Option C - Task 2: Data processing 1

I created the load_stock_data function which basically loads the data, filters out the data that is not useful, and splits the dataset into training and testing.
For the first part, the parameters start_date: str, end_date: str, are added, which are used in df = yf.download(symbol, start=start_date, end=end_date, progress=False) and start and end dates are chosen.

For the data that is not useful or null, nan_method is used: str = "ffill"

```
if df.isnull().sum().sum() > 0:

    if nan_method == "drop":

        df = df.dropna()

    elif nan_method == "ffill":

        df = df.fillna(method="ffill")

    elif nan_method == "bfill":

        df = df.fillna(method="bfill")

    elif nan_method == "mean":

        for col in df.select_dtypes(include=["float64", "int64"]).columns:

            df[col].fillna(df[col].mean(), inplace=True)

    else:

        raise ValueError(f"Unknown nan_method '{nan_method}'.")
```

and an if block that allows you to delete the data, use the next data, the previous data, or the average.

For the third part of split data, these parameters were added:

```
split_method: str | None = None,

part: str = "all", # "train" | "test" | "all"

train_size: float = 0.8,

split_date: str | None = None,

random_state: int = 42
```

which allows splitting into training and test, by date, ratio, or randomness.

```
if split_method is None or part == "all":

    return df
```

```
if split_method == "date":
    cutoff = pd.to_datetime(split_date)
    return df[df.index <= cutoff] if part == "train" else df[df.index > cutoff]


if split_method == "ratio":
    cut = int(len(df) * train_size)
    return df.iloc[:cut] if part == "train" else df.iloc[cut:]


if split_method == "random":
    tr_idx, te_idx = train_test_split(df.index, train_size=train_size,
                                      shuffle=True, random_state=random_state)
    return df.loc[tr_idx].sort_index() if part == "train" else df.loc[te_idx].sort_index()
```

An if block is used again to handle the logic for splitting it into whatever is required (dates, ratio, or randomness).

I was unable to perform the last two, save the data set to use later without downloading and the scale

This is due to a date confusion and for the next part it should already be implemented and documented.