In [56]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import GradientBoostingClassifier
```

In [22]:

```
df = pd.read_csv('training_titanic_x_y_train.csv', usecols=['Pclass', 'Sex', 'Age',
print(df.isnull().sum())
df1 = pd.read_csv('test_titanic_x_test.csv', usecols=['Pclass', 'Sex', 'Age', 'Fare
print(df1.isnull().sum())
```

```
Pclass        0
Sex           0
Age         132
Fare          0
Survived      0
dtype: int64
Pclass     0
Sex        0
Age       45
Fare       0
dtype: int64
```

In [24]:

```
df.Age.fillna(df.Age.mean(), inplace=True)
df1.Age.fillna(df1.Age.mean(), inplace=True)
```

In [26]:

```
print(df.isnull().sum())
print(df1.isnull().sum())
```

```
Pclass      0
Sex         0
Age         0
Fare        0
Survived    0
dtype: int64
Pclass    0
Sex       0
Age       0
Fare      0
dtype: int64
```

In [28]:

```
df.Sex.replace(['female', 'male'], [0, 1], inplace=True)
df1.Sex.replace(['female', 'male'], [0, 1], inplace=True)
```

In [31]:

```
df
```

Out[31]:

	Pclass	Sex	Age	Fare	Survived
0	2	0	29.00000	26.0000	1
1	3	1	29.70056	8.0500	0
2	2	1	39.00000	26.0000	0
3	3	0	29.00000	21.0750	0
4	3	1	25.00000	7.0500	0
...	...	...	...	...	...
663	2	0	17.00000	10.5000	1
664	3	1	29.70056	7.7500	0
665	3	1	32.00000	56.4958	1
666	3	0	22.00000	9.8375	0
667	3	0	29.70056	15.5000	1

668 rows × 5 columns

In [34]:

```
X_train = df.iloc[:, :4]
Y_train = df.iloc[:, 4]
X_test = df1.iloc[:, :4]
```

In [35]:

```
X_train.shape, Y_train.shape, X_test.shape
```

Out[35]:

```
((668, 4), (668,), (223, 4))
```

In [47]:

```
scaler = StandardScaler()
scaler.fit(X_train)
X_train_scaler = scaler.transform(X_train)
X_test_scaler = scaler.transform(X_test)
```

In [52]:

```
model = GradientBoostingClassifier(random_state=0, learning_rate=0.01, max_depth=9,
alg = LogisticRegression(random_state=0, solver='liblinear')
alg.fit(X_train, Y_train)
Y_pred = alg.predict(X_test)
model.fit(X_train, Y_train)
Y_pred = alg.predict(X_test)
Y_pred_gradient = model.predict(X_test)
```

In [53]:

```
np.savetxt('LogisticPredictions.csv', Y_pred)
np.savetxt('LogisticPredictionsGradient.csv', Y_pred_gradient)
```

In [54]:

```
alg.score(X_train, Y_train)
```

Out[54]:

```
0.7784431137724551
```

In [55]:

```
model.score(X_train, Y_train)
```

Out[55]:

```
0.9431137724550899
```