一.1

In [2]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
data['workclass'] = labelencoder.fit_transform(data['workclass'])
data['education'] = labelencoder.fit_transform(data['education'])
data['marital-status'] = labelencoder.fit_transform(data['marital-status'])
data['occupation'] = labelencoder.fit_transform(data['occupation'])
data['relationship'] = labelencoder.fit_transform(data['relationship'])
data['race'] = labelencoder.fit_transform(data['race'])
data['gender'] = labelencoder.fit_transform(data['gender'])
data['native-country'] = labelencoder.fit_transform(data['native-country'])
data['income'] = labelencoder.fit_transform(data['income'])
```

In [3]:
```
X=data.drop(['income'],axis=1)
y=data['income']
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.3,random_state=15)
clf=KNeighborsClassifier(n_neighbors=3,p=2,weights='distance',algorithm='brute')
clf.fit(X_train,y_train)
```

Out[3]:
```
KNeighborsClassifier(algorithm='brute', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                     weights='distance')
```

In [4]:
```
clf.score(X_test,y_test)
```

Out[4]:
```
0.614149008885851
```

一.2

5]:
```
from mlxtend.feature_selection import SequentialFeatureSelector
from sklearn.ensemble import RandomForestClassifier

sfs = SequentialFeatureSelector(RandomForestClassifier(),
          k_features=(1,14),
          forward=True,
          floating=False,
          scoring='accuracy',
          cv=5)
sfs.fit(X_train, y_train)
print(f"Best score achieved: {sfs.k_score_}, Feature's names: {sfs.k_feature_names_}")
display(pd.DataFrame(sfs.get_metric_dict()))
```

```
Best score achieved: 0.8249589453981839, Feature's names: ('age', 'workclass', 'fnlwgt', 'educational-num', 'marital-status',
'occupation', 'relationship', 'race', 'gender', 'capital-gain', 'capital-loss', 'hours-per-week', 'native-country')
```

	1	2	3	4	5	6	
feature_idx	(7,)	(7, 10)	(6, 7, 10)	(6, 7, 10, 11)	(6, 7, 9, 10, 11)	(5, 6, 7, 9, 10, 11)	(5, 6, 7, 8, 9
cv_scores	[0.7503660322108345, 0.7562225475841874, 0.761...	[0.780380673499268, 0.7840409956076134, 0.7897...	[0.7928257686676428, 0.8045387994143485, 0.813...	[0.8038067349926794, 0.8184480234260615, 0.819...	[0.8045387994143485, 0.8169838945827232, 0.819...	[0.8023426061493412, 0.8125915080527086, 0.816...	[0.801610541 0.8111273792
avg_score	0.750255	0.779404	0.800645	0.814561	0.813828	0.812364	
feature_names	(relationship,)	(relationship, capital-gain)	(occupation, relationship, capital-gain)	(occupation, relationship, capital-gain, capit...	(occupation, relationship, gender, capital-gai...	(marital-status, occupation, relationship, gen...	(marita occ relationsh
ci_bound	0.0114394	0.00967668	0.0122221	0.0146535	0.0136151	0.0157531	0.0
std_dev	0.00890026	0.00752878	0.00950918	0.0114009	0.010593	0.0122564	0.0
std_err	0.00445013	0.00376439	0.00475459	0.00570047	0.0052965	0.0061282	0.0

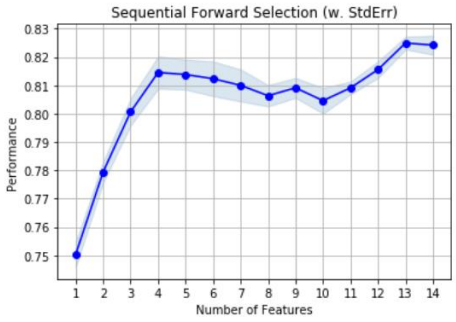
一.3、4

In [10]:
```
from mlxtend.plotting import plot_sequential_feature_selection as plot_sfs

print(sfs.k_feature_idx_)
print('CV Score:')
print(sfs.k_score_)

fig = plot_sfs(sfs.get_metric_dict(), kind='std_err')
plt.title('Sequential Forward Selection (w. StdErr)')
plt.grid()
plt.show()
```

```
(0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)
CV Score:
0.8249589453981839
```



一.5

把 education 欄位去除之後訓練發現跟一.1 的準確度比有些微上升

