技术报告

本项目在原框架的基础上，在minimatrix.py中实现了matrix类，并在main.py中进行了测试。下展开对minimatrix.py中函数的说明。

\_\_init\_\_()初始化函数接收三个参数：矩阵数据，维数，与起始值。若用户输入矩阵数据，则判断数据中各行列表长度是否相同，即能否生成矩阵，若不能则报错。若用户输入维数，则根据初始值生成该大小的矩阵。

shape()函数返回矩阵的维数。

reshape()函数接收一个元组作为新维数，返回重新排列后的矩阵。先判断新维数对应元素数量是否与原矩阵相符，若否则报错。将原矩阵数据按序存储为列表，再根据新维数进行重新排列生成新矩阵。

dot()函数接收另一个矩阵类对象，实现两对象的乘积，返回答案矩阵。先判断二者是否可乘，若否则报错。先创建一个答案矩阵大小的全零数据列表，随后按照矩阵乘法运算法则逐位运算。最后根据答案列表生成对象并输出。

T()函数不接受参数，实现矩阵的转置，返回答案矩阵。先建立目标大小的空矩阵，随后根据原矩阵数据填充信息。

sum()函数接收axis参数作为按列或按行或按元素相加的指针，返回求和结果。

copy()函数生成一个原矩阵的复制。重新按元素创造列表以防止嵌套列表id相同产生的问题。

Kronecker\_product()用于求Kronecker积。先创造空答案列表。使用嵌套循环遍历第一个矩阵的每个元素，并在内部循环中构建一个临时矩阵 num\_mat，该矩阵在对角线上具有第一个矩阵元素的值。将这个临时矩阵与第二个输入矩阵相乘，并将结果放入结果矩阵的相应位置。最后将答案输出。

\_\_getitem\_\_()依据行和列的输入数据类型分类，获取元素生成矩阵输出。

\_\_setitem\_\_()依据行和列的输入数据类型分类，更改元素。

\_\_pow\_\_()接受幂指数，先判断矩阵是否为方阵，随后不断做累乘得出目标矩阵。

\_\_add\_\_(),\_\_sub\_\_(),\_\_mul\_\_()先判断二矩阵是否大小相同，若否则报错。随后遍历所有矩阵，逐位进行操作。

\_\_len\_\_()返回总元素个数。

\_\_str\_\_()用于美化矩阵输出，使输出结果右对齐。对于每个元素，判断其是否为整数，如果是则将其转换为整数类型。根据元素类型和大小更新宽度值。如果存在大整数，将宽度设置为最大整数的宽度。如果最大整数宽度不超过5，且存在浮点数，则将宽度设置为7。否则，将宽度设置为最大整数宽度、7和当前元素宽度的最小值。

Gauss\_elimination()用于实现高斯消元法，返回高斯消元后的矩阵和交换行的次数。先创建变量seq用于追踪交换行的次数，同时复制输入矩阵以避免修改原始数据。通过循环迭代矩阵的每一列，对角线元素用作主元素进行消元操作。寻找当前主元素所在列的非零元素，如果主元素为零，则交换行以确保主元素非零。使用主元素对下方的元素进行消元，确保主元素下方的元素都为零。返回经过高斯消元后的矩阵和交换行的次数。

Det()用于求矩阵行列式。先判断是否为方阵，随后调用Gauss\_elimination()，对经过高斯消元法后的矩阵的主对角线上的元素做乘积，并根据交换行的次数判断是否取相反数。最后输出答案。

Inverse()通过初等行变换法求逆。先在原矩阵右侧拼接一个单位矩阵，并通过高斯消元法进行初等行变换直到左侧变为单位矩阵，右侧即为逆矩阵，最后将右半矩阵输出。

Rank()通过初等行变换法求秩。对原矩阵调用Gauss\_elimination()。非零行数即为秩，作为结果输出。

I(n)用于生成大小为 n x n 的单位矩阵。narray(dim, init\_value=1)用于生成指定维度的矩阵，所有元素初始化为给定的初始值。arange(start, end, step=1)用于生成一个行向量， start 是起始值，end 是终止值（不包含），step 是步长，默认为1。zeros(dim)用于生成一个零矩阵zeros\_like(matrix)用于生成一个与给定矩阵相同维度的零矩阵。ones(dim)用于生成一个元素全为1的矩阵。ones\_like(matrix)用于生成一个与给定矩阵相同维度的全1矩阵。nrandom(dim)用于生成一个指定维度的矩阵，元素为 [0, 1) 之间的随机数。nrandom\_like(matrix)用于生成一个与给定矩阵相同维度的随机数矩阵。

Concatenate()函数用于实现矩阵的拼接。函数首先验证输入参数的合法性，包括拼接对象的存在、类型为矩阵、拼接维度的正确性等。随后根据指定的维度，进行行或列的拼接操作。

Vectorize（）函数接受一个函数作为参数，返回一个新的函数，该新函数可以接受矩阵作为输入，并将函数应用于输入矩阵的每个元素或标量。如果输入 mat 是矩阵类型，则创建矩阵的副本ans\_mat，并遍历矩阵的每个元素，将 func 应用于每个元素。最后将处理后的矩阵输出。