

```
In [1]: 1 import pandas as pd  
2 import pycaret  
3 import seaborn as sns  
4 import matplotlib.pyplot as plt
```

```
In [2]: 1 dataset = pd.read_csv('data_java_21_aggregated_not_zero.csv', sep = ';')
```

```
In [3]: 1 dataset.shape
```

Out[3]: (102, 27)

```
In [4]: 1 dataset.head()
```

Out[4]:

	intro	intro-VPL	methods	methods-VPL	encap	encap-VPL	constructors	constructors-VPL	classes1	classes1-VPL	...	polymorphism-VPL	intro-VPL.1	methods-VPL-f	encap-VPL-f	constructors-VPL-f	clas
0	110	60	160	630	148	100	110	165	80	0	...	45	60	630	100	165	
1	110	60	160	630	150	100	110	165	80	0	...	45	60	630	100	165	
2	110	60	160	630	150	100	110	165	80	0	...	45	60	630	100	165	
3	110	60	160	630	150	100	110	165	80	0	...	45	60	630	100	165	
4	110	60	160	630	150	100	110	157	80	0	...	45	60	630	100	157	

5 rows × 27 columns



In [5]: 1 | dataset.tail(20)

Out[5]:

	intro	intro-VPL	methods	methods-VPL	encap	encap-VPL	constructors	constructors-VPL	classes1	classes1-VPL	...	polymorphism-VPL	intro-VPL.1	methods-VPL-f	encap-VPL-f	constructors-VPL-f	c
82	50	0	60	0	30	0	50	0	0	0	...	0	0	0	0	0	0
83	97	20	0	100	10	70	0	124	0	0	...	0	20	100	70	124	
84	10	20	0	140	50	40	0	60	0	0	...	0	20	140	40	60	
85	10	20	0	140	50	40	0	60	0	0	...	0	20	140	40	60	
86	0	60	0	60	10	70	0	98	0	0	...	0	60	60	70	98	
87	71	60	93	183	0	0	0	0	0	0	...	0	60	183	0	0	
88	0	0	0	188	0	40	0	65	0	0	...	0	0	188	40	65	
89	85	0	128	80	26	34	0	0	0	0	...	0	0	80	34	0	
90	107	60	0	100	0	20	0	0	0	0	...	0	60	100	20	0	
91	106	60	50	40	0	0	0	0	0	0	...	0	60	40	0	0	
92	102	0	0	80	0	0	0	0	0	0	...	0	0	80	0	0	
93	71	0	30	40	20	0	0	0	0	0	...	0	0	40	0	0	
94	0	0	0	90	0	40	0	30	0	0	...	0	0	90	40	30	
95	97	60	0	0	0	0	0	0	0	0	...	0	60	0	0	0	
96	91	20	0	40	0	0	0	0	0	0	...	0	20	40	0	0	
97	40	0	0	97	0	10	0	0	0	0	...	0	0	97	10	0	
98	98	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
99	0	0	0	0	20	0	0	0	0	0	...	0	0	0	0	0	
100	20	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
101	0	20	0	0	0	0	0	0	0	0	...	0	20	0	0	0	

20 rows × 27 columns



In [6]: 1 dataset.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102 entries, 0 to 101
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   intro            102 non-null    int64  
 1   intro-VPL        102 non-null    int64  
 2   methods          102 non-null    int64  
 3   methods-VPL      102 non-null    int64  
 4   encaps           102 non-null    int64  
 5   encaps-VPL       102 non-null    int64  
 6   constructors     102 non-null    int64  
 7   constructors-VPL 102 non-null    int64  
 8   classes1         102 non-null    int64  
 9   classes1-VPL     102 non-null    int64  
 10  classes2         102 non-null    int64  
 11  classes2-VPL     102 non-null    int64  
 12  static            102 non-null    int64  
 13  static-VPL        102 non-null    int64  
 14  inher             102 non-null    int64  
 15  inher-VPL        102 non-null    int64  
 16  polymorphism      102 non-null    int64  
 17  polymorphism-VPL 102 non-null    int64  
 18  intro-VPL.1      102 non-null    int64  
 19  methods-VPL-f    102 non-null    int64  
 20  encaps-VPL-f    102 non-null    int64  
 21  constructors-VPL-f 102 non-null    int64  
 22  classes2-VPL-f   102 non-null    int64  
 23  static-VPL-f     102 non-null    int64  
 24  inher-VPL-f      102 non-null    int64  
 25  polymorphism-VPL-f 102 non-null    int64  
 26  sum_vpl          102 non-null    int64  
dtypes: int64(27)
memory usage: 21.6 KB
```

In [7]: 1 | dataset.describe().T

Out[7]:

	count	mean	std	min	25%	50%	75%	max
<b>intro</b>	102.0	94.509804	30.101066	0.0	97.25	110.0	110.00	110.0
<b>intro-VPL</b>	102.0	46.196078	23.239621	0.0	40.00	60.0	60.00	60.0
<b>methods</b>	102.0	123.421569	59.649544	0.0	102.50	159.0	160.00	160.0
<b>methods-VPL</b>	102.0	348.843137	224.051875	0.0	107.50	346.0	584.50	630.0
<b>encap</b>	102.0	108.745098	58.633901	0.0	71.75	146.5	150.00	150.0
<b>encap-VPL</b>	102.0	71.598039	38.642115	0.0	40.00	100.0	100.00	100.0
<b>constructors</b>	102.0	79.117647	44.749533	0.0	51.00	105.5	110.00	110.0
<b>constructors-VPL</b>	102.0	112.637255	63.217179	0.0	60.00	139.0	165.00	165.0
<b>classes1</b>	102.0	52.401961	34.236007	0.0	0.00	73.0	80.00	80.0
<b>classes1-VPL</b>	102.0	0.000000	0.000000	0.0	0.00	0.0	0.00	0.0
<b>classes2</b>	102.0	109.450980	71.353482	0.0	32.50	150.0	169.75	170.0
<b>classes2-VPL</b>	102.0	54.166667	41.345462	0.0	0.00	65.0	95.00	95.0
<b>static</b>	102.0	57.362745	33.316303	0.0	40.00	77.0	80.00	80.0
<b>static-VPL</b>	102.0	147.950980	109.225027	0.0	38.00	170.0	270.00	270.0
<b>inher</b>	102.0	116.568627	75.730569	0.0	40.00	161.5	186.50	190.0
<b>inher-VPL</b>	102.0	139.941176	110.768944	0.0	27.00	158.5	270.00	270.0
<b>polymorphism</b>	102.0	45.107843	31.401238	0.0	0.00	69.0	70.00	70.0
<b>polymorphism-VPL</b>	102.0	26.647059	21.434518	0.0	0.00	45.0	45.00	45.0
<b>intro-VPL.1</b>	102.0	46.196078	23.239621	0.0	40.00	60.0	60.00	60.0
<b>methods-VPL-f</b>	102.0	348.843137	224.051875	0.0	107.50	346.0	584.50	630.0
<b>encap-VPL-f</b>	102.0	71.598039	38.642115	0.0	40.00	100.0	100.00	100.0
<b>constructors-VPL-f</b>	102.0	112.637255	63.217179	0.0	60.00	139.0	165.00	165.0
<b>classes2-VPL-f</b>	102.0	54.166667	41.345462	0.0	0.00	65.0	95.00	95.0
<b>static-VPL-f</b>	102.0	147.950980	109.225027	0.0	38.00	170.0	270.00	270.0
<b>inher-VPL-f</b>	102.0	139.941176	110.768944	0.0	27.00	158.5	270.00	270.0
<b>polymorphism-VPL-f</b>	102.0	26.647059	21.434518	0.0	0.00	45.0	45.00	45.0
<b>sum_vpl</b>	102.0	1734.666667	927.242140	20.0	969.00	2025.5	2570.50	2753.0

In [ ]:

1

## reduced dataset

In [8]:

```
1 dataset_reduced = dataset[['intro','intro-VPL','methods','methods-VPL','encap','encap-VPL','constructors',
2                               'constructors-VPL','classes1','classes2','classes2-VPL','static','static-VPL',
3                               'inher','inher-VPL','polymorphism','polymorphism-VPL']]
```

In [9]:

```
1 dataset_reduced = dataset_reduced.astype(float)
```

In [10]:

```
1 dataset_reduced.describe().T
```

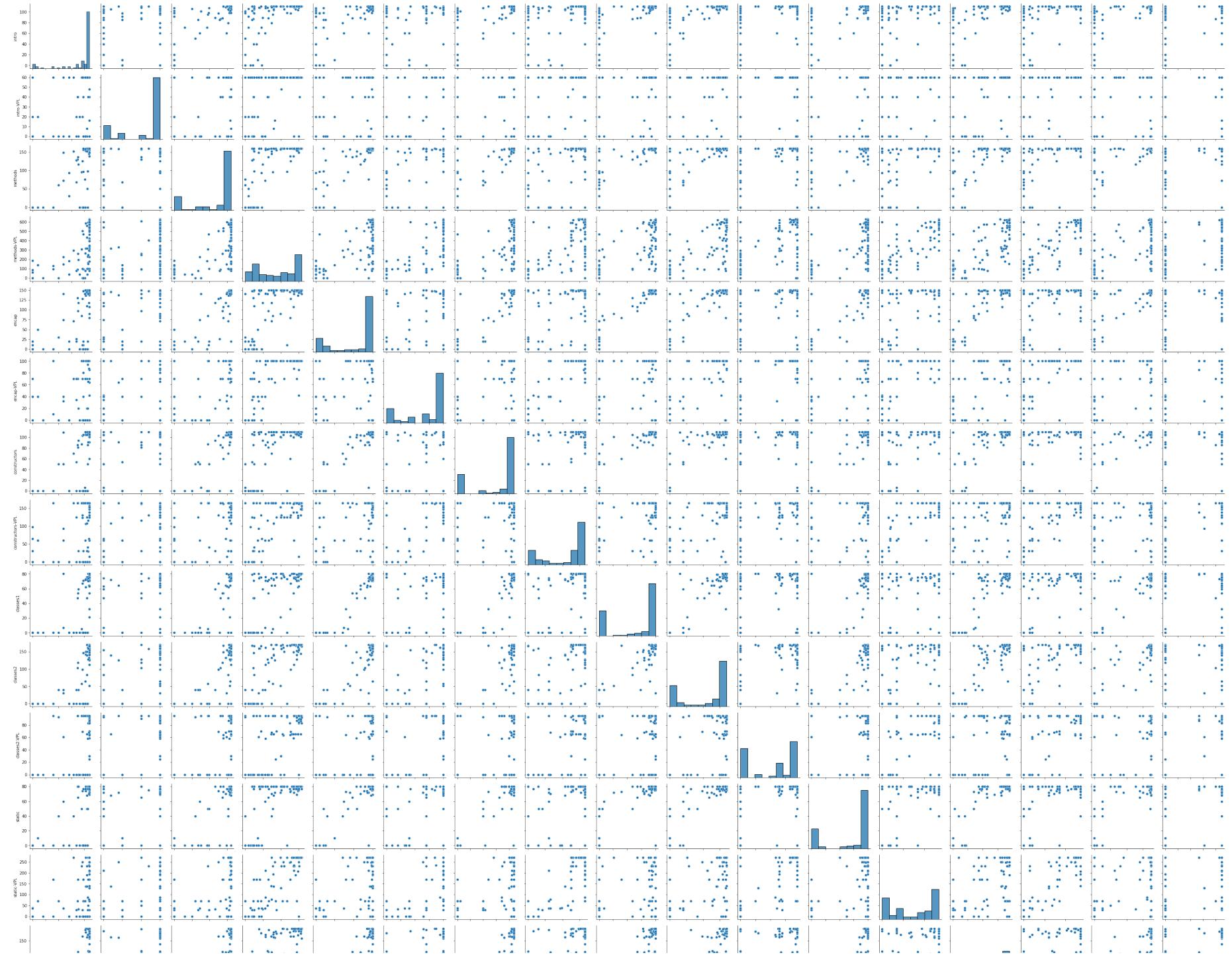
Out[10]:

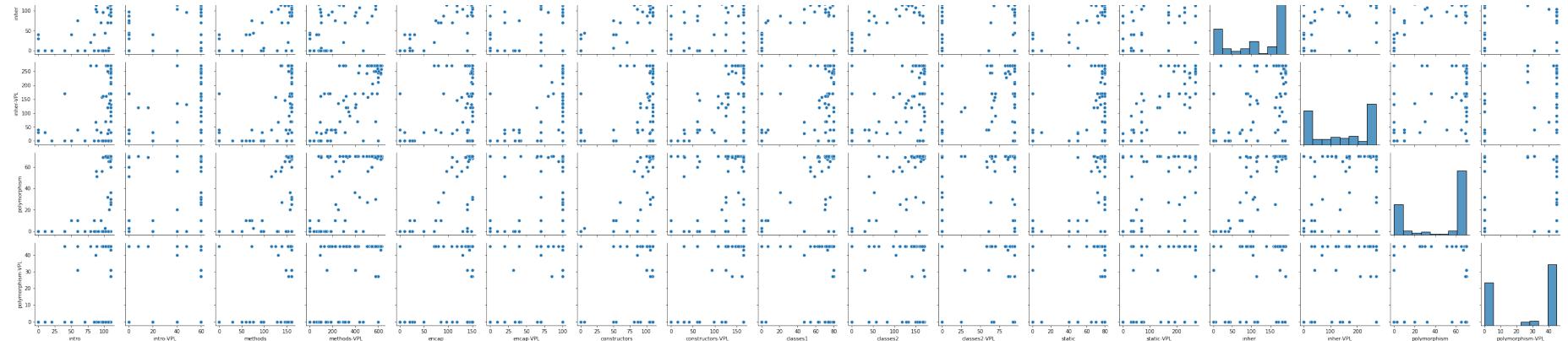
	count	mean	std	min	25%	50%	75%	max
<b>intro</b>	102.0	94.509804	30.101066	0.0	97.25	110.0	110.00	110.0
<b>intro-VPL</b>	102.0	46.196078	23.239621	0.0	40.00	60.0	60.00	60.0
<b>methods</b>	102.0	123.421569	59.649544	0.0	102.50	159.0	160.00	160.0
<b>methods-VPL</b>	102.0	348.843137	224.051875	0.0	107.50	346.0	584.50	630.0
<b>encap</b>	102.0	108.745098	58.633901	0.0	71.75	146.5	150.00	150.0
<b>encap-VPL</b>	102.0	71.598039	38.642115	0.0	40.00	100.0	100.00	100.0
<b>constructors</b>	102.0	79.117647	44.749533	0.0	51.00	105.5	110.00	110.0
<b>constructors-VPL</b>	102.0	112.637255	63.217179	0.0	60.00	139.0	165.00	165.0
<b>classes1</b>	102.0	52.401961	34.236007	0.0	0.00	73.0	80.00	80.0
<b>classes2</b>	102.0	109.450980	71.353482	0.0	32.50	150.0	169.75	170.0
<b>classes2-VPL</b>	102.0	54.166667	41.345462	0.0	0.00	65.0	95.00	95.0

```
In [11]: 1 g = sns.PairGrid(dataset_reduced)
2 g.map_diag(sns.histplot)
3 g.map_offdiag(sns.scatterplot)
```

```
Out[11]: <seaborn.axisgrid.PairGrid at 0x166165bc908>
```



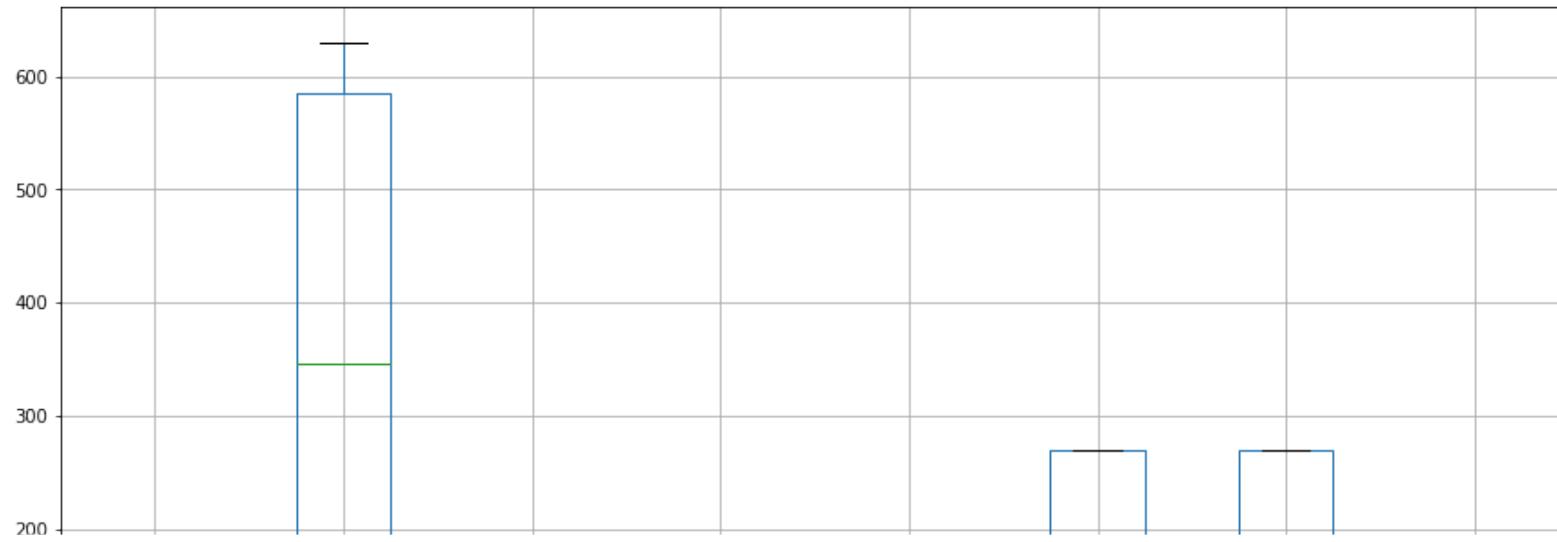




## VPL

```
In [12]: 1 dataset_reduced.boxplot(column = ['intro-VPL','methods-VPL','encap-VPL','constructors-VPL','classes2-VPL','static-VPL',
2                                         'inher-VPL', 'polymorphism-VPL'],figsize=(15,8))
```

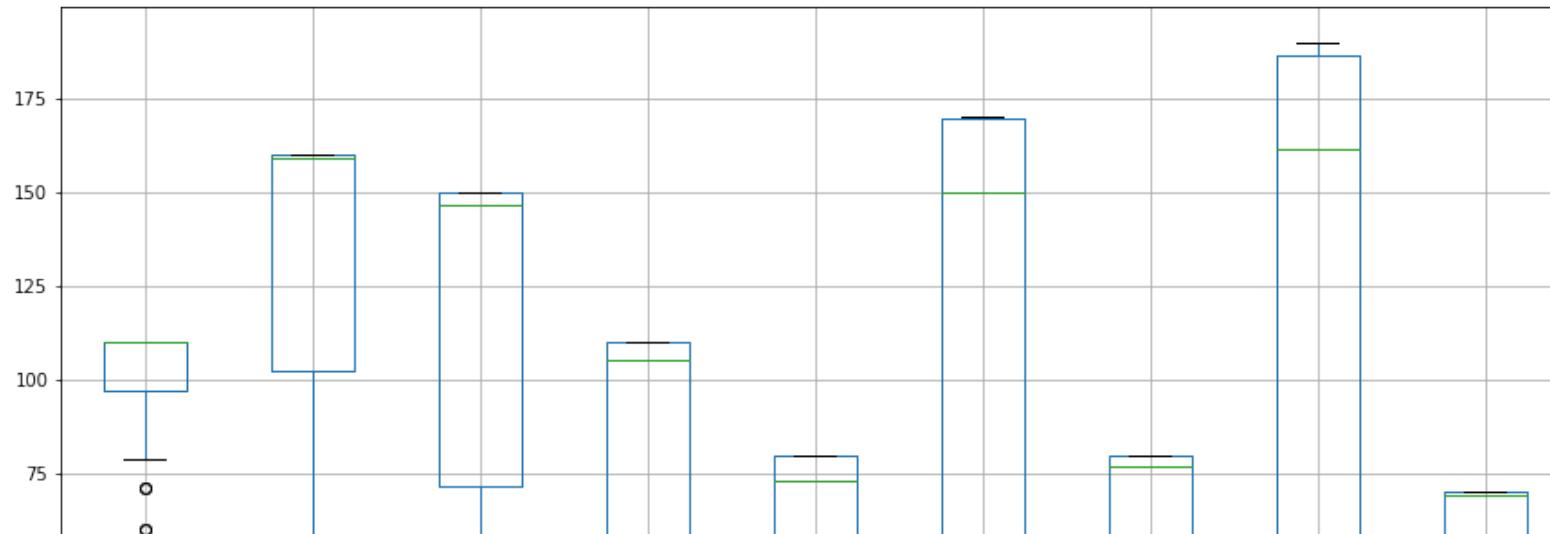
Out[12]: <AxesSubplot:>



## Micro-lessons

```
In [13]: 1 dataset_reduced.boxplot(column = ['intro','methods','encap','constructors','classes1','classes2','static', 'inher', 'polymorphis'])
```

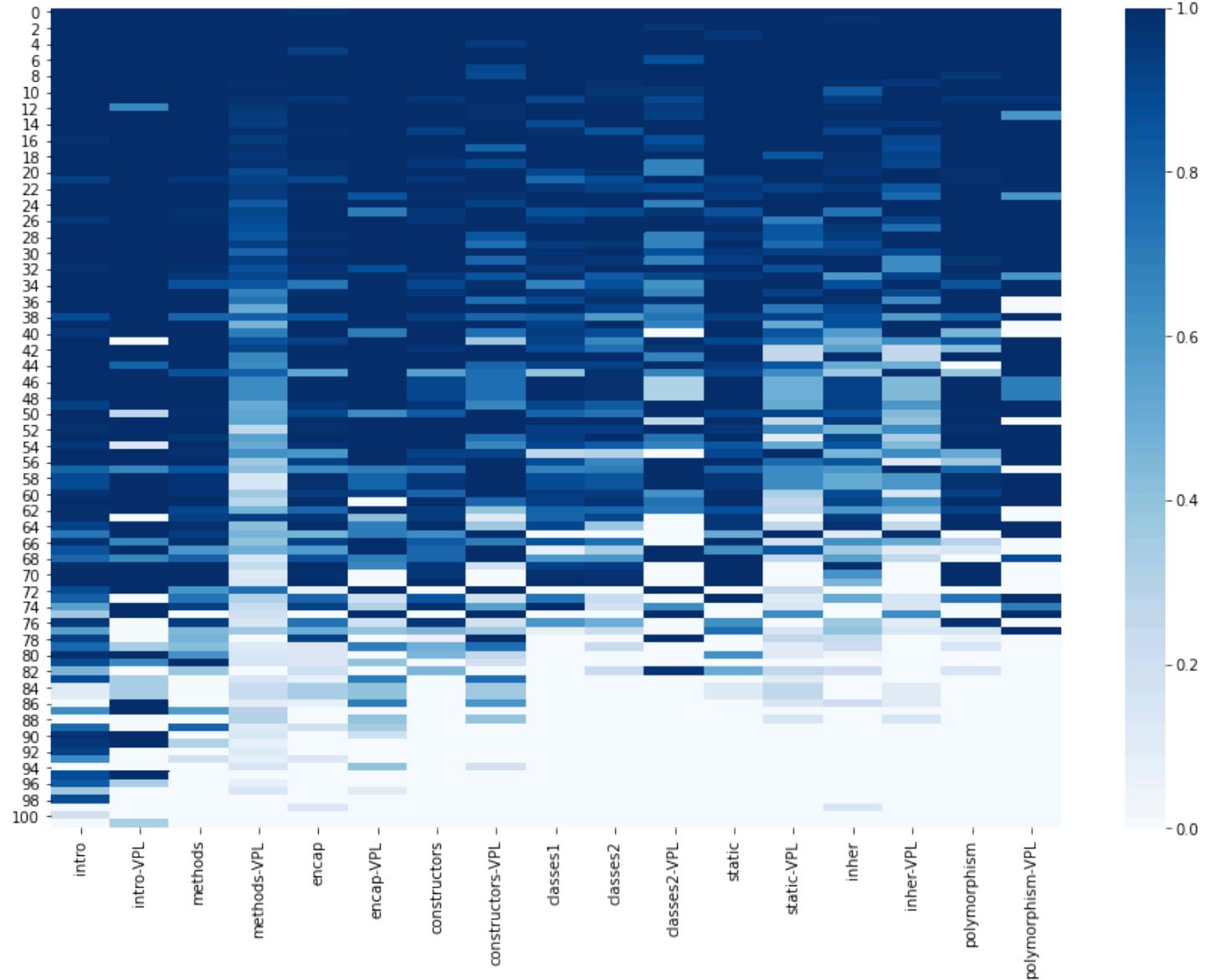
```
Out[13]: <AxesSubplot:>
```



```
In [14]: 1 # copy the data
2 dataset_reduced_min_max_scaled = dataset_reduced.copy()
3
4 # apply normalization techniques
5 for column in dataset_reduced_min_max_scaled.columns:
6     dataset_reduced_min_max_scaled[column] = (dataset_reduced_min_max_scaled[column] - dataset_reduced_min_max_scaled[column].min()
7
8
```

```
In [15]: 1 fig, ax = plt.subplots(figsize=(15, 10))
2 sns.heatmap(dataset_reduced_min_max_scaled, cmap='Blues')
```

Out[15]: <AxesSubplot:>



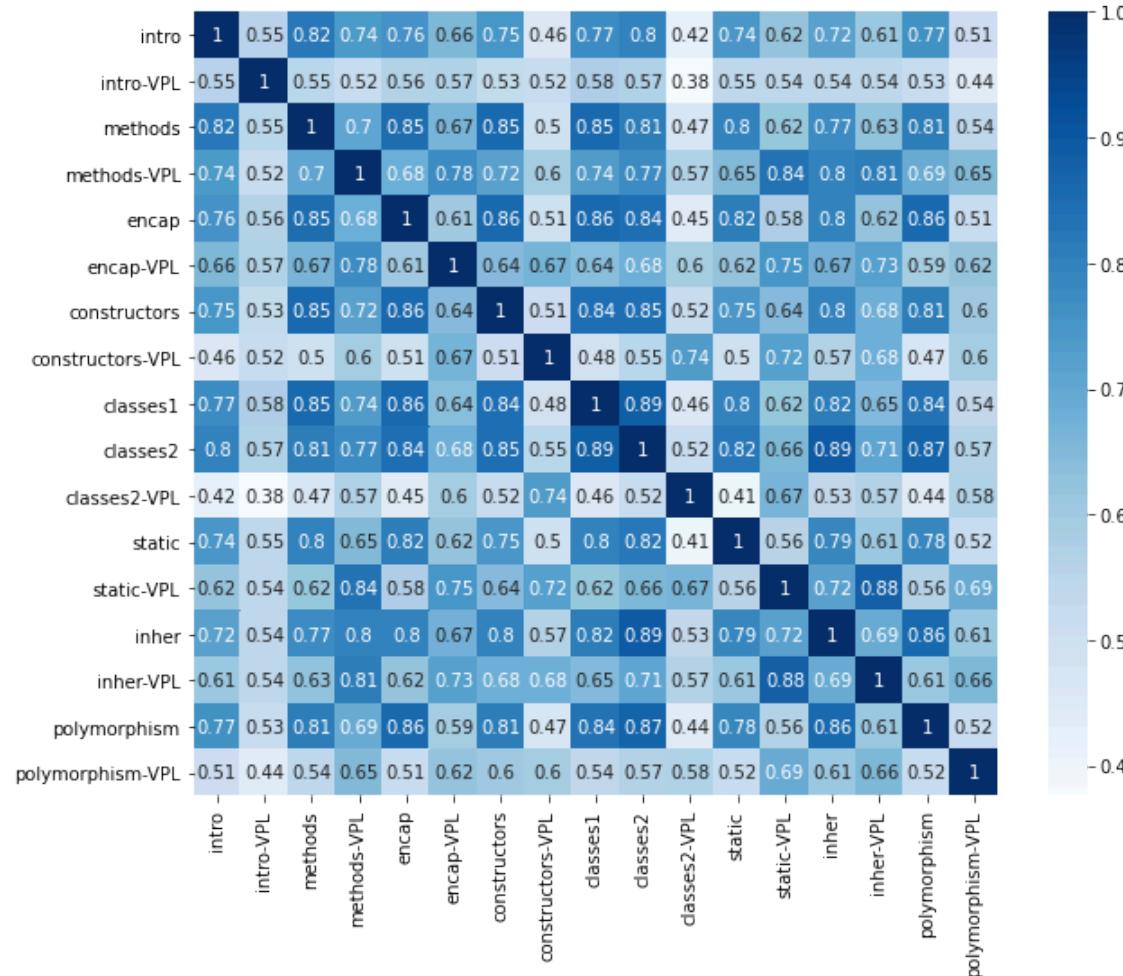
```
In [16]: 1 dataset_reduced.corr(method='spearman')
```

Out[16]:

	intro	intro-VPL	methods	methods-VPL	encap	encap-VPL	constructors	constructors-VPL	classes1	classes2	classes2-VPL	static	static-VPL	inh
intro	1.000000	0.554696	0.823184	0.741630	0.756987	0.655632	0.750949	0.462903	0.769973	0.799437	0.422968	0.744914	0.617819	0.71944
intro-VPL	0.554696	1.000000	0.553167	0.521474	0.564133	0.570029	0.529508	0.519096	0.578190	0.573242	0.376203	0.548742	0.539914	0.5422
methods	0.823184	0.553167	1.000000	0.703647	0.850211	0.668541	0.854673	0.501322	0.854209	0.814883	0.471693	0.804898	0.615815	0.77444
methods-VPL	0.741630	0.521474	0.703647	1.000000	0.684170	0.780068	0.718098	0.597180	0.738520	0.771162	0.571024	0.651329	0.837327	0.80310
encap	0.756987	0.564133	0.850211	0.684170	1.000000	0.607419	0.862756	0.505264	0.862313	0.842227	0.446467	0.816903	0.576520	0.80180
encap-VPL	0.655632	0.570029	0.668541	0.780068	0.607419	1.000000	0.644660	0.669163	0.642104	0.681103	0.597559	0.620922	0.751434	0.67362
constructors	0.750949	0.529508	0.854673	0.718098	0.862756	0.644660	1.000000	0.512340	0.844933	0.845751	0.516844	0.748485	0.644834	0.79720
constructors-VPL	0.462903	0.519096	0.501322	0.597180	0.505264	0.669163	0.512340	1.000000	0.483317	0.551548	0.738707	0.502234	0.724223	0.56539
classes1	0.769973	0.578190	0.854209	0.738520	0.862313	0.642104	0.844933	0.483317	1.000000	0.886645	0.460544	0.797561	0.616972	0.81638
classes2	0.700437	0.573242	0.814883	0.771162	0.842227	0.681103	0.845751	0.551548	0.551548	1.000000	0.502234	0.800736	0.656600	0.80176

