In [56]:

```
from scipy.stats import binom
import numpy as np
```

In [57]:

```
def incorrect_pred_prob(n, p):
    k = np.floor(n/2)
    return binom.sf(k, n, p)
```

In [58]:

```
#The ensemble contains 11 independent models, all of which have an error rate of 0.2.
ensemble_02 = incorrect_pred_prob(11,0.2)
ensemble_02
```

Out[58]:

```
0.011654205440000008
```

In [59]:

```
#The ensemble contains 11 independent models, all of which have an error rate of 0.49.
ensemble_49 = incorrect_pred_prob(11,0.49)
ensemble_49
```

Out[59]:

```
0.47294772571497473
```

In [60]:

```
#The ensemble contains 21 independent models, all of which have an error rate of 0.49.
ensemble_21_49 = incorrect_pred_prob(21,0.49)
ensemble_21_49
```

Out[60]:

```
0.4630479010127354
```

In [61]:

```
"""Using the binom library from scipy created a function to find the probability of the
```

Out[61]:

```
'Using the binom library from scipy created a function to find the probability of the en
semble being incorrect and numpy to achieve my results. It seems the probability has a g
reater impact that the number of takes if we have the same probability based off of the
size of 21 and size 11 with both having 49 percent.'
```

In [ ]: