Group 24

CS3244 Stock Trend Prediction

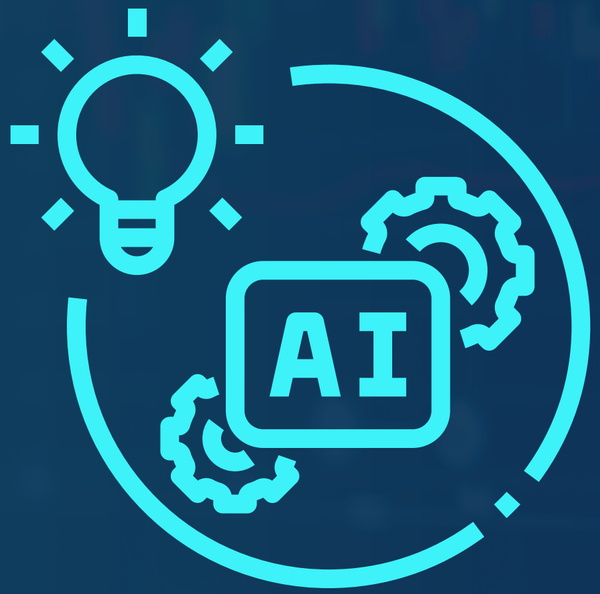
Team Members: Chew Yun Yi Wang Chengmao
Gao Yening Brandon Ong Cae Jun
Leong Deng Jun Zhang Yijian

Check out our GitHub Repo here!

https://github.com/brandono7/stock_price_prediction

Content:

Group 24



Introduction

Stock Market **Analysis**

Group 24

Introduction

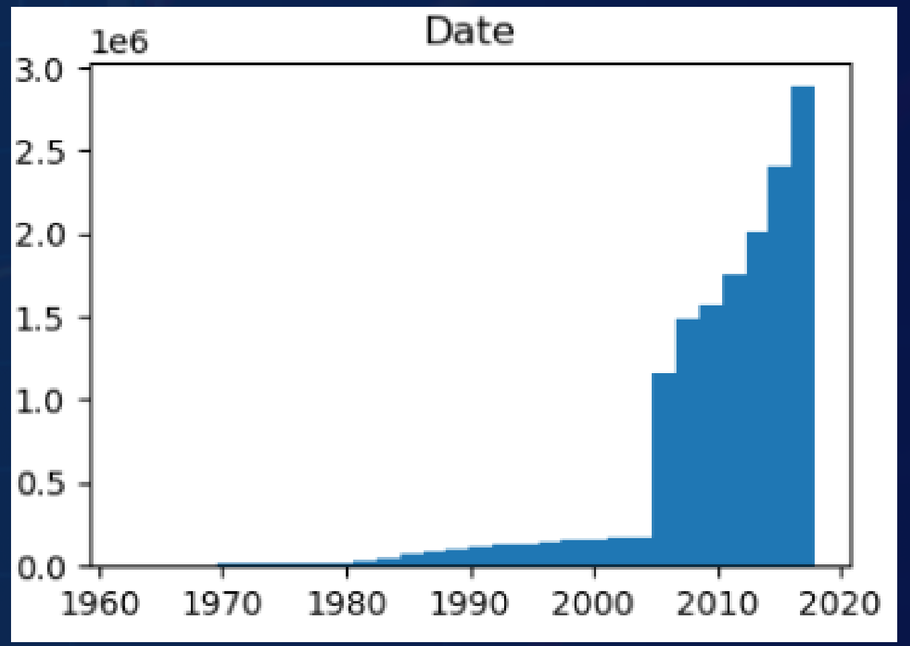
In this project, we are given historical daily prices and volumes for all U.S. stocks and ETFs. To narrow our focus, we select 50 U.S. stocks and apply machine learning techniques to analyze their price movements.

Rather than predicting exact stock prices—which can be highly volatile and noisy—we simplify the problem into a classification task: predicting the daily price direction (up or down). This transformation makes the problem more practical for short-term trading decisions and reduces sensitivity to extreme values.

Our goal is to explore and compare various machine learning models to determine which ones perform best in forecasting the directional movement of stock prices.

Group 24

Data Preprocessing



Each stock's data includes:

- Date
- Open, High, Low, Close prices
- Volume traded

To prepare the data for modeling, we perform the following preprocessing steps:

- Remove any missing or duplicate entries.
- Ensure all selected stocks have sufficient historical data.

Group 24

Data Preprocessing

We filter out the stocks that have full data from 2011-01-01 to 2017-01-01

Use marketcap data from yahoo finance to filter out the top 50 stocks from the stocks we have.

Ticker	Company Name	Industry
AAPL	**Apple Inc.**	**Information Technology**
NVDA	**NVIDIA Corporation**	**Information Technology**
MSFT	**Microsoft Corporation**	**Information Technology**
AMZN	Amazon.com, Inc.	Consumer Discretionary
GOOGL	Alphabet Inc.	Communication Services
LLY	Eli Lilly and Company	Health Care
WMT	Walmart Inc.	Consumer Staples
JPM	JPMorgan Chase & Co.	Financials
XOM	Exxon Mobil Corporation	Energy
UNH	UnitedHealth Group Incorporated	Health Care
ORCL	Oracle Corporation	Information Technology
SYK	Stryker Corporation	Health Care

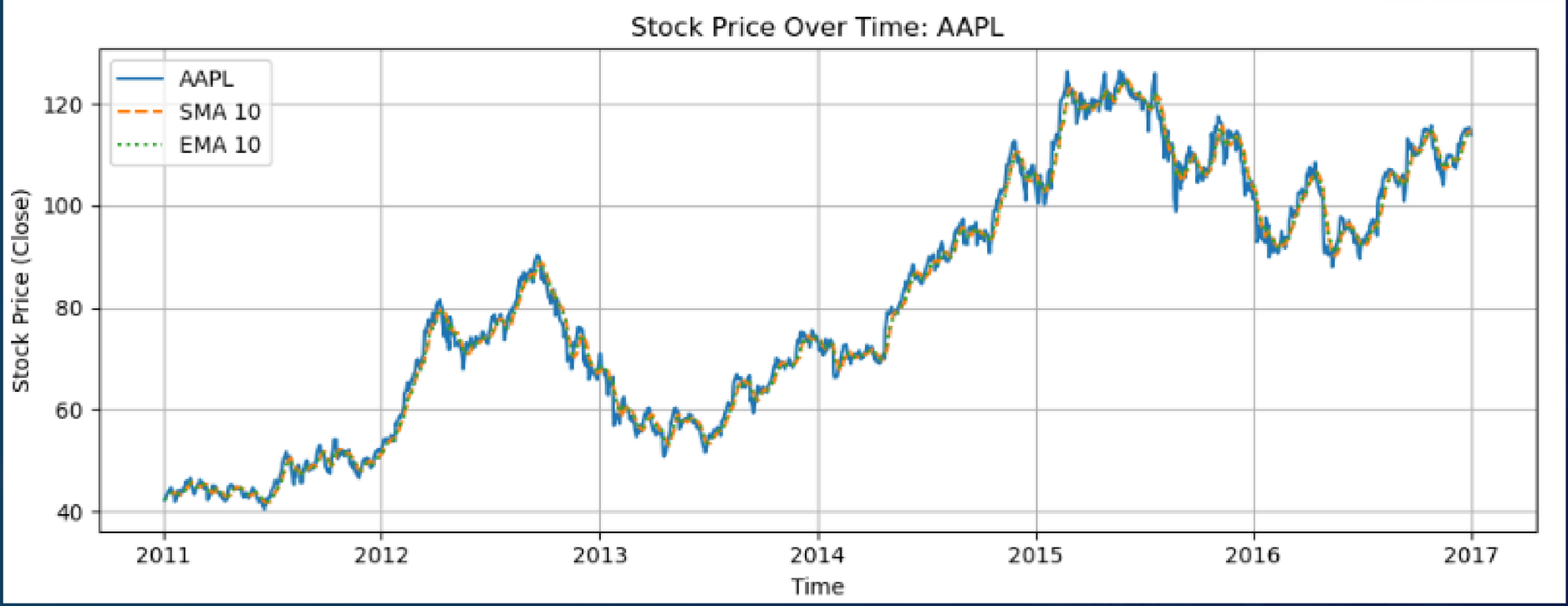
Ticker	Company Name	Industry
COST	**Costco Wholesale Corporation**	**Consumer Staples**
JNJ	**Johnson & Johnson**	**Health Care**
PG	**Procter & Gamble Company**	**Consumer Staples**
HD	Home Depot, Inc.	Consumer Discretionary
BAC	Bank of America Corporation	Financials
SMFG	Sumitomo Mitsui Financial Group, Inc.	Financials
KO	The Coca-Cola Company	Consumer Staples
CVX	Chevron Corporation	Energy
WFC	Wells Fargo & Company	Financials
CSCO	Cisco Systems, Inc.	Information Technology
MRK	Merck & Co., Inc.	Health Care
UNP	Union Pacific Corporation	Industrials

Ticker	Company Name	Industry
IBM	International Business Machines Corp.	Information Technology
GE	General Electric Company	Industrials
ABT	Abbott Laboratories	Health Care
MCD	McDonald's Corporation	Consumer Discretionary
PEP	PepsiCo, Inc.	Consumer Staples
TMO	Thermo Fisher Scientific Inc.	Health Care
AXP	American Express Company	Financials
T	AT&T Inc.	Communication Services
AMD	Advanced Micro Devices, Inc.	Information Technology
VZ	Verizon Communications Inc.	Communication Services
DIS	The Walt Disney Company	Communication Services
GS	The Goldman Sachs Group, Inc.	Financials

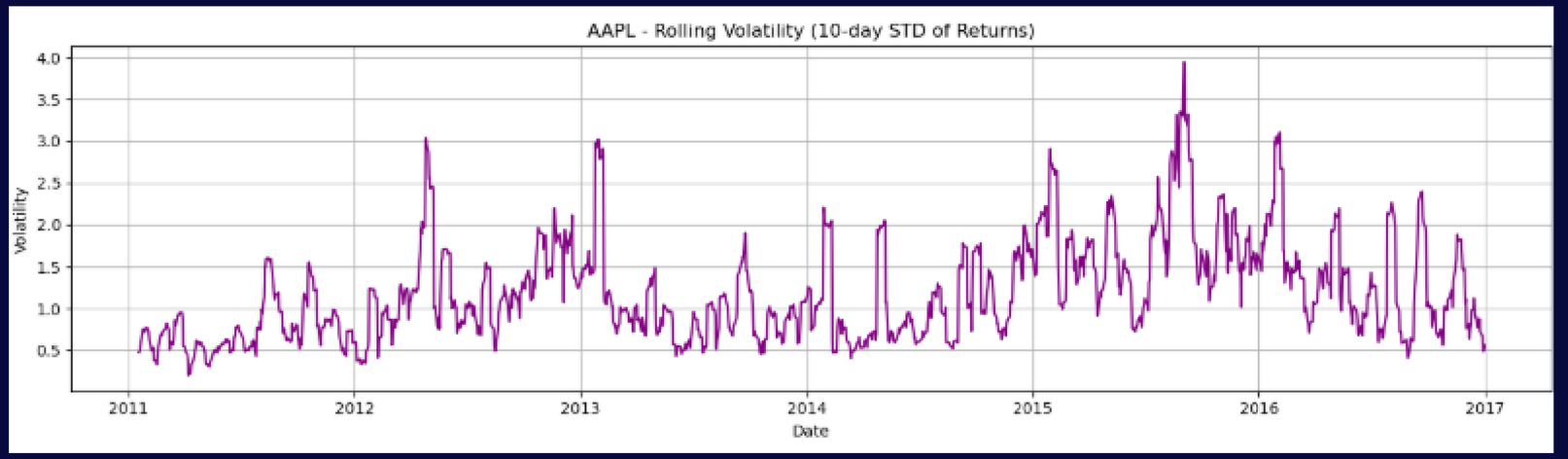
Ticker	Company Name	Industry
QCOM	QUALCOMM Incorporated	Information Technology
ADBE	Adobe Inc.	Information Technology
INTU	Intuit Inc.	Information Technology
AMGN	Amgen Inc.	Health Care
TXN	Texas Instruments Incorporated	Information Technology
CAT	Caterpillar Inc.	Industrials
PGR	The Progressive Corporation	Financials
SPGI	S&P Global Inc.	Financials
DHR	Danaher Corporation	Health Care
BSX	Boston Scientific Corporation	Health Care
PFE	Pfizer Inc.	Health Care
C	Citigroup Inc.	Financials
CMCSA	Comcast Corporation	Communication Services
HON	Honeywell International Inc.	Industrials

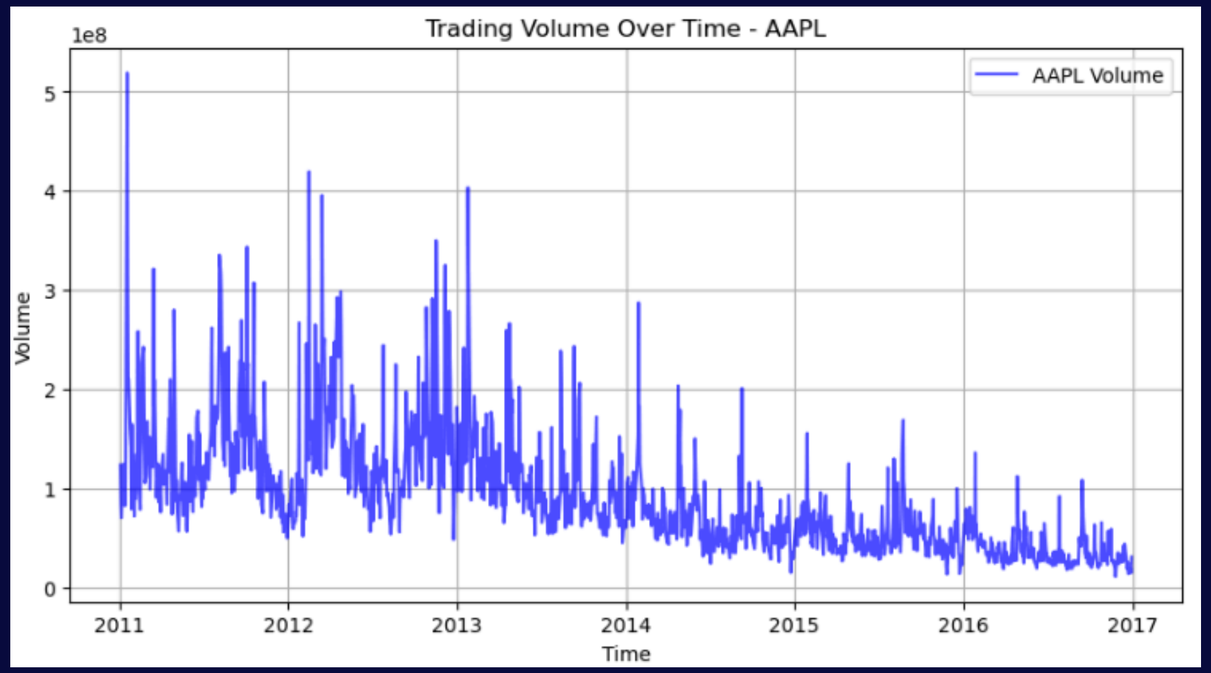
Group 24

EXPLORATORY DATA ANALYSIS



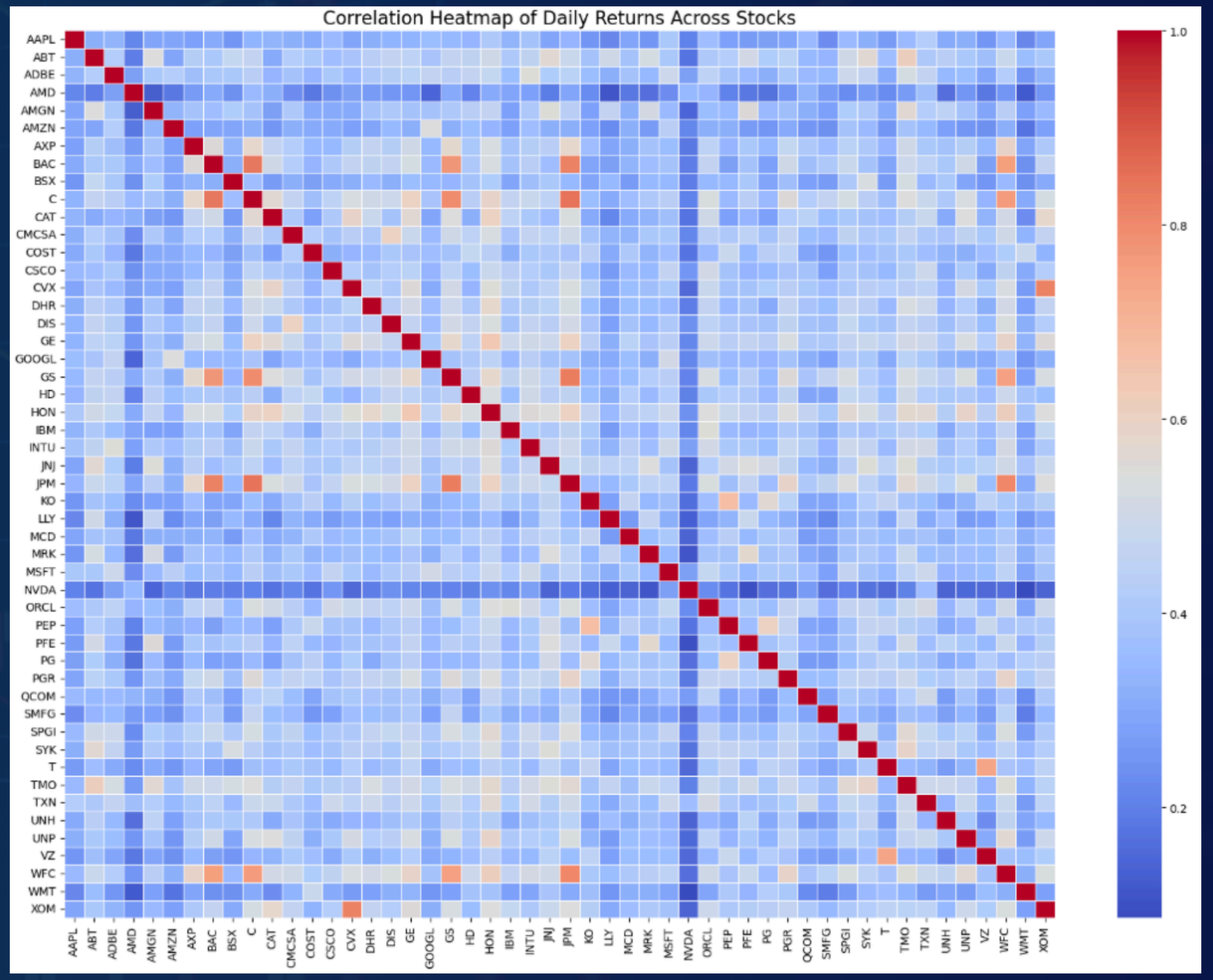
Stock Market **Analysis**


AAPL - Rolling Volatility (10-day STD of Returns)
Volatility
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
2011
2012
2013
2014
2015
2016
2017
Date


Trading Volume Over Time - AAPL
1e8
AAPL Volume
Volume
5
4
3
2
1
0
2011
2012
2013
2014
2015
2016
2017
Time

Group 24

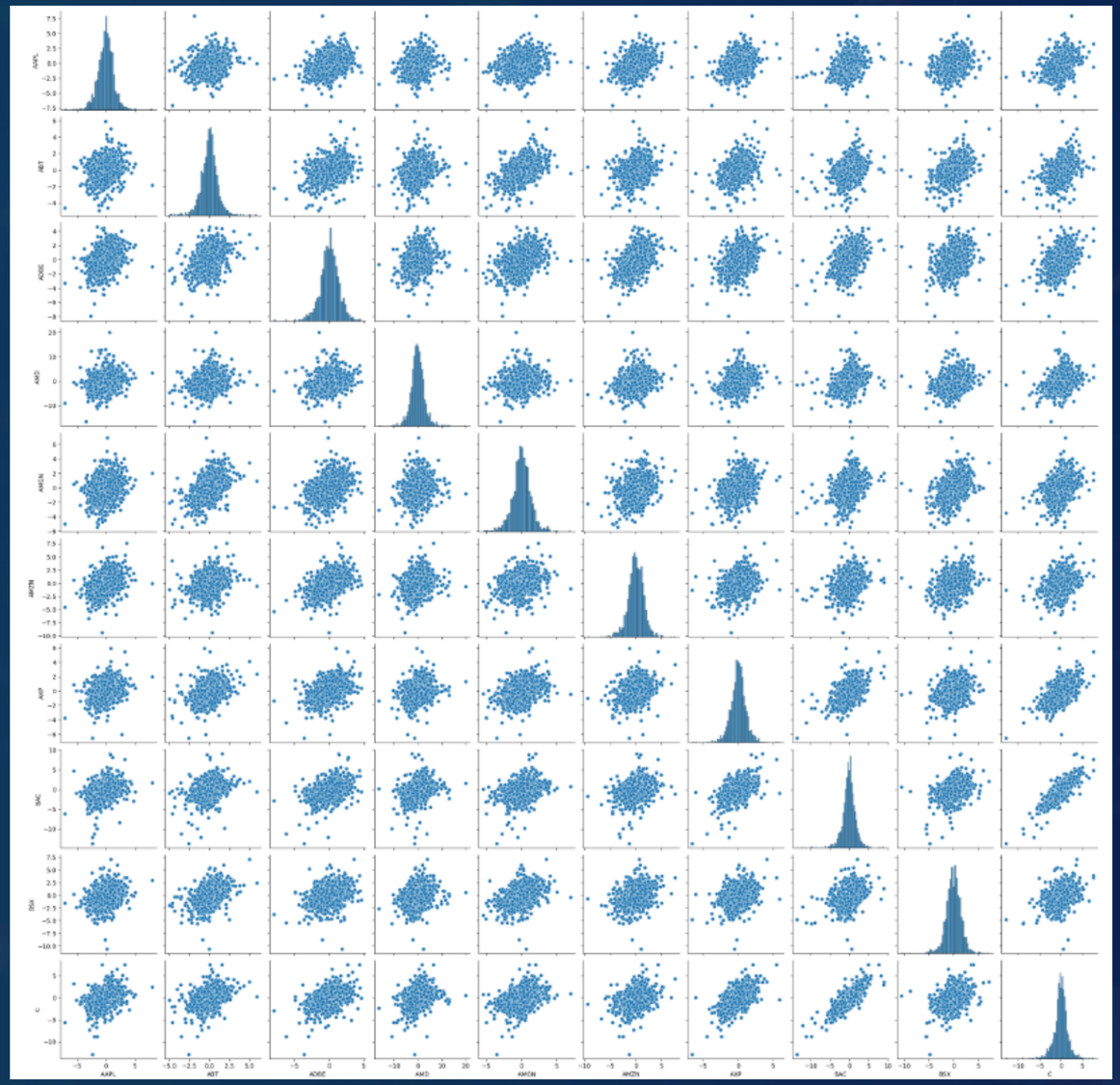
EXPLORATORY DATA ANALYSIS



COMMON SHARES

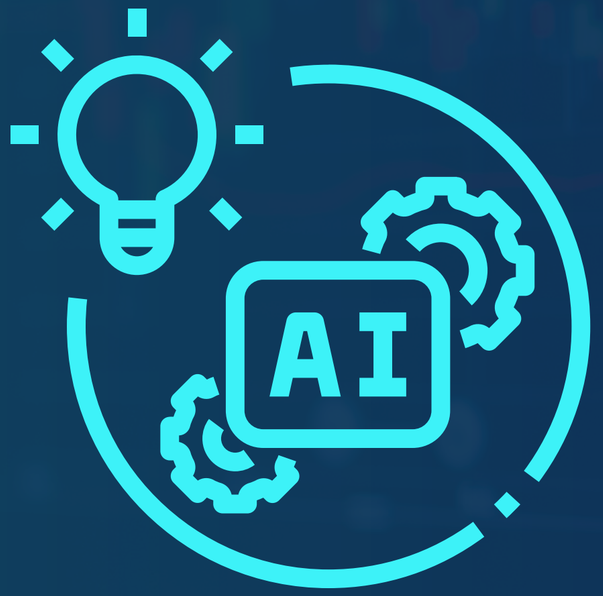
Stock Market **Analysis**

Group 24



Stock Market **Analysis**

Group 24



Evaluation Metrics

Stock Market **Analysis**

Group 24

Evaluation Metrics

Accuracy

Percentage of total predictions that are correct

F1 Score

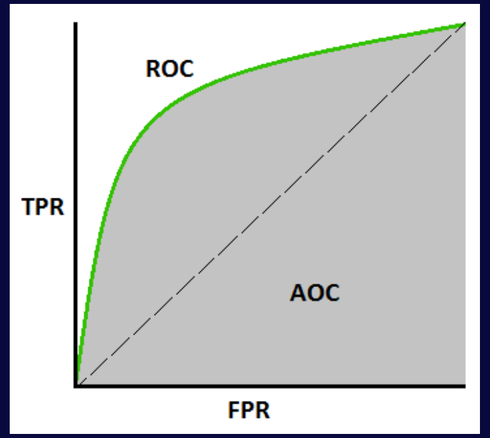
The harmonic mean of Precision and Recall

AUC – ROC

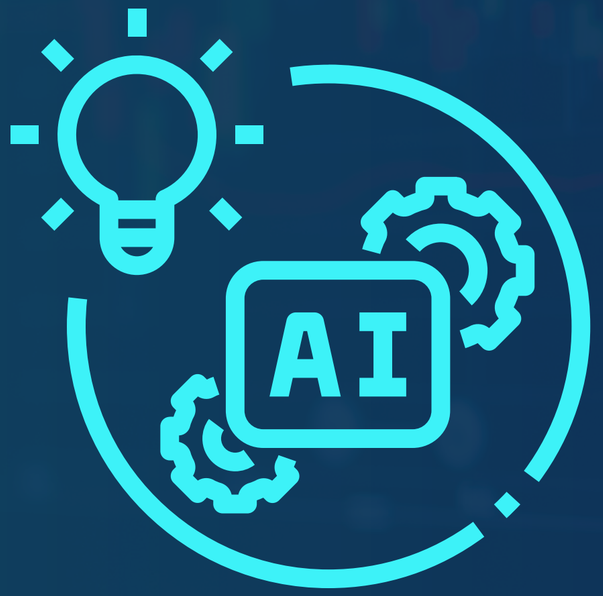
Area under the Receiver Operating Characteristic Curve

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$



Group 24

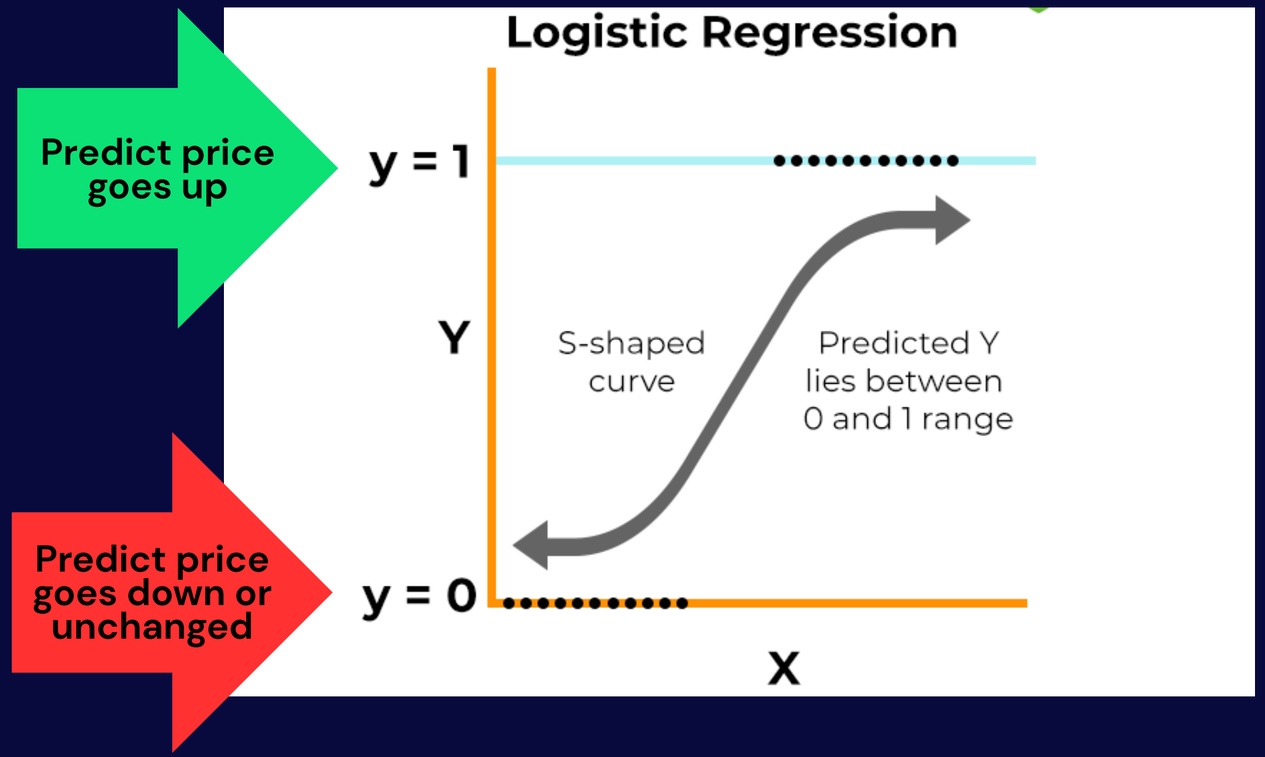


Logistic Regression (Benchmark)

Stock Market **Analysis**

Group 24

Logistic Regression (Benchmark)



- Simple and Interpretable model
- Computationally efficient to train
- Widely used as a benchmark against more complex and complicated models for classification problems

Group 24

Logistic Regression (Benchmark)

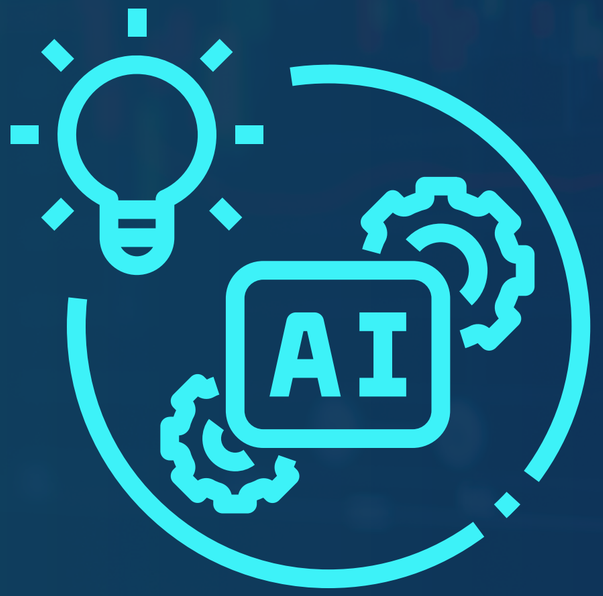
- Train-Test Split (no random shuffling) : 80% in the train set and 20% in the test set.
- No hyperparameter tuning

Key Performance Metrics	
Accuracy	0.7647
F1 Score	0.7777
AUC-ROC	0.8280

Confusion Matrix:

	Predicted Down	Predicted Up
Actual Down	5326	2004
Actual Up	1547	6213

Group 24

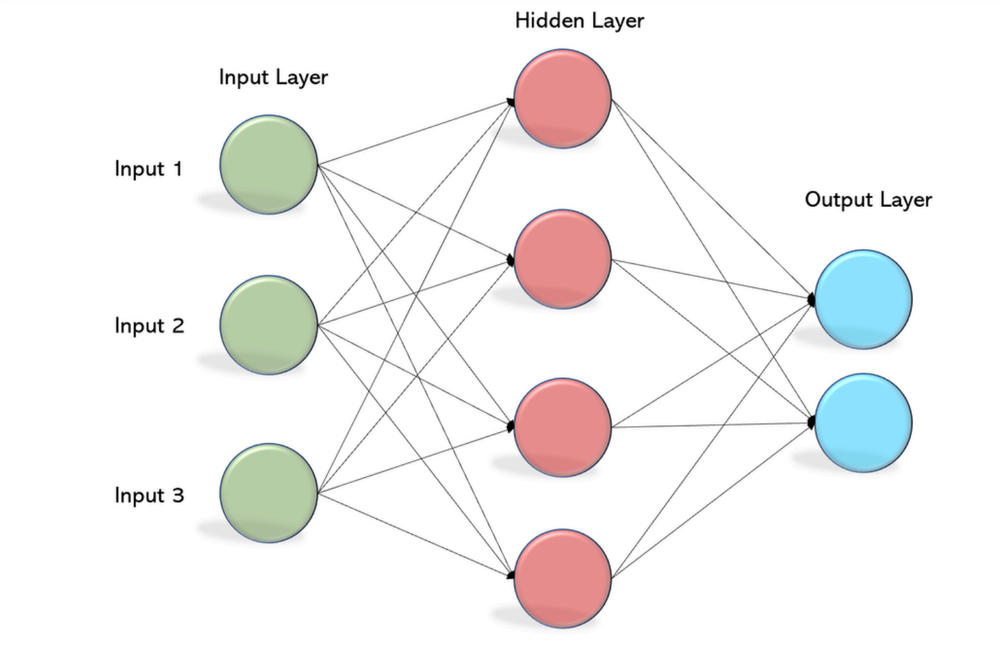


MultiLayer Perception

MLP

Stock Market **Analysis**

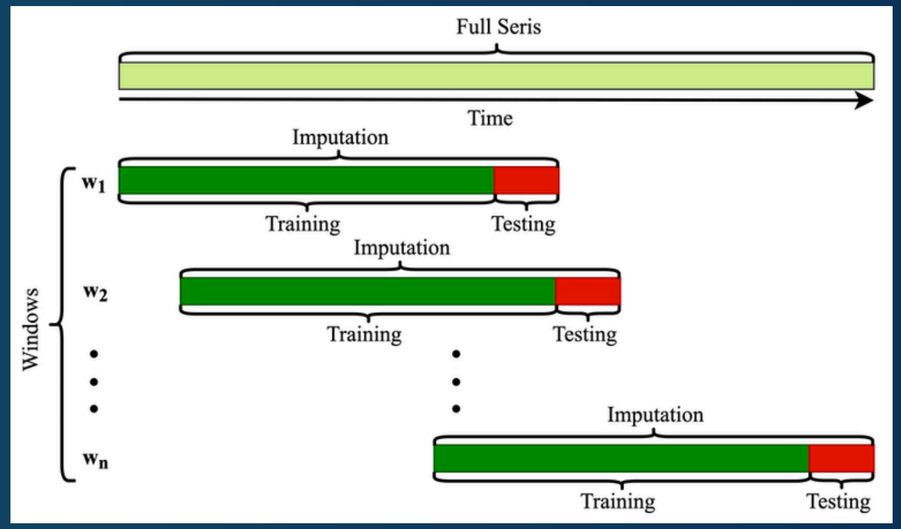
Why MLP



Fundamental type of feedforward Neural network

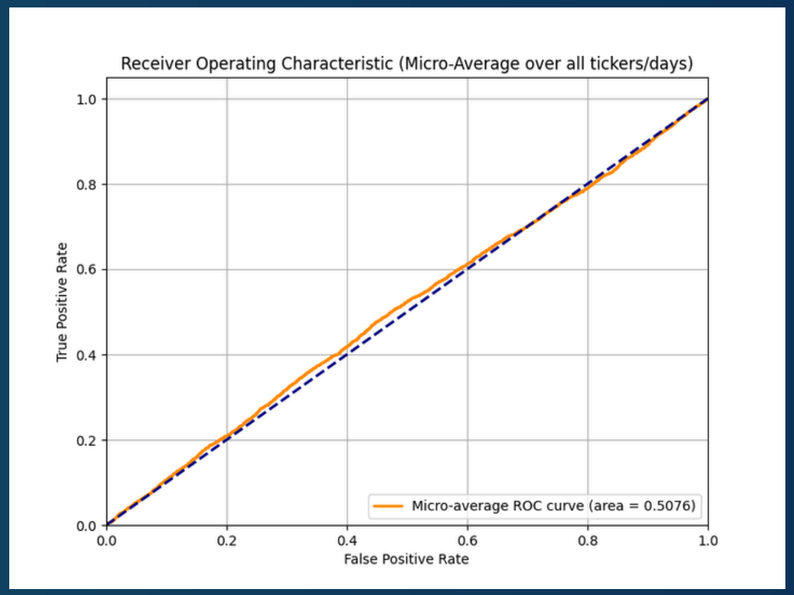
- Non-linearity
- Universal Approximator
- The hidden layers learn the intermediate features from the input data.

Basic Implementation



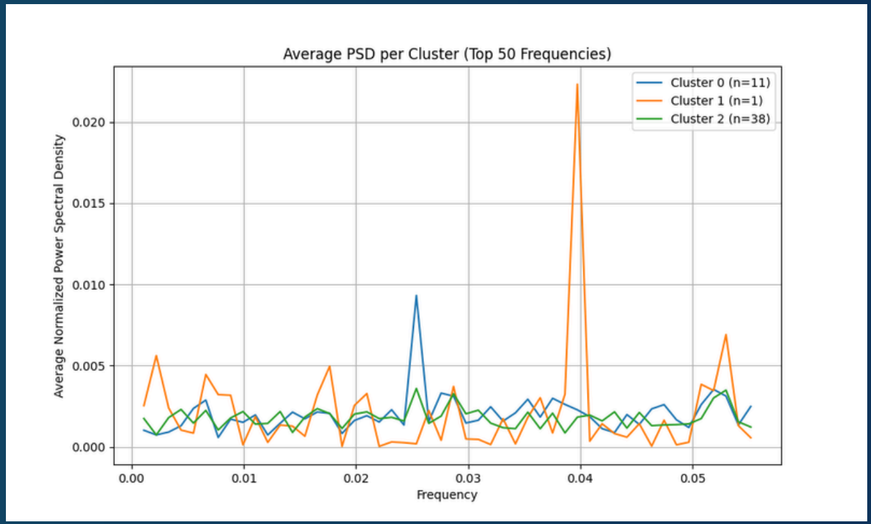
The Rolling Window Approach (size = 10)

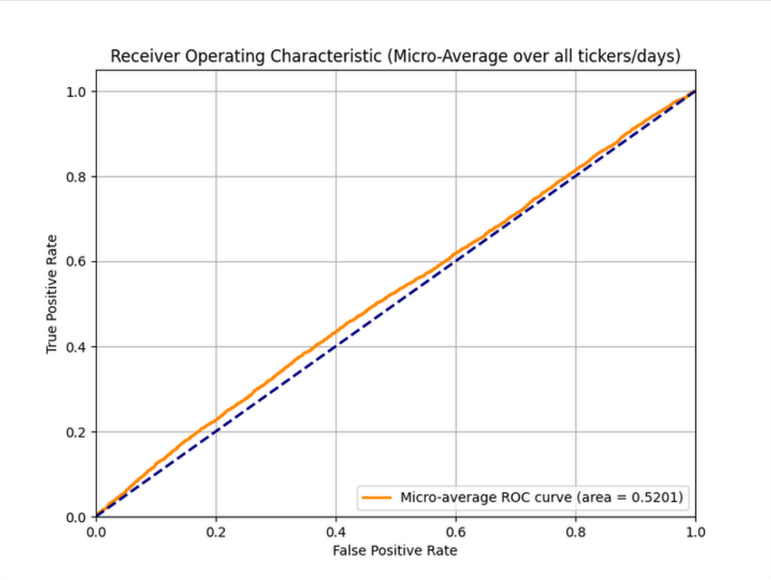
- Define a window_size (e.g., 10 days).
- Take the features from the past window_size days.
- **Flatten** these features into a single input vector for the MLP.
- The target is the Open and Close price of the next day (day window_size + 1).
- Slide this window one day forward in time and repeat.



Overall Test Accuracy: 0.5118
Overall Test F1 Score (Weighted): 0.5121
Overall Test ROC AUC Score (Weighted): 0.5155

MLP Optimisations





Fast Fourier Transform Clustering

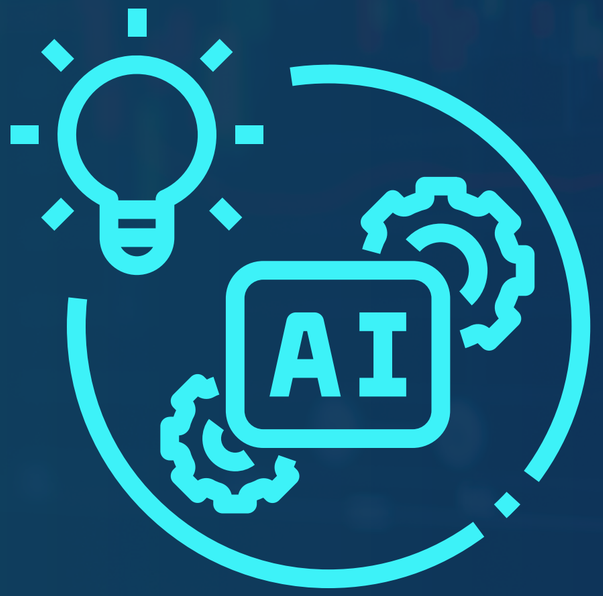
- Capture the frequency component of stock
- Cluster the stocks into groups with similar frequency component
- Determine best number of cluster using Silhouette Score
- Train different groups separately .

Overall Test Accuracy: 0.5130
Overall Test F1 Score (Weighted): 0.5072
Overall Test ROC AUC Score (Weighted): 0.5201

MLP Performance and Anslysis

- Complex and Noisy Data:
  - Stock prices are influenced by numerous unpredictable factors (e.g., news, sentiment, geopolitics).
  - MLPs struggle to capture these non-linear, noisy relationships effectively.
- Overfitting Risk:
  - MLPs' high flexibility can lead to overfitting historical patterns, reducing generalization to future data.
- Non-Stationarity:
  - Stock data's statistical properties change over time, challenging MLPs' assumption of stable patterns.
- Temporal Dependencies:
  - MLPs lack inherent mechanisms to model sequential or long-term dependencies, unlike RNNs or LSTMs.
- Feature Engineering Dependency:
  - MLPs require extensive feature engineering, which may not fully capture market dynamics.

Group 24



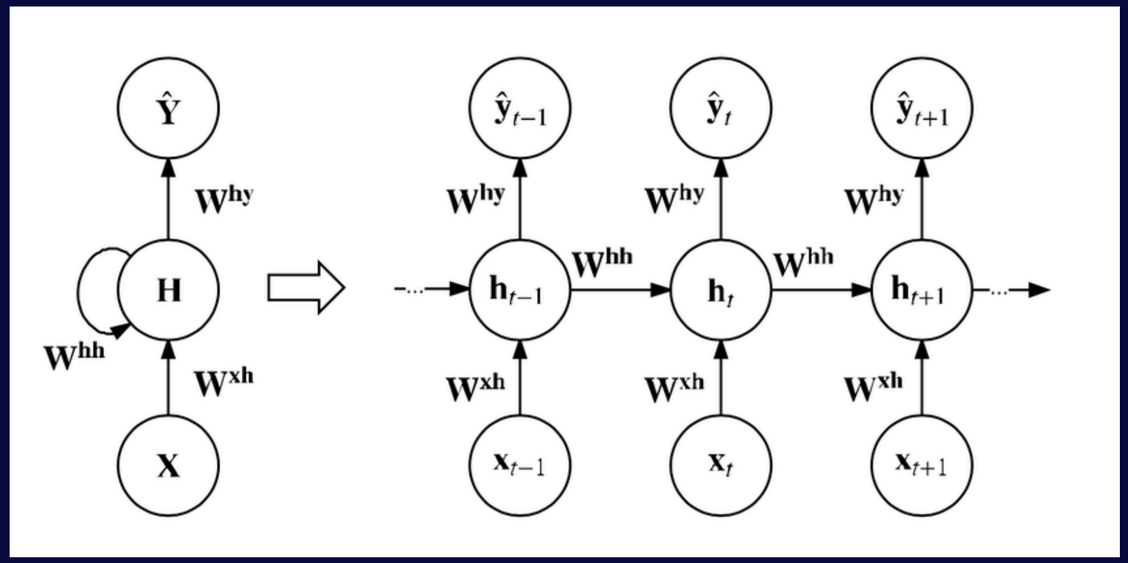
Recurrent Neural Network

RNN

Stock Market **Analysis**

Group 24

RNN



- Neural network that “remembers” previous inputs
- Learn long-range dependencies
  - “unroll” network across time
  - Apply backpropagation to compute gradients
- Supports Multivariate Time Series inputs
  - Feed entire feature vector
  - Learn interactions between our various features (open, high, low etc)

RNN Optimizations

Multiple optimizations were investigated with keras autotuner to find the optimal RNN model:

1. Single layer basic RNN
2. Deep RNN
3. RNN with more dense layers (Best Result)
4. RNN with gradient clipping
5. RNN with layer normalization
6. RNN on individual stocks

Group 24

RNN Analysis

Key Performance Metrics (Best)	
Accuracy	0.5165
F1 Score	0.6706
AUC-ROC	0.5098

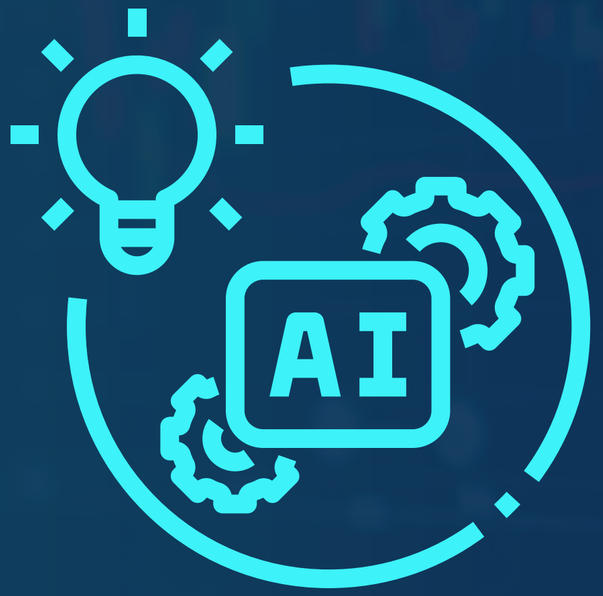
```
Confusion Matrix:
                Predicted Down  Predicted Up
Actual Down                369          6918
Actual Up                  373          7420
```

- Terrible accuracy and AUC-ROC → almost random predictions even after various optimization attempts
- Poor F1 scores: recall > precision, some attempts result in predictions being heavily weighted towards 1 class
- Earlier inputs are lost in RNN, information gets scattered the further we get in a sequence
  - Vanishing/exploding gradient problem
  - Normalization and gradient clipping can only do so much

Remarks & Improvements

- Poor accuracy might be due to lack of informative features → can easily input more features so it can focus on learning rather than inferring these features (e.g. momentum indicators)
- Have a way to retain past inputs more effectively → LSTM

Group 24



Long Short-Term Memory

LSTM

Stock Market **Analysis**

Group 24

Why LSTM?

Long Short-Term Memory

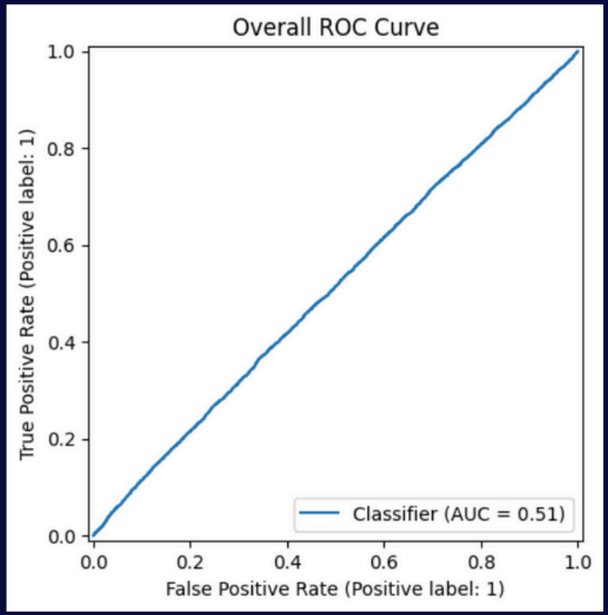
- Handles Vanishing Gradient Problem that RNN has
- Captures Temporal Dependencies (automatically learns time dependent structures, hence can capture patterns in the financial market such as lags)
- Supports Multivariate Time Series (e.g. interactions between “Open”, “High”, “Low”, “log_returns”)
- Effective for Sequential Labeling

Group 24

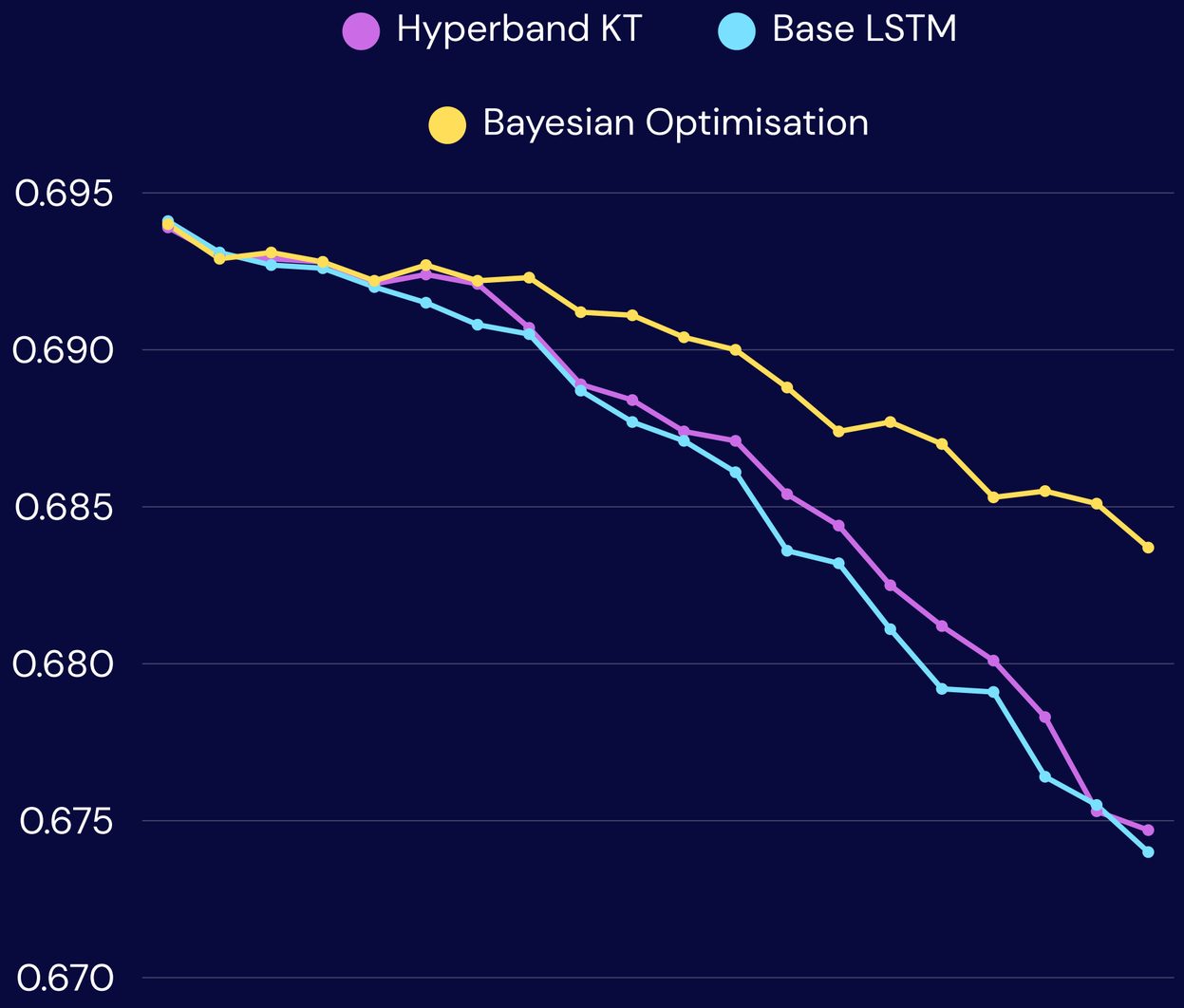
Types of LSTM Tested

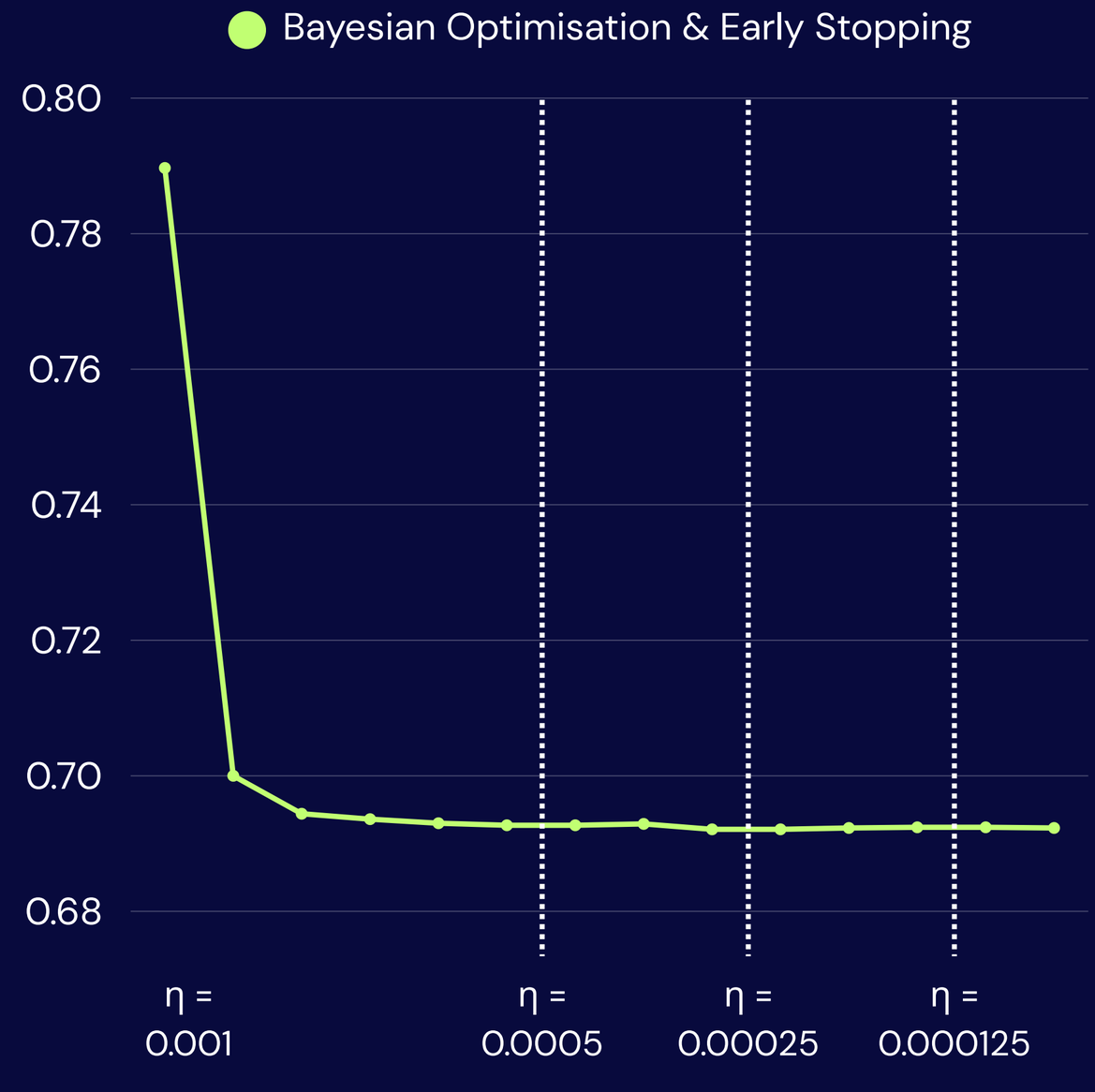
1. Base LSTM
2. LSTM with Rolling Window & Hyperhand Keras Tuning
3. LSTM with Rolling Window & Bayesian Optimisation Keras Tuning
4. LSTM with Rolling Window & Bayesian Optimisation + Early Stopping through Dynamic Learning Rate Reduction
5. Training with Individual Stocks with Rolling Window

```
===== OVERALL EVALUATION =====
Accuracy: 0.5119
F1 Score: 0.3550
AUC-ROC: 0.5119

Confusion Matrix:
             Predicted Down  Predicted Up
Actual Down            5531          1866
Actual Up              5285          1968
```



Training Loss





Group 24

Why LSTM might not be suitable?

Market Noise & Non-Stationarity

Stock prices are influenced by unpredictable events; patterns change over time.

Insufficient Features

Using only basic price data misses out on key signals like technical indicators or sentiment.

Binary Target Oversimplifies

Direction labels (up/down) may flip due to minor, random fluctuations, adding noise.

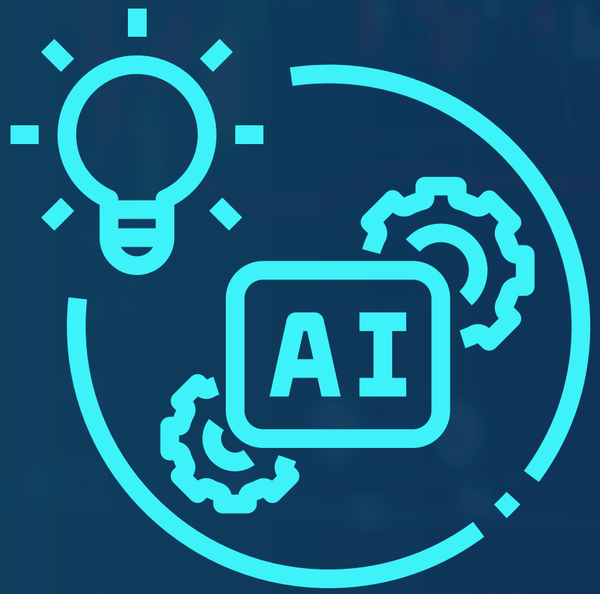
Overfitting to Historical Noise

LSTM may memorize irrelevant patterns that don’t generalize to future data.

LSTM Limitations

Slow training, no attention mechanism, struggles with very long-term dependencies

Group 24



XGBoost

Stock Market **Analysis**

eXtreme Gradient Boosting

- Gradient-Boosted Trees
  - Uses second-order Hessian approximation for better splits and weight updates (compared to gradient descent)

- Automatically learn complex interactions (e.g., between price movements and volume) that a linear model cannot

- Built-in regularization and early-stopping mechanisms help prevent overfitting on volatile financial series

Variations

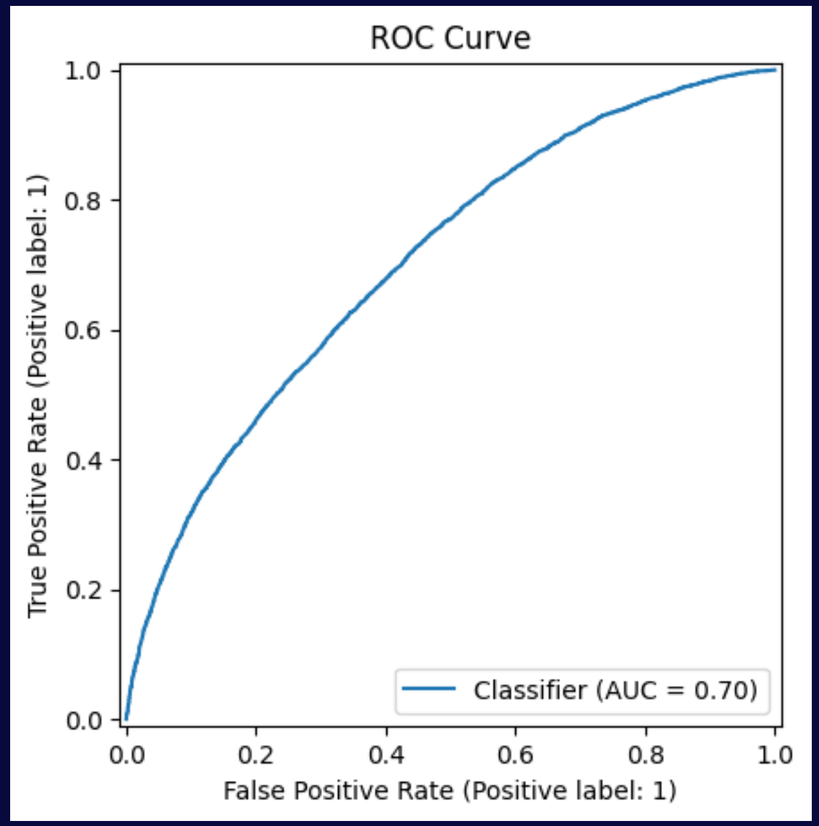
- Base XGBoost
- Base XGBoost using RandomizedSearchCV
- Base XGBoost using RandomizedSearchCV + GridSearchCV

- XGBoost with Rolling Window
- XGBoost with Rolling Window using RandomizedSearchCV
- XGBoost with Rolling Window on Individual Stocks

Best Result

Base XGBoost using RandomizedSearchCV + GridSearchCV

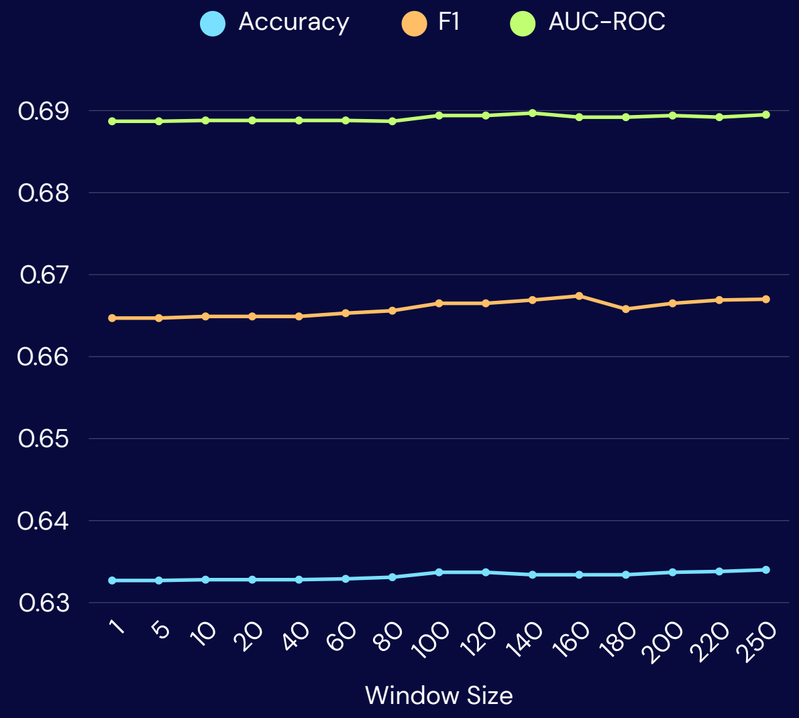
```
Accuracy: 0.6419
F1 Score: 0.6734
AUC-ROC Score: 0.7047

Confusion Matrix:
              Predicted Down  Predicted Up
Actual Down             4114          3216
Actual Up               2188          5572
```



Group 24

Non-rolling window outperformed rolling window

- Fewer total observations at each retraining than the non-rolling approach
- Parameter estimates remain noisier and more sensitive to the idiosyncrasies of any given quarter
- A rolling window discards older data that may still carry signal so it can’t fully leverage long-term structural patterns



Base Model Outperformed XGBoost

- Only four raw features (Open, High, Low, Volume) were used
- Logistic regression was less prone to overfit on the limited feature set
- XGBoost’s flexibility and capacity to model complex interactions became a liability instead
- As a result, the simpler logistic model generalized better on unseen, time-ordered data

Other Limitations

Non-Stationarity & Concept Drift

Static tree ensembles, once trained, won't automatically adapt to shifts in market patterns

Binary-Target Oversimplification

Framing "up/down" as a two-class problem can amplify label noise

Limited Extrapolation Beyond Observed Ranges

Tree-based methods interpolate within the data they've seen; they cannot predict extreme moves outside the historical feature space.

Future Improvements

Improved Feature Engineering

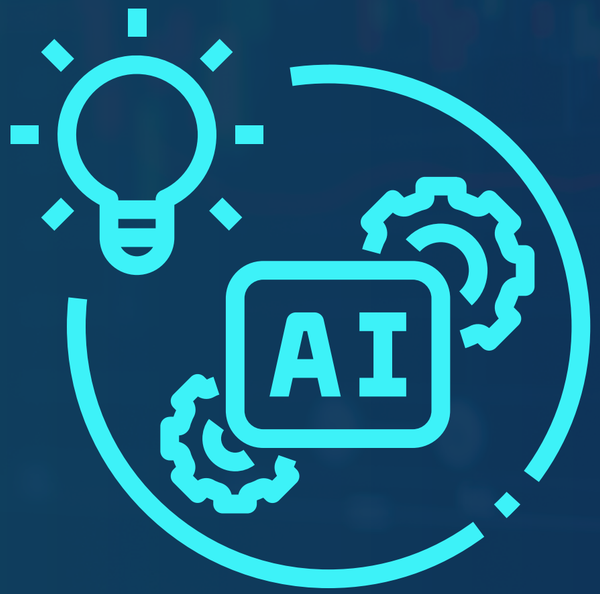
Additional features such as momentum indicators, volatility measures, or cross-asset signals.

Incorporating external macroeconomic or sentiment data.

Refining the rolling-window scheme

Optimise window length, overlap, or retraining frequency via cross-validation

Group 24



Comparison of Results

Stock Market **Analysis**

Model Performances

	Accuracy	F1 Score	AUC-ROC
Logistic Regression	**0.7647**	**0.7777**	**0.8280**
MLP	0.5130	0.5072	0.5201
RNN	0.5165	0.6706	0.5098
LSTM	0.5119	0.3550	0.5119
XGBoost	0.6419	0.6734	0.7047

Thank **You!**