同济大学

机器学习线性回归实验报告

1 实验综述

1.1 实验说明

成员名单及分工（按学号排序）：

学号：2151531　姓名：栾佳浩　任务：搭建应用线性回归模型

学号：2152486　姓名：刘翌帆　任务：数据预处理，撰写报告

学号：2152496　姓名：郭桉齐　任务：手写最小二乘法，研究正则化参数、数据升降维

学号：2154312　姓名：郑博远　任务：手写梯度下降法，研究学习率，撰写报告

装

订

线

1.2 实验目标

本次实验旨在探索和应用线性回归模型。我们使用 kaggle 的"Calculate Concrete Strength"数据集作为实验基础，该数据集包括水泥（Cement）、高炉矿渣（Blast Furnace Slag）、粉煤灰（Fly Ash）、水（Water）、高效减水剂（Super-plasticizer）、粗骨料（Coarse Aggregate）、细骨料（Fine Aggregate）以及养护龄期（Age）多个特征，以及与之对应的水泥强度（Strength）目标值。

在本次实验实验中，我们的首要目标是建立线性回归模型，通过学习特征与水泥强度之间的关系进行预测。同时，我们希望深入了解数据预处理的重要性，包括如何处理缺失值、进行特征标准化等，通过上述操作提高线性回归模型的准确性。通过实验，我们希望掌握线性回归模型的构建和评估方法，并通过模型的训练和优化来提高对混凝土强度的预测准确性，从而加深对机器学习中线性回归任务基本概念的理解。

1.3 实验数据集

本实验使用的数据集来自于 kaggle 数据库的 Calculate Concrete Strength。该数据集的主要目标是预测混凝土的强度，其特征涵盖了混凝土制备过程中使用的各种成分，包括水泥（Cement）、高炉矿渣（Blast Furnace Slag）、粉煤灰（Fly Ash）、水（Water）、高效减水剂（Super-plasticizer）、粗骨料（Coarse Aggregate）、细骨料（Fine Aggregate）以及养护龄期（Age）。

通过深入研究这些特征与混凝土强度之间的关系，我们可以构建一个线性回归模型，从而预测混凝土的强度。这不仅在工程和建筑领域具有实际应用价值，还为学习者提供了一个与生活实际相贴合的数据集，以探索深度学习和回归分析的基本概念。

2 实验报告设计

2.1 数据准备

本实验使用的数据集来自于 kaggle 数据库的 Calculate Concrete Strength（https://www.kaggle.com/datasets/prathamtripathi/regression-with-neural-networking/data）。该数据集包括八个关键特征，每个特征在预测混凝土强度方面都起着重要作用。这些特征包括水泥（Cement）、高炉矿渣（Blast Furnace Slag）、粉煤灰（Fly Ash）、水（Water）、高效减水剂（Super-plasticizer）、粗骨料（Coarse Aggregate）、细骨料（Fine Aggregate）以及养护龄期（Age）。目标变量则是水泥强度（Strength），这是表征混凝土质量和耐久性的基本参数。数据集包含共 1030 条数据，其中混凝土的养护龄期特征为整型，其余特征均为浮点型。

