

by НТІ

EUROPEAN INVESTMENT MANAGEMENT

EXPLORATORY AND MODELLING
ANALYSIS

KASDD 2023

Our GOAT Team



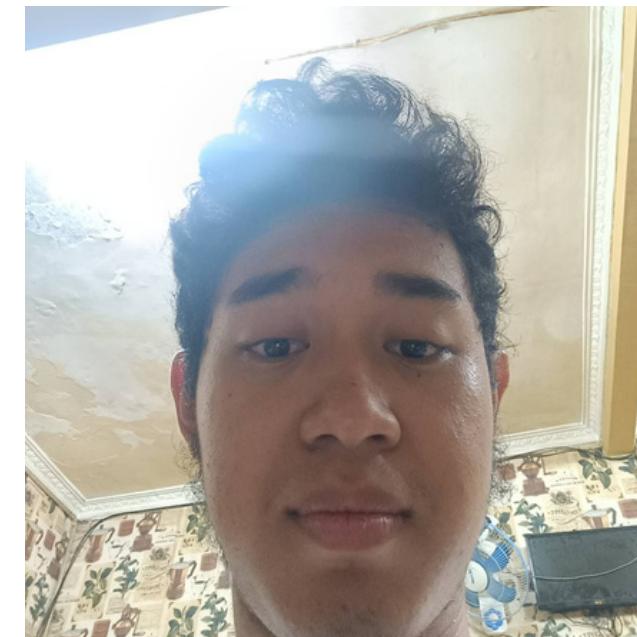
**Kristo Jeremy
Thady Tobing**

2106633310



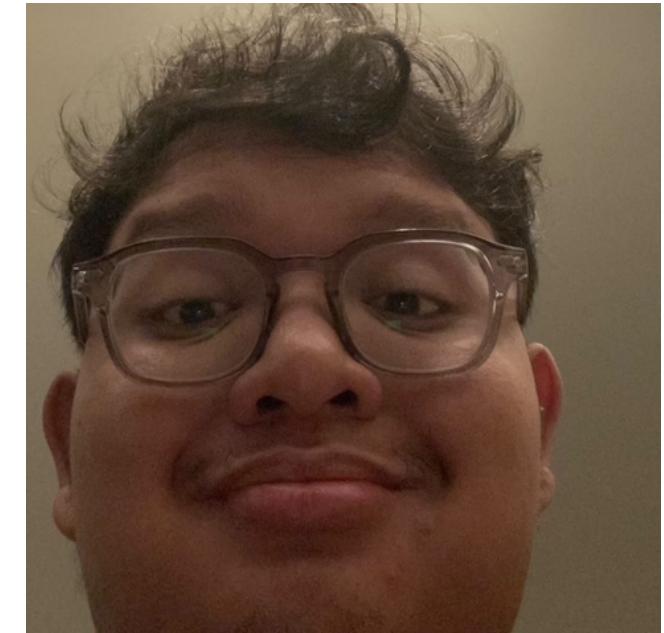
**Muhammad
Reyvan Natachnoury**

2106654353



Rafli Wasis Anggito

2106751442



**Michael Baptiswa
Marully Pangaribuan**

2106752054

Apa itu European Investment Management ?

European Investment Management (EIM) adalah kegiatan pengelolaan investasi di wilayah Eropa yang mencakup strategi investasi, seperti pengelolaan dana, portofolio, dan aset, dengan fokus pada pasar keuangan dan investasi di Eropa.

Tujuan utama

Mengoptimalkan pengembalian investasi dan mengelola risiko untuk klien atau entitas yang menginvestasikan dana mereka melalui berbagai instrumen keuangan yang tersedia di pasar Eropa.



Ringkasan Dataset

Dataset ini tentang kategori investasi beserta dengan atribut berupa nilai equity, cash flow, sales growth, fund return, dsb.

Jumlah Baris

22420

Jumlah Kolom

117

Numerikal

107

Kategorikal

10

Tahapan Analisis

**Data
Preprocessing**

**Exploratory
Analysis**

**Predictive
Modelling**

Data Preprocessing

Missing Values

Melakukan handling terhadap value yang bernilai NaN atau Null.

Duplicates Values

Melakukan handling terhadap instance data yang memiliki value yang sama dengan instance data lainnya

Outliers

Melakukan handling terhadap outliers dalam persebaran data

MISSING VALUES

1 Checking Missing Values

```
for i in eim_df.columns:  
    percentage_missing = eim_df[i].isnull().sum() / len(eim_df) * 100  
    print(f"{i} : {round(percentage_missing, 2)} %")
```

```
ticker : 0.0 %  
category : 0.0 %  
dividend_frequency : 53.46 %  
equity_style : 0.0 %  
equity_size : 0.0 %  
equity_size_score : 0.0 %  
price_prospective_earnings : 0.03 %  
price_book_ratio : 0.02 %  
price_sales_ratio : 0.04 %  
price_cash_flow_ratio : 0.25 %  
dividend_yield_factor : 0.0 %
```

2 Checking Duplicate Values

```
[8] print("Rows duplicated : " + str(eim_df.duplicated().sum()))  
  
Rows duplicated : 0
```

3 Handling Missing Values

Drop Columns dan Rows

- Drop kolom dengan **missing value lebih dari 30%**
- Drop baris dengan **null features lebih dari 20**

Impute remaining missing values

- Imputasi data NaN dengan **median** untuk kolom numerik
- Imputasi data NaN dengan **modus** untuk kolom kategorikal

```
ticker : 0.0 %
category : 0.0 %
equity_style : 0.0 %
equity_size : 0.0 %
equity_size_score : 0.0 %
price_prospective_earnings : 0.0 %
price_book_ratio : 0.0 %
price_sales_ratio : 0.0 %
price_cash_flow_ratio : 0.0 %
dividend_yield_factor : 0.0 %
```

4

Checking Outliers

	Total Outliers	Percentage Outliers
asset_bond	5104	24.02
asset_stock	4389	20.66
involvement_palm_oil	4209	19.81
asset_other	3652	17.19
fund_return_2015	3545	16.69
involvement_gmo	3357	15.80

Outlier tidak dihandle karena :

- Nilai-nilai ekstrem bisa saja sejalan dengan sifat real data. Dengan menghapusnya akan bisa merusak representasi data yang akurat.

Exploratory Analysis

**5 Besar Investment
Management berdasarkan
Dana Kelolaan**

**Perbedaan antar
equity_style**

**Top 5 Investment
management berdasarkan
equity_size_score**

**Perbandingan untuk tiap
sektor**

**Hubungan antara
maangement_fees dengan
pertumbuhan return
investasi**

**Hubungan antara Fund Size
dan Trailing return year to
date**

1

5 Besar Investment Management berdasarkan Dana Kelolaan

Exploratory

Grouping data

Melakukan grouping berdasarkan kategori management

```
▶ group_category = eim_df.groupby(['category']).mean().reset_index()
```

Sorting data

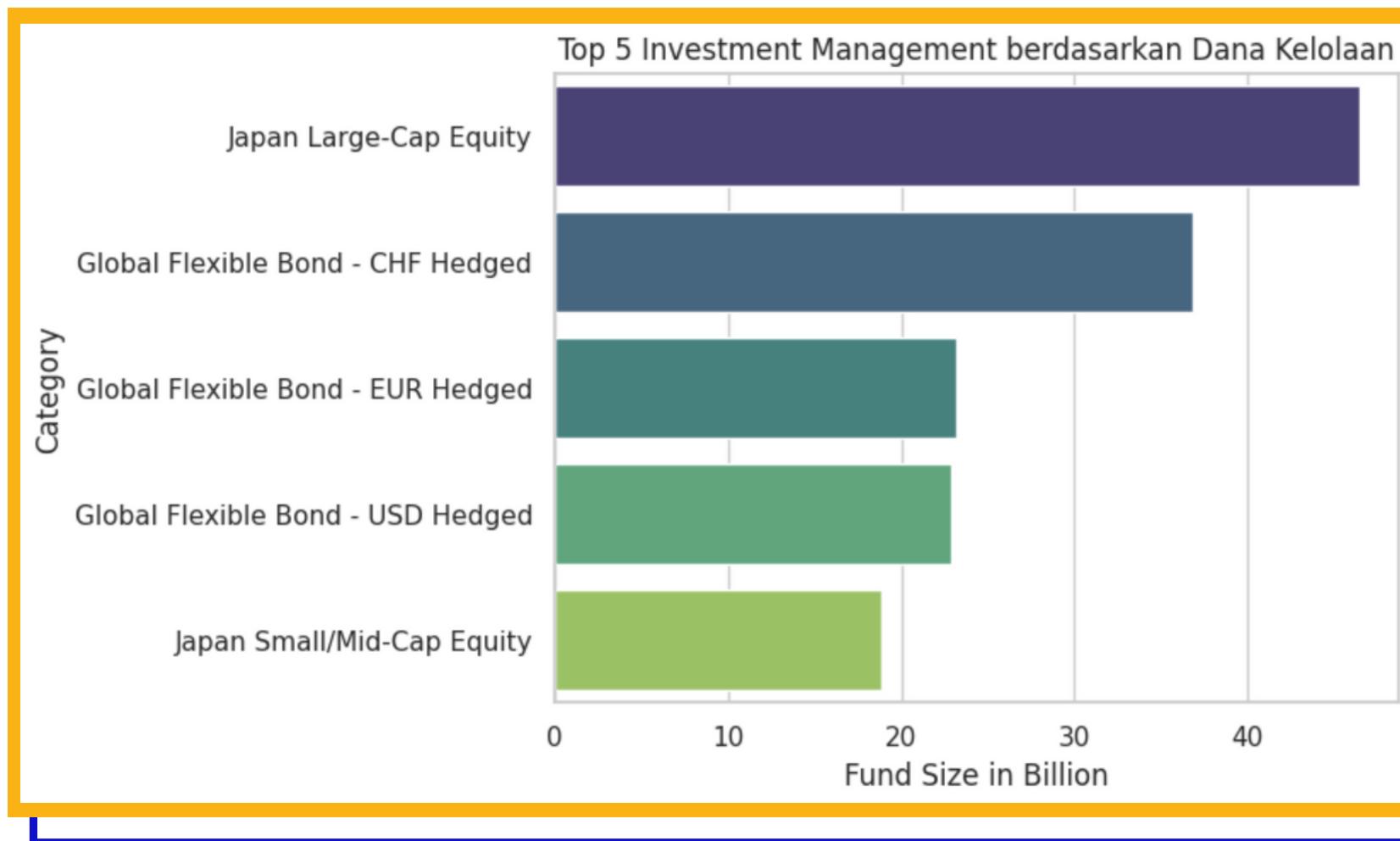
Melakukan sorting kategori management bedasarkan fund size terbesar

```
▶ # Mengurutkan data berdasarkan rata-rata dana kelolaan secara descending
sorted_category = group_category.sort_values(by='fund_size', ascending=False).reset_index()
sorted_category = sorted_category.drop(columns=['index'])
top_5_fund_size = sorted_category[['category', 'fund_size']].head(5)
top_5_fund_size['fund_size'] = top_5_fund_size['fund_size'] / 1e9
top_5_fund_size.round(2)
```

1

5 Besar Investment Management berdasarkan Dana Kelolaan

Exploratory



- **Dominasi Japan Large-Cap:** Japan Large-Cap Equity memimpin dengan aset sebesar \$46.46 miliar USD
- **Persaingan Ketat di Kategori Obligasi Global:** Kategori Global Flexible Bond - EUR Hedged dan Global Flexible Bond - USD Hedged bersaing ketat dengan masing-masing aset sekitar \$23.19 miliar USD dan \$22.91 miliar USD.

②

Tunjukkan perbandingan untuk tiap sektor

Exploratory

Grouping data

Melakukan grouping berdasarkan kategori sektor

```
[ ] ## Group data by each sector

aggregations = {
    col: 'mean' for col in numeric_cols
}

aggregations.update({
    col: lambda x: x.mode().iloc[0] for col in categorical_columns
})

filtered_df = eim_df[eim_df['category'].str.startswith("Sector")]

sector_grouped = filtered_df.groupby('category').agg(
    aggregations
)

sector_grouped.drop(columns=['category', 'ticker'])
```

②

Tunjukkan perbandingan untuk tiap sektor

Exploratory

Select fitur

Memilih fitur penting yang tepat untuk dibandingkan tiap sektor, yaitu equity_size_score, sales_growth, fund_trailing_return_5years

```
selected_columns = ['category','equity_size_score', 'sales_growth',
                     'fund_trailing_return_5years']

selected_data = sector_grouped[selected_columns]

selected_data
```

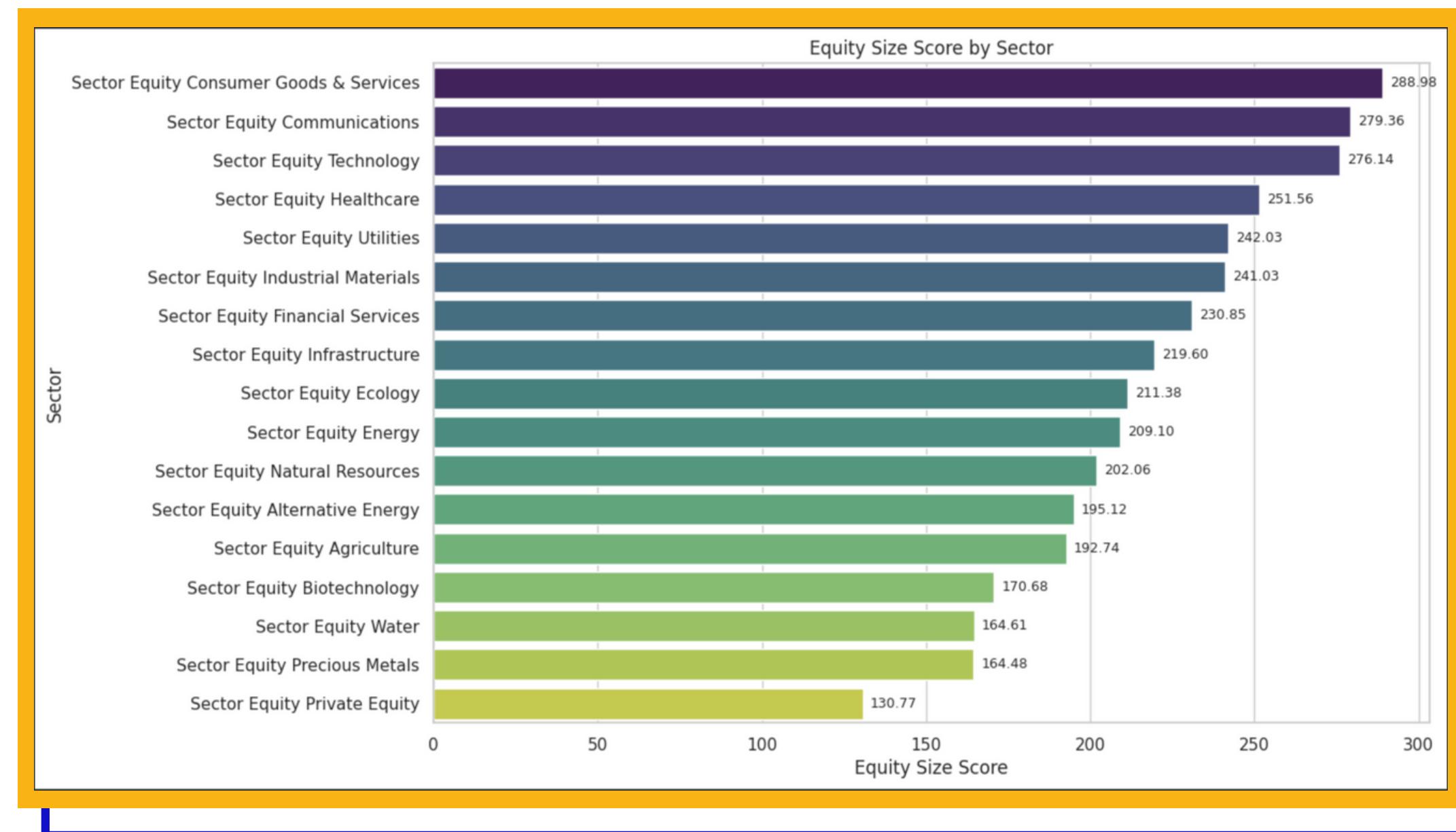
- **equity_size_score:** Dapat memengaruhi tingkat risiko dan potensi pengembalian investasi.
- **sales_growth:** Menunjukkan seberapa cepat penjualan perusahaan berkembang.
- **fund_trailing_return_5years:** Menunjukkan informasi tentang kinerja dana investasi selama 5 tahun terakhir. Ini dapat menjadi indikator konsistensi dan hasil historis, yang penting untuk pertimbangan investor.

②

Tunjukkan perbandingan untuk tiap sektor

Exploratory

Berdasarkan Equity Size Score



Berdasarkan Equity Size Score:

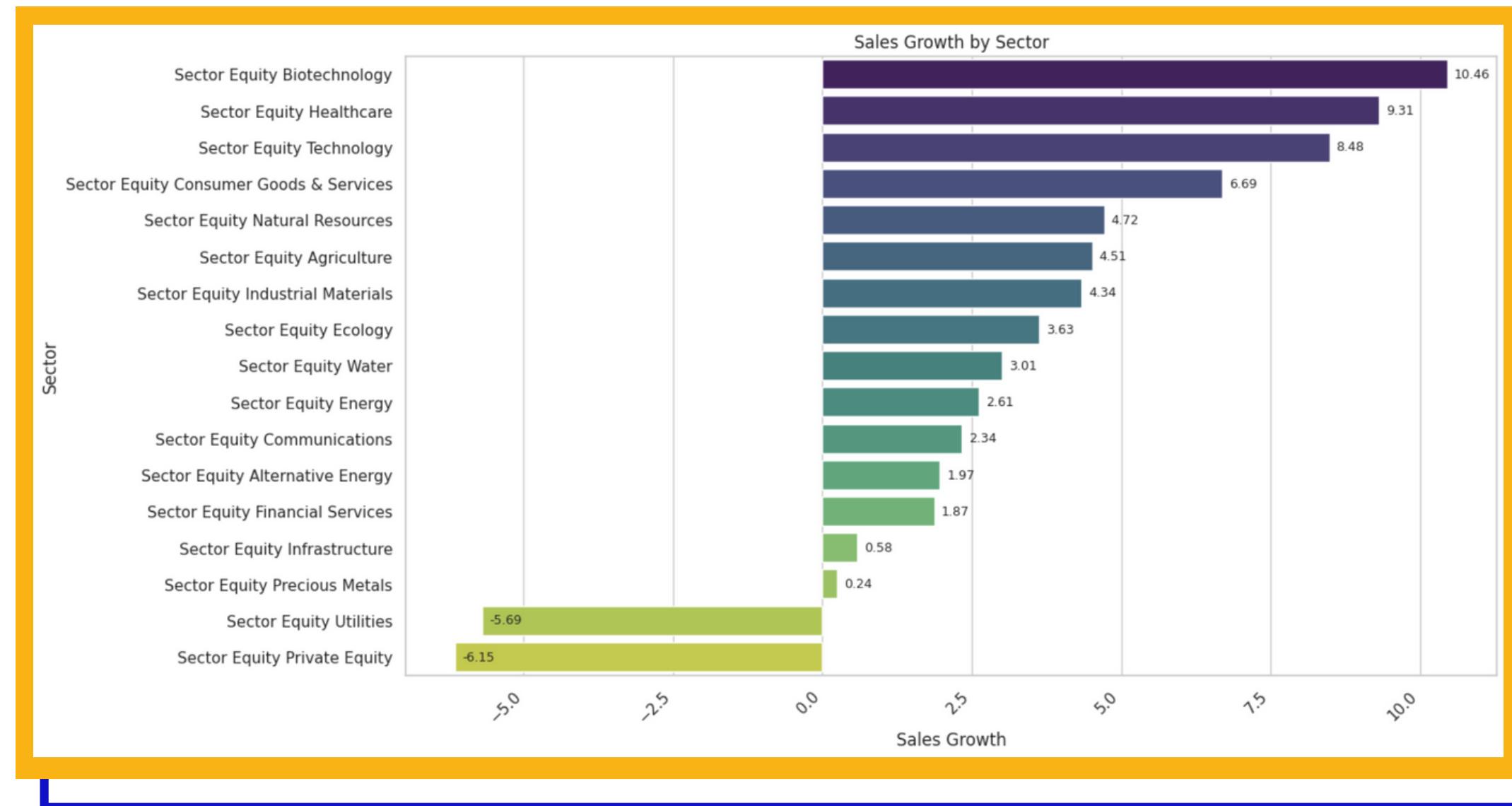
- Sector terbaik adalah **Consumer Goods & Services** dengan score 288.98.
- Sector terburuk adalah **Private Equity** dengan score 130.77.

②

Tunjukkan perbandingan untuk tiap sektor

Exploratory

Berdasarkan Sales Growth



Berdasarkan Sales Growth:

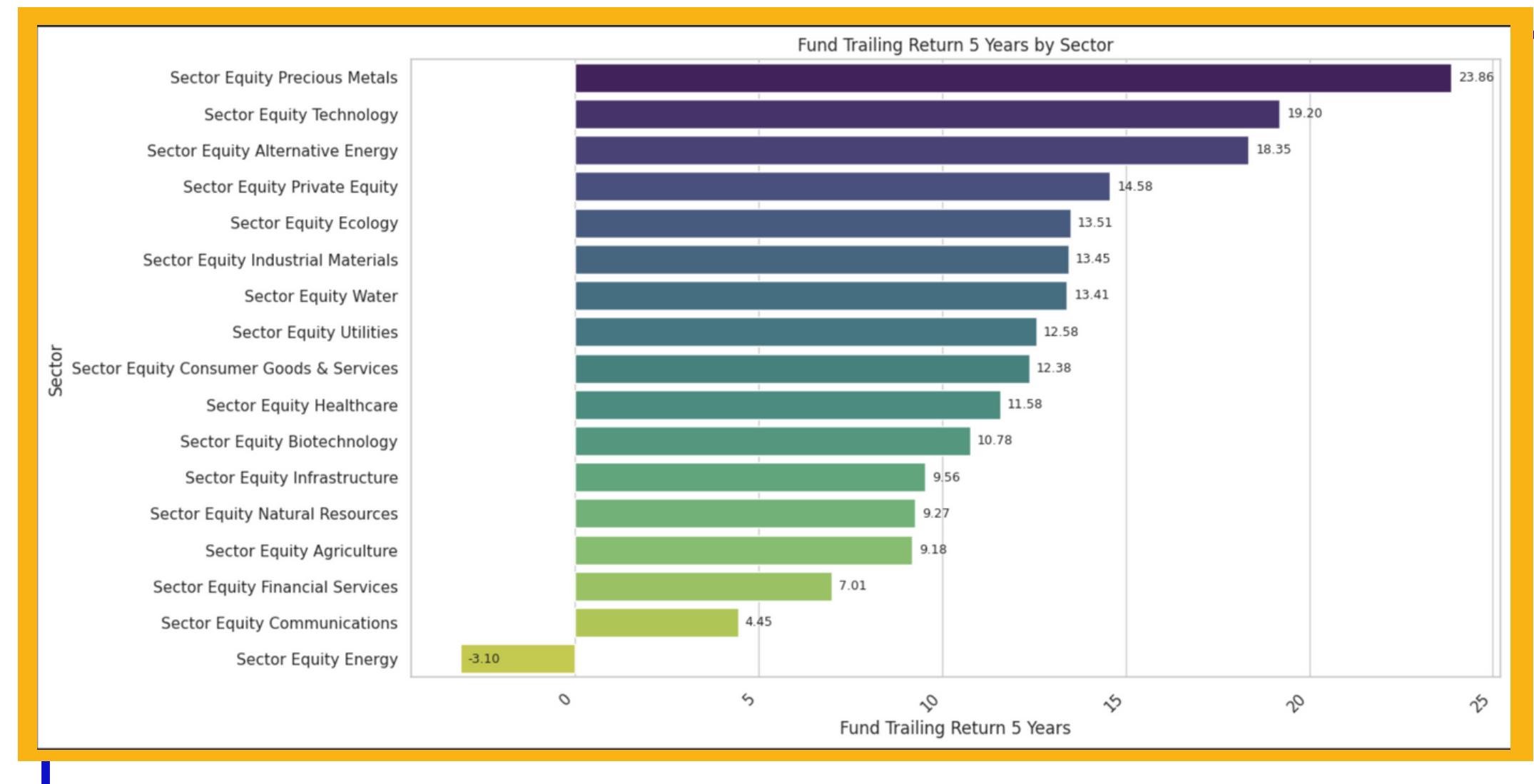
- Sector terbaik adalah **Biotechnology** dengan score 10.46.
- Sector terburuk adalah **Private Equity** dengan score -6.15.

②

Tunjukkan perbandingan untuk tiap sektor

Exploratory

Berdasarkan Fund Trailing Return 5 Years



Berdasarkan Fund Trailing Return 5 Years:

- Sector terbaik adalah **Precious Metal** dengan score 23.86.
- Sector terburuk adalah **Energy** dengan score -3.10.

3

Apakah terdapat hubungan antara management_fees dengan pertumbuhan return investasi?

