Data Engineer Test: Report

Project Summary

As a Data Engineer, my primary focus was to clean, normalize, and transform supermarket transaction data into a structured format suitable for analysis and machine learning. The dataset spans two years across multiple branches in two provinces, categorized into four item types. I leveraged Python and key libraries such as Pandas, NumPy, Scikit-learn, Matplotlib, and Jupyter to process and analyze the data. JSON-based configurations were used to manage file paths and parameters dynamically, ensuring a flexible and automated workflow. The Linear Regression model was implemented to predict transaction amounts, forming the foundation for data-driven decision-making in retail operations. Below is my step-by-step approach to handling the task.

1. Data Extraction & Loading

- Implemented a file extraction module that downloads and extracts zipped datasets automatically into a raw folder.
- Used file_reader.py module to load raw CSV files into Pandas DataFrames for processing.
- Applied data cleaning functions to standardize and transform datasets.
- Used the file_writer.py module to save cleaned files into a separate directory (data/clean/) for further analysis.
- Configured a config_loader.py to manage file paths and parameters dynamically.

2. Data Cleaning & Preprocessing

To ensure data consistency and usability for analysis and machine learning, I developed custom cleaning functions after conducting Exploratory Data Analysis (EDA) in Jupyter Notebooks. Initially, I explored the data, handled string manipulations, and identified inconsistencies. I then automated the cleaning and transformation process by building generalized functions, which were integrated into Python scripts to efficiently process multiple datasets dynamically.

- Item Data: Cleaned descriptions, standardized item sizes, and extracted unit measurements.
- Sales, Supermarket, and Promotions Data: Renamed columns and handled missing values.

3. Data Transformation & Feature Engineering

I engineered time-based features (month, season) from weekly sales data to capture seasonal trends and merged datasets into a unified structure for machine learning. Missing values were handled by filling defaults or removing unusable records, while categorical variables were encoded using Label Encoding for consistency. To improve model performance, numerical features such as transaction amount and quantity were normalized with MinMaxScaler.

Additionally, promotions were mapped to corresponding sales records to enhance predictive insights.

4. Machine Learning Model for Sales Prediction

The dataset was split into training and test sets, maintaining time-based ordering to preserve trends. A Linear Regression model was trained to predict transaction amounts using historical data. Model performance was evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to assess prediction accuracy. To handle unseen categorical values in the test set, an "UnknownCategory" label was introduced, ensuring robust and consistent model performance.

Challenges

One of the main challenges I faced was the lack of formal training in data science and machine learning. To overcome this, I self-learned key concepts, explored new libraries and functions, and gained hands-on experience with data preprocessing, feature engineering, and model evaluation. This process not only strengthened my technical skills but also enhanced my ability to apply machine learning techniques effectively in a real-world scenario.

Outcome & Next Steps

The project successfully transformed raw retail transaction data into a structured, machine-learning-ready format. The predictive model provides a foundation for sales forecasting, supporting data-driven decision-making in retail operations.
Next steps could include:

- Testing advanced models (e.g., Random Forest, XGBoost) for better accuracy.
- Adding more features like customer segmentation or regional economic indicators.
- Automating model retraining and performance monitoring for real-time predictions.

Supervised Learning Model Task: Sales Prediction

Problem Definition and Objectives

The goal of this supervised learning task is to predict transaction amounts for supermarket sales based on historical data. The business objective is to enable better demand forecasting, optimize inventory, and enhance promotional strategies. By analyzing past transactions, item attributes, promotions, and store locations, the model aims to provide actionable insights into future sales trends.

Chosen Model, Features, and Training Process

The Linear Regression model was chosen for its simplicity, interpretability, and efficiency in numerical prediction tasks. It effectively establishes a relationship between transaction amounts (target variable) and various input features, allowing for straightforward analysis of how

different factors influence sales. This model provides a strong baseline for predicting trends while maintaining computational efficiency and ease of implementation.

Features Used:

- Time-based Features: *Month, season* (engineered from the week variable).
- Item Attributes: *Item type, brand, size* (encoded categorical variables).
- Promotional Features: *Discounts, promotions at different stores*.
- Store Information: *Supermarket attributes affecting sales performance*.
- Sales Metrics: *Quantity of items sold*.

Training Process:

Data Preprocessing: The datasets, including sales, items, promotions, and supermarkets, were merged to create a unified structure for analysis. Categorical variables were converted into a numerical format using Label Encoding, ensuring consistency across features. To maintain a uniform scale, MinMaxScaler was applied to normalize numerical variables. Additionally, new time-based features such as month and season were created to capture temporal trends. Missing values were filled with zeros to preserve data integrity and prevent disruptions in the modeling process.

Data Splitting: The data was split into 80% training and 20% test sets using train_test_split(), ensuring a balanced distribution for model training. Time-based ordering was preserved to prevent data leakage and maintain the integrity of historical sales trends.

Metrics used for analysis of the model's performance:

The model's performance was assessed using the following metrics. These metrics help gauge the model's effectiveness in predicting sales.

- Mean Absolute Error (MAE): Measures the average magnitude of errors.
- Root Mean Squared Error (RMSE): Quantifies prediction deviations with an emphasis on large errors.

Insights and Business Value

1. Categorical Feature Influence
   The model leverages key categorical features like item type, brand, unit of measure (UOM), promotions (display, feature), and location data (postal code, province). These insights help explain how product attributes and in-store promotions impact sales, allowing businesses to fine-tune their strategies.
2. Data Integrity & Preprocessing Success
   - No missing values were found in X_train or X_test, confirming that the data pipeline effectively handled gaps.
   - Proper categorical encoding ensured that all features were numeric, making the dataset model-ready.
3. Model Performance & Sales Prediction

- The Mean Absolute Error (MAE) of 0.6234 indicates that, on average, the model's sales predictions are within 0.62 units of actual values.
- The Root Mean Squared Error (RMSE) of 0.9366 shows that most errors are reasonable, but there's still room for improvement.
- The predicted transaction amount of 8.51 suggests that the model estimates sales figures close to actual trends.

4. Business Implications
    - Better Inventory Management: By understanding how item types, brands, and promotions influence sales, businesses can reduce overstocking.
    - Optimizing Promotions: The feature and display data highlight the impact of promotions, helping businesses refine marketing strategies for better results.
    - Regional Sales Trends: Location-based data (province, postal code) helps pinpoint high-performing areas, enabling businesses to adjust supply chain logistics and distribution more effectively.

Next Steps for Improvement

- Enhance Feature Engineering – Introduce additional factors, like seasonal demand shifts and price elasticity, to improve accuracy.
- Test Advanced Models – Exploring Random Forest or Gradient Boosting could enhance prediction performance.
- Analyze Prediction Errors – Investigating where the model miscalculates sales trends could reveal hidden sales drivers.

Overall, this model serves as a strong foundation for predicting sales transactions, empowering businesses to make data-driven decisions in inventory, marketing, and sales planning.

Conclusion

The supervised learning model successfully integrates historical sales data, promotions, and store attributes to predict transaction amounts. While Linear Regression provides a solid baseline, further improvements could include testing more advanced models like Random Forest or Gradient Boosting for better accuracy.