Final Project: Statistical Data Analysis of Jane Street Market Prediction

December 8th, 2024

Author Contributions

Partner 1 and 2 worked on all parts simultaneously and contributed even work to the assignment through paired-programming.

1. Introduction

In this project, we analyze the dataset provided by the Jane Street Market Prediction competition on Kaggle. Each row in the dataset symbolizes a unique trading opportunity, providing critical information about the market at a specific moment.

The dataset's most important assets are its anonymized market features, the key response variable resp, and the weight variable: weight and resp together quantify the overall return associated with the trading opportunity represented by each row. This value is pivotal for evaluating the profitability of trades. Additionally, the dataset contains several related response variables, resp_1, resp_2, resp_3, and resp_4, which represent returns calculated over different time horizons where the larger the number the longer the time period is respectively. These variables provide a nuanced view of how the market evolves and how profitability might differ depending on the holding period of the trade.

Further enriching the dataset are the temporal indicators. The date column, stored as an integer, represents the day on which the trading opportunity occurred, providing a chronological framework for market behavior analysis. Complementing this is the ts_id column, which serves as a unique identifier for the time ordering of trades within each day. Together, these columns allow for a detailed exploration of time-series patterns, such as intraday trading dynamics and market trends over time.

By delving into this dataset, our goal is to explore the relationships between these features and the returns, evaluate trade profitability, and understand the dynamics of market behavior over time to hopefully be used when creating a predictive model that can predict resp.

We focus on answering the following key questions:

1. **What are the correlations between the features and the target responses? Are there patterns or redundancies in these correlations that indiciate dependencies or noise?**
2. **How does trade weight impact overall profitability? Are there specific ranges of weight that consistently lead to higher returns?**
3. **What are the point estimates (mean) and confidence intervals for the resp values across different time horizons (resp_1, resp_2, . . . , resp_4)? Are there statistically significant differences in expected returns across these horizons, and how consistent are these differences?**
4. **Are there specific trading days that exhibit higher or lower overall returns? Can trends or seasonality be identified over periods of time?**

We will also conduct an advanced anaylsis section were we aim to combine what we have learned in the our 4 basic analyses sections to further our investigation on a deeper level.

Before starting our analysis, we removed all rows with missing (NA) values. This decision was made to ensure that our model is trained on complete data, as imputing missing values could introduce inaccuracies, especially given the complex relationships inherent in financial datasets. By excluding incomplete rows, we enhance the integrity of our dataset and avoid potential biases caused by imputed data.

After cleaning the data, we randomly shuffled the train.csv dataset and selected 100,000 rows. The original dataset contained approximately 650 million rows, which was too computationally expensive to process with our available resources. By shuffling and randomly sampling the cleaned data, we minimize selection bias and ensure that our subset is representative of the broader population.

While the commonly referenced 10% sampling guideline suggests larger sample sizes, this rule is less relevant for datasets with extremely large populations, such as ours. Instead, under the Central Limit Theorem, a sufficiently large random sample—like our 100,000 rows—is adequate to approximate the properties of the full dataset and derive meaningful insights. This approach balances computational feasibility with statistical rigor while ensuring that our analyses are conducted on high-quality, complete data.