Exploratory

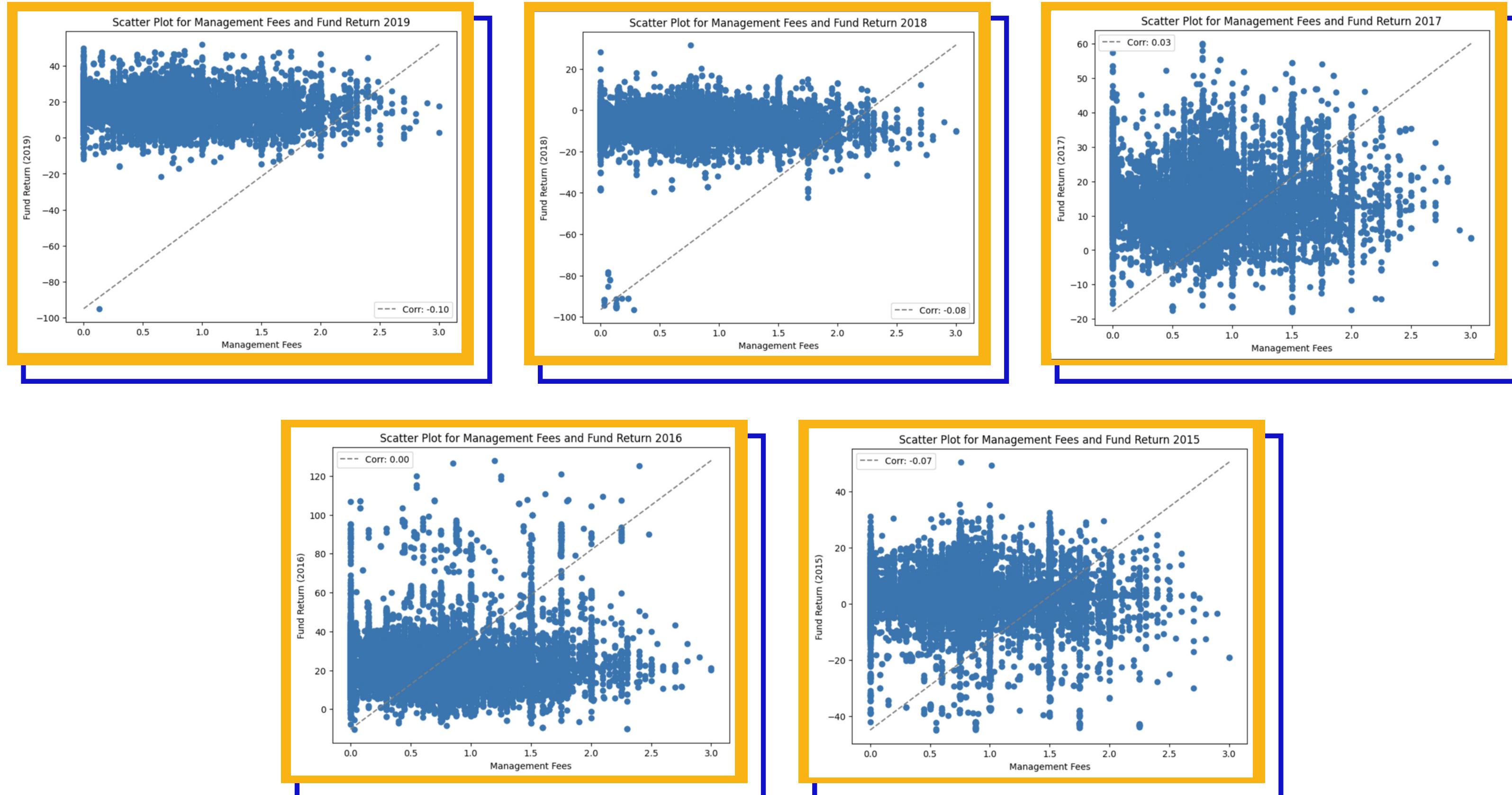
Korelasi management_fees dengan fund return masing-masing tahun

Analisis dimulai dengan melihat perbandingan per tahun-tahun secara masing-masing. Visualisasi data korelasi ditampilkan melalui scatter plot.

Korelasi management_fees dengan pertumbuhan fund return

Analisis dimulai dengan melihat perbandingan pertumbuhan dari tahun 2015-2019. Visualisasi data korelasi ditampilkan melalui scatter plot.

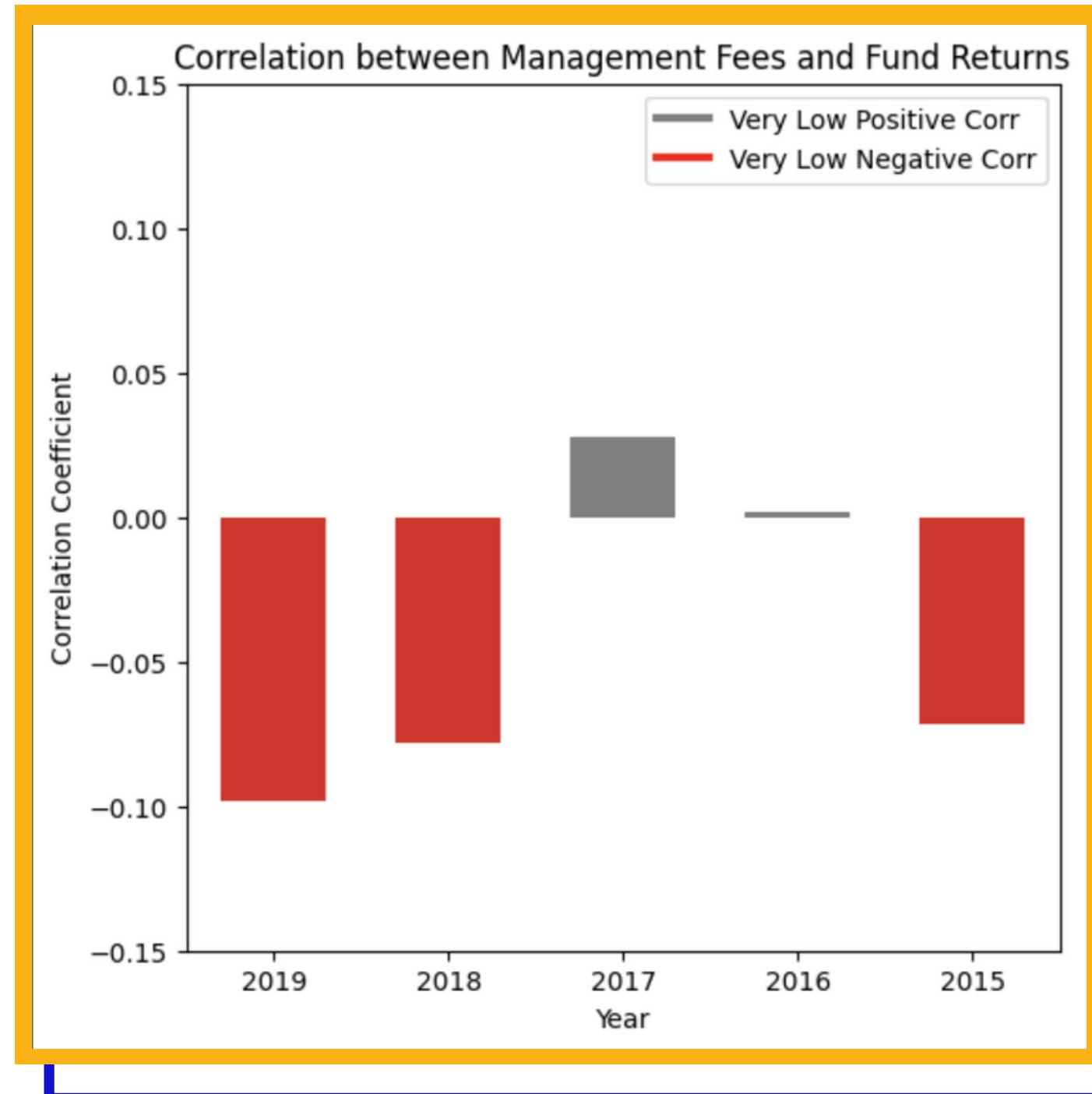
Korelasi management_fees dengan fund return masing-masing tahun



3

Apakah terdapat hubungan antara management_fees dengan pertumbuhan return investasi? (per tahun)

Exploratory



Tidak terdapat korelasi yang berarti untuk management_fees dengan fund return masing-masing tahun

Jika dibandingkan berdasarkan masing-masing tahun, maka korelasi yang dihasilkan adalah very low positive correlation dan very low negative correlation.

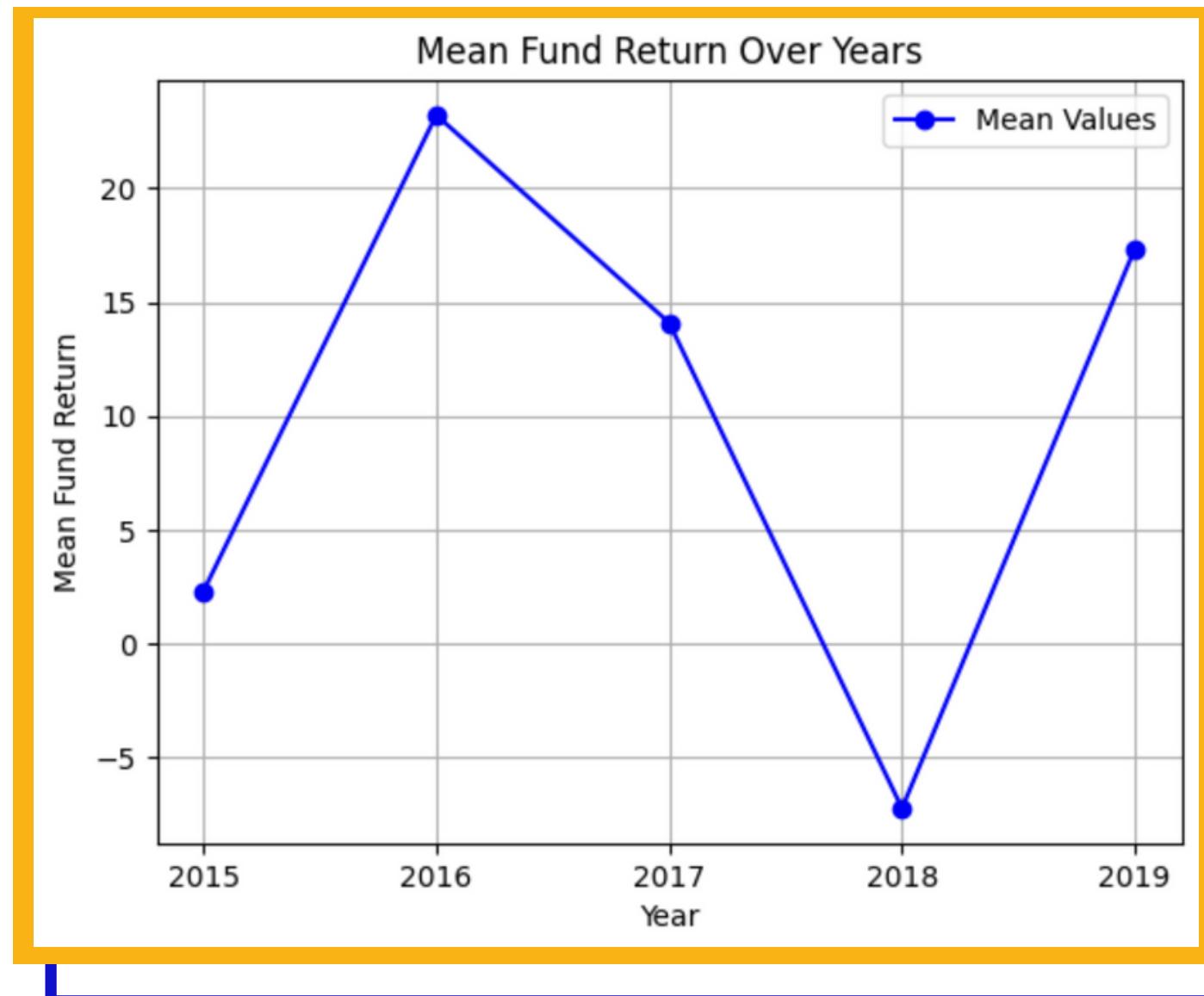
Maka dari itu, dapat disimpulkan bahwa management_fees **tidak berhubungan** terhadap fund return untuk masing-masing tahun

3

Apakah terdapat hubungan antara management_fees dengan pertumbuhan return investasi? (pertumbuhan)

Exploratory

Tren Pertumbuhan fund return 2015-2019



Pertumbuhan fund return memiliki tren tidak menentu: tren naik pada tahun 2016 kemudian berangsung-angsur turun pada tahun 2017 dan 2018 lalu kembali naik pada tahun 2019

3

Apakah terdapat hubungan antara management_fees dengan pertumbuhan return investasi? (pertumbuhan)

Exploratory

Mengkalkulasi variable pertumbuhan fund return

Variabel baru ini digunakan sebagai target korelasi baru. Variabel didapatkan dari jumlah selisih tahun(x) dengan tahun (x-1)

```
difference_2016_2015 = eim_df['fund_return_2016'] - eim_df['fund_return_2015']
difference_2017_2016 = eim_df['fund_return_2017'] - eim_df['fund_return_2016']
difference_2018_2017 = eim_df['fund_return_2018'] - eim_df['fund_return_2017']
difference_2019_2018 = eim_df['fund_return_2019'] - eim_df['fund_return_2018']

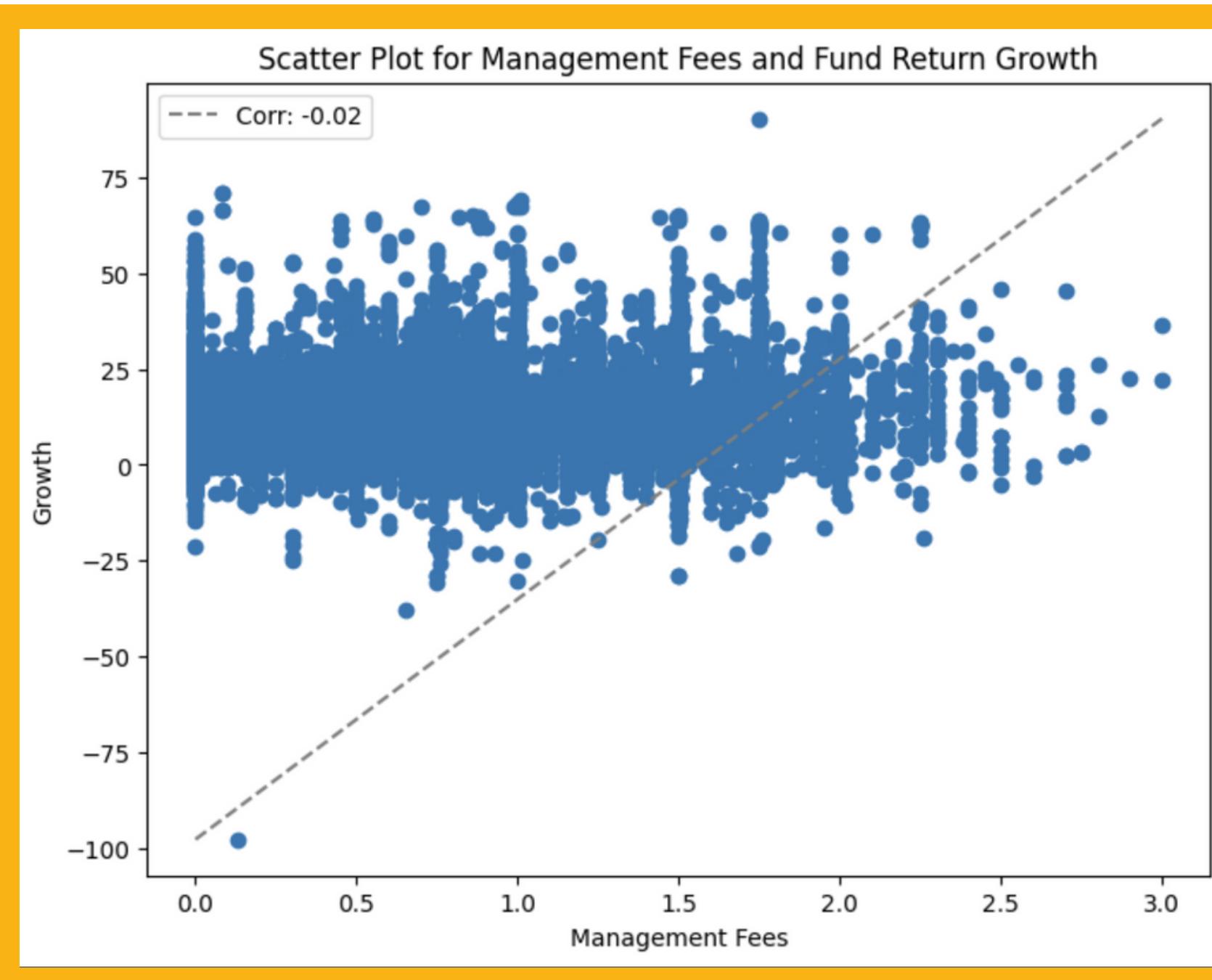
fund_return_trend = difference_2016_2015 + difference_2017_2016 + difference_2018_2017 + difference_2019_2018
fund_return_trend

man_fees_ret_invs_trend = pd.DataFrame()
man_fees_ret_invs_trend['fund_trend_ovr_year'] = fund_return_trend
man_fees_ret_invs_trend['management_fees'] = man_fees_ret_invs['management_fees']
```

3

Apakah terdapat hubungan antara management_fees dengan pertumbuhan return investasi? (pertumbuhan)

Exploratory



Tidak terdapat korelasi yang berarti untuk management_fees dengan pertumbuhan fund return

corr = -0,02 (**very low negative correlation**)

Maka dari itu, dapat disimpulkan bahwa management_fees **tidak berhubungan** terhadap pertumbuhan fund return.

4

Perbedaan antar equity_style

Exploratory

Grouping by equity_style

```
equity_style_grouped = eim_encoded.groupby('equity_style')

grouped = eim_encoded.groupby('equity_style')
```

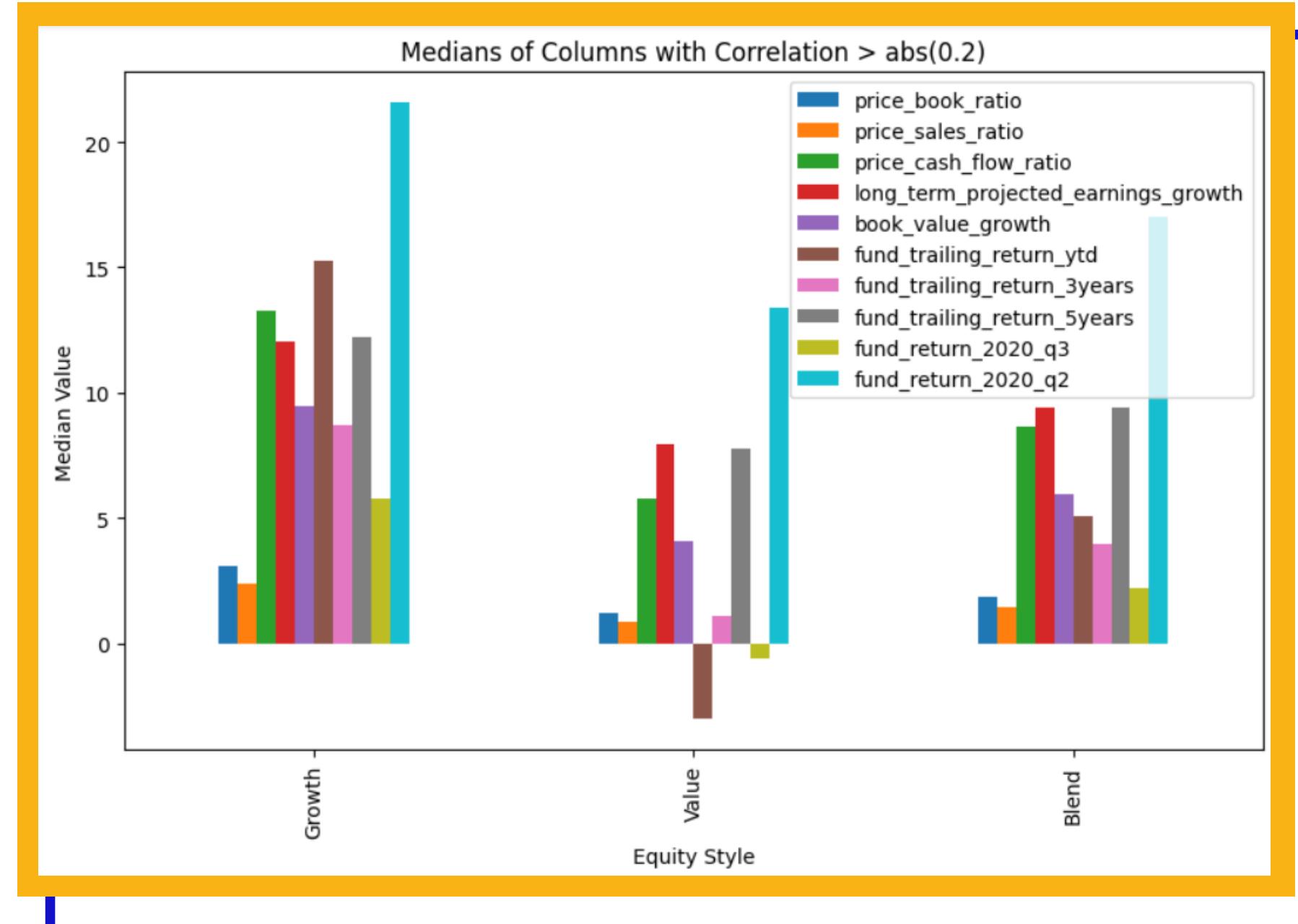
Menyeleksi fitur yang memiliki korelasi > 0,2

```
corr_matrix = eim_encoded.corr()

# Extract correlations involving equity_style column
equity_style_corr = corr_matrix["equity_style"]

# Filter the columns with correlation higher than abs(0.2)
corr_columns = equity_style_corr[abs(equity_style_corr) > 0.2].index.tolist()
```

Perbedaan antar equity_style



Performa terbaik: growth

Dari semua equity style yang ada, tipe equity style Growth memiliki rata - rata tertinggi dibandingkan dengan Value dan Blend

5

Top 5 Investment management berdasarkan equity_size_score

Exploratory

Grouping data

Melakukan grouping berdasarkan kategori management

```
▶ group_category = eim_df.groupby(['category']).mean().reset_index()
```

Sorting data

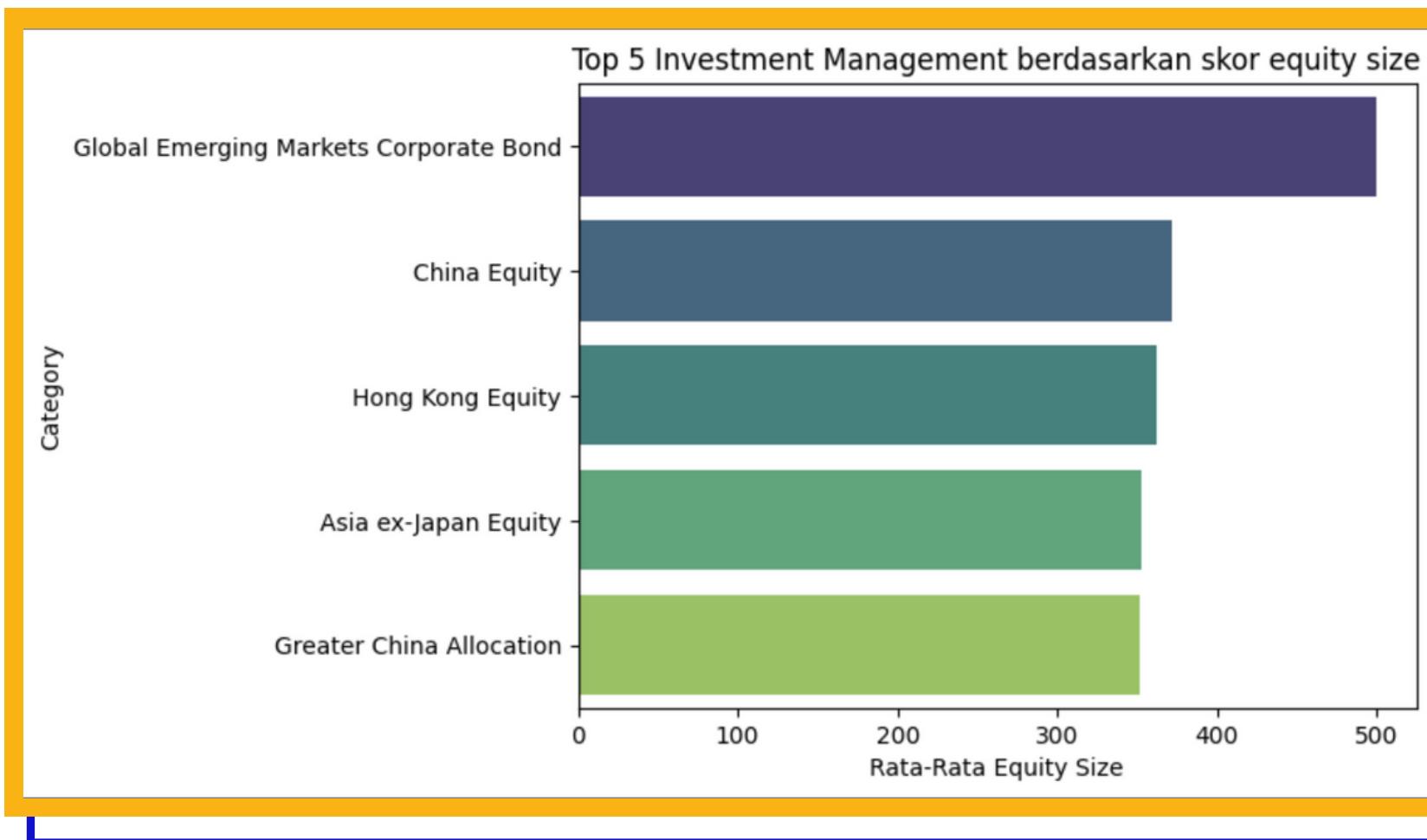
Melakukan sorting kategori management bedasarkan equity size score terbesar

```
[ ] # Mengurutkan data berdasarkan rata-rata dana kelolaan secara descending
sorted_category = group_category.sort_values(by='equity_size_score', ascending=False).reset_index()
sorted_category = sorted_category.drop(columns=['index'])
top_5_equity_size = sorted_category[['category', 'equity_size_score']].head(5)
top_5_equity_size
```

5

Top 5 Investment management berdasarkan equity_size_score

Exploratory



- **Dominasi Global Emerging Markets Corporate Bond:** Global Emerging Markets Corporate Bond memimpin dengan rataan equity size 499.5 satuan.
- **Persaingan Ketat antara Asia ex-japan Euqity dan Greater China Allocation:** Keduanya bersaing ketat dengan masing-masing equity sekitar 352.6 satuan dan 351.2 satuan.

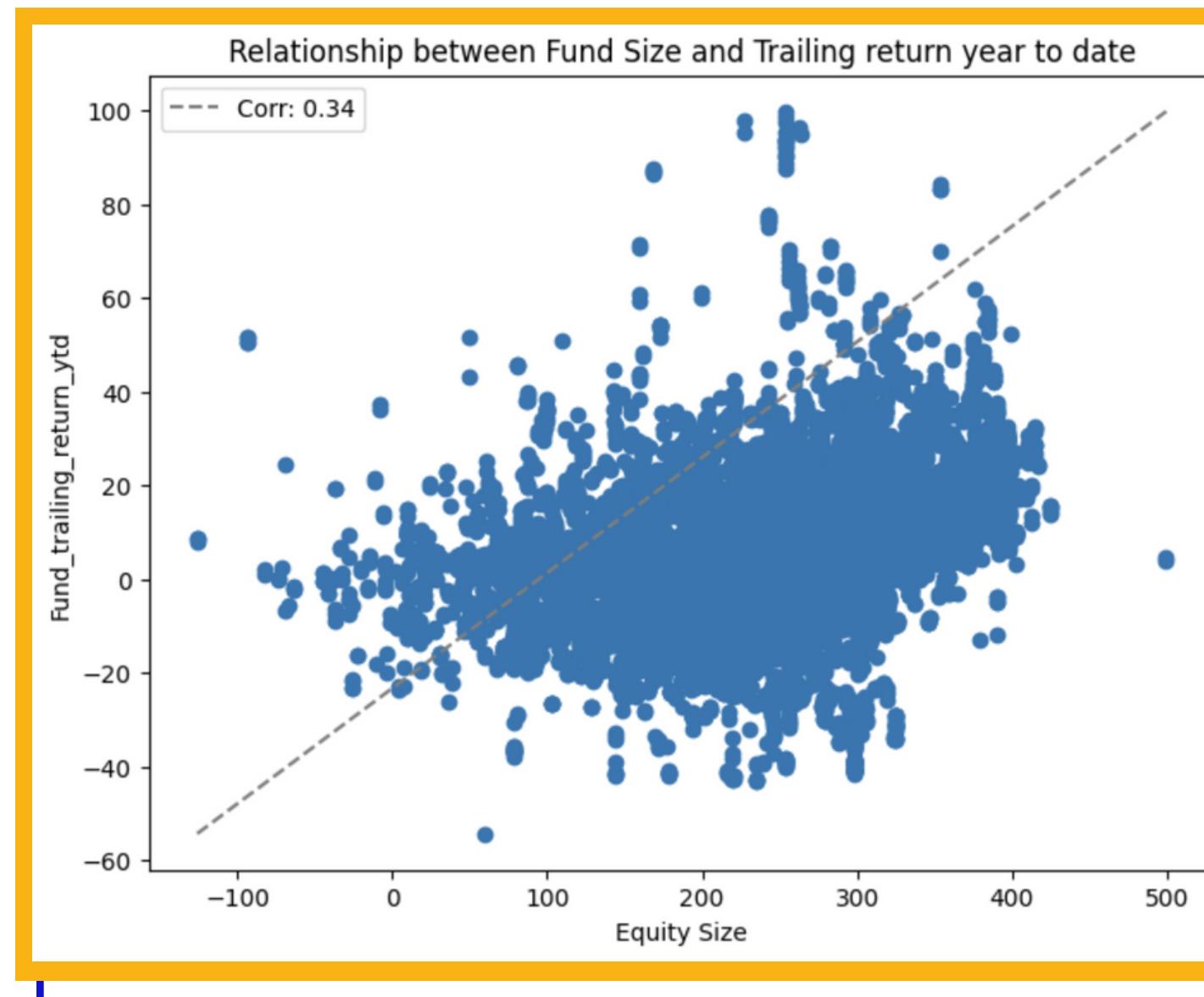
6

Hubungan antara Fund Size dan Trailing return year to date

Exploratory

Korelasi antara fund size dan trailing return year to date

Mencari korelasi antara kedua fitur tersebut.



Terdapat korelasi yang cukup untuk und size dan trailing return year to date

corr = 0.34 (moderate positive correlation)

Maka dari itu, dapat disimpulkan bahwa fund size **memiliki hubungan** dengan trailing return year to date

Predictive Modelling

**Model untuk memprediksi
"long_term_projected_
earnings_growth"**

**Model untuk
mengklasifikasi "rating"**

**Clustering dan analisis dari
jenis-jenis manajer
investasi yang terdapat
dalam suatu cluster**

1

Model untuk memprediksi "long_term_projected_earnings_growth"

Modelling

Melakukan label encoding untuk column yang bersifat kategorikal

```
# Label encoding

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

eim_df_regress = eim_df_regress.drop(columns=["ticker"])
eim_df_regress["category"] = le.fit_transform(eim_df_regress["category"]) +1
eim_df_regress['equity_style'] = eim_df_regress['equity_style'].replace({'Value': 1, 'Blend': 2, 'Growth': 3})
eim_df_regress['equity_size'] = eim_df_regress['equity_size'].replace({'Small': 10, 'Medium': 20, 'Large': 30})
eim_df_regress["nav_per_share_currency"] = le.fit_transform(eim_df_regress["nav_per_share_currency"]) +1
eim_df_regress["shareclass_size_currency"] = le.fit_transform(eim_df_regress["shareclass_size_currency"]) +1
eim_df_regress["fund_size_currency"] = le.fit_transform(eim_df_regress["fund_size_currency"]) +1
eim_df_regress
```

equity_style	equity_size	category
Value	Large	1
Blend	Medium	2



	category	equity_style	equity_size
0	89	1	30
1	182	2	20

Model untuk memprediksi "long_term_projected_earnings_growth"

Feature selection menggunakan wrapper method

```
# Feature selection

from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression

model = LinearRegression()
rfe = RFE(model, n_features_to_select=20)
X = eim_df_regress.drop(columns=['long_term_projected_earnings_growth'])
y = eim_df_regress['long_term_projected_earnings_growth']
fit = rfe.fit(X, y)

selected_features = X.columns[fit.support_]

eim_df_regress_selected = eim_df_regress[selected_features]
eim_df_regress_selected.columns
```

Memilih 20 dari 117 fitur

- Dengan memilih hanya 20 fitur, kami dapat menghindari redundansi dan memastikan setiap fitur yang dipilih memberikan informasi yang unik.

```
Index(['equity_style', 'price_book_ratio', 'price_sales_ratio',
       'dividend_yield_factor', 'roa', 'roe', 'asset_stock', 'asset_bond',
       'asset_cash', 'asset_other', 'market_cap_giant', 'market_cap_large',
       'market_cap_medium', 'market_cap_small', 'market_cap_micro',
       'sustainability_rank', 'involvement_controversial_weapons',
       'involvement_palm_oil', 'involvement_small_arms',
       'fund_return_2018_q3'],
      dtype='object')
```

1

Model untuk memprediksi "long_term_projected_earnings_growth"

Modelling

Splitting data training and testing

```
X = eim_df_regress_selected  
  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

- Membagi data training dan testing sebesar 80% dan 20%, dimana variabel independent (X) kami pilih dari feature selection menggunakan wrapper method dan variabel dependen atau target (y) adalah long_term_projected_earnings_growth

1

Model untuk memprediksi "long_term_projected_earnings_growth"

Modelling

Regression Modelling

Lasso Regression

Model ini bisa beberapa koefisien fitur menjadi nol

Random Forest

Kuat dalam menangani non-linearitas dan interaksi fitur.

Gradient Boosting

Mampu menangani data yang rumit dan tidak linear

Membuat lasso regression dengan beberapa nilai alpha yang berbeda

```
alpha_values = [0.2, 0.1, 0.5, 1.0, 2.0, 5.0]

lasso_models = {}

for alpha in alpha_values:
    lasso = Lasso(alpha=alpha)
    lasso.fit(X_train, y_train)

    lasso_models[alpha] = lasso

lasso_models

for alpha, model in lasso_models.items():
    test_pred_lasso = model.predict(X_test)
    print(f'Lasso with Alpha: {alpha}')
    regression_metrics(test_pred_lasso, y_test)
    print("=====")
```

```
Lasso with Alpha: 0.2
MAE: 1.78
MSE: 8.68
RMSE: 2.95
R_squared: 0.34
=====
Lasso with Alpha: 0.1
MAE: 1.77
MSE: 8.50
RMSE: 2.91
R_squared: 0.36
=====
Lasso with Alpha: 0.5
MAE: 1.84
MSE: 9.11
RMSE: 3.02
R_squared: 0.31
```

```
Lasso with Alpha: 1.0
MAE: 1.97
MSE: 10.00
RMSE: 3.16
R_squared: 0.24
=====
Lasso with Alpha: 2.0
MAE: 2.24
MSE: 11.88
RMSE: 3.45
R_squared: 0.10
=====
Lasso with Alpha: 5.0
MAE: 2.30
MSE: 12.34
RMSE: 3.51
R_squared: 0.07
=====
```

- Model lasso yang paling optimal adalah model lasso yang menggunakan nilai **alpha = 0,1** dengan **R_squared sebesar 0,36**

B

Random Forest Regression

Modelling

Membuat Random Forest Regression dengan criterion squared_error

```
from sklearn.ensemble import RandomForestRegressor  
  
rf_regressor = RandomForestRegressor(criterion='squared_error')  
rf_regressor.fit(X_train, y_train)  
y_pred_rf = rf_regressor.predict(X_test)  
regression_metrics(y_pred_rf, y_test)
```

MAE: 0.31
MSE: 1.18
RMSE: 1.09
R_squared: 0.91

- Model Random Forest memiliki R_squared sebesar 0,91

C

Gradient Boosting Regression

Membuat Gradient Boosting dengan n_estimators=200 dan learning_rate=0.2

```
from sklearn.ensemble import GradientBoostingRegressor  
  
gradient_boosting_model = GradientBoostingRegressor(n_estimators=200, learning_rate=0.2, random_state=42)  
gradient_boosting_model.fit(X_train, y_train)  
y_pred_gb = gradient_boosting_model.predict(X_test)  
regression_metrics(y_pred_gb, y_test)
```

MAE: 1.09
MSE: 3.05
RMSE: 1.75
R_squared: 0.77

- Model Gradient Boosting memiliki R_squared sebesar 0,77

1

Model untuk memprediksi "long_term_projected_earnings_growth"

Modelling

Lasso Regression

MAE: 1.77
MSE: 8.50
RMSE: 2.91
R_squared: 0.36

Random Forest

MAE: 0.31
MSE: 1.18
RMSE: 1.09
R_squared: 0.91

Gradient Boosting

MAE: 1.09
MSE: 3.05
RMSE: 1.75
R_squared: 0.77



Model Terbaik

▼ RandomForestRegressor
RandomForestRegressor()

Model Random Forest adalah model yang kami pilih karena:

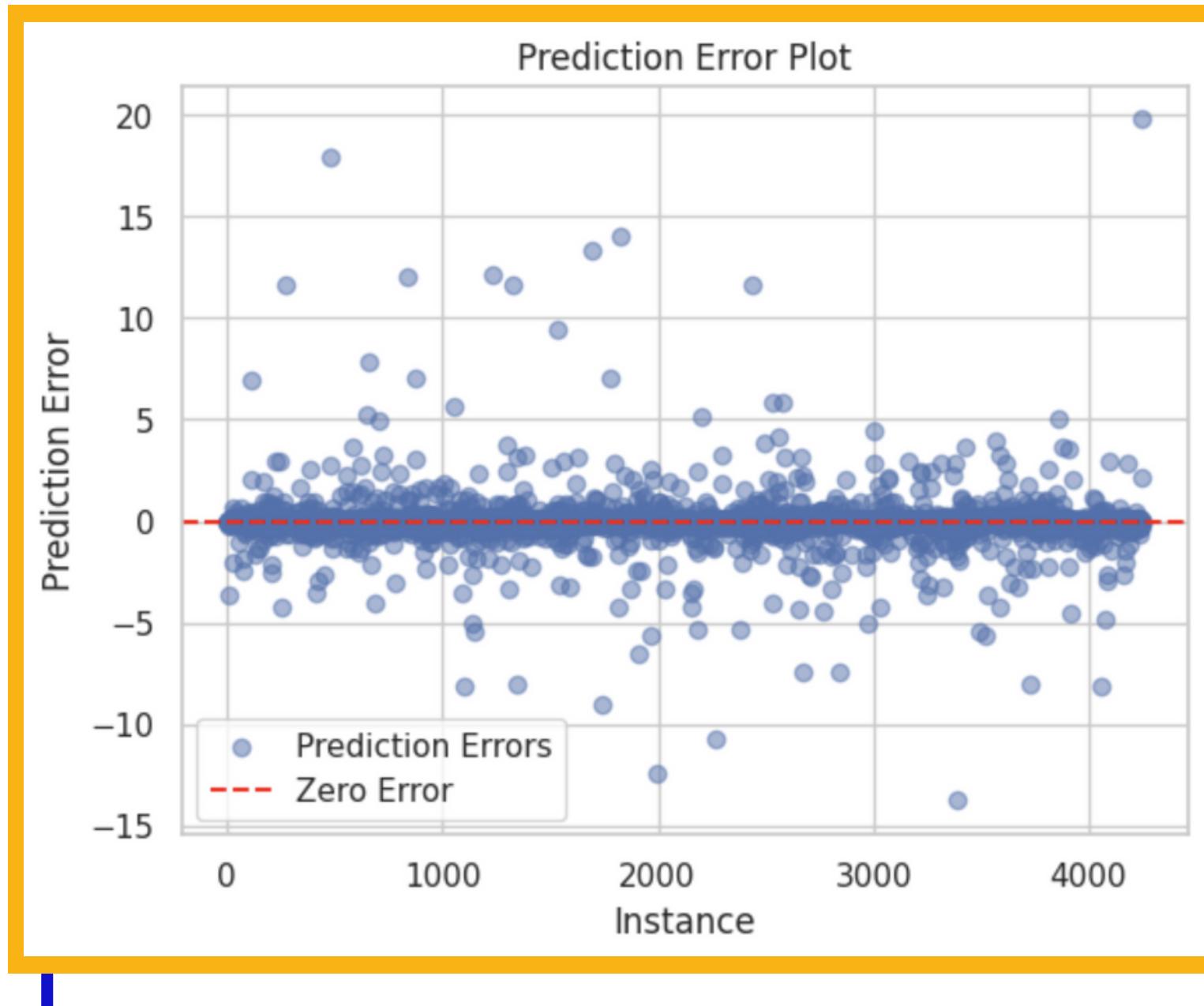
- **Mean Average Error terkecil:** model secara keseluruhan memiliki tingkat kesalahan yang rendah dalam memprediksi nilai
- **R_Squared tertinggi:** model secara baik menjelaskan variasi dalam data target

1

Model untuk memprediksi "long_term_projected_earnings_growth"

Modelling

Evaluasi kinerja model regresi



Prediction error plot memvisualisasikan error values, yaitu selisih actual dan prediction.

Berdasarkan plot disamping dapat diketahui bahwa **majoritas data memiliki error value yang kecil**, ditandai dengan persebaran yang cenderung mendekati garis 0

Value error terbesar yaitu sekitar 20 satuan pengukuran.

Melakukan label encoding untuk column yang bersifat kategorikal

```
# Label encoding

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

eim_df_regress = eim_df_regress.drop(columns=["ticker"])
eim_df_regress["category"] = le.fit_transform(eim_df_regress["category"]) +1
eim_df_regress['equity_style'] = eim_df_regress['equity_style'].replace({'Value': 1, 'Blend': 2, 'Growth': 3})
eim_df_regress['equity_size'] = eim_df_regress['equity_size'].replace({'Small': 10, 'Medium': 20, 'Large': 30})
eim_df_regress["nav_per_share_currency"] = le.fit_transform(eim_df_regress["nav_per_share_currency"]) +1
eim_df_regress["shareclass_size_currency"] = le.fit_transform(eim_df_regress["shareclass_size_currency"]) +1
eim_df_regress["fund_size_currency"] = le.fit_transform(eim_df_regress["fund_size_currency"]) +1
eim_df_regress
```

equity_style	equity_size	category
Value	Large	1
Blend	Medium	2



	category	equity_style	equity_size
0	89	1	30
1	182	2	20

Feature selection menggunakan wrapper method

```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression

model = LinearRegression()
rfe = RFE(model, n_features_to_select=20)
X_classif = df_eim_train.drop(columns=['rating'])
y_classif = df_eim_train['rating']
fit = rfe.fit(X_classif, y_classif)

selected_features = X_classif.columns[fit.support_]

eim_df_classif_selected = df_eim_train[selected_features]
eim_df_classif_selected.columns
```

Memilih 20 dari 117 fitur

- Dengan memilih hanya 20 fitur, kami dapat menghindari redundansi dan memastikan setiap fitur yang dipilih memberikan informasi yang unik.

```
Index(['equity_style', 'asset_stock', 'asset_bond', 'asset_cash',
       'asset_other', 'market_cap_giant', 'market_cap_large',
       'market_cap_micro', 'ongoing_cost', 'management_fees',
       'environmental_score', 'social_score', 'sustainability_rank',
       'involvement_palm_oil', 'involvement_pesticides', 'involvement_tobacco',
       'fund_trailing_return_5years', 'fund_return_2020_q1',
       'fund_return_2019_q2', 'quarters_up'],
      dtype='object')
```

Splitting data training and testing

```
x_classif = eim_df_classif_selected
] ✓ 0.0s

x_train, x_test, y_train, y_test = train_test_split(x_classif, y_classif, test_size=0.2, random_state=42)
] ✓ 0.0s
```

- Membagi data training dan testing sebesar 80% dan 20%, dimana variabel independent (X) kami pilih dari feature selection menggunakan wrapper method dan variabel dependen atau target (y) adalah rating

2

Model untuk mengklasifikasi "rating"

Modelling

Classification Modelling

Random Forest

Dapat menangani jumlah fitur yang besar dan memilih subset fitur yang relevan dalam proses pembentukannya.

A

Random Forest Classifier

Modelling

Membuat Random Forest Classifier dengan criterion gini

```
# rfc_1 = RandomForestClassifier(**clf_rfc.best_params_)  
rfc_1 = RandomForestClassifier(criterion='gini')  
/ 0.0s
```

```
Accuracy: 0.82  
F1 Score: 0.82  
Recall Score: 0.80  
Precision Score: 0.83
```

- Model Random Forest Classifier memiliki **Accuracy sebesar 0,82**

```
Accuracy: 0.82  
F1 Score: 0.82  
Recall Score: 0.80  
Precision Score: 0.83
```



Model Terbaik

```
▼ RandomForestRegressor  
RandomForestRegressor()
```

Model Random Forest adalah model yang kami pilih karena:

- **Accuracy:** model **sebesar 0.82** menunjukkan seberapa baik model dapat memprediksi kelas dengan benar secara keseluruhan
- **F1 Score:** model **sebesar 0.82** menunjukkan keseimbangan antara kemampuan model untuk mengidentifikasi kelas positif dan menghindari mengklasifikasikan yang seharusnya negatif.

3

Clustering dan analisis dari jenis-jenis manajer investasi

Modelling

K-Means

Cocok untuk eksplorasi data dan
untuk data dengan bentuk bulat

K-Means

```
# Feature Selection
corr_matrix = eim_encoded.corr()

correlation_threshold = 0.9 # Define the correlation threshold

high_corr_features_set = set() # Set of features with high correlation

for feature in corr_matrix.columns:
    high_corr_features = corr_matrix.index[(abs(corr_matrix[feature]) > correlation_threshold)
                                             & (corr_matrix.index != feature)]
    
    if len(high_corr_features) > 0:
        print("Feature:", feature)
        for high_corr_feature in high_corr_features:
            correlation_value = corr_matrix.loc[high_corr_feature, feature]
            print(f"- {high_corr_feature}: {correlation_value}")
        print()
    high_corr_features_set.update(set(high_corr_features))

high_corr_features_set = list(high_corr_features_set)
print(high_corr_features_set)
```

Feature Selection dengan memilih feature yang memiliki skor korelasi > 0.9 dengan satu sama lain

Clustering dan analisis dari jenis-jenis manajer investasi

```
num_of_cluster = [2, 3, 4, 5, 6, 7, 8]

fig, ax = plt.subplots(4, 2, figsize=(20,10))
for k in num_of_cluster:
    # Create KMeans instance for different number of clusters
    clusterer = KMeans(n_clusters = k, n_init=10)

    # Draw silhouette diagram
    q, mod = divmod(k, 2)
    visualizer = SilhouetteVisualizer(clusterer, colors = 'yellowbrick', ax = ax[q-1][mod])
    visualizer.fit(X)

