```
import numpy as np
```

```
import pandas as pd
```

```
import seaborn as snb
```

```
from matplotlib import pyplot as plt
from matplotlib import style
```

```
from sklearn import linear_model
```

```
from sklearn.linear_model import LogisticRegression
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.linear_model import Perceptron
```

```
from sklearn.linear_model import SGDClassifier
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
 from sklearn.svm import SVC, LinearSVC
```

```
from sklearn.naive_bayes import GaussianNB
```

```
test_df = pd.read_csv("test.csv")
```

```
train_df = pd.read_csv("train.csv")
```

```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
train_df.head(8)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282

```
total = train_df.isnull().sum().sort_values(ascending=False)
```

```
 percent_1 =train_df.isnull().sum()/train_df.isnull().count()*100
```

```
percent_2 = (round(percent_1,1)).sort_values(ascending=False)
```

```
missing_data = pd.concat([total, percent_2], axis=1, keys=['Total',
'%'])
```

```
missing_data.head(5)
```

	Total	%
Cabin	687	77.1

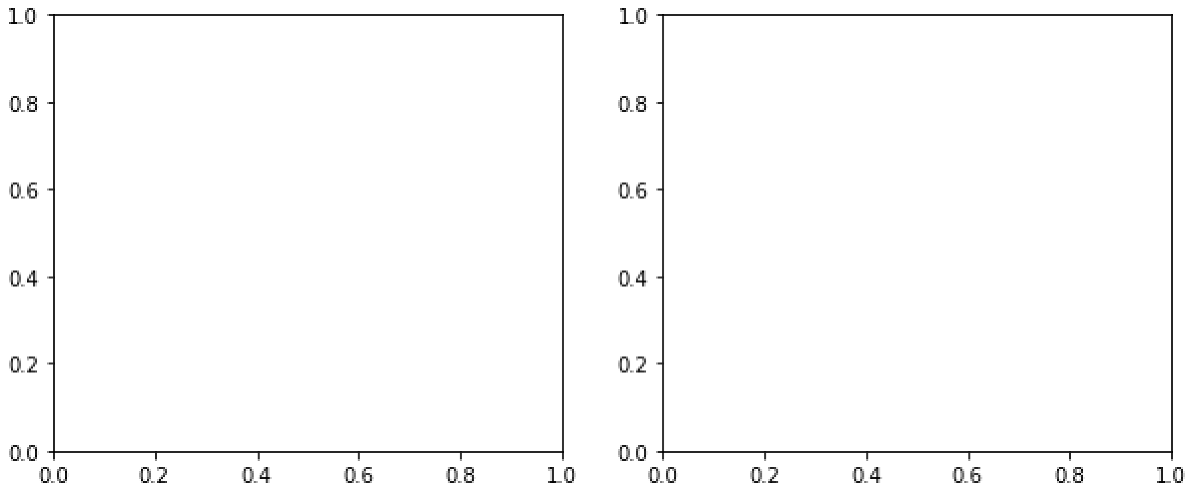
```
train_df.columns.values
```

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

Survived	0	0.0

```
survived = 'survived'
not_survived = 'not survived'
```

```
fig, axes = plt.subplots(nrows=1, ncols=2,figsize=(10, 4))
```



```
women = train_df[train_df['Sex']=='female']
```

```
men = train_df[train_df['Sex']=='male']
```

```
ax = snb.distplot(women[women['Survived']==1].Age.dropna(), bins=18, label = survived,  ax
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
```

```
ax = snb.distplot(women[women['Survived']==0].Age.dropna(), bins=40, label = not_survived,
axes[0], kde =False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
```

```
ax.legend()
```

```
<matplotlib.legend.Legend at 0x7f1dc6fd88d0>
```

```
ax.set_title('Female')
```

```
Text(0.5, 1.0, 'Female')
```

```
ax = snb.distplot(men[men['Survived']==1].Age.dropna(), bins=18, label
= survived, ax = axes[1], kde = False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
```

```
ax = snb.distplot(men[men['Survived']==0].Age.dropna(), bins=40, label
= not_survived, ax = axes[1], kde = False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
```

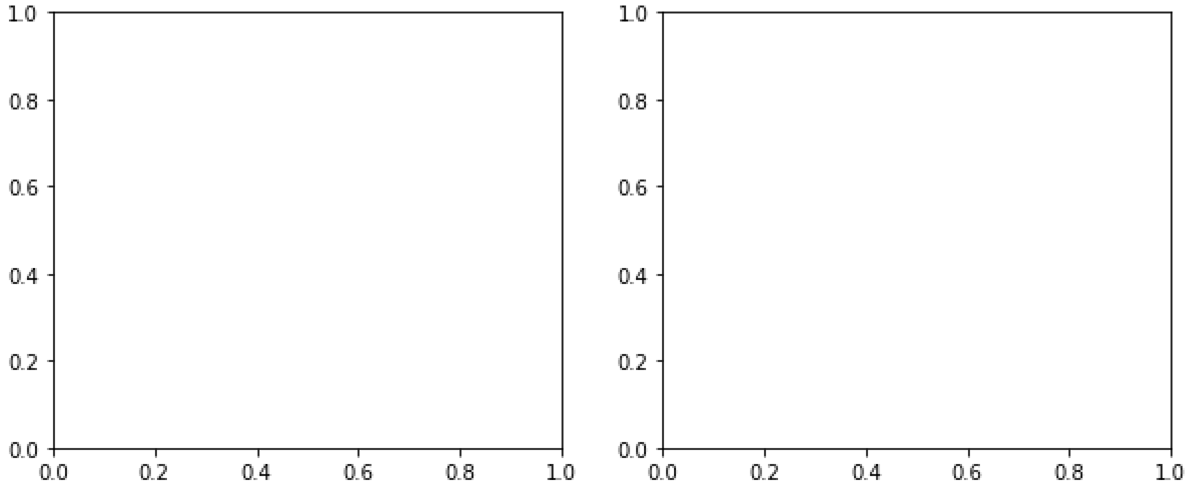
```
ax.legend()
```

