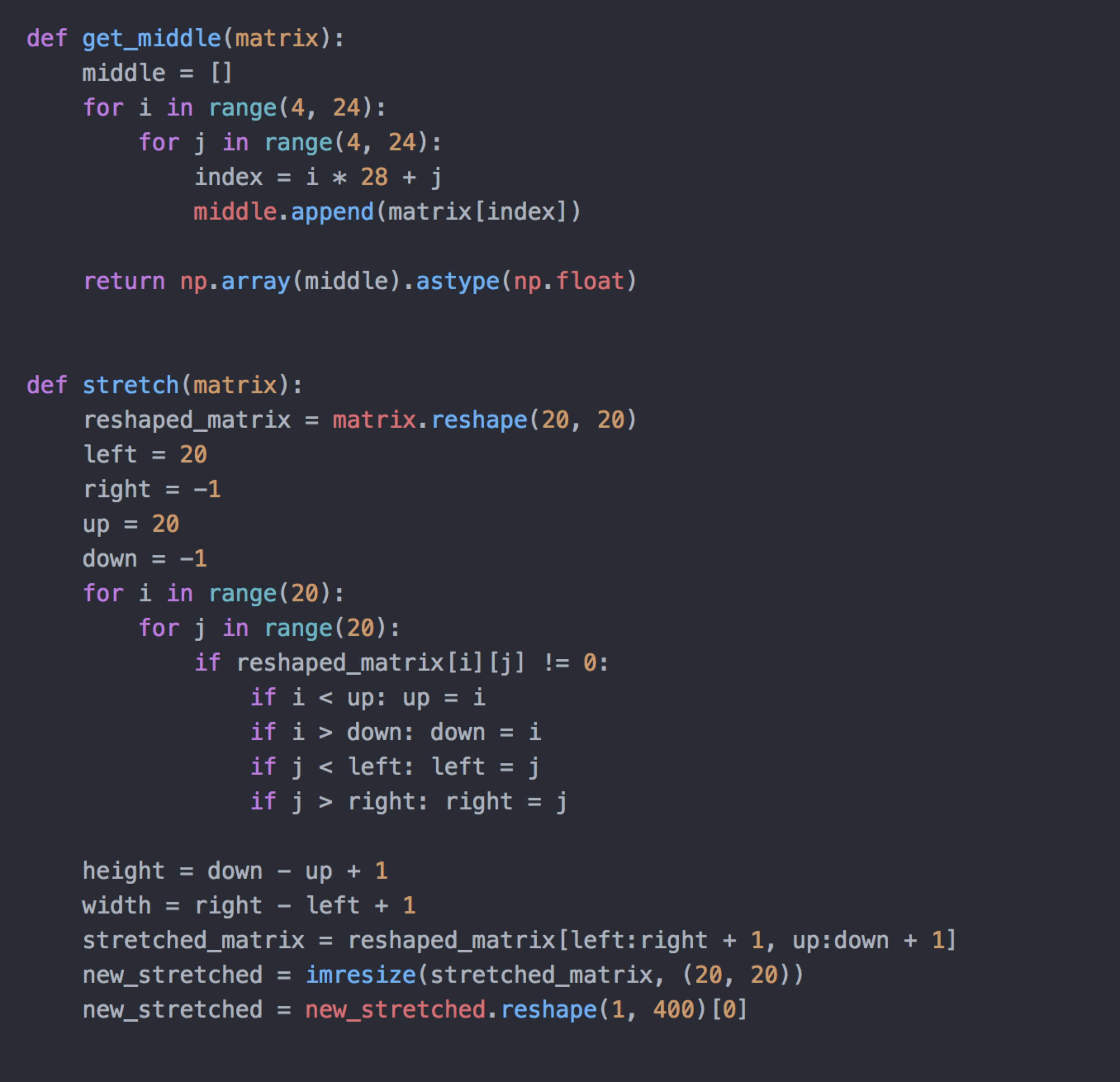
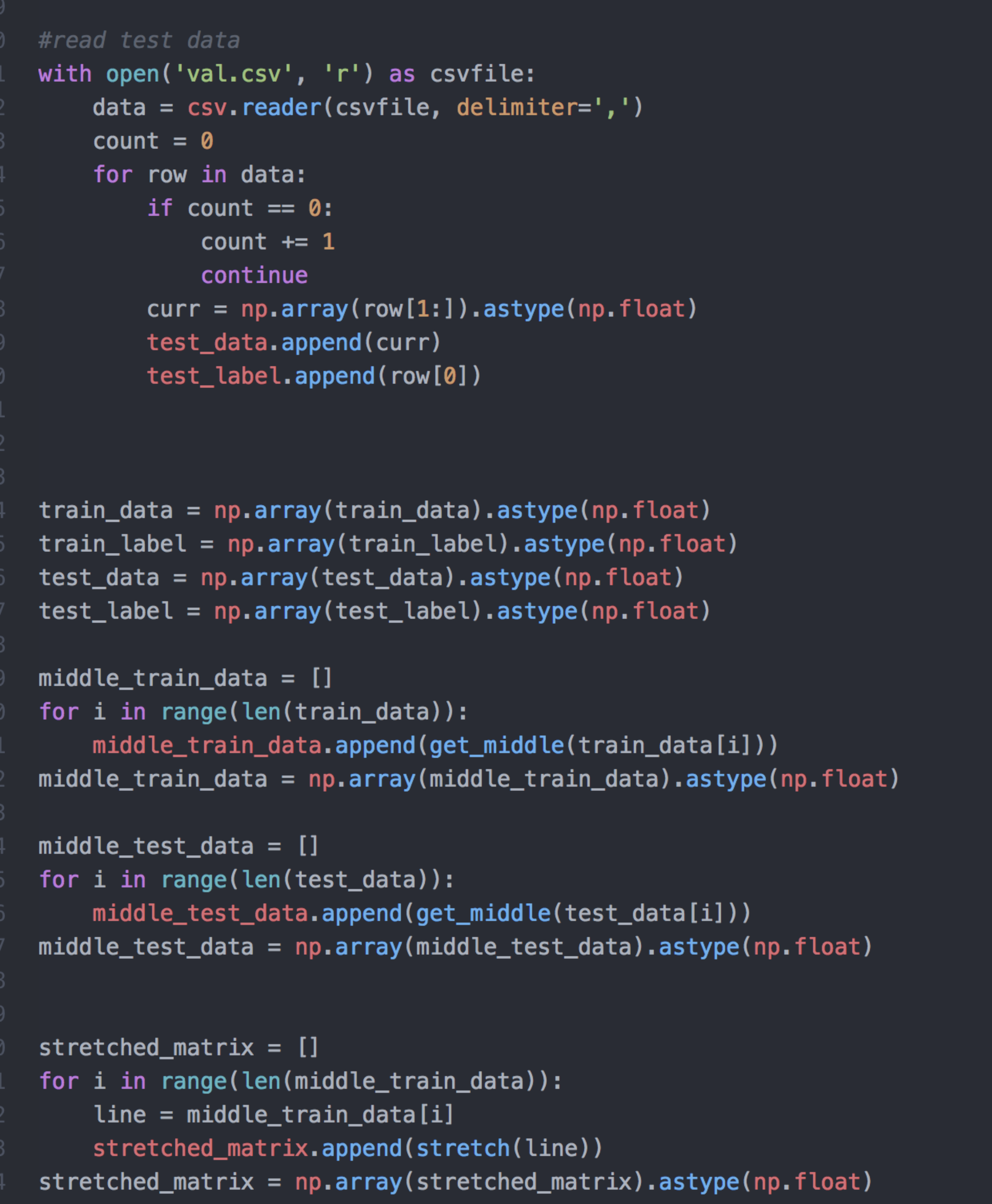
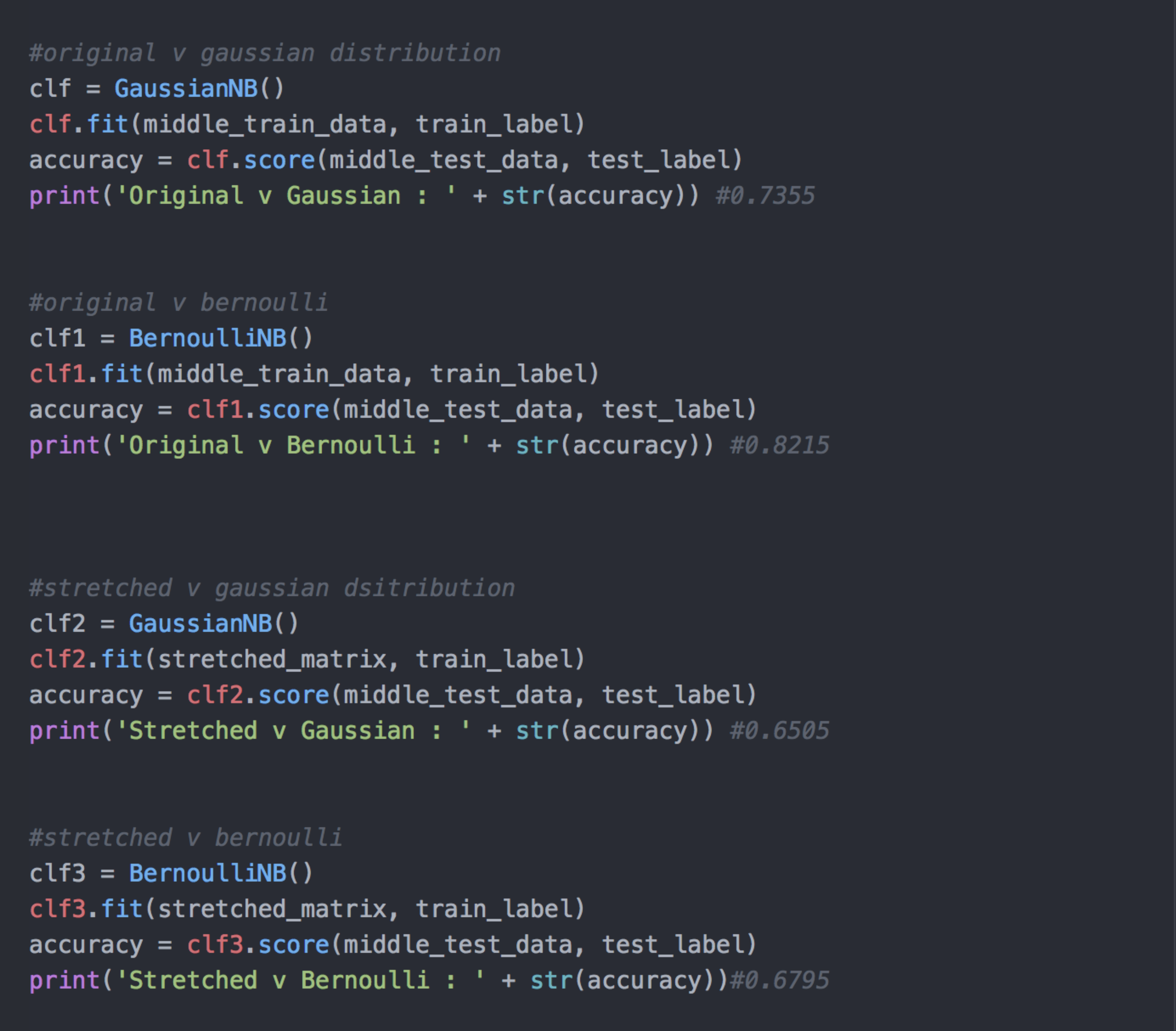


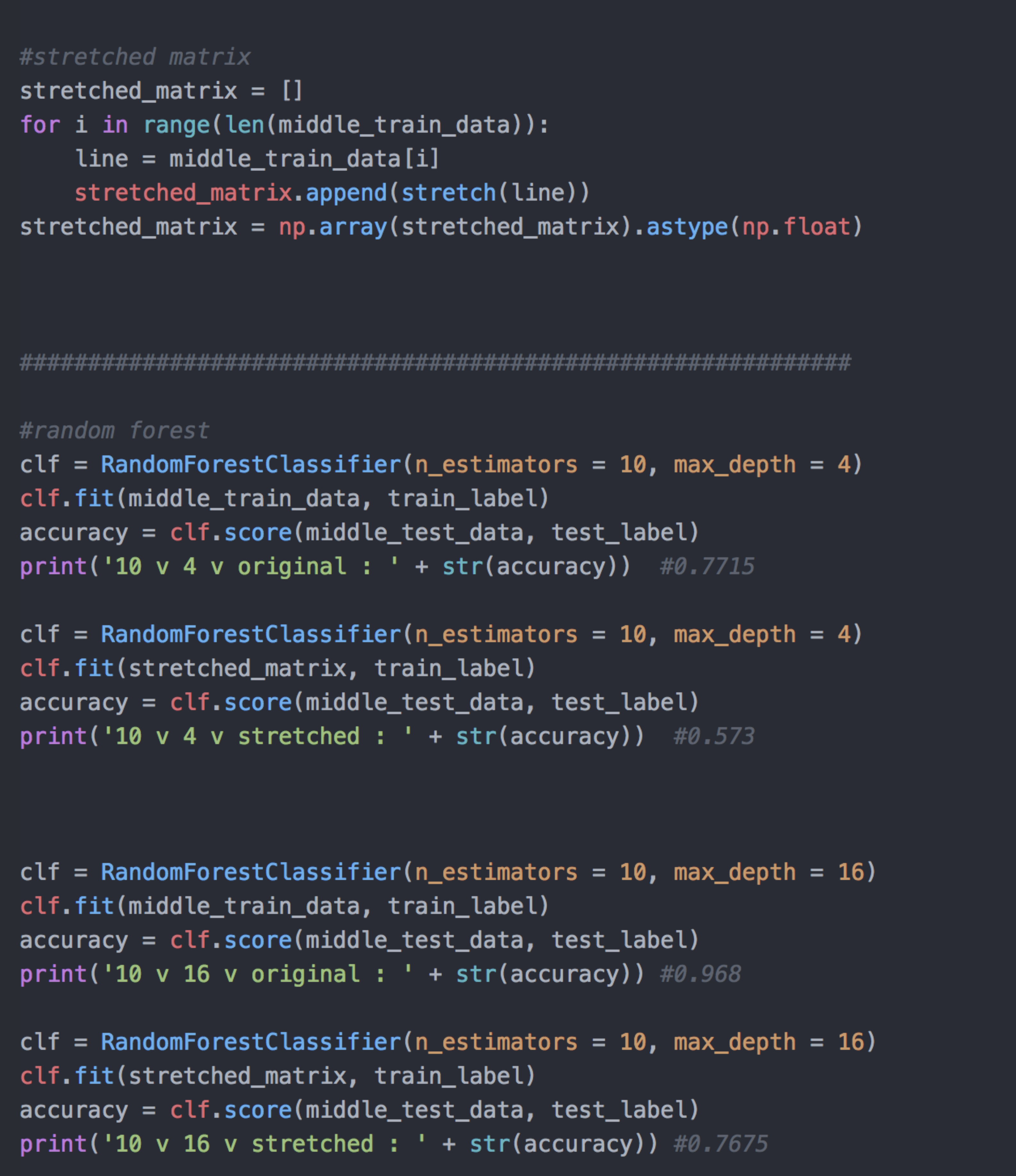
```python
def get_middle(matrix):
    middle = []
    for i in range(4, 24):
        for j in range(4, 24):
            index = i * 28 + j
            middle.append(matrix[index])

    return np.array(middle).astype(np.float)


def stretch(matrix):
    reshaped_matrix = matrix.reshape(20, 20)
    left = 20
    right = -1
    up = 20
    down = -1
    for i in range(20):
        for j in range(20):
            if reshaped_matrix[i][j] != 0:
                if i < up: up = i
                if i > down: down = i
                if j < left: left = j
                if j > right: right = j

    height = down - up + 1
    width = right - left + 1
    stretched_matrix = reshaped_matrix[left:right + 1, up:down + 1]
    new_stretched = imresize(stretched_matrix, (20, 20))
    new_stretched = new_stretched.reshape(1, 400)[0]
```



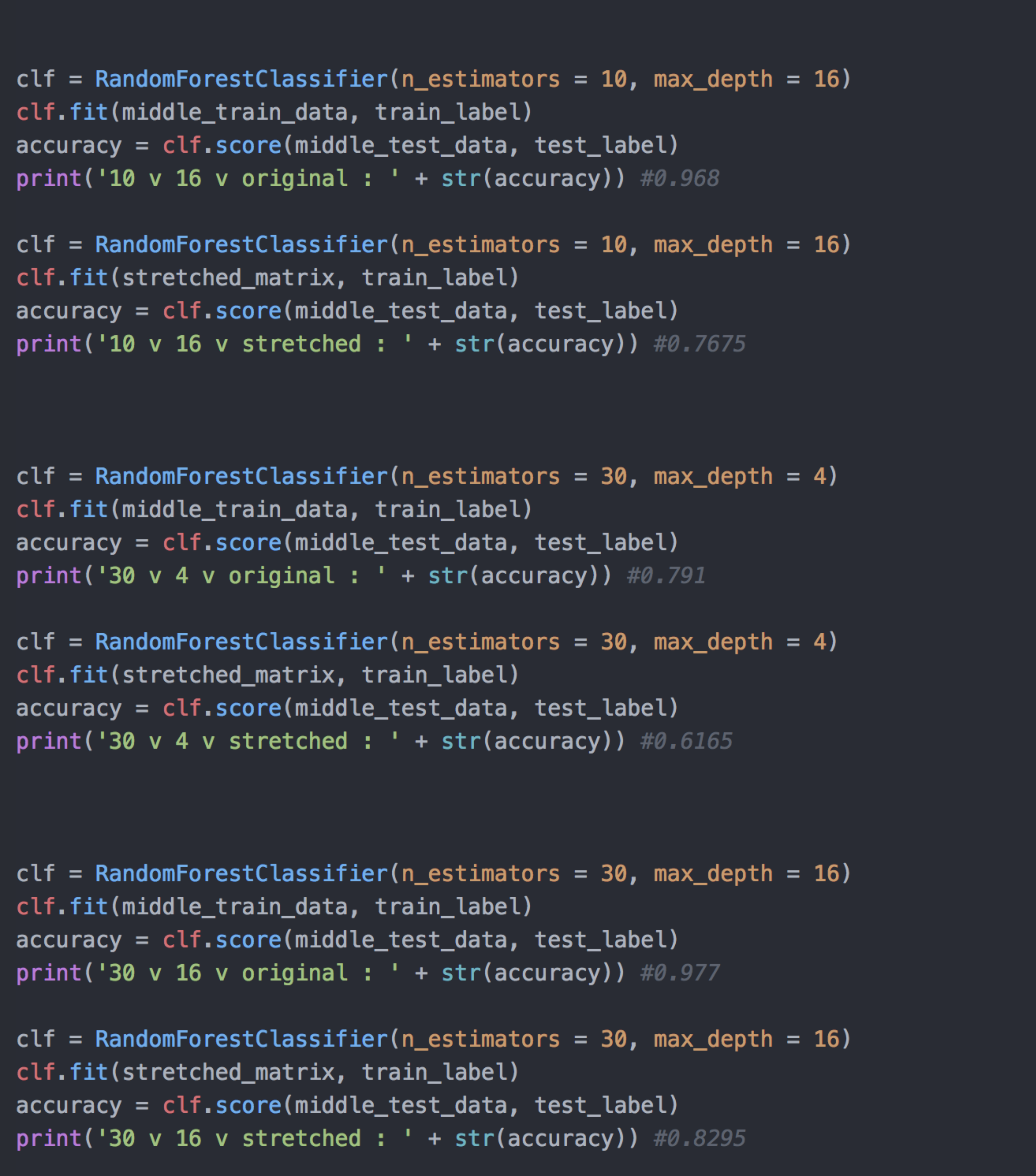
```
#read test data
with open('val.csv', 'r') as csvfile:
    data = csv.reader(csvfile, delimiter=',')
    count = 0
    for row in data:
        if count == 0:
            count += 1
            continue
        curr = np.array(row[1:]).astype(np.float)
        test_data.append(curr)
        test_label.append(row[0])



train_data = np.array(train_data).astype(np.float)
train_label = np.array(train_label).astype(np.float)
test_data = np.array(test_data).astype(np.float)
test_label = np.array(test_label).astype(np.float)

middle_train_data = []
for i in range(len(train_data)):
    middle_train_data.append(get_middle(train_data[i]))
middle_train_data = np.array(middle_train_data).astype(np.float)

middle_test_data = []
for i in range(len(test_data)):