    # Compute silhouette score
    # This gives a perspective into the density and separation of the formed clusters
    cluster_labels = clusterer.fit_predict(X)
    silhouette_avg = silhouette_score(X, cluster_labels)
    print(
        "For n_clusters =",
        k,
        "The average silhouette_coefficient is :",
        silhouette_avg,
    )
```

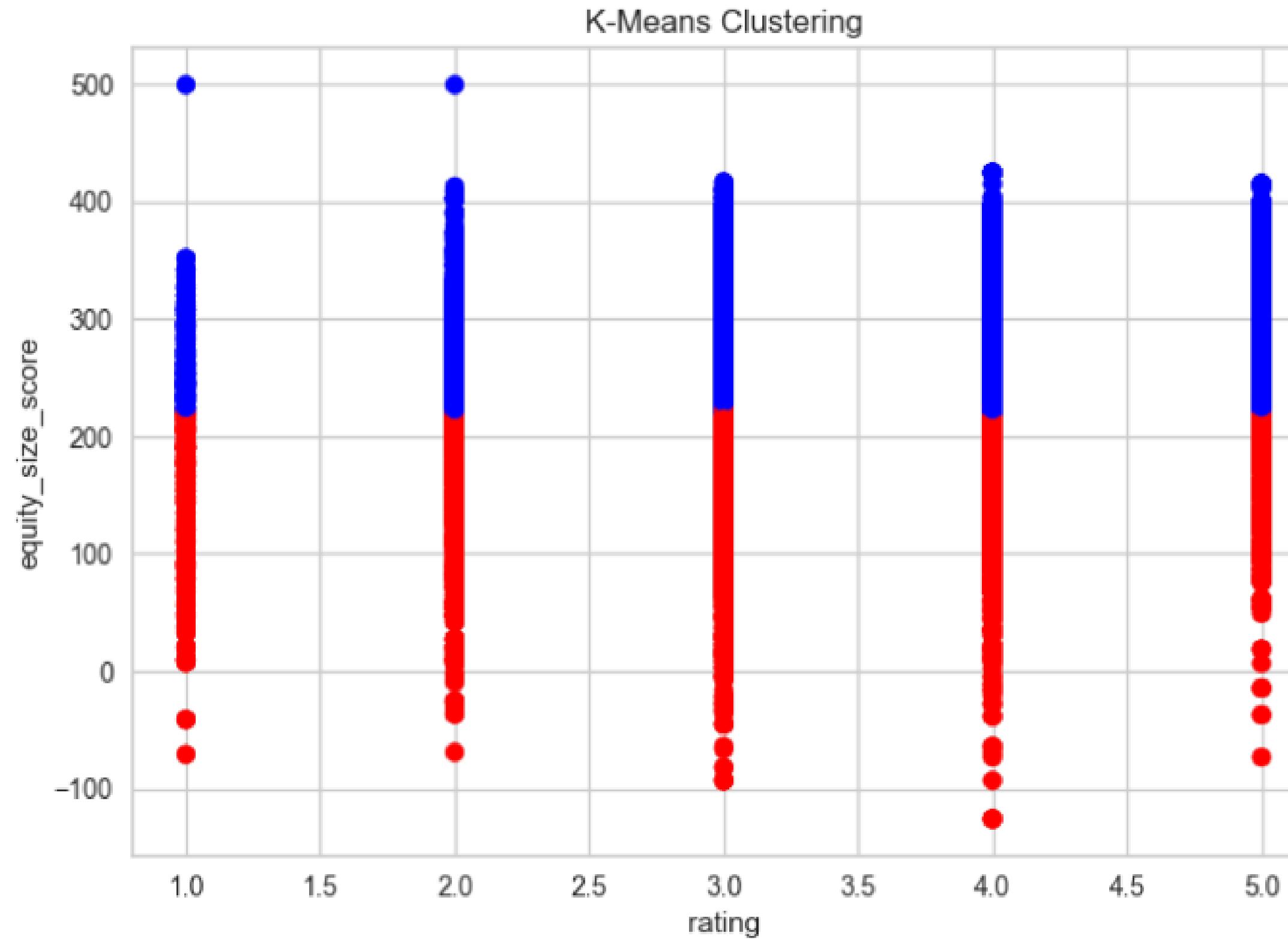
```
For n_clusters = 2 The average silhouette_coefficient is : 0.5007525732867555
For n_clusters = 3 The average silhouette_coefficient is : 0.3986353587466371
For n_clusters = 4 The average silhouette_coefficient is : 0.3638698143225515
For n_clusters = 5 The average silhouette_coefficient is : 0.33583612042253225
For n_clusters = 6 The average silhouette_coefficient is : 0.3235488043524996
For n_clusters = 7 The average silhouette_coefficient is : 0.3265932917262976
```

