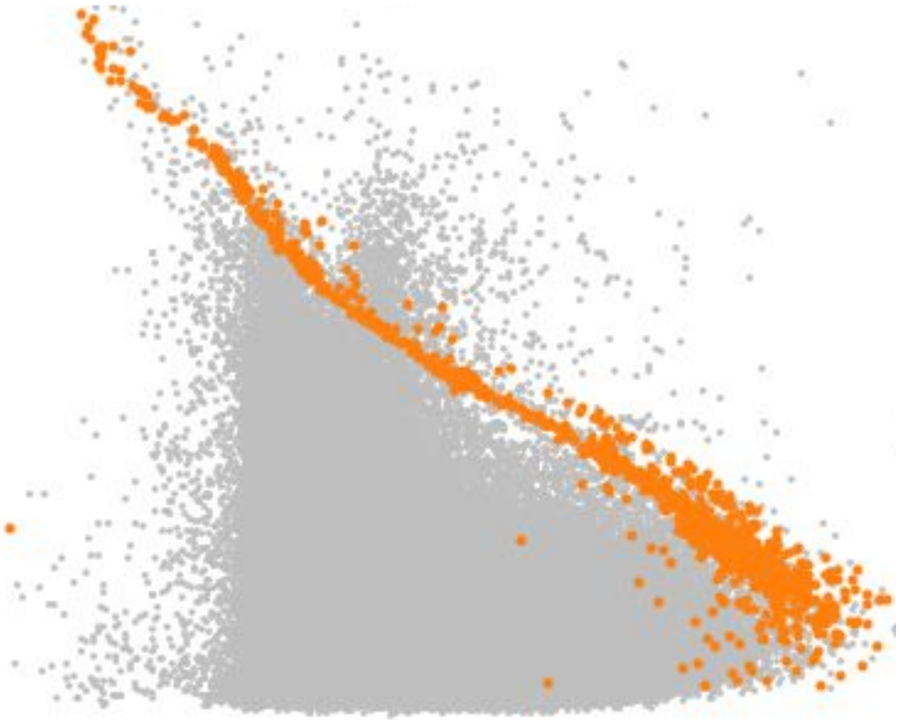
Clustering with HDBSCAN Model

Case study using "Gaia DR3" data for M45 (Pleiades) Star Cluster

Include:

- Data Preprocessing
- Building the model
- Hyperparameter tuning
- Clustering result interpretation



Data Source Overview

Source: https://www.cosmos.esa.int/web/gaia/dr3

gaiaedr3_M45.csv

```
solution_id,designation,source_id,random_index,ref_epoch,ra,ra_error,dec,dec_error,parallax,parallax_error,p
1636148068921376768,Gaia DR3 66715208773782912,66715208773782912,901684164,2016.0,56.84288916450584,3.964365
1636148068921376768,Gaia DR3 66715105696289024,66715105696289024,4198555,2016.0,56.83776002013594,0.0459730
1636148068921376768,Gaia DR3 66714452859095296,66714452859095296,18782241,2016.0,56.85167368558746,0.444679
1636148068921376768,Gaia DR3 66714487218820864,66714487218820864,1206046094,2016.0,56.86271261575877,0.4183
1636148068921376768,Gaia DR3 66714384141811712,66714384141811712,514743767,2016.0,56.85677438391798,0.055781
1636148068921376768,Gaia DR3 66715208774291968,66715208774291968,1445210981,2016.0,56.83561557768164,1.8954
1636148068921376768,Gaia DR3 66715243135238912,66715243135238912,1289519816,2016.0,56.849720188322955,0.0440
1636148068921376768,Gaia DR3 66715105694123520,66715105694123520,756531118,2016.0,56.82982016982982,0.485260
1636148068921376768,Gaia DR3 66714418501518848,66714418501518848,1704287240,2016.0,56.87239647545671,0.08812
1636148068921376768,Gaia DR3 66714384139954816,66714384139954816,1730405377,2016.0,56.862005757399984,18.511
1636148068921376768,Gaia DR3 66715174415766144,66715174415766144,1576654840,2016.0,56.82575575197103,0.03245
1636148068921376768,Gaia DR3 66714384142368256,66714384142368256,398398949,2016.0,56.87125084725649,0.468529
1636148068921376768,Gaia DR3 66715105694124928,66715105694124928,1386094566,2016.0,56.82503612242155,0.65377
1636148068921376768,Gaia DR3 66714281062596736,66714281062596736,385793905,2016.0,56.85273970848359,0.018342
1636148068921376768,Gaia DR3 66714865178148224,66714865178148224,16995418,2016.0,56.87614465716478,0.1477162
1636148068921376768,Gaia DR3 66714281062569600,66714281062569600,152744160,2016.0,56.85831297825326,0.041157
1636148068921376768,Gaia DR3 66714796458639872,66714796458639872,1545118827,2016.0,56.8790013973075,0.036180
1636148068921376768,Gaia DR3 66715174415766656,66715174415766656,1485601332,2016.0,56.82045603037588,0.09814
1636148068921376768,Gaia DR3 66714315420148224,66714315420148224,1126810326,2016.0,56.84057747543486,0.5186
1636148068921376768,Gaia DR3 66714418499341184,66714418499341184,570654621,2016.0,56.88036683835673,0.374680
1636148068921376768,Gaia DR3 66715071336552320,66715071336552320,557616627,2016.0,56.82506416684544,0.024904
1636148068921376768,Gaia DR3 66715273197982848,66715273197982848,1499585071,2016.0,56.83070036887997,0.02529
1636148068921376768,Gaia DR3 66715174413940992,66715174413940992,1374092020,2016.0,56.81753229775729,1.95926
```

