Stock Price Forecasting & Risk Analysis

This project analyzes three major stocks (AAPL, MSFT, TSLA) using historical prices from Yahoo Finance. It includes:

- Trend analysis
- Volatility indicators
- Bollinger Bands
- Short-term forecasts using Holt-Winters
- Side-by-side comparisons
- Business interpretation

Single-Stock Analysis: AAPL

```
import yfinance as yf
import pandas as pd

df = yf.download("AAPL", start="2015-01-01", end="2024-12-31")
df.head()
```

```
/tmp/ipython-input-1618978389.py:4: FutureWarning: YF.download() has changed argument auto_adjust default to True
  df = yf.download("AAPL", start="2015-01-01", end="2024-12-31")
[*********************100%***********************]  1 of 1 completed
```

Price	Close	High	Low	Open	Volume
Ticker	AAPL	AAPL	AAPL	AAPL	AAPL
Date					
2015-01-02	24.237549	24.705318	23.798599	24.694233	212818400
2015-01-05	23.554745	24.086805	23.368524	24.006996	257142000
2015-01-06	23.556959	23.816338	23.195601	23.619033	263188400
2015-01-07	23.887276	23.987036	23.654499	23.765345	160423600
2015-01-08	24.805084	24.862725	24.097887	24.215385	237458000

```
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Make plots look nicer
plt.style.use("seaborn-v0_8")

# Download stock data
df = yf.download("AAPL", start="2015-01-01", end="2024-12-31")
df.head()
```

```
/tmp/ipython-input-3271675975.py:11: FutureWarning: YF.download() has changed argument auto_adjust default to True
  df = yf.download("AAPL", start="2015-01-01", end="2024-12-31")
[*********************100%***********************]  1 of 1 completed
```

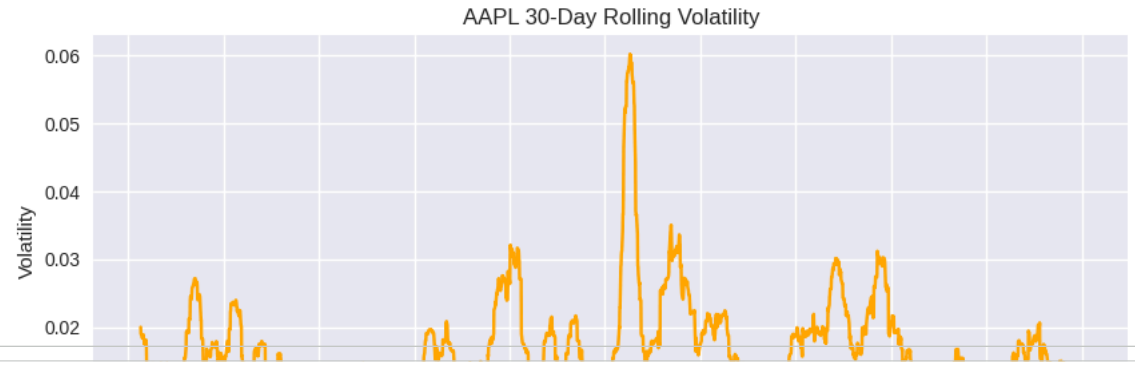
Price	Close	High	Low	Open	Volume
Ticker	AAPL	AAPL	AAPL	AAPL	AAPL
Date					
2015-01-02	24.237549	24.705318	23.798599	24.694233	212818400
2015-01-05	23.554745	24.086805	23.368524	24.006996	257142000
2015-01-06	23.556959	23.816338	23.195601	23.619033	263188400
2015-01-07	23.887276	23.987036	23.654499	23.765345	160423600
2015-01-08	24.805084	24.862725	24.097887	24.215385	237458000

```
plt.figure(figsize=(10,4))
plt.plot(df["Close"])
plt.title("AAPL Closing Price")
plt.xlabel("Date")
plt.ylabel("Price ($)")
plt.show()
```


```
df["Returns"] = df["Close"].pct_change()
df["RollingVol"] = df["Returns"].rolling(30).std()

# Quick preview
df[["Close", "Returns", "RollingVol"]].head()
```

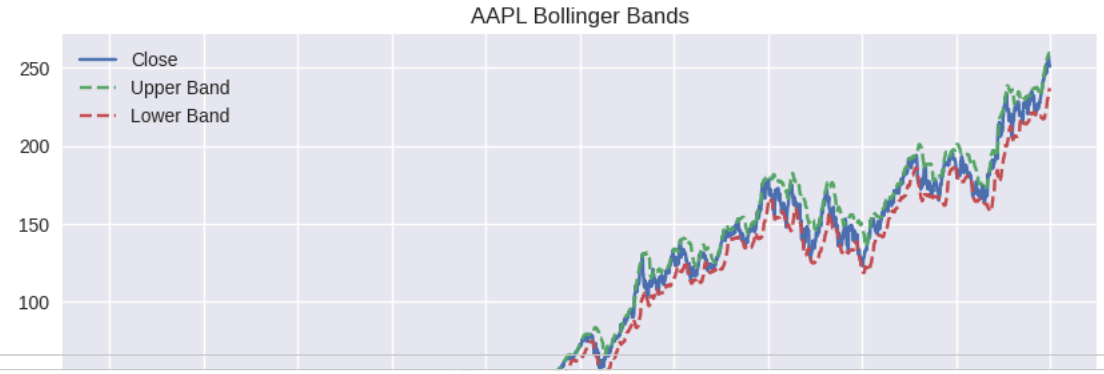
Price	Close	Returns	RollingVol
Ticker	AAPL		
Date			
2015-01-02	24.237549	NaN	NaN
2015-01-05	23.554745	-0.028171	NaN
2015-01-06	23.556959	0.000094	NaN
2015-01-07	23.887276	0.014022	NaN
2015-01-08	24.805084	0.038422	NaN

```
plt.figure(figsize=(10,4))
plt.plot(df["RollingVol"], color="orange")
plt.title("AAPL 30-Day Rolling Volatility")
plt.ylabel("Volatility")
plt.show()
```




```
close = df["Close"]
sma20 = close.rolling(20).mean()
std20 = close.rolling(20).std()

df["SMA20"] = sma20
df["Upper"] = sma20 + (2 * std20)
df["Lower"] = sma20 - (2 * std20)

plt.figure(figsize=(10,4))
```

```
plt.plot(close, label="Close")
plt.plot(df["Upper"], label="Upper Band", linestyle="--")
plt.plot(df["Lower"], label="Lower Band", linestyle="--")
plt.title("AAPL Bollinger Bands")
plt.legend()
plt.show()
```

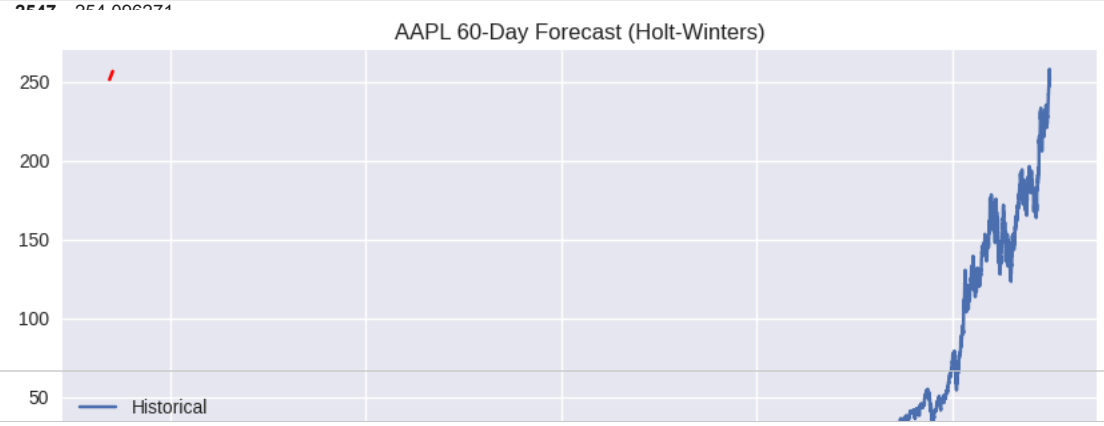



```
from statsmodels.tsa.holtwinters import ExponentialSmoothing

series = df["Close"].dropna()

# Build model
model = ExponentialSmoothing(series, trend="add", seasonal=None)
fitted = model.fit()

# Forecast next 60 days
forecast = fitted.forecast(60)
forecast
```

```
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Pr
  return get_prediction_index(
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: FutureWarning: No supported index is available. I
  return get_prediction_index(
```

	0
2515	251.206587
2516	251.296890
2517	251.387193
2518	251.477495
2519	251.567798
2520	251.658100
2521	251.748403
2522	251.838706
2523	251.929008
2524	252.019311
2525	252.109614
2526	252.199916
2527	252.290219
2528	252.380521
2529	252.470824
2530	252.561127
2531	252.651429
2532	252.741732
2533	252.832034
2534	252.922337
2535	253.012640
2536	253.102942
2537	253.193245
2538	253.283548
2539	253.373850
2540	253.464153
2541	253.554455

```
plt.figure(figsize=(10,4))
plt.plot(series, label="Historical")
plt.plot(forecast.index, forecast, label="Forecast", color="red")
plt.title("AAPL 60-Day Forecast (Holt-Winters)")
plt.legend()
plt.show()
```



Price, volatility, and Bollinger Band movements show clear regime shifts in Apple's stock. Periods where bands widen and volatility increases typically align with macro-level disruptions. When volatility cools and bands tighten, the trend stabilizes.