    middle_test_data.append(get_middle(test_data[i]))
middle_test_data = np.array(middle_test_data).astype(np.float)


stretched_matrix = []
for i in range(len(middle_train_data)):
    line = middle_train_data[i]
    stretched_matrix.append(stretch(line))
stretched_matrix = np.array(stretched_matrix).astype(np.float)
```



```
#original v gaussian distribution
clf = GaussianNB()
clf.fit(middle_train_data, train_label)
accuracy = clf.score(middle_test_data, test_label)
print('Original v Gaussian : ' + str(accuracy)) #0.7355


#original v bernoulli
clf1 = BernoulliNB()
clf1.fit(middle_train_data, train_label)
accuracy = clf1.score(middle_test_data, test_label)
print('Original v Bernoulli : ' + str(accuracy)) #0.8215



#stretched v gaussian dsitribution
clf2 = GaussianNB()
clf2.fit(stretched_matrix, train_label)
accuracy = clf2.score(middle_test_data, test_label)
print('Stretched v Gaussian : ' + str(accuracy)) #0.6505


#stretched v bernoulli
clf3 = BernoulliNB()
clf3.fit(stretched_matrix, train_label)
accuracy = clf3.score(middle_test_data, test_label)
print('Stretched v Bernoulli : ' + str(accuracy))#0.6795
```



```python
#stretched matrix
stretched_matrix = []
for i in range(len(middle_train_data)):
    line = middle_train_data[i]
    stretched_matrix.append(stretch(line))
stretched_matrix = np.array(stretched_matrix).astype(np.float)



#############################################################


#random forest
clf = RandomForestClassifier(n_estimators = 10, max_depth = 4)
clf.fit(middle_train_data, train_label)
accuracy = clf.score(middle_test_data, test_label)
print('10 v 4 v original : ' + str(accuracy))  #0.7715

clf = RandomForestClassifier(n_estimators = 10, max_depth = 4)
clf.fit(stretched_matrix, train_label)