```
In [17]: 1 fig, ax = plt.subplots(figsize=(10, 8))
2 sns.heatmap(dataset_reduced.corr(method='spearman'),
3             xticklabels=dataset_reduced.columns,
4             yticklabels=dataset_reduced.columns, cmap='Blues', annot=True)
```

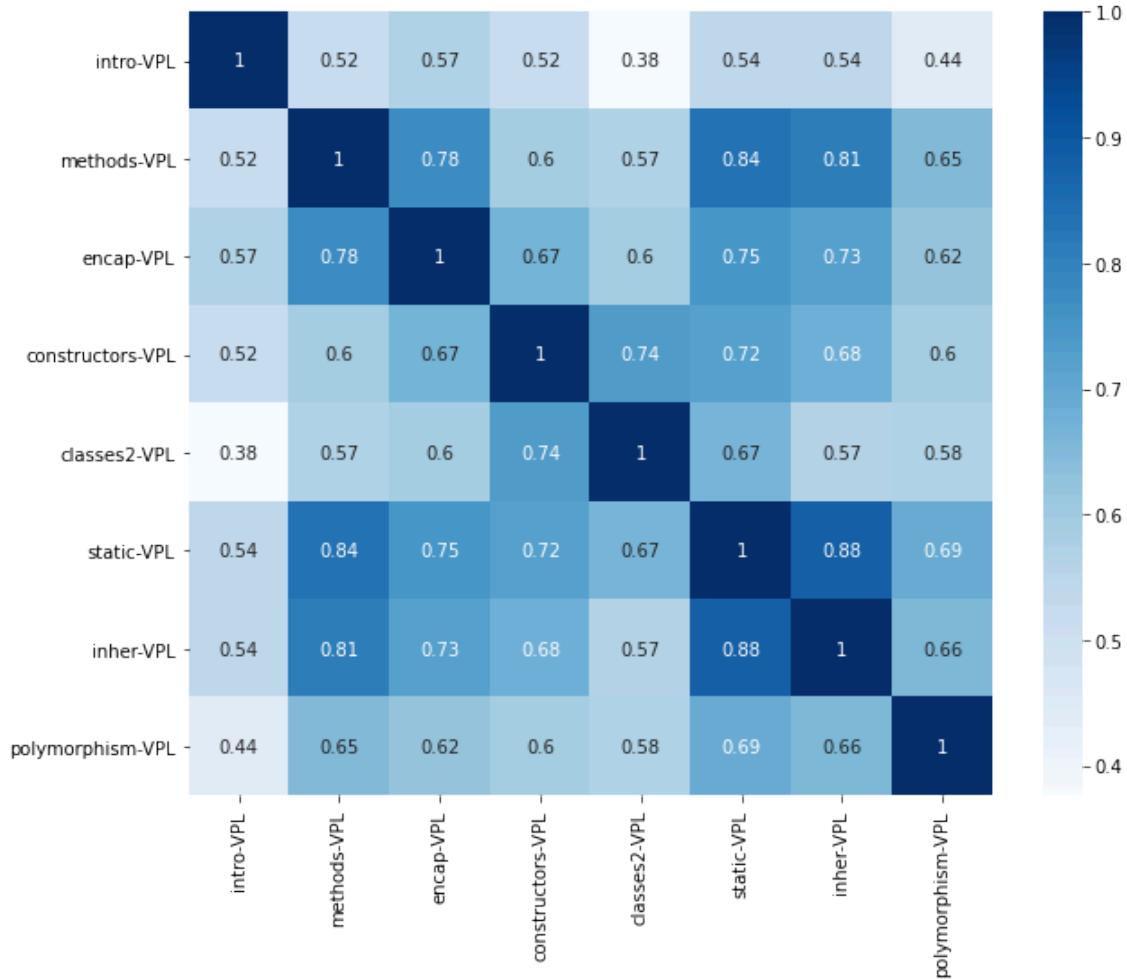
Out[17]: <AxesSubplot:>



## VPL

```
In [18]: 1 dataset_reduced_VPL = dataset_reduced[['intro-VPL', 'methods-VPL', 'encap-VPL', 'constructors-VPL', 'classes2-VPL', 'static-VPL',
2                                         'inher-VPL',   'polymorphism-VPL']]
3 fig, ax = plt.subplots(figsize=(10, 8))
4 sns.heatmap(dataset_reduced_VPL.corr(method='spearman'),
5             xticklabels=dataset_reduced_VPL.columns,
6             yticklabels=dataset_reduced_VPL.columns, cmap='Blues', annot=True)
```

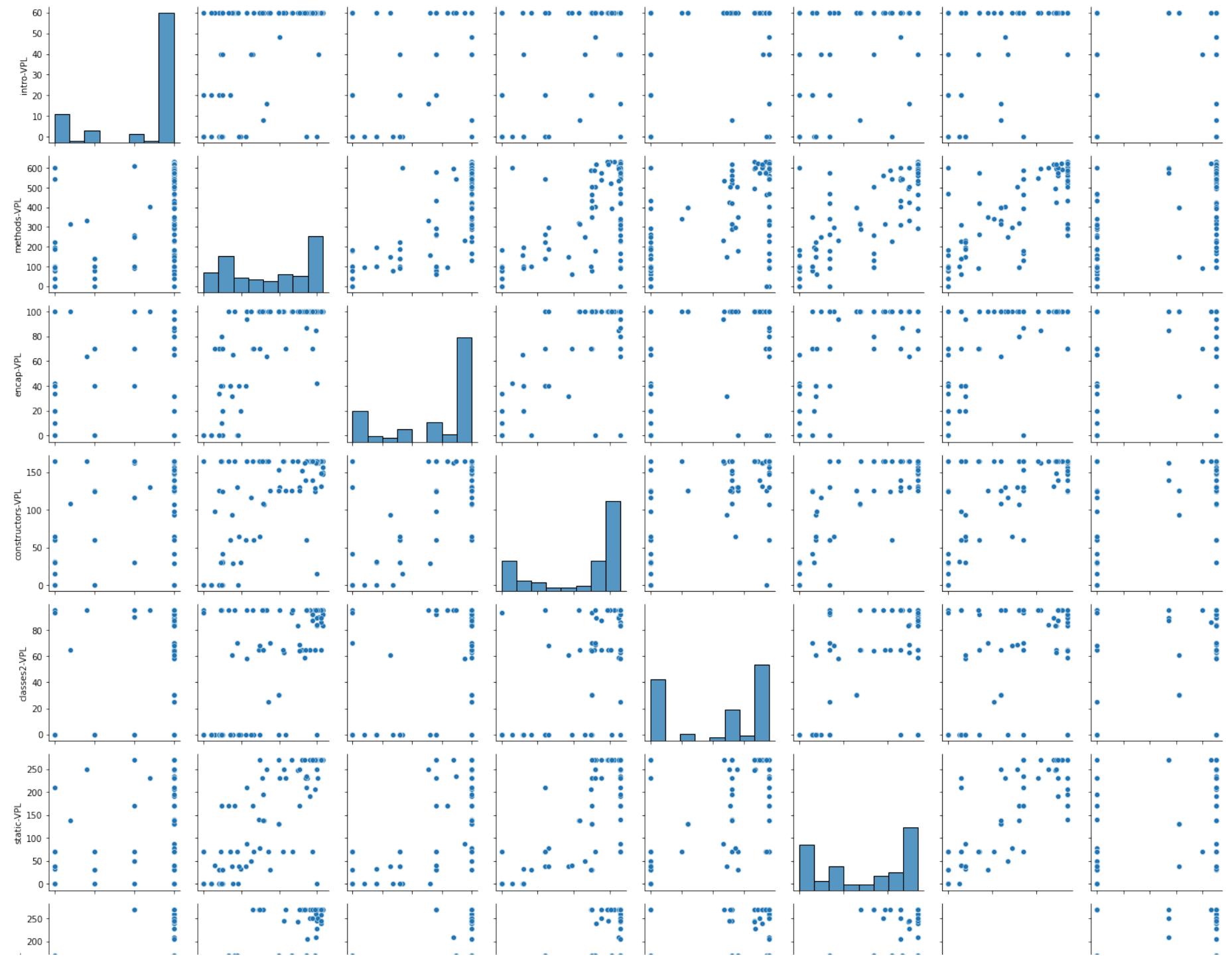
Out[18]: <AxesSubplot:>

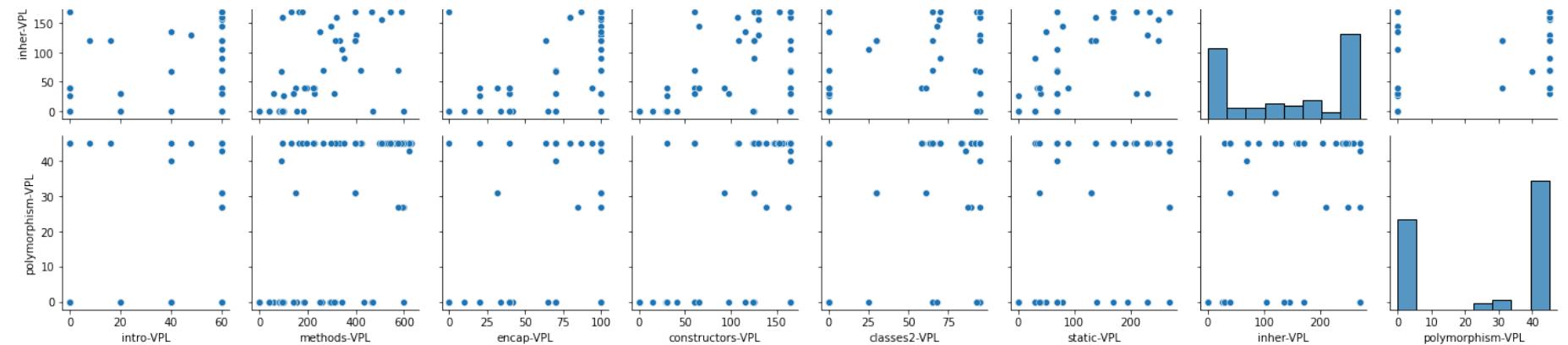


```
In [19]: 1 g_vpl = sns.PairGrid(dataset_reduced_VPL)
2 g_vpl.map_diag(sns.histplot)
3 g_vpl.map_offdiag(sns.scatterplot)
```

```
Out[19]: <seaborn.axisgrid.PairGrid at 0x166243b4c50>
```



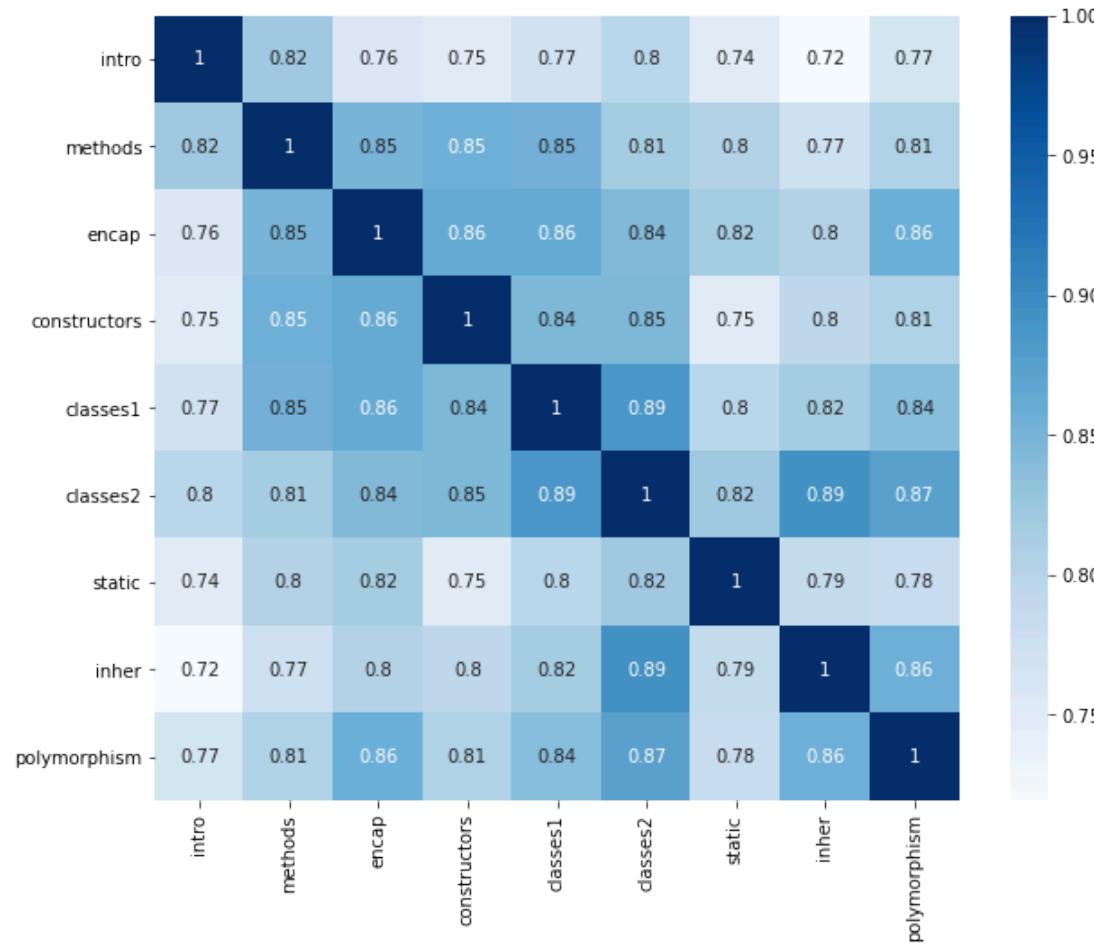




## Micro-lessons

```
In [20]: 1 dataset_reduced_mc_lessons = dataset_reduced[['intro', 'methods', 'encap', 'constructors', 'classes1', 'classes2', 'static', 'inher',  
2 fig, ax = plt.subplots(figsize=(10, 8))  
3 sns.heatmap(dataset_reduced_mc_lessons.corr(method='spearman'),  
4             xticklabels=dataset_reduced_mc_lessons.columns,  
5             yticklabels=dataset_reduced_mc_lessons.columns, cmap='Blues', annot=True)
```

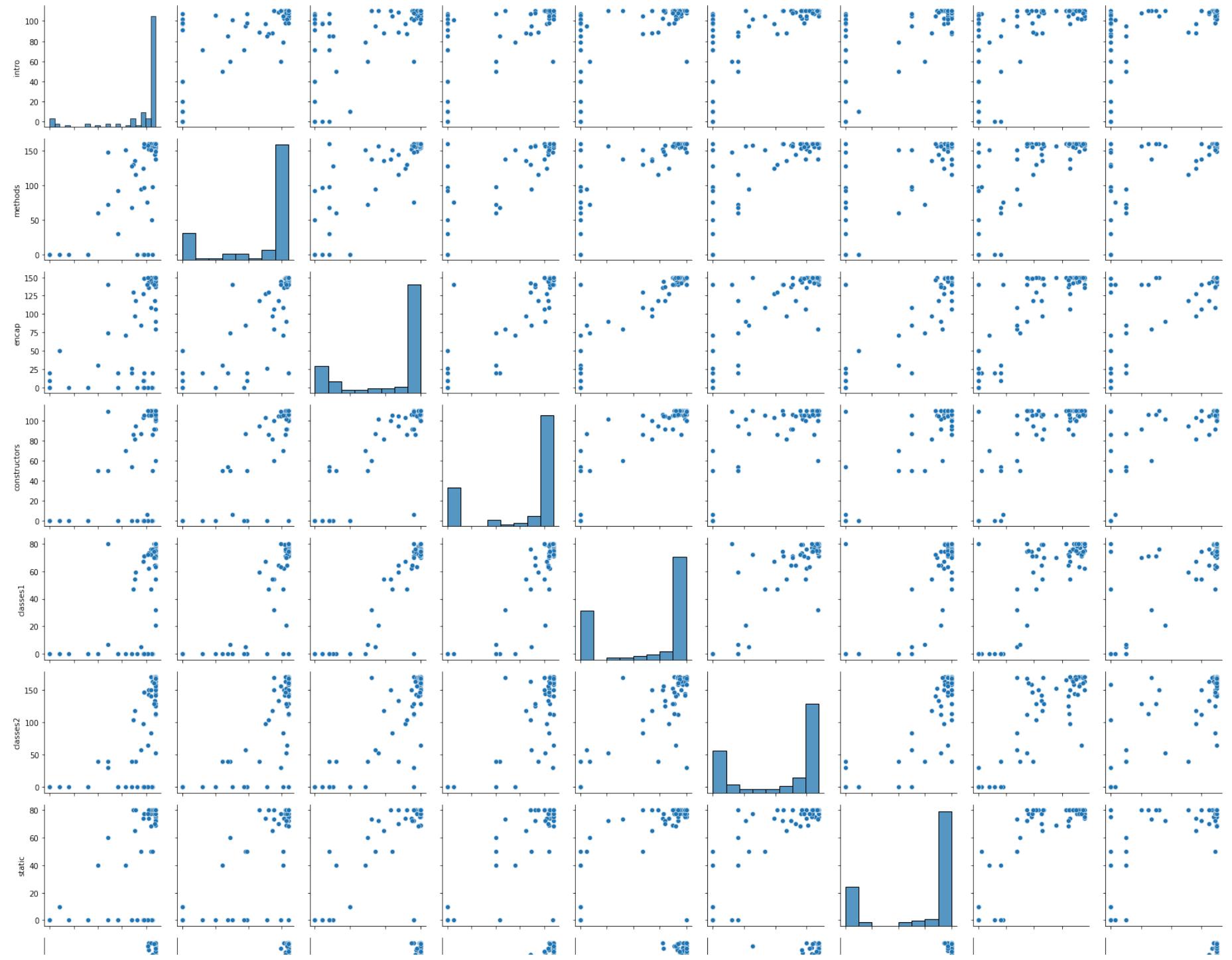
Out[20]: <AxesSubplot:>

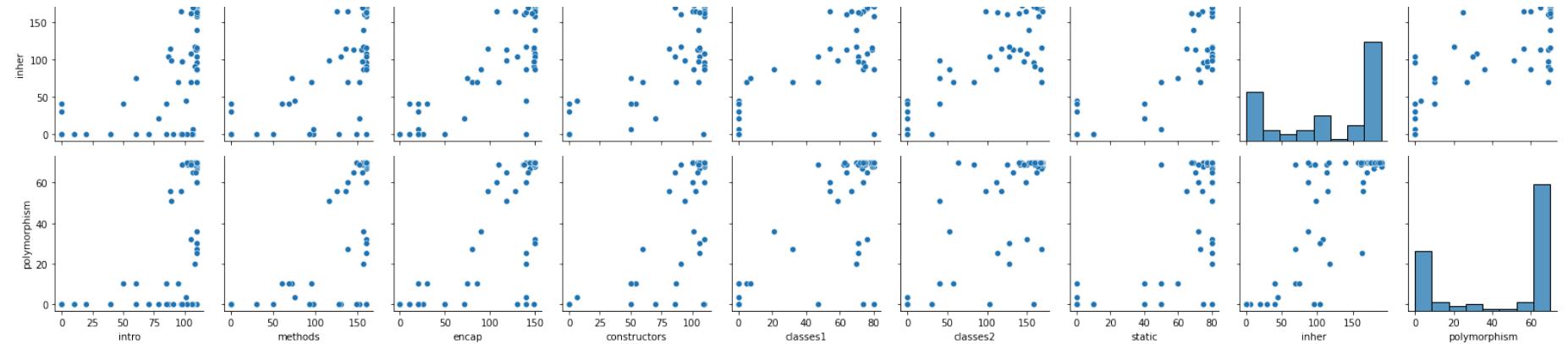


```
In [21]: 1 g_mc = sns.PairGrid(dataset_reduced_mc_lessons)
2 g_mc.map_diag(sns.histplot)
3 g_mc.map_offdiag(sns.scatterplot)
```

```
Out[21]: <seaborn.axisgrid.PairGrid at 0x16628eac1d0>
```







## Term periods

```
In [22]: 1 dataset_T1 = dataset_reduced[['intro','intro-VPL','methods','methods-VPL']]
2 dataset_T2 = dataset_reduced[['encap','encap-VPL','constructors','constructors-VPL','classes1','classes2','classes2-VPL','static']]
3 dataset_T3 = dataset_reduced[['inher','inher-VPL', 'polymorphism', 'polymorphism-VPL']]
```

```
In [23]: 1 dataset_T1 = dataset_T1.astype(float)
2 dataset_T2 = dataset_T2.astype(float)
3 dataset_T3 = dataset_T3.astype(float)
```

## Alltogether

```
In [24]: 1 from pycaret.clustering import *
2
3 # input_dataset1 = setup(dataset_reduced, normalize = True, pca=True, pca_components=2,
4 #                         numeric_features = ['intro-VPL','methods','methods-VPL','encap','encap-VPL','constructors','constructors-VPL',
5 #                         session_id = 123)
6
7 input_dataset1 = setup(dataset_reduced, normalize = True,
8                         numeric_features = ['intro-VPL','methods','methods-VPL','encap','encap-VPL','constructors','constructors-VPL','cl
9                         session_id = 123)
```

	Description	Value
0	session_id	123
1	Original Data	(102, 17)
2	Missing Values	False
3	Numeric Features	17
4	Categorical Features	0
5	Ordinal Features	False
6	High Cardinality Features	False
7	High Cardinality Method	None
8	Transformed Data	(102, 17)
9	CPU Jobs	-1
10	Use GPU	False
11	Log Experiment	False
12	Experiment Name	cluster-default-name
13	USI	680d
14	Imputation Type	simple
15	Iterative Imputation Iteration	None
16	Numeric Imputer	mean
17	Iterative Imputation Numeric Model	None
18	Categorical Imputer	mode
19	Iterative Imputation Categorical Model	None
20	Unknown Categoricals Handling	least_frequent
21	Normalize	True
22	Normalize Method	zscore
23	Transformation	False
24	Transformation Method	None
25	PCA	False
26	PCA Method	None
27	PCA Components	None

	Description	Value
28	Ignore Low Variance	False
29	Combine Rare Levels	False
30	Rare Level Threshold	None
31	Numeric Binning	False
32	Remove Outliers	False
33	Outliers Threshold	None
34	Remove Multicollinearity	False
35	Multicollinearity Threshold	None
36	Clustering	False
37	Clustering Iteration	None
38	Polynomial Features	False
39	Polynomial Degree	None
40	Trigonometry Features	False
41	Polynomial Threshold	None
42	Group Features	False
43	Feature Selection	False
44	Feature Selection Method	classic
45	Features Selection Threshold	None
46	Feature Interaction	False
47	Feature Ratio	False
48	Interaction Threshold	None

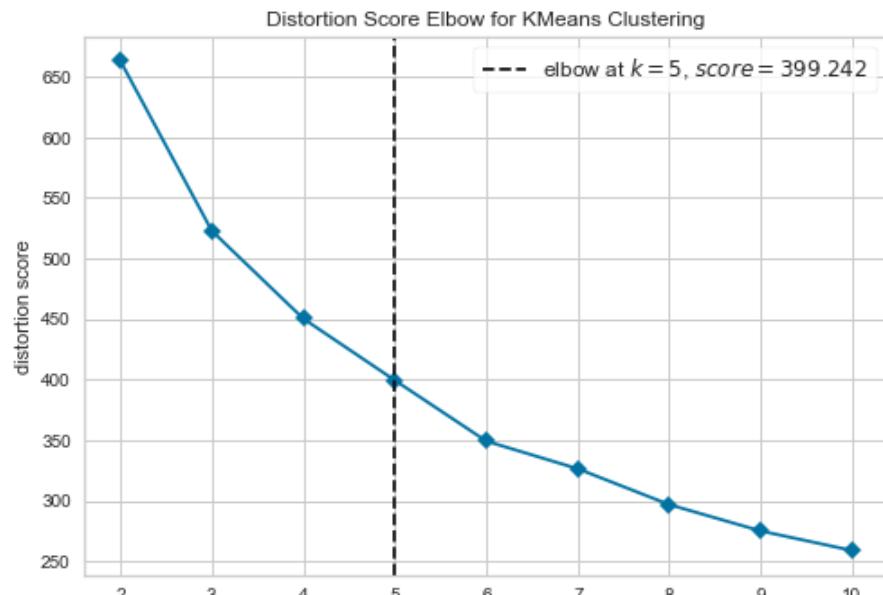
In [25]: 1 kmeans = create\_model('kmeans')

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.3747	93.1811	1.3109	0	0	0

```
In [26]: 1 kmeans
```

```
Out[26]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
                 n_clusters=4, n_init=10, n_jobs=-1, precompute_distances='deprecated',
                 random_state=123, tol=0.0001, verbose=0)
```

```
In [27]: 1 plot_model(kmeans, plot = 'elbow')
```



```
In [28]: 1 evaluate_model(kmeans)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance  
Distribution

## 2D Cluster PCA Plot



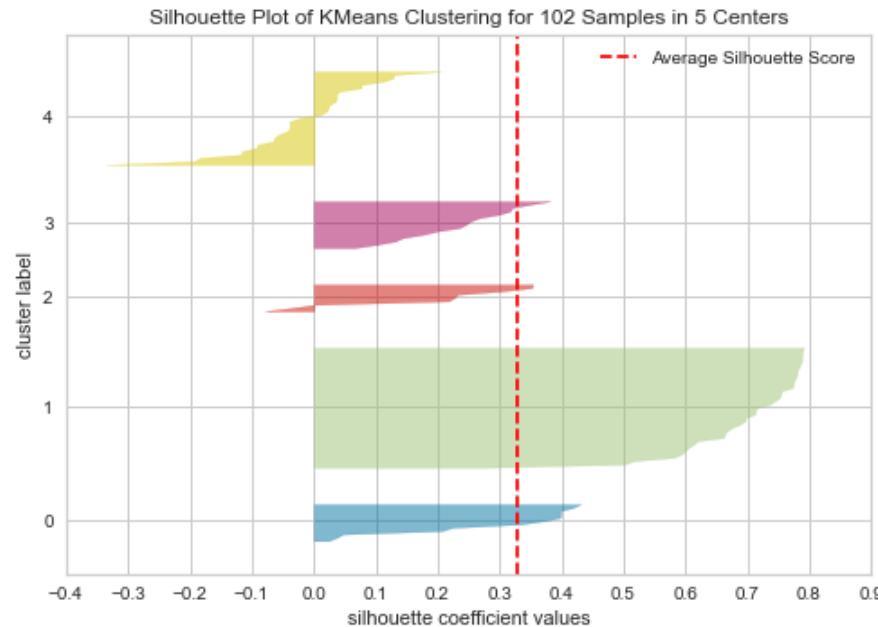
```
In [29]: 1 kmeans5 = create_model('kmeans', num_clusters = 5)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.3284	81.0733	1.483	0	0	0

```
In [30]: 1 kmeans5
```

```
Out[30]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
                 n_clusters=5, n_init=10, n_jobs=-1, precompute_distances='deprecated',
                 random_state=123, tol=0.0001, verbose=0)
```

