### **Predicting the Sales of Products of a Retail Chain using Supervised Learning**

This analysis provides a comprehensive overview of the project "Predicting the Sales of Products of a Retail Chain using Supervised Learning," detailing its objectives, methodology, results, technologies used, and the nature of the problem (regression). The project focuses on forecasting product sales for a large Indian retail chain, leveraging supervised learning to enhance inventory management and strategic planning.

#### **Project Background and Objectives**

The project aims to predict future sales for a retail chain operating in Maharashtra, Telangana, and Kerala, stocking diverse product categories such as Fast Moving Consumer Goods (FMCG), eatables/perishables, and others. The primary objective is to develop accurate sales forecasting models to optimize inventory management, reduce operational costs (e.g., overstocking or stockouts), and support data-driven strategic planning. Accurate predictions enable the retail chain to maintain optimal stock levels, improve operational efficiency, and enhance decision-making in a competitive retail environment.

#### **Methodology and Implementation**

The project adopted a structured methodology, following the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework, with detailed steps in data preparation, feature engineering, modeling, and evaluation.

##### **Data Understanding and Preparation**

The project utilized four datasets to build the predictive model:

* **Train Data**: Included columns such as date, product\_identifier, department\_identifier, category\_of\_product, outlet, state, and sales (the target variable, representing the number of sales for a product).
* **Test Data**: Contained similar features (id, date, product\_identifier, department\_identifier, category\_of\_product, outlet, state) but lacked the sales column, used for final predictions.
* **Product Prices**: Provided details like outlet, product\_identifier, week\_id, and sell\_price, indicating the price at which a product was sold in a specific outlet and week.
* **Date to Week ID Map**: Mapped date to week\_id, facilitating temporal analysis.

Key features analyzed included:

* date: Observation date for sales data.
* product\_identifier: Unique identifier for each product.
* department\_identifier: ID for a specific department within a store.
* category\_of\_product: Product category (e.g., FMCG, eatables/perishables, others).
* outlet: Store ID.
* state: State name (Maharashtra, Telangana, Kerala).
* sales: Number of sales (target for training).
* week\_id: Unique identifier for a week.
* sell\_price: Selling price of a product in a specific outlet and week.

The preprocessing steps were comprehensive:

* **Data Merging**: Datasets were merged into a single dataframe using common keys (date, product\_identifier, week\_id, outlet) to create a comprehensive training set, ensuring all relevant information was combined for analysis.
* **Exploratory Data Analysis (EDA)**: Involved datatype conversion, feature extraction from dates (e.g., extracting the month from the date column), and correlation analysis to identify relationships between variables. For instance, high correlation was observed between product\_identifier and department\_identifier, which was noted but both retained for their individual importance.
* **Data Cleaning and Transformation**: The date column was converted to datetime format, and the month was extracted as a new feature, replacing the original date column. Features were ensured to be in appropriate data types (categorical and numerical). No missing values were found in the dataset, simplifying the cleaning process.
* **Feature Selection**: Irrelevant or redundant features were dropped based on correlation analysis. The final features for modeling included product\_identifier, department\_identifier, category\_of\_product, outlet, state, week\_id, sell\_price, and month, with sales as the target variable.
* **Data Splitting**: The dataset was split into training (70%) and testing (30%) sets to ensure robust model evaluation and prevent overfitting.

##### **Modeling**

Several supervised learning models were implemented to predict sales, reflecting a comprehensive approach to model selection:

* **Linear Regression**: A baseline model for predicting continuous outcomes.
* **Decision Tree Regressor**: A tree-based model capturing non-linear relationships.
* **Random Forest Regressor**: An ensemble method combining multiple decision trees, achieving the best performance.
* **K-Nearest Neighbors (KNN) Regressor**: A distance-based model, competitive but less scalable.
* **Gaussian Naive Bayes**: A probabilistic model, less effective for this dataset.

Each model was trained on the training set and evaluated on the testing set. The Random Forest Regressor emerged as the most effective model due to its lowest Root Mean Squared Error (RMSE), attributed to its ability to handle non-linear relationships and its robustness against overfitting, making it suitable for real-world applications.

##### **Evaluation**

The models were evaluated using the following metrics:

* **Root Mean Squared Error (RMSE)**: Measures the square root of the average squared differences between predicted and actual values, emphasizing larger errors.
* **Mean Absolute Error (MAE)**: Measures the average absolute differences between predicted and actual values, providing a linear score of error.
* **R² Score**: Indicates the proportion of variance in the dependent variable that is predictable from the independent variables, with higher values indicating better fit.

The performance comparison is summarized in the following table:

| **Model** | **RMSE (Lower is Better)** | **MAE (Lower is Better)** | **R² (Higher is Better)** |
| --- | --- | --- | --- |
| Linear Regression | Higher | Higher | Lower |
| Decision Tree | Moderate | Moderate | Moderate |
| Random Forest | Lowest | Lowest | Highest |
| KNN | Low | Low | High |
| Gaussian Naive Bayes | Highest | Highest | Lowest |

The Random Forest Regressor had the lowest RMSE, making it the preferred choice due to its scalability, stability, and suitability for real-world use. KNN showed competitive performance but was less preferred due to its computational intensity and sensitivity to data scaling. Gaussian Naive Bayes performed poorly, likely due to its assumption of feature independence, which may not hold for sales data.

##### **Final Model and Prediction**

The Random Forest Regressor was selected as the final model and applied to the test dataset for sales prediction. The test data underwent the same preprocessing steps (merging, date conversion, feature encoding, etc.) to ensure consistency. Predicted sales values were rounded to the nearest integer for practical use (e.g., predicting whole sales units) and added back to the test dataset for final output.

##### **Classification vs. Regression Analysis**

To confirm the problem type:

* **Target Variable**: The target variable sales is a continuous numerical value representing the number of product units sold (e.g., 10, 50). Predicting a continuous outcome is characteristic of regression tasks.
* **Modeling Approach**: The project uses regressors (Linear Regression, Decision Tree Regressor, Random Forest Regressor, KNN Regressor), designed for predicting numerical values. Evaluation metrics like RMSE, MAE, and R² are standard for regression, not classification, which typically uses accuracy or confusion matrices for categorical outcomes.
* **Contrast with Classification**: If the project were predicting a category (e.g., whether sales are "high" or "low"), it would be a classification problem. However, the focus on predicting the exact number of sales confirms it as a regression problem.

Thus, this is a **regression problem**, specifically predicting a continuous numerical outcome.

#### **Key Technologies Used**

The project was executed using Python, leveraging several key libraries:

* **Pandas**: For data manipulation and analysis, particularly for merging datasets and handling dataframes ([Pandas Documentation](https://pandas.pydata.org/docs/)).
* **NumPy**: For numerical computations and array operations ([NumPy Documentation](https://numpy.org/doc/stable/)).
* **Matplotlib and Seaborn**: For data visualization, used in EDA to plot correlations and distributions ([Matplotlib Documentation](https://matplotlib.org/stable/contents.html), [Seaborn Documentation](https://seaborn.pydata.org/)).
* **Scikit-learn**: For machine learning tasks, including model implementation (e.g., RandomForestRegressor, KNeighborsRegressor), data preprocessing (e.g., StandardScaler), and evaluation metrics (e.g., RMSE, MAE, R²) ([Scikit-learn Documentation](https://scikit-learn.org/stable/)).

The environment for execution was Google Colab, providing a cloud-based platform for coding, running, and sharing the notebook, which facilitated collaboration and access to computational resources ([Google Colab](https://colab.research.google.com/)).

#### **Additional Aspects**

* **Data Visualization**: Correlation analysis was visualized using heatmaps created with Matplotlib and Seaborn, helping identify key predictors like product\_identifier and department\_identifier. These visualizations ensured clarity in understanding feature relationships.
* **Feature Importance**: While not explicitly computed using advanced methods, feature importance was inferred from correlation analysis, prioritizing features with stronger relationships to sales.
* **Challenges Faced**:
  + **Data Preprocessing**: Merging multiple datasets required careful alignment of keys to avoid data loss or inconsistencies.
  + **Feature Selection**: Deciding which features to retain based on correlation analysis was challenging, especially for highly correlated features like product\_identifier and department\_identifier.
  + **Evaluation Metrics**: Relying on RMSE, MAE, and R² provided a comprehensive evaluation, but additional metrics like Mean Absolute Percentage Error (MAPE) could enhance interpretability for business applications.
  + **Computational Resources**: Training ensemble models like Random Forest was resource-intensive, mitigated by using Google Colab’s computational capabilities.

#### **Results and Implications**

The project successfully developed a sales prediction model, with the Random Forest Regressor demonstrating the best performance due to its lowest RMSE. This model can be integrated into the retail chain’s inventory management system to forecast sales accurately, reducing costs associated with overstocking or stockouts and supporting strategic planning. The comparative analysis of multiple models ensured the chosen model was robust and reliable, making it suitable for real-world retail applications. The project’s outcomes contribute to improved operational efficiency and data-driven decision-making in the retail sector.