Feature selection

January 29, 2016

In [153]:
```
import pandas as pd
from sklearn import ensemble
from sklearn import cross_validation
from sklearn.metrics import classification_report
from sklearn.grid_search import GridSearchCV

data = pd.read_csv('train.csv')

def test_model(X, y, boost=True):
    X_train, X_test, y_train, y_test = cross_validation.train_test_split(
        X, y, test_size=0.3, random_state=0)

    params = {
        "n_estimators": [150, 200, 250],
        "learning_rate": [1.0, 0.9, 0.2, 0.1],
        #"base_estimator__max_depth": [1, 5]
    }

    if boost:
        clf = ensemble.AdaBoostClassifier()
        clf = GridSearchCV(clf, params, cv=5)
    else:
        params = {
          'n_estimators': [150, 200, 250],
          'min_samples_split': [2, 5, 10]
        }
        clf = ensemble.RandomForestClassifier(n_estimators=200,
            min_samples_leaf=2, min_samples_split=10, random_state=1)
        #clf = GridSearchCV(clf, params, cv=5)

    clf.fit(X_train, y_train)

    #print clf.best_estimator_

    y_true, y_pred = y_test, clf.predict(X_test)

    print classification_report(y_true, y_pred)

data['sex_num'] = data.Sex.replace({'male': 1, 'female': 0})
data['others'] = data['Parch'] + data['SibSp']
data['alone'] = data.others.map(lambda v: 0 if v > 0 else 1)

import re
```

```
        data['title'] = data.Name.map(
                    lambda n: re.search('\w+\.', n).group().lower()[:-1])

        data.Cabin = data.Cabin.map(lambda c: str(c).replace('F ', '').lower())

        data['cabin_type'] = data.Cabin.map(lambda c: str(c).split()[0][0])
```

```
In [14]: ind = data.Age > 0
         test_model(data[ind][['Age']], data[ind]['Survived'])
```

```
precision    recall  f1-score   support

          0       0.60      0.93      0.73       125
          1       0.57      0.13      0.22        90

avg / total       0.59      0.60      0.51       215
```

```
In [15]: test_model(data[['sex_num']], data['Survived'])
```

```
precision    recall  f1-score   support

          0       0.83      0.83      0.83       168
          1       0.72      0.71      0.71       100

avg / total       0.79      0.79      0.79       268
```

```
In [16]: test_model(data[ind][['Age', 'sex_num']], data[ind]['Survived'])
```

```
precision    recall  f1-score   support

          0       0.81      0.86      0.83       125
          1       0.79      0.71      0.75        90

avg / total       0.80      0.80      0.80       215
```

```
In [17]: data['age_fixed'] = data.Age

         data.age_fixed.fillna(data.Age.mean(), inplace=True)
         test_model(data[['age_fixed', 'sex_num']], data['Survived'])
```

```
precision    recall  f1-score   support

          0       0.83      0.83      0.83       168
          1       0.72      0.72      0.72       100

avg / total       0.79      0.79      0.79       268
```

```
In [88]: data['age_fixed'] = data.groupby(['Sex']).Age.transform(
             lambda grp: grp.fillna(grp.mean()))

         test_model(data[['age_fixed', 'sex_num']], data['Survived'])
```

```
precision    recall  f1-score   support

          0       0.83      0.83      0.83       168
          1       0.72      0.72      0.72       100

avg / total       0.79      0.79      0.79       268
```

```
In [32]: data['age_fixed'] = data.groupby(['Sex', 'title']).Age.transform(
             lambda grp: grp.fillna(grp.mean()))

         test_model(data[['age_fixed', 'sex_num']], data['Survived'])

precision    recall  f1-score   support

          0       0.83      0.83      0.83       168
          1       0.72      0.72      0.72       100

avg / total       0.79      0.79      0.79       268

In [33]: test_model(data[['age_fixed', 'sex_num', 'Pclass']], data['Survived'])

precision    recall  f1-score   support

          0       0.84      0.83      0.84       168
          1       0.72      0.73      0.73       100

avg / total       0.80      0.79      0.79       268

In [34]: test_model(data[ind][['Age', 'sex_num', 'Pclass']], data[ind]['Survived'])

precision    recall  f1-score   support

          0       0.86      0.84      0.85       125
          1       0.78      0.81      0.80        90

avg / total       0.83      0.83      0.83       215

In [79]: data['age_fixed'] = data.groupby(['Sex']).Age.transform(
             lambda grp: grp.fillna(grp.mean()))

         test_model(data[['age_fixed', 'sex_num', 'Pclass']], data['Survived'])

precision    recall  f1-score   support

          0       0.84      0.83      0.83       168
          1       0.72      0.73      0.72       100

avg / total       0.79      0.79      0.79       268
```

