Predicting Short-Term FAANG Stock Price Movements

Why This Project?

- Goal: Explore whether machine learning can improve short-term FAANG stock predictions
- Motivation: Help investors better understand momentum & volatility signals
- Outcome: Develop a data-driven framework to support smarter decision-making

Introduction

- Goal: Predict next-day price direction for FAANG stocks
- Binary classification: up (1) or down (0)
- The most challenge -> Short-term returns are always noisy

Dataset Overview

- Daily OHLCV data across FAANG stocks
- Date converted to datetime, sorted by Company & Date
- Missing values are due to adding feature engineering and have been handled, duplicates removed

	Date	Open	High	Low	Close	Adj Close	Volume	Company
0	1997-05-15	2.437500	2.500000	1.927083	1.958333	1.958333	72156000.0	Amazon
1	1997-05-16	1.968750	1.979167	1.708333	1.729167	1.729167	14700000.0	Amazon
2	1997-05-19	1.760417	1.770833	1.625000	1.708333	1.708333	6106800.0	Amazon
3	1997-05-20	1.729167	1.750000	1.635417	1.635417	1.635417	5467200.0	Amazon
4	1997-05-21	1.635417	1.645833	1.375000	1.427083	1.427083	18853200.0	Amazon

Feature Engineering

- Returns: 1, 3, 5, 10-day pct change
- Moving averages: MA7, MA14, MA30
- Volatility: 7,14,30-day rolling std
- Intraday spreads: High-Low, Open-Close
- Lagged returns: Lag1, Lag2, Lag3

```
faang.head(10)
```

Date	Open	High	Low	Close	Adj Close	Volume	Company	Return1	Return3	...	MA30	Volatility7	Volatility14	Volatility30	High
1997-06-27	1.515625	1.515625	1.479167	1.489583	1.489583	1188000.0	Amazon	-0.013794	-0.013794	...	1.542188	0.010086	0.024538	0.047449	
1997-06-30	1.510417	1.598958	1.479167	1.541667	1.541667	2746800.0	Amazon	0.034965	0.020690	...	1.535938	0.017291	0.021896	0.043339	
1997-07-01	1.541667	1.541667	1.510417	1.515625	1.515625	1292400.0	Amazon	-0.016892	0.003448	...	1.529514	0.018406	0.021215	0.043384	
1997-07-02	1.515625	1.593750	1.510417	1.588542	1.588542	3882000.0	Amazon	0.048110	0.066434	...	1.527951	0.024352	0.022418	0.043690	
1997-07-03	1.598958	1.916667	1.593750	1.911458	1.911458	12577200.0	Amazon	0.203278	0.239864	...	1.544097	0.077456	0.058538	0.051471	
1997-07-07	1.833333	2.020833	1.833333	2.000000	2.000000	8053200.0	Amazon	0.046322	0.319588	...	1.564236	0.075776	0.058747	0.051475	

Target Definition

- Target = 1 if tomorrow Adj Close > today
- Computed within each stock company

```
# 7. Target variable: tomorrow movement
faang["Target"] = (
    faang.groupby("Company", observed=False)["Adj Close"].shift(-1) > faang["Adj Close"]
).astype(int)
```

Train/Test Strategy

- 70% training / 30% testing split
- shuffle=False because this is time series and to preserve time order

```
#TRAIN / TEST SPLIT
X = faang[feature_cols]
y = faang["Target"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, shuffle=False
)
```

