Chapter 9: Demo Cross Validation

In [1]:
```
#Import scikit-learn dataset library
from sklearn import datasets

#Load dataset
iris = datasets.load_iris()
```

In [2]:
```
type(iris)
```

Out[2]:
```
sklearn.utils.Bunch
```

In [3]:
```
# print the label species(setosa, versicolor,virginica)
print(iris.target_names)

# print the names of the four features
print(iris.feature_names)
```

```
['setosa' 'versicolor' 'virginica']
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (c
m)']
```

In [4]:
```
# print the iris data (top 5 records)
print(iris.data[0:5])

# print the iris labels (0:setosa, 1:versicolor, 2:virginica)
print(iris.target[:5])
```

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]]
[0 0 0 0 0]
```

In [5]:
```
# Creating a DataFrame of given iris dataset.
import pandas as pd
data=pd.DataFrame({
    'sepal length':iris.data[:,0],
    'sepal width':iris.data[:,1],
    'petal length':iris.data[:,2],
    'petal width':iris.data[:,3],
    'species':iris.target
})
data.head()
```

Out[5]:

	sepal length	sepal width	petal length	petal width	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

In [7]:
```
X=data[['petal length', 'petal width']]
y=data['species']
```

Select model

In [8]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
```

In [9]:
```
# Tính độ chính xác theo: DecisionTree, RandomForest, Logistic, Naive Bayes, SVM,
from sklearn.model_selection import cross_val_score
models = [
    KNeighborsClassifier(n_neighbors=6),
    DecisionTreeClassifier(),
    RandomForestClassifier(n_estimators=30),
    SVC(kernel='linear'),
    GaussianNB(),
    LogisticRegression(solver='lbfgs', multi_class='auto')
]
CV = 10 # số lần lặp
# lưu kết quả acc của 6 model, mỗi model 10 giá trị acc
cv_df = pd.DataFrame(index=range(CV * len(models)))
entries = [] # lưu 2 thông tin là model_name và accuracies.mean()

for model in models: # duyệt từng model trong ds model
    model_name = model.__class__.__name__
    accuracies = cross_val_score(model, X, y, scoring='accuracy', cv=CV)
    print(accuracies)
    entries.append([model_name, accuracies.mean()])

cv_df = pd.DataFrame(entries, columns=['model_name', 'accuracy'])
```

```
[1.         0.93333333 1.         0.93333333 0.93333333 0.93333333
 0.93333333 1.         1.         1.        ]
[1.         0.93333333 1.         0.93333333 0.93333333 0.8
 0.93333333 0.93333333 1.         1.        ]
[1.         0.93333333 1.         0.93333333 0.93333333 0.86666667
 0.86666667 1.         1.         1.        ]
[1.         0.93333333 1.         0.93333333 0.93333333 0.93333333
 0.86666667 1.         1.         1.        ]
[1.         0.93333333 1.         0.93333333 0.93333333 0.93333333
 0.86666667 1.         1.         1.        ]
[1.         0.93333333 1.         0.93333333 0.93333333 0.93333333
 0.86666667 1.         1.         1.        ]
```

In [10]:
```
cv_df
```

Out[10]:

	model_name	accuracy
0	KNeighborsClassifier	0.966667
1	DecisionTreeClassifier	0.946667
2	RandomForestClassifier	0.953333
3	SVC	0.960000
4	GaussianNB	0.960000
5	LogisticRegression	0.960000

In [11]:
```
import matplotlib.pyplot as plt
```

In [12]:
```
plt.figure(figsize=(8,4))
plt.bar(cv_df['model_name'],cv_df['accuracy'], )
plt.ylim(0.5, 1)
plt.xlabel('model_name')
plt.ylabel('Mean of accuracies')
plt.xticks(rotation='vertical')
plt.title("Accuracies of Algorithms")

plt.show()
```