

# ANALYZING SUDOKU DIFFICULTY

CS-556

Performance modelling

YRKAV

# OUTLINE



# WHAT IS SUDOKU

- It's a logic based number placement puzzle.
- A filled 9x9 sudoku should have all digits from 1 to 9 in each row, column and 3x3 grid.
- One of the world's most popular puzzles<sup>1</sup>
  - People solve it on newspapers, websites, apps
  - People of all ages solve it

[1] Rosenhouse, J. & Taalman, L. *Taking Sudoku Seriously: The Math Behind the World's most Popular Pencil Puzzle* (Oxford University Press, New York, 2011)

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

# IMPORTANCE & BENEFITS OF PROBLEM

Why are we considering this problem?

1. This analysis will help us understand one of the most popular and participated puzzles in the world. (Any advances may lead to lots of \$\$)
2. Will help up give rating to an unsolved Sudoku. Sudoku's are almost always provided with a difficulty rating.
3. Classification of Sudoku according to difficulty useful for -
  - Beginners who want to solve only easy ones first.
  - Experts who want to solve only hard ones.

Each Sudoku has different level of difficulty.

Time taken to solve may range from a few minutes to a couple of hours.

What factors determine the difficulty?

# FACTORS AFFECTING DIFFICULTY

- We are considering those factors which are superficial, which can be seen immediately by looking at the unsolved Sudoku.
- **MISSION** - analyze whether such superficial factors play a role in determining the difficulty of a Sudoku.

## Justification

We are considering simple, superficial factors because we want to establish if we can know whether a Sudoku is easy/difficult just by looking at it.

# FACTOR 1: NUMBER OF GIVENS

7	6			5			8	
4				3		2		
	3	2				4		6
					8			
		9		6		8		
			2					
6		8				9	1	
		7		4				5
	2			9			3	8

Number of givens  
here is **27**

## Type

- Controllable

## Values

- $< A$
- $\geq A$

$$A = 28$$

## FACTOR 2: NUMBER OF 3X3 GRIDS WITH LESS GIVENS

7	6			5			8	
4				3		2		
	3	2				4		6
					8			
		9		6		8		
			2					
6		8				9	1	
		7		4				5
	2			9			3	8

Here we see **2** 3x3 grids which have only one given

### Type

- Controllable

### How much less?

- Those 3x3 grids with  $\leq B$  givens

### Values

- $> A$
- $\leq A$

$$A = 2$$

$$B = 2$$

# FACTOR 3: NUMBER OF ROWS WITH LESS GIVENS

7	6			5			8	
4				3		2		
	3	2				4		6
					8			
		9		6		8		
			2					
6		8				9	1	
		7		4				5
	2			9			3	8

Here we see **2** row  
with low givens

Type

- Controllable

Which rows?

- Those with  $\leq B$  givens

Values

- $> A$
- $\leq A$

$A = 2$

$B = 2$



# FACTOR 4: NUMBER OF COLUMNS WITH LESS GIVENS

7	6			5			8	
4				3		2		
	3	2				4		6
					8			
		9		6		8		
			2					
6		8				9	1	
		7		4				5
	2			9			3	8

Here we see **2**  
columns with low  
givens

Type

- Controllable

Which columns?

- Those with  $\leq B$  givens

Values

- $> A$
- $\leq A$

$A = 2$

$B = 2$

# FACTOR 5: NUMBER OF RARE DIGITS

7	6			5			8	
4				3		2		
	3	2				4		6
					8			
		9		6		8		
			2					
6		8				9	1	
		7		4				5
	2			9			3	8

Here we see **1** digit which occurs rarely

## Type

- Controllable

## How rare?

- Number of digits occurring  $\leq B$  times

## Values

- $> A$
- $\leq A$

$$A = 0$$

$$B = 1$$

# FACTOR 6: NUMBER OF CELLS WITH LOW CANDIDATES

7	6			5			8	
4				3		2		
	3	2				4		6
					8			
		9		6		8		
			2					
6		8				9	1	
		7		4				5
	2			9			3	8

Here we see **12** such cells

Type

- Controllable

How low?

- Number of cells which have  $\leq B$  candidates

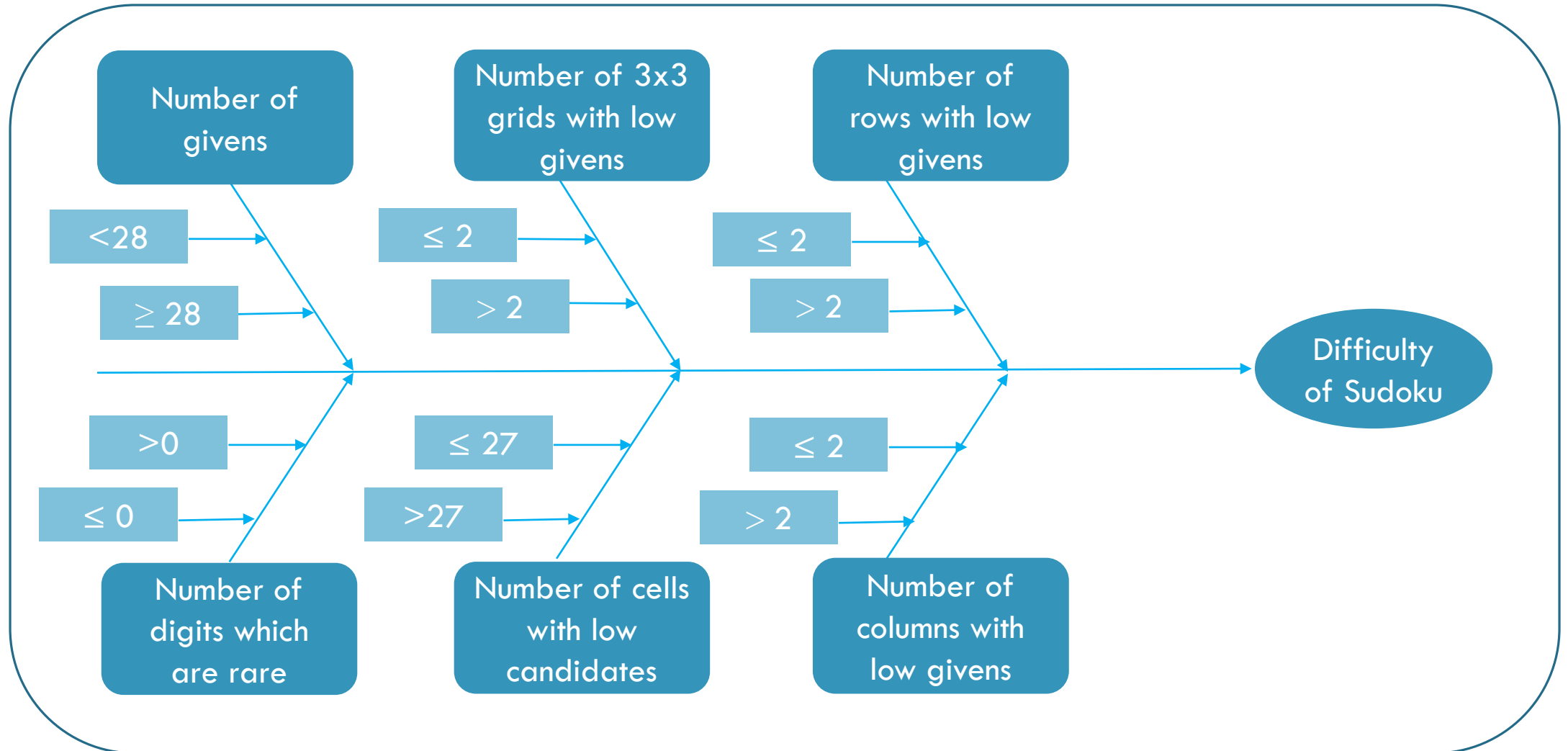
Values

- $\leq A$
- $> A$

$A = 9$

$B = 2$

# FISHBONE DIAGRAM



# NECESSITY AND ADEQUACY OF FACTORS

- Since we want to analyze only superficial factors – the factors which we consider are adequate.
  1. We consider the number of givens. (Factor-1)
  2. We consider the distribution of givens. (Factor-5).
  3. We consider the layout of the givens across the Sudoku – number of rows, cols and 3x3 grids having low number of givens. (Factors-2,3,4)
- We have clearly considered all factors related to the physical appearance of the Sudoku. Hence, they are adequate and necessary.
- We have considered those cells which can be filled up easily at first. Hence, this is also a superficial factor. (Factor-6)

# GENERATING SUDOKU PUZZLES — FLOW CHART

Generate many  
FILLED Sudokus



Eliminate numbers  
while maintaining  
unique solution



Classify these  
unsolved Sudokus  
into  $2^6$  groups



Pick 15 from  
each of those  
 $2^6$  groups

# GENERATING FILLED SUDOKU

Latin Squares are 3x3 sudokus with digits from 0 to 2. There are only 12 such squares

0	1	2
1	2	0
2	0	1

0	1	2
2	0	1
1	2	0

0	2	1
1	0	2
2	1	0

0	2	1
2	1	0
1	0	2

1	0	2
0	2	1
2	1	0

1	0	2
2	1	0
0	2	1

1	2	0
0	1	2
2	0	1

1	2	0
2	0	1
0	1	2

2	0	1
0	1	2
1	2	0

2	0	1
1	2	0
0	1	2

2	1	0
0	2	1
1	0	2

2	1	0
1	0	2
0	2	1

0	2	1
1	0	2
2	1	0

1	0	2
0	2	1
2	1	0

0	1	2
1	2	0
2	0	1

2	0	1
1	2	0
0	1	2

2	1	0
1	0	2
0	2	1

2	1	0
0	2	1
1	0	2

0	1	2
2	0	1
1	2	0

2	0	1
0	1	2
1	2	0

0	2	1
2	1	0
1	0	2

7	9	8	5	4	6	1	2	3
8	7	9	4	6	5	2	3	1
9	8	7	6	5	4	3	1	2
6	4	5	3	2	1	9	8	7
5	6	4	2	1	3	7	9	8
4	5	6	1	3	2	8	7	9
1	2	3	9	7	8	4	6	5
3	1	2	7	8	9	6	5	4
2	3	1	8	9	7	5	4	6

7	9	8	5	4	6	1	2	3
6	4	5	3	2	1	9	8	7
1	2	3	9	7	8	4	6	5
8	7	9	4	6	5	2	3	1
5	6	4	2	1	3	7	9	8
3	1	2	7	8	9	6	5	4
9	8	7	6	5	4	3	1	2
4	5	6	1	3	2	8	7	9
2	3	1	8	9	7	5	4	6

Take 9 such Latin squares randomly (P1-P9) and form a 9x9 grid. Take another Latin square Q which is superimposed in the below diagram.



