In [12]:
```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

In [13]:
```
x=pd.read_csv('train.csv')
x.head()
```

Out[13]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

In [14]:
```
y=x.pop('Survived')
y.head()
```

Out[14]:
```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

In [17]:
```
numeric_variables=list(x.dtypes[x.dtypes!="object"].index)
x[numeric_variables].head()
```

Out[17]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare
0	1	3	22.0	1	0	7.2500
1	2	1	38.0	1	0	71.2833
2	3	3	26.0	0	0	7.9250
3	4	1	35.0	1	0	53.1000
4	5	3	35.0	0	0	8.0500

In [18]:
```
x["Age"].fillna(x.Age.mean(),inplace=True)
```

In [19]:
```
x.tail()
```

Out[19]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
886	887	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.00	NaN
887	888	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.00	B42
888	889	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.45	NaN
889	890	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.00	C148
890	891	3	Dooley, Mr. Patrick	male	32.000000	0	0	370376	7.75	NaN

In [20]:
```
x[numeric_variables].head()
```

Out[20]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare
0	1	3	22.0	1	0	7.2500
1	2	1	38.0	1	0	71.2833
2	3	3	26.0	0	0	7.9250
3	4	1	35.0	1	0	53.1000
4	5	3	35.0	0	0	8.0500

In [23]:
```
model=RandomForestClassifier(n_estimators=100)
model.fit(x[numeric_variables],y)
```

Out[23]:
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
```

In [29]:
```
print("Train Accuracy :",accuracy_score(y,model.predict(x[numeric_variables])))
```

```
Train Accuracy : 1.0
```

In [30]:
```
test=pd.read_csv("test.csv")
```

In [31]:
```
test[numeric_variables].head()
```

Out[31]:

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

In [32]:
```
test["Age"].fillna(test.Age.mean(),inplace=True)
```

In [33]:
```
test=test[numeric_variables].fillna(test.mean()).copy()
```

In [36]:
```
y_pred=model.predict(test[numeric_variables])
y_pred
```

Out[36]:
```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1,
       1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1,
       1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1,
       0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0,
       1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0,
       1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1,
       0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0,
       0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0,
       1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0,
       1, 0, 0, 0], dtype=int64)
```

In [35]:
```
submission=pd.DataFrame({
    "PassengerId": test["PassengerId"],
    "Survived":y_pred
})
submission.to_csv('titanic.csv',index=False)
```

In [39]:
```
submission
```

404	1296	1
405	1297	0
406	1298	0
407	1299	1
408	1300	0
409	1301	1
410	1302	0
411	1303	1
412	1304	0
413	1305	0
414	1306	1
415	1307	0

In [ ]: