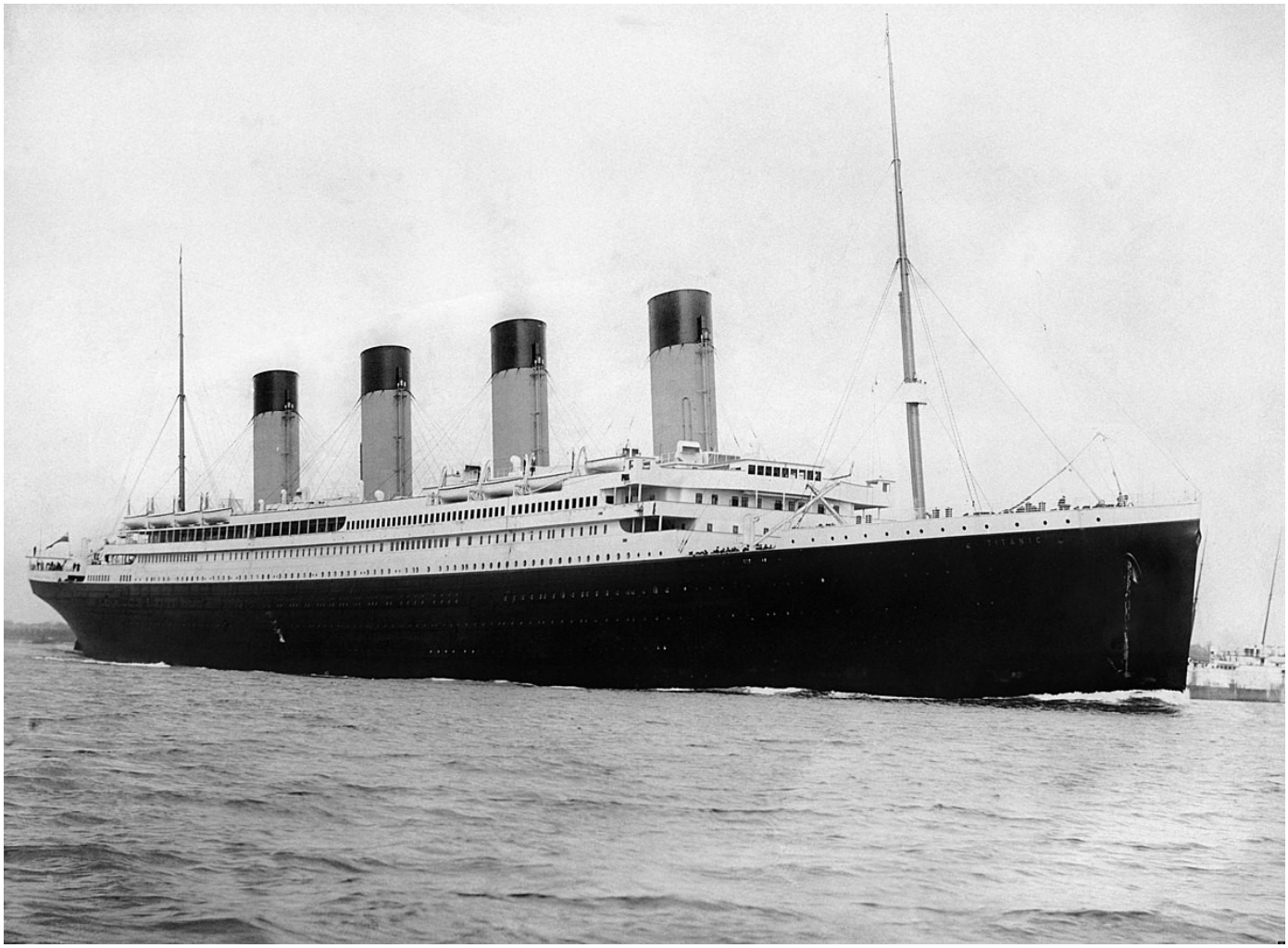


The titanic ML competition is the competition with the most participants on kaggle

I first tried the challenge more than a years ago. My score was quite low at the time, 0.74641. This notebook has a slightly higher score but there's still many improvements that can be made. The first day, this notebook had a score of 0.75598. After a few more edits, the score climbed to 0.76794

In [1]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
/kaggle/input/titanic/train.csv
/kaggle/input/titanic/test.csv
/kaggle/input/titanic/gender_submission.csv
```

In [2]:

```
train = pd.read_csv('/kaggle/input/titanic/train.csv')
train.head()
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	7
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	5
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8

In [3]:

```
test = pd.read_csv('/kaggle/input/titanic/test.csv')
test.head()
```

Out[3]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	N
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	N
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	N
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	N
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	N

In [4]:

```
train.drop(['PassengerId', 'Ticket'], axis=1, inplace=True)
```

In [5]:

```
test.drop('Ticket', axis=1, inplace=True)
```

In [6]:

```
train.head()
```

Out[6]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarke
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	N+N	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S

I read that extracting the deck from the cabin number was useful for the prediction. Here is the plan of the different decks

In [7]:

```
train['Deck'] = train['Cabin'].str[0] #.replace('\d+', '')
train.head()
```

Out[7]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarke
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S

In [8]:

```
train.Deck.value_counts()
```

Out[8]:

```
C    59
B    47
D    33
E    32
A    15
F    13
G     4
T     1
Name: Deck, dtype: int64
```

In [9]:

```
test['Deck'] = test['Cabin'].str[0] #.replace('\d+', '')
test.head()
```

Out[9]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Emba
0	892	3	Kelly, Mr. James	male	34.5	0	0	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	12.2875	NaN	S

In [10]:

```
test.Deck.value_counts()
```

Out[10]:

```
C    35
B    18
D    13
E     9
F     8
A     7
G     1
Name: Deck, dtype: int64
```

In [11]:

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

In [12]:

```
train.head()
```

Out[12]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S

In [13]:

```
train.isna().sum()
```

Out[13]:

```
Survived      0
Pclass        0
Name          0
Sex           0
Age         177
SibSp         0
Parch         0
Fare          0
Embarked      2
Deck        687
dtype: int64
```

In [14]:

```
test.isna().sum()
```

Out[14]:

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Fare             1
Embarked         0
Deck           327
dtype: int64
```

In the train set we have 177 missing values for the age, 2 for the embarkment place, and 687 for the deck.

In the test set, we have 86 missing values for the age, 1 for the fare, and 327 for the deck

In [15]:

```
train[train['Deck'] == 'T']
```

Out[15]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	Deck
339	0	1	Blackwell, Mr. Stephen Weart	male	45.0	0	0	35.5	S	T

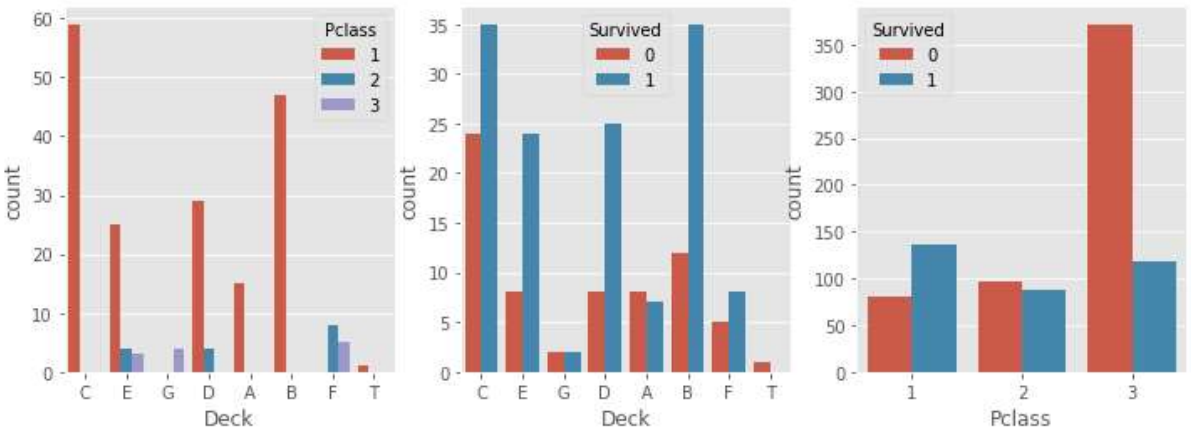
Mr. Blackwell is the only passenger listed on the 'T' deck. The deck does not seems to appear on the plan. Further research show that his cabin was on the boat deck, the higher layer of the titanic

In [16]:

