In [1]:

```
##Titanic Survival Predictions
```

In [2]:

```
###Import Necessary Libraries
```

In [3]:

```
#data analysis libraries
import numpy as np
import pandas as pd

#visualization libraries
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

#ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

```
G:\Anaconda\lib\site-packages\pandas\compat\_optional.py:138: UserWarning:
Pandas requires version '2.7.0' or newer of 'numexpr' (version '2.6.9' cur
rently installed).
  warnings.warn(msg, UserWarning)
```

In [4]:

```
###Read in and Explore the Data
```

In [8]:

```
#import train and test CSV files
train = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")

#take a look at the training data
train.describe(include="all")
```

Out[8]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Pa
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000
unique	NaN	NaN	NaN	891	2	NaN	NaN	
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	NaN	NaN	
freq	NaN	NaN	NaN	1	577	NaN	NaN	
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000
max	891.000000	1.000000	3.000000	NaN	NaN	80.000000	8.000000	6.000

In [9]:

```
###Data Analysis
```

In [10]:

```
#get a list of the features within the dataset
print(train.columns)
```

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

In [11]:

```
#see a sample of the dataset to get an idea of the variables
train.sample(5)
```

Out[11]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
814	815	0	3	Tomlin, Mr. Ernest Portage	male	30.5	0	0	364499	8.0500
178	179	0	2	Hale, Mr. Reginald	male	30.0	0	0	250653	13.0000
638	639	0	3	Panula, Mrs. Juha (Maria Emilia Ojala)	female	41.0	0	5	3101295	39.6875
240	241	0	3	Zabour, Miss. Thamine	female	NaN	1	0	2665	14.4542
151	152	1	1	Pears, Mrs. Thomas (Edith Wearne)	female	22.0	1	0	113776	66.6000

In [12]:

```
#see a summary of the training dataset
train.describe(include = "all")
```

Out[12]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Pa
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000
unique	NaN	NaN	NaN	891	2	NaN	NaN	
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	NaN	NaN	
freq	NaN	NaN	NaN	1	577	NaN	NaN	
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000
max	891.000000	1.000000	3.000000	NaN	NaN	80.000000	8.000000	6.000

In [13]:

```
##Some Observations:
#There are a total of 891 passengers in our training set.
#The Age feature is missing approximately 19.8% of its values. I'm guessing that the Ag
e feature is pretty important to survival, so we should probably attempt to fill these
 gaps.
#The Cabin feature is missing approximately 77.1% of its values. Since so much of the f
eature is missing, it would be hard to fill in the missing values. We'll probably drop
 these values from our dataset.
#The Embarked feature is missing 0.22% of its values, which should be relatively harmle
ss.
```

In [14]:

```
#check for any other unusable values
print(pd.isnull(train).sum())
```

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

In [15]:

```
# Some Predictions:
#Sex: Females are more likely to survive.
#SibSp/Parch: People traveling alone are more likely to survive.
#Age: Young children are more likely to survive.
```

In [16]:

```
# Data Visualization
# It's time to visualize our data so we can see whether our predictions were accurate!
```

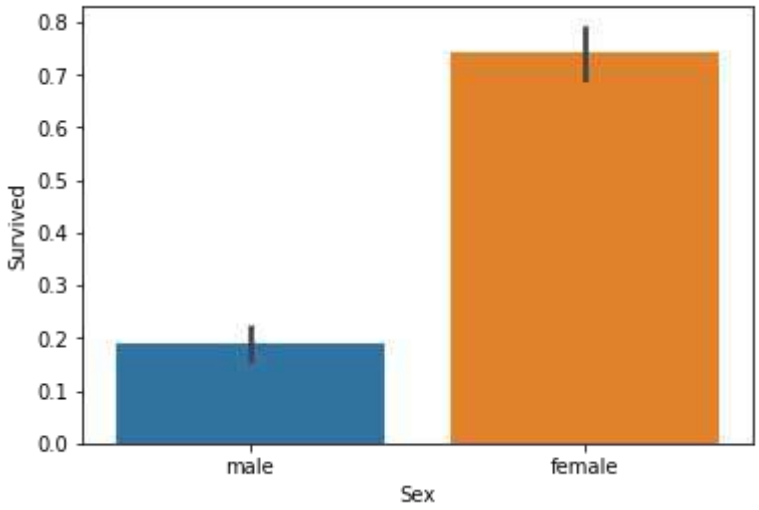
In [17]:

```
#SEX Feature
#draw a bar plot of survival by sex
sns.barplot(x="Sex", y="Survived", data=train)

#print percentages of females vs. males that survive
print("Percentage of females who survived:", train["Survived"][train["Sex"] == 'female'
].value_counts(normalize = True)[1]*100)

print("Percentage of males who survived:", train["Survived"][train["Sex"] == 'male'].va
lue_counts(normalize = True)[1]*100)
```

```
Percentage of females who survived: 74.20382165605095
Percentage of males who survived: 18.890814558058924
```



In [18]:

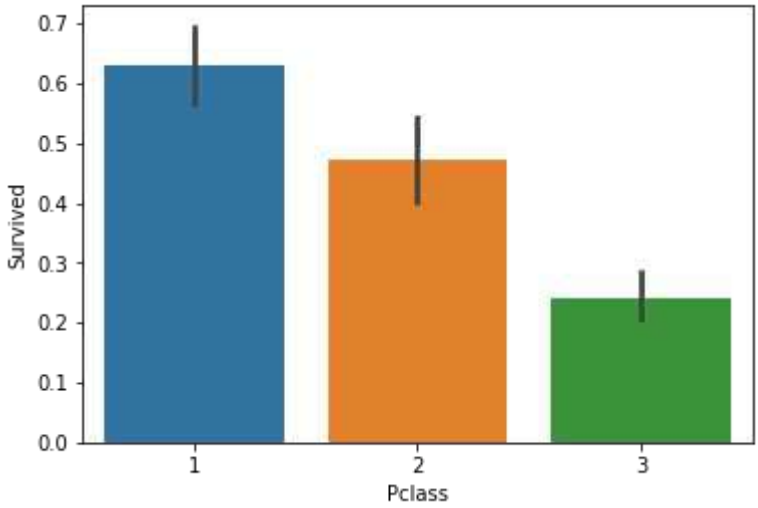
```
#PClass
#draw a bar plot of survival by Pclass
sns.barplot(x="Pclass", y="Survived", data=train)

#print percentage of people by Pclass that survived
print("Percentage of Pclass = 1 who survived:", train["Survived"][train["Pclass"] == 1]
.value_counts(normalize = True)[1]*100)

print("Percentage of Pclass = 2 who survived:", train["Survived"][train["Pclass"] == 2]
.value_counts(normalize = True)[1]*100)

print("Percentage of Pclass = 3 who survived:", train["Survived"][train["Pclass"] == 3]
.value_counts(normalize = True)[1]*100)
```

```
Percentage of Pclass = 1 who survived: 62.96296296296296
Percentage of Pclass = 2 who survived: 47.28260869565217
Percentage of Pclass = 3 who survived: 24.236252545824847
```



In [19]:

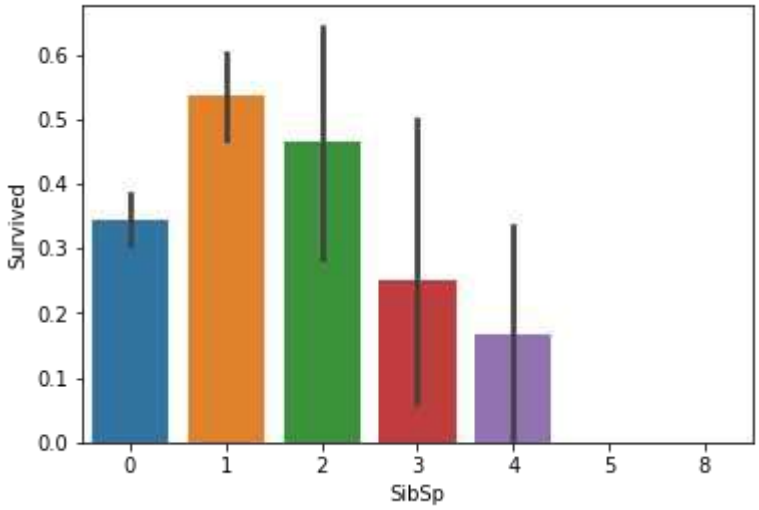
```
#SibSp Feature
#draw a bar plot for SibSp vs. survival
sns.barplot(x="SibSp", y="Survived", data=train)

#I won't be printing individual percent values for all of these.
print("Percentage of SibSp = 0 who survived:", train["Survived"][train["SibSp"] == 0].v
alue_counts(normalize = True)[1]*100)

print("Percentage of SibSp = 1 who survived:", train["Survived"][train["SibSp"] == 1].v
alue_counts(normalize = True)[1]*100)

print("Percentage of SibSp = 2 who survived:", train["Survived"][train["SibSp"] == 2].v
alue_counts(normalize = True)[1]*100)
```

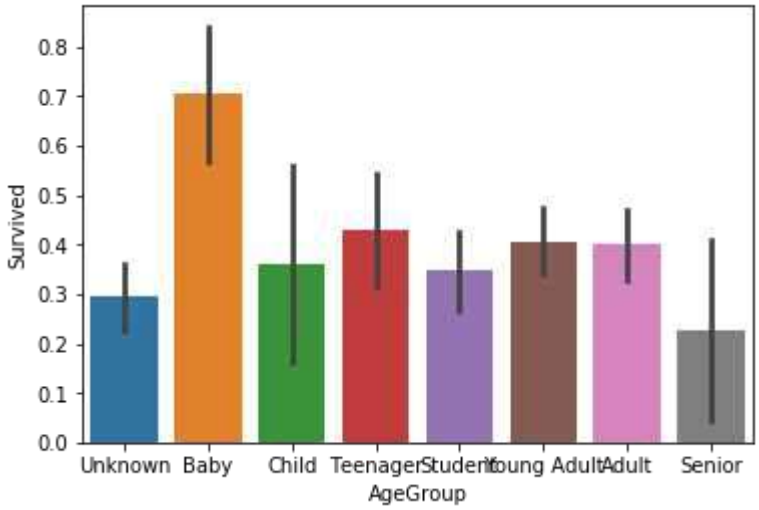
```
Percentage of SibSp = 0 who survived: 34.53947368421053
Percentage of SibSp = 1 who survived: 53.588516746411486
Percentage of SibSp = 2 who survived: 46.42857142857143
```



In [21]:

```
#Age Feature
#sort the ages into logical categories
train["Age"] = train["Age"].fillna(-0.5)
test["Age"] = test["Age"].fillna(-0.5)
bins = [-1, 0, 5, 12, 18, 24, 35, 60, np.inf]
labels = ['Unknown', 'Baby', 'Child', 'Teenager', 'Student', 'Young Adult', 'Adult', 'S
enior']
train['AgeGroup'] = pd.cut(train["Age"], bins, labels = labels)
test['AgeGroup'] = pd.cut(test["Age"], bins, labels = labels)

#draw a bar plot of Age vs. survival
sns.barplot(x="AgeGroup", y="Survived", data=train)
plt.show()
```



In [22]:

```
#Babies are more likely to survive than any other age group.
#People with a recorded Cabin number are, in fact, more likely to survive. (66.6% vs 2
9.9%)
```

In [23]:

```
###Cleaning Data
```

In [24]:

```
# Looking at the Test Data
```

In [25]:

```
test.describe(include="all")
```

Out[25]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
count	418.000000	418.000000	418	418	418.000000	418.000000	418.000000	418
unique	NaN	NaN	418	2	NaN	NaN	NaN	363
top	NaN	NaN	Kelly, Mr. James	male	NaN	NaN	NaN	PC 17608
freq	NaN	NaN	1	266	NaN	NaN	NaN	5
mean	1100.500000	2.265550	NaN	NaN	23.941388	0.447368	0.392344	NaN
std	120.810458	0.841838	NaN	NaN	17.741080	0.896760	0.981429	NaN
min	892.000000	1.000000	NaN	NaN	-0.500000	0.000000	0.000000	NaN
25%	996.250000	1.000000	NaN	NaN	9.000000	0.000000	0.000000	NaN
50%	1100.500000	3.000000	NaN	NaN	24.000000	0.000000	0.000000	NaN
75%	1204.750000	3.000000	NaN	NaN	35.750000	1.000000	0.000000	NaN
max	1309.000000	3.000000	NaN	NaN	76.000000	8.000000	9.000000	NaN

In [26]:

```
#We have a total of 418 passengers.
#1 value from the Fare feature is missing.
#Around 20.5% of the Age feature is missing, we will need to fill that in.
```

In [27]:

```
#Cabin Feature
#we'll start off by dropping the Cabin feature since not a lot more useful information
 can be extracted from it.
train = train.drop(['Cabin'], axis = 1)
test = test.drop(['Cabin'], axis = 1)
```

In [28]:

```
#Tcket Feature
#we can also drop the Ticket feature since it's unlikely to yield any useful informatio
n
train = train.drop(['Ticket'], axis = 1)
test = test.drop(['Ticket'], axis = 1)
```

In [29]:

```
#Embarked Feature
#now we need to fill in the missing values in the Embarked feature
print("Number of people embarking in Southampton (S):")
southampton = train[train["Embarked"] == "S"].shape[0]
print(southampton)

print("Number of people embarking in Cherbourg (C):")
cherbourg = train[train["Embarked"] == "C"].shape[0]
print(cherbourg)

print("Number of people embarking in Queenstown (Q):")
queenstown = train[train["Embarked"] == "Q"].shape[0]
print(queenstown)
```

```
Number of people embarking in Southampton (S):
644
Number of people embarking in Cherbourg (C):
168
Number of people embarking in Queenstown (Q):
77
```

In [30]:

```
#replacing the missing values in the Embarked feature with S
train = train.fillna({"Embarked": "S"})
```

In [31]:

```
#Age Feature
#Next we'll fill in the missing values in the Age feature. Since a higher percentage of values are missing, it would be illogical to fill all of them with the same value (as we did with Embarked). Instead, let's try to find a way to predict the missing ages.
```

In [32]:

```
#create a combined group of both datasets
combine = [train, test]

#extract a title for each Name in the train and test datasets
for dataset in combine:
    dataset['Title'] = dataset.Name.str.extract(' ([A-Za-z]+)\.', expand=False)

pd.crosstab(train['Title'], train['Sex'])
```

Out[32]:

Sex Title	female	male
Capt	0	1
Col	0	2
Countess	1	0
Don	0	1
Dr	1	6
Jonkheer	0	1
Lady	1	0
Major	0	2
Master	0	40
Miss	182	0
Mlle	2	0
Mme	1	0
Mr	0	517
Mrs	125	0
Ms	1	0
Rev	0	6
Sir	0	1

In [33]:

```
#replace various titles with more common names
for dataset in combine:
    dataset['Title'] = dataset['Title'].replace(['Lady', 'Capt', 'Col',
    'Don', 'Dr', 'Major', 'Rev', 'Jonkheer', 'Dona'], 'Rare')

    dataset['Title'] = dataset['Title'].replace(['Countess', 'Lady', 'Sir'], 'Royal')
    dataset['Title'] = dataset['Title'].replace('Mlle', 'Miss')
    dataset['Title'] = dataset['Title'].replace('Ms', 'Miss')
    dataset['Title'] = dataset['Title'].replace('Mme', 'Mrs')

train[['Title', 'Survived']].groupby(['Title'], as_index=False).mean()
```

Out[33]:

	Title	Survived
0	Master	0.575000
1	Miss	0.702703
2	Mr	0.156673
3	Mrs	0.793651
4	Rare	0.285714
5	Royal	1.000000

In [34]:

```
#map each of the title groups to a numerical value
title_mapping = {"Mr": 1, "Miss": 2, "Mrs": 3, "Master": 4, "Royal": 5, "Rare": 6}
for dataset in combine:
    dataset['Title'] = dataset['Title'].map(title_mapping)
    dataset['Title'] = dataset['Title'].fillna(0)

train.head()
```

Out[34]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S

In [35]:

```
# fill missing age with mode age group for each title
mr_age = train[train["Title"] == 1]["AgeGroup"].mode() #Young Adult
miss_age = train[train["Title"] == 2]["AgeGroup"].mode() #Student
mrs_age = train[train["Title"] == 3]["AgeGroup"].mode() #Adult
master_age = train[train["Title"] == 4]["AgeGroup"].mode() #Baby
royal_age = train[train["Title"] == 5]["AgeGroup"].mode() #Adult
rare_age = train[train["Title"] == 6]["AgeGroup"].mode() #Adult

age_title_mapping = {1: "Young Adult", 2: "Student", 3: "Adult", 4: "Baby", 5: "Adult",
6: "Adult"}

