实验六：深度学习与聚类算法

11. 在上一节课作业的基础上，请利用深度学习方法，对各学科做一个排名模型，能够较好的预测出排名位置，并且利用MSE，MAPE等指标来进行评价模型的优劣

在上一节课的作业中，我发现原数据的第一行和最后一行有冗余信息，可能会干扰表格的读取

第一行是表头，应该去掉：

	A	B	C	D	E	F	G	H	I	J	K	L
1	Indicators	Results List: Institutions		Filter Results By: ResearchFields		Filter Value(s): AGRICULTURAL SCIENCES		Show: Top				
2		Institution	Countries/	Web of Sc	Cites	Cites/Pap	Top Papers					
3	1	CHINESE /	CHINA MA	15661	332254	21.22	251					
4	2	CHINESE /	CHINA MA	12222	223855	18.32	198					
5	3	UNITED S	USA	12564	220644	17.56	105					
6	4	CHINA AC	CHINA MA	10052	207779	20.67	166					
7	5	INRAE	FRANCE	9314	187838	20.17	118					
8	6	JIANGNAI	CHINA MA	8027	181325	22.59	166					
9	7	NORTHW	CHINA MA	9024	180634	20.02	162					
10	8	WAGENIN	NETHERLA	6819	172734	25.33	152					
11	9	CONSEJO	SPAIN	7656	167593	21.89	94					
12	10	NANJING	CHINA MA	6480	137921	21.28	99					
13	11	UNIVERSI	USA	6175	129479	20.97	110					
14	12	UNIVERSI	CHINA MA	5950	127682	21.46	95					
15	13	ZHEJIANG	CHINA MA	5313	125715	23.66	148					
16	14	EGYPTIAN	EGYPT	7861	117405	14.94	103					
17	15	INDIAN C	INDIA	12753	111262	8.72	66					
18	16	SOUTH CI	CHINA MA	3524	108282	30.73	117					
19	17	CGIAR	FRANCE	5262	107193	20.37	69					
20	18	HUAZHOI	CHINA MA	4855	97482	20.08	68					
21	19	UNIVERSII	BRAZIL	6169	95242	15.44	35					
22	20	JIANGSU U	CHINA MA	4084	89040	21.8	176					
23	21	STATE UN	USA	5172	86527	16.73	62					
24	22	CENTRE N	FRANCE	4113	81348	19.78	60					
25	23	CORNELL	USA	3551	78744	22.18	59					

最后一行是 Copyright ?2025 Clarivate，也应该去掉：

	A	B	C	D	E	F	G	H	I	J	K	L
1359	1356	MAE FAH	THAILAND	236	3627	15.37	2					
1360	1358	NAMIK KE	TURKIYE	352	3625	10.3	1					
1361	1359	GANGNEU	SOUTH KO	268	3621	13.51	1					
1362	1360	TAMPERE	FINLAND	234	3619	15.47	0					
1363	1361	HOSPITAL	CANADA	202	3618	17.91	2					
1364	1362	BRAUNSC	GERMANY	188	3614	19.22	1					
1365	1362	ICAR - INI	INDIA	393	3614	9.2	6					
1366	1364	HUNGARI	HUNGARY	180	3605	20.03	2					
1367	1365	LIAONING	CHINA MA	333	3604	10.82	0					
1368	1366	DAIRYNZ	NEW ZEAI	239	3599	15.06	0					
1369	1367	SMITHSOI	USA	209	3598	17.22	0					
1370	1367	BAYER CR	GERMANY	250	3598	14.39	2					
1371	1369	GENERAL	USA	107	3595	33.6	4					
1372	1370	STEFAN C	ROMANIA	161	3591	22.3	5					
1373	1371	KAGOSHI	JAPAN	306	3588	11.73	1					
1374	1372	UNIVERSI	USA	127	3580	28.19	4					
1375	1373	HOPITAL I	FRANCE	123	3573	29.05	3					
1376	1373	HEINRICH	GERMANY	136	3573	26.27	4					
1377	1375	UNIVERSI	SOUTH AF	184	3572	19.41	6					
1378	1376	BANGLAD	BANGLAD	170	3567	20.98	2					
1379	1376	NEAR EAS	TURKIYE	204	3567	17.49	2					
1380	1378	GRIFFITH	AUSTRALI	205	3562	17.38	2					
1381	1378	UMM AL-	SAUDI AR	308	3562	11.56	3					
1382	1380	SAINT LO	USA	108	3561	32.97	4					
1383	1381	VIT VELLC	INDIA	144	3558	24.71	2					
1384	Copyright ?2025 Clarivate											
1385												

在 Powershell 执行下面命令：

```
$src="D:\ESI\download"; $dst="D:\ESI\clean"; New-Item -ItemType Directory -Force
-Path $dst | Out-Null
Get-ChildItem $src -Filter *.csv | ForEach-Object {
  $lines = Get-Content $_.FullNameGet-ChildItem $src -Filter *.csv
  $lines[1..($lines.Count-2)] | Set-Content -Encoding UTF8 (Join-Path $dst
$_.Name)
}
```

解释一下：

1. src 是原始文件路径，dst 是目标文件路径
2. New-Item 用于创建一个新目录，如果 D:\ESI\clean 不存在，会自动创建
3. Get-ChildItem $src -Filter *.csv 会列出 src 中所有扩展名为 .csv 的文件
4. ForEach-Object 遍历每个文件
5. $lines = Get-Content $_.FullName 会把整个文件的内容读成一个字符串数组，每一行是一个元素
6. $lines[1..($lines.Count-2)] 是数组切片语法，表示从第二行到倒数第二行，删除文件的第一行和最后一行
7. Set-Content 把上一步得到的“去掉第一行的内容”写入到新的文件中

可以看到，第一行已经成功去掉了：

	A	B	C	D	E	F	G	H	I	J	K	L
1		Institution	Countries/	Web of Sc	Cites	Cites/Pape	Top Papers					
2	1	CHINESE /	CHINA M/	15661	332254	21.22	251					
3	2	CHINESE /	CHINA M/	12222	223855	18.32	198					
4	3	UNITED S	USA	12564	220644	17.56	105					
5	4	CHINA AC	CHINA M/	10052	207779	20.67	166					
6	5	INRAE	FRANCE	9314	187838	20.17	118					
7	6	JIANGNAI	CHINA M/	8027	181325	22.59	166					
8	7	NORTHW	CHINA M/	9024	180634	20.02	162					
9	8	WAGENIN	NETHERLA	6819	172734	25.33	152					
10	9	CONSEJO	SPAIN	7656	167593	21.89	94					
11	10	NANJING	CHINA M/	6480	137921	21.28	99					
12	11	UNIVERSI	USA	6175	129479	20.97	110					
13	12	UNIVERSI	CHINA M/	5950	127682	21.46	95					
14	13	ZHEJIANG	CHINA M/	5313	125715	23.66	148					
15	14	EGYPTIAN	EGYPT	7861	117405	14.94	103					
16	15	INDIAN C	INDIA	12753	111262	8.72	66					
17	16	SOUTH CI	CHINA M/	3524	108282	30.73	117					
18	17	CGIAR	FRANCE	5262	107193	20.37	69					
19	18	HUAZHOI	CHINA M/	4855	97482	20.08	68					
20	19	UNIVERSI	BRAZIL	6169	95242	15.44	35					
21	20	JIANGSU U	CHINA M/	4084	89040	21.8	176					
22	21	STATE UN	USA	5172	86527	16.73	62					
23	22	CENTRE N	FRANCE	4113	81348	19.78	60					
24	23	CORNELL	USA	3551	78744	22.18	59					
25	24	UNIVERSI	USA	3908	78346	20.05	64					

最后一行也成功去掉了：

	A	B	C	D	E	F	G	H	I	J	K	L
1366	1365	LIAONING	CHINA MA	333	3604	10.82	0					
1367	1366	DAIRYNZ	NEW ZEAI	239	3599	15.06	0					
1368	1367	SMITHSOI	USA	209	3598	17.22	0					
1369	1367	BAYER CR	GERMANY	250	3598	14.39	2					
1370	1369	GENERAL	USA	107	3595	33.6	4					
1371	1370	STEFAN C	ROMANIA	161	3591	22.3	5					
1372	1371	KAGOSHI	JAPAN	306	3588	11.73	1					
1373	1372	UNIVERSI	USA	127	3580	28.19	4					
1374	1373	HOPITAL I	FRANCE	123	3573	29.05	3					
1375	1373	HEINRICH	GERMANY	136	3573	26.27	4					
1376	1375	UNIVERSI	SOUTH AF	184	3572	19.41	6					
1377	1376	BANGLAD	BANGLAD	170	3567	20.98	2					
1378	1376	NEAR EAS	TURKIYE	204	3567	17.49	2					
1379	1378	GRIFFITH	AUSTRALI	205	3562	17.38	2					
1380	1378	UMM AL-	SAUDI AR	308	3562	11.56	3					
1381	1380	SAINT LO	USA	108	3561	32.97	4					
1382	1381	VIT VELLC	INDIA	144	3558	24.71	2					
1383												
1384												
1385												
1386												
1387												
1388												
1389												
1390												
1391												

我整理了一下邱学长提到的深度学习步骤：

> 定义数据集
> 定义模型
> 定义训练过程
> 训练多少轮次
> 参数冻结
> 在测试集上跑一遍
> 深层次MMP，每层加一些归一化、正则化、非线性组合

首先，题目要求我们对各学科做一个排名模型，预测排名位置

电脑通过学习一些国家的（Web of Science Documents，Cites，Cites/Paper，Cites/Paper，排名），

再给其他国家的（Web of Science Documents，Cites，Cites/Paper，Cites/Paper），可以推测出排名

我们来明确一下输入和输出：

输入：多个 CSV，每个 CSV 代表一个学科，第一列就是排名，这是通过人工观察发现的

输出：对于每个学科，在**测试值**上跑预测模型，得出的**真实值**对比**预测值**的表，以及各项指标的总表

我们要明确一点，每个学科的排名预测模型都应该是**独立**的

因此，要循环遍历每个csv文件，**写一个循环**

在循环中，依次遍历存放多个 CSV 文件路径的列表，并同时获取每个文件的编号（从 1 开始）和路径

```
for i, csv_path in enumerate(csvs, 1):
```

然后，取出当前表格最左侧的第一列列名，并打印当前处理进度，包括文件编号、总文件数、学科文件名以及该排名列的列名，用来提示程序正在处理哪个学科的排名数据

```
label_col = df.columns[0]
print(f"\n=== [{i}/{len(csvs)}] 学科: {csv_path.name} | 排名列: {label_col} ===")
```

前面提到过了，我们通过一个学校的：

X =（Web of Science Documents，Cites，Cites/Paper，Cites/Paper）

来预测它的排名，设为 Y

将表格中最左侧的排名列转成数值型，然后把剩下的所有列作为特征矩阵 X，并通过 to_numeric_df() 函数将这些特征列全部转换为数值类型，以便后续模型训练或聚类分析使用

```
y_raw = pd.to_numeric(df[label_col], errors="coerce").astype(float)
X = df.drop(columns=[label_col])
X = to_numeric_df(X)
```

不要忘了过滤掉在排名列中存在缺失值的行，只保留排名有效的样本，使得 X 和 Y 严格对应且都是完整的数值型数据，从而确保后续训练或聚类过程不会受到缺失数据干扰

```
mask = y_raw.notna()
X = X.loc[mask].copy()
y_raw = y_raw.loc[mask].astype(float)
```

然后，划分数据集

课堂上，老师讲解了测试集、验证集、测试集

邱学长也提到了合适的比例：60%，20%，20%

我们还要计算当前样本总数 N，并生成一个从 0 到 N-1 的索引数组 idx_all，然后调用 split_60_20_20_idx() 函数，基于给定随机种子 SEED 进行数据集的划分

```
N = len(row_ids)
idx_all = np.arange(N)
idx_tr, idx_va, idx_te = split_60_20_20_idx(idx_all, seed=SEED)
```

然后，我们要 scale 一下特征，因为特征的大小范围很不统一

创建一个标准化器 StandardScaler()，以训练集数据为基准进行特征标准化处理——即让每个特征的均值为 0、标准差为 1

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_np[idx_tr])
X_val   = scaler.transform(X_np[idx_va])
X_test  = scaler.transform(X_np[idx_te])
```

排名也要 scale 一下

经过下面的处理后，模型预测输出也是 [0,1] 之间的小数，更易收敛、数值稳定，并可在评估阶段反归一化回真实排名

```
y_min = float(np.min(y_np))
y_max_v = float(np.max(y_np))
y_max = y_max_v if y_max_v != y_min else (y_min + 1.0)
def norm(y): return (y - y_min) / (y_max - y_min)

y_train = norm(y_np[idx_tr])
y_val   = norm(y_np[idx_va])
y_test  = norm(y_np[idx_te])
```

然后，调用自定义函数 train_one() 来训练一个深度学习模型，并返回在测试集上的预测结果

```
_, predict_rank = train_one(
    X_train, y_train, X_val, y_val, X_test, y_test,
    y_min=y_min, y_max=y_max, device=device
)
```

先插叙一下，深度学习模型的**内部**是怎么工作的?

首先，动态选择合适的归一化方式，构建深度学习模型，并定义一个稳健的损失函数

```
norm_kind = "batch" if len(X_train) >= 2 else "layer"
model = DeepRankMLP(X_train.shape[1], norm_kind=norm_kind).to(device)
huber = nn.SmoothL1Loss(beta=1.0)  # robust to outliers
```

训练分为两个阶段，full training 和 freeze and fine-tune

1. full training

先用 AdamW 优化器，并配上 StepLR 学习率调度器（每 40 个 epoch 将学习率乘以 0.5），随后把训练集与验证集的 NumPy 特征/标签 转成 PyTorch 张量 并封装为 TensorDataset，为后续用 DataLoader 批量迭代训练与评估做准备

```
opt = torch.optim.AdamW(model.parameters(), lr=LR1, weight_decay=WEIGHT_DECAY)
sched = torch.optim.lr_scheduler.StepLR(opt, step_size=40, gamma=0.5) tr_ds =
TensorDataset(torch.from_numpy(X_train).float(),
torch.from_numpy(y_train_norm).float()) va_ds =
TensorDataset(torch.from_numpy(X_val).float(),
torch.from_numpy(y_val_norm).float())
```

构建 PyTorch 的 DataLoader，也就是让模型能够按批次取出训练数据和验证数据的关键部分

```
bs_eff = min(BATCH_SIZE, len(tr_ds))
drop_last = (len(tr_ds) >= 2)
tr_dl = DataLoader(tr_ds, batch_size=bs_eff, shuffle=True, drop_last=drop_last)
va_dl = DataLoader(va_ds, batch_size=min(BATCH_SIZE, len(va_ds)), shuffle=False)
```

把模型输出的归一化预测结果（0~1之间）反变换回原始排名范围

因为我们前面把排名 scale 到了 0~1之间，可是题目想要的肯定是预测原始排名

```
def denorm(y_hat_norm: np.ndarray) -> np.ndarray:
    return y_min + np.clip(y_hat_norm, 0.0, 1.0) * (y_max - y_min)
```

下面实现一个完整的训练轮次循环

每轮训练都会遍历整个训练集，将模型设为训练模式，再逐批取出数据送入网络进行前向传播，计算预测值与真实值的 Huber 损失

接着清空梯度、反向传播、裁剪梯度防止爆炸，最后用 AdamW 优化器更新参数

每个 epoch 结束后，学习率调度器按计划降低学习率，以帮助模型在后期更平稳地收敛

```
nonlocal best_state, best_val_mse_rank
for ep in range(1, epochs + 1):
    if len(tr_dl) == 0:
        break
    model.train()
    for xb, yb in tr_dl:
        xb, yb = xb.to(device), yb.to(device)
        pred = model(xb)
        loss = huber(pred, yb)
        opt.zero_grad()
        loss.backward()
        nn.utils.clip_grad_norm_(model.parameters(), 5.0)
        opt.step()
    sched.step()
```

先切换到 eval 模式并关闭梯度计算，然后用验证集逐批预测，得到所有预测值和真实值

接着将它们反归一化为原始排名，用 MSE 计算误差

若当前误差小于历史最优值，就更新 best_val_mse_rank 并保存当前模型参数

```
model.eval()
with torch.no_grad():
    yp, yt = [], []
    for xb, yb in va_dl:
        xb = xb.to(device)
        yp.append(model(xb).cpu().numpy())
        yt.append(yb.cpu().numpy())
    if len(yp) == 0:
        continue
    y_pred_norm = np.concatenate(yp).reshape(-1)
    y_true_norm = np.concatenate(yt).reshape(-1)
    y_pred_rank = denorm(y_pred_norm)
    y_true_rank = denorm(y_true_norm)
    v_mse_rank = mse_np(y_true_rank, y_pred_rank)
    if v_mse_rank < best_val_mse_rank:
        best_val_mse_rank = v_mse_rank
        best_state = {k: v.clone() for k, v in model.state_dict().items()}
```

2. freeze and fine-tune

接下来是第二部分，冻结参数训练

首先调用冻结模型前半部分的参数，只让后半部分参与训练

接着，用 AdamW 优化器重新初始化，只更新未冻结部分的参数，并设置一个更小的学习率 LR2 与相同的 weight_decay 来避免过拟合

然后建立新的 StepLR 调度器，每 20 轮将学习率减半

最后调用 run_epochs(EPOCHS_2) 进行多轮训练

```
freeze_first_half(model, True)
opt = torch.optim.AdamW(filter(lambda p: p.requires_grad, model.parameters()),
lr=LR2, weight_decay=WEIGHT_DECAY)
sched = torch.optim.lr_scheduler.StepLR(opt, step_size=20, gamma=0.5)
run_epochs(EPOCHS_2)
```

最后是预测排名阶段

首先，将模型切换到评估模式，关闭 dropout 和 batch norm 的随机行为

关闭梯度计算，避免浪费显存和计算资源

把输入的 NumPy 数组 X_np 转换为 PyTorch 张量并移动到设备上

调用模型获得预测的归一化输出 yhat_norm，再转回 NumPy 格式

通过 denorm() 函数反归一化，将结果映射回真实的排名区间

```
def predict_rank(X_np: np.ndarray) -> np.ndarray:
    model.eval()
    with torch.no_grad():
        X_t = torch.from_numpy(X_np).float().to(device)
        yhat_norm = model(X_t).cpu().numpy().reshape(-1)
    return denorm(yhat_norm)
```

最后，利用训练好的模型在测试集上进行预测，并计算预测结果与真实排名之间的多种评估指标，用于衡量模型性能

```
y_pred_te = predict_rank(X_test)
y_true_te = y_np[idx_te]

metrics = compute_metrics(y_true_te, y_pred_te)
```

关于输出部分，我们前面提到过了：

对于每个学科，在**测试值**上跑预测模型，得出的**真实值**对比**预测值**的表，以及各项指标的总表

首先是各项指标的输出，需要的数据都在metrics字典中

依次输出即可

```
print("[测试] " + "  ".join([
    f"MSE={metrics['MSE']:.4f}",
    f"RMSE={metrics['RMSE']:.2f}",
    f"MAE={metrics['MAE']:.2f}",
    f"MAPE={metrics['MAPE']:.2f}%",
    f"MedAE={metrics['MedianAE']:.2f}",
    f"R2={metrics['R2']:.3f}",
    f"nRMSE={metrics['nRMSE']:.3f}",
    f"Spearman={metrics['Spearman']:.3f}",
    f"Kendall={metrics['Kendall']:.3f}",
]))
```

然后是对比表

把模型在测试集上某所大学的真实排名和预测排名保存为一个独立的结果表

```
pred_df = pd.DataFrame({
    "row_id": row_ids[idx_te],
    "true_rank": y_true_te,
    "pred_rank": y_pred_te,
})
```

生成每个学科预测结果的 CSV 文件名并保存预测结果表格

```
stem = csv_path.stem
safe = "".join([ch if ch.isalnum() or ch in ("-","_") else "_" for ch in stem])
pred_path = (Path(__file__).resolve().parent / f"predictions_{safe}.csv")
pred_df.to_csv(pred_path, index=False)
print(f"已保存测试对比表: {pred_path.resolve()}")
```

但是我想把所有学科的各项指标汇总到一张表中，以便**纵向对比**

把每个学科对应的模型评估指标保存到一个总表列表中，以便最后统一生成总表

```
row = {"subject_csv": csv_path.name}
row.update(metrics)
rows.append(row)
```

最后，在所有学科模型的预测模型评测结果汇总后，生成一个综合评价表 deep_learning.csv，并按 MSE 升序排序显示和保存

```
if rows:
    out = pd.DataFrame(rows).sort_values("MSE")
    out_path = (Path(__file__).resolve().parent / "deep_learning.csv")
    out.to_csv(out_path, index=False)
    print("\n=== 总结（按 MSE 升序） ===")
    print(out.to_string(index=False))
    print(f"\n结果已保存到: {out_path.resolve()}")
```

以上的完整代码，请查看 deep_learning.py

各学科排名模型在**测试集**上的预测值与真实值对比表格，都在附件中：

名称	修改日期	类型	大小
predictions_CHEMISTRY.csv	30/10/2025 10:20	Microsoft Excel ...	10 KB
predictions_CLINICAL_MEDICINE.csv	30/10/2025 10:21	Microsoft Excel ...	30 KB
predictions_COMPUTER_SCIENCE.csv	30/10/2025 10:21	Microsoft Excel ...	4 KB
predictions_ECONOMICS__BUSINES...	30/10/2025 10:22	Microsoft Excel ...	3 KB
predictions_ENGINEERING.csv	30/10/2025 10:22	Microsoft Excel ...	12 KB
predictions_ENVIRONMENT_ECOLOG...	30/10/2025 10:22	Microsoft Excel ...	9 KB
predictions_GEOSCIENCES.csv	30/10/2025 10:22	Microsoft Excel ...	5 KB
predictions_IMMUNOLOGY.csv	30/10/2025 10:22	Microsoft Excel ...	5 KB
predictions_MATERIALS_SCIENCE.csv	30/10/2025 10:22	Microsoft Excel ...	7 KB
predictions_MATHEMATICS.csv	30/10/2025 10:22	Microsoft Excel ...	2 KB
predictions_MICROBIOLOGY.csv	30/10/2025 10:23	Microsoft Excel ...	4 KB
predictions_MOLECULAR_BIOLOGY__...	30/10/2025 10:23	Microsoft Excel ...	5 KB
predictions_MULTIDISCIPLINARY.csv	30/10/2025 10:23	Microsoft Excel 逗号分隔值文件	3
predictions_NEUROSCIENCE__BEHA...	30/10/2025 10:23	Microsoft Excel ...	6 KB
predictions_PHARMACOLOGY__TO...	30/10/2025 10:23	Microsoft Excel ...	6 KB
predictions_PHYSICS.csv	30/10/2025 10:23	Microsoft Excel ...	4 KB
predictions_PLANT__ANIMAL_SCIE...	30/10/2025 10:23	Microsoft Excel ...	9 KB
predictions_PSYCHIATRY_PSYCHOLO...	30/10/2025 10:23	Microsoft Excel ...	5 KB
predictions_SOCIAL_SCIENCES__GEN...	30/10/2025 10:24	Microsoft Excel ...	11 KB
predictions_SPACE_SCIENCE.csv	30/10/2025 10:24	Microsoft Excel ...	1 KB
8_build.csv	30/10/2025 10:24	Microsoft Excel ...	5 KB

我这里随机选一个学科展示一下：

	A	B	C	D	E	F	G	H	I	J
1	row_id	true_rank	pred_rank							
2	581	582	580.8528							
3	175	176	210.1509							
4	548	549	538.756							
5	654	655	642.4933							
6	1926	1927	1934.248							
7	1244	1245	1256.446							
8	290	291	295.6851							
9	221	222	268.3793							
10	1461	1461	1466.63							
11	433	434	452.868							
12	121	122	190.9796							
13	891	892	879.5628							
14	286	287	314.0567							
15	867	868	851.5161							
16	347	348	367.6648							
17	1962	1963	1974.138							
18	2731	2731	2611.857							
19	787	788	669.9355							
20	1798	1799	1808.691							
21	188	189	216.5818							
22	1446	1447	1464.342							
23	1910	1911	1919.203							
24	303	304	291.6348							
25	1621	1622	1635.82							
26	2164	2165	2184.776							
27	1933	1934	1946.557							
28	409	410	437.9659							
29	1716	1717	1724.965							

predictions_ENGINEERING

可以看到，预测值与真实值还是比较接近的，说明模型预测效果好

预测的排名值是一个小数，而不是整数，虽然排名只能是整数

如果需要整数，把预测的排名值四舍五入即可，或者向偶数取整

不过我觉得小数也是有意义的，因为它本身只是一个估计

比如一个大学预测排名为 2.5，那可以认为它的排名大概是第二、第三

各学科排名模型在**测试集**上的**各项指标**如下：

```
=== 总结（按 MSE 升序） ===
                      subject_csv         MSE      RMSE       MAE      MAPE  MedianAE       R2    nRMSE  Spearman  Kendall
                MICROBIOLOGY.csv  252.269248 15.882986 10.887834  9.391180  8.071838 0.995208 0.019954  0.999330 0.984664
              SPACE SCIENCE.csv  341.910472 18.490821 12.745177 17.596886  9.387699 0.916745 0.081818  0.964611 0.867021
               GEOSCIENCES.csv  393.698578 19.841839 14.370080  7.106743 11.384811 0.996415 0.017076  0.999041 0.980594
                IMMUNOLOGY.csv  508.733315 22.555117 14.377933  9.384470  9.383499 0.995366 0.019278  0.998747 0.982618
MOLECULAR BIOLOGY & GENETICS.csv  617.998418 24.859574 16.262368  9.345132  9.079315 0.994337 0.021449  0.999509 0.988023
       AGRICULTURAL SCIENCES.csv  630.600022 25.111751 15.397359  5.749851  8.184601 0.995703 0.018237  0.999727 0.988554
     NEUROSCIENCE & BEHAVIOR.csv  805.005098 28.372612 23.433166  8.089908 23.185997 0.994044 0.022114  0.999609 0.989159
          MATERIALS SCIENCE.csv  812.457454 28.503639 19.236143  9.710216 11.946030 0.996106 0.018295  0.999076 0.979174
                PHYSICS.csv  907.911353 30.131567 21.446334  9.820175 15.024963 0.989004 0.030528  0.997779 0.972158
   PHARMACOLOGY & TOXICOLOGY.csv  958.344377 30.957138 22.375154  6.753941 17.531281 0.993706 0.022449  0.999778 0.991520
     PSYCHIATRY PSYCHOLOGY.csv 1191.735976 34.521529 17.262220  7.539682 10.504194 0.988529 0.030362  0.995185 0.968633
        ENVIRONMENT ECOLOGY.csv 1425.751402 37.759123 28.113645  7.195055 24.663239 0.995981 0.018482  0.999325 0.987613
          COMPUTER SCIENCE.csv 1430.191239 37.817869 20.458793 11.278869 12.577209 0.977369 0.044180  0.992381 0.978220
             ENGINEERING.csv 1504.758002 38.791210 24.294541  9.406422 12.861816 0.997717 0.013969  0.999850 0.993642
          MULTIDISCIPLINARY.csv 1763.283374 41.991468 23.607978 56.734116  8.507751 0.446612 0.203842  0.711346 0.610994
                MATHEMATICS.csv 2138.738504 46.246497 27.812698 73.346139 15.793304 0.840698 0.118581  0.928669 0.830385
    BIOLOGY & BIOCHEMISTRY.csv 2251.553616 47.450539 34.840483 13.596073 23.239258 0.989958 0.029200  0.999768 0.992530
        CLINICAL MEDICINE.csv 2467.755633 49.676510 40.941861  3.610824 36.800293 0.999367 0.007373  0.999911 0.994969
      ECONOMICS & BUSINESS.csv 2705.859523 52.017877 24.491297 30.210407 12.215485 0.904894 0.095974  0.953637 0.892438
               CHEMISTRY.csv 3585.305891 59.877424 44.487254 16.364376 32.726349 0.990791 0.028098  0.999752 0.994537
    PLANT & ANIMAL SCIENCE.csv 3697.386316 60.806137 48.659946 21.578761 38.319824 0.988808 0.031311  0.999848 0.992538
  SOCIAL SCIENCES, GENERAL.csv 4389.281090 66.251650 49.070618 12.624892 41.315399 0.991133 0.027674  0.998471 0.989669
```

其中的 nRMSE 是 RMSE 按数值范围归一化之后的，很有参考价值：

> nRMSE < 0.10：很好
>
> 0.10–0.20：可用

而各学科排名模型在**测试集**上的 nRMSE 都小于 0.10，说明模型预测效果**很好**

各项指标的总表也在附件中，叫 deep_learning.csv

12. 对ESI的数据进行聚类，发现与华师大类似的学校有哪些，并分析下原因

数据是每个学科一个表格，适合观察某个学科哪些高校厉害

但是如果想将全球高校聚类，应该观察**同一个高校的不同学科**，数据的组织形式是不适合这种观察的

我的思路是把所有表格合并成一个二维表格，每一行表示一个高校，每一列表示一个学科

某行某列的元素代表**某个高校在某个学科的排名**

手动构建这个二维表格的工作量是巨大的，我们肯定要写一个python程序

下面来构建二维表格，重复一下我的思路：

> 把所有表格合并成一个二维表格，每一行表示一个高校，每一列表示一个学科
>
> 某行某列的元素代表**某个高校在某个学科的排名**

我们读取clean目录下的所有CSV文件，把二维表格输出到一个CSV文件中：

```
INPUT_DIR = (Path(__file__).resolve().parent / "clean")
OUTPUT_CSV = (Path(__file__).resolve().parent / "build.csv")
```

先写一个函数，列出所有CSV文件：

```
def list_subject_files(root: Path) -> list[Path]:
    files = sorted(root.glob("*.csv"))
    if not files:
        raise FileNotFoundError(f"No CSV files found in {root}")
    return files
```

对于每个学科表，我们只关心大学名称、学科名称、学科排名这三个信息

学科排名：rank 取文件的第一列（清洗后的名次列）

大学名称：university 取 'Institutions'

学科名称：subject 用文件名（去掉扩展名）

对于每个学科表，处理后返回一个 pd.DataFrame，包含三列：['university', 'subject', 'rank']

读入 CSV，找到表示名次的第一列，指定表示学校名称的 Institutions 列：

```
df = pd.read_csv(csv_path, dtype=str)
rank_col = df.columns[0]
inst_col = "Institutions"

out = pd.DataFrame({
    "university": df[inst_col].astype(str).str.strip(),
    "rank": pd.to_numeric(df[rank_col], errors="coerce"),
}).dropna(subset=["university", "rank"])

out["rank"] = out["rank"].astype(int)
out["subject"] = csv_path.stem.strip()
return out[["university", "subject", "rank"]]
```

每个学科都会返回一个 pd.DataFrame，将所有 pd.DataFrame 合并：

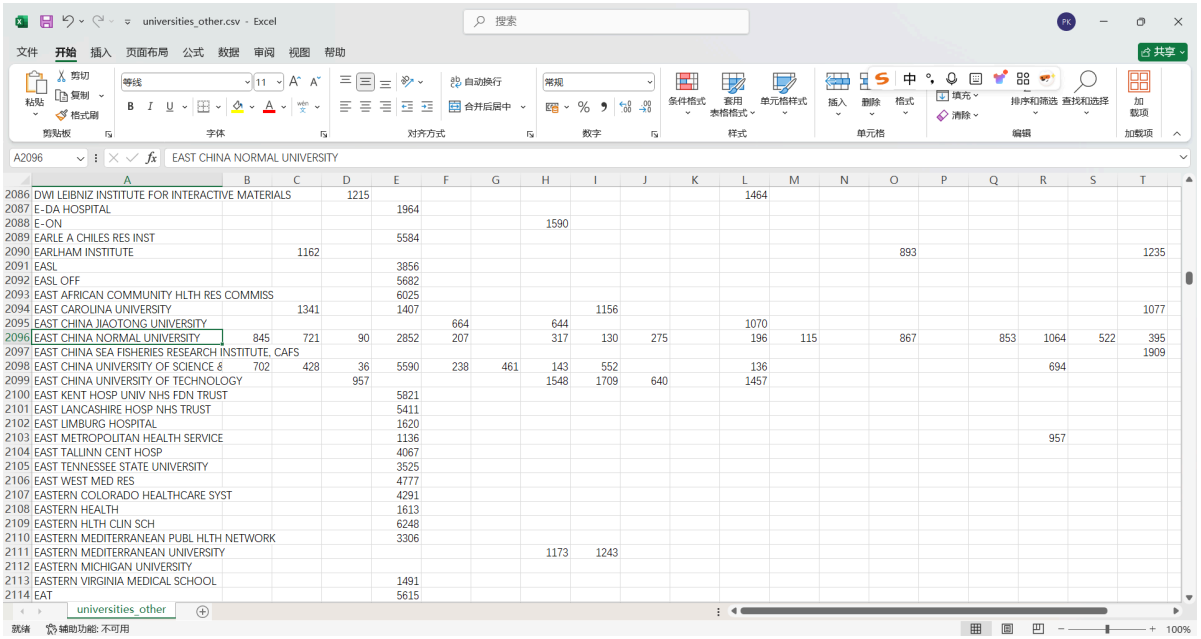
```
frames = [read_subject(p) for p in list_subject_files(input_dir)]
long_df = pd.concat(frames, ignore_index=True)
```

最后，按行、列排序：

```
return pivot.sort_index().sort_index(axis=1)
```

以上预处理的完整代码请老师阅读 build.py

运行后，得到二维表格，取名叫 build.csv，同样在附件中：



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
2086	DWI LEIBNIZ INSTITUTE FOR INTERACTIVE MATERIALS			1215								1464								
2087	E-DA HOSPITAL				1964															
2088	E-ON							1590												
2089	EARLE A CHILES RES INST				5584															
2090	EARLHAM INSTITUTE		1162												893					1235
2091	EASL				3856															
2092	EASL OFF				5682															
2093	EAST AFRICAN COMMUNITY HLTH RES COMMISS				6025															
2094	EAST CAROLINA UNIVERSITY		1341		1407				1156											1077
2095	EAST CHINA JIAOTONG UNIVERSITY					664		644				1070								
2096	EAST CHINA NORMAL UNIVERSITY	845	721	90	2852	207		317	130	275		196	115		867		853	1064	522	395
2097	EAST CHINA SEA FISHERIES RESEARCH INSTITUTE, CAFS																			1909
2098	EAST CHINA UNIVERSITY OF SCIENCE &	702	428	36	5590	238	461	143	552			136						694		
2099	EAST CHINA UNIVERSITY OF TECHNOLOGY			957				1548	1709	640		1457								
2100	EAST KENT HOSP UNIV NHS FDN TRUST				5821															
2101	EAST LANCASHIRE HOSP NHS TRUST				5411															
2102	EAST LIMBURG HOSPITAL				1620															
2103	EAST METROPOLITAN HEALTH SERVICE				1136													957		
2104	EAST TALLINN CENT HOSP				4067															
2105	EAST TENNESSEE STATE UNIVERSITY				3525															
2106	EAST WEST MED RES				4777															
2107	EASTERN COLORADO HEALTHCARE SYST				4291															
2108	EASTERN HEALTH				1613															
2109	EASTERN HLTH CLIN SCH				6248															
2110	EASTERN MEDITERRANEAN PUBL HLTH NETWORK				3306															
2111	EASTERN MEDITERRANEAN UNIVERSITY							1173	1243											
2112	EASTERN MICHIGAN UNIVERSITY																			
2113	EASTERN VIRGINIA MEDICAL SCHOOL				1491															
2114	EAT				5615															

空白元素是正常的，代表这个高校在这个学科排名中**没有上榜**

有了这个表格，我们就可以直观地将全球高校进行分类

python 库中，sklearn 是非常好用的

它里面有一个 KMeans 模组，可以很方便地实现 KMeans 算法

首先，读取二维表格

第一列是学校名，单独存到 universities 中

其余列为各学科排名，存到一个矩阵中

某些学校在某些学科没有上榜，那么这一项就是空白的，怎么办?

我的创意是，填充这个学科最大排名的 2 倍，这样能体现该校这个学科的实力比较弱

