DMW_p7.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

21162101012_CBA_Yash_Lakhtariya

DMW Practical 7

From the given dataset, you need to create a classification model which will predict whether the particular person has survived the tragic incident or not. And to create the model, you first have to convert the dataset into appropriate form, e.g - remove the special characters, remove null/duplicate entries, etc. A decision tree should be constructed from the given data. More, to test the model if any new tuple assigned to it then, it should be able to classify that the given detail of the person has actually survived the incident or not.

[1]
```
import pandas as pd
from bs4 import BeautifulSoup
```

[4]
```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving titanicdata.htm to titanicdata.htm
```

[5]
```
file = uploaded['titanicdata.htm']
soup = BeautifulSoup(file,"html.parser")
```

[6]
```
table = soup.find('table')
```

[7]
```
data=pd.read_html(str(table),flavor='bs4')[0]
data.head(10)
```

	Name	Age	Class/Dept	Ticket	Joined	Job	Boat [Body]	Unnamed: 7
0	ABƒ´-AL-MUN√†, Mr NƒÅsƒ´f QƒÅsim	27	3rd Class Passenger	2699 £18 15s 9d	Cherbourg	NaN	15	NaN
1	ABBING, Mr Anthony	42	3rd Class Passenger	5547 £7 11s	Southampton	Blacksmith	NaN	NaN
2	ABBOTT, Mrs Rhoda Mary 'Rosa'	39	3rd Class Passenger	CA2673 £20 5s	Southampton	NaN	A	NaN
3	ABBOTT, Mr Rossmore Edward	16	3rd Class Passenger	CA2673 £20 5s	Southampton	Jeweller	[190]	NaN
4	ABBOTT, Mr Eugene Joseph	13	3rd Class Passenger	CA2673 £20 5s	Southampton	Scholar	NaN	NaN
5	ABBOTT, Mr Ernest Owen	21	Victualling Crew	NaN	Southampton	Lounge Pantry Steward	NaN	NaN
6	ABELSETH, Miss Karen Marie	16	3rd Class Passenger	348125 £7 13s	Southampton	NaN	16	NaN
7	ABELSETH, Mr Olaus J√∏rgensen	25	3rd Class Passenger	348122 £7 13s	Southampton	Farmer	A	NaN
8	ABELSON, Mr Samuel	30	2nd Class Passenger	3381 £24	Cherbourg	NaN	NaN	NaN
9	ABELSON, Mrs Hannah	28	2nd Class Passenger	3381 £24	Cherbourg	NaN	10	NaN

[8]
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2456 entries, 0 to 2455
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Name         2456 non-null   object
 1   Age          2427 non-null   object
 2   Class/Dept   2456 non-null   object
 3   Ticket       1348 non-null   object
 4   Joined       2456 non-null   object
 5   Job          1809 non-null   object
 6   Boat [Body]  873 non-null    object
 7   Unnamed: 7   0 non-null      float64
dtypes: float64(1), object(7)
memory usage: 153.6+ KB
```

[9]
```
import re
data['Name']=data['Name'].map(str).apply(lambda x:x.encode('utf-8').decode('ascii','ignore'))
data['Boat [Body]']=data['Boat [Body]'].map(str).apply(lambda x:x.encode('utf-8').decode('ascii','ignore'))
```

[10]
```
data.head(10)
df=data
```

[11]
```
def proc_age(value):
  if'm' in value:
    return float(re.findall(r'-?\d+\.?\d*',value)[0])/12
  else:
    return(float(value))

data['Age']= data['Age'].map(str).apply(proc_age)
data.head()
```

	Name	Age	Class/Dept	Ticket	Joined	Job	Boat [Body]	Unnamed: 7
0	AB-AL-MUN, Mr Nsf Qsim	27.0	3rd Class Passenger	2699 £18 15s 9d	Cherbourg	NaN	15	NaN
1	ABBING, Mr Anthony	42.0	3rd Class Passenger	5547 £7 11s	Southampton	Blacksmith	nan	NaN
2	ABBOTT, Mrs Rhoda Mary 'Rosa'	39.0	3rd Class Passenger	CA2673 £20 5s	Southampton	NaN	A	NaN
3	ABBOTT, Mr Rossmore Edward	16.0	3rd Class Passenger	CA2673 £20 5s	Southampton	Jeweller	[190]	NaN
4	ABBOTT, Mr Eugene Joseph	13.0	3rd Class Passenger	CA2673 £20 5s	Southampton	Scholar	nan	NaN

[12]
```
data_trim = data[['Name','Age','Class/Dept','Boat [Body]']]
data_trim.head()
```

	Name	Age	Class/Dept	Boat [Body]
0	AB-AL-MUN, Mr Nsf Qsim	27.0	3rd Class Passenger	15
1	ABBING, Mr Anthony	42.0	3rd Class Passenger	nan
2	ABBOTT, Mrs Rhoda Mary 'Rosa'	39.0	3rd Class Passenger	A
3	ABBOTT, Mr Rossmore Edward	16.0	3rd Class Passenger	[190]
4	ABBOTT, Mr Eugene Joseph	13.0	3rd Class Passenger	nan

[13]
```
def categories_person(value):
  if'PASENGER' in value.upper():
    return 'Passenger'
  else:
    return 'Crew'

data_trim['Crew/Passenger']=data_trim['Class/Dept'].map(str).apply(categories_person)
data_trim.head()
```

```
<ipython-input-13-56c8aa19ae24>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data_trim['Crew/Passenger']=data_trim['Class/Dept'].map(str).apply(categories_person)
```

	Name	Age	Class/Dept	Boat [Body]	Crew/Passenger
0	AB-AL-MUN, Mr Nsf Qsim	27.0	3rd Class Passenger	15	Crew
1	ABBING, Mr Anthony	42.0	3rd Class Passenger	nan	Crew
2	ABBOTT, Mrs Rhoda Mary 'Rosa'	39.0	3rd Class Passenger	A	Crew
3	ABBOTT, Mr Rossmore Edward	16.0	3rd Class Passenger	[190]	Crew
4	ABBOTT, Mr Eugene Joseph	13.0	3rd Class Passenger	nan	Crew

[14]
```
def categories_age(age):
  if age>18:
    return 'Adult'
  else:
    return 'Child'

data_trim['Adult/Chld']=data_trim['Age'].apply(categories_age)
data_trim.head()
```

```
<ipython-input-14-61d4e648fffd>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data_trim['Adult/Chld']=data_trim['Age'].apply(categories_age)
```

	Name	Age	Class/Dept	Boat [Body]	Crew/Passenger	Adult/Chld
0	AB-AL-MUN, Mr Nsf Qsim	27.0	3rd Class Passenger	15	Crew	Adult
1	ABBING, Mr Anthony	42.0	3rd Class Passenger	nan	Crew	Adult
2	ABBOTT, Mrs Rhoda Mary 'Rosa'	39.0	3rd Class Passenger	A	Crew	Adult
3	ABBOTT, Mr Rossmore Edward	16.0	3rd Class Passenger	[190]	Crew	Child
4	ABBOTT, Mr Eugene Joseph	13.0	3rd Class Passenger	nan	Crew	Child

[15]
```
def categories_gender(name):
  firstname= name[name.index(',')+2:]
  saluation = firstname.split(" ")[0]
  if saluation.upper() in ['MR','MASTER']:
    return 'Male'
  else:
    return 'Female'

data_trim['Gender']=data_trim['Name'].map(str).apply(categories_gender)
data_trim.head()
```

```
<ipython-input-15-a32d13c5d6ee>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data_trim['Gender']=data_trim['Name'].map(str).apply(categories_gender)
```

	Name	Age	Class/Dept	Boat [Body]	Crew/Passenger	Adult/Chld	Gender
0	AB-AL-MUN, Mr Nsf Qsim	27.0	3rd Class Passenger	15	Crew	Adult	Male
1	ABBING, Mr Anthony	42.0	3rd Class Passenger	nan	Crew	Adult	Male
2	ABBOTT, Mrs Rhoda Mary 'Rosa'	39.0	3rd Class Passenger	A	Crew	Adult	Female
3	ABBOTT, Mr Rossmore Edward	16.0	3rd Class Passenger	[190]	Crew	Child	Male
4	ABBOTT, Mr Eugene Joseph	13.0	3rd Class Passenger	nan	Crew	Child	Male

[16]
```
def categories_survival(value):
  if value.strip()=='nan' or '[' in value:
    return 0
  else:
    return 1

data_trim['Survival']=data_trim['Boat [Body]'].map(str).apply(categories_survival)
data_trim.head(15)
```

	Name	Age	Class/Dept	Boat [Body]	Crew/Passenger	Adult/Chld	Gender	Survival
0	AB-AL-MUN, Mr Nsf Qsim	27.0	3rd Class Passenger	15	Crew	Adult	Male	1
1	ABBING, Mr Anthony	42.0	3rd Class Passenger	nan	Crew	Adult	Male	0
2	ABBOTT, Mrs Rhoda Mary 'Rosa'	39.0	3rd Class Passenger	A	Crew	Adult	Female	1
3	ABBOTT, Mr Rossmore Edward	16.0	3rd Class Passenger	[190]	Crew	Child	Male	0
4	ABBOTT, Mr Eugene Joseph	13.0	3rd Class Passenger	nan	Crew	Child	Male	0
5	ABBOTT, Mr Ernest Owen	21.0	Victualling Crew	nan	Crew	Adult	Male	0
6	ABELSETH, Miss Karen Marie	16.0	3rd Class Passenger	16	Crew	Child	Female	1
7	ABELSETH, Mr Olaus Jrgensen	25.0	3rd Class Passenger	A	Crew	Adult	Male	1
8	ABELSON, Mr Samuel	30.0	2nd Class Passenger	nan	Crew	Adult	Male	0
9	ABELSON, Mrs Hannah	28.0	2nd Class Passenger	10	Crew	Adult	Female	1
10	ABRAHAMSSON, Mr Abraham August Johannes	20.0	3rd Class Passenger	15	Crew	Adult	Male	1
11	ABRAMS, Mr William Thomas	34.0	Engineering Crew	nan	Crew	Adult	Male	0
12	ADAMS, Mr John	27.0	3rd Class Passenger	[103]	Crew	Adult	Male	0
13	ADAMS, Mr Robert John	26.0	Engineering Crew	nan	Crew	Adult	Male	0
14	AHIER, Mr Percy Snowden	20.0	Victualling Crew	nan	Crew	Adult	Male	0

Along with that try to analyze the given data by answering the below mentioned questions:

[19]
```
from google.colab import files
datafile = files.upload()
```

Choose Files train.csv

- **train.csv**(text/csv) - 61194 bytes, last modified: 26/4/2024 - 100% done

```
Saving train.csv to train.csv
```

[20]
```
data=pd.read_csv('train.csv')
data.head()
```

	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

Next steps: View recommended plots

[21]
```
mean_age = data['Age'].mean()
data['Age'].fillna(mean_age, inplace=True)
data_trim = data[['Name','Age','Pclass','Survived']]
data_trim.head()
```

	Name	Age	Pclass	Survived
0	Braund, Mr. Owen Harris	22.0	3	0
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1	1
2	Heikkinen, Miss. Laina	26.0	3	1
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	1
4	Allen, Mr. William Henry	35.0	3	0

Next steps: View recommended plots

[22]
```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
```

[23]
```
X = data_trim[['Age', 'Pclass']]
y = data_trim['Survived']

# Train Decision Tree Classifier
clf = DecisionTreeClassifier()
clf.fit(X, y)

# Test the model (predict survival for new data)
new_data = {
    "Name": ["New Passenger"],
    "Age": [30.0],
    "Pclass": [2]
}

new_data_trim = pd.DataFrame(new_data)
X_new = new_data_trim[['Age', 'Pclass']]
prediction = clf.predict(X_new)

if prediction[0] == 1:
    print("The passenger is predicted to have survived.")
else:
    print("The passenger is predicted to have not survived.")
```

```
The passenger is predicted to have not survived.
```

What is the survival percentage of Crew members ?

[24]
```
crew_survival_percentage = data_trim['Survived'].mean() * 100
print("Survival percentage of Crew members:", crew_survival_percentage)
```