2.2 数据预处理

数据预处理是数据分析和机器学习中的关键步骤，旨在使数据集更适合模型训练和分析。在数据预处理阶段，我们对原始数据进行一系列操作，以确保数据质量和适应模型的需求。

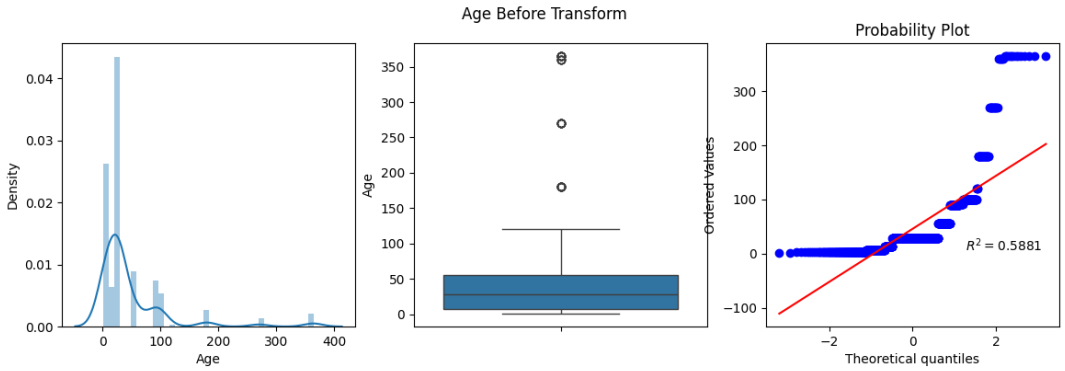
2.2.1 检查缺失值

当数据表格中存在缺失值时，我们采取了一种简单的策略，即将缺失值用各列的平均值来填充。缺失值填充有助于保持数据的完整性，确保模型训练不会受到缺失数据的干扰。

2.2.2 探索性数据分析（EDA）与数据转换

在探索性数据分析（EDA）的阶段，我们致力于了解数据的特征分布，以便更好地准备数据进行建模。对于所有的八个特征列（ 'Cement'、'Blast Furnace Slag'、'Water'、'Superplasticizer'、'Fine Aggregate'、'Age'、'Strength'、'Fly Ash' 和 'Coarse Aggregate'），我们执行了一系列不同的数据变换，探究数据变换后的特征分布变化。

通过编写自定义函数 apply_transform，绘制每个特征的分布图。该函数在应用数据变换之前和之后绘制了直方图、箱线图和概率图，以便直观地观察特征的分布变化。我们执行了四种不同类型的数据变换，分别是 Power 转换、Log 转换、Sqrt 转换和平方转换。这个分析帮助我们确定哪些特征列需要进行哪种特征变换。下图以'Age'特征为例，显示了在 Sqrt 转换前后的分布：



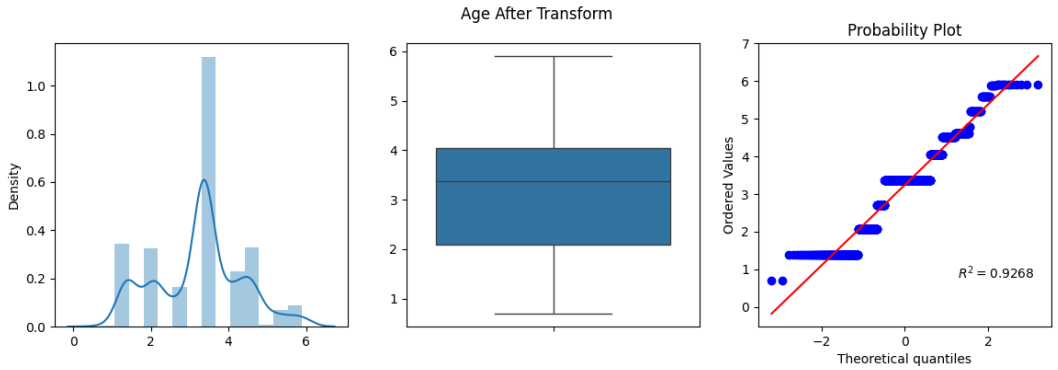


图 1 'Age'特征转换前后分布图

通过分析各个特征列的直方图、箱线图和 Probability Plot 概率图，我们选择对不同的特征列执行不同的特征变换，使得数据分布能够更符合正态分布，从而使得线性关系更容易捕捉，模型的拟合效果更好。最终通过分析，我们对以下特征列执行了以下特征变换：

- 'Cement'、'Superplasticizer'、'Water' 和 'Coarse Aggregate' 分别进行 Power 转换；
- 'Age' 列执行了 Log 转换；
- 'Blast Furnace Slag' 和 'Fly Ash' 分别进行了 Sqrt 转换。

通过这些 EDA 和特征转换步骤，我们为建模准备了经过分析和适当转换的特征，以更好地捕获数据中的模式和关系，从而提高建模性能。

2.2.3 数据规范化（标准化）

数据规范化是为了确保不同特征的值处于相似的尺度范围内，避免模型受到特征值范围的影响。在这个步骤中，我们使用了标准化方法，通过减去均值并除以标准差，将数据标准化为均值为 0，标准差为 1 的分布，有助于提高模型的稳定性和性能。

在这一部分中，代码使用了标准化方法，通过 StandardScaler 类创建了一个标准化的变换器，然后将数值列进行了标准化处理，将它们的值缩放到均值为 0，标准差为 1 的范围内，有助于消除不同特征之间的尺度差异，以确保模型训练的稳定性和性能。

2.2.4 Pearson 相关系数

本实验使用 Pearson 相关系数来考量不同特征之间的线性相关性，可以看到目标特征 Strength（混凝土强度） 与 养护龄期（Age）和高效减水剂（Super-plasticizer）相关度较高，而与其他特征相关性都处于略低的水平。

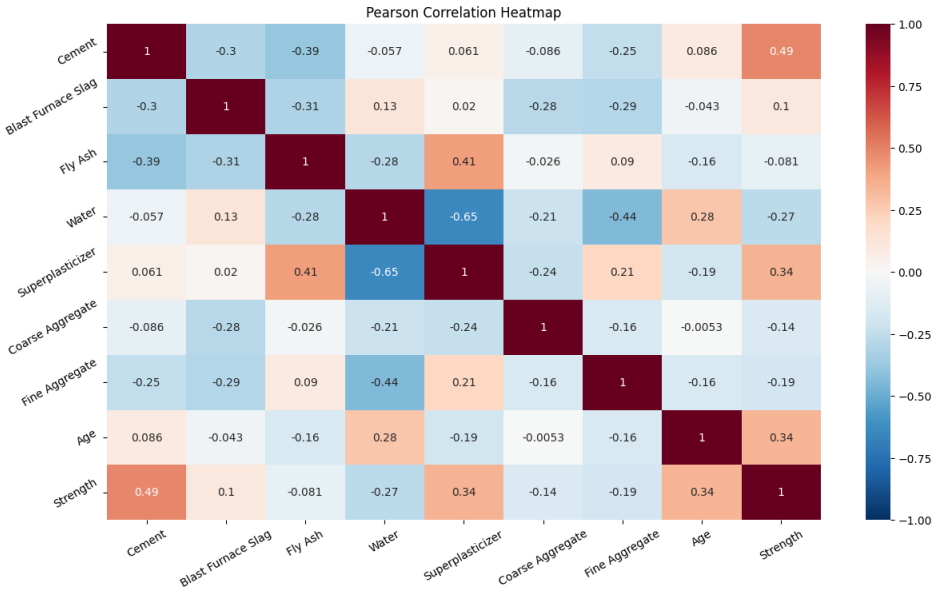


图 2 各特征之间 Pearson 相关系数热力图

2.3 模型搭建

2.3.1 手写实现批量梯度下降法

在本模块中，我们通过手写实现梯度下降法（包括可选的 AdaGrad 优化方法）来训练线性回归模型，以拟合训练数据。在定义的 LinearRegression 类中，我们定义了一个名为`_train_for_one_batch`的方法，它的任务是计算损失函数的梯度，并使用学习率来更新模型参数。其中，梯度表示损失函数对模型参数的偏导数，而学习率则决定了每次参数更新的步长。在迭代训练的过程中，通过多次调用`_train_for_one_batch`方法，每次从数据中随机选择一批样本进行小批量训练。这有助于模型逐渐收敛到最优参数，以最小化损失函数。

我们定义了 LinearRegression 类，初始化时可以指定特征个数、批次大小、学习率等超参数。为了与 sklearn 中的 LinearRegression 统一，自定义类包含了以下主要方法和功能：

- fit: 用于训练线性回归模型。在这个方法中，我们将数据转换成 NumPy 数组，并通过多次迭代来不断调整模型参数以最小化损失函数。
- predict: 用于在训练集上预测回归值。这个方法可以根据训练好的模型参数对新数据进行回归预测。
- _train_for_one_batch: 用于执行一次小批量训练的内部方法。在这个方法中，我们计算损失函数的梯度并根据学习率更新模型参数。

此外，我们还实现了一个 Parameter 类，用于管理线性回归模型的参数。这个类包括参数的初

始化和更新方法，以及参数的存储和处理。

在手写实现梯度下降法的模型中，我们还加入了 AdaGrad 优化方法的选项，该方法会根据 AdaGrad 算法来自适应调整学习率。这种自适应学习率的调整可以帮助更精细地调整参数，以提高训练的效率和性能。AdaGrad 优化后的性能对比将在后文中详细介绍。

通过手写实现 LinearRegression 的梯度下降类，我们能够自定义调整批量梯度下降的参数（包括学习率和优化方法），以便更好地适应不同的数据和任务需求，从而为我们的后续实验训练线性回归模型提供了更多可控性与灵活性。

2.3.2 手写实现最小二乘法

在本模块中，我们将介绍如何手写实现最小二乘法梯度下降的线性回归模型。这个模型可以用于拟合数据并估计包括截距项在内的线性关系回归系数。在手写的最小二乘法模型文件中，我们定义了 LinearRegression 类。为了与 sklearn 中的 LinearRegression 统一，自定义类包含了以下主要方法和功能：

- fit：这个方法用于拟合线性回归模型。在这之前，我们需要将特征矩阵 `X` 的第一列添加一列全为 1 的列，以处理截距。然后，我们使用最小二乘法来计算回归系数。这涉及到以下步骤：
    1. 计算特征矩阵 `X` 的转置；
    2. 使用矩阵运算来计算回归系数；
    3. 提取截距和系数，将它们存储在 `self.intercept_` 和 `self.coef_` 中。
- predict：这个方法用于进行预测。与拟合时一样，我们需要在特征矩阵 `X` 的第一列添加一列全为 1 的列，以处理截距。然后，我们使用拟合得到的系数进行预测，并返回预测结果。

该部分手写实现的线性回归模型能够通过最小二乘法来拟合数据，估计回归系数和截距项，并用于进行预测。通过实现手写最小二乘法，我们更好地理解了线性回归模型的工作原理，并能够根据数据估计回归系数，以便之后对混凝土数据集进行回归预测。

2.3.3 模型应用流程

在加载数据集并并执行数据预处理后，我们进行了数据的分割。将读入到的 DataFrame 划分为特征矩阵（X）和目标变量（y）。接下来，我们将数据集划分为训练集和测试集。这一步骤对于模型的评估至关重要。我们将 25%的数据作为测试集，以验证模型的泛化性能。

在选择了要使用的特征列之后，我们对这些特征进行了列变换。其中包括 PowerTransformer、log 变换和 sqrt 变换等操作，以使特征更适合满足线性模型的假设（具体操作见前文数据预处理部分介绍）。该转换过程通过 Scikit-Learn 的 ColumnTransformer 实现。随后，我们对特征进行了

标准化，使用了 StandardScaler，以确保不同特征具有相同的尺度。这有助于模型更好地拟合数据和提高模型性能。

通过上述步骤，我们准备好用于训练和评估的最终数据，将其存储在 final_train 和 final_test 中。这些数据经过特征选择、变换和标准化，已经准备好用于建模和评估。在使用不同的线性回归模型（最小二乘法、梯度下降法）进行模型训练后，在测试集上验证回归效果。

2.4 模型训练测试

表 1 训练测试结果

R-Square	训练集	测试集
库函数最小二乘法（未进行数据转换）	0.5997	0.6047
库函数梯度下降法	0.8110	0.8144
手写最小二乘法	0.8112	0.8152
手写梯度下降法	0.8107	0.8147

根据测试结果表格对比，可以明显发现未经过数据转换的数据明显回归效果更差。在进行数据变换后的数据集上尝试不同实现方式的各种模型，可以看到几种方法在训练集和测试集上的表现类似，模型泛化能力强，并没有出现过拟合。梯度下降回归方式的结果未与最小二乘回归方式有明显差距，R-Square 分数仅略低于手写的最小二乘法，这可能由于梯度下降回归方式得到的是局部最优解。在 2.6 章节中，将继续讨论学习率在内的因素对回归效果的影响。

2.5 结果可视化

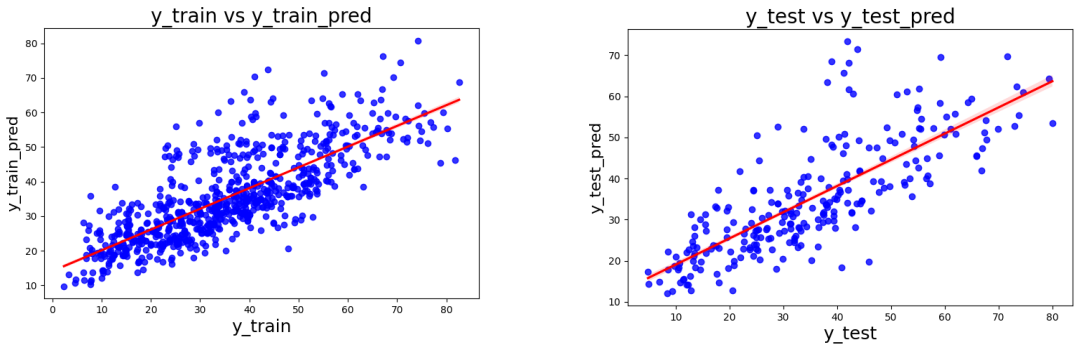


图 3 库函数最小二乘法（未进行数据转换）回归效果图

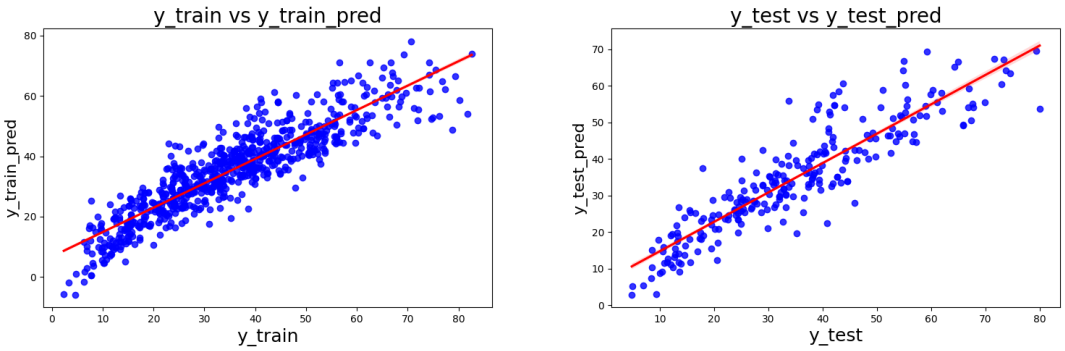


图 4 库函数梯度下降法回归效果图

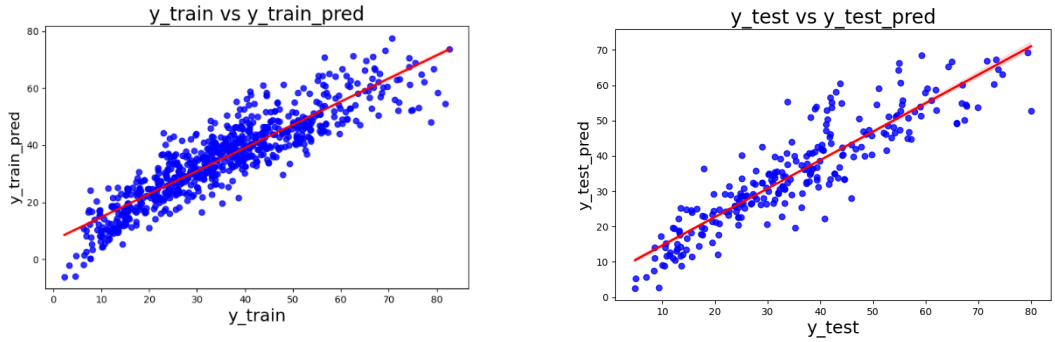


图 5 手写最小二乘法回归效果图

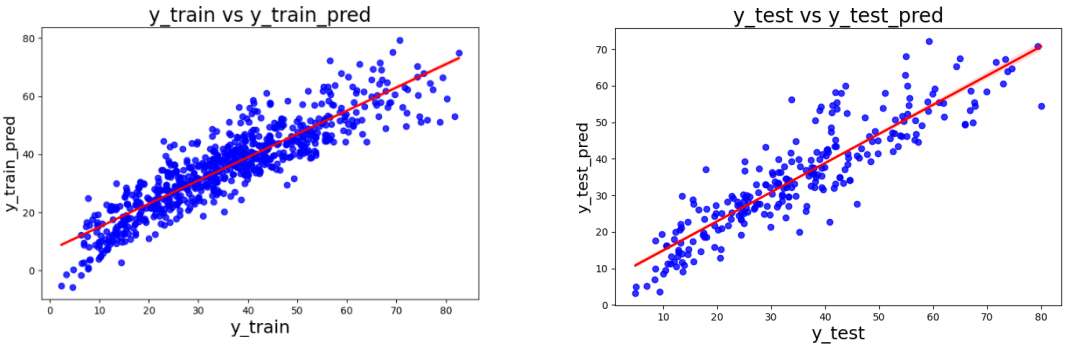


图 6 手写梯度下降法回归效果图

2.6 分析和优化

2.6.1 正则化参数

在本次实验的分析优化环节中，我们使用了 l1 和 l2 正则化参数来对线性回归模型进行优化。对比结果显示，这些正则化参数对结果的影响较小的，尽管可能在某些情况下能够略微提升模型的性能，但总体趋势显示，R-Square 分数随正则化参数的增加而下降。

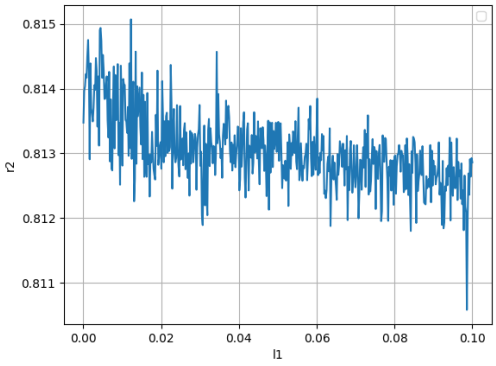


图 7 R-Square 随正则化参数 l1 变化图

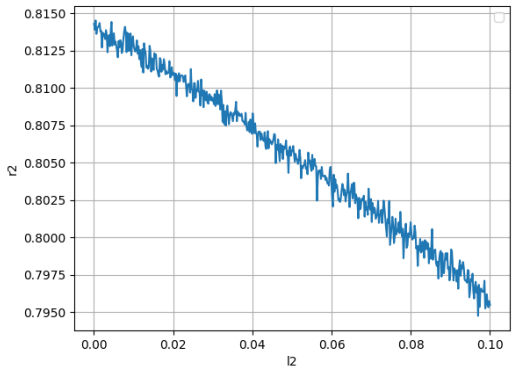


图 8 R-Square 随正则化参数 l2 变化图

这一结果显示，尽管正则化参数有时会对模型产生积极影响，但并不是所有情况都需要进行正则化。在本次情况下，模型已经足够简单，不需要额外的正则化来限制其复杂性。

2.6.2 数据转换对比

通过下列图片对比，由 R-Square 数值进行分析，当我们选择合适的 transform 方式时，可以使 R-Square 值从 0.6 左右上升到 0.8 左右，对于回归效果具有显著的提升作用。

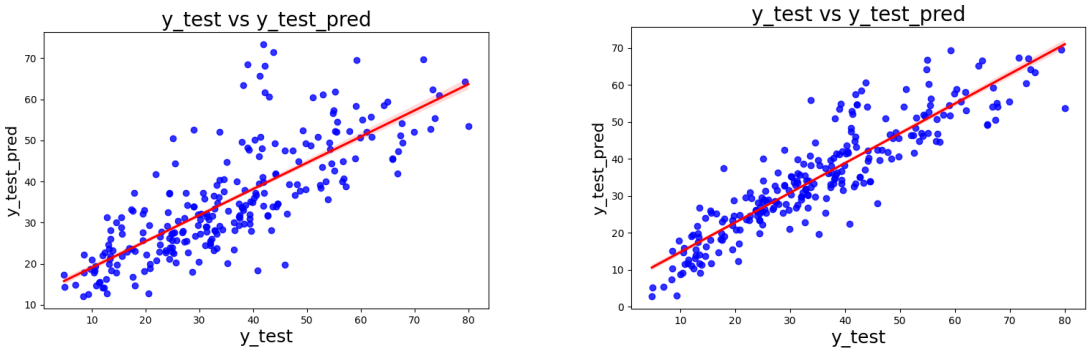


图 9 数据转换前（图左）后（图右）回归预测效果对比图

2.6.3 PCA 降维对比

PCA 通常情况下可以用于降低特征维度，提高训练速度。但是根据实验结果对比，PCA 降维使得结果 R-Square 产生小幅度下降，因此我们在本实验中选择不降维。