accuracy = clf.score(middle_test_data, test_label)
print('10 v 4 v stretched : ' + str(accuracy))  #0.573



clf = RandomForestClassifier(n_estimators = 10, max_depth = 16)
clf.fit(middle_train_data, train_label)
accuracy = clf.score(middle_test_data, test_label)
print('10 v 16 v original : ' + str(accuracy)) #0.968

clf = RandomForestClassifier(n_estimators = 10, max_depth = 16)
clf.fit(stretched_matrix, train_label)
accuracy = clf.score(middle_test_data, test_label)
print('10 v 16 v stretched : ' + str(accuracy)) #0.7675
```



```
clf = RandomForestClassifier(n_estimators = 10, max_depth = 16)
clf.fit(middle_train_data, train_label)
accuracy = clf.score(middle_test_data, test_label)
print('10 v 16 v original : ' + str(accuracy)) #0.968

clf = RandomForestClassifier(n_estimators = 10, max_depth = 16)
clf.fit(stretched_matrix, train_label)
accuracy = clf.score(middle_test_data, test_label)
print('10 v 16 v stretched : ' + str(accuracy)) #0.7675



clf = RandomForestClassifier(n_estimators = 30, max_depth = 4)
clf.fit(middle_train_data, train_label)
accuracy = clf.score(middle_test_data, test_label)
print('30 v 4 v original : ' + str(accuracy)) #0.791

clf = RandomForestClassifier(n_estimators = 30, max_depth = 4)
clf.fit(stretched_matrix, train_label)
accuracy = clf.score(middle_test_data, test_label)
print('30 v 4 v stretched : ' + str(accuracy)) #0.6165



clf = RandomForestClassifier(n_estimators = 30, max_depth = 16)
clf.fit(middle_train_data, train_label)
accuracy = clf.score(middle_test_data, test_label)
print('30 v 16 v original : ' + str(accuracy)) #0.977

clf = RandomForestClassifier(n_estimators = 30, max_depth = 16)
clf.fit(stretched_matrix, train_label)
accuracy = clf.score(middle_test_data, test_label)
print('30 v 16 v stretched : ' + str(accuracy)) #0.8295
```