For each cell multiply the digit in Q with 3 and add its number. Further add 1.  
Ex: A1 cell above has value  $3 \times 2 + 0 + 1 = 7$   
This is a Sudoku satisfying row and column constraints.

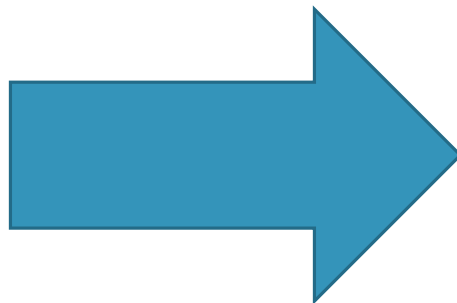


To satisfy the 3x3 grid constraints, we perform 3 row replacements as shown. This gives us a valid complete sudoku

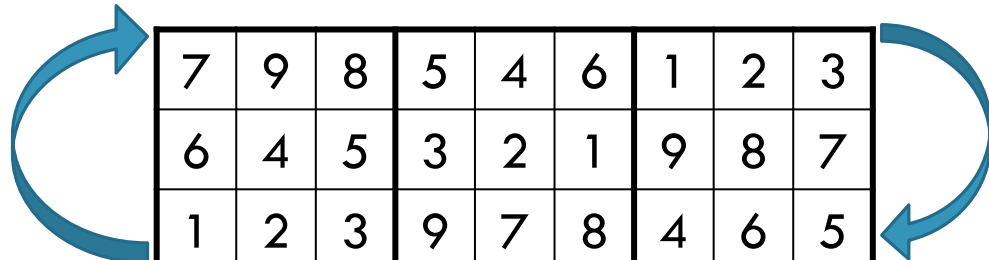


# TRANSFORMATIONS

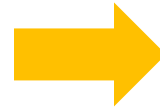
- We apply the following operations on the Sudoku which PRESERVE ALL SUDOKU PROPERTIES. i.e. a transformation takes a valid Sudoku to another valid Sudoku.
- All transformations are applied multiple times in a **mixed random order**.



# PERMUTATIONS OF ROWS

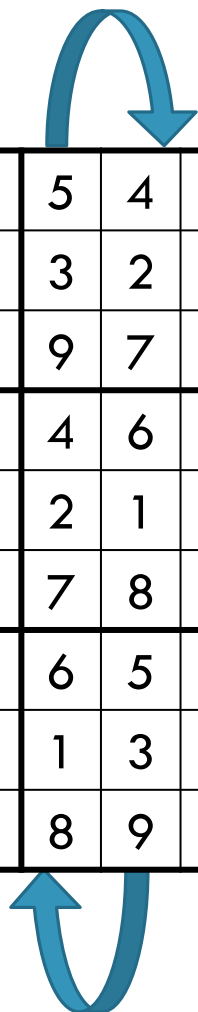


7	9	8	5	4	6	1	2	3
6	4	5	3	2	1	9	8	7
1	2	3	9	7	8	4	6	5
8	7	9	4	6	5	2	3	1
5	6	4	2	1	3	7	9	8
3	1	2	7	8	9	6	5	4
9	8	7	6	5	4	3	1	2
4	5	6	1	3	2	8	7	9
2	3	1	8	9	7	5	4	6

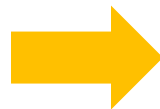


1	2	3	9	7	8	4	6	5
6	4	5	3	2	1	9	8	7
7	9	8	5	4	6	1	2	3
8	7	9	4	6	5	2	3	1
5	6	4	2	1	3	7	9	8
3	1	2	7	8	9	6	5	4
9	8	7	6	5	4	3	1	2
4	5	6	1	3	2	8	7	9
2	3	1	8	9	7	5	4	6

# PERMUTATIONS OF COLUMNS



7	9	8	5	4	6	1	2	3
6	4	5	3	2	1	9	8	7
1	2	3	9	7	8	4	6	5
8	7	9	4	6	5	2	3	1
5	6	4	2	1	3	7	9	8
3	1	2	7	8	9	6	5	4
9	8	7	6	5	4	3	1	2
4	5	6	1	3	2	8	7	9
2	3	1	8	9	7	5	4	6



7	9	8	4	5	6	1	2	3
6	4	5	2	3	1	9	8	7
1	2	3	7	9	8	4	6	5
8	7	9	6	4	5	2	3	1
5	6	4	1	2	3	7	9	8
3	1	2	8	7	9	6	5	4
9	8	7	5	6	4	3	1	2
4	5	6	3	1	2	8	7	9
2	3	1	9	8	7	5	4	6

# TRANSPOSE

7	9	8	5	4	6	1	2	3
6	4	5	3	2	1	9	8	7
1	2	3	9	7	8	4	6	5
8	7	9	4	6	5	2	3	1
5	6	4	2	1	3	7	9	8
3	1	2	7	8	9	6	5	4
9	8	7	6	5	4	3	1	2
4	5	6	1	3	2	8	7	9
2	3	1	8	9	7	5	4	6



7	6	1	8	5	3	9	4	2
9	4	2	7	6	1	8	5	3
8	5	3	9	4	2	7	6	1
5	3	9	4	2	7	6	1	8
4	2	7	6	1	8	5	3	9
6	1	8	5	3	9	4	2	7
1	9	4	2	7	6	3	8	5
2	8	6	3	9	5	1	7	4
3	7	5	1	8	4	2	9	6

# MAPPING DIGITS

1	2	3	4	5	6	7	8	9
3	8	5	6	7	4	1	9	2



7	9	8	5	4	6	1	2	3
6	4	5	3	2	1	9	8	7
1	2	3	9	7	8	4	6	5
8	7	9	4	6	5	2	3	1
5	6	4	2	1	3	7	9	8
3	1	2	7	8	9	6	5	4
9	8	7	6	5	4	3	1	2
4	5	6	1	3	2	8	7	9
2	3	1	8	9	7	5	4	6



1	2	9	7	6	4	3	8	5
4	6	7	5	8	3	2	9	1
3	8	5	2	1	9	6	4	7
9	1	2	6	4	7	8	5	3
7	4	6	8	3	5	1	2	9
5	3	8	1	9	2	4	7	6
2	9	1	4	7	6	5	3	8
6	7	4	3	5	8	9	1	2
8	5	3	9	2	1	7	6	4

# COMPUTER ALGORITHM TO SOLVE A SUDOKU

The two key ideas used to solve a Sudoku are:

## Constraint Propagation

1. If a square has only one possible value, then eliminate that value from the square's peers.
2. If a unit has only one possible place for a value, then put the value there.

These updates to a square may in turn cause further updates to its peers, and the peers of those peers, and so on. This process is called constraint propagation.

## Backtracking Search

If we haven't already found a solution or a contradiction, choose one unfilled square and consider all its possible values. One at a time, we try assigning the square each value, and searching from the resulting position. In other words, we search for a value  $d$  such that we can successfully search for a solution from the result of assigning square  $s$  to  $d$ . If the search leads to a failed position, go back and consider another value of  $d$ . This is a recursive search, and we call it a depth-first search because we (recursively) consider all possibilities under values(s)  $d$  before we consider a different value for  $s$ .

# ELIMINATING NUMBERS

An important property of Sudoku is that it has a unique solution. The only parameter that can be controlled while eliminating numbers to generate an unsolved Sudoku is the number of givens in it.

We use the Sudoku Solving Algorithm to determine if it has  $>1$  solutions. Q. How are we ensuring it has atleast 1 soln?

Pick a random cell to eliminate

9		6		7		4		3
			4			2		
	7			2	3		1	
5						1		
	4		2		8		6	
		3		9				5
	3		7				5	
		7			5			
4		5		1		7		8

This grid has 2 solutions

9		6		7		4		3
			4			2		
	7			2	3		1	
5						1		
	4		2		8		6	
		3						5
	3		7				5	
		7			5			
4		5		1		7		8

9	2	6	5	7	1	4	8	3
3	5	1	4	8	6	2	7	9
8	7	4	9	2	3	5	1	6
5	8	2	3	6	7	1	9	4
1	4	9	2	5	8	3	6	7
7	6	3	1			8	2	5
2	3	8	7			6	5	1
6	1	7	8	3	5	9	4	2
4	9	5	6	1	2	7	3	8

9	4
4	9

4	9
9	4

So go back and pick another cell to eliminate. At each step we choose those cells for elimination which maintain the solution. We proceed to remove the number of givens that we want to.

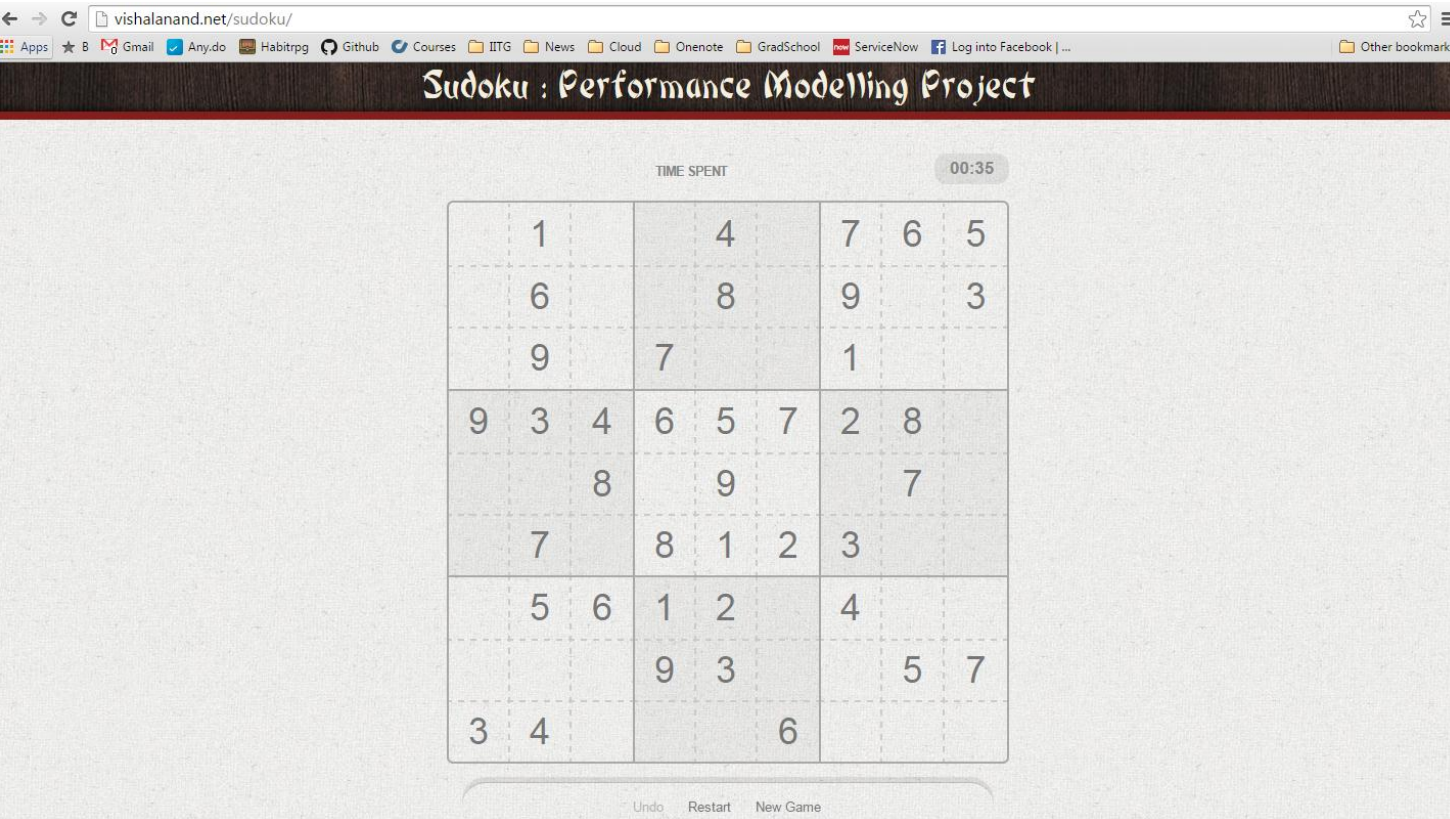
# CLASSIFY GENERATED SUDOKUS

- Following procedure described earlier, we generate 5000 Sudokus.
- The only factor we could control was factor-1 (Number of Givens) while generating Sudokus.
- We chose cutoffs for each factor (As and Bs) such that we get the most even split of the 5000s Sudokus across 64 categories.
- We could get 15 Sudoku's in each category.

**FOOTNOTE:** We wrote A LOT of python scripts. The total lines in all scripts was 1024!



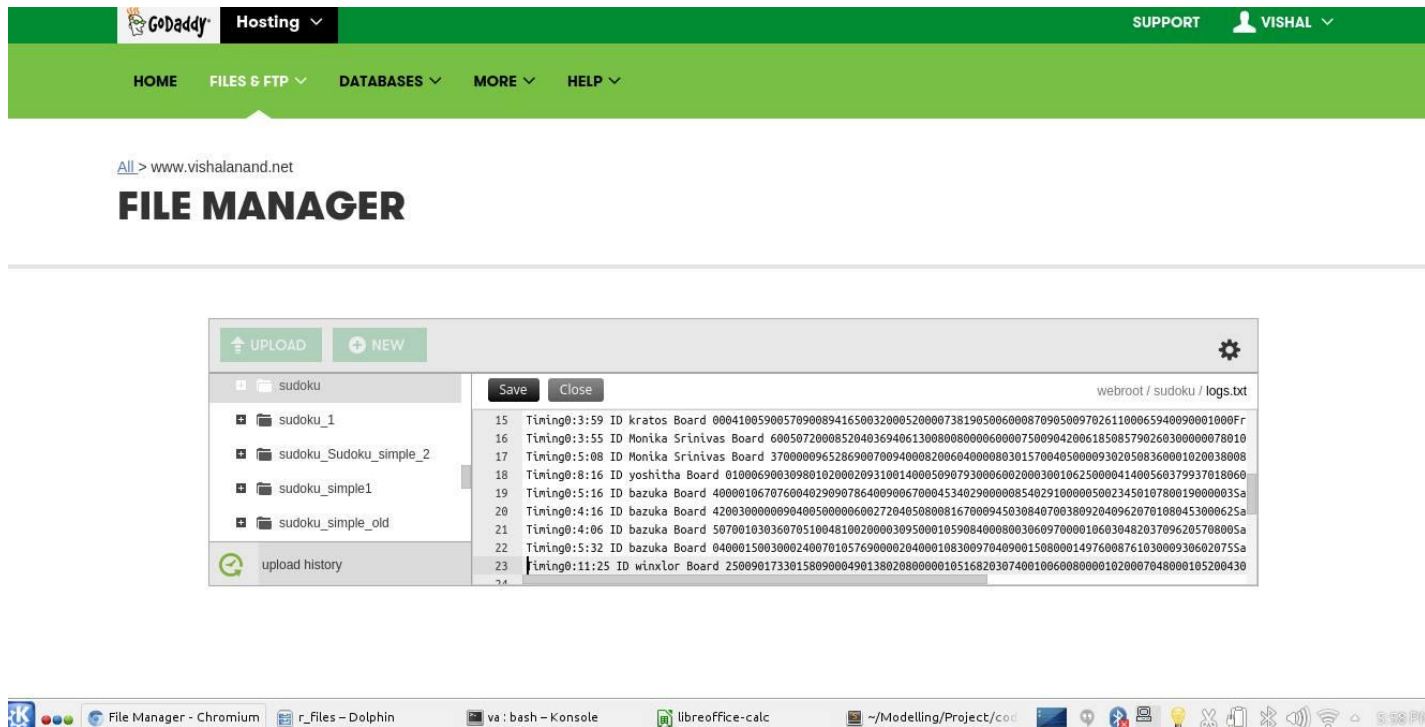
# DATA COLLECTION - WEBSITE



<http://vishalanand.net/sudoku/>

- Created a web portal to collect data.  
<http://vishalanand.net/sudoku/>
- Shared among friends in IIT Guwahati and other IITs. Posted link on Sudoku websites and blogs and got reasonable external participation.
- Collected 768 Sudoku data out of which we used 640. (64x10)

# DATA COLLECTION



- We asked user for his/her name to identify user every time he/she plays.
- We gave a user Sudokus from all different categories before repeating the same category.
- This minimizes operator error accumulating in one category.

# ANALYSIS

We used three regression models to figure out which factor contributes how much towards difficulty of a Sudoku.

1. Linear Regression
2. Full factorial regression
3. Multiplicative Regression

We also use 2 machine learning models for further analysis.

# MODEL SELECTION — WHY WE CHOSE THEM?

1. Our factors have two levels each -
  1. One which makes Sudoku intuitively hard  
eg. Number of rows with more givens is less
  2. One which makes Sudoku intuitively easy  
eg. Number of rows with more givens is more.

Hence these models in which factors have 2 levels with replications are ideal.

2. These models cover both cases - when there are no interactions between factors (logistic regression) and there may be interactions (full factorial model).
3. We are using both linear model (full factorial) and multiplicative model.

We are ensuring that all types of models are being considered.  
This is because we may need to show that superficial factors don't  
work for any model type.

# MODELS

## 1. Linear Regression

$$y_{ij} = q_0 + q_A X_A + q_B X_B + e_{ij}$$

## 2. Full Factorial Regression

$$y_{ij} = q_0 + q_A X_A + q_B X_B + q_{AB} X_{AB} + e_{ij}$$

## 3. Multiplicative (logistic) Full-Factorial Regression

$$y_{ij} = 10^{q_0} 10^{q_A X_A} 10^{q_B X_B} 10^{q_{AB} X_{AB}} 10^{e_{ij}}$$

$$\log(y_{ij}) = q_0 + q_A X_A + q_B X_B + q_{AB} X_{AB} + e_{ij}$$

Linear

Multiplicative

Interactions

No Interactions

# ANOVA ANALYSIS

“ANalysis Of VAriation”

We try to see what are the sources of variation in the data. If error term contributes highly to the total variation, then the model is not a good fit.

Variation

$$SSE = \sum_{i=1}^{2^k} \sum_{j=1}^r e_{ij}^2$$

$$SST = SSA + SSB + SSAB + SSE$$

Variance

$$s_e^2 = \frac{SSE}{2^k(r-1)}$$

$$s_{q_0} = s_{q_0} = s_{q_0} = s_{q_0} = \frac{s_e}{\sqrt{2^k r}}$$

Confidence Interval:  $q_i \mp t_{[\frac{1-\alpha}{2}; 2^k(r-1)]} s_{q_i}$   
(0 should lie outside this interval)

# LINEAR REGRESSION

Terms describing model	Value
Error	99.45%
Goodness of the model	0.55%
F-value	1.11
Critical f-value (with $\alpha = 0.05$ )	1.14
F-test result	Failed

# LINEAR REGRESSION

First few factors with most (and negligible) contribution to variation

Factor	Contribution	Coefficient values	F-Test	Confidence interval test
Factor 5: Number of rare digits	0.25%	150.24	Fail	Fail
Factor 1: Number of givens	0.19%	131.17	Fail	Fail
Factor 3: Number of rows with low givens	0.05%	-65.73	Fail	Fail



# LINEAR REGRESSION

- We can see that even the top contributing factors contributes very less to total variation.
- No factor has passed the F-Test.
- No factor has passed the confidence interval test.
- So we conclude that this model does not fit our data well.
- What if we introduce interactions? (Error reduces, but not much)

# FULL FACTORIAL REGRESSION

Terms describing model	Value
Error	90.5%
Goodness of the model	9.5%
F-value	1.10
Critical f-value (with $\alpha = 0.05$ )	1.14
F-test result	Failed

# FULL FACTORIAL REGRESSION

First few factors with most (and negligible) contribution to variation

Factor	Contribution	Coefficient values	F-test	Confidence interval test
Factor 1 & Factor 6	0.76%	259.04	Pass	Pass
Factor 1, Factor 2, Factor 3, Factor 5	0.52%	214.81	Fail	Fail
Factor 3, Factor 4, Factor 6	0.48%	206.73	Fail	Fail

# FULL FACTORIAL REGRESSION

- We can see that even the top contributing factors contributes very less.
- Only one factor has passed the F-Test.
- Only one has passed the confidence interval test.
- So we conclude that this model does not fit our data.
- What if we introduce a multiplicative model? (Error reduces, but not much)

# MULTIPLICATIVE MODEL

Terms describing model	Value
Error	81.89%
Goodness of the model	18.11%
F-value	1.09
Critical f-value (with $\alpha = 0.05$ )	1.14
F-test result	Failed

# MULTIPLICATIVE MODEL

First few factors with most (and negligible) contribution to variation

Factor	Contribution	Coefficient values	F-test	Confidence interval test
Factor 1	5.65%	0.104	Pass	Pass
Factor 1, Factor 6	1.98%	0.06	Pass	Pass
Factor 2, Factor 3, Factor 5, Factor 6	0.95%	0.04	Pass	Pass

# MULTIPLICATIVE MODEL

- We can see that even the top contributing factors contribute very less, a bit more than previous models.
- Few factors have passed the F-Test.
- Only a few factors have passed the confidence interval test.
- So we conclude that this model, although better, also does not fit our data.
- We give up on regression models.

## FOOTNOTE

Since the regression models failed important tests and we rejected the models, we did not proceed with other tests like Visual Test etc.

# MACHINE LEARNING MODELS

Since the standard regression models don't fit Sudoku data well, we try to fit some Machine learning models

We divide the time taken to solve into 9 classes with equal ranges and apply machine learning techniques for multi-class classification.

- Naïve Bayes
- Decision Trees



# NAÏVE BAYES

- It is a conditional probability model which given a vector  $\mathbf{x} = (x_1, \dots, x_n)$  representing  $n$  features, assigns probabilities  $p(C_k | x_1, \dots, x_n)$  for each of the  $k$  possible classes.
- An important **assumption** here is that in each class, the features/factors are independent. We make this assumption for our model and try to fit it.
- Using Bayes theorem, the conditional probability can be decomposed as  $p(C_k | \mathbf{x}) = \frac{p(C_k) p(\mathbf{x} | C_k)}{p(\mathbf{x})}$ .

where  $p(\mathbf{x} | C_k) = p(x_1 | C_k) p(x_2 | C_k) p(x_3 | C_k) \dots$

- The model assigns class label to an example as  $\hat{y} = \operatorname{argmax}_{k \in \{1, \dots, K\}} p(C_k) \prod_{i=1}^n p(x_i | C_k)$ .

# RESULTS

Correctly classified instances	40.15%
Incorrectly classified instances	59.85%
Kappa statistic	0.185
Relative absolute error	92.85%

$$\text{Kappa statistic} = (\text{observed accuracy} - \text{expected accuracy}) / (1 - \text{expected accuracy})$$

Expected accuracy is the accuracy obtained by a purely random classifier. Kappa Statistic is a true measure of the performance of a multiclass classifier, instead of just the error %

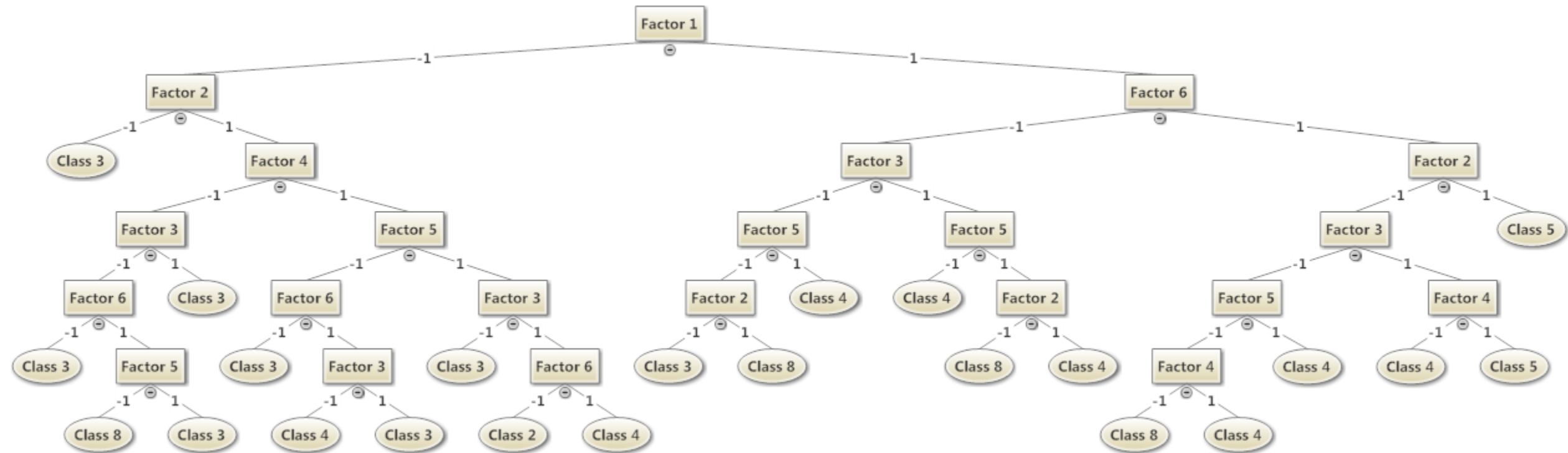
# DECISION TREE

Since in Naïve Bayes, we made the assumption that factors are independent within a class, we now chose a model where factors need not be independent – Decision Trees.

- Decision tree is a tree where a decision is made at the node and is split according to the decision taken. The leaves are tried to be made as pure as possible (examples from same category).
- **Algorithm:** split the node at a decision which maximizes information gain. Information is gained when entropy is minimized.

Entropy :  $H(S) = - \sum_{x \in X} p(x) \log_2 p(x)$     Information Gain:  $IG(A, S) = H(S) - \sum_{t \in T} p(t) H(t)$

This is the decision tree obtained on our data by running it on a decision-tree algorithm.



# RESULTS

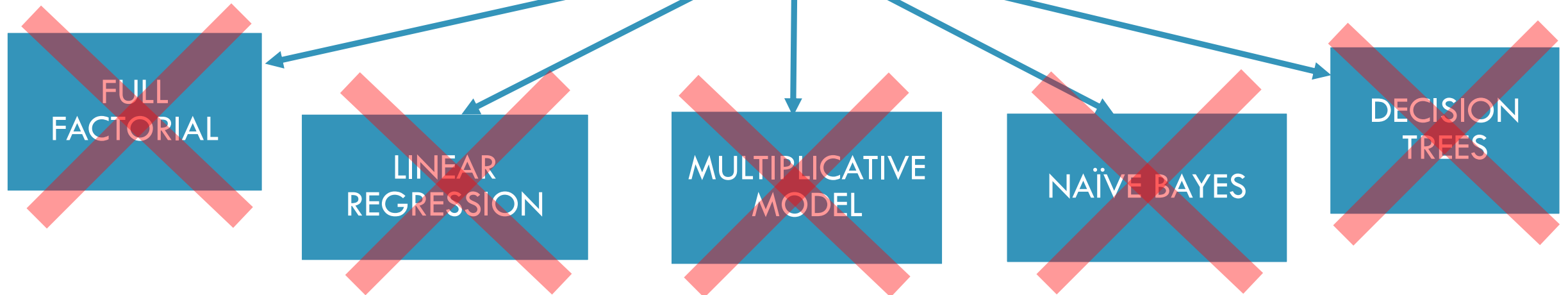
Correctly classified instances	49.7%
Incorrectly classified instances	50.3%
Kappa statistic	0.325
Relative absolute error	81.5%

There is an improvement here w.r.t. Naïve Bayesian model which might be because Decision trees don't assume independence of factors.

We see here that both the ML models still have high errors:  
Kappa statistic  $< 0.4$  are considered unacceptable. [*J.L. Fleiss, Fleiss' Kappa Test*]

# FINDINGS AND CONCLUSIONS

We find that our given factors do not fit many models.



# FINDINGS AND CONCLUSIONS

- We conclude that the factors we chose for our Sudoku problem do not fit any model. They are **NOT GOOD FACTORS**.



- Superficial factors such as number of givens, layout of the givens etc. **ARE NOT** good indicators of Sudoku difficulty. This has been corroborated by existing literature and popular opinion.

*“Assessing difficulty levels of sudoku is surprisingly complicated, and can often only be really determined by solving the puzzle, which kinda defeats the purpose. The rankings given in newspapers and book collections of these tricky puzzles are often misleading.”*

-Denise, Puzzle Writer and Author

*“It is natural to expect a correlation with the number of clues - the fewer clues are given, the harder the puzzle. This is not universally true.”*

-M. Ercsey-Ravasz, Z. Toroczkai, “The Chaos Within Sudoku”, Scientific Reports 2, pp. 755-762, 2012

*“You cannot judge the difficulty level of a Sudoku puzzle by looking at the number of givens or the layout of givens. You can have an easy puzzle and a ridiculously hard puzzle with exactly the same layout of givens. The number of givens does not correlate to difficulty. You can have an easy puzzle with fewer givens and a difficult puzzle with more.”*

- Roy Leban, Founder & CEO, Puzzazz (a puzzle analysis company)

and many more .....



# LESSONS LEARNT

1. We have proved that superficial factors are bad indicators of Sudoku difficulty. One cannot determine how hard a Sudoku is unless one starts solving it.
2. It is very hard to convince people to solve Sudokus.

## **What would we have done differently?**

1. We could have collected more data.
2. We could have checked more models to validate that our data doesn't magically fit into a new model.
3. We could possibly have used user's Sudoku experience as a factor.

# FUTURE WORK

1. We have established that superficial factors are not good for predicting Sudoku difficulty. We can ascertain this by analyzing for more complex models.
2. Identify more complex factors which do play a role in Sudoku difficulty.
  - Time taken by computer algorithm to solve Sudoku.
  - How many strategies can be successfully applied to solve the Sudoku.
3. Performance Analysis for the complex factors.

QUESTIONS??



THANK YOU !!!