```
In [31]: 1 plot_model(kmeans5, plot = 'silhouette')
```



In [32]: 1 | evaluate\_model(kmeans5)

Plot Type: Cluster PCA Plot (2d)

Cluster TSnE (3d)

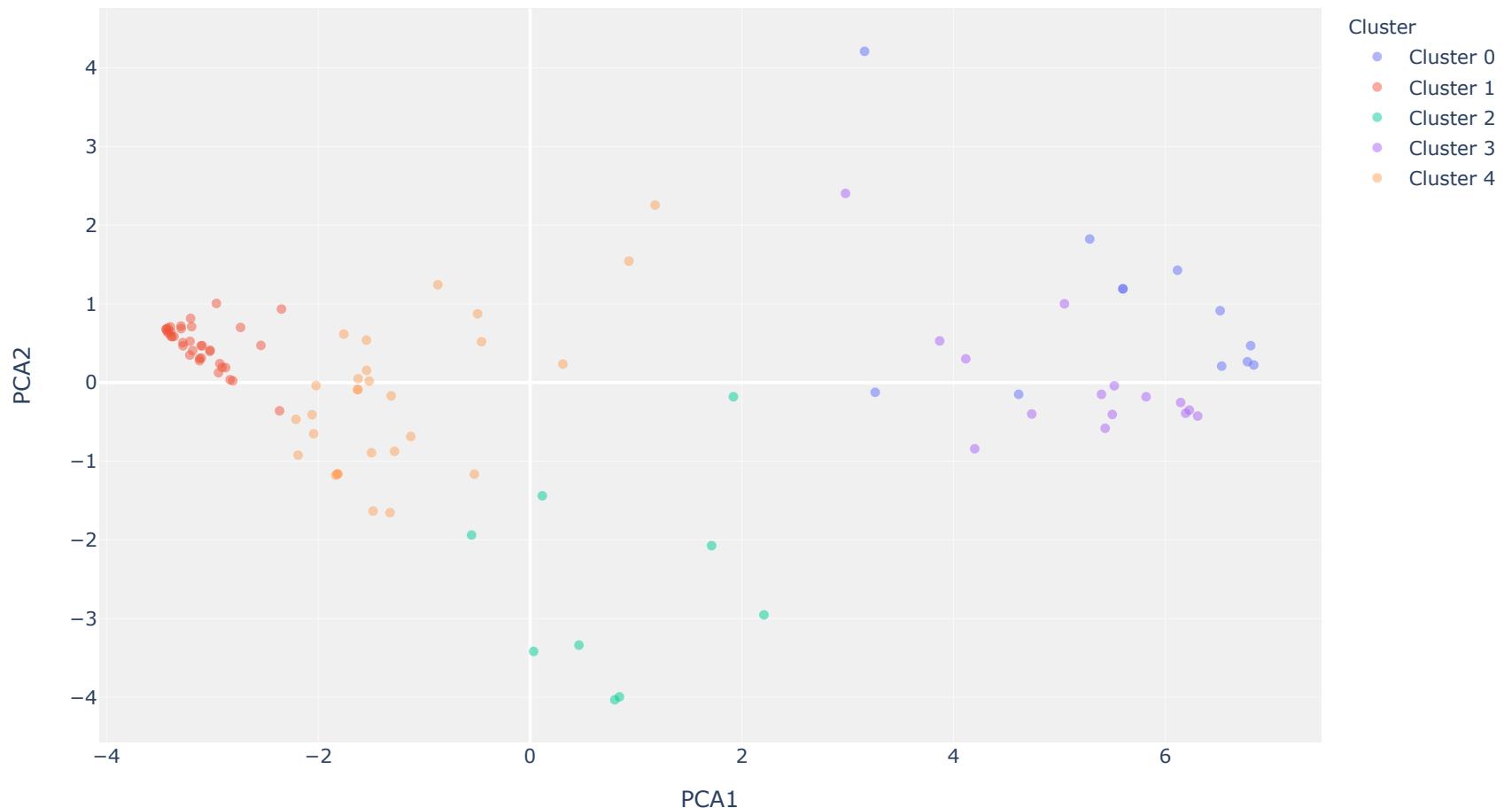
Elbow

Silhouette

Distance

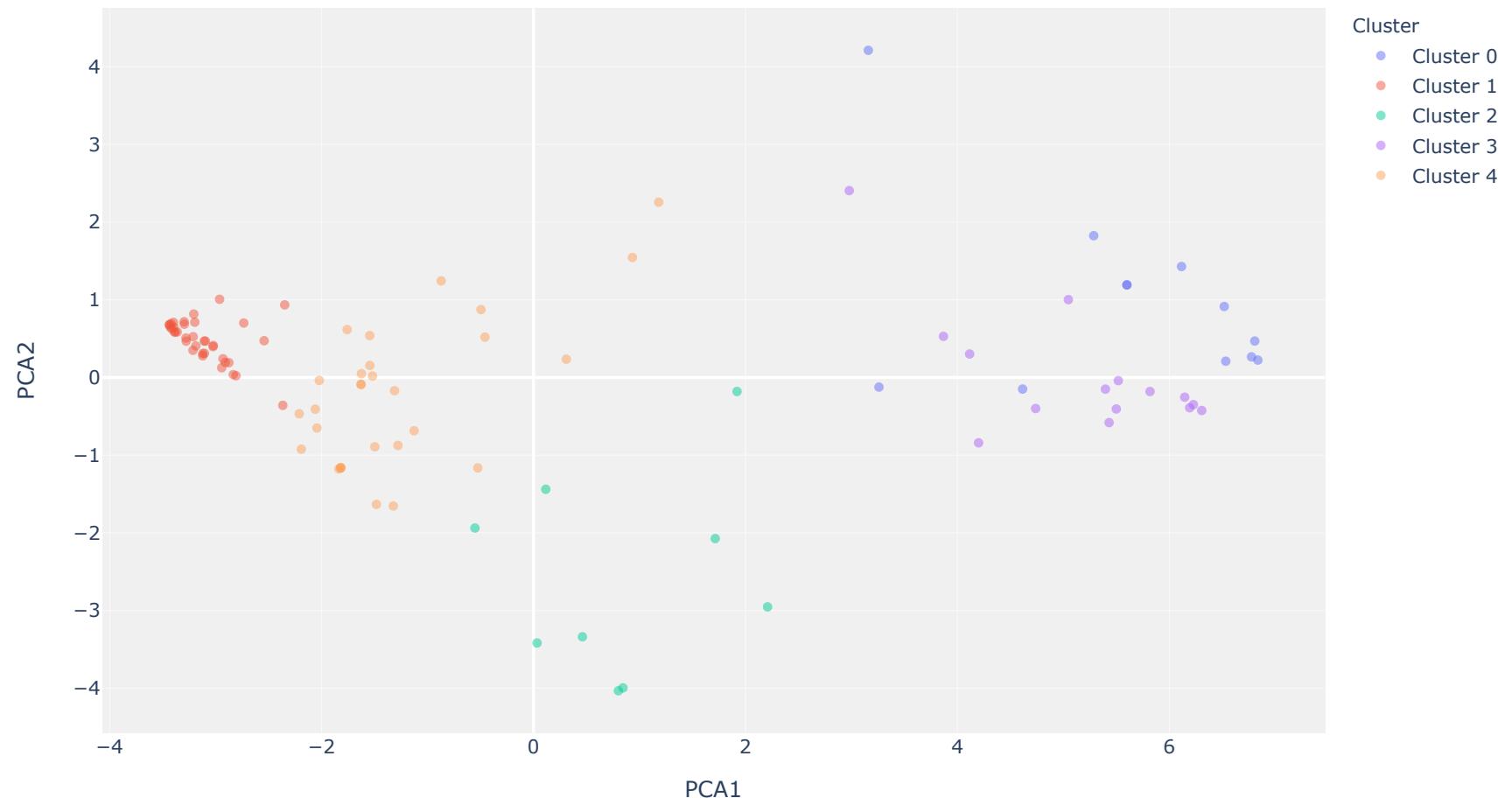
Distribution

## 2D Cluster PCA Plot



```
In [33]: 1 plot_model(kmeans5)
```

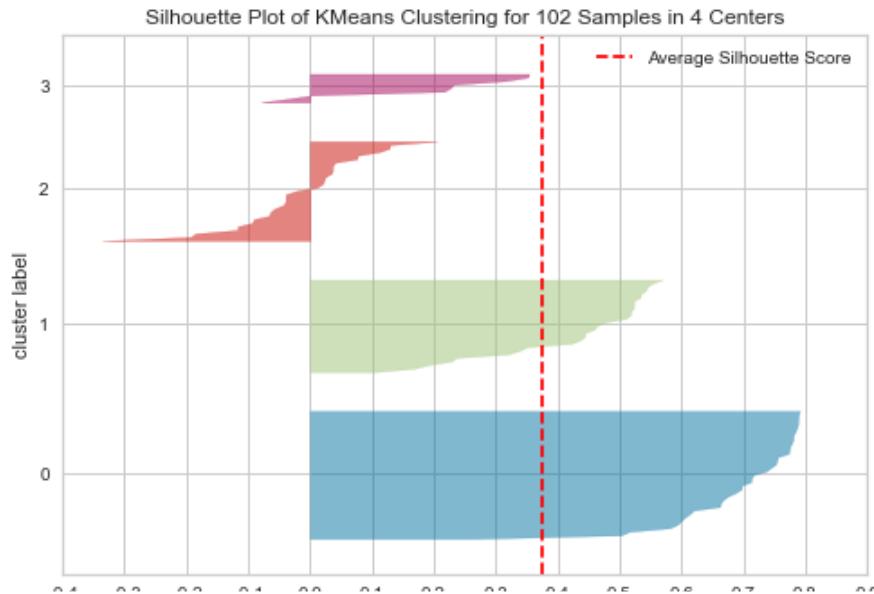
2D Cluster PCA Plot



```
In [34]: 1 kmeans4 = create_model('kmeans', num_clusters = 4)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.3747	93.1811	1.3109	0	0	0

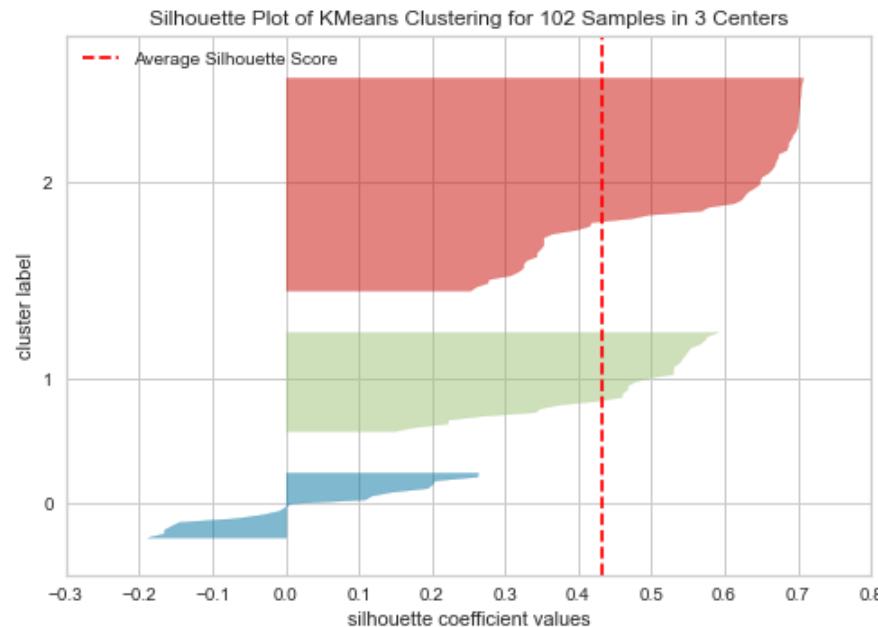
```
In [35]: 1 plot_model(kmeans4, plot = 'silhouette')
```



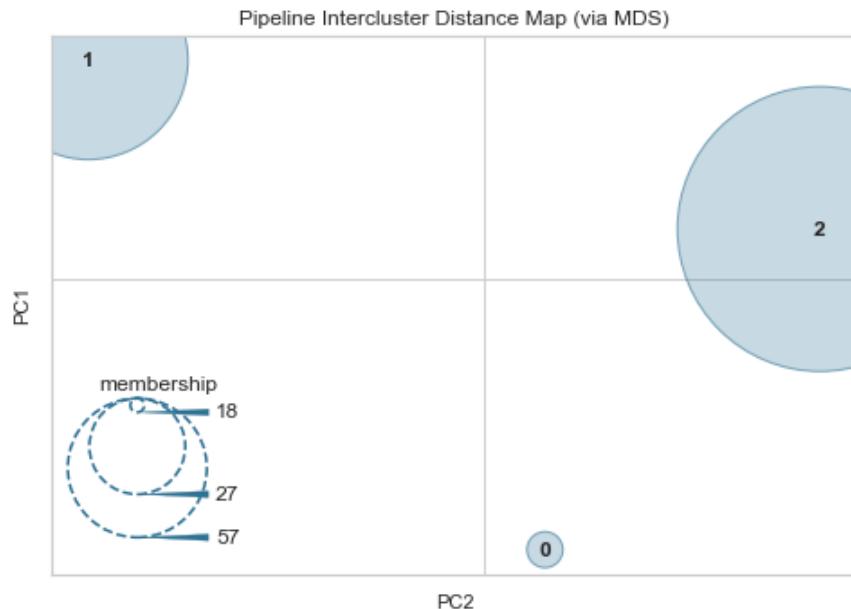
```
In [36]: 1 kmeans3 = create_model('kmeans', num_clusters = 3)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.4313	114.7803	1.2559	0	0	0

```
In [37]: 1 plot_model(kmeans3, plot = 'silhouette')
```



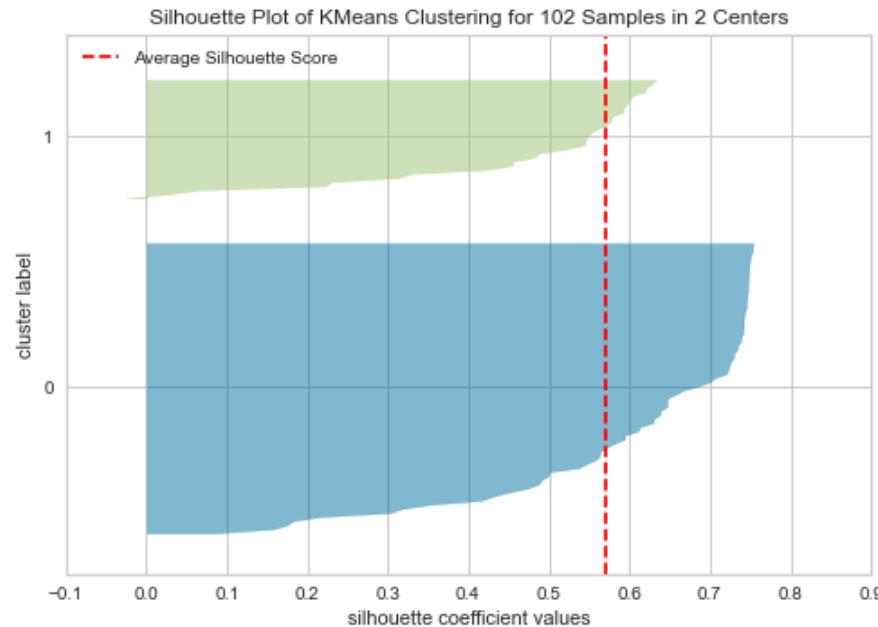
```
In [38]: 1 plot_model(kmeans3, plot = 'distance')
```



```
In [39]: 1 kmeans2 = create_model('kmeans', num_clusters = 2)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.57	161.6648	0.6984	0	0	0

```
In [40]: 1 plot_model(kmeans2, plot = 'silhouette')
```



In [41]: 1 | evaluate\_model(kmeans2)

Plot Type: Cluster PCA Plot (2d)

Cluster TSnE (3d)

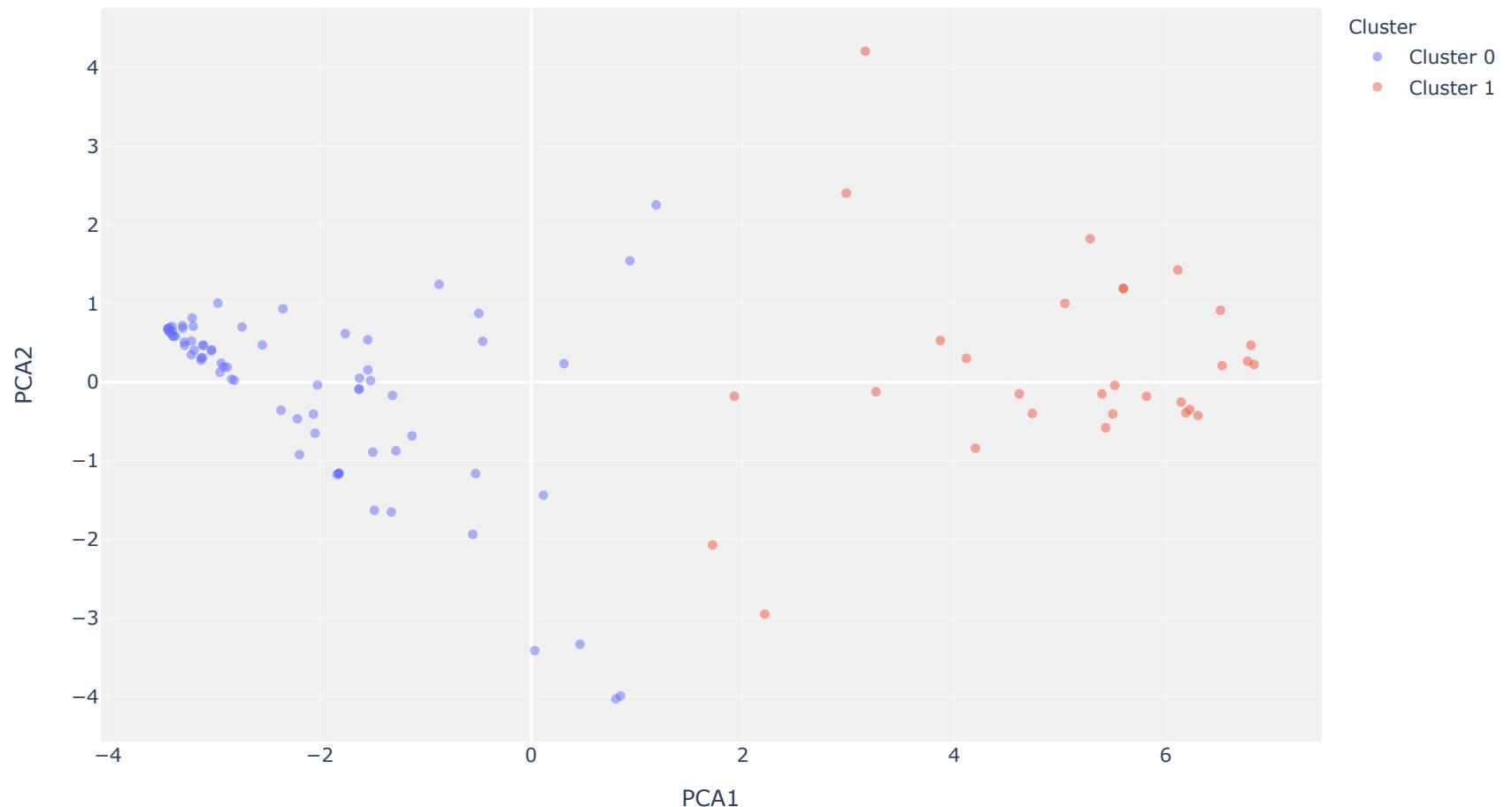
Elbow

Silhouette

Distance

Distribution

## 2D Cluster PCA Plot



In [42]: 1 | plot\_model(kmeans2)

2D Cluster PCA Plot

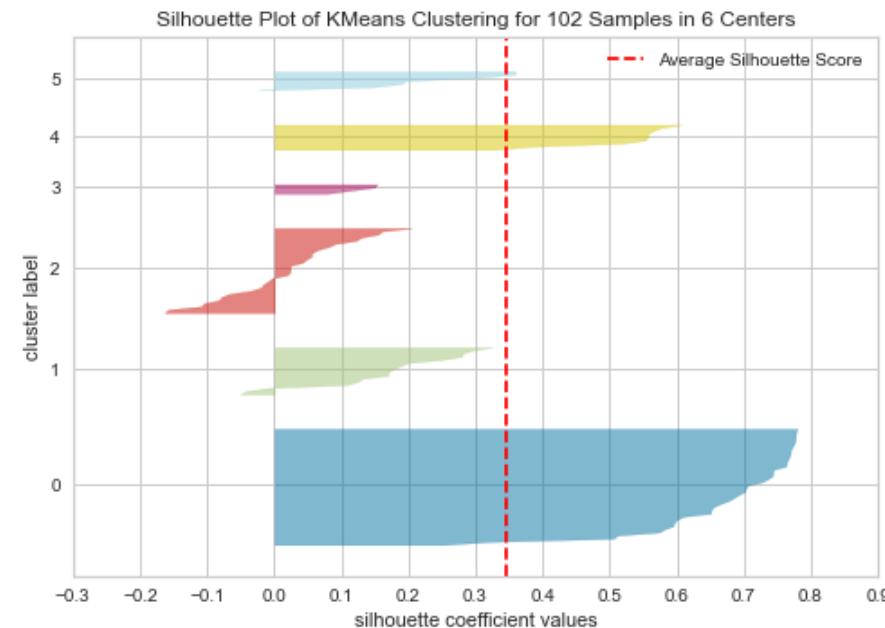


In [ ]: 1

```
In [43]: 1 kmeans6 = create_model('kmeans', num_clusters = 6)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.3465	76.1701	1.2466	0	0	0

```
In [44]: 1 plot_model(kmeans6, plot = 'silhouette')
```



```
In [45]: 1 evaluate_model(kmeans6)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

## 2D Cluster PCA Plot



```
In [46]: 1 kmeans_df5 = assign_model(kmeans5)
```

```
In [47]: 1 kmeans_df5.head()
```

Out[47]:

	intro	intro-VPL	methods	methods-VPL	encap	encap-VPL	constructors	constructors-VPL	classes1	classes2	classes2-VPL	static	static-VPL	inher	inher-VPL	polymorphism	polymorp
0	110.0	60.0	160.0	630.0	148.0	100.0	110.0	165.0	80.0	170.0	95.0	80.0	270.0	190.0	270.0	70.0	
1	110.0	60.0	160.0	630.0	150.0	100.0	110.0	165.0	80.0	170.0	95.0	80.0	270.0	188.0	270.0	70.0	
2	110.0	60.0	160.0	630.0	150.0	100.0	110.0	165.0	80.0	170.0	92.0	80.0	270.0	190.0	270.0	70.0	
3	110.0	60.0	160.0	630.0	150.0	100.0	110.0	165.0	80.0	170.0	95.0	77.0	270.0	190.0	270.0	70.0	
4	110.0	60.0	160.0	630.0	150.0	100.0	110.0	157.0	80.0	170.0	95.0	80.0	270.0	190.0	270.0	70.0	



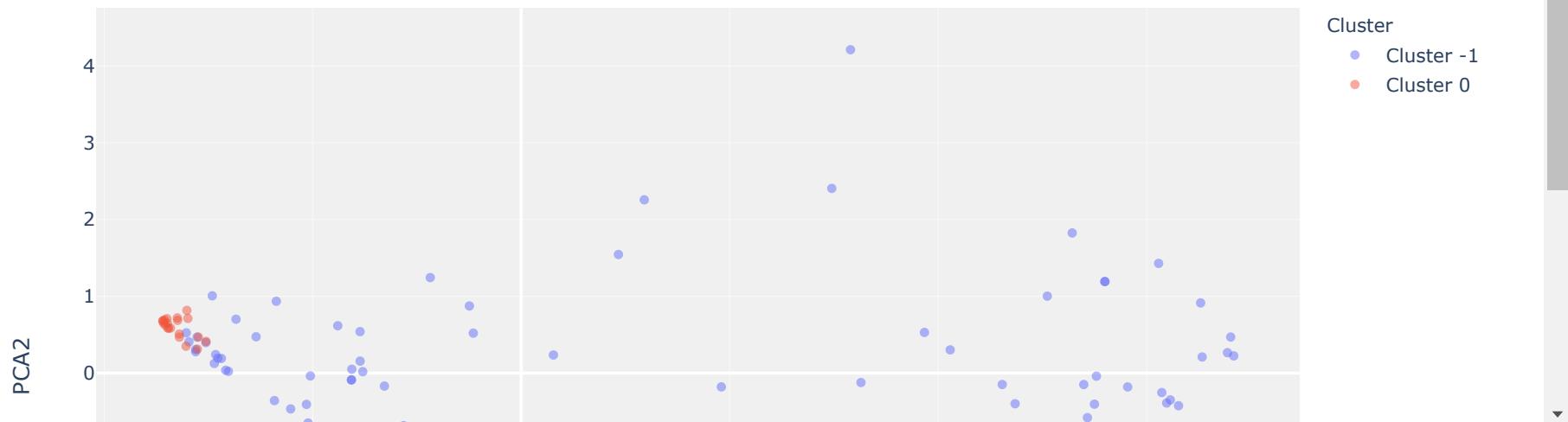
## DBSCAN

```
In [48]: 1 model_dbSCAN = create_model('dbSCAN')
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.0397	19.7449	1.0447	0	0	0

```
In [49]: 1 plot_model(model_dbSCAN)
```

2D Cluster PCA Plot



```
In [50]: 1 model_dbSCAN2 = create_model('dbSCAN', eps = 0.7)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.1734	33.4757	1.0102	0	0	0

```
In [51]: 1 model_dbSCAN3 = create_model('dbSCAN', eps = 0.3)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	-0.0864	10.5508	1.0586	0	0	0

```
In [52]: 1 model_dbSCAN4 = create_model('dbSCAN', eps = 0.9)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.2354	42.3797	0.9726	0	0	0

```
In [53]: 1 model_dbSCAN5 = create_model('dbSCAN', eps = 2.0)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.3653	82.1147	1.2025	0	0	0

```
In [54]: 1 model_dbSCAN6 = create_model('dbSCAN', eps = 3.20)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.4583	94.3754	1.4969	0	0	0

```
In [55]: 1 model_dbSCAN6
```

```
Out[55]: DBSCAN(algorithm='auto', eps=3.2, leaf_size=30, metric='euclidean',
metric_params=None, min_samples=5, n_jobs=-1, p=None)
```

```
In [56]: 1 evaluate_model(model_dbSCAN6)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

## 2D Cluster PCA Plot

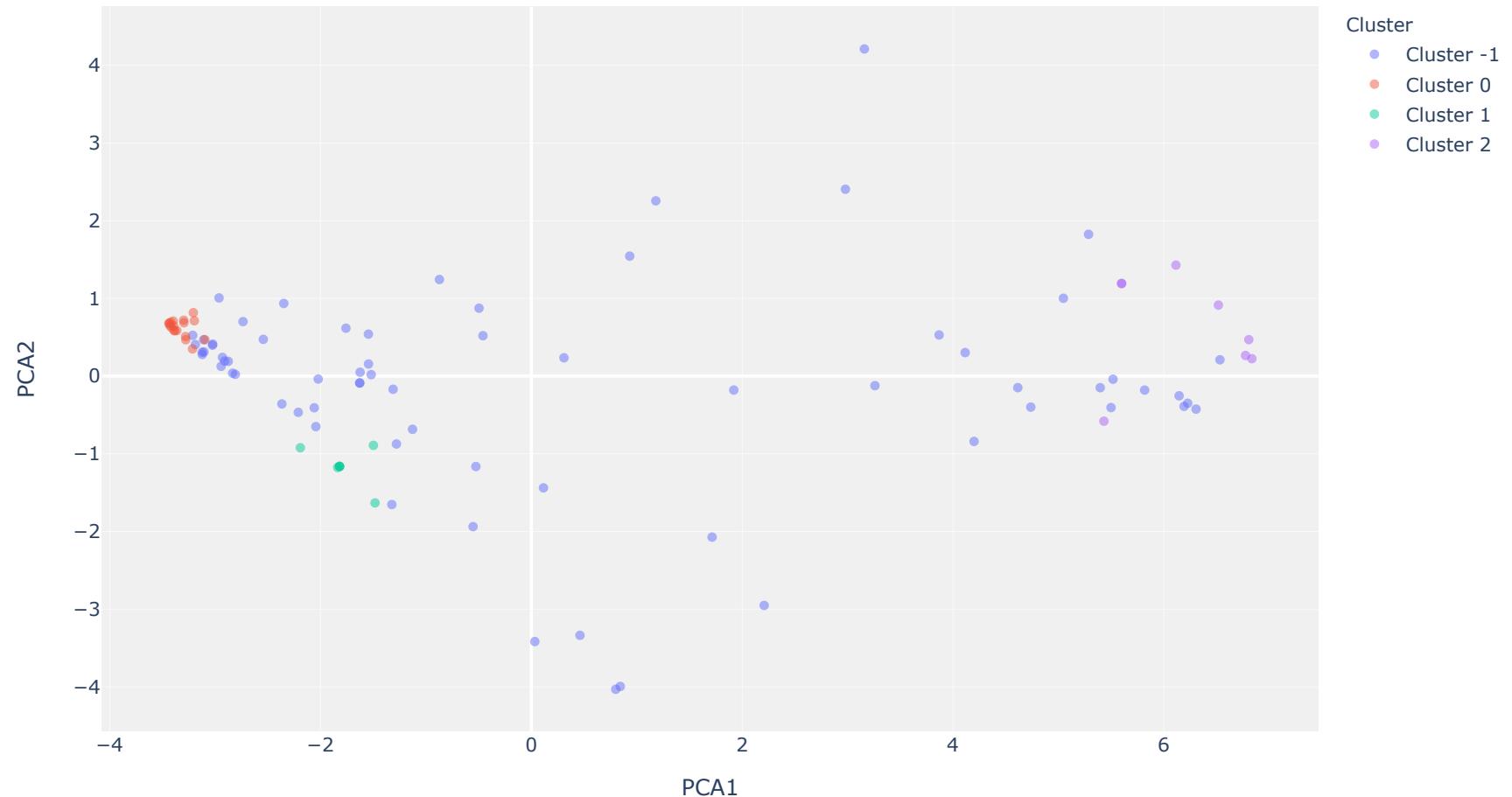


```
In [57]: 1 model_optics = create_model('optics')
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	-0.0615	17.0579	1.3876	0	0	0

```
In [58]: 1 plot_model(model_optics)
```

2D Cluster PCA Plot

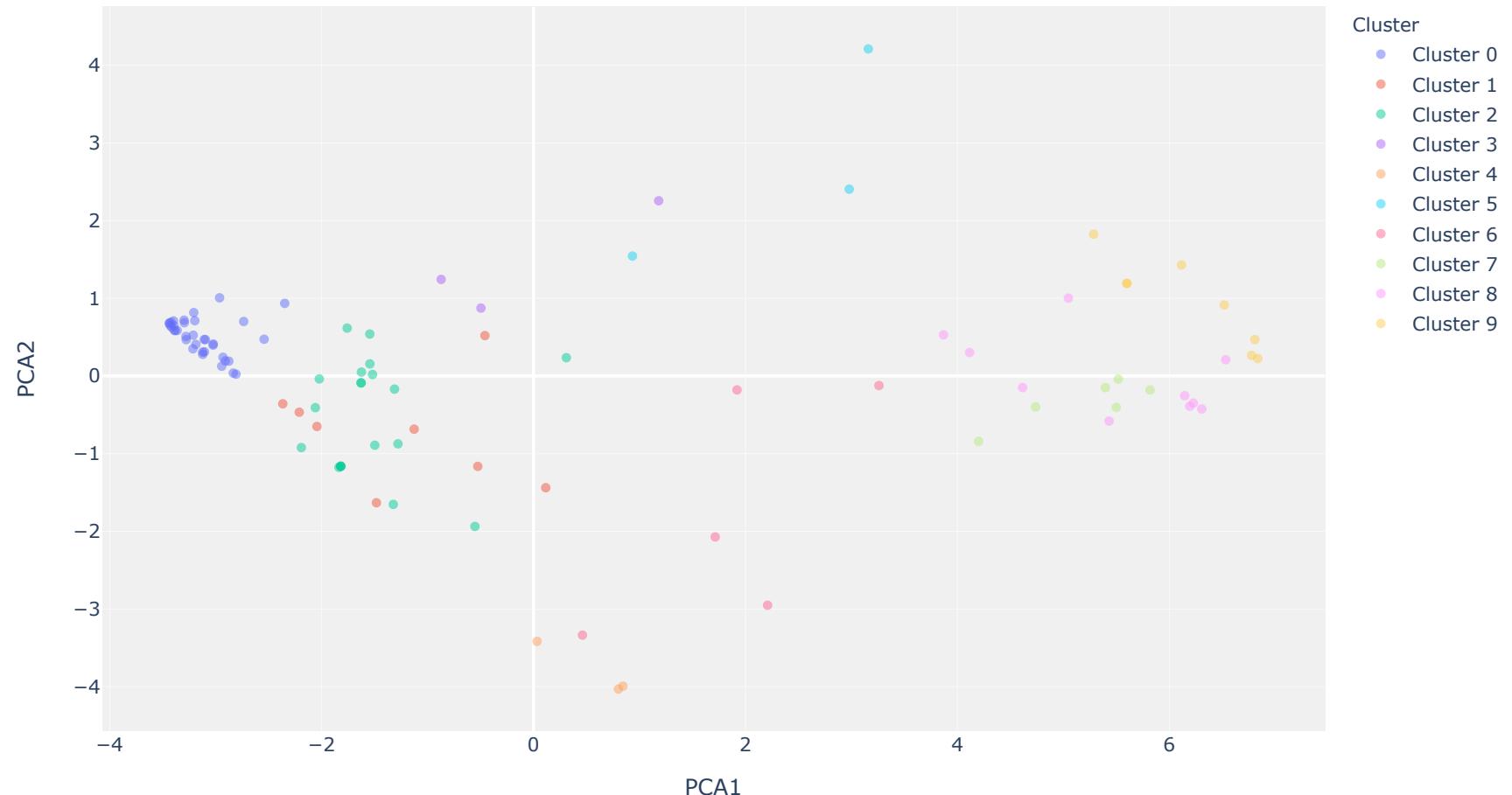


```
In [59]: 1 model_ap = create_model('ap')
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.3662	57.6724	1.2593	0	0	0

```
In [60]: 1 plot_model(model_ap)
```

2D Cluster PCA Plot



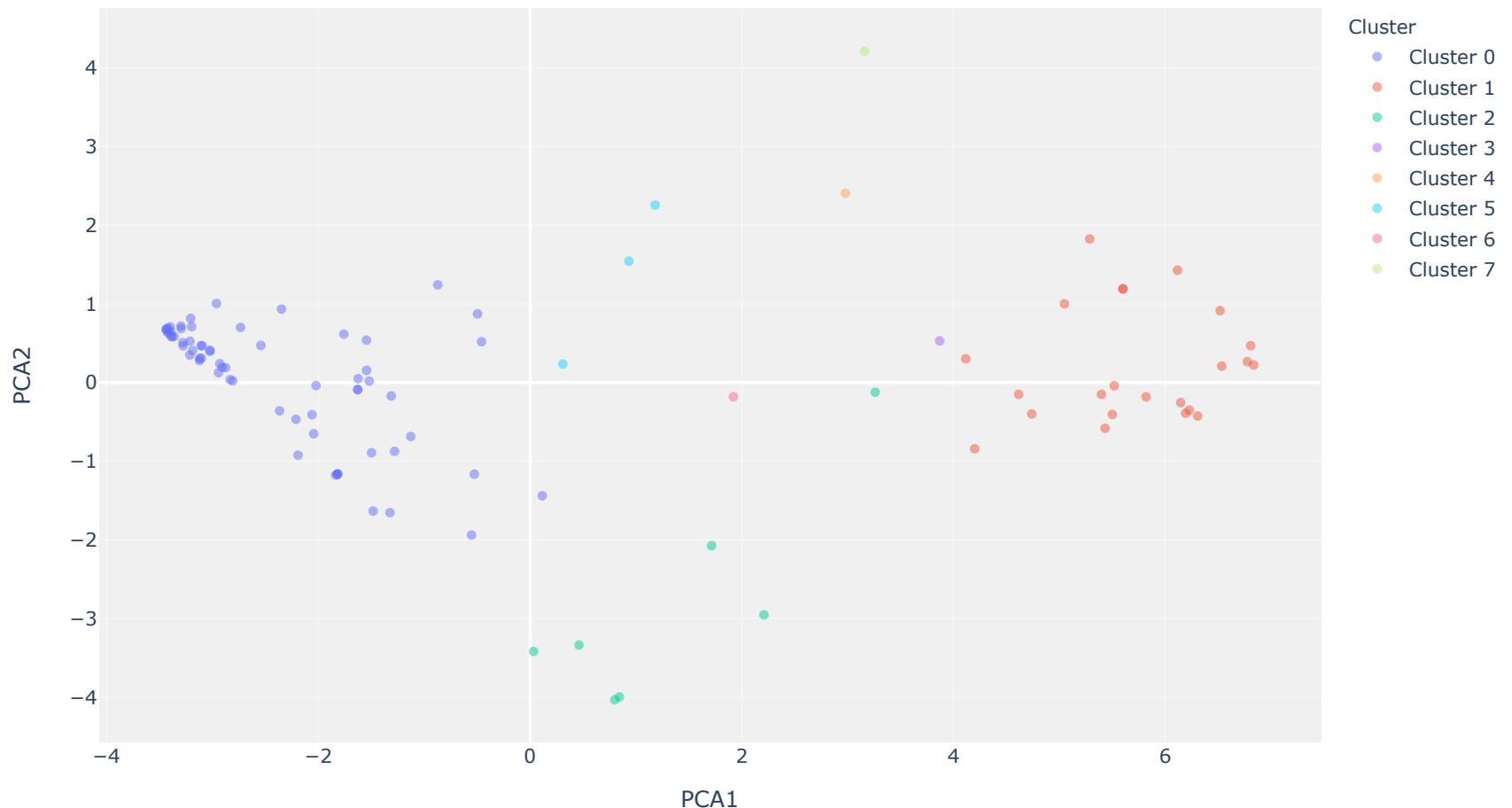
```
In [61]: 1 model_meanshift = create_model('meanshift')
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.3891	41.8296	0.7929	0	0	0

```
In [62]: 1 evaluate_model(model_meanshift)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

## 2D Cluster PCA Plot

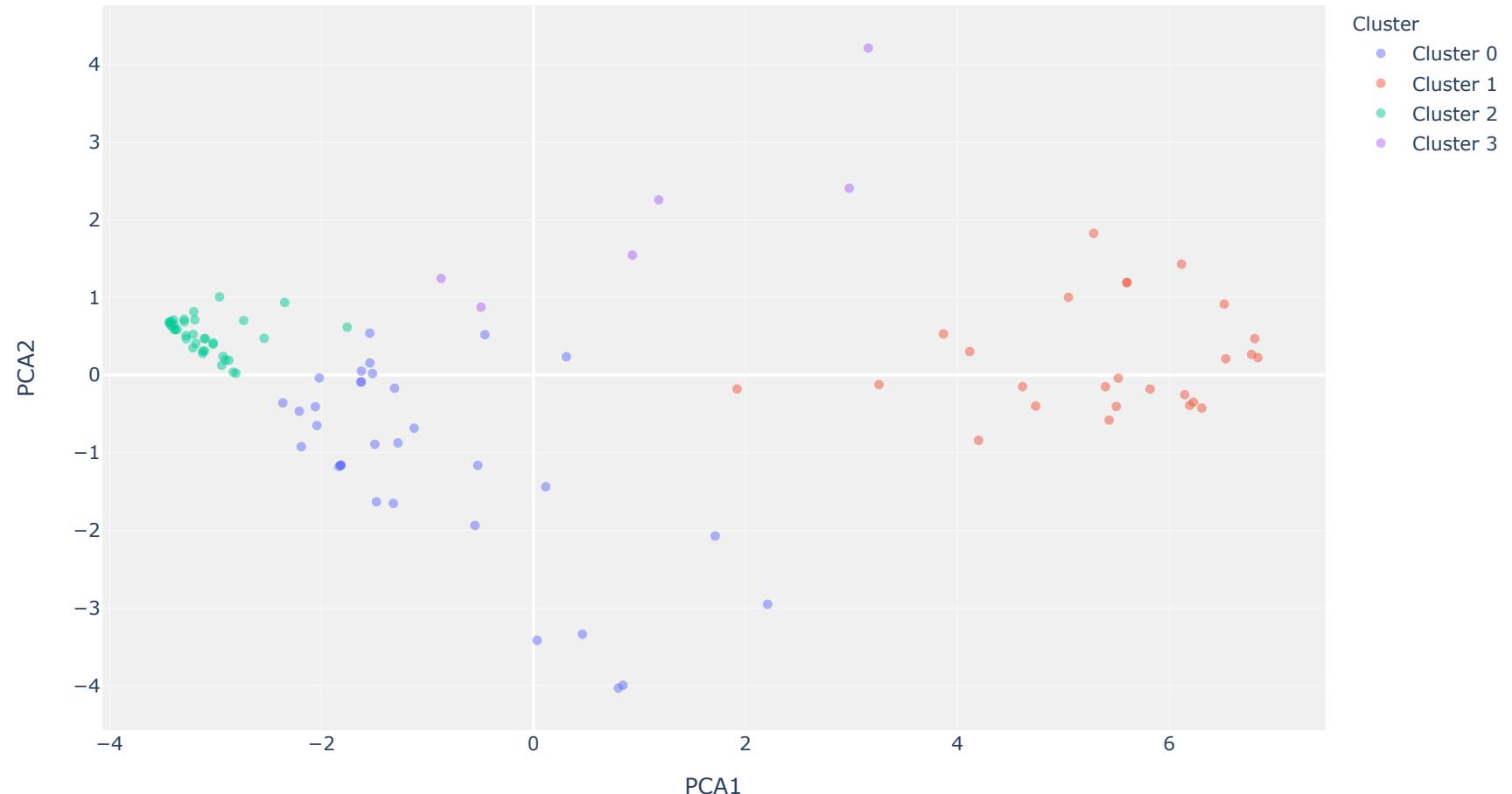


```
In [63]: 1 model_hclust = create_model('hclust')
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.3688	85.9097	1.2241	0	0	0

```
In [64]: 1 plot_model(model_hclust)
```

2D Cluster PCA Plot



```
In [65]: 1 model_birch = create_model('birch')
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.4499	83.2682	1.0465	0	0	0

```
In [66]: 1 plot_model(model_birch)
```

2D Cluster PCA Plot



```
In [67]: 1 get_metrics()
```

Out[67]:

ID	Name	Display Name	Score Function	Scorer	Target	Args	Greater is Better	Needs Ground Truth	Custom
<b>ID</b>									
<b>silhouette</b>	Silhouette	Silhouette	<function silhouette_score at 0x000001662E0C4730>	make_scoring(silhouette_score)	pred	{}	True	False	False
<b>chs</b>	Calinski-Harabasz	Calinski-Harabasz	<function calinski_harabasz_score at 0x0000016...	make_scoring(calinski_harabasz_score)	pred	{}	True	False	False
<b>db</b>	Davies-Bouldin	Davies-Bouldin	<function davies_bouldin_score at 0x000001662E...	make_scoring(davies_bouldin_score)	pred	{}	True	False	False
<b>hs</b>	Homogeneity Score	Homogeneity	<function homogeneity_score at 0x000001662E0A3...	make_scoring(homogeneity_score)	pred	{}	True	True	False
<b>ari</b>	Rand Index	Rand Index	<function adjusted_rand_score at 0x000001662E0...	make_scoring(adjusted_rand_score)	pred	{}	True	True	False
<b>cs</b>	Completeness Score	Completeness	<function completeness_score at 0x000001662E0A...	make_scoring(completeness_score)	pred	{}	True	True	False

```
In [ ]:
```

1

## Agglomerative Clustering

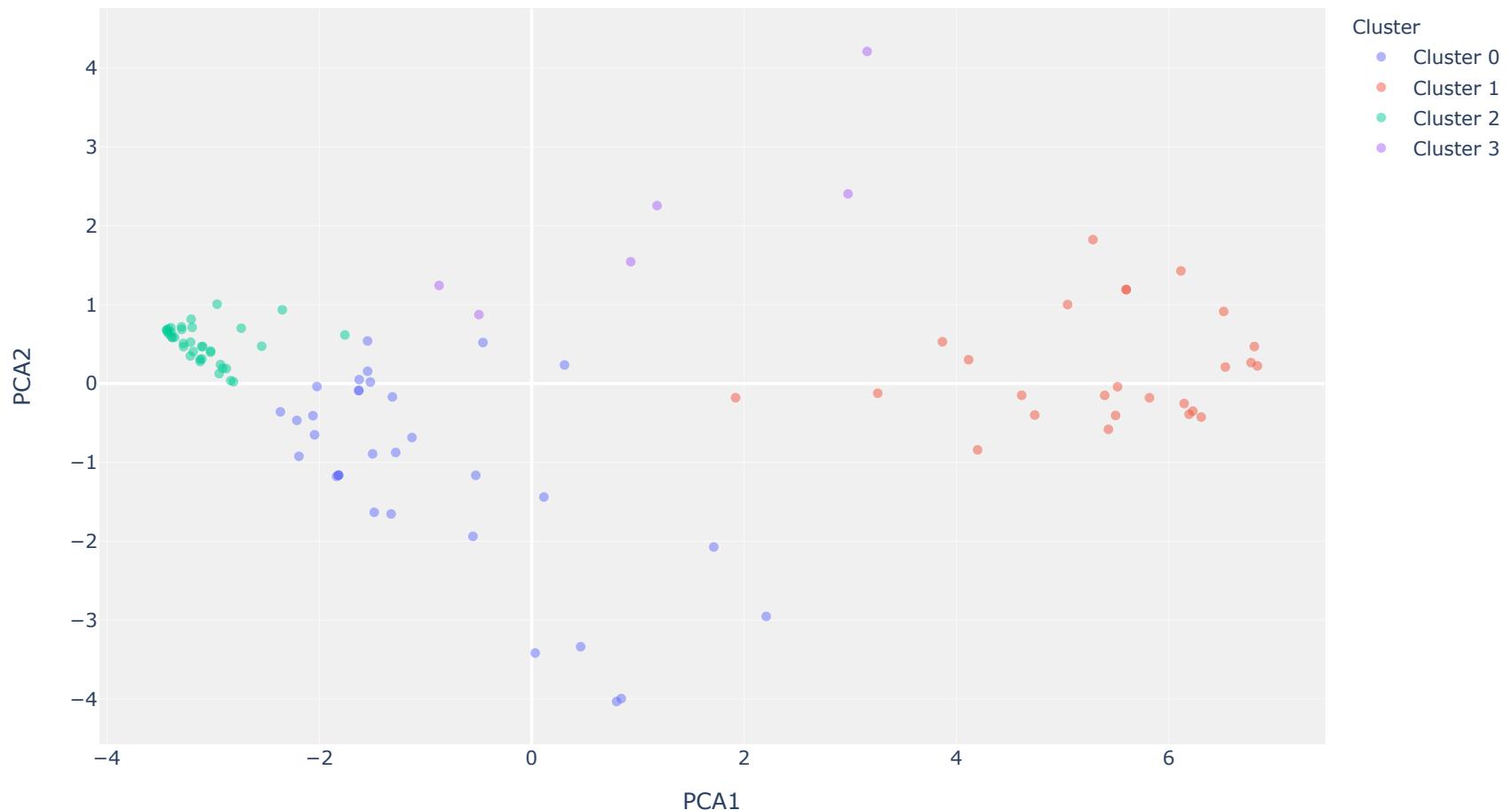
```
In [68]: 1 model_agglomerative4 = create_model('hclust', n_clusters = 4)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.3688	85.9097	1.2241	0	0	0

```
In [69]: 1 evaluate_model(model_agglomerative4)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

## 2D Cluster PCA Plot



## Term Periods

```
In [70]: 1 from pycaret.clustering import *
2
3 dataset_T1 = dataset_T1.astype(float)
4
5 input_dataset2 = setup(dataset_T1, normalize = True,
6                         session_id = 123)
```

	Description	Value
0	session_id	123
1	Original Data	(102, 4)
2	Missing Values	False
3	Numeric Features	4
4	Categorical Features	0
5	Ordinal Features	False
6	High Cardinality Features	False
7	High Cardinality Method	None
8	Transformed Data	(102, 4)
9	CPU Jobs	-1
10	Use GPU	False
11	Log Experiment	False
12	Experiment Name	cluster-default-name
13	USI	bc74
14	Imputation Type	simple
15	Iterative Imputation Iteration	None
16	Numeric Imputer	mean
17	Iterative Imputation Numeric Model	None
18	Categorical Imputer	mode
19	Iterative Imputation Categorical Model	None
20	Unknown Categoricals Handling	least_frequent
21	Normalize	True
22	Normalize Method	zscore
23	Transformation	False
24	Transformation Method	None
25	PCA	False
26	PCA Method	None
27	PCA Components	None

	Description	Value
28	Ignore Low Variance	False
29	Combine Rare Levels	False
30	Rare Level Threshold	None
31	Numeric Binning	False
32	Remove Outliers	False
33	Outliers Threshold	None
34	Remove Multicollinearity	False
35	Multicollinearity Threshold	None
36	Clustering	False
37	Clustering Iteration	None
38	Polynomial Features	False
39	Polynomial Degree	None
40	Trigonometry Features	False
41	Polynomial Threshold	None
42	Group Features	False
43	Feature Selection	False
44	Feature Selection Method	classic
45	Features Selection Threshold	None
46	Feature Interaction	False
47	Feature Ratio	False
48	Interaction Threshold	None

In [ ]: 1

In [71]: 1 kmeansT1 = create\_model('kmeans')

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.4917	118.7943	0.8273	0	0	0

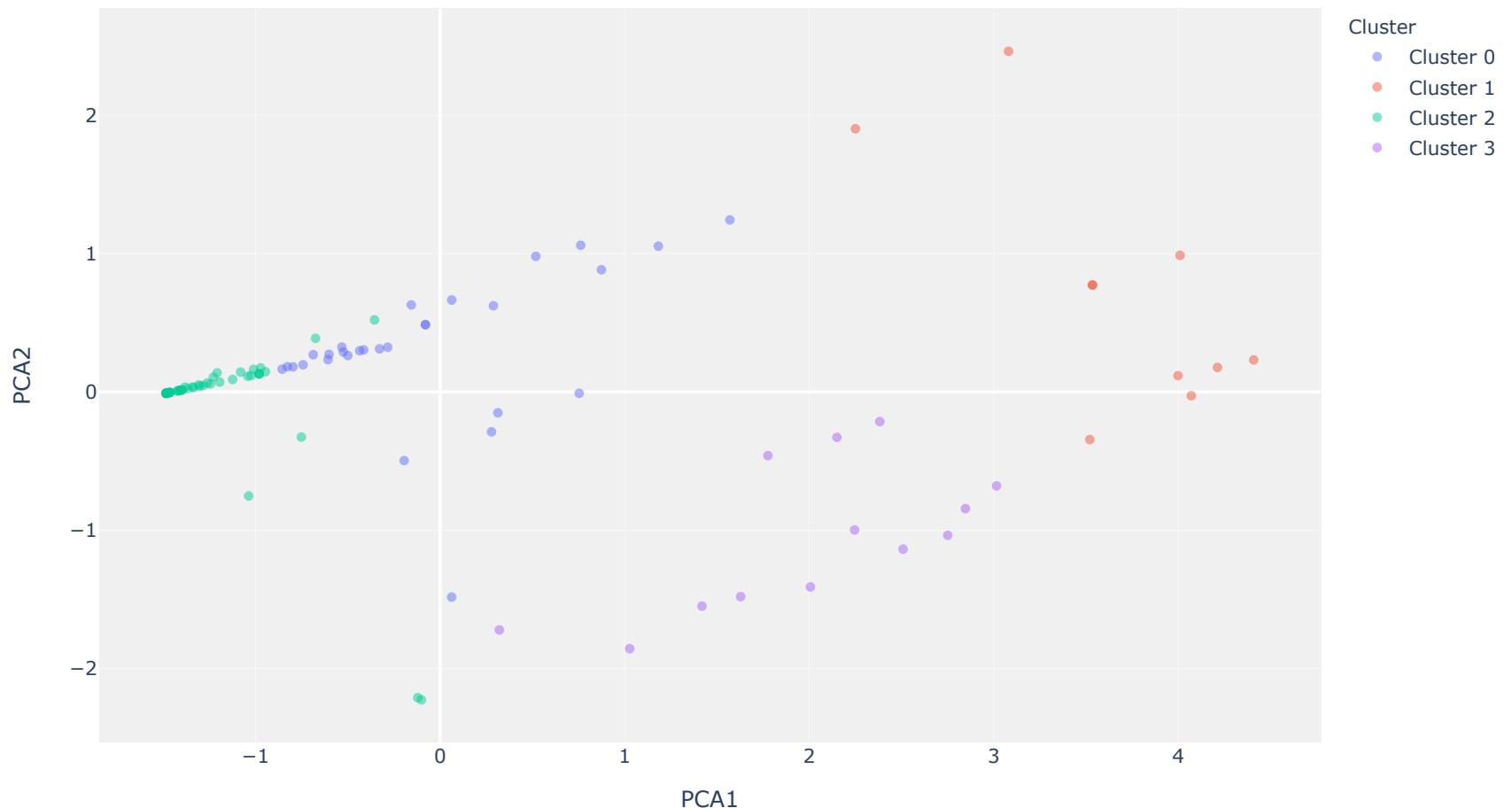
In [72]: 1 kmeansT1

Out[72]: KMeans(algorithm='auto', copy\_x=True, init='k-means++', max\_iter=300,  
n\_clusters=4, n\_init=10, n\_jobs=-1, precompute\_distances='deprecated',  
random\_state=123, tol=0.0001, verbose=0)

In [73]: 1 | evaluate\_model(kmeansT1)

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

## 2D Cluster PCA Plot



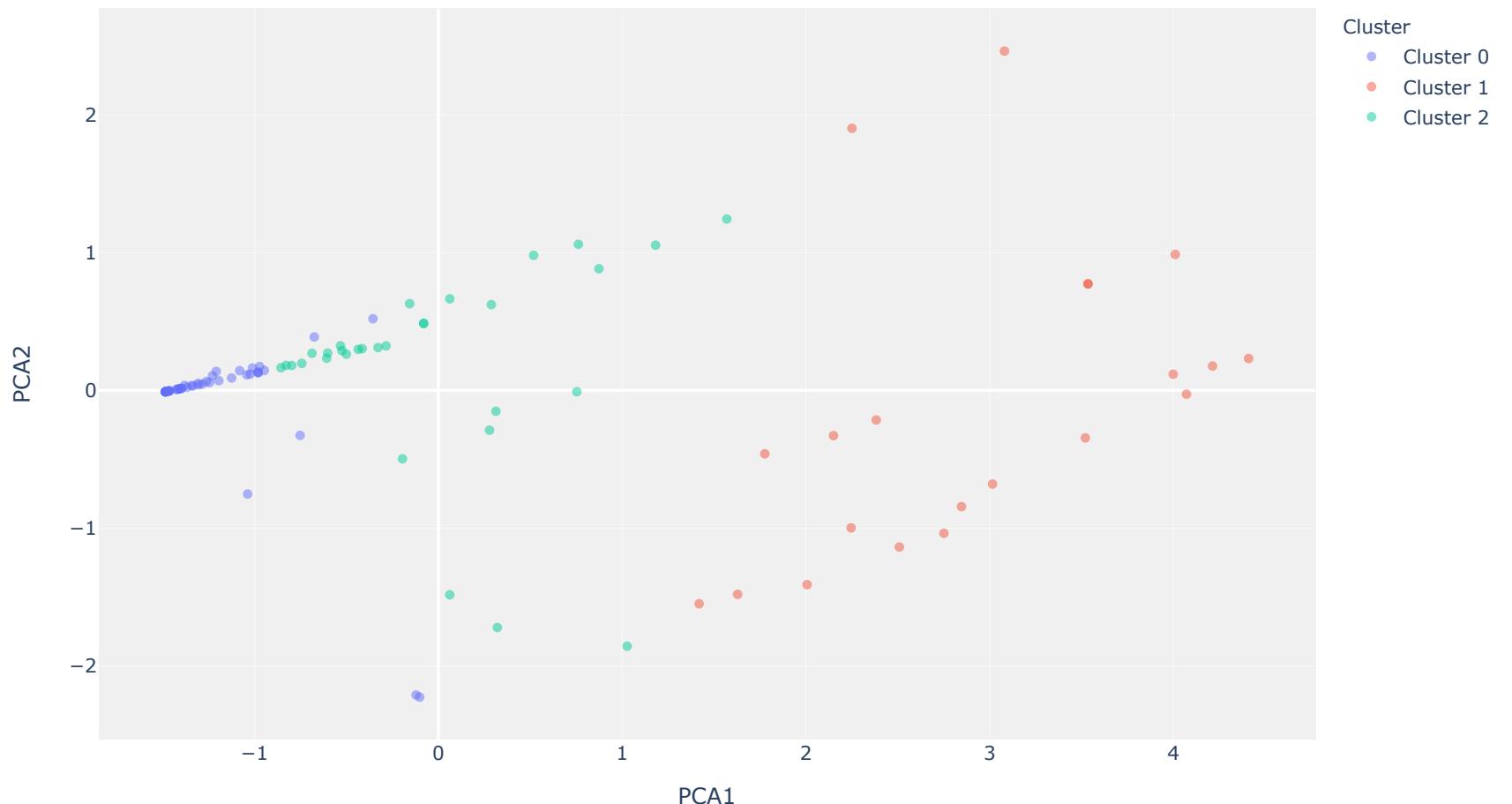
```
In [74]: 1 kmeansT1_3 = create_model('kmeans', num_clusters=3)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.4654	109.8245	0.8889	0	0	0

In [75]: 1 | evaluate\_model(kmeansT1\_3)

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

## 2D Cluster PCA Plot



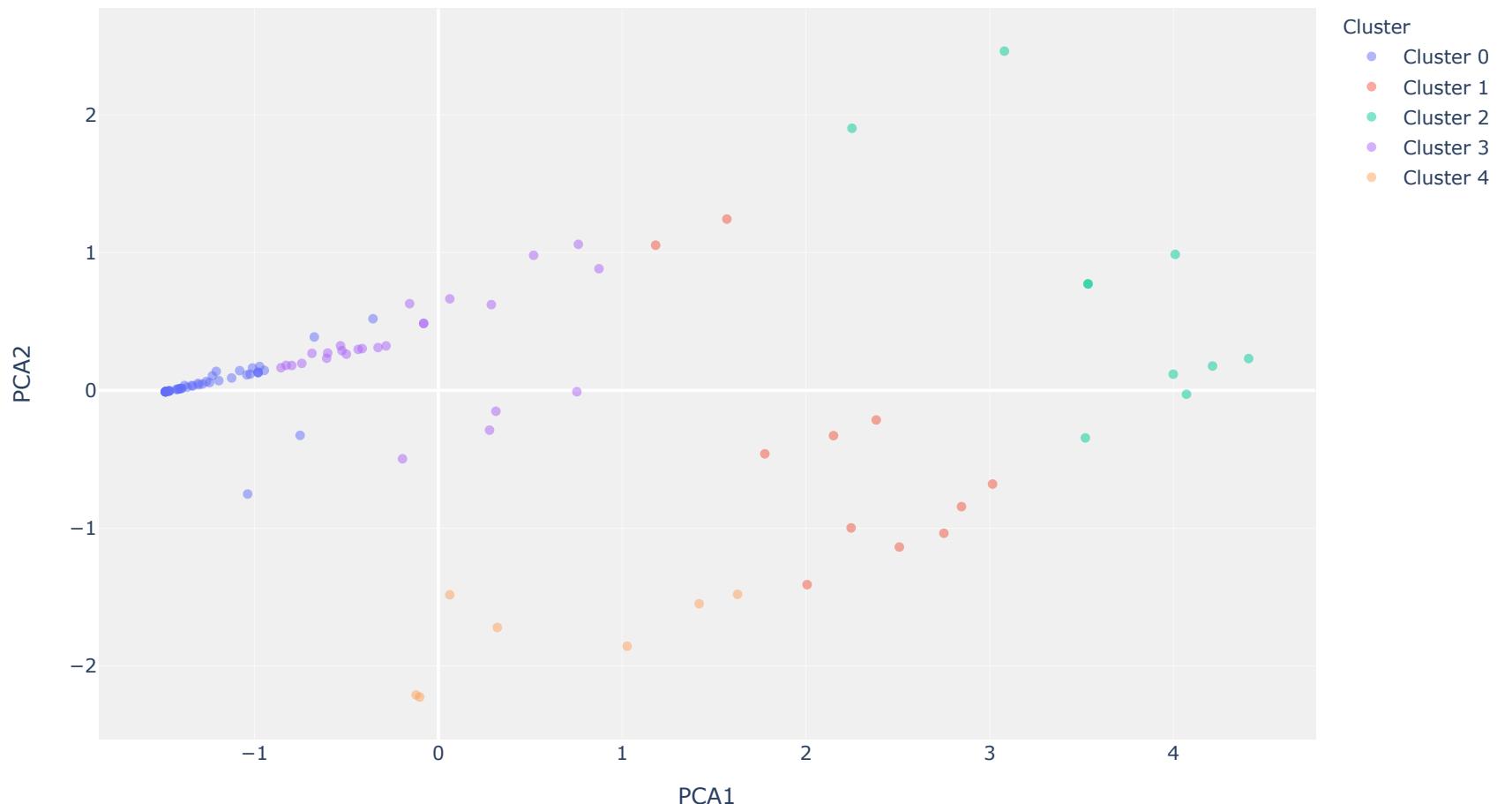
```
In [76]: 1 kmeansT1_5 = create_model('kmeans', num_clusters=5)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.5219	128.2155	0.8026	0	0	0

```
In [77]: 1 | evaluate_model(kmeansT1_5)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

## 2D Cluster PCA Plot



```
In [78]: 1 kmeansT1_2 = create_model('kmeans', num_clusters=2)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.5947	132.5591	0.7669	0	0	0

```
In [79]: 1 evaluate_model(kmeansT1_2)
```

Plot Type: Cluster PCA Plot (2d)

Cluster TSnE (3d)

## Elbow

## Silhouette

## Distance

## Distribution

## 2D Cluster PCA Plot

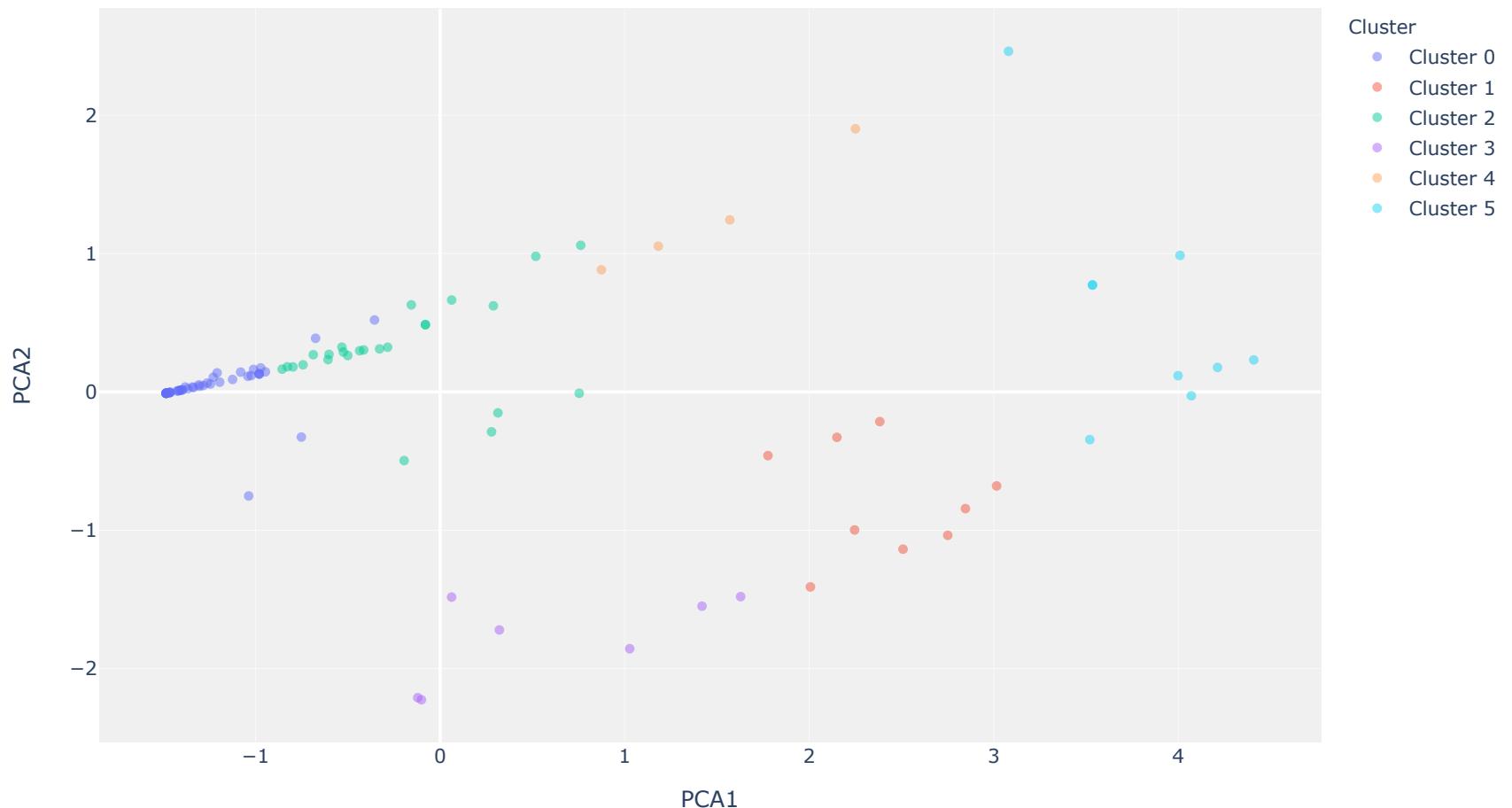
```
In [80]: 1 kmeansT1_6 = create_model('kmeans', num_clusters=6)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.5316	133.9519	0.7479	0	0	0

```
In [81]: 1 evaluate_model(kmeansT1_6)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

2D Cluster PCA Plot



```
In [82]: 1 kmeans_T1_4_assign = assign_model(kmeansT1)
```

```
In [83]: 1 kmeans_T1_6_assign = assign_model(kmeansT1_6)
```

```
In [84]: 1 kmeans_T1_2_assign = assign_model(kmeansT1_2)
```

```
In [85]: 1 kmeans_T1_6_assign.head()
```

Out[85]:

	intro	intro-VPL	methods	methods-VPL	Cluster
0	110.0	60.0	160.0	630.0	Cluster 0
1	110.0	60.0	160.0	630.0	Cluster 0
2	110.0	60.0	160.0	630.0	Cluster 0
3	110.0	60.0	160.0	630.0	Cluster 0
4	110.0	60.0	160.0	630.0	Cluster 0

```
In [86]: 1 kmeans_T1_2_assign.head()
```

Out[86]:

	intro	intro-VPL	methods	methods-VPL	Cluster
0	110.0	60.0	160.0	630.0	Cluster 0
1	110.0	60.0	160.0	630.0	Cluster 0
2	110.0	60.0	160.0	630.0	Cluster 0
3	110.0	60.0	160.0	630.0	Cluster 0
4	110.0	60.0	160.0	630.0	Cluster 0

```
In [87]: 1 clustersT = pd.DataFrame()  
2 clustersTT = pd.DataFrame()  
3 clustersTTT = pd.DataFrame()
```

```
In [88]: 1 clustersT['T1'] = kmeans_T1_6_assign['Cluster'] = kmeans_T1_6_assign['Cluster'].str.replace('\D+', '')
```

```
In [89]: 1 clustersTT = kmeans_T1_2_assign['Cluster']
```

```
In [90]: 1 clustersTTT['T1'] = kmeans_T1_2_assign['Cluster'] = kmeans_T1_2_assign['Cluster'].str.replace('\D+', '')
```

```
In [91]: 1 clustersTTT['T1'] = kmeans_T1_4_assign['Cluster'] = kmeans_T1_4_assign['Cluster'].str.replace('\D+', '')
```

```
In [92]: 1 clustersTT['T1']
```

```
Out[92]: 0      0  
1      0  
2      0  
3      0  
4      0  
..  
97     1  
98     1  
99     1  
100    1  
101    1  
Name: Cluster, Length: 102, dtype: object
```

## T2

```
In [93]: 1 from pycaret.clustering import *
2
3 input_dataset3 = setup(dataset_T2, normalize = True,
4                         session_id = 123)
```

	Description	Value
0	session_id	123
1	Original Data	(102, 9)
2	Missing Values	False
3	Numeric Features	9
4	Categorical Features	0
5	Ordinal Features	False
6	High Cardinality Features	False
7	High Cardinality Method	None
8	Transformed Data	(102, 9)
9	CPU Jobs	-1
10	Use GPU	False
11	Log Experiment	False
12	Experiment Name	cluster-default-name
13	USI	df80
14	Imputation Type	simple
15	Iterative Imputation Iteration	None
16	Numeric Imputer	mean
17	Iterative Imputation Numeric Model	None
18	Categorical Imputer	mode
19	Iterative Imputation Categorical Model	None
20	Unknown Categoricals Handling	least_frequent
21	Normalize	True
22	Normalize Method	zscore
23	Transformation	False
24	Transformation Method	None
25	PCA	False
26	PCA Method	None
27	PCA Components	None

	Description	Value
28	Ignore Low Variance	False
29	Combine Rare Levels	False
30	Rare Level Threshold	None
31	Numeric Binning	False
32	Remove Outliers	False
33	Outliers Threshold	None
34	Remove Multicollinearity	False
35	Multicollinearity Threshold	None
36	Clustering	False
37	Clustering Iteration	None
38	Polynomial Features	False
39	Polynomial Degree	None
40	Trigonometry Features	False
41	Polynomial Threshold	None
42	Group Features	False
43	Feature Selection	False
44	Feature Selection Method	classic
45	Features Selection Threshold	None
46	Feature Interaction	False
47	Feature Ratio	False
48	Interaction Threshold	None

In [94]: 1 kmeansT2\_5 = create\_model('kmeans', num\_clusters=5)

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.4677	133.5979	1.0457	0	0	0

```
In [95]: 1 kmeansT2_3 = create_model('kmeans', num_clusters=3)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.5643	167.2016	0.7829	0	0	0

```
In [96]: 1 kmeansT2_4 = create_model('kmeans', num_clusters=4)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.5701	144.1638	0.9049	0	0	0

```
In [97]: 1 kmeansT2_2 = create_model('kmeans', num_clusters=2)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.6297	204.7539	0.5791	0	0	0

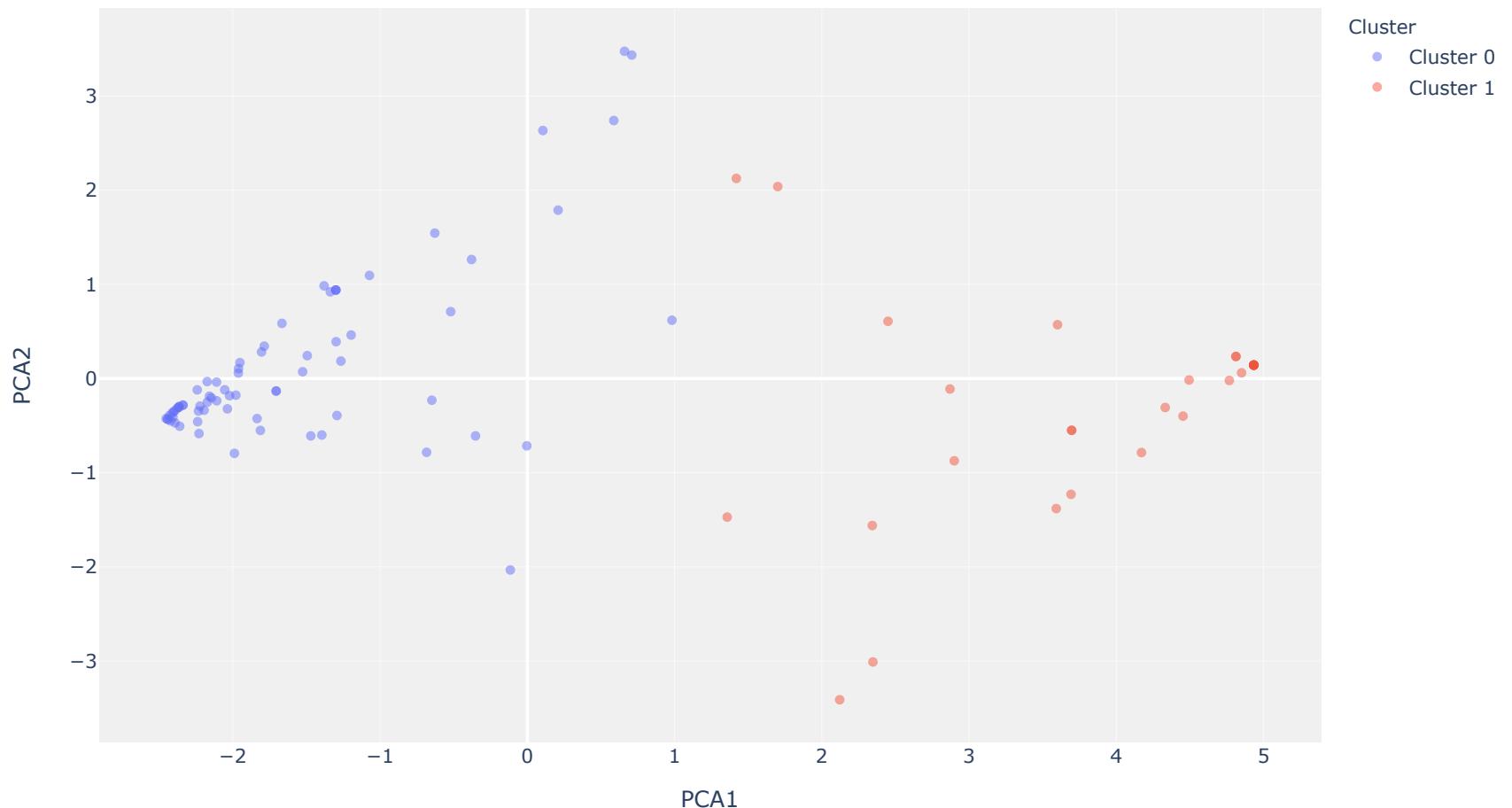
```
In [98]: 1 kmeansT2_6 = create_model('kmeans', num_clusters=6)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.4565	123.8955	1.0332	0	0	0

In [99]: 1 | evaluate\_model(kmeansT2\_2)

Plot Type: Cluster PCA Plot (2d) Cluster tSNE (3d) Elbow Silhouette Distance Distribution

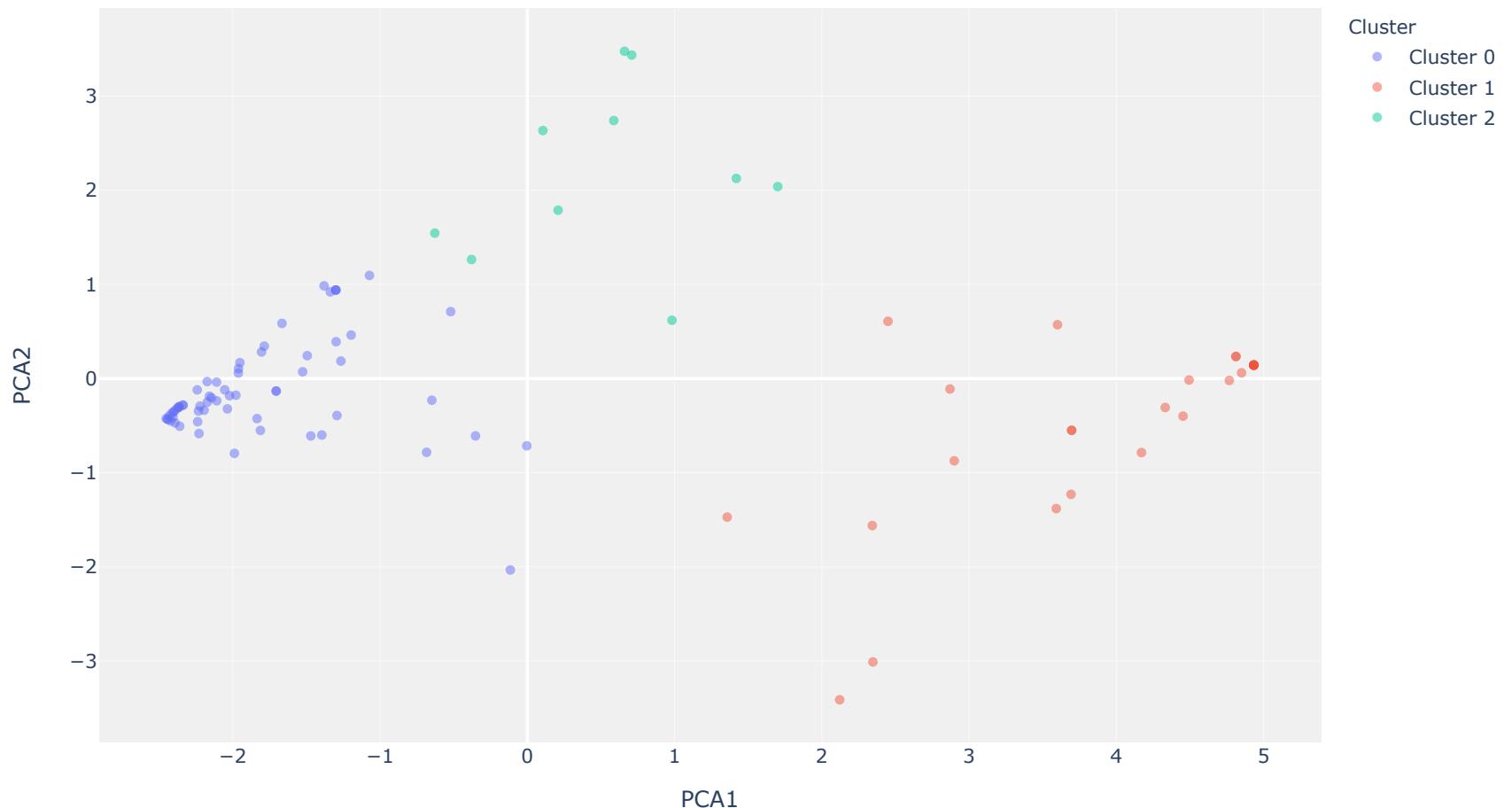
## 2D Cluster PCA Plot



```
In [100]: 1 evaluate_model(kmeansT2_3)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

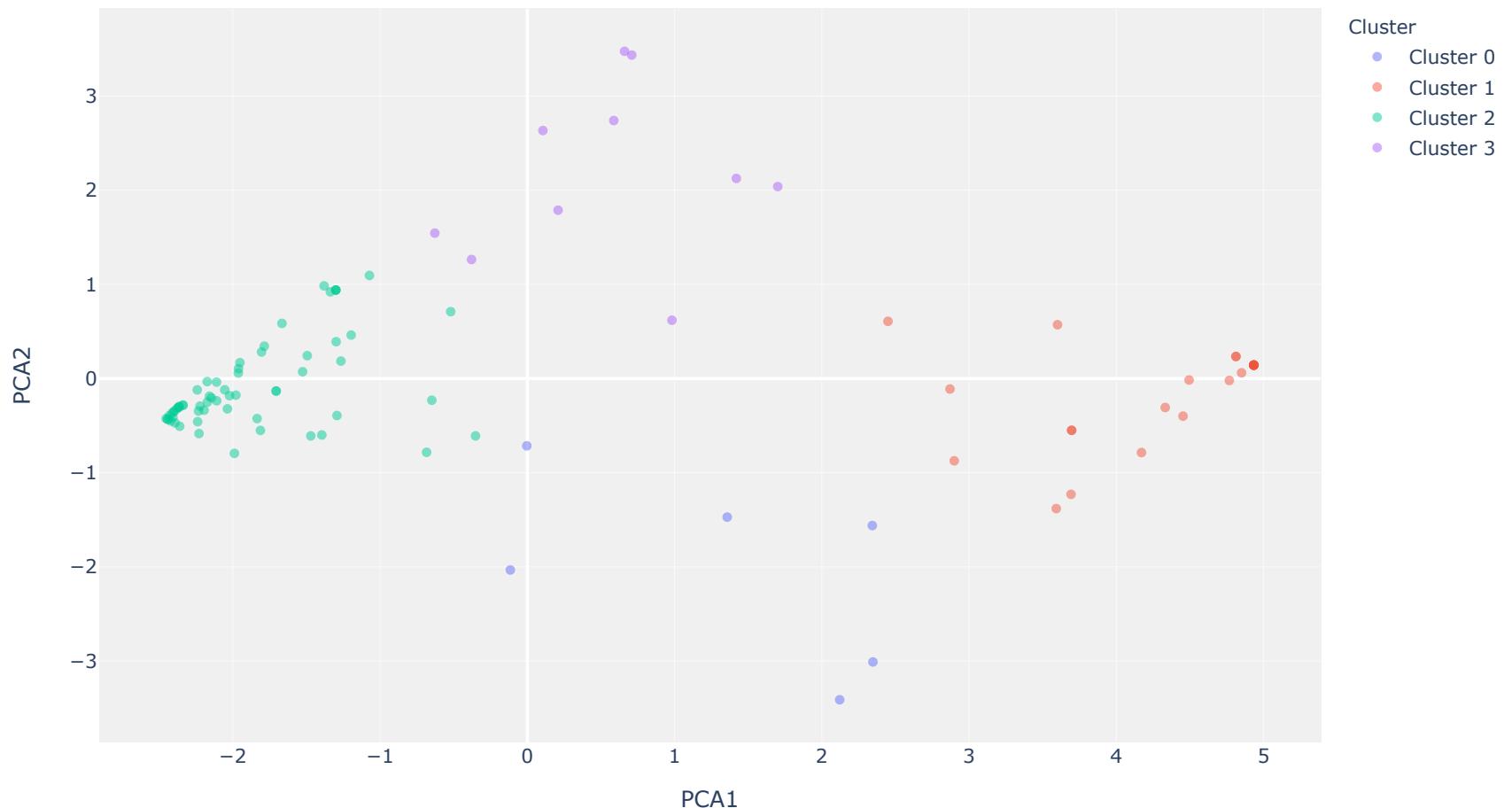
## 2D Cluster PCA Plot



```
In [101]: 1 evaluate_model(kmeansT2_4)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

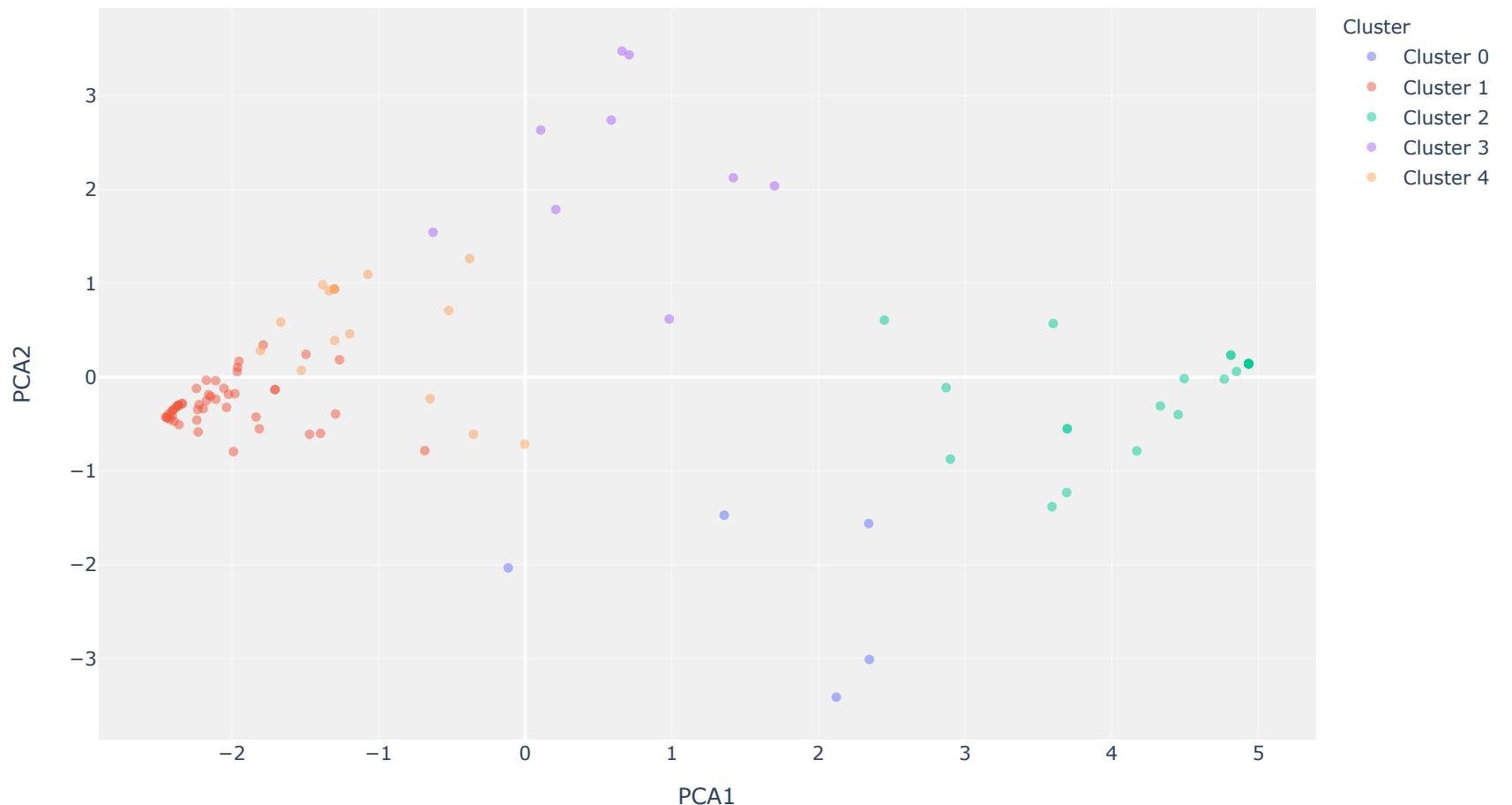
## 2D Cluster PCA Plot



```
In [102]: 1 evaluate_model(kmeansT2_5)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

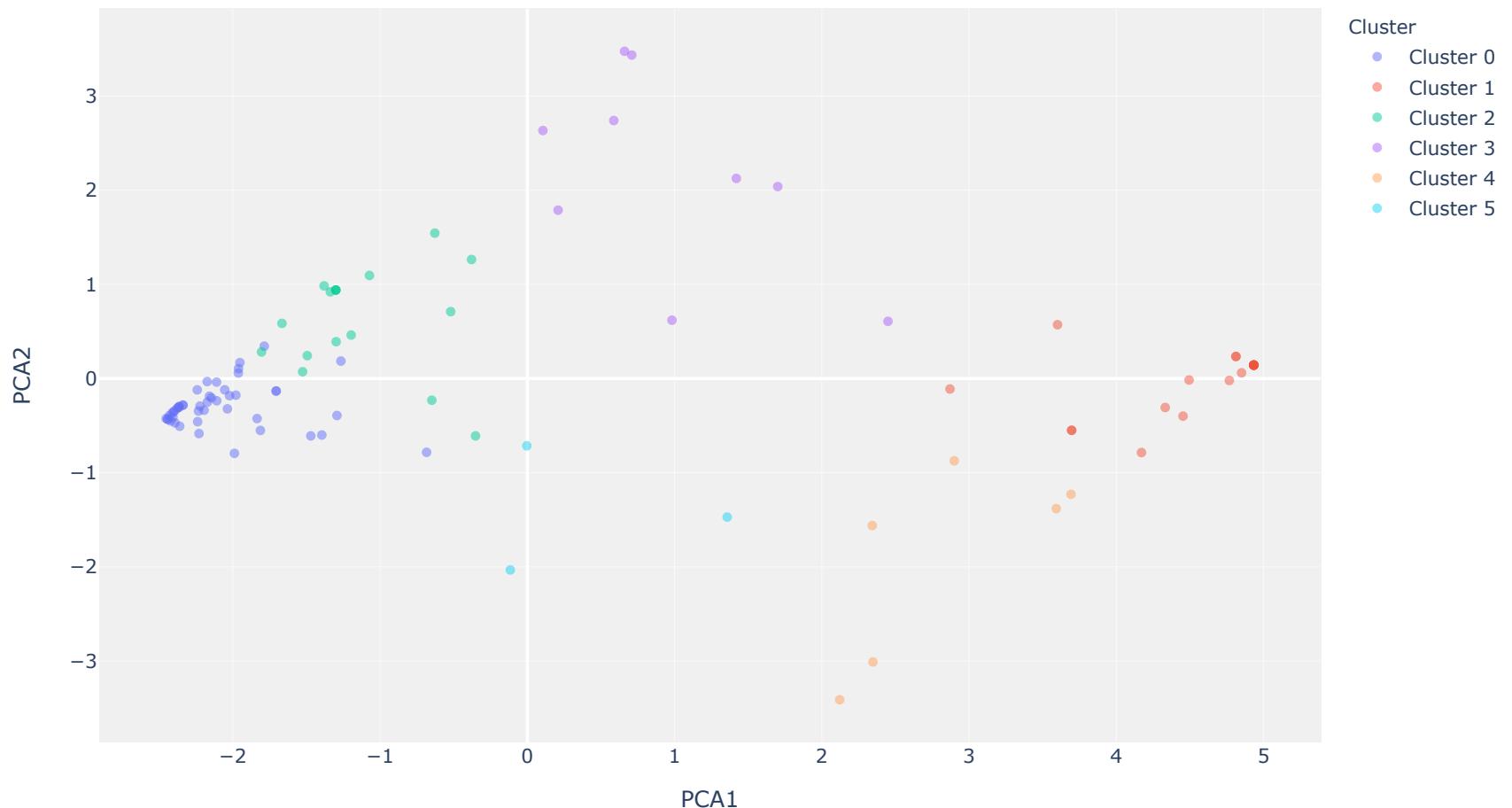
## 2D Cluster PCA Plot



```
In [103]: 1 evaluate_model(kmeansT2_6)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

2D Cluster PCA Plot



```
In [104]: 1 kmeans_T2_6_assign = assign_model(kmeansT2_6)
```

```
In [105]: 1 kmeans_T2_6_assign.head()
```

Out[105]:

	encap	encap-VPL	constructors	constructors-VPL	classes1	classes2	classes2-VPL	static	static-VPL	Cluster
0	148.0	100.0	110.0	165.0	80.0	170.0	95.0	80.0	270.0	Cluster 0
1	150.0	100.0	110.0	165.0	80.0	170.0	95.0	80.0	270.0	Cluster 0
2	150.0	100.0	110.0	165.0	80.0	170.0	92.0	80.0	270.0	Cluster 0
3	150.0	100.0	110.0	165.0	80.0	170.0	95.0	77.0	270.0	Cluster 0
4	150.0	100.0	110.0	157.0	80.0	170.0	95.0	80.0	270.0	Cluster 0

```
In [106]: 1 clustersT['T2'] = kmeans_T2_6_assign['Cluster'] = kmeans_T2_6_assign['Cluster'].str.replace('\D+', '')
```

```
In [107]: 1 clustersT['T2']
```

```
Out[107]: 0      0
1      0
2      0
3      0
4      0
..
97     1
98     1
99     1
100    1
101    1
Name: T2, Length: 102, dtype: object
```

```
In [108]: 1 kmeans_T2_2_assign = assign_model(kmeansT2_2)
```

```
In [109]: 1 clustersTT['T2'] = kmeans_T2_2_assign['Cluster'] = kmeans_T2_2_assign['Cluster'].str.replace('\D+', '')
```

```
In [110]: 1 clustersTT['T2']
```

```
Out[110]: 0      0  
1      0  
2      0  
3      0  
4      0  
..  
97     1  
98     1  
99     1  
100    1  
101    1  
Name: Cluster, Length: 102, dtype: object
```

```
In [111]: 1 kmeans_T2_4_assign = assign_model(kmeansT2_4)
```

```
In [112]: 1 clustersTTT['T2'] = kmeans_T2_4_assign['Cluster'] = kmeans_T2_4_assign['Cluster'].str.replace('\D+', '')
```

```
In [ ]: 1
```

## T3

```
In [113]: 1 from pycaret.clustering import *
```

```
2
3 input_dataset4 = setup(dataset_T3, normalize = True,
4                         session_id = 123)
```

	Description	Value
0	session_id	123
1	Original Data	(102, 4)
2	Missing Values	False
3	Numeric Features	4
4	Categorical Features	0
5	Ordinal Features	False
6	High Cardinality Features	False
7	High Cardinality Method	None
8	Transformed Data	(102, 4)
9	CPU Jobs	-1
10	Use GPU	False

```
In [114]: 1 kmeansT3_6 = create_model('kmeans', num_clusters=6)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.5948	187.3231	0.7725	0	0	0

```
In [115]: 1 kmeansT3_5 = create_model('kmeans', num_clusters=5)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.5775	177.7359	0.7942	0	0	0

```
In [116]: 1 kmeansT3_4 = create_model('kmeans', num_clusters=4)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.6057	167.6004	0.7765	0	0	0

```
In [117]: 1 kmeansT3_3 = create_model('kmeans', num_clusters=3)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.5677	150.9572	0.8597	0	0	0

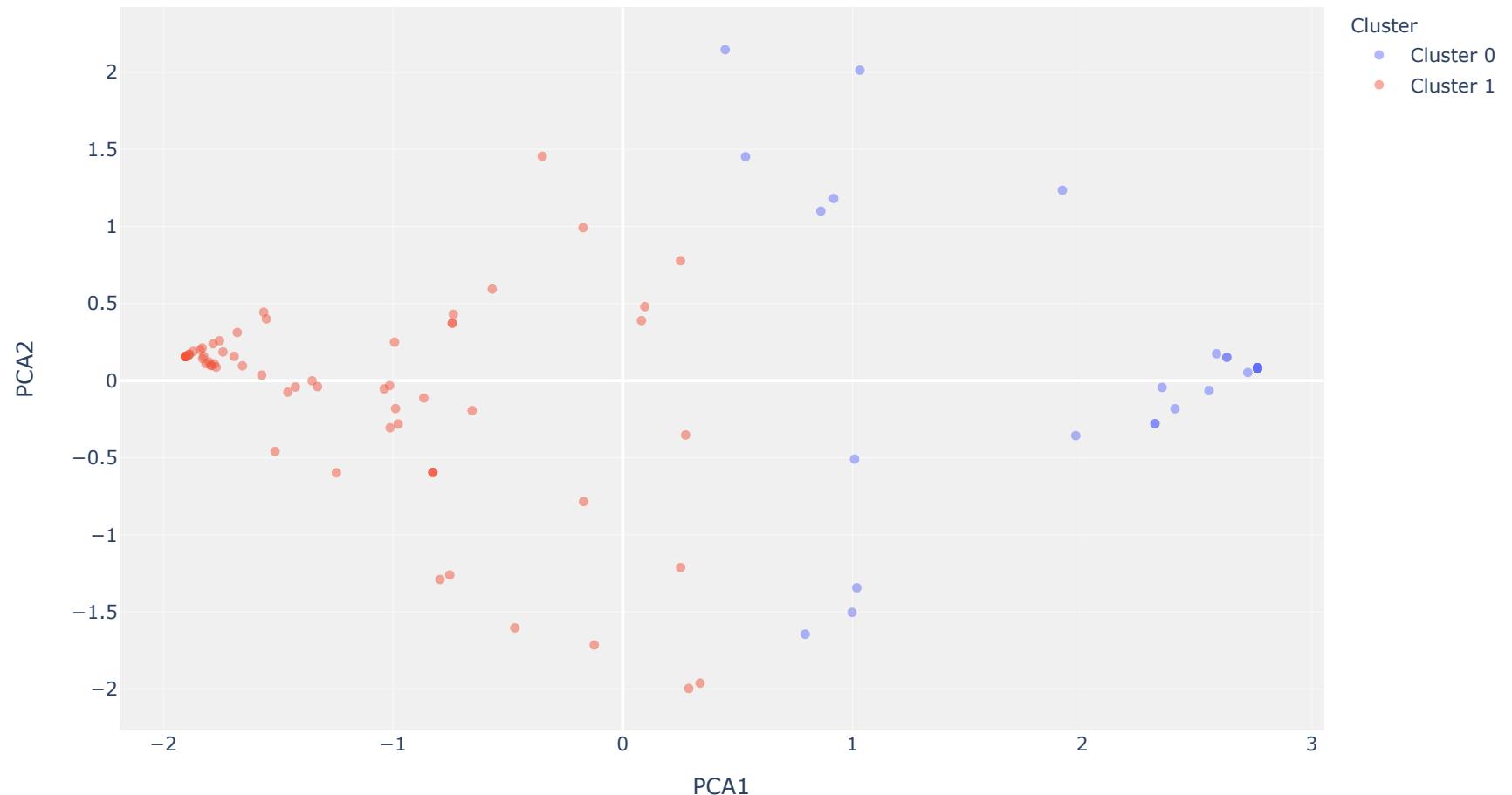
```
In [118]: 1 kmeansT3_2 = create_model('kmeans', num_clusters=2)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.5956	182.6024	0.6036	0	0	0

```
In [119]: 1 | evaluate_model(kmeansT3_2)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

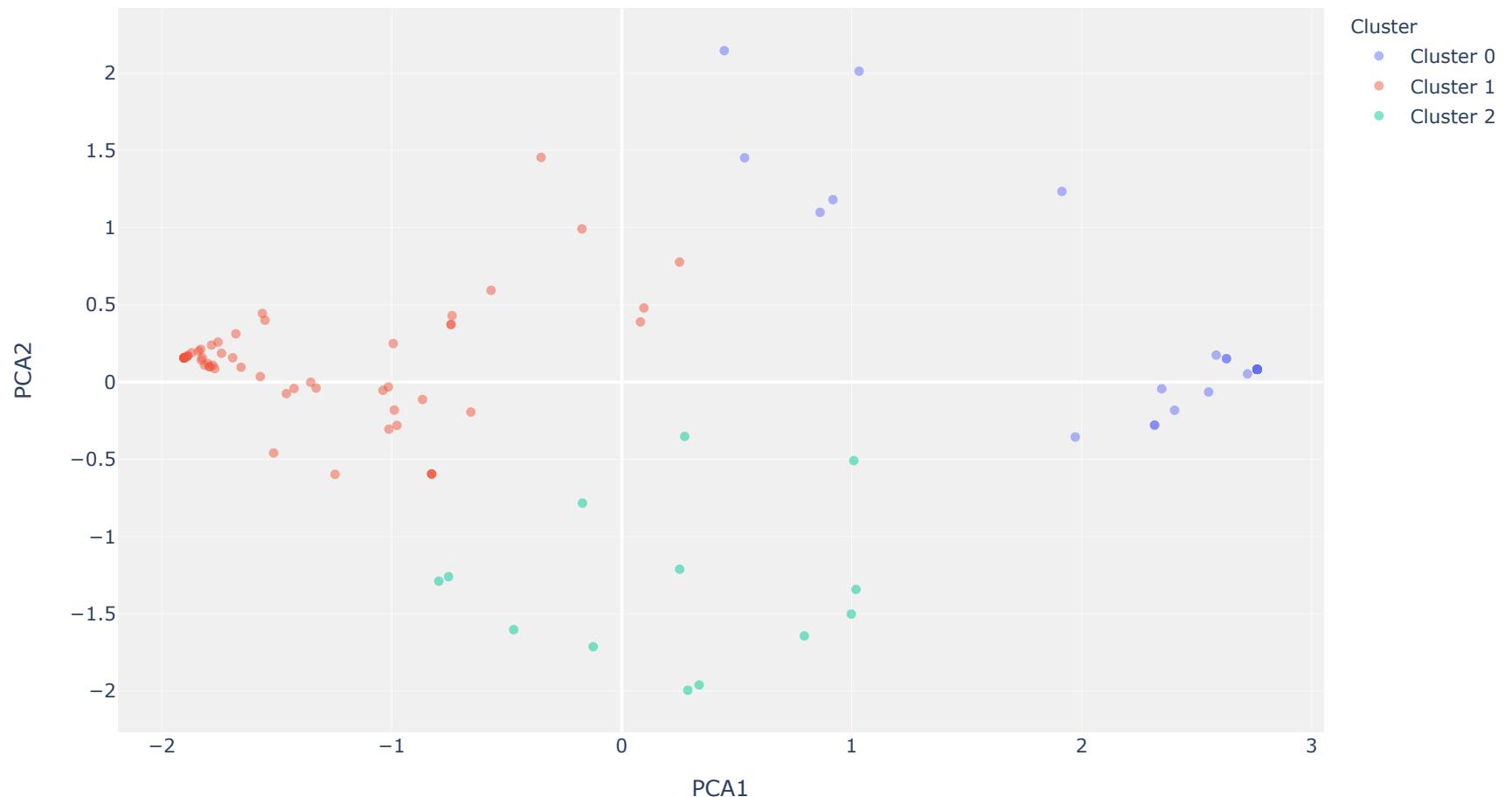
## 2D Cluster PCA Plot



In [120]: 1 | evaluate\_model(kmeansT3\_3)

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

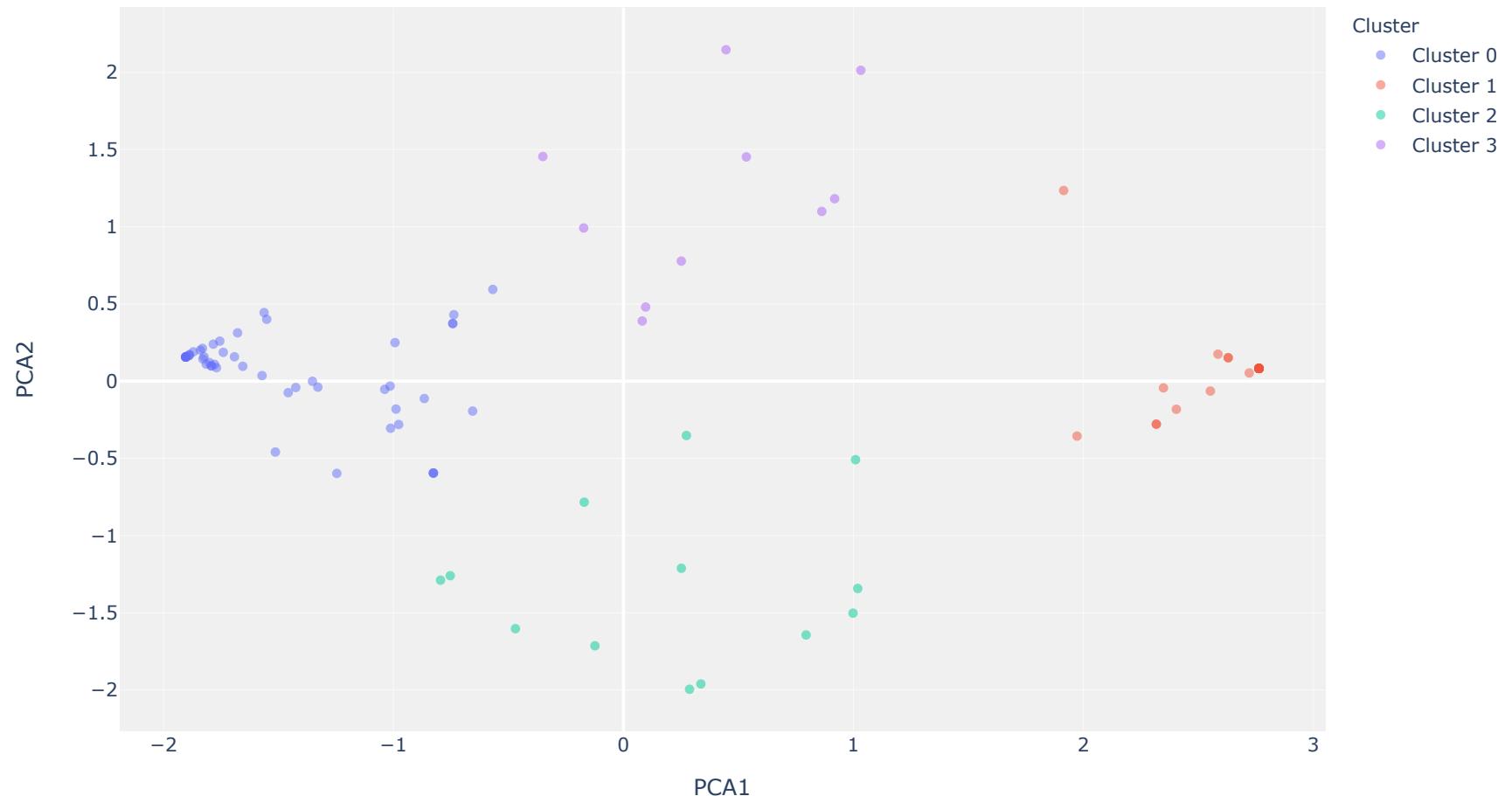
## 2D Cluster PCA Plot



In [121]: 1 | evaluate\_model(kmeansT3\_4)

Plot Type: Cluster PCA Plot (2d) Cluster tSNE (3d) Elbow Silhouette Distance Distribution

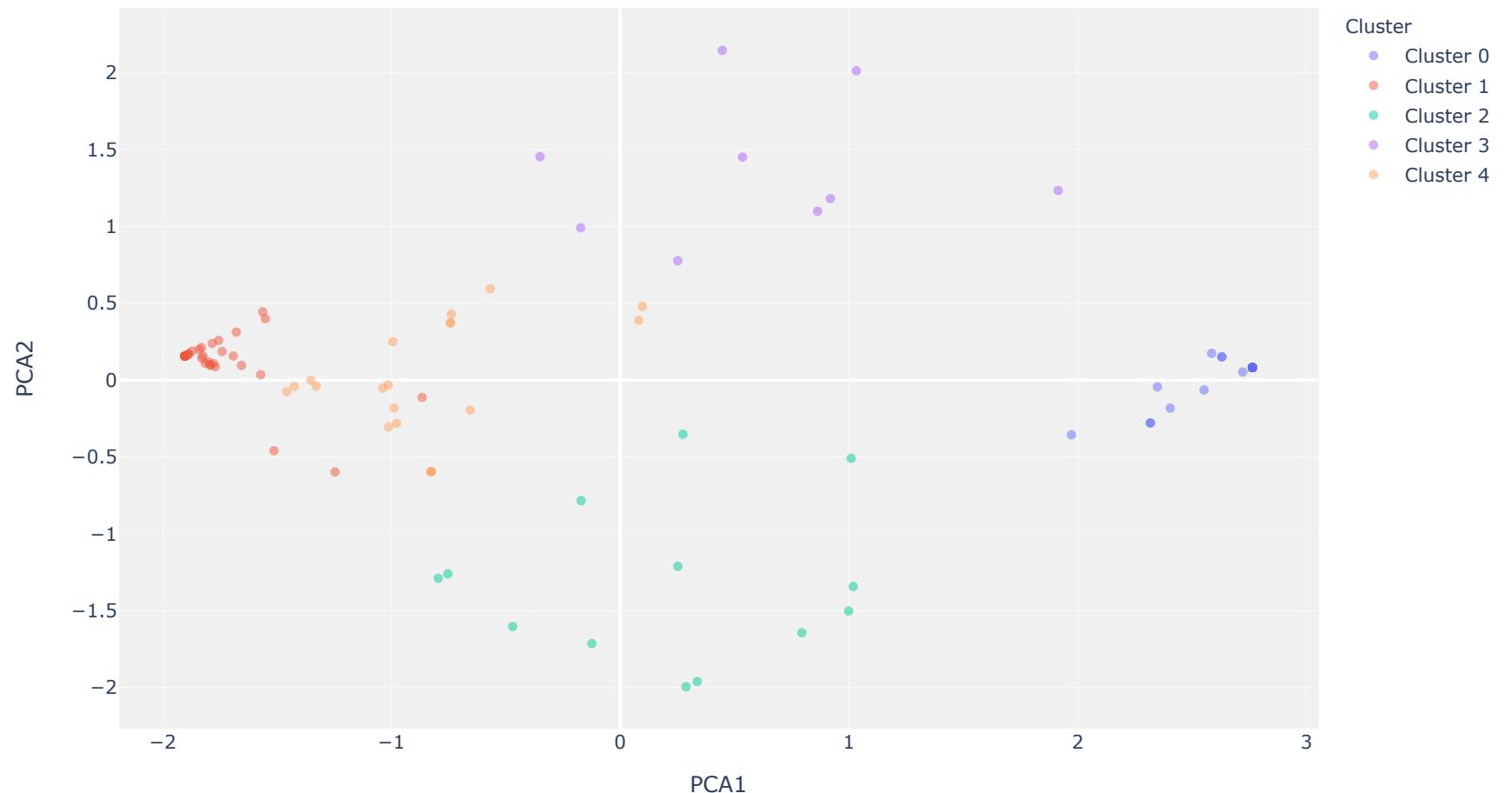
## 2D Cluster PCA Plot



In [122]: 1 | evaluate\_model(kmeansT3\_5)

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

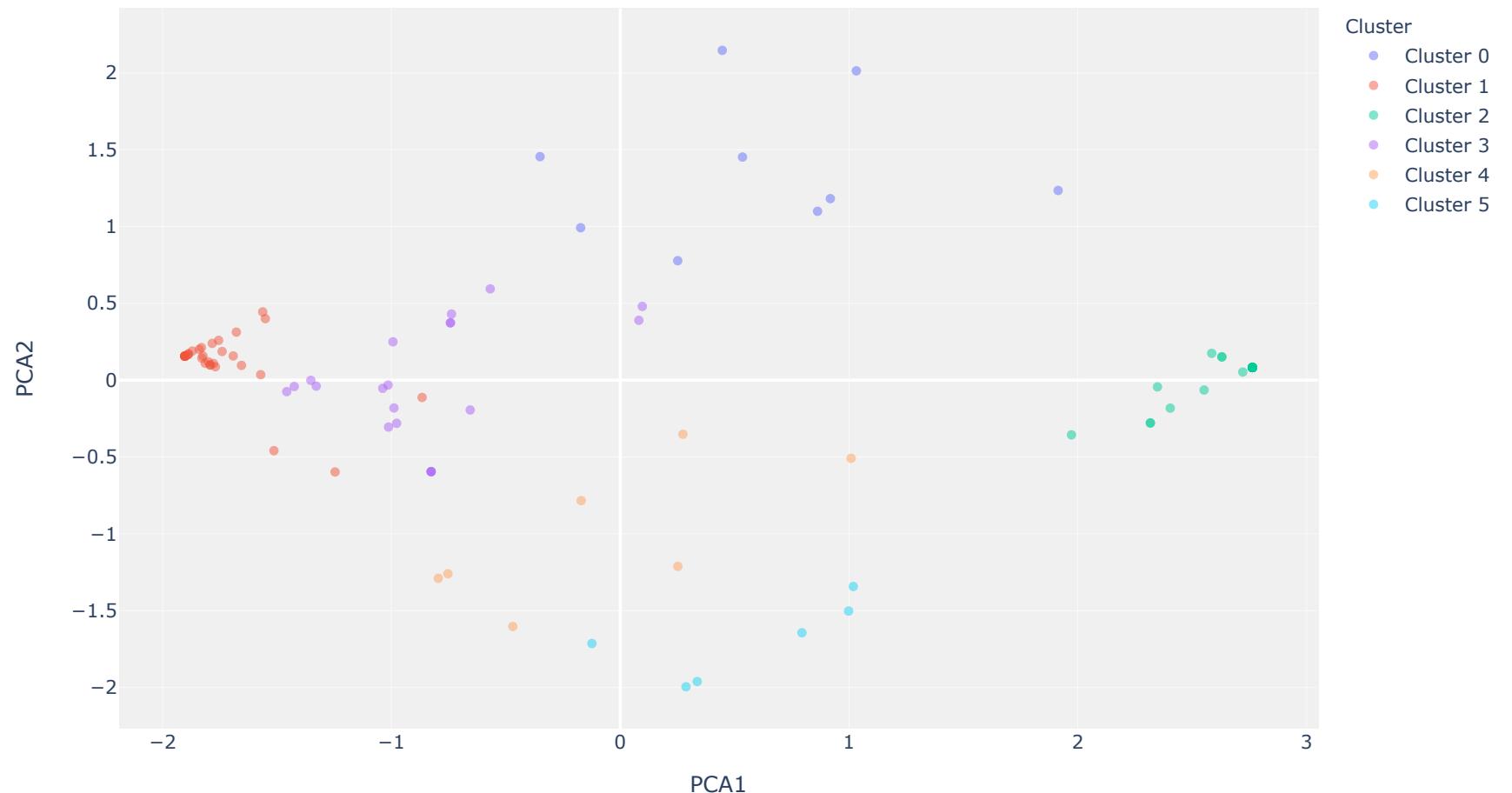
## 2D Cluster PCA Plot



In [123]: 1 | evaluate\_model(kmeansT3\_6)

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

## 2D Cluster PCA Plot



```
In [124]: 1 kmeans_T3_6_assign = assign_model(kmeansT3_6)
2 kmeans_T3_6_assign.head()
3 clustersT['T3'] = kmeans_T3_6_assign['Cluster'] = kmeans_T3_6_assign['Cluster'].str.replace('\D+', '')
```

```
In [125]: 1 clustersT['T3']
```

```
Out[125]: 0      1
1      1
2      1
3      1
4      1
..
97     2
98     2
99     2
100    2
101    2
Name: T3, Length: 102, dtype: object
```

```
In [126]: 1 kmeans_T3_2_assign = assign_model(kmeansT3_2)
```

```
In [127]: 1 clustersTT['T3'] = kmeans_T3_2_assign['Cluster'] = kmeans_T3_2_assign['Cluster'].str.replace('\D+', '')
```

```
In [128]: 1 clustersTT['T3']
```

```
Out[128]: 0      1
1      1
2      1
3      1
4      1
..
97     0
98     0
99     0
100    0
101    0
Name: Cluster, Length: 102, dtype: object
```

```
In [129]: 1 kmeans_T3_4_assign = assign_model(kmeansT3_4)
```

```
In [130]: 1 clustersTTT['T3'] = kmeans_T3_4_assign['Cluster'] = kmeans_T3_4_assign['Cluster'].str.replace('\D+', '')
```

## VPL clustering

```
In [131]: 1 from pycaret.clustering import *
```

```
2
3 input_dataset5 = setup(dataset_reduced_VPL, normalize = True,
4                         numeric_features = ['intro-VPL','methods-VPL','encap-VPL','constructors-VPL','classes2-VPL','static-VPL',
5                         'inher-VPL', 'polymorphism-VPL'],
6                         session_id = 123)
```

	Description	Value
0	session_id	123
1	Original Data	(102, 8)
2	Missing Values	False
3	Numeric Features	8
4	Categorical Features	0
5	Ordinal Features	False
6	High Cardinality Features	False
7	High Cardinality Method	None
8	Transformed Data	(102, 8)
9	CPU Jobs	-1
10	Use GPU	False

```
In [132]: 1 kmeans_vpl2 = create_model('kmeans', num_clusters = 2)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.5482	162.3892	0.703	0	0	0

```
In [133]: 1 kmeans_vpl3 = create_model('kmeans', num_clusters = 3)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.4275	121.716	1.1015	0	0	0

```
In [134]: 1 kmeans_vpl4 = create_model('kmeans', num_clusters = 4)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.396	100.6032	1.0784	0	0	0

```
In [135]: 1 kmeans_vpl5 = create_model('kmeans', num_clusters = 5)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.4148	91.6369	1.18	0	0	0

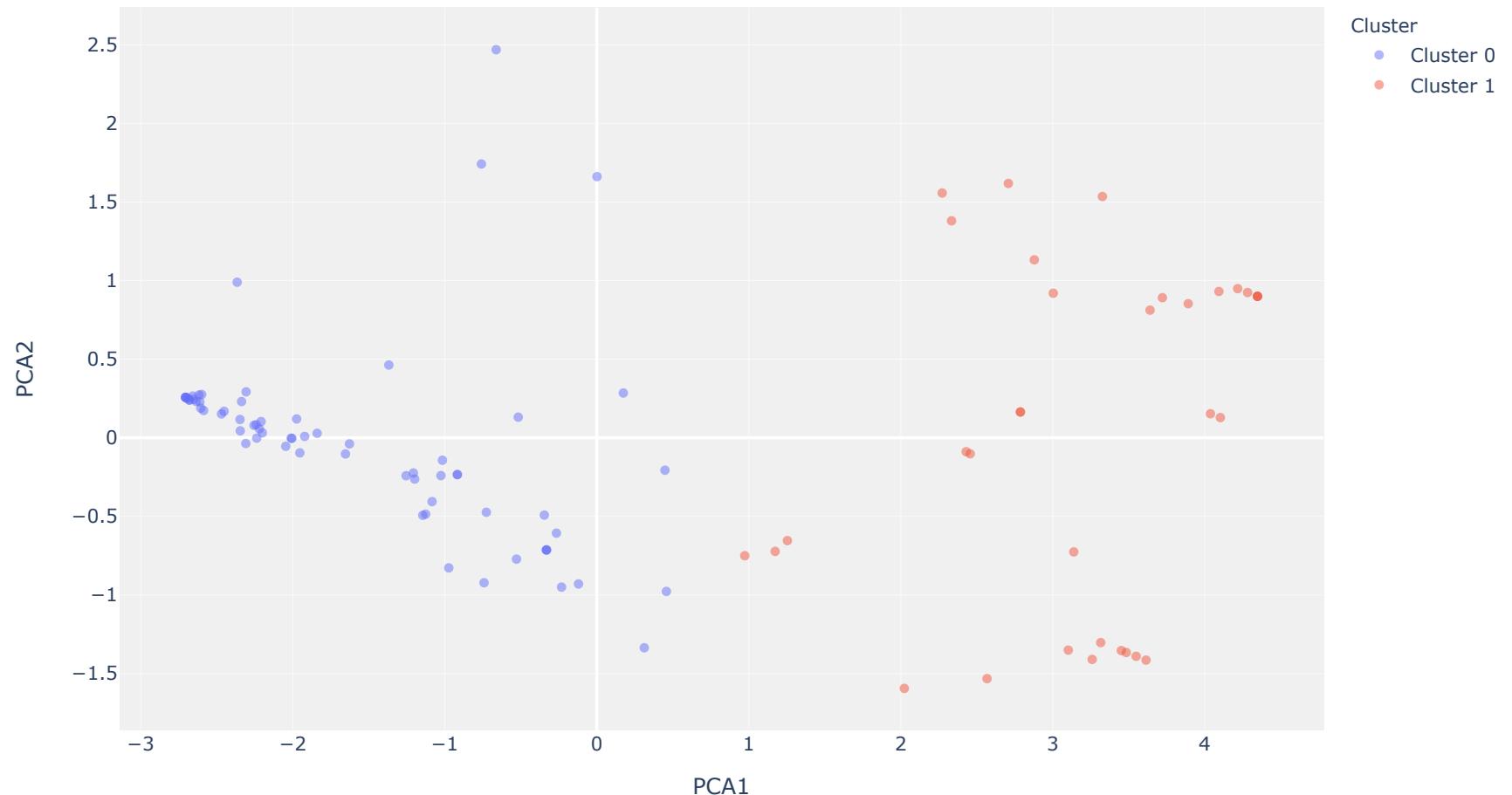
```
In [136]: 1 kmeans_vpl6 = create_model('kmeans', num_clusters = 6)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.4228	81.6195	1.152	0	0	0

In [137]: 1 | evaluate\_model(kmeans\_vp12)

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

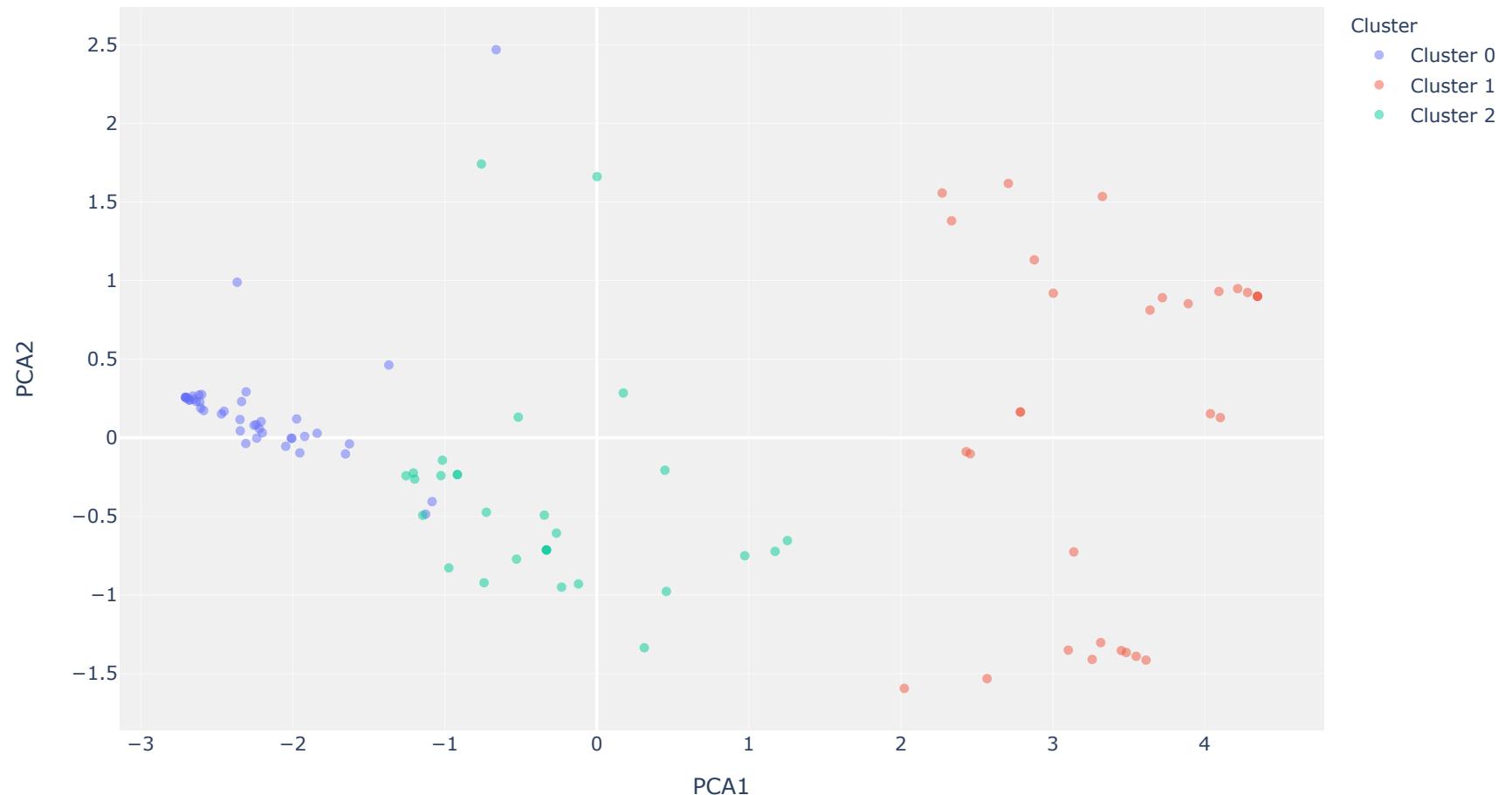
## 2D Cluster PCA Plot



In [138]: 1 | evaluate\_model(kmeans\_vp13)

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

## 2D Cluster PCA Plot



```
In [139]: 1 evaluate_model(kmeans_vp14)
```

Plot Type: Cluster PCA Plot (2d)

Cluster TSnE (3d)

Elbow

Silhouette

Distance

Distribution

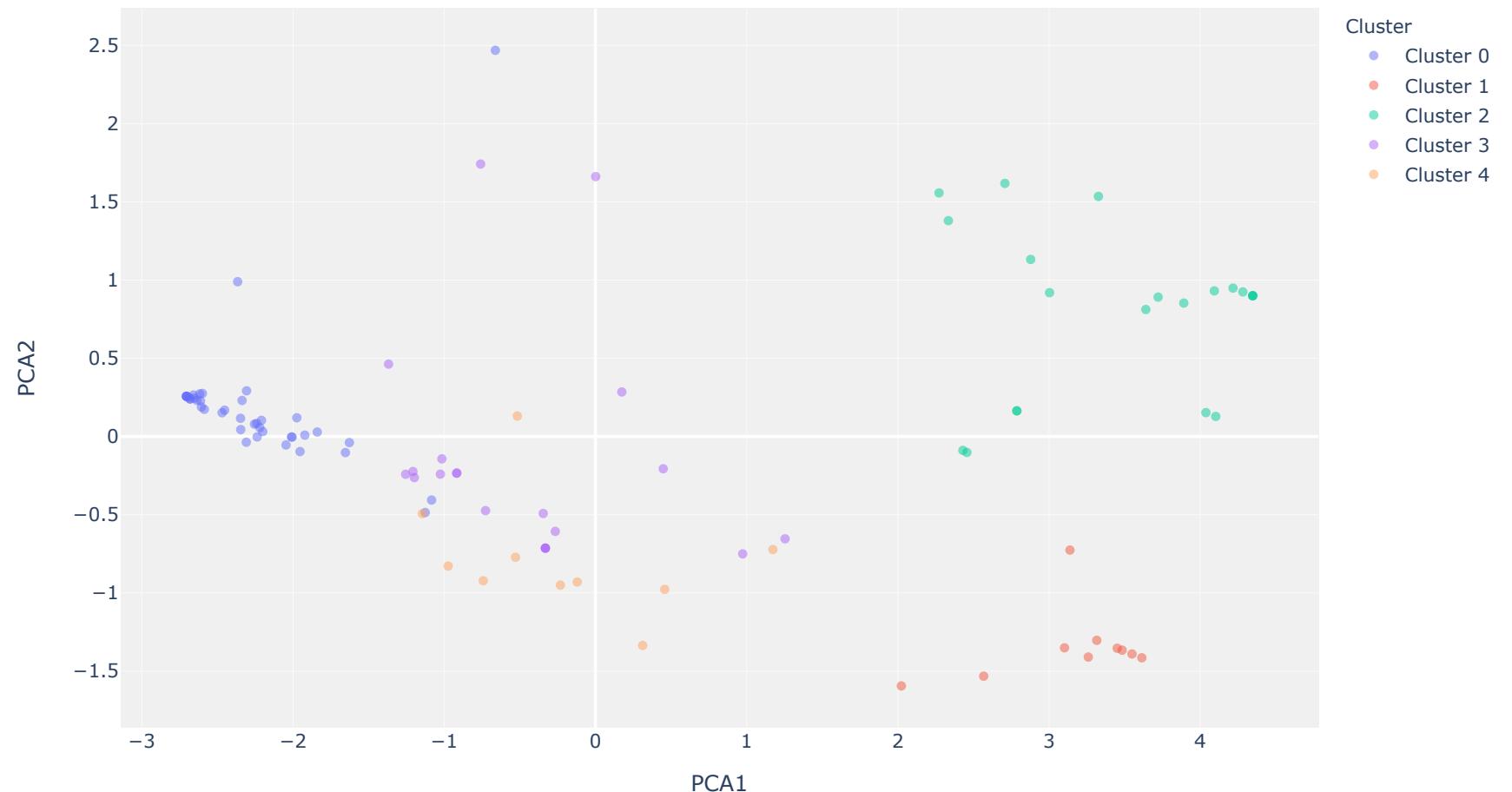
## 2D Cluster PCA Plot



```
In [140]: 1 evaluate_model(kmeans_vp15)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

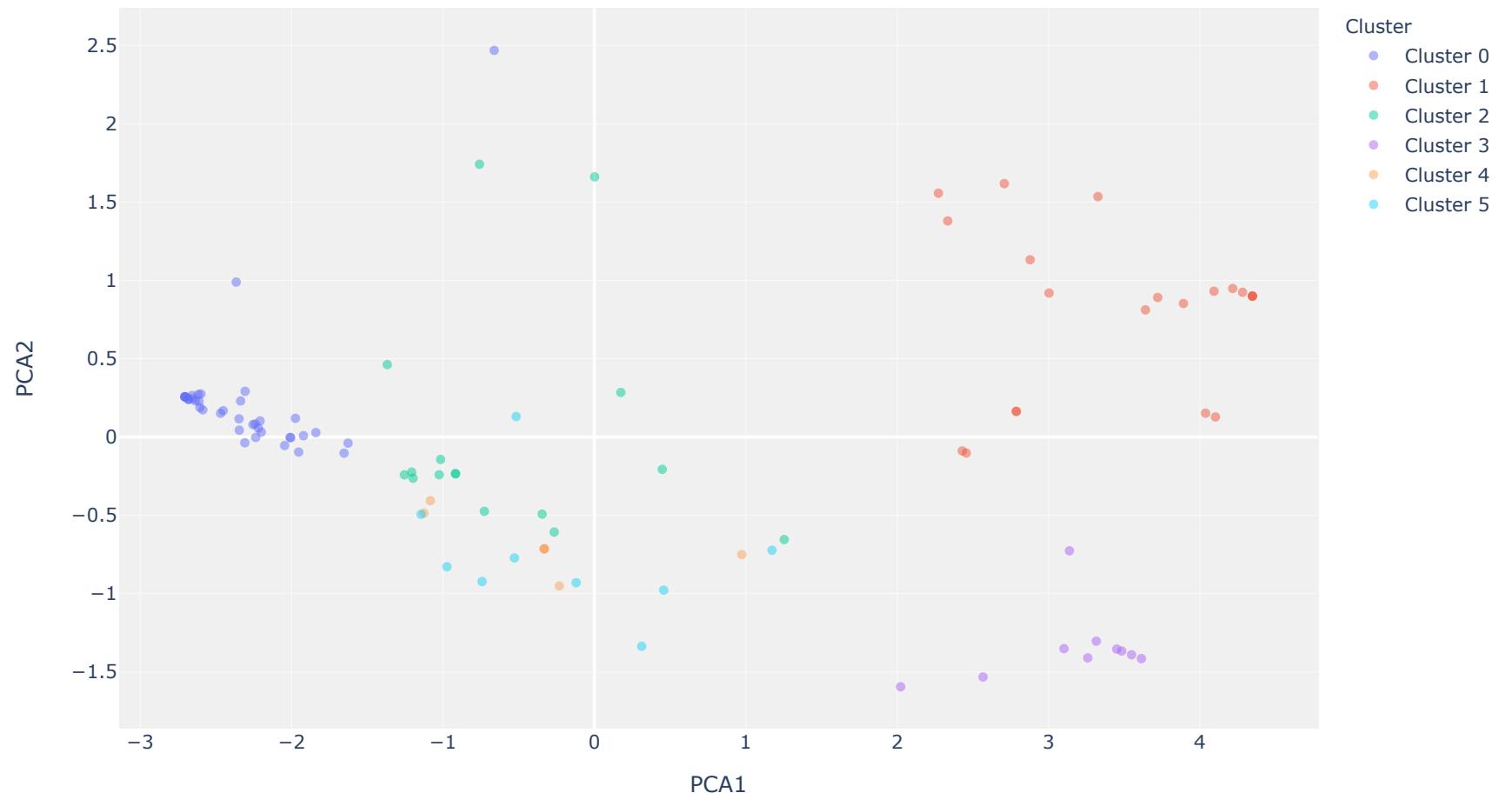
## 2D Cluster PCA Plot



```
In [141]: 1 evaluate_model(kmeans_vp16)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

2D Cluster PCA Plot



```
In [142]: 1 kmeans_VPL_2_assign = assign_model(kmeans_vp12)
```

```
In [143]: 1 clustersVPL = pd.DataFrame()
```

```
In [144]: 1 clustersVPL['VPL'] = kmeans_VPL_2_assign['Cluster'] = kmeans_VPL_2_assign['Cluster'].str.replace('\D+', '')
```

```
In [145]: 1 clustersVPL.head()
```

Out[145]:

VPL
0 0
1 0
2 0
3 0
4 0

```
In [ ]: 1
```

In [ ]: 1

## Micro-lessons

```
In [146]: 1 from pycaret.clustering import *
2
3 input_dataset6 = setup(dataset_reduced_mc_lessons, normalize = True,
4                         numeric_features = ['intro','methods','encap','constructors','classes1','classes2','static', 'inher', 'pol
5                         session_id = 123)
```

	Description	Value
0	session_id	123
1	Original Data	(102, 9)
2	Missing Values	False
3	Numeric Features	9
4	Categorical Features	0
5	Ordinal Features	False
6	High Cardinality Features	False
7	High Cardinality Method	None
8	Transformed Data	(102, 9)
9	CPU Jobs	-1
10	Use GPU	False
11	Log Experiment	False
12	Experiment Name	cluster-default-name
13	USI	a27e
14	Imputation Type	simple
15	Iterative Imputation Iteration	None
16	Numeric Imputer	mean
17	Iterative Imputation Numeric Model	None
18	Categorical Imputer	mode
19	Iterative Imputation Categorical Model	None
20	Unknown Categoricals Handling	least_frequent
21	Normalize	True
22	Normalize Method	zscore
23	Transformation	False
24	Transformation Method	None
25	PCA	False
26	PCA Method	None
27	PCA Components	None

	Description	Value
28	Ignore Low Variance	False
29	Combine Rare Levels	False
30	Rare Level Threshold	None
31	Numeric Binning	False
32	Remove Outliers	False
33	Outliers Threshold	None
34	Remove Multicollinearity	False
35	Multicollinearity Threshold	None
36	Clustering	False
37	Clustering Iteration	None
38	Polynomial Features	False
39	Polynomial Degree	None
40	Trigonometry Features	False
41	Polynomial Threshold	None
42	Group Features	False
43	Feature Selection	False
44	Feature Selection Method	classic
45	Features Selection Threshold	None
46	Feature Interaction	False
47	Feature Ratio	False
48	Interaction Threshold	None

In [147]: 1 kmeans\_mc\_2 = create\_model('kmeans', num\_clusters = 2)

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.7363	383.3815	0.4624	0	0	0

```
In [148]: 1 kmeans_mc_3 = create_model('kmeans', num_clusters = 3)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.6896	286.8699	0.6816	0	0	0

```
In [149]: 1 kmeans_mc_4 = create_model('kmeans', num_clusters = 4)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.5769	274.3522	0.8355	0	0	0

```
In [150]: 1 kmeans_mc_5 = create_model('kmeans', num_clusters = 5)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.5605	262.406	0.9686	0	0	0

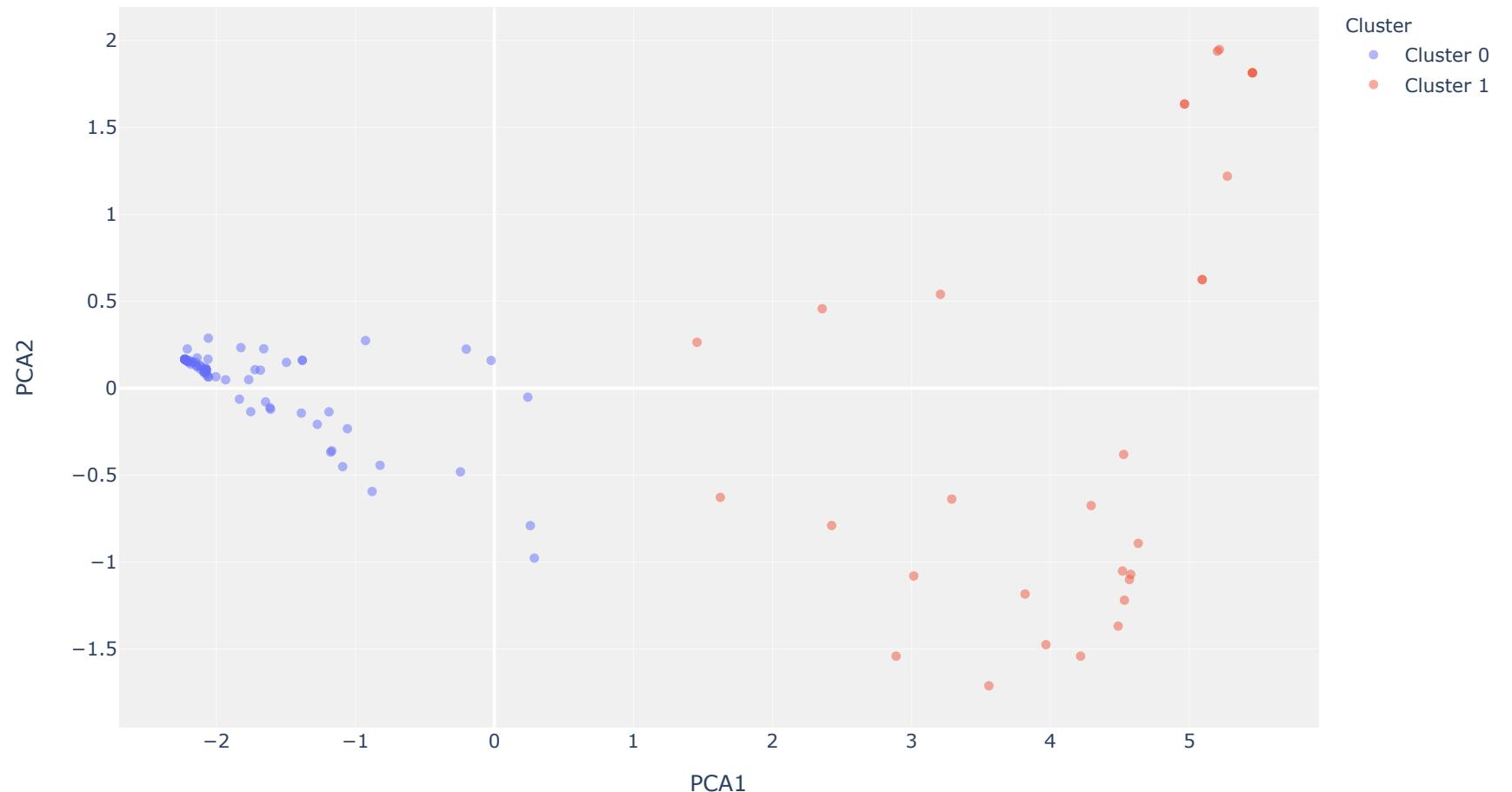
```
In [151]: 1 kmeans_mc_6 = create_model('kmeans', num_clusters = 6)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.5601	241.889	0.9954	0	0	0

```
In [152]: 1 evaluate_model(kmeans_mc_2)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

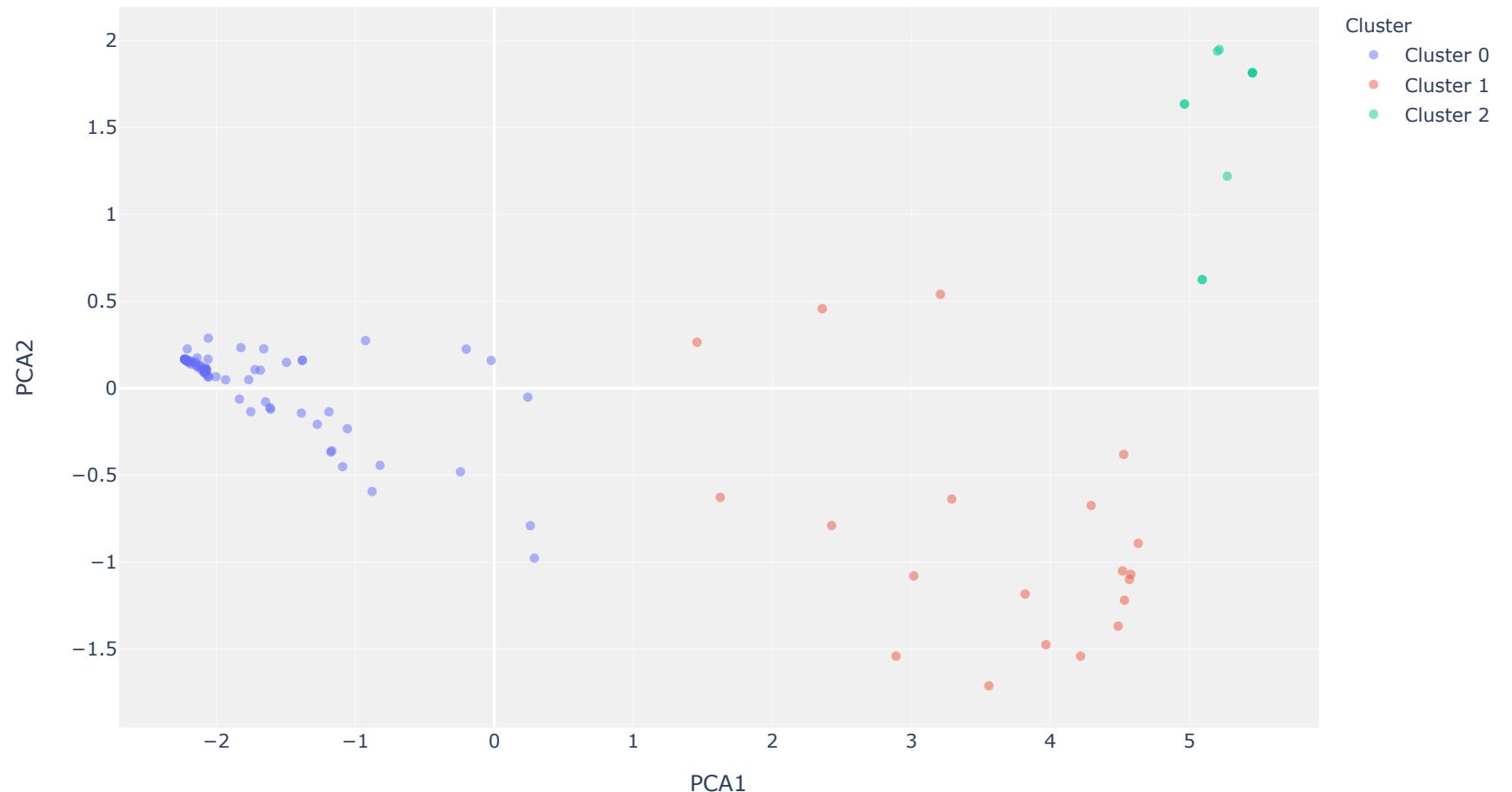
## 2D Cluster PCA Plot



```
In [153]: 1 evaluate_model(kmeans_mc_3)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

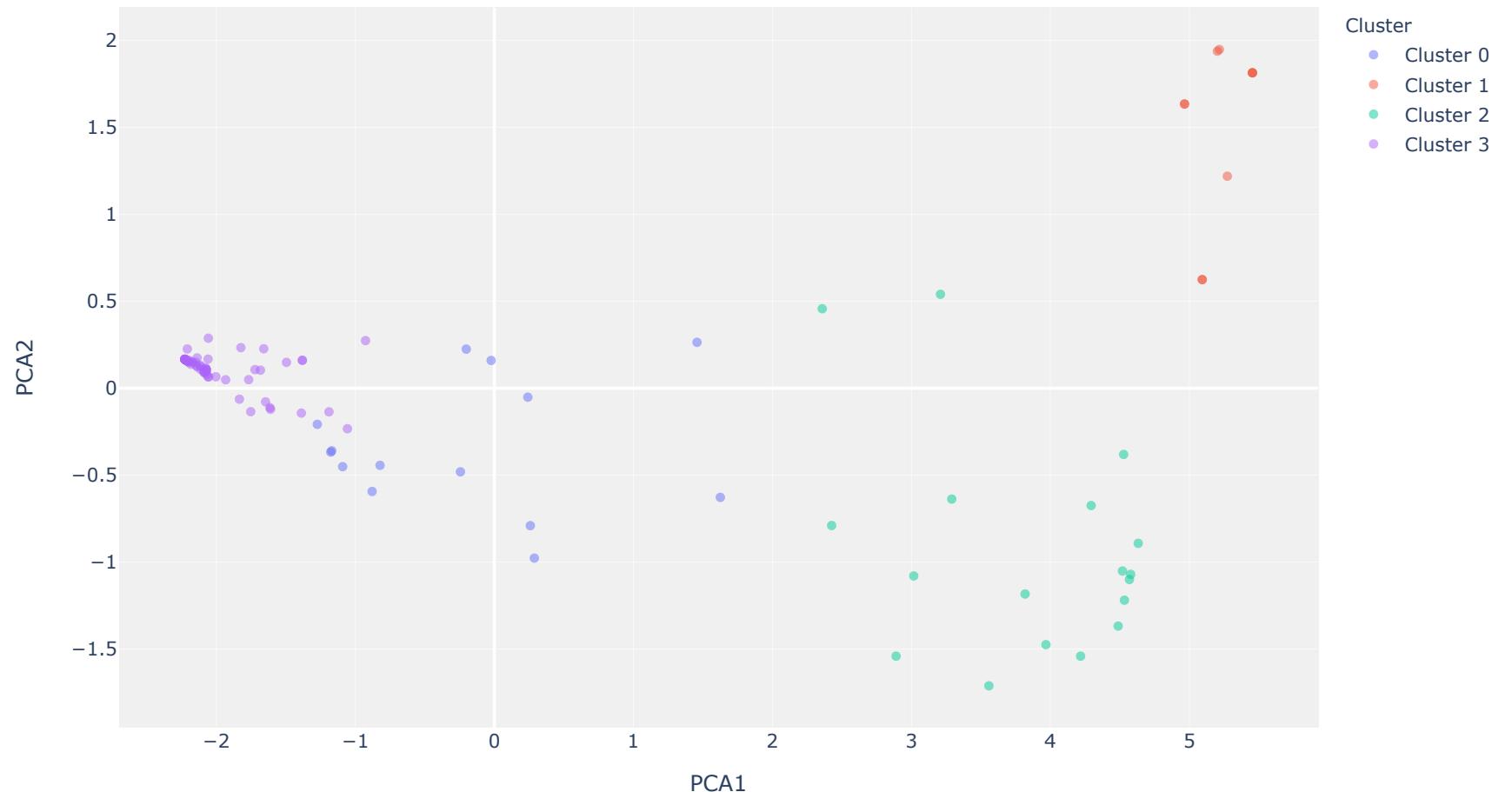
## 2D Cluster PCA Plot



```
In [154]: 1 evaluate_model(kmeans_mc_4)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

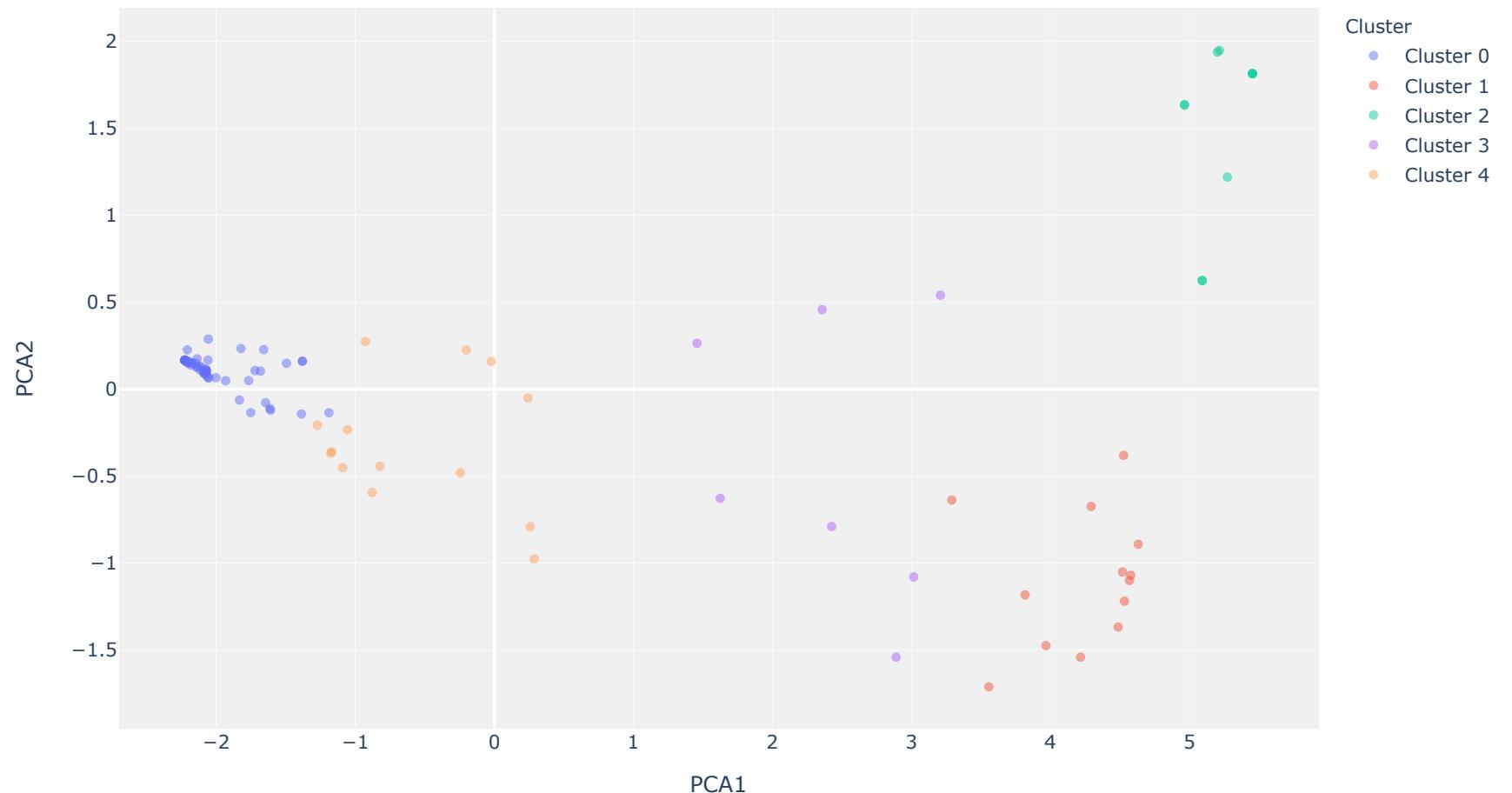
## 2D Cluster PCA Plot



```
In [155]: 1 evaluate_model(kmeans_mc_5)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

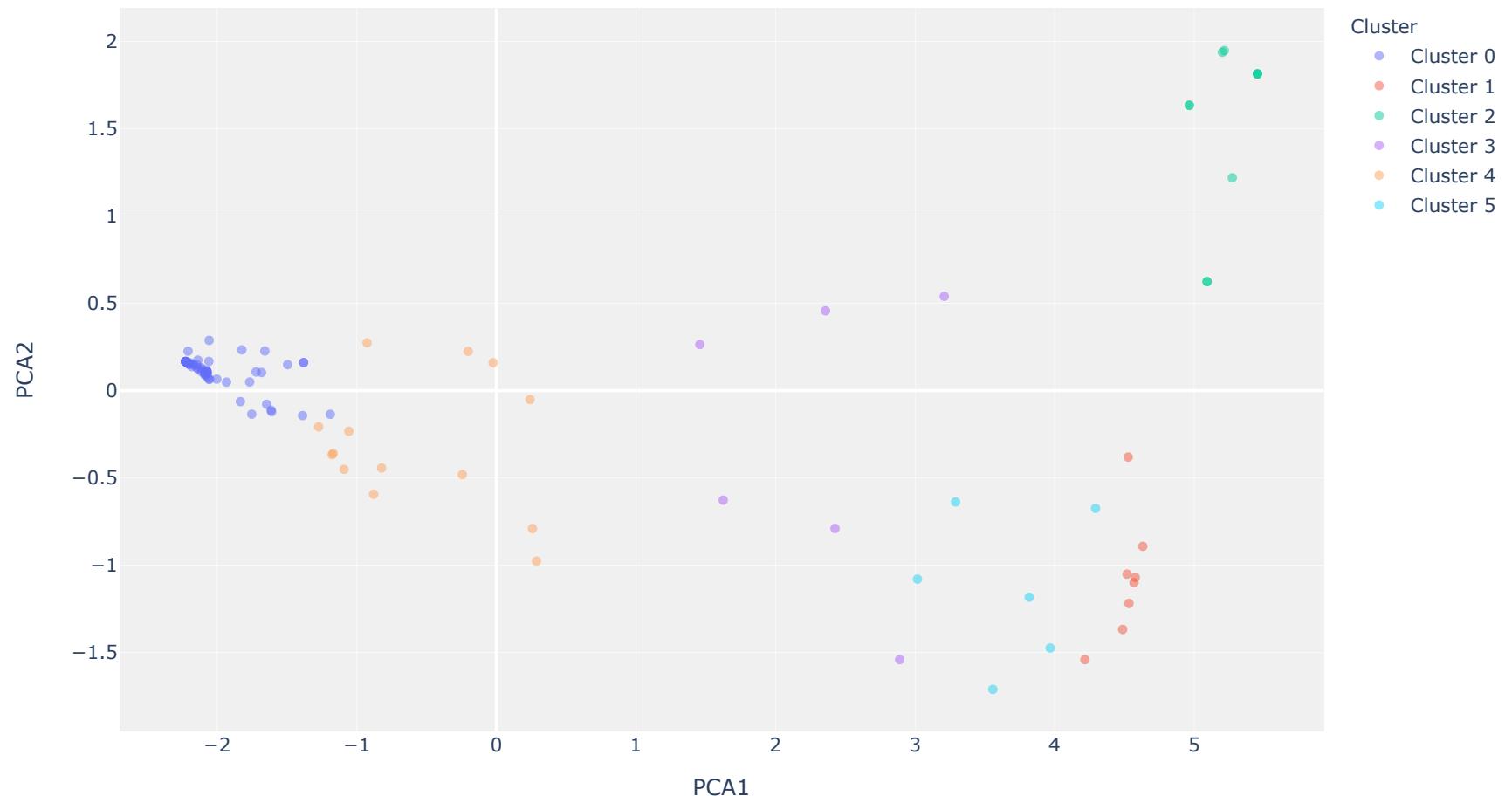
## 2D Cluster PCA Plot



```
In [156]: 1 evaluate_model(kmeans_mc_6)
```

