 27 value = [20, 7]
Pclass <= 2.5 gini = 0.226 samples = 154 value = [134, 20]
gini = 0.278 samples = 18 value = [15, 3]
Age <= 36.25 gini = 0.038 samples = 51 value = [50, 1]
gini = 0.117 samples = 32 value = [30, 2]
Fare <= 9.492 gini = 0.252 samples = 122 value = [104, 18]
gini = 0.091 samples = 21 value = [20, 1]
gini = 0.0 samples = 30 value = [30, 0]
Fare <= 7.812 gini = 0.208 samples = 102 value = [90, 12]
gini = 0.42 samples = 20 value = [14, 6]
Age <= 24.5 gini = 0.278 samples = 48 value = [40, 8]
Age <= 21.5 gini = 0.137 samples = 54 value = [50, 4]
gini = 0.198 samples = 27 value = [24, 3]
gini = 0.363 samples = 21 value = [16, 5]
gini = 0.266 samples = 19 value = [16, 3]
gini = 0.056 samples = 35 value = [34, 1]

[39]:

```
import plotly.offline as offline
import plotly.graph_objs as go
offline.init_notebook_mode()
```

[42]:

```
train_auc=clf.cv_results_["mean_train_score"]
train_std=clf.cv_results_["std_train_score"]
test_auc=clf.cv_results_["mean_test_score"]
test_std=clf.cv_results_["std_test_score"]
```

[44]:

```
print(len(train_auc))
print(len(test_auc))
```

```
390
390
```

[51]:

```
x1=[]
y1=[]
max_depth=list(np.arange(10,100,3))
min_samples_leaf=list(np.arange(7,20))
print(len(max_depth))
print(len(min_samples_leaf))
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

```
30
13
```

[55]:

```
from itertools import repeat
train_auc_score=clf.cv_results_["mean_train_score"]
test_auc_score=clf.cv_results_["mean_test_score"]
x1 = [x for item in max_depth for x in repeat(item, 13)]
y1 = [y for item in min_samples_leaf for y in repeat(item, 30)]
```

[57]:

```
trace1 = go.Scatter3d(x=x1,y=y1,z=train_auc_score, name="train auc")
trace2 = go.Scatter3d(x=x1,y=y1,z=test_auc_score, name="test auc")
data = [trace1, trace2]
layout = go.Layout(scene = dict(
xaxis = dict(title='max_depth'),
yaxis = dict(title='min_samples_leaf'),
zaxis = dict(title='AUC'),))
fig = go.Figure(data=data, layout=layout)
offline.iplot(fig, filename='3d-scatter-colorscale')
```

[ ]: