Exploratory Data Analysis (EDA)

Thursday 11th December, 2025

Curvenote

In this notebook, we want to visualize different parts of the S&P 500 data we have to understand how to optimize our models with the trends or properties in the data we discover from this section. It will greatly help focus our research questions and validate our models' assumptions.

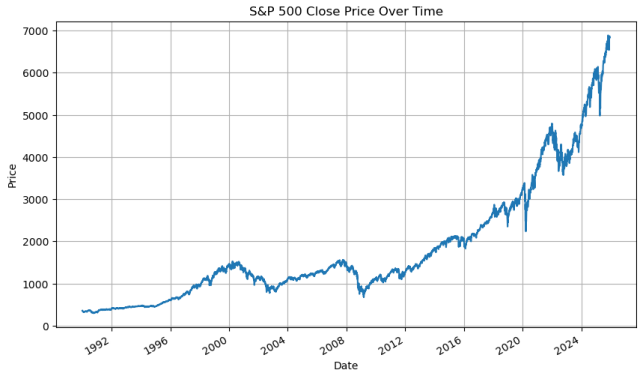
```
#installing the relevant libraries and dataset
import pandas as pd
import matplotlib.pyplot as plt
import os

df = pd.read_csv("../data/sp500.csv", index_col = "Date", parse_dates = True)
```

1 Trend Analysis

Visualizing the raw closing price allows us to identify long-term market behavior. Overall, we should see an increasing pattern as that is how our market has grown over time.

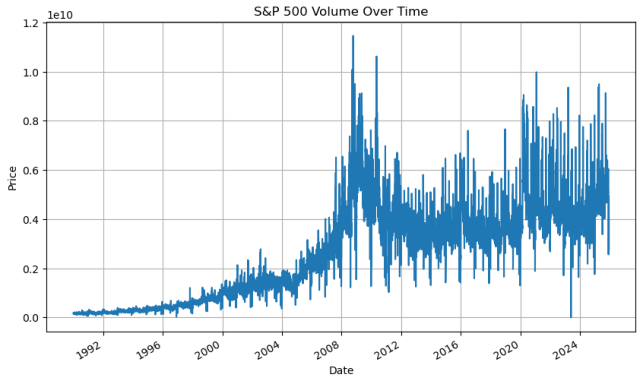
```
#First
plt.figure(figsize=(10, 6))
df["Close"].plot(title="S&P 500 Close Price Over Time")
plt.ylabel("Price")
plt.grid(True)
plt.savefig("../images/close_price.png")
plt.show()
```



2 Volatility Clustering

We also want to take a look at the volatility clustering, where essentially the idea is that large price changes tend to be followed by large price changes. We want to see how much risk poor markets have, such as 2008, compared to market conditions considered strong.

```
#Second
plt.figure(figsize=(10, 6))
df["Volume"].plot(title="S&P 500 Volume Over Time")
plt.ylabel("Price")
plt.grid(True)
plt.savefig("../images/volume.png")
plt.show()
```

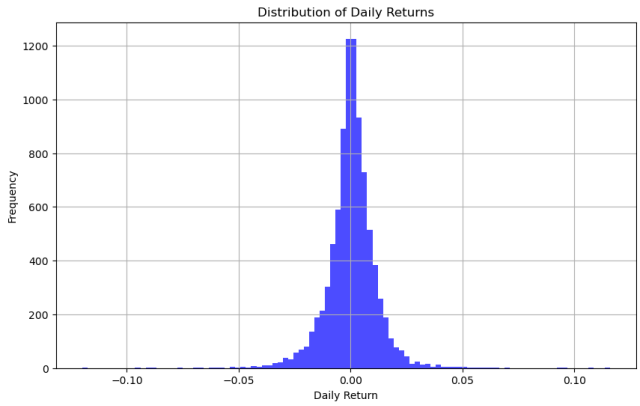


We see that amount of risk/volatility are not random in that they tend to cluster together. High volatility today gives a high indicator or increases the probability of high volatility tomorrow. Therefore, we can predict the magnitude of movement with reasonable accuracy using past volatility.

3 3. Distribution of Returns

We look at the distribution of daily Log Returns.

```
#Third
plt.figure(figsize=(10, 6))
df["Return"].hist(bins=100, color= "blue", alpha=0.7)
plt.title("Distribution of Daily Returns")
plt.xlabel("Daily Return")
plt.ylabel("Frequency")
plt.savefig("../images/returns.png")
plt.show()
```



The distribution is centered almost exactly at 0.0, indicating that on an average day, the market does not move much. Below, we give a more numerical explanation of our visualization.

```
print("Basic stats of daily returns:")
print(df["Return"].describe())
```

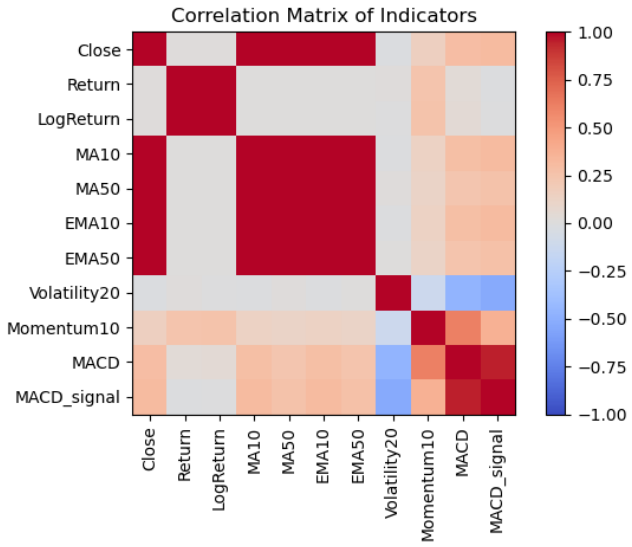
```
Basic stats of daily returns:
count    9051.000000
mean        0.000390
std         0.011377
min        -0.119841
25%        -0.004429
50%         0.000603
75%         0.005710
max         0.115800
Name: Return, dtype: float64
```

4 Feature Correlation

We visualize the correlation matrix to see if there is any multicollinearity in order to better work our models. Ideally, we find features that have a non-zero correlation with LogReturn. If correlations are near zero, linear models are not optimal.

```
#Fourth
corr = df[["Close", "Return", "LogReturn", "MA10", "MA50", "EMA10", "EMA50", "Volatility20",

plt.imshow(corr, cmap="coolwarm", vmin= -1, vmax=1)
plt.colorbar()
plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
plt.yticks(range(len(corr.columns)), corr.columns)
plt.title("Correlation Matrix of Indicators")
plt.tight_layout()
plt.savefig("../images/correlation_matrix.png")
plt.show()
```



We see that there is high correlation between MA10, MA50, and EMA features. Moreover, in LogReturn row, the correlation between our technical indicators and the LogReturn is near 0. This means that there is no simple linear relationship between yesterday's indicators and tomorrow's return. Thus, the need for non-linear models like Random Forest or LSTMs seem to be a good idea.