Preparing the Data and Feature Engineering

Thursday 11th December, 2025

Curvenote

This notebook is responsible for getting all the financial data we need from the yfinance API.

```
import pandas as pd
import os
import sys

#Going to use the scripts folder now cus project said so.
sys.path.append(os.path.abspath(os.path.join("..")))
from scripts.data_process import download_sp500, add_technical_indicators
```

1 Ingesting Data

We get the S&P500 Index historical data beginning from 01/01/1990. This captures a lot of various market conditions, thus allowing us to work with less biased data.

```
#Downloading the Data

sp500 = download_sp500(start_date="1990 -01 -01")
print(sp500.head())
print(sp500.tail())
```

```
[*********************100%***********************]  1 of 1 completed

Close        High         Low        Open     Volume
Date
1990 -01 -02  359.690002  359.690002  351.980011  353.399994  162070000
1990 -01 -03  358.760010  360.589996  357.890015  359.690002  192330000
1990 -01 -04  355.670013  358.760010  352.890015  358.760010  177000000
1990 -01 -05  352.200012  355.670013  351.350006  355.670013  158530000
1990 -01 -08  353.790009  354.239990  350.540009  352.200012  140110000
                  Close         High          Low         Open      Volume
```

```
Date
2025 -12 -03  6849.720215  6862.419922  6810.430176  6815.290039  4736780000
2025 -12 -04  6857.120117  6866.470215  6827.120117  6866.470215  4872440000
2025 -12 -05  6870.399902  6895.779785  6858.290039  6866.319824  4944560000
2025 -12 -08  6846.509766  6878.270020  6827.189941  6875.200195  4757130000
2025 -12 -09  6840.509766  6864.919922  6837.430176  6840.609863  2757882000
```

2 Feature Engineering

Since we already acknowledged that financial data is very noisy and messy, we wanted to add meaningful technical indicators.

- **Moving Averages (MA10, MA50):** Deals with short-term fluctuations to identify the underlying trend.
- **Momentum (Momentum10):** Captures price changes (Close price today vs. 10 days ago).
- **Volatility (Volatility20):** Rolling standard deviation of returns, being a proxy for market risk.
- **MACD:** An indicator that shows the relationship between two moving averages of a stock's price.
- **Log Returns:** Like everyone, we model *log returns* rather than actual prices to ensure stability.

```
# feature engineering data
sp500_feat = add_technical_indicators(sp500)
print(sp500_feat.head())
```

```
Close        High         Low        Open     Volume  MA10  \
Date
1990 -01 -02  359.690002  359.690002  351.980011  353.399994  162070000   NaN
1990 -01 -03  358.760010  360.589996  357.890015  359.690002  192330000   NaN
1990 -01 -04  355.670013  358.760010  352.890015  358.760010  177000000   NaN
1990 -01 -05  352.200012  355.670013  351.350006  355.670013  158530000   NaN
1990 -01 -08  353.790009  354.239990  350.540009  352.200012  140110000   NaN

            MA50       EMA10       EMA50    Return  LogReturn  Volatility20  \
Date
1990 -01 -02   NaN  359.690002  359.690002       NaN        NaN           NaN
1990 -01 -03   NaN  359.520913  359.653532 -0.002586  -0.002589           NaN
1990 -01 -04   NaN  358.820749  359.497316 -0.008613  -0.008650           NaN
1990 -01 -05   NaN  357.616979  359.211147 -0.009756  -0.009804           NaN
```

```
1990 -01 -08   NaN  356.921166  358.998553  0.004514   0.004504            NaN

            Momentum10      MACD  MACD_signal
Date
1990 -01 -02         NaN  0.000000     0.000000
1990 -01 -03         NaN -0.074187    -0.014837
1990 -01 -04         NaN -0.377962    -0.087462
1990 -01 -05         NaN -0.888463    -0.247662
1990 -01 -08         NaN -1.151466    -0.428423
```

3 Storing Data

Now we are just going to save the data we used into the data folder so we do not have to call to yfinance api everytime we want to analyze or use the dataset.

```
final_data = "../data/sp500.csv"
sp500_feat.to_csv(final_data)
```