Applied Data Analysis – Assessment 1

Group id: A1Group_12

Members: Nasser Alshuhayli, Vallotto Nicolo

Date: 22/09/2025

Introduction

This report describes the construction of a model capable of forecasting future performance of a portfolio of 650 stock returns against the Monash Index. We are aiming to get both qualitative and quantitative predictions:

- **Classification task:** predicting whether stocks will underperform or overperform compared to the Monash index in July 2023
- **Regression task:** predicting the excess return of the stock when compared to the index for July 2023

NB: It is important to note that since we are trying to predict the performance of the stock index in 2023, taking into consideration the months prior, we will be applying a **lag** to our features. What this means is that we will be using the features from month 2020_01 to predict the labels for month 2020_02 and so forth.

Data Exploration

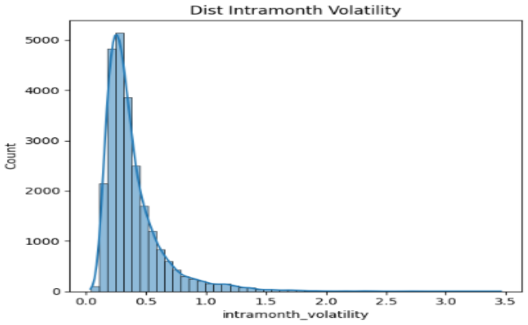
As a first step in the analysis, all the data files were read and merged into 1.

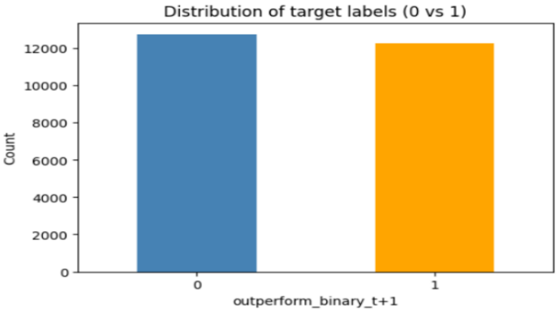
For numerical features, statistics such as standard deviation, mean and median have been computed. For categorical features, instances and number of categories have been computed. The dataset includes 38 different features and 25618 rows.

In total, 135 rows contain missing values in the dataset. Most of them come from rolling computations like return_1m and return_3m.

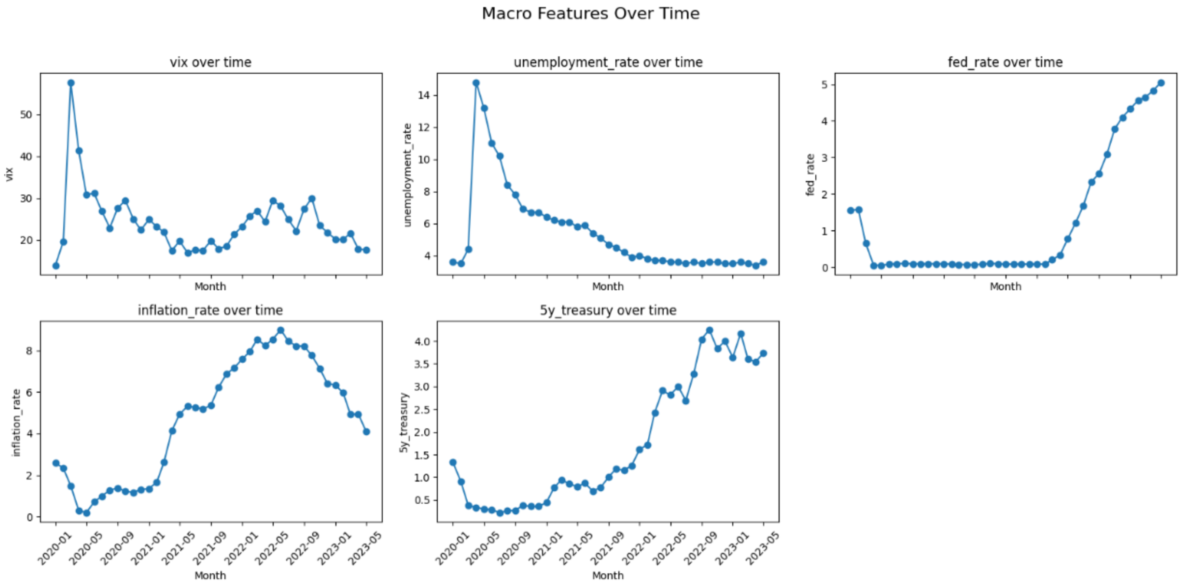
Each feature's distribution was plotted individually, and we observed the following:

- Most of the features show heavily skewed distributions.
- Returns are skewed but can also be negative. This is notable as they cannot be log-transformed for adjustments
- Distribution of the classification labels shows that the classes are balanced



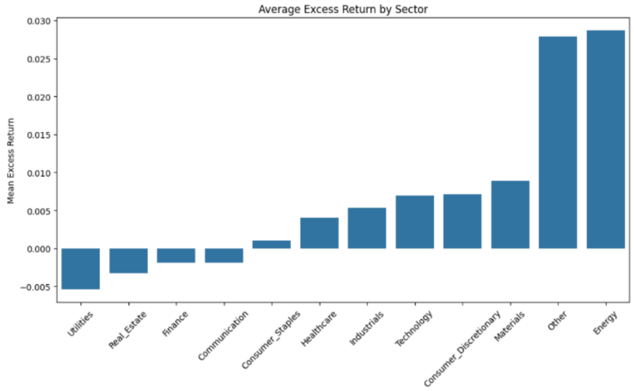


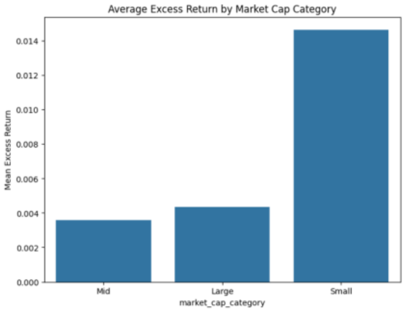
We also explored the variation of macro variables across the years, to possibly detect different economic regimes that have been present throughout the years Notable findings:



- VIX and unemployment spiked in 2020, indicating a crisis regime, characterised by lower returns and volatility spikes
- 2021 appears to be a recovery regime, where unemployment decreased and Vix went back down
- Between 2022 and 2023, inflation, federal rates and bond rates grew, indicating a tightening regime

Furthermore, we observed that different sectors and market caps have a strong influence in determining returns, as they show considerably different amounts of average excess returns.

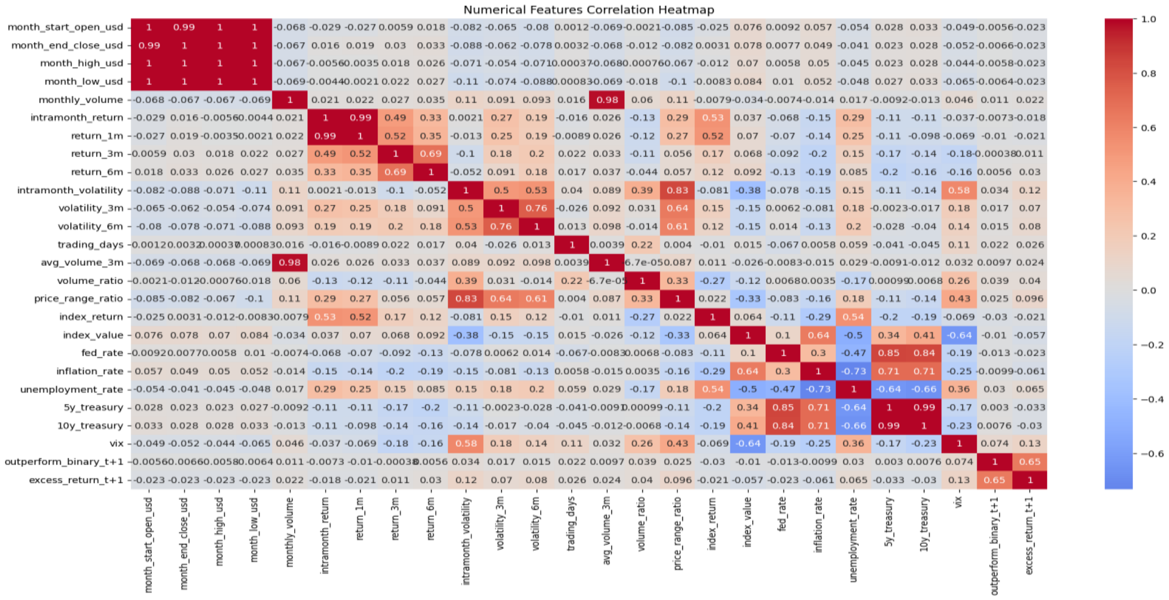




Correlation analysis

In order to understand the relationships between different variables, we computed the correlation matrix of the full data. Looking at the results, we may observe the following:

- Some variables show very high correlations between each other. To avoid risking multicollinearity problems, it is important to detect them.
- We do not observe very strong correlations between predictors and targets, which suggests non- linear relationships between predictors and targets are present.



Classification Modelling

Dataset Filtering

Looking at the plots for macro-features presented during the EDA, we observed an unusual spike in both unemployment and vix for most part of the year 2020. This is probably connected to the Covid-19 phenomenon. We made the decision to **not include any data before October 2020**, as it presents a very different regime compared to the rest of the dataset.

Feature engineering

The following features were added/removed :

- **Stock specific features:** clv, log_dollar_vol were added. These features are aimed at capturing liquidity and stock price information.
- **Macro financial features:** term_spread, real_rate and bond_inflation_spread: give additional context to the optional macro features provided. These features proved to cause the model to overfit on the first few months, hurting generalization, so we did not end up using them in the final model.
- **Dropped features :** Features such as month_high_usd and month_low usd, aswell as monthly_volume and avg_volume_3m showed very high correlations, so only 1 of them were kept to avoid collinearity. Returns and volatility over different timeframes have been kept to account for short, long and medium range trends.
- **Cyclical / time-based features:** Initially, time features such as month, year, seasonality had been used to capture time-based shifts. These features have then been removed from the final model, as they hurt generalization.

Transformations and handling of anomalies

Handling outliers turned out to be a significant challenge for this task. While the initial goal was to detect and remove them entirely, their large number meant that important information was being lost in the process. Instead, a **winsorization** approach was applied, capping extreme values at the 1st and 99th percentiles to preserve data integrity while reducing the impact of outliers. **Missing values** have been handled by median imputation.

We followed the following preprocessing pipeline for different kind of features:

- **Skewed features** like volatility, volume and month_high have been median imputed, winsorised, log-transformed and standardised
- **Returns** are median imputed, winsorised, they are transformed through
- **Categorical features** are One-Hot Encoded, while **Ordinal features** like market_cap have been turned into ordinal variables.
- All other numerical values are only standardized

Models

Three different models have been applied and tuned:

- Boost
- Logistic Regression
- Support vector classification

The Baseline has been estimated to be around 0.50 of an F1 score. This is equivalent to random guessing, so any score beyond that will be considered an improvement and validate the usefulness of our model.

Tuning methodology for model selection

When deciding which model to choose, rigorous standards have been applied to correctly assess the model performance and avoid overfitting:

- **Time Aware CV:** When dealing with time-related data, it is important to create the splits in a way that ensures the model doesn't "peek into the future" for its predictions, as this would cause higher bias/overfitting. Using Time Series Cross Validation ensures that the validation data is always coming from a "future" split of the data. CV itself is necessary as it ensures we are not looking at an overly optimistic split in our data. In this case, we divided the data across 5 different splits (data is not much, so doing 10 splits seemed too unstable).
- **Random Grid Selection:** Each candidate model has been tuned by making use of a random grid. Random grids ensure that the time it takes to tune the model is not exaggerate, while also exploring a wide range of parameters. Each parameter combination was cross-validated, and the results were explored further to make sure no anomalies occurred. For example, if a split had an high F1 score, but also showed high standard deviation across splits, it was deemed too unstable and helped us further understand model quality.

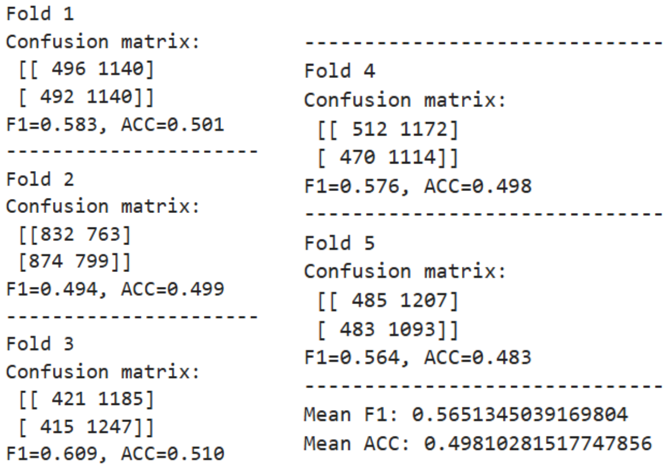
Models and performances

The first model to be explored was XGBoost, because of its robustness to poor feature engineering and scaling issues. A well-tuned XGBoost is often capable of gathering both linear interactions and non in the dataset. This served as proof that our preprocessing was sound. Two models were fit after XGBoost being Logistic Regression and SVC. Since both of these models are very sensible to different scales and distances, as well as requiring normal distribution of features in the case of Logistic Regression, standardisation was essential for a better performance of the models.

The cross-validated mean F1 scores obtained for each model are as follow:

- **XGBoost: F1 score of 0.501** with standard deviation of 0.03. This is barely above a random prediction, so XGBoost wasn't found to have a good performance
- **Linear Regression:** This model has shown to be a bit inconsistent across different tunings. For lower values of C, the model sometimes degenerates and defaults to predicting only positives/negatives. With strong regularisation, the model shows a stronger performance. Using l1=1 in elastic net (lasso) and a value of C=1000, logistic regression was able to reach an **F1 score of 0.51**. While this is just a small improvement over random guessing, it is still significant and proves that our features provide some predicting capabilities
- **Support vector classifier:** This is the model that has performed the best. Different Kernels have been explored such as linear or rbf. The linear kernel has shown to be superior, able to reach an **F1 score of 0.57** at its best. The parameters used to get this performance were a strong regularization of **C=500**, **linear Kernel** and **balanced class weights**. To make sure this value was real, and not the result of chance or a majority class prediction, we computed the confusion matrixes across folds, which can be seen down below.

  As we can observe, the F1 scores are consistent across folds, and the predictions do not favour one class over the other. This model has been selected as the best performing model to make the final predictions.



```
Fold 1
Confusion matrix:
 [[ 496 1140]
 [ 492 1140]]
F1=0.583, ACC=0.501
----------------------
Fold 2
Confusion matrix:
 [[832 763]
 [874 799]]
F1=0.494, ACC=0.499
----------------------
Fold 3
Confusion matrix:
 [[ 421 1185]
 [ 415 1247]]
F1=0.609, ACC=0.510
------------------------------
Fold 4
Confusion matrix:
 [[ 512 1172]
 [ 470 1114]]
F1=0.576, ACC=0.498
------------------------------
Fold 5
Confusion matrix:
 [[ 485 1207]
 [ 483 1093]]
F1=0.564, ACC=0.483
------------------------------
Mean F1: 0.5651345039169804
Mean ACC: 0.49810281517747856
```

Regression Modelling

For the regression modelling, we utilized the same best performing model that gave us the best predictions for the classification problem. The chosen model was the Support Vectors Regression model. Once again, the model was implemented using time-aware cross validation and testing both linear and rbf kernels. The best performing model has reached an **RMSE of 0.095**, with weak regularization of **C = 0.0012** and a **linear kernel.**