Plot Type: Cluster PCA Plot (2d) Cluster TSnE (3d) Elbow Silhouette Distance Distribution

2D Cluster PCA Plot



```
In [157]: 1 kmeans_MC_2_assign = assign_model(kmeans_mc_2)
```

```
In [158]: 1 clustersMC = pd.DataFrame()
```

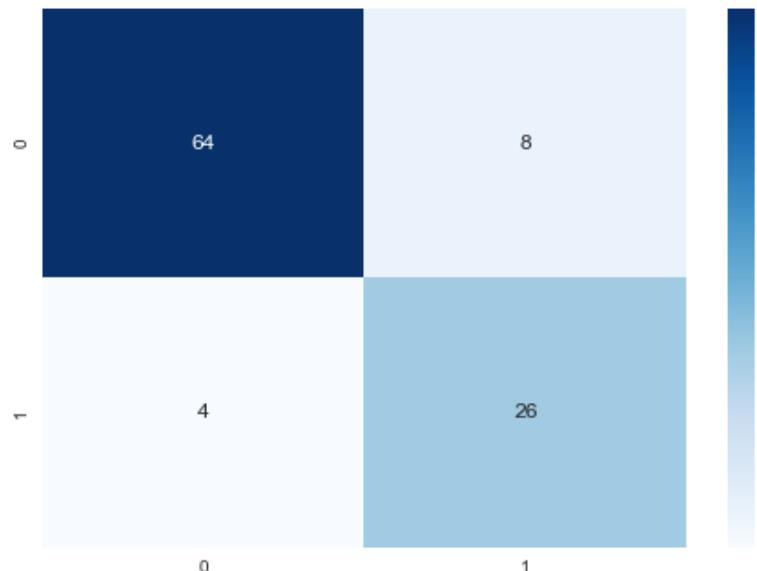
```
In [159]: 1 clustersMC['MC'] = kmeans_MC_2_assign['Cluster'] = kmeans_MC_2_assign['Cluster'].str.replace('\D+', '')
```

```
In [ ]: 1
```

```
In [160]: 1 from sklearn.metrics.cluster import contingency_matrix
```

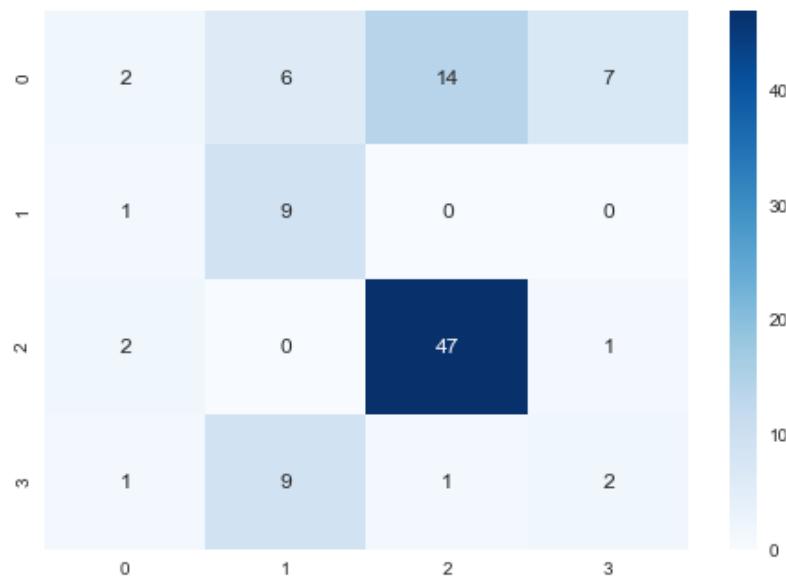
```
In [161]: 1 sns.heatmap(contingency_matrix(clustersMC['MC'], clustersVPL['VPL']), annot=True, cmap='Blues')
```

```
Out[161]: <AxesSubplot:>
```



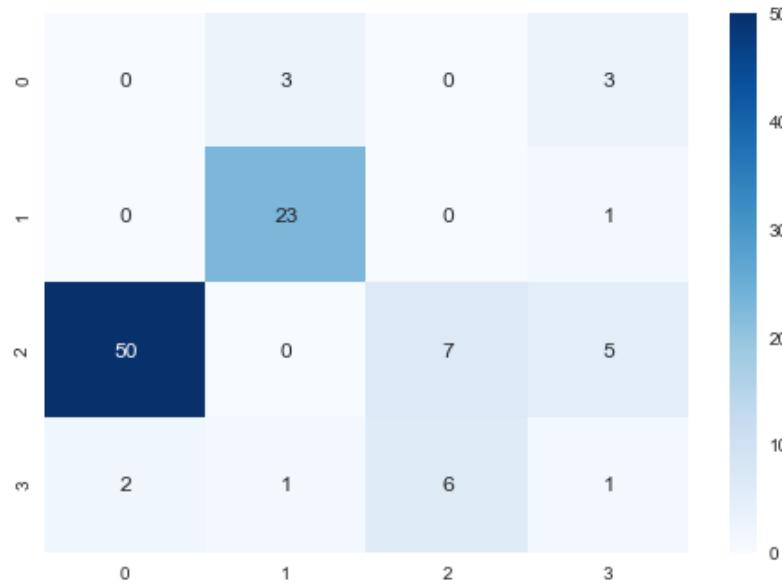
```
In [162]: 1 sns.heatmap(contingency_matrix(clustersTTT['T1'], clustersTTT['T2']), annot=True,cmap='Blues')
```

```
Out[162]: <AxesSubplot:>
```



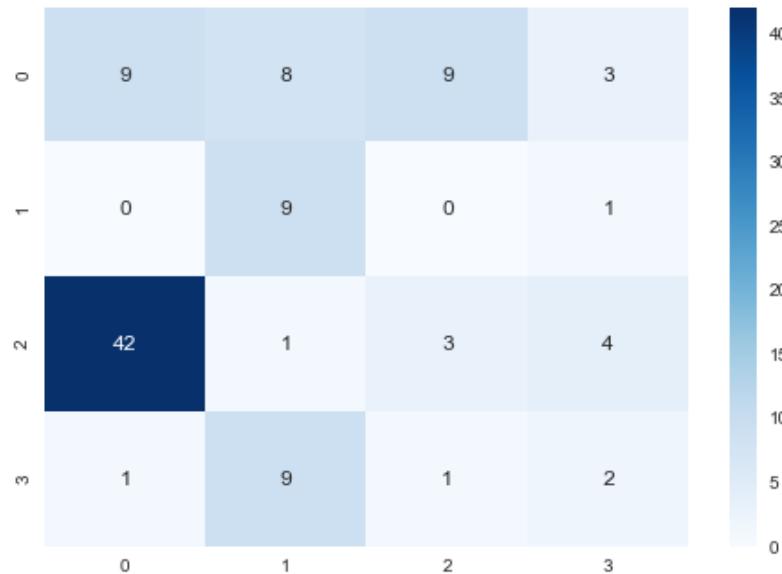
```
In [163]: 1 sns.heatmap(contingency_matrix(clustersTTT['T2'], clustersTTT['T3']), annot=True,cmap='Blues')
```

Out[163]: <AxesSubplot:>



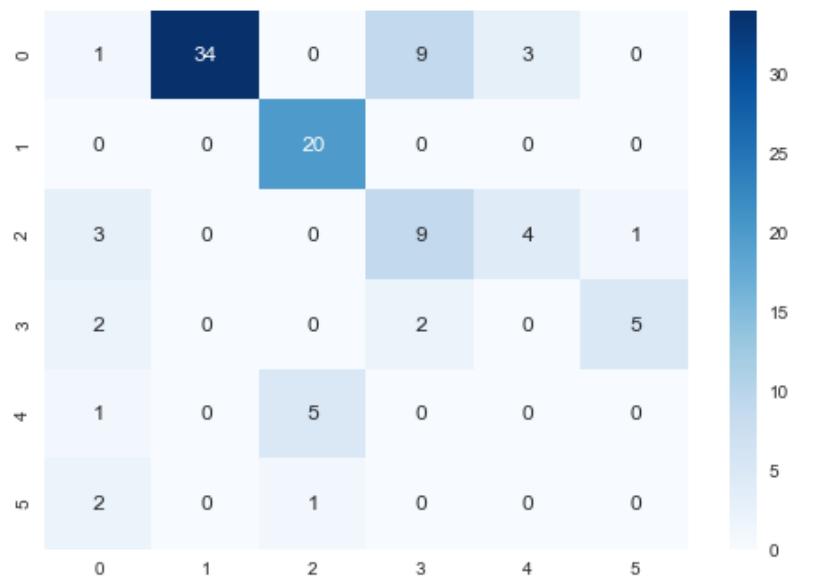
```
In [164]: 1 sns.heatmap(contingency_matrix(clustersTTT['T1'], clustersTTT['T3']), annot=True,cmap='Blues')
```

Out[164]: <AxesSubplot:>



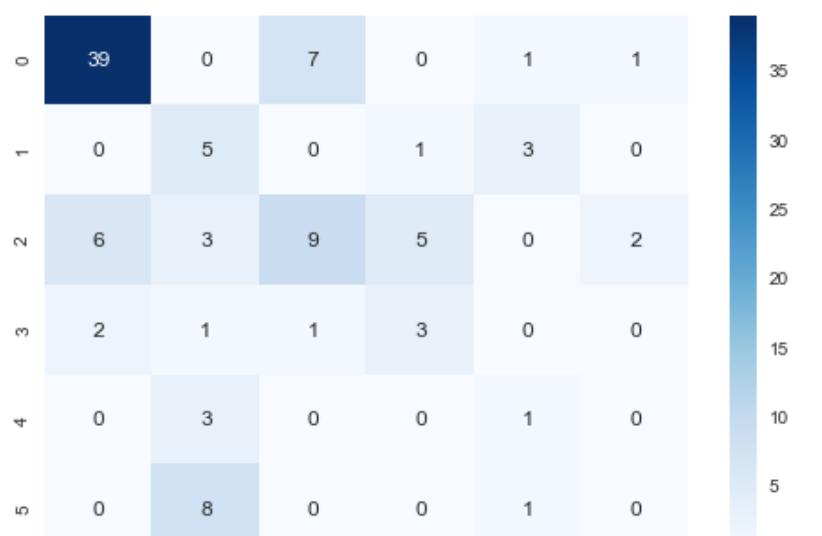
```
In [165]: 1 sns.heatmap(contingency_matrix(clustersT['T2'], clustersT['T3']), annot=True, cmap='Blues')
```

Out[165]: <AxesSubplot:>



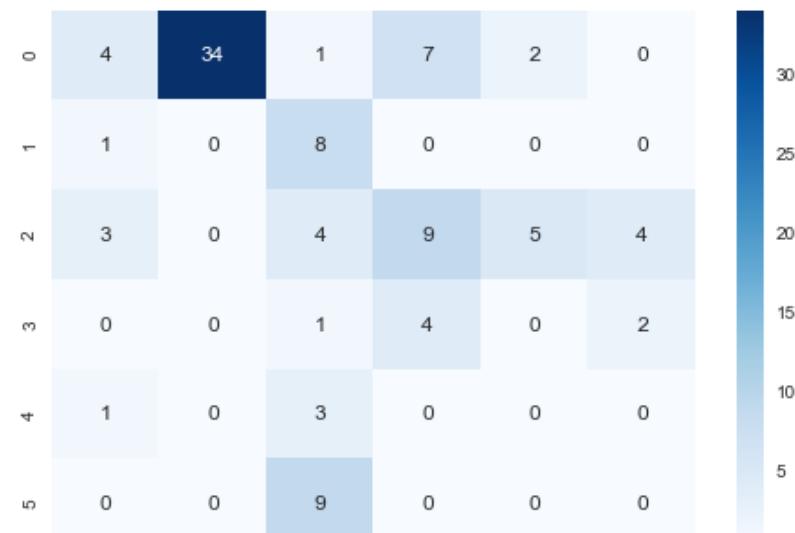
```
In [166]: 1 sns.heatmap(contingency_matrix(clustersT['T1'], clustersT['T2']), annot=True, cmap='Blues')
```

Out[166]: <AxesSubplot:>



```
In [167]: 1 sns.heatmap(contingency_matrix(clustersT['T1'], clustersT['T3']), annot=True, cmap='Blues')
```

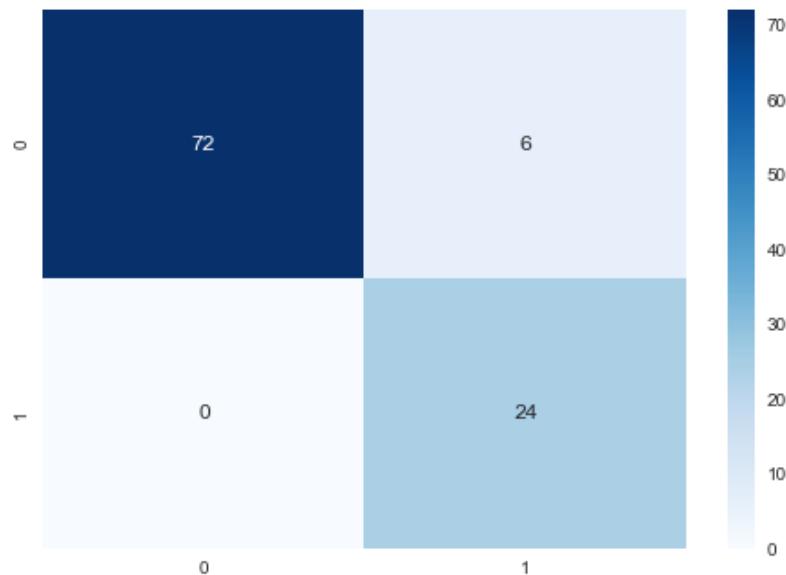
Out[167]: <AxesSubplot:>



```
In [ ]: 1
```

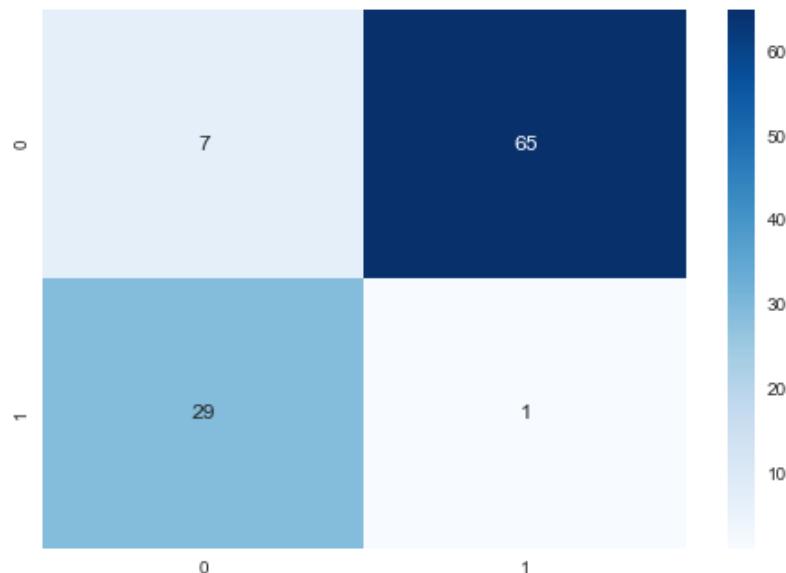
```
In [168]: 1 sns.heatmap(contingency_matrix(clustersTT['T1'], clustersTT['T2']), annot=True, cmap='Blues')
```

Out[168]: <AxesSubplot:>



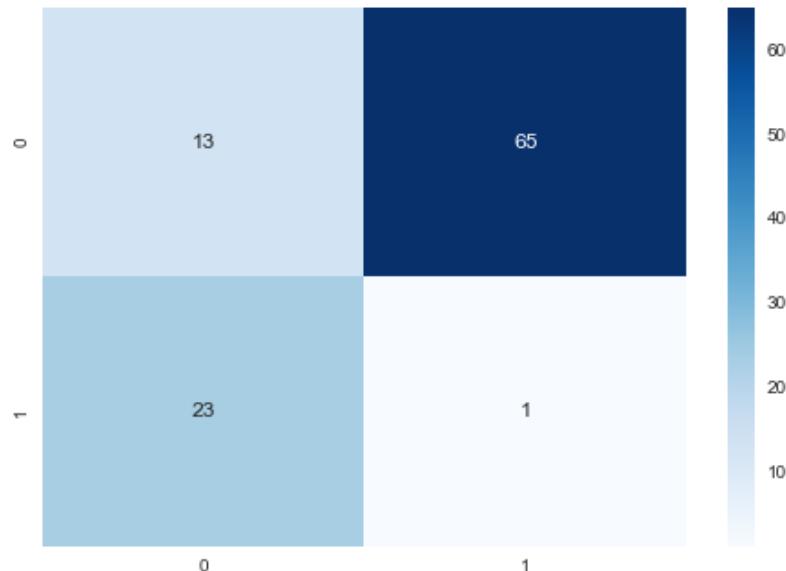
```
In [169]: 1 sns.heatmap(contingency_matrix(clustersTT['T2'], clustersTT['T3']), annot=True, cmap='Blues')
```

```
Out[169]: <AxesSubplot:>
```



```
In [170]: 1 sns.heatmap(contingency_matrix(clustersTT['T1'], clustersTT['T3']), annot=True, cmap='Blues')
```

```
Out[170]: <AxesSubplot:>
```



```
In [ ]: 1
```

```
In [171]: 1 from sklearn.metrics.cluster import adjusted_rand_score
```

```
In [172]: 1 adjusted_rand_score(clustersMC['MC'], clustersVPL['VPL'])
```

```
Out[172]: 0.573626442352794
```

```
In [173]: 1 adjusted_rand_score(clustersT['T1'], clustersT['T2'])
```

```
Out[173]: 0.41652791113497356
```

```
In [174]: 1 adjusted_rand_score(clustersT['T2'], clustersT['T3'])
```

```
Out[174]: 0.5438701342049281
```

```
In [175]: 1 adjusted_rand_score(clustersT['T1'], clustersT['T3'])
```

```
Out[175]: 0.38476411751152273
```

```
In [176]: 1 adjusted_rand_score(clustersTT['T1'], clustersTT['T2'])
```

```
Out[176]: 0.7660444202207385
```

```
In [177]: 1 adjusted_rand_score(clustersTT['T2'], clustersTT['T3'])
```

```
Out[177]: 0.7043229112279911
```

```
In [178]: 1 adjusted_rand_score(clustersTT['T1'], clustersTT['T3'])
```

```
Out[178]: 0.5113062795825317
```

```
In [179]: 1 adjusted_rand_score(clustersTTT['T1'], clustersTTT['T2'])
```

```
Out[179]: 0.42580973489845253
```

```
In [180]: 1 adjusted_rand_score(clustersTTT['T2'], clustersTTT['T3'])
```

```
Out[180]: 0.6094009866667736
```

```
In [181]: 1 adjusted_rand_score(clustersTTT['T1'], clustersTTT['T3'])
```

```
Out[181]: 0.3720311461048107
```

```
In [182]: 1 from sklearn.decomposition import PCA
```

```
In [183]: 1 from sklearn.preprocessing import StandardScaler
```

```
In [184]: 1 scaler = StandardScaler()
```

```
In [185]: 1 dataset_reduced_std = scaler.fit_transform(dataset_reduced)
```

```
In [186]: 1 pca = PCA()  
2 pca.fit(dataset_reduced_std)
```

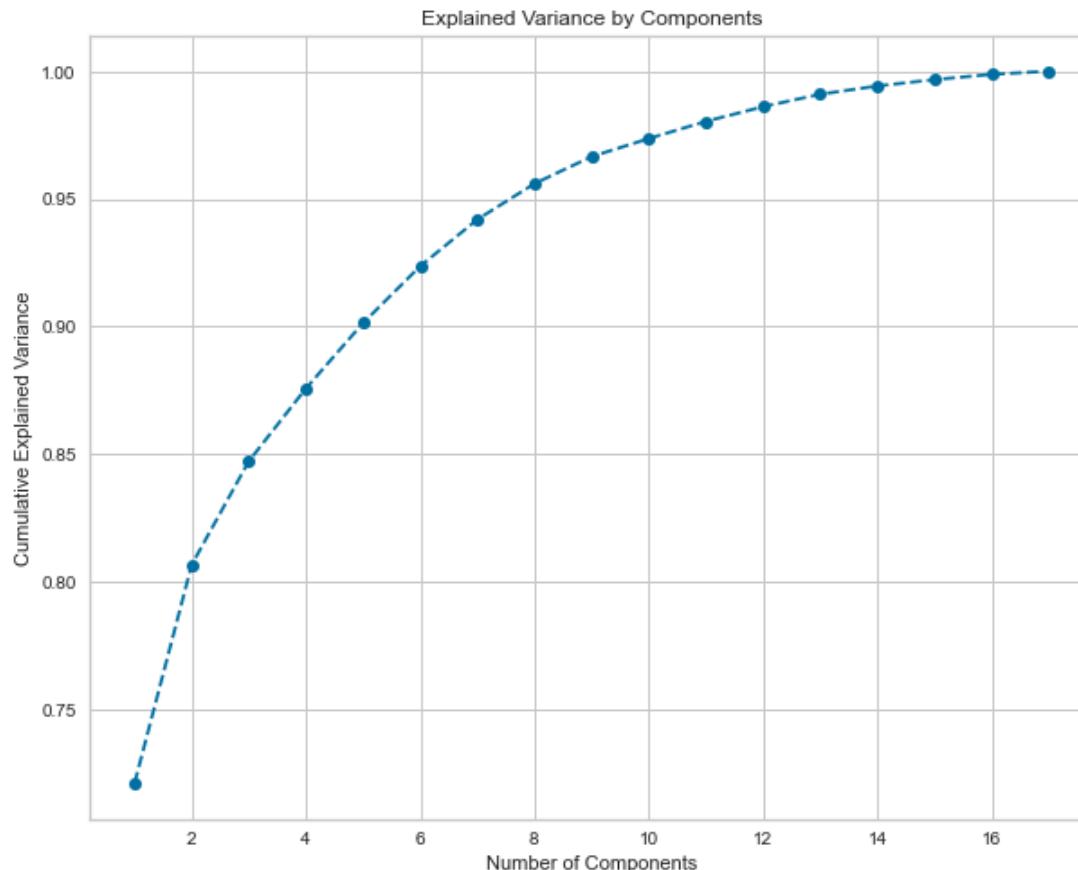
```
Out[186]: PCA(copy=True, iterated_power='auto', n_components=None, random_state=None,  
svd_solver='auto', tol=0.0, whiten=False)
```

```
In [187]: 1 pca.explained_variance_ratio_
```

```
Out[187]: array([0.7213091 , 0.08549269, 0.04057368, 0.02849862, 0.02558219,
 0.02225842, 0.01827447, 0.01402138, 0.01056267, 0.00706838,
 0.00671829, 0.00574404, 0.00488592, 0.00321943, 0.00249536,
 0.00206142, 0.00123395])
```

```
In [188]: 1 plt.figure(figsize = (10,8))
2 plt.plot(range(1,18), pca.explained_variance_ratio_.cumsum(),'bo', linestyle='--')
3 plt.title('Explained Variance by Components')
4 plt.xlabel('Number of Components')
5 plt.ylabel('Cumulative Explained Variance')
```

```
Out[188]: Text(0, 0.5, 'Cumulative Explained Variance')
```



```
In [189]:
```

```
1 pca2 = PCA()  
2 pca2.fit(dataset_reduced_min_max_scaled)  
3
```

```
Out[189]: PCA(copy=True, iterated_power='auto', n_components=None, random_state=None,  
      svd_solver='auto', tol=0.0, whiten=False)
```

```
In [190]:
```

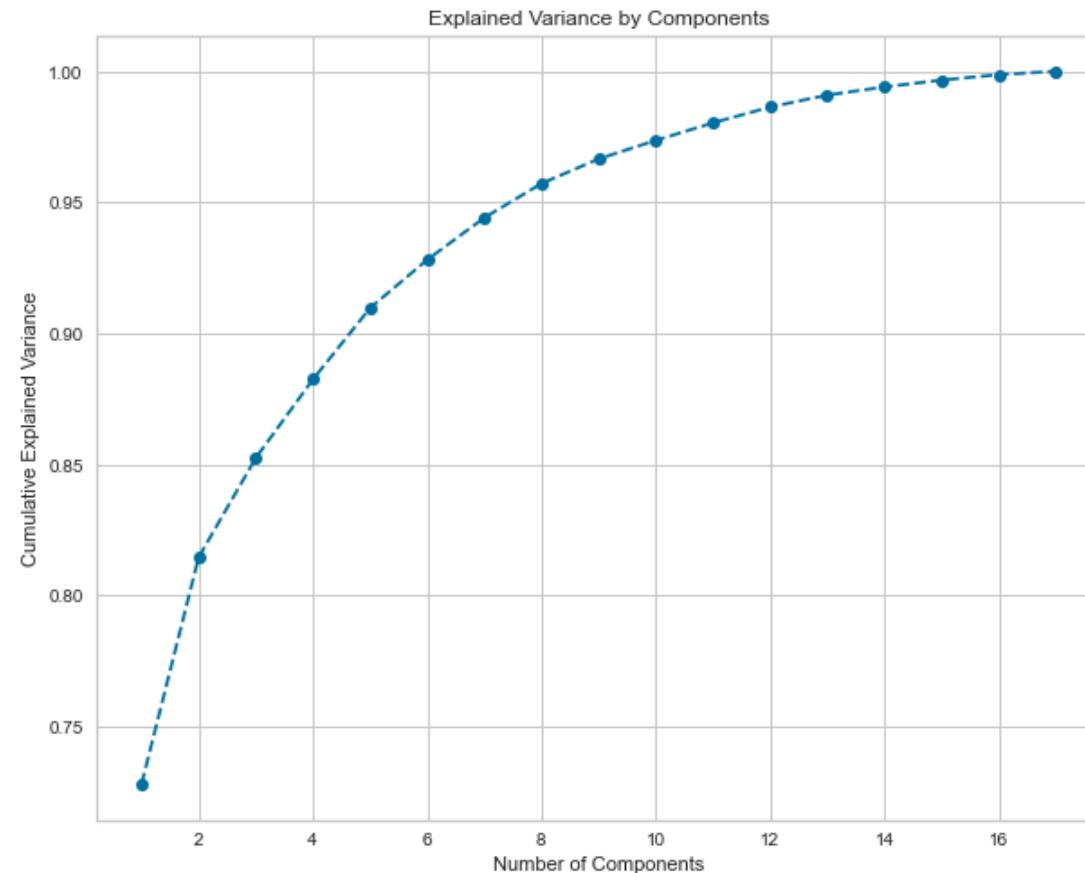
```
1 pca2.explained_variance_ratio_
```

```
Out[190]: array([0.72795641, 0.08659105, 0.03790389, 0.02998272, 0.02722763,  
      0.01842525, 0.01603878, 0.01304666, 0.00941073, 0.00704008,  
      0.0066948 , 0.00603634, 0.00454973, 0.00312783, 0.00259642,  
      0.00207652, 0.00129516])
```

In [191]:

```
1 plt.figure(figsize = (10,8))
2 plt.plot(range(1,18), pca2.explained_variance_ratio_.cumsum(),'bo', linestyle='--')
3 plt.title('Explained Variance by Components')
4 plt.xlabel('Number of Components')
5 plt.ylabel('Cumulative Explained Variance')
```

Out[191]: Text(0, 0.5, 'Cumulative Explained Variance')



```
In [192]: 1 dataset_reduced_min_max_scaled.describe().T
```

Out[192]:

	count	mean	std	min	25%	50%	75%	max
<b>intro</b>	102.0	0.859180	0.273646	0.0	0.884091	1.000000	1.000000	1.0
<b>intro-VPL</b>	102.0	0.769935	0.387327	0.0	0.666667	1.000000	1.000000	1.0
<b>methods</b>	102.0	0.771385	0.372810	0.0	0.640625	0.993750	1.000000	1.0
<b>methods-VPL</b>	102.0	0.553719	0.355638	0.0	0.170635	0.549206	0.927778	1.0
<b>encap</b>	102.0	0.724967	0.390893	0.0	0.478333	0.976667	1.000000	1.0
<b>encap-VPL</b>	102.0	0.715980	0.386421	0.0	0.400000	1.000000	1.000000	1.0
<b>constructors</b>	102.0	0.719251	0.406814	0.0	0.463636	0.959091	1.000000	1.0
<b>constructors-VPL</b>	102.0	0.682650	0.383134	0.0	0.363636	0.842424	1.000000	1.0
<b>classes1</b>	102.0	0.655025	0.427950	0.0	0.000000	0.912500	1.000000	1.0
<b>classes2</b>	102.0	0.643829	0.419726	0.0	0.191176	0.882353	0.998529	1.0
<b>classes2-VPL</b>	102.0	0.570175	0.435215	0.0	0.000000	0.684211	1.000000	1.0
<b>static</b>	102.0	0.717034	0.416454	0.0	0.500000	0.962500	1.000000	1.0
<b>static-VPL</b>	102.0	0.547967	0.404537	0.0	0.140741	0.629630	1.000000	1.0
<b>inher</b>	102.0	0.613519	0.398582	0.0	0.210526	0.850000	0.981579	1.0
<b>inher-VPL</b>	102.0	0.518301	0.410255	0.0	0.100000	0.587037	1.000000	1.0
<b>polymorphism</b>	102.0	0.644398	0.448589	0.0	0.000000	0.985714	1.000000	1.0
<b>polymorphism-VPL</b>	102.0	0.592157	0.476323	0.0	0.000000	1.000000	1.000000	1.0

```
In [193]: 1 pca3 = PCA(n_components = 6)
```

```
In [194]: 1 pca3.fit(dataset_reduced_min_max_scaled)
```

Out[194]: PCA(copy=True, iterated\_power='auto', n\_components=6, random\_state=None, svd\_solver='auto', tol=0.0, whiten=False)

```
In [195]: 1 pca3.explained_variance_ratio_
```

Out[195]: array([0.72795641, 0.08659105, 0.03790389, 0.02998272, 0.02722763, 0.01842525])

