In [90]:
```python
# Using pandas for loading the csv file into a dataframe
import pandas as pd

# Using matplotlib library for basic graph plotting
import matplotlib.pyplot as plt

# Using seaborn library that allows us to optimize matplotlib's output
import seaborn as sns

# Using numpy library
import numpy as np
```

In [91]:
```python
# Loading the csv file into the pandas dataframe

stocks = pd.read_csv('2018_Financial_Data.csv')
stocks
```

Out[91]:

	Unnamed: 0	Revenue	Revenue Growth	Cost of Revenue	Gross Profit	R&D Expenses	SG&A Expense	Operating Expenses	Operating Income	Interest Expense	...	Receivables growth	Inventory Growth	Asset Growth	Book Value per Share Growth	Debt Growth	R&D Expense Growth	SG&A Expenses Growth	Secto
0	CMCSA	9.450700e+10	0.1115	0.000000e+00	9.450700e+10	0.000000e+00	6.482200e+10	7.549800e+10	1.900900e+10	3.542000e+09	...	0.2570	0.0000	0.3426	0.0722	0.7309	0.0000	0.1308	Consume Cyclica
1	KMI	1.414400e+10	0.0320	7.288000e+09	6.856000e+09	0.000000e+00	6.010000e+08	3.062000e+09	3.794000e+09	1.917000e+09	...	0.0345	-0.0920	-0.0024	0.0076	-0.0137	0.0000	-0.1265	Energ
2	INTC	7.084800e+10	0.1289	2.711100e+10	4.373700e+10	1.354300e+10	6.750000e+09	2.042100e+10	2.331600e+10	-1.260000e+08	...	0.1989	0.0387	0.0382	0.1014	-0.0169	0.0390	-0.0942	Technolog
3	MU	3.039100e+10	0.4955	1.250000e+10	1.789100e+10	2.141000e+09	8.130000e+08	2.897000e+09	1.499400e+10	3.420000e+08	...	0.4573	0.1511	0.2275	0.6395	-0.5841	0.1738	0.0942	Technolog
4	GE	1.216150e+11	0.0285	9.546100e+10	2.615400e+10	0.000000e+00	1.811100e+10	4.071100e+10	-1.455700e+10	5.059000e+09	...	-0.2781	-0.2892	-0.1575	-0.4487	-0.2297	0.0000	0.0308	Industrial
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
4387	YRIV	0.000000e+00	0.0000	0.000000e+00	0.000000e+00	0.000000e+00	3.755251e+06	3.755251e+06	-3.755251e+06	1.105849e+07	...	0.0000	0.0000	-0.0508	-0.1409	-0.0152	0.0000	-0.2602	Rea Estat
4388	YTEN	5.560000e+05	-0.4110	0.000000e+00	5.560000e+05	4.759000e+06	5.071000e+06	9.830000e+06	-9.274000e+06	0.000000e+00	...	0.3445	0.0000	-0.2323	-0.8602	0.0000	0.0352	-0.0993	Basi Material
4389	ZKIN	5.488438e+07	0.2210	3.659379e+07	1.829059e+07	1.652633e+06	7.020320e+06	8.672953e+06	9.617636e+06	1.239170e+06	...	0.1605	0.7706	0.2489	0.4074	-0.0968	0.2415	0.8987	Basi Material
4390	ZOM	0.000000e+00	0.0000	0.000000e+00	0.000000e+00	1.031715e+07	4.521349e+06	1.664863e+07	-1.664863e+07	0.000000e+00	...	0.8980	0.0000	0.1568	-0.2200	0.0000	2.7499	0.1457	Industrial
4391	ZYME	5.301900e+07	0.0243	0.000000e+00	5.301900e+07	5.668400e+07	2.945700e+07	8.614600e+07	-3.312700e+07	1.660000e+05	...	-0.4185	0.0000	0.8519	0.1325	0.0000	0.3577	0.5880	Healthcar

4392 rows × 225 columns

In [92]:
```python
# Checking if tnere are null values

null_values = stocks.isnull().sum()
null_values
```

Out[92]:
```
Unnamed: 0               0
Revenue                 46
Revenue Growth         139
Cost of Revenue        185
Gross Profit            64
                      ...
R&D Expense Growth     259
SG&A Expenses Growth   248
Sector                   0
2019 PRICE VAR [%]       0
Class                    0
Length: 225, dtype: int64
```