#I tried to get this code to work with using .map(), but couldn't.
#I've put down a less elegant, temporary solution for now.
#train = train.fillna({"Age": train["Title"].map(age_title_mapping)})
#test = test.fillna({"Age": test["Title"].map(age_title_mapping)})

for x in range(len(train["AgeGroup"])):
    if train["AgeGroup"][x] == "Unknown":
        train["AgeGroup"][x] = age_title_mapping[train["Title"][x]]

for x in range(len(test["AgeGroup"])):
    if test["AgeGroup"][x] == "Unknown":
        test["AgeGroup"][x] = age_title_mapping[test["Title"][x]]
```

In [36]:

```
#map each Age value to a numerical value
age_mapping = {'Baby': 1, 'Child': 2, 'Teenager': 3, 'Student': 4, 'Young Adult': 5, 'A
dult': 6, 'Senior': 7}
train['AgeGroup'] = train['AgeGroup'].map(age_mapping)
test['AgeGroup'] = test['AgeGroup'].map(age_mapping)

train.head()

#dropping the Age feature for now, might change
train = train.drop(['Age'], axis = 1)
test = test.drop(['Age'], axis = 1)
```

In [37]:

```
#Name Feature
#drop the name feature since it contains no more useful information.
train = train.drop(['Name'], axis = 1)
test = test.drop(['Name'], axis = 1)
```

In [38]:

```
#Sex Feature
#map each Sex value to a numerical value
sex_mapping = {"male": 0, "female": 1}
train['Sex'] = train['Sex'].map(sex_mapping)
test['Sex'] = test['Sex'].map(sex_mapping)

train.head()
```

Out[38]:

	PassengerId	Survived	Pclass	Sex	SibSp	Parch	Fare	Embarked	AgeGroup	Title
0	1	0	3	0	1	0	7.2500	S	4.0	1
1	2	1	1	1	1	0	71.2833	C	6.0	3
2	3	1	3	1	0	0	7.9250	S	5.0	2
3	4	1	1	1	1	0	53.1000	S	5.0	3
4	5	0	3	0	0	0	8.0500	S	5.0	1

In [39]:

```
#Embarked Feature
#map each Embarked value to a numerical value
embarked_mapping = {"S": 1, "C": 2, "Q": 3}
train['Embarked'] = train['Embarked'].map(embarked_mapping)
test['Embarked'] = test['Embarked'].map(embarked_mapping)

train.head()
```

Out[39]:

	PassengerId	Survived	Pclass	Sex	SibSp	Parch	Fare	Embarked	AgeGroup	Title
0	1	0	3	0	1	0	7.2500	1	4.0	1
1	2	1	1	1	1	0	71.2833	2	6.0	3
2	3	1	3	1	0	0	7.9250	1	5.0	2
3	4	1	1	1	1	0	53.1000	1	5.0	3
4	5	0	3	0	0	0	8.0500	1	5.0	1

In [40]:

```
#Fare Feature
#fill in missing Fare value in test set based on mean fare for that Pclass
for x in range(len(test["Fare"])):
    if pd.isnull(test["Fare"][x]):
        pclass = test["Pclass"][x] #Pclass = 3
        test["Fare"][x] = round(train[train["Pclass"] == pclass]["Fare"].mean(), 4)

#map Fare values into groups of numerical values
train['FareBand'] = pd.qcut(train['Fare'], 4, labels = [1, 2, 3, 4])
test['FareBand'] = pd.qcut(test['Fare'], 4, labels = [1, 2, 3, 4])

#drop Fare values
train = train.drop(['Fare'], axis = 1)
test = test.drop(['Fare'], axis = 1)
```

In [41]:

```
#check train data
train.head()
```

Out[41]:

	PassengerId	Survived	Pclass	Sex	SibSp	Parch	Embarked	AgeGroup	Title	FareBanc
0	1	0	3	0	1	0	1	4.0	1	1
1	2	1	1	1	1	0	2	6.0	3	4
2	3	1	3	1	0	0	1	5.0	2	2
3	4	1	1	1	1	0	1	5.0	3	4
4	5	0	3	0	0	0	1	5.0	1	2

In [42]:

```
#check test data
test.head()
```

Out[42]:

	PassengerId	Pclass	Sex	SibSp	Parch	Embarked	AgeGroup	Title	FareBand
0	892	3	0	0	0	3	5.0	1	1
1	893	3	1	1	0	1	6.0	3	1
2	894	2	0	0	0	3	7.0	1	2
3	895	3	0	0	0	1	5.0	1	2
4	896	3	1	1	1	1	4.0	3	2

In [43]:

```
# Splitting the Training Data
```

In [44]:

```
from sklearn.model_selection import train_test_split

predictors = train.drop(['Survived', 'PassengerId'], axis=1)
target = train["Survived"]
x_train, x_val, y_train, y_val = train_test_split(predictors, target, test_size = 0.22,
random_state = 0)
```

In [45]:

```
###Choosing the Best Model
```

In [46]:

```
# Logistic Regression
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score


logreg = LogisticRegression()
logreg.fit(x_train, y_train)
y_pred = logreg.predict(x_val)
acc_logreg = round(accuracy_score(y_pred, y_val) * 100, 2)
print(acc_logreg)
```

```
G:\Anaconda\lib\site-packages\sklearn\linear_model\least_angle.py:35: Depr
ecationWarning: `np.float` is a deprecated alias for the builtin `float`.
To silence this warning, use `float` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar typ
e, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  eps=np.finfo(np.float).eps,
G:\Anaconda\lib\site-packages\sklearn\linear_model\least_angle.py:597: Dep
recationWarning: `np.float` is a deprecated alias for the builtin `float`.
To silence this warning, use `float` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar typ
e, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  eps=np.finfo(np.float).eps, copy_X=True, fit_path=True,
G:\Anaconda\lib\site-packages\sklearn\linear_model\least_angle.py:836: Dep
recationWarning: `np.float` is a deprecated alias for the builtin `float`.
To silence this warning, use `float` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar typ
e, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  eps=np.finfo(np.float).eps, copy_X=True, fit_path=True,
G:\Anaconda\lib\site-packages\sklearn\linear_model\least_angle.py:862: Dep
recationWarning: `np.float` is a deprecated alias for the builtin `float`.
To silence this warning, use `float` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar typ
e, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  eps=np.finfo(np.float).eps, positive=False):
G:\Anaconda\lib\site-packages\sklearn\linear_model\least_angle.py:1097: De
precationWarning: `np.float` is a deprecated alias for the builtin `float
`. To silence this warning, use `float` by itself. Doing this will not mod
ify any behavior and is safe. If you specifically wanted the numpy scalar
type, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  max_n_alphas=1000, n_jobs=None, eps=np.finfo(np.float).eps,
G:\Anaconda\lib\site-packages\sklearn\linear_model\least_angle.py:1344: De
precationWarning: `np.float` is a deprecated alias for the builtin `float
`. To silence this warning, use `float` by itself. Doing this will not mod
ify any behavior and is safe. If you specifically wanted the numpy scalar
type, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  max_n_alphas=1000, n_jobs=None, eps=np.finfo(np.float).eps,
G:\Anaconda\lib\site-packages\sklearn\linear_model\least_angle.py:1480: De
precationWarning: `np.float` is a deprecated alias for the builtin `float
`. To silence this warning, use `float` by itself. Doing this will not mod
ify any behavior and is safe. If you specifically wanted the numpy scalar
type, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  eps=np.finfo(np.float).eps, copy_X=True, positive=False):
G:\Anaconda\lib\site-packages\sklearn\linear_model\randomized_l1.py:152: D
eprecationWarning: `np.float` is a deprecated alias for the builtin `float
`. To silence this warning, use `float` by itself. Doing this will not mod
ify any behavior and is safe. If you specifically wanted the numpy scalar
type, use `np.float64` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  precompute=False, eps=np.finfo(np.float).eps,
G:\Anaconda\lib\site-packages\sklearn\linear_model\randomized_l1.py:320: D
eprecationWarning: `np.float` is a deprecated alias for the builtin `float
`. To silence this warning, use `float` by itself. Doing this will not mod
ify any behavior and is safe. If you specifically wanted the numpy scalar
type, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  eps=np.finfo(np.float).eps, random_state=None,
G:\Anaconda\lib\site-packages\sklearn\linear_model\randomized_l1.py:580: D
eprecationWarning: `np.float` is a deprecated alias for the builtin `float
`. To silence this warning, use `float` by itself. Doing this will not mod
ify any behavior and is safe. If you specifically wanted the numpy scalar
type, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  eps=4 * np.finfo(np.float).eps, n_jobs=None,
```

```
79.7
```

```
G:\Anaconda\lib\site-packages\sklearn\linear_model\base.py:283: Deprecatio
nWarning: `np.int` is a deprecated alias for the builtin `int`. To silence
this warning, use `int` by itself. Doing this will not modify any behavior
and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64`
or `np.int32` to specify the precision. If you wish to review your current
use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  indices = (scores > 0).astype(np.int)
```

In [47]:

```
# Random Forest
from sklearn.ensemble import RandomForestClassifier

randomforest = RandomForestClassifier()
randomforest.fit(x_train, y_train)
y_pred = randomforest.predict(x_val)
acc_randomforest = round(accuracy_score(y_pred, y_val) * 100, 2)
print(acc_randomforest)
```

```
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:34: De
precationWarning: `np.bool` is a deprecated alias for the builtin `bool`.
To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar typ
e, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  from ._gradient_boosting import predict_stages
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:34: De
precationWarning: `np.bool` is a deprecated alias for the builtin `bool`.
To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar typ
e, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  from ._gradient_boosting import predict_stages
G:\Anaconda\lib\site-packages\sklearn\ensemble\forest.py:487: DeprecationW
arning: `np.int` is a deprecated alias for the builtin `int`. To silence t
his warning, use `int` by itself. Doing this will not modify any behavior
and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64`
or `np.int32` to specify the precision. If you wish to review your current
use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  y_store_unique_indices = np.zeros(y.shape, dtype=np.int)
G:\Anaconda\lib\site-packages\sklearn\tree\tree.py:149: DeprecationWarnin
g: `np.int` is a deprecated alias for the builtin `int`. To silence this w
arning, use `int` by itself. Doing this will not modify any behavior and i
s safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `n
p.int32` to specify the precision. If you wish to review your current use,
check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  y_encoded = np.zeros(y.shape, dtype=np.int)
G:\Anaconda\lib\site-packages\sklearn\tree\tree.py:149: DeprecationWarnin
g: `np.int` is a deprecated alias for the builtin `int`. To silence this w
arning, use `int` by itself. Doing this will not modify any behavior and i
s safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `n
p.int32` to specify the precision. If you wish to review your current use,
check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  y_encoded = np.zeros(y.shape, dtype=np.int)
G:\Anaconda\lib\site-packages\sklearn\tree\tree.py:149: DeprecationWarnin
g: `np.int` is a deprecated alias for the builtin `int`. To silence this w
arning, use `int` by itself. Doing this will not modify any behavior and i
s safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `n
p.int32` to specify the precision. If you wish to review your current use,
check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  y_encoded = np.zeros(y.shape, dtype=np.int)
G:\Anaconda\lib\site-packages\sklearn\tree\tree.py:149: DeprecationWarnin
g: `np.int` is a deprecated alias for the builtin `int`. To silence this w
arning, use `int` by itself. Doing this will not modify any behavior and i
s safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `n
p.int32` to specify the precision. If you wish to review your current use,
check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  y_encoded = np.zeros(y.shape, dtype=np.int)
```

