```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
train = pd.read_csv("/content/train.csv")
train.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Far
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.250
1	2	1	1	Cumings, Mrs. John Bradley (Florence	female	38.0	1	0	PC 17599	71.283

Next steps: View recommended plots

```
test = pd.read_csv("/content/test.csv")
test.head()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Em
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	

Next steps: View recommended plots

```
train.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
```

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Name         418 non-null    object 
 3   Sex          418 non-null    object 
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64  
 6   Parch        418 non-null    int64
 7   Ticket       418 non-null    object
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object
 10  Embarked     418 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

```
all = pd.concat([train, test], sort = False)
all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1309 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  1309 non-null   int64
 1   Survived     891 non-null    float64
 2   Pclass       1309 non-null   int64
 3   Name         1309 non-null   object
 4   Sex          1309 non-null   object
 5   Age          1046 non-null   float64
 6   SibSp        1309 non-null   int64
 7   Parch        1309 non-null   int64
 8   Ticket       1309 non-null   object
 9   Fare         1308 non-null   float64
 10  Cabin        295 non-null    object
 11  Embarked     1307 non-null   object
dtypes: float64(3), int64(4), object(5)
memory usage: 132.9+ KB
```

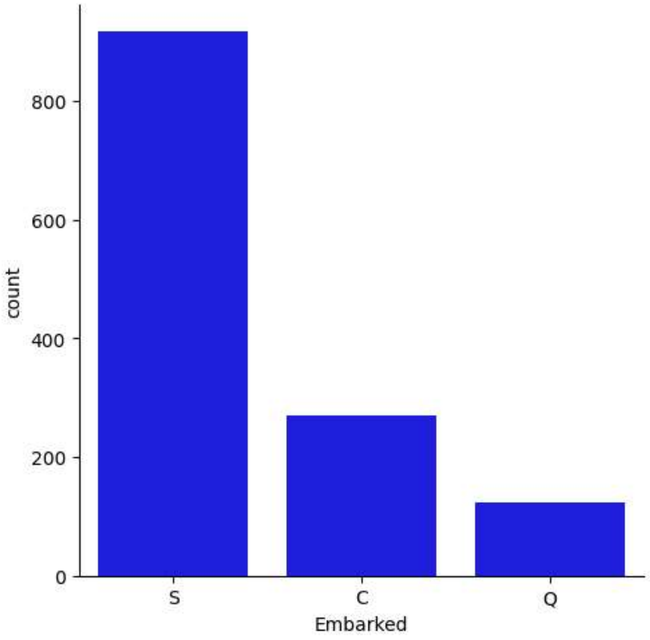
```
all['Age'] = all['Age'].fillna(value=all['Age'].median())
all['Fare'] = all['Fare'].fillna(value=all['Fare'].median())
```

```
all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1309 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  1309 non-null   int64
 1   Survived     891 non-null    float64
 2   Pclass       1309 non-null   int64
 3   Name         1309 non-null   object
 4   Sex          1309 non-null   object
 5   Age          1309 non-null   float64
 6   SibSp        1309 non-null   int64
 7   Parch        1309 non-null   int64
 8   Ticket       1309 non-null   object
 9   Fare         1309 non-null   float64
 10  Cabin        295 non-null    object
 11  Embarked     1307 non-null   object
dtypes: float64(3), int64(4), object(5)
memory usage: 132.9+ KB
```

```
sns.catplot(x = 'Embarked', kind = 'count', data = all, color='blue')
```

```
<seaborn.axisgrid.FacetGrid at 0x794686ea76a0>
```



```
all['Embarked'] = all['Embarked'].fillna('S')