Data Preparation: Loading the Dataset

Loading the dataset

I am using a cone search directed at the center of the M45 star cluster with a radius of 3 degrees. The data used is sourced from Gaia DR3.

```python
# ## Uncomment this block of script if you've done it once
# ## so you don't have to download it multiple times
# Gaia.ROW_LIMIT = -1 # Set to -1 for no row limit
# ra, dec = 56.85, 24.1167 # RA, Dec of M45
# coord = SkyCoord(ra=ra, dec=dec, unit=(u.degree, u.degree), frame='icrs')
# j = Gaia.cone_search_async(coord, radius=u.Quantity(3.0, u.deg)) # Cone search with 3 degrees radius
# gaia_result = j.get_results()
# raw_df = gaia_result.to_pandas() # Covert to pandas dataframe
# raw_df.to_csv('gaiaedr3_M45.csv', index=False)
```

✓ 0.0s Python

```python
# Load the data
FILENAME = "./gaiaedr3_M45.csv"
raw_df = pd.read_csv(FILENAME, delimiter=",")
# Select columns
columns = ["pmra", "pmdec", "parallax", "ra", "dec",
           "phot_g_mean_flux_error", "phot_g_mean_flux",
           "phot_bp_mean_flux_error", "phot_bp_mean_flux",
           "phot_rp_mean_flux_error", "phot_rp_mean_flux",
           "phot_g_mean_mag", "phot_bp_mean_mag",
           "phot_rp_mean_mag"]
raw_df = raw_df[columns]
```

✓ 4.4s Python

Data Preparation: Data Validation

Data Validation

```python
# Checking the data
print("Data shape:", raw_df.shape)
print("Data summary:")
raw_df.describe().T
```

✓ 0.2s Python

```
Data shape: (257231, 14)
Data summary:
```

	count	mean	std	min	25%	50%	75%	max
pmra	222794.0	2.869325	9.002622e+00	-293.338097	-0.275285	1.275331	4.163522	8.724843e+02
pmdec	222794.0	-4.585140	9.363686e+00	-1157.434442	-5.877124	-2.357292	-0.731895	1.969323e+02
parallax	222794.0	0.776594	1.354555e+00	-17.562348	0.178540	0.538342	1.115612	7.498865e+01
ra	257231.0	57.065388	1.632845e+00	53.564199	55.777809	57.162409	58.390606	6.013532e+01
dec	257231.0	24.272511	1.495057e+00	21.116861	23.090854	24.331559	25.485114	2.711657e+01
ot_g_mean_flux_error	256839.0	59.419778	1.043421e+04	0.565921	0.940130	1.197986	1.849912	4.226650e+06
phot_g_mean_flux	256839.0	40915.025673	3.609524e+06	23.241533	135.627003	353.201289	1481.922653	1.307657e+09
t_bp_mean_flux_error	252953.0	142.399084	4.245321e+04	0.000013	7.440292	9.383103	12.431332	2.092370e+07
phot_bp_mean_flux	252953.0	25745.099325	2.525265e+06	1.221373	60.469705	131.429209	611.449817	8.410149e+08
ot_rp_mean_flux_error	253878.0	73.742053	1.269126e+04	0.001949	8.510300	10.379328	13.372285	6.209676e+06
phot_rp_mean_flux	253878.0	24385.742214	1.585557e+06	3.199736	147.901167	351.591426	1307.829598	5.596059e+08
phot_g_mean_mag	256839.0	18.816181	1.989501e+00	2.896132	17.760303	19.317310	20.356502	2.227171e+01
phot_bp_mean_mag	252953.0	19.398061	1.999410e+00	3.026533	18.372640	20.041813	20.884697	2.512142e+01
phot_rp_mean_mag	253878.0	17.906101	1.877766e+00	2.878190	16.956518	18.382800	19.322966	2.348511e+01

Data Preparation: Data Validation (2)

```
print("Missing values:")
raw_df.isna().sum().apply(lambda x: f"{x/raw_df.shape[0]:.2%}")[list(map(lambda x: x > 0, raw_df.isna().sum()))]
```

✓ 0.0s Python

```
Missing values:

pmra                       13.39%
pmdec                      13.39%
parallax                   13.39%
phot_g_mean_flux_error      0.15%
phot_g_mean_flux            0.15%
phot_bp_mean_flux_error     1.66%
phot_bp_mean_flux           1.66%
phot_rp_mean_flux_error     1.30%
phot_rp_mean_flux           1.30%
phot_g_mean_mag             0.15%
phot_bp_mean_mag            1.66%
phot_rp_mean_mag            1.30%
dtype: object
```

```
print("Duplicated values: ", raw_df[raw_df.duplicated()].shape[0])
```

✓ 0.6s Python

```
Duplicated values:  0
```

Data Cleaning

Data Cleaning

I proceed the stars that satisfy the following criteria (Agarwal et al. 2021):

- Each source must have the five astrometric parameters, positions, proper motions, and parallax as well as valid measurements in the three photometric passbands G, GBP, and GRP in the Gaia DR3 catalogue.
- Their parallax values must be non-negative.
- To eliminate sources with high uncertainty while still retaining a fraction of sources down to G ~ 21 mag, the errors in their G-mag must be less than 0.005.

```python
# Removing NaN values in raw_df
modified_df = raw_df.dropna().reset_index()

# Validate the data
print("Data shape:", raw_df.shape)
print("Missing values:")
modified_df.isna().sum().apply(lambda x: f"{x/modified_df.shape[0]:.2%}")[
    list(map(lambda x: x > 0, modified_df.isna().sum()))]
```

✓ 0.0s Python

```
Data shape: (257231, 14)
Missing values:

Series([], dtype: object)
```

Data Cleaning (2)

```python
# Removing negative-values parallax
modified_df = modified_df[modified_df['parallax'] > 0]

# Validate the data
print("Data shape:", raw_df.shape)
```

✓ 0.0s Python

```
Data shape: (257231, 14)
```

To eliminate sources with high uncertainty while still retaining a fraction of sources down to G ~ 21 mag, we need to calculate the error of G ($|\sigma_G|$) and select the measurement with the errors less than 0.005.

$$|\sigma_G| = -\frac{2.5}{ln\,10}\frac{\sigma_{F_G}}{F_G}$$

```python
# Calculate the errors
modified_df['e_Gmag'] = abs(-2.5*modified_df['phot_g_mean_flux_error']/math.log(10)/modified_df['phot_g_mean_flu

# Filter the error
modified_df = modified_df[modified_df['e_Gmag'] < 0.005].reset_index(drop=True)

# Validate the data
print("Data shape:", raw_df.shape)
```

✓ 0.0s Python

```
Data shape: (257231, 14)
```

Feature Addition

Feature Addition

```python
# Add BP - RP color index
modified_df['bp_rp'] = modified_df['phot_bp_mean_mag'] - modified_df['phot_rp_mean_mag']
```

✓ 0.0s

```python
# Adjust the RA and Dec of the stars
coor_all  = SkyCoord(ra=modified_df.ra, dec=modified_df.dec, frame='icrs', unit=(u.deg, u.deg))

modified_df['ra']  = coor_all.ra.wrap_at(180 * u.deg).degree
modified_df['dec']  = coor_all.dec.degree
```

✓ 4.9s

Building the Model

Building HDBSCAN Model

```python
# Feature selection
X = modified_df[["pmra", "pmdec", "parallax"]]
```

✓ 0.0s

```python
# Data Scaling
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

✓ 0.0s

```python
# Building HDBSCAN model
clusterer = hdbscan.HDBSCAN() # Create a model with default hyperparameter
cluster_labels = clusterer.fit_predict(X)

modified_df['label'] = cluster_labels
```

✓ 12.3s

Model Evaluation

Model Evaluation

```
# View the clustering result
modified_df['label'].value_counts()
```

✓ 0.0s

```
-1        104327
 7          1078
2959          73
2348          71
3814          63
           ...
495            5
1922           5
1410           5
3435           5
520            5
Name: label, Length: 3883, dtype: int64
```

Noise!

Model Evaluation (2)

```python
# Remove the noise from evaluation
evaluated_df = modified_df[modified_df['label'] != -1]

