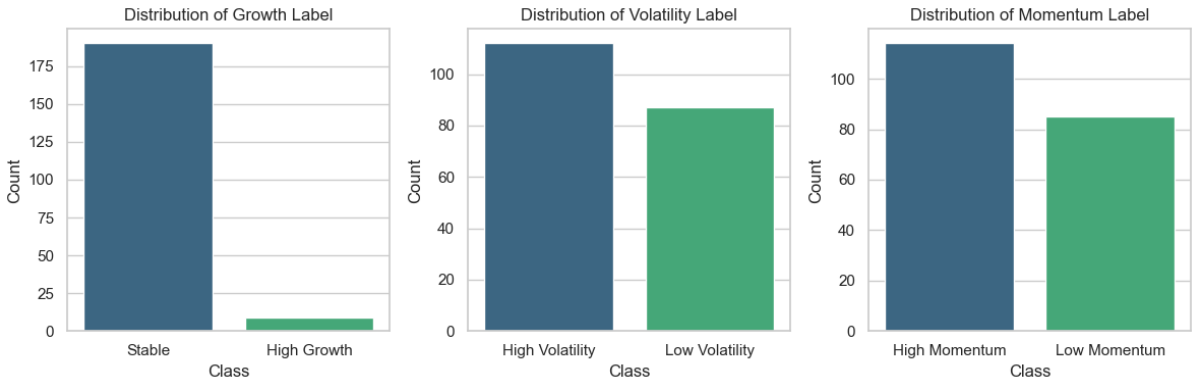
MULTI LABEL STOCK CLASSIFICATION

I have selected stock data from 200 companies, focusing on computing financial features such as Return, Volatility, Momentum, and Moving Averages. Each stock is classified into categories like High Growth, Stable, Low Growth, High Volatility, Low Volatility, High Momentum, and Low Momentum based on its performance over the latest 3 months, using data sourced from Yahoo Finance.

The goal of using recent 3-month data is to capture the most current market trends.

This project combines my interest in applying data science and machine learning to finance with the company's mission as a data services provider. And also I see the company's objective is to generate actionable insights for the venture capitalists and private-sector stakeholders. The project content consists of four files.

- ETL.ipynb : This file handles data scraping, transformation, and preprocessing. The final cleaned dataset is saved for further analysis.
- EDA.ipynb : This file performs initial exploratory data analysis, including data visualization and balancing. To address class imbalance, additional weight is assigned to the minority class during training.



- multilabel_classification_ml_models.py : This script builds traditional probabilistic machine learning models for multilabel classification. Each model is configured using MultiOutputClassifier, enabling it to handle multiple labels simultaneously. For each target label, separate models (e.g., Logistic Regression) are trained to output probabilities for classes 0

and 1. A threshold of 0.5 is applied to these probabilities to generate binary predictions.

- multilabel_classification_dnn.py : This script applies a deep neural network (DNN) to the same dataset for multilabel classification. The DNN's final layer outputs a probability matrix (e.g., 0.8 for Growth = 1, 0.3 for Volatility = 1). The model then calculates its error by comparing predicted probabilities with actual labels. Like the ML models, a threshold of 0.5 is applied to convert these probabilities into binary predictions for each label.

Model Performance: The following metric are used for model performance.

Model	F1 Score	Hamming Loss	Subest Accuracy
Logistic Regression	0	0.625	0
Naïve Bayes	0.78	0.32	0.25
Random Forest	1	0	1
XgBoost	1	0	1
Deep Neural Network	0.69	0.29	0.3

Note: Model performance is currently limited due to the small dataset size and the relatively few features used. Efforts to pull a larger dataset encountered some issues.

Future Work : To improve model performance, additional stock data and more features will be incorporated. Integrating text data from news articles, social media, and financial threads could add valuable context to the dataset. Additionally, further model parameter tuning will be explored to optimize performance.

Reference : The following tool or sources were considered for doing this project ChatGPT, Gemini and Google Search.