In [93]:
```python
# Dropping columns with more than 200 null values (just a soft cleansing)

columns_to_drop = null_values[null_values > 200].index
stocks_dropped = stocks.drop(columns=columns_to_drop)
stocks_dropped
```

Out[93]:

	Unnamed: 0	Revenue	Revenue Growth	Cost of Revenue	Gross Profit	SG&A Expense	Operating Expenses	Operating Income	Interest Expense	Earnings before Tax	...	Gross Profit Growth	Operating Income Growth	EPS Growth	EPS Diluted Growth	Weighted Average Shares Diluted Growth	Operating Cash Flow growth	Receivables growth	Sec
0	CMCSA	9.450700e+10	0.1115	0.000000e+00	9.450700e+10	6.482200e+10	7.549800e+10	1.900900e+10	3.542000e+09	1.511100e+10	...	0.1115	0.0550	-0.4700	-0.4674	-0.0305	0.1428	0.2570	Consu Cyc
1	KMI	1.414400e+10	0.0320	7.288000e+09	6.856000e+09	6.010000e+08	3.062000e+09	3.794000e+09	1.917000e+09	2.196000e+09	...	0.0564	0.0751	65.0000	65.0000	-0.0063	0.0961	0.0345	Ene
2	INTC	7.084800e+10	0.1289	2.711100e+10	4.373700e+10	6.750000e+09	2.042100e+10	2.331600e+10	-1.260000e+08	2.331700e+10	...	0.1187	0.2917	1.2402	1.2513	-0.0277	0.3312	0.1989	Technol
3	MU	3.039100e+10	0.4955	1.250000e+10	1.789100e+10	8.130000e+08	2.897000e+09	1.499400e+10	3.420000e+08	1.430300e+10	...	1.1208	1.5552	1.6274	1.6100	0.0650	1.1342	0.4573	Technol
4	GE	1.216150e+11	0.0285	9.546100e+10	2.615400e+10	1.811100e+10	4.071100e+10	-1.455700e+10	5.059000e+09	-2.177200e+10	...	0.7216	-1.4225	-1.5437	-1.5437	0.0005	-0.2961	-0.2781	Industr
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4387	YRIV	0.000000e+00	0.0000	0.000000e+00	0.000000e+00	3.755251e+06	3.755251e+06	-3.755251e+06	1.105849e+07	-1.482451e+07	...	0.0000	0.2602	-0.3333	-0.3333	NaN	-2.0649	0.0000	F Es
4388	YTEN	5.560000e+05	-0.4110	0.000000e+00	5.560000e+05	5.071000e+06	9.830000e+06	-9.274000e+06	0.000000e+00	-9.170000e+06	...	-0.4110	0.0010	0.7204	0.7204	2.0203	-0.0673	0.3445	Ba Mater
4389	ZKIN	5.488438e+07	0.2210	3.659379e+07	1.829059e+07	7.020320e+06	8.672953e+06	9.617636e+06	1.239170e+06	8.416324e+06	...	0.3953	0.1903	-0.0714	-0.0714	0.2420	-2.9498	0.1605	Ba Mater
4390	ZOM	0.000000e+00	0.0000	0.000000e+00	0.000000e+00	4.521349e+06	1.664863e+07	-1.664863e+07	0.000000e+00	-1.664769e+07	...	0.0000	-1.0594	-1.0000	-1.0000	0.0691	-0.5716	0.8980	Industr
4391	ZYME	5.301900e+07	0.0243	0.000000e+00	5.301900e+07	2.945700e+07	8.614600e+07	-3.312700e+07	1.660000e+05	-3.438500e+07	...	0.0243	-2.6816	-1.4706	-0.9688	0.3644	109.3973	-0.4185	Healthc

