DATA COMPUTING VISUALIZATION ANALYSIS STATISTICS

WORK

中国大学生计算机设计大赛

上海第二工业大学

面向统计学的数据计算可视化分析平台

作者：王贤安 唐振川 李辰彦 何秀敏 张倬诚

指导老师：朱丹丽 于程程

演示：王贤安

作品小类：Web应用与开发

CONTENTS

01
平台创意

统计学分析平台的作用，统计学分析数据可视化的意义

02
模块简介

一共分为七个模块，分别代表七个网页，主页和相应的六个功能，不断更新中

03
功能实现

功能：判别分析，一元回归分析，多元回归分析，逻辑回归分析、支持向量机分析、机器学习案例

04
使用技术

基于Web.py框架的后端，bootstrap框架的前端，部分前端使用JavaScript辅助计算

ONE

平台创意

专业课课程《统计学基础》

A platform for visualizing the statistical calculation and analysis process.

工作汇报

设计创意

当我们在做统计学数据分析的题目时，我们会发现……

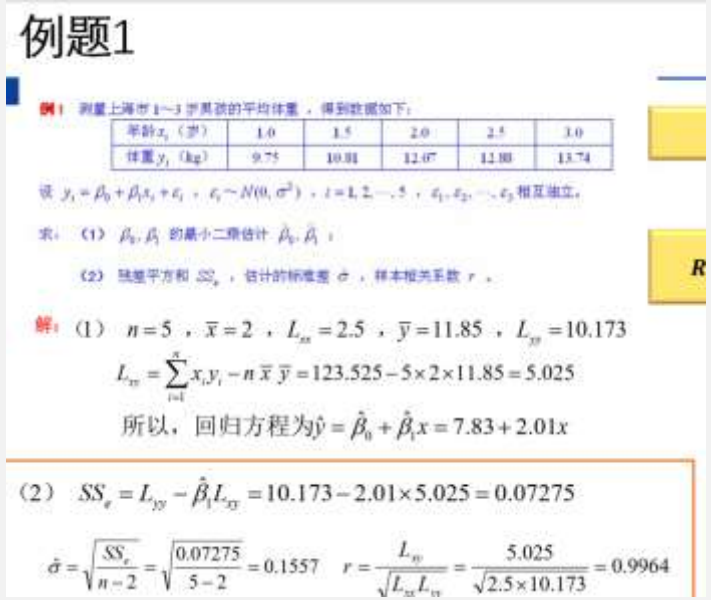
痛点1:

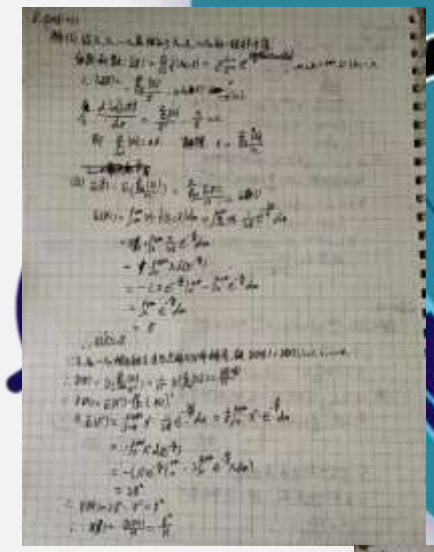
统计学公式记不住

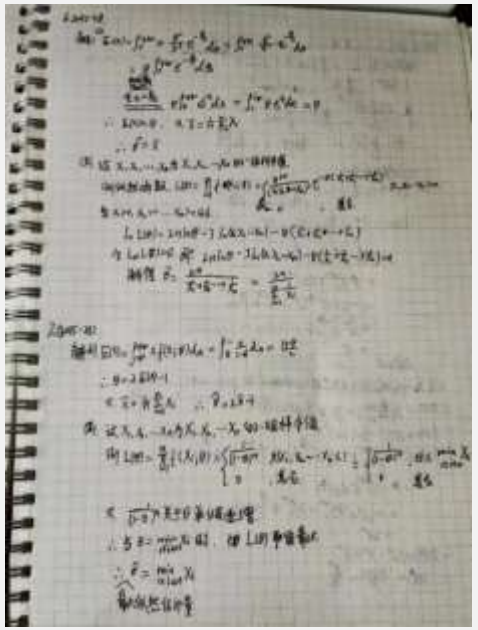
痛点2:

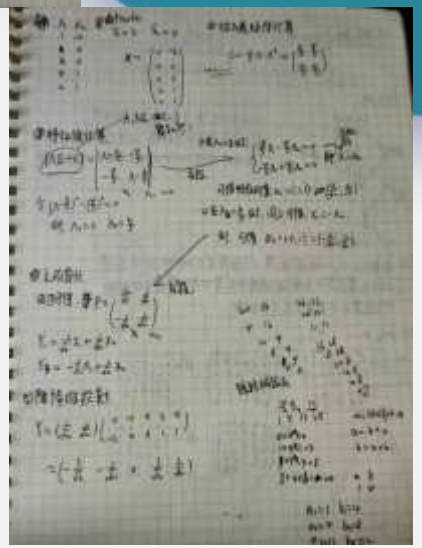
统计学分析数据计算复杂，做题时，无法精确定位错误

该怎么办呢?









工作汇报

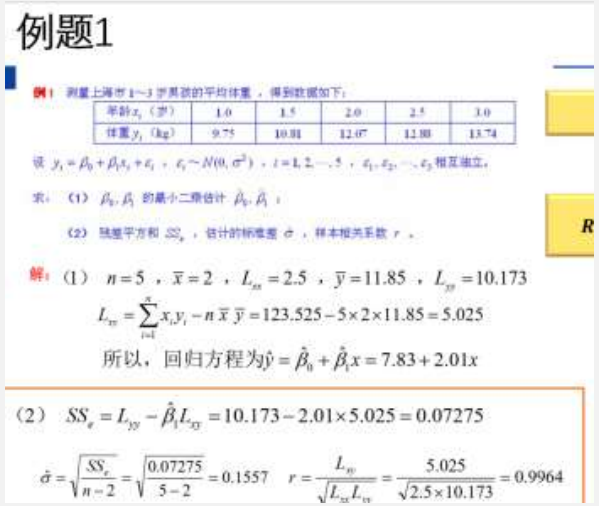
设计创意

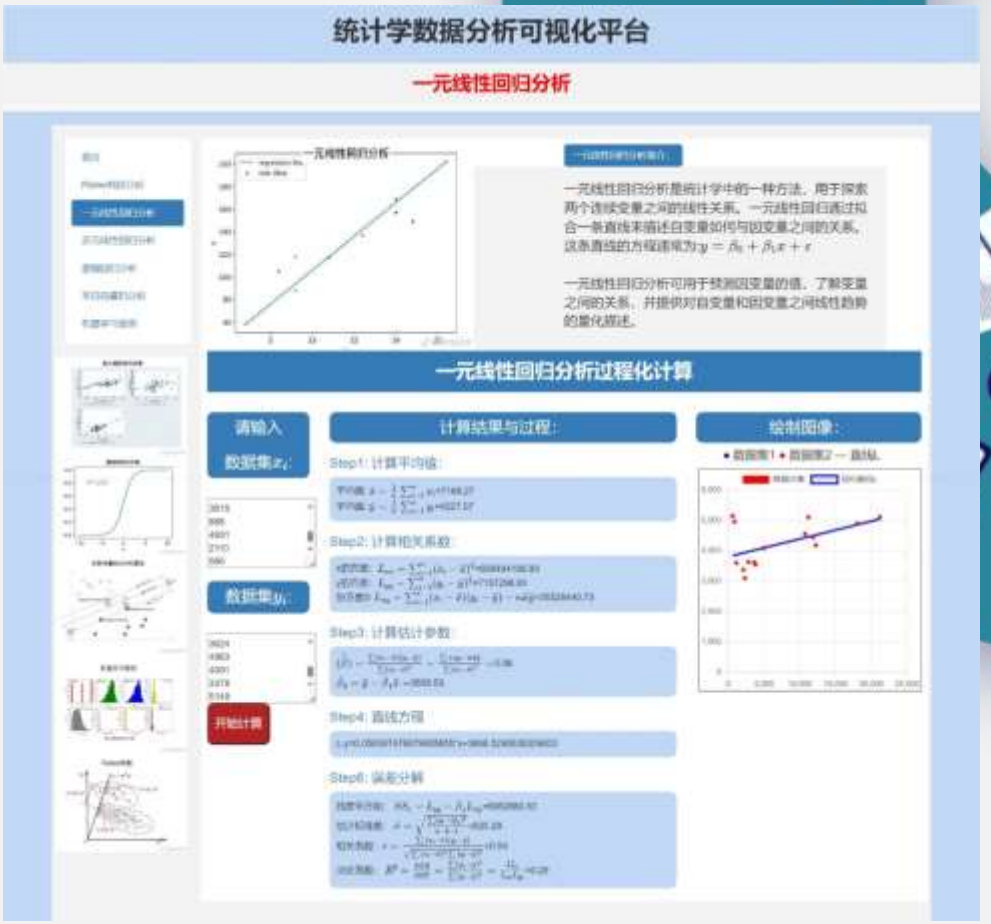
设计思路：

1.自动提供计算思路

2.根据需求提供计算过程及结果

3.可视化展示





解决方案：

在网络平台上进行统计学计算，即可获得对应的公式与运算过程、计算结果

即：

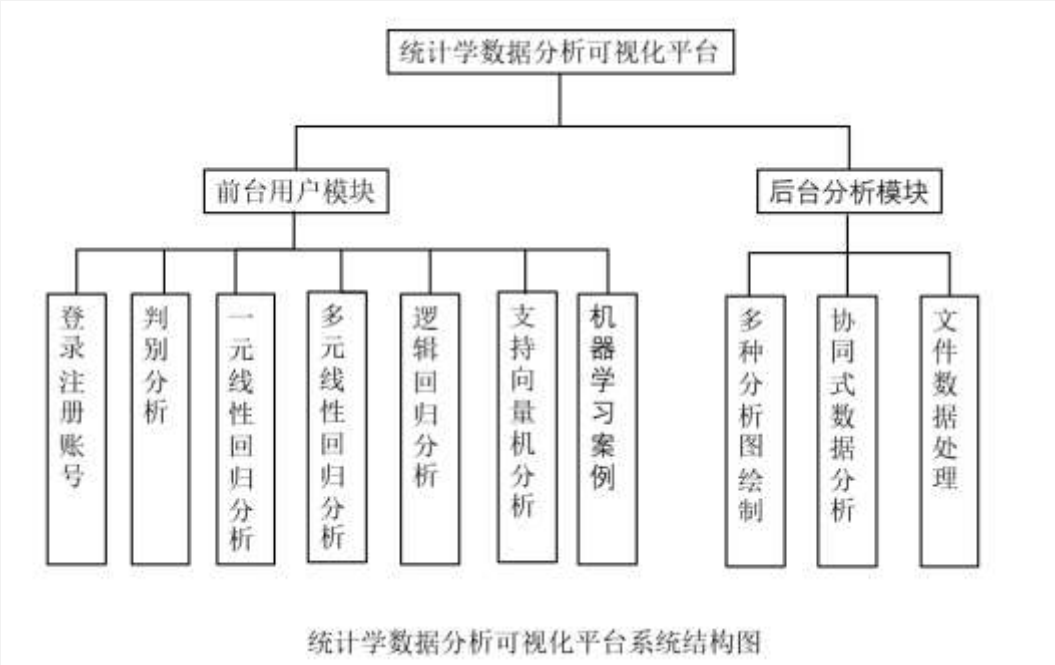
面向统计学的数据计算可视化分析平台

TWO

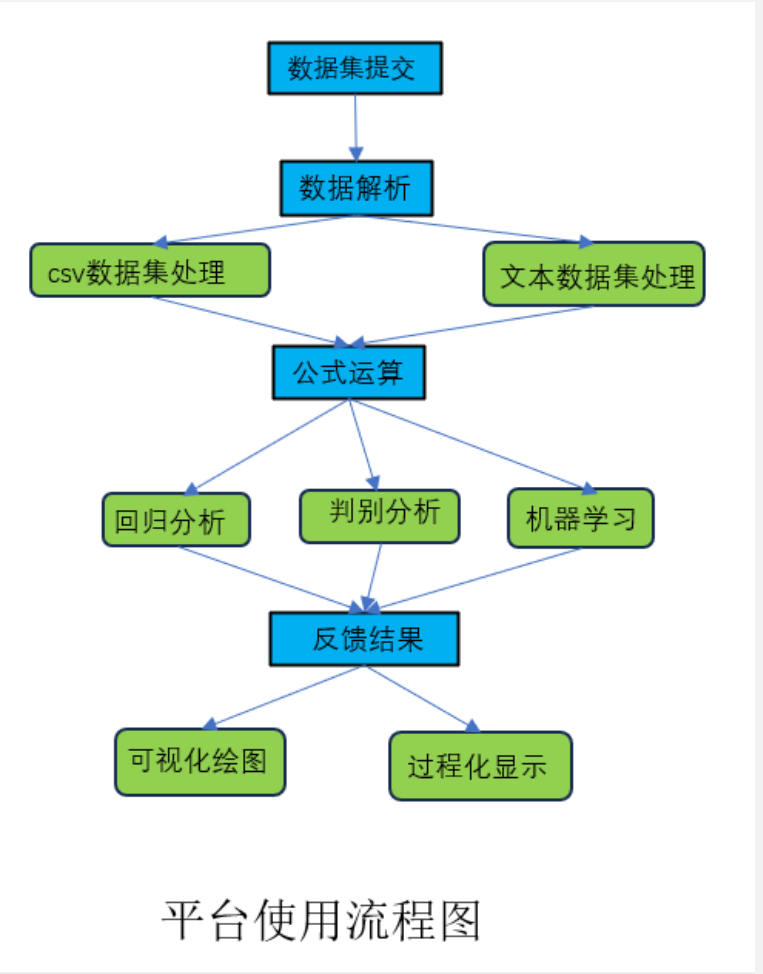
模块简介

平台具有的功能，网页呈现.
目前一共分为七个模块，分别
代表七个网页，即主页和相应
的七个功能

工作汇报



统计学数据分析可视化平台系统结构图



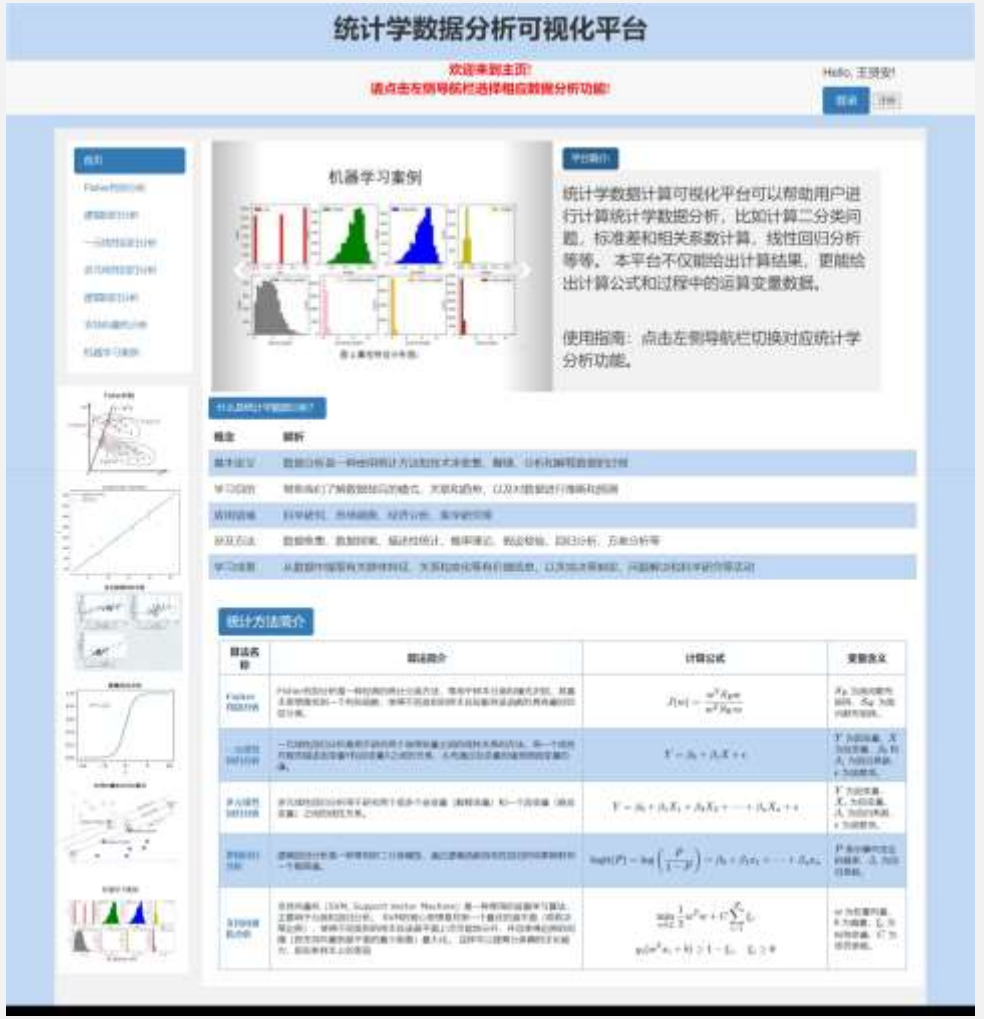
平台使用流程图