n_cluster = 2 merupakan jumlah kluster dengan skor silhouette coefficient tertinggi, sehingga akan dilakukan clustering dengan 2 kluster

3

Clustering dan analisis dari jenis-jenis manajer investasi

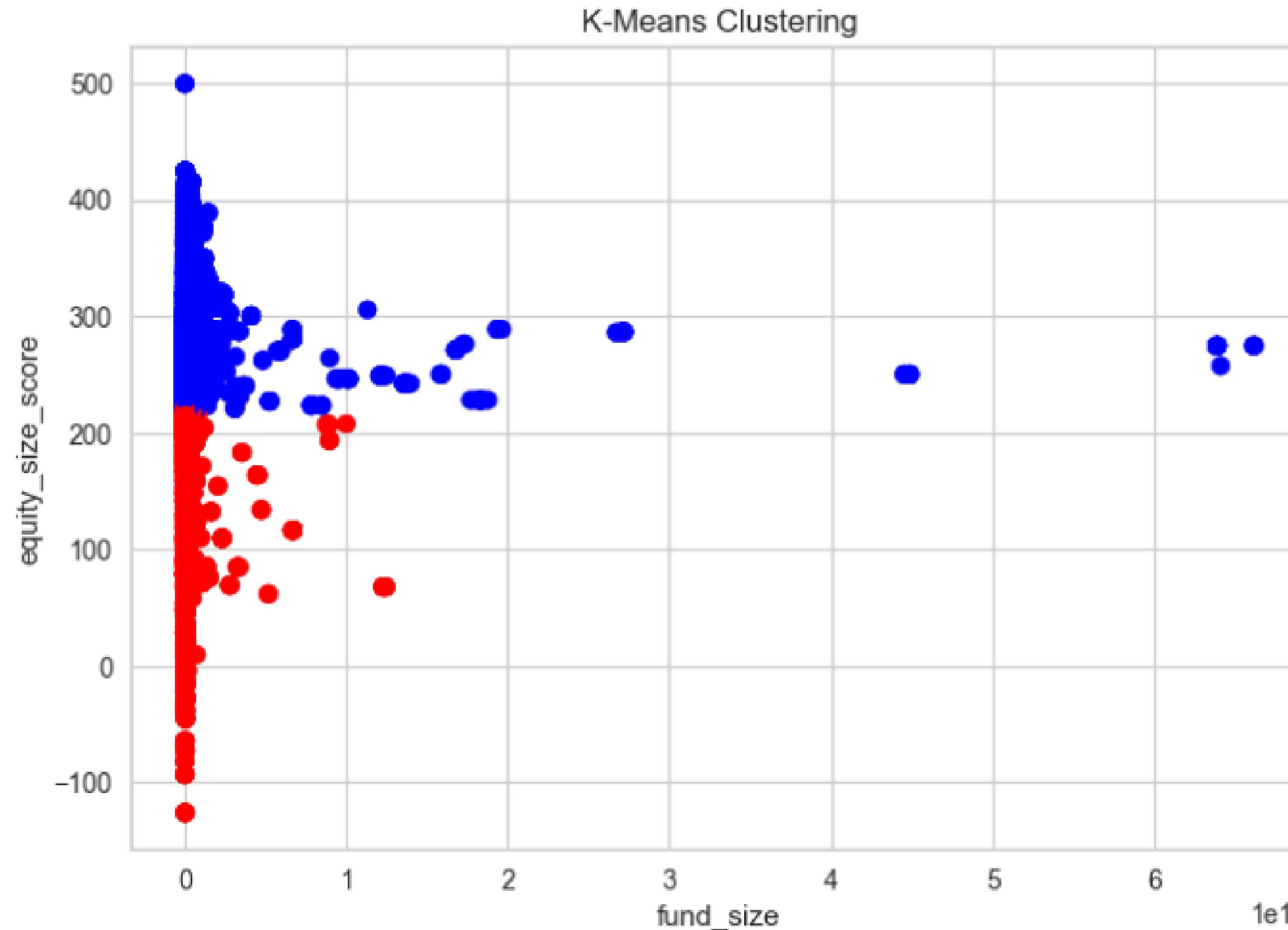
Modelling



3

Clustering dan analisis dari jenis-jenis manajer investasi

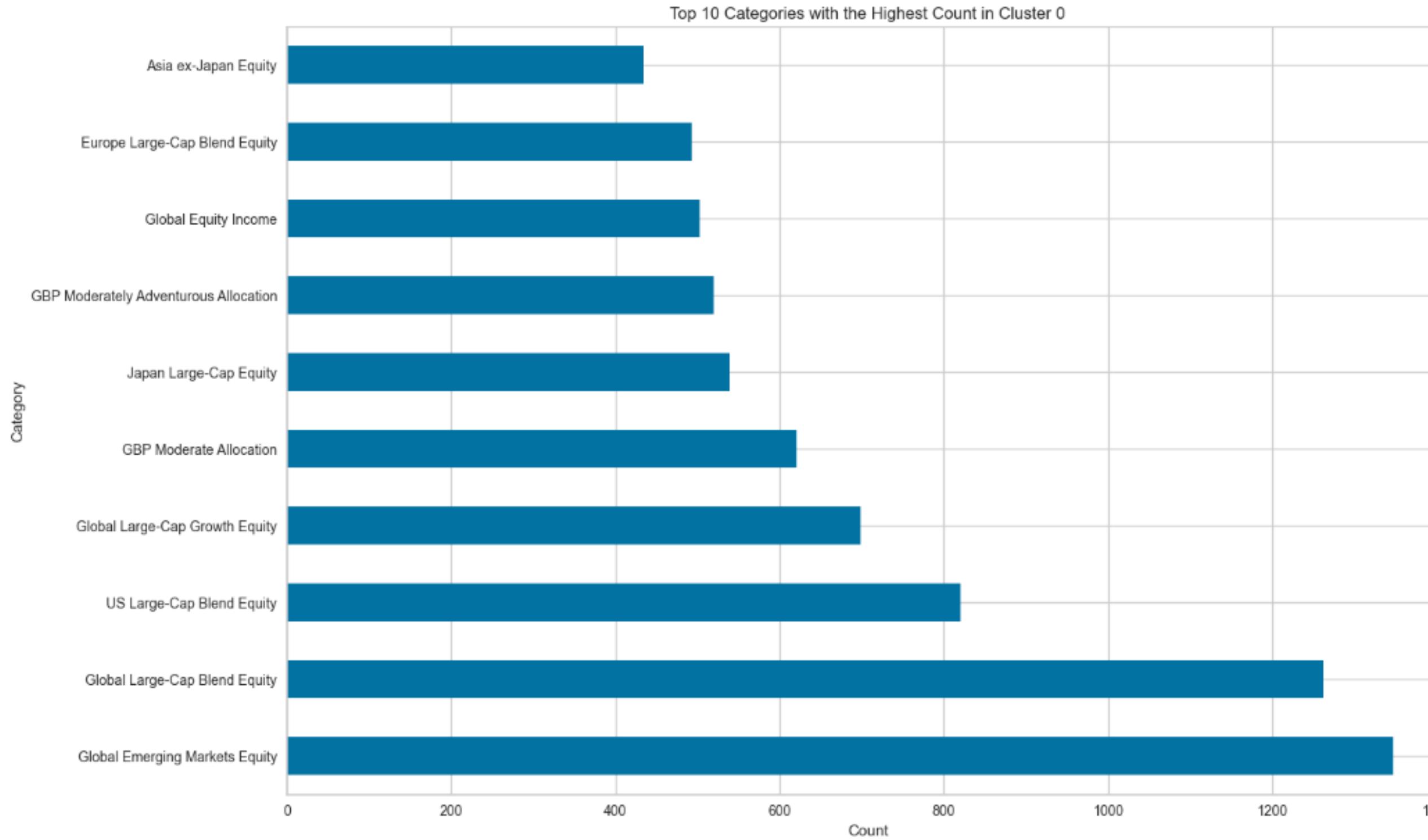
Modelling



3

Clustering dan analisis dari jenis-jenis manajer investasi

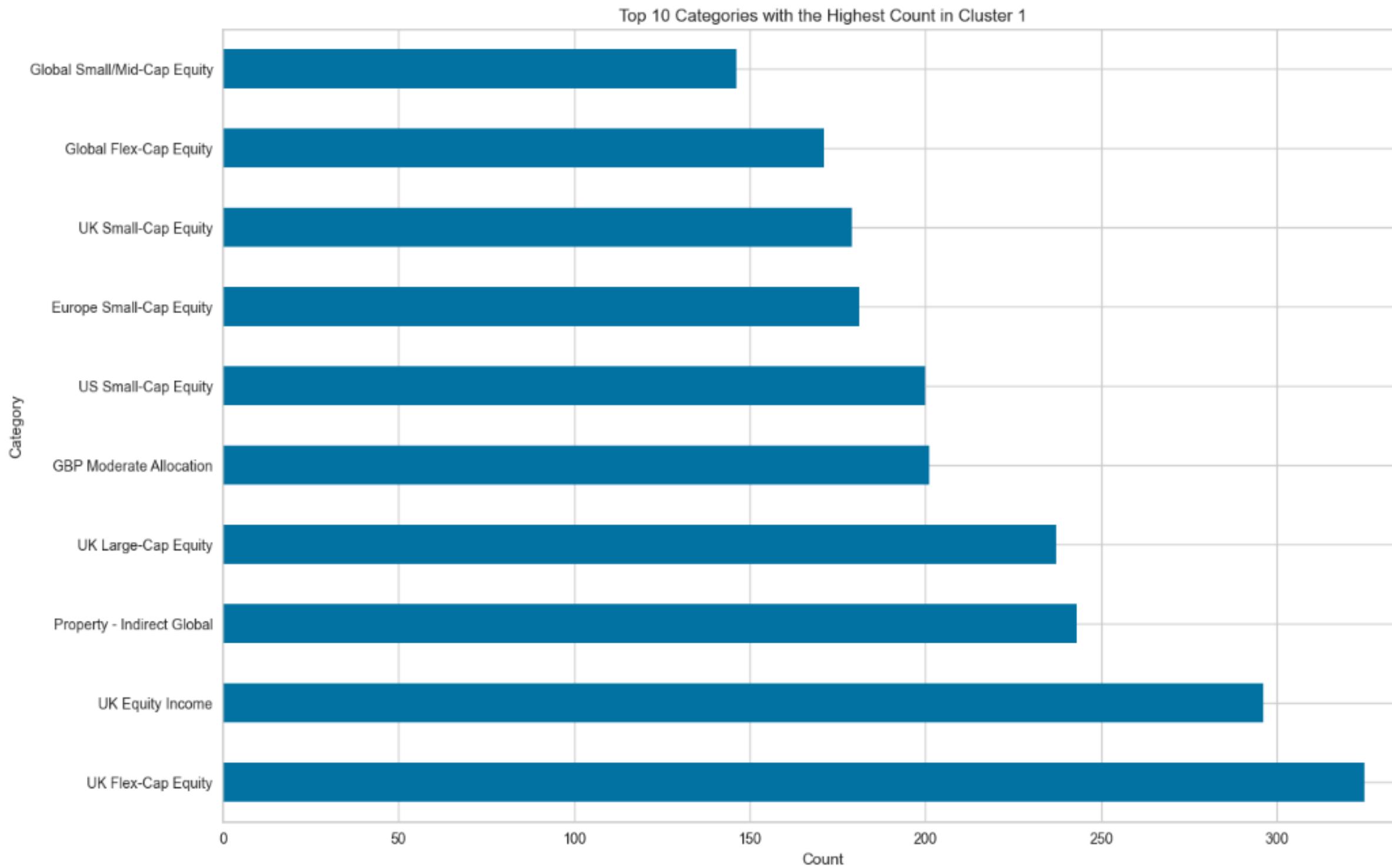
Modelling



3

Clustering dan analisis dari jenis-jenis manajer investasi

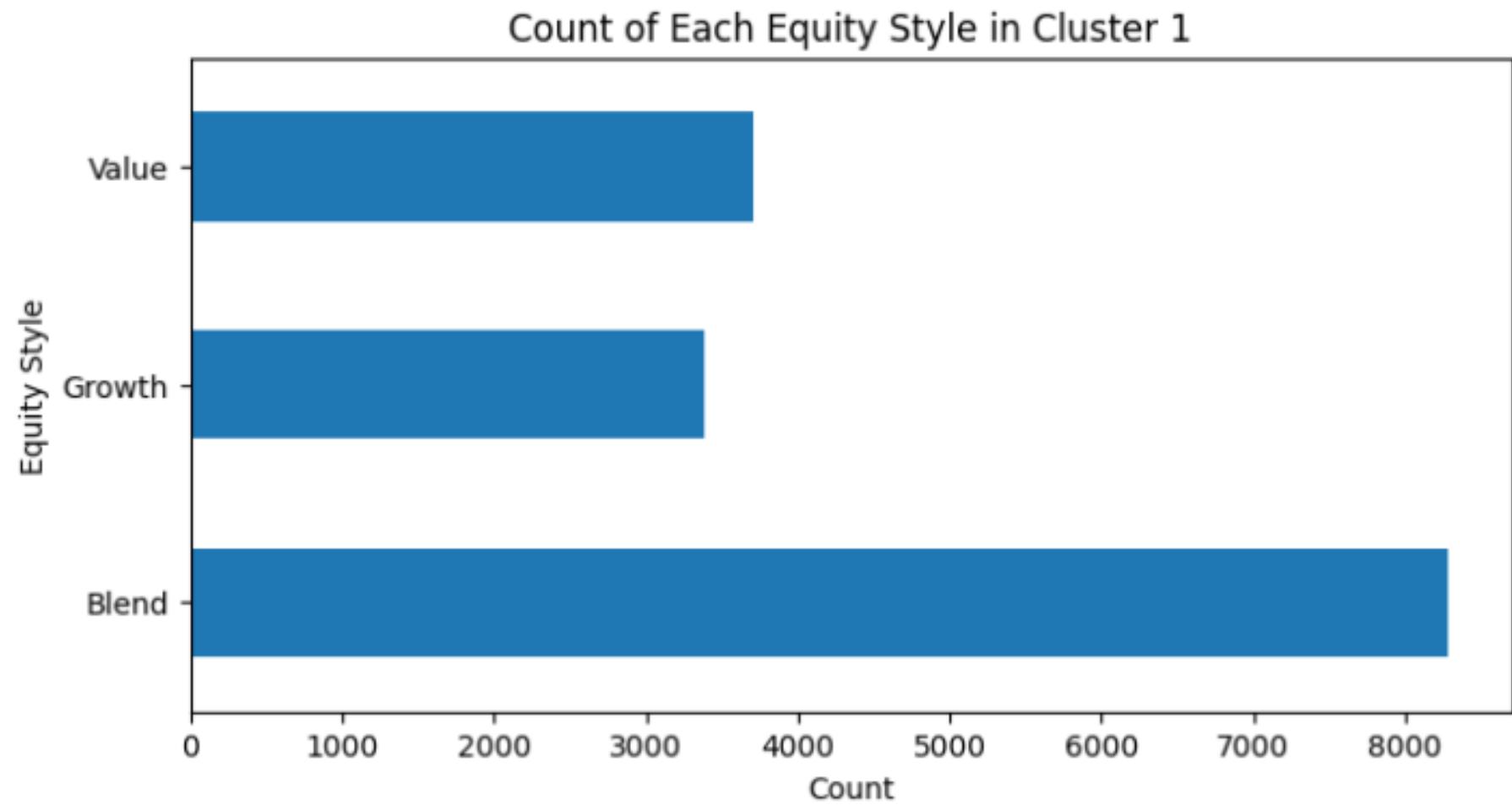
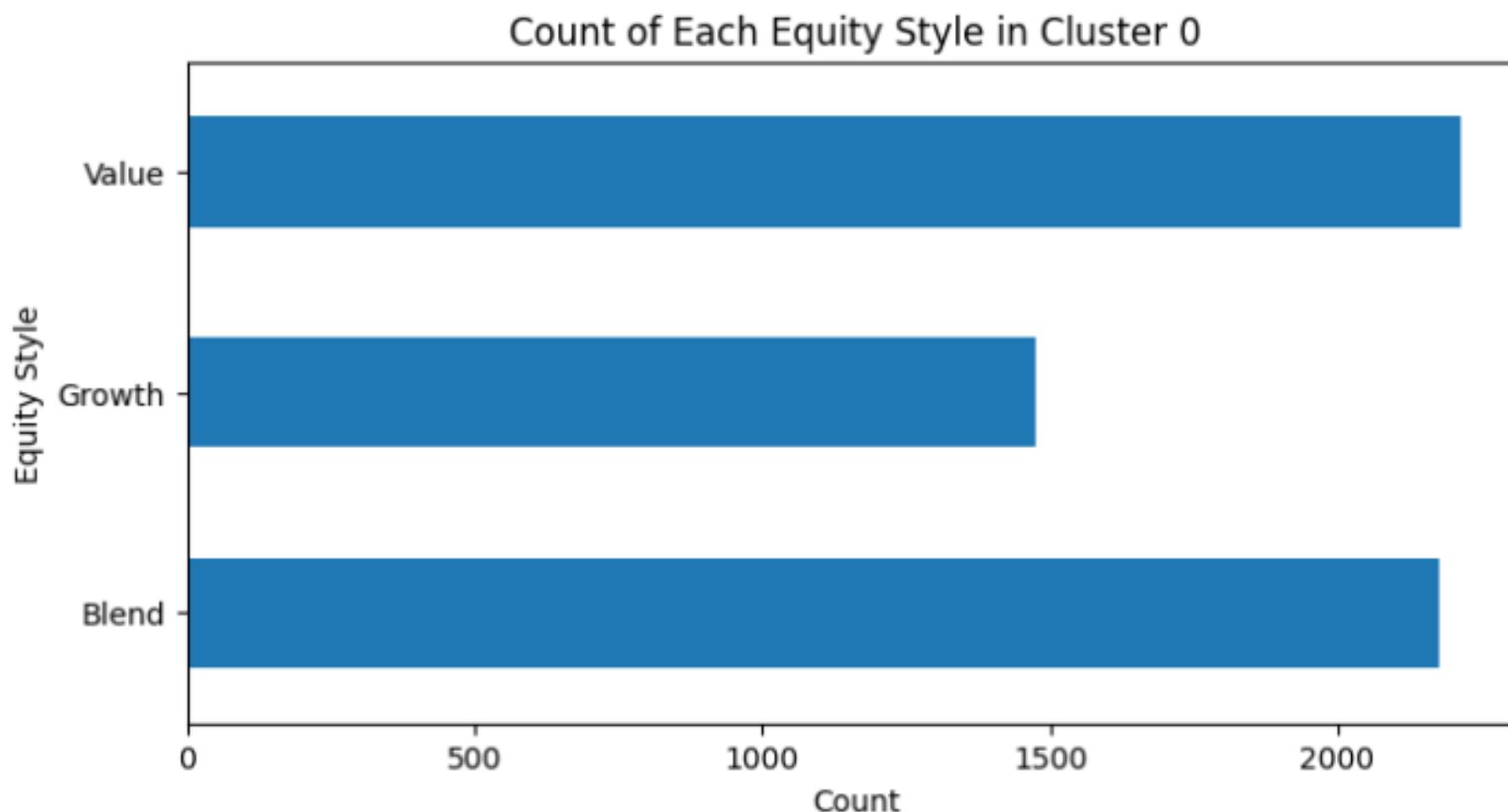
Modelling



3

Clustering dan analisis dari jenis-jenis manajer investasi

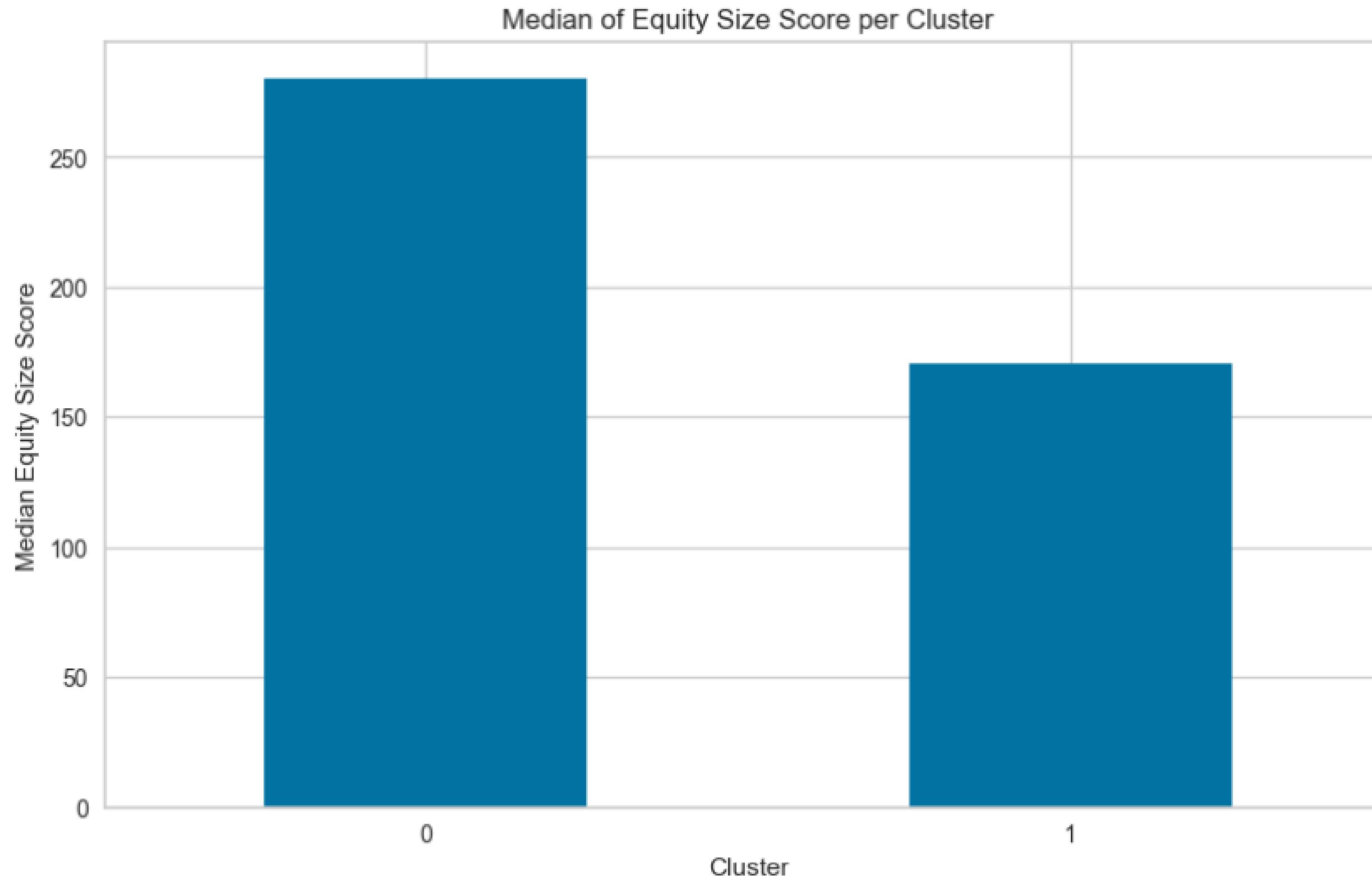
Modelling



3

Clustering dan analisis dari jenis-jenis manajer investasi

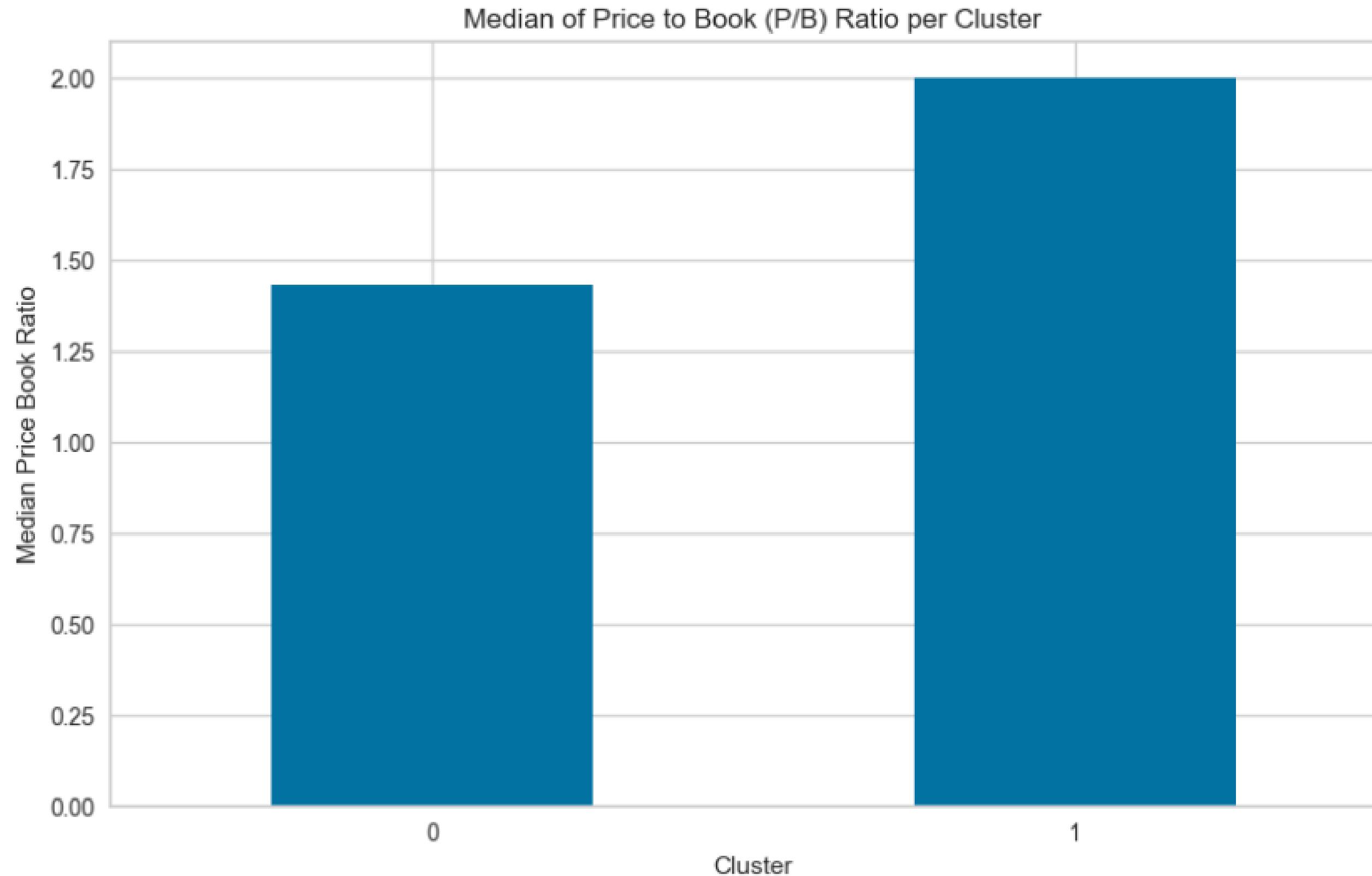
Modelling



3

Clustering dan analisis dari jenis-jenis manajer investasi

Modelling



by НТП



THANK YOU