```
path = "build.csv"
df = pd.read_csv(path)
universities = df.iloc[:, 0]
X = df.iloc[:, 1:].fillna(df.max() * 2)
```

然后对各个学科排名数据进行标准化，都转化为 均值为 0、标准差为 1 的分布

```
X_scaled = StandardScaler().fit_transform(X)
```

下面是整个聚类分析的核心步骤

先创建一个 KMeans 聚类器，n_clusters=8 表示希望将所有高校分成 8个聚类

random_state=42，固定随机数种子，保证每次运行结果一致

n_init=10 表示算法会尝试 10 次不同的随机初始中心，选择其中效果最好的一次作为最终模型，从而提高聚类稳定性

再将聚类标签作为新的一列 "Cluster" 添加到原始 DataFrame

```
kmeans = KMeans(n_clusters=8, random_state=42, n_init=10)
labels = kmeans.fit_predict(X_scaled)
df["Cluster"] = labels
```

然后，我们来发现与华师大归到了同一个类别的学校有哪些

首先，找到 EAST CHINA NORMAL UNIVERSITY 所在的那一行

提取那一行的 Cluster 值，也就是华师大被归到的类别

然后，获取所有学校的名字构成的列表，也就是总表的第一列

在表格中查找 Cluster 值等于华师大的 Cluster 值，有哪些行

把这些行中的学校名称记录到一个列表中

最后输出出来

以上的完整代码，请查看 similar.py

```
target = "EAST CHINA NORMAL UNIVERSITY"
if target in universities.values:
    target_cluster = df.loc[universities == target, "Cluster"].values[0]
    name_col = df.columns[0]
    similar_schools = df[df["Cluster"] == target][name_col].values
    print(f"与 {target} 聚类相似的高校：")
    print(similar_schools)
```

```
PS D:\Program Files\Microsoft VS Code> & D:\python\python.exe d:/ECNU/数据科学导论/第六次作业/similar.py
1
与 EAST CHINA NORMAL UNIVERSITY 聚类相似的高校：
['AALTO UNIVERSITY' 'ACADEMIA SINICA - TAIWAN'
 'AGENCY FOR SCIENCE TECHNOLOGY & RESEARCH (A*STAR)'
 'ARISTOTLE UNIVERSITY OF THESSALONIKI' 'ARIZONA STATE UNIVERSITY-TEMPE'
 'AUTONOMOUS UNIVERSITY OF MADRID' 'BEIJING NORMAL UNIVERSITY'
 'BEN-GURION UNIVERSITY OF THE NEGEV' 'BROWN UNIVERSITY'
 'CAIRO UNIVERSITY' 'CALIFORNIA STATE UNIVERSITY SYSTEM'
 'CARDIFF UNIVERSITY' 'CARNEGIE MELLON UNIVERSITY'
 'CASE WESTERN RESERVE UNIVERSITY'
 'CATHOLIC UNIVERSITY OF THE SACRED HEART' 'CENT VAL LOIRE COMUE'
 'CENTRAL SOUTH UNIVERSITY' 'CHARLES UNIVERSITY PRAGUE'
 'CHINA AGRICULTURAL UNIVERSITY' 'CHINA MEDICAL UNIVERSITY TAIWAN'
 'CHINESE ACADEMY OF AGRICULTURAL SCIENCES'
 'CHINESE ACADEMY OF MEDICAL SCIENCES - PEKING UNION MEDICAL COLLEGE'
 'CHINESE UNIVERSITY OF HONG KONG' 'CHULALONGKORN UNIVERSITY'
 'CIBER - CENTRO DE INVESTIGACION BIOMEDICA EN RED'
 'CITY UNIVERSITY OF HONG KONG'
 'CITY UNIVERSITY OF NEW YORK (CUNY) SYSTEM'
 'COLORADO STATE UNIVERSITY FORT COLLINS'
 'COLORADO STATE UNIVERSITY SYSTEM'
 'COMMONWEALTH SCIENTIFIC & INDUSTRIAL RESEARCH ORGANISATION (CSIRO)'
 'COMMUNAUTE UNIVERSITE GRENOBLE ALPES' 'COMPLUTENSE UNIVERSITY OF MADRID'
 'CONSEJO NACIONAL DE INVESTIGACIONES CIENTIFICAS Y TECNICAS (CONICET)'
 'COUNCIL OF SCIENTIFIC & INDUSTRIAL RESEARCH (CSIR) - INDIA'
 'CURTIN UNIVERSITY' 'CZECH ACADEMY OF SCIENCES' 'DALHOUSIE UNIVERSITY'
 'DARTMOUTH COLLEGE' 'DEAKIN UNIVERSITY' 'DELFT UNIVERSITY OF TECHNOLOGY'
 'DREXEL UNIVERSITY' 'EAST CHINA NORMAL UNIVERSITY'
 'EBERHARD KARLS UNIVERSITY OF TUBINGEN'
 'ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE' 'EMORY UNIVERSITY'
 'ERASMUS UNIVERSITY ROTTERDAM' 'FLINDERS UNIVERSITY SOUTH AUSTRALIA'
 'FLORIDA STATE UNIVERSITY' 'FRIEDRICH SCHILLER UNIVERSITY OF JENA'
 'GEORGE WASHINGTON UNIVERSITY' 'GEORGIA INSTITUTE OF TECHNOLOGY'
 'GEORGIA STATE UNIVERSITY' 'GOETHE UNIVERSITY FRANKFURT'
 'GRIFFITH UNIVERSITY' 'HANYANG UNIVERSITY'
```

不过归到华师大这一类的学校有很多，不知道哪几个最相似

我有两个思路：

1. 计算相似度，按照相似度从大到小排序，输出出来

```
target_vec = X_scaled[universities.str.upper() == target.upper()][0].reshape(1,
-1)
similarity = cosine_similarity(target_vec, X_scaled)[0]
df["Similarity"] = similarity

print("\n最相似的高校（按相似度降序）：")
print(df.sort_values("Similarity", ascending=False).head(50)[[name_col,
"Similarity"]])
```

运行结果如下，输出了与华师大相似度前30名的高校，以及相似度

