In [1]:
```
import numpy as np
import pandas as pd
df = pd.read_stata('ccm.dta')
#df.head()
pd.set_option('display.max_columns', None)
#print(df.columns)
```

Selecting Firm types

- went with slightly bigger firms than originally expected
- wanted more data

In [2]:
```
# Filter Market Caps and companies that dont have gross profit
MiddleMarket_data = df[(df['mkvalt'] >= 0) & (df['mkvalt'] <= 12000)]
# Calculate market capitalization
MiddleMarket_data['Avg_Price'] = (MiddleMarket_data['prch_c'] + MiddleMarket_data['prcl_c']) / 2
MiddleMarket_data['Market_Cap'] = MiddleMarket_data['Avg_Price'] * MiddleMarket_data['csho']
```

```
/var/folders/_8/vpmh2zyd23z9mnscjwzk9t1r0000gn/T/ipykernel_45215/2959442128.py:4: PerformanceWarn
ing: DataFrame is highly fragmented.  This is usually the result of calling `frame.insert` many t
imes, which has poor performance.  Consider joining all columns at once using pd.concat(axis=1) i
nstead. To get a de-fragmented frame, use `newframe = frame.copy()`
  MiddleMarket_data['Avg_Price'] = (MiddleMarket_data['prch_c'] + MiddleMarket_data['prcl_c']) /
2
/var/folders/_8/vpmh2zyd23z9mnscjwzk9t1r0000gn/T/ipykernel_45215/2959442128.py:4: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/ind
exing.html#returning-a-view-versus-a-copy
  MiddleMarket_data['Avg_Price'] = (MiddleMarket_data['prch_c'] + MiddleMarket_data['prcl_c']) /
2
/var/folders/_8/vpmh2zyd23z9mnscjwzk9t1r0000gn/T/ipykernel_45215/2959442128.py:5: PerformanceWarn
ing: DataFrame is highly fragmented.  This is usually the result of calling `frame.insert` many t
imes, which has poor performance.  Consider joining all columns at once using pd.concat(axis=1) i
nstead. To get a de-fragmented frame, use `newframe = frame.copy()`
  MiddleMarket_data['Market_Cap'] = MiddleMarket_data['Avg_Price'] * MiddleMarket_data['csho']
/var/folders/_8/vpmh2zyd23z9mnscjwzk9t1r0000gn/T/ipykernel_45215/2959442128.py:5: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/ind
exing.html#returning-a-view-versus-a-copy
  MiddleMarket_data['Market_Cap'] = MiddleMarket_data['Avg_Price'] * MiddleMarket_data['csho']
```

Renaming/Reorganizing downloaded data

In [3]:
```
#(MiddleMarket_data['ebitda'] > 1).sum()
#filtered_data = MiddleMarket_data[MiddleMarket_data['ebitda'] > 1]
#filtered_data['ebitda']
filtered_data = MiddleMarket_data.loc[MiddleMarket_data['ebitda'] > 0, ['tic','ebitda','dvc','ob'
#filtered_data
filtered_data.columns = ['Ticker', 'EBITDA', 'Dividends', 'Operating Income', 'Company Name', 'Fi
filtered_data.reindex()
```

Out[3]:

	Ticker	EBITDA	Dividends	Operating Income	Company Name	Fiscal Date	Long-Term Debt	Fiscal Year	Research and Development Expenses	Cost of Goods Sold	Dep
0	AIR	101.800	0.100	750.0	AAR CORP	2021-07-22	565.300	5	NaN	1364.600	
1	AIR	149.300	0.000	850.0	AAR CORP	2022-07-22	539.400	5	NaN	1470.300	
2	AIR	179.300	0.000	740.0	AAR CORP	2023-07-21	734.000	5	NaN	1591.300	
5	AAL	4103.000	0.000	NaN	AMERICAN AIRLINES GROUP INC	2023-02-26	70515.000	12	NaN	37631.000	
6	AAL	6267.000	0.000	NaN	AMERICAN AIRLINES GROUP INC	2024-02-23	68260.000	12	NaN	38716.000	
...	...	...	...	...	...	...	...	...	...	...	...
21958	HYFM	36.742	0.000	NaN	HYDROFARM HLDNG GP INC	2022-03-31	256.062	12	NaN	374.733	
21968	KARO	84.155	0.000	NaN	KAROOOOO LTD	2022-06-20	59.330	2	9.673	27.559	
21969	KARO	81.977	15.995	NaN	KAROOOOO LTD	2023-06-21	57.738	2	9.645	37.580	
21976	HSHP	23.744	0.000	NaN	HIMALAYA SHIPPING LTD	2024-04-12	445.001	12	NaN	9.146	
21985	CLCO	277.522	87.511	NaN	COOL COMPANY LTD	2024-03-28	1250.363	12	NaN	77.315	

9342 rows × 32 columns

Selecting Varriables/Financial metrics of Interest

In [4]:
```
# Reorder the columns
filtered_data = filtered_data[['Ticker', 'Company Name', 'Market_Cap','Fiscal Date', 'Fiscal Year
                               'EBITDA', 'Operating Income', 'Dividends',
                               'Long-Term Debt', 'Research and Development Expenses',
                               'Cost of Goods Sold', 'Operating Income Before Depreciation',
                               'Capital Expenditures', 'Total Assets', 'Revenue',
                               'Selling, General, and Administrative Expenses',
                               'Property, Plant, and Equipment Net',
                               'Gross Property, Plant, and Equipment', 'Cash and Equivalents',
                               'Common Equity', 'Current Liabilities', 'Common Shares Outstanding
                               '52-Week Low Price', '52-Week High Price', 'Avg_Price',
                                'Inventory', 'OpInc After Dep','Interest Expense','Taxes', 'Curre

# Display the reordered DataFrame
filtered_data = filtered_data[(filtered_data['Operating Income'] >= .00)]
filtered_data.replace([np.inf, -np.inf], np.nan, inplace=True)
filtered_data.fillna(0, inplace=True)
filtered_data = filtered_data[(filtered_data['Market_Cap'] >= 100)]
filtered_data.describe()
```

```
/var/folders/_8/vpmh2zyd23z9mnscjwzk9t1r0000gn/T/ipykernel_45215/362217121.py:17: FutureWarning:
Setting an item of incompatible dtype is deprecated and will raise in a future error of pandas. V
alue '0' has dtype incompatible with datetime64[ns], please explicitly cast to a compatible dtype
first.
  filtered_data.fillna(0, inplace=True)
```

Out[4]:

	Market_Cap	Fiscal Year	Employees	SP Index Code	EBITDA	Operating Income	Dividends	Long-Term Debt
count	1865.000000	1865.000000	1865.000000	1865.000000	1865.000000	1865.000000	1865.000000	1865.000000
mean	2565.155154	9.635925	12.313823	4672.390885	345.178820	1477.291645	28.439949	2500.088561
std	2717.436271	3.803079	23.895366	2077.657616	520.606361	4531.653675	115.608116	7288.660880
min	101.146145	1.000000	0.000000	0.000000	0.188000	0.000000	-1.000000	0.000000
25%	567.022015	9.000000	1.767000	3530.000000	62.692000	0.000000	0.000000	372.338000
50%	1467.084995	12.000000	4.850000	3845.000000	178.222000	117.873000	0.000000	1007.659000
75%	3671.073840	12.000000	13.000000	5900.000000	442.800000	904.600000	25.241000	2461.595000
max	16531.765360	12.000000	300.000000	9997.000000	7101.000000	48600.000000	4207.700000	182626.000000

Calculating mutliples and metrics (features/targets)

In [5]:

```
# EV/EBITDA
filtered_data['Enterprise_Value'] = filtered_data['Market_Cap'] + filtered_data['Long-Term Debt']
filtered_data['EV/EBITDA'] = filtered_data['Enterprise_Value'] / filtered_data['EBITDA']
filtered_data['EV/Sales'] = filtered_data['Enterprise_Value'] / filtered_data['Revenue']
# P/E Ratio
filtered_data['EPS'] = filtered_data['Operating Income Before Depreciation'] / filtered_data['Com
filtered_data['P/E'] = filtered_data['Avg_Price'] / filtered_data['EPS']
# Debt-to-Equity Ratio
filtered_data['Debt_to_Equity'] = filtered_data['Long-Term Debt'] / filtered_data['Common Equity'
# Return on Equity (ROE)
filtered_data['ROE'] = filtered_data['Operating Income Before Depreciation'] / filtered_data['Com
# Current Ratio
filtered_data['Current_Ratio'] = filtered_data['Total Assets'] / filtered_data['Current Liabiliti
# Quick Ratio
filtered_data['Quick_Ratio'] = (filtered_data['Total Assets'] - filtered_data['Inventory']) / fil
# Interest Covered Ratio
filtered_data['Interest_Coverage_Ratio'] = filtered_data['EBITDA'] / filtered_data['Interest Expe
# Gross Margin
filtered_data['Gross_Margin'] = (filtered_data['Revenue'] - filtered_data['Cost of Goods Sold'])
# Operating Margin
filtered_data['Operating_Margin'] = filtered_data['Operating Income Before Depreciation'] / filte
# Calculate Gross Profit
filtered_data['Gross Profit'] = filtered_data['Revenue'] - filtered_data['Cost of Goods Sold']
# Calculate Gross Profit Margin
filtered_data['Gross Profit Margin'] = (filtered_data['Gross Profit'] / filtered_data['Revenue'])
#EV/GP
filtered_data['EV/GP'] = filtered_data['Enterprise_Value'] / filtered_data['Gross Profit']
# Calculate EBITDA - Capex
filtered_data['EBITDA - Capex'] = filtered_data['EBITDA'] - filtered_data['Capital Expenditures']
# Calculate EBITDA - Capex margin
filtered_data['EBITDA - Capex Margin'] = (filtered_data['EBITDA - Capex'] / filtered_data['EBITDA
filtered_data['EV/EBITDA-Capex'] = filtered_data['Enterprise_Value'] / filtered_data['EBITDA - Ca
filtered_data['Free Cash Flow'] = (filtered_data['EBITDA']
                                   - filtered_data['Taxes']
                                   - filtered_data['Interest Expense']
                                   + (filtered_data['Current Assets'] - filtered_data['Current Li
```

```
                                    - filtered_data['Capital Expenditures'])
filtered_data['FCF_Positive'] = (filtered_data['Free Cash Flow'] > 0).astype(int)
filtered_data['FCF Yield'] = filtered_data['Free Cash Flow'] / filtered_data['Market_Cap']
filtered_data['Invested_Capital'] = filtered_data['Long-Term Debt'] + filtered_data['Common Equit
filtered_data['EBIT'] = filtered_data['Operating Income'] + filtered_data['Interest Expense'] + f
filtered_data['EV_EBIT'] = filtered_data['Enterprise_Value'] / filtered_data['EBIT']
filtered_data['ROIC'] = filtered_data['EBIT'] / filtered_data['Invested_Capital']
filtered_data['EBIT Margin (%)'] = (filtered_data['EBIT'] / filtered_data['Revenue'])
filtered_data['EBIT_Positive'] = (filtered_data['EBIT'] > 0).astype(int)
filtered_data

filtered_data['Revenue per Employee'] = filtered_data['Revenue'] / filtered_data['Employees']

filtered_data['Total Debt Service'] = filtered_data['Long-Term Debt'] + filtered_data['Interest E
filtered_data['Debt_Coverage_Ratio'] = filtered_data['Operating Income Before Depreciation'] / fi

filtered_data['Dividend (y/n)'] = (filtered_data['Dividends'] > 0).astype(int)

filtered_data['Price_Range_Ratio'] = (filtered_data['52-Week High Price'] - filtered_data['52-Wee

filtered_data.describe()
```

Out[5]:

	Market_Cap	Fiscal Year	Employees	SP Index Code	EBITDA	Operating Income	Dividends	Long-Term Debt
count	1865.000000	1865.000000	1865.000000	1865.000000	1865.000000	1865.000000	1865.000000	1865.000000
mean	2565.155154	9.635925	12.313823	4672.390885	345.178820	1477.291645	28.439949	2500.088561
std	2717.436271	3.803079	23.895366	2077.657616	520.606361	4531.653675	115.608116	7288.660880
min	101.146145	1.000000	0.000000	0.000000	0.188000	0.000000	-1.000000	0.000000
25%	567.022015	9.000000	1.767000	3530.000000	62.692000	0.000000	0.000000	372.338000
50%	1467.084995	12.000000	4.850000	3845.000000	178.222000	117.873000	0.000000	1007.659000
75%	3671.073840	12.000000	13.000000	5900.000000	442.800000	904.600000	25.241000	2461.595000
max	16531.765360	12.000000	300.000000	9997.000000	7101.000000	48600.000000	4207.700000	182626.000000

Derriving Marginal Effects

In [6]:

```
# Group the data by "Ticker" and sort by "Fiscal Year"
grouped_data = filtered_data.sort_values(by=['Ticker', 'Fiscal Year']).groupby('Ticker')

# Calculate year-over-year changes in margins
filtered_data['YOY_Gross_Margin_Change'] = grouped_data['Gross_Margin'].pct_change()
filtered_data['YOY_EBIT_Margin_Change'] = grouped_data['EBIT Margin (%)'].pct_change()
filtered_data['YOY_Operating_Margin_Change'] = grouped_data['Operating_Margin'].pct_change()
filtered_data['YOY_Gross_Profit_Change'] = grouped_data['Gross Profit'].pct_change()
filtered_data['YOY_Revenue_Change'] = grouped_data['Revenue'].pct_change()
filtered_data['YOY_EBIT_Change'] = grouped_data['EBIT'].pct_change()
filtered_data
```

Out[6]:

	Ticker	Company Name	Market_Cap	Fiscal Date	Fiscal Year	Employees	SP Index Code	EBITDA	Operating Income	Dividends	
0	AIR	AAR CORP	1000.228125	2021-07-22 00:00:00	5	4.700	5080.0	101.800	750.0	0.1	
1	AIR	AAR CORP	1351.759245	2022-07-22 00:00:00	5	4.500	5080.0	149.300	850.0	0.0	
2	AIR	AAR CORP	1511.513640	2023-07-21 00:00:00	5	5.000	5080.0	179.300	740.0	0.0	
7	CECO	CECO ENVIRONMENTAL CORP	221.591938	2021-04-01 00:00:00	12	0.730	3564.0	30.514	183.1	0.0	
8	CECO	CECO ENVIRONMENTAL CORP	266.913360	2022-04-10 00:00:00	12	0.730	3564.0	21.893	213.9	0.0	
...	...	...	...	...	...	...	...	...	...	...	
21898	NVT	NVENT ELECTRIC PLC	8013.274605	2024-03-04 00:00:00	12	11.300	3440.0	772.300	639.1	119.7	3
21909	ACA	ARCOSA INC	2058.140000	2021-03-06 00:00:00	12	6.410	3440.0	283.700	525.1	9.8	
21910	ACA	ARCOSA INC	2804.056500	2022-03-11 00:00:00	12	6.170	3440.0	267.000	552.2	9.8	1
21911	ACA	ARCOSA INC	2643.124000	2023-03-06 00:00:00	12	5.230	3440.0	324.500	896.4	9.8	1
21912	ACA	ARCOSA INC	3310.875000	2024-03-01 00:00:00	12	6.075	3440.0	339.400	1621.2	9.8	1

1865 rows × 70 columns

Defining Industry

In [7]:

```
# Define a function to map SP Index Codes to divisions
def map_sp_index_to_division(sp_index_code):
    if sp_index_code >= 100 and sp_index_code <= 999:
        return 'Agriculture, Forestry and Fishing'
    elif sp_index_code >= 1000 and sp_index_code <= 1499:
        return 'Mining'
    elif sp_index_code >= 1500 and sp_index_code <= 1799:
        return 'Construction'
    elif sp_index_code >= 2000 and sp_index_code <= 3999:
        return 'Manufacturing'
    elif sp_index_code >= 4000 and sp_index_code <= 4999:
        return 'Transportation, Communications, Electric, Gas and Sanitary service'
    elif sp_index_code >= 5000 and sp_index_code <= 5199:
        return 'Wholesale Trade'
```

```
    elif sp_index_code >= 5200 and sp_index_code <= 5999:
        return 'Retail Trade'
    elif sp_index_code >= 6000 and sp_index_code <= 6799:
        return 'Finance, Insurance and Real Estate'
    elif sp_index_code >= 7000 and sp_index_code <= 8999:
        return 'Services'
    elif sp_index_code >= 9100 and sp_index_code <= 9729:
        return 'Public Administration'
    elif sp_index_code >= 9900 and sp_index_code <= 9999:
        return 'Nonclassifiable'
    else:
        return 'Unknown'

# Add a new column 'Division' based on 'SP Index Code'
filtered_data['Division'] = filtered_data['SP Index Code'].apply(map_sp_index_to_division)
```

More reorganizing for readability

In [8]:
```
# Select columns of interest
selected_columns = ['Ticker', 'EV/EBITDA','EV/Sales', 'P/E', 'EV_EBIT','Debt_to_Equity', 'ROE','F
                    'Current_Ratio', 'Quick_Ratio', 'Interest_Coverage_Ratio', 'Gross_Margin', 'D
                    'Operating_Margin', 'EV/GP' , 'EBIT Margin (%)', 'EV/EBITDA-Capex','YOY_Gross
                    'YOY_Operating_Margin_Change', 'YOY_Gross_Profit_Change', 'YOY_Revenue_Change

# Create a new DataFrame with selected columns
selected_df = filtered_data[selected_columns]
selected_df
```

Out[8]:

	Ticker	EV/EBITDA	EV/Sales	P/E	EV_EBIT	Debt_to_Equity	ROE	FCF_Positive	FCF Yield	ROI
0	AIR	14.787113	0.911547	9.825424	1.946881	0.580152	0.104475	1	0.667348	0.50217
1	AIR	12.272333	1.008343	9.053980	2.084482	0.521411	0.144321	1	0.563710	0.55848
2	AIR	12.067561	1.086966	8.430082	2.761248	0.667819	0.163133	1	0.564070	0.42747
7	CECO	13.091825	1.264146	7.261976	2.099155	1.064370	0.150569	1	0.421906	0.45488
8	CECO	20.333365	1.373352	12.191722	2.027659	1.027797	0.107028	1	0.322052	0.52928
...	...	...	...	...	...	...	...	...	...	...
21898	NVT	14.046063	3.323868	10.375857	16.665808	0.961013	0.245791	1	0.161232	0.10563
21909	ACA	9.576454	1.403616	7.254635	4.789071	0.398742	0.149931	1	0.249740	0.21434
21910	ACA	14.853770	1.954925	10.502084	6.726521	0.632161	0.136692	1	0.195574	0.18493
21911	ACA	11.213941	1.622492	8.145220	3.646947	0.529299	0.148553	1	0.217243	0.29868
21912	ACA	13.117192	1.929016	9.755082	2.640555	0.534262	0.145540	1	0.166693	0.47122

1865 rows × 29 columns

FILTERING COMMENTED OUT BELOW (raw)

In [9]:
```
selected_df.replace([np.inf, -np.inf], np.nan, inplace=True)
```

```
selected_df.fillna(0, inplace=True)
```

```
/var/folders/_8/vpmh2zyd23z9mnscjwzk9t1r0000gn/T/ipykernel_45215/2665936439.py:1: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/ind
exing.html#returning-a-view-versus-a-copy
  selected_df.replace([np.inf, -np.inf], np.nan, inplace=True)
/var/folders/_8/vpmh2zyd23z9mnscjwzk9t1r0000gn/T/ipykernel_45215/2665936439.py:3: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/ind
exing.html#returning-a-view-versus-a-copy
  selected_df.fillna(0, inplace=True)
```

selected_df = selected_df[(selected_df['Operating_Margin'] <= 2)& (selected_df['Operating_Margin'] >= 0.0)] selected_df = selected_df[(selected_df['EV/EBITDA'] <= 30)& (selected_df['EV/EBITDA'] >= 1)] selected_df = selected_df[(selected_df['EV_EBIT'] <= 30)& (selected_df['EV_EBIT'] >= 0)] selected_df = selected_df[(selected_df['EBIT Margin (%)'] <= 1) & (selected_df['EBIT Margin (%)'] >= 0)] selected_df = selected_df[(selected_df['Debt_to_Equity'] >= 0)] selected_df = selected_df[(selected_df['YOY_Operating_Margin_Change'] >= -.75) & (selected_df['YOY_Operating_Margin_Change'] <= 5)] selected_df = selected_df[(selected_df['Revenue per Employee'] <= 1200)] selected_df = selected_df[(selected_df['ROE'] <= 1.3)] # Display the reordered DataFrame selected_df.replace([np.inf, -np.inf], np.nan, inplace=True) selected_df.fillna(0, inplace=True) selected_df= selected_df[(filtered_data['Market_Cap'] >= 100)] selected_df.head()

In [10]:
```
selected_df.describe()
```

Out[10]:

	EV/EBITDA	EV/Sales	P/E	EV_EBIT	Debt_to_Equity	ROE	FCF_Positive	FCF Yield
count	1865.000000	1865.000000	1865.000000	1865.000000	1865.000000	1865.000000	1865.000000	1865.000000
mean	27.859417	2.369837	19.133466	27.361607	2.429292	0.169719	0.900268	0.274336
std	118.632207	2.800362	93.089335	316.553944	63.492889	11.603919	0.299723	0.449700
min	1.979820	0.189080	0.182750	-6509.207271	-2130.757991	-483.593607	0.000000	-8.210014
25%	10.338713	0.971582	5.218299	2.094525	0.706916	0.145914	1.000000	0.095800
50%	14.390089	1.634341	8.841054	7.448310	1.259737	0.238566	1.000000	0.208952
75%	20.935576	2.814968	14.413687	37.576915	2.355021	0.381787	1.000000	0.386244
max	3042.196552	47.323948	3071.858621	6556.449349	961.885321	75.535714	1.000000	6.303995

Grouping Together Small Industries

In [11]:
```
# Create a mapping for grouping divisions
division_mapping = {
    'Unknown': 'Unknown_Nonclassifiable',
    'Nonclassifiable': 'Unknown_Nonclassifiable',
    'Mining': 'Mining_Trans_Comm_Gas_Sanitary',
    'Transportation, Communications, Electric, Gas and Sanitary service': 'Mining_Trans_Comm_Gas_
}

# Apply the mapping to the Division column in your DataFrame
selected_df['Division'] = selected_df['Division'].replace(division_mapping)

# Check the updated distribution
division_counts = selected_df['Division'].value_counts()
print(division_counts)
```

```
Division
Manufacturing                          753
Retail Trade                           412
Services                               317
Construction                           119
Wholesale Trade                        103
Unknown_Nonclassifiable                 55
Finance, Insurance and Real Estate      55
Mining_Trans_Comm_Gas_Sanitary          51
Name: count, dtype: int64
```

```
/var/folders/_8/vpmh2zyd23z9mnscjwzk9t1r0000gn/T/ipykernel_45215/487303733.py:10: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/ind
exing.html#returning-a-view-versus-a-copy
  selected_df['Division'] = selected_df['Division'].replace(division_mapping)
```

Corrrelations between Multiples various financial metrics... By Industry

- Multiples of Choice:

took this out, but kept it incase we wanted later... Getting exact Correlations

- above weak correlations... >0.3
- correlations between multiples and metrics across industries

Define the multiples and metrics multiples = ['EV/EBITDA','EV/Sales', 'P/E', 'EV/GP','EV_EBIT','FCF Yield','EV/EBITDA-Capex'] metrics = ['Debt_Coverage_Ratio', 'ROE', 'EBIT Margin (%)', 'YOY_Gross_Profit_Change','ROIC','FCF_Positive', 'Gross_Margin','Quick_Ratio','Current_Ratio','Debt_to_Equity', 'Revenue per Employee','FCF_Positive','YOY_Revenue_Change','YOY_EBIT_Change'] # Group data by sector grouped_data = selected_df.groupby('Division') # Calculate correlation between multiples and metrics for each sector correlation_results = {} for sector, group in grouped_data: correlation_results[sector] = {} for multiple in multiples: for metric in metrics: correlation = group[multiple].corr(group[metric]) # Only store correlations with absolute value higher than 0.3 if abs(correlation) > 0.3: correlation_results[sector][(multiple, metric)] = correlation # Display correlation results for sector, result in correlation_results.items(): print(f"---****** {sector} ******---") for (multiple, metric), correlation in result.items(): print(f"{multiple} vs {metric}: {correlation}")

Counting number of non-weak correlations across industries

In [12]:
```
multiples = ['EV/EBITDA','EV/Sales', 'P/E', 'EV/GP','EV_EBIT','FCF Yield','EV/EBITDA-Capex']
metrics = ['Debt_Coverage_Ratio', 'ROE', 'EBIT Margin (%)',
           'YOY_Gross_Profit_Change','ROIC','FCF_Positive',
           'Gross_Margin','Quick_Ratio','Current_Ratio','Debt_to_Equity',
           'Revenue per Employee','FCF_Positive','YOY_Revenue_Change','YOY_EBIT_Change']

# Group data by sector
grouped_data = selected_df.groupby('Division')

# Calculate correlation between multiples and metrics for each sector
correlation_counts = {multiple: 0 for multiple in multiples}
for sector, group in grouped_data:
    print(f"---****** {sector} ******---")
    for multiple in multiples:
        for metric in metrics:
            correlation = group[multiple].corr(group[metric])
            if abs(correlation) > 0.3:
                correlation_counts[multiple] += 1
                print(f"{multiple} vs {metric}: {correlation}")
```

```python
# Display the count of correlations with absolute value greater than 0.3 for each multiple
print("\nNumber of correlations with |value| > 0.3 for each multiple:")
for multiple, count in correlation_counts.items():
    print(f"{multiple}: {count}")
```

---****** Construction ******---
EV/EBITDA vs Debt_Coverage_Ratio: -0.6000929319263598
EV/EBITDA vs EBIT Margin (%): 0.35001681850772576
EV/EBITDA vs YOY_Gross_Profit_Change: -0.33318687031519634
EV/EBITDA vs FCF_Positive: -0.47947432436965204
EV/EBITDA vs Gross_Margin: -0.4769550463288401
EV/EBITDA vs Quick_Ratio: 0.49063103814358827
EV/EBITDA vs Current_Ratio: 0.506207182171044
EV/EBITDA vs Revenue per Employee: -0.4874965658994136
EV/EBITDA vs FCF_Positive: -0.47947432436965204
EV/Sales vs FCF_Positive: -0.35958153102039403
EV/Sales vs Gross_Margin: 0.31023406006415466
EV/Sales vs Quick_Ratio: 0.4261318235921708
EV/Sales vs Current_Ratio: 0.44857638865683824
EV/Sales vs FCF_Positive: -0.35958153102039403
P/E vs Debt_Coverage_Ratio: -0.35566967270768346
P/E vs Quick_Ratio: 0.5186692321997179
P/E vs Current_Ratio: 0.5392367490483434
P/E vs Revenue per Employee: -0.4443995177927245
EV/GP vs Debt_Coverage_Ratio: -0.4739965864330813
EV/GP vs EBIT Margin (%): 0.46254802781529586
EV/GP vs FCF_Positive: -0.37718704679738924
EV/GP vs Gross_Margin: -0.41640354863259493
EV/GP vs Quick_Ratio: 0.5206659189843458
EV/GP vs Current_Ratio: 0.5325836557828691
EV/GP vs Revenue per Employee: -0.4359446187730493
EV/GP vs FCF_Positive: -0.37718704679738924
EV_EBIT vs EBIT Margin (%): -0.4643636760723289
EV_EBIT vs ROIC: -0.39316318095216746
EV_EBIT vs Gross_Margin: 0.5324592943715856
---****** Finance, Insurance and Real Estate ******---
EV/Sales vs EBIT Margin (%): 0.4072335956639599
EV/Sales vs Gross_Margin: 0.40573292372981284
EV/Sales vs Revenue per Employee: 0.4249513103479582
P/E vs Debt_Coverage_Ratio: 0.49382650190735733
P/E vs ROIC: 0.588071796119638
P/E vs Quick_Ratio: 0.6648632305940008
P/E vs Current_Ratio: 0.6761439253808492
P/E vs Debt_to_Equity: -0.3495162694999225
EV/GP vs Debt_Coverage_Ratio: -0.3215224235622979
EV/GP vs EBIT Margin (%): 0.34964452178916194
EV/GP vs FCF_Positive: -0.33200742695889623
EV/GP vs FCF_Positive: -0.33200742695889623
EV_EBIT vs ROE: -0.49746587793991054
EV_EBIT vs EBIT Margin (%): -0.3219454668105717
EV_EBIT vs FCF_Positive: -0.3177693039142977
EV_EBIT vs Debt_to_Equity: -0.5020025078977257
EV_EBIT vs FCF_Positive: -0.3177693039142977
FCF Yield vs FCF_Positive: 0.5361824903152305
FCF Yield vs FCF_Positive: 0.5361824903152305
EV/EBITDA-Capex vs FCF_Positive: -0.6285387785610335
EV/EBITDA-Capex vs Gross_Margin: -0.3184065429472358
EV/EBITDA-Capex vs FCF_Positive: -0.6285387785610335
---****** Manufacturing ******---
EV/Sales vs Gross_Margin: 0.47315647353470075
EV/Sales vs Quick_Ratio: 0.375629736517443
EV/Sales vs Current_Ratio: 0.3298182637813385
EV/GP vs EBIT Margin (%): 0.30786253673143
---****** Mining_Trans_Comm_Gas_Sanitary ******---
EV/EBITDA vs Debt_Coverage_Ratio: -0.4824422865850665
EV/EBITDA vs ROE: -0.3926828916959983
EV/EBITDA vs Gross_Margin: -0.37731376748812717
EV/Sales vs EBIT Margin (%): 0.6077614567806208
EV/Sales vs FCF_Positive: -0.6441149501727796
EV/Sales vs Gross_Margin: 0.5163282222863625
EV/Sales vs Quick_Ratio: 0.39232980742446144
EV/Sales vs Current_Ratio: 0.3828174850245315

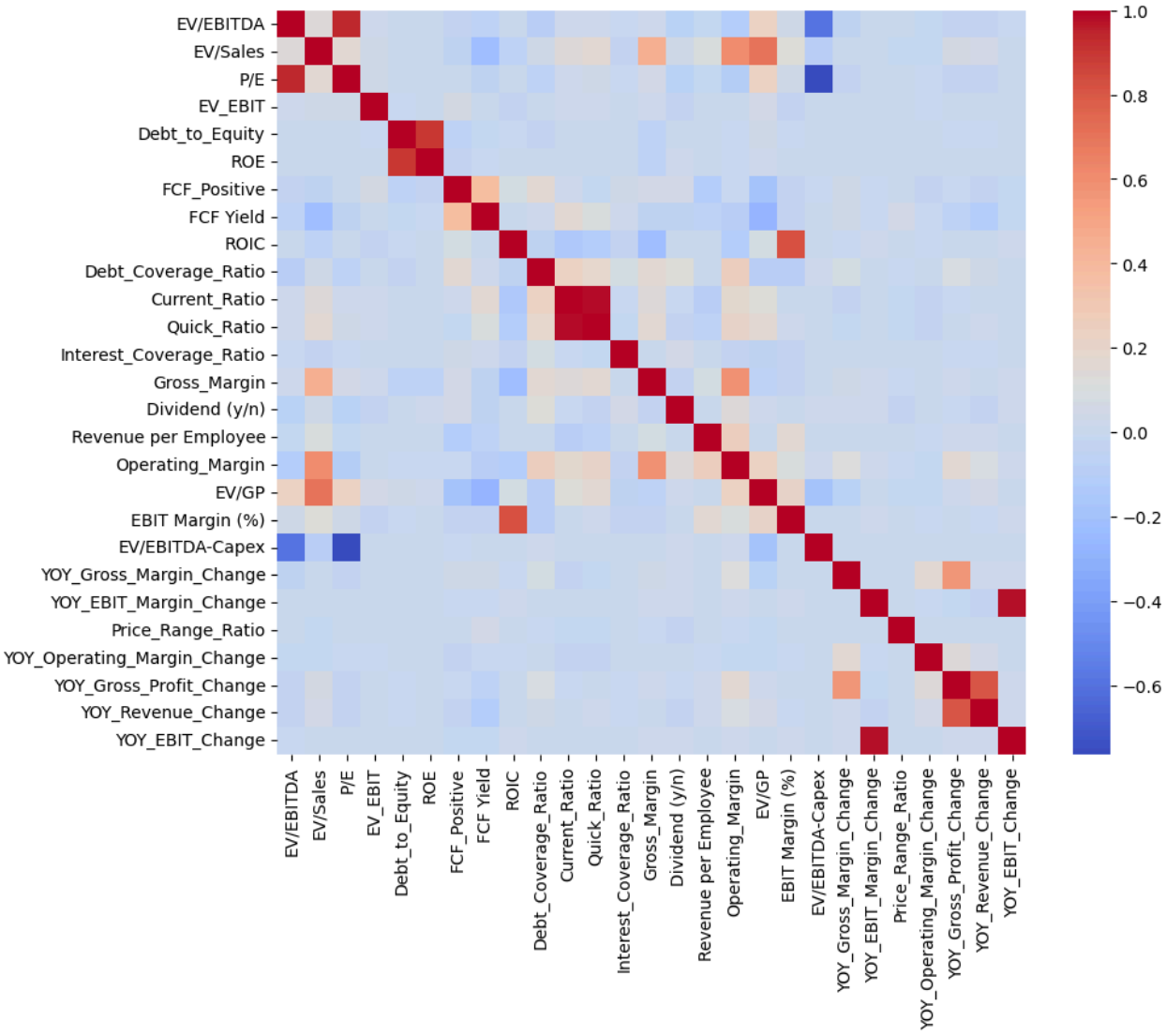
EV/Sales vs Revenue per Employee: 0.4400611459686633
EV/Sales vs FCF_Positive: -0.6441149501727796
P/E vs ROE: -0.33798357686992797
P/E vs EBIT Margin (%): -0.33847684735519656
P/E vs Gross_Margin: -0.41757968365296927
EV/GP vs Debt_Coverage_Ratio: -0.42518085057926747
EV/GP vs ROE: -0.32416175313924833
EV/GP vs FCF_Positive: -0.33190306486991883
EV/GP vs Quick_Ratio: 0.409261581403271
EV/GP vs Current_Ratio: 0.4030288483038599
EV/GP vs FCF_Positive: -0.33190306486991883
EV_EBIT vs EBIT Margin (%): -0.4990794498622054
EV_EBIT vs ROIC: -0.6461684677192694
EV_EBIT vs YOY_Revenue_Change: 0.38586634136578657
FCF Yield vs ROE: 0.33175545259392675
FCF Yield vs FCF_Positive: 0.48267837568194955
FCF Yield vs Gross_Margin: -0.3344793302629631
FCF Yield vs Debt_to_Equity: 0.46576485217704566
FCF Yield vs FCF_Positive: 0.48267837568194955
---****** Retail Trade ******---
EV/Sales vs Quick_Ratio: 0.3969671466400984
EV/Sales vs Current_Ratio: 0.3695403829630308
EV/GP vs Quick_Ratio: 0.38590521093685054
EV/GP vs Current_Ratio: 0.3457459826274817
EV/GP vs YOY_Revenue_Change: 0.3101163756126926
FCF Yield vs FCF_Positive: 0.5299479246636398
FCF Yield vs Gross_Margin: 0.3645724850543398
FCF Yield vs FCF_Positive: 0.5299479246636398
---****** Services ******---
EV/Sales vs Gross_Margin: 0.4731547638507673
EV/GP vs Gross_Margin: -0.4167166313810165
FCF Yield vs FCF_Positive: 0.5032062134275097
FCF Yield vs FCF_Positive: 0.5032062134275097
---****** Unknown_Nonclassifiable ******---
EV/EBITDA vs ROIC: 0.3255114532985711
EV/Sales vs EBIT Margin (%): 0.45466177111075257
EV/Sales vs Gross_Margin: 0.4693685847070018
EV/Sales vs Quick_Ratio: 0.33874843656850345
EV/Sales vs Current_Ratio: 0.3173205099955567
EV/Sales vs Revenue per Employee: 0.4450658115018997
P/E vs ROE: -0.41337002749511725
P/E vs ROIC: 0.3682350207727263
P/E vs Debt_to_Equity: -0.39196568308818286
EV_EBIT vs EBIT Margin (%): -0.34340648586269
EV_EBIT vs ROIC: -0.33434773354900466
FCF Yield vs FCF_Positive: 0.3816102903713165
FCF Yield vs Gross_Margin: 0.5366637542457964
FCF Yield vs Quick_Ratio: 0.7609264456310928
FCF Yield vs Current_Ratio: 0.7618725473696331
FCF Yield vs Revenue per Employee: 0.5041398526642962
FCF Yield vs FCF_Positive: 0.3816102903713165
---****** Wholesale Trade ******---
EV/EBITDA vs Debt_Coverage_Ratio: -0.39791502813862795
EV/Sales vs Debt_Coverage_Ratio: 0.4442140551848722
EV/Sales vs Gross_Margin: 0.6200023702544293
EV/Sales vs Revenue per Employee: -0.38442918167767576
EV/GP vs Quick_Ratio: 0.48117433751816224
EV/GP vs Current_Ratio: 0.4141225146313788
FCF Yield vs Revenue per Employee: 0.4770285322220833

Number of correlations with |value| > 0.3 for each multiple:
EV/EBITDA: 14
EV/Sales: 29
P/E: 15
EV/GP: 25
EV_EBIT: 13

```
FCF Yield: 19
EV/EBITDA-Capex: 3
```

Try to Identify Multiple of choice from above and below

In [13]:
```
import seaborn as sns
import matplotlib.pyplot as plt
# Correlation Analysis

columns_to_drop = ['Ticker'] + ['Division']

# Create a new dataframe without the specified columns
corr_df = selected_df.drop(columns=columns_to_drop)
corr = corr_df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(corr, cmap='coolwarm', annot=False)
plt.show()
```

