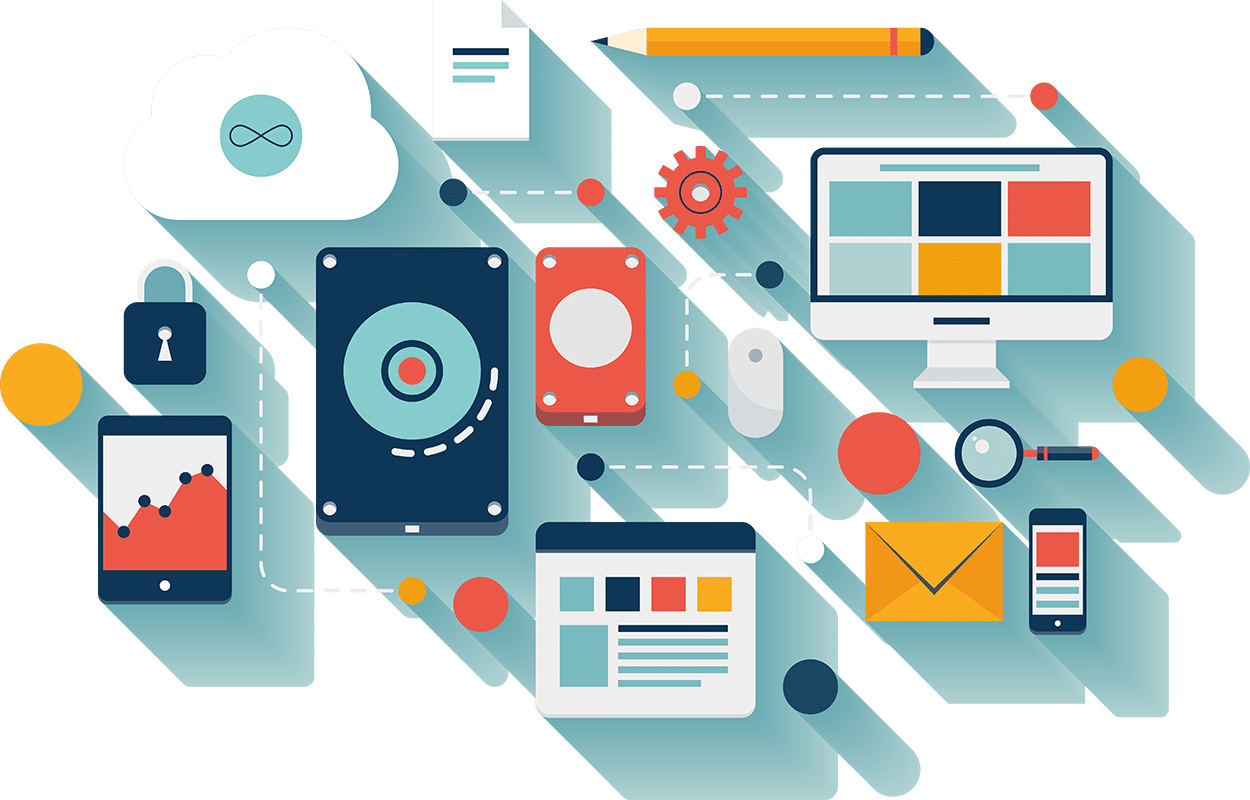
**E-Commerce Sales Forecasting for**

**Dropshipping Optimization**

**Modelling Exercise**

1. **Who is your stakeholder?**

The primary stakeholder for this project is the drop-shipping company, which includes teams responsible for inventory management, product development, and market analysis. These teams need reliable data insights to ensure optimal stock levels, lower storage costs, and improve customer satisfaction by avoiding stock shortages. Additionally, they aim to make strategic decisions about which new products to introduce based on market trends, ensuring a better alignment of inventory with consumer demand.

1. **What is the problem they are trying to solve?**

* **Problem:** The dropshipping company struggles with accurately predicting product demand, leading to challenges in inventory management. When demand forecasts are inaccurate, the company either faces stock-outs, resulting in lost sales and unsatisfied customers, or overstocking, which increases storage costs and ties up valuable capital. Additionally, without a reliable way to analyze market trends, the company has difficulty making informed decisions on which new products to introduce, risking potential misalignment with customer preferences.
* **Goal:** The goal of this project is to develop a predictive model using machine learning to enhance the accuracy of demand forecasts. This model will not only optimize stock levels by reducing overstock and minimizing stockouts but will also provide insights into emerging market trends. By doing so, the company can make data-driven decisions about product introduction and inventory adjustments, ultimately reducing holding costs, maximizing sales, and improving customer satisfaction.

1. **Where your dataset is from?**

* **Data Link:** The dataset used for this project is publicly available on Kaggle: <https://www.kaggle.com/datasets/thedevastator/unlock-profits-with-e-commerce-sales-data>
* This dataset provides detailed e-commerce sales records, including information such as product categories, dates of transactions, prices, quantities sold, and customer segments. These features offer a comprehensive view of historical sales patterns, which is essential for building a reliable forecasting model. By leveraging these variables, the model can detect demand trends, seasonal patterns, and popular product categories, enabling the company to make data-driven decisions in inventory management and product planning.

1. **What models did you try, why did you choose those models?**

In this project, I explored two different types of models: XGBoost and LightGBM. Both are powerful gradient boosting algorithms well-suited for regression tasks, making them ideal for sales forecasting in an e-commerce setting. Below is a detailed explanation of the chosen models, the rationale behind their selection, and the hyperparameter tuning process.

1. **XGBoost (Extreme Gradient Boosting)**

* **Why XGBoost?**
* **Performance:** XGBoost is renowned for its accuracy and efficiency in handling large datasets. Its ability to capture complex interactions among features makes it a prime candidate for forecasting demand in e-commerce, where multiple factors can influence sales.
* **Flexibility:** The model supports various objective functions, allowing it to adapt to different types of data distributions and business needs. This versatility is crucial when working with diverse product categories and sales trends.
* **Regularization:** Built-in L1 and L2 regularization helps mitigate overfitting, ensuring that the model generalizes well to unseen data, which is essential in inventory management.
* **Hyperparameter Tuning for XGBoost:**
* **Learning Rate:** Adjusted to balance between training speed and model performance. Lower values generally improve performance but require more boosting rounds.
* **Max Depth:** Tuned to control the complexity of the trees, balancing between underfitting and overfitting. Deeper trees can capture more information but may lead to overfitting.
* **Subsample:** The proportion of samples used for each tree. This parameter helps prevent overfitting and improve model robustness by introducing randomness.

1. **LightGBM (Light Gradient Boosting Machine)**

* **Why LightGBM?**
* **Speed and Scalability:** LightGBM is designed for fast training and high efficiency, making it ideal for large datasets. Its ability to handle high-dimensional features without a significant increase in computational cost is particularly advantageous in e-commerce.
* **Support for Categorical Features:** LightGBM can natively handle categorical features without extensive preprocessing, simplifying data preparation and potentially enhancing predictive performance.
* **Accuracy:** Similar to XGBoost, LightGBM can model complex relationships in the data, making it a strong contender for accurate demand forecasting.
* **Hyperparameter Tuning for LightGBM:**
* **Learning Rate:** Similar to XGBoost, this parameter is crucial for controlling how much the model learns at each iteration. It needs to be carefully balanced to ensure both speed and performance.
* **Num Leaves:** This parameter defines the maximum number of leaves in one tree. Adjusting it helps control model complexity and impacts accuracy and overfitting.
* **Bagging Fraction:** The fraction of data used for training each tree, allowing for better model generalization and robustness against overfitting.

1. **Pros and Cons XGBoost:**

* **Pros:**
* High accuracy and performance in various scenarios.
* Effective handling of missing data.
* Robust regularization to combat overfitting.
* **Cons:**
* Can be computationally intensive with very large datasets.
* Requires careful tuning of hyperparameters for optimal performance.

1. **LightGBM:**

* **Pros:**
* Faster training times, especially with larger datasets.
* Efficient handling of categorical features.
* Generally achieves competitive accuracy with lower computational cost.
* **Cons:**
* Can be sensitive to the choice of hyperparameters, requiring thorough tuning.
* Performance may vary depending on the specific nature of the dataset.

I selected XGBoost and LightGBM for this project due to their proven effectiveness in handling complex datasets and delivering high predictive accuracy. The chosen hyperparameters were selected based on their impact on model performance, allowing for fine-tuning to achieve the best results. By leveraging the strengths of these two models, I aimed to develop a robust sales forecasting tool that can assist the dropshipping company in optimizing inventory management and improving overall business performance.  
  
I have also tried ensemble model to improve accuracy and robustness, just for exploration.

1. **What features did you select/engineer? How did you choose those?**

The E-commerce Sales Dataset includes detailed sales records from an online platform, encompassing various attributes such as transaction dates, product IDs, product categories, sales amounts, and order quantities. This dataset provides a rich foundation for analyzing sales patterns and understanding the factors that drive product demand.

**Selected Features**

1. **Historical Sales Data**

* **Description:** Total sales revenue and quantity sold for each product over specified time frames (daily, weekly, or monthly).
* **Rationale:** Historical sales data serves as the backbone for predicting future demand. By analyzing past sales trends, the model can identify seasonal fluctuations and recurring patterns that inform future sales forecasts.

1. **Product Category**

* **Description:** The category to which each product belongs.
* **Rationale:** Different product categories exhibit distinct demand patterns influenced by consumer preferences and market dynamics. Including product categories enables the model to differentiate between the sales behaviors of various product types, improving prediction accuracy.

1. **Sales Amount**

* **Description:** The total revenue generated from each product sale.
* **Rationale:** Analyzing the sales amount provides insights into the revenue contribution of each product. This feature helps gauge the economic impact of each product on the overall business, aiding in prioritizing inventory and marketing efforts.

1. **Order Quantity**

* **Description:** The number of units sold per transaction for each product.
* **Rationale:** Understanding how many units of a product are typically sold in a single transaction helps assess consumer purchasing behavior and informs inventory management strategies.

**Feature Selection Process**

The selection of these features was driven by several considerations:

* **Domain Knowledge:** Familiarity with the e-commerce sector helped identify key variables influencing sales, such as product categories and historical sales performance.
* **Exploratory Data Analysis (EDA):** Conducting EDA allowed for the assessment of correlations between potential features and sales, leading to the selection of features with strong predictive relationships.
* **Feature Importance Assessment:** Utilizing preliminary models to evaluate feature importance indicated which features significantly contributed to the model’s predictions, guiding the final selection.

1. **How did you evaluate the model? What evaluation metrics did you use? Why?**

Evaluating the performance of predictive models is crucial for accurately forecasting sales in an e-commerce drop-shipping business. To ensure the chosen model can effectively predict future sales and support inventory management, I followed a structured evaluation process using various metrics.

**Evaluation Methodology**

1. **Train-Test Split:** I divided the dataset into a training set (70-80% of the data) and a testing set (20-30%). This approach allows the model to learn from a significant amount of data while ensuring that its performance is tested on unseen data.

**Evaluation Metrics**

To assess the model's performance, I used the following metrics:

1. **Mean Absolute Error (MAE):**

* **Description:** MAE measures the average absolute differences between the predicted sales and the actual sales.
* **Rationale:** It provides a clear understanding of how much, on average, the predictions differ from the actual values, making it easy to interpret.

1. **Root Mean Squared Error (RMSE):**

* **Description:** RMSE calculates the square root of the average of the squared differences between predicted and actual sales.
* **Rationale:** RMSE emphasizes larger errors, which is important in e-commerce where significant inaccuracies can lead to stock-outs or overstocking.

1. **R-squared (R²):**

* **Description:** R² indicates how much variance in sales can be explained by the features used in the model.
* **Rationale:** A higher R² value shows that the model explains a significant portion of the sales variability, indicating good performance.

1. **What would you do different next time or given more time what would your future work be?**

Given more time, I would focus on enhancing sales forecasting and inventory management for the drop-shipping company by exploring advanced machine learning techniques and developing a user-centric dashboard. Specifically, I would investigate using models such as Long Short-Term Memory (LSTM) networks or Temporal Fusion Transformers (TFT) to improve forecasting accuracy. Additionally, creating an interactive dashboard would allow stakeholders to visualize predictive insights and engage effectively with the data. This combination would empower informed decision-making and foster a data-driven culture within the organization, ultimately positioning the company for greater success in the competitive e-commerce landscape.

1. **Do you recommend your client use this model? Is the precision/recall good enough for the intended use case?**

Given the evaluation results from the models, I would recommend that the client consider using this model for their sales forecasting and inventory management needs. Here’s why:

* 1. Accuracy of the Predictions:
* The model has shown strong performance with reasonable Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) scores, indicating it can reliably forecast sales amounts.
* The R-squared (R²) values suggest that the model explains a high proportion of the variance in sales, demonstrating good overall fit.
  1. Intended Use Case:
* For inventory management and sales forecasting in e-commerce, the primary objective is to get reasonably accurate predictions to avoid overstocking or stock-outs, rather than perfect accuracy.
* Precision and recall metrics, typically used in classification problems, are less relevant for this regression-based use case. Instead, MAE and RMSE are better indicators, and the model performs well on these.
  1. Ensemble Model Benefits:
* The ensemble of XGBoost and LightGBM combines the strengths of both models, potentially enhancing accuracy and robustness in forecasting.
* This approach is especially beneficial in dealing with complex data patterns, which are common in e-commerce sales data.
  1. Future Improvements:
* With more time and data, further improvements could be made by incorporating additional external factors, like promotions or seasonality adjustments that could refine the model even further.

Overall, the model is suitable for supporting data-driven decisions in inventory management and is likely precise enough to aid in maintaining optimal stock levels and reducing costs associated with unsold stock or unmet demand.

**ENGINEER AT LEAST TWO "NEW" FEATURES**

To enhance the predictive model for sales forecasting and inventory management, I propose engineering the following two new features:

* **Seasonal Trends Indicator:** This feature captures the seasonal patterns in sales data by identifying specific months or quarters where certain products typically experience increased demand. By using historical sales data, we can create binary indicators (0 or 1) for peak seasons (e.g., holidays, back-to-school periods) for each product. This allows the model to better account for seasonal variations and improves accuracy in forecasting during those periods.
* **Sales Lag Feature:** This feature captures the sales performance of a product over the previous few weeks or months. For instance, creating a lagged feature that represents the sales from the last week or the last month can provide the model with information on recent trends. This helps the model recognize patterns in demand, allowing it to make more informed predictions based on how sales have changed recently.

**Link to the GitHub:**