##

File: RattanavilayThip_7_3_Calculate_Probability_of_a_Model_Ensemble

Name: Thip Rattanavilay

Date: 07/22/2021

Course: DSC 630 7.3 Assignment: Calculate Probability of a Model Ensemble

##

Calculate the probability of a model ensemble that uses simple majority voting making an incorrect prediction in the following scenarios. (Hint: Understanding how to use the binomial distribution will be useful in answering this question.)

1) The ensemble contains 11 independent models, all of which have an error rate of 0.2.

2) The ensemble contains 11 independent models, all of which have an error rate of 0.49.

3) The ensemble contains 21 independent models, all of which have an error rate of 0.49.

#1 - The ensemble contains 11 independent models, all of which have an error rate of 0.2 (20%)

In [1]:
```python
# Loading Libraries
import numpy as np
import matplotlib.pyplot as plt

#The stats submodule of the scipy module does numerous calculations in probability and statistics.
from scipy.stats import binom
```

In [2]:
```python
#define cdf distro
def binomcdf(k, n, p) :
        return 1 - binom.cdf (k, n, p)
```

In [3]:
```python
#define pmf distro
def binompmf(k, n, p) :
        return binom.pmf (k, n, p)
```

Establishing Parameters

In [4]:
```python
#set parameters 11, .2, // 2
Num_of_Models=11
Error_Rates=0.2
Num_of_Fails = Num_of_Models // 2
```

Using the CDF Function

Finds the probability that x successes or fewer occur during n trials where the probability of success on a given trial is equal to p.

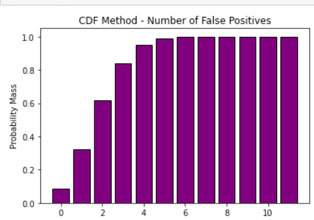
In [5]:
```python
# values of k,p,n and binomcdf will return an array consist of probability.
binomcdf(k=Num_of_Fails,
         p=Error_Rates,
         n=Num_of_Models)
```

Out[5]: 0.011654205439999954

In [6]:
```python
#Finding the R Value and assign list to R
r = list(range(Num_of_Models + 1))

# list of cdf values
dist = [binom.cdf(r,Num_of_Models,Error_Rates) for r in r ]

# plotting the graph
plt.bar(r, dist, color=['purple'], edgecolor='black')
plt.title("CDF Method - Number of False Positives")
plt.xlabel("X")
plt.ylabel("Probability Mass")
plt.show()
```



Using the PMF Function

"pmf" stands for "probability mass function"other name for the distribution of a variable that has finitely many values.

The formula for the binomial probability mass function is

P(x;p,n)=(nx)(p)x(1−p)(n−x)

In [7]:
```python
# values of k,P,N and binompmf will return an array consist of probability.
binompmf(k=Num_of_Fails,
         p=Error_Rates,
         n=Num_of_Models)
```

Out[7]: 0.038755368959999995

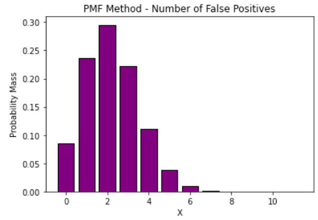
For an ensemble with 11 models with an error rate of 20% per model, the probability of getting PMF misclassifications is 3.8755%

In [8]:
```python
#Finding the R Value and assign list to R
r = list(range(Num_of_Models + 1))

#pmf values
dist = [binom.pmf(r,Num_of_Models,Error_Rates) for r in r ]

#plotting graph
plt.bar(r, dist, color=['purple'], edgecolor='black')
plt.title("PMF Method - Number of False Positives")
plt.xlabel("X")
plt.ylabel("Probability Mass")
```

```
plt.show()
```



Results:

For an ensemble with 11 models with an error rate of 20% per model, the probability of getting misclassifications is 1.1654%

#2 - The ensemble contains 11 independent models, all of which have an error rate of 0.49 (49%)

Establishing Parameters

In [9]:
```
#set parameters 11,.49,/ 2
Num_of_Models=11
Error_Rates=0.49
Num_of_Fails = np.ceil(Num_of_Models / 2)
```

Using the CDF Function

Finds the probability that x successes or fewer occur during n trials where the probability of success on a given trial is equal to p.

