Inductive Learning from Imbalanced Data Sets

Standard Assumption

- The data sets are balanced: i.e., there are as many positive examples of the concept as there are negative ones.
- ***Example:*** Our database of sick and healthy patients contains as many examples of sick patients as it does of healthy ones.

The Standard Assumption is not Always Correct

- There exist many domains that do not have a balanced data set.
- ***Examples:***
  - Helicopter Gearbox Fault Monitoring
  - Discrimination between Earthquakes and Nuclear Explosions
  - Document Filtering
  - Detection of Oil Spills
  - Detection of Fraudulent Telephone Calls

But What is the Problem?

- Standard learners are often biased towards the majority class.
- That is because these classifiers attempt to reduce global quantities such as the error rate, not taking the data distribution into consideration.
- As a result examples from the overwhelming class are well-classified whereas examples from the minority class tend to be misclassified.

Significance of the problem for Machine Learners/Data Miners

- For the past 16 years, there has been a lot of research on the problem.
- There are excellent review articles on the subject as well as comparative studies.
- Some of the dominant approaches are SMOTE (and its many variants) and bagged random undersampling, but there are many others as well.
- The problem occurs in many domains.

Several Common Approaches

- At the data Level: Re-Sampling
  - Oversampling (Random or Directed)
  - Undersampling (Random or Directed)
  - SMOTE
  - One class learning (ignore the small class altogether)
- At the Algorithmic Level:
  - Adjusting the Costs
  - Adjusting the decision threshold / probabilistic estimate at the tree leaf

Analysis and Approaches

Fundamental

- What domain characteristics aggravate the problem?
- Class imbalances or small disjuncts?
- Are all classifiers sensitive to class imbalances?
- Which proposed solutions to the class imbalance problem are more appropriate?

Some Approaches

- SMOTE
- Specialized Resampling: within-class versus between-class imbalances
- One class versus two-class learning
- Multiple Resampling

Part I: Fundamentals

I. What domain characteristics aggravate the problem?
II. Class Imbalances or Small Disjuncts?
III. Are all classifiers sensitive to class imbalances?
IV. Which proposed solutions to the class imbalance problem are more appropriate?

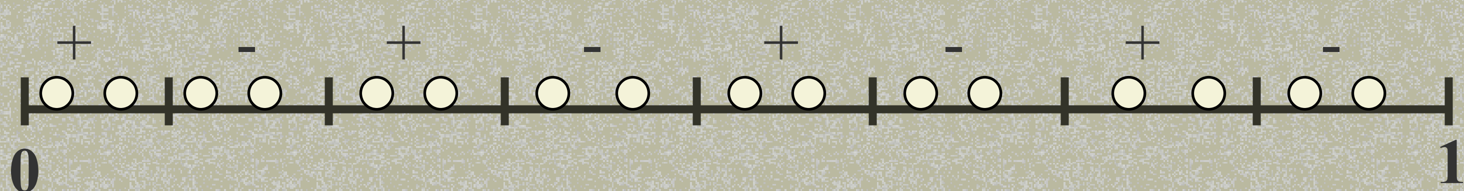
I. I What domain characteristics aggravate the Problem?

To answer this question, I generated artificial domains that vary along three different axes:

- The degree of ***concept complexity***

- The ***size*** of the training set

- The degree of ***imbalance*** between the two classes.

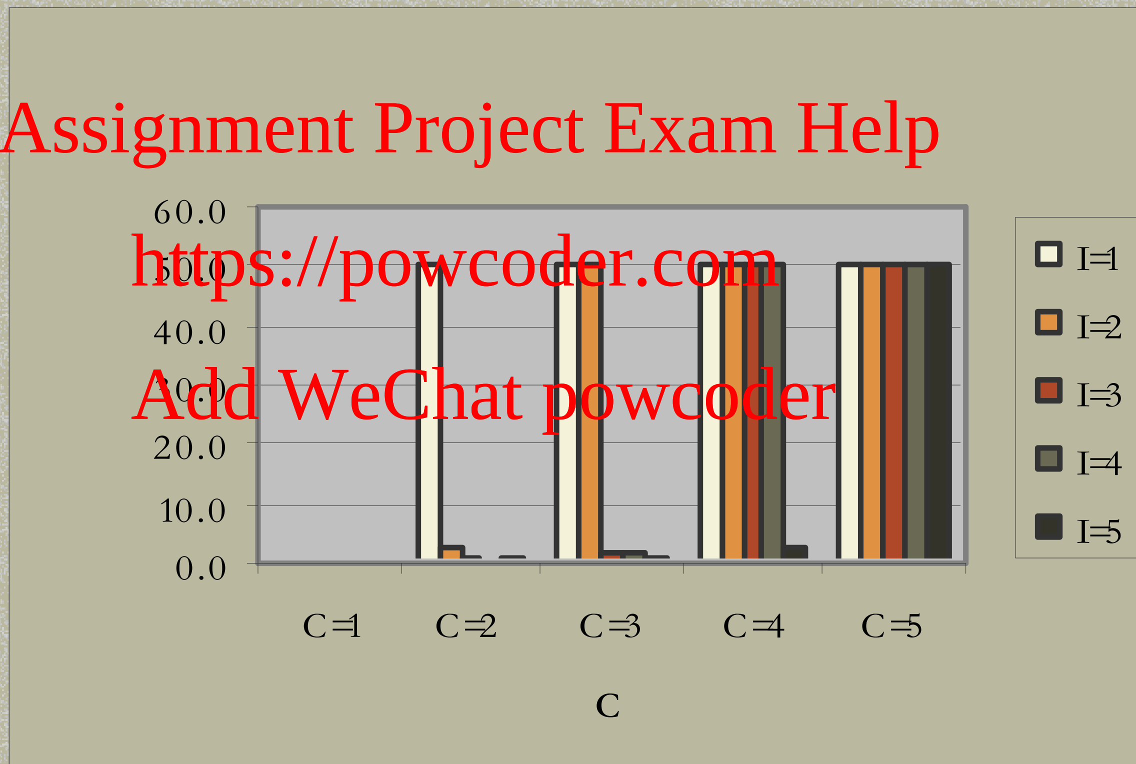


I. I What domain characteristics aggravate the Problem?

- I created 125 domains, each representing a different type of class imbalance, by varying the concept complexity (C), the size of the training set (S) and the degree of imbalance (I) at different rates (5 settings were used per domain characteristics).
- I ran C5.0 [a decision tree learning algorithm] on these various imbalanced domains and plotted its error rate on each domain.
- Each experiment was repeated 5 times and the results averaged.