- 系统结构与使用流程

工作汇报

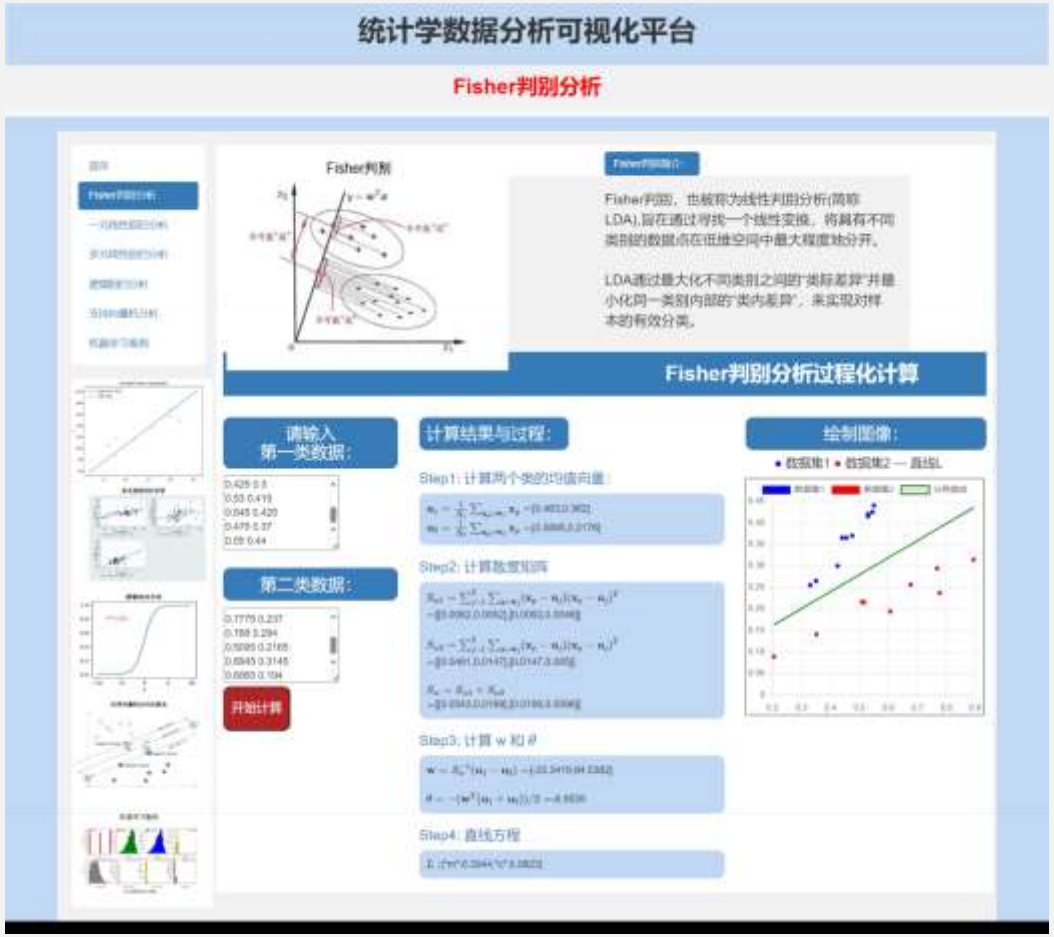
- 登录账号
- 跳转页面
- 系统介绍

主界面=系统简介+操作指南



主界面截图

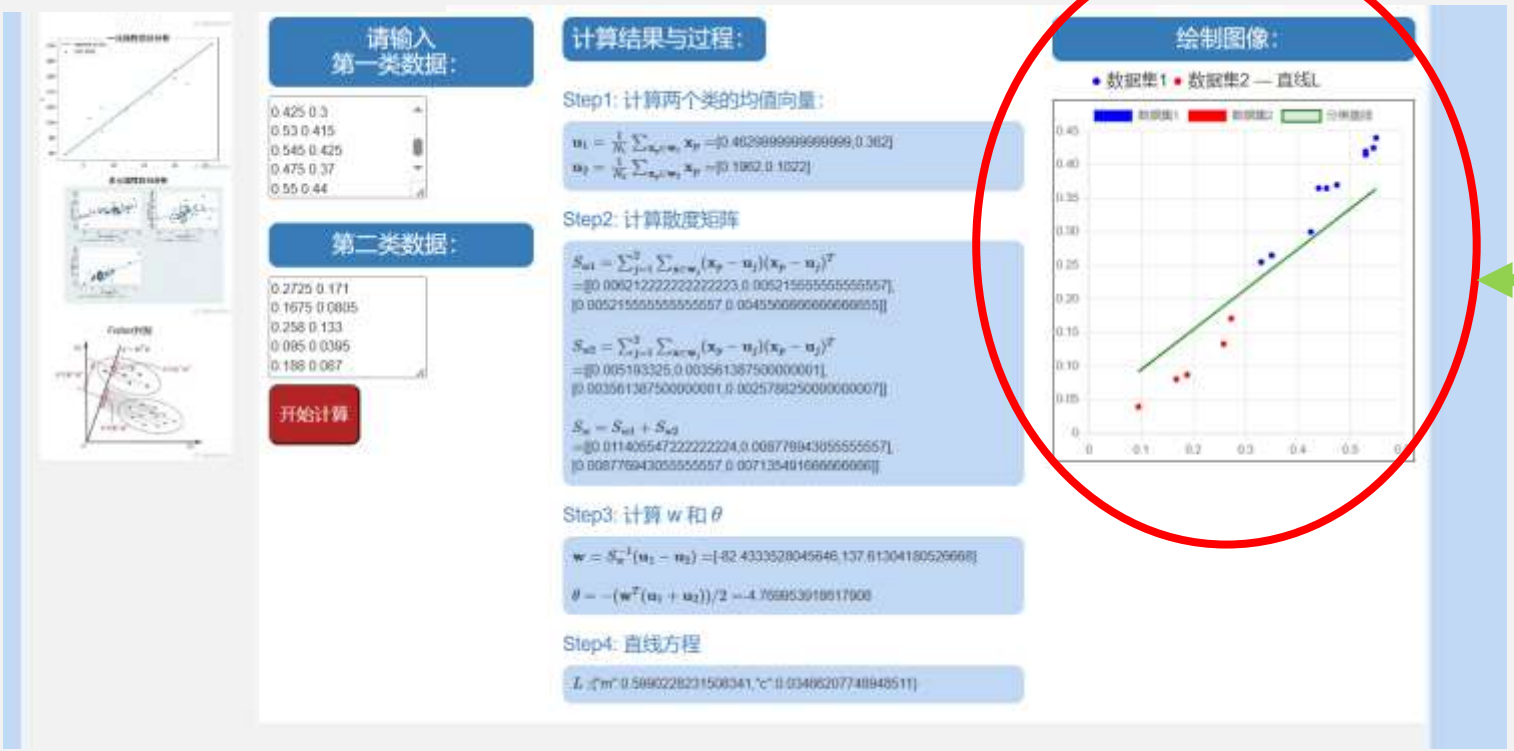
工作汇报



FISHER判别分析主界面

Fisher判别，也被称为线性判别分析(简称LDA),旨在通过寻找一个线性变换，将具有不同类别的数据点在低维空间中最大程度地分开，实现对样本的有效分类。

工作汇报



可自动调节刻度，鼠标位于点上即可显示坐标

FISHER判别分析计算界面

输入数据集 → 统计学 → 后台计算分析 → 可视化 → 公式结果 / 绘制图像

工作汇报

逻辑回归分析主页图

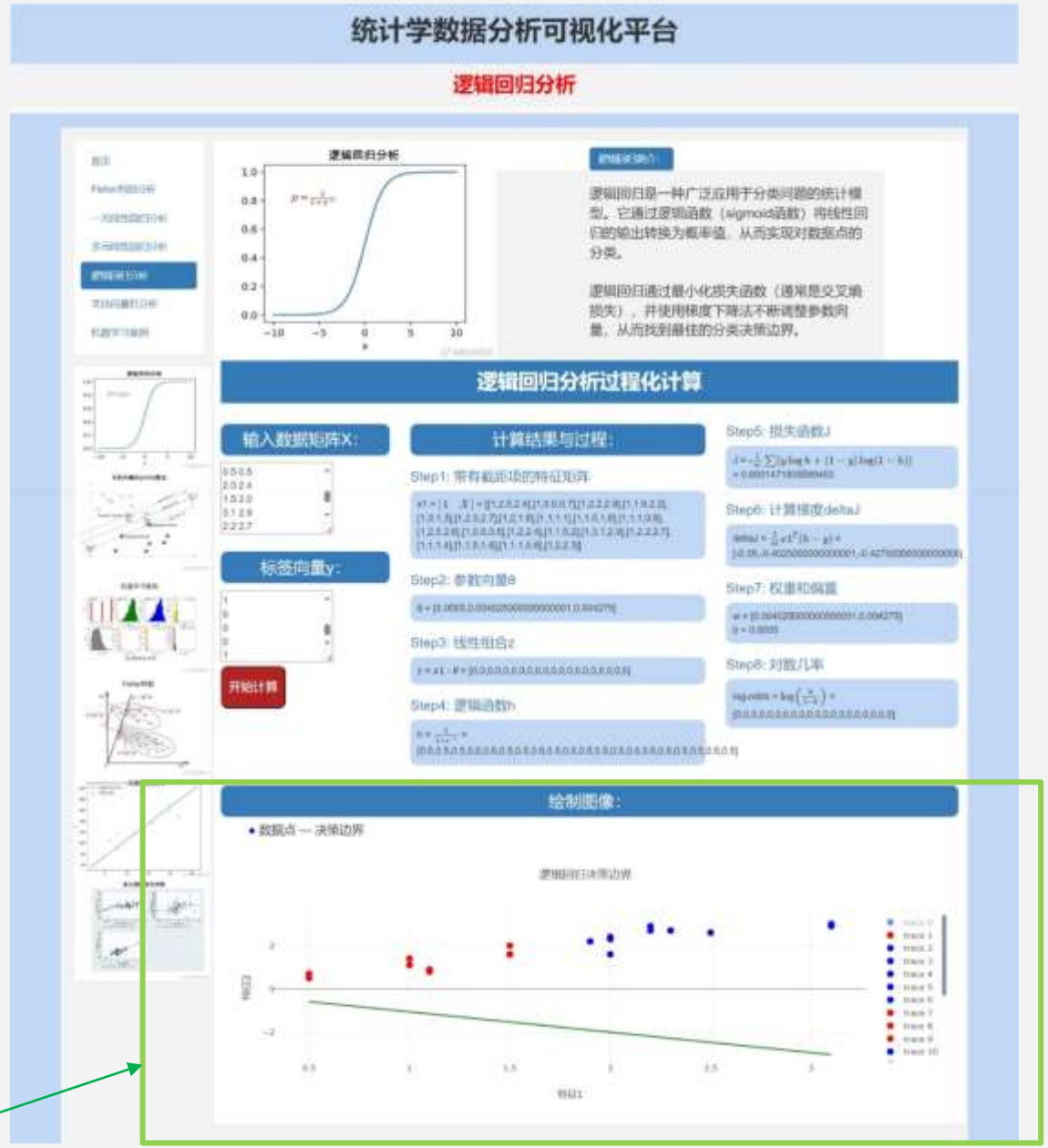
数据矩阵

标签向量

对数几率

- 逻辑回归

PlotlyChart绘图



工作汇报

支持向量机分析主页图

数据矩阵

标签向量

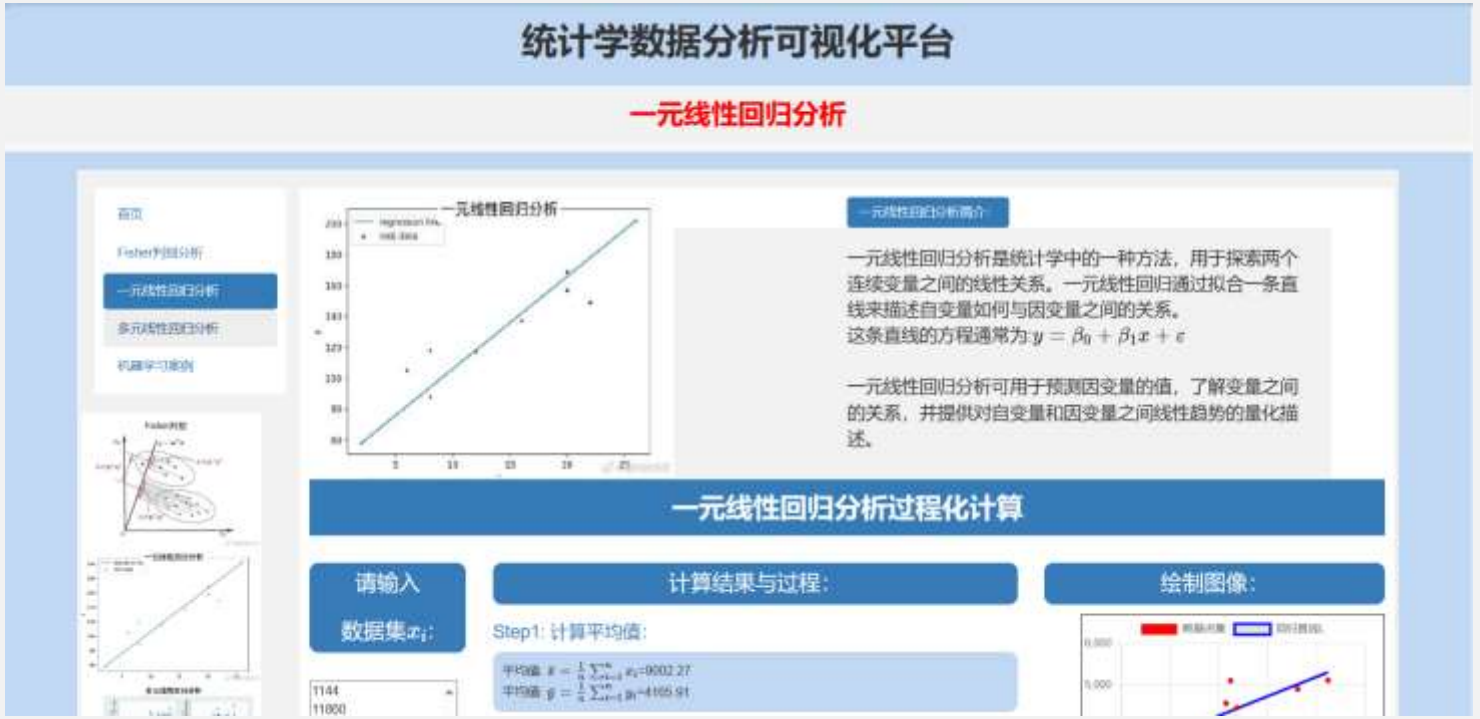
对数几率

PlotlyChart绘图

可读性进一步增强

- 支持向量机（Support Vector Machine）

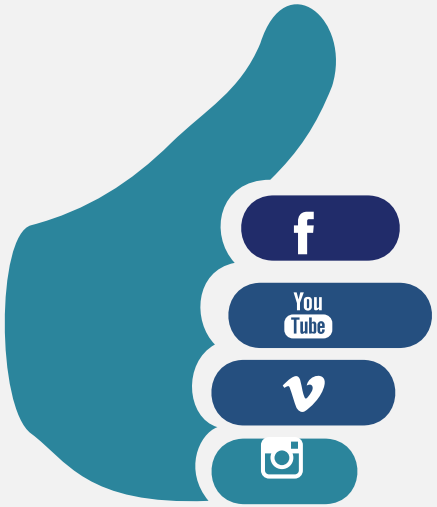
工作汇报



一元线性回归分析主页图

- 描述:
- 一元线性回归分析是统计学中的一种方法，用于探索两个连续变量之间的线性关系。
- 一元线性回归通过拟合一条直线来描述自变量如何与因变量之间的关系。

工作汇报



文板框支持大量数据输入

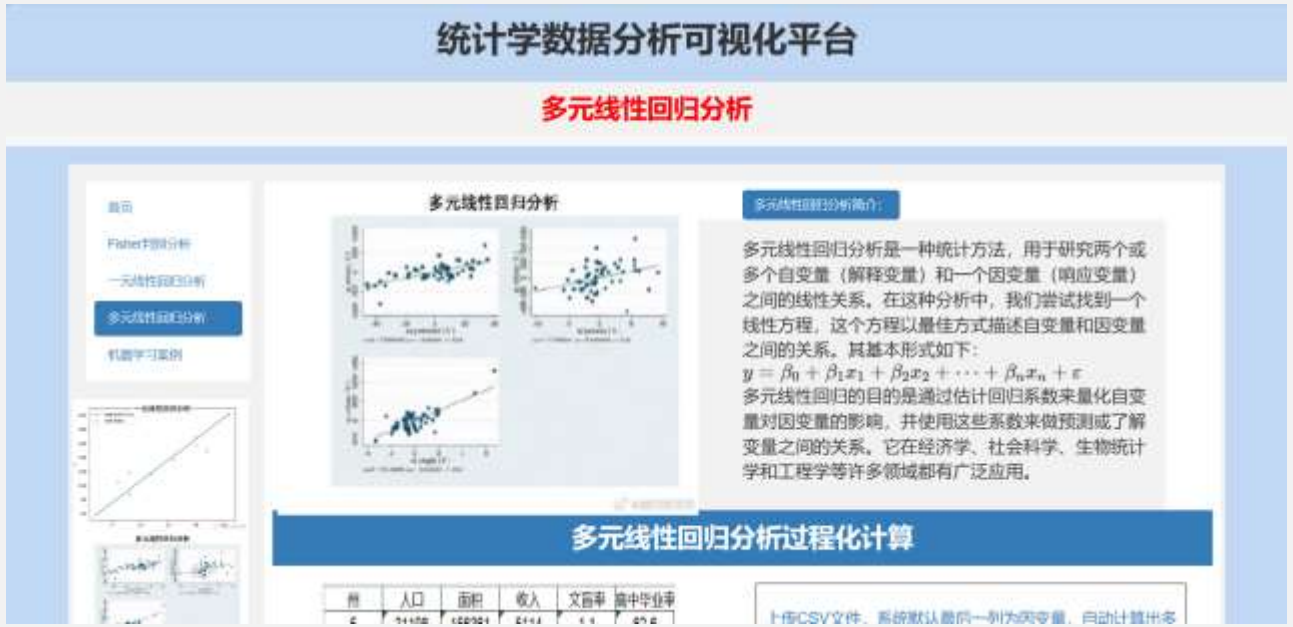
一元线性回归分析计算区截图

- 自动调节刻度
- 红点蓝线

- 平均值
- 相关系数
- 估计参数
- 直线方程
- 误差分解