The Holt-Winters forecast points to a modest continuation of the trend with relatively low near-term volatility. From a strategic standpoint, this kind of modeling helps teams evaluate short-term expectations, plan around potential price stability, and identify when risk conditions are improving or deteriorating.

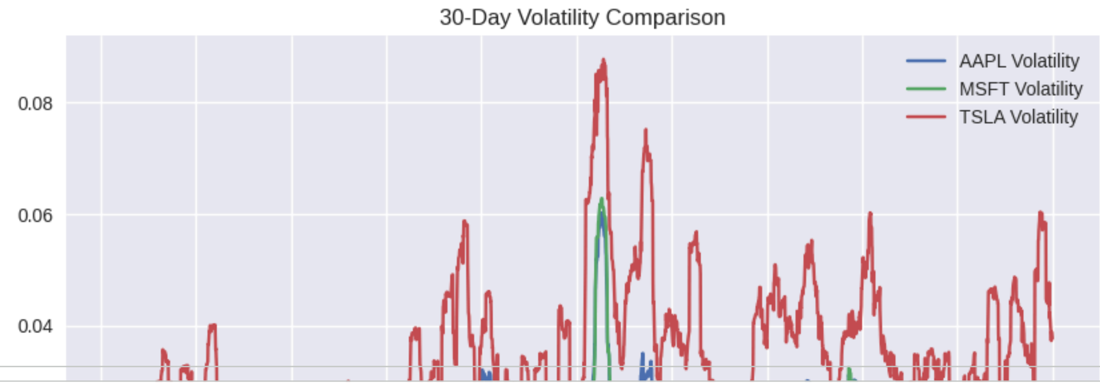
Multi-Stock Analysis: AAPL, MSFT, TSLA

```
tickers = ["AAPL", "MSFT", "TSLA"]

data = {}
for t in tickers:
    print(f"Downloading {t}...")
    data[t] = yf.download(t, start="2015-01-01", end="2024-12-31")
```

```
Downloading AAPL...
/tmp/ipython-input-1653847169.py:6: FutureWarning: YF.download() has changed argument auto_adjust default to True
  data[t] = yf.download(t, start="2015-01-01", end="2024-12-31")
[*********************100%***********************]  1 of 1 completed
/tmp/ipython-input-1653847169.py:6: FutureWarning: YF.download() has changed argument auto_adjust default to True
  data[t] = yf.download(t, start="2015-01-01", end="2024-12-31")
Downloading MSFT...
[*********************100%***********************]  1 of 1 completed
/tmp/ipython-input-1653847169.py:6: FutureWarning: YF.download() has changed argument auto_adjust default to True
  data[t] = yf.download(t, start="2015-01-01", end="2024-12-31")
Downloading TSLA...
[*********************100%***********************]  1 of 1 completed
```

```
plt.figure(figsize=(10,5))

for t in tickers:
    plt.plot(data[t]["Close"], label=t)

plt.title("Stock Price Comparison")
plt.legend()
plt.show()
```


```
for t in tickers:
    df = data[t]
    df["Returns"] = df["Close"].pct_change()
    df["Vol30"] = df["Returns"].rolling(30).std()

plt.figure(figsize=(10,5))
for t in tickers:
    plt.plot(data[t]["Vol30"], label=f"{t} Volatility")

plt.title("30-Day Volatility Comparison")
plt.legend()
plt.show()
```




```
from statsmodels.tsa.holtwinters import ExponentialSmoothing

forecasts = {}

for t in tickers:
    series = data[t]["Close"].dropna()
    model = ExponentialSmoothing(series, trend="add", seasonal=None)
    fitted = model.fit()
    forecasts[t] = fitted.forecast(60)
```

```
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Pr
  return get_prediction_index(
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: FutureWarning: No supported index is available. I
  return get_prediction_index(
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Pr
  return get_prediction_index(
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: FutureWarning: No supported index is available. I
  return get_prediction_index(
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Pr
  return get_prediction_index(
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837: FutureWarning: No supported index is available. I
  return get_prediction_index(
```

```
t = "AAPL"   # change to MSFT or TSLA if you want
series = data[t]["Close"]
forecast = forecasts[t]

plt.figure(figsize=(10,4))
plt.plot(series, label="Historical")
```