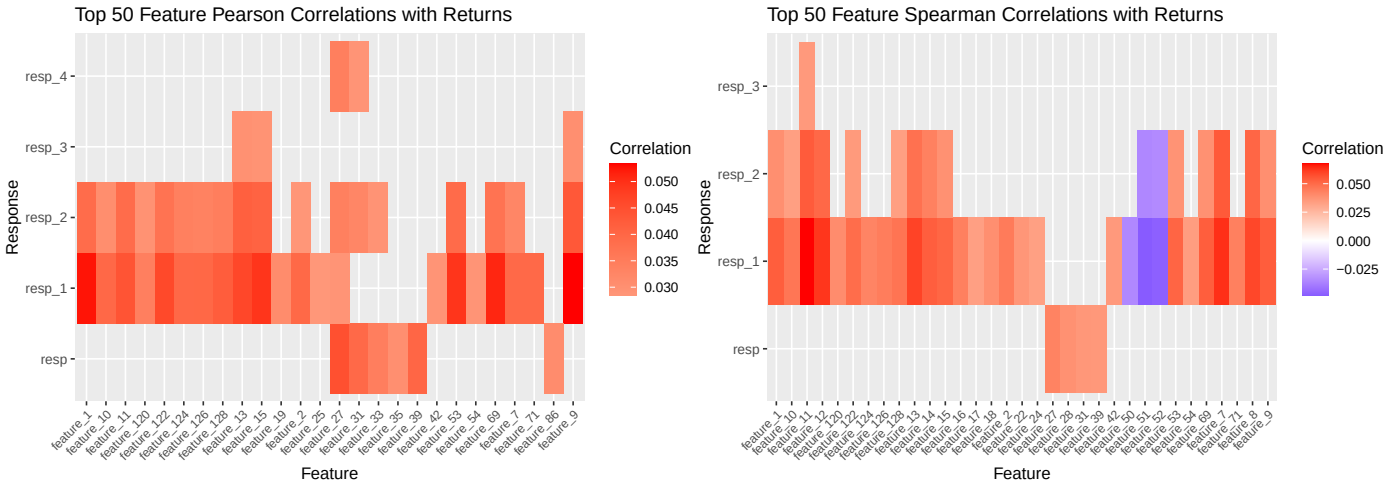
2. Analysis

2.1 - Feature Correlations

Methods

In this section, we examined the relationship between the anonymized features and the various response variables (`resp`,'resp_1',...,'resp_4') using three methods: Pearson correlation, Spearman correlation, and Random Forest Regresion. As the results of these methods provided little value, we pivoted in exploring colinearities between the features to extract any actionable insights.

Analysis




```
## [1] ""
```

```
##         Feature1   Feature2 Correlation
##           <char>     <char>       <num>
##   1:  feature_67 feature_68   0.9987588
##   2:  feature_62 feature_63   0.9973821
##   3:  feature_60 feature_61   0.9967653
##   4:  feature_65 feature_66   0.9959741
##   5: feature_113 feature_89   0.9924283
##  ---
## 140:  feature_73 feature_97  -0.8034003
```

```
## 141:  feature_18 feature_21   0.8030749
## 142:  feature_24 feature_25   0.8016457
## 143:  feature_75 feature_76   0.8015218
## 144: feature_110 feature_96   0.8004828
```

Conclusion

After creating a heatmap of Pearson correlations between the anonymized features and the resp variables, we observed that the relationships between these variables are not linear. The strongest Pearson correlation was only 0.05, indicating an extremely weak linear association. Despite the weak correlations, certain features, such as 27, 31, 33, 35, and 39, showed stronger relationships with resp compared to others. This suggests that specific subsets of features may be more relevant for modeling, even though their individual contributions remain limited. Such low correlation values overall suggest that the relationships between the features and the response variables are likely non-linear in nature.

To explore potential non-linear monotonic relationships, we pivoted to Spearman correlations. A similar process to that of Pearson correlations was followed: a correlation matrix was computed, and the top 50 features were visualized; keep in mind that the heatmap does not display resp_4 because the correlations involving resp_4 did not rank among the top 50 strongest correlations (by absolute value). This indicates that the features do not exhibit a strong monotonic relationship with resp_4. However, this approach also revealed weak relationships, with the strongest correlation being 0.05 and the lowest being -0.025. While this did not provide significant insights, it reinforced the need to explore more complex relationships beyond monotonicity.

Recognizing the limitations of correlation-based methods, we attempted to use a Random Forests model to detect both linear and non-linear interactions between the features and the response variables. Random Forests also provide feature importance metrics, which would have been valuable for guiding feature selection and identifying the most impactful predictors. However, the computational expense of training the Random Forest on 100,000 rows and a large number of features made this approach infeasible, prompting us to abandon it. Had computational resources permitted, Random Forests could have provided insights into the relative importance of features and detected interactions missed by simpler methods.

With individual feature-response relationships yielding limited insights, we shifted focus to examining collinearities among features themselves. A feature-to-feature correlation matrix was computed, and feature pairs with high absolute correlation ($|\text{Correlation}| > 0.8$) were identified. To avoid redundancies, we retained only unique pairs by ensuring that Feature1 appeared lexicographically before Feature2. This analysis identified redundant features and provided a better understanding of collinearities within the dataset. This is critical for feature selection, as it helps identify redundant features that can inflate variance in predictive models. By removing or combining highly correlated features, we can improve the interpretability and efficiency of future models. We plan on leveraging this information when building our final predictive model.