```
<matplotlib.legend.Legend at 0x7f1dc6f96f90>
```

```
 ax.set_title('Male')
```

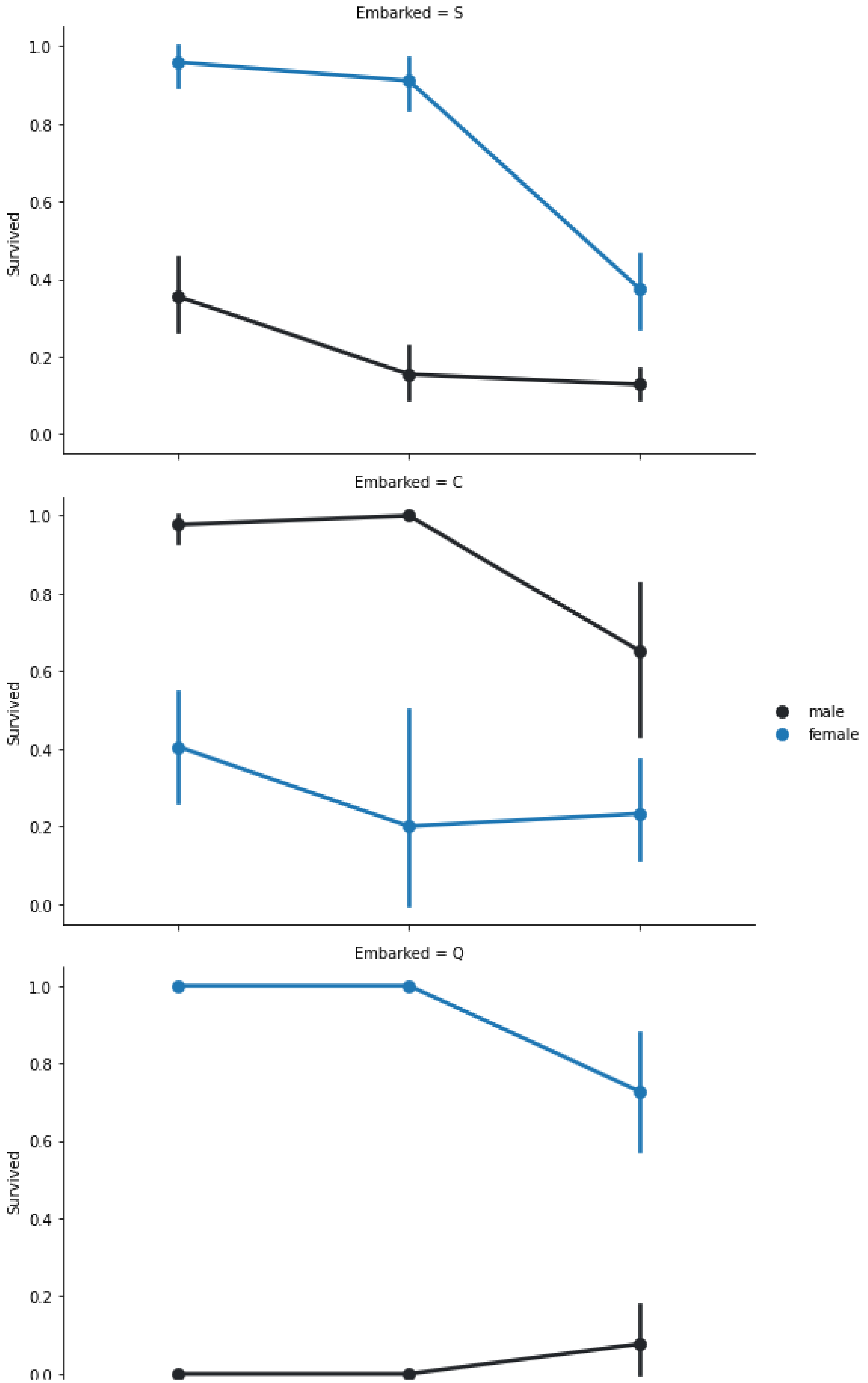
```
Text(0.5, 1.0, 'Male')
```