输入数据集 → 统计学 → 后台计算分析 → 可视化 → 公式结果 / 绘制图像

工作汇报



多元线性回归分析界面
(CSV表格分析)

- 多元线性回归分析研究的线性关系与一元线性回归类似，同样是拟合直线，但是其比一元线性回归方程需要更多自变量，因此计算复杂度更高，本系统采取csv文件分析。

工作汇报

- 分析样本

- 下载学习数据

- 输出方程

多元线性回归分析计算界面
(CSV表格分析)

文件上传 → 后台计算 → 输出结果

工作汇报

机器学习：鲍鱼年龄预测

- 鲍鱼机器学习案例，使用鲍鱼各项物理测量（如长度、直径等）进行机器学习训练，以预测其生物年龄。
- 此界面体现出本系统高聚合、低耦合，要增添新功能稳定且操作简单。

工作汇报

- 下载数据集
- 下载学习模型

机器学习：鲍鱼年龄预测

输入编号 → 读取数据 → 调用数据库 → 机器学习 → 预测结果

工作汇报


面向统计学的数据计算可视化分析平台
历史记录
当前用户：王贤安
返回主页
Fisher判别分析
一元线性回归分析
多元线性回归分析
逻辑回归分析
支持向量机分析
Fisher判别分析历史记录
ID
Username
Data 1
Data 2
u1
u2
S1
S2
Sw
w
theta
L
录入时间
2
王贤安
31335 14256
12117 11169 1
[29467.7629,1
[9718.9157,11
[[13444482.91
[[3071434.673
[[16515917.58
[0.0017,-0.00
-20.6664
{"m": 1.8628,"c
2024-07-15 1
1
王贤安
0.455 0.365 0
0.677 0.2565
[0.463,0.362]
[0.5895,0.217
[[0.0062,0.005
[[0.0481,0.014
[[0.0543,0.015
[-33.3419,84.5
-6.9535
{"m": 0.3944,"c
2024-07-09 1

