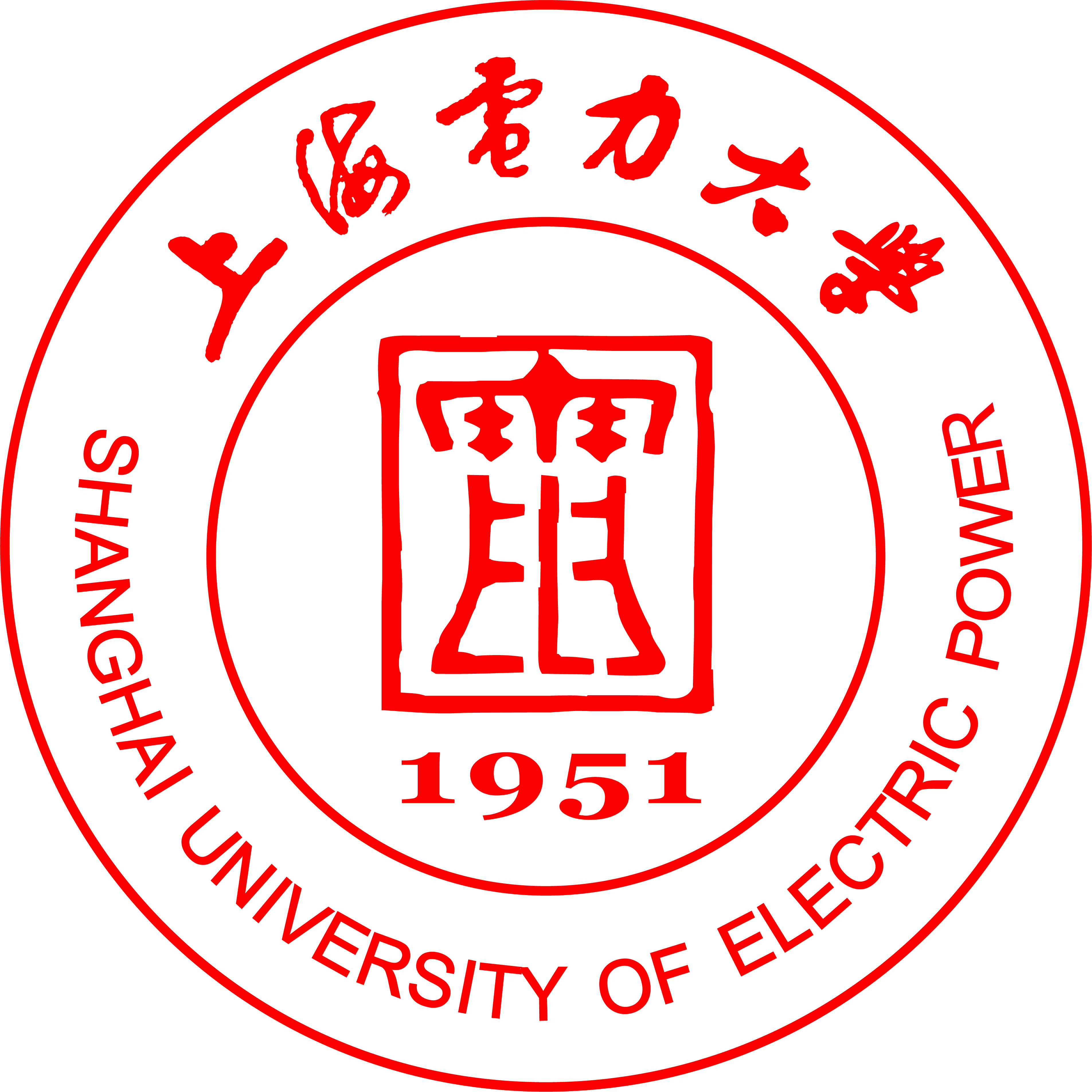
**上海电力大学**

**实践课程报告**



学 院： 数理学院

专 业： 信息与计算科学专业

课程编号： 2812097.01 课程名称： 数据分析实验

学生姓名： 刘伟涛 学号： 20222421 班级： 2022122

指导老师： 于娜

2025 年 1 月 10 日

成绩：

|  |
| --- |
| 教师评语： |

## 关于…的回归与分类问题分析

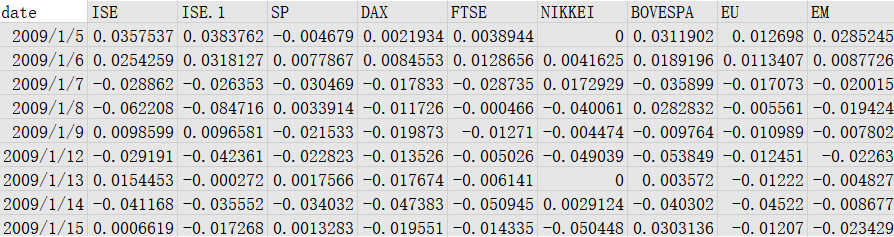
## 回归分析

1、数据来源与说明

数据下载地址：[ISTANBUL STOCK EXCHANGE - UCI Machine Learning Repository](https://archive.ics.uci.edu/dataset/247/istanbul+stock+exchange)

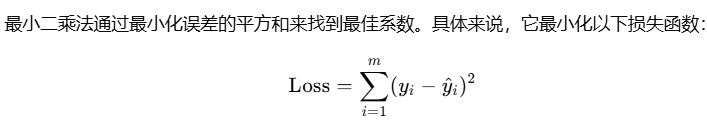
数据内容是伊斯坦布尔证券交易所的回报数据，以及七个国际指数的回报数据：标准普尔（SP）、德国DAX、英国富时100（FTSE）、日本日经225（NIKKEI）、巴西博维斯帕（BOVESPA）、MSCI欧洲（EU）、MSCI新兴市场（EM）。数据收集时间范围为2009年6月5日至2011年2月22日。