all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1309 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  1309 non-null   int64  
 1   Survived     891 non-null    float64
 2   Pclass       1309 non-null   int64  
 3   Name         1309 non-null   object 
 4   Sex          1309 non-null   object 
 5   Age          1309 non-null   float64
 6   SibSp        1309 non-null   int64  
 7   Parch        1309 non-null   int64  
 8   Ticket       1309 non-null   object 
 9   Fare         1309 non-null   float64
 10  Cabin        295 non-null    object 
 11  Embarked     1309 non-null   object 
dtypes: float64(3), int64(4), object(5)
memory usage: 132.9+ KB
```

```
#Age
all.loc[ all['Age'] <= 16, 'Age'] = 0
all.loc[(all['Age'] > 16) & (all['Age'] <= 32), 'Age'] = 1
all.loc[(all['Age'] > 32) & (all['Age'] <= 48), 'Age'] = 2
all.loc[(all['Age'] > 48) & (all['Age'] <= 64), 'Age'] = 3
all.loc[ all['Age'] > 64, 'Age'] = 4
```

```
#Title
import re
def get_title(name):
    title_search = re.search(' ([A-Za-z]+\.)', name)

    if title_search:
        return title_search.group(1)
    return ""
```

```
all['Title'] = all['Name'].apply(get_title)
all['Title'].value_counts()
```

```
Mr.         757
Miss.       260
Mrs.        197
Master.      61
Rev.          8
Dr.           8
Col.          4
```

```
Mlle.          2
Major.         2
Ms.            2
Lady.          1
Sir.           1
Mme.           1
Don.           1
Capt.          1
Countess.      1
Jonkheer.      1
Dona.          1
Name: Title, dtype: int64
```

```
all['Title'] = all['Title'].replace(['Capt.', 'Dr.', 'Major.', 'Rev.'], 'Officer.')
all['Title'] = all['Title'].replace(['Lady.', 'Countess.', 'Don.', 'Sir.', 'Jonkheer.', 'Dona.'], 'Royal.')
all['Title'] = all['Title'].replace(['Mlle.', 'Ms.'], 'Miss.')
all['Title'] = all['Title'].replace(['Mme.'], 'Mrs.')
all['Title'].value_counts()
```

```
Mr.         757
Miss.       264
Mrs.        198
Master.      61
Officer.     19
Royal.        6
Col.          4
Name: Title, dtype: int64
```

```
#Cabin
all['Cabin'] = all['Cabin'].fillna('Missing')
all['Cabin'] = all['Cabin'].str[0]
all['Cabin'].value_counts()
```

```
M    1014
C      94
B      65
D      46
E      41
A      22
F      21
G       5
T       1
Name: Cabin, dtype: int64
```

```
#Family Size & Alone
all['Family_Size'] = all['SibSp'] + all['Parch'] + 1
all['IsAlone'] = 0
all.loc[all['Family_Size']==1, 'IsAlone'] = 1
all.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title	Family_Size	IsAlone
0	1	0.0	3	Braund, Mr. Owen Harris	male	1.0	1	0	A/5 21171	7.2500	M	S	Mr.	2	0
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	2.0	1	0	PC 17599	71.2833	C	C	Mrs.	2	0
2	3	1.0	3	Heikkinen, Miss. Laina	female	1.0	0	0	STON/O2. 3101282	7.9250	M	S	Miss.	1	1

Next steps: View recommended plots

