

```
In [7]: import pandas as pd
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.linear_model import LogisticRegression
        from sklearn.svm import SVC
        from sklearn.datasets import load_iris
        import numpy as np
```

```
In [8]: iris = load_iris()
```

```
In [9]: dir(iris)
```

```
Out[9]: ['DESCR',
         'data',
         'feature_names',
         'filename',
         'frame',
         'target',
         'target_names']
```

```
In [10]: from sklearn.model_selection import train_test_split
```

```
In [11]: x_train, x_test, y_train, y_test = train_test_split(iris.data, iris.target , test
```

```
In [12]: lr = LogisticRegression(solver = 'liblinear' , multi_class = 'ovr')
```

```
In [13]: lr.fit(x_train, y_train)
```

```
Out[13]: LogisticRegression(multi_class='ovr', solver='liblinear')
```

```
In [14]: lr.score(x_test , y_test)
```

```
Out[14]: 0.9333333333333333
```

```
In [15]: rf = RandomForestClassifier(n_estimators = 40)
```

```
In [16]: rf.fit(x_train , y_train)
```

```
Out[16]: RandomForestClassifier(n_estimators=40)
```

```
In [17]: rf.score(x_test , y_test)
```

```
Out[17]: 0.9333333333333333
```

```
In [18]: svm = SVC (gamma = 'auto')
```

```
In [19]: svm.fit(x_train , y_train)
```

```
Out[19]: SVC(gamma='auto')
```

```
In [20]: svm.score(x_test , y_test)
```

```
Out[20]: 1.0
```

```
In [21]: def get_score(model,x_train, x_test, y_train, y_test ):
          model.fit(x_train, y_train)
          return model.score(x_test, y_test)
```

```
In [22]: from sklearn.model_selection import StratifiedKFold
          folds = StratifiedKFold(n_splits = 3)

          scores_lr = []
          scores_rf = []
          scores_svm =[]
```

```
In [24]: for train_index , test_index in folds.split(iris.data , iris.target):
          x_train, x_test, y_train, y_test = iris.data[train_index] , iris.data[test_index]
          iris.target[train_index] , iris.target[test_index]
          scores_lr.append(get_score(LogisticRegression( solver = 'liblinear', multi_class = 'ovr'
```

```
In [25]: scores_lr
```

```
Out[25]: [0.96, 0.96, 0.94]
```

```
In [26]: from sklearn.model_selection import cross_val_score
```

```
In [38]: lr = cross_val_score(LogisticRegression(solver = 'liblinear', multi_class = 'ovr',
          lr
          np.average(lr)
```

```
Out[38]: 0.9533333333333333
```

```
In [41]: rf = cross_val_score(RandomForestClassifier(n_estimators = 40), iris.data, iris.target, cv=5)
          print(rf)
          np.average(rf)
```

```
[0.98 0.94 0.94]
```

```
Out[41]: 0.9533333333333333
```

```
In [40]: svm = cross_val_score(SVC (gamma = 'auto'), iris.data, iris.target, cv = 3)
          print(svm)
          np.average(svm)
```

[0.98 0.98 0.96]

Out[40]: 0.9733333333333333

In []: