1. 模型

2.1 普通最小二乘法

普通最小二乘法（OLS）是一种经典的线性回归方法，通过最小化残差平方和来估计模型参数。OLS假设误差项服从正态分布且具有同方差性。然而，当数据中存在异常值时，OLS的表现可能会受到影响。为了增强模型的鲁棒性，我们使用了Huber回归，它结合了OLS和L1正则化的优点，能够更好地处理异常值。

我们使用scikit-learn库中的HuberRegressor实现Huber回归。通过网格搜索法优化epsilon（Huber损失函数的阈值参数）和alpha（正则化强度）两个超参数。具体步骤如下：

1. 定义参数网格：epsilon和alpha的取值范围；
2. 遍历所有参数组合，训练模型并计算验证集上的R2分数；
3. 选择最优参数组合，并在测试集上进行预测。

伪代码：

for epsilon, alpha in parameter\_grid:

model = HuberRegressor(epsilon, alpha)

model.fit(X\_train, y\_train)

2.2 主成分回归

主成分回归（PCR）通过主成分分析（PCA）将原始特征降维为少数几个主成分，然后在这些主成分上进行线性回归。PCR能够有效处理多重共线性问题，并减少模型的过拟合风险。其核心思想是通过降维保留数据的主要信息，同时去除噪声。

PCR的实现分为两步：主成分分析（PCA）和线性回归。我们使用scikit-learn库中的PCA和LinearRegression类。通过遍历不同的主成分数量，选择最优的主成分个数。

伪代码：

for n\_components in candidate\_values:

pca = PCA(n\_components)

X\_train\_pca = pca.fit\_transform(X\_train)

model = LinearRegression()

model.fit(X\_train\_pca, y\_train)

2.3 偏最小二乘法

偏最小二乘法（PLS）是一种结合了降维和回归的技术。与PCR不同，PLS在降维时同时考虑了自变量和因变量的信息。它通过最大化自变量和因变量之间的协方差来提取主成分，从而在降维的同时保留对因变量的解释能力。

我们使用scikit-learn库中的PLSRegression实现偏最小二乘法。通过遍历不同的主成分数量，选择最优的主成分个数。

伪代码：

for n\_components in candidate\_values:

model = PLSRegression(n\_components)

model.fit(X\_train, y\_train)

2.4 广义线性模型

广义线性模型（GLM）是线性回归的扩展，允许因变量服从非正态分布。GLM通过链接函数将线性预测器与因变量的期望值联系起来。在本报告中，我们实现了一个结合样条基函数和Group Lasso正则化的GLM模型。样条基函数用于捕捉非线性关系，而Group Lasso正则化用于增强模型的稀疏性。具体步骤如下：

1. 生成样条基函数：基于分位数确定节点位置，并构建二阶样条基函数；
2. 定义Group Lasso正则化：对每个特征对应的样条基函数进行分组正则化；
3. 使用Huber损失函数和Group Lasso正则化训练模型。

伪代码：

for num\_knots in candidate\_values:

knots = calculate\_knots(X\_train, num\_knots)

X\_train\_spline = spline\_basis(X\_train, knots)

model = GroupLassoHuberRegressor()

model.fit(X\_train\_spline, y\_train)

1. 验证样本分割与调参

2.1 样本分割方法

为了确保模型的预测性能能够在样本外数据上得到有效验证，我们参考文献中的时间序列分割方法，将样本划分为三个不相交的时间段：训练集、验证集和测试集。这种划分方法不仅保持了数据的时间顺序，还模拟了模型在实际应用中的样本外测试场景。其中，**训练集**用于根据一组特定的调优参数值估计模型；**验证集**用于调优超参数，基于训练集估计的模型对验证集进行预测，并根据预测误差计算目标函数，迭代搜索最优超参数；**测试集**既不用于模型估计，也不用于超参数调优，是真正的样本外数据，用于评估模型的最终预测性能。

通过这种划分方法，我们能够有效避免模型在训练和调参过程中对测试数据的过拟合，从而更准确地评估模型的泛化能力。

2.2 时间序列滚动划分

为了充分利用时间序列数据，我们采用滚动窗口的方法对数据集进行划分。具体划分逻辑为：

1. 从数据集的起点开始，依次划分训练集、验证集和测试集；
2. 每次划分后，将窗口向前滚动roll\_freq个月，重复上述划分过程，直到测试集的结束日期超出数据范围。
3. 哈哈哈