SALES FORECASTING FOR E-GROCERY WAREHOUSE

DAB422-25W-003 CAPSTONE PROJECT 2

Group 7 – Section 3

Contents

PROJECT SUMMARY

The project focuses on building a predictive sales model for a grocery store in Frankfurt. The goal is forecasting sales for the next 14 days. The project aims to enhance historical sales data, weather patterns, holiday information, and inventory details to predict future sales trends. Improvement of sales forecasting accuracy by using advanced machine learning and *XGBoost and LightGBM* is the goal.

PROBLEM STATEMENT

Accurate sales forecasting is a challenge for grocery stores. Inaccurate predictions can lead to revenue losses, overstocking or understocking, increased operational costs. Traditional forecasting methods often struggle to capture complex time-dependent sales patterns, which are impacted by seasonality, holidays, weather conditions.

This project aims to address the following key business problems:

- Predicting sales with high accuracy
- Reducing forecasting errors

The project aims to create a reliable predictive model by applying machine learning techniques such as *XGBoost* and *LightGBM.*

OBJECTIVE

The primary objective of the project is to build a robust predictive model for sales forecasting. The goal is to predict sales for the next 14 days.

DATA

The data originates form the Kaggle competition in predicting sales for a grocery network in Europe.
Additionally, weather data was scrapped using Visual Crossing weather API.

Data contains the following tables:
Calendar: contains information about holidays
Sales_train: training set contains historical sales data for given date
Sales_test: testing set
Inventory: additional information about inventory
Sales_train data consists of 202,153 rows and contains 2 years sales data.

Weather: additional climate information

METHODOLOGY

Tools Used

- Python
- Pandas & NumPy
- Matplotlib & Seaborn
- Scikit-learn
- Google Colab
- Microsoft VS Code
- Visual Crossing API
- Recursive Feature Elimination
- Optuna

Time Series Forecasting Models

- *XGBoost, LightGBM* – machine learning models adapted for time series prediction.

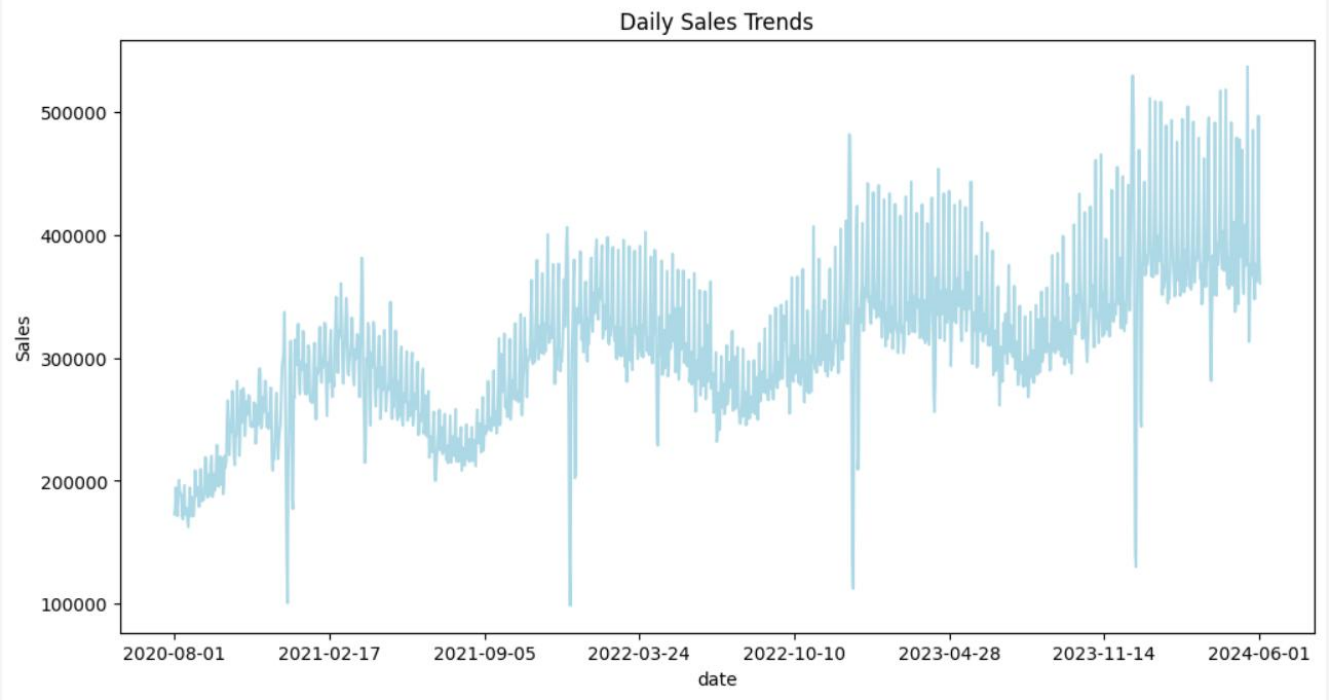
Metric Evaluation

- Mean Absolute Error (MAE)
- R-squared
- Root Mean Squared Error (RMSE)