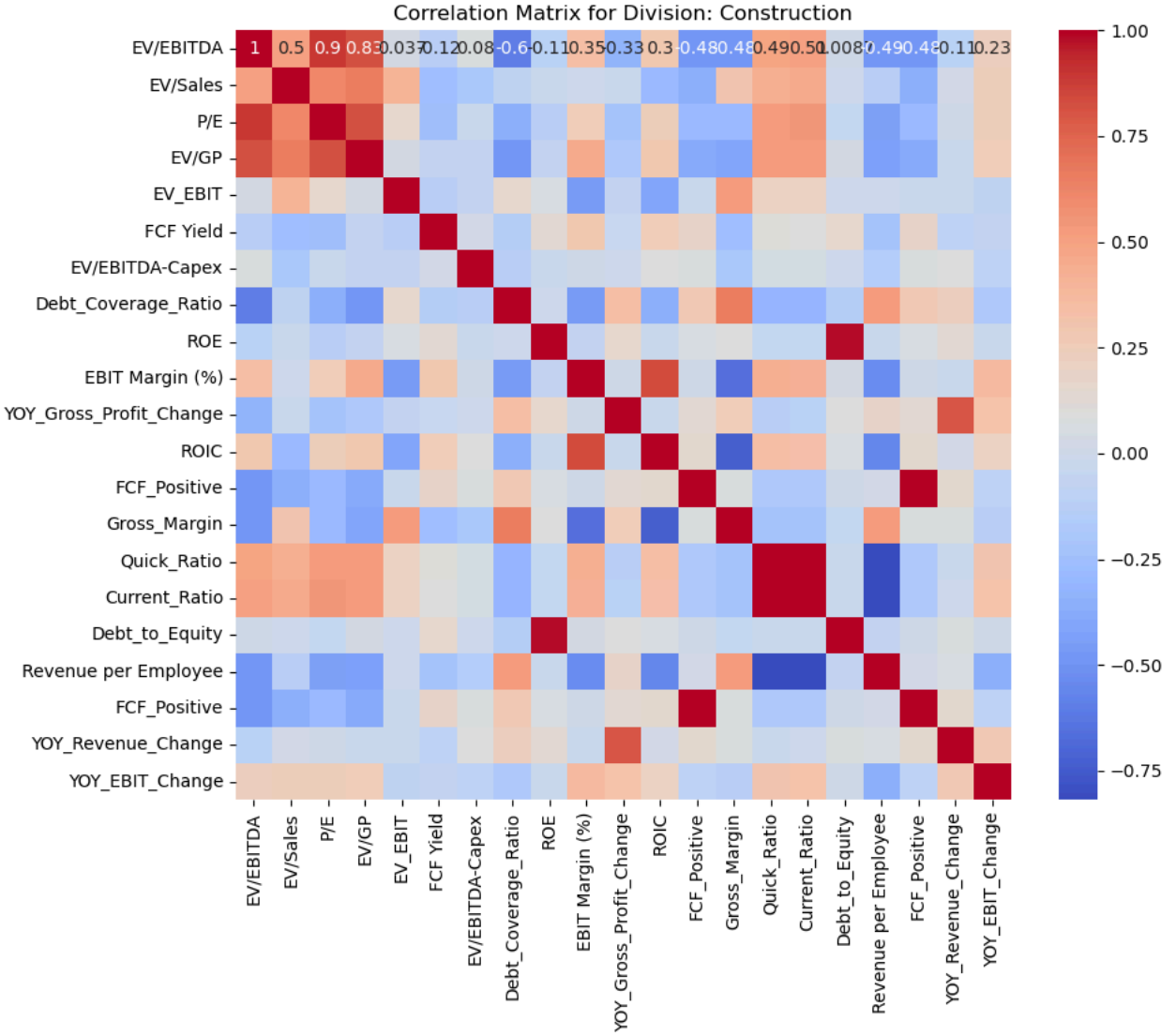


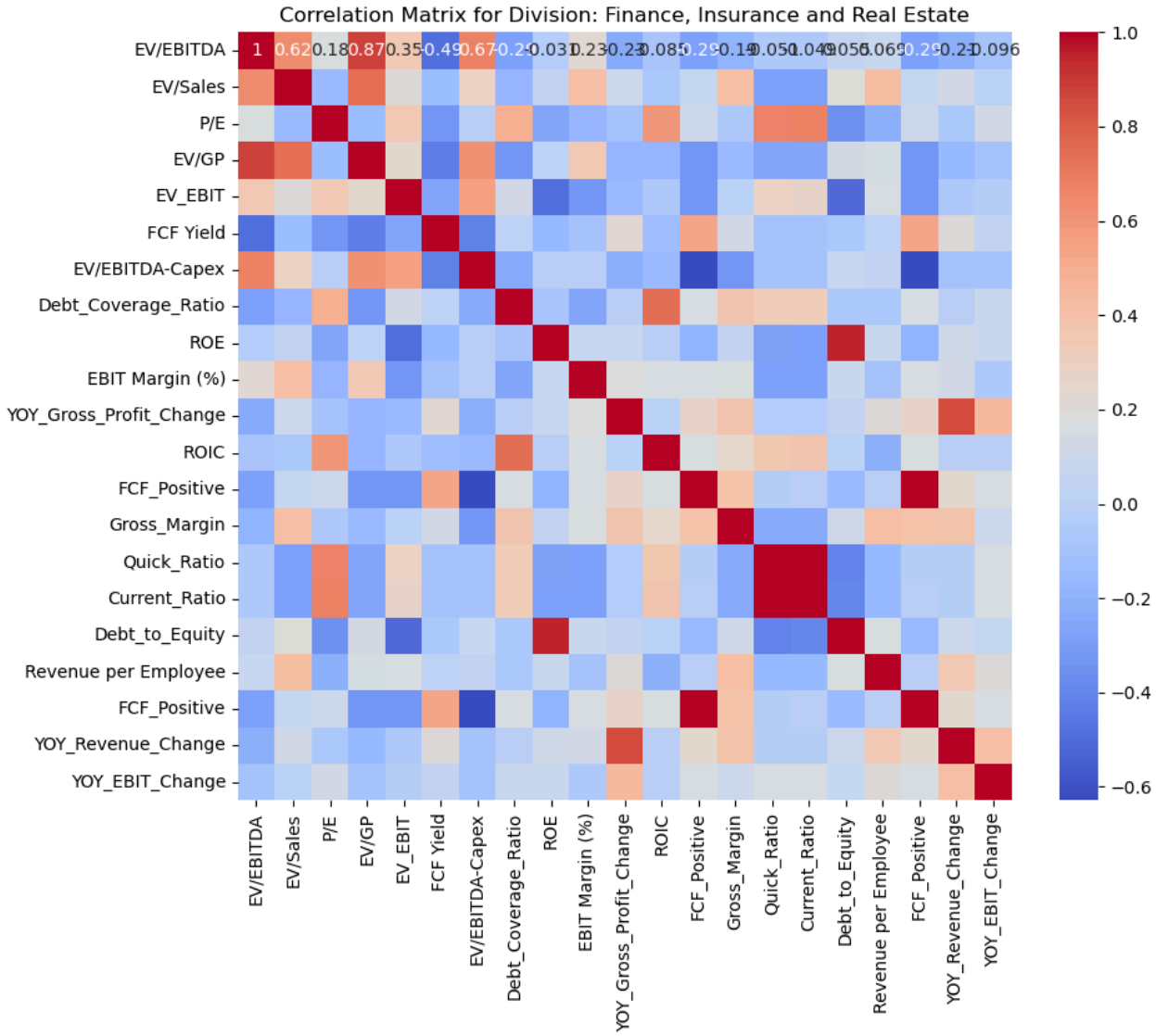
Visualize the correlations across industries... see if significant differences

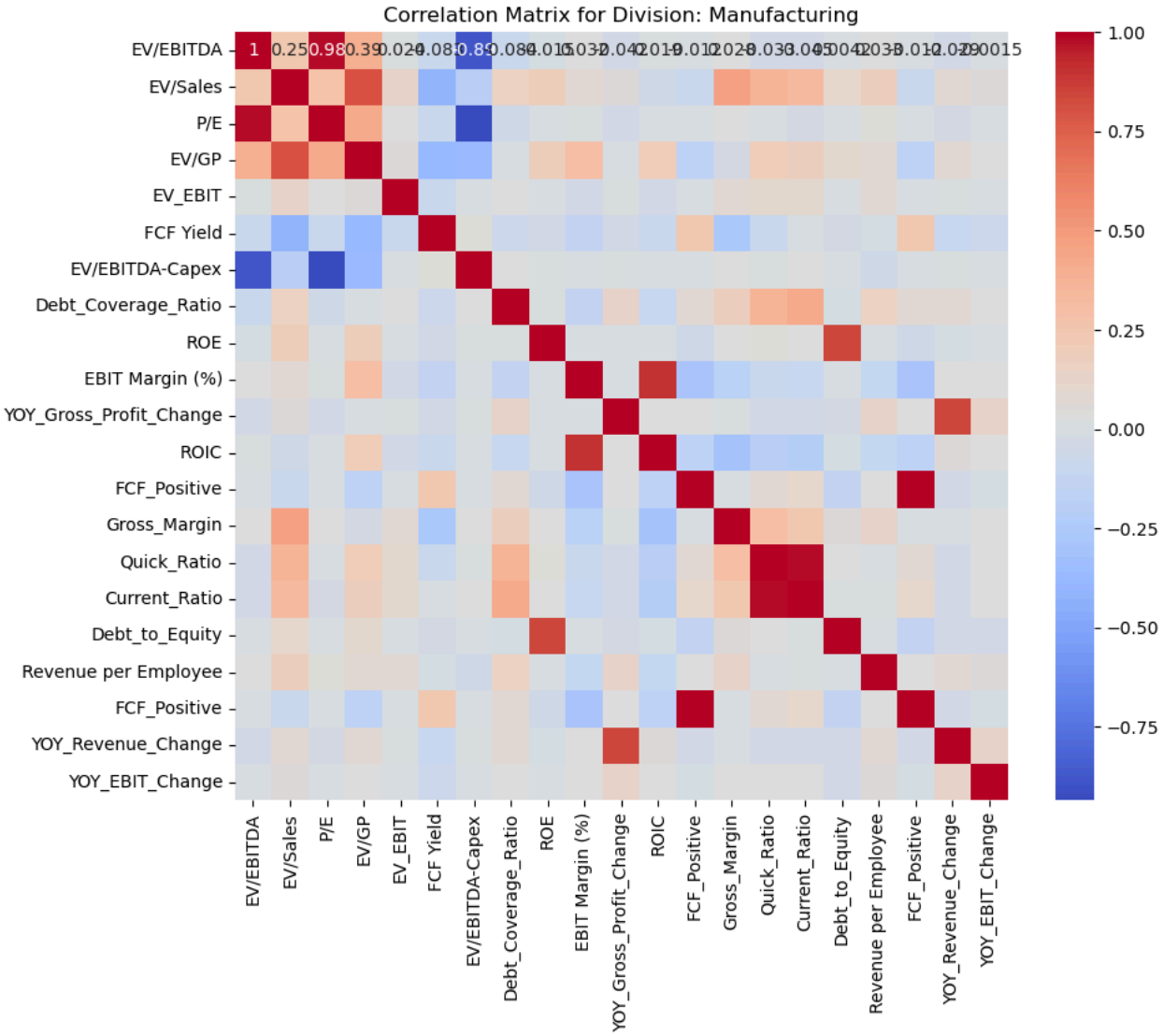
- we may end up focusing in on the most populated industry

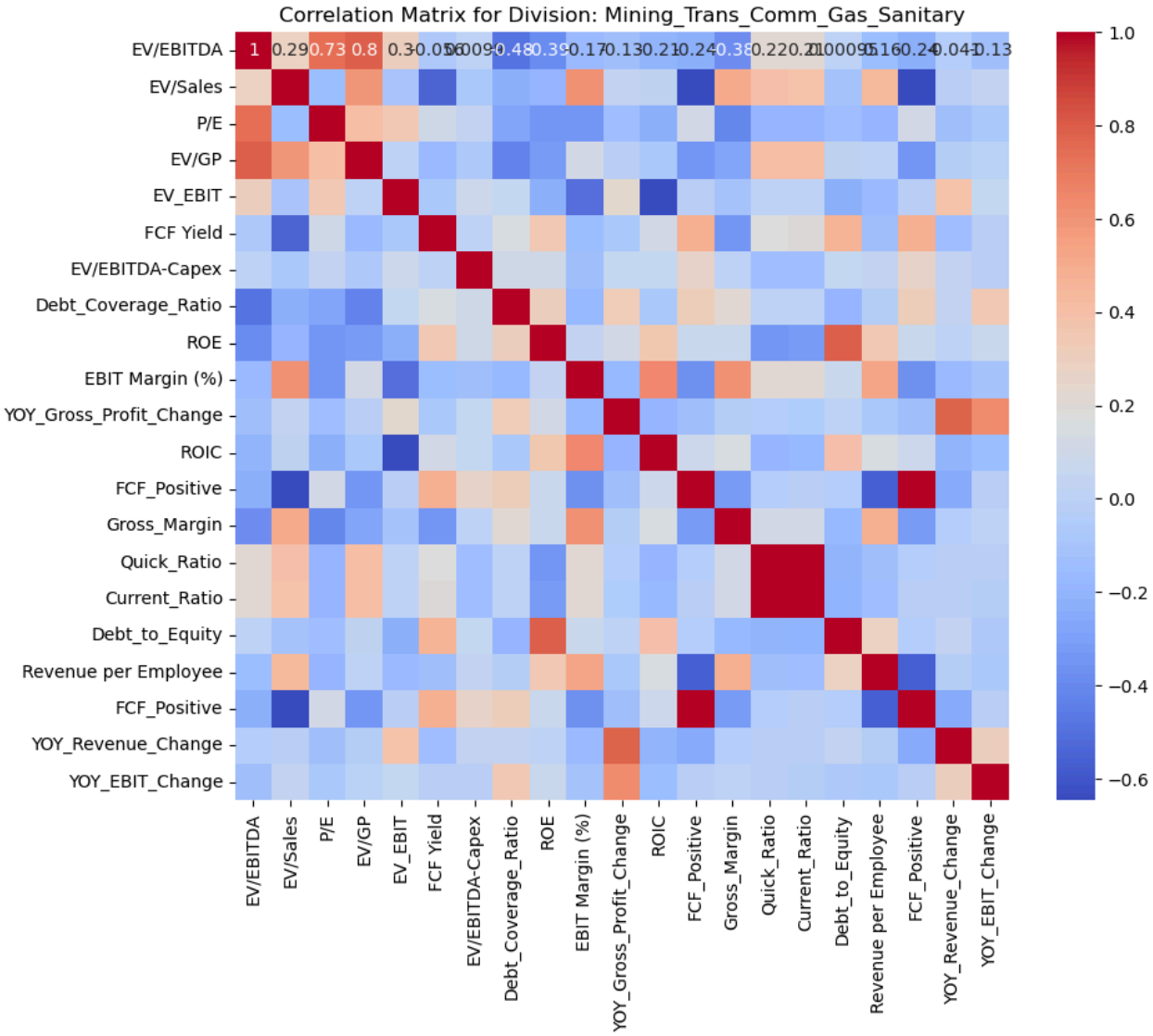
In [14]:

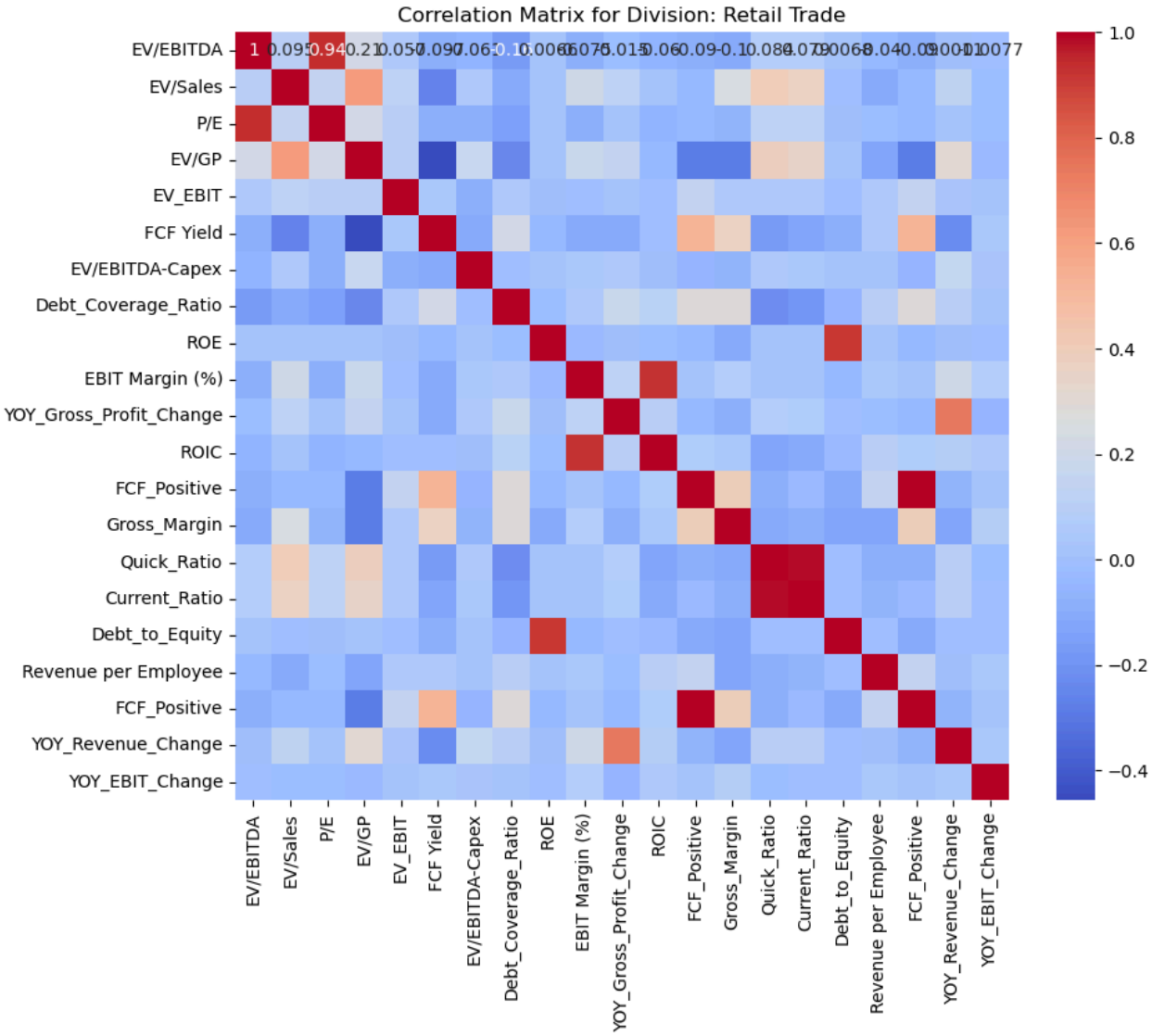
```
# Define the multiples and metrics
multiples = ['EV/EBITDA', 'EV/Sales', 'P/E', 'EV/GP','EV_EBIT','FCF Yield','EV/EBITDA-Capex']
metrics = ['Debt_Coverage_Ratio', 'ROE', 'EBIT Margin (%)',
           'YOY_Gross_Profit_Change','ROIC','FCF_Positive',
           'Gross_Margin','Quick_Ratio','Current_Ratio','Debt_to_Equity',
           'Revenue per Employee','FCF_Positive','YOY_Revenue_Change','YOY_EBIT_Change']

# Drop the specified non-numeric columns
columns_to_drop = ['Ticker', 'Division']

# Iterate over each unique Division in the dataset
grouped_data = selected_df.groupby('Division')
for division, group in grouped_data:
    # Drop the non-numeric columns
    group = group.drop(columns=columns_to_drop, errors='ignore')

    # Keep only the columns that are multiples or metrics
    subset = group[multiples + metrics].select_dtypes(include='number')

    # Calculate the correlation matrix
    corr = subset.corr()

    # Plot the correlation matrix using seaborn heatmap
    plt.figure(figsize=(10, 8))
    sns.heatmap(corr, cmap='coolwarm', annot=True)
    plt.title(f'Correlation Matrix for Division: {division}')
    plt.show()
```

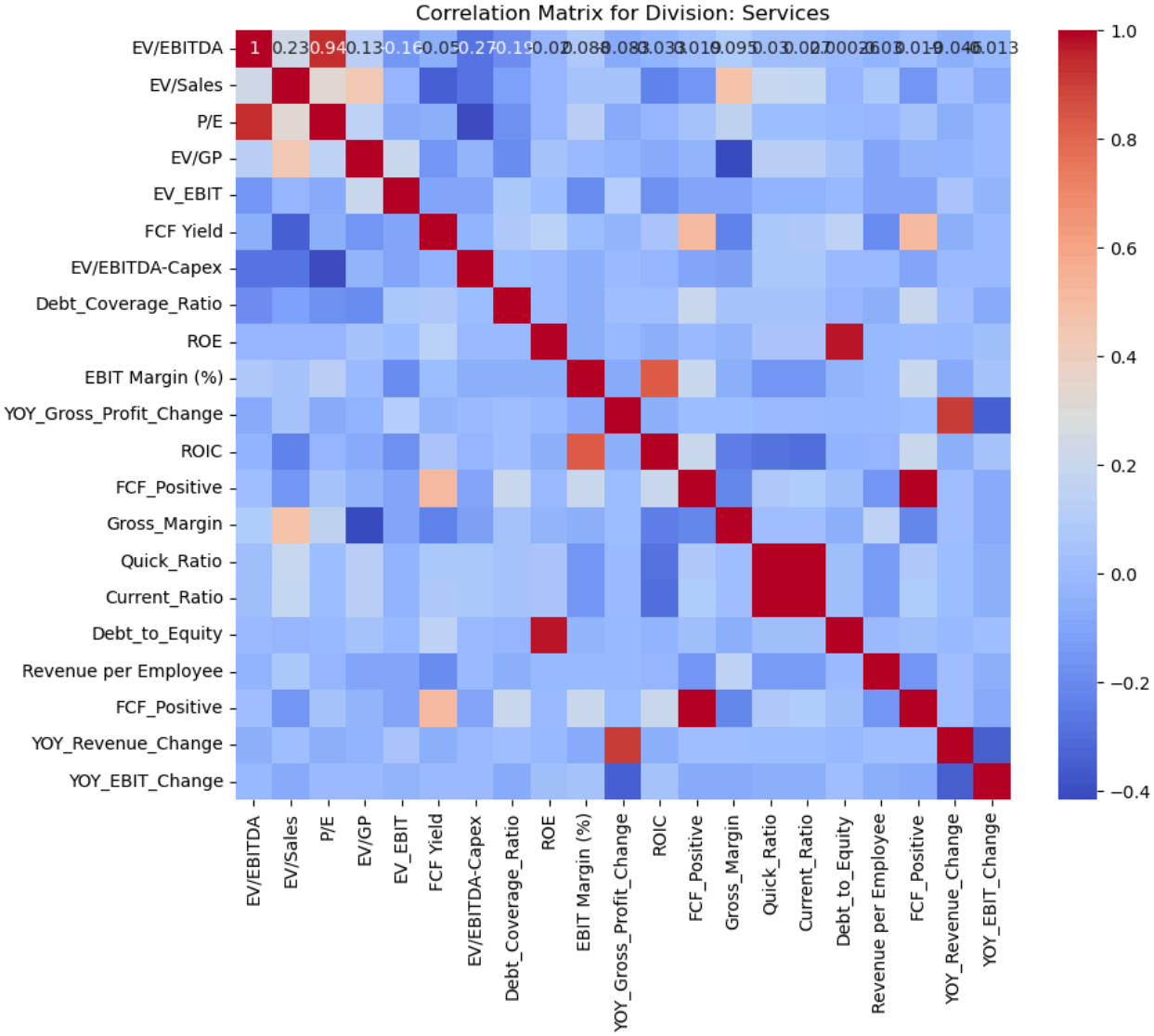


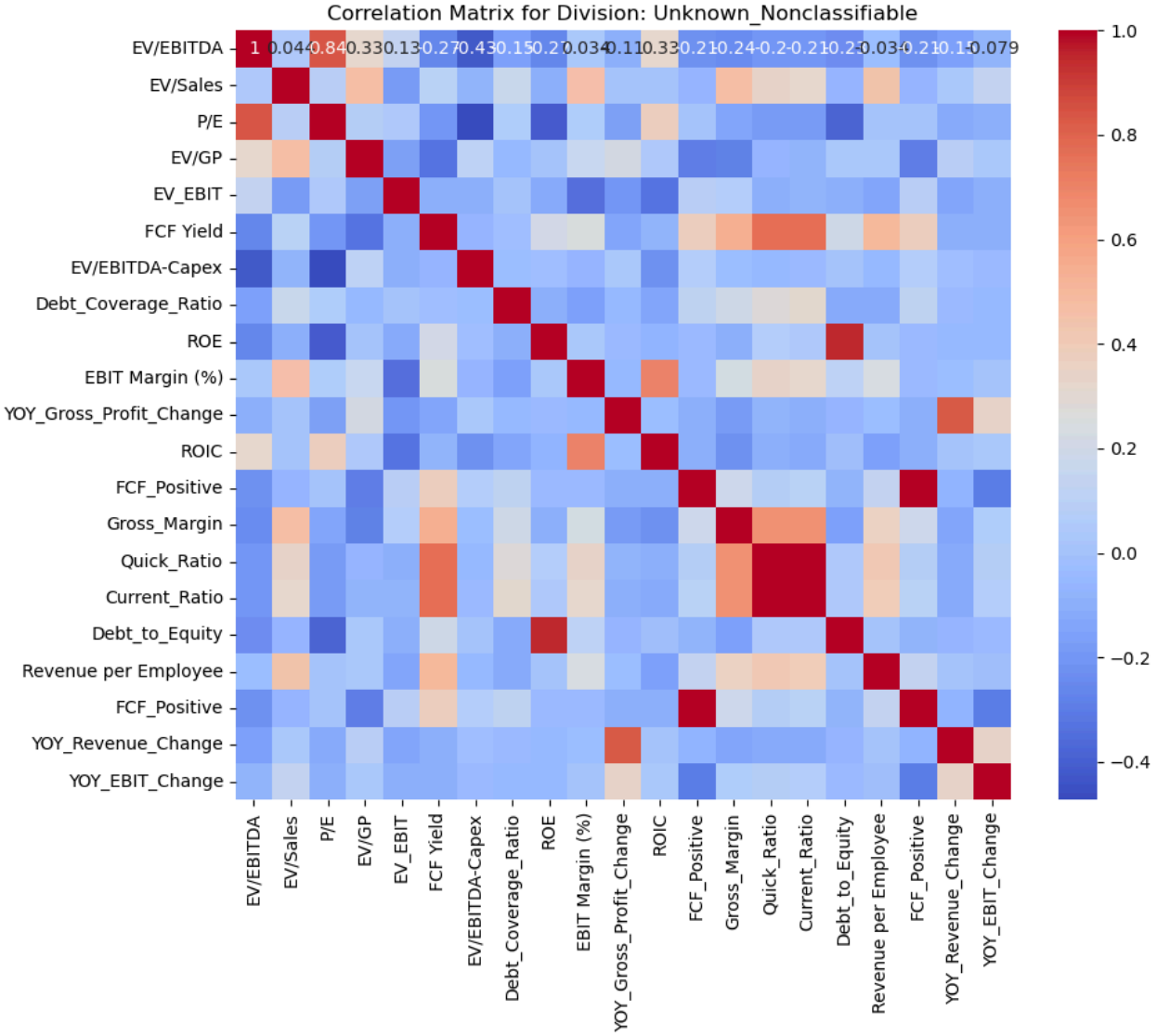

Correlation Matrix for Division: Finance, Insurance and Real Estate
EV/EBITDA
EV/Sales
P/E
EV/GP
EV_EBIT
FCF Yield
EV/EBITDA-Capex
Debt_Coverage_Ratio
ROE
EBIT Margin (%)
YOY_Gross_Profit_Change
ROIC
FCF_Positive
Gross_Margin
Quick_Ratio
Current_Ratio
Debt_to_Equity
Revenue per Employee
FCF_Positive
YOY_Revenue_Change
YOY_EBIT_Change
1
0.62
0.18
0.87
0.35
-0.49
0.67
-0.23
-0.031
0.23
-0.23
0.085
-0.29
-0.19
-0.051
-0.049
-0.056
0.069
-0.29
-0.23
0.096
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

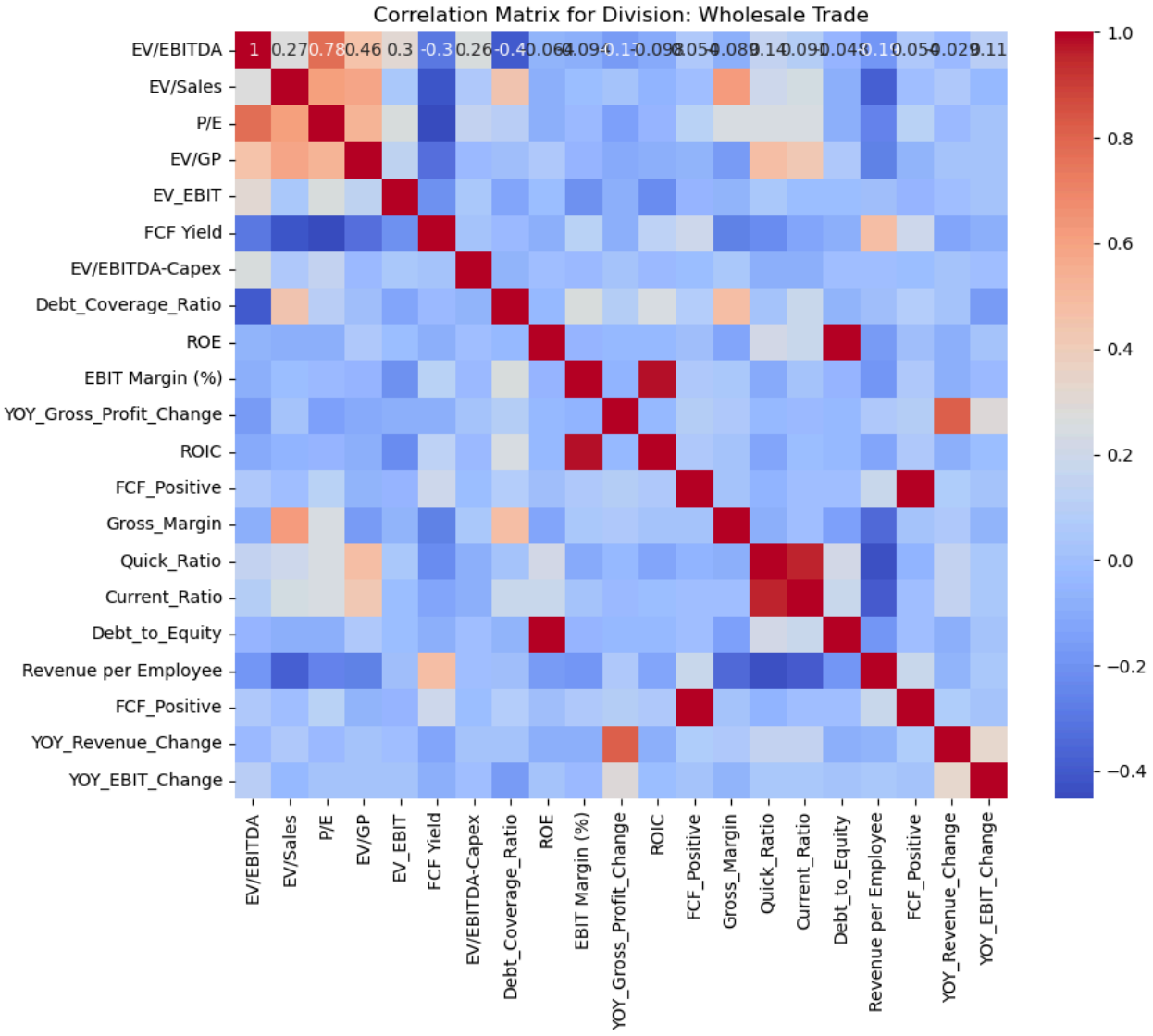

Correlation Matrix for Division: Manufacturing
EV/EBITDA
EV/Sales
P/E
EV/GP
EV_EBIT
FCF Yield
EV/EBITDA-Capex
Debt_Coverage_Ratio
ROE
EBIT Margin (%)
YOY_Gross_Profit_Change
ROIC
FCF_Positive
Gross_Margin
Quick_Ratio
Current_Ratio
Debt_to_Equity
Revenue per Employee
FCF_Positive
YOY_Revenue_Change
YOY_EBIT_Change
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75


Correlation Matrix for Division: Mining_Trans_Comm_Gas_Sanitary
EV/EBITDA
EV/Sales
P/E
EV/GP
EV_EBIT
FCF Yield
EV/EBITDA-Capex
Debt_Coverage_Ratio
ROE
EBIT Margin (%)
YOY_Gross_Profit_Change
ROIC
FCF_Positive
Gross_Margin
Quick_Ratio
Current_Ratio
Debt_to_Equity
Revenue per Employee
FCF_Positive
YOY_Revenue_Change
YOY_EBIT_Change
1
0.29
0.73
0.8
0.3
-0.05
-0.48
-0.17
-0.13
-0.21
-0.24
-0.38
0.22
0.24
0.16
0.24
-0.13
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6


Correlation Matrix for Division: Retail Trade
EV/EBITDA
EV/Sales
P/E
EV/GP
EV_EBIT
FCF Yield
EV/EBITDA-Capex
Debt_Coverage_Ratio
ROE
EBIT Margin (%)
YOY_Gross_Profit_Change
ROIC
FCF_Positive
Gross_Margin
Quick_Ratio
Current_Ratio
Debt_to_Equity
Revenue per Employee
FCF_Positive
YOY_Revenue_Change
YOY_EBIT_Change
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4


Correlation Matrix for Division: Services
EV/EBITDA
EV/Sales
P/E
EV/GP
EV_EBIT
FCF Yield
EV/EBITDA-Capex
Debt_Coverage_Ratio
ROE
EBIT Margin (%)
YOY_Gross_Profit_Change
ROIC
FCF_Positive
Gross_Margin
Quick_Ratio
Current_Ratio
Debt_to_Equity
Revenue per Employee
FCF_Positive
YOY_Revenue_Change
YOY_EBIT_Change
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4


Correlation Matrix for Division: Unknown_Nonclassifiable
EV/EBITDA
EV/Sales
P/E
EV/GP
EV_EBIT
FCF Yield
EV/EBITDA-Capex
Debt_Coverage_Ratio
ROE
EBIT Margin (%)
YOY_Gross_Profit_Change
ROIC
FCF_Positive
Gross_Margin
Quick_Ratio
Current_Ratio
Debt_to_Equity
Revenue per Employee
FCF_Positive
YOY_Revenue_Change
YOY_EBIT_Change
1 0.04 0.84 0.33 0.13 0.27 0.43 0.15 0.27 0.03 0.11 0.33 0.21 0.24 0.2 0.21 0.25 0.03 0.21 0.1 0.079
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4



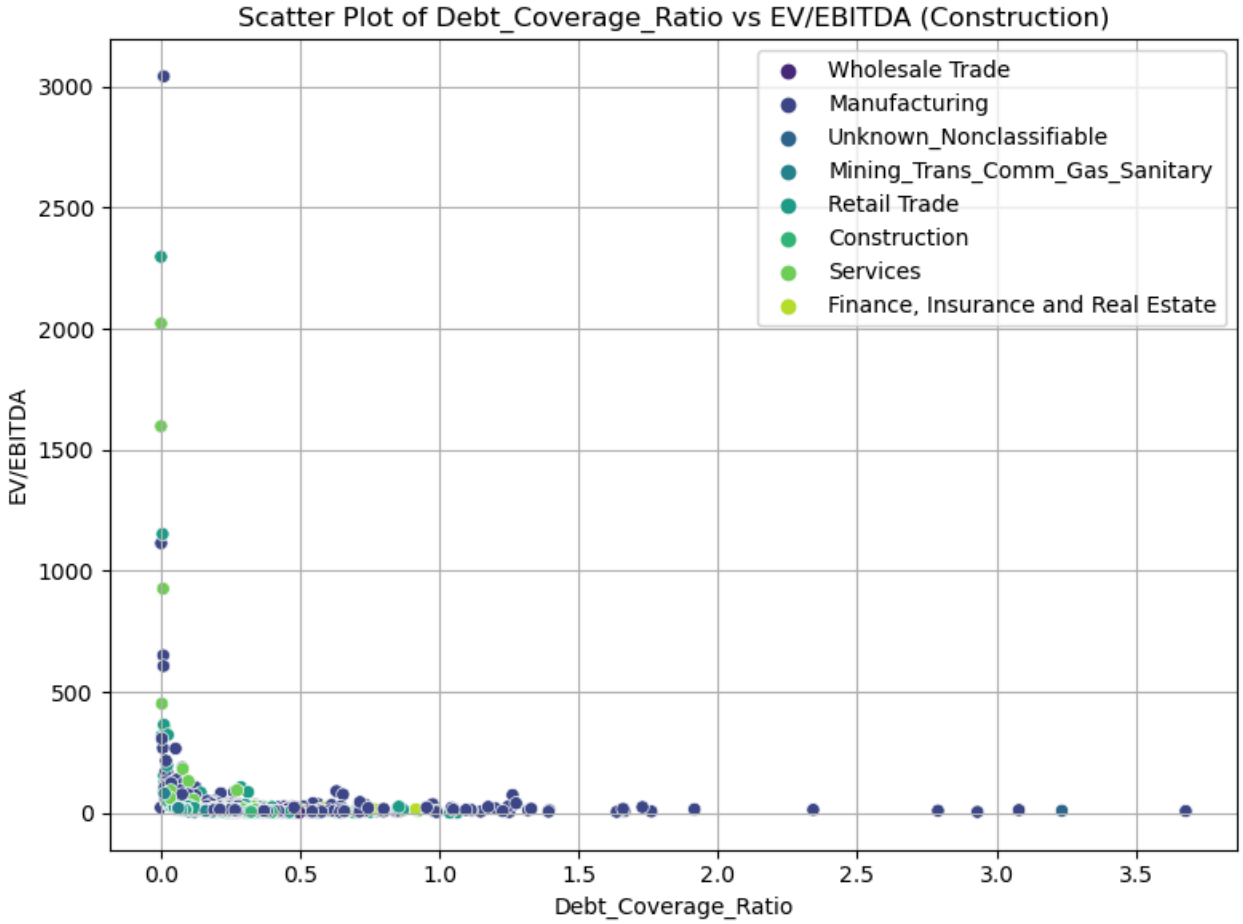
Creating Scatter Plots between selected Multiple of Interest vs a Financial Metric

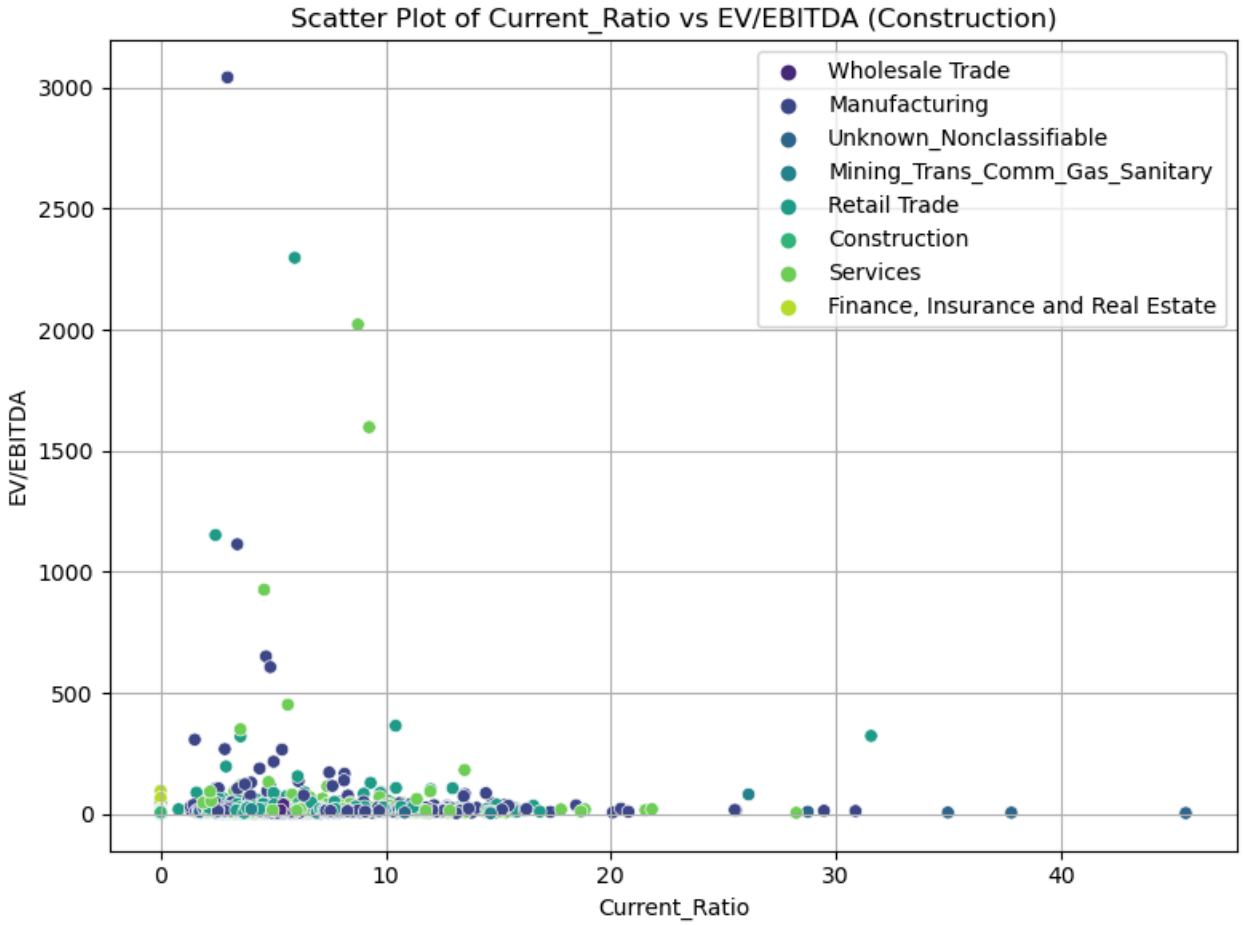
In [15]:

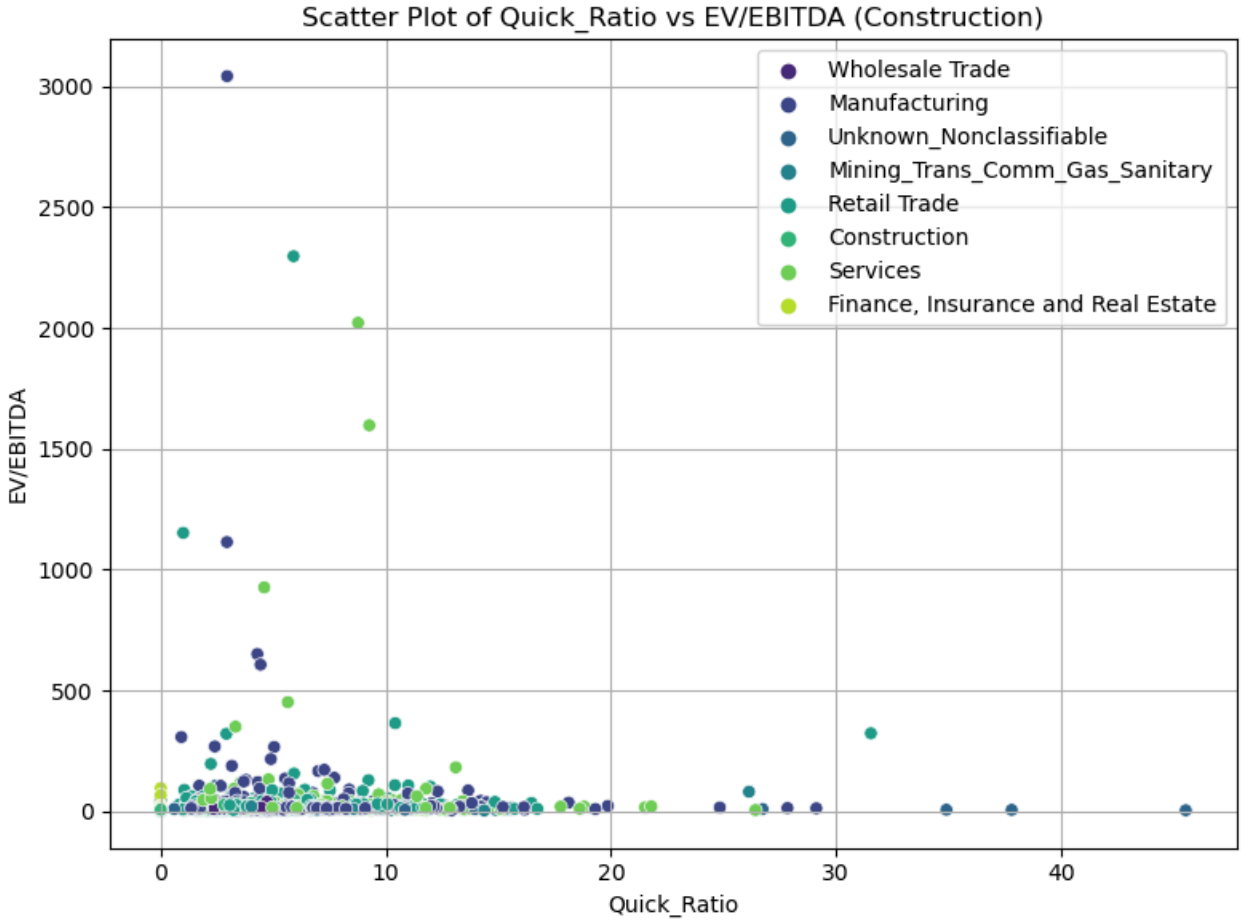
```
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import linregress

# Define the multiples and metrics
multiples = ['EV/EBITDA','EV/Sales']  #multiples = ['EV/EBITDA', 'P/E', 'EV/GP','EV_EBIT','FCF Yi
metrics = ['Debt_Coverage_Ratio', 'ROE', 'EBIT Margin (%)',
           'YOY_Gross_Profit_Change','ROIC','FCF_Positive',
           'Gross_Margin','Quick_Ratio','Current_Ratio','Debt_to_Equity',
           'Revenue per Employee','FCF_Positive','YOY_Revenue_Change','YOY_EBIT_Change']

# Group data by sector
grouped_data = selected_df.groupby('Division')

# Calculate correlation between multiples and metrics for each sector
correlation_results = {}
for sector, group in grouped_data:
    correlation_results[sector] = {}
    for multiple in multiples:
        for metric in metrics:
            correlation = group[multiple].corr(group[metric])
            correlation_results[sector][(multiple, metric)] = correlation
```

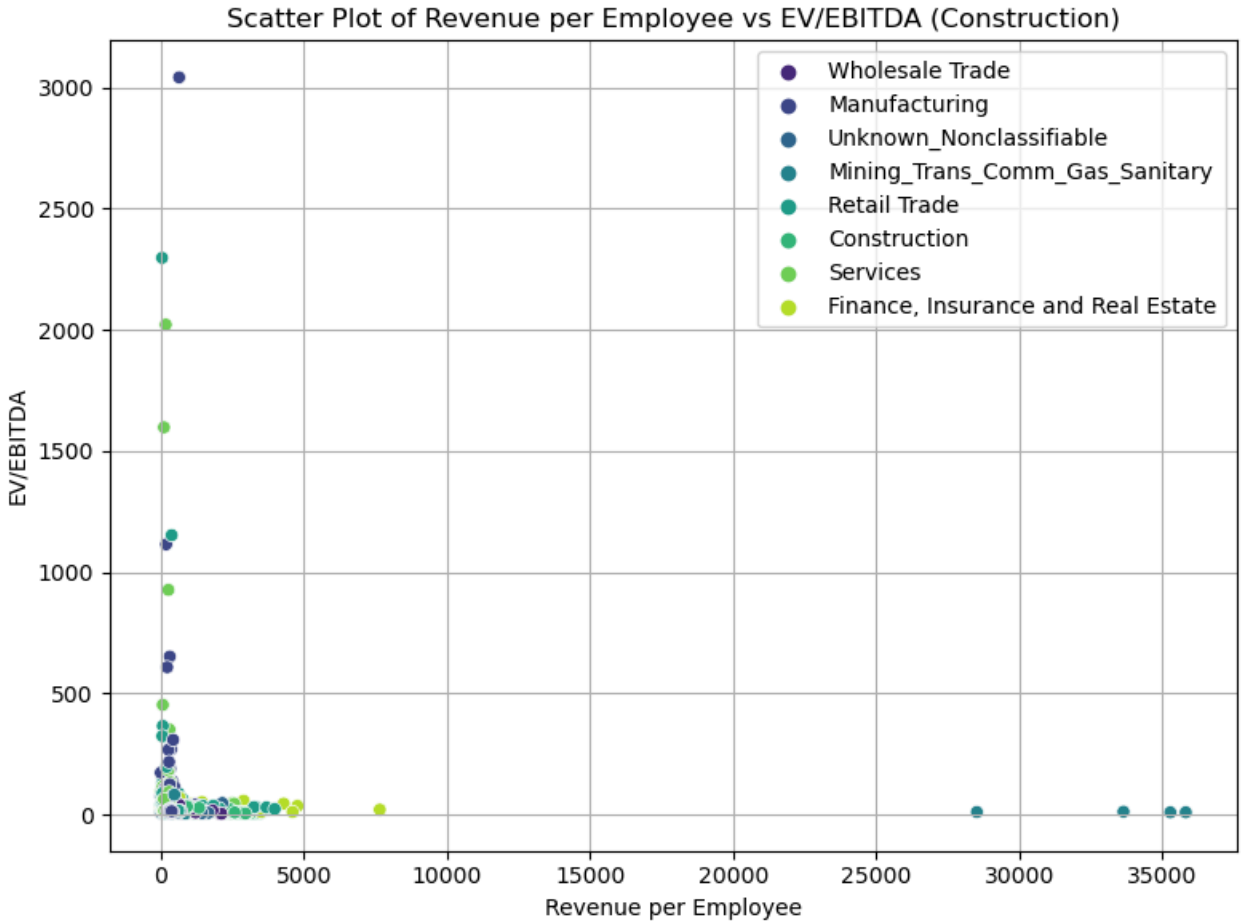
```python
# Sort correlation results and select top 5 correlations
top_correlations = {}
for sector, result in correlation_results.items():
    top_correlations[sector] = sorted(result.items(), key=lambda x: abs(x[1]), reverse=True)[:5]

# Plot scatter plots for top 5 correlations
for sector, correlations in top_correlations.items():
    print(f"--- {sector} ---")
    for (multiple, metric), correlation in correlations:
        plt.figure(figsize=(8, 6))
        sns.scatterplot(data=selected_df, x=metric, y=multiple, hue='Division', palette='viridis'
        plt.title(f'Scatter Plot of {metric} vs {multiple} ({sector})')
        plt.xlabel(metric)
        plt.ylabel(multiple)
        plt.legend()
        plt.grid(True)
        plt.tight_layout()
        plt.show()
```

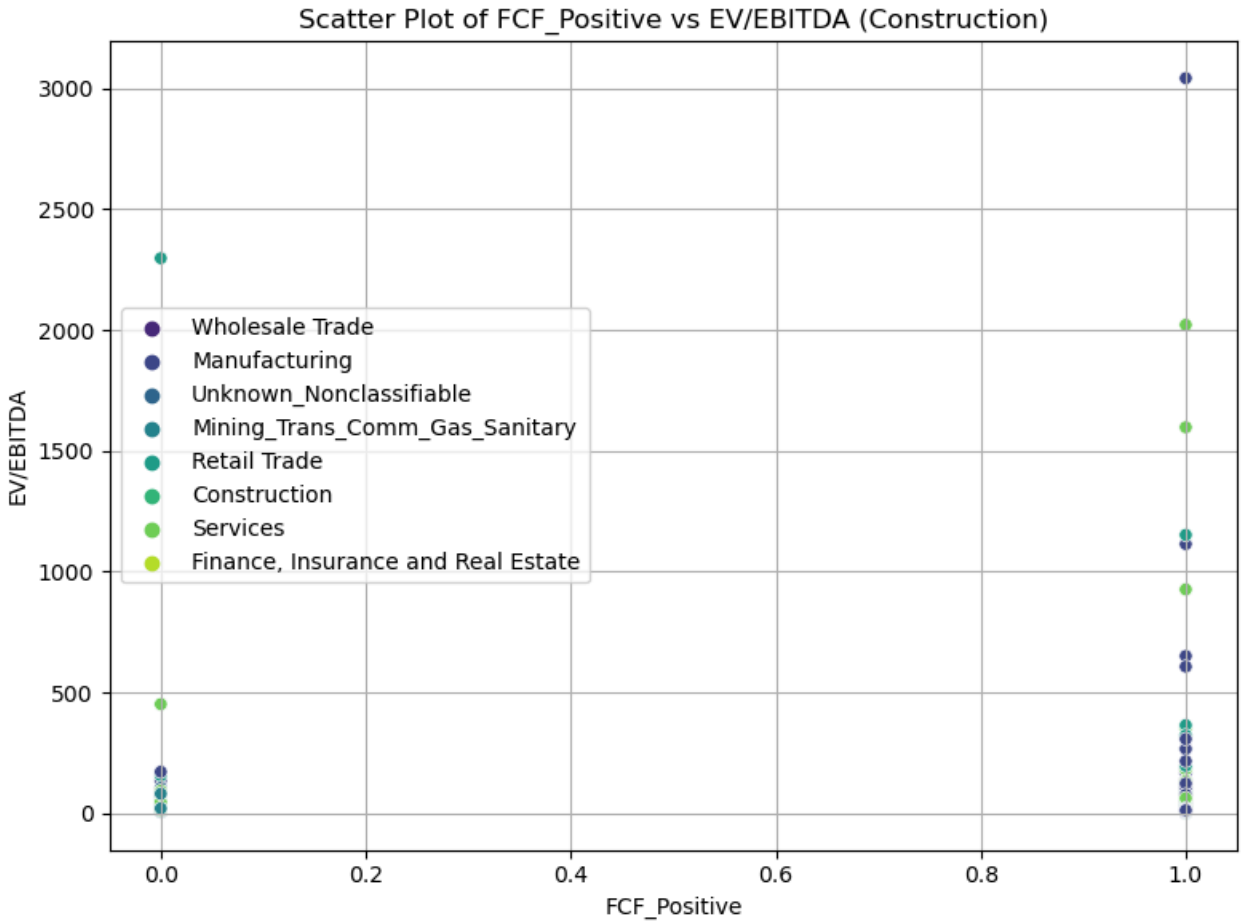
```
--- Construction ---
```



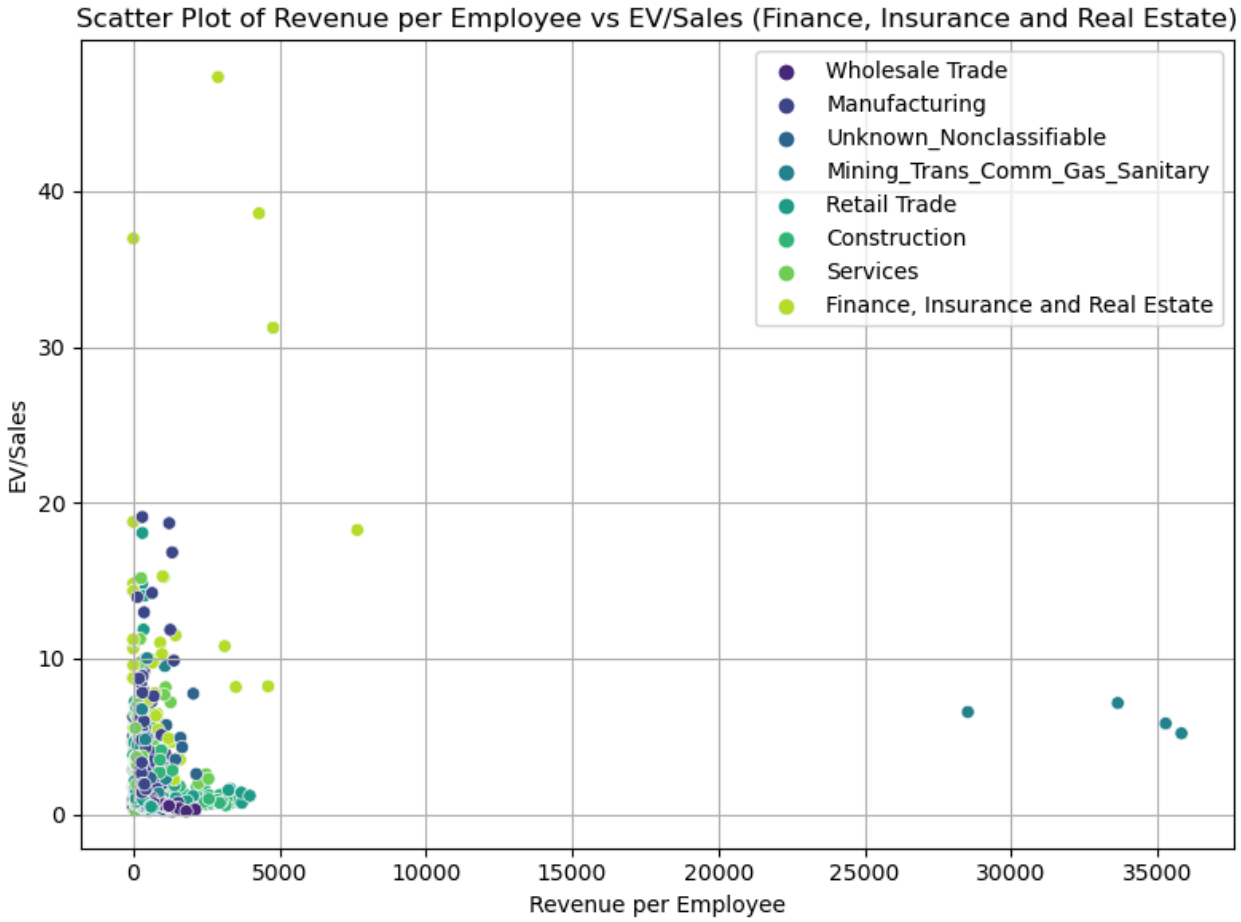

Scatter Plot of Current_Ratio vs EV/EBITDA (Construction)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/EBITDA
Current_Ratio
0
500
1000
1500
2000
2500
3000
0
10
20
30
40

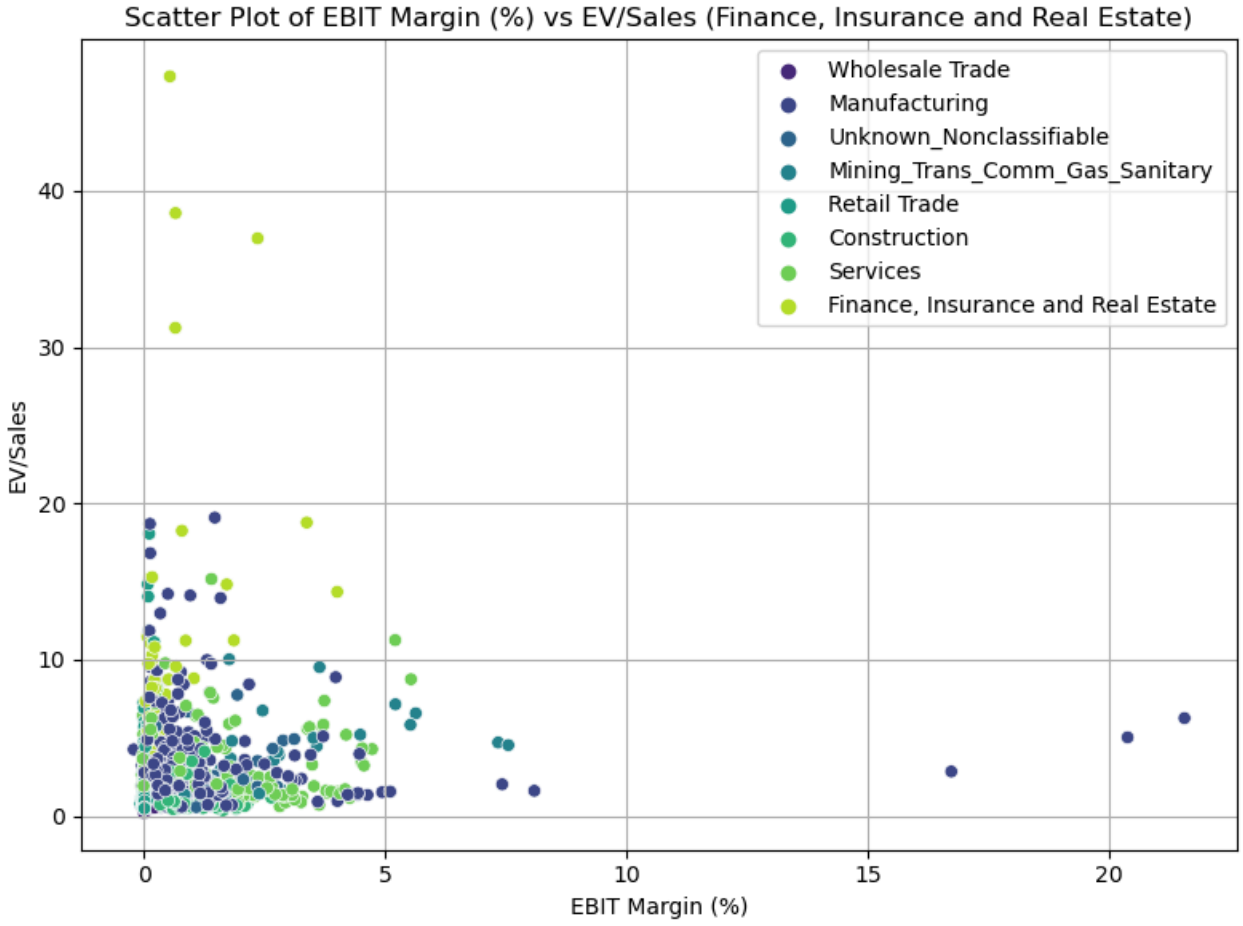

Scatter Plot of Quick_Ratio vs EV/EBITDA (Construction)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/EBITDA
Quick_Ratio
0
500
1000
1500
2000
2500
3000
0
10
20
30
40

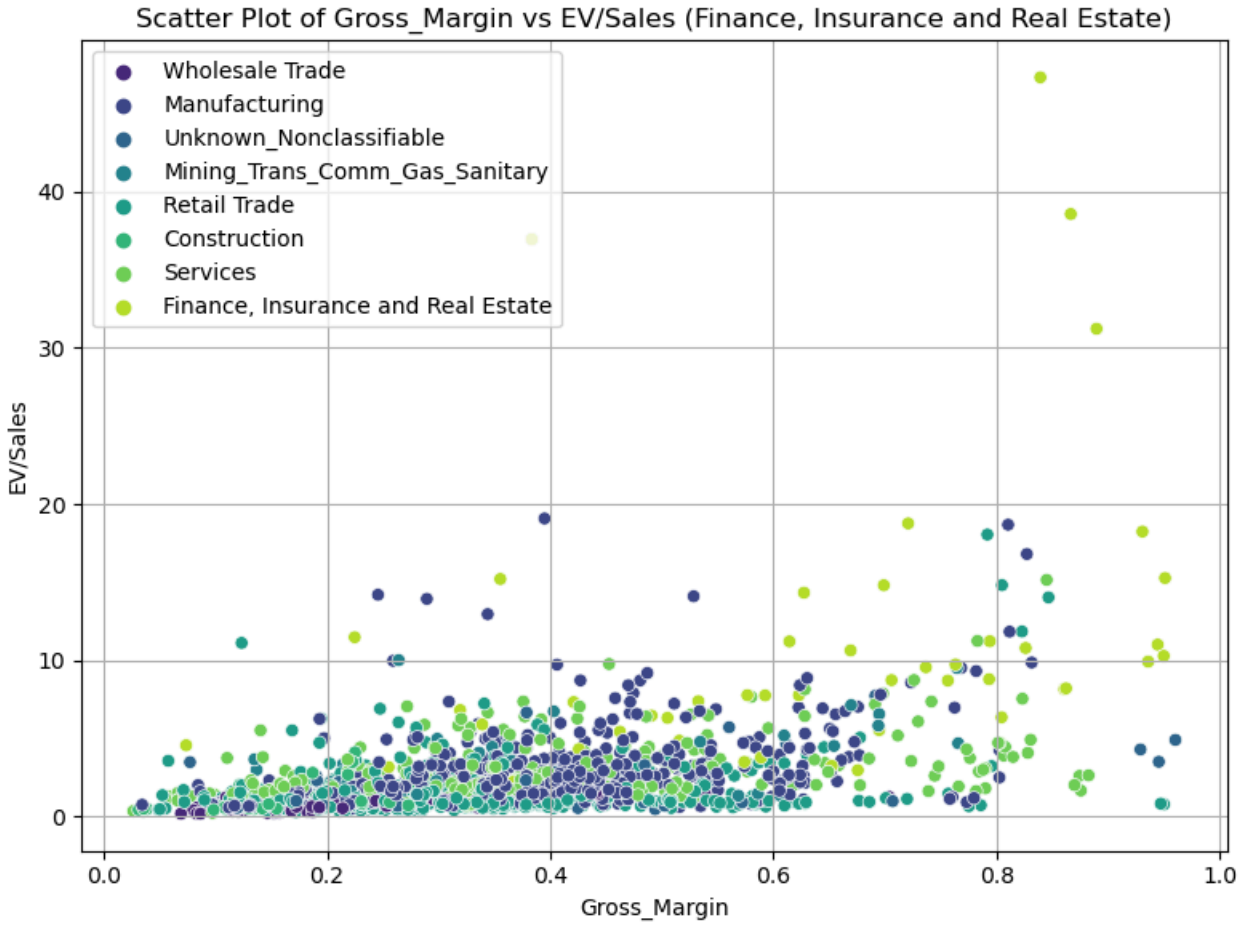


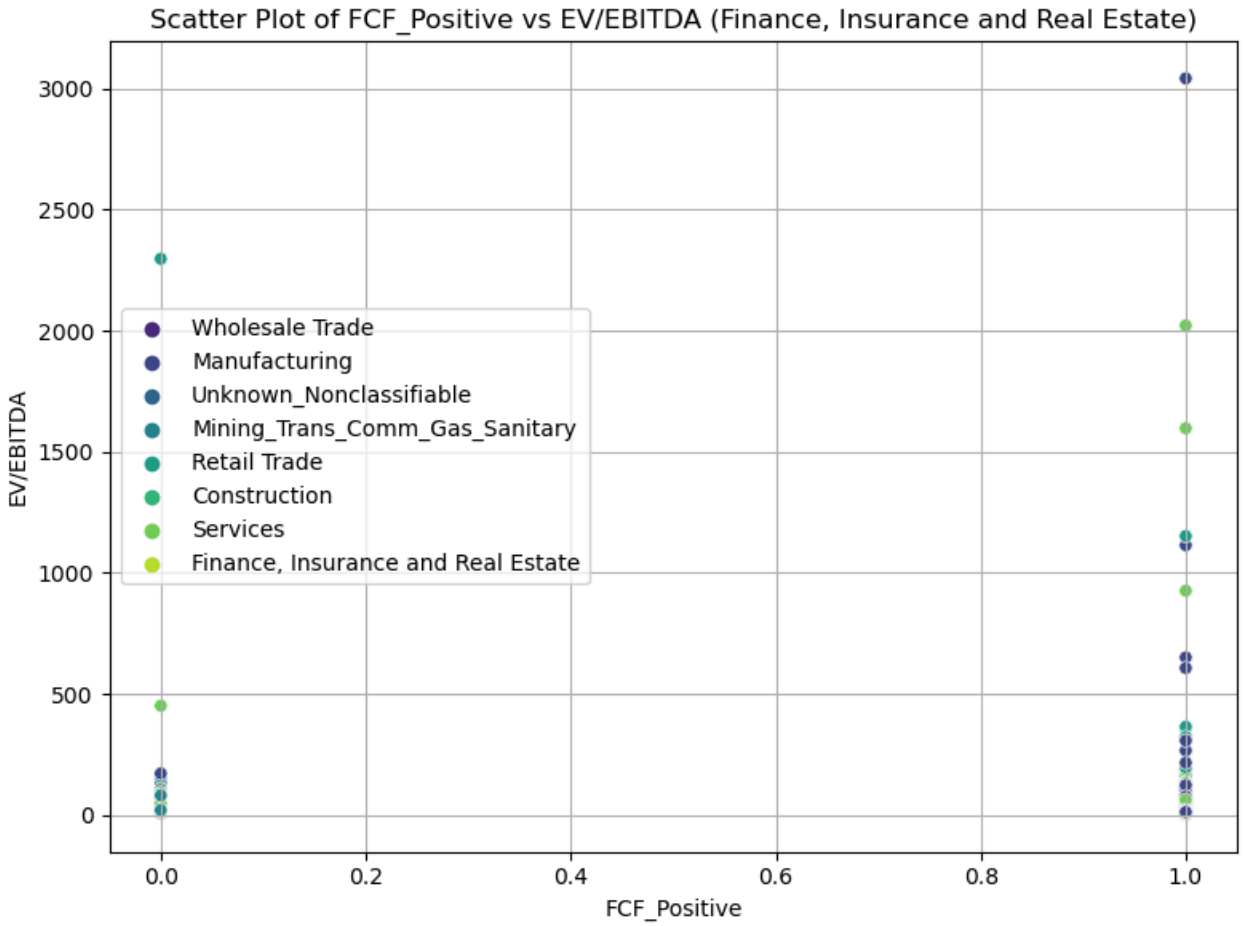


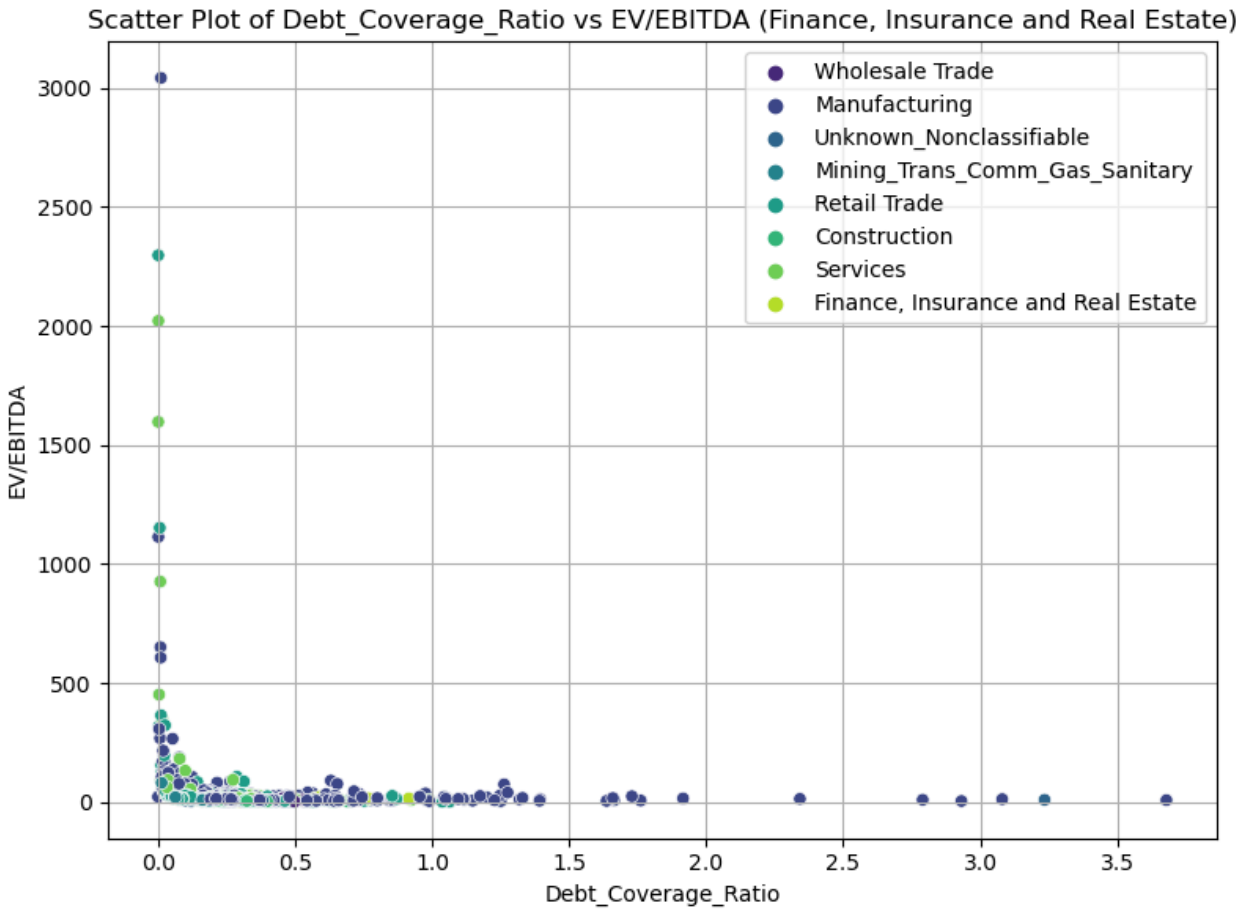

```
--- Finance, Insurance and Real Estate ---
```


Scatter Plot of Revenue per Employee vs EV/Sales (Finance, Insurance and Real Estate)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
0
10
20
30
40
0
5000
10000
15000
20000
25000
30000
35000
Revenue per Employee

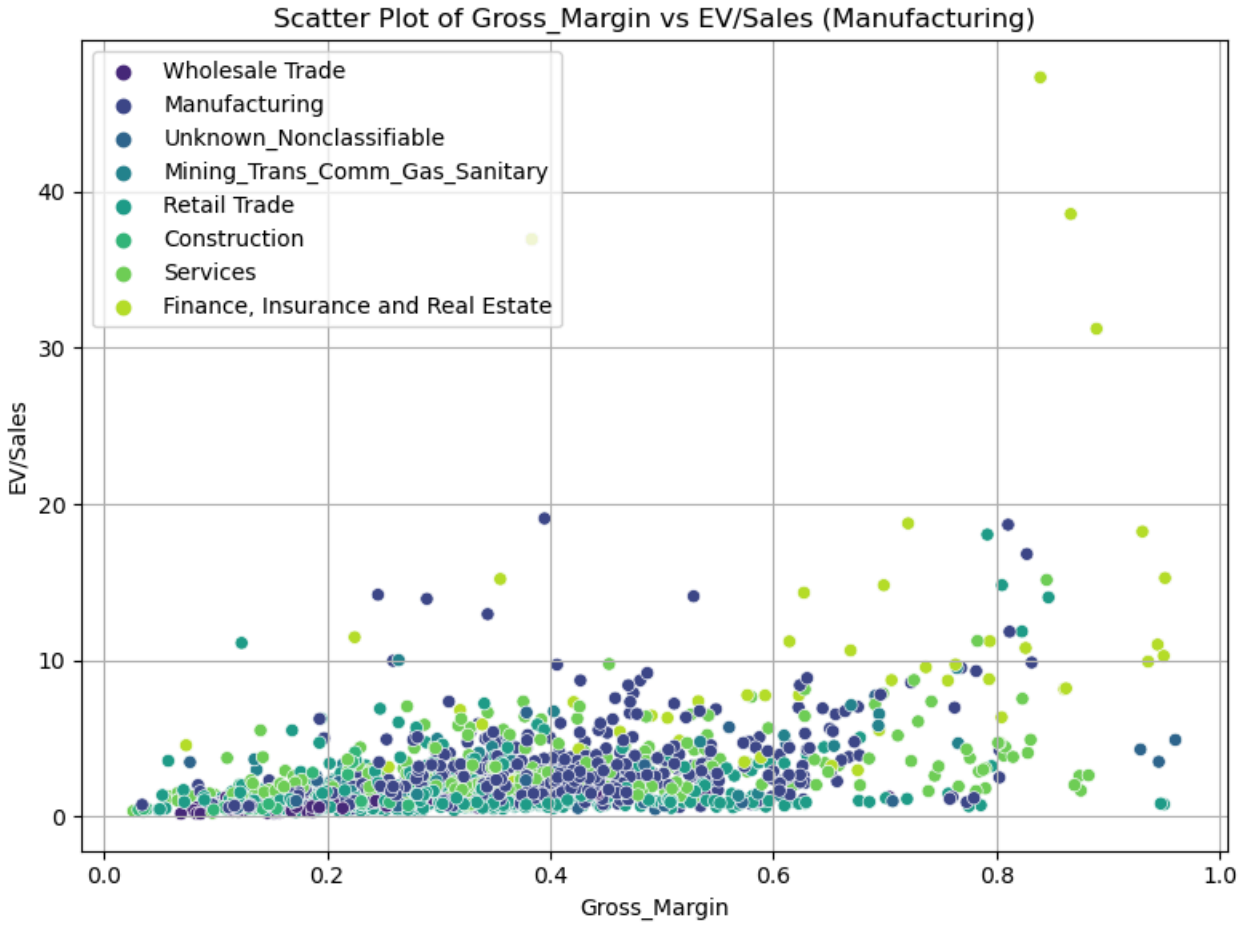

Scatter Plot of EBIT Margin (%) vs EV/Sales (Finance, Insurance and Real Estate)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
0
10
20
30
40
0
5
10
15
20
EBIT Margin (%)

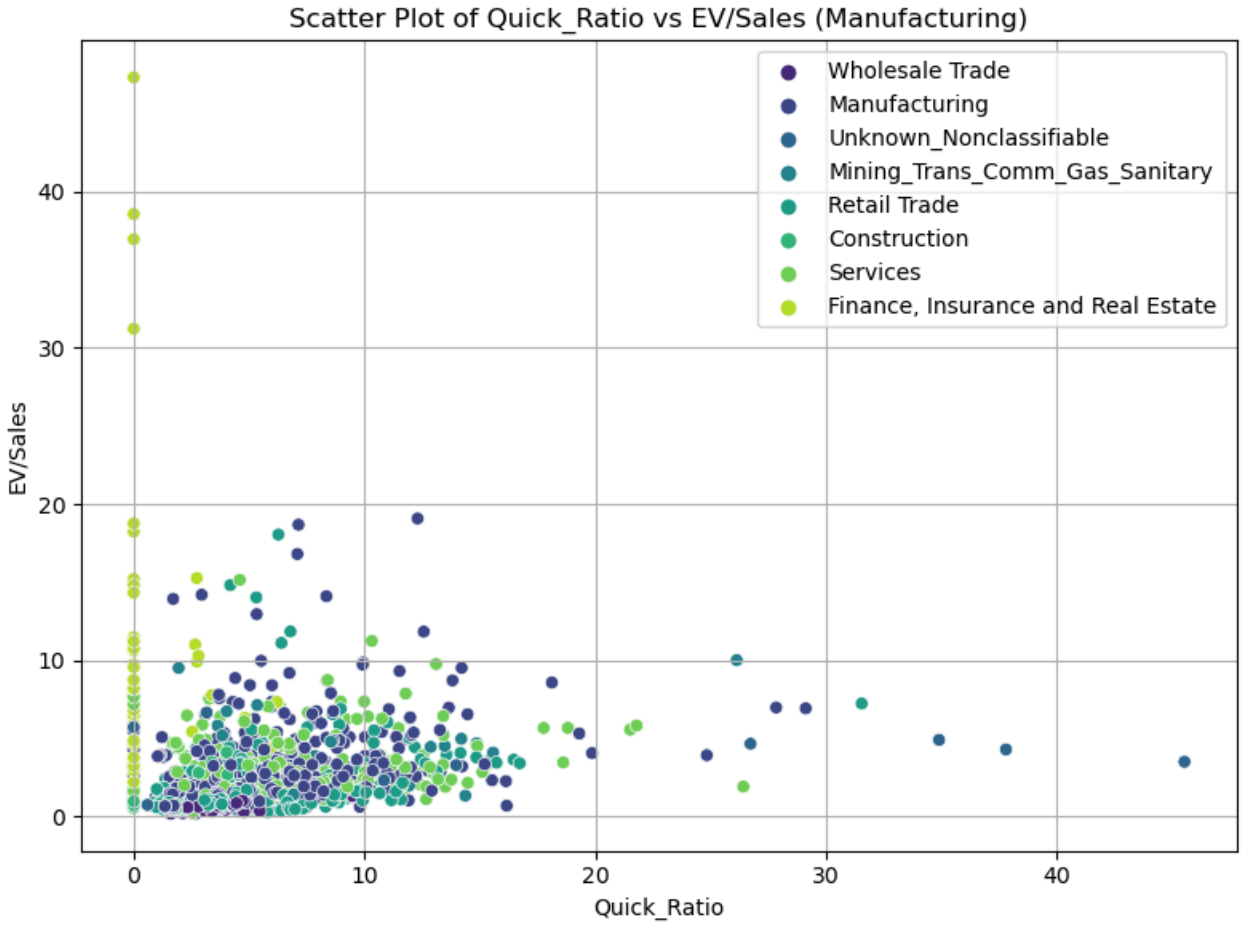

Scatter Plot of Gross_Margin vs EV/Sales (Finance, Insurance and Real Estate)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
Gross_Margin
0
10
20
30
40
0.0
0.2
0.4
0.6
0.8
1.0

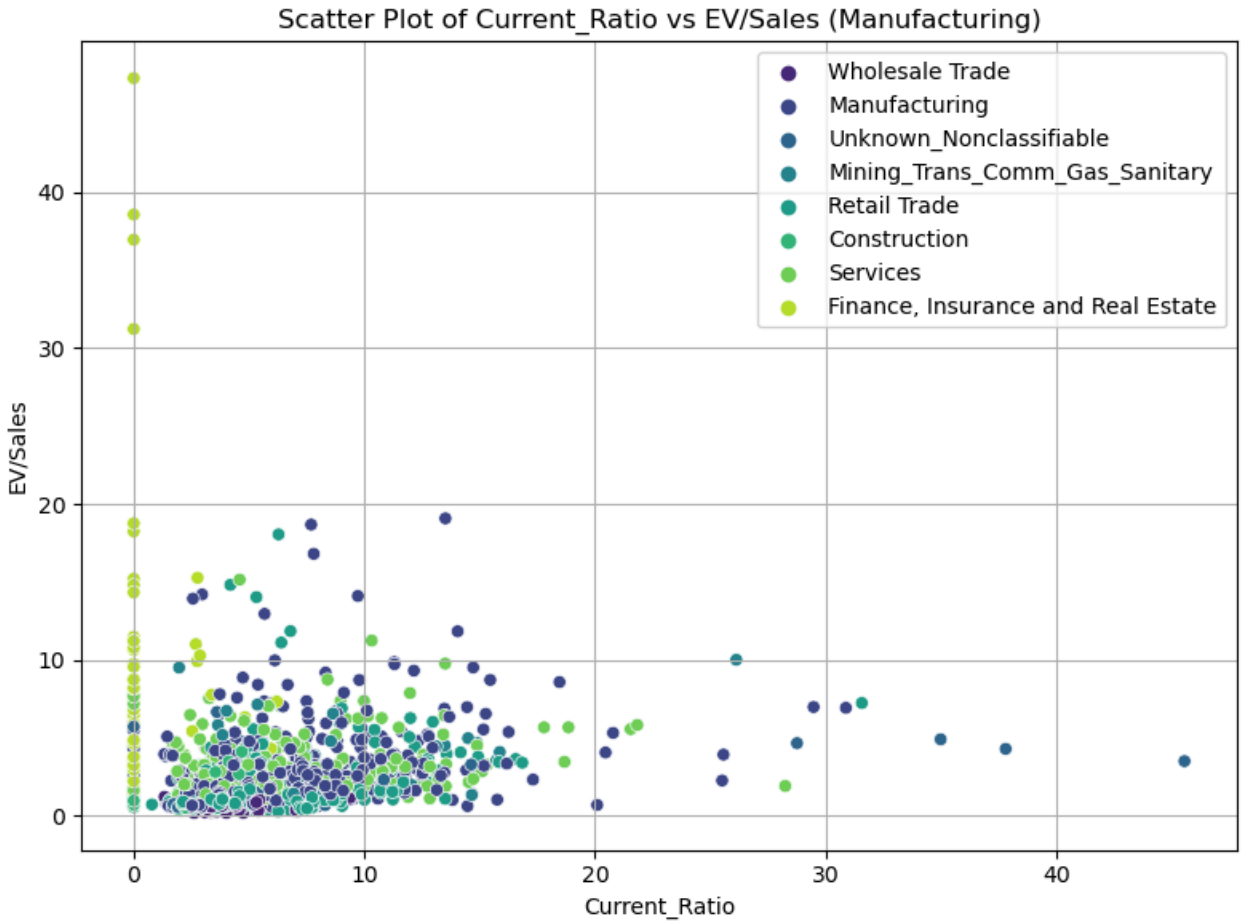

Scatter Plot of FCF_Positive vs EV/EBITDA (Finance, Insurance and Real Estate)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/EBITDA
FCF_Positive
0
500
1000
1500
2000
2500
3000
0.0
0.2
0.4
0.6
0.8
1.0

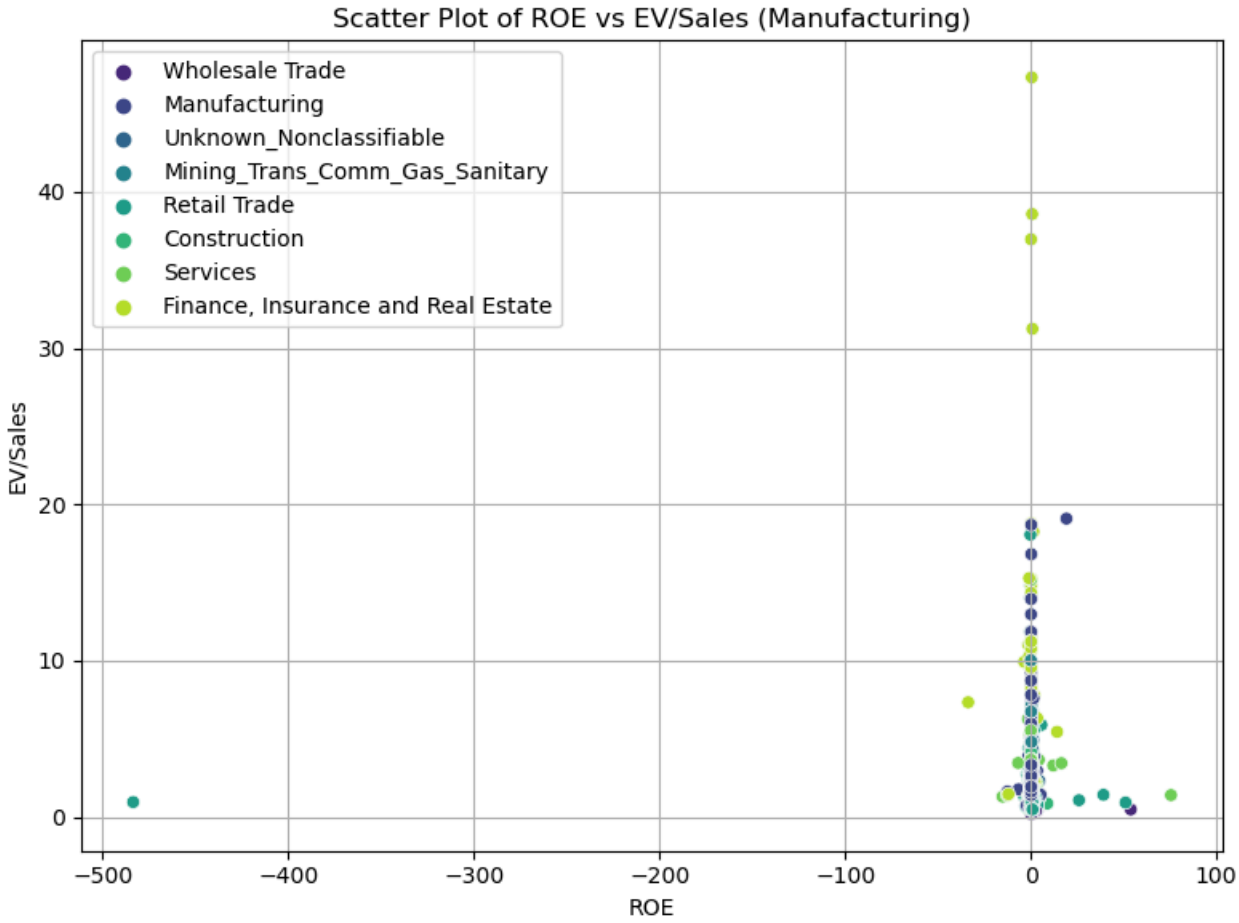


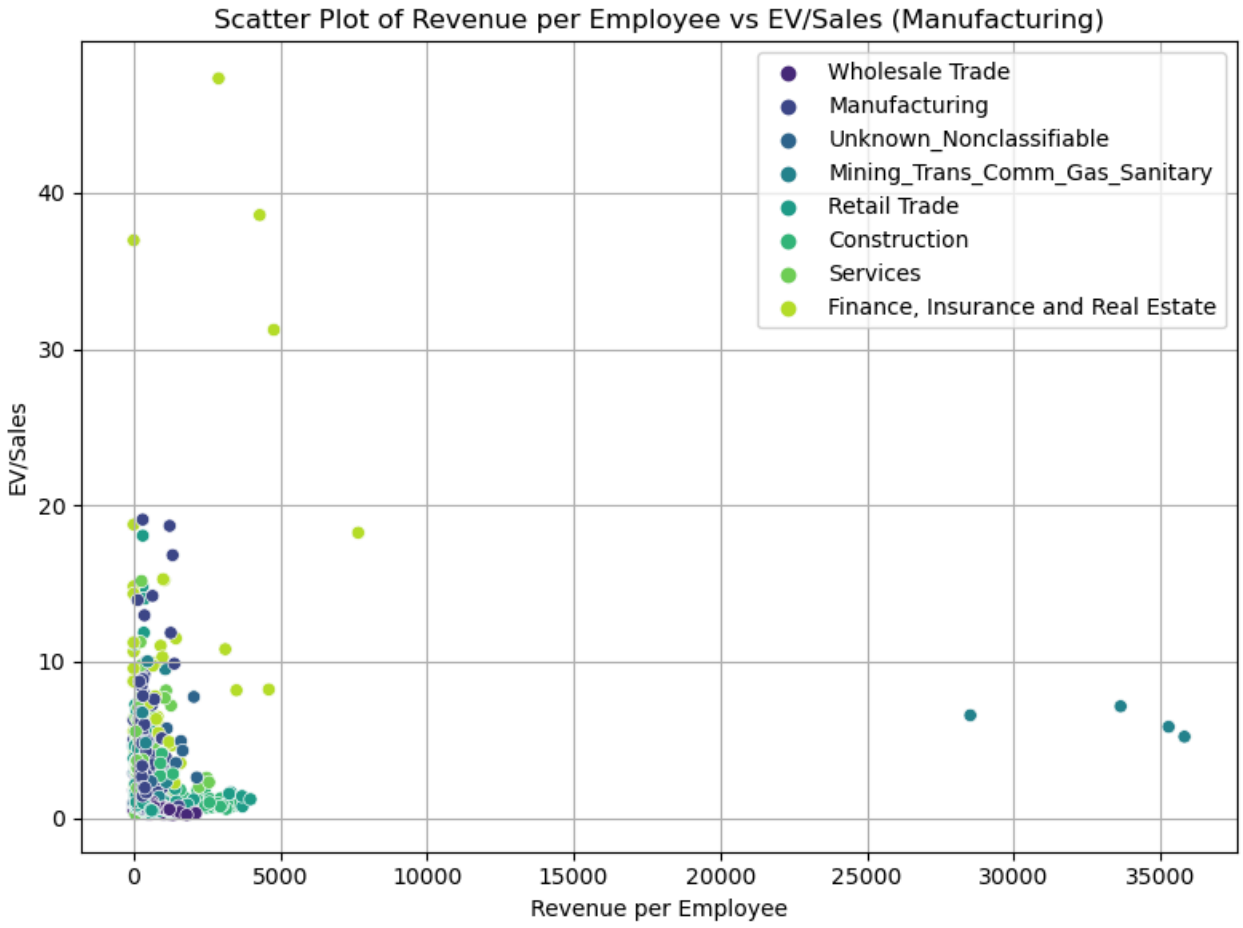
--- Manufacturing ---



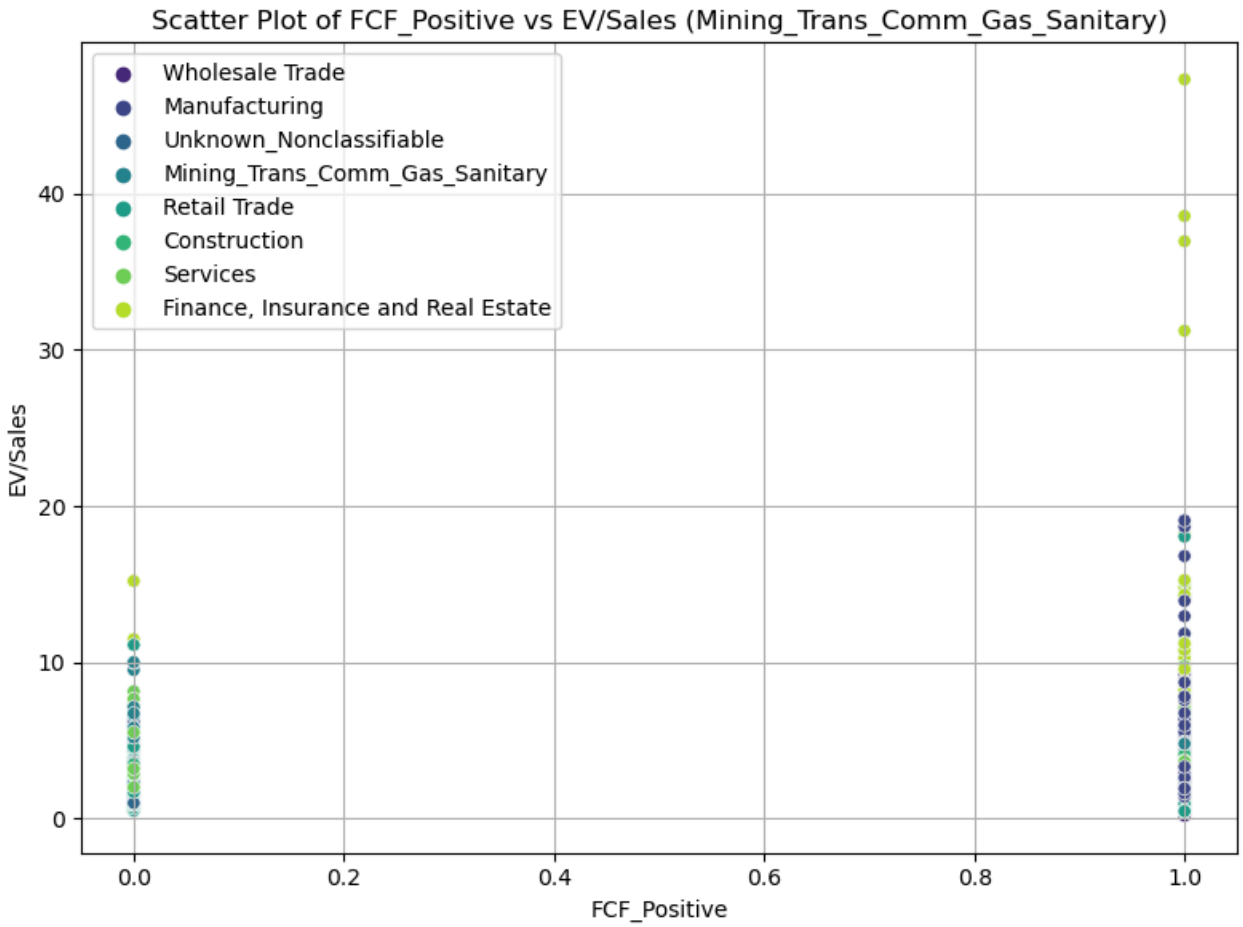

Scatter Plot of Quick_Ratio vs EV/Sales (Manufacturing)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
Quick_Ratio
0
10
20
30
40

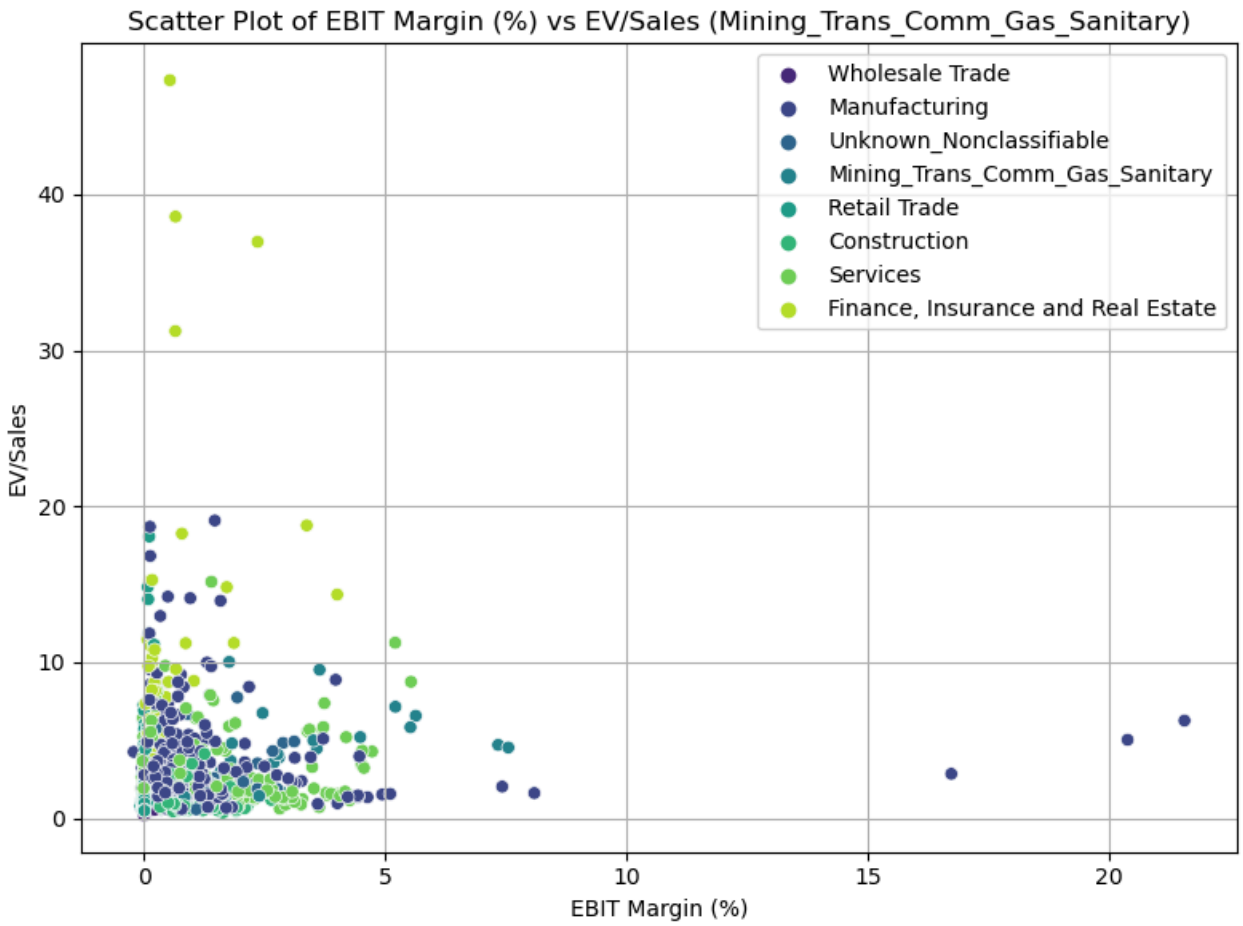

Scatter Plot of Current_Ratio vs EV/Sales (Manufacturing)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
Current_Ratio
0
10
20
30
40

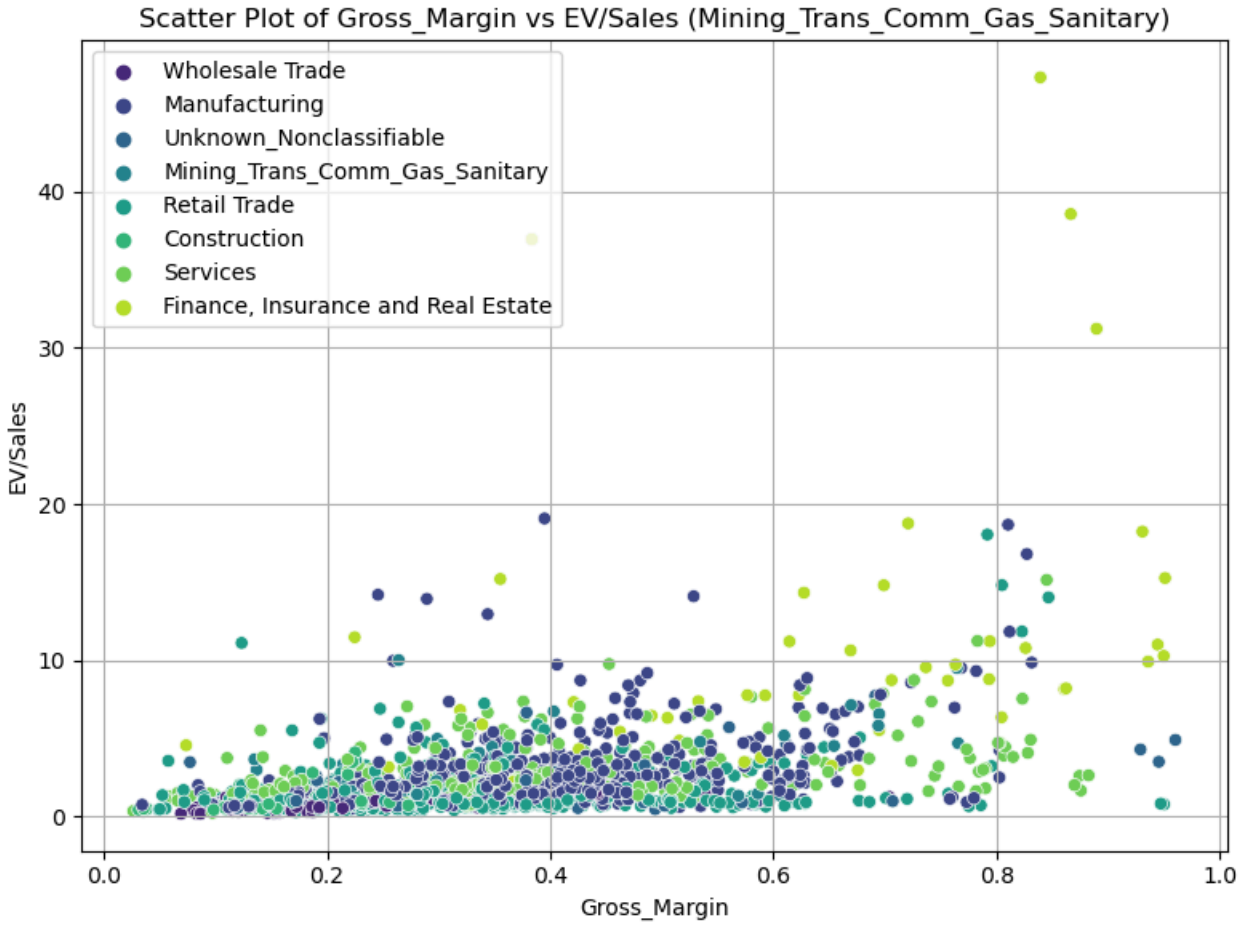


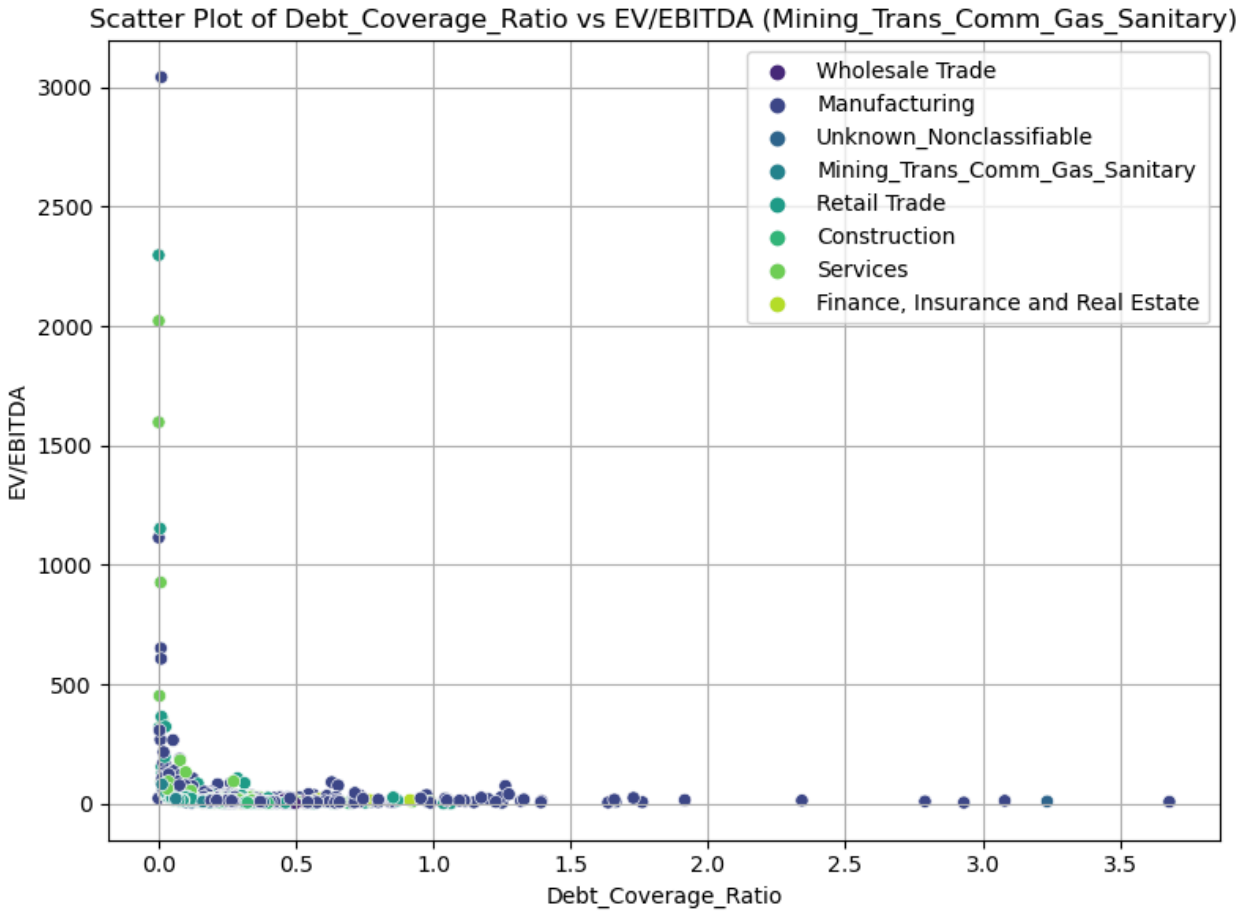


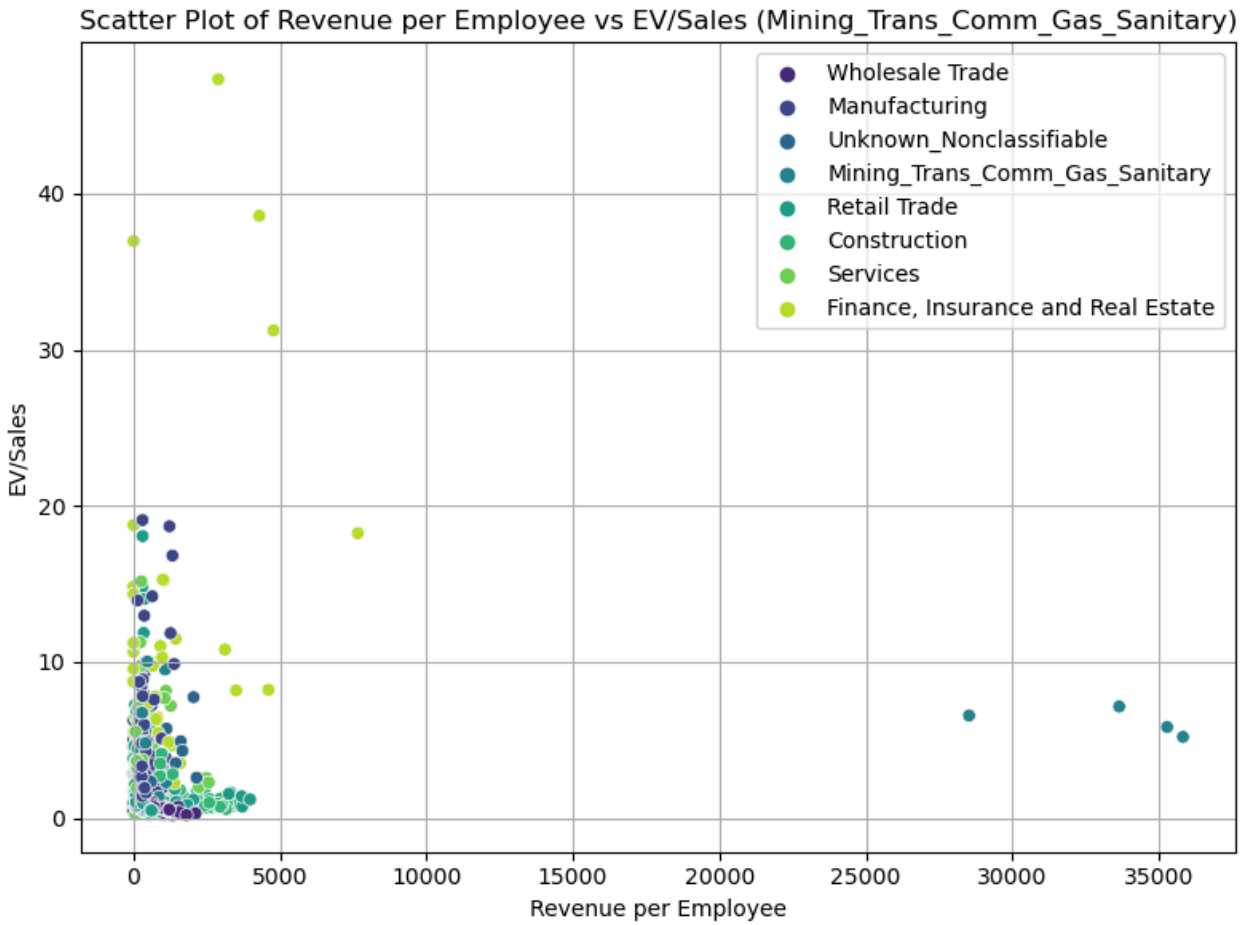
--- Mining_Trans_Comm_Gas_Sanitary ---


Scatter Plot of FCF_Positive vs EV/Sales (Mining_Trans_Comm_Gas_Sanitary)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
FCF_Positive
0
10
20
30
40
0.0
0.2
0.4
0.6
0.8
1.0

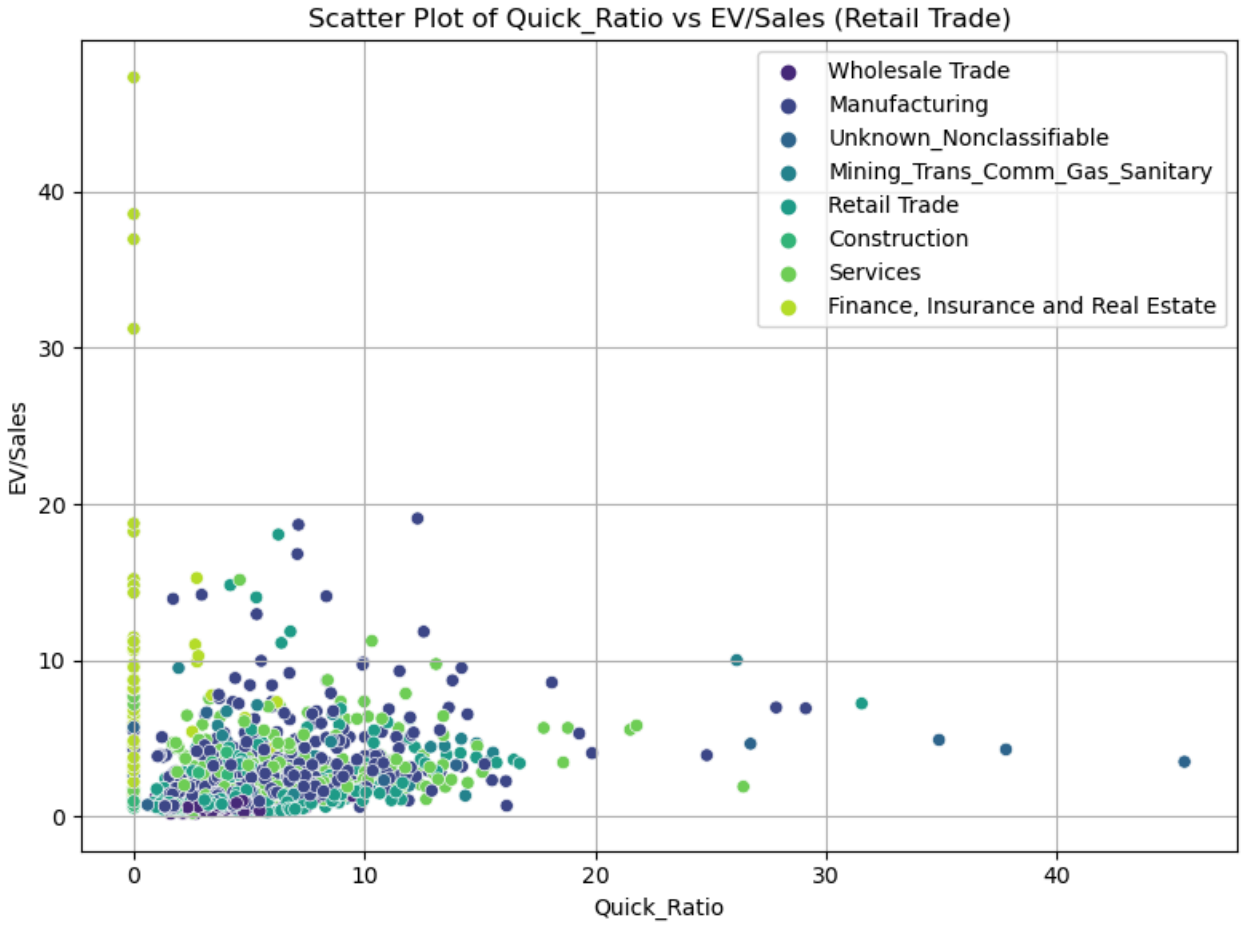

Scatter Plot of EBIT Margin (%) vs EV/Sales (Mining_Trans_Comm_Gas_Sanitary)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
EBIT Margin (%)
0
10
20
30
40
0
5
10
15
20

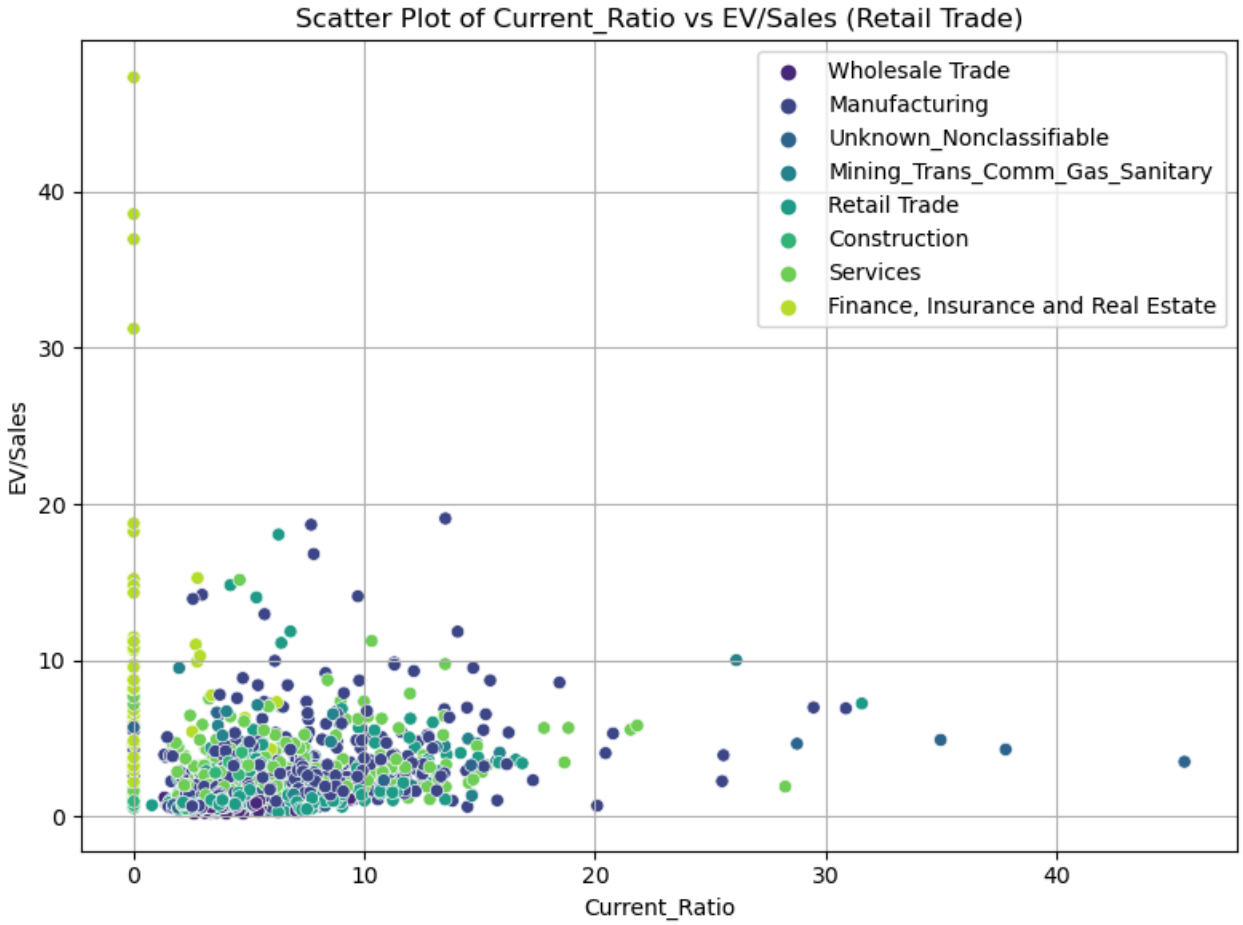

Scatter Plot of Gross_Margin vs EV/Sales (Mining_Trans_Comm_Gas_Sanitary)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
Gross_Margin
0
10
20
30
40
0.0
0.2
0.4
0.6
0.8
1.0

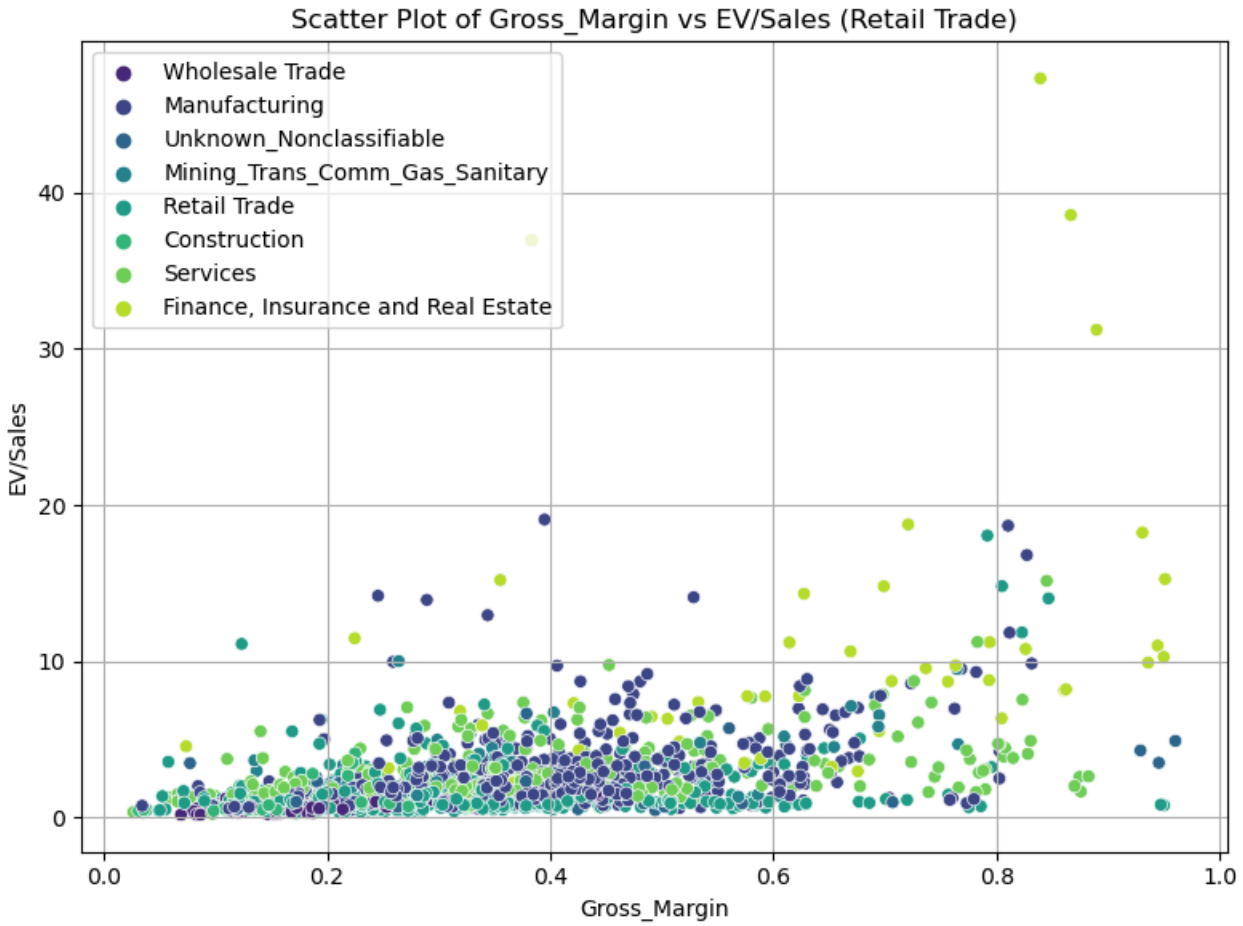

Scatter Plot of Debt_Coverage_Ratio vs EV/EBITDA (Mining_Trans_Comm_Gas_Sanitary)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/EBITDA
Debt_Coverage_Ratio
0
500
1000
1500
2000
2500
3000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

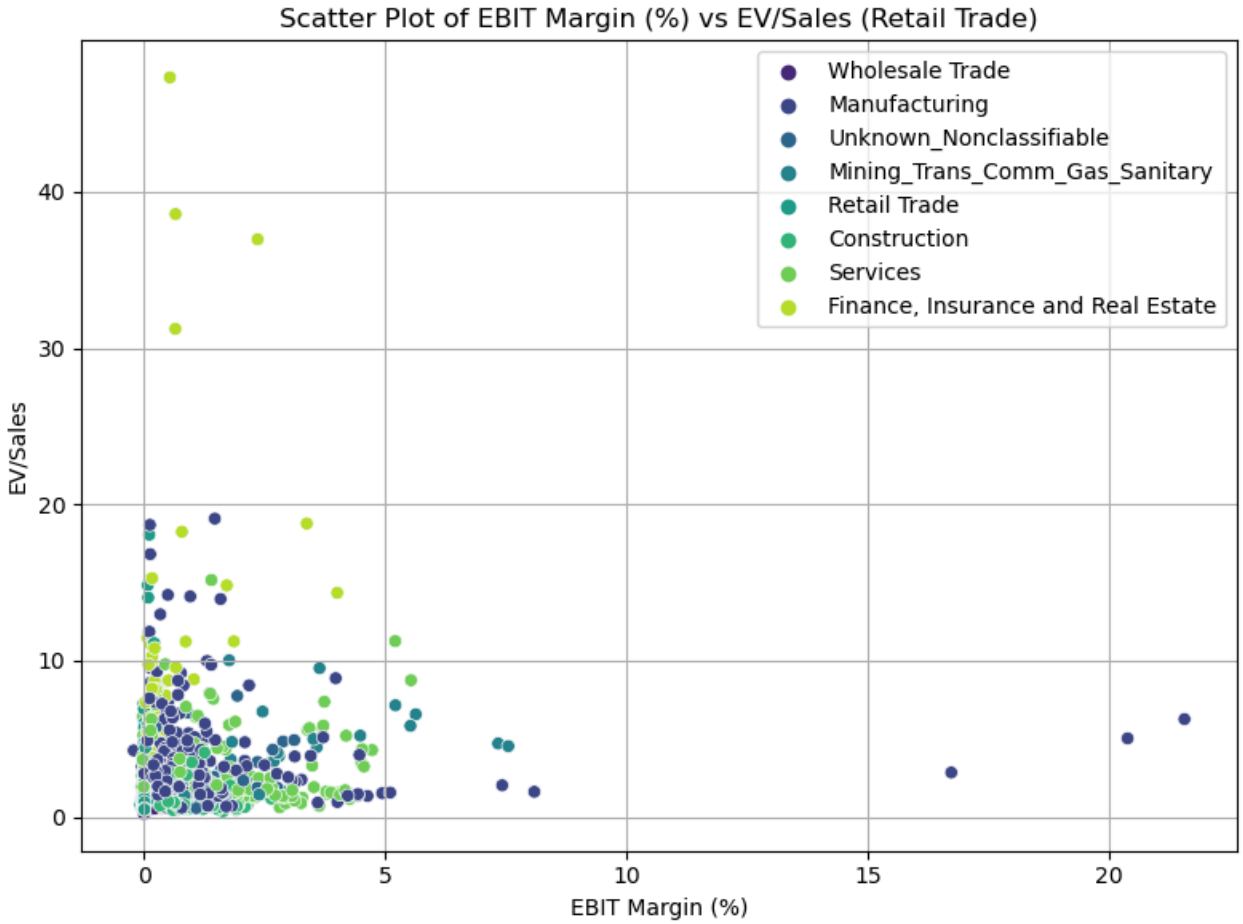


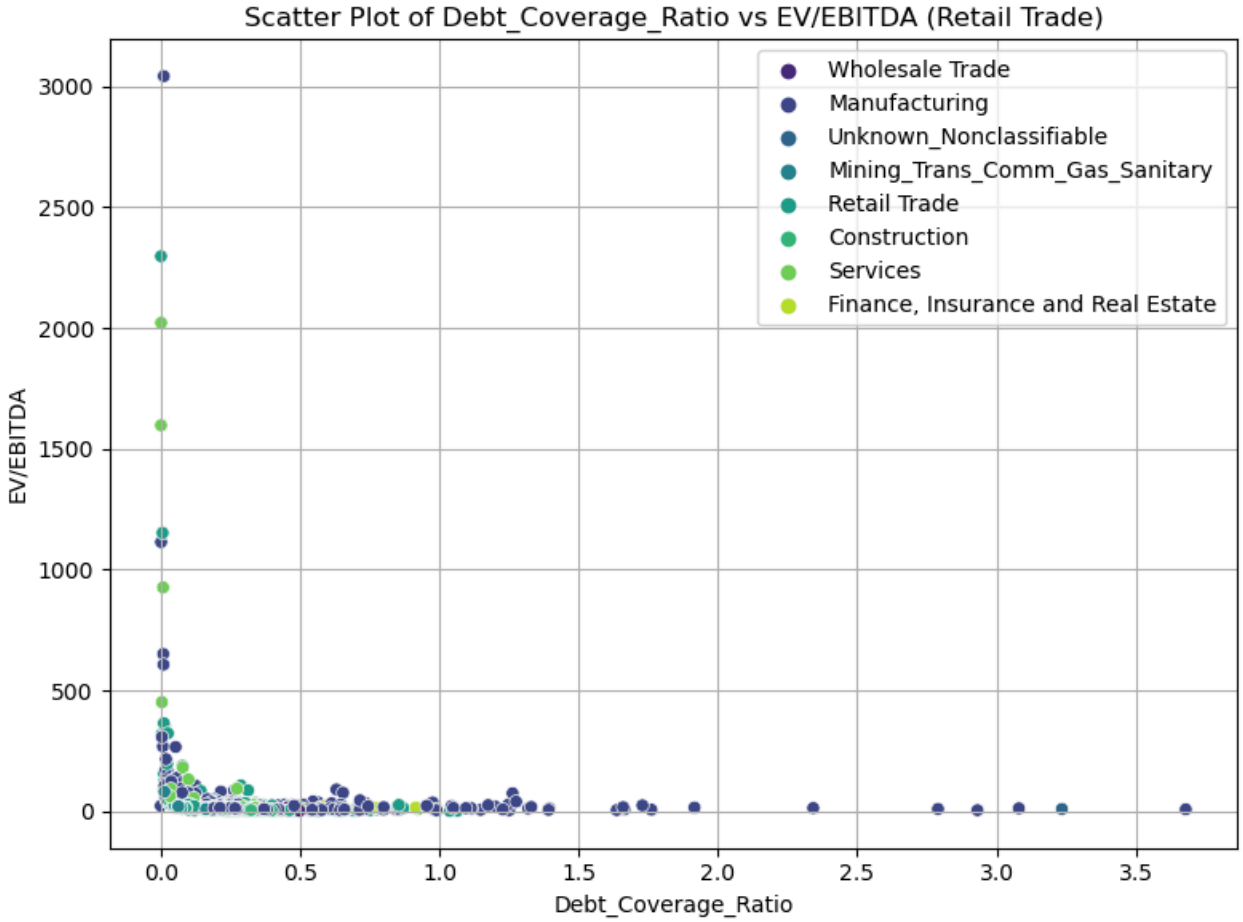

```
--- Retail Trade ---
```



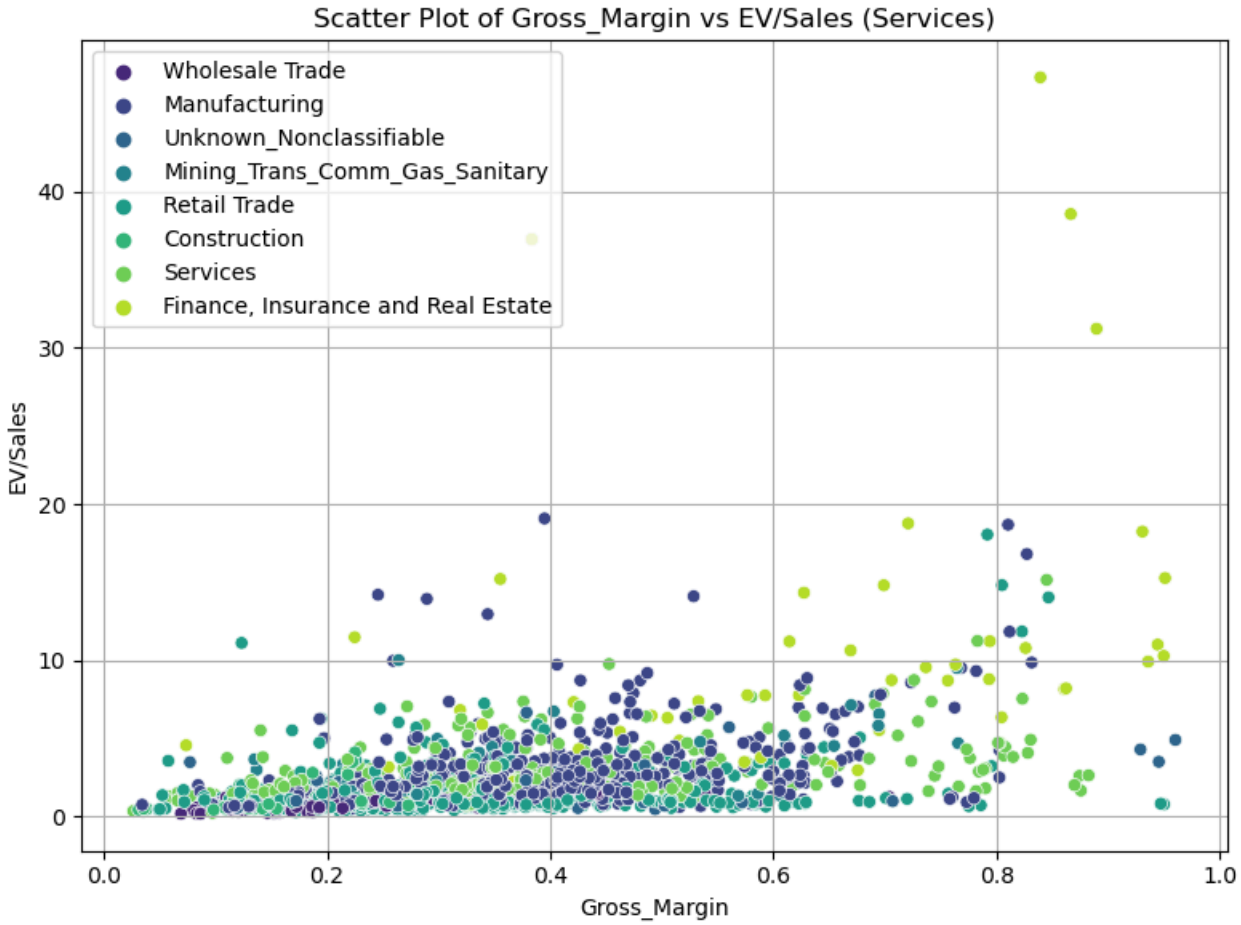

Scatter Plot of Current_Ratio vs EV/Sales (Retail Trade)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
Current_Ratio
0
10
20
30
40

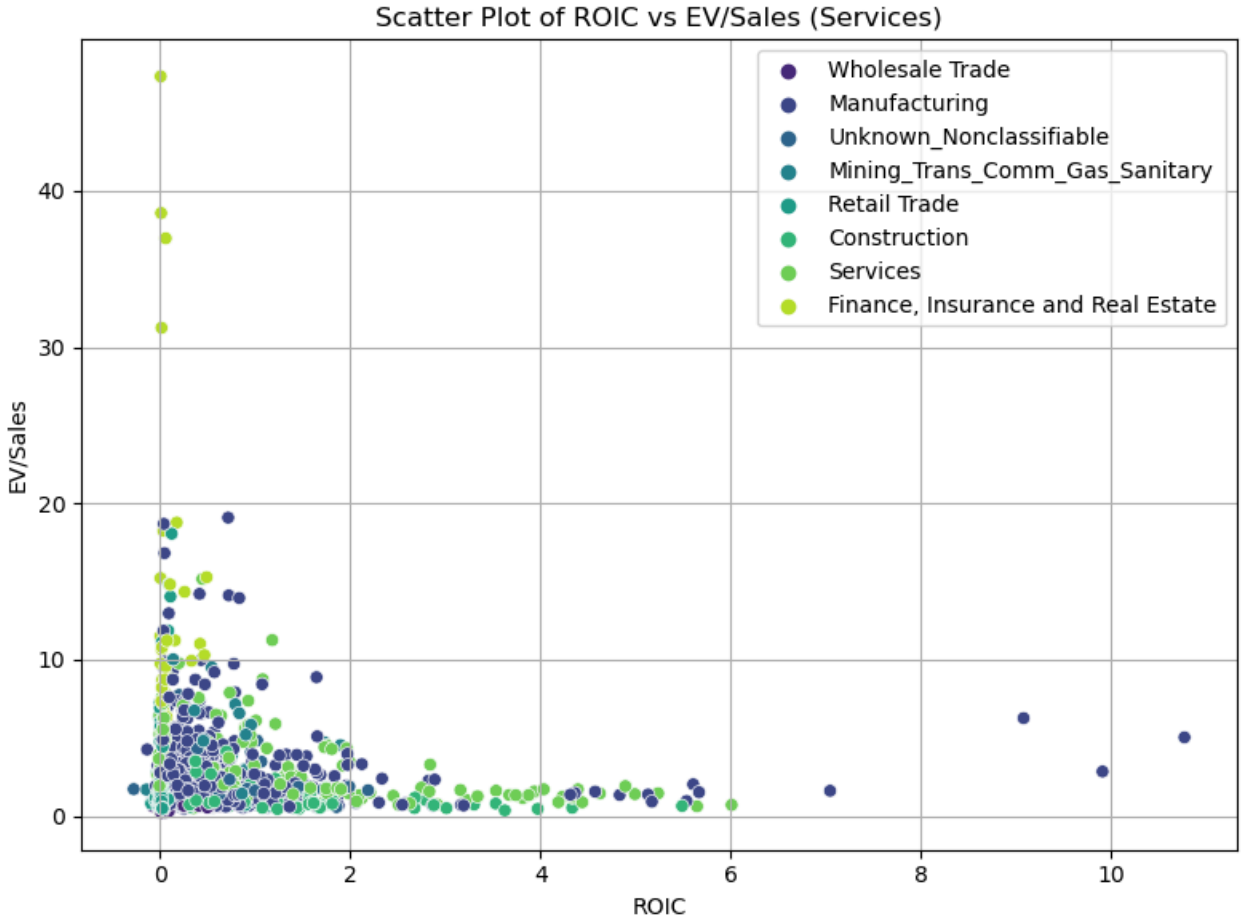

Scatter Plot of Gross_Margin vs EV/Sales (Retail Trade)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
Gross_Margin
0.0
0.2
0.4
0.6
0.8
1.0

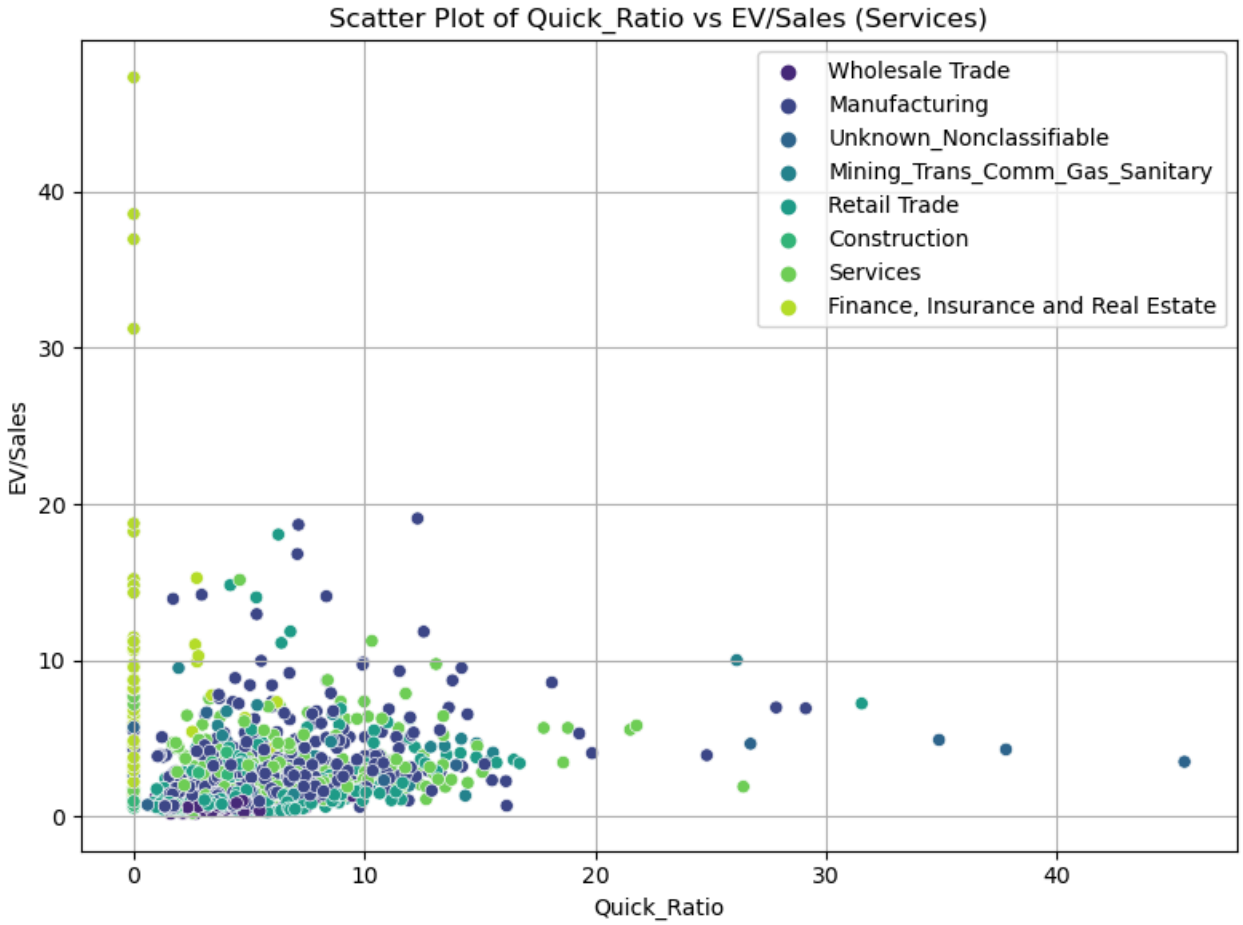

Scatter Plot of EBIT Margin (%) vs EV/Sales (Retail Trade)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
EBIT Margin (%)
0
10
20
30
40
0
5
10
15
20

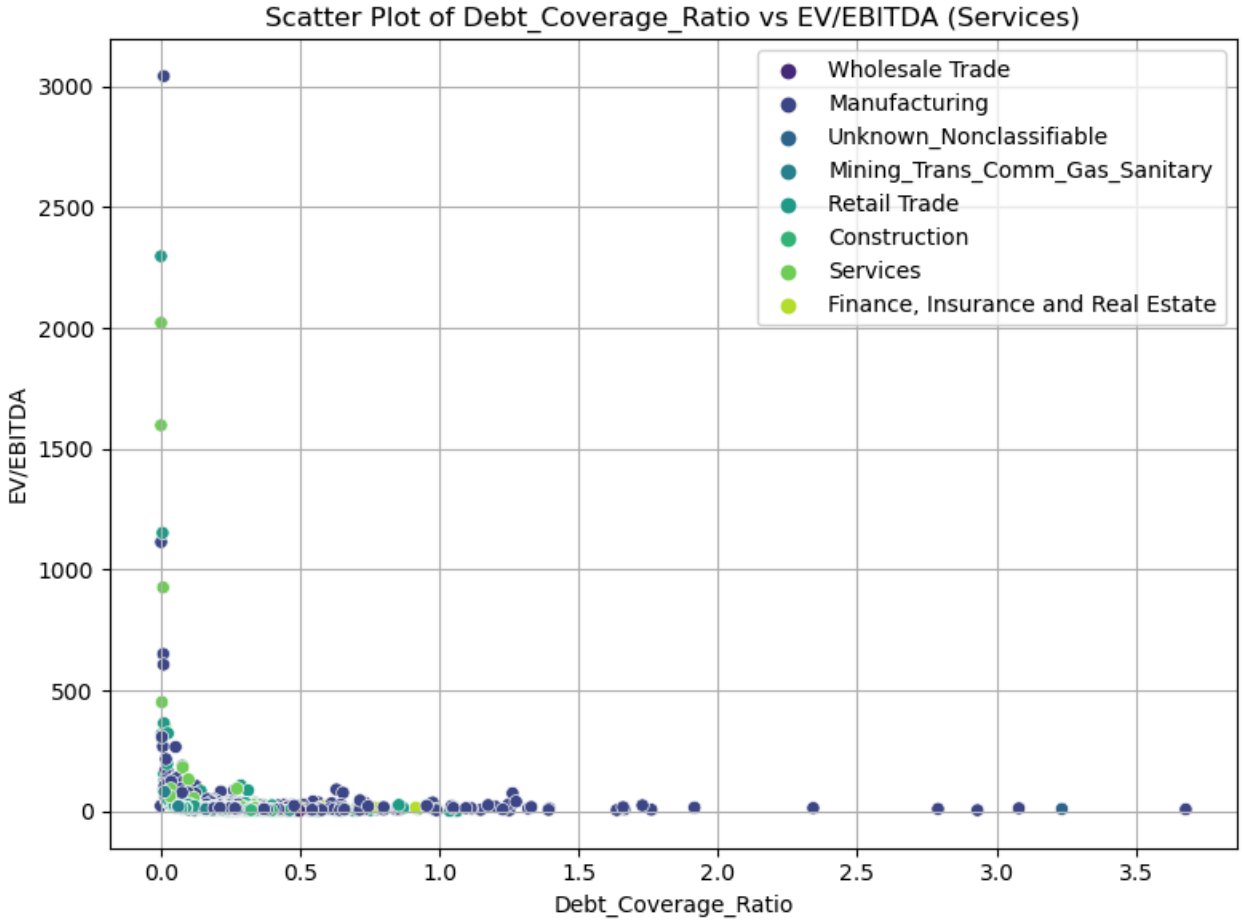

Scatter Plot of Debt_Coverage_Ratio vs EV/EBITDA (Retail Trade)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/EBITDA
Debt_Coverage_Ratio
0
500
1000
1500
2000
2500
3000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

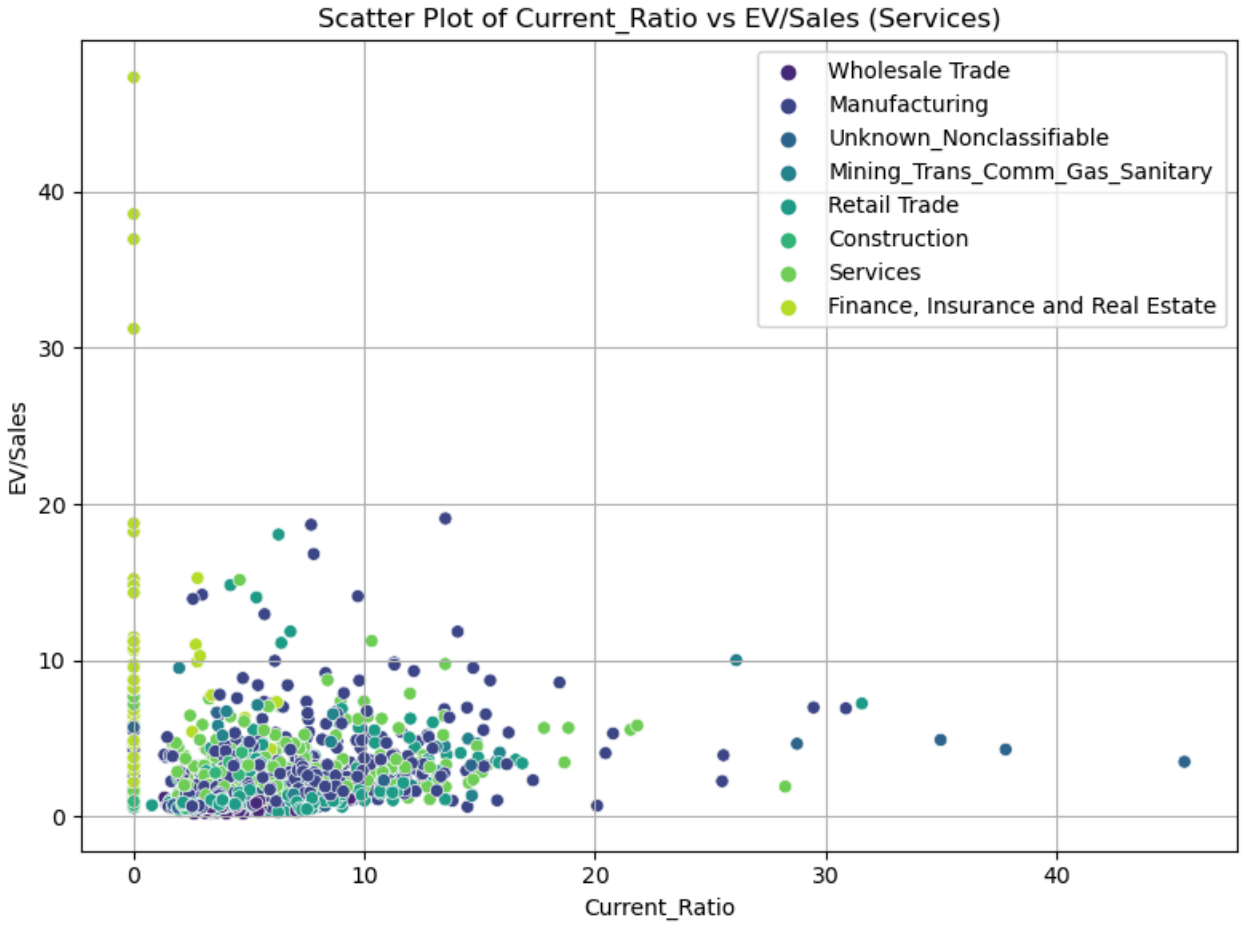
```
--- Services ---
```


Scatter Plot of Gross_Margin vs EV/Sales (Services)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
Gross_Margin
0
10
20
30
40
0.0
0.2
0.4
0.6
0.8
1.0

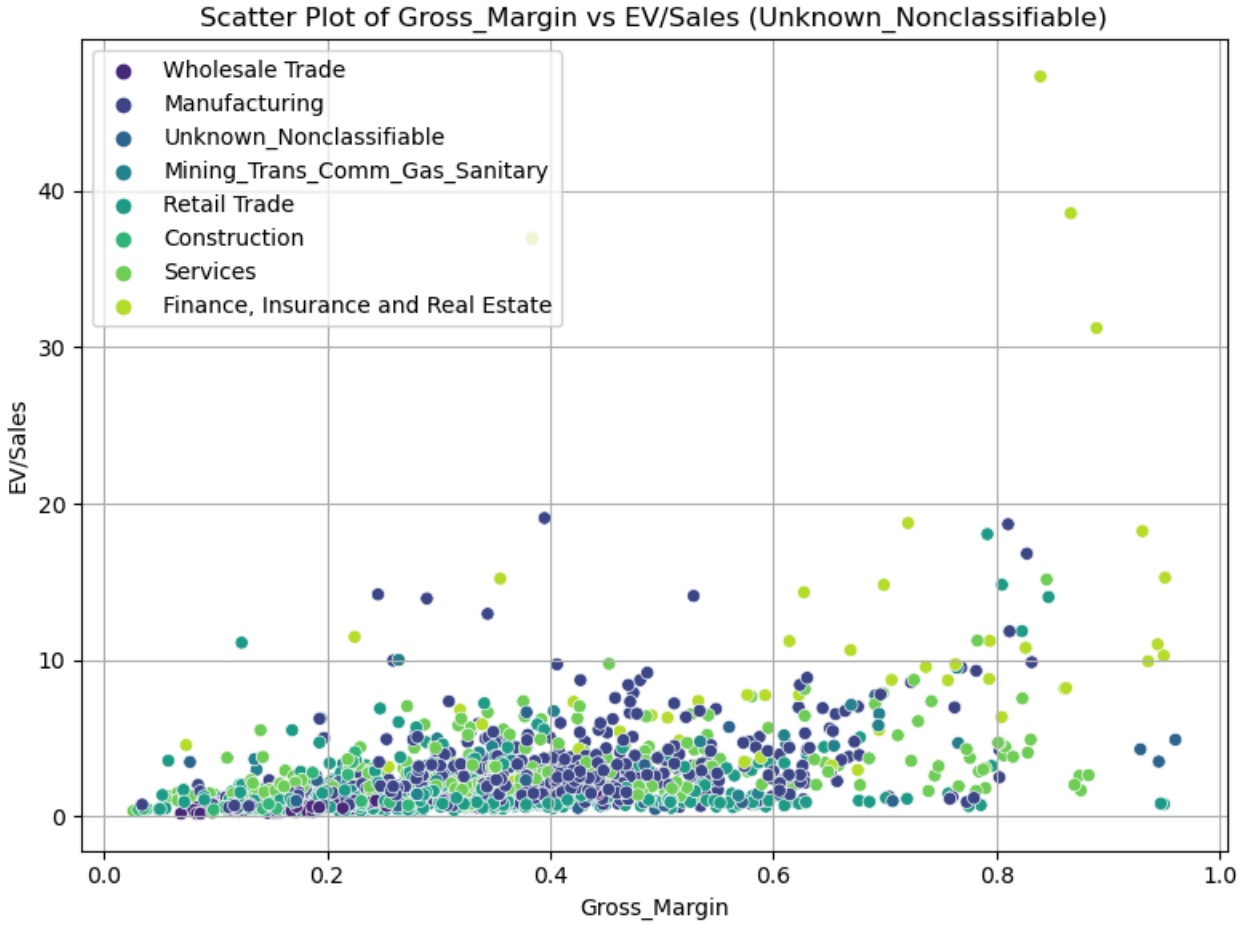

Scatter Plot of ROIC vs EV/Sales (Services)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
ROIC
0
10
20
30
40
0
2
4
6
8
10

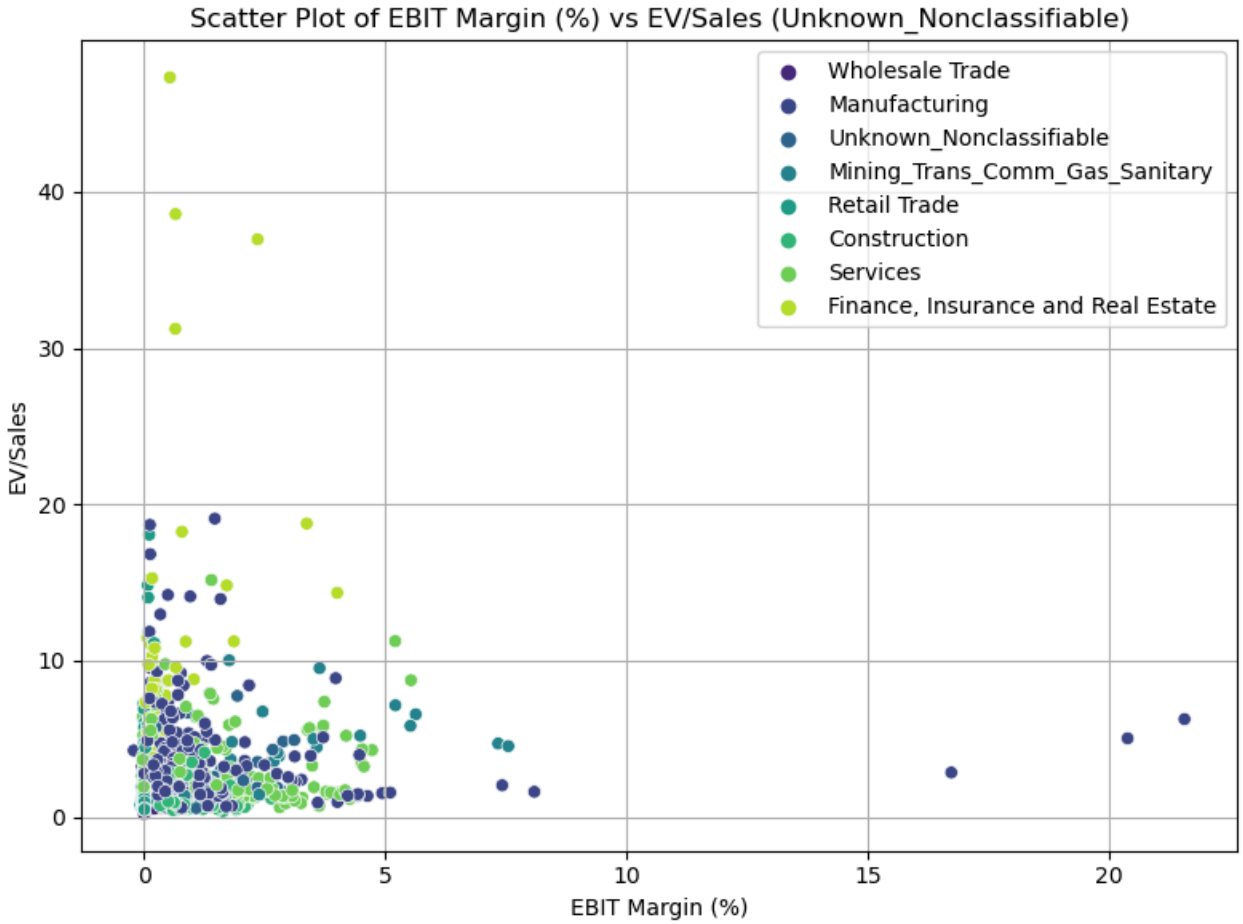

Scatter Plot of Quick_Ratio vs EV/Sales (Services)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
Quick_Ratio
0
10
20
30
40

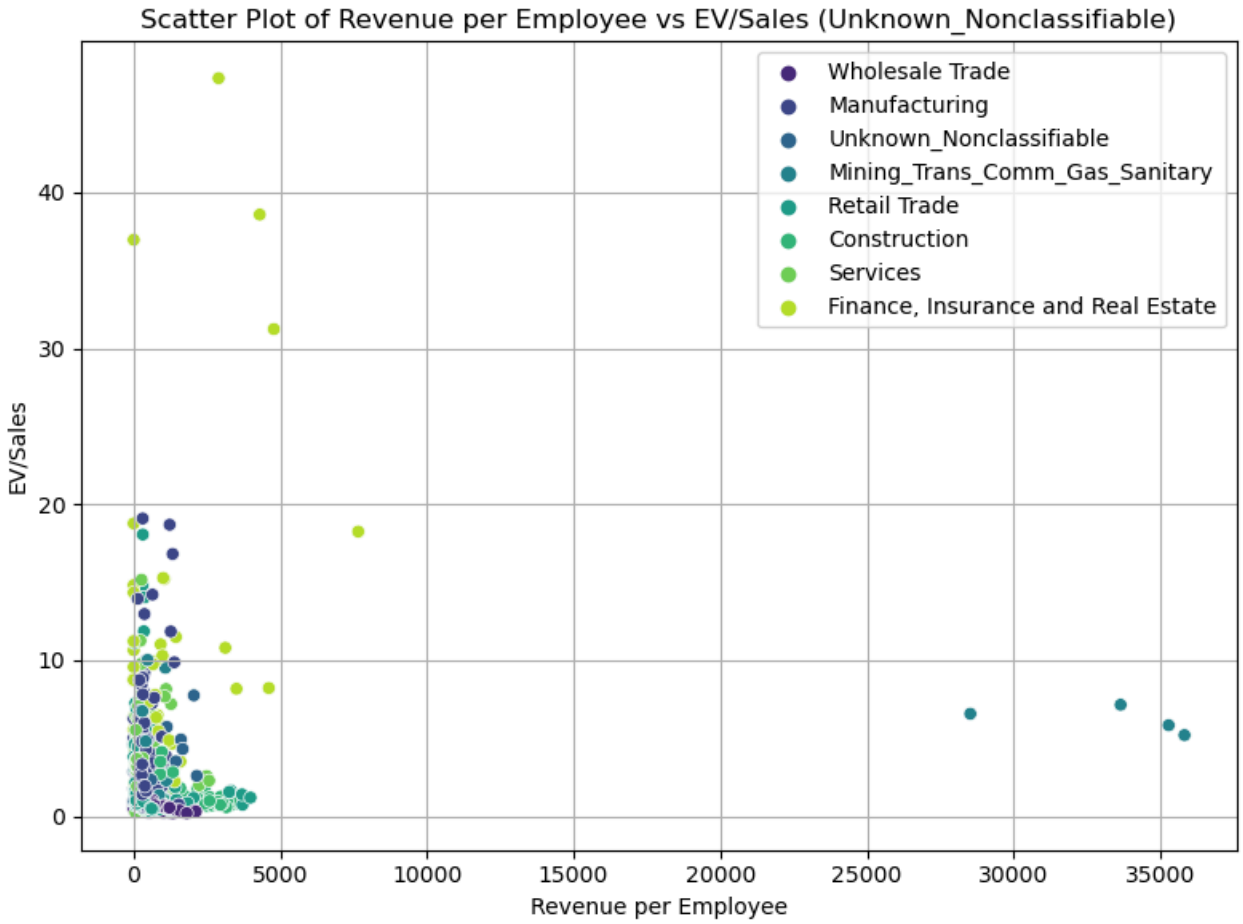

Scatter Plot of Debt_Coverage_Ratio vs EV/EBITDA (Services)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/EBITDA
Debt_Coverage_Ratio
0
500
1000
1500
2000
2500
3000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

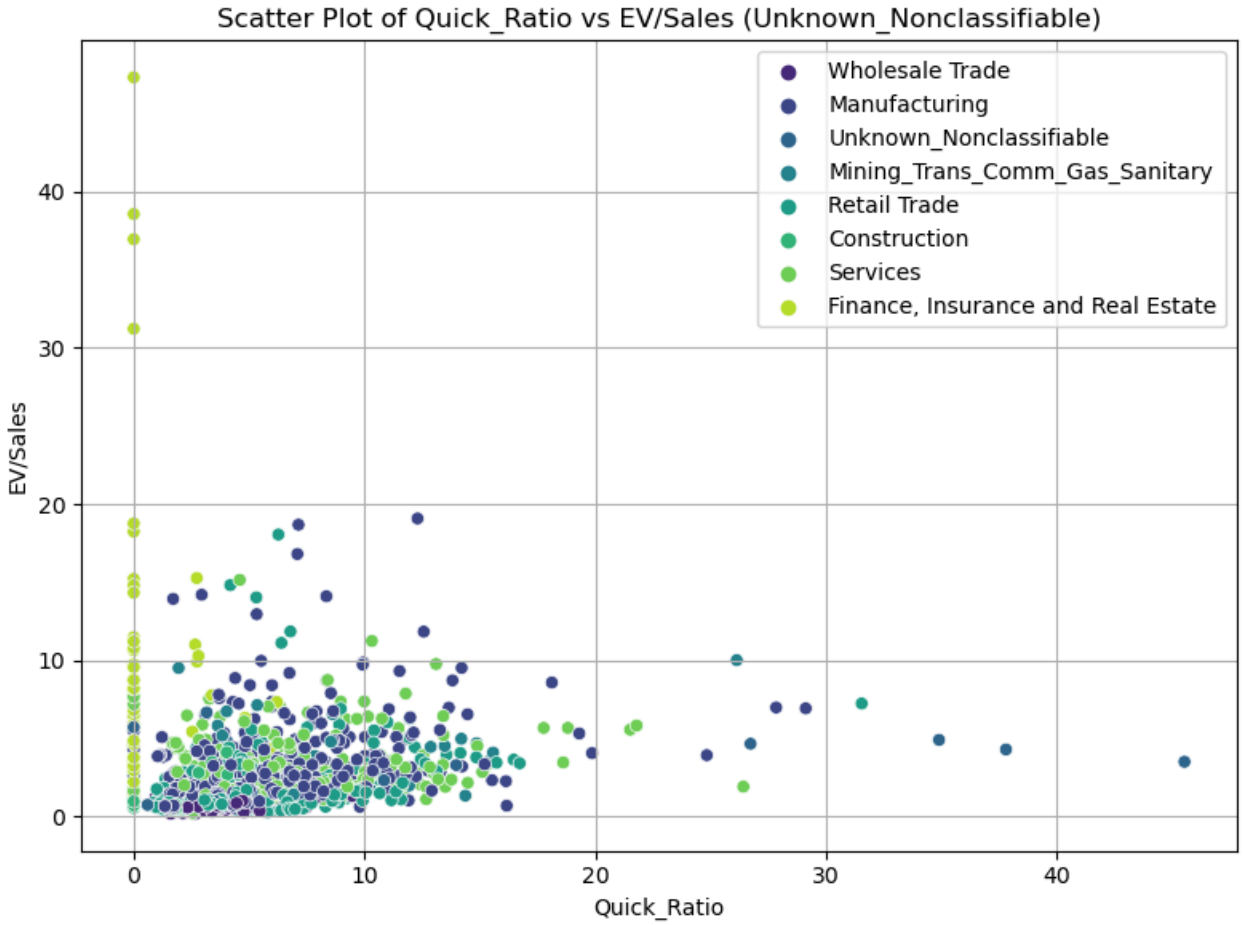


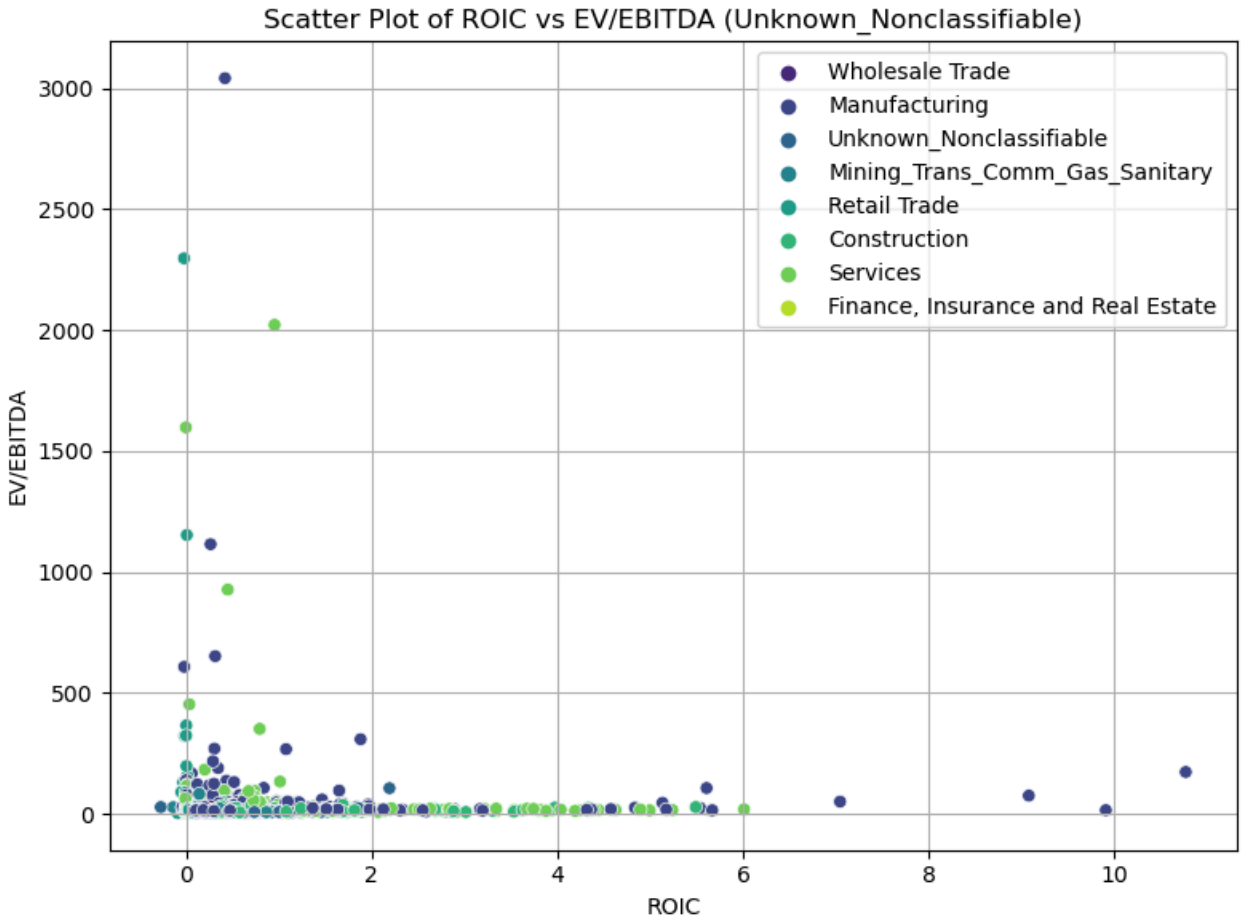

```
--- Unknown_Nonclassifiable ---
```



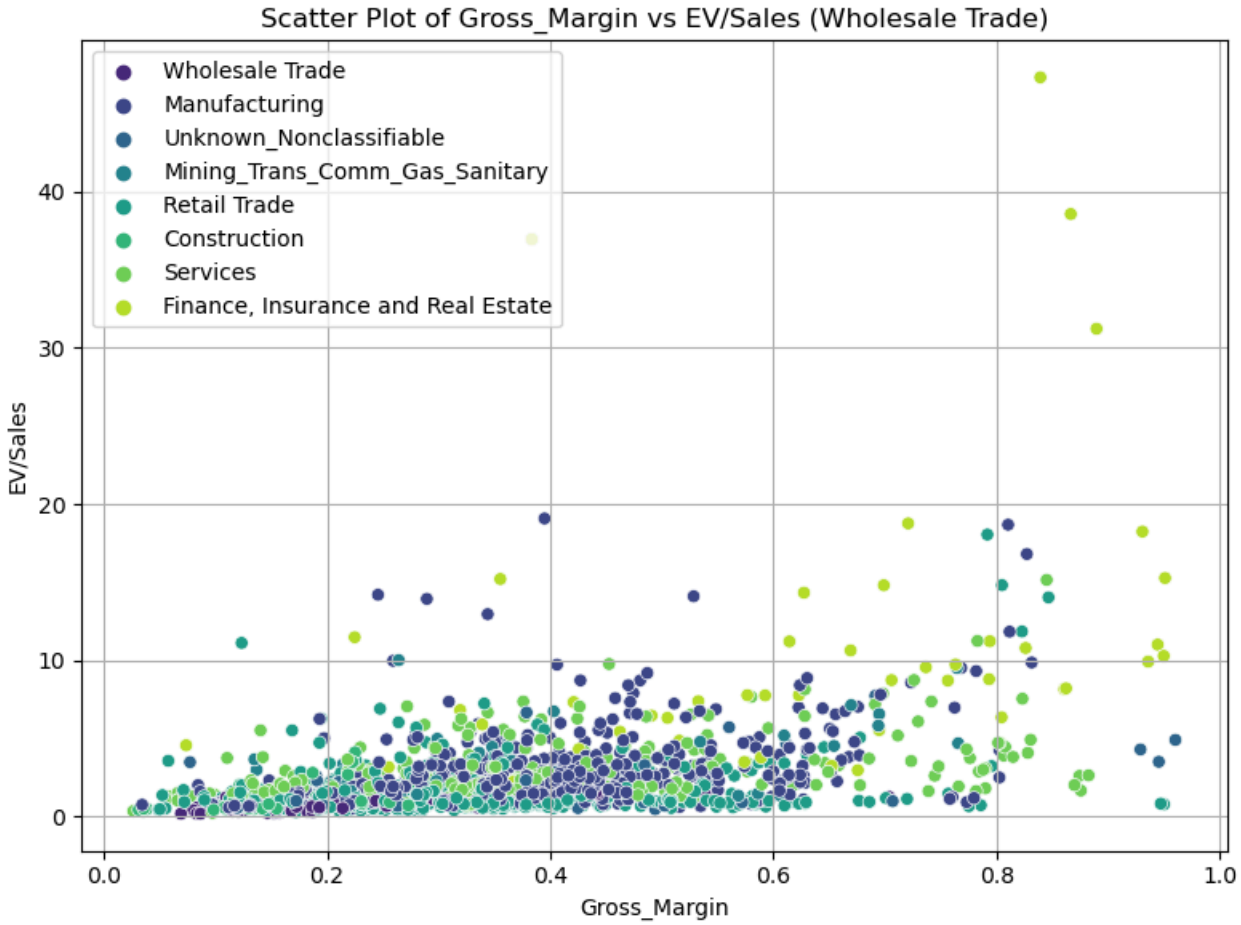

Scatter Plot of EBIT Margin (%) vs EV/Sales (Unknown_Nonclassifiable)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
EBIT Margin (%)
0
10
20
30
40
0
5
10
15
20

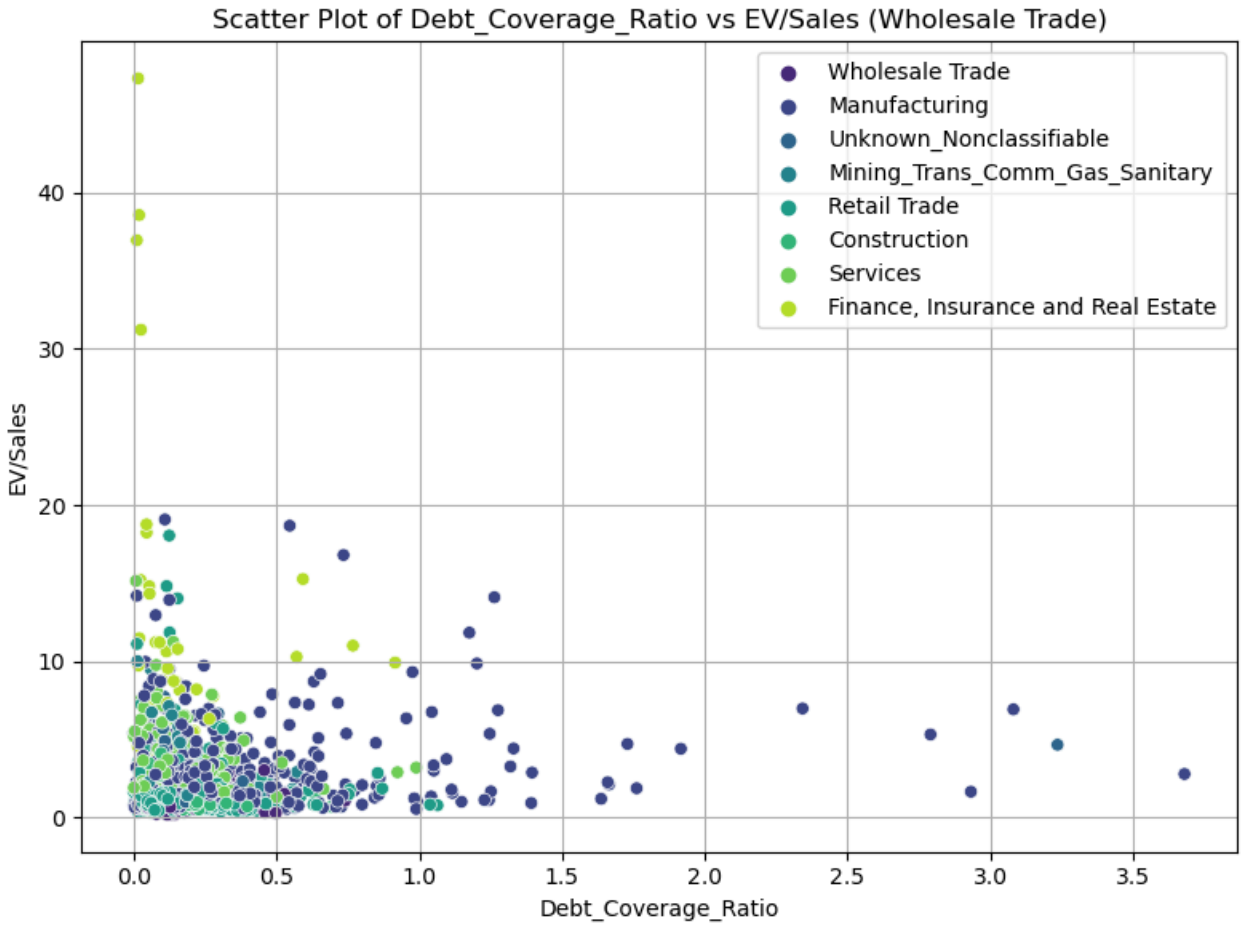

Scatter Plot of Revenue per Employee vs EV/Sales (Unknown_Nonclassifiable)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
Revenue per Employee
0
10
20
30
40
0
5000
10000
15000
20000
25000
30000
35000

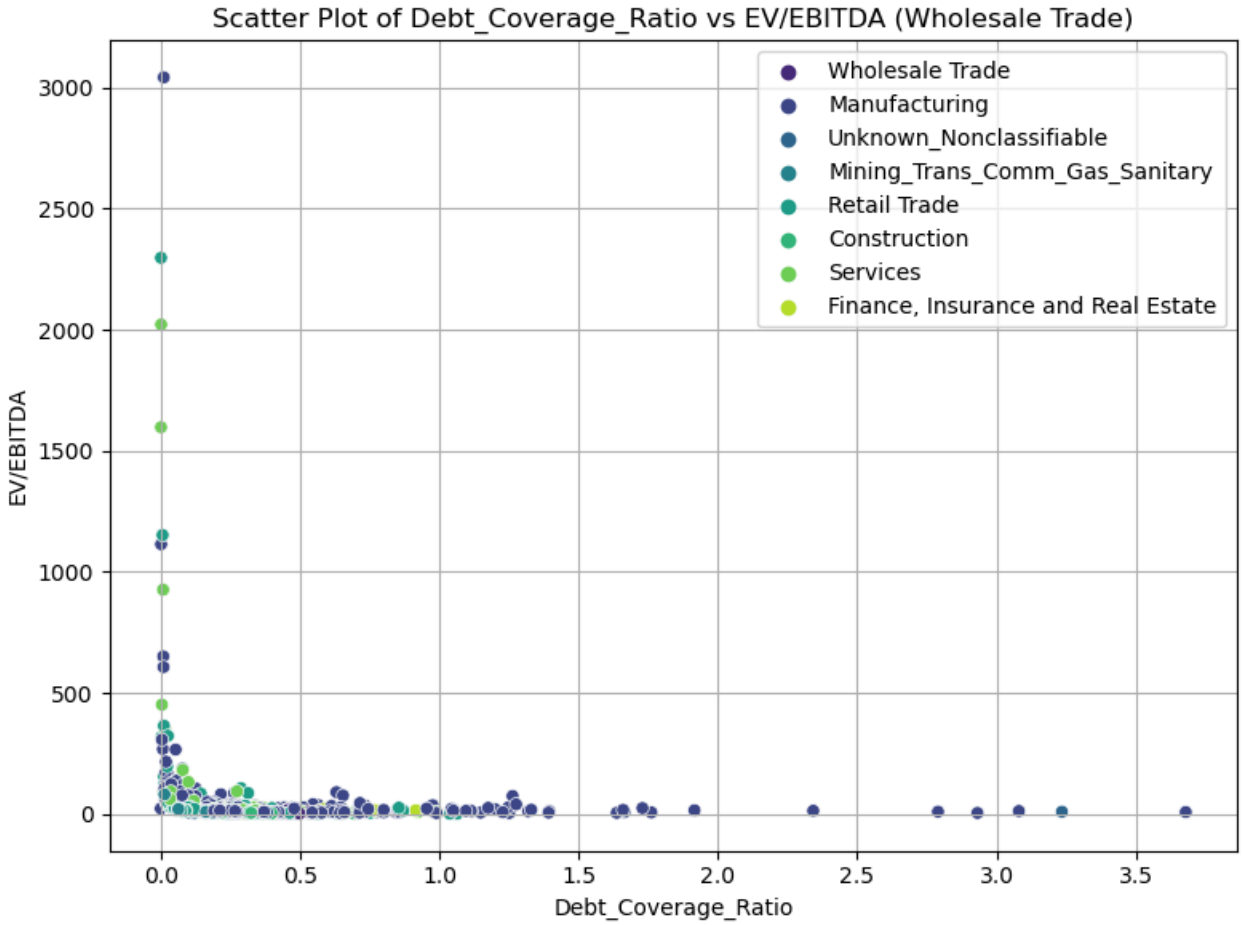


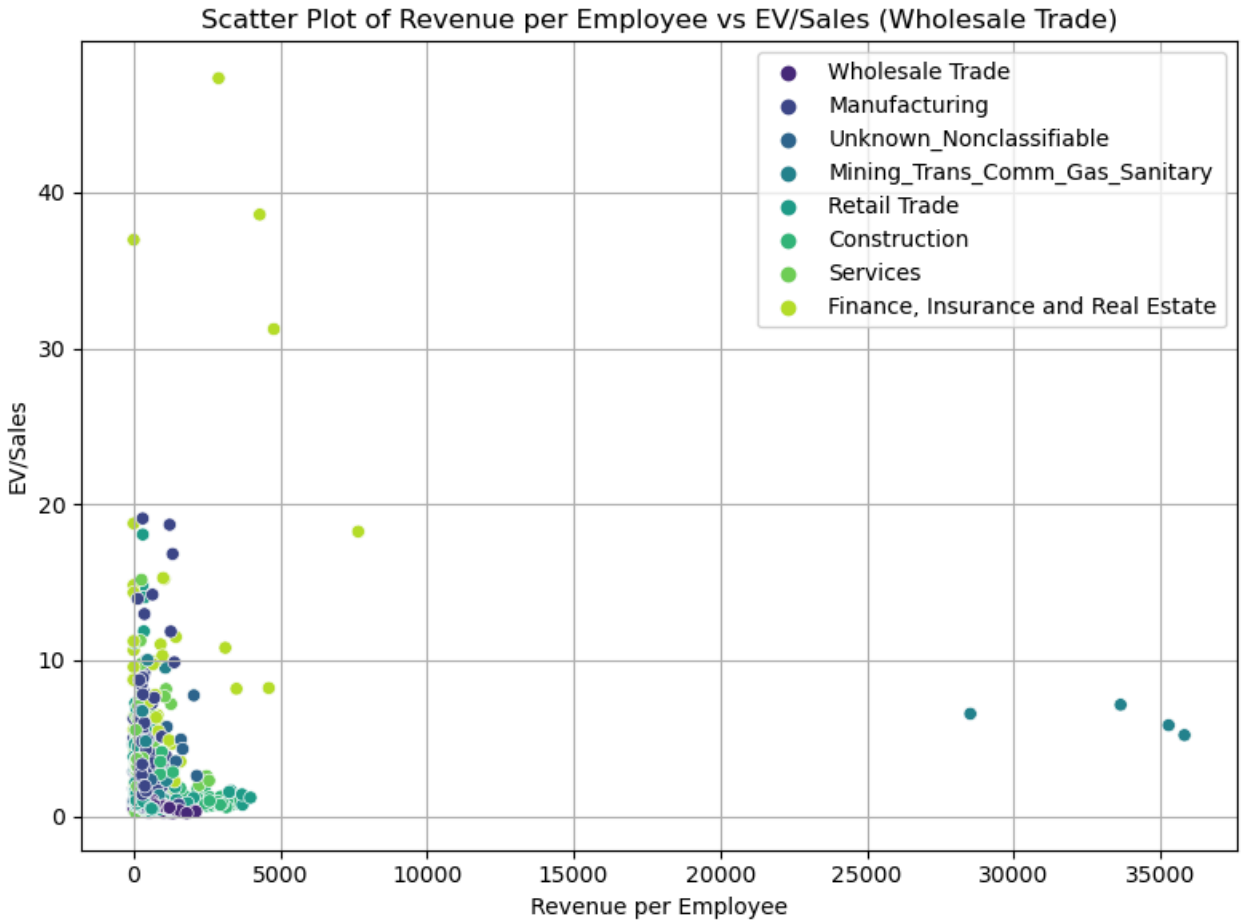


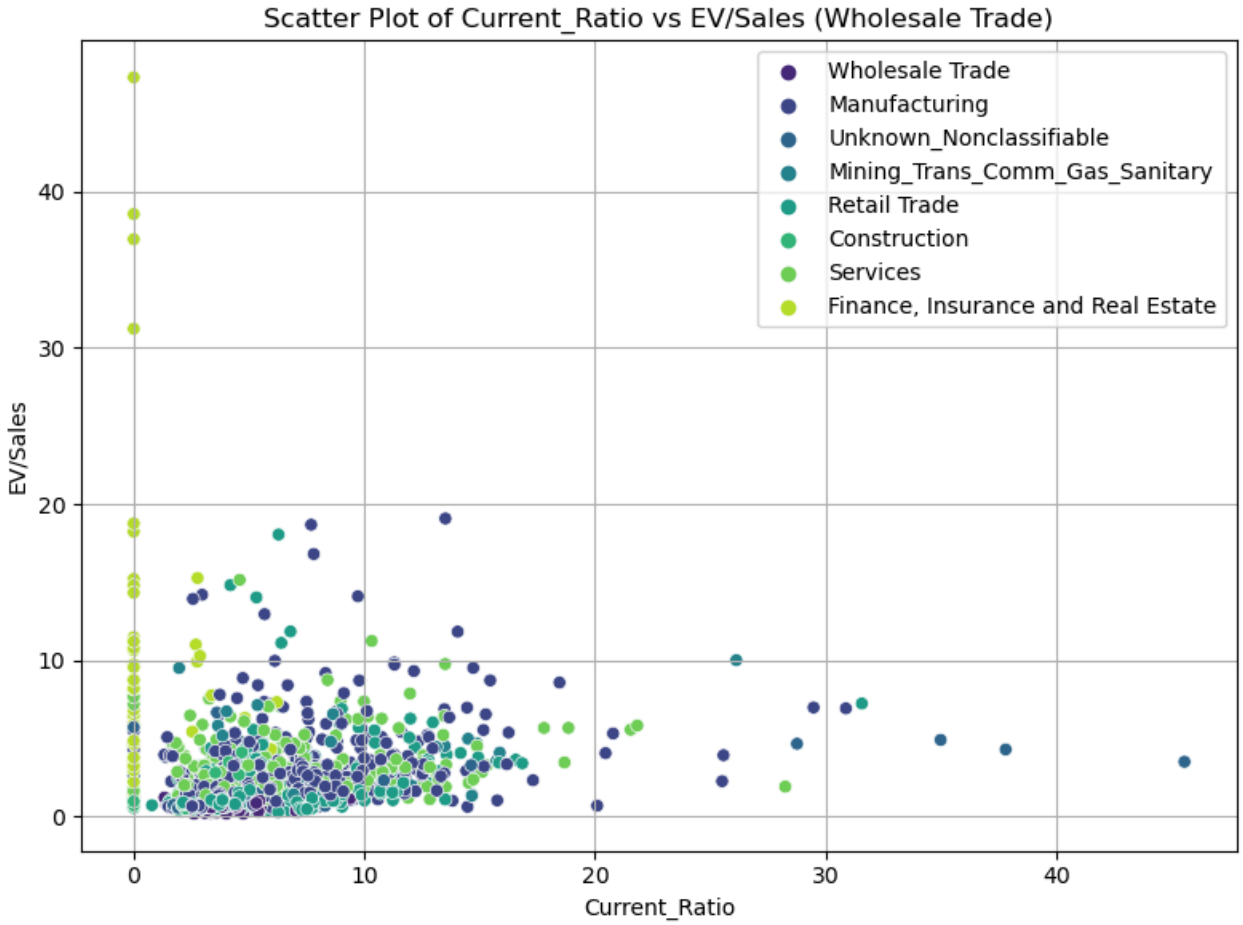

```
--- Wholesale Trade ---
```


Scatter Plot of Gross_Margin vs EV/Sales (Wholesale Trade)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
Gross_Margin
0
10
20
30
40
0.0
0.2
0.4
0.6
0.8
1.0


Scatter Plot of Debt_Coverage_Ratio vs EV/Sales (Wholesale Trade)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
Debt_Coverage_Ratio
0
10
20
30
40
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5


Scatter Plot of Debt_Coverage_Ratio vs EV/EBITDA (Wholesale Trade)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/EBITDA
Debt_Coverage_Ratio
0
500
1000
1500
2000
2500
3000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5


Scatter Plot of Revenue per Employee vs EV/Sales (Wholesale Trade)
Wholesale Trade
Manufacturing
Unknown_Nonclassifiable
Mining_Trans_Comm_Gas_Sanitary
Retail Trade
Construction
Services
Finance, Insurance and Real Estate
EV/Sales
Revenue per Employee
0
10
20
30
40
0
5000
10000
15000
20000
25000
30000
35000



Running Regressions (MLRs) between selected Multiples of Interest vs financial metrics

In [16]:

```
import statsmodels.api as sm

# Assuming 'selected_df' is the DataFrame containing the data

# Define the multiples (dependent variables) and metrics (independent variables)
multiples = ['EV/EBITDA','EV/Sales']  #multiples = ['EV/EBITDA', 'P/E', 'EV/GP','EV_EBIT','FCF Yi
metrics = ['Debt_Coverage_Ratio', 'ROE', 'EBIT Margin (%)',
           'YOY_Gross_Profit_Change','ROIC','FCF_Positive',
           'Gross_Margin','Quick_Ratio','Current_Ratio','Debt_to_Equity',
           'Revenue per Employee','FCF_Positive','YOY_Revenue_Change','YOY_EBIT_Change']

# Group data by Division
grouped_data = selected_df.groupby('Division')

# Perform MLR for each multiple by each industry
for division, group in grouped_data:
    print(f'\nDivision: {division}')

    # Drop non-numeric columns and keep only columns with multiples and metrics
    group = group[multiples + metrics].select_dtypes(include='number')

    for multiple in multiples:
        # Prepare the dependent (y) and independent (X) variables
        if multiple in group.columns:
            y = group[multiple].dropna()  # Drop missing values
            X = group.loc[y.index, metrics].dropna()

            # Add a constant to the model (intercept term)
```

```
            X = sm.add_constant(X)

            # Fit the model
            model = sm.OLS(y, X).fit()

            # Print the summary of the model
            print(f'\nMultiple: {multiple}')
            print(model.summary())
```

Division: Construction

Multiple: EV/EBITDA

OLS Regression Results

Dep. Variable:	EV/EBITDA	R-squared:	0.699
Model:	OLS	Adj. R-squared:	0.662
Method:	Least Squares	F-statistic:	18.76
Date:	Mon, 06 May 2024	Prob (F-statistic):	8.54e-22
Time:	17:49:14	Log-Likelihood:	-334.51
No. Observations:	119	AIC:	697.0
Df Residuals:	105	BIC:	735.9
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	16.4562	3.841	4.284	0.000	8.840	24.072
Debt_Coverage_Ratio	-7.2383	5.291	-1.368	0.174	-17.729	3.252
ROE	-18.8543	4.862	-3.878	0.000	-28.494	-9.214
EBIT Margin (%)	-2.0618	1.919	-1.075	0.285	-5.866	1.742
YOY_Gross_Profit_Change	-11.3433	3.366	-3.370	0.001	-18.017	-4.669
ROIC	1.3264	1.008	1.315	0.191	-0.673	3.326
FCF_Positive	-1.9021	0.565	-3.368	0.001	-3.022	-0.782
FCF_Positive	-1.9021	0.565	-3.368	0.001	-3.022	-0.782
Gross_Margin	8.2427	14.562	0.566	0.573	-20.630	37.116
Quick_Ratio	-4.3678	3.181	-1.373	0.173	-10.675	1.939
Current_Ratio	5.5941	3.051	1.833	0.070	-0.456	11.645
Debt_to_Equity	3.0080	0.780	3.858	0.000	1.462	4.554
Revenue per Employee	0.0008	0.001	0.785	0.434	-0.001	0.003
FCF_Positive	-1.9021	0.565	-3.368	0.001	-3.022	-0.782
FCF_Positive	-1.9021	0.565	-3.368	0.001	-3.022	-0.782
YOY_Revenue_Change	12.6295	4.374	2.887	0.005	3.957	21.302
YOY_EBIT_Change	2.3348	1.611	1.449	0.150	-0.859	5.529

Omnibus:	33.615	Durbin-Watson:	1.841
Prob(Omnibus):	0.000	Jarque-Bera (JB):	81.242
Skew:	1.081	Prob(JB):	2.28e-18
Kurtosis:	6.422	Cond. No.	2.55e+22

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 5.68e-37. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Multiple: EV/Sales

OLS Regression Results

Dep. Variable:	EV/Sales	R-squared:	0.703
Model:	OLS	Adj. R-squared:	0.667
Method:	Least Squares	F-statistic:	19.14
Date:	Mon, 06 May 2024	Prob (F-statistic):	4.19e-22
Time:	17:49:14	Log-Likelihood:	-31.981
No. Observations:	119	AIC:	91.96
Df Residuals:	105	BIC:	130.9
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.5717	0.302	-1.892	0.061	-1.171	0.028
Debt_Coverage_Ratio	-0.4278	0.416	-1.028	0.307	-1.253	0.398
ROE	-1.1562	0.383	-3.022	0.003	-1.915	-0.398
EBIT Margin (%)	0.6149	0.151	4.073	0.000	0.316	0.914
YOY_Gross_Profit_Change	-0.5080	0.265	-1.918	0.058	-1.033	0.017
ROIC	-0.2698	0.079	-3.400	0.001	-0.427	-0.112

FCF_Positive	-0.0430	0.044	-0.968	0.335	-0.131	0.045
FCF_Positive	-0.0430	0.044	-0.968	0.335	-0.131	0.045
Gross_Margin	6.2668	1.146	5.469	0.000	3.995	8.539
Quick_Ratio	-1.1394	0.250	-4.552	0.000	-1.636	-0.643
Current_Ratio	1.2375	0.240	5.154	0.000	0.761	1.714
Debt_to_Equity	0.1880	0.061	3.064	0.003	0.066	0.310
Revenue per Employee	0.0001	7.99e-05	1.485	0.141	-3.98e-05	0.000
FCF_Positive	-0.0430	0.044	-0.968	0.335	-0.131	0.045
FCF_Positive	-0.0430	0.044	-0.968	0.335	-0.131	0.045
YOY_Revenue_Change	0.6618	0.344	1.923	0.057	-0.021	1.344
YOY_EBIT_Change	0.0817	0.127	0.644	0.521	-0.170	0.333

Omnibus:	22.778	Durbin-Watson:	1.209
Prob(Omnibus):	0.000	Jarque-Bera (JB):	38.425
Skew:	0.863	Prob(JB):	4.53e-09
Kurtosis:	5.184	Cond. No.	2.55e+22

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 5.68e-37. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Division: Finance, Insurance and Real Estate

Multiple: EV/EBITDA

OLS Regression Results

Dep. Variable:	EV/EBITDA	R-squared:	0.365
Model:	OLS	Adj. R-squared:	0.163
Method:	Least Squares	F-statistic:	1.811
Date:	Mon, 06 May 2024	Prob (F-statistic):	0.0742
Time:	17:49:14	Log-Likelihood:	-221.85
No. Observations:	55	AIC:	471.7
Df Residuals:	41	BIC:	499.8
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	32.0787	9.288	3.454	0.001	13.322	50.836
Debt_Coverage_Ratio	-32.7677	29.253	-1.120	0.269	-91.846	26.310
ROE	-3.4598	1.510	-2.292	0.027	-6.509	-0.411
EBIT Margin (%)	6.1338	4.348	1.411	0.166	-2.648	14.916
YOY_Gross_Profit_Change	-6.3487	10.698	-0.593	0.556	-27.954	15.257
ROIC	4.4187	49.914	0.089	0.930	-96.386	105.223
FCF_Positive	-4.5011	2.294	-1.962	0.057	-9.134	0.131
FCF_Positive	-4.5011	2.294	-1.962	0.057	-9.134	0.131
Gross_Margin	7.2021	17.202	0.419	0.678	-27.538	41.943
Quick_Ratio	-18.1231	45.804	-0.396	0.694	-110.625	74.379
Current_Ratio	19.9893	45.457	0.440	0.662	-71.814	111.792
Debt_to_Equity	0.4468	0.210	2.127	0.039	0.023	0.871
Revenue per Employee	0.0005	0.002	0.239	0.812	-0.004	0.005
FCF_Positive	-4.5011	2.294	-1.962	0.057	-9.134	0.131
FCF_Positive	-4.5011	2.294	-1.962	0.057	-9.134	0.131
YOY_Revenue_Change	-7.2781	17.860	-0.408	0.686	-43.347	28.791
YOY_EBIT_Change	1.9217	4.222	0.455	0.651	-6.606	10.449

Omnibus:	40.800	Durbin-Watson:	1.675
Prob(Omnibus):	0.000	Jarque-Bera (JB):	163.408
Skew:	1.945	Prob(JB):	3.28e-36
Kurtosis:	10.495	Cond. No.	1.34e+22

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 9.81e-37. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

Multiple: EV/Sales

OLS Regression Results

Dep. Variable:	EV/Sales	R-squared:	0.576
Model:	OLS	Adj. R-squared:	0.441
Method:	Least Squares	F-statistic:	4.279
Date:	Mon, 06 May 2024	Prob (F-statistic):	0.000173
Time:	17:49:14	Log-Likelihood:	-176.36
No. Observations:	55	AIC:	380.7
Df Residuals:	41	BIC:	408.8
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-1.1875	4.061	-0.292	0.771	-9.389	7.014
Debt_Coverage_Ratio	-13.9303	12.791	-1.089	0.282	-39.763	11.902
ROE	-2.1126	0.660	-3.200	0.003	-3.446	-0.779
EBIT Margin (%)	4.4523	1.901	2.342	0.024	0.612	8.292
YOY_Gross_Profit_Change	-1.0150	4.678	-0.217	0.829	-10.462	8.432
ROIC	-6.7307	21.826	-0.308	0.759	-50.808	37.347
FCF_Positive	-0.7657	1.003	-0.763	0.450	-2.791	1.260
FCF_Positive	-0.7657	1.003	-0.763	0.450	-2.791	1.260
Gross_Margin	19.3049	7.522	2.567	0.014	4.114	34.496
Quick_Ratio	1.6883	20.028	0.084	0.933	-38.759	42.136
Current_Ratio	-0.4344	19.877	-0.022	0.983	-40.576	39.707
Debt_to_Equity	0.2989	0.092	3.254	0.002	0.113	0.484
Revenue per Employee	0.0012	0.001	1.312	0.197	-0.001	0.003
FCF_Positive	-0.7657	1.003	-0.763	0.450	-2.791	1.260
FCF_Positive	-0.7657	1.003	-0.763	0.450	-2.791	1.260
YOY_Revenue_Change	-4.1435	7.810	-0.531	0.599	-19.915	11.628
YOY_EBIT_Change	0.2179	1.846	0.118	0.907	-3.511	3.947

Omnibus:	20.894	Durbin-Watson:	1.322
Prob(Omnibus):	0.000	Jarque-Bera (JB):	36.916
Skew:	1.191	Prob(JB):	9.64e-09
Kurtosis:	6.230	Cond. No.	1.34e+22

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 9.81e-37. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Division: Manufacturing

Multiple: EV/EBITDA

OLS Regression Results

Dep. Variable:	EV/EBITDA	R-squared:	0.016
Model:	OLS	Adj. R-squared:	-0.001
Method:	Least Squares	F-statistic:	0.9171
Date:	Mon, 06 May 2024	Prob (F-statistic):	0.535
Time:	17:49:14	Log-Likelihood:	-4691.2
No. Observations:	753	AIC:	9410.
Df Residuals:	739	BIC:	9475.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	6.7887	36.968	0.184	0.854	-65.786	79.363
Debt_Coverage_Ratio	-25.9830	15.363	-1.691	0.091	-56.143	4.177
ROE	-7.3439	9.041	-0.812	0.417	-25.094	10.406
EBIT Margin (%)	5.1253	8.224	0.623	0.533	-11.019	21.270

YOY_Gross_Profit_Change	-32.5782	37.608	-0.866	0.387	-106.410	41.254
ROIC	-3.5850	12.666	-0.283	0.777	-28.451	21.281
FCF_Positive	3.7461	8.453	0.443	0.658	-12.849	20.341
FCF_Positive	3.7461	8.453	0.443	0.658	-12.849	20.341
Gross_Margin	30.2925	38.271	0.792	0.429	-44.840	105.425
Quick_Ratio	6.4231	7.298	0.880	0.379	-7.905	20.751
Current_Ratio	-6.6106	6.691	-0.988	0.323	-19.746	6.525
Debt_to_Equity	0.3709	0.521	0.711	0.477	-0.653	1.395
Revenue per Employee	0.0374	0.026	1.419	0.156	-0.014	0.089
FCF_Positive	3.7461	8.453	0.443	0.658	-12.849	20.341
FCF_Positive	3.7461	8.453	0.443	0.658	-12.849	20.341
YOY_Revenue_Change	15.9588	48.489	0.329	0.742	-79.234	111.151
YOY_EBIT_Change	0.1634	1.776	0.092	0.927	-3.323	3.650

Omnibus:	1740.777	Durbin-Watson:	1.964
Prob(Omnibus):	0.000	Jarque-Bera (JB):	7018017.629
Skew:	20.326	Prob(JB):	0.00
Kurtosis:	474.200	Cond. No.	2.24e+21

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.19e-35. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Multiple: EV/Sales