```
G:\Anaconda\lib\site-packages\sklearn\tree\tree.py:149: DeprecationWarnin
g: `np.int` is a deprecated alias for the builtin `int`. To silence this w
arning, use `int` by itself. Doing this will not modify any behavior and i
s safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `n
p.int32` to specify the precision. If you wish to review your current use,
check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  y_encoded = np.zeros(y.shape, dtype=np.int)
G:\Anaconda\lib\site-packages\sklearn\tree\tree.py:149: DeprecationWarnin
g: `np.int` is a deprecated alias for the builtin `int`. To silence this w
arning, use `int` by itself. Doing this will not modify any behavior and i
s safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `n
p.int32` to specify the precision. If you wish to review your current use,
check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  y_encoded = np.zeros(y.shape, dtype=np.int)
G:\Anaconda\lib\site-packages\sklearn\tree\tree.py:149: DeprecationWarnin
g: `np.int` is a deprecated alias for the builtin `int`. To silence this w
arning, use `int` by itself. Doing this will not modify any behavior and i
s safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `n
p.int32` to specify the precision. If you wish to review your current use,
check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  y_encoded = np.zeros(y.shape, dtype=np.int)
G:\Anaconda\lib\site-packages\sklearn\tree\tree.py:149: DeprecationWarnin
g: `np.int` is a deprecated alias for the builtin `int`. To silence this w
arning, use `int` by itself. Doing this will not modify any behavior and i
s safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `n
p.int32` to specify the precision. If you wish to review your current use,
check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  y_encoded = np.zeros(y.shape, dtype=np.int)
G:\Anaconda\lib\site-packages\sklearn\tree\tree.py:149: DeprecationWarnin
g: `np.int` is a deprecated alias for the builtin `int`. To silence this w
arning, use `int` by itself. Doing this will not modify any behavior and i
s safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `n
p.int32` to specify the precision. If you wish to review your current use,
check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  y_encoded = np.zeros(y.shape, dtype=np.int)
G:\Anaconda\lib\site-packages\sklearn\tree\tree.py:149: DeprecationWarnin
g: `np.int` is a deprecated alias for the builtin `int`. To silence this w
arning, use `int` by itself. Doing this will not modify any behavior and i
s safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `n
p.int32` to specify the precision. If you wish to review your current use,
check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  y_encoded = np.zeros(y.shape, dtype=np.int)
```

```
84.77
```

```
G:\Anaconda\lib\site-packages\sklearn\ensemble\base.py:158: DeprecationWar
ning: `np.int` is a deprecated alias for the builtin `int`. To silence thi
s warning, use `int` by itself. Doing this will not modify any behavior an
d is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or
`np.int32` to specify the precision. If you wish to review your current us
e, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
```

In [48]:

```
# Gradient Boosting Classifier
from sklearn.ensemble import GradientBoostingClassifier

gbk = GradientBoostingClassifier()
gbk.fit(x_train, y_train)
y_pred = gbk.predict(x_val)
acc_gbk = round(accuracy_score(y_pred, y_val) * 100, 2)
print(acc_gbk)
```

```
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1302:
DeprecationWarning: `np.object` is a deprecated alias for the builtin `obj
ect`. To silence this warning, use `object` by itself. Doing this will not
modify any behavior and is safe.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  dtype=np.object)
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1489:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  sample_mask = np.ones((n_samples, ), dtype=np.bool)
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
```

```
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
```

```
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
```

```
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
```

```
83.25
```

```
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
```

```
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
```

```
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
```

```
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
```

```
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
```

```
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
```

```
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
```

```
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
G:\Anaconda\lib\site-packages\sklearn\ensemble\gradient_boosting.py:1162:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modi
fy any behavior and is safe. If you specifically wanted the numpy scalar t
ype, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations
  assert sample_mask.dtype == np.bool
```

In [49]:

```
models = pd.DataFrame({
    'Model': ['Logistic Regression',
              'Random Forest','Gradient Boosting Classifier'],
    'Score': [acc_logreg,
              acc_randomforest,acc_gbk]})
models.sort_values(by='Score', ascending=False)
```

Out[49]:

	Model	Score
1	Random Forest	84.77
2	Gradient Boosting Classifier	83.25
0	Logistic Regression	79.70

In [50]:

```
#Creating Submission File
```

In [52]:

```
#set ids as PassengerId and predict survival
ids = test['PassengerId']
predictions = gbk.predict(test.drop('PassengerId', axis=1))

#set the output as a dataframe and convert to csv file named submission.csv
output = pd.DataFrame({ 'PassengerId' : ids, 'Survived': predictions })
output.to_csv('submission.csv', index=False)
```

In [ ]:

```

```