Problem Chosen	2023 NEUQMCM Summary	Control Team Number
A		JS06012024001

2024NEUQMCM PreContest

The summary is an essential part of your MCM/ICM paper and should appear as the first page of your solution report. The judges place considerable weight on the summary, and winning papers are often distinguished from other papers based on the quality of the summary.

- To write a good summary, imagine that a reader will choose whether to read the body of the paper based on your summary: Your concise presentation in the summary should inspire a reader to learn about the details of your work.
- You should write the summary last, as it should clearly describe your approach to the problem and, most prominently, your most important conclusions. Ensure you plan time after solving your problem to write a comprehensive and articulate summary.
- Summaries that are mere restatements of the contest problem, or are a cut-and-paste boilerplate from the Introduction are generally considered to be weak.

Key Words: cardiovascular; machine learning; logistic regression

Contents

1 Problem Background and Related Research Overview

Provide citation and references, following guidance provided here. Double-check citations to ensure they are accurate and are properly referenced.

1.1 Problem Background

The team's solution should be articulate, concise, and organized in order to allow the reader to easily follow the solution process and conclusions. Key statements should present major ideas and results.

- A Table of Contents assists the reader in previewing the organization of your report.
- Present a clarification or restatement of the problem as appropriate.
- Present a clear exposition of all variables and hypotheses.
- State and justify reasonable assumptions that bear on the problem.
- Present an analysis of the problem, motivating or justifying the model being used.
- Summarize derivations, computations, or illustrative examples in the main body of the solution, and leave lengthy derivations and/or calculations and data in appropriate appendices.
- Include a design of the model. Discuss how the model could be tested, to include error analysis, sensitivity, and/or stability.
- Discuss any apparent strengths or weaknesses to your model or approach.
- Provide a conclusion and report results explicitly.
- Document resources and references.

1.2 Related Work

For mathematical modeling beginners, the book named 'A First Course in Mathematical Modeling' [1] is a good reference.

2 Assumptions and Data Preprocessing

1. Clearly indicate the use of LLMs or other AI tools in their report, including which model was used and for what purpose. Please use inline citations and the reference section. Also append the Report on Use of AI (described below) after your 25-page solution.
2. Verify the accuracy, validity, and appropriateness of the content and any citations generated by language models and correct any errors or inconsistencies.
3. Provide citation and references, following guidance provided here. Double-check citations to ensure they are accurate and are properly referenced.

4. Be conscious of the potential for plagiarism since LLMs may reproduce substantial text from other sources. Check the original sources to be sure you are not plagiarizing someone else's work.

2.1 Assumptions

2.2 Data Preprocessing

3 Exploratory Data Analysis

- minimizes the discomfort to the hands, or
- maximizes the outgoing velocity of the ball.

We focus exclusively on the second definition.

- the initial velocity and rotation of the ball,
- the initial velocity and rotation of the bat,
- the relative position and orientation of the bat and ball, and
- the force over time that the hitter hands applies on the handle.

Theorem 3.1. *No free lunch.*

In mathematical folklore, the "no free lunch" (NFL) theorem (sometimes pluralized) of David Wolpert and William Macready appears in the 1997 "No Free Lunch Theorems for Optimization" [2]. Wolpert had previously derived no free lunch theorems for machine learning (statistical inference) [3].

Lemma 3.2 (Moore's Law)**.** *Processor speeds, or overall processing power for computers will double every two years.*

Proof. Moore's law is a 1965 observation made by Intel co-founder Gordon E. Moore that the number of transistors placed in an integrated circuit (IC) or chip doubles approximately every two years. Because Moore's observation has been frequently cited and used for research and development by multiple organizations, and it has been proven repeatedly, it is known as Moore's law. □

It follows that

$$a^2 + b^2 = c^2 \tag{1}$$

Include your model design and justification for type model used or developed.
Calculating and simplifying the model:

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x)g(y)$$

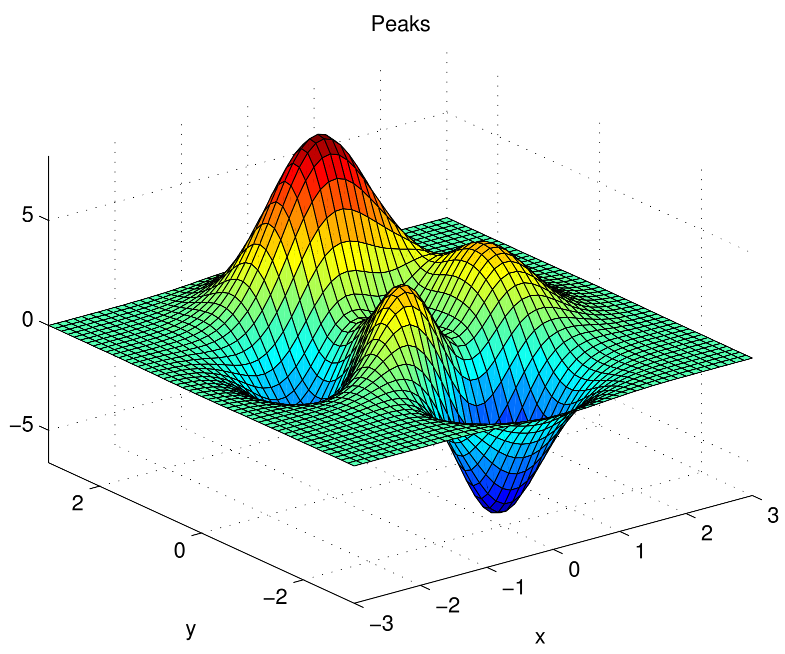

Figure 1: The Curve Plane of System with form of pdf

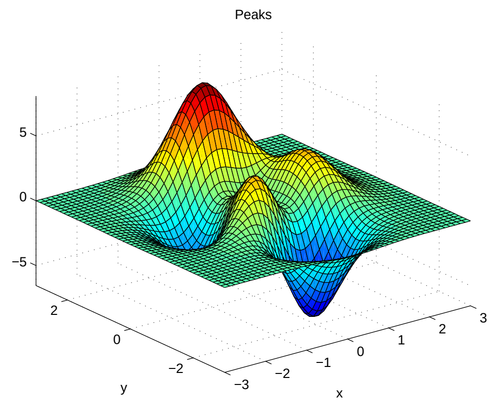

Figure 2: The Curve Plane of System with form of eps

4 Machine Learning Methods for Classification

The equation (1) has told that

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \frac{\text{Opposite}}{\text{Hypotenuse}} \cos^{-1}\theta \arcsin\theta$$

$$p_j = \begin{cases} 0, & \text{if } j \text{ is odd} \\ r!\,(-1)^{j/2}, & \text{if } j \text{ is even} \end{cases}$$

$$\arcsin\theta = \oint_{\varphi} \lim_{x\to\infty} \frac{n!}{r!(n-r)!}$$

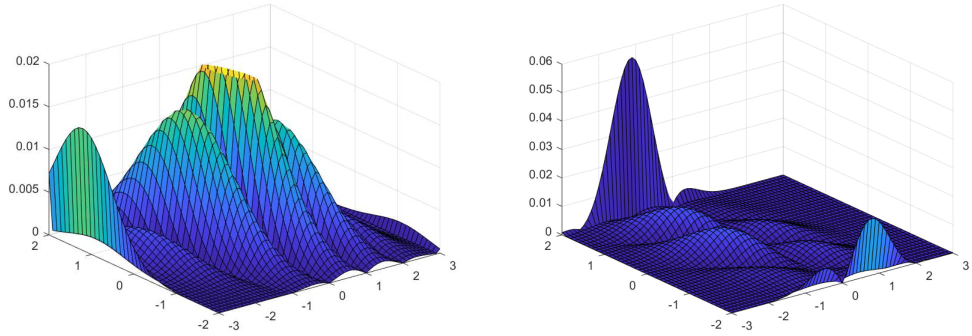

Figure 3: Comparison

The normal distribution, also known as the Gaussian distribution, would be the most important continuous distribution. For $-\infty < \mu < +\infty$ and $\sigma > 0$, the normal distribution is denoted by $N(\mu, \sigma^2)$, and its probability density is given by

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}.$$

The cumulative distribution function (cdf) of a normal variable X is written by

$$F(x) = P(X \le x) = \int_{-\infty}^{x} f(t)\,\mathrm{d}t = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(t-\mu)^2}{2\sigma^2}}\,\mathrm{d}t,$$

denoted by $X \sim N(\mu, \sigma^2)$ with probability density function (pdf) $f(x) = \frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

The frequency property of a controlled system is analyzed via

$$G(\mathrm{i}\omega) = G(s)\Big|_{s=\mathrm{i}\omega} = A(\omega)\mathrm{e}^{\mathrm{i}\phi(\omega)} = U(\omega) + \mathrm{i}V(\omega).$$

5 Simulation Result and Model evaluation

Describe model testing and sensitivity analysis, including error analysis, etc.

Table 1: A three-line form

system	Version	Editor
Windows	MikTEX	TEXMakerX
Unix/Linux	teTEX	Kile

6 Conclusion

The team's solution should be articulate, concise, and organized in order to allow the reader to easily follow the solution process and conclusions. Key statements should present major ideas and results.

- A Table of Contents assists the reader in previewing the organization of your report.
- Present a clarification or restatement of the problem as appropriate.
- Present a clear exposition of all variables and hypotheses.
- State and justify reasonable assumptions that bear on the problem.
- Present an analysis of the problem, motivating or justifying the model being used.
- Summarize derivations, computations, or illustrative examples in the main body of the solution, and leave lengthy derivations and/or calculations and data in appropriate appendices.
- Include a design of the model. Discuss how the model could be tested, to include error analysis, sensitivity, and/or stability.
- Discuss any apparent strengths or weaknesses to your model or approach.
- Provide a conclusion and report results explicitly.
- Document resources and references.

References

[1] Frank R. Giordano, William P. Fox, Steven B. Horton. A First Course in Mathematical Modeling, 5th edition, Brooks/Cole Cengage Learning, 2013.

[2] D. H. Wolpert, W. G. Macready. No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation 1(1), 67-82 (1997).

[3] D. H. Wolpert. The Lack of A Priori Distinctions between Learning Algorithms. Neural Computation 8(7), 1341-1390 (1996).

Appendices

Appendix A: Programmes Codes

Here are simulation programmes we used in our model as follow.

Input matlab source:

```
function [t,seat,aisle]=OI6Sim(n,target,seated)
pab=rand(1,n);
for i=1:n
    if pab(i)<0.4
       aisleTime(i)=0;
    else
        aisleTime(i)=trirnd(3.2,7.1,38.7);
    end
end
```

Input C++ source:

```
//=====================================================
// Name        : Sudoku.cpp
// Author      : wzlf11
// Version     : a.0
// Copyright   : Your copyright notice
// Description : Sudoku in C++.
//=====================================================

#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int table[9][9];

int main() {

    for(int i = 0; i < 9; i++){
        table[0][i] = i + 1;
    }

    srand((unsigned int)time(NULL));

    shuffle((int *)&table[0], 9);

    while(!put_line(1))
    {
        shuffle((int *)&table[0], 9);
    }

    for(int x = 0; x < 9; x++){
        for(int y = 0; y < 9; y++){
            cout << table[x][y] << " ";
        }

        cout << endl;
    }

    return 0;
}
```

Input Python source:

```python
"""
mnist_svm
~~~~~~~~~

A classifier program for recognizing handwritten digits from the MNIST
data set, using an SVM classifier."""

#### Libraries
# My libraries
import mnist_loader

# Third-party libraries
from sklearn import svm

def svm_baseline():
    training_data, validation_data, test_data = mnist_loader.load_data()
    # train
    clf = svm.SVC()
    clf.fit(training_data[0], training_data[1])
    # test
    predictions = [int(a) for a in clf.predict(test_data[0])]
    num_correct = sum(int(a == y) for a, y in zip(predictions, test_data[1]))
    print "Baseline classifier using an SVM."
    print "%s of %s values correct." % (num_correct, len(test_data[1]))

if __name__ == "__main__":
    svm_baseline()
```

Input R source:

```r
####################################################
### How to Read this Book?
####################################################

R.version


####################################################
### Starting with R
####################################################

install.packages('DMwR')

installed.packages()

library()

old.packages()

update.packages()
y <- 43
```

```
y


z <- 5
w <- z^2
w
i <- (z*2 + 45)/2
i


(34 + 90)/12.5


ls()
rm(y)
```

Report on Use of AI

Think carefully about how to document and reference whatever tools the team may choose to use. A variety of style guides are beginning to incorporate policies for the citation and referencing of AI tools. Use inline citations and list all AI tools used in the reference section of your 25-page solution. Whether or not a team chooses to use AI tools, the main solution report is still limited to 25 pages. If a team chooses to utilize AI, following the end of your report, add a new section titled Report on Use of AI. This new section has no page limit and will not be counted as part of the 25-page solution. Examples (this is not exhaustive these examples to your situation):

1. OpenAI ChatGPT (Jan 8, 2024 version, ChatGPT-4,)

Query1: <insert the exact wording you input into the AI tool>
Output1: <insert the complete output from the AI tool>
Query2: <insert the exact wording you input into the AI tool>
Output2: <insert the complete output from the AI tool>

2. OpenAI Ernie (Jan 8, 2024 version, Ernie 4.0)

Query: <insert the exact wording of any subsequent input into the AI tool>
Output: <insert the complete output from the second query>

3. Github CoPilot (Jan 8, 2024 version)

Query: <insert the exact wording you input into the AI tool>
Output: <insert the complete output from the AI tool>

4. Google Bard (Jan 9, 2024 version)

Query: <insert the exact wording of your query>
Output: <insert the complete output from the AI tool>