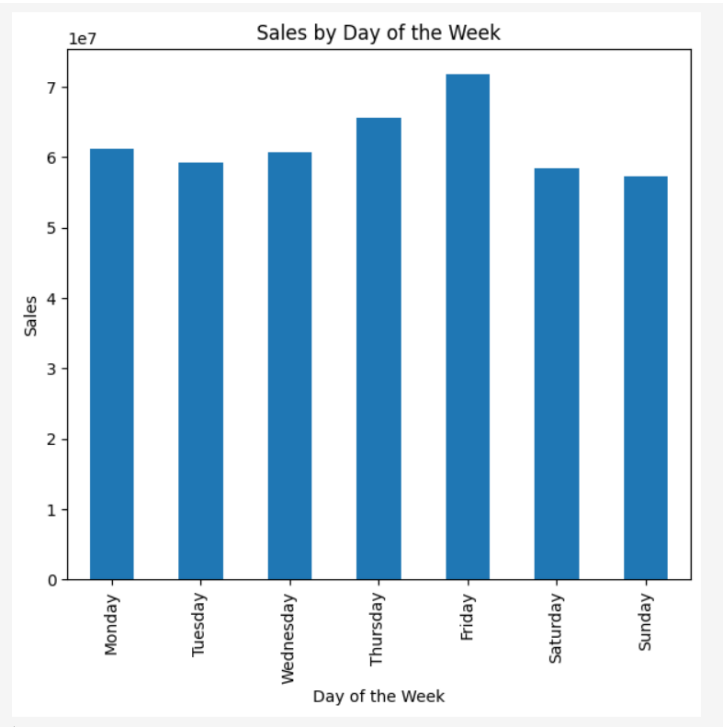
FEATURE ENGINEERING

- **Date features:** day, month, year, day of week were extracted from the date, days_till_next_holiday, day_before_shops_closed – were created based on calendar data.
- **Lag features:** lag features like 7_day_lag, 14_days_lag, 365_days_lag – to capture historical patterns. These features show us sales from the same day 7 and 14 days ago correspondingly.
- **Fourier Features:** day_sin, day_cos, year_sin, year_cos – to allow model to recognize the cyclic nature of data.
- **Rolling features:** moving average of 7, 14, 30 days sales – to provide model average sales for these periods.
- **Target Encoding:** means_sales by each unique_id – for compact category-target relationships
- **One-hot encoding:** Used to ensured that encoded categorical variables are treated independently

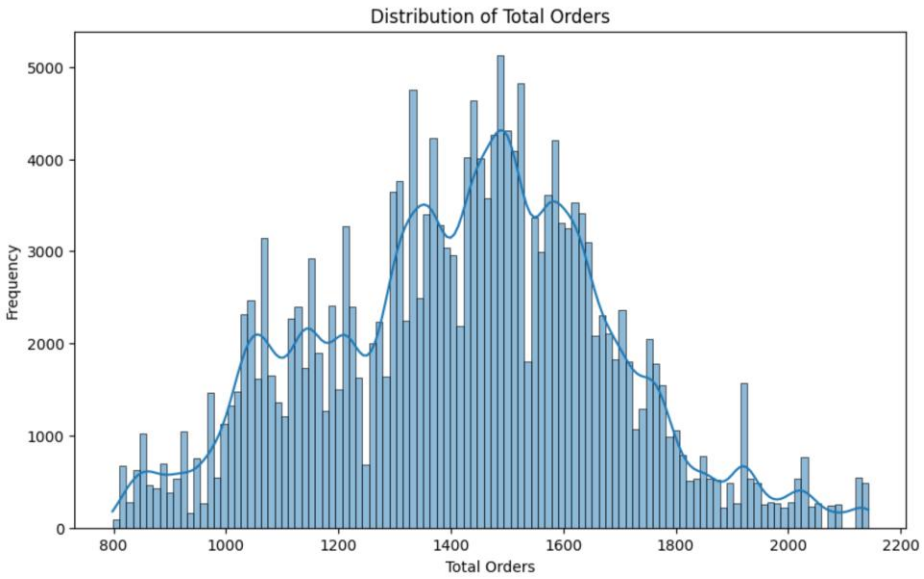
EXPLORATORY DATA ANALYSIS (EDA)



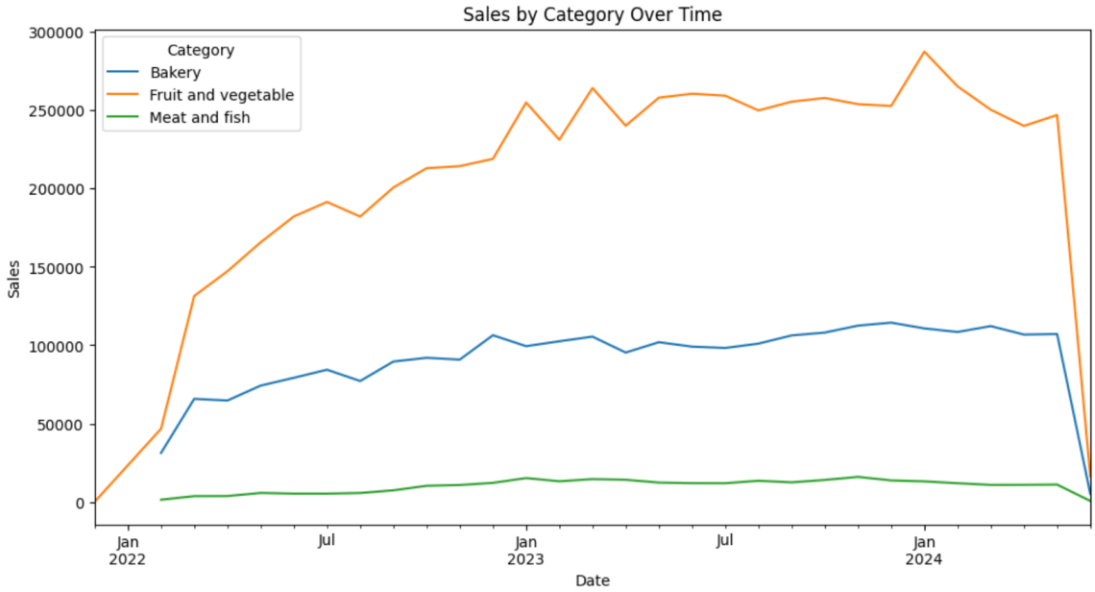
The chart "Daily Sales Trends" shows the fluctuations in daily sales over time. The sales show a general upward trend with periodic dips and spikes, indicating seasonal or cyclical patterns. The overall trend suggests growth in sales.



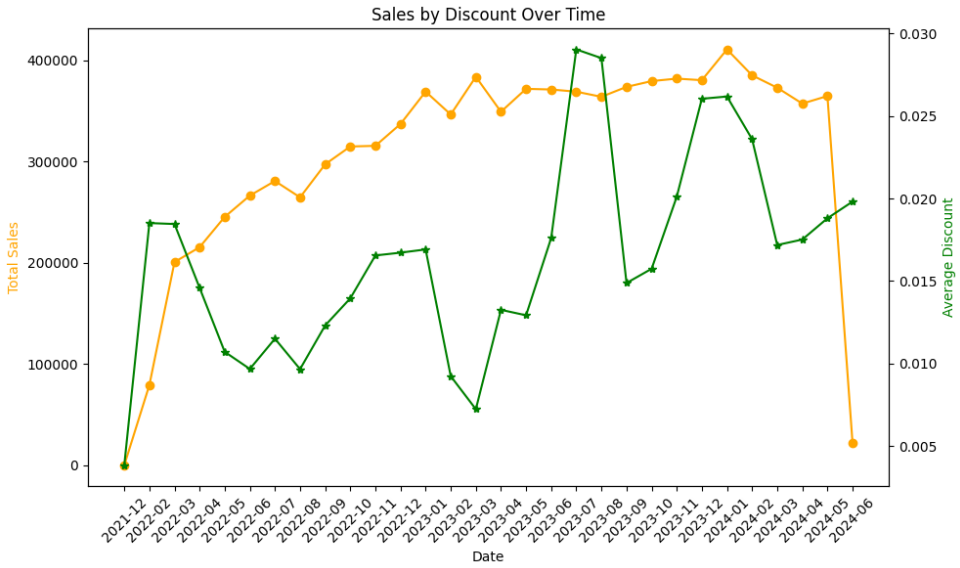
The bar plot "Sales by Day of the Week" displays total sales for each day. Sales are relatively consistent across the week, with Friday showing the highest sales. Saturday and Sunday have slightly lower sales compared to weekdays.



The histogram "Distribution of Total Orders" shows the frequency distribution of total orders. The data is almost normally distributed, with a peak around 1400 orders. The density plot overlaid suggests multiple minor peaks, which indicates about possible subgroups within the data.



The line plot "sales by Category Over Time" shows the sales fluctuations by each of the three categories in warehouse. The 'Fruit and vegetable' category has significantly higher sales compared to other two categories. And as we can see it was the first category for the grocery shop to start. We can see the trend of sales dropping in December and July and increasing in January.



The line plot "Sales by Discount Over Time" show us how sales and discount correlate over time. We can see that as discount increase the sales start also to grow up. However, there is no clear straightforward correlation between sales volume and discount level.

Chi-Square Test of Independence

```
Chi-Square Statistic: 673.8173579726122
P-value: 7.039654482576168e-134
Degrees of Freedom: 15
Expected Frequencies:
[[ 2139.02270637  5438.00672595  3815.41200718 21803.81982698
   6353.76565744 10223.97307609]
 [ 2137.38966777  5433.85507538  3812.49912784 21787.17368334
   6348.91486998 10216.16757569]
 [ 2135.92852798  5430.14044066  3809.89286738 21772.27976535
   6344.57469173 10209.18370691]
 [ 2136.65909788  5431.99775802  3811.19599761 21779.72672434
   6346.74478085 10212.6756413 ]]
There is a significant association between sales quartiles and the days till next holiday.
```

Figure 1 – Chi-Square Test of Independence (Sales/ Days Till Next Holiday)

```
Chi-Square Statistic: 505.11473193969255
P-value: 3.7172513621389243e-109
Degrees of Freedom: 3
Expected Frequencies:
[[47941.98172231  1832.01827769]
 [47905.38037812  1830.61962188]
 [47872.63180701  1829.36819299]
 [47889.00609256  1829.99390744]]
There is a significant association between sales quartiles and the day before shop closure.
```

Figure 2 – Chi-Square Test of Independence (Sales/Day before Shops Closed)

Figure 1 and Figure 2 show the results of Chi-Square Test of Independence between Sales and Days till next holiday and day before shops closed correspondingly. Both tests have showed a significant association between sales and the second tested variable. This information will be considered in further modelling.

RESULTS

Model	MAE	Train R2	Test R2	Train RMSE	Test RMSE	Outliers Cleaned	Hyperparameter Tuning	Recursive Feature Elimination (N)
XGBoost	14.83	0.95	0.82	15.13	29.96	No	No	No
LightGBM	14.54	0.95	0.83	15.13	28.61	No	No	No
XGBoost	8.21	0.86	0.71	7.48	11	Yes	No	No
LightGBM	8.34	0.83	0.71	8.22	11.1	Yes	No	No
XGBoost	8.03	0.85	0.73	7.56	10.66	Yes	No	Yes (50)
LightGBM	8.43	0.83	0.71	8.22	11.16	Yes	No	Yes (50)
XGboost	8.13	0.84	0.71	7.83	11	Yes	No	Yes (40)
LightGBM	8.63	0.83	0.69	8.2	11.47	Yes	No	Yes (40)
XGBoost	8.49	0.86	0.7	7.51	11.32	Yes	No	Yes (60)
LightGBM	8.54	0.83	0.7	8.19	11.28	Yes	No	Yes (60)
XGBoost	7.84	0.83	0.72	8.18	10.48	Yes	Yes	Yes (50)
Light GBM	8.16	0.85	0.72	7.54	10.93	Yes	Yes	Yes (50)

Figure 3 - Results.

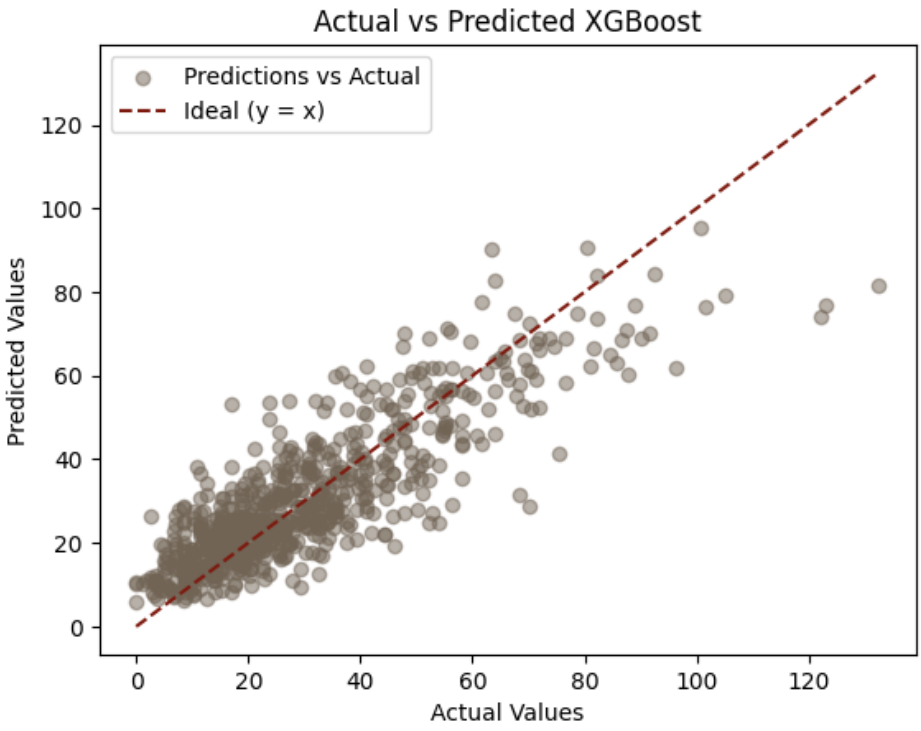
We have tested *XGBoost and LightGB* for sales forecasting. The models have shown similar accuracy results. However, the XGBoost model presented the lowest *Mean Absolute Error.* The *Root Mean Squared Error* in the Train set versus the Test set, indicates that our model is slightly overfitting, something that will be addressed in further work.

Key performance metrics:

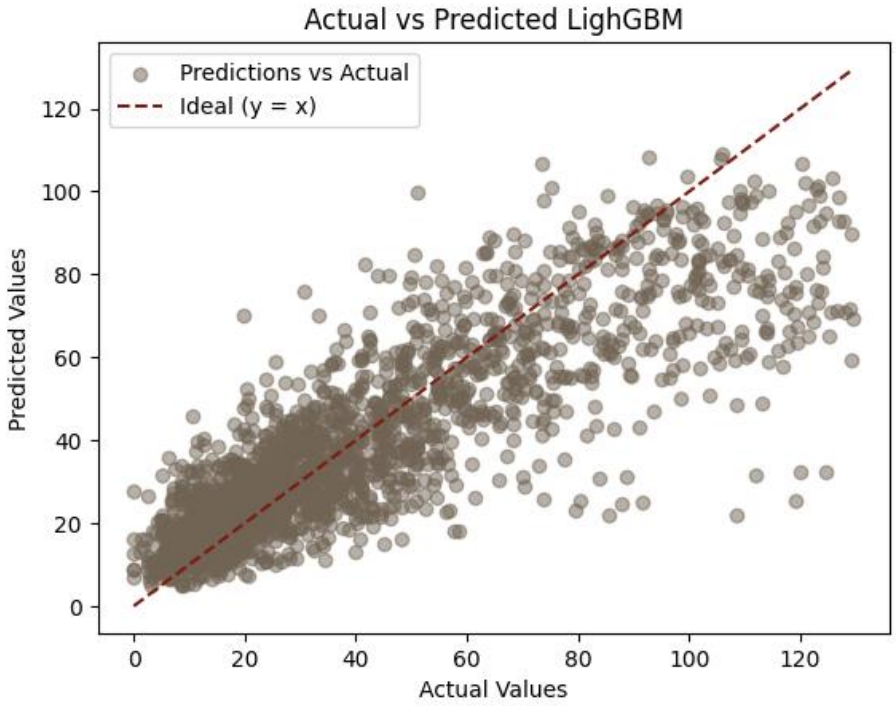
- *XGBoost:* MAE, RMSE, R^2
- *LightGBM*: MAE, RMSE, R^2

Plot 1 and Plot 2 show the predicted vs actual values for XGBoost and LightGBM. These plots show us that for small values the data points are very close to ideal line (x=y), but as the values increase the spread of the points also increases, meaning that our model predicts larger values worse. This is the point for future improvement.

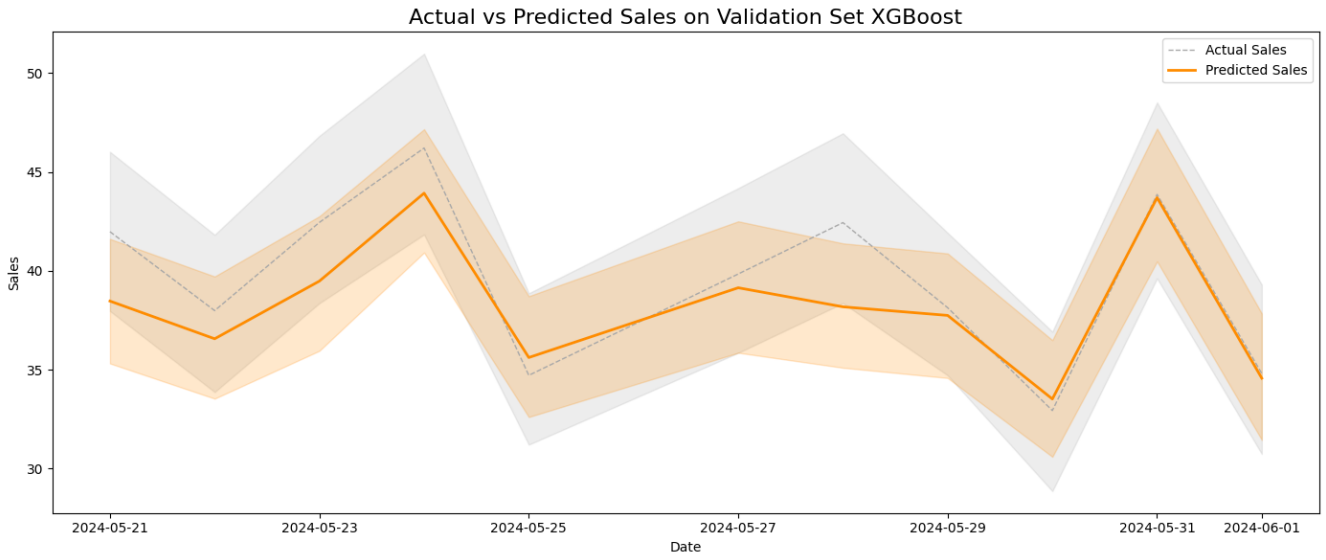
Plot 3 and Plot 4 show the predictions over time vs actual values.


Actual vs Predicted XGBoost
Predictions vs Actual
Ideal (y = x)
Predicted Values
Actual Values
0
20
40
60
80
100
120

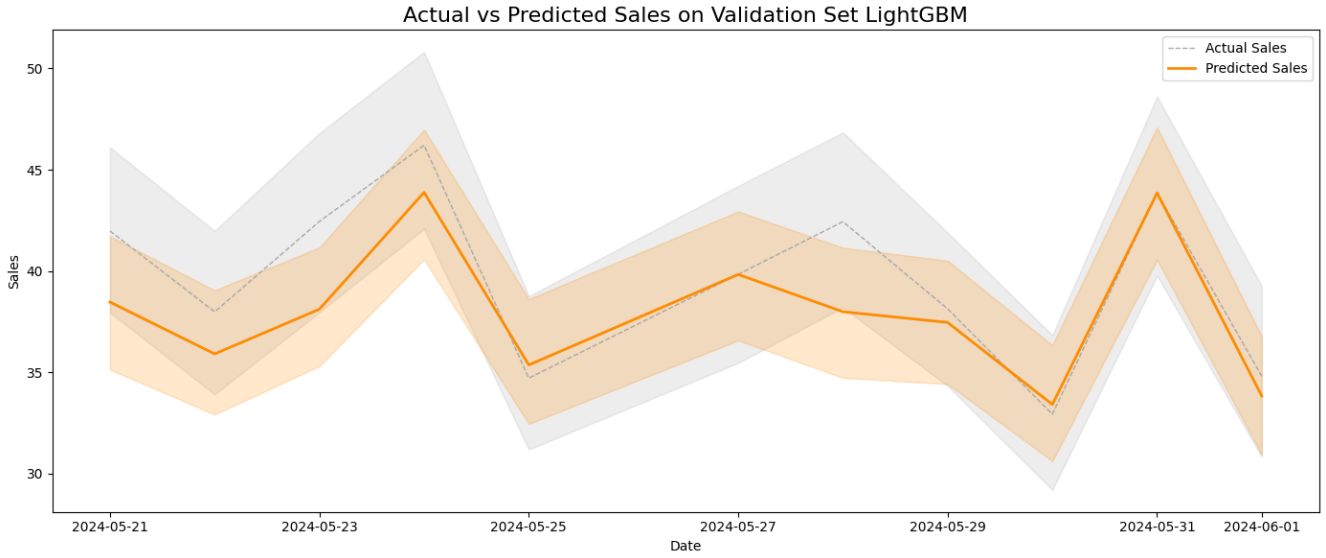
Plot 1


Actual vs Predicted LighGBM
Predictions vs Actual
Ideal (y = x)
Predicted Values
Actual Values
0
20
40
60
80
100
120

Plot 2



Plot 3



Plot 4

CHALLENGES FACED

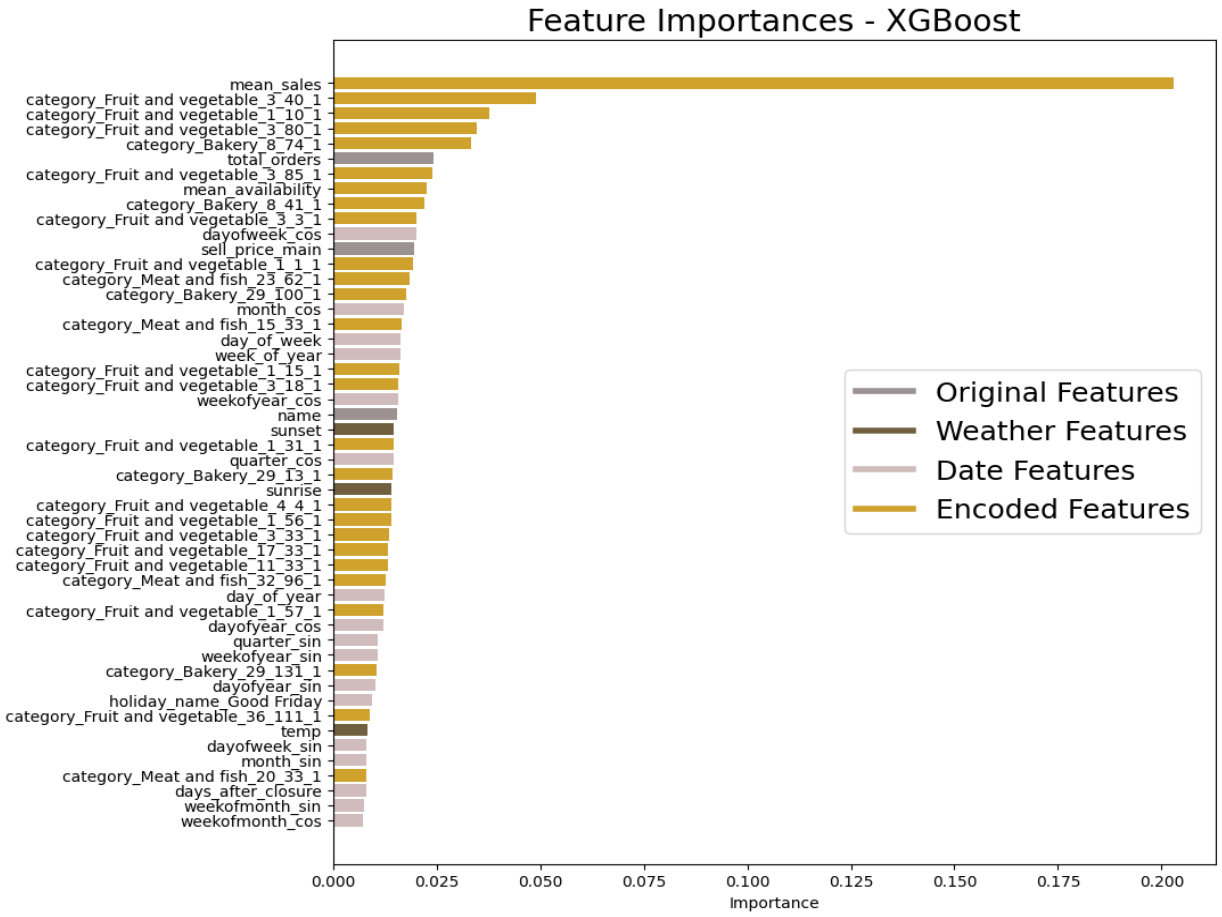
- **Hyperparameter Tuning:** Finding the best settings for *XGBoost, LightGBM* to improve performance.
- **Utilizing the Full Dataset:** Ensuring that all data is used effectively without overloading the model with unnecessary features.
- **Overfitting:** All the models show slight overfitting, thus require regularization techniques.

KEY TAKEAWAYS

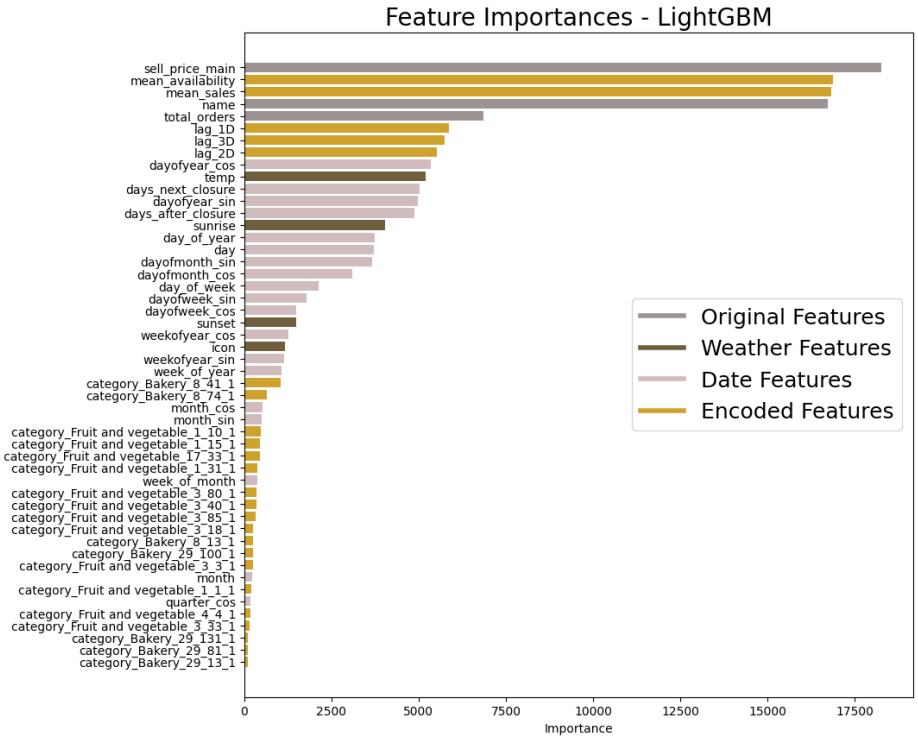
- Importance of Feature Engineering

  If we look at the plot of the Feature Importances for both of our models (Picture 1, Picture 2), we can see that there are only few original features among those that were selected for model training, using Recursive Feature Elimination method. And the biggest part of the features was engineered in different ways: encoding, external implementation, extracted from original data.
- Effectiveness of Recursive Feature Elimination

  If we look at the Results Table (Figure 3), it is visible that Recursive Feature Elimination improved the performance of both models compared to baseline, and the number of features also has impact on the model performance. Implementing this method helped us a lot in automotive feature selection and saved a lot of time.



Picture 1 – Feature Importance for XGBoost Model



Picture 2 – Feature Importance for LightGBM Model

NEXT STEPS

- Conduct more precise hyperparameter tuning for *XGBoost, LightGBM*, and *RandomForestRegressor* to improve prediction accuracy.
- Focus on addressing overfitting by implementing regularization techniques such as early stopping, L1/L2 regularization.
- Research about ways to reduce MAE and experiment with different techniques.
- Deploy the best-performing model for final evaluation and implement it for forecasting future sales in the business context.

CONCLUSION

This project aimed to develop a model for accurate sales forecasting for an e-grocery warehouse using machine learning. Methodology included feature engineering, exploratory data analysis, and model testing. The team built predictive models using XGBoost and LightGBM. The feature engineering and particularly lag features, Fourier transformations, rolling averages, data encoding, extracting data features significantly improved model performance by capturing underlying patterns in the data.

XGBoost was the top-performing model based on key evaluation metrics such as Mean Absolute Error (MAE) and R-squared, but slight overfitting took place. The team used Recursive Feature Elimination for feature selection and that enhanced model performance significantly.

The results demonstrate that machine learning models can effectively handle the complexity of time-series data in the retail domain. Despite challenges such as tuning hyperparameters and addressing model generalization, the project establishes a solid foundation for deploying a reliable forecasting system.

Future work will focus on refining these models further, incorporating additional data sources, exploring deep learning model for more advanced forecasting, and implementing the best model into a real-world business context to support inventory and operational decisions.

REFERENCES

Galli, Soledad. 2022. *Python Feature Engineering Cookbook - Second Edition*. s.l. : Packt Publishing, 2022.

M5 accuracy competition: Results, findings, and conclusions. **Spyros Makridakis, Vassilios Assimakopoulos. 2022.** 4, s.l. : International Journal of Forecasting on ScienceDirect, 2022, Vol. 38. https://doi.org/10.1016/j.ijforecast.2021.11.013.

Mouna Labiadh. 2023. Understanding Temporal Fusion Transformer. *Medium*. [Online] 04 12, 2023. https://medium.com/dataness-ai/understanding-temporal-fusion-transformer-9a7a4fcde74b.

Terence Parr, Jeremy Howard. 2018-2019. The Mechanics of Machine Learning. [Online] 2018-2019. https://mlbook.explained.ai/.