Financial analysis of S&P500 stocks

Tanmay Nandanikar

Contents

1. Introduction

The Standard and Poor's 500, or simply the S&P 500, is a stock market index tracking the stock performance of 500 large companies listed on exchanges in the United States. It is one of the most commonly followed equity indices. As of December 31, 2020, more than $5.4 trillion was invested in assets tied to the performance of the index.

In this project, we will try to understand the dataset, analyse it, and perform data visualizations using various Python libraries to find patterns and insights within the dataset.

2. Data Overview and Pre-processing

In this section, we'll be working to understand our dataset and later cleaning it before processing it further.

IMPORTING LIBRARIES AND MODULES

```
# linear algebra
import numpy as np

# data processing, CSV file I/O (e.g. pd.read_csv)
import pandas as pd

# data visualization
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go

# stocks related missing info
import yfinance as yf

# ignoring the warnings
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

ABOUT THE DATASET

```
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

SP500_Comp = pd.read_csv('sp500_companies.csv')
print(SP500_Comp.head(5))
```

Printing the first 5 rows of the dataset

```
  Exchange Symbol                   Shortname                Longname  \
0      NMS   AAPL                  Apple Inc.              Apple Inc.
1      NMS   MSFT       Microsoft Corporation   Microsoft Corporation
2      NMS  GOOGL              Alphabet Inc.           Alphabet Inc.
3      NMS   TSLA                Tesla, Inc.             Tesla, Inc.
4      NYQ  BRK-B  Berkshire Hathaway Inc. New  Berkshire Hathaway Inc.

                 Sector                     Industry  Currentprice  \
0            Technology         Consumer Electronics        167.57
```

```
1             Technology         Software—Infrastructure        277.75
2  Communication Services  Internet Content & Information        114.24
3       Consumer Cyclical             Auto Manufacturers        869.74
4      Financial Services          Insurance—Diversified        288.69

       Marketcap        Ebitda  Revenuegrowth           City State  \
0  2749220519936  1.295570e+11          0.019      Cupertino    CA
1  2085341364224  9.798300e+10          0.124        Redmond    WA
2  1495607148544  9.688700e+10          0.126  Mountain View    CA
3   873445064704  1.403000e+10          0.416         Austin    TX
4   646454837248  1.177540e+11          0.096          Omaha    NE

         Country  Fulltimeemployees  \
0  United States           154000.0
1  United States           221000.0
2  United States           174014.0
3  United States            99290.0
4  United States           372000.0

                              Longbusinesssummary    Weight
0  Apple Inc. designs, manufactures, and markets ...  0.078980
1  Microsoft Corporation develops, licenses, and ...  0.059908
2  Alphabet Inc. provides various products and pl...  0.042966
3  Tesla, Inc. designs, develops, manufactures, l...  0.025093
4  Berkshire Hathaway Inc., through its subsidiar...  0.018572
```

Output for the above code

Column no	Features	Description
0	Exchange	An open, organized marketplace where stocks, bonds, commodities, options and futures are traded
1	Symbol	"Ticker". Unique code given to a company listed on the exchange
2	Shortname	Company's short name
3	Longname	Company's long name
4	Sector	Sector of the Company
5	Industry	Industry of the Company
6	Currentprice	Most recent selling price of a stock
7	Marketcap	Market value of the company's outstanding shares. Calculated using $Current\ Price \times Outstanding\ shares$
8	Ebitda	It's a profitability calculation that measures how profitable a company is before paying interest to creditors, taxes to the government, and taking paper expenses like depreciation and amortization. It is calculated as $EBITDA = Net\ Income + Interest + Taxes + Depreciation + Amortization$
9	Revenuegrowth	Increase (or decrease) in a company's sales from one period to the next. It is calculated as $\frac{Current\ Period\ Sales - Prior\ Period\ Sales}{Prior\ Period\ Sales}$
10	City	City of the Company's HQ
11	State	State of the Company's HQ
12	Country	Company's country of origin
13	Fulltimeemployees	Total full time employees in the company
14	Longbusinesssummary	Summary about the company's business
15	Weight	S&P 500 uses marketcap weighing method, where weight of each stock is calculated as $\frac{Company\ marker\ cap}{Total\ of\ all\ market\ cap}$

Definitions for each column

```
print(SP500_Comp.shape)
print(SP500_Comp.count(axis=0))
```

To view shape and number of null values in each column

```
(495, 16)
Exchange               495
Symbol                 495
Shortname              495
Longname               495
Sector                 493
Industry               493
Currentprice           493
Marketcap              495
Ebitda                 463
Revenuegrowth          492
City                   493
State                  474
Country                493
Fulltimeemployees      487
Longbusinesssummary    493
Weight                 495
dtype: int64
```

Number of non-null values in respective columns

DEALING WITH MISSING DATA

I noticed that 2 companies, namely "Honeywell International Inc." and "Host Hotels & Resorts, Inc." have missing data for nearly all fields. Hence, these companies will be dropped.

Missing Revenue Growth

```
def replace_null(df, sym, col, missing):
    ticker = yf.Ticker(sym)
    df.loc[df['Symbol']==sym, col]= ticker.info[missing]

replace_null(SP500_Comp,'ROP', 'Revenuegrowth', 'revenueGrowth')
```

yFinance library was used to get the correct Revenue Growth numbers

Missing States

Since the "State" column isn't used, we can safely drop the entire column as below.

```
SP500_Comp = SP500_Comp.drop(['State'], axis=1)
```

Dropping the "State" column

Missing Fulltimeemployees

```
SP500_Comp.loc[SP500_Comp['Fulltimeemployees'].isnull(), 'Fulltimeemployees'] =
SP500_Comp['Fulltimeemployees'].mode()[0]

print(SP500_Comp[SP500_Comp['Fulltimeemployees'].isnull()])
```

Replacing the null "Fulltimeemployees" rows with the mode of the column

Missing EBITDA

Let's count the number of missing values from each sector.

```
Sector              Industry
Financial Services  Asset Management          3
                    Banks Diversified         4
                    Banks Regional           14
                    Capital Markets           4
                    Credit Services           4
                    Insurance Reinsurance     1
```

It seems all companies with missing EBITDA values are from the Financial Services sector.

COLUMN BREAKDOWN

```
for col in SP500_Comp.columns:
    b = SP500_Comp[col].unique()
    if len(b)<20:
        print(f'{col} has {len(b)} unique values -->> {b}', end = '\n\n')
```

Code to check columns with low numbers of unique values

```
Exchange has 4 unique values -->> ['NMS' 'NYQ' 'NGM' 'BTS']

Sector has 11 unique values -->> ['Technology' 'Communication Services' 'Consumer Cyclical'
 'Financial Services' 'Healthcare' 'Energy' 'Consumer Defensive'
 'Industrials' 'Utilities' 'Basic Materials' 'Real Estate']

Country has 7 unique values -->> ['United States' 'Ireland' 'United Kingdom' 'Switzerland'
'Netherlands'
 'Israel' 'Bermuda']
```

Output for above code

3. Data analysis and visualization

PAIRPLOTS

```
sns.set(style='darkgrid')
plt.figure(figsize=(15,12))
sns.pairplot(SP500_Comp, corner=True, hue='Exchange')
plt.tight_layout()
plt.savefig('foo.png')
```

Code to save all possible pairplots with exchanges highlighted in different colors

Various pairplots from the code above

OBSERVATIONS:

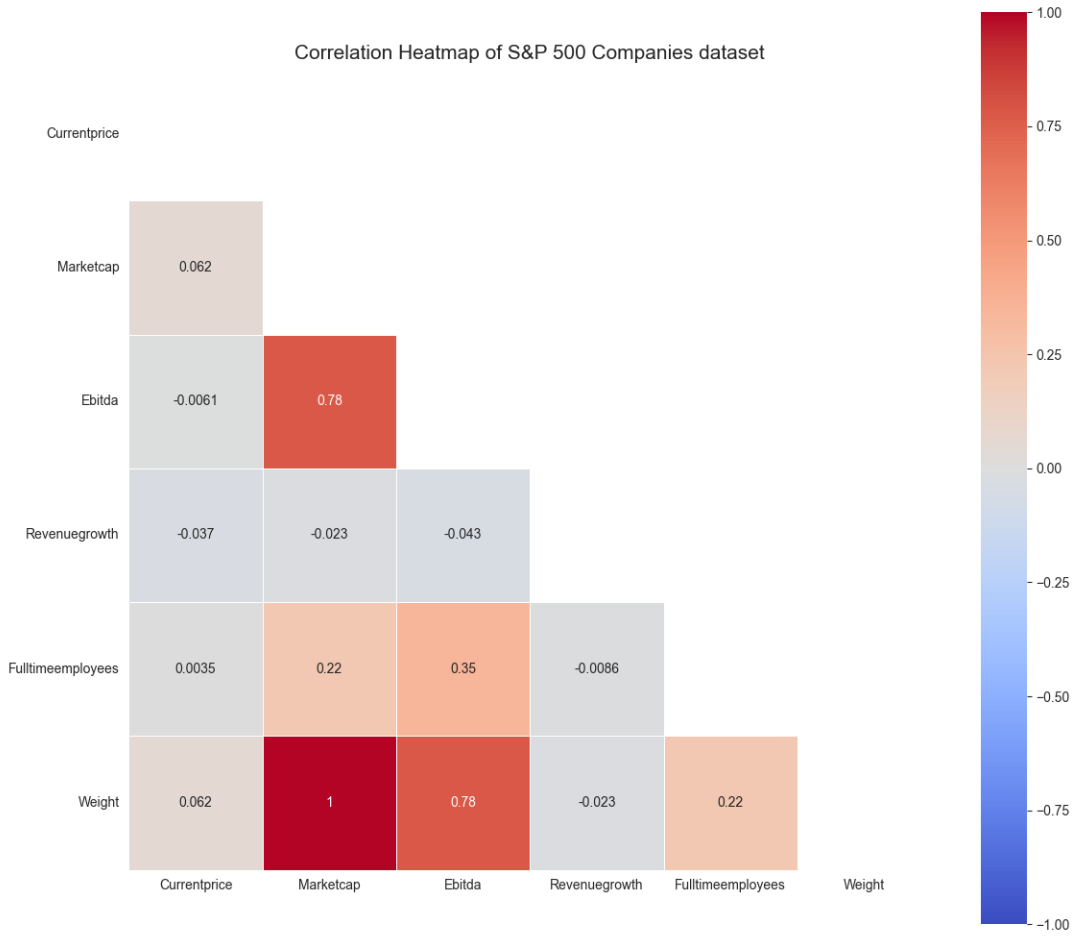
A linear correlation is observed between the marketcap and weights. The reason for this observation is that the weights in the S&P500 index are directly calculated based on the marketcap of each stock.

HEATMAPS

```
SP_corr = SP500_Comp.corr()
mask = np.zeros_like(SP_corr)
mask[np.triu_indices_from(mask)] = True
with sns.axes_style("white"):
    f, ax = plt.subplots(figsize=(12, 10))
```

```
    ax = sns.heatmap(SP_corr, mask=mask, vmax=1, vmin=-1, linewidths=.5, square=True,
cmap='coolwarm', annot=True)
    plt.title('Correlation Heatmap of S&P 500 Companies dataset', fontsize = 15)
    plt.yticks(rotation=0)
    plt.tight_layout()
    plt.savefig('foo1.png')
```

Code to generate heatmaps for linear correlations between pairplots

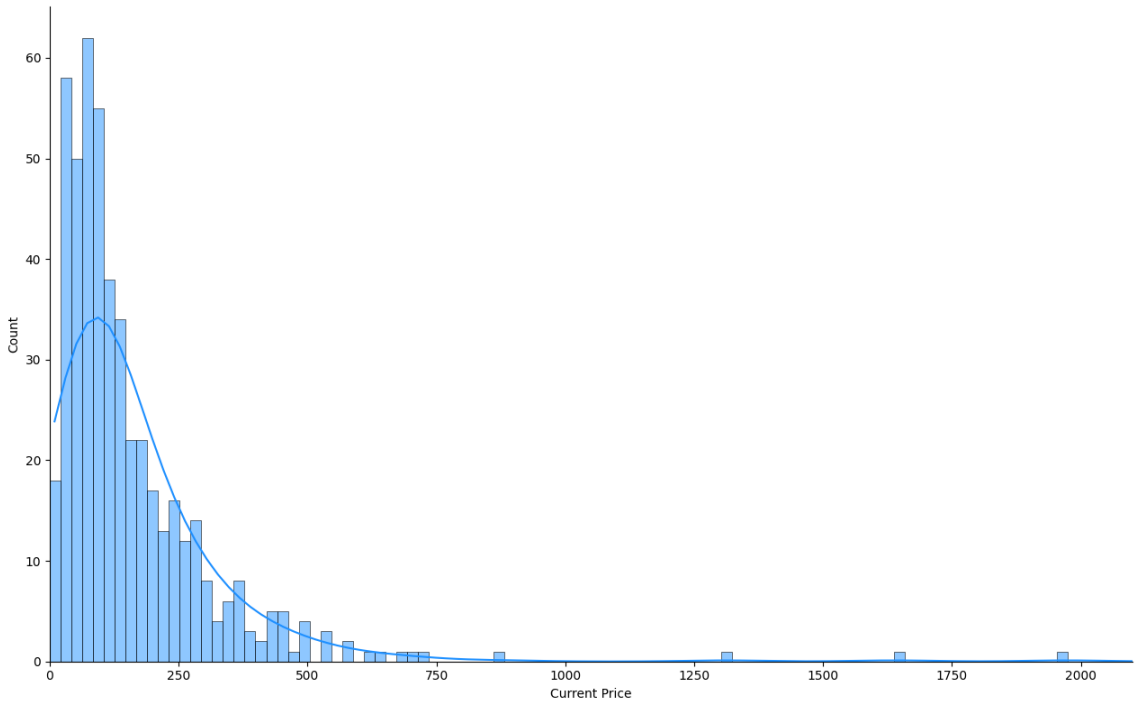


- As we can see, there's a high positive correlation = 1, between Marketcap & Weight, which has a very obvious reason, i.e., weights in S&P 500 are calculated using Marketcap. Formula: weights=Companymarketcap/Totalofallmarketcapweights
- EBITDA also shows a strong correlation with Weights as well, that is companies with higher weights, tend to show high EBITDA.
- Other features have weak positive or negative relationship with each other.

DISTRIBUTION OF CURRENTPRICE

```
d2 = sns.displot(data=SP500_Comp, x='Currentprice', kde=True, height=8, aspect=1.6,
bins=100, binrange=(0, 2100), color='dodgerblue')
d2.set(xlabel='Current Price')
plt.xlim(0, 2100)
plt.savefig('foo2.png')
```

Code to plot distribution of currentprice



Plot for currentprice vs number of stocks at that price

- The distribution of Current Price is skewed right that means, the vast majority of stocks in S&P 500 have low current price as compared to few minority stocks aka outliers seen on the right side of the plot.
- Also, there are seem to be 2 distinct groups that has the highest count of Current Price, which seems to be between 50-150.

```
print(f'The mode of the Current Price column is {SP500_Comp.Currentprice.mode()[0]}.')
```

```
The mode of the Current Price column is 13.71.
```

4. Conclusion

We successfully analysed various S&P500 index stocks as of August 2022.