# Evaluate silhouette score
score = silhouette_score(evaluated_df, evaluated_df['label'])
print(f'Silhouette Score: {score}')

# Evaluate Davies-Bouldin Index score
db_score = davies_bouldin_score(evaluated_df, evaluated_df['label'])
print(f'Davies-Bouldin Score: {db_score}')
```

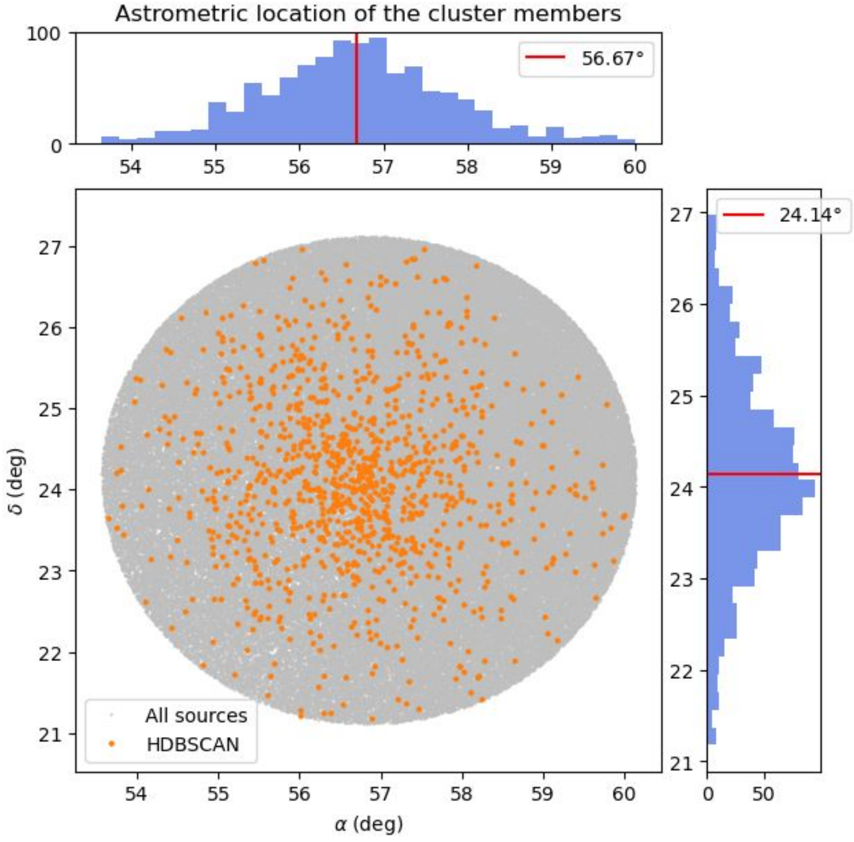
✓ 34.5s

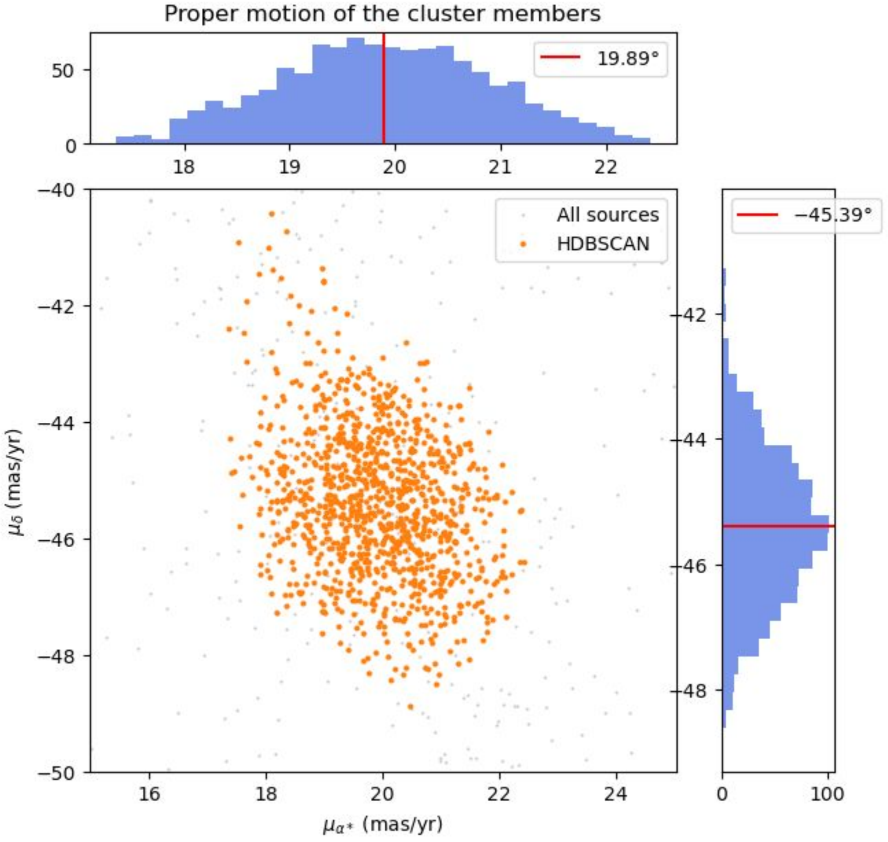
```
Silhouette Score: -0.7612576267146131
Davies-Bouldin Score: 184.33163436958793
```

Bad clustering!

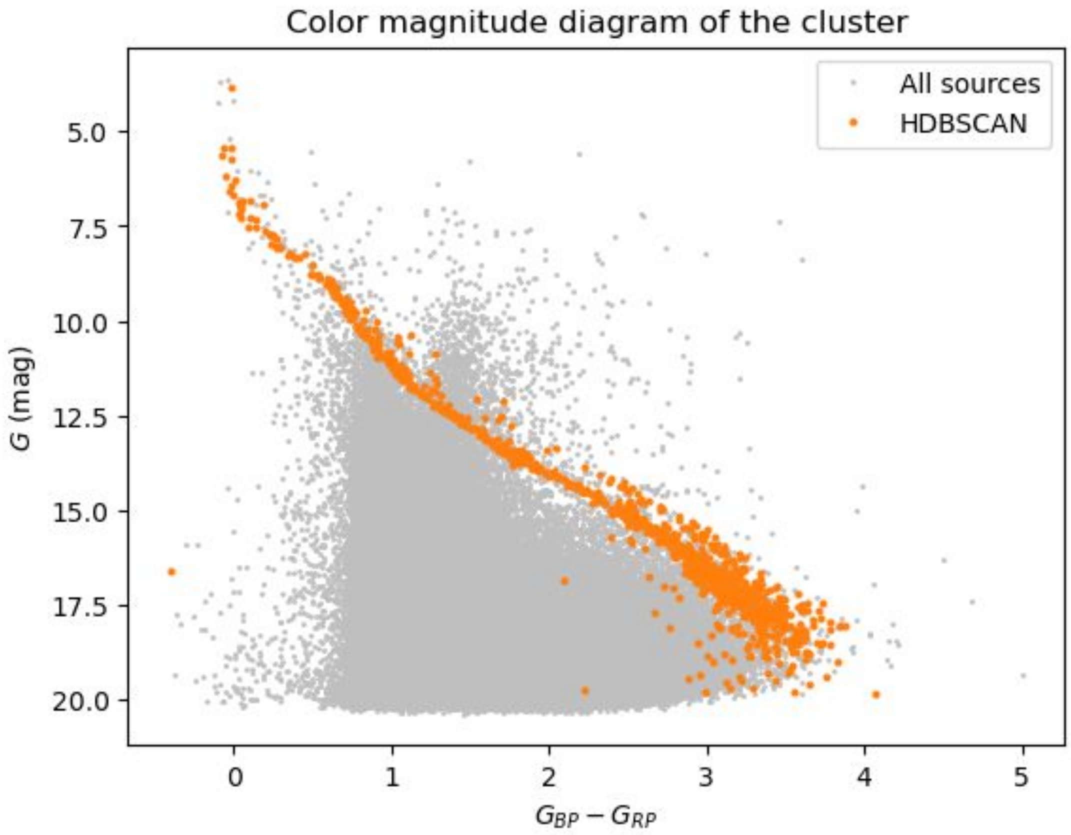
Model Evaluation (3)

Position and proper motions are distributed in "normal" fashion, indicating that the presence of a star cluster.





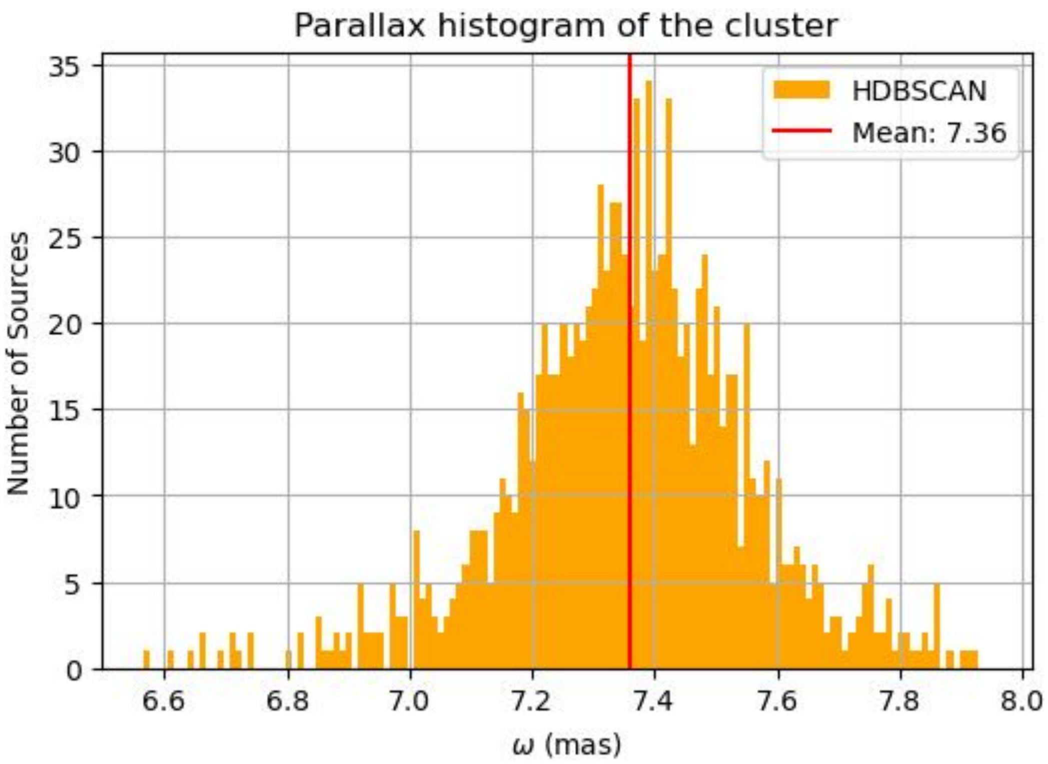
Model Evaluation (4)



Even though the clustering performance metrics indicate a bad result (not well-separated and not compact), it provides a good insight about the cluster.

We can see the clustered results “formed” an isochrone of a “star cluster” with a certain age.

Model Evaluation



The parallax (distance) also shows a well-defined peak in the distribution, suggesting a presence of a star cluster in a certain distance.

Hyperparameter Tuning