I. I What domain characteristics aggravate the Problem?

Error rate



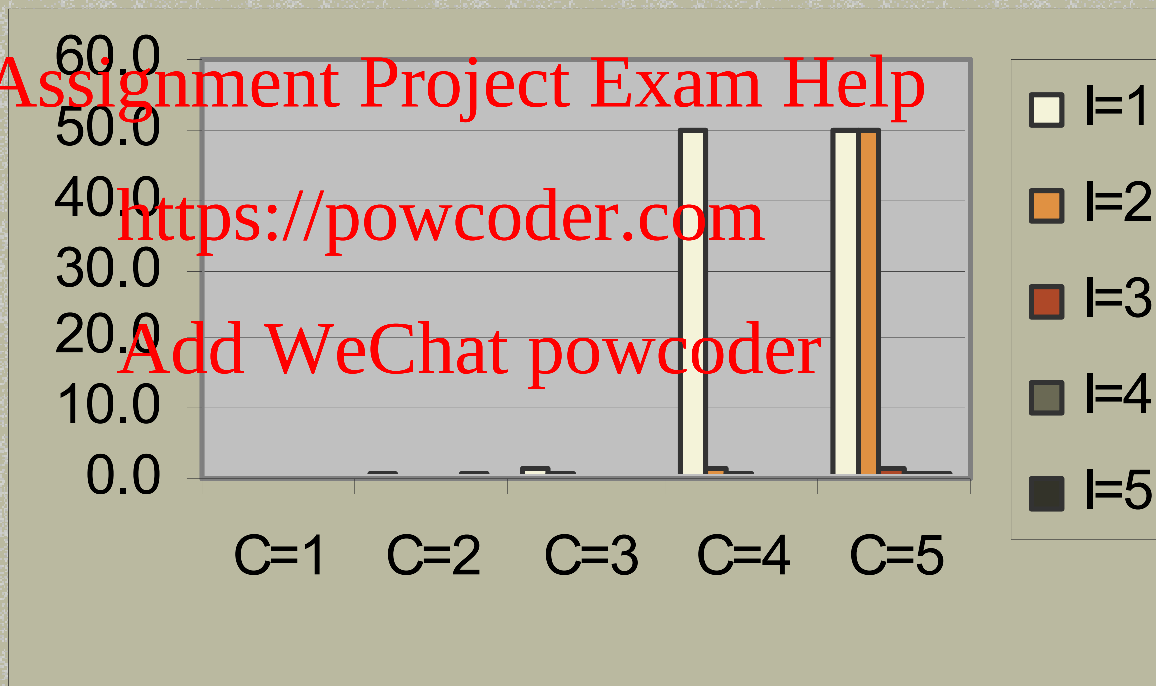
Large Imbal.

Full balance

S= 1

I.I What domain characteristics aggravate the Problem?

Error rate



Large Imbal.

Full balance

S= 5

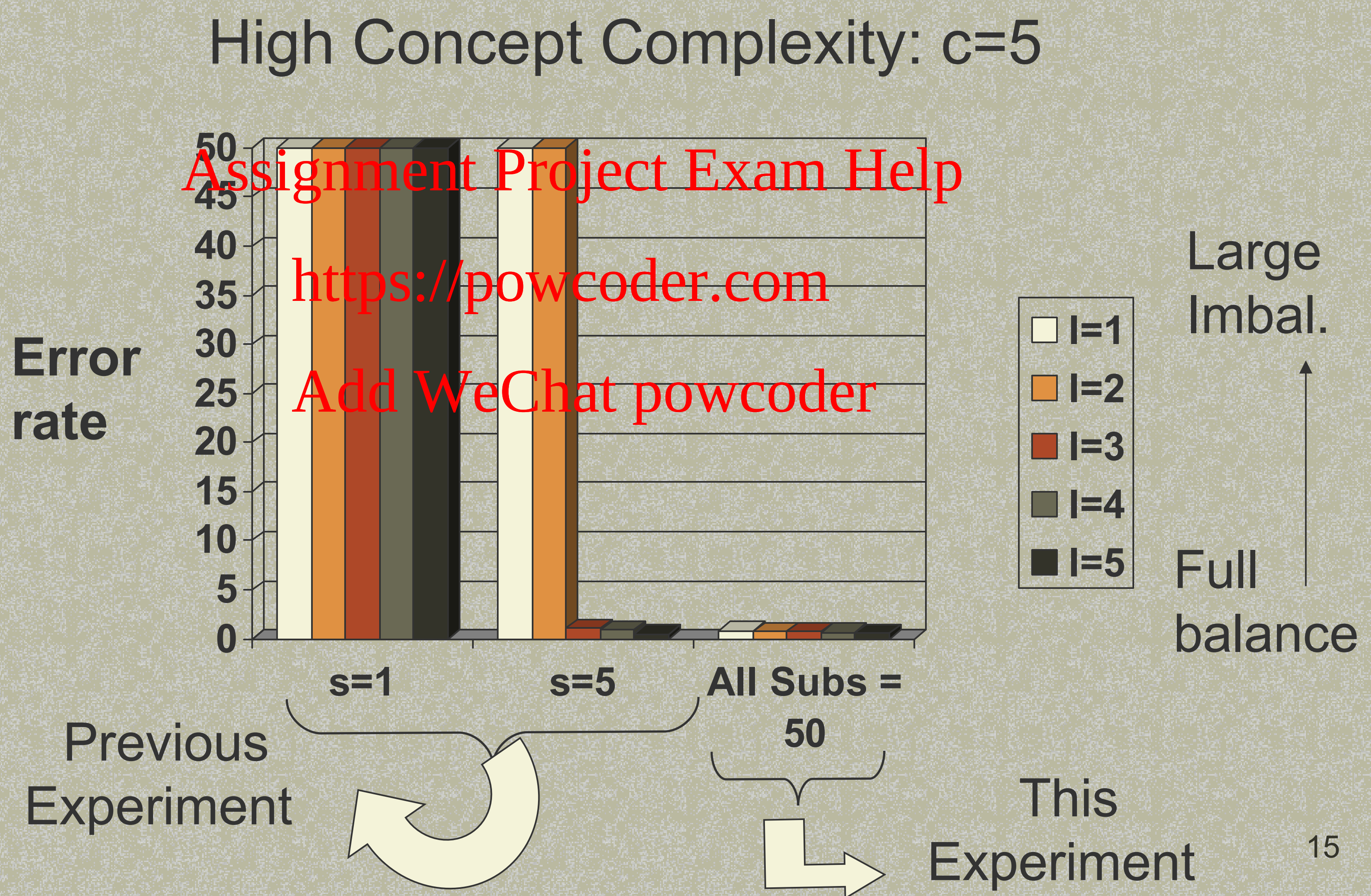
I. I What domain characteristics aggravate the Problem?

- The problem is aggravated by two factors:
  - An increase in the degree of class imbalance
  - An increase in problem complexity – class imbalances do not hinder the classification o simple problems (e.g., linearly separable ones)
- However, the problem is simultaneously mitigate by one factor:
  - The size of the training set – large training sets yield low sensitivity to class imbalances

I.II: Clas Imbalances or Small Disjuncts?

- Studying the training sets from the previous experiments, it can be inferred that when i and c are large, and s, small, the domain contains many very small subclusters.
- These were also the conditions under which C5.0 performed the worst.
- To test whether it is these small subclusters that cause performance degradation, we disregarded the value of s and set the size of all subclusters to 50 examples.

I.II: Clas Imbalances or Small Disjuncts?



I.II: Class Imbalances or Small Disjuncts?

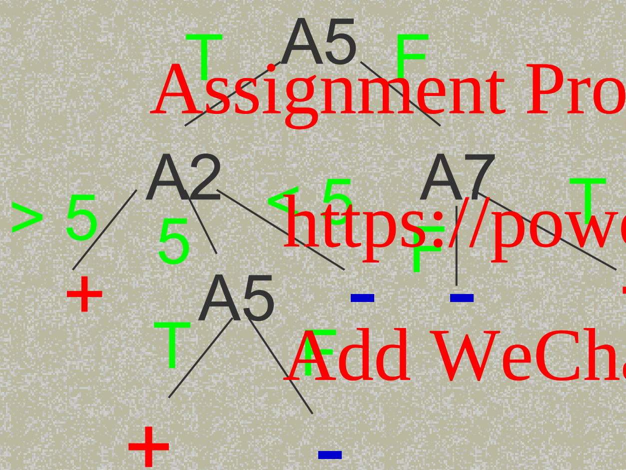
- When all the subclusters are of size 50, even at the highest degree of concept complexity, no matter what the class imbalance is, the error is below 1% → **it is negligible.**
- This suggests that it is not the class imbalance per se that causes a performance decrease, but rather, that it is the small disjunct problem created by the class imbalance (in highly complex and small-sized domains) that cause that loss of performance.

I.III Class Overlap

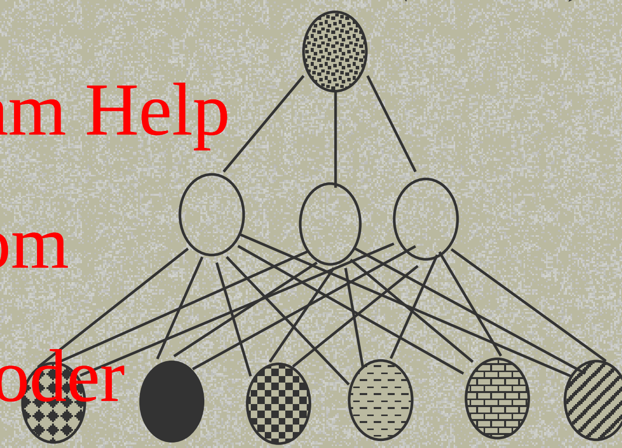
- Another factor that was studied by [Batista et al.] along with the other ones is the question of class overlap.
- They showed that as the class overlap increases, classifiers become more and more sensitive to the class imbalance problem.
- Overlap, they showed, is a very significant factor that cannot be overlooked since it is present in most real-world domains.

I.IV Are all classifiers sensitive to class imbalances?

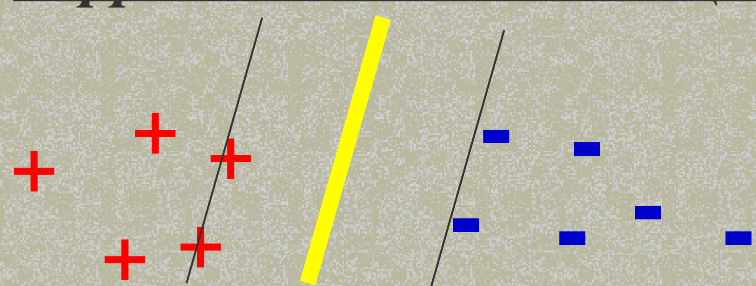
Decision Tree (C5.0)



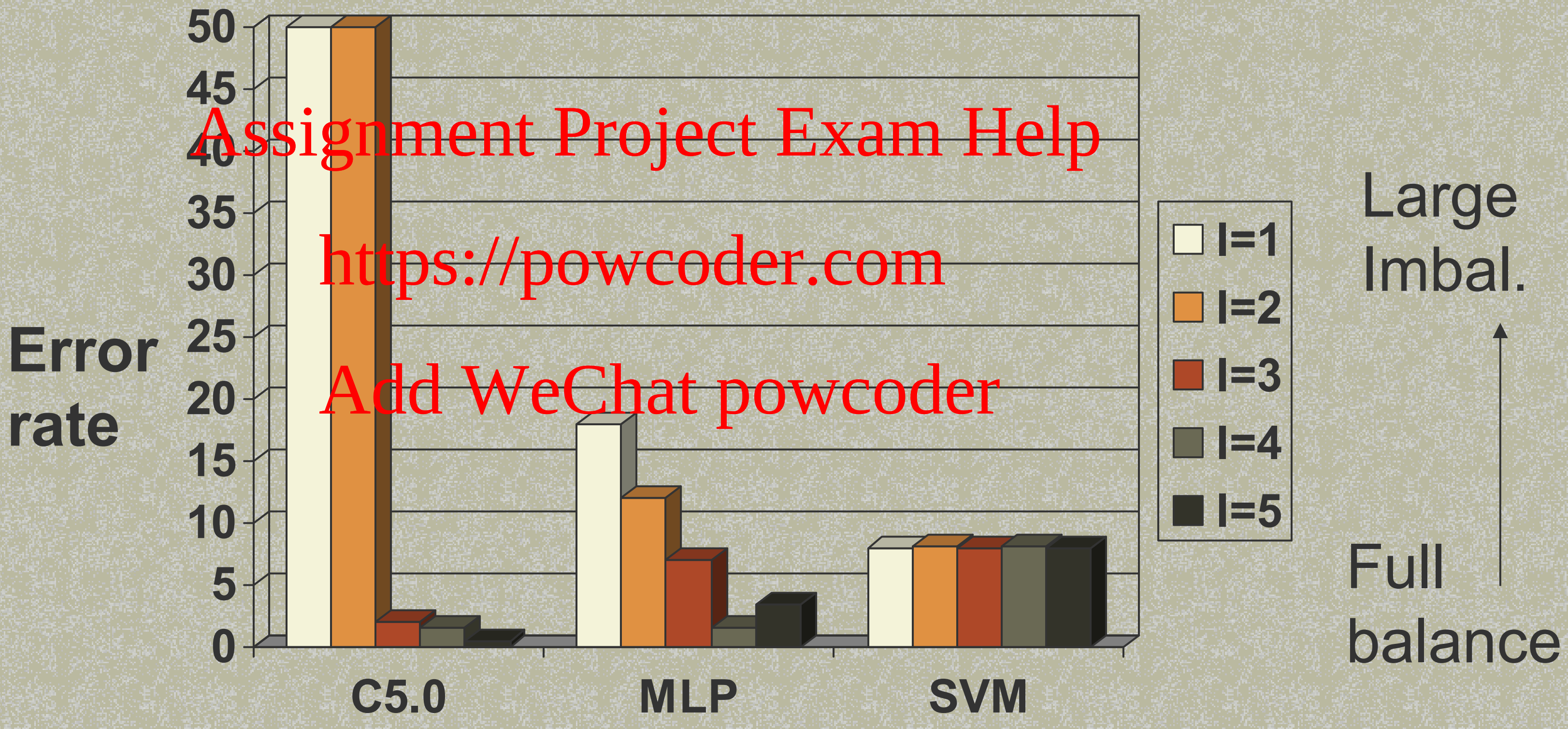
Neural Net (MLPs)



Support Vector Machines (SVMs)



I.III Are all classifiers sensitive to class imbalances?



S = 1; C = 3

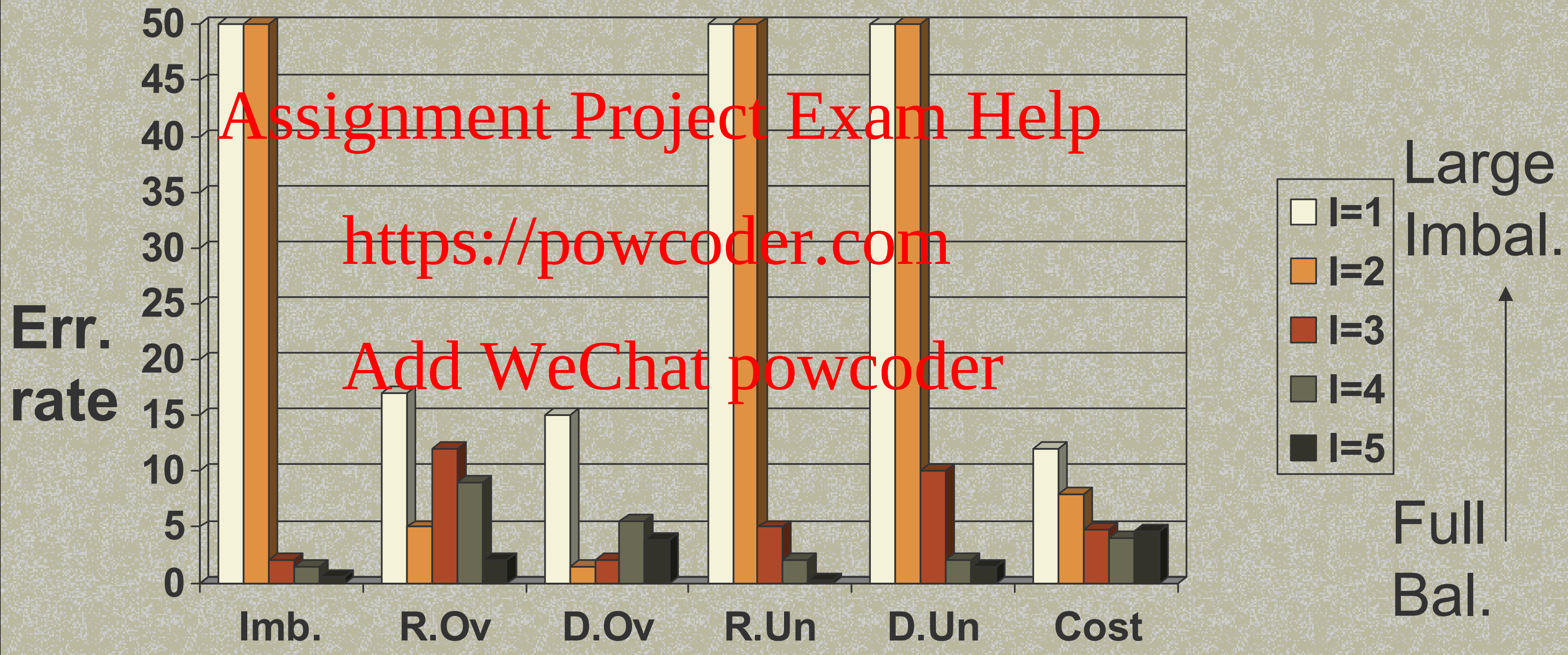
I.III Are all classifiers sensitive to class imbalances?

- **Decision Tree (C5.0)** C5.0 is the most sensitive to class imbalances. This is because C5.0 works globally, not paying attention to specific data points.
- **Multi-Layer perceptrons (MLPs)** MLPs are less prone to the class imbalance problem than C5.0. This is because of their flexibility: their solution gets adjusted by each data point in a bottom-up manner as well as by the overall data set in a top-down manner.
- **Support Vector Machines (SVMs)** SVMs are even less prone to the class imbalance problem than MLPs because they are only concerned with a few support vectors, the data points located close to the boundaries.

I.IV Which Solution is Best?

- Random Oversampling
- Directed Oversampling
- Random Undersampling
- Directed Undersampling
- Adjusting the Costs

I.IV Which Solution is Best?



S = 1; C = 3

I.IV Which Solution is Best?

- Three of the five methods considered present an improvement over C5.0 at S=1 and C=3: Random oversampling, Directed oversampling and Cost-modifying.
- Undersampling (random and directed) is not effective and can even hurt the performance.
- Random oversampling helps quite dramatically at all complexity. Directed oversampling makes a bit of a difference by helping slightly more.
- On the graph of the previous slide, Cost-adjusting is about as effective as Directed oversampling. Generally, however, it is found to be slightly more useful.

I.IV Which Solution is Best?

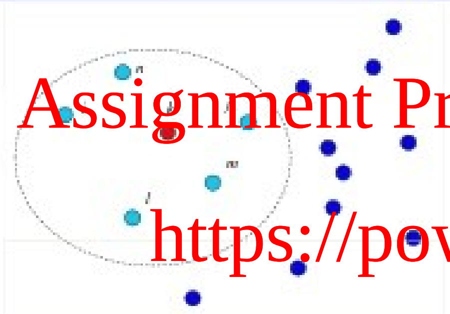
- However, note that:
- The results obtained using random oversampling (directed or not) have been discarded in later research (See Drummond and Holte, 2003: http://www.site.uottawa.ca/~nat/Workshop2003/drummondc.pdf) on more general domains.
- On more general domains, it was shown that random oversampling has a tendency to cause classifiers to overfit the data.
- When simple approaches are sought, which do not generate artificial data, it is now believed that random undersampling is preferable.

Part II:More sophisticated approaches

I. SMOTE (Chawla et al., 2002)
II. Specialized Resampling: within-class versus between-class imbalances
III. One class versus two-class learning
IV. Multiple Resampling

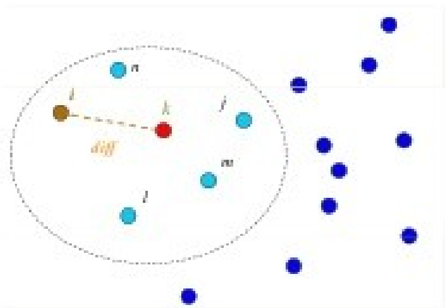
II.I SMOTE

SMOTE, R package [16]

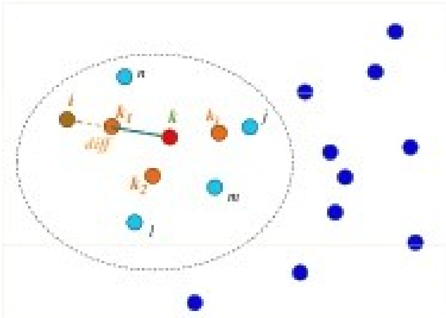


1. For each minority example k compute nearest minority class examples (i, j, l, n, m)

3. Synthetically generate event k_1, such that k_1 lies between k and i



2. Randomly choose an example out of 5 closest points



4. Dataset after applying SMOTE 3 times

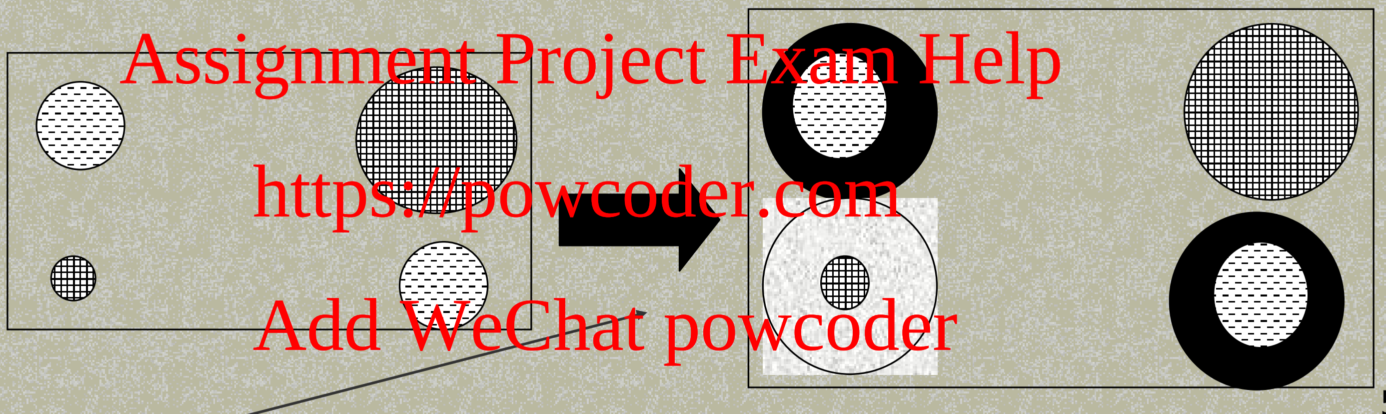
II.II: Within-class versus Between-class Imbalances

Idea:

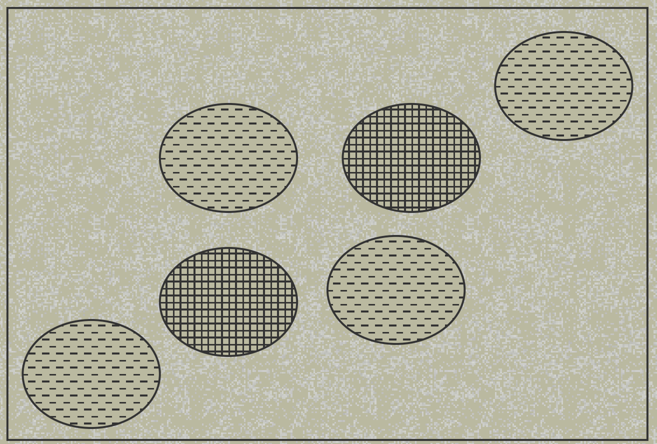
- Use unsupervised learning to identify subclusters in each class separately.