2.2 - Trade Weight and Profitability

Methods

To understand how trade weight influences profitability, we binned the trade weights into quintiles. This allowed us to categorize the trades into groups with different weight ranges. We then calculated the average profitability (resp) within each weight bin to assess whether heavier trades are associated with higher or lower profitability.

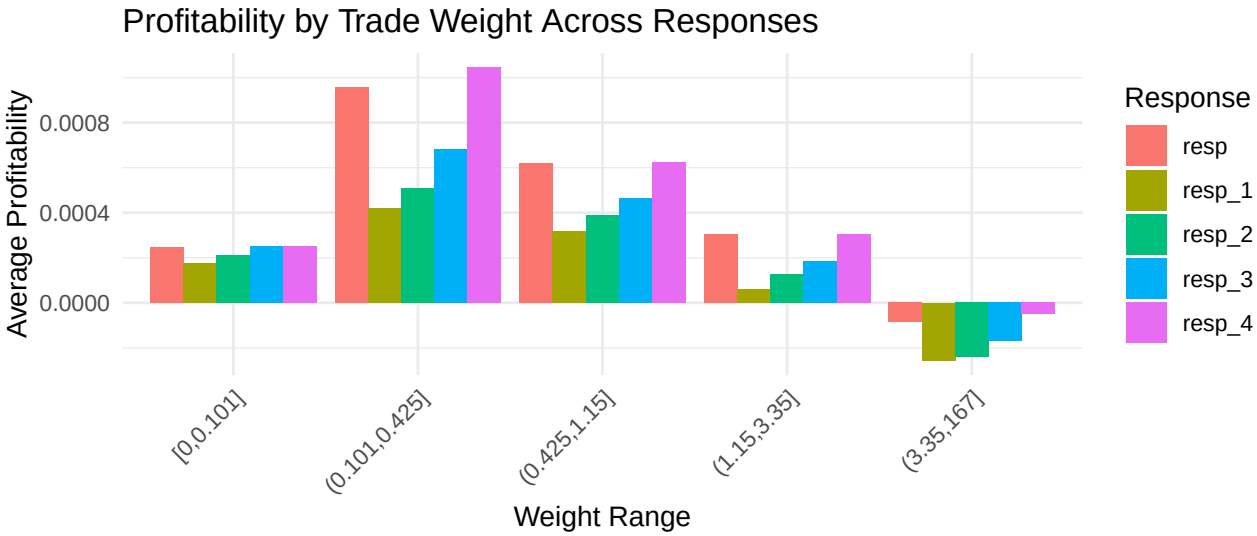
The bins were created based on quintiles of the weight variable, ensuring that each bin contained roughly the same number of trades.

Unlike other computationally intensive analyses in this study, this approach was computationally feasible to run on the entire dataset,enabling us to leverage all available data for a more robust analysis.

After categorizing the data into weight bins, we computed the mean profitability for each group and visualized the results in a bar chart. This approach provided a clear view of trends in profitability across different trade

weights.

Analysis



The bar chart of profitability by trade weight revealed the following insights:

Lower-weight trades, particularly those in the range [0, 0.101], consistently demonstrate positive profitability across all responses, suggesting that smaller trades may benefit from reduced market impact and better pricing opportunities.

Trades in the mid-range (0.425, 1.15] exhibit the highest profitability, indicating this range could represent an optimal trade size that balances capturing market inefficiencies with minimizing transaction costs.

In contrast, trades in the highest weight range (3.35, 167] consistently result in negative returns, likely due to increased market impact, slippage, and reduced flexibility.

Conclusion

The analysis of trade weight and profitability reveals significant insights into how weight impacts returns across the response variables (resp, resp_1, resp_2, resp_3, resp_4).

These findings provide actionable insights for predictive modeling and trading strategies. Trade weight should be included as a critical feature in predictive models, with emphasis on segmenting weight ranges to capture the non-linear relationships observed. Features that highlight the mid-range weight segment (0.425, 1.15] could help models identify trades with the highest potential for profitability. Conversely, high-weight trades should either be avoided or adjusted to mitigate their negative impact on returns. By leveraging these insights, predictive models can better capture the dynamics between trade weight and profitability, leading to more informed and effective trading strategies.

2.3 - Return Behavior Across Time Horizons

Methods

To address the question of point estimates (mean) and confidence intervals for the resp values across different time horizons (resp, resp_1, resp_2, resp_3, resp_4), we computed the mean and confidence intervals for each response variable using a standard statistical approach. The confidence intervals for each response variable were calculated using the formula:

$$\text{CI} = \text{mean} \pm t_{\alpha/2} \cdot \frac{\text{SD}}{\sqrt{n}}$$

where:

- mean is the sample mean of the response variable
- $t_{\alpha/2}$ is the critical value from the t-distribution for a 95% confidence level
- SD is the standard deviation of the response variable
- n is the sample size

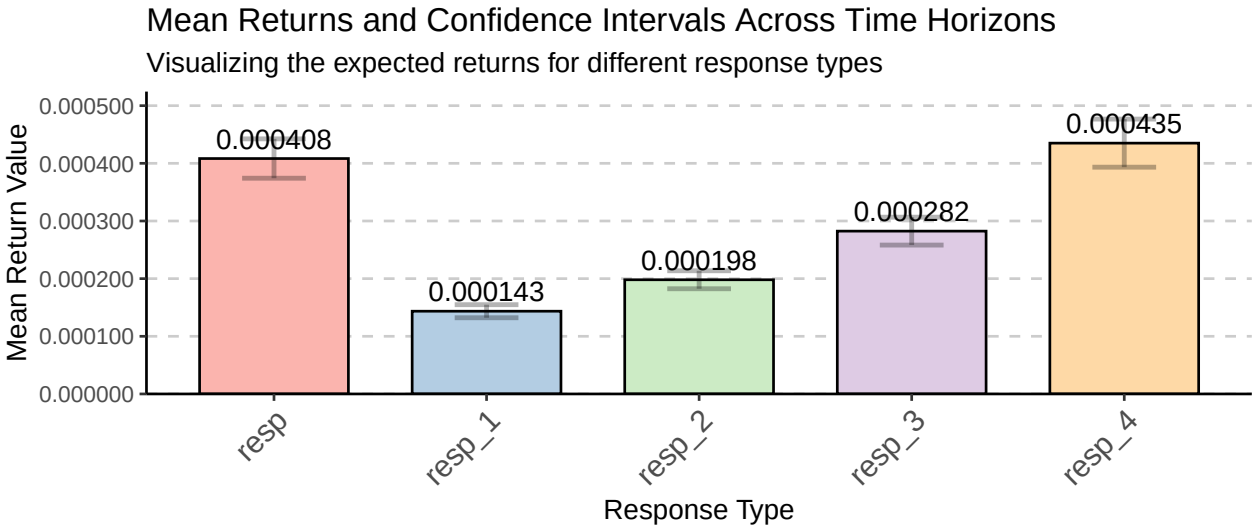
These calculations allow us to quantify the uncertainty around the mean estimates.

Unlike other computationally intensive analyses in this study, this approach was computationally feasible to run on the entire dataset, enabling us to leverage all available data for a more robust analysis.

To visualize the results, we constructed a bar chart showing the mean returns for each response variable, with error bars representing the confidence intervals. This approach highlights the relative magnitude of the expected returns across the different time horizons and visually assesses the overlap (or lack thereof) in confidence intervals, which provides an indication of whether differences are statistically significant.

Analysis

```
##   resp_type        mean margin_of_error    ci_lower    ci_upper
## 1      resp 0.000408311    0.0000341460 0.000374165 0.000442457
## 2    resp_1 0.000143497    0.0000113205 0.000132176 0.000154817
## 3    resp_2 0.000198075    0.0000155953 0.000182480 0.000213670
## 4    resp_3 0.000282418    0.0000241729 0.000258245 0.000306591
## 5    resp_4 0.000435020    0.0000417217 0.000393298 0.000476742
```



Conclusions

The analysis of point estimates (mean) and confidence intervals for the resp values across different time horizons (resp, resp_1, resp_2, resp_3, and resp_4) reveals several important trends and insights. Mean returns steadily increase from resp_1 (0.000143) to resp_4 (0.000435), indicating that profitability grows as the time horizon expands. Specifically, the mean returns are:

resp_1: 0.000143 (CI: 0.000132 to 0.000155)

resp_2: 0.000198 (CI: 0.000182 to 0.000214)

resp_3: 0.000282 (CI: 0.000258 to 0.000307)

resp_4: 0.000435 (CI: 0.000393 to 0.000477)

Additionally, the confidence intervals, represented by the error bars in the visualization, vary across response types. Shorter time horizons, such as resp_1 and resp_2, have the narrowest confidence intervals, suggesting more consistent and less variable returns. In contrast, longer time horizons like resp_3 and resp_4 exhibit wider confidence intervals, reflecting greater variability and higher risk associated with longer-term trades.

Statistically significant differences in expected returns are evident, as the confidence intervals for some response types, such as resp_1 and resp_4, do not overlap. This highlights clear distinctions in profitability across time horizons. Specifically, resp_1, representing the shortest horizon, has the lowest mean return and is significantly lower than the longer horizons like resp_3 and resp_4. Meanwhile, resp_4, the longest horizon, achieves the highest mean return, which significantly exceeds those of the shorter horizons such as resp_1 and resp_2.

These findings have several implications for predictive modeling:

- Incorporate Time Horizon as a Feature: The increasing mean returns across time horizons suggest that time horizon information (resp_1, resp_2, etc.) is a critical feature to include in the predictive model.
- Balance Risk and Reward: The greater variability in returns for longer horizons indicates that models must account for risk when making predictions. While longer horizons offer higher profitability, they also come with increased uncertainty. Feature Prioritization: Shorter horizons provide more consistent outcomes and may serve as reliable predictors, but the higher profitability of longer horizons emphasizes the need to balance risk and reward dynamics in the model.
- Consider Multi-Target Regression: The significant differences across time horizons highlight the importance of considering them independently, potentially through multi-target regression to predict profitability for each horizon simultaneously.

By incorporating these findings, predictive models can better optimize trade execution and profitability forecasting, tailoring strategies to different time horizons to achieve a balance between expected returns and associated risks.

2.4 - Temporal Analysis

Method

Our goal for this analysis was to investigate patterns in average returns over three distinct temporal dimensions: days of the week, months of the year, and daily trends throughout the entire dataset.

The first step involved examining whether returns varied systematically across the days of the week. The central hypothesis was that market behavior could differ depending on the day due to factors such as trading volumes, investor sentiment, or external economic events. The data was grouped by day, and the average return was computed for each day to smooth out daily fluctuations and highlight overarching trends. A bar chart was used to visualize the results, with the x-axis representing the days of the week and the y-axis showing the average returns for each day of the week.

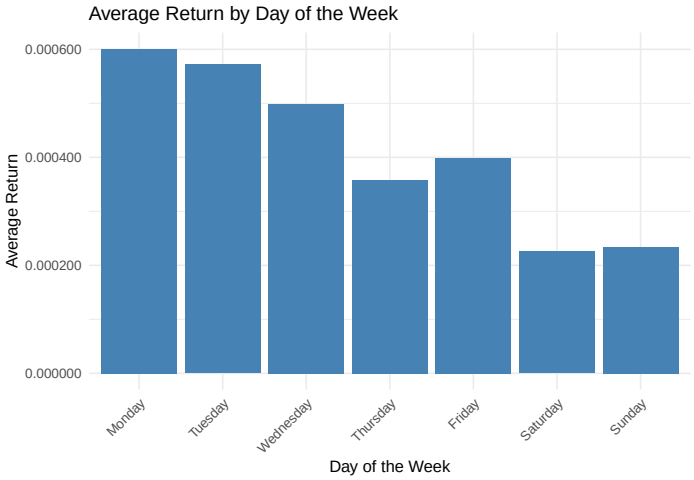
The second part focused on identifying seasonal trends in returns by analyzing monthly averages. Each transaction was mapped to its respective month, creating a categorization based on the calendar year. The dataset was grouped by month, and the average return was calculated for each group. To facilitate interpretation and identify seasonal patterns, the results were visualized using a bar chart where the x-axis represented the months and the y-axis depicted the average returns. A discrete color palette was applied to enhance differentiation among the months.

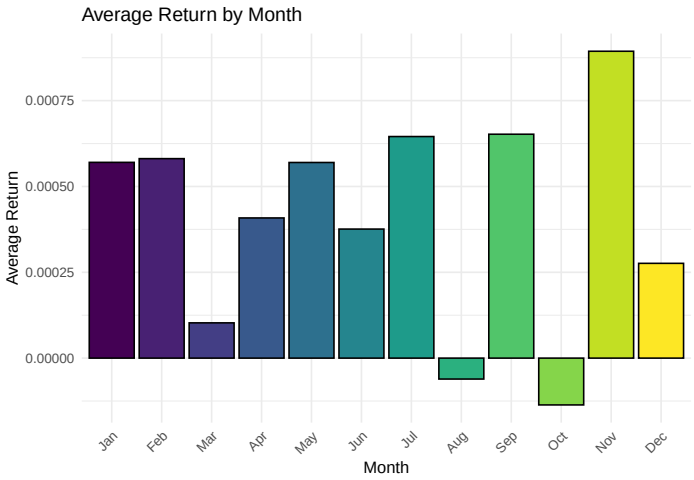
Lastly, we aimed to capture the overarching trend in returns across the entire dataset at a daily granularity. The data was grouped by individual dates, and the average return was calculated for each day. To visualize these daily trends over time, we constructed a line graph with the x-axis representing the timeline in chronological order and the y-axis showing the average return for each day.

Unlike other computationally intensive analyses in this study, this approach was computationally feasible to run on the entire dataset, enabling us to leverage all available data for a more robust analysis.

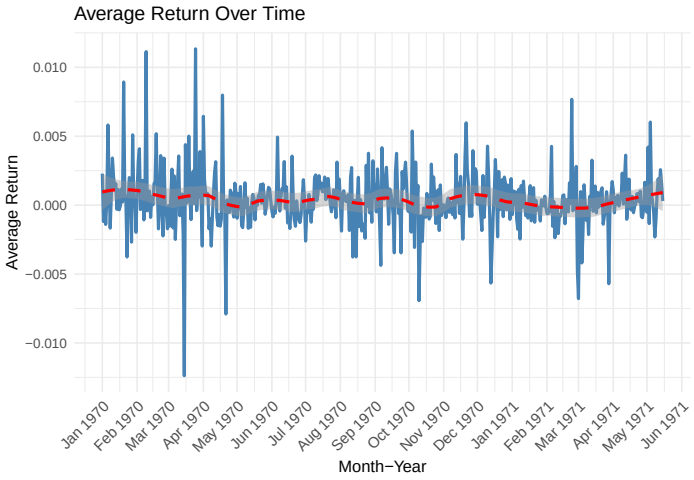
By addressing all three temporal dimensions, we ensured a comprehensive exploration of potential time-based patterns in returns. These findings served as a foundation for understanding seasonality and incorporating temporal features into predictive modeling.

Analysis






```
## `geom_smooth()` using formula = 'y ~ x'
```



The bar plot analyzing average returns by the day of the week reveals distinct patterns in trading behavior and market dynamics. Mondays consistently show the highest average return, which could be attributed to a "weekend effect," as traders respond to events or news that occurred while markets were closed. After Monday, there is a gradual decline in returns on Tuesday and Wednesday, suggesting a stabilization of market activity as early-week information is absorbed. However, there is a notable drop in average returns on Thursday, potentially due to pre-weekend adjustments, increased risk aversion, or the influence of external factors like economic data releases. This trend reverses on Friday, with average returns recovering, possibly driven by traders closing positions or adjusting portfolios ahead of the weekend. Over the weekend, returns drop significantly, reflecting the limited activity or closures of most financial markets during this period.

The bar plot of average returns by month reveals a distinct seasonal pattern in profitability across the year. January and February exhibit relatively high average returns, suggesting a possible "January effect," a well-documented phenomenon in financial markets where returns tend to be higher at the beginning of the year. This could be due to factors such as portfolio rebalancing or increased market activity following the holiday season.

The average return drops significantly in March and April, indicating a potential slowdown in market

momentum during this period. However, there is a recovery in May and the subsequent summer months of June and July, which show a consistent upward trend in average returns. This may reflect increased trading activity or other market dynamics during the mid-year period.