```
data['education'] = labelencoder.fit_transform(data['education'])
data['marital-status'] = labelencoder.fit_transform(data['marital-status'])
data['occupation'] = labelencoder.fit_transform(data['occupation'])
data['relationship'] = labelencoder.fit_transform(data['relationship'])
data['race'] = labelencoder.fit_transform(data['race'])
data['gender'] = labelencoder.fit_transform(data['gender'])
data['native-country'] = labelencoder.fit_transform(data['native-country'])
data['income'] = labelencoder.fit_transform(data['income'])


X=data.drop(['income','education'],axis=1)
y=data['income']
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.3,random_state=15)
clf=KNeighborsClassifier(n_neighbors=3,p=2,weights='distance',algorithm='brute')
clf.fit(X_train,y_train)
```

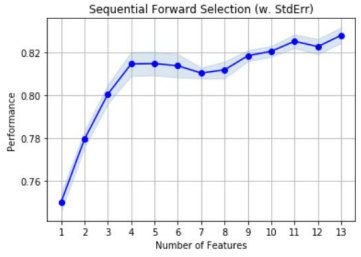
Out[15]:
```
KNeighborsClassifier(algorithm='brute', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                     weights='distance')
```

In [16]:
```
clf.score(X_test,y_test)
```

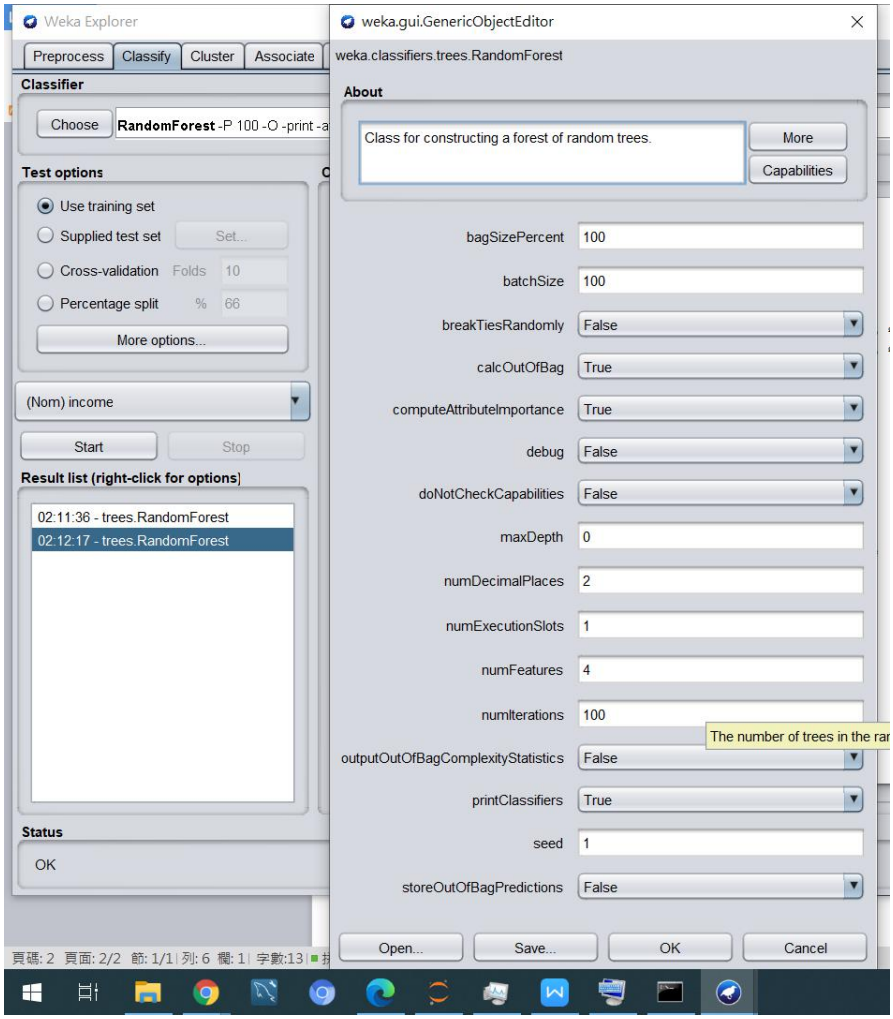
Out[16]:
```
0.6144907723855092
```

```
fig = plot_sfs(sfs.get_metric_dict(), kind='std_err')
plt.title('Sequential Forward Selection (w. StdErr)')
plt.grid()
plt.show()
```

```
Best score achieved: 0.8278884902310963, Feature's names: ('age', 'workclass', 'fnlwgt', 'educational-num', 'marital-status',
'occupation', 'relationship', 'race', 'gender', 'capital-gain', 'capital-loss', 'hours-per-week', 'native-country')
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
('age', 'workclass', 'fnlwgt', 'educational-num', 'marital-status', 'occupation', 'relationship', 'race', 'gender', 'capital-ga
in', 'capital-loss', 'hours-per-week', 'native-country')
CV Score:
0.8278884902310963
```



二.1



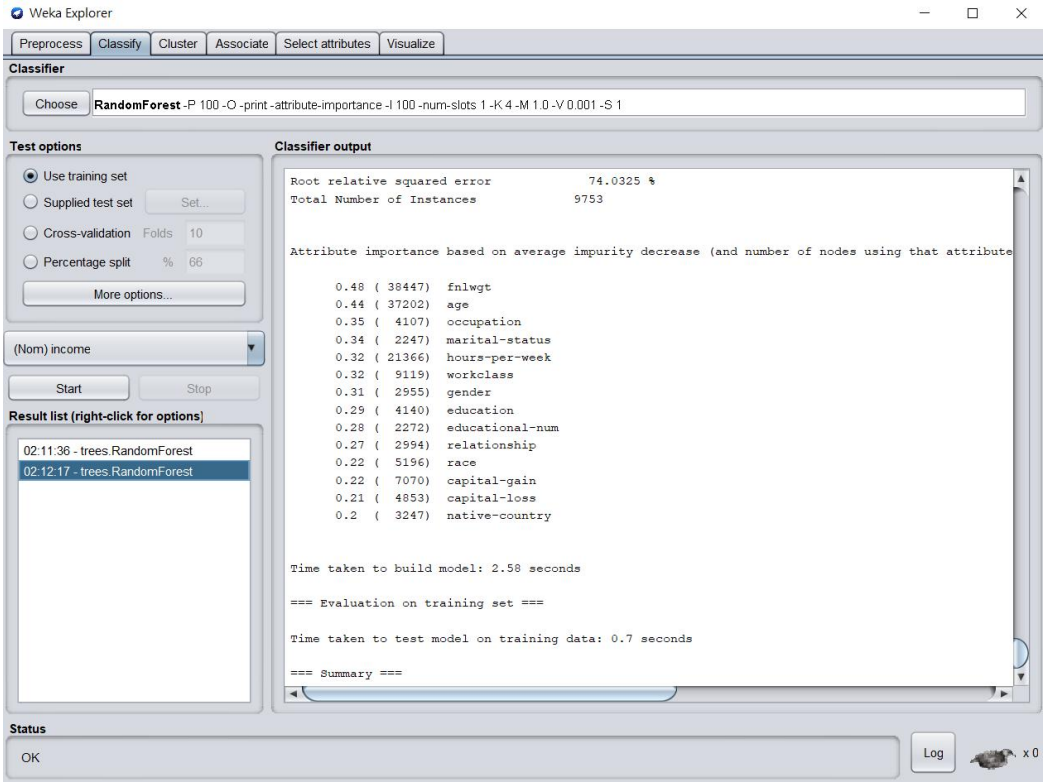
二.2

挑: fnlwgt

age

occupation

marital-status


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
RandomForest -P 100 -O -print -attribute-importance -I 100 -num-slots 1 -K 4 -M 1.0 -V 0.001 -S 1
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) income
Start
Stop
Result list (right-click for options)
02:11:36 - trees.RandomForest
02:12:17 - trees.RandomForest
Classifier output
Root relative squared error 74.0325 %
Total Number of Instances 9753
Attribute importance based on average impurity decrease (and number of nodes using that attribute
0.48 (38447) fnlwgt
0.44 (37202) age
0.35 (4107) occupation
0.34 (2247) marital-status
0.32 (21366) hours-per-week
0.32 (9119) workclass
0.31 (2955) gender
0.29 (4140) education
0.28 (2272) educational-num
0.27 (2994) relationship
0.22 (5196) race
0.22 (7070) capital-gain
0.21 (4853) capital-loss
0.2 (3247) native-country
Time taken to build model: 2.58 seconds
=== Evaluation on training set ===
Time taken to test model on training data: 0.7 seconds
=== Summary ===
Status
OK
Log
x 0