- Re-sample the subclusters of each class until no within-class imbalance and no between-class imbalance are present (although the subclusters of each class can have different sizes)

II.II: Within-class versus Between-class Imbalances



Symmetric Case



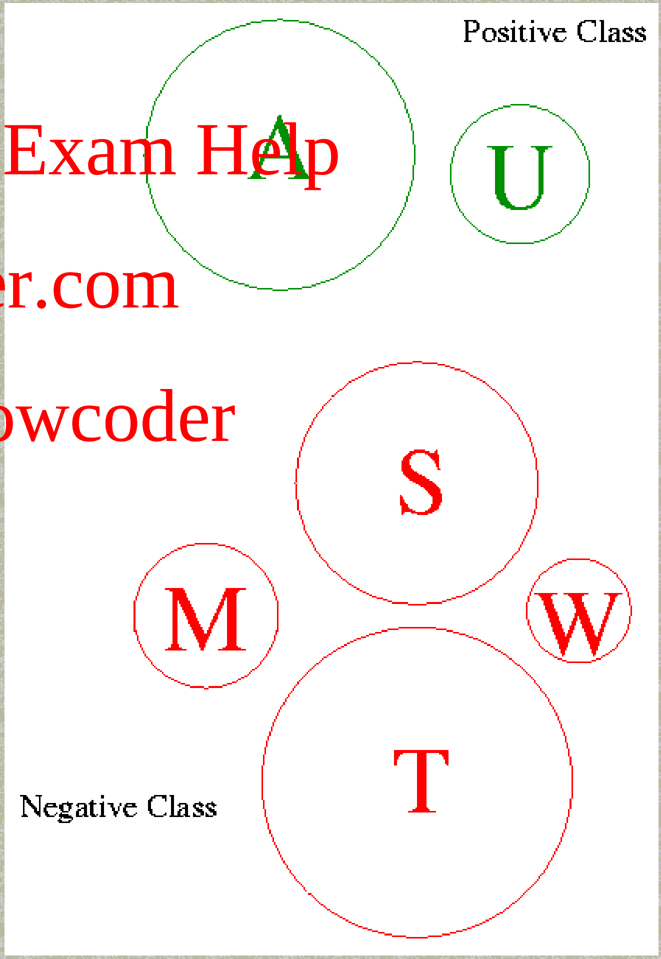
Asymmetric Case

II.II: Within-class vs Between- class Imbalances: Experiments

- Imbalances
- Random Oversampling
  - Between class imbalance eliminated
- Guided Oversampling I (# Clusters Known)
  - Use prior knowledge of classes to guide clustering
- Guided Oversampling II (# Clusters Unknown)
  - Let clustering algorithm determine the number of clusters

II.II: Within-class vs Between- class Imbalances: Letters

- Subset of the *Letters* dataset found at the UCI Repository
- Positive class contains the vowels *a* and *u*
- Negative class contains the consonants *m, s, t* and *w*.
- All letters are distributed according to their frequency in English texts.

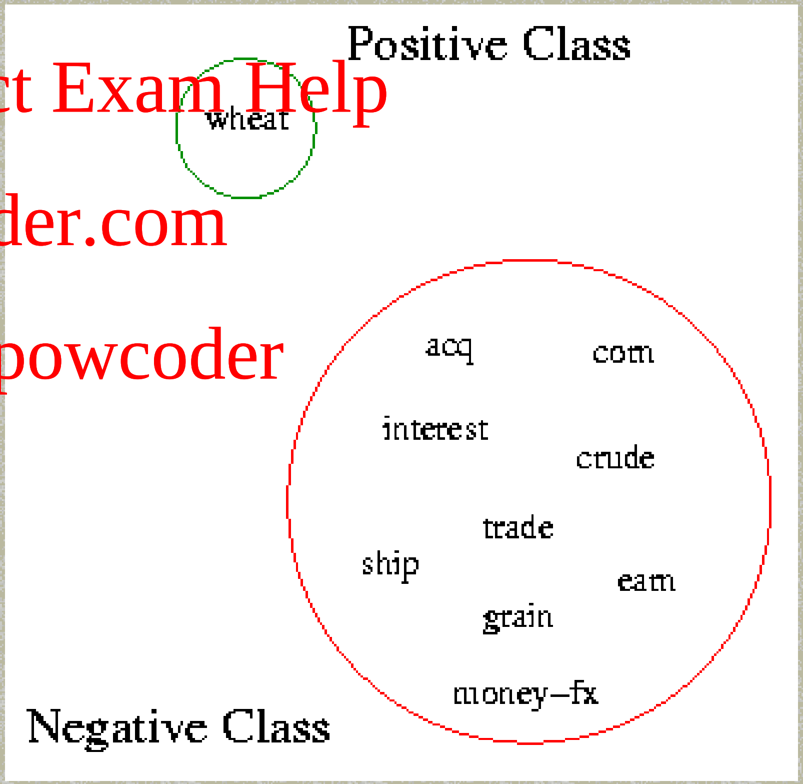


II.II: Within-class vs Between- class Imbalances: Letters

Method	Precision	Recall	F-Measure
Imbalanced	0.905	0.818	0.859
Random Oversampling	0.905	0.818	0.859
Guided Oversampling I (# Clusters Unknown)	0.923	0.914	0.919
Guided Oversampling II (Using Known Clusters)	0.935	0.877	0.905

II.I:I Within-class vs Between- class Imbalances: Text Classification

- Reuters-21578 Dataset
- Classifying a document according to its topic
- Positive class is a particular topic
- Negative class is every other topic