In contrast, August and September show a sharp decline in average returns, with September having the lowest returns of any month. This aligns with historical market patterns, as September is often regarded as a challenging month for financial markets. The reasons for this could include seasonal investor behavior or broader economic factors.

The final quarter of the year shows a rebound, with October and November experiencing significant increases in average returns. This might be influenced by end-of-year trading strategies, such as tax-loss harvesting or institutional investors making adjustments before closing their books. December's returns remain positive but show a slight dip compared to November, possibly reflecting reduced market activity during the holiday season.

The analysis of average returns over time reveals significant short-term fluctuations with periodic spikes and dips, indicative of market turbulence or event-driven impacts. The LOESS trendline shows a slight upward trend, suggesting improving profitability over time, but also highlights cyclical declines, reflecting potential seasonal or external market dynamics.

These patterns emphasize the importance of incorporating temporal features, such as seasonality or recent return trends, into predictive models to account for market behavior. However, the historical context of this data may limit its direct applicability to future predictions, as market conditions, trading strategies, and external factors may have evolved significantly. Models must account for these historical differences to ensure robust and adaptive forecasting.

Conclusion

This analysis reveals a clear weekly trend in average returns, with Mondays consistently outperforming other days of the week, followed by Tuesday and Wednesday. Returns decrease towards the end of the week, with the lowest averages occurring over the weekend. These findings align with well-documented market phenomena, such as the Monday Effect and there usually being a reduction of activity on weekends. For investors, this suggests that timing trades earlier in the week may offer slightly higher returns, though the practical significance of this pattern depends on the magnitude of the differences and the specific asset class being considered.

Future analyses could expand on this by examining whether the observed trends persist across different years or market conditions. Additionally, incorporating external factors such as trading volumes, economic news releases, or volatility indexes could provide a deeper understanding of the drivers behind these weekly 2patterns.

3. Advanced Analysis

Method

In this analysis, we implemented a predictive modeling pipeline using XGBoost to predict the target variable resp. To begin, we addressed multicollinearity by removing highly correlated features: we referred to our analysis from part 2.1 to do so. This step ensured that redundant features did not inflate the model's variance or negatively impact interpretability. Next, we split the dataset into three subsets: 70% for training, 15% for validation, and 15% for testing. The split was performed randomly, with set.seed(123) ensuring reproducibility and consistent data distributions across subsets. After splitting, we excluded non-predictive columns (date, ts_id) and response variables other than resp, using only the remaining features to train the model.

For hyperparameter tuning, we employed a reduced grid search approach to balance computational efficiency with performance optimization. The grid included key hyperparameters such as max_depth (2, 4, 6, 8), eta (0.05, 0.1), colsample_bytree (0.7), subsample (0.8), and gamma (0). Each hyperparameter combination was evaluated using up to 150 boosting rounds, with early stopping after 10 rounds of no improvement in

validation performance to prevent overfitting. The best hyperparameter combination, based on the lowest validation RMSE, was selected for the final model. The model was then trained with the optimal parameters on the training set and evaluated on the test set, where RMSE was calculated to measure performance on unseen data.

To contextualize the model's performance, we created a baseline model using a simple mean prediction for all test samples. This baseline provided benchmarks for evaluating the XGBoost model's effectiveness. Finally, we examined feature importance using xgb.importance to identify the most impactful predictors, offering insights into the key drivers of the target variable. This systematic methodology ensured the development of a robust and interpretable model while maintaining computational efficiency.

Analysis

```
## Model Performance Results

## Baseline Model RMSE: 0.025396

## Final Model RMSE: 0.025223
```

Table 1: Top 5 Features by Importance (XGBoost)

Feature	Gain	Cover	Frequency
feature_43	0.0624514	0.0843561	0.0409535
feature_39	0.0360273	0.0902383	0.0265892
feature_27	0.0345603	0.0855684	0.0267421
feature_45	0.0322204	0.0124668	0.0239914
feature_5	0.0308217	0.0474106	0.0377445

Conclusion

The results of our predictive modeling demonstrate a modest success in improving performance over the baseline model. The final XGBoost model achieved a test RMSE of 0.02522255, which is lower than the baseline RMSE derived from the mean prediction (0.02539697). This indicates that the XGBoost model was able to capture more nuanced relationships in the data compared to a naive mean-prediction approach. The success of the XGBoost model can be attributed to its ability to leverage the predictive power of individual features and its capacity to handle complex relationships and interactions among them.

