ML_4_Assignment

March 18, 2019

1 Predicting Survival in the Titanic Data Set

In [1]:
```
# Importing necessary libraries

import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

In [2]:
```
url = "https://raw.githubusercontent.com/BigDataGal/Python-for-Data-Science/master/tita
titanic = pd.read_csv(url)
titanic.columns
```

Out[2]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

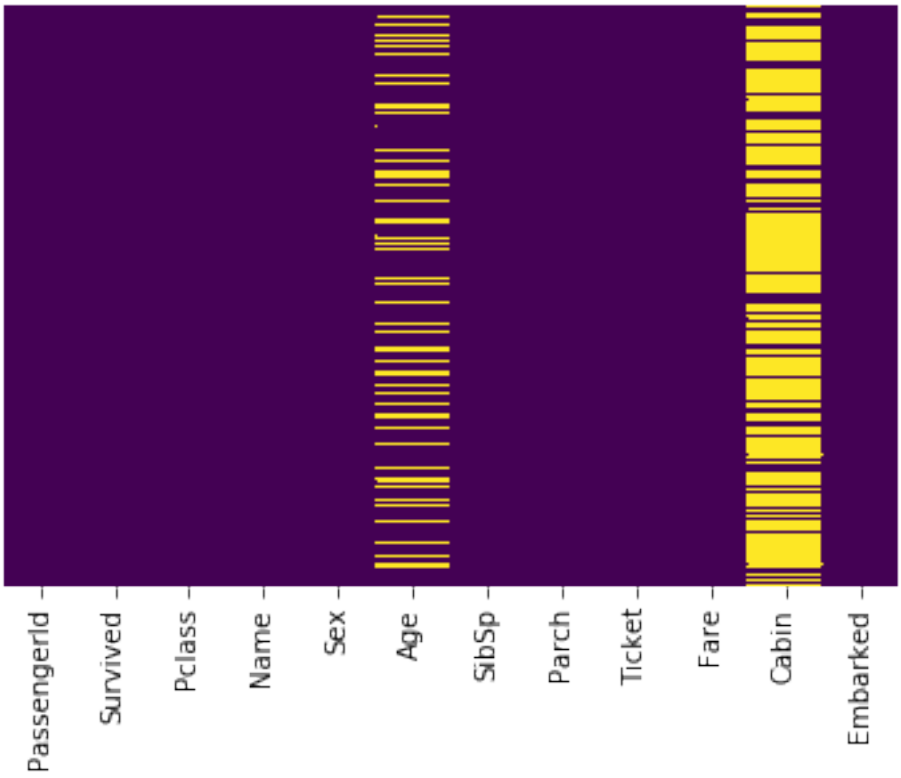
In [3]:
```
titanic.isnull().any().any(),titanic.shape
```

Out[3]:
```
(True, (891, 12))
```

In [4]:
```
# Visualizing the missing records column wise using seaborn heatmap

sb.heatmap(titanic.isnull(),yticklabels = False,cbar = False,cmap='viridis')
```

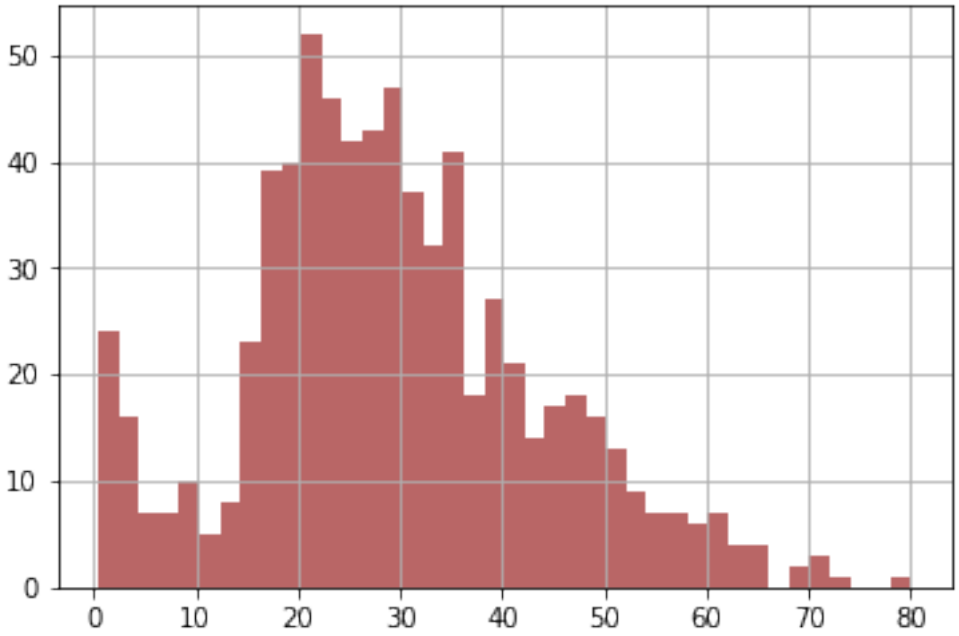
Out[4]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x20381d06ac8>
```



In [5]:
```
# From the above we can see that only 2 columns 'Age' and 'Cabin' has NaN values.
# Cabin has so many NaN values,So we should go with dropping the entire column.
# Let's see the distribution of age

titanic['Age'].hist(bins=40,color='darkred',alpha=0.6)
```

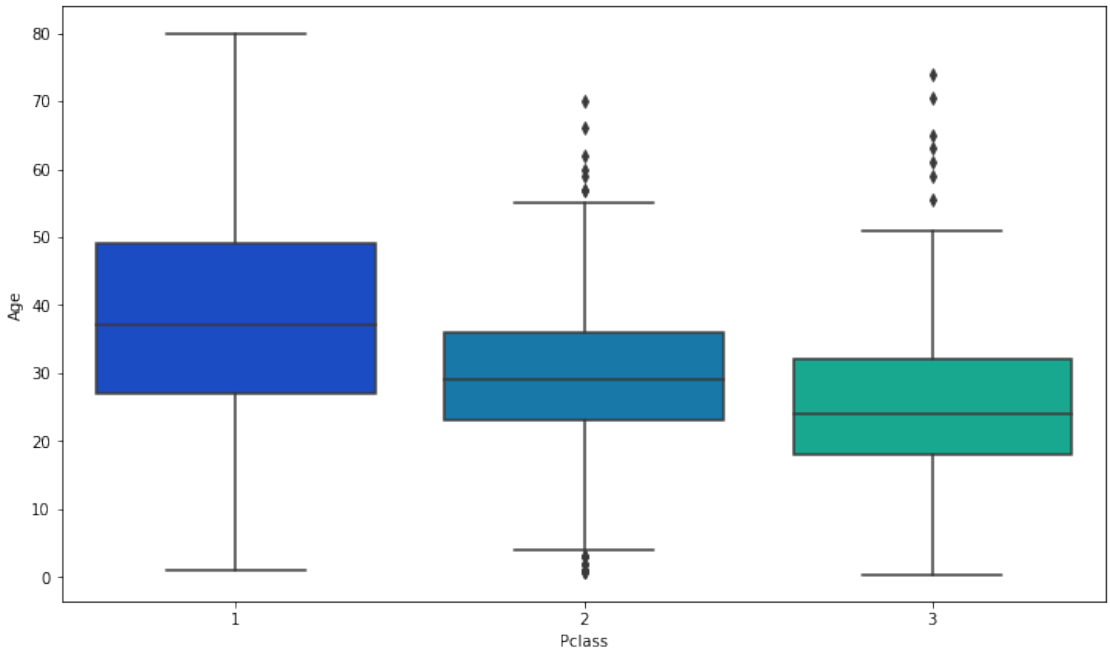
Out[5]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x20381c91908>
```



In [6]:
```
# We can replace the NaN values in Age column by their mean age in each passenger clas
# To find the mean age in each passenger class, we can use boxplot in seaborn

plt.figure(figsize=(12,7))
sb.boxplot(x='Pclass',y='Age',data=titanic,palette='winter')
```

Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x203fc803f60>



In [7]:
```
# Replacing the missing age value with the mean of age in each passenger class.

def impute_age(cols):
    Age = cols[0]
    Pclass = cols[1]
    if pd.isnull(Age):
        if  Pclass == 1:
            return 37
        elif Pclass == 2:
            return 29
        else:
            return 24
    else:
        return Age

titanic['Age'] = titanic[['Age','Pclass']].apply(impute_age,axis=1)
```

In [16]:
```
# Let's check if an Nan Values are there in Age column

 titanic['Age'].isnull().any()
```

Out[16]: False

In [8]:
```
# Now as we can see there are so many NaN values Cabin column.We can drop the same.

titanic.drop('Cabin',axis = 1,inplace = True)
```

In [9]:
```
# Changing categorical variables and dropping the columns that are not required.

sex = pd.get_dummies(titanic['Sex'],drop_first = True)
embark = pd.get_dummies(titanic['Embarked'],drop_first= True)
titanic.drop(['Sex','Embarked','Name','Ticket'],axis=1,inplace = True)
titanic = pd.concat([titanic,sex,embark],axis=1)
titanic.head()
```

Out[9]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	male	Q	S
0	1	0	3	22.0	1	0	7.2500	1	0	1
1	2	1	1	38.0	1	0	71.2833	0	0	0
2	3	1	3	26.0	0	0	7.9250	0	0	1
3	4	1	1	35.0	1	0	53.1000	0	0	1
4	5	0	3	35.0	0	0	8.0500	1	0	1

In [11]:
```
# Splitting into train and test parts and applying the model
```

```
X = titanic.drop('Survived',axis = 1,)
y = titanic['Survived']
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.33,random_state =
lr = LogisticRegression()
lr.fit(X_train,y_train)
y_pred = lr.predict(X_test)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning
  FutureWarning)
```

In [13]:
```
# Accuracy Score

from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test,y_pred)
accuracy
```

Out[13]:
```
0.7966101694915254
```

In [17]:
```
# Applying decession tree

from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(X_train,y_train)
y_pred1 = dt.predict(X_test)
accuracy1 = accuracy_score(y_test,y_pred1)
accuracy1
```

Out[17]:
```
0.7322033898305085
```

In [ ]:
```
# As we are getting a better accuracy with LogisticRegression we should go with that.
```