II.II: Within-class vs Between- class Imbalances: Text Classification

Method	Precision	Recall	F-Measure
Imbalanced	0.617	0.394	0.455
Random Oversampling	0.580	0.545	0.560
Guided Oversampling I (# Clusters Unknown)	0.650	0.510	0.544
Guided OversamplingII (Using Known Clusters)	0.601	0.751	0.665

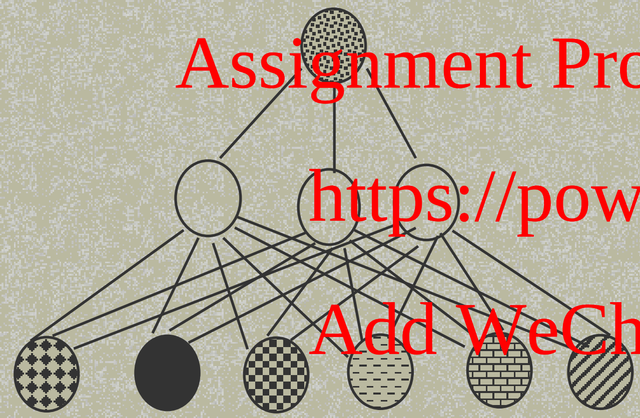
II.II: Within-class versus Between-class Imbalances

Results:

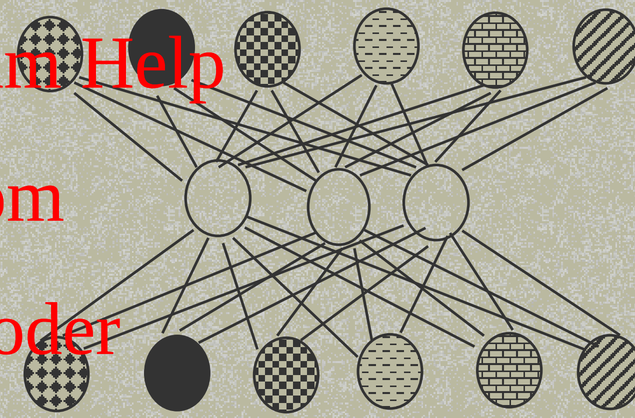
- On letter and text categorization tasks, this strategy worked better than the random over-sampling strategy.
- Noise in the small subclusters, however, caused problems since it got too magnified.
- This promising strategy requires more study..

II.III One-Class versus Two-Class Learning

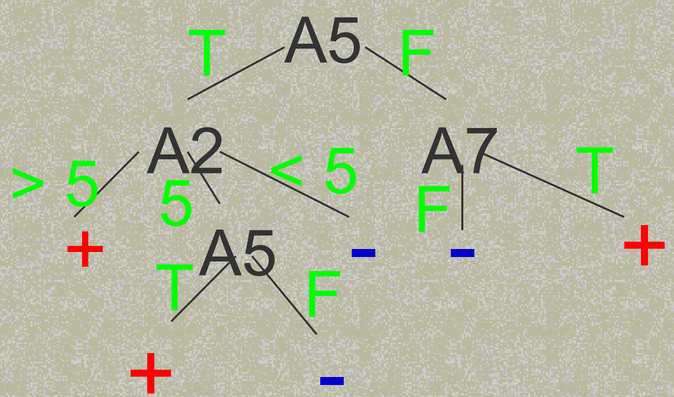
DMLP



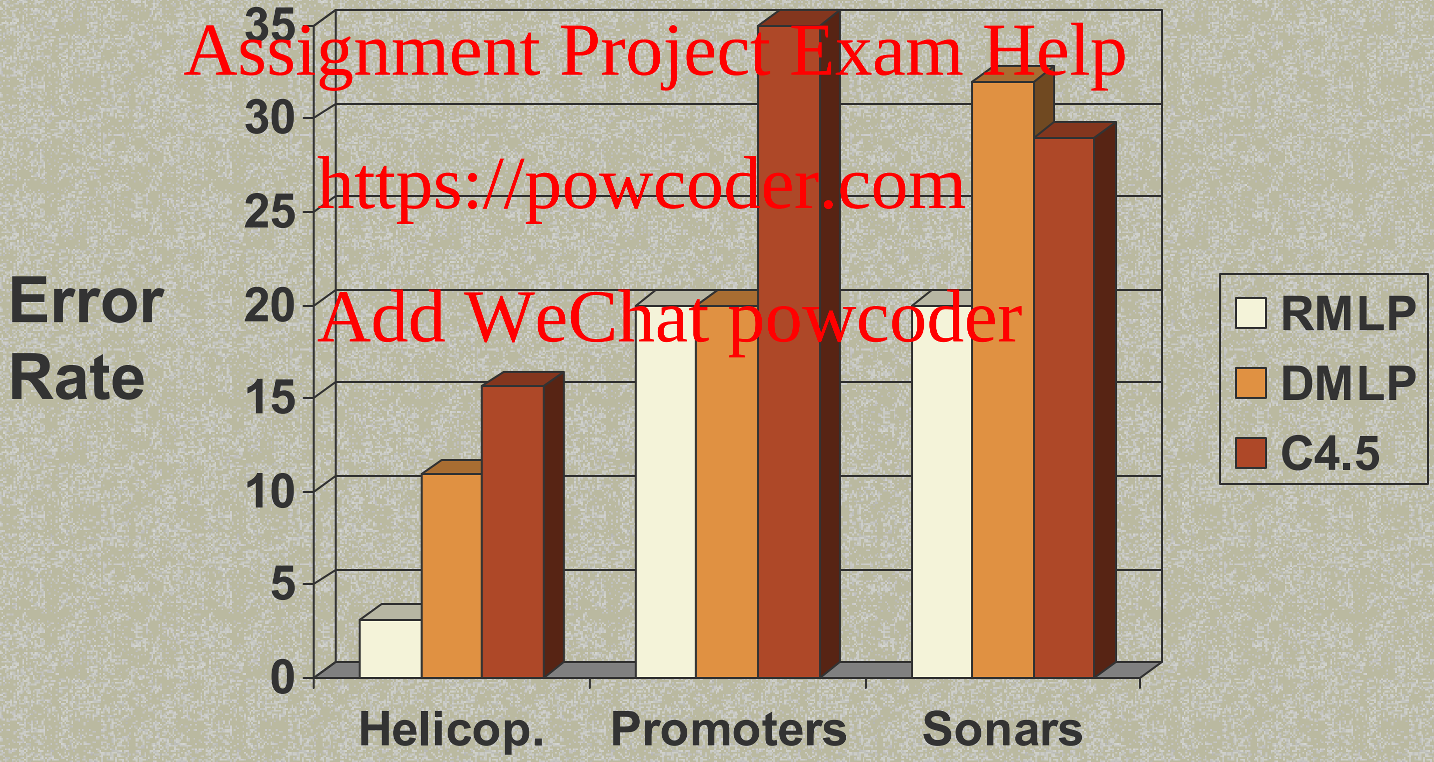
RMLP



Decision Tree (C5.0)



