Author: Ganesh Kale

Week#9

Date: Feb 10, 2021

Calculate Probability of Model Ensemble:

In [1]:
```
# Import required packages

import numpy as np  # for numeric operations
import pandas as pd  # for data manipulation
import matplotlib.pyplot as plt  # for data visualization
import seaborn as sns  # for data visualization
from scipy.stats import binom # for binom distribution


from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = 'all'

import warnings
warnings.filterwarnings('ignore')
```

What is Model Ensemble:

Model Ensemble is a machine learning technique that combines several base models in order to produce one optimal predictive model. A voting ensemble is an ensemble machine learning model that combines the predictions from multiple other models.

A voting ensemble can be used for classification or regression. In the case of classification, the predictions for each label are summed and the label with the majority vote is predicted.

There are two approaches to the majority vote prediction for classification; they are hard voting and soft voting.

- Hard Voting: Predict the class with the largest sum of votes from models
- Soft Voting: Predict the class with the largest summed probability from models

In this assignment the majority voting i.e. Hard Voting is used to predict incorrect outcome/prediction.

Scenario#1 - The ensemble contains 11 independent models, all of which have an error rate of 0.2

In this scenario, there are 11 models used in the ensemble for prediction and the error rate or failure is 0.2 i.e. 20% we need to calculate the probability of ensemble to predict incorrect prediction.

Here, binomial probability formula is used from scipy stats module.

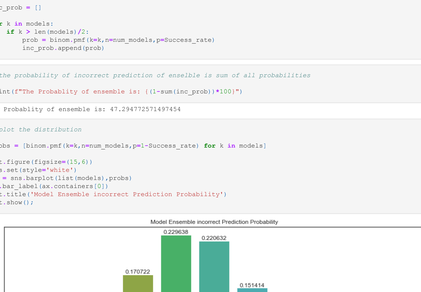
Since the voting used here is Hard voting, so 6 and/or more(up to 11) models predicting incorrect prediction are considered calculating probability.

In [2]:
```
# calc probability of enseble

num_models = 11   # number of models in ensemble

models = range(num_models,0,-1)

Error_rate = 0.2    # indv model error/failure rate

Success_rate = 1-0.2    # prob of success = 1- prob of failure/error

inc_prob = []

for k in models:
    if k > len(models)/2:
        prob = binom.pmf(k=k,n=num_models,p=Success_rate)
        inc_prob.append(prob)
```

In [3]:
```
# the probability of incorrect prediction of enselble is sum of all probabilities

print(f"The Probability of ensemble is: {(1-sum(inc_prob))*100}")
```

```
The Probability of ensemble is: 1.1654205439999954
```

In [4]:
```
# plot the distribution

probs = [binom.pmf(k=k,n=num_models,p=1-Success_rate) for k in models]

plt.figure(figsize=(15,6))
sns.set(style='white')
ax = sns.barplot(list(models),probs)
ax.bar_label(ax.containers[0])
plt.title('Model Ensemble incorrect Prediction Probability')
plt.show();
```



Insights:

The above bar chart shows the incorrect prediction of combination of 11 models, combination of 6 models or more are very low (almost 0) considering the failure rate.

The probability of model ensemble to predict incorrect prediction would be: 1.16%

This means the success rate of model ensemble would be: 98.84%

Which is quite better than individual models(80%).

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

In this scenario, there are 11 models used in the ensemble for prediction and the error rate or failure is 0.49 i.e. 49% we need to calculate the probability of ensemble to predict incorrect prediction.

Here, binomial probability formula is used from scipy stats module.

Since the voting used here is Hard voting, so 6 and/or more(up to 11) models predicting incorrect prediction are considered for calculating probability.

In [5]:
```
# calc probability of enseble

num_models = 11   # number of models in ensemble

models = range(num_models,0,-1)

Error_rate = 0.49    # indv model error/failure rate

Success_rate = 1-0.49   # prob of success = 1- prob of failure/error

inc_prob = []

for k in models:
    if k > len(models)/2:
        prob = binom.pmf(k=k,n=num_models,p=Success_rate)
        inc_prob.append(prob)
```

In [6]:
```
# the probability of incorrect prediction of enselble is sum of all probabilities

print(f"The Probability of ensemble is: {(1-sum(inc_prob))*100}")
```

```
The Probability of ensemble is: 47.29477257149745 4
```

In [7]:
```
# plot the distribution

probs = [binom.pmf(k=k,n=num_models,p=1-Success_rate) for k in models]

plt.figure(figsize=(15,6))
sns.set(style='white')
ax = sns.barplot(list(models),probs)
ax.bar_label(ax.containers[0])
plt.title('Model Ensemble incorrect Prediction Probability')
plt.show();
```

Insights:

The above bar chart shows the probability of incorrect prediction of 11 models, The out of 11 models, 6 models predicting incorrect prediction is 22%, so on when we add up all the probabilities from 6 to 11 , combined shows the probability of model ensemble, which is 47.29%

This means the success rate of model enseble would be: 52.71%

Which is greater than individual models(51%).

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

In this scenario, there are 21 models used in the ensemble for prediction and the error rate or failure is 0.49 i.e. 49% we need to calculate the probability of ensemble to predict incorrect prediction.

Here, binomial probability formula is used from scipy stats module.

Since the voting used here is Hard voting, so 11 and/or more(up to 21) models predicting incorrect prediction are considered for calculating probability.

In [8]:
```
# calc probability of enseble

num_models = 21   # number of models in ensemble

models = range(num_models,0,-1)

Error_rate = 0.49    # indv model error/failure rate

Success_rate = 1-0.49   # prob of success = 1- prob of failure/error

inc_prob = []

for k in models:
    if k > len(models)/2:
        prob = binom.pmf(k=k,n=num_models,p=Success_rate)
        inc_prob.append(prob)
```

In [9]:
```
# the probability of incorrect prediction of enselble is sum of all probabilities

print(f"The Probability of ensemble is: {(1-sum(inc_prob))*100}")
```

```
The Probability of ensemble is: 46.30479010127353
```

In [10]:
```
# plot the distribution

probs = [binom.pmf(k=k,n=num_models,p=1-Success_rate) for k in models]

plt.figure(figsize=(19,6))
sns.set(style='white')
ax = sns.barplot(list(models),probs)
ax.bar_label(ax.containers[0])
plt.title('Model Ensemble incorrect Prediction Probability')
plt.show();
```


Insights:

The above bar chart shows the probability of incorrect prediction of 21 models, The out of 21 models, 11 models predicting incorrect prediction is 16%, so on, when we add up all the probabilities from 11 to 21 , combined shows the probability of model ensemble, which is 46.31%

In this scenario, the number of models are increased and we observed that the incorrect prediction rate got reduced, we can say that in model ensemble if we increase the models, there are chances of reducing the incorrect prediction which means increaseing the accuracy.

This means the success rate of model enseble would be: 53.69%

Which is greater than individual models(51%).

END