OLS Regression Results

Dep. Variable:	EV/Sales	R-squared:	0.411
Model:	OLS	Adj. R-squared:	0.400
Method:	Least Squares	F-statistic:	39.59
Date:	Mon, 06 May 2024	Prob (F-statistic):	5.60e-76
Time:	17:49:14	Log-Likelihood:	-1427.5
No. Observations:	753	AIC:	2883.
Df Residuals:	739	BIC:	2948.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.7061	0.485	-1.457	0.146	-1.658	0.245
Debt_Coverage_Ratio	-0.0739	0.201	-0.367	0.714	-0.469	0.322
ROE	0.8404	0.119	7.089	0.000	0.608	1.073
EBIT Margin (%)	0.4671	0.108	4.332	0.000	0.255	0.679
YOY_Gross_Profit_Change	-1.1569	0.493	-2.346	0.019	-2.125	-0.189
ROIC	-0.3896	0.166	-2.346	0.019	-0.716	-0.064
FCF_Positive	-0.1542	0.111	-1.391	0.165	-0.372	0.063
FCF_Positive	-0.1542	0.111	-1.391	0.165	-0.372	0.063
Gross_Margin	5.5401	0.502	11.041	0.000	4.555	6.525
Quick_Ratio	0.3815	0.096	3.987	0.000	0.194	0.569
Current_Ratio	-0.1974	0.088	-2.250	0.025	-0.370	-0.025
Debt_to_Equity	-0.0309	0.007	-4.525	0.000	-0.044	-0.018
Revenue per Employee	0.0020	0.000	5.725	0.000	0.001	0.003
FCF_Positive	-0.1542	0.111	-1.391	0.165	-0.372	0.063
FCF_Positive	-0.1542	0.111	-1.391	0.165	-0.372	0.063
YOY_Revenue_Change	2.1496	0.636	3.381	0.001	0.901	3.398
YOY_EBIT_Change	0.0110	0.023	0.472	0.637	-0.035	0.057

Omnibus:	568.195	Durbin-Watson:	1.100
Prob(Omnibus):	0.000	Jarque-Bera (JB):	12569.776
Skew:	3.187	Prob(JB):	0.00
Kurtosis:	21.974	Cond. No.	2.24e+21

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.19e-35. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

Division: Mining_Trans_Comm_Gas_Sanitary

Multiple: EV/EBITDA

OLS Regression Results

Dep. Variable:	EV/EBITDA	R-squared:	0.618
Model:	OLS	Adj. R-squared:	0.484
Method:	Least Squares	F-statistic:	4.613
Date:	Mon, 06 May 2024	Prob (F-statistic):	0.000121
Time:	17:49:14	Log-Likelihood:	-184.47
No. Observations:	51	AIC:	396.9
Df Residuals:	37	BIC:	424.0
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	36.8044	7.439	4.947	0.000	21.731	51.878
Debt_Coverage_Ratio	-19.7657	23.795	-0.831	0.411	-67.978	28.447
ROE	-55.3904	27.590	-2.008	0.052	-111.293	0.512
EBIT Margin (%)	-5.0273	2.980	-1.687	0.100	-11.066	1.012
YOY_Gross_Profit_Change	1.7479	4.965	0.352	0.727	-8.313	11.809
ROIC	9.9092	8.893	1.114	0.272	-8.110	27.928
FCF_Positive	-3.2541	1.373	-2.369	0.023	-6.037	-0.471
FCF_Positive	-3.2541	1.373	-2.369	0.023	-6.037	-0.471
Gross_Margin	-9.3699	14.524	-0.645	0.523	-38.799	20.059
Quick_Ratio	24.9881	12.436	2.009	0.052	-0.211	50.187
Current_Ratio	-24.0822	12.442	-1.936	0.061	-49.292	1.128
Debt_to_Equity	5.9856	3.328	1.798	0.080	-0.758	12.730
Revenue per Employee	3.862e-05	0.000	0.108	0.915	-0.001	0.001
FCF_Positive	-3.2541	1.373	-2.369	0.023	-6.037	-0.471
FCF_Positive	-3.2541	1.373	-2.369	0.023	-6.037	-0.471
YOY_Revenue_Change	-3.7874	2.721	-1.392	0.172	-9.300	1.725
YOY_EBIT_Change	-0.4092	3.601	-0.114	0.910	-7.706	6.888

Omnibus:	16.615	Durbin-Watson:	1.680
Prob(Omnibus):	0.000	Jarque-Bera (JB):	21.716
Skew:	1.150	Prob(JB):	1.93e-05
Kurtosis:	5.220	Cond. No.	6.12e+22

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.2e-36. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Multiple: EV/Sales

OLS Regression Results

Dep. Variable:	EV/Sales	R-squared:	0.754
Model:	OLS	Adj. R-squared:	0.668
Method:	Least Squares	F-statistic:	8.737
Date:	Mon, 06 May 2024	Prob (F-statistic):	9.58e-08
Time:	17:49:14	Log-Likelihood:	-74.513
No. Observations:	51	AIC:	177.0
Df Residuals:	37	BIC:	204.1
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	3.1779	0.861	3.689	0.001	1.433	4.923
Debt_Coverage_Ratio	-3.3319	2.755	-1.209	0.234	-8.914	2.250
ROE	-0.5415	3.195	-0.170	0.866	-7.014	5.931
EBIT Margin (%)	0.4852	0.345	1.406	0.168	-0.214	1.184

YOY_Gross_Profit_Change	1.1907	0.575	2.071	0.045	0.026	2.356
ROIC	-0.9721	1.030	-0.944	0.351	-3.058	1.114
FCF_Positive	-0.4994	0.159	-3.140	0.003	-0.822	-0.177
FCF_Positive	-0.4994	0.159	-3.140	0.003	-0.822	-0.177
Gross_Margin	2.1661	1.682	1.288	0.206	-1.241	5.574
Quick_Ratio	0.2588	1.440	0.180	0.858	-2.659	3.176
Current_Ratio	-0.1575	1.441	-0.109	0.914	-3.076	2.761
Debt_to_Equity	0.0414	0.385	0.107	0.915	-0.739	0.822
Revenue per Employee	-2.564e-06	4.15e-05	-0.062	0.951	-8.67e-05	8.16e-05
FCF_Positive	-0.4994	0.159	-3.140	0.003	-0.822	-0.177
FCF_Positive	-0.4994	0.159	-3.140	0.003	-0.822	-0.177
YOY_Revenue_Change	-0.7891	0.315	-2.505	0.017	-1.427	-0.151
YOY_EBIT_Change	-0.0676	0.417	-0.162	0.872	-0.913	0.777

Omnibus:	10.329	Durbin-Watson:	1.763
Prob(Omnibus):	0.006	Jarque-Bera (JB):	13.591
Skew:	0.662	Prob(JB):	0.00112
Kurtosis:	5.154	Cond. No.	6.12e+22

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.2e-36. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Division: Retail Trade

Multiple: EV/EBITDA

OLS Regression Results

Dep. Variable:	EV/EBITDA	R-squared:	0.045
Model:	OLS	Adj. R-squared:	0.014
Method:	Least Squares	F-statistic:	1.444
Date:	Mon, 06 May 2024	Prob (F-statistic):	0.136
Time:	17:49:14	Log-Likelihood:	-2578.7
No. Observations:	412	AIC:	5185.
Df Residuals:	398	BIC:	5242.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	64.7411	22.545	2.872	0.004	20.419	109.063
Debt_Coverage_Ratio	-129.3227	50.417	-2.565	0.011	-228.439	-30.206
ROE	0.2613	0.681	0.384	0.701	-1.078	1.601
EBIT Margin (%)	-675.8440	338.690	-1.995	0.047	-1341.690	-9.998
YOY_Gross_Profit_Change	-18.3181	42.086	-0.435	0.664	-101.056	64.420
ROIC	333.8206	203.753	1.638	0.102	-66.747	734.388
FCF_Positive	-4.2211	4.966	-0.850	0.396	-13.984	5.542
FCF_Positive	-4.2211	4.966	-0.850	0.396	-13.984	5.542
Gross_Margin	-16.1869	47.871	-0.338	0.735	-110.298	77.924
Quick_Ratio	-2.5837	14.799	-0.175	0.861	-31.677	26.510
Current_Ratio	5.5887	15.134	0.369	0.712	-24.163	35.341
Debt_to_Equity	-0.0605	0.134	-0.451	0.652	-0.324	0.203
Revenue per Employee	-0.0086	0.014	-0.610	0.542	-0.036	0.019
FCF_Positive	-4.2211	4.966	-0.850	0.396	-13.984	5.542
FCF_Positive	-4.2211	4.966	-0.850	0.396	-13.984	5.542
YOY_Revenue_Change	42.7090	51.544	0.829	0.408	-58.624	144.042
YOY_EBIT_Change	0.1711	2.371	0.072	0.943	-4.491	4.833

Omnibus:	847.708	Durbin-Watson:	1.955
Prob(Omnibus):	0.000	Jarque-Bera (JB):	953959.872
Skew:	14.456	Prob(JB):	0.00
Kurtosis:	236.954	Cond. No.	2.86e+21

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.72e-35. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Multiple: EV/Sales

OLS Regression Results

Dep. Variable:	EV/Sales	R-squared:	0.389
Model:	OLS	Adj. R-squared:	0.370
Method:	Least Squares	F-statistic:	19.53
Date:	Mon, 06 May 2024	Prob (F-statistic):	2.20e-35
Time:	17:49:14	Log-Likelihood:	-728.08
No. Observations:	412	AIC:	1484.
Df Residuals:	398	BIC:	1540.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0224	0.253	-0.089	0.929	-0.519	0.474
Debt_Coverage_Ratio	-1.2024	0.565	-2.129	0.034	-2.312	-0.092
ROE	0.0054	0.008	0.704	0.482	-0.010	0.020
EBIT Margin (%)	25.2925	3.793	6.668	0.000	17.835	32.750
YOY_Gross_Profit_Change	1.0740	0.471	2.279	0.023	0.147	2.001
ROIC	-12.9886	2.282	-5.692	0.000	-17.475	-8.502
FCF_Positive	-0.0138	0.056	-0.247	0.805	-0.123	0.096
FCF_Positive	-0.0138	0.056	-0.247	0.805	-0.123	0.096
Gross_Margin	3.6361	0.536	6.782	0.000	2.582	4.690
Quick_Ratio	0.7436	0.166	4.487	0.000	0.418	1.069
Current_Ratio	-0.5901	0.169	-3.481	0.001	-0.923	-0.257
Debt_to_Equity	-0.0003	0.002	-0.209	0.835	-0.003	0.003
Revenue per Employee	0.0002	0.000	1.120	0.263	-0.000	0.000
FCF_Positive	-0.0138	0.056	-0.247	0.805	-0.123	0.096
FCF_Positive	-0.0138	0.056	-0.247	0.805	-0.123	0.096
YOY_Revenue_Change	-0.5355	0.577	-0.928	0.354	-1.670	0.599
YOY_EBIT_Change	-0.0293	0.027	-1.105	0.270	-0.082	0.023

Omnibus:	405.722	Durbin-Watson:	0.804
Prob(Omnibus):	0.000	Jarque-Bera (JB):	16445.673
Skew:	4.210	Prob(JB):	0.00
Kurtosis:	32.784	Cond. No.	2.86e+21

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.72e-35. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Division: Services

Multiple: EV/EBITDA

OLS Regression Results

Dep. Variable:	EV/EBITDA	R-squared:	0.096
Model:	OLS	Adj. R-squared:	0.057
Method:	Least Squares	F-statistic:	2.464
Date:	Mon, 06 May 2024	Prob (F-statistic):	0.00337
Time:	17:49:14	Log-Likelihood:	-2033.0
No. Observations:	317	AIC:	4094.
Df Residuals:	303	BIC:	4147.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	50.4699	36.410	1.386	0.167	-21.178	122.118
Debt_Coverage_Ratio	-231.8618	76.229	-3.042	0.003	-381.867	-81.857

ROE	-10.3972	8.654	-1.201	0.231	-27.426	6.632
EBIT Margin (%)	35.8724	13.304	2.696	0.007	9.692	62.053
YOY_Gross_Profit_Change	-104.9760	59.254	-1.772	0.077	-221.577	11.625
ROIC	-34.9696	14.329	-2.441	0.015	-63.166	-6.773
FCF_Positive	5.4283	6.404	0.848	0.397	-7.174	18.030
FCF_Positive	5.4283	6.404	0.848	0.397	-7.174	18.030
Gross_Margin	54.1312	47.337	1.144	0.254	-39.020	147.282
Quick_Ratio	15.9173	34.429	0.462	0.644	-51.832	83.667
Current_Ratio	-16.0762	34.304	-0.469	0.640	-83.581	51.429
Debt_to_Equity	0.8861	0.757	1.170	0.243	-0.604	2.376
Revenue per Employee	-0.0291	0.030	-0.980	0.328	-0.087	0.029
FCF_Positive	5.4283	6.404	0.848	0.397	-7.174	18.030
FCF_Positive	5.4283	6.404	0.848	0.397	-7.174	18.030
YOY_Revenue_Change	71.8924	57.792	1.244	0.214	-41.832	185.617
YOY_EBIT_Change	-3.4545	5.587	-0.618	0.537	-14.450	7.540

Omnibus:	531.738	Durbin-Watson:	1.646
Prob(Omnibus):	0.000	Jarque-Bera (JB):	136406.146
Skew:	9.324	Prob(JB):	0.00
Kurtosis:	102.898	Cond. No.	5.36e+21

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.74e-36. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Multiple: EV/Sales

OLS Regression Results

Dep. Variable:	EV/Sales	R-squared:	0.377
Model:	OLS	Adj. R-squared:	0.351
Method:	Least Squares	F-statistic:	14.13
Date:	Mon, 06 May 2024	Prob (F-statistic):	1.25e-24
Time:	17:49:14	Log-Likelihood:	-571.30
No. Observations:	317	AIC:	1171.
Df Residuals:	303	BIC:	1223.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.5696	0.362	4.336	0.000	0.857	2.282
Debt_Coverage_Ratio	-1.2140	0.758	-1.602	0.110	-2.705	0.278
ROE	-0.0335	0.086	-0.390	0.697	-0.203	0.136
EBIT Margin (%)	0.8817	0.132	6.665	0.000	0.621	1.142
YOY_Gross_Profit_Change	0.6731	0.589	1.143	0.254	-0.486	1.832
ROIC	-0.9039	0.142	-6.345	0.000	-1.184	-0.624
FCF_Positive	-0.0748	0.064	-1.174	0.241	-0.200	0.051
FCF_Positive	-0.0748	0.064	-1.174	0.241	-0.200	0.051
Gross_Margin	3.2423	0.471	6.889	0.000	2.316	4.168
Quick_Ratio	0.2336	0.342	0.682	0.495	-0.440	0.907
Current_Ratio	-0.1796	0.341	-0.527	0.599	-0.851	0.492
Debt_to_Equity	0.0028	0.008	0.366	0.715	-0.012	0.018
Revenue per Employee	3.656e-05	0.000	0.124	0.901	-0.001	0.001
FCF_Positive	-0.0748	0.064	-1.174	0.241	-0.200	0.051
FCF_Positive	-0.0748	0.064	-1.174	0.241	-0.200	0.051
YOY_Revenue_Change	-0.5427	0.575	-0.944	0.346	-1.673	0.588
YOY_EBIT_Change	-0.0477	0.056	-0.858	0.392	-0.157	0.062

Omnibus:	157.976	Durbin-Watson:	1.201
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1007.189
Skew:	1.986	Prob(JB):	1.96e-219
Kurtosis:	10.776	Cond. No.	5.36e+21

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.74e-36. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Division: Unknown_Nonclassifiable

Multiple: EV/EBITDA

OLS Regression Results

Dep. Variable:	EV/EBITDA	R-squared:	0.374
Model:	OLS	Adj. R-squared:	0.176
Method:	Least Squares	F-statistic:	1.888
Date:	Mon, 06 May 2024	Prob (F-statistic):	0.0612
Time:	17:49:14	Log-Likelihood:	-215.15
No. Observations:	55	AIC:	458.3
Df Residuals:	41	BIC:	486.4
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	25.6354	7.473	3.430	0.001	10.543	40.728
Debt_Coverage_Ratio	-2.5755	5.383	-0.478	0.635	-13.447	8.296
ROE	-5.7160	6.296	-0.908	0.369	-18.431	6.998
EBIT Margin (%)	-7.9273	4.069	-1.948	0.058	-16.145	0.291
YOY_Gross_Profit_Change	5.2907	15.521	0.341	0.735	-26.055	36.636
ROIC	16.6498	6.453	2.580	0.014	3.618	29.681
FCF_Positive	-1.6827	1.349	-1.247	0.219	-4.407	1.041
FCF_Positive	-1.6827	1.349	-1.247	0.219	-4.407	1.041
Gross_Margin	-5.1690	15.178	-0.341	0.735	-35.821	25.483
Quick_Ratio	3.8559	3.723	1.036	0.306	-3.664	11.375
Current_Ratio	-3.7730	3.694	-1.021	0.313	-11.234	3.688
Debt_to_Equity	0.1540	0.557	0.276	0.784	-0.971	1.279
Revenue per Employee	0.0054	0.005	1.083	0.285	-0.005	0.016
FCF_Positive	-1.6827	1.349	-1.247	0.219	-4.407	1.041
FCF_Positive	-1.6827	1.349	-1.247	0.219	-4.407	1.041
YOY_Revenue_Change	-19.6364	18.166	-1.081	0.286	-56.323	17.050
YOY_EBIT_Change	-1.0566	1.191	-0.887	0.380	-3.462	1.349

Omnibus:	36.565	Durbin-Watson:	2.015
Prob(Omnibus):	0.000	Jarque-Bera (JB):	119.277
Skew:	1.803	Prob(JB):	1.26e-26
Kurtosis:	9.249	Cond. No.	9.23e+20

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 3.02e-35. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Multiple: EV/Sales

OLS Regression Results

Dep. Variable:	EV/Sales	R-squared:	0.648
Model:	OLS	Adj. R-squared:	0.536
Method:	Least Squares	F-statistic:	5.801
Date:	Mon, 06 May 2024	Prob (F-statistic):	6.86e-06
Time:	17:49:14	Log-Likelihood:	-75.837
No. Observations:	55	AIC:	179.7
Df Residuals:	41	BIC:	207.8
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.3129	0.594	2.212	0.033	0.114	2.512
Debt_Coverage_Ratio	1.6087	0.428	3.763	0.001	0.745	2.472

ROE	-0.3757	0.500	-0.751	0.457	-1.386	0.634
EBIT Margin (%)	1.3802	0.323	4.270	0.000	0.728	2.033
YOY_Gross_Profit_Change	0.0022	1.233	0.002	0.999	-2.487	2.492
ROIC	-1.5655	0.513	-3.055	0.004	-2.601	-0.530
FCF_Positive	-0.0652	0.107	-0.609	0.546	-0.282	0.151
FCF_Positive	-0.0652	0.107	-0.609	0.546	-0.282	0.151
Gross_Margin	1.7284	1.206	1.434	0.159	-0.706	4.163
Quick_Ratio	0.4970	0.296	1.681	0.100	-0.100	1.094
Current_Ratio	-0.5694	0.293	-1.940	0.059	-1.162	0.023
Debt_to_Equity	0.0164	0.044	0.371	0.712	-0.073	0.106
Revenue per Employee	0.0009	0.000	2.282	0.028	0.000	0.002
FCF_Positive	-0.0652	0.107	-0.609	0.546	-0.282	0.151
FCF_Positive	-0.0652	0.107	-0.609	0.546	-0.282	0.151
YOY_Revenue_Change	-0.0610	1.443	-0.042	0.966	-2.975	2.853
YOY_EBIT_Change	0.1193	0.095	1.260	0.215	-0.072	0.310

Omnibus:	39.108	Durbin-Watson:	2.125
Prob(Omnibus):	0.000	Jarque-Bera (JB):	122.340
Skew:	1.998	Prob(JB):	2.72e-27
Kurtosis:	9.116	Cond. No.	9.23e+20

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 3.02e-35. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Division: Wholesale Trade

Multiple: EV/EBITDA

OLS Regression Results

Dep. Variable:	EV/EBITDA	R-squared:	0.308
Model:	OLS	Adj. R-squared:	0.207
Method:	Least Squares	F-statistic:	3.052
Date:	Mon, 06 May 2024	Prob (F-statistic):	0.000924
Time:	17:49:14	Log-Likelihood:	-316.68
No. Observations:	103	AIC:	661.4
Df Residuals:	89	BIC:	698.2
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	8.6250	5.221	1.652	0.102	-1.749	18.999
Debt_Coverage_Ratio	-10.8287	8.130	-1.332	0.186	-26.983	5.325
ROE	-11.3560	4.866	-2.334	0.022	-21.024	-1.688
EBIT Margin (%)	-15.9390	19.983	-0.798	0.427	-55.645	23.767
YOY_Gross_Profit_Change	-4.7974	6.179	-0.776	0.440	-17.075	7.480
ROIC	9.2691	14.178	0.654	0.515	-18.903	37.441
FCF_Positive	1.5275	1.041	1.467	0.146	-0.541	3.597
FCF_Positive	1.5275	1.041	1.467	0.146	-0.541	3.597
Gross_Margin	6.1713	7.344	0.840	0.403	-8.422	20.764
Quick_Ratio	-0.5511	1.481	-0.372	0.711	-3.495	2.392
Current_Ratio	1.0651	1.372	0.776	0.440	-1.662	3.792
Debt_to_Equity	1.5792	0.685	2.304	0.024	0.217	2.941
Revenue per Employee	-0.0013	0.002	-0.648	0.519	-0.005	0.003
FCF_Positive	1.5275	1.041	1.467	0.146	-0.541	3.597
FCF_Positive	1.5275	1.041	1.467	0.146	-0.541	3.597
YOY_Revenue_Change	-1.0536	5.039	-0.209	0.835	-11.066	8.958
YOY_EBIT_Change	0.2889	0.266	1.084	0.281	-0.241	0.818

Omnibus:	25.663	Durbin-Watson:	1.527
Prob(Omnibus):	0.000	Jarque-Bera (JB):	38.351
Skew:	1.150	Prob(JB):	4.70e-09
Kurtosis:	4.909	Cond. No.	1.14e+23

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 6.58e-39. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Multiple: EV/Sales
                            OLS Regression Results
==============================================================================
Dep. Variable:                EV/Sales   R-squared:                       0.514
Model:                            OLS   Adj. R-squared:                  0.443
Method:                 Least Squares   F-statistic:                     7.233
Date:                Mon, 06 May 2024   Prob (F-statistic):           2.00e-09
Time:                        17:49:14   Log-Likelihood:                -41.836
No. Observations:                 103   AIC:                             111.7
Df Residuals:                      89   BIC:                             148.6
Df Model:                          13
Covariance Type:            nonrobust
================================================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                          0.2883      0.362      0.796      0.428      -0.431       1.008
Debt_Coverage_Ratio            1.0828      0.564      1.920      0.058      -0.038       2.203
ROE                            0.1936      0.338      0.574      0.568      -0.477       0.864
EBIT Margin (%)                0.6711      1.386      0.484      0.629      -2.083       3.425
YOY_Gross_Profit_Change       -0.0849      0.429     -0.198      0.843      -0.937       0.767
ROIC                          -0.8526      0.983     -0.867      0.388      -2.807       1.102
FCF_Positive                   0.0320      0.072      0.443      0.659      -0.112       0.175
FCF_Positive                   0.0320      0.072      0.443      0.659      -0.112       0.175
Gross_Margin                   1.8624      0.509      3.656      0.000       0.850       2.875
Quick_Ratio                    0.0705      0.103      0.686      0.494      -0.134       0.275
Current_Ratio                 -0.0349      0.095     -0.367      0.714      -0.224       0.154
Debt_to_Equity                -0.0286      0.048     -0.602      0.549      -0.123       0.066
Revenue per Employee          -0.0003      0.000     -1.989      0.050      -0.001    -3.3e-07
FCF_Positive                   0.0320      0.072      0.443      0.659      -0.112       0.175
FCF_Positive                   0.0320      0.072      0.443      0.659      -0.112       0.175
YOY_Revenue_Change            -0.0448      0.350     -0.128      0.898      -0.739       0.650
YOY_EBIT_Change                0.0117      0.018      0.632      0.529      -0.025       0.048
==============================================================================
Omnibus:                       57.412   Durbin-Watson:                   1.119
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              235.145
Skew:                           1.874   Prob(JB):                     8.69e-52
Kurtosis:                       9.383   Cond. No.                     1.14e+23
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 6.58e-39. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Feature Importance

- identifying what financial metrics drive/influence our multiples of interest
- also doing this at the industry level

In [17]:
```
from sklearn.model_selection import train_test_split
from interpret.glassbox import ExplainableBoostingRegressor
from interpret import show
```

EV/EBITDA

In [18]:

```
from sklearn.model_selection import train_test_split
from interpret.glassbox import ExplainableBoostingRegressor
from interpret import show

# Adjust the data by replacing the division mapping
# This assumes you have already run the division grouping mapping mentioned earlier

# Loop through each unique Division and compute feature importance
for division, group in selected_df.groupby('Division'):
    print(f"\nDivision: {division}")

    # Prepare features (X) and target (y)
    X = group.drop(columns=['Ticker', 'Division','EV/EBITDA', 'P/E', 'EV/GP','EV_EBIT','EV/Sales'
    y = group['EV/EBITDA']

    # Check if there are enough samples to perform a split and modeling
    if len(y) < 10:
        print(f"Not enough samples to perform feature selection for {division}")
        continue

    # Split the dataset into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42)

    # Initialize the ExplainableBoostingRegressor
    ebm = ExplainableBoostingRegressor(random_state=42)

    # Fit the model
    ebm.fit(X_train, y_train)

    # Get the global feature importances from the model
    ebm_global = ebm.explain_global()
    show(ebm_global)
```

```
Division: Construction
```

Division: Finance, Insurance and Real Estate

Division: Manufacturing

Division: Mining_Trans_Comm_Gas_Sanitary

Division: Retail Trade

Division: Services

Division: Unknown_Nonclassifiable

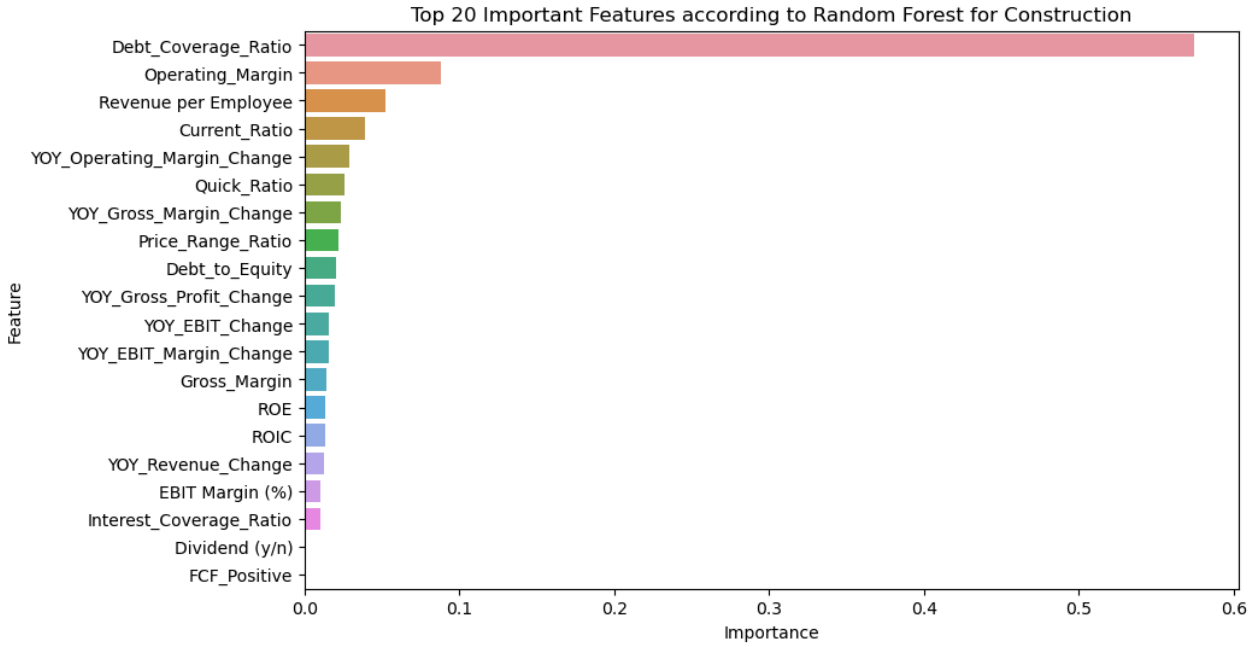
Division: Wholesale Trade

In [19]:

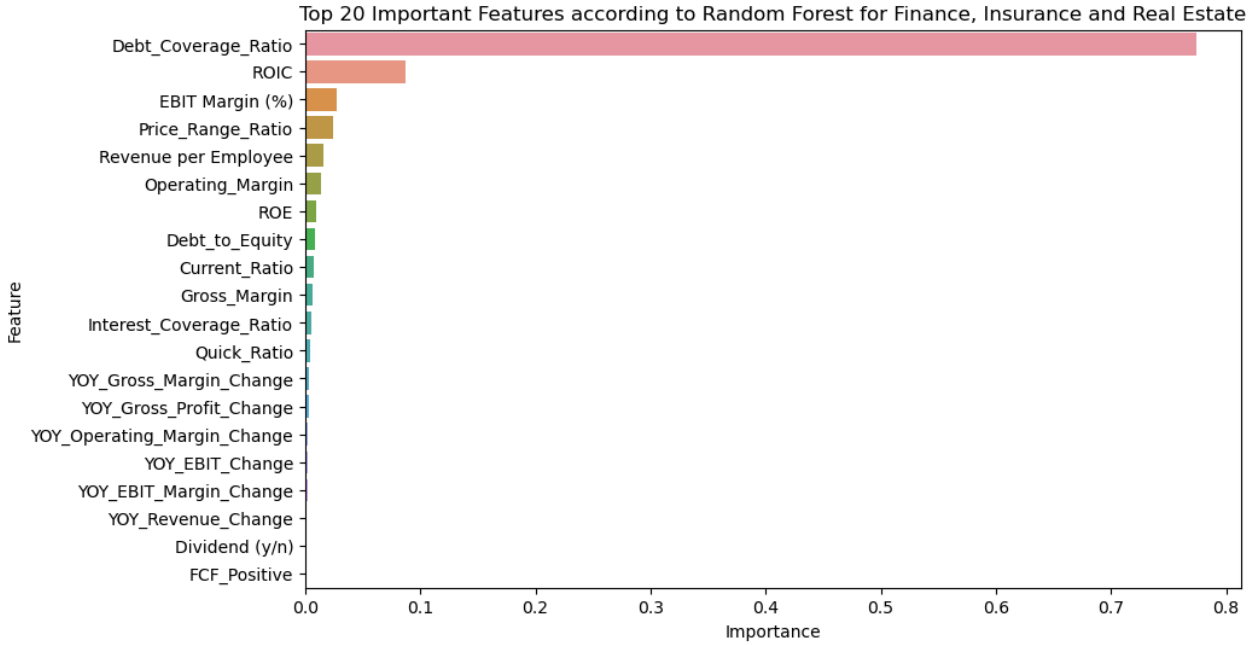
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
import seaborn as sns
import matplotlib.pyplot as plt

# Group the data by 'Division'
for division, group in selected_df.groupby('Division'):
    print(f"\nDivision: {division}")

    # Prepare the features (X) and target (y)
    X = group.drop(columns=['Ticker', 'Division', 'EV/EBITDA', 'P/E', 'EV/GP', 'EV_EBIT','EV/Sale
    y = group['EV/EBITDA']

    # Check if there are enough samples to train the model
    if len(y) < 10:
        print(f"Not enough samples to perform feature selection for {division}")
```

```
        continue

    # Scale the features
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    # Use RandomForestRegressor to get feature importances
    rf = RandomForestRegressor(random_state=42)
    rf.fit(X_scaled, y)
    feature_importances = rf.feature_importances_

    # Plot feature importances
    features = pd.DataFrame({'Feature': X.columns, 'Importance': feature_importances})
    features = features.sort_values(by='Importance', ascending=False)

