In [44]:
```
'''
Patrick Ballou
ID: 801130521
ECGR 4105
Homework 2
Problem 4
'''
```

Out[44]:
```
'\nPatrick Ballou\nID: 801130521\nECGR 4105\nHomework 2\nProblem 4\n'
```

In [45]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import KFold
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_breast_cancer
from sklearn.preprocessing import MinMaxScaler, StandardScaler
```

In [46]:
```
breast = load_breast_cancer()
breast_data = breast.data
breast_data.shape
breast_input = pd.DataFrame(breast_data)
```

In [47]:
```
breast_labels = breast.target
breast_labels.shape
labels = np.reshape(breast_labels,(569,1))
final_breast_data = np.concatenate([breast_data, labels],axis=1)
final_breast_data.shape
```

Out[47]:
```
(569, 31)
```

In [48]:
```
breast_dataset = pd.DataFrame(final_breast_data)
features = breast.feature_names
features_labels = np.append(features, 'label')
breast_dataset.columns = features_labels
breast_dataset.head()
```

Out[48]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	dim
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	(
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	(
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	(
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	(
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	(

5 rows × 31 columns

In [49]:
```
x = breast_dataset[features]
Y = breast_dataset['label']
```

In [50]:
```
#standard scaler is better here too
scaler = StandardScaler()
#scaler = MinMaxScaler()
X = scaler.fit_transform(x)
```

In [51]:
```
#k-fold with k = 5 and k = 10
kfold1 = KFold(n_splits=5, random_state=7, shuffle=True)
kfold2 = KFold(n_splits=10, random_state=7, shuffle=True)
model = LogisticRegression()
results_5 = cross_val_score(model, X, Y, cv=kfold1)
results_10 = cross_val_score(model, X, Y, cv=kfold2)
```

In [52]:
```
#basically the same accuracy
print("Accuracy for K = 5: %.3f%% (%.3f%%)" % (results_5.mean()*100, results_5.std()*1
print("Accuracy for K = 10: %.3f%% (%.3f%%)" % (results_10.mean()*100, results_10.std(
```

```
Accuracy for K = 5: 97.539% (1.024%)
Accuracy for K = 10: 97.716% (1.578%)
```

In [53]:
```
#4b: add penalty
kfold1 = KFold(n_splits=5, random_state=7, shuffle=True)
kfold2 = KFold(n_splits=10, random_state=7, shuffle=True)
model = LogisticRegression(penalty='l2')
results_5 = cross_val_score(model, X, Y, cv=kfold1)
results_10 = cross_val_score(model, X, Y, cv=kfold2)
```

In [54]:
```
#very similar results
print("Accuracy for K = 5: %.3f%% (%.3f%%)" % (results_5.mean()*100, results_5.std()*1
print("Accuracy for K = 10: %.3f%% (%.3f%%)" % (results_10.mean()*100, results_10.std(
```

```
Accuracy for K = 5: 97.539% (1.024%)
Accuracy for K = 10: 97.716% (1.578%)
```