```
fig, axes = plt.subplots(nrows=1, ncols=2,figsize=(10, 4))
```



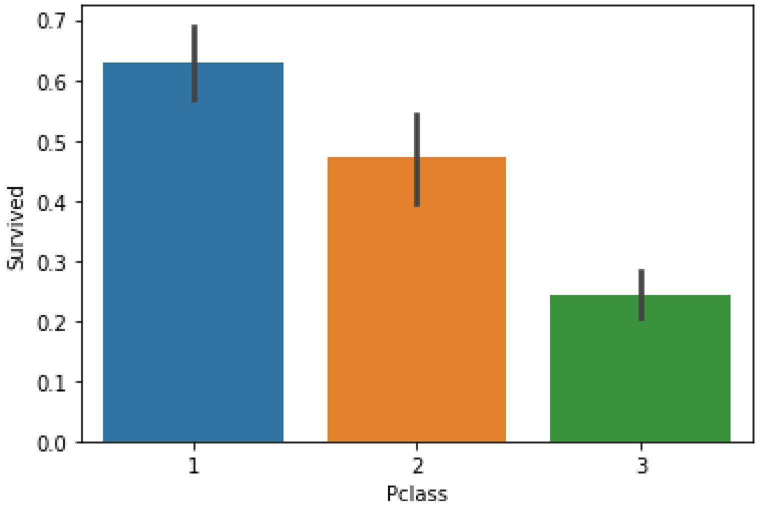
```
FacetGrid = snb.FacetGrid(train_df, row='Embarked', size=4.5, aspect=1.6)
FacetGrid.map(snb.pointplot, 'Pclass', 'Survived', 'Sex',
palette=None, order=None, hue_order=None )
FacetGrid.add_legend()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning: The
  warnings.warn(msg, UserWarning)
<seaborn.axisgrid.FacetGrid at 0x7f1dc6d23dd0>
```



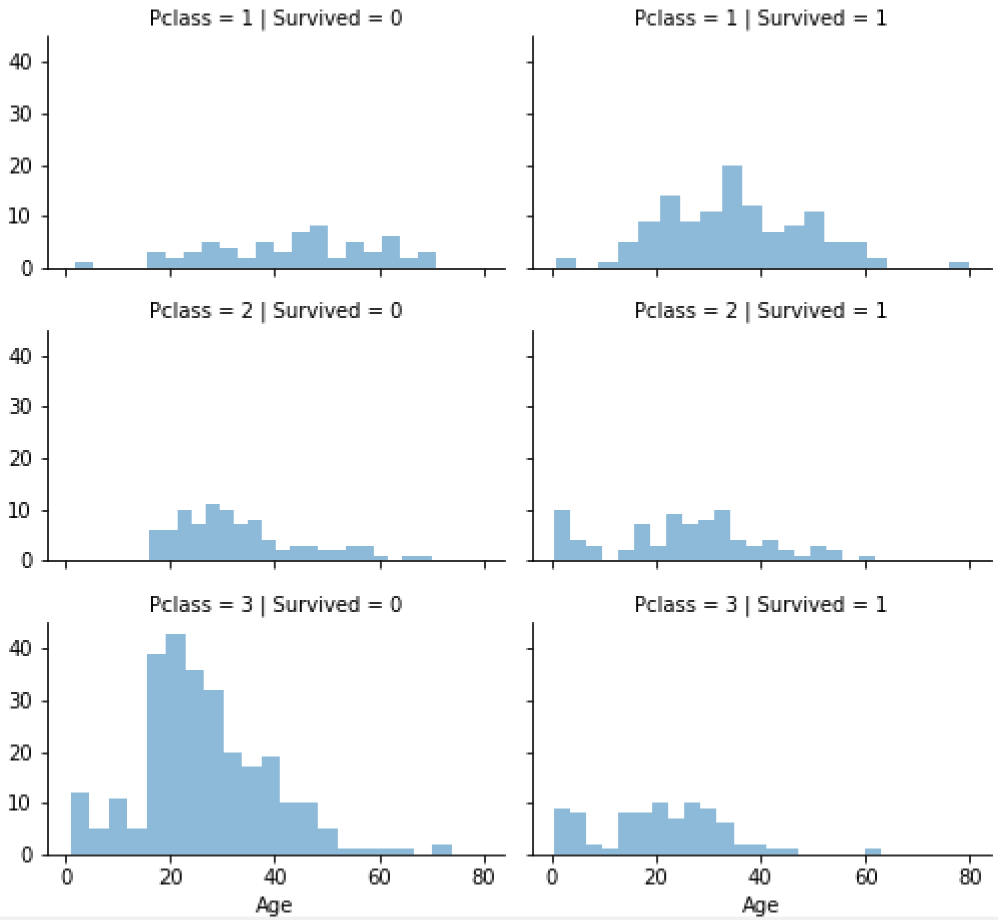
```
snb.barplot(x='Pclass', y='Survived', data=train_df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1dc4319dd0>
```



```
grid = snb.FacetGrid(train_df, col='Survived', row='Pclass', size=2.2, aspect=1.6)
grid.map(plt.hist, 'Age', alpha=.5, bins=20)
grid.add_legend();
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning: The `
  warnings.warn(msg, UserWarning)
```



```
data = [train_df, test_df]


for dataset in data:
        dataset['relatives'] = dataset['SibSp'] + dataset['Parch']
```

```
dataset.loc[dataset['relatives'] > 0, 'not_alone'] =0
```

```
 dataset.loc[dataset['relatives'] == 0, 'not_alone'] = 1
```

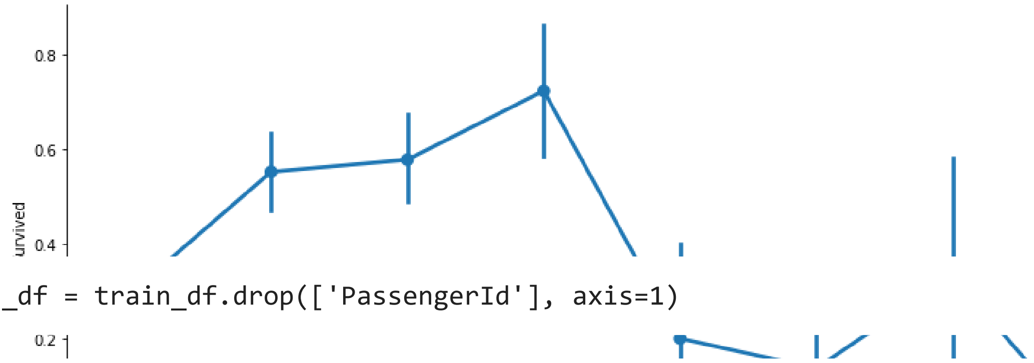
```
dataset['not_alone'] =dataset['not_alone'].astype(int)
```

```
train_df['not_alone'].value_counts()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.7/dist-packages/pandas/core/indexes/base.py in
get_loc(self, key, method, tolerance)
   3360             try:
-> 3361                 return self._engine.get_loc(casted_key)
   3362             except KeyError as err:

                          ⇕ 4 frames

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'not_alone'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.7/dist-packages/pandas/core/indexes/base.py in
get_loc(self, key, method, tolerance)
   3361                 return self._engine.get_loc(casted_key)
   3362             except KeyError as err:
-> 3363                 raise KeyError(key) from err
   3364 
   3365         if is_scalar(key) and isna(key) and not self.hasnans:
```

```
axes = snb.factorplot('relatives','Survived', data=train_df, aspect = 2.5, )
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3717: UserWarning: The
  warnings.warn(msg)
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
  FutureWarning
```



```
train_df = train_df.drop(['PassengerId'], axis=1)
```

```
import re
deck = {"A": 1, "B": 2, "C": 3, "D": 4, "E": 5, "F": 6, "G": 7, "U":
8}


data = [train_df, test_df]


for dataset in data:
   dataset['Cabin'] = dataset['Cabin'].fillna("U0")



dataset['Deck'] = dataset['Cabin'].map(lambda x: re.compile("([azA-Z]+)").search(x).group(



dataset['Deck'] = dataset['Deck'].map(deck)



dataset['Deck'] = dataset['Deck'].fillna(0)




dataset['Deck'] = dataset['Deck'].astype(int)# we can now drop the cabin feature
train_df = train_df.drop(['Cabin'], axis=1)
test_df = test_df.drop(['Cabin'], axis=1)


data = [train_df, test_df]


for dataset in data:
   mean = train_df["Age"].mean()



std = test_df["Age"].std()



is_null = dataset["Age"].isnull().sum()
# compute random numbers between the mean, std and is_null
```

```
rand_age = np.random.randint(mean - std, mean + std, size =
is_null)
# fill NaN values in Age column with random values generated
```

```
age_slice = dataset["Age"].copy()
```

```
age_slice[np.isnan(age_slice)] = rand_age
```

```
dataset["Age"] =age_slice
```

```
dataset["Age"] =train_df["Age"].astype(int)
```

```
---------------------------------------------------------------------------
IntCastingNaNError                        Traceback (most recent call last)
<ipython-input-93-5d9b0307c3cb> in <module>
----> 1 dataset["Age"] =train_df["Age"].astype(int)
```

7 frames

```
/usr/local/lib/python3.7/dist-packages/pandas/core/dtypes/cast.py in
astype_float_to_int_nansafe(values, dtype, copy)
   1212     if not np.isfinite(values).all():
   1213         raise IntCastingNaNError(
-> 1214             "Cannot convert non-finite values (NA or inf) to integer"
   1215         )
   1216     return values.astype(dtype, copy=copy)

IntCastingNaNError: Cannot convert non-finite values (NA or inf) to integer
```

SEARCH STACK OVERFLOW

```
train_df["Age"].isnull().sum()
```

```
177
```

```
train_df['Embarked'].describe()
```

```
count     889
unique      3
top         S
freq      644
Name: Embarked, dtype: object
```

● ✕