    plt.figure(figsize=(10, 6))
    sns.barplot(x='Importance', y='Feature', data=features.head(20))
    plt.title(f'Top 20 Important Features according to Random Forest for {division}')
    plt.show()
```

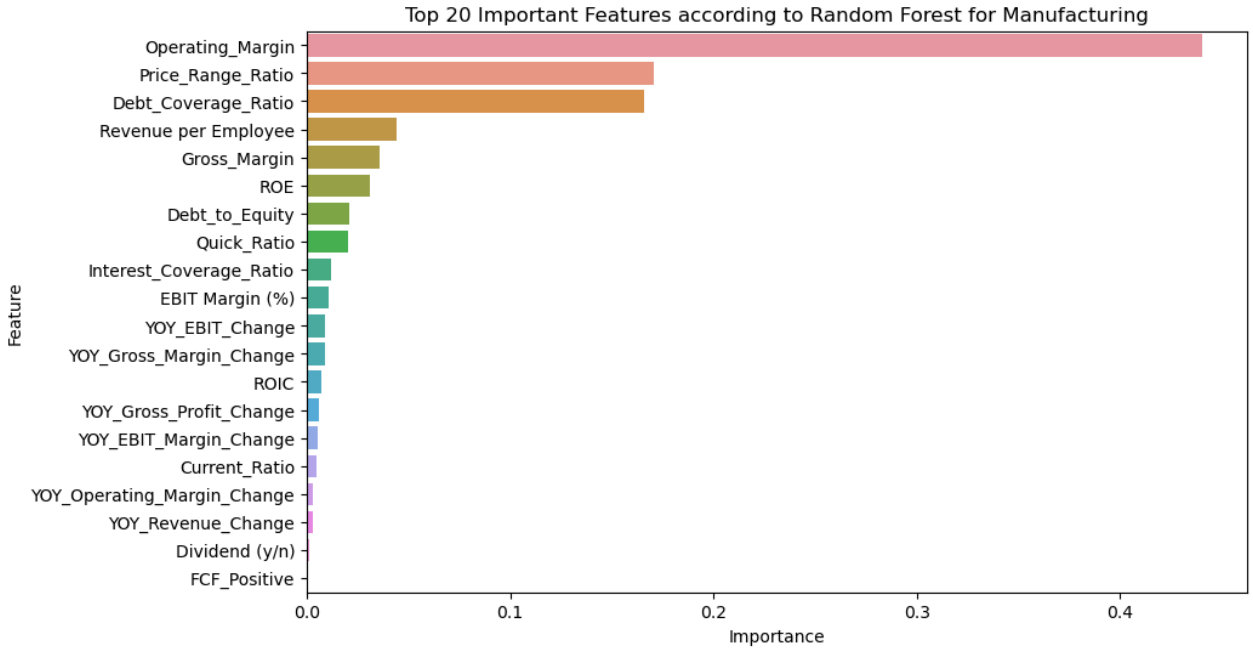
Division: Construction



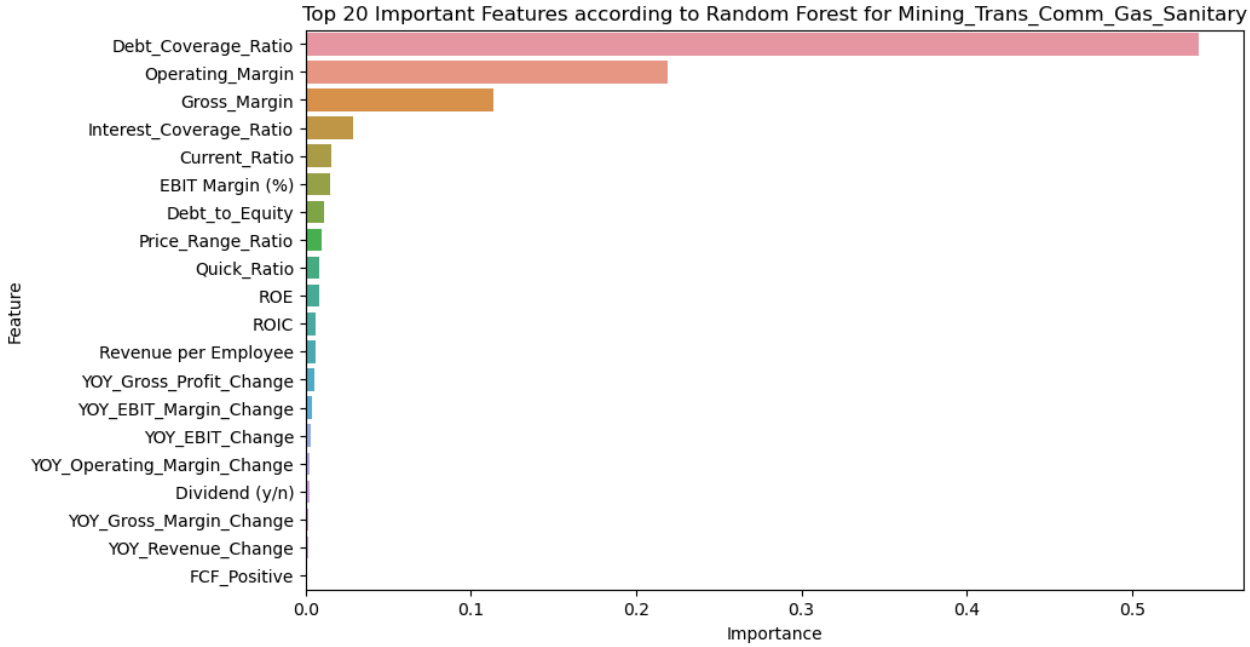
Division: Finance, Insurance and Real Estate



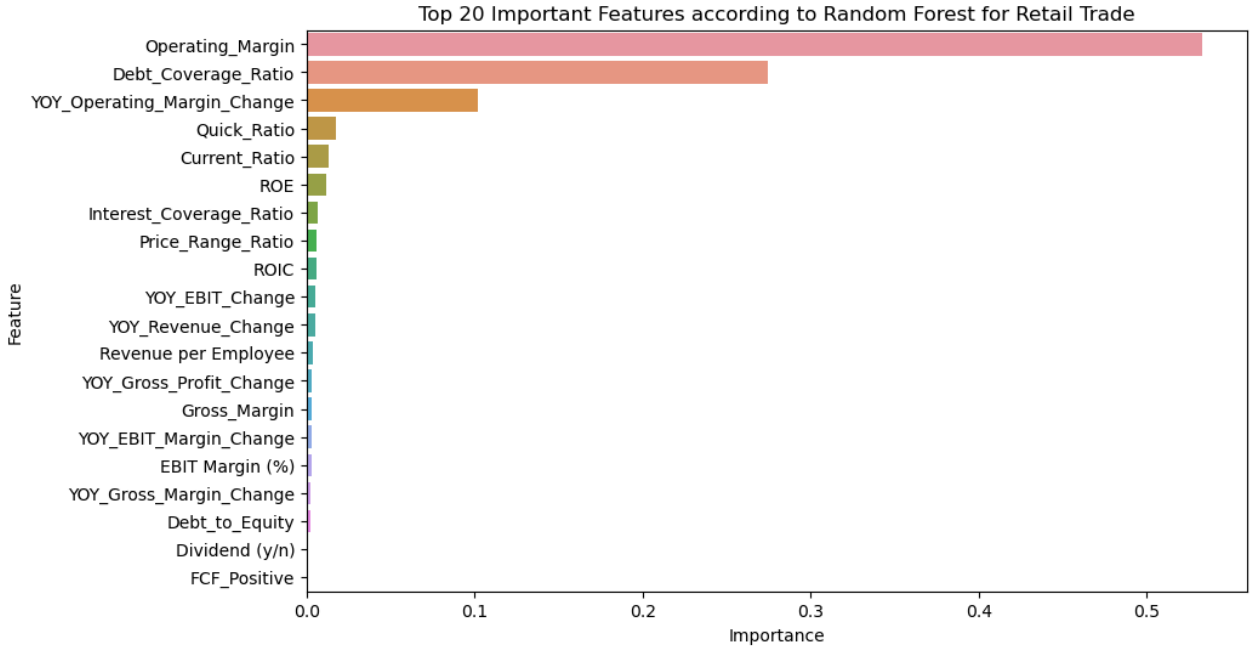
Division: Manufacturing



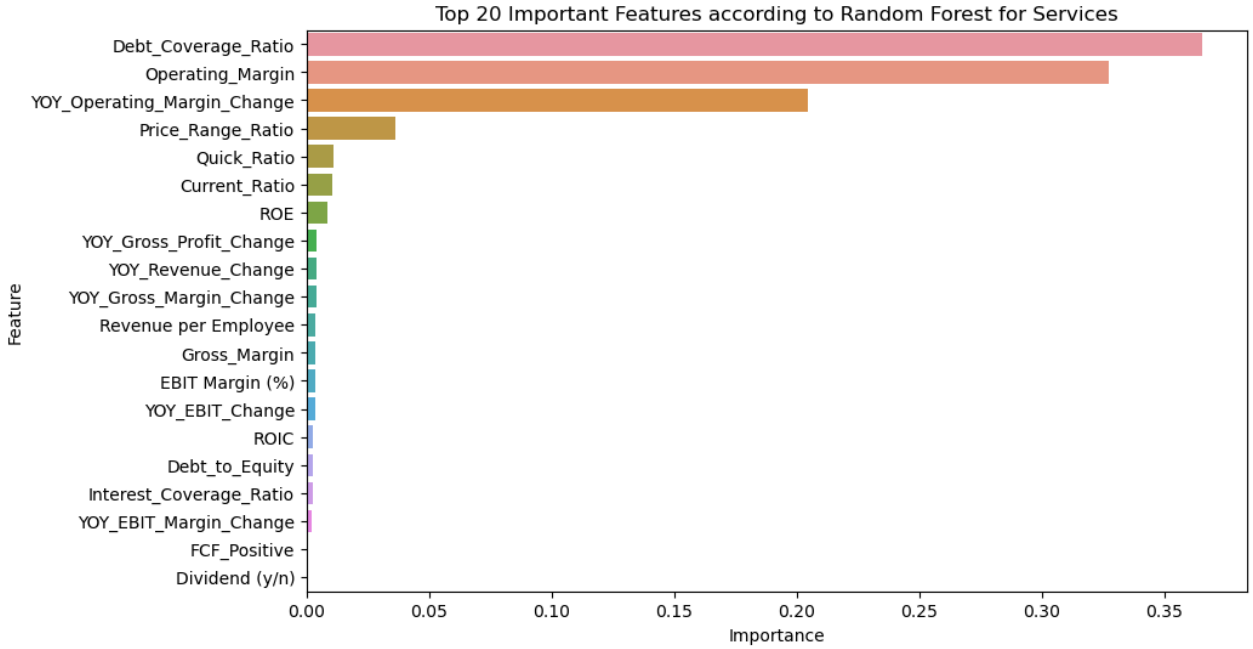
Division: Mining_Trans_Comm_Gas_Sanitary



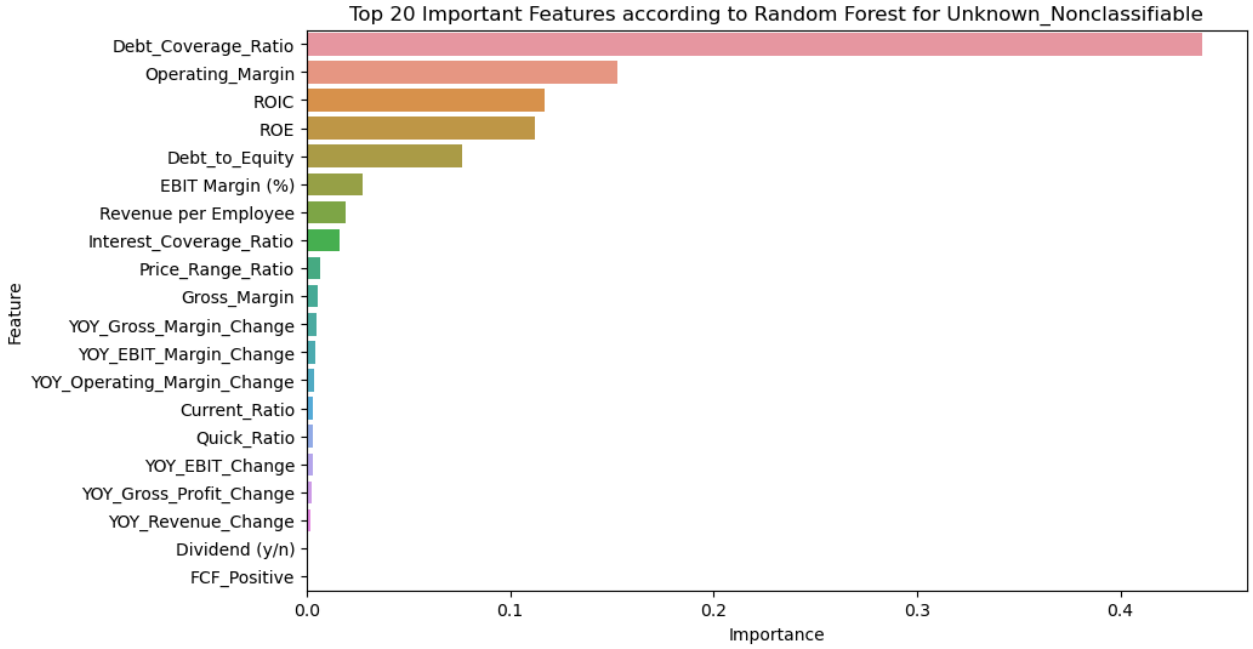
Division: Retail Trade



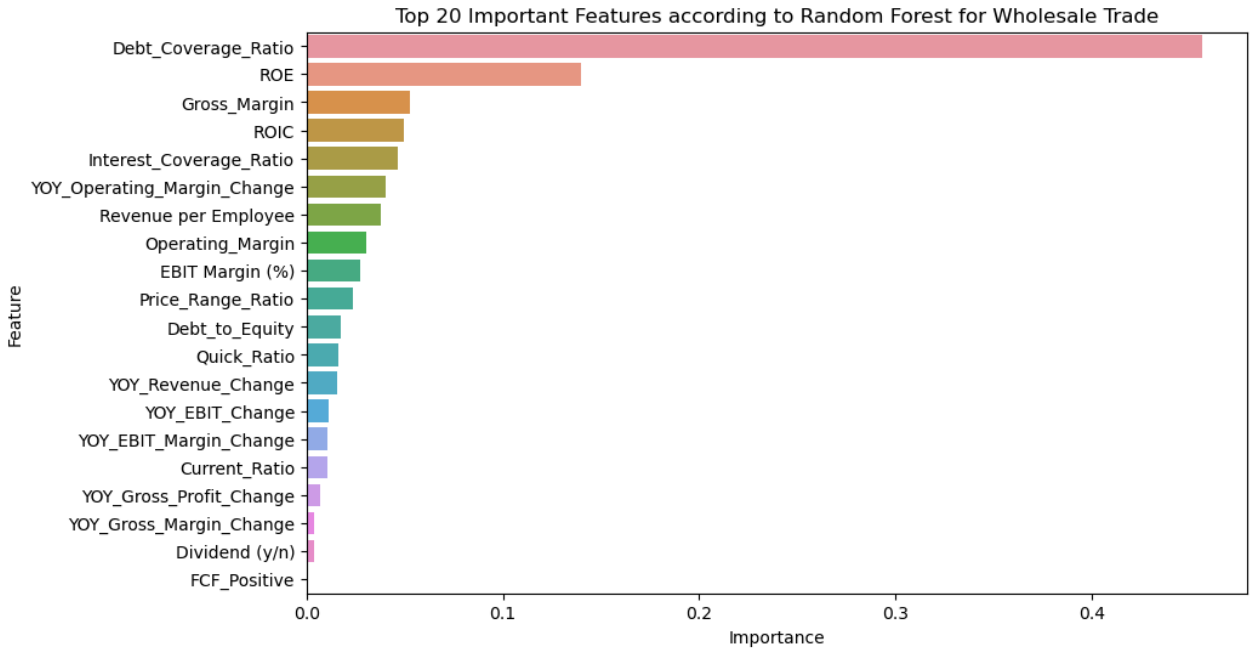
Division: Services



Division: Unknown_Nonclassifiable



Division: Wholesale Trade



In [20]:
```python
from sklearn.feature_selection import RFE

# Recursive Feature Elimination (RFE)
rfe = RFE(estimator=rf, n_features_to_select=10)
rfe.fit(X_scaled, y)

# Features selected by RFE
selected_features = X.columns[rfe.support_]
print("Features selected by RFE:", selected_features)
```

```
Features selected by RFE: Index(['ROE', 'ROIC', 'Debt_Coverage_Ratio', 'Interest_Coverage_Ratio',
       'Gross_Margin', 'Revenue per Employee', 'Operating_Margin',
       'EBIT Margin (%)', 'Price_Range_Ratio', 'YOY_Operating_Margin_Change'],
      dtype='object')
```

In [21]:
```python
from sklearn.feature_selection import RFE
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler

# Group the data by 'Division'
for division, group in selected_df.groupby('Division'):
    print(f"\nDivision: {division}")

    # Prepare the features (X) and target (y)
    X = group.drop(columns=['Ticker', 'Division', 'EV/EBITDA', 'P/E', 'EV/GP', 'EV_EBIT', 'EV/Sal
    y = group['EV/EBITDA']

    # Check if there are enough samples to train the model
    if len(y) < 10:
        print(f"Not enough samples to perform feature selection for {division}")
        continue

    # Scale the features
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    # Initialize the RandomForestRegressor and RFE
    rf = RandomForestRegressor(random_state=42)
    rfe = RFE(estimator=rf, n_features_to_select=10)

    # Fit the RFE model
    rfe.fit(X_scaled, y)
```

```
    # Get the features selected by RFE
    selected_features = X.columns[rfe.support_]
    print("Features selected by RFE:", selected_features.tolist())
```

```
Division: Construction
Features selected by RFE: ['Debt_to_Equity', 'Debt_Coverage_Ratio', 'Current_Ratio', 'Quick_Rati
o', 'Revenue per Employee', 'Operating_Margin', 'YOY_Gross_Margin_Change', 'Price_Range_Ratio',
'YOY_Operating_Margin_Change', 'YOY_EBIT_Change']

Division: Finance, Insurance and Real Estate
Features selected by RFE: ['Debt_to_Equity', 'ROE', 'ROIC', 'Debt_Coverage_Ratio', 'Current_Rati
o', 'Interest_Coverage_Ratio', 'Revenue per Employee', 'Operating_Margin', 'EBIT Margin (%)', 'Pr
ice_Range_Ratio']

Division: Manufacturing
Features selected by RFE: ['Debt_to_Equity', 'ROE', 'Debt_Coverage_Ratio', 'Interest_Coverage_Rat
io', 'Gross_Margin', 'Revenue per Employee', 'Operating_Margin', 'EBIT Margin (%)', 'YOY_Gross_Ma
rgin_Change', 'Price_Range_Ratio']

Division: Mining_Trans_Comm_Gas_Sanitary
Features selected by RFE: ['Debt_to_Equity', 'ROIC', 'Debt_Coverage_Ratio', 'Current_Ratio', 'Int
erest_Coverage_Ratio', 'Gross_Margin', 'Revenue per Employee', 'Operating_Margin', 'EBIT Margin
(%)', 'Price_Range_Ratio']

Division: Retail Trade
Features selected by RFE: ['ROE', 'ROIC', 'Debt_Coverage_Ratio', 'Current_Ratio', 'Quick_Ratio',
'Interest_Coverage_Ratio', 'Gross_Margin', 'Operating_Margin', 'EBIT Margin (%)', 'YOY_Operating_
Margin_Change']

Division: Services
Features selected by RFE: ['Debt_to_Equity', 'ROE', 'Debt_Coverage_Ratio', 'Current_Ratio', 'Quic
k_Ratio', 'Gross_Margin', 'Operating_Margin', 'YOY_Gross_Margin_Change', 'Price_Range_Ratio', 'YO
Y_Operating_Margin_Change']

Division: Unknown_Nonclassifiable
Features selected by RFE: ['Debt_to_Equity', 'ROE', 'ROIC', 'Debt_Coverage_Ratio', 'Interest_Cove
rage_Ratio', 'Revenue per Employee', 'Operating_Margin', 'EBIT Margin (%)', 'Price_Range_Ratio',
'YOY_Operating_Margin_Change']

Division: Wholesale Trade
Features selected by RFE: ['ROE', 'ROIC', 'Debt_Coverage_Ratio', 'Interest_Coverage_Ratio', 'Gros
s_Margin', 'Revenue per Employee', 'Operating_Margin', 'EBIT Margin (%)', 'Price_Range_Ratio', 'Y
OY_Operating_Margin_Change']
```

EV/SALES

In [22]:
```
# Loop through each unique Division and compute feature importance
for division, group in selected_df.groupby('Division'):
    print(f"\nDivision: {division}")

    # Prepare features (X) and target (y)
    X = group.drop(columns=['Ticker', 'Division','EV/EBITDA', 'P/E', 'EV/GP','EV_EBIT','EV/Sales'
    y = group['EV/Sales']

    # Check if there are enough samples to perform a split and modeling
    if len(y) < 10:
        print(f"Not enough samples to perform feature selection for {division}")
        continue

    # Split the dataset into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42)

    # Initialize the ExplainableBoostingRegressor
    ebm = ExplainableBoostingRegressor(random_state=42)
```

```
    # Fit the model
    ebm.fit(X_train, y_train)

    # Get the global feature importances from the model
    ebm_global = ebm.explain_global()
    show(ebm_global)
```

Division: Construction

Division: Finance, Insurance and Real Estate

Division: Manufacturing

Division: Mining_Trans_Comm_Gas_Sanitary

Division: Retail Trade

Division: Services

Division: Unknown_Nonclassifiable

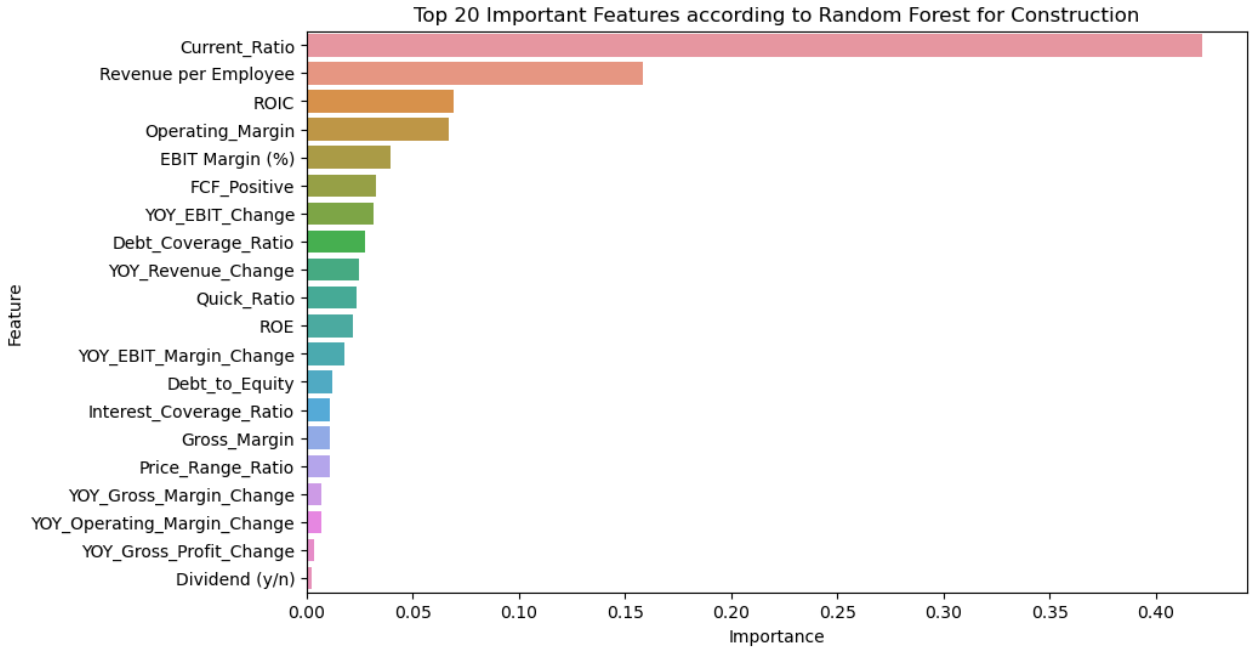
Division: Wholesale Trade

In [23]:

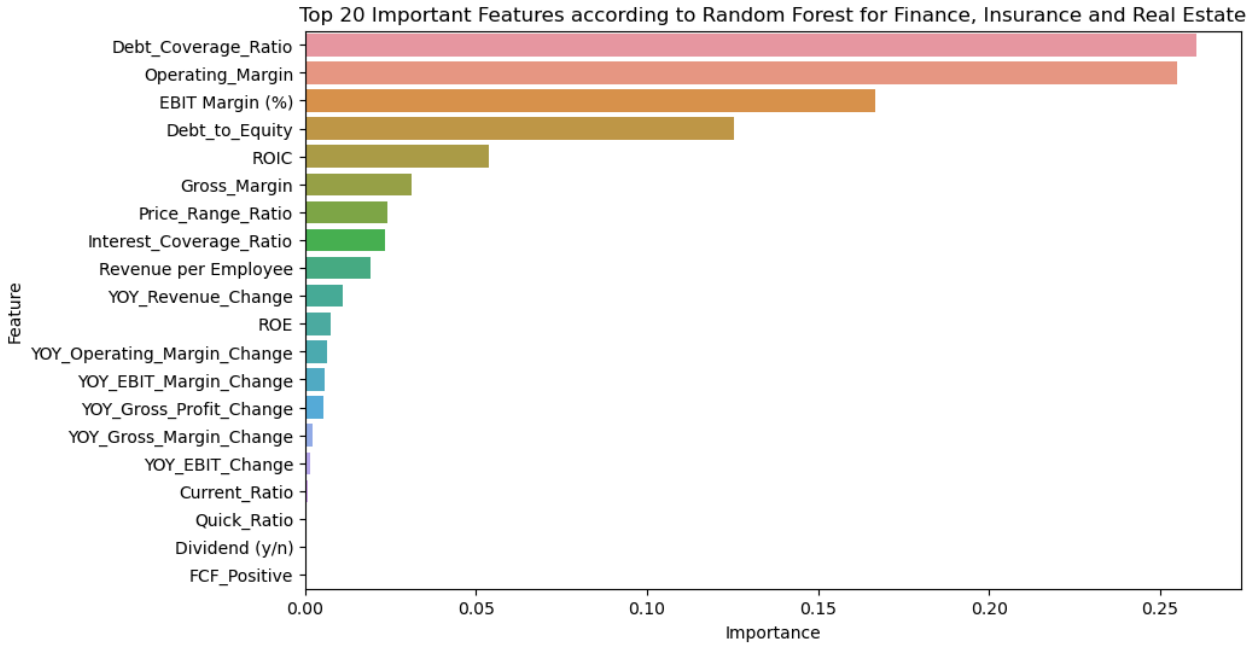
```
# Group the data by 'Division'
for division, group in selected_df.groupby('Division'):
    print(f"\nDivision: {division}")

    # Prepare the features (X) and target (y)
    X = group.drop(columns=['Ticker', 'Division', 'EV/EBITDA', 'P/E', 'EV/GP', 'EV_EBIT','EV/Sale
    y = group['EV/Sales']

    # Check if there are enough samples to train the model
    if len(y) < 10:
        print(f"Not enough samples to perform feature selection for {division}")
        continue

    # Scale the features
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
```

```
    # Use RandomForestRegressor to get feature importances
    rf = RandomForestRegressor(random_state=42)
    rf.fit(X_scaled, y)
    feature_importances = rf.feature_importances_

    # Plot feature importances
    features = pd.DataFrame({'Feature': X.columns, 'Importance': feature_importances})
    features = features.sort_values(by='Importance', ascending=False)

    plt.figure(figsize=(10, 6))
    sns.barplot(x='Importance', y='Feature', data=features.head(20))
    plt.title(f'Top 20 Important Features according to Random Forest for {division}')
    plt.show()
```

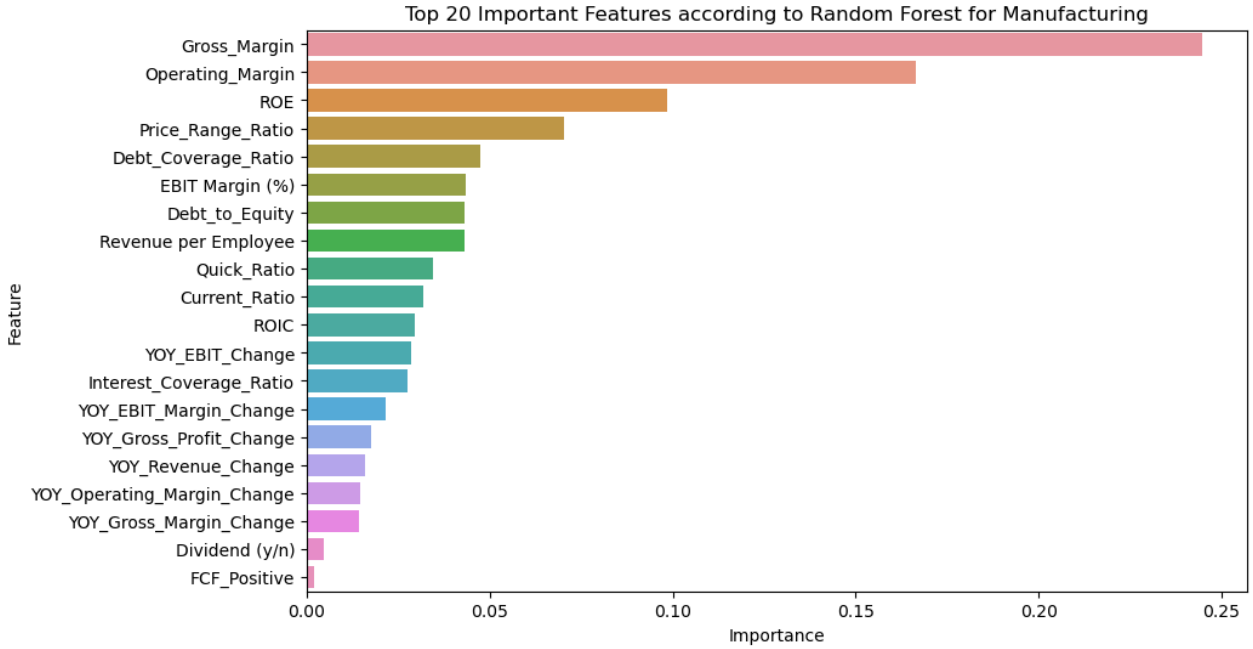
```
Division: Construction
```



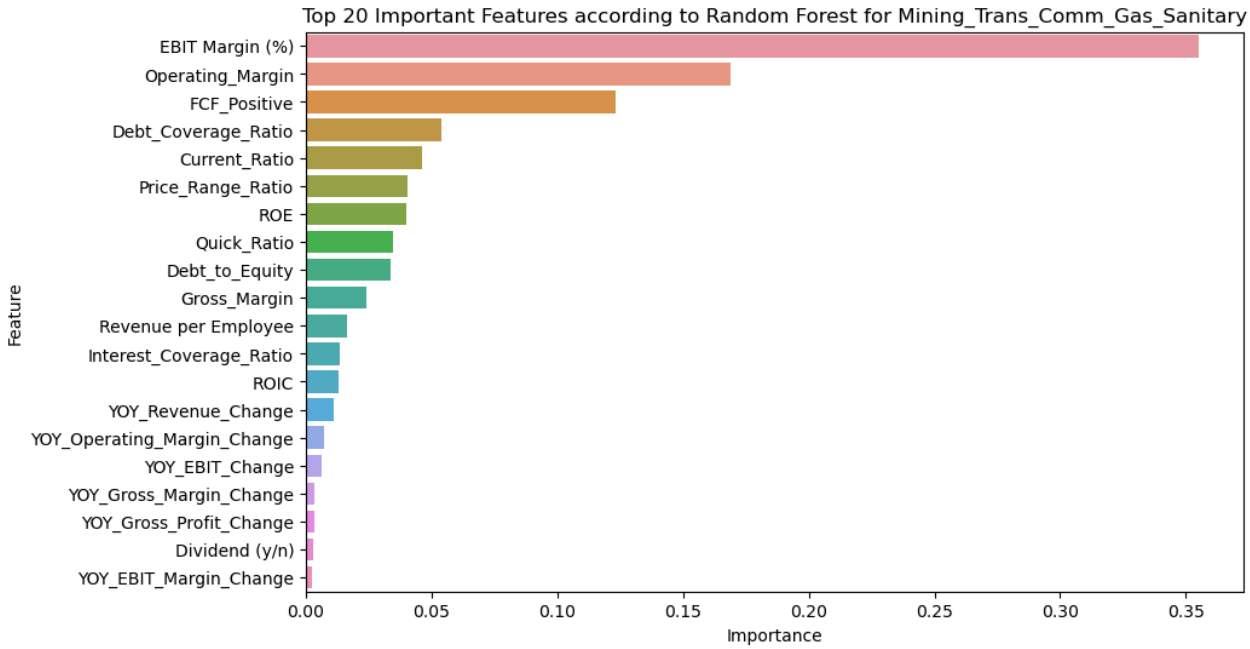

```
Division: Finance, Insurance and Real Estate
```



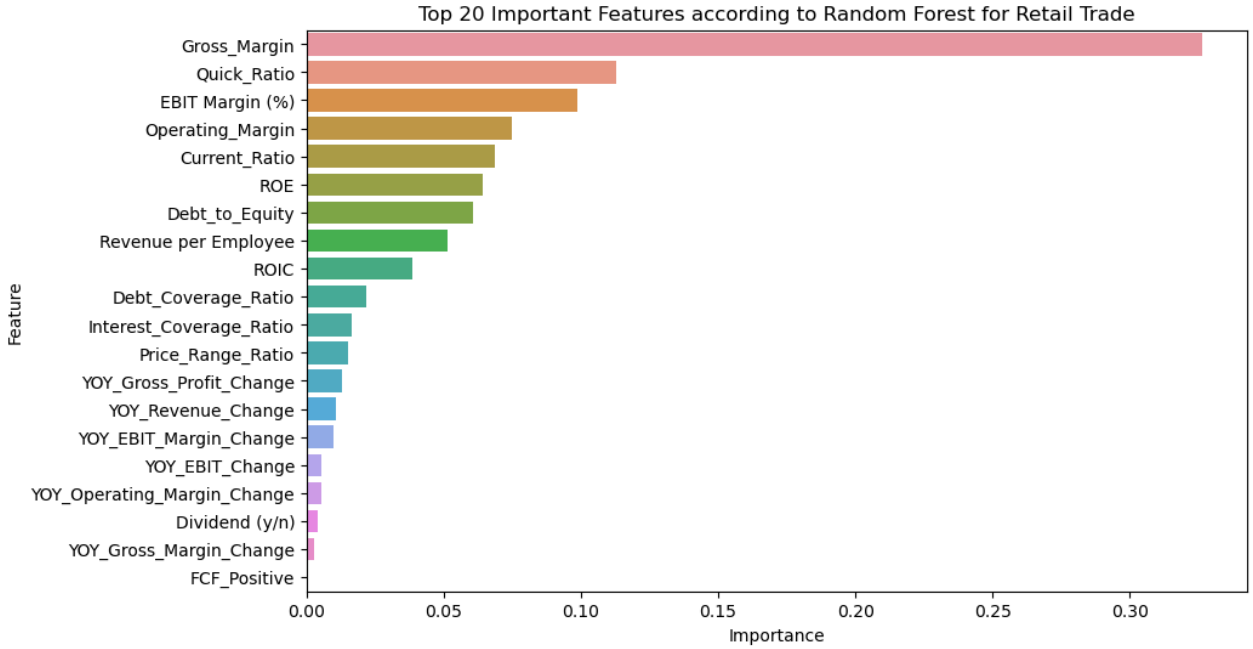

```
Division: Manufacturing
```



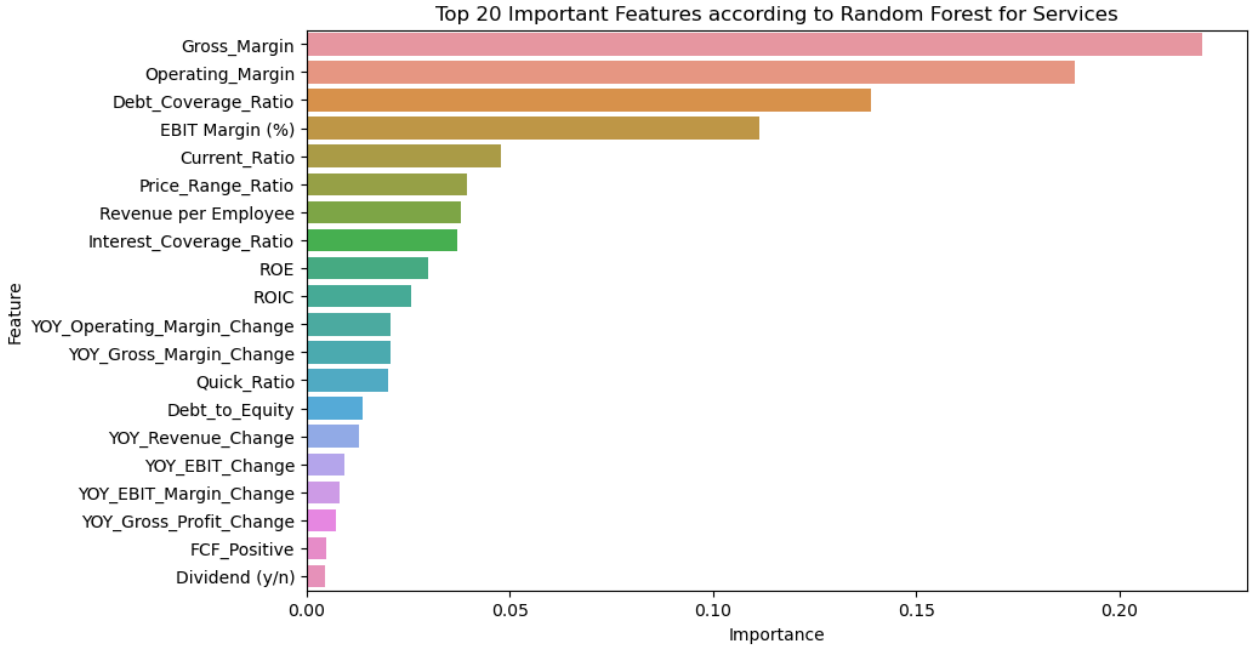
Division: Mining_Trans_Comm_Gas_Sanitary



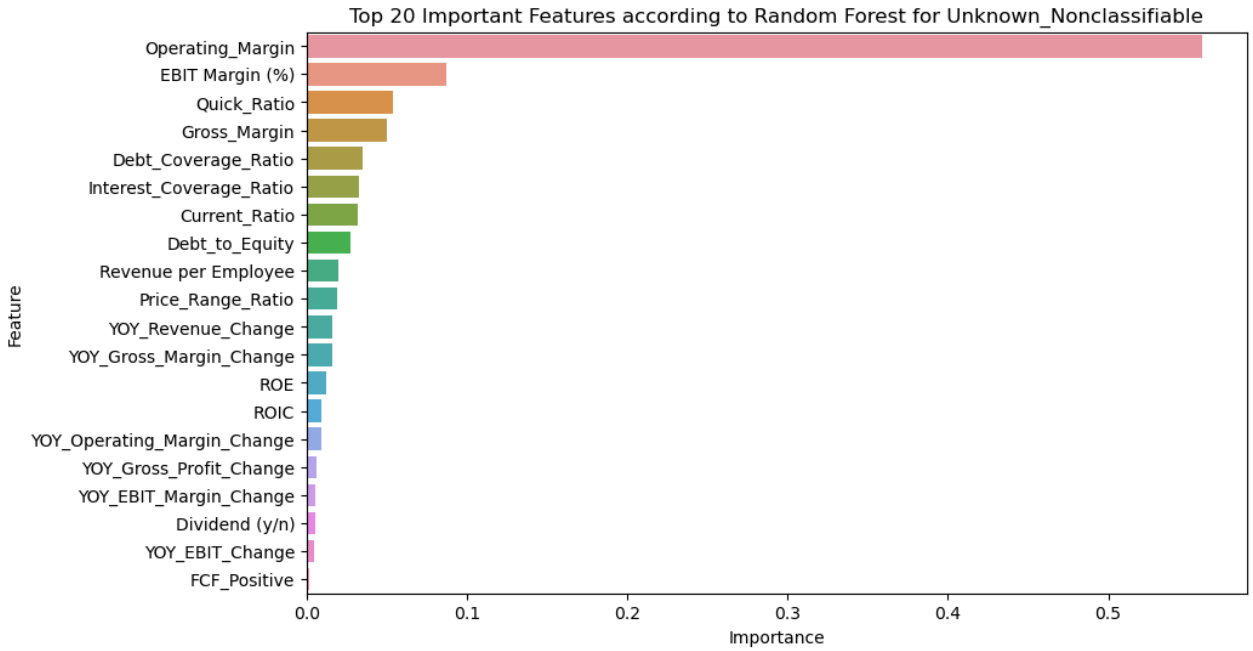
Division: Retail Trade



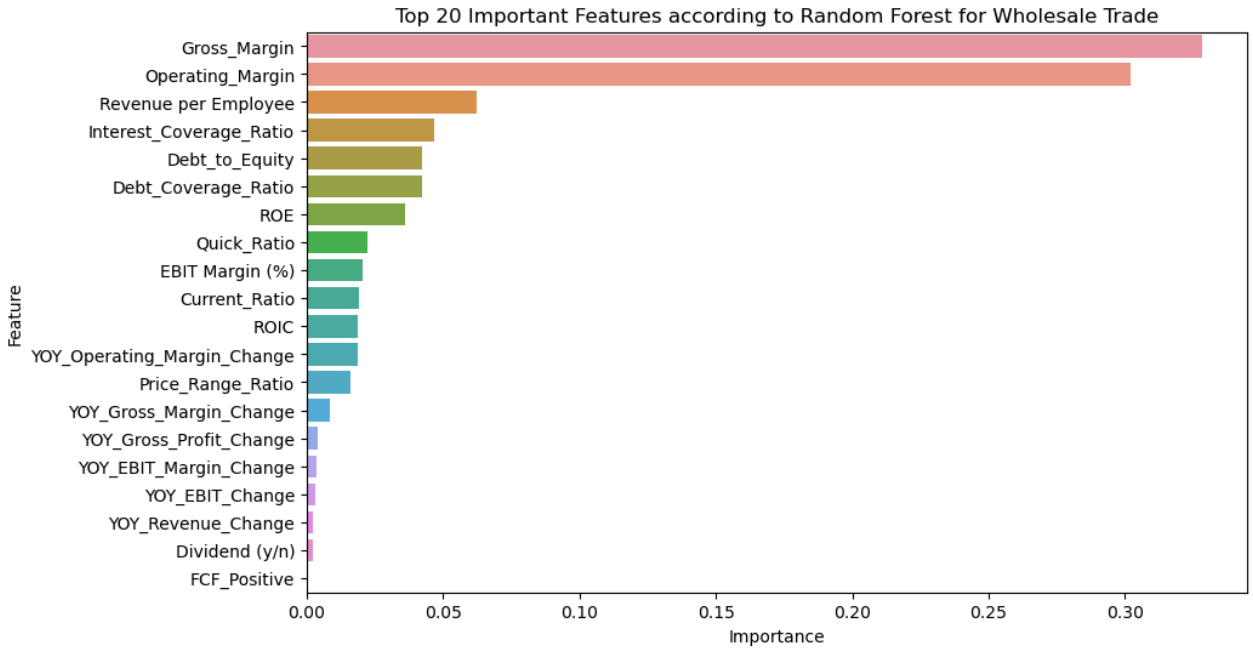
Division: Services



Division: Unknown_Nonclassifiable



Division: Wholesale Trade



In [24]:

```python
from sklearn.feature_selection import RFE
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler

# Group the data by 'Division'
for division, group in selected_df.groupby('Division'):
    print(f"\nDivision: {division}")

    # Prepare the features (X) and target (y)
    X = group.drop(columns=['Ticker', 'Division', 'EV/EBITDA', 'EV/Sales','P/E', 'EV/GP', 'EV_EBI
    y = group['EV/Sales']

    # Check if there are enough samples to train the model
    if len(y) < 10:
        print(f"Not enough samples to perform feature selection for {division}")
        continue

    # Scale the features
    scaler = StandardScaler()
```

```
    X_scaled = scaler.fit_transform(X)

    # Initialize the RandomForestRegressor and RFE
    rf = RandomForestRegressor(random_state=42)
    rfe = RFE(estimator=rf, n_features_to_select=10)

    # Fit the RFE model
    rfe.fit(X_scaled, y)

    # Get the features selected by RFE
    selected_features = X.columns[rfe.support_]
    print("Features selected by RFE:", selected_features.tolist())
```

```
Division: Construction
Features selected by RFE: ['FCF_Positive', 'ROIC', 'Debt_Coverage_Ratio', 'Current_Ratio', 'Quick
_Ratio', 'Revenue per Employee', 'Operating_Margin', 'EBIT Margin (%)', 'YOY_EBIT_Margin_Change',
'YOY_EBIT_Change']

Division: Finance, Insurance and Real Estate
Features selected by RFE: ['Debt_to_Equity', 'ROE', 'ROIC', 'Debt_Coverage_Ratio', 'Interest_Cove
rage_Ratio', 'Gross_Margin', 'Revenue per Employee', 'Operating_Margin', 'EBIT Margin (%)', 'Pric
e_Range_Ratio']

Division: Manufacturing
Features selected by RFE: ['Debt_to_Equity', 'ROE', 'Debt_Coverage_Ratio', 'Quick_Ratio', 'Gross_
Margin', 'Revenue per Employee', 'Operating_Margin', 'EBIT Margin (%)', 'Price_Range_Ratio', 'YOY
_EBIT_Change']

Division: Mining_Trans_Comm_Gas_Sanitary
Features selected by RFE: ['Debt_to_Equity', 'ROE', 'FCF_Positive', 'Debt_Coverage_Ratio', 'Curre
nt_Ratio', 'Quick_Ratio', 'Gross_Margin', 'Operating_Margin', 'EBIT Margin (%)', 'Price_Range_Rat
io']

Division: Retail Trade
Features selected by RFE: ['Debt_to_Equity', 'ROE', 'ROIC', 'Debt_Coverage_Ratio', 'Current_Rati
o', 'Quick_Ratio', 'Gross_Margin', 'Revenue per Employee', 'Operating_Margin', 'EBIT Margin (%)']

Division: Services
Features selected by RFE: ['ROE', 'ROIC', 'Debt_Coverage_Ratio', 'Current_Ratio', 'Interest_Cover
age_Ratio', 'Gross_Margin', 'Revenue per Employee', 'Operating_Margin', 'EBIT Margin (%)', 'Price
_Range_Ratio']

Division: Unknown_Nonclassifiable
Features selected by RFE: ['Debt_to_Equity', 'Debt_Coverage_Ratio', 'Current_Ratio', 'Quick_Rati
o', 'Interest_Coverage_Ratio', 'Gross_Margin', 'Revenue per Employee', 'Operating_Margin', 'EBIT
Margin (%)', 'YOY_Gross_Margin_Change']

Division: Wholesale Trade
Features selected by RFE: ['Debt_to_Equity', 'ROE', 'ROIC', 'Debt_Coverage_Ratio', 'Quick_Ratio',
'Interest_Coverage_Ratio', 'Gross_Margin', 'Revenue per Employee', 'Operating_Margin', 'YOY_Opera
ting_Margin_Change']
```