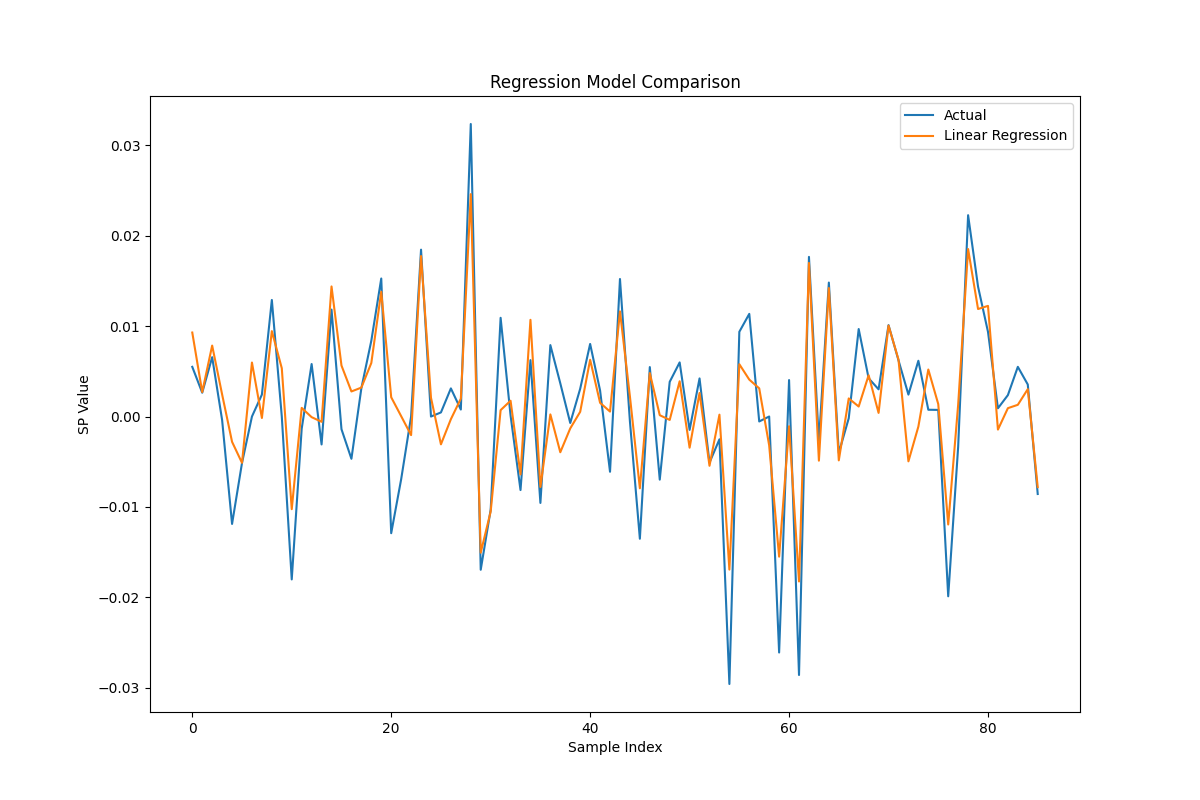
截取前十数据展示：



2、回归分析

（1）线性回归





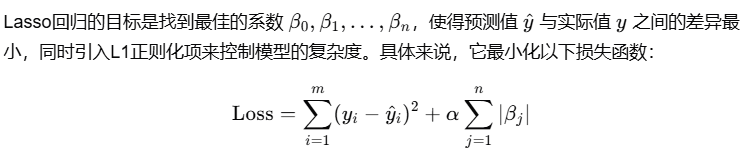
代码：

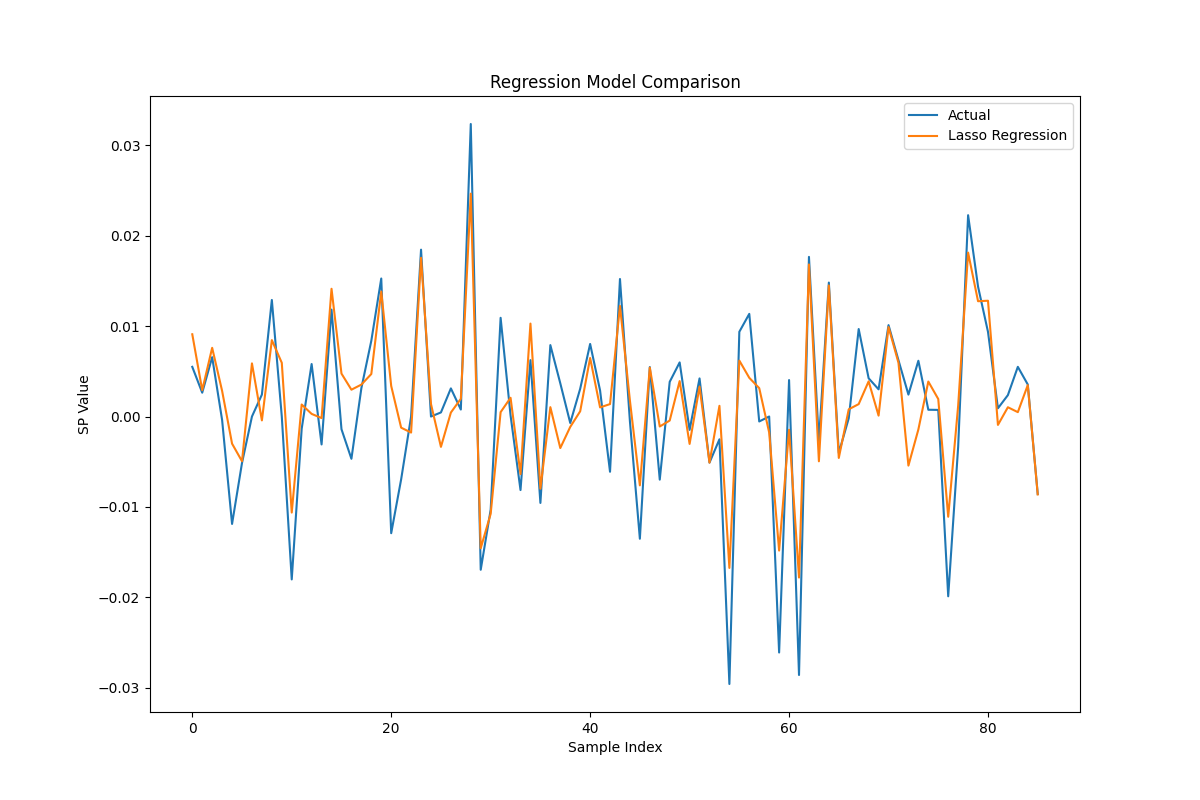
linear\_reg = LinearRegression()

linear\_reg.fit(X\_train, y\_train)

y\_pred\_linear = linear\_reg.predict(X\_test)

1. Lasso回归





代码：

from sklearn.linear\_model import LassoCV

lasso\_cv = LassoCV(alphas=[0.0001, 0.001, 0.01, 0.1, 1, 10], cv=5, random\_state=42)

lasso\_cv.fit(X\_train, y\_train)

print(f"Best alpha for Lasso: {lasso\_cv.alpha\_}")

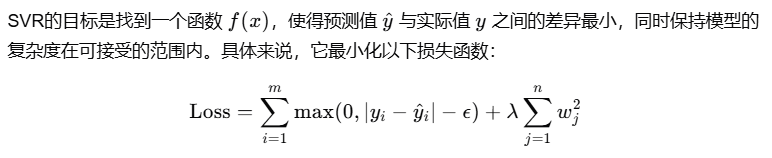
feature\_importance = pd.Series(lasso\_cv.coef\_, index=features)

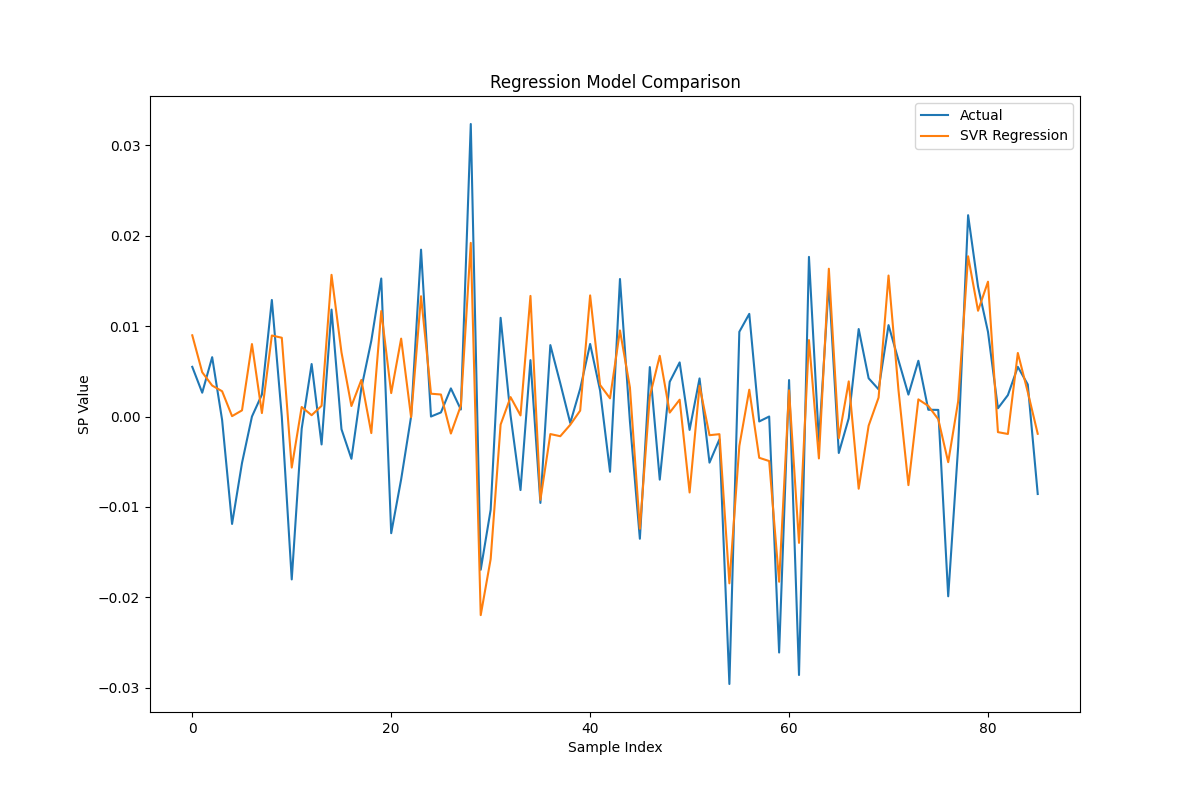
print("Feature Importance:")

print(feature\_importance.sort\_values(ascending=False))

y\_pred\_lasso = lasso\_cv.predict(X\_test)

1. SVR回归





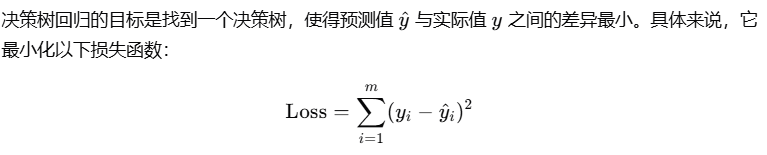
代码：

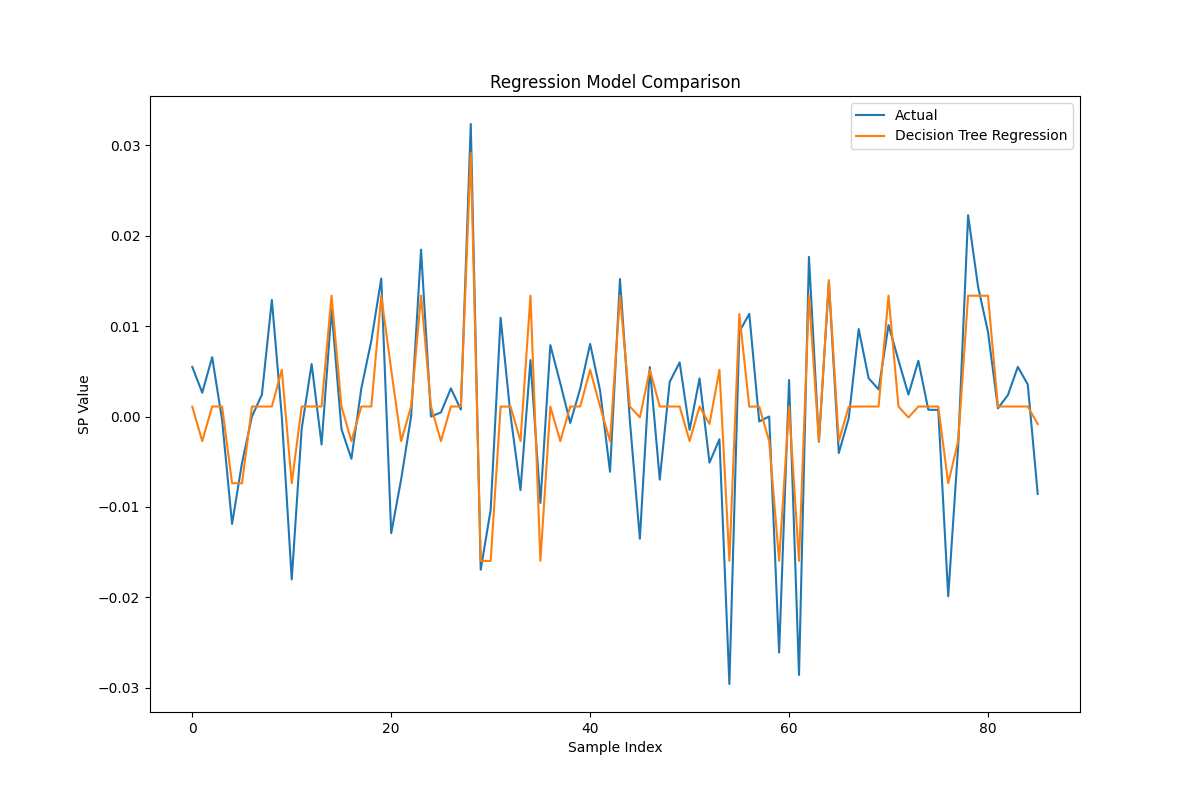
svr\_reg = SVR(kernel='rbf', C=10.0, epsilon=0.01)

svr\_reg.fit(X\_train, y\_train)

y\_pred\_svr = svr\_reg.predict(X\_test)

1. 决策树回归





代码：

tree\_reg = DecisionTreeRegressor(max\_depth=5, random\_state=42)

tree\_reg.fit(X\_train, y\_train)

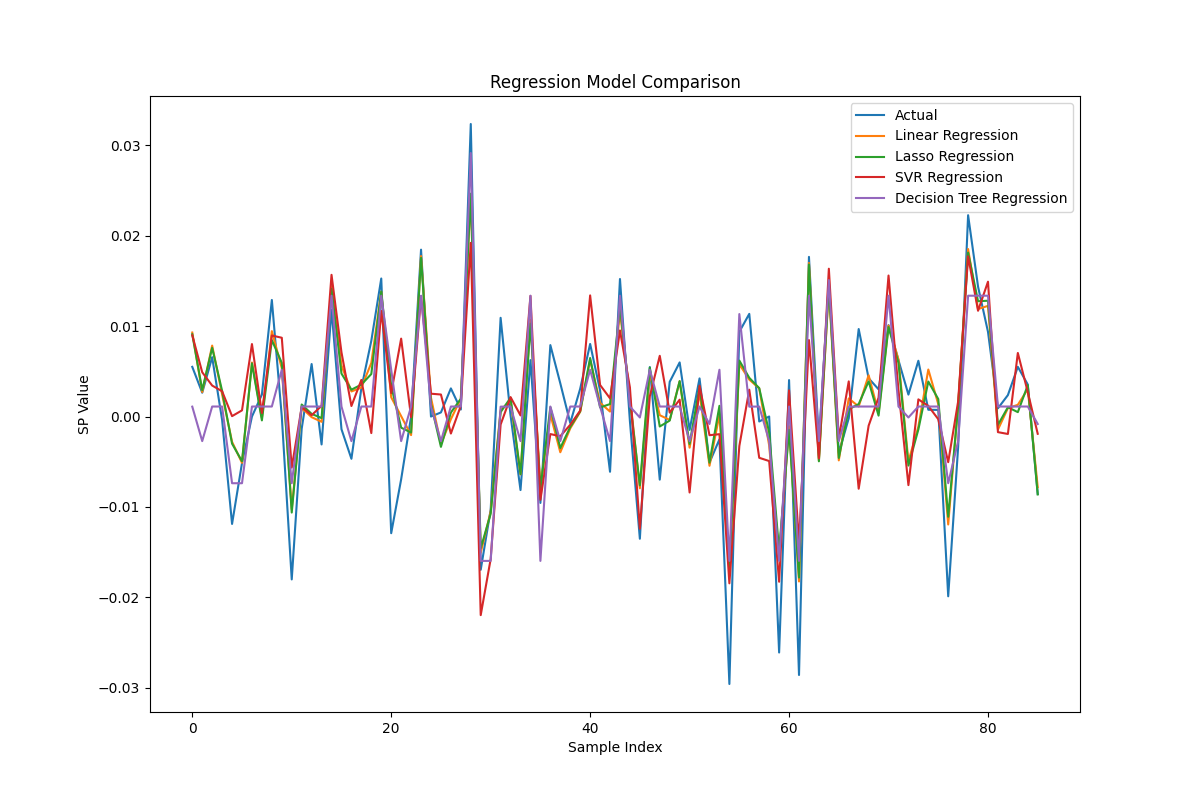
y\_pred\_tree = tree\_reg.predict(X\_test)

1. 回归结果与分析

模型偏差：线性回归和Lasso回归的曲线在某些区间内与实际值较为接近，这表明它们在这些区间的偏差较小。而SVR回归和决策树回归在某些点上与实际值有较大偏差，这可能是由于模型对数据的某些特征过于敏感或不够敏感。

模型稳定性：决策树回归（紫色）在某些区间内波动较大，这可能是由于决策树模型对数据的噪声较为敏感，导致模型在训练过程中出现过拟合。

模型复杂度：SVR回归（红色）在某些区间内与实际值的偏差较大，可能是由于核函数和正则化参数的选择不当，导致模型复杂度不足以捕捉数据的复杂性。



四种回归分析中，没有单一模型在所有区间都表现最佳。这表明不同的回归模型可能适用于数据的不同特征或在不同的预测任务中表现更好。

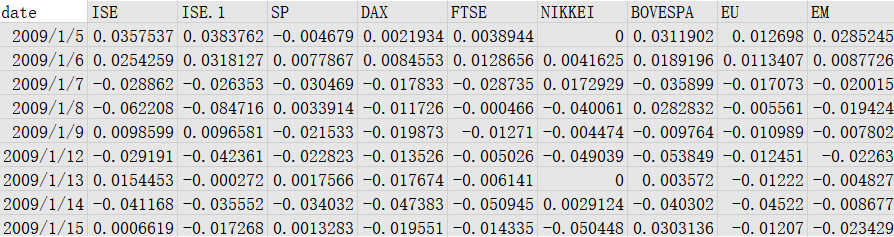
## 二、 分类分析

1、数据来源与说明

数据下载地址：[ISTANBUL STOCK EXCHANGE - UCI Machine Learning Repository](https://archive.ics.uci.edu/dataset/247/istanbul+stock+exchange)

数据内容是伊斯坦布尔证券交易所的回报数据，以及七个国际指数的回报数据：标准普尔（SP）、德国DAX、英国富时100（FTSE）、日本日经225（NIKKEI）、巴西博维斯帕（BOVESPA）、MSCI欧洲（EU）、MSCI新兴市场（EM）。数据收集时间范围为2009年6月5日至2011年2月22日。

截取前十数据展示：



分类分析中选取SP组作为数据集。

1. 分类分析

评估方式介绍：

混淆矩阵（Confusion Matrix）：

用于评估分类模型性能的一种工具，它通过展示模型预测结果与实际标签之间的关系，帮助我们理解模型在不同类别上的表现。混淆矩阵通常用于二分类问题，但也可以扩展到多分类问题。

对于一个二分类问题，混淆矩阵是一个2x2的矩阵，其元素定义如下：

真阳性（True Positive, TP）：模型正确预测为正类的样本数。

假阳性（False Positive, FP）：模型错误预测为正类的样本数。

真阴性（True Negative, TN）：模型正确预测为负类的样本数。

假阴性（False Negative, FN）：模型错误预测为负类的样本数。

|  |  |  |
| --- | --- | --- |
| 真实标签\预测标签 | 正类 | 负类 |
| 正类 | TP | FN |
| 负类 | FP | TN |

精确率（Precision）：

定义：精确率是指在所有被模型预测为正类的样本中，实际为正类的比例。

计算公式：

意义：高精确率意味着模型预测为正类的样本大多数是正确的，但并不保证所有实际的正类都被正确识别。

召回率（Recall）：

定义：召回率是指在所有实际为正类的样本中，被模型正确预测为正类的比例。

计算公式：

意义：高召回率意味着模型能够识别出大多数的实际正类样本，但可能会有较多的误报（FP）。

F1分数（F1 Score）：

定义：F1分数是精确率和召回率的调和平均数，它试图在精确率和召回率之间找到一个平衡。

计算公式：

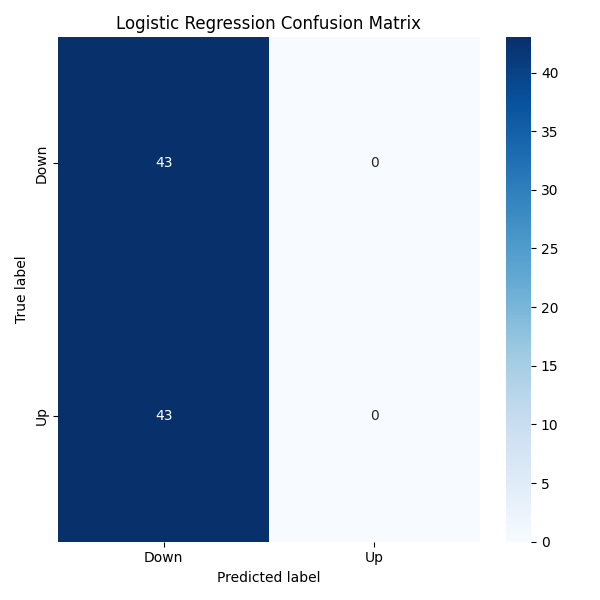
意义：F1分数是一个综合指标，当模型的精确率和召回率差距较大时，F1分数会降低，这有助于识别模型在这两个指标上的不平衡。

支持率（Support）：

定义：支持率是指每个类别在测试集中出现的次数。

意义：支持率提供了每个类别样本数量的上下文信息，有助于理解模型在不同类别上的性能。

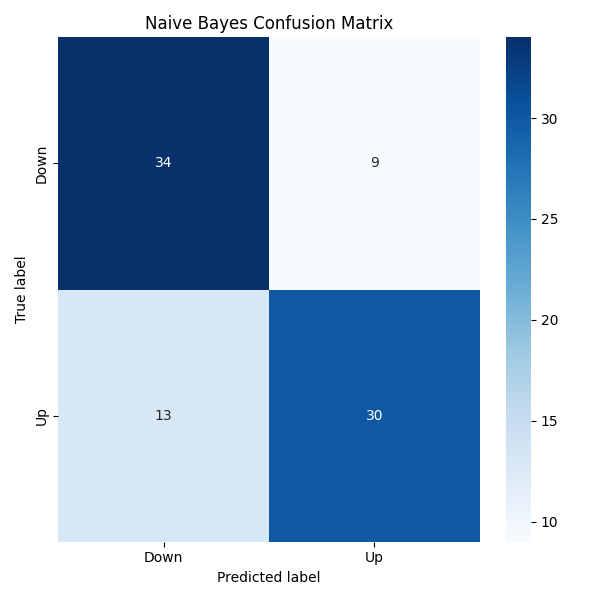
1. 逻辑回归



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 精确率 | 召回率 | F1分数 | 支持率 |
| 0 | 0.5 | 1 | 0.67 | 43 |
| 1 | 0 | 0 | 0 | 43 |
| 宏平均 | 0.25 | 0.5 | 0.33 | 86 |
| 加权平均 | 0.25 | 0.5 | 0.33 | 86 |

总体准确度：50%

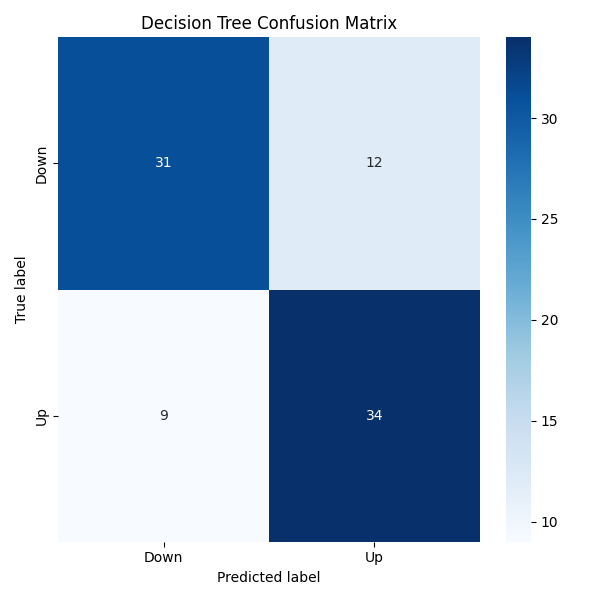
1. 朴素贝叶斯分类



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 精确率 | 召回率 | F1分数 | 支持率 |
| 0 | 0.72 | 0.79 | 0.76 | 43 |
| 1 | 0.77 | 0.7 | 0.73 | 43 |
| 宏平均 | 0.75 | 0.74 | 0.74 | 86 |
| 加权平均 | 0.75 | 0.74 | 0.74 | 86 |

总体准确度：74.41%

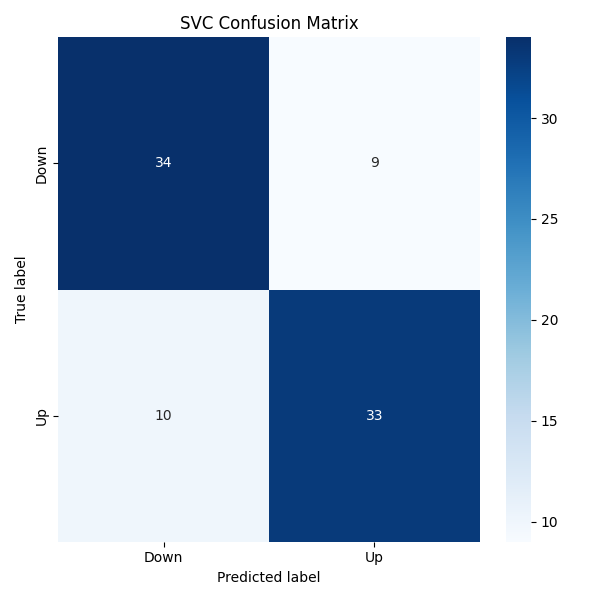
1. 决策树分类



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 精确率 | 召回率 | F1分数 | 支持率 |
| 0 | 0.79 | 0.77 | 0.78 | 43 |
| 1 | 0.77 | 0.79 | 0.78 | 43 |
| 宏平均 | 0.78 | 0.78 | 0.78 | 86 |
| 加权平均 | 0.78 | 0.78 | 0.78 | 86 |

总体准确度：77.9%

1. SVC分类



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 精确率 | 召回率 | F1分数 | 支持率 |
| 0 | 0.77 | 0.79 | 0.78 | 43 |
| 1 | 0.79 | 0.77 | 0.78 | 43 |
| 宏平均 | 0.78 | 0.78 | 0.78 | 86 |
| 加权平均 | 0.78 | 0.78 | 0.78 | 86 |