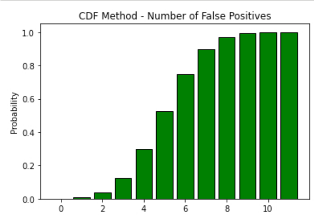
In [10]:
```
# values of k,P,N and binomcdf will return an array consist of probability.
binomcdf(k=Num_of_Fails-1,
         p=Error_Rates,
         n=Num_of_Models)
```

Out[10]: 0.4729477257149748

In [11]:
```
#Finding the R Value and assign list to R
r = list(range(Num_of_Models + 1))

# cdf values
dist = [binom.cdf(r,Num_of_Models,Error_Rates) for r in r ]

# plotting graph
plt.bar(r, dist, color=['green'], edgecolor='black')
plt.title("CDF Method - Number of False Positives")
plt.xlabel("X")
plt.ylabel("Probability")
plt.show()
```



Using the PMF Function

"pmf" stands for "probability mass function"other name for the distribution of a variable that has finitely many values.

The formula for the binomial probability mass function is

P(x;p,n)=(nx)(p)x(1−p)(n−x)

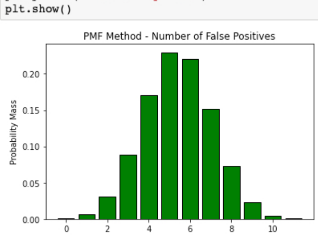
In [12]:
```
# values of k,p,n and binompmf will return an array consist of probability.
binompmf(k=Num_of_Fails-1,
         p=Error_Rates,
         n=Num_of_Models)
```

Out[12]: 0.2296378289465168

In [13]:
```
#Finding the R Value and assign list to R
r = list(range(Num_of_Models + 1))

# pmf values
dist = [binom.pmf(r,Num_of_Models,Error_Rates) for r in r ]

# plotting graph
plt.bar(r, dist, color=['green'], edgecolor='black')
plt.title("PMF Method - Number of False Positives")
plt.xlabel("X")
plt.ylabel("Probability Mass")
plt.show()
```



Results:

For an ensemble with 11 models with an error rate of 49% per model, the probability of getting misclassifications is 47.2948%

#3 - The ensemble contains 21 independent models, all of which have an error rate of 0.49

```
In [14]: #set parameters 21,.49, //2
Num_of_Models=21
Error_Rates=0.49
Num_of_Fails = Num_of_Models // 2
```

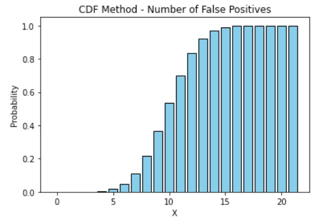
```
In [15]: # values of k,p,n and binomcdf will return an array consist of probability.
binomcdf(k=Num_of_Fails,
         p=Error_Rates,
         n=Num_of_Models)
```

```
Out[15]: 0.4630479010127354
```

```
In [16]: #Finding the R Value and assign list to R
r = list(range(Num_of_Models + 1))

# list of cdf values
dist = [binom.cdf(r,Num_of_Models,Error_Rates) for r in r ]

# plotting the graph
plt.bar(r, dist, color=['skyblue'], edgecolor='black')
plt.title("CDF Method - Number of False Positives")
plt.xlabel("X")
plt.ylabel("Probability")
plt.show()
```



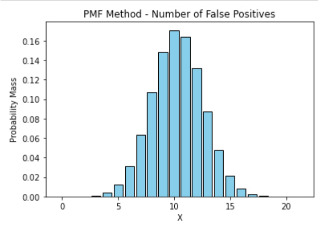

```
In [17]: # values of k,P,N and binompmf will return an array consist of probability.
binompmf(k=Num_of_Fails,
         p=Error_Rates,
         n=Num_of_Models)
```

```
Out[17]: 0.17086688342342418
```

```
In [18]: #Finding the R Value and assign list to R
r = list(range(Num_of_Models + 1))

#pmf values
dist = [binom.pmf(r,Num_of_Models,Error_Rates) for r in r ]

#plotting graph
plt.bar(r, dist, color=['skyblue'], edgecolor='black')
plt.title("PMF Method - Number of False Positives")
plt.xlabel("X")
plt.ylabel("Probability Mass")
plt.show()
```



Results:

For an ensemble with 21 models with an error rate of 49% per model, the probability of getting misclassifications is 46.3047%

Reference: https://www.itl.nist.gov/div898/handbook/eda/section3/eda366i.htm

```
In [ ]:
```