In [1]:

```
%matplotlib notebook
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
np.set_printoptions(precision = 2)
from sklearn.datasets import make_regression
```

In [2]:

```
fruits = pd.read_table('fruit_data_with_colors.txt')

X_fruits_2d = fruits[['height', 'width', 'mass', 'color_score']]
y_fruits_2d = fruits['fruit_label']
```

Logistic Regression

In [39]:

```
from sklearn.linear_model import LogisticRegression

y_fruits_apple = y_fruits_2d == 1
X_train, X_test, y_train, y_test = train_test_split(X_fruits_2d, y_fruits_apple, random_state = 0)

clf = LogisticRegression().fit(X_train, y_train)

print('Accuracy of Logistic regression classifier on test set: {:.2f}'.format(clf.score(X_test, y_te
```

```
Accuracy of Logistic regression classifier on test set: 0.67
```

In [40]:

```
LR = format(clf.score(X_test, y_test))
```

Support Vector Machine

In [41]:

```
from sklearn.svm import SVC

X_train, X_test, y_train, y_test = train_test_split(X_fruits_2d, y_fruits_apple, random_state = 0)

clf = SVC(kernel = 'linear').fit(X_train, y_train)

print('Accuracy of Logistic regression classifier on test set: {:.2f}'.format(clf.score(X_test, y_te
```

```
Accuracy of Logistic regression classifier on test set: 0.67
```

In [42]:

```
SVM = format(clf.score(X_test, y_test))
```

Decision Tree

In [30]:

```
from sklearn.tree import DecisionTreeClassifier

X_train, X_test, y_train, y_test = train_test_split(X_fruits_2d, y_fruits_apple, random_state = 0)

clf = DecisionTreeClassifier().fit(X_train, y_train)

print('Accuracy of Logistic regression classifier on test set: {:.2f}'.format(clf.score(X_test, y_te
```

```
Accuracy of Logistic regression classifier on test set: 0.80
```

In [31]:

```
DT = format(clf.score(X_test, y_test))
```

Random Forest

In [32]:

```
from sklearn.ensemble import RandomForestClassifier

X_train, X_test, y_train, y_test = train_test_split(X_fruits_2d, y_fruits_apple, random_state = 0)

clf = RandomForestClassifier(n_estimators = 10, random_state = 0).fit(X_train, y_train)

print('Accuracy of Logistic regression classifier on test set: {:.2f}'.format(clf.score(X_test, y_te
```

```
Accuracy of Logistic regression classifier on test set: 1.00
```

In [33]:

```
RF = format(clf.score(X_test, y_test))
```

In [43]:

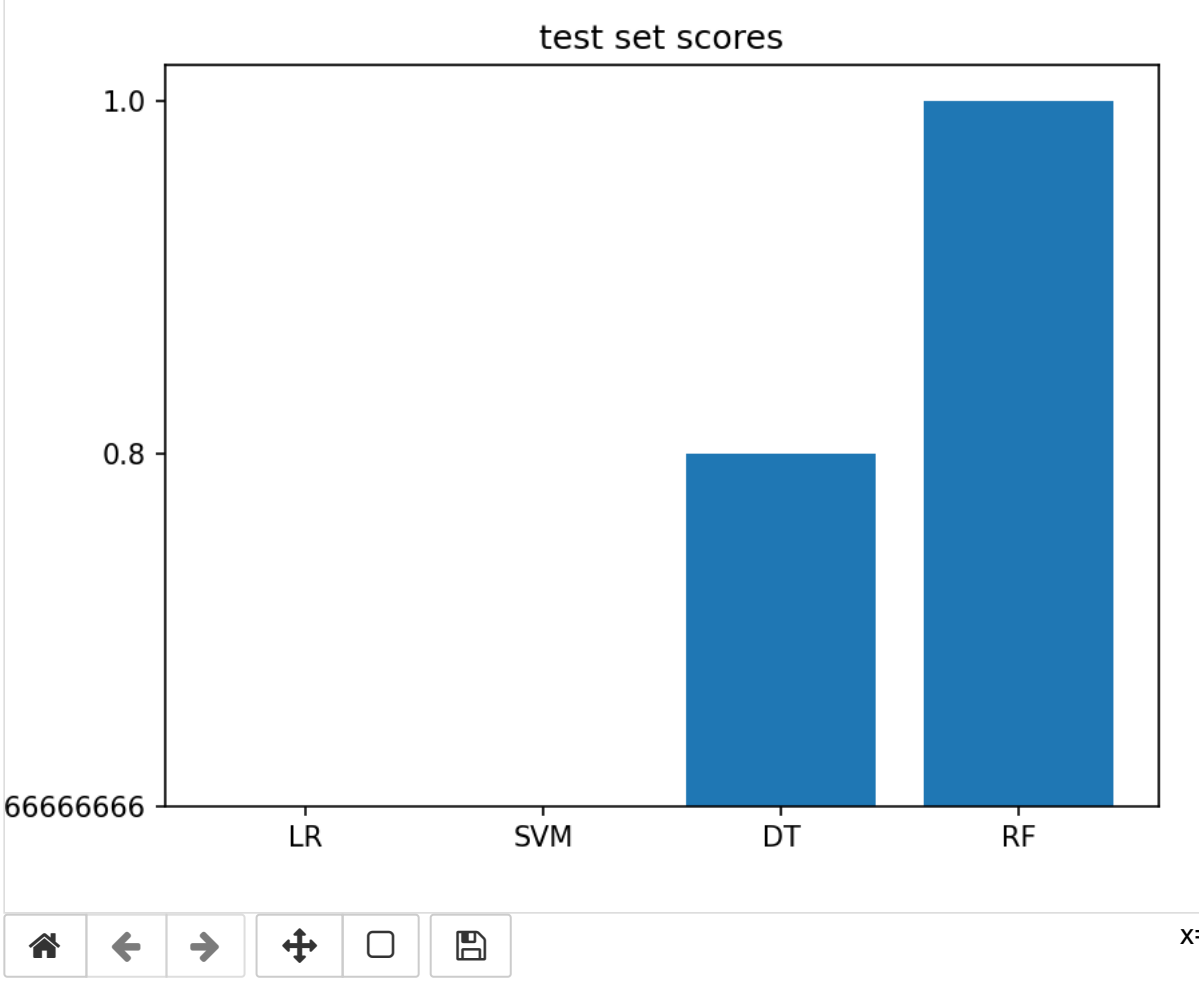
```
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(4)

model = ['LR', 'SVM', 'DT', 'RF']
score = [LR, SVM, DT, RF]

plt.title('test set scores')
plt.bar(x, score)
plt.xticks(x, model)
plt.show()
```

Figure 1



1.가장 성능이 좋은 모델은 Random Forest이다.

2.[무게가 120, 너비가 6, 높이가 8, color_score가 0.7]인 과일은 레몬으로 추정한다.