In [ ]:
```
【ハイパーパラメタ・チューニング】
# 問4
# 同様に、ランダムフォレストのハイパーパラメータの中からいくつかを選び、チューニ
# ングなしとグリッドサーチしたものを比較してみてください。
```

In [37]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

df = pd.read_csv('train.csv')
df = df.drop(['PassengerId','Age','SibSp','Parch','Ticket','Cabin',
              'Name'], axis=1)
df['Embarked'] = df['Embarked'].fillna('S')
df['Embarked'] = df['Embarked'].map({'S': 0, 'C': 1, 'Q': 2})
df['Sex'] = df['Sex'].apply(lambda x: 0 if x=='male' else 1)
df.head()
```

Out[37]:

	Survived	Pclass	Sex	Fare	Embarked
0	0	3	0	7.2500	0
1	1	1	1	71.2833	1
2	1	3	1	7.9250	0
3	1	1	1	53.1000	0
4	0	3	0	8.0500	0

In [38]:
```
# チューニングなし
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

x = df.drop('Survived', axis=1)
y = df['Survived']
x_train, x_test, y_train, y_test = train_test_split(x, y,
                                                    test_size=0.3,
                                                    random_state=0)


rf = RandomForestClassifier()
rf.fit(x_train, y_train)

pred = rf.predict(x_test)
acc = accuracy_score(pred, y_test)

print('accuracy score : {:.5f}' .format(acc))
```

```
accuracy score : 0.81716
```

In [39]:
```
rf.get_params()
```

Out[39]:
```
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': 'auto',
 'max_leaf_nodes': None,
 'max_samples': 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': None,
 'oob_score': False,
 'random_state': None,
 'verbose': 0,
 'warm_start': False}
```

In [40]:
```
# グリッドサーチ
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV

params = {
    'n_estimators': [10, 50, 100, 500, 1000],
    'max_depth': [1, 2, 10, 100, 1000],
    'max_features': [1, 2, 2.2, 3, 3.5, 4]
}

grid = GridSearchCV(estimator=RandomForestClassifier(),
                    param_grid=params, cv=5, n_jobs=-1)
grid.fit(x_train, y_train)
```

Out[40]:
```
GridSearchCV(cv=5, estimator=RandomForestClassifier(), n_jobs=-1,
             param_grid={'max_depth': [1, 2, 10, 100, 1000],
                         'max_features': [1, 2, 2.2, 3, 3.5, 4],
                         'n_estimators': [10, 50, 100, 500, 1000]}
)
```

In [41]:
```
print('results :{}' .format(grid.cv_results_))
print()
print('best score : {:.5f}' .format(grid.best_score_))
print()
print('best parameters : {}' .format(grid.best_params_))
```

```
,
       0.01517634, 0.05662665, 0.10587144, 0.49745607, 0.9886279
1,
       0.03136878, 0.13589334, 0.26784277, 1.26500359, 2.5773800
4,
       0.0183619 , 0.05604243, 0.10468378, 0.48066659, 0.9929690
4,
       0.03521957, 0.13557906, 0.27390103, 1.30557919, 2.5387589
5,
       0.03357382, 0.13819323, 0.26835761, 1.32805471, 2.6879606
7,
       0.03156304, 0.13908744, 0.28023286, 1.39174848, 2.7521384
2,
       0.01520953, 0.05331826, 0.10402112, 0.48277774, 0.9954827
3,
       0.03386145, 0.15157032, 0.28865037, 1.40606294, 2.8294035
9,
       0.01614685, 0.05601478, 0.10896311, 0.49206128, 0.9686502
,
       0.03615885, 0.14420938, 0.28896985, 1.43151855, 2.6502268
```

In [42]:
```
from sklearn.metrics import accuracy_score

rf = RandomForestClassifier(max_depth=10, max_features=3,
                            n_estimators=500)
rf.fit(x_train, y_train)

pred = rf.predict(x_test)
acc = accuracy_score(pred, y_test)

print('accuracy score : {:.5f}' .format(acc))
```

```
accuracy score : 0.81716
```

In [ ]:
```
# グリッドサーチの甲斐がありませんでした。SVMと違いランダムフォレストの方は効果
# がないケースもあるようですが、例えば特徴量が多い時(↑は5個) などでは効果が出る
# 場合もありそうな気がします。
```