查看历史记录

THREE

功能实现

实现具体的功能，需要网页与用户进行交互，用户输入数据即可实现计算分析。

具体运行过程可参考程序演示视频

FOUR

技术手段

基于Web.py框架的后端，
bootstrap框架的前端，部分前端
使用JavaScript辅助计算

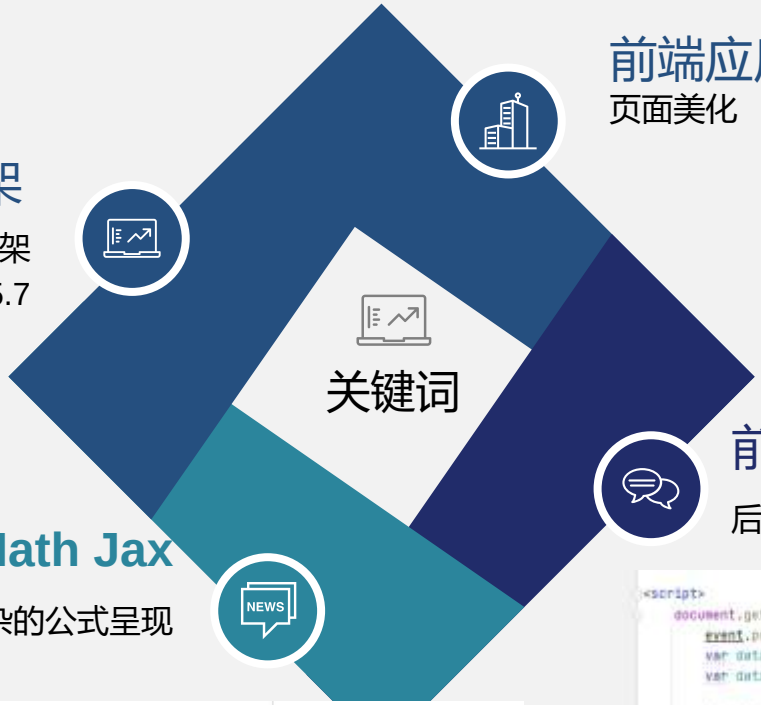
工作汇报

```
import web
import pymysql
import tempfile
import csv
from sklearn import metrics
from sklearn.linear_model import LinearRegression
import joblib
import json
import numpy as np
import pandas as pd
import statsmodels.api as sm
```

web.py框架

服务端构建依赖于Python的web.py框架

数据库MySQL5.7



公式使用Math Jax

复杂的公式呈现

```
<div id="step1" >
    <h2  class="text2">Step1: 计算两个类的均值向量: </h2>

    <p class="myElement">\( \mathbf{u}_1 = \frac{1}{N_1} \sum_{\mathbf{x}_p \in \mathbf{w}_1} \mathbf{x}_p = \)<span
    <br>\( \mathbf{u}_2 = \frac{1}{N_2} \sum_{\mathbf{x}_p \in \mathbf{w}_2} \mathbf{x}_p =\)<span id="u2Result"> </
</div>
```

```
<div class="row">
    <div class="col-md-2"><!-- 2.左侧栏 -->
        <div class ="module1">
        <ul class="nav nav-pills nav-stacked">
            <li><a href="calculateApp.html">首页</a></li>
            <li class="active"><a href="Fisher.html"> Fisher判别分析</a></li>
            <li><a href="Linear.html">一元线性回归分析</a></li>
            <li><a href="MLinear.html">多元线性回归分析</a></li>
            <li><a href="baoyu.html">机器学习案例</a></li>
        </ol>
        </div>
    </div>
    <div class="col-md-10">
        <div class="row">
            <div class="col-md-6"><!-- 3.右侧 -->
```

前端应用bootstrap框架

页面美化

前后端协同计算

后端**Python库**，前端**JavaScript**

```
<script>
    document.getElementById('inputForm').addEventListener('submit', function(event) {
        event.preventDefault(); // 阻止表单的默认提交行为
        var dataset1 = parseData(document.getElementById('data1').value);
        var dataset2 = parseData(document.getElementById('data2').value);

        // 确保数据集不为空
        if (dataset1.length === 0 || dataset2.length === 0 || dataset1.length !== dataset2.length)
            alert('Please enter an equal number of valid values in both datasets.');
            return;
        }

        // 计算
```

工作汇报

公式使用Math Jax

复杂的公式呈现与绘图

公式可视化

数据可视化

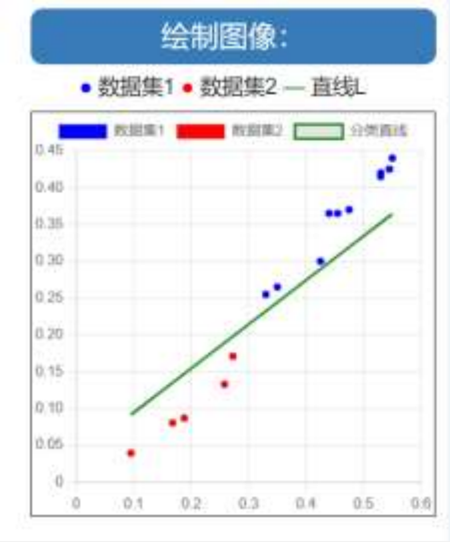
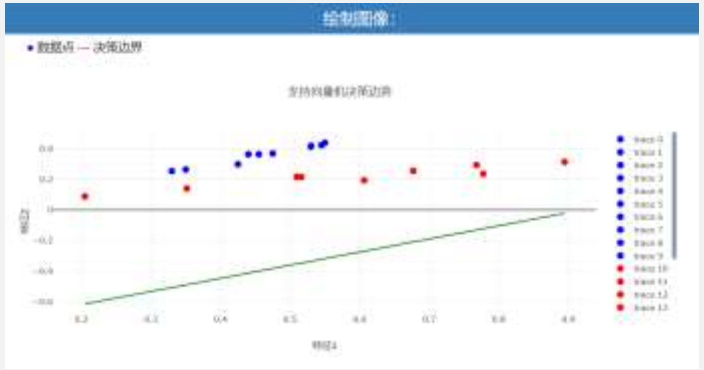


