```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LOLTeamCounterPick.Classes
{
    class ChampionData
    {
        public string name;
        public double votedYesTotalCount;
        public List<ChampionDataRecord> Records;

        public ChampionData(string str,int mode)
        {
            switch (mode)
            {
                case 0:
                    name = str;
                    Records = new List<ChampionDataRecord>();
                    break;
                case 1:
                    //change this part if the str format changed!
                    Records = new List<ChampionDataRecord>();
                    votedYesTotalCount = 0;
                    name = GFs.parseElement(str, "<title>", " counters");
                    name = GFs.originChampionName(name);
                    List<string> names = GFs.parseElements(str, "How to counter ", string.Format(" as {0}", ↙
    name));
                    List<string> votedYesStr = GFs.parseElements(str, "<div class='uvote tag_green'><img src ↙
    =\"/resources/img/up.png\" />", "</div></a>");
                    List<string> votedNoStr = GFs.parseElements(str, "<div class='dvote tag_red'><img src=\ ↙
    "/resources/img/down.png\" />", "</div></a>");

                    if (votedNoStr.Count == votedYesStr.Count && votedYesStr.Count == names.Count)
                    {
                        for (int i = 0; i < votedYesStr.Count; i++)
                        {
                            int votedYes = Convert.ToInt32(votedYesStr[i].Replace(",", ""));
                            int votedNo = Convert.ToInt32(votedNoStr[i].Replace(",", ""));
                            if (i == votedYesStr.Count - 1) { addRecord(names[i], votedYes, votedNo, true); ↙
     }
                            else { addRecord(names[i], votedYes, votedNo, false); }
                        }
                    }
                    else { throw new Exception("Error: Picked up different amount for names, votedYes, and ↙
    votedNo!"); }
                    break;
                default:
                    throw new Exception("Wrong mode used in code! Contact Developers Please!");
            }
        }
        private void updateAllRecords()
        {
            for (int i = 0; i < Records.Count; i++)
            {
                for (int j = i + 1; j < Records.Count; j++)
                {
                    if (Records[j].name == Records[i].name)
                    {
                        Records[i].votedYes += Records[j].votedYes;
                        Records[i].votedNo += Records[j].votedNo;
                        Records[i].correctness = Records[i].votedYes / (Records[i].votedYes + Records[i]. ↙
    votedNo);
                        Records.RemoveAt(j);
                    }
                }
                Records[i].supportness = Records[i].votedYes / votedYesTotalCount;
                Records[i].value = Records[i].supportness * Records[i].correctness;


            }
        }
```

```
        public void addRecord(string _name, int _votedYes, int _votedNo, bool isDoneAdding)
        {
            Records.Add(new ChampionDataRecord(_name, _votedYes, _votedNo));
            votedYesTotalCount += _votedYes;
            if (isDoneAdding) { updateAllRecords(); }
        }

    }
    class ChampionDataRecord
    {
        public string name;
        public double votedYes;
        public double votedNo;
        public double correctness;
        public double supportness;
        public double value;
        public ChampionDataRecord(string _name, int _votedYes, int _votedNo)
        {
            name = _name;
            votedYes = _votedYes;
            votedNo = _votedNo;
            correctness = votedYes / (votedYes + votedNo);
        }
        public ChampionDataRecord(string _name, double _value)
        {
            name = _name;
            value = _value;
        }
    }
}
```