In [27]:
```
'''
Patrick Ballou
ID: 801130521
ECGR 4105
Homework 2
Problem 2
'''
```

Out[27]:
```
'\nPatrick Ballou\nID: 801130521\nECGR 4105\nHomework 2\nProblem 2\n'
```

In [28]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import KFold
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import MinMaxScaler, StandardScaler
```

In [29]:
```
df = pd.read_csv("diabetes.csv")
df.head()
```

Out[29]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33

In [30]:
```
#don't need to split into train and test for k-fold
inputs = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI'

x = df[inputs]
Y = df['Outcome']
```

In [4]:
```
#standard scaler performs better here
scaler = StandardScaler()
#scaler = MinMaxScaler()
X = scaler.fit_transform(x)
```

In [25]:
```
#test model 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 [26]:
```
#5 and 10 perform about the same
```

```
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: 77.341% (1.944%)
Accuracy for K = 10: 77.346% (4.689%)
```