II.III One-Class versus Two-Class Learning



II.III One-Class versus Two-Class Learning

- One-Class learning is more accurate than two-class learning on two of our three domains considered and as accurate on the third.
- It can thus be quite useful in class imbalanced situations.
- Further comparisons with other proposed methods are required.

II.IV Multiple Resampling

<u>**Idea:**</u>

- Although the results reported here suggest that undersampling is not as useful as oversampling, other studies of ours and others (on different data sets) suggest that it can be ➔ It shouldn't be abandoned
- Further experiments of ours (not reported here) suggest that rather than oversampling or undersampling until a full balance is achieved may not always be optimal ➔ A different re-sampling rate should be used

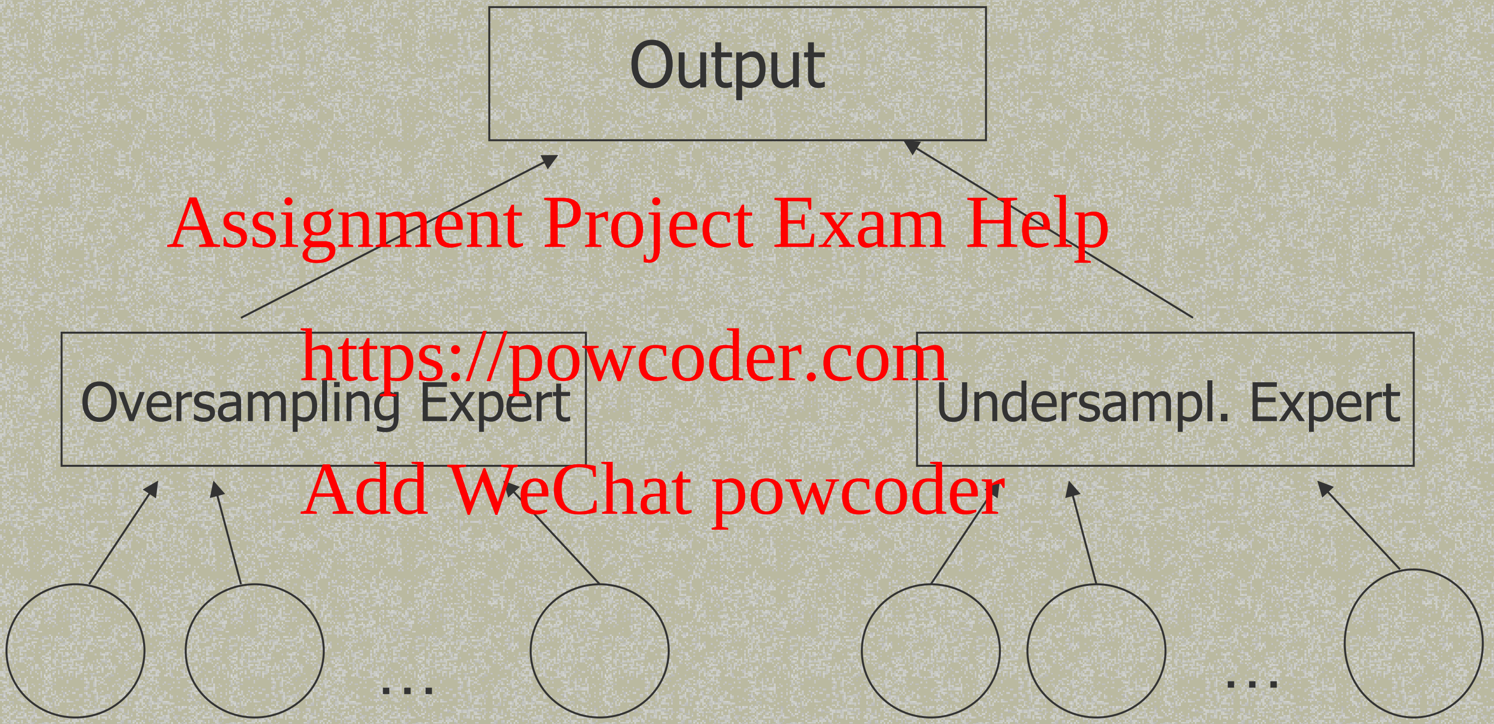
II.IV Multiple Resampling

Idea (Continued):

- It is not possible to know, a-priori, whether a given domain favours oversampling or undersampling and what resampling rate is best.

- Therefore, we decided to create a self-adaptive combination scheme that considers both strategies at various rates.

II.IVMultiple Resampling



Oversampling Classifiers (sampled at different rates)

Undersampling Classifiers (sampled at diff.rates)

II.IV Multiple Resampling

- The combination scheme was compared to ***C4.5-Adaboost*** (with 20 classifiers) with respect to the ***F_B-measures*** on a text classification task (Reuters-21578, Top 10 categories)
- The F_B-measure combines ***precision*** (the proportion of examples classified as positive that are truly positive) and ***recall*** (the proportion of truly positive examples that are classified as positive) in the following way:
  - $F_1 \rightarrow$ precision = recall
  - $F_2 \rightarrow$ 2 * precision = recall
  - $F_{0.5} \rightarrow$ precision = 2 * recall

II.IV Testing the Combination Scheme → Results

In all cases, the mixture scheme is superior to Adaboost However, though it helps ***both*** recall and precision, it helps ***recall more***.