```
#Drop unwanted variables
all_1 = all.drop(['Name', 'Ticket'], axis = 1)
all_1.head
```

```
pandas.core.generic.NDFrame.head
def head(n: int=5) -> NDFrameT
```

```
/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py
Return the first `n` rows.

This function returns the first `n` rows for the object based
on position. It is useful for quickly testing if your object
has the right type of data in it.
```

```
all_dummies = pd.get_dummies(all_1, drop_first = True)
all_dummies.head()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	Family_Size	IsAlone	Sex_male	...	Cabin_M	Cabin_T	Embarked_Q	Embarked_
0	1	0.0	3	1.0	1	0	7.2500	2	0	1	...	1	0	0	
1	2	1.0	1	2.0	1	0	71.2833	2	0	0	...	0	0	0	
2	3	1.0	3	1.0	0	0	7.9250	1	1	0	...	1	0	0	
3	4	1.0	1	2.0	1	0	53.1000	2	0	0	...	0	0	0	
4	5	0.0	3	2.0	0	0	8.0500	1	1	1	...	1	0	0	

5 rows × 26 columns

```
all_train = all_dummies[all_dummies['Survived'].notna()]
all_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 891 entries, 0 to 890
Data columns (total 26 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   PassengerId    891 non-null    int64  
 1   Survived       891 non-null    float64
 2   Pclass         891 non-null    int64  
 3   Age            891 non-null    float64
 4   SibSp          891 non-null    int64  
 5   Parch          891 non-null    int64  
 6   Fare           891 non-null    float64
 7   Family_Size    891 non-null    int64  
 8   IsAlone        891 non-null    int64  
 9   Sex_male       891 non-null    uint8  
 10  Cabin_B        891 non-null    uint8  
 11  Cabin_C        891 non-null    uint8  
 12  Cabin_D        891 non-null    uint8  
 13  Cabin_E        891 non-null    uint8  
 14  Cabin_F        891 non-null    uint8  
 15  Cabin_G        891 non-null    uint8  
 16  Cabin_M        891 non-null    uint8  
 17  Cabin_T        891 non-null    uint8  
 18  Embarked_Q     891 non-null    uint8  
 19  Embarked_S     891 non-null    uint8  
 20  Title_Master.  891 non-null    uint8  
 21  Title_Miss.    891 non-null    uint8  
 22  Title_Mr.      891 non-null    uint8  
 23  Title_Mrs.     891 non-null    uint8  
 24  Title_Officer. 891 non-null    uint8  
 25  Title_Royal.   891 non-null    uint8  
dtypes: float64(3), int64(6), uint8(17)
memory usage: 84.4 KB
```

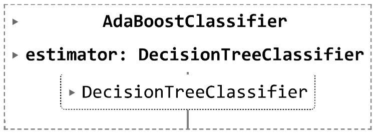
```
all_test = all_dummies[all_dummies['Survived'].isna()]
all_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 418 entries, 0 to 417
Data columns (total 26 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   PassengerId    418 non-null    int64  
 1   Survived       0 non-null      float64
 2   Pclass         418 non-null    int64  
 3   Age            418 non-null    float64
 4   SibSp          418 non-null    int64  
 5   Parch          418 non-null    int64  
 6   Fare           418 non-null    float64
 7   Family_Size     418 non-null    int64
 8   IsAlone         418 non-null    int64
 9   Sex_male        418 non-null    uint8
 10  Cabin_B         418 non-null    uint8
 11  Cabin_C         418 non-null    uint8
 12  Cabin_D         418 non-null    uint8
 13  Cabin_E         418 non-null    uint8
 14  Cabin_F         418 non-null    uint8
 15  Cabin_G         418 non-null    uint8
 16  Cabin_M         418 non-null    uint8
 17  Cabin_T         418 non-null    uint8
 18  Embarked_Q      418 non-null    uint8
 19  Embarked_S      418 non-null    uint8
 20  Title_Master.   418 non-null    uint8
 21  Title_Miss.     418 non-null    uint8
 22  Title_Mr.       418 non-null    uint8
 23  Title_Mrs.      418 non-null    uint8
 24  Title_Officer.  418 non-null    uint8
 25  Title_Royal.    418 non-null    uint8
dtypes: float64(3), int64(6), uint8(17)
memory usage: 39.6 KB
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(all_train.drop(['PassengerId','Survived'],axis=1),
                                                    all_train['Survived'], test_size=0.30,
                                                    random_state=101, stratify = all_train['Survived'])
```

```
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
```

```
ada = AdaBoostClassifier(DecisionTreeClassifier(),n_estimators=100, random_state=0)
ada.fit(X_train,y_train)
```



▸ **AdaBoostClassifier**

▸ **estimator: DecisionTreeClassifier**

▸ DecisionTreeClassifier

```
predictions = ada.predict(X_test)
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

         0.0       0.77      0.87      0.82       165
         1.0       0.74      0.58      0.65       103

    accuracy                           0.76       268
   macro avg       0.76      0.73      0.74       268
weighted avg       0.76      0.76      0.75       268
```

```
print (f'Train Accuracy - : {ada.score(X_train,y_train):.3f}')
print (f'Test Accuracy - : {ada.score(X_test,y_test):.3f}')
```

```
Train Accuracy - : 0.961
Test Accuracy - : 0.761
```

```
TestForPred = all_test.drop(['PassengerId', 'Survived'], axis = 1)
```

```
t_pred = ada.predict(TestForPred).astype(int)
```

```
PassengerId = all_test['PassengerId']
```

```
adaSub = pd.DataFrame({'PassengerId': PassengerId, 'Survived':t_pred })
adaSub.head()
```

	PassengerId	Survived
0	892	0
1	893	1
2	894	1