2.6.4 学习率分析与 Adagrad 优化

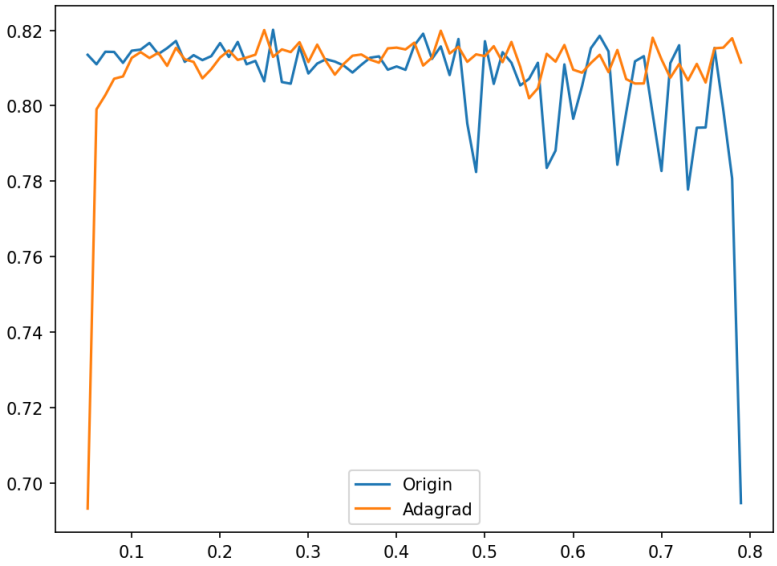


图 10 有无 Adagrad 优化 R-Square 随学习率的变化曲线

根据上图不难发现，在不使用 Adagrad 优化时，随着学习率的增加，R-Square 的震荡越发明显，训练效果趋于不稳定；由此可得，将学习率设置为 0.1 左右就已经能够达到比较理想的回归效果。在使用 Adagrad 优化后，当学习率设置大于 0.1 时，观察到 R-Square 明显较未优化时变化更为平滑稳定。在学习率较大时，Adagrad 能够起到很好的优化效果。

2.6.5 模型优化总结

实验中的模型优化部分主要包括以下几个方面：

（1）特征工程。通过应用不同的特征变换方法，如 PowerTransformer、log 变换和 sqrt 变换，对原始数据进行了特征工程。这有助于使数据更符合线性模型的假设，从而提高模型的性能。通过可视化，还展示了特征变换前后的分布情况，以便更好地理解数据变换的效果。

（2）特征标准化。使用 StandardScaler 对特征进行标准化，确保各个特征具有相似的尺度，从而有助于梯度下降等算法更快地收敛。

（3）PCA 降维。PCA 通常情况下可以用于降低特征维度，提高训练速度。但是根据实验结果对比，PCA 降维使得结果 R-Square 产生小幅度下降，因此我们在本实验中选择不降维。

（4）正则化参数设置。使用了 SGDRegressor 模型，并通过网格搜索不同的正则化参数（l1 和 l2）来调优模型。网格搜索了一系列正则化参数值，然后通过评估指标（如 R-Square 分数）来选择最佳参数。最终发现，不使用正则化参数即可获得比较理想的回归结果。

（5）学习率与 Adagrad 优化。在使用梯度下降法时，通过改变学习率观察回归结果，并对比

有无 Adagrad 优化时的结果，选择更适宜的学习率与优化方式。

综上所述，实验对线性回归模型进行了一系列的优化改进的尝试，包括特征工程、特征选择、特征标准化、正则化参数调优等方面，以提高模型的性能和泛化能力。这些优化步骤的尝试有助于发现更好地解决方式，使模型更适应数据，并提高其在测试集上的表现。

3 总结

本次实验的主要目标是探索和应用线性回归模型，以预测混凝土的强度。我们使用了"Calculate Concrete Strength"数据集，其中包含了多个特征（如水泥、高炉矿渣、粉煤灰、水、高效减水剂、粗骨料、细骨料、养护龄期）以及对应的水泥强度目标值。选定数据及后，我们旨在构建线性回归模型，通过学习特征与水泥强度之间的关系来进行准确的预测。

在数据准备与预处理阶段，我们仔细分析了数据集，处理了缺失值以确保数据的一致性。接下来，我们进行了包括特征工程和标准化在内的数据预处理工作，为接下来模型的训练和分析奠定了良好的基础。随后，我们不仅使用 sklearn 的库函数进行线性回归任务，还手写完成了梯度下降法和最小二乘法两种不同的线性回归模型，并对它们进行了训练和测试。

在结果可视化方面，我们对这两种不同的优化方法进行了详细分析和比较。梯度下降法能够通过不断迭代找寻最优解，适用于较大型的数据集，但对参数初始值敏感，对学习率选择也有所要求，还有可能陷入局部最优解。最小二乘法具有闭式解、适用于小型数据集、简单实现等优点，但对异常值敏感，不适用于大型数据集。

最后，我们通过一系列的模型优化步骤，包括正则化参数、学习率的调整等，提高了模型的性能和泛化能力。总结而言，这次实验不仅实现了混凝土强度的准确预测，还积累了机器学习和回归建模的宝贵经验，有助于更深入地理解机器学习领域的核心概念。通过这个实验，我们不仅获得了更好的预测结果，还在探索和实践的过程中提高了机器学习技能。