Project Team Goals

- Generate feature matrices for references, citations, journals, MSC, etc.
- Perform SVD and PCA as an initial step for dimensionality reduction.
- Check the quality of the resulting clusters and use it to describe the department's strength.
- Explore further clustering methods: specifically, Spectral Clustering.

Introduction of Datasets

- Citations:
- **Rows:** Represent different professors.
- **Columns:** Represent unique identifiers for different papers.
- **Entries:** Indicate the number of times a particular paper (column) has cited any paper by a particular professor (row).

	MR4504450	MR3933391	MR4358671	MR4500640	MR4312365	MR4310891	MR4292179	MR4457373	MR4411466
AhlgrenScottD	0	0	0	0	0	0	0	0	0
AlbinPierre	0	0	0	0	0	0	0	0	0
BaloghJózsef	0	0	0	0	0	0	0	0	0
BaryshnikovYuliyM	0	0	0	0	0	0	0	0	0
Berwick-EvansDaniel	3	1	2	1	1	1	1	1	1
...	...	...	...	...	...	...	...	...	...
WuXuan	0	0	0	0	0	0	0	0	0
YongAlexanderTF	0	0	0	0	0	0	0	0	0
YoungAmanda	0	0	0	0	0	0	0	0	0
ZaharescuAlexandru	0	0	0	0	0	0	0	0	0
ZharnitskyVadim	0	0	0	0	0	0	0	0	0

70 rows × 22176 columns

Introduction of Datasets

- Papers:
- **Rows:** Represent different professors.
- **Columns:** represent papers (by their unique identifiers)
- **Entries:** indicate the number of times a particular professor (in a given row) has cited a particular paper (in the corresponding column)

	MR2273359	MR2275343	MR1833071	MR2773200	MR2763082	MR2543662	MR3097158	MR4107507	MR2145
AhlgrenScottD_papers	0	0	0	0	0	0	0	0	
AlbinPierre_papers	0	0	0	0	0	0	0	0	
BaloghJózsef_papers	0	0	0	0	0	0	0	0	
BaryshnikovYuliyM_papers	0	0	0	0	0	0	0	0	
Berwick-EvansDaniel_papers	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	
WuXuan_papers	0	0	0	0	0	0	0	0	
YongAlexanderTF_papers	0	0	0	0	0	0	0	0	
YoungAmanda_papers	0	0	0	0	0	0	0	0	
ZaharescuAlexandru_papers	1	1	13	2	0	2	1	0	
ZharnitskyVadim_papers	0	0	0	0	0	0	0	0	

70 rows × 26304 columns

Data Preprocess

- More than 90 percent of the cols have just 1 faculty linked to it. In other words, the vectors spanning the columns are nearly orthogonal.

```
#remove all the cols with one or zero non-zero entries.
mask = (papers != 0).sum(axis=0) == 1
cols_to_drop = mask[mask].index.tolist()
papers = papers.drop(columns=cols_to_drop)
mask = (papers != 0).sum(axis=0) == 0
cols_to_drop = mask[mask].index.tolist()
papers = papers.drop(columns=cols_to_drop)
```

[9] ✓ 0.0s Python

Data Normalization

- Then apply L1 and L2 normalization separately

```python
#L1 & L2 Noramalization

papers_l1 = papers.divide(papers.abs().sum(axis=1), axis=0)
l2_norm = papers.pow(2).sum(axis=1).pow(0.5)
papers_l2 = papers.divide(l2_norm, axis=0)
```

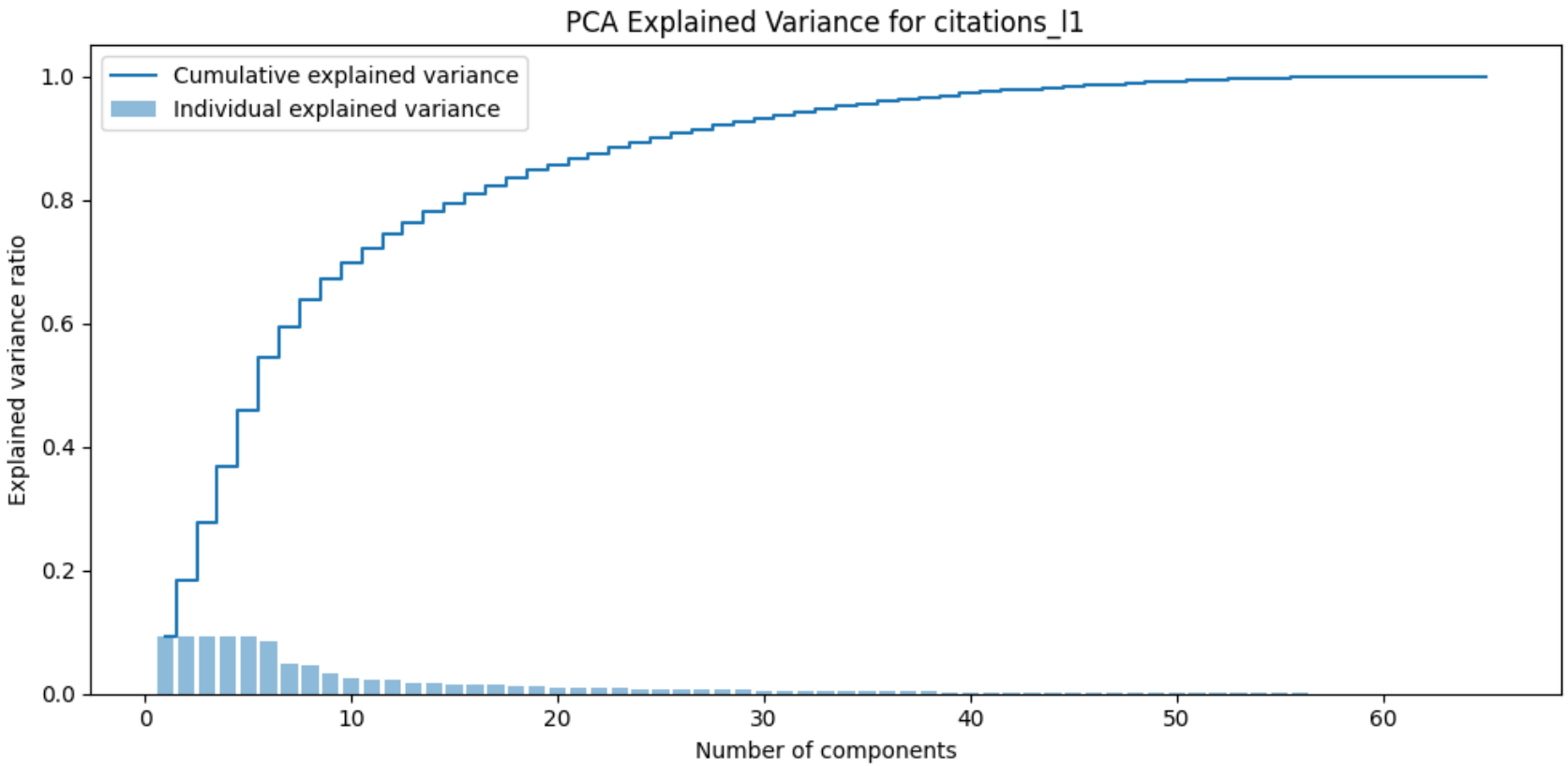
✓ 0.0s Python

```python
citations_l1 = citations.divide(citations.abs().sum(axis=1), axis=0)
l2_norm = citations.pow(2).sum(axis=1).pow(0.5)
citations_l2 = citations.divide(l2_norm, axis=0)
```

✓ 0.0s Python

PCA Process

- Plot the explained variance for each component
- Plot the cumulative explained variance



PCA Process

- Apply PCA transformation to each data frame

```
#Apply pca to the our df
pca = PCA(n_components=2)

papers_principalComponents_l1 = pca.fit_transform(papers_l1)
papers_principalComponents_l2 = pca.fit_transform(papers_l2)

citations_principalComponents_l1 = pca.fit_transform(citations_l1)
citations_principalComponents_l2 = pca.fit_transform(citations_l2)
```

✓ 0.0s Python

```
papers_l1_pca = pd.DataFrame(papers_principalComponents_l1, columns=['PC1', 'PC2'])
papers_l1_pca.index = papers_l1.index

print(papers_l1_pca)

papers_l2_pca = pd.DataFrame(papers_principalComponents_l2, columns=['PC1', 'PC2'])
papers_l2_pca.index = papers_l2.index

print(papers_l2_pca)
```

✓ 0.0s Python

Scatterplot

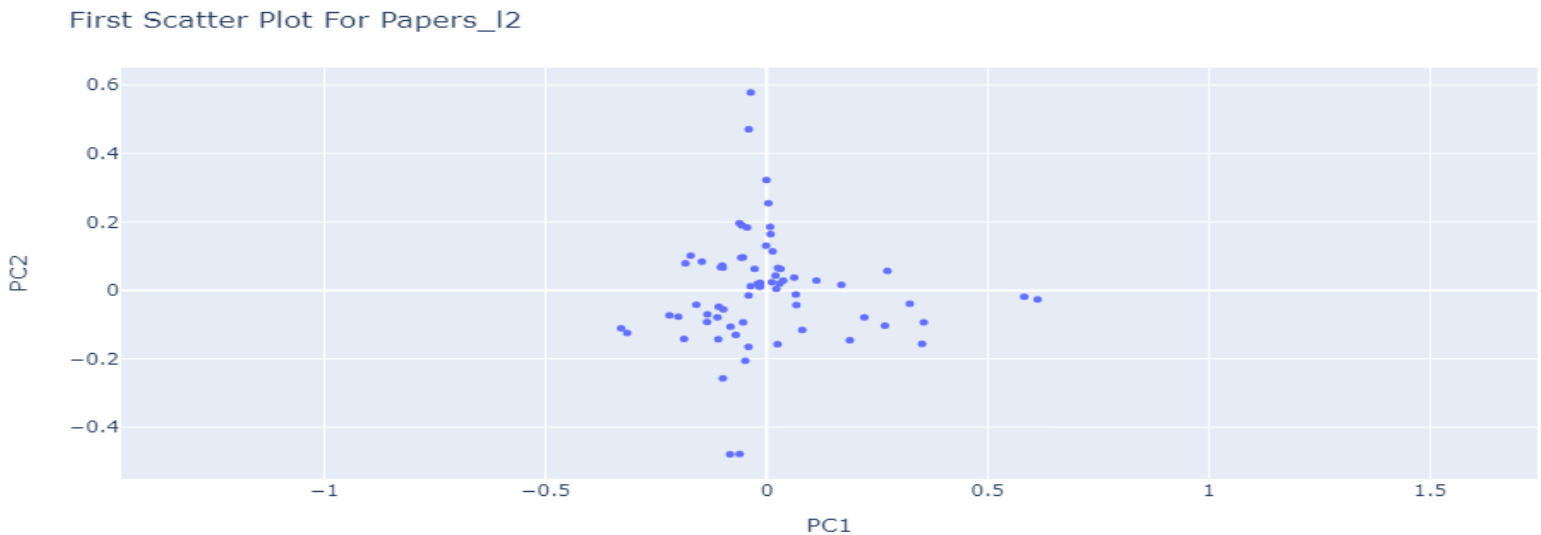


Figure 1: First Papers Plot with L2

Scatterplot

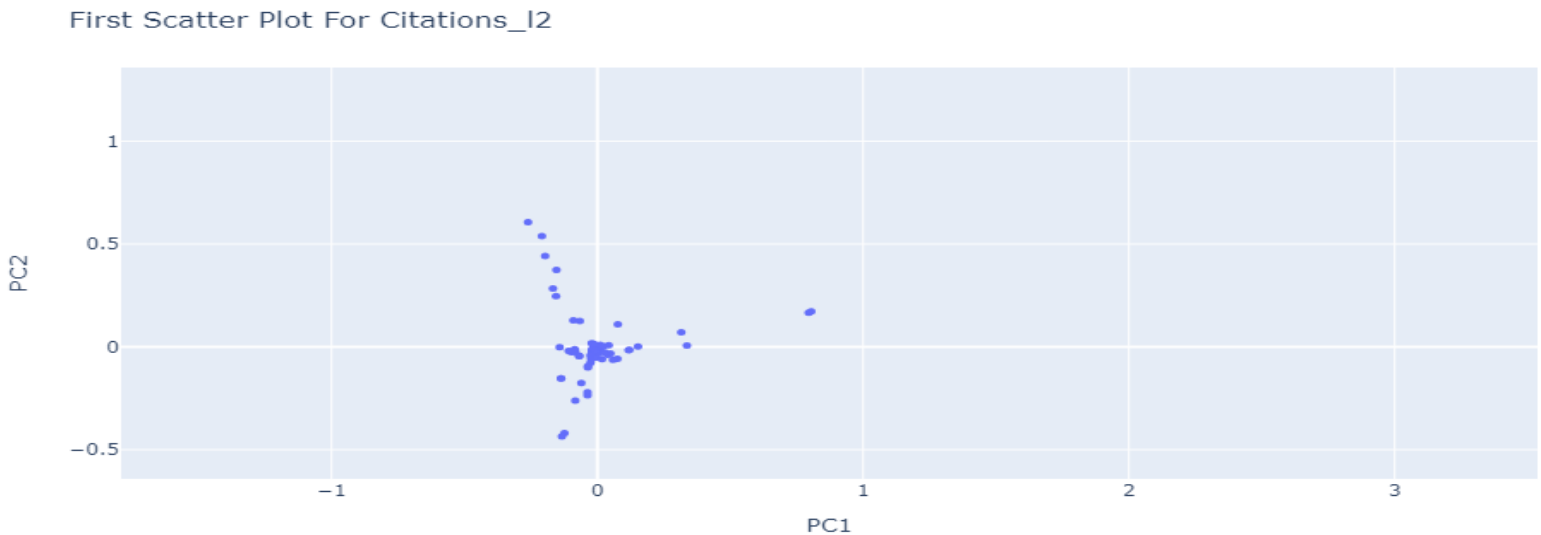


Figure 2: First Citations Plot with L2

K-Means

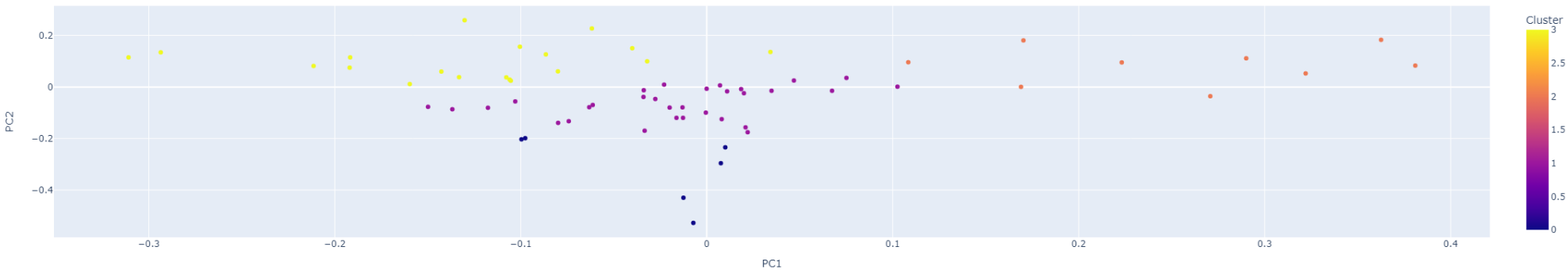


Figure 3: Papers with L2 after removing outliers

K-Means

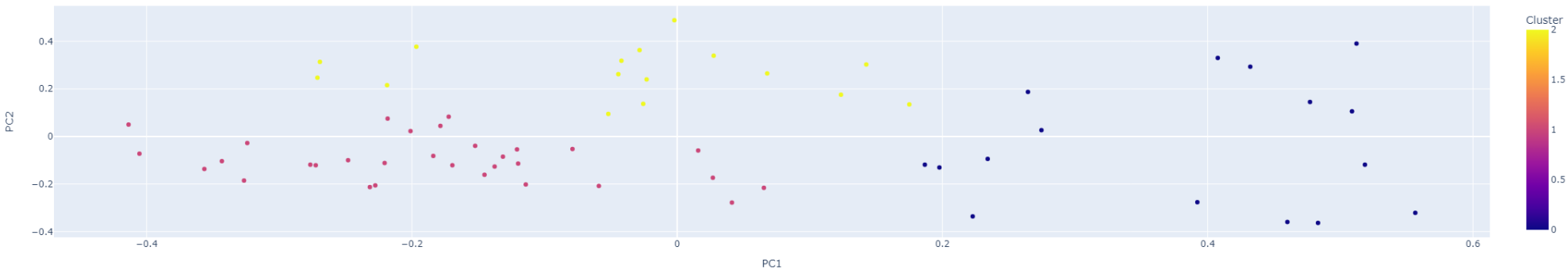


Figure 4: Citations with L2 after removing outliers