总体准确度：77.9%

1. 分类结果与分析

首先将四个机器学习模型在二分类任务上的性能指标保存为classification\_results.csv文件

用可视化的方法将四种四种方法准确度作比较：



逻辑回归在类别1上的表现非常差，导致总体准确度只有50%。

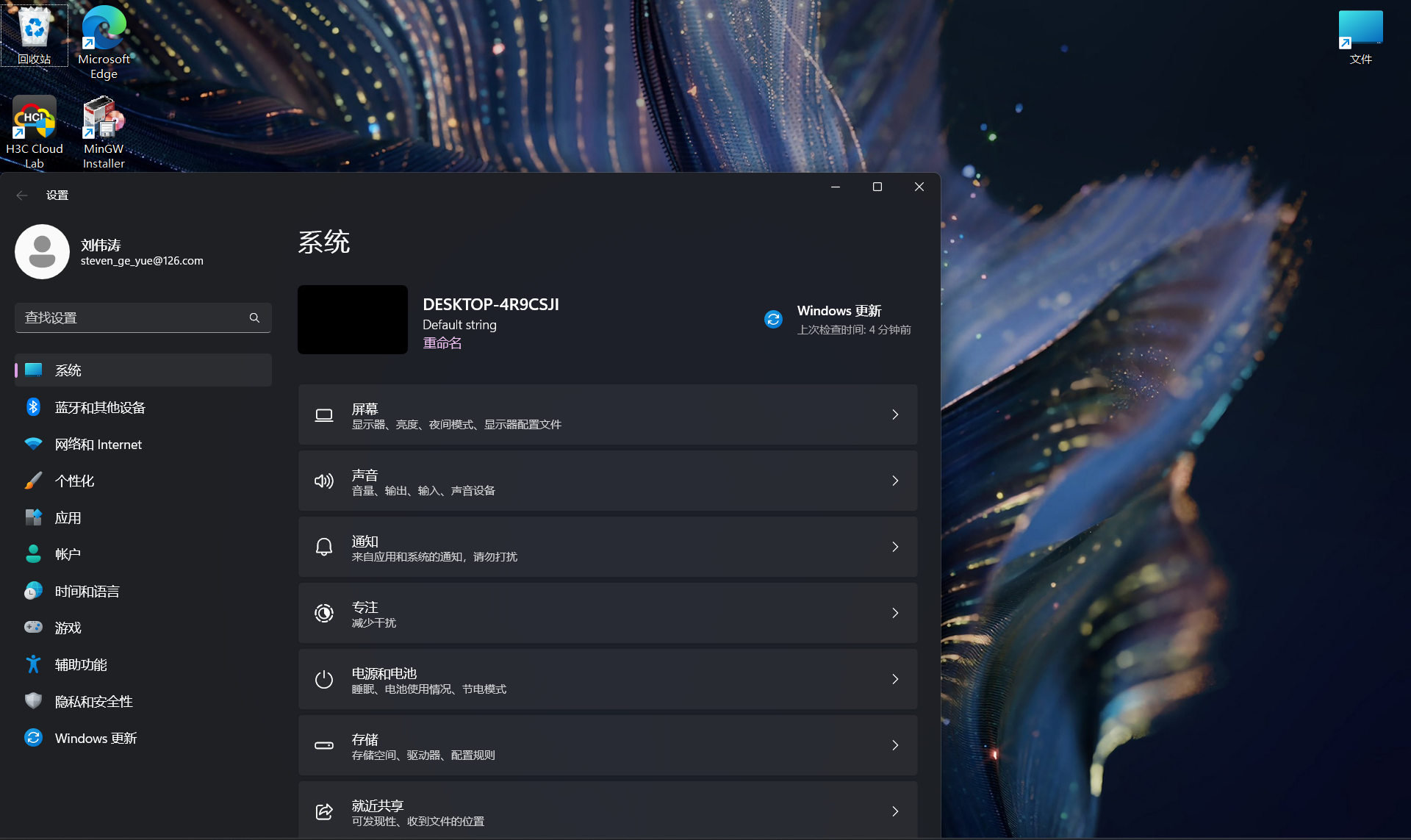
朴素贝叶斯和SVC在两个类别上的表现相对均衡，总体准确度分别为74.41%和77.9%。