```python
# Building HDBSCAN model
clusterer = hdbscan.HDBSCAN(500) # Create a model with arbitrary hyperparameter
cluster_labels = clusterer.fit_predict(X)
```

Min cluster = 5 (Default) → 500 (New)

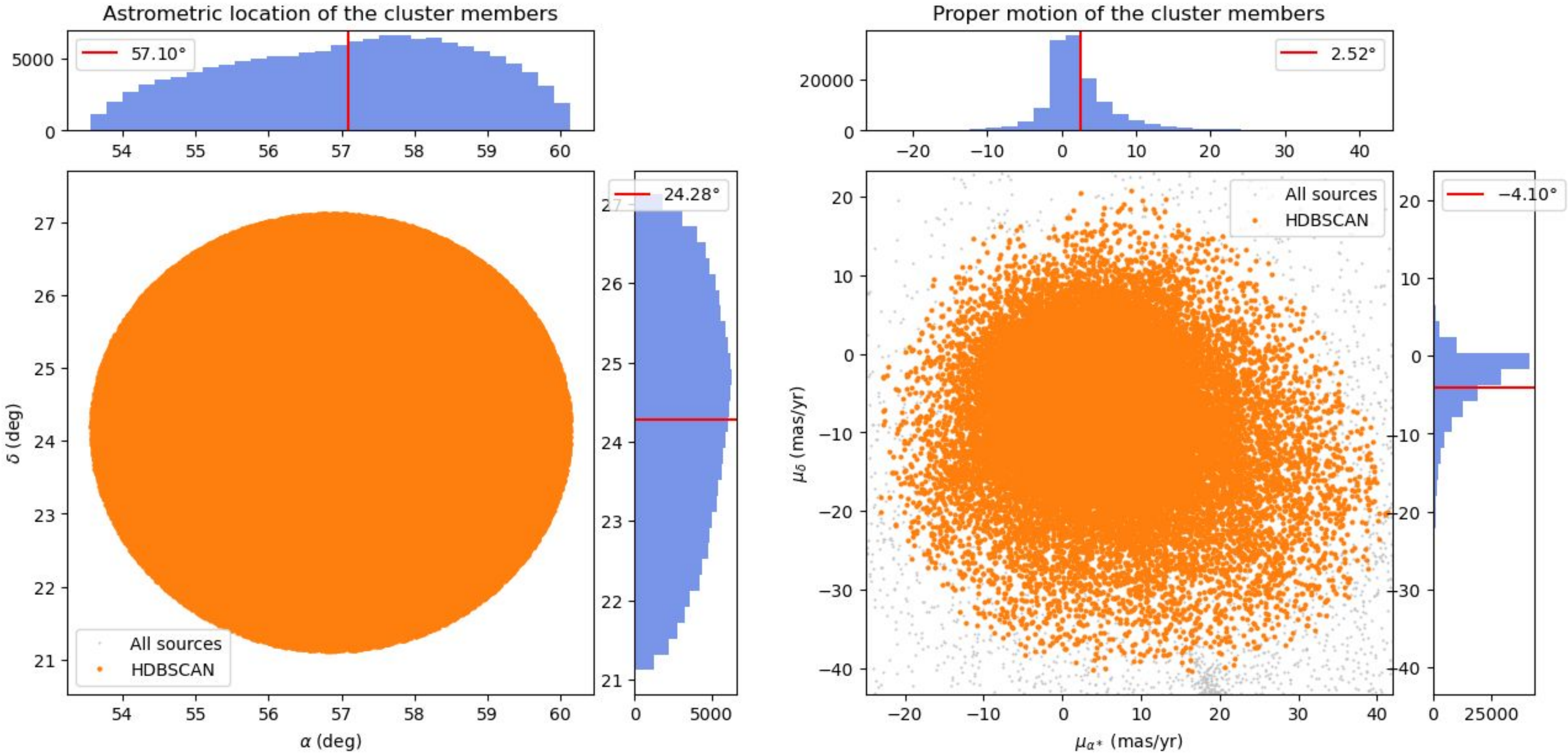
```python
# Remove the noise from evaluation
evaluated_df = modified_df[modified_df['label'] != -1]