More precise age makes things only worse??

```
In [36]: test_model(data[ind][['Age', 'sex_num', 'Pclass', 'Fare']], data[ind]['Survived'])

precision    recall  f1-score   support

          0       0.81      0.86      0.84       125
          1       0.79      0.72      0.76        90

avg / total       0.80      0.80      0.80       215

In [37]: test_model(data[ind][['age_fixed', 'sex_num', 'Pclass', 'Fare']], data[ind]['Survived'])
```

```
precision    recall  f1-score   support

          0       0.81      0.86      0.84       125
          1       0.79      0.72      0.76        90

avg / total       0.80      0.80      0.80       215
```

```
In [39]: test_model(data[ind][['Age', 'sex_num', 'Pclass', 'others']], data[ind]['Survived'])
```

```
precision    recall  f1-score   support

          0       0.85      0.86      0.85       125
          1       0.80      0.79      0.79        90

avg / total       0.83      0.83      0.83       215
```

```
In [43]: test_model(data[ind][['Age', 'sex_num', 'Pclass', 'alone']], data[ind]['Survived'])
```

```
precision    recall  f1-score   support

          0       0.86      0.84      0.85       125
          1       0.78      0.81      0.80        90

avg / total       0.83      0.83      0.83       215
```

```
In [52]: cabin = pd.get_dummies(data.cabin_type, prefix='cabin')

         sdata = data[ind][['Age', 'sex_num', 'Pclass']]
         sdata = pd.merge(sdata, cabin, left_index=True, right_index=True)

         test_model(sdata, data[ind]['Survived'])
```

```
precision    recall  f1-score   support

          0       0.85      0.84      0.84       125
          1       0.78      0.79      0.78        90

avg / total       0.82      0.82      0.82       215
```

```
In [53]: test_model(data[ind][['Age', 'sex_num', 'Fare']], data[ind]['Survived'])
```

```
precision    recall  f1-score   support

          0       0.82      0.82      0.82       125
          1       0.75      0.74      0.75        90

avg / total       0.79      0.79      0.79       215
```

```
In [82]: data['age_bin'] = pd.qcut(data.Age, 10)

         age_bin = pd.get_dummies(data.age_bin, prefix='bin')

         sdata = data[['Age', 'sex_num', 'Pclass']]
         sdata = pd.merge(sdata, age_bin, left_index=True, right_index=True)

         test_model(sdata[ind], data[ind]['Survived'])
```

```
precision    recall  f1-score   support

          0       0.86      0.86      0.86       125
          1       0.80      0.80      0.80        90

avg / total       0.83      0.83      0.83       215
```

```
In [68]: test_model(data[ind][['Age', 'sex_num', 'Pclass', 'Parch']], data[ind]['Survived'])
```

```
precision    recall  f1-score   support

          0       0.85      0.82      0.83       125
          1       0.76      0.80      0.78        90

avg / total       0.81      0.81      0.81       215
```

```
In [69]: test_model(data[ind][['Age', 'sex_num', 'Pclass', 'SibSp']], data[ind]['Survived'])
```

```
precision    recall  f1-score   support

          0       0.86      0.82      0.84       125
          1       0.76      0.81      0.78        90

avg / total       0.82      0.81      0.81       215
```

```
In [90]: test_model(data[['age_fixed', 'sex_num', 'Pclass', 'Fare', 'others']], data['Survived'], False
```

```
precision    recall  f1-score   support

          0       0.84      0.88      0.86       168
          1       0.78      0.71      0.74       100

avg / total       0.82      0.82      0.82       268
```

```
In [154]: data['age_bin'] = pd.qcut(data.Age, 10)
          data['crew'] = data.Fare.map(lambda f: 0 if f > 0 else 1)
          data['cabin_num'] = data.Cabin.map(lambda c: len(c.split(' ')))

          nums = data.Cabin.map(lambda c: len(str(c).split(' ')))
          data['fare_fixed'] = (data.Fare / nums)

          age_bin = pd.get_dummies(data.age_bin, prefix='bin')
          title = pd.get_dummies(data.title, prefix='title')
          embarked = pd.get_dummies(data.Embarked, prefix='title')

          sdata = data[['sex_num', 'Pclass', 'Fare', 'others']]
          sdata = pd.merge(sdata, age_bin, left_index=True, right_index=True)
          sdata = pd.merge(sdata, title, left_index=True, right_index=True)
          #sdata = pd.merge(sdata, embarked, left_index=True, right_index=True)

          test_model(sdata, data['Survived'], False)
```

```
precision    recall  f1-score   support

          0       0.85      0.88      0.86       168
          1       0.78      0.75      0.77       100

avg / total       0.83      0.83      0.83       268
```