4392 rows × 68 columns

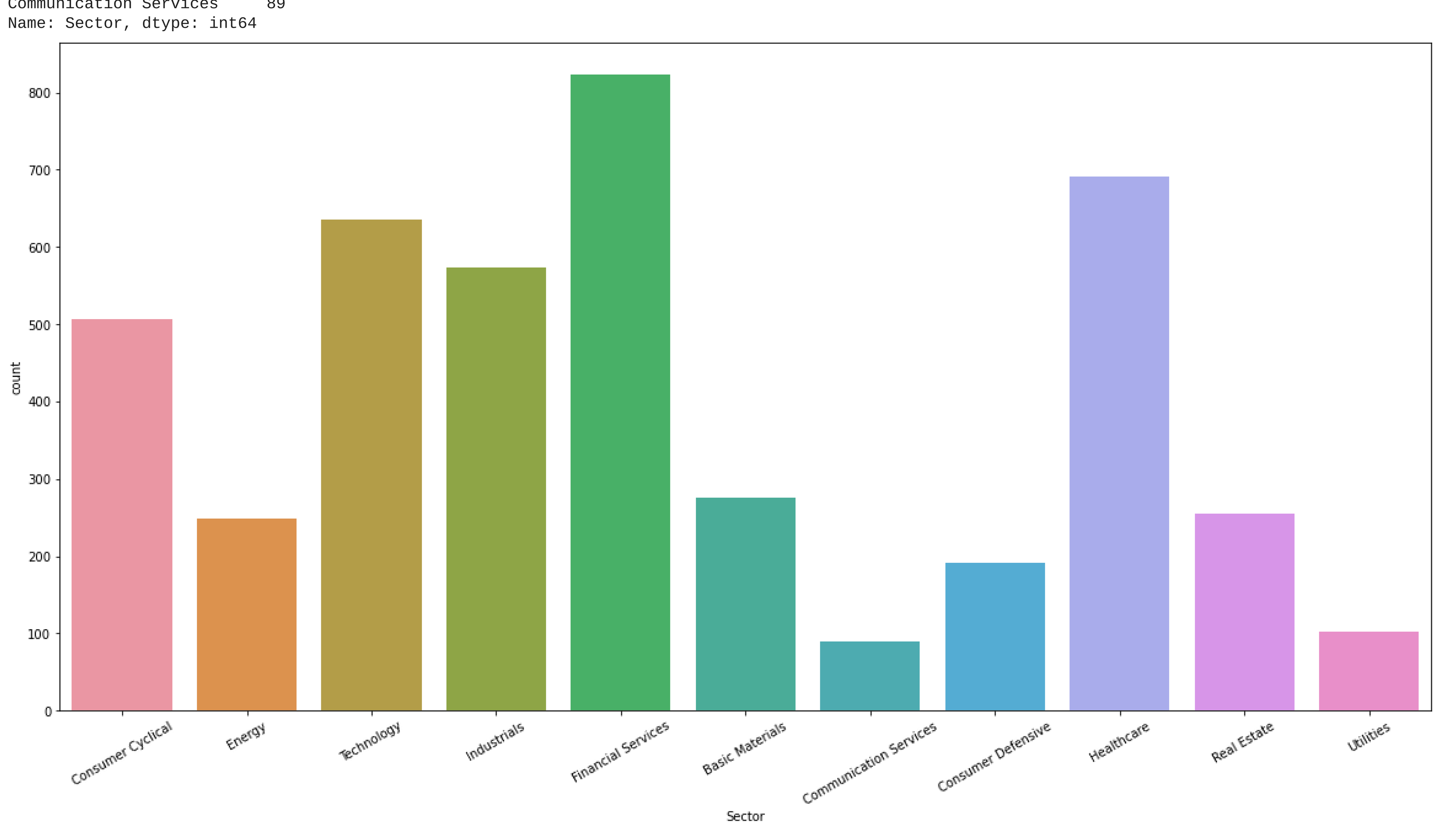
In [94]:
```python
# Visualization to see the stocks per sector

plt.figure(figsize = (20,10))
plot = sns.countplot(stocks_dropped['Sector'])

plot.set_xticklabels(plot.get_xticklabels(), rotation=30)
stocks_dropped.Sector.value_counts()
```

```
C:\Users\hagay\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[94]:
```
Financial Services        824
Healthcare                691
Technology                636
Industrials               574
Consumer Cyclical         506
Basic Materials           276
Real Estate               255
Energy                    248
Consumer Defensive        191
Utilities                 102
Communication Services     89
Name: Sector, dtype: int64
```



In [95]:
```python
# Renaming some columns

stocks_dropped.rename(columns={'Unnamed: 0': 'stock_symbol', 'Class': 'buy_worthy?'}, inplace=True)
stocks_dropped
```

Out[95]:

	stock_symbol	Revenue	Revenue Growth	Cost of Revenue	Gross Profit	SG&A Expense	Operating Expenses	Operating Income	Interest Expense	Earnings before Tax	...	Gross Profit Growth	Operating Income Growth	EPS Growth	EPS Diluted Growth	Weighted Average Shares Diluted Growth	Operating Cash Flow growth	Receivables growth	
0	CMCSA	9.450700e+10	0.1115	0.000000e+00	9.450700e+10	6.482200e+10	7.549800e+10	1.900900e+10	3.542000e+09	1.511100e+10	...	0.1115	0.0550	-0.4700	-0.4674	-0.0305	0.1428	0.2570	Co
1	KMI	1.414400e+10	0.0320	7.288000e+09	6.856000e+09	6.010000e+08	3.062000e+09	3.794000e+09	1.917000e+09	2.196000e+09	...	0.0564	0.0751	65.0000	65.0000	-0.0063	0.0961	0.0345	
2	INTC	7.084800e+10	0.1289	2.711100e+10	4.373700e+10	6.750000e+09	2.042100e+10	2.331600e+10	-1.260000e+08	2.331700e+10	...	0.1187	0.2917	1.2402	1.2513	-0.0277	0.3312	0.1989	Tec
3	MU	3.039100e+10	0.4955	1.250000e+10	1.789100e+10	8.130000e+08	2.897000e+09	1.499400e+10	3.420000e+08	1.430300e+10	...	1.1208	1.5552	1.6274	1.6100	0.0650	1.1342	0.4573	Tec
4	GE	1.216150e+11	0.0285	9.546100e+10	2.615400e+10	1.811100e+10	4.071100e+10	-1.455700e+10	5.059000e+09	-2.177200e+10	...	0.7216	-1.4225	-1.5437	-1.5437	0.0005	-0.2961	-0.2781	Inc
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4387	YRIV	0.000000e+00	0.0000	0.000000e+00	0.000000e+00	3.755251e+06	3.755251e+06	-3.755251e+06	1.105849e+07	-1.482451e+07	...	0.0000	0.2602	-0.3333	-0.3333	NaN	-2.0649	0.0000	
4388	YTEN	5.560000e+05	-0.4110	0.000000e+00	5.560000e+05	5.071000e+06	9.830000e+06	-9.274000e+06	0.000000e+00	-9.170000e+06	...	-0.4110	0.0010	0.7204	0.7204	2.0203	-0.0673	0.3445	M
4389	ZKIN	5.488438e+07	0.2210	3.659379e+07	1.829059e+07	7.020320e+06	8.672953e+06	9.617636e+06	1.239170e+06	8.416324e+06	...	0.3953	0.1903	-0.0714	-0.0714	0.2420	-2.9498	0.1605	M
4390	ZOM	0.000000e+00	0.0000	0.000000e+00	0.000000e+00	4.521349e+06	1.664863e+07	-1.664863e+07	0.000000e+00	-1.664769e+07	...	0.0000	-1.0594	-1.0000	-1.0000	0.0691	-0.5716	0.8980	Inc
4391	ZYME	5.301900e+07	0.0243	0.000000e+00	5.301900e+07	2.945700e+07	8.614600e+07	-3.312700e+07	1.660000e+05	-3.438500e+07	...	0.0243	-2.6816	-1.4706	-0.9688	0.3644	109.3973	-0.4185	Hea

4392 rows × 68 columns

In [96]:
```python
# Getting a list of all the columns

stocks_dropped.columns.tolist()
```

Out[96]:
```
['stock_symbol',
 'Revenue',
 'Revenue Growth',
 'Cost of Revenue',
 'Gross Profit',
 'SG&A Expense',
 'Operating Expenses',
 'Operating Income',
 'Interest Expense',
 'Earnings before Tax',
 'Income Tax Expense',
 'Net Income',
 'Net Income Com',
 'EPS',
 'EPS Diluted',
 'Weighted Average Shs Out',
 'Weighted Average Shs Out (Dil)',
 'Gross Margin',
 'EBIT Margin',
 'EBITDA',
 'EBIT',
 'Consolidated Income',
 'Earnings Before Tax Margin',
 'Net Profit Margin',
 'Cash and cash equivalents',
 'Short-term investments',
 'Cash and short-term investments',
 'Receivables',
 'Inventories',
 'Total current assets',
 'Property, Plant & Equipment Net',
 'Goodwill and Intangible Assets',
 'Long-term investments',
 'Total assets',
 'Payables',
 'Total current liabilities',
 'Long-term debt',
 'Total debt',
 'Total liabilities',
 'Other comprehensive income',
 'Retained earnings (deficit)',
 'Total shareholders equity',
 'Other Liabilities',
 'Depreciation & Amortization',
 'Stock-based compensation',
 'Operating Cash Flow',
 'Capital Expenditure',
 'Investment purchases and sales',
 'Investing Cash flow',
 'Issuance (repayment) of debt',
 'Financing Cash Flow',
 'Effect of forex changes on cash',
 'Net cash flow / Change in cash',
 'Free Cash Flow',
 'daysOfInventoryOutstanding',
 'Average Receivables',
 'Days Sales Outstanding',
 'Receivables Turnover',
 'Gross Profit Growth',
 'Operating Income Growth',
 'EPS Growth',
 'EPS Diluted Growth',
 'Weighted Average Shares Diluted Growth',
 'Operating Cash Flow growth',
 'Receivables growth',
 'Sector',
 '2019 PRICE VAR [%]',
 'buy_worthy?']
```

In [97]:
```python
# Getting a list of the first 40 columns and how many nulls each

column_null_counts = stocks_dropped.isnull().sum()
column_null_counts.head(40)
```

Out[97]:
```
stock_symbol                         0
Revenue                             46
Revenue Growth                     139
Cost of Revenue                    185
Gross Profit                        64
SG&A Expense                       166
Operating Expenses                 184
Operating Income                    35
Interest Expense                   184
Earnings before Tax                 71
Income Tax Expense                 140
Net Income                         151
Net Income Com                     117
EPS                                 65
EPS Diluted                         63
Weighted Average Shs Out            66
Weighted Average Shs Out (Dil)     200
Gross Margin                        51
EBIT Margin                        174
EBITDA                             183
EBIT                               143
Consolidated Income                151
Earnings Before Tax Margin          51
Net Profit Margin                  176
Cash and cash equivalents           49
Short-term investments             179
Cash and short-term investments    119
Receivables                         38
Inventories                        197
Total current assets               185
Property, Plant & Equipment Net    169
Goodwill and Intangible Assets     142
Long-term investments              181
Total assets                       108
Payables                           174
Total current liabilities          184
Long-term debt                      30
Total debt                         168
Total liabilities                  127
Other comprehensive income         158
dtype: int64
```

In [98]:
```python
# Handling missing values

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

In [99]:
```python
# Splitting the dataset into features and targets

y = stocks_dropped[['buy_worthy?']]
X = stocks_dropped[['Total debt', 'Total assets']]
```

In [100…
```python
# Splitting the dataset into training and test sets in the ratio 70/30

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 10)
```

In [101…
```python
# Creating a Logistic Regression Model

from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()
```

In [102…
```python
# Training the Linear Regression model using the training data

lr.fit(X_train, y_train)
```

```
C:\Users\hagay\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
```

Out[102…
```
LogisticRegression()
```

In [103…
```python
# Make predictions using your test data

y_pred = lr.predict(X_test)
```

In [104…
```python
# Generating the confusion matrix using scikit-learn's confusion matrix method

from sklearn.metrics import confusion_matrix

confusion_matrix(y_test, y_pred)
```

Out[104…
```
array([[  0, 395],
       [  0, 923]], dtype=int64)
```

In [105…
```python
# Generate Classification Report

from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       395
           1       0.70      1.00      0.82       923

    accuracy                           0.70      1318
   macro avg       0.35      0.50      0.41      1318
weighted avg       0.49      0.70      0.58      1318
```

```
C:\Users\hagay\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels wit
h no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\hagay\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels wit
h no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\hagay\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels wit
h no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```