# Evaluate silhouette score
score = silhouette_score(evaluated_df, evaluated_df['label'])
print(f'Silhouette Score: {score}')

# Evaluate Davies-Bouldin Index score
db_score = davies_bouldin_score(evaluated_df, evaluated_df['label'])
print(f'Davies-Bouldin Score: {db_score}')
```

✓ 5m 55.6s

```
Silhouette Score: 0.9545380503439873
Davies-Bouldin Score: 1.8217291942993028
```

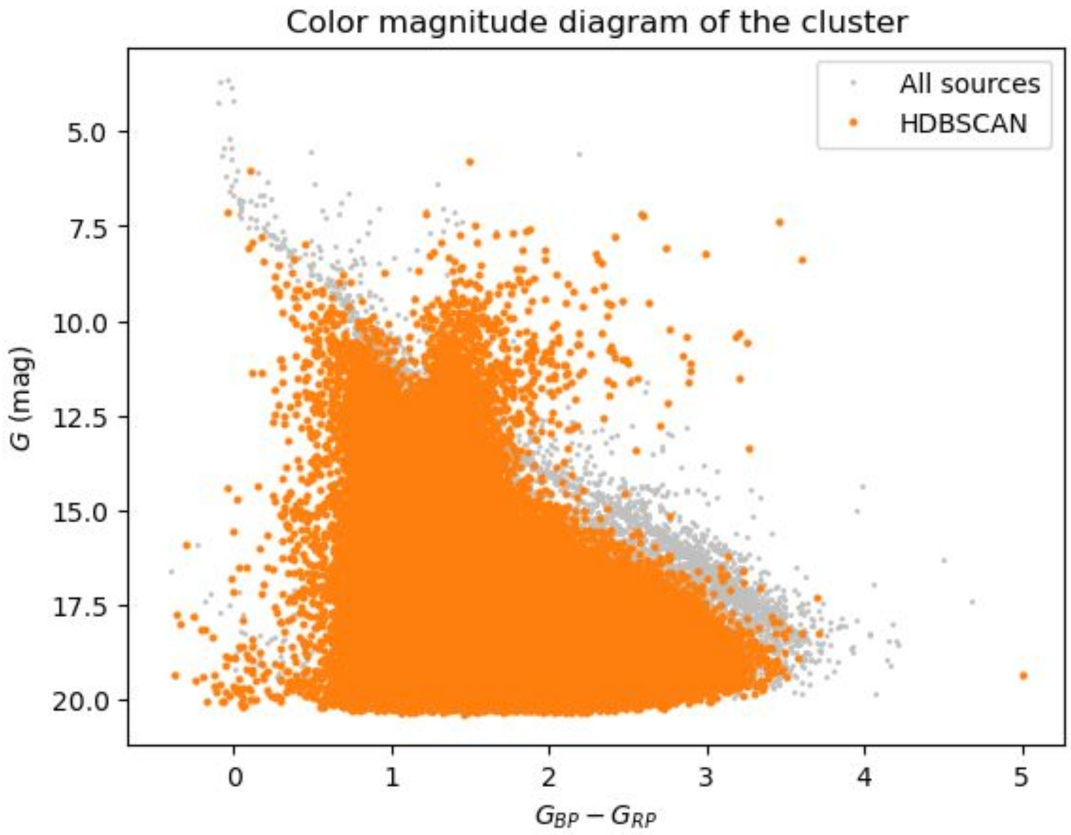
Better clustering!

Model Evaluation After Hyperparameter Tuning



Position and proper motions are “not so normally” distributed

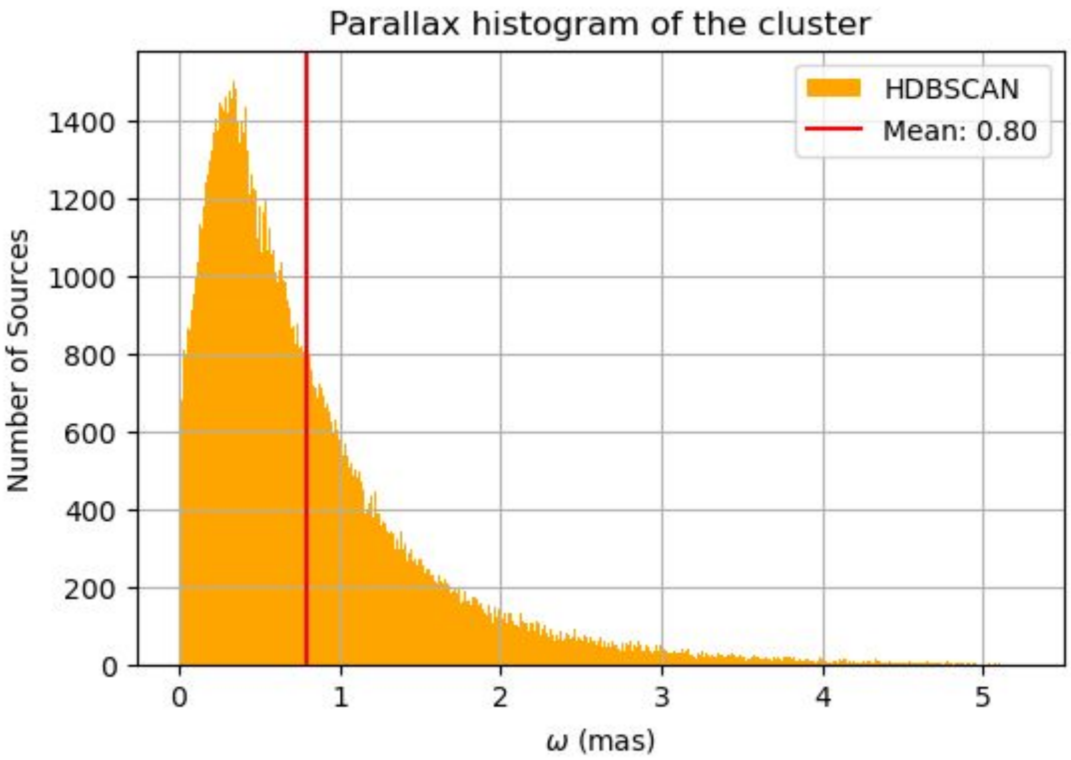
Model Evaluation After Hyperparameter Tuning (2)



Even though the clustering performance metrics indicate a good result (well-separated and compact), it can’t provides a good insight about the cluster.

We can’t see an isochrone of a “star cluster” with a certain age.

Model Evaluation After Hyperparameter Tuning (2)



The parallax of the stars varies greatly, indicating that the star cluster has not been detected.

I use the default hyperparameter for the end result.

Conclusion

Despite the clustering performance indicating poor results (silhouette score of -0.76 and Davies-Bouldin score of 184), the outcome provides valuable insights of physical properties of the star cluster. Additionally, the results are not significantly different from known cluster attributes.

- ra_c = 56.672 degree
- dec_c = 24.140 degree
- pmra_c = 19.892 mas/year
- pmdec_c = -45.390 mas/year
- parallax_c = 7.360 mas
- dist_c = 0.136 kpc

Suggestion: try different clustering methods, refine preprocessing steps, or explore more advanced techniques to improve the clustering performance while still capturing some meaningful astronomical information.

Pleiades

A color-composite image of the Pleiades from the Digitized Sky Survey

Observation data (J2000 epoch)	
Right ascension	$03^h\ 47^m\ 24^s$[1]
Declination	+24° 07′ 00″[1]
Distance	444 ly on average[2][3][4][5] (136.2±1.2 pc)