Models Compared

- Logistic Regression
- Random Forest
- Gradient Boosting

Scaling Decisions

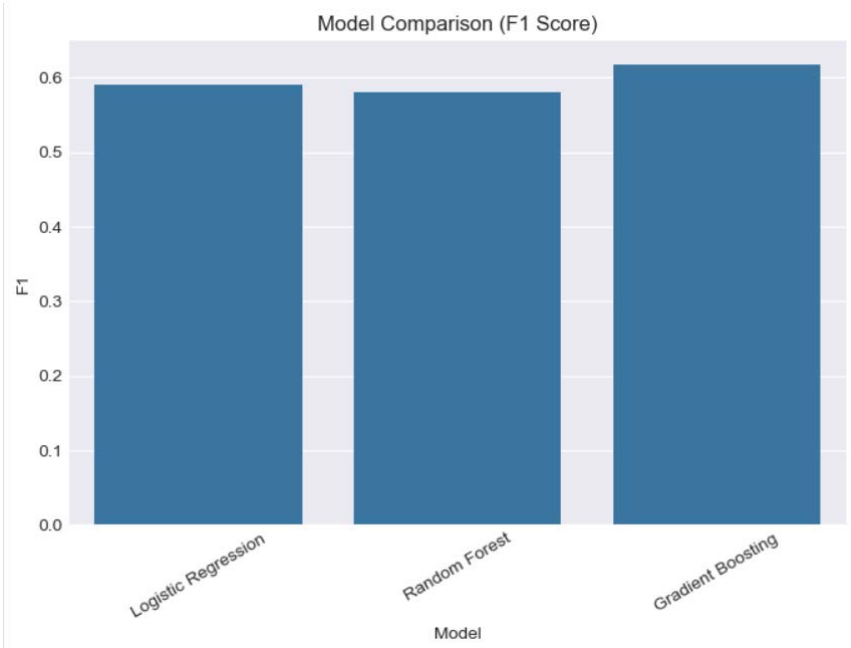
- StandardScaler applied to Logistic Regression
- Random Forest & Gradient Boosting models use raw features

Evaluation Metrics

- Accuracy
- Precision
- Recall
- F1-score (primary metric)

Model Results Summary

- Logistic Regression: Accuracy ~0.51, F1 ~0.59
- Random Forest: Accuracy ~0.52, F1 ~0.58
- Gradient Boosting: Accuracy ~0.52, F1 ~0.62

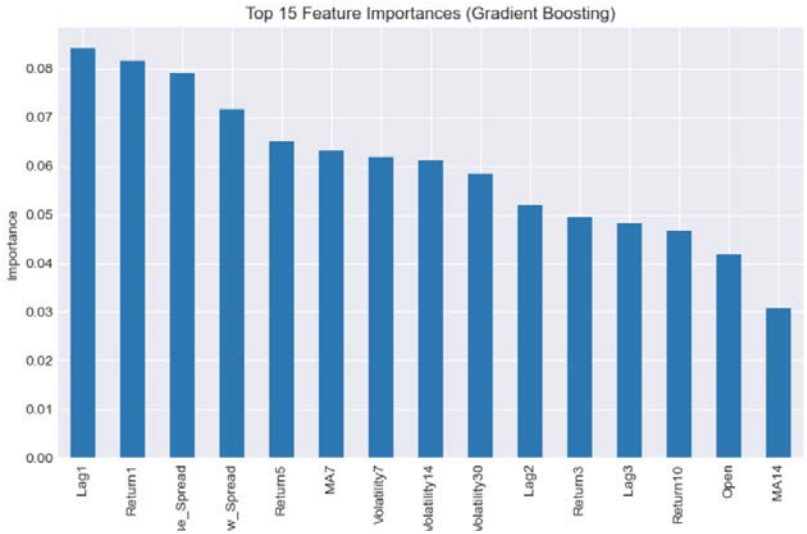


Why Gradient Boosting Wins

- Highest recall → captures more true upward moves
- Best F1-score across models
- Handles non-linear interactions

Important Features

- Lag1, Lag3 returns
- High_Low_Spread, Open_Close_Spread
- Return1, Return5
- MA7, rolling volatilities



Interpretation

- Momentum signals contribute strongly
- Volatility regime helps identify trend strength
- Small accuracy gains are expected in noisy markets

Recommendations

- Daily stock returns are noisy and close to random walk. Use Gradient Boosting as ranking tool for long entries
- Combine with risk management rules
- Retrain regularly with rolling window

Future Work

- Incorporate macro factors or sector ETFs
- Try multi-class forecasting (big up / up / flat / down)
- Consider transaction cost modeling