```
Survival percentage of Crew members: 38.38383838383838
```

What is the survival percentage of first class passengers?

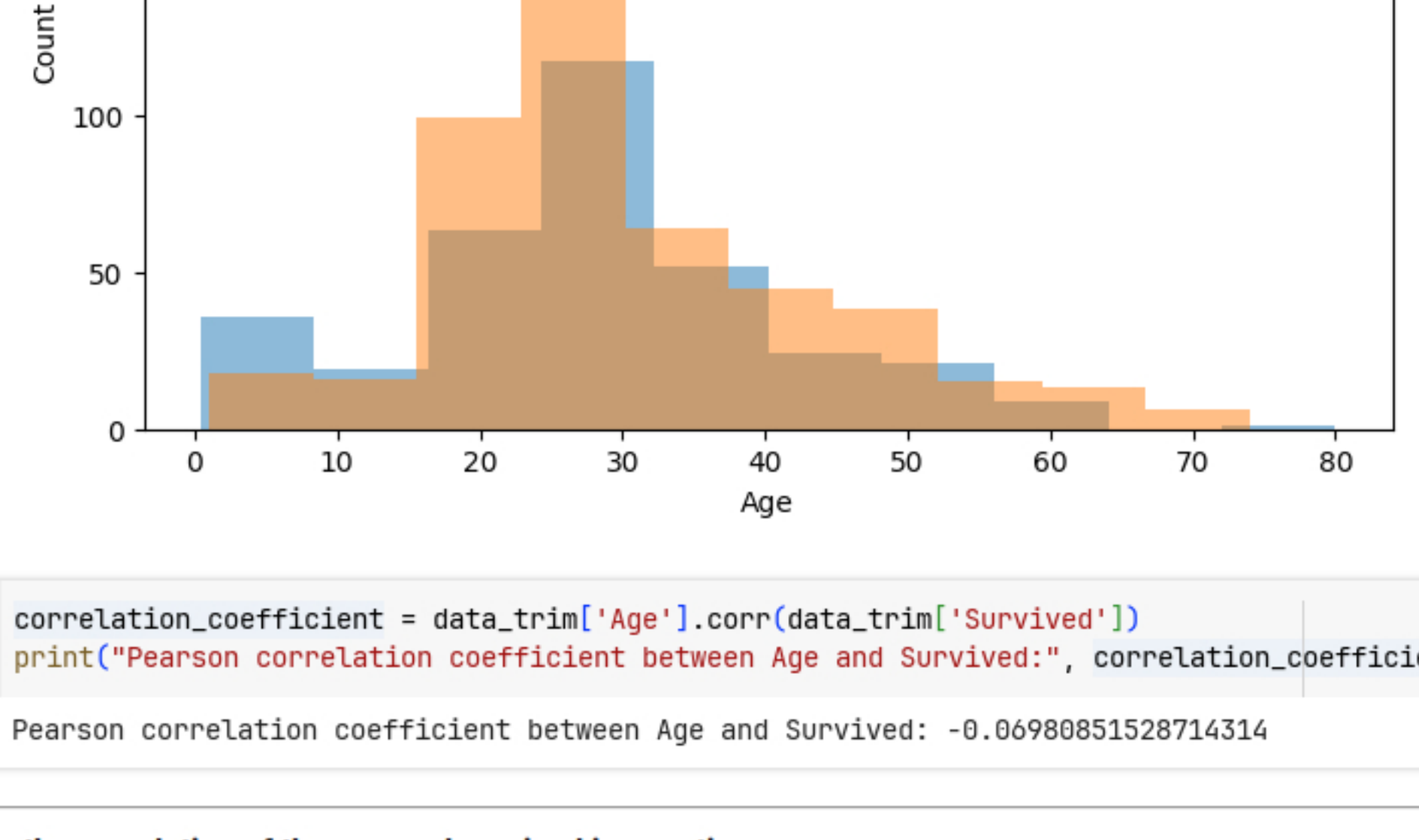
[25]
```
first_class_survival_percentage = data_trim[data_trim['Pclass'] == 1]['Survived'].mean() * 100
print("Survival percentage of first class passengers:", first_class_survival_percentage.round(2))
```

```
Survival percentage of first class passengers: 62.96
```

Does age also play a role in survival ? How can you say that ?

[26]
```
survivors_mean_age = data_trim[data_trim['Survived'] == 1]['Age'].mean()
non_survivors_mean_age = data_trim[data_trim['Survived'] == 0]['Age'].mean()
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))
plt.hist(data_trim[data_trim['Survived'] == 1]['Age'], bins=10, alpha=0.5, label='Survived')
plt.hist(data_trim[data_trim['Survived'] == 0]['Age'], bins=10, alpha=0.5, label='Did not survive')
plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Age Distribution of Survivors vs Non-survivors')
plt.legend()
plt.show()
```




```
correlation_coefficient = data_trim['Age'].corr(data_trim['Survived'])
print("Pearson correlation coefficient between Age and Survived:", correlation_coefficient)
```

```
Pearson correlation coefficient between Age and Survived: -0.06980851528714314
```

Here, the correlation of the age and survived is negative.

Hence, it can be said that more age means less chances of being survived and vice versa, but as it near to zero, age has negligible effect of survival chances