```
plt.style.use('ggplot')
```

In [17]:

```
fig, ax = plt.subplots(1,3, figsize=(12, 4))

sns.countplot(ax=ax[0], x=train['Deck'], hue=train['Pclass'])
sns.countplot(ax=ax[1], x=train['Deck'], hue=train['Survived'])
sns.countplot(ax=ax[2], x=train['Pclass'], hue=train['Survived'])

plt.show()
```



Despite the large number of missing data, we can see a trend. The higher the deck, the higher the class

In [18]:

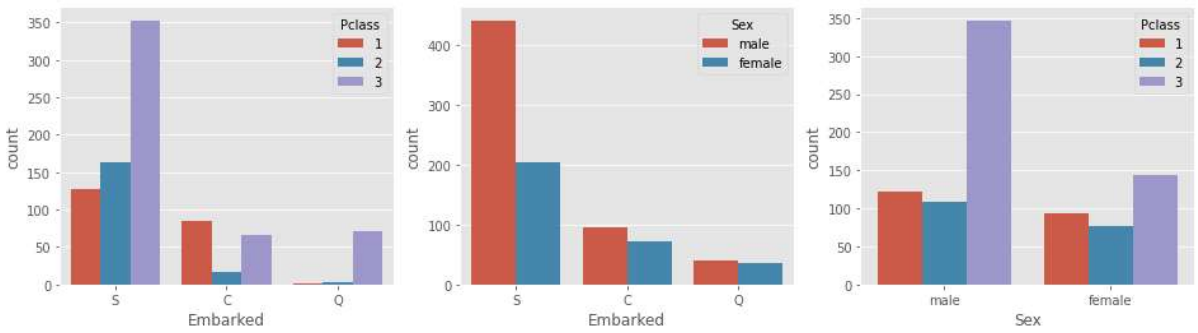
```
train[train.Embarked.isna()]['Embarked'] = 'S'
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: Set
tingWithCopyWarning:
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/pan
das-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-
copy
  """Entry point for launching an IPython kernel.
```

We know there are three embarkment places. Southampton, Cherbourg, and Queenstown. a quick google search told us that Amelie Icard and Marth Evelyn Stone both embarked in Southampton

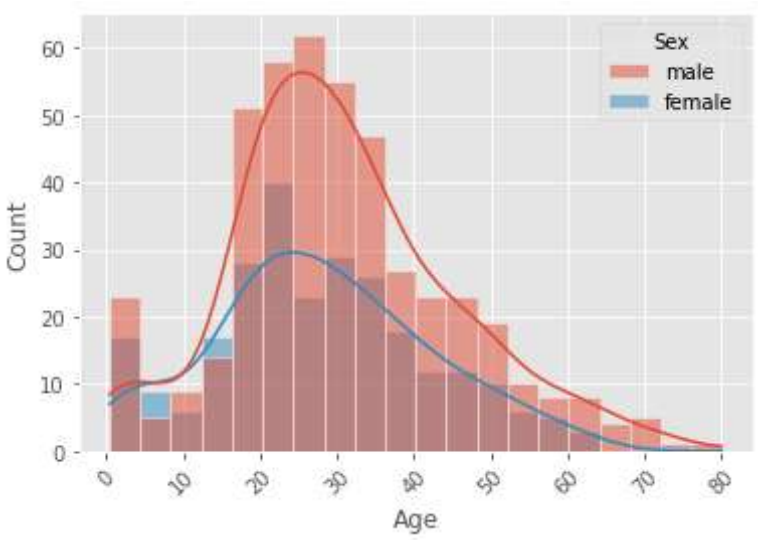
In [19]:

```
fig, ax = plt.subplots(1,3, figsize=(16, 4))

sns.countplot(ax=ax[0], x=train['Embarked'], hue=train['Pclass'])

sns.countplot(ax=ax[1], x=train['Embarked'], hue=train['Sex'])

sns.countplot(ax=ax[2], x=train['Sex'], hue=train['Pclass'])

plt.show()
```



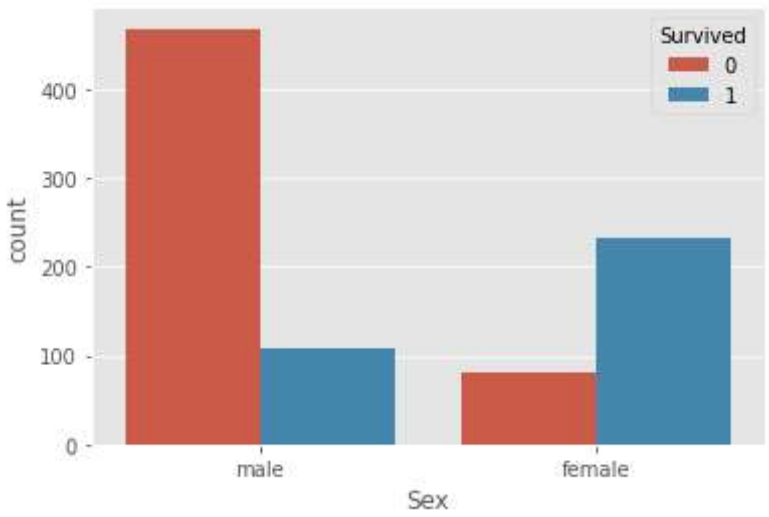
In [20]:

```
sns.histplot(data=train, x='Age', hue='Sex', bins=20, kde=True)
plt.xticks(rotation=45)
plt.show()
```



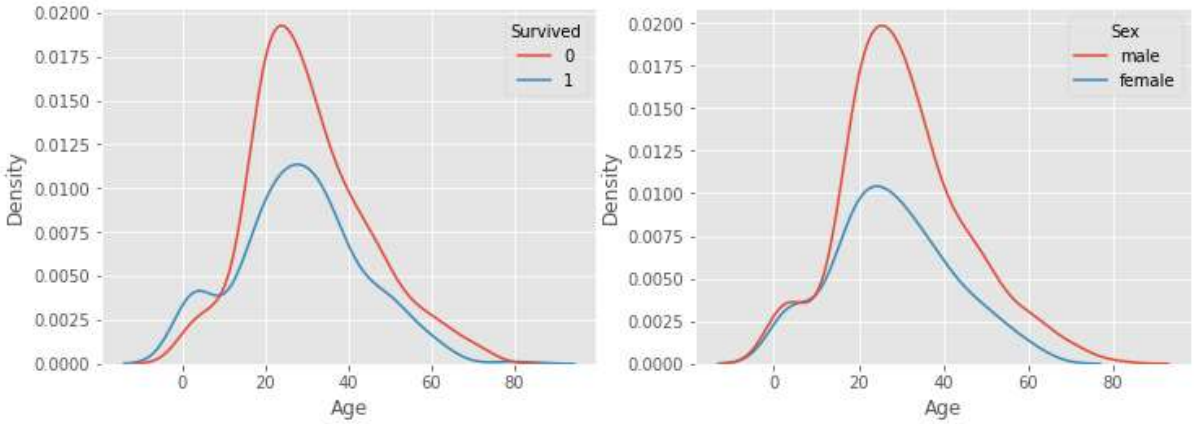
In [21]:

```
sns.countplot(data=train, x='Sex', hue='Survived')

plt.show()
```



In [22]:

```
fig, ax = plt.subplots(1, 2, figsize=(12, 4))

sns.kdeplot(ax=ax[0], data=train, x='Age', hue='Survived')

sns.kdeplot(ax=ax[1], data=train, x='Age', hue='Sex')

plt.show()
```



In [23]:

```
train['Title'] = train['Name'].str.split(',')
train['Title'] = train['Title'].str[1]
train['Title'] = train['Title'].str.split('.')
train['Title'] = train['Title'].str[0]
train.head()
```

Out[23]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	Deck
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S	NaN
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S	NaN
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S	C
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S	NaN

In [24]:

```
test['Title'] = test['Name'].str.split(',')
test['Title'] = test['Title'].str[1]
test['Title'] = test['Title'].str.split('.')
test['Title'] = test['Title'].str[0]
test.head()
```

Out[24]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	D
0	892	3	Kelly, Mr. James	male	34.5	0	0	7.8292	Q	N
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	7.0000	S	N
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	9.6875	Q	N
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	8.6625	S	N
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	12.2875	S	N

In [25]:

```
train.Title.value_counts()
```

Out[25]:

```
Mr              517
Miss            182
Mrs             125
Master           40
Dr                7
Rev               6
Mlle              2
Major             2
Col               2
the Countess      1
Capt              1
Ms                1
Sir               1
Lady              1
Mme               1
Don               1
Jonkheer          1
Name: Title, dtype: int64
```

In [26]:

```
test.Title.value_counts()
```

Out[26]:

```
Mr        240
Miss       78
Mrs        72
Master     21
Col         2
Rev         2
Ms          1
Dr          1
Dona        1
Name: Title, dtype: int64
```

is there any use of knowing the title of someone for the model. Yes, it can help us determine the unknown age. We can also exclude some outliers who might lower the quality of our model

In [27]:

```
train.Title.unique()
```

Out[27]:

```
array([' Mr', ' Mrs', ' Miss', ' Master', ' Don', ' Rev', ' Dr', '
Mme',
       ' Ms', ' Major', ' Lady', ' Sir', ' Mlle', ' Col', ' Capt',
       ' the Countess', ' Jonkheer'], dtype=object)
```

In [28]:

```
train[train['Title'] == ' Capt']
```

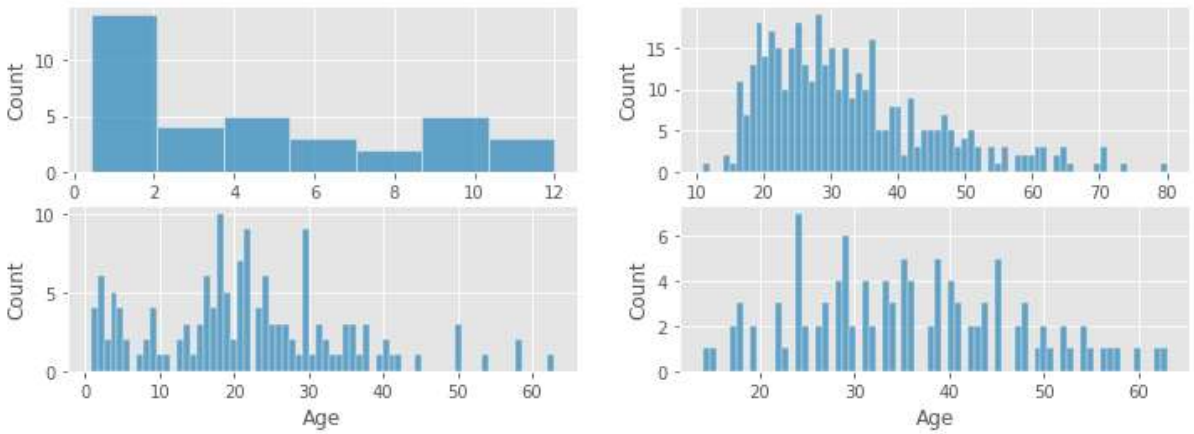
Out[28]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	Deck
745	0	1	Crosby, Capt. Edward Gifford	male	70.0	1	1	71.0	S	B

note: Capt. Edward Gifford Crosby is not the captain of the Titanic, he is one of passenger

In [29]:

```
master = train[train['Title'] == ' Master']
master.head()
mr = train[train['Title'] == ' Mr']
miss = train[train['Title'] == ' Miss']
mrs = train[train['Title'] == ' Mrs']

fig, ax = plt.subplots(2,2, figsize=(12, 4))

sns.histplot(ax=ax[0,0], data=master, x='Age')
sns.histplot(ax=ax[0,1], data=mr, x='Age', bins=70)
sns.histplot(ax=ax[1,0], data=miss, x='Age', bins=70)
sns.histplot(ax=ax[1,1], data=mrs, x='Age', bins=70)

plt.show()
```



There is a clear age separation between Master and Mr.

However it is not as clear between Miss and Mrs

In [30]:

```
train[(train['Title'] == ' Master') & (train['Age'].isna())]
```

Out[30]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	Dec
65	1	3	Moubarek, Master. Gerios	male	NaN	1	1	15.2458	C	NaN
159	0	3	Sage, Master. Thomas Henry	male	NaN	8	2	69.5500	S	NaN
176	0	3	Lefebre, Master. Henry Forbes	male	NaN	3	1	25.4667	S	NaN
709	1	3	Moubarek, Master. Halim Gonios ("William George")	male	NaN	1	1	15.2458	C	NaN

In [31]:

```
test[(test['Title'] == ' Master') & (test['Age'].isna())]
```

Out[31]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	D
244	1136	3	Johnston, Master. William Arthur Willie""	male	NaN	1	2	23.4500	S	N
339	1231	3	Betros, Master. Seman	male	NaN	0	0	7.2292	C	N
344	1236	3	van Billiard, Master. James William	male	NaN	1	1	14.5000	S	N
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	22.3583	C	N

In [32]:

```
test['AgeCategory'] = 'Adult'
test['AgeCategory'][test['Age'] <= 12] = 'Child'
test['AgeCategory'][test['Title'] == ' Master'] = 'Child'
test.head(20)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:2: Set
tingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pan
das-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-
copy

/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: Set
tingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pan
das-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-
copy
  This is separate from the ipykernel package so we can avoid doing
imports until
```

Out[32]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	7.8292	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	7.0000	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	9.6875	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	8.6625	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	12.2875	S
5	897	3	Svensson, Mr. Johan Cervin	male	14.0	0	0	9.2250	S
6	898	3	Connolly, Miss. Kate	female	30.0	0	0	7.6292	Q
7	899	2	Caldwell, Mr. Albert Francis	male	26.0	1	1	29.0000	S
8	900	3	Abrahim, Mrs. Joseph (Sophie Halaut Easu)	female	18.0	0	0	7.2292	C
9	901	3	Davies, Mr. John Samuel	male	21.0	2	0	24.1500	S
10	902	3	Ilieff, Mr. Ylio	male	NaN	0	0	7.8958	S
11	903	1	Jones, Mr. Charles Cresson	male	46.0	0	0	26.0000	S
12	904	1	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female	23.0	1	0	82.2667	S
13	905	2	Howard, Mr. Benjamin	male	63.0	1	0	26.0000	S
14	906	1	Chaffee, Mrs. Herbert Fuller (Carrie Constance...	female	47.0	1	0	61.1750	S
15	907	2	del Carlo, Mrs. Sebastiano (Argenia Genovesi)	female	24.0	1	0	27.7208	C

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked
16	908	2	Keane, Mr. Daniel	male	35.0	0	0	12.3500	Q
17	909	3	Assaf, Mr. Gerios	male	21.0	0	0	7.2250	C
18	910	3	Ilmakangas, Miss. Ida Livija	female	27.0	1	0	7.9250	S
19	911	3	Assaf Khalil, Mrs. Mariana (Miriam")"	female	45.0	0	0	7.2250	C

In [33]:

```
train['AgeCategory'] = 'Adult'
train['AgeCategory'][train['Age'] <= 12] = 'Child'
train['AgeCategory'][train['Title'] == ' Master'] = 'Child'
#train[train['Age'] > 12]['AgeCategory'] = 'Adult'
train.head(10)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:2: Set
tingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pan
das-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-
copy

/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: Set
tingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pan
das-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-
copy
  This is separate from the ipykernel package so we can avoid doing
imports until
```

Out[33]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	Decl
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S	NaN
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S	NaN
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S	C
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S	NaN
5	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	Q	NaN
6	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	S	E
7	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	S	NaN
8	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	S	NaN
9	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	C	NaN

In [34]:

```
train.drop(['Name', 'Title', 'Deck'], axis=1, inplace=True)
test.drop(['Name', 'Title', 'Deck'], axis=1, inplace=True)
```

In [35]:

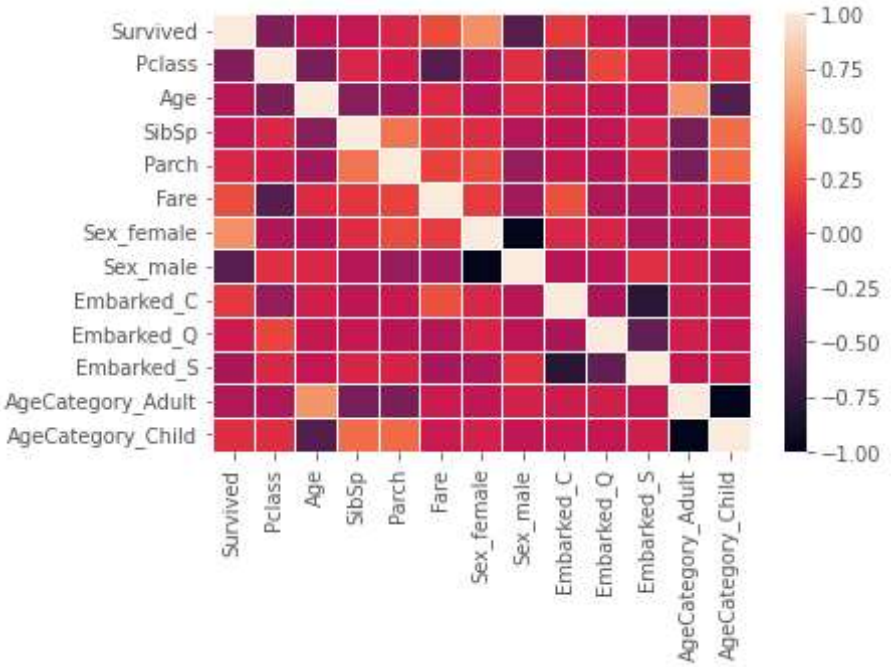
```
train = pd.get_dummies(train)
test = pd.get_dummies(test)
test.head()
```

Out[35]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare	Sex_female	Sex_male	Embarked
0	892	3	34.5	0	0	7.8292	0	1	0
1	893	3	47.0	1	0	7.0000	1	0	0
2	894	2	62.0	0	0	9.6875	0	1	0
3	895	3	27.0	0	0	8.6625	0	1	0
4	896	3	22.0	1	1	12.2875	1	0	0

In [36]:

```
sns.heatmap(train.corr(), linewidth=0.5)
plt.show()
```



In [37]:

```
train.corr()
```

Out[37]:

	Survived	Pclass	Age	SibSp	Parch	Fare
Survived	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.54950
Age	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000
Sex_female	0.543351	-0.131900	-0.093254	0.114631	0.245489	0.182333
Sex_male	-0.543351	0.131900	0.093254	-0.114631	-0.245489	-0.18233
Embarked_C	0.168240	-0.243292	0.036261	-0.059528	-0.011069	0.269335
Embarked_Q	0.003650	0.221009	-0.022405	-0.026354	-0.081228	-0.11721
Embarked_S	-0.155660	0.081720	-0.032523	0.070941	0.063036	-0.16660
AgeCategory_Adult	-0.117636	-0.124732	0.561675	-0.385491	-0.376629	0.004070
AgeCategory_Child	0.117636	0.124732	-0.561675	0.385491	0.376629	-0.00407

let's give a try with our age categories because it have a higher correlation with the survival compare to the age

Age/sex	Class/crew	Number aboard	Number saved	Number lost	Percentage saved	Percentage lost
Children	First Class	6	5	1	83%	17%
	Second Class	24	24	0	100%	0%
	Third Class	79	27	52	34%	66%
Women	First Class	144	140	4	97%	3%
	Second Class	93	80	13	86%	14%
	Third Class	165	76	89	46%	54%
	Crew	23	20	3	87%	13%
Men	First Class	175	57	118	33%	67%
	Second Class	168	14	154	8%	92%
	Third Class	462	75	387	16%	84%
	Crew	885	192	693	22%	78%
Total		**2224**	**710**	**1514**	**32%**	**68%**

Machine Learning Model

In [38]:

```
y_train = train['Survived']
X_train = train.drop(['Age', 'Survived'], axis=1)
X_test = test.drop(['Age', 'PassengerId'], axis=1)

X_train.head()
```

Out[38]:

	Pclass	SibSp	Parch	Fare	Sex_female	Sex_male	Embarked_C	Embarked_Q
0	3	1	0	7.2500	0	1	0	0
1	1	1	0	71.2833	1	0	1	0
2	3	0	0	7.9250	1	0	0	0
3	1	1	0	53.1000	1	0	0	0
4	3	0	0	8.0500	0	1	0	0

In [39]:

```
y_train.head()
```

Out[39]:

```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

In [40]:

```
X_test.head()
```

Out[40]:

	Pclass	SibSp	Parch	Fare	Sex_female	Sex_male	Embarked_C	Embarked_Q	Er
0	3	0	0	7.8292	0	1	0	1	0
1	3	1	0	7.0000	1	0	0	0	1
2	2	0	0	9.6875	0	1	0	1	0
3	3	0	0	8.6625	0	1	0	0	1
4	3	1	1	12.2875	1	0	0	0	1

In [41]:

```
X_test.columns
```

Out[41]:

```
Index(['Pclass', 'SibSp', 'Parch', 'Fare', 'Sex_female', 'Sex_mal
e',
       'Embarked_C', 'Embarked_Q', 'Embarked_S', 'AgeCategory_Adul
t',
       'AgeCategory_Child'],
      dtype='object')
```

In [42]:

```
X_test.isna().sum()
```

Out[42]:

```
Pclass               0
SibSp                0
Parch                0
Fare                 1
Sex_female           0
Sex_male             0
Embarked_C           0
Embarked_Q           0
Embarked_S           0
AgeCategory_Adult    0
AgeCategory_Child    0
dtype: int64
```

In [43]:

```
X_test['Fare'][X_test['Fare'].isna()] = X_test['Fare'].median()
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: Set
tingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pan
das-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-
copy
  """Entry point for launching an IPython kernel.
```

In [44]:

```
X_test.isna().sum()
```

Out[44]:

```
Pclass               0
SibSp                0
Parch                0
Fare                 0
Sex_female           0
Sex_male             0
Embarked_C           0
Embarked_Q           0
Embarked_S           0
AgeCategory_Adult    0
AgeCategory_Child    0
dtype: int64
```

In [45]:

```
result = pd.read_csv('/kaggle/input/titanic/gender_submission.csv')
result.head()
```

Out[45]:

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1

In [46]:

```
result.shape
```

Out[46]:

```
(418, 2)
```

In [47]:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

model = DecisionTreeClassifier()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

y_true = result.Survived

score = accuracy_score(y_pred, y_true)
print(f"The accuracy score is {round(score * 100)}")
```

```
The accuracy score is 86
```

In [48]:

```
y_pred_df = pd.DataFrame(y_pred)
y_pred_df
```

Out[48]:

	0
0	0
1	1
2	0
3	0
4	1
...	...
413	0
414	1
415	0
416	0
417	1

418 rows × 1 columns

In [49]:

```
submit_df = test['PassengerId']
submit_df = pd.DataFrame(submit_df)
submit_df['Survived'] = y_pred

submit_df.head()
```

Out[49]:

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1

In [50]:

```
#submission = submit_df.to_csv('submission.csv')
```

To further improve the accuracy, I will try to remove the outliers