Although the improvement over the baseline is numerically small, its significance becomes more apparent when considering the weighted nature of the target variable. By incorporating weights, which reflect the magnitude of the returns, even slight reductions in RMSE could translate into substantial differences in profitability from a financial perspective. This highlights the practical relevance of the XGBoost model's improvement over the baseline. The baseline models, while simpler, fail to account for the complex dynamics captured by the XGBoost model, particularly in scenarios where larger returns hold greater weight.

To further enhance performance while minimizing computational demands, we focused on simplifying the hyperparameter search space and reducing feature redundancy. By restricting the range of hyperparameters and using a grid search over fewer configurations, we balanced computational efficiency with model optimization. Additionally, as identified in section 2.1 of our analyses, collinear features were removed, reducing the number of features processed during training and ensuring that redundant information did not inflate computational costs or distort the model's learning process. However, one limitation of the final model is that it did not include all the insights derived from earlier exploratory data analysis (EDA). Features such as temporal trends, day-of-the-week patterns, and monthly seasonality, which showed promise during EDA, were excluded in the current iteration to keep the computational requirements manageable. Future work could integrate these features to improve predictive power further while exploring methods to handle the increased computational load.

The feature importance analysis offers valuable insights into the drivers of the model's predictive power. Feature_43 emerged as the most important predictor, followed by feature_39, feature_27, and feature_45. These features likely capture critical market dynamics or trading patterns closely linked to the target variable, resp. The varying importance scores across features indicate their diverse roles in prediction, with some providing strong individual signals and others contributing complementarily. Leveraging these insights, future iterations could refine the dataset by prioritizing high-importance features and engineering new ones based on their relationships to market behavior. For example, examining the specific characteristics of feature_43 could inspire domain-specific transformations or interactions that enhance the model's performance.

In conclusion, the XGBoost model demonstrated its superiority over the baseline models, both in terms of predictive accuracy and potential financial impact. The incorporation of weighted returns underscores the value of even slight RMSE improvements in a financial context, where profitability depends on accurately predicting large-magnitude returns. Despite these successes, the model's predictive power could be further improved by reintroducing features identified in EDA, such as temporal patterns, and by experimenting with more advanced optimization techniques like random search or Bayesian optimization. By strategically balancing feature selection, computational efficiency, and advanced model tuning, future iterations can aim for greater predictive accuracy and practical applicability in the challenging domain of financial market modeling.

4. Discussion and Conclusion

In this report, we explored the Jane Street Market Prediction dataset to uncover insights into market dynamics, profitability drivers, and predictive modeling. The analysis revealed key patterns that align with common themes in stock market behavior. Feature correlations highlighted the lack of strong linear or monotonic relationships between individual features and returns, indicating the need for non-linear modeling approaches to capture the complexity of financial markets. The study of trade weights demonstrated that profitability varies significantly with trade size, with mid-range weights offering optimal returns, reflecting the balance between market impact and liquidity—a recurring challenge in portfolio management and trade execution. Temporal analysis uncovered weekly and seasonal patterns, such as higher returns on Mondays and during specific months like January, echoing phenomena such as the "Monday Effect" and the "January Effect," which are well-documented in financial literature.

Additionally, the confidence intervals for returns across different time horizons revealed that longer-term trades offer higher average returns but come with increased variability, underscoring the classic risk-reward tradeoff in finance. Finally, our predictive modeling efforts showed that an XGBoost model, optimized through hyperparameter tuning, outperformed the baseline mean-prediction model, albeit marginally. The importance of features like trade weight and specific anonymized attributes reinforces the idea that predictive models must carefully weigh input features to extract actionable insights.

Overall, the report highlights the intricacies of financial data and the importance of tailoring analysis and models to the unique characteristics of the market. While our findings align with established themes such as the influence of trade size, temporal patterns, and risk-return dynamics, the complexity and variability of the dataset suggest that real-world trading strategies must be adaptive and robust to evolving market conditions. Future work could extend this analysis by incorporating external factors like macroeconomic indicators, historical context, trading volume, or volatility metrics to further contextualize predictions and enhance model performance.