Chart.js



Plotly Chart

工作汇报

前后端协同计算

后端Python库，前端JavaScript

```
<script>
document.getElementById('inputForm').addEventListener('submit', function(event) {
    event.preventDefault();
    var data1 = document.getElementById('data1').value.split('\n').map(x => x.split(' ').map(Number));
    var data2 = document.getElementById('data2').value.split('\n').map(x => x.split(' ').map(Number));

    // 使用 AJAX 请求后端接口
    var xhr = new XMLHttpRequest();
    xhr.open('POST', '/calculate', true);
    xhr.setRequestHeader('Content-Type', 'application/json');
    xhr.onreadystatechange = function () {
        if (xhr.readyState == 4 && xhr.status == 200) {
            var result = JSON.parse(xhr.responseText);
            var dataset1 = data1.map(function(point) { return { x: point[0], y:  point[1] }; });
            var dataset2 = data2.map(function(point) { return { x: point[0], y: point[1] }; });

            // 将结果显示在页面上
            document.getElementById('u1Result').textContent = JSON.stringify(result.u1);
            document.getElementById('u2Result').textContent = JSON.stringify(result.u2);
            document.getElementById('S1Result').textContent = JSON.stringify(result.S1);
            document.getElementById('S2Result').textContent = JSON.stringify(result.S2);
            document.getElementById('SwResult').textContent = JSON.stringify(result.Sw);
            document.getElementById('wResult').textContent = JSON.stringify(result.w);
            document.getElementById('thetaResult').textContent = JSON.stringify(result.theta);
            document.getElementById('LResult').textContent = JSON.stringify(result.L);

        // 根据最小和最大x值计算直线的起点和终点
        var minX = Math.min(...dataset1.concat(dataset2).map(point => point.x));
        var maxX = Math.max(...dataset1.concat(dataset2).map(point => point.x));
        var lineStart = { x: minX, y: result.L.m * minX + result.L.c };
        var lineEnd = { x: maxX, y: result.L.m * maxX + result.L.c };
        var ctx = document.getElementById('myChart').getContext('2d');
```

前后端协同运算

```
class Calculate:
    def POST(self):
        data = json.loads(web.data())
        coordinates1 = np.array(data['coordinates1'])
        coordinates2 = np.array(data['coordinates2'])

        u1 = np.mean(coordinates1, axis=0)
        u2 = np.mean(coordinates2, axis=0)

        S1 = np.cov(coordinates1, rowvar=False)
        S2 = np.cov(coordinates2, rowvar=False)
        Sw = S1 + S2

        reg_lambda = 1e-5
        Sw_reg = Sw + np.eye(Sw.shape[0]) * reg_lambda

        w = np.linalg.inv(Sw_reg).dot(u1 - u2)

        theta = -0.5 * np.dot(w.T, (u1 + u2))

        m = -w[0] / w[1]
        c = -theta / w[1]

        result = {
            'u1': u1.tolist(),
            'u2': u2.tolist(),
            'S1': S1.tolist(),
            'S2': S2.tolist(),
            'Sw': Sw.tolist(),
            'w': w.tolist(),
            'theta': theta,
            'L': {'m': m, 'c': c}
            #`y = ${L.m.toFixed(2)}x + ${L.c.toFixed(2)}`;
        }
        web.header('Content-Type', 'application/json')
        return json.dumps(result)
```

工作汇报

多种计算模型

判别分析，回归分析，机器学习

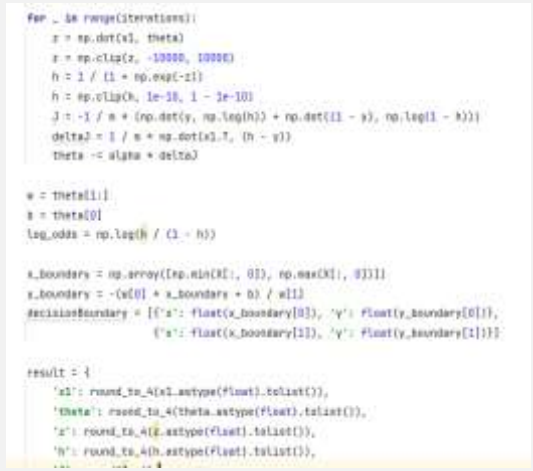
```
class Calculate:
    def POST(self):
        data = json.loads(web.data())
        coordinates1 = np.array(data['coordinates1'])
        coordinates2 = np.array(data['coordinates2'])

        u1 = np.mean(coordinates1, axis=0)
        u2 = np.mean(coordinates2, axis=0)

        S1 = np.cov(coordinates1, rowvar=False)
        S2 = np.cov(coordinates2, rowvar=False)
        Sw = S1 + S2

        reg_lambda = 1e-5
        Sw_reg = Sw + np.eye(Sw.shape[0]) * reg_lambda

        w = np.linalg.inv(Sw_reg).dot(u1 - u2)

        theta = -0.5 * np.dot(w.T, (u1 + u2))

        m = -w[0] / w[1]
        c = -theta / w[1]

        result = {
            'u1': u1.tolist(),
            'u2': u2.tolist(),
            'S1': S1.tolist(),
            'S2': S2.tolist(),
            'Sw': Sw.tolist(),
            'w': w.tolist(),
            'theta': theta,
            'L': {'m': m, 'c': c}
            # y = ${l.m.toFixed(2)}x + ${l.c.toFixed(2)}
        }
        web.header('Content-Type', 'application/json')
        return json.dumps(result)
```

Fisher判别分析代码节选

多元线性回归分析代码节选

机器学习模型代码节选

一元线性回归分析代码节选



逻辑回归分析代码节选

DATA COMPUTING VISUALIZATION ANALYSIS STATISTICS

WORK

总结

专业领域+程序设计=实际应用创新

感想：

设计扩展更多数据可视化和数学公式展示模块

引入更多高级的数据分析方法学习模型。

帮助学习统计学课程，提高计算效率，学习效率！

DATA COMPUTING VISUALIZATION ANALYSIS STATISTICS

中国大学生计算机设计大赛

THANK YOU

演示ppt

面向统计学的计算分析过程可视化平台

A platform for visualizing the statistical calculation and analysis process.

WORK