决策树和SVC在两个类别上的表现非常接近，总体准确度都是77.9%，且在精确率、召回率和F1分数上都表现良好。

在这些模型中，决策树和SVC是最优选择，因为它们在所有性能指标上都表现均衡，且总体准确度最高。

## 三、图形处理

原图：



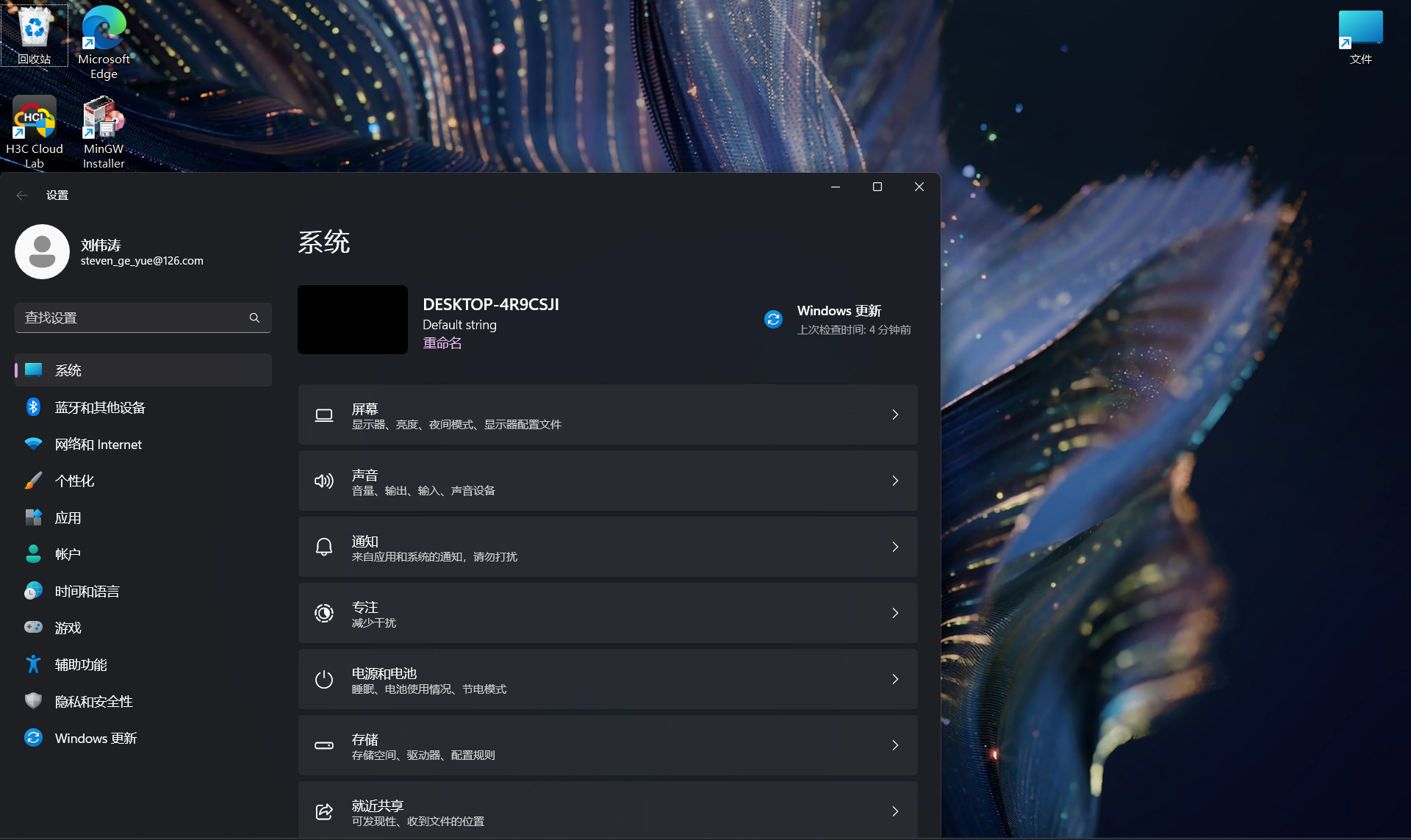
尺寸: 1920x1143 pixels

大小: 1365405 bytes

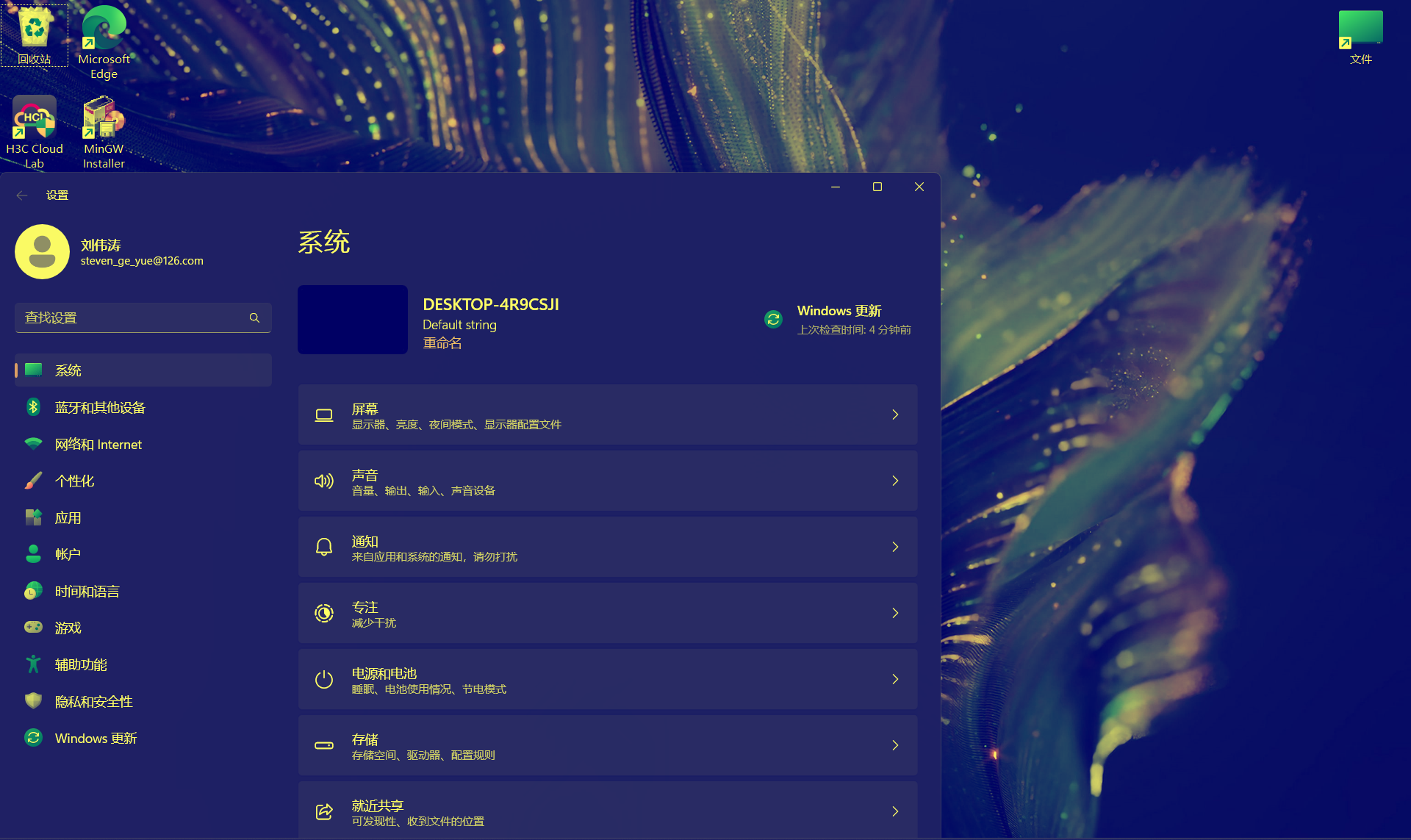
压缩结果:

压缩尺寸: 1160094 bytes

压缩后：

 压缩率：85%

截取中间区域：

 将蓝色通道设置为100：

部分代码：

    img\_array\_blue = img\_array.copy()

    img\_array\_blue[:, :, 2] = 100

    # 保存修改后的图像

    Image.fromarray(img\_array\_blue).save('blue\_channel\_modified.png')

    original\_shape = img\_array.shape

    img\_flat = img\_array.reshape(-1, original\_shape[2])

    pca = PCA(n\_components=3)

    compressed = pca.fit\_transform(img\_flat)

    reconstructed = pca.inverse\_transform(compressed)

## 四、 感想与体会

在完成这次作业的过程中，我深刻体会到了理论与实践相结合的重要性。通过比较逻辑回归、朴素贝叶斯、决策树和SVC等不同分类算法的性能，我学会了如何从多个维度评估模型，包括精确率、召回率、F1分数和支持率，这让我意识到了没有完美的模型，每个模型都有其优势和局限性。在编程实现和参数调优的过程中，我的技能得到了锻炼和提升，同时我也学会了如何查找资料、分析问题和找到解决方案。而且还学会了用AI助手提供信息、解释概念、解答问题、推荐资源。这是一次很有意义的课程。