```
最相似的高校（按相似度降序）：
                                      university  Similarity
2174                     EAST CHINA NORMAL UNIVERSITY    1.000000
8863                     UNIVERSITY OF BASQUE COUNTRY    0.975276
8966  UNIVERSITY OF ELECTRONIC SCIENCE & TECHNOLOGY ...    0.966522
7634                     SOUTHWEST UNIVERSITY - CHINA    0.951955
9404                          UNIVERSITY OF VICTORIA    0.940694
7585                    SOUTH CHINA NORMAL UNIVERSITY    0.939383
4246                         ISLAMIC AZAD UNIVERSITY    0.937946
7604                     SOUTHEAST UNIVERSITY - CHINA    0.936249
9130                     UNIVERSITY OF MILANO-BICOCCA    0.930900
8994                            UNIVERSITY OF GRANADA    0.930886
5667                       NANJING NORMAL UNIVERSITY    0.917725
4697                            KING SAUD UNIVERSITY    0.909850
4673                       KING ABDULAZIZ UNIVERSITY    0.909119
8674                          UNIVERSITE DE LORRAINE    0.907234
4477                                 JILIN UNIVERSITY    0.904901
6293  NORWEGIAN UNIVERSITY OF SCIENCE & TECHNOLOGY (...    0.904843
9224                               UNIVERSITY OF PAVIA    0.902431
9151                            UNIVERSITY OF MUNSTER    0.902096
9233                               UNIVERSITY OF PISA    0.899670
728                          BEIJING NORMAL UNIVERSITY    0.898977
8642              UNIVERSITAT POLITECNICA DE CATALUNYA    0.896695
1360                   CHINA MEDICAL UNIVERSITY TAIWAN    0.895067
7415                             SHENZHEN UNIVERSITY    0.888467
9964                             ZHENGZHOU UNIVERSITY    0.888001
2215          ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE    0.884566
9416                             UNIVERSITY OF WATERLOO    0.883734
8290                UNIV BOURGOGNE FRANCHE-COMTE COMUE    0.880574
7782                             STONY BROOK UNIVERSITY    0.879509
8372                          UNIV NANTES ANGERS LE MANS    0.878168
3649  INDIAN INSTITUTE OF TECHNOLOGY SYSTEM (IIT SYS...    0.878145
8547                  UNIVERSIDADE ESTADUAL DE CAMPINAS    0.877905
8672                               UNIVERSITE DE LILLE    0.876648
1536         CITY UNIVERSITY OF NEW YORK (CUNY) SYSTEM    0.875992
9229                             UNIVERSITY OF PERUGIA    0.875758
8871                              UNIVERSITY OF BERGEN    0.873826
```

2. 让聚类算法分的更细一些

   第一种方法其实也挺好的，但是题目毕竟要求使用**聚类算法**

   我还是觉得应该**把切题放到首位**

   在前面我提到过，n_clusters=8 表示希望将所有高校分成 8个聚类

   如果人工调一下这个参数，比如 令n_clusters=100

   那所有高校就会被分成100个聚类，每个聚类的高校数量也会变少

   这样改动：

```
kmeans = KMeans(n_clusters=100, random_state=42, n_init=10)
labels = kmeans.fit_predict(X_scaled)
df["Cluster"] = labels
```

改动代码后，KMeans 聚类算法的运行结果如下：

```
PS D:\Program Files\Microsoft VS Code> & D:\python\python.exe d:/ECNU/数据科学导论/第六次作业/similar.py
85
与 EAST CHINA NORMAL UNIVERSITY 聚类相似的高校：
['BEIJING NORMAL UNIVERSITY' 'CARNEGIE MELLON UNIVERSITY'
 'CITY UNIVERSITY OF NEW YORK (CUNY) SYSTEM' 'CURTIN UNIVERSITY'
 'DELFT UNIVERSITY OF TECHNOLOGY' 'EAST CHINA NORMAL UNIVERSITY'
 'FLORIDA STATE UNIVERSITY' 'HONG KONG BAPTIST UNIVERSITY'
 'HONG KONG POLYTECHNIC UNIVERSITY'
 'HONG KONG UNIVERSITY OF SCIENCE & TECHNOLOGY'
 'NORWEGIAN UNIVERSITY OF SCIENCE & TECHNOLOGY (NTNU)'
 'QUEENSLAND UNIVERSITY OF TECHNOLOGY (QUT)' 'RICE UNIVERSITY'
 'UNIVERSITY OF BASQUE COUNTRY' 'UNIVERSITY OF BATH'
 'UNIVERSITY OF ELECTRONIC SCIENCE & TECHNOLOGY OF CHINA'
 'UNIVERSITY OF GRANADA' 'UNIVERSITY OF MACAU'
 'UNIVERSITY OF MISSOURI COLUMBIA' 'UNIVERSITY OF SOUTH CAROLINA COLUMBIA'
 'UNIVERSITY OF SOUTH CAROLINA SYSTEM' 'UNIVERSITY OF VICTORIA'
 'UNIVERSITY OF WATERLOO']
```

在实验报告的最后，我来分析一下这些高校与华师大相似的**原因**：

1. 师范类 / 教育与基础科学强校

   代表高校：北京师范大学、香港教育类大学

   这些学校在教育学、心理学、数学、环境科学等人文理交叉学科上布局与华师大非常接近；都是“教育+理科+环境+计算机”结构。

2. 理工与信息科学综合型大学

   代表高校：华南理工、电子科技大学、香港科技大学、挪威科技大学

   华师大近年理工科（尤其是计算机、信息科学、环境科学）发展迅猛，与这些“中等规模理工型大学”结构接近。

3. 工科强校但文理基础扎实

   代表高校：卡内基梅隆大学、德尔夫特理工大学、昆士兰科技大学

   这些大学在信息科学、人工智能、计算机、地球环境等领域的科研产出与华师大相似，尤其是AI与地球科学交叉研究。

4. 综合性大学

   代表高校：南加州大学体系、南卡罗来纳大学、马卡奥大学、格拉纳达大学、巴斯大学

   它们的科研特征是“多学科中等水平、部分学科突出”，整体分布平衡，与华师大ESI曲线（无极端强势学科，但全面均衡）接近。

5. 技术应用型大学

   代表高校：香港理工大学、香港城市大学、佛罗里达州立大学

   注重工程、社会科学与教育、环境等跨领域融合，这正是华师大在“双一流”中努力拓展的方向。