I used random_state=1 in the “train_test_split” function so the values will be the same to use them in both *k*= 1 and *k*= 3.

```
X_train, X_test, y_train, y_test = train_test_split(
    evidence, labels, test_size=TEST_SIZE, random_state=1
)
```

For *k*= 1

```
Correct: 4226
Incorrect: 706
True Positive Rate: 35.94%
True Negative Rate: 94.51%
[[3959  230]
 [ 476  267]]
Accuracy: 85.69%
```

The confusion matrix:

	Positive	Negative
Positive	3959	230
Negative	476	267

For *k*= 3

```
Correct: 4109
Incorrect: 823
True Positive Rate: 40.78%
True Negative Rate: 90.86%
[[3806  383]
 [ 440  303]]
Accuracy: 83.31%
```

The confusion matrix:

	Positive	Negative
Positive	3806	383
Negative	440	303

It is clearly that using *k-nearest-neighbor classifier* = 1 will give more accuracy than 3, which the correct predictions are more. The accuracy for k=1 is 85.69, while its 83.31 for k=3.