TEBONG ROLAND

DSC680-T301 Applied Data Science (2243-1)

Superkart sales forecast

Superkart sales forecast

1. Business Problem

SuperKart, a renowned supermarket and food mart chain, seeks to enhance its business strategy by accurately predicting future sales across its various outlets. This initiative aims to optimize sales operations in different city tiers and effectively manage inventory levels. To realize this goal, SuperKart has engaged a data science firm, providing them with last quarter's sales data from multiple outlets. The firm's task is to develop a predictive model that estimates the total sales for each store in the upcoming quarter.

2. Background/History

Sales forecasting is pivotal in projecting future revenue by analyzing past sales, industry trends, and current sales pipeline status. It's a critical tool for estimating sales over different timeframes (weekly, monthly, quarterly, yearly), enabling organizations to plan and strategize effectively. Accurate sales forecasts offer numerous benefits, including enhanced decision-making, reduced risks related to sales pipeline and forecasts, more efficient territory coverage planning, and establishing benchmarks for future trend analysis. Such forecasts are integral to coordinating sales operations across regions and informing the supply chain team about procurement needs..

3. Data Explanation

- **Data Loading:** The analysis begins with importing data from the 'SuperKart.csv' file using panda’s library. This dataset contains 8,763 records across 12 columns. The primary focus is on the 'Product_Store_Sales_Total' variable, representing the total sales for each store, which we'll predict for the next quarter. The remaining variables serve as predictors for this target variable.
- **Data Dictionary:** The dataset encompasses various attributes of products and stores. The detailed data dictionary is given below.
    1. Product_Id: A unique identifier for each product, starting with two letters followed by a number.
    2. Product_Weight: Weight of each product.
    3. Product_Sugar_Content: Sugar content category for each product (e.g., low sugar, regular, no sugar).
    4. Product_Allocated_Area: Ratio of a product's display area to the total display area in a store.
    5. Product_Type: General category of each product (e.g., meat, snack foods, drinks, dairy, hygiene products).
    6. Product_MRP: Maximum retail price for each product.
    7. Store_Id: Unique identifier for each store.
    8. Store_Establishment_Year: Year when the store was established.
    9. Store_Size: Size classification of the store (e.g., high, medium, low square footage).

10. Store_Location_City_Type: City tier classification based on living standards (Tier 1, Tier 2, Tier 3).
11. Store_Type: Type of store based on the range of products sold (e.g., Departmental Store, various Supermarket Types, Food Mart).
12. Product_Store_Sales_Total: Total revenue from sales of each product in a specific store.

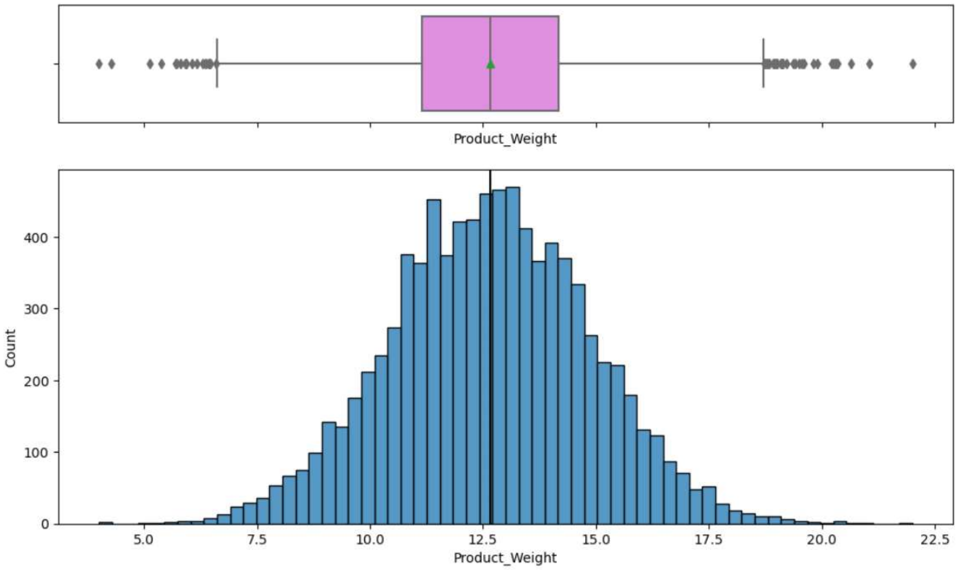
4. Methods & Analysis

4.1 Exploratory Data Analysis (EDA):

Let's check the statistical summary of the data.

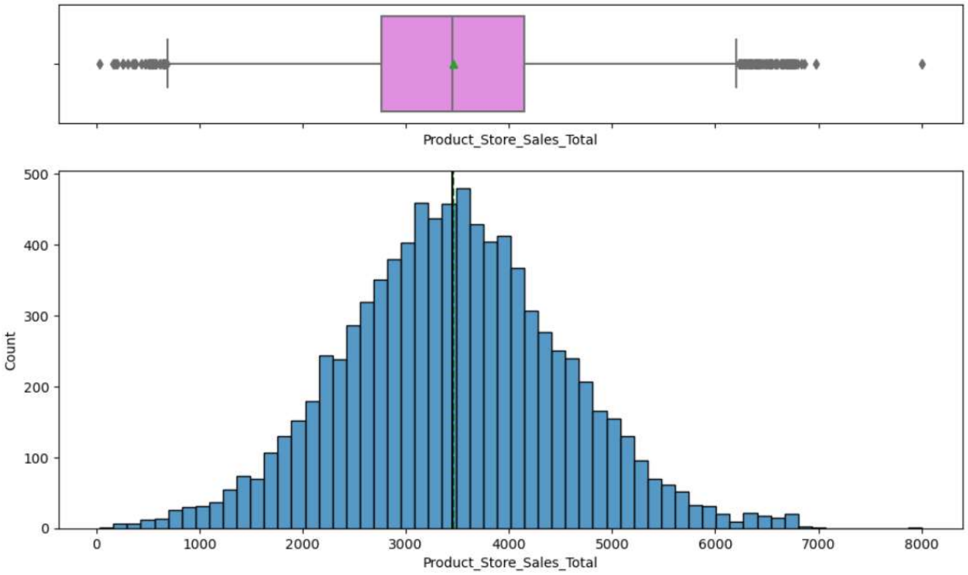
1) There are 16 different product types and fruits and vegetables have been sold the highest number of times (1249).
2) There are 4 unique stores in the dataset.
3) The revenue generated from the sale of a particular product at a certain outlet varies from 33 to 8000 with 50% of the values lying above 2762.
4) The 75th percentile of Store_Id is 0. It indicates that the vast majority of these stores doesn't have unique identifier.
5) The mean store sales is approx. USD 34,640, whereas the median of the store sales is approx. USD 34,523. This indicates that the Product_Store_Sales_Total distribution is only slightly skewed towards the right side.

After reviewing the statistical summary of numeric variables, let’s delve into the univariate analysis, which reveals the most useful patterns in our dataset. The histogram plot of Product_Weight variable is shown below:



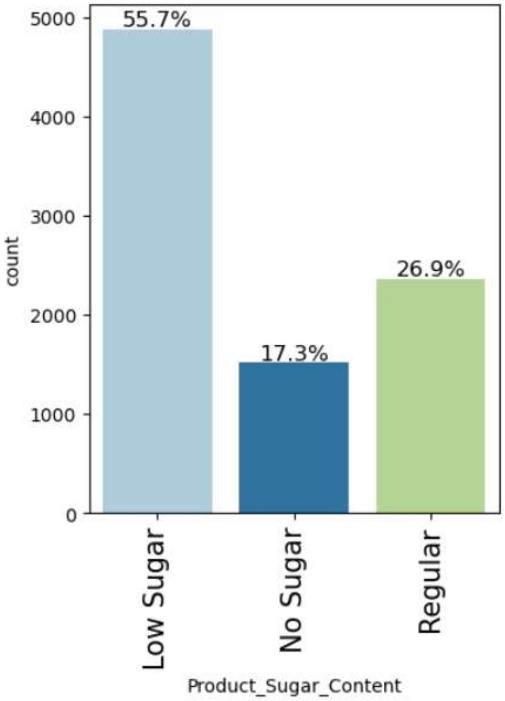
- The product weight is uniformly distributed with mean and median lying around 12.5.

As our target variable is Product_Store_Sales_Total, let's observe the distribution of the target variable as shown below:



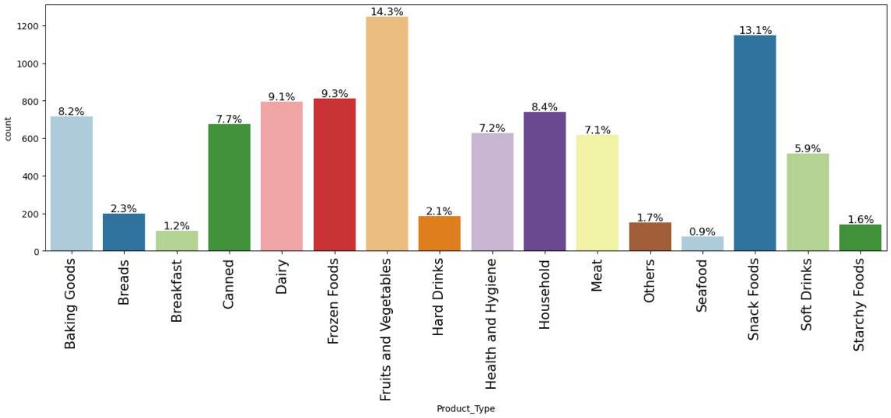
- The revenue generated from each product at a particular store is normally distributed with mean and median lying around 3500.

Let’s observe the Product_Sugar_Content variable as shown below:



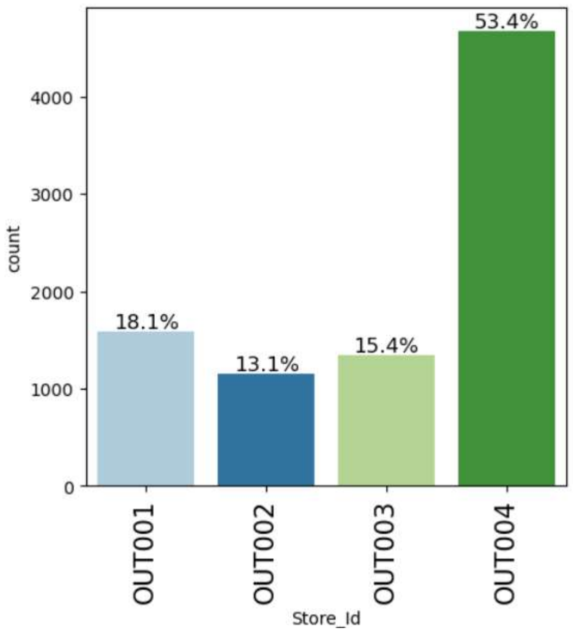
- Around 56% of the products are having low sugar followed by 27% products which are having regular sugar content.
- Around 17% of the products are having no sugar content.

Furthermore, it’s important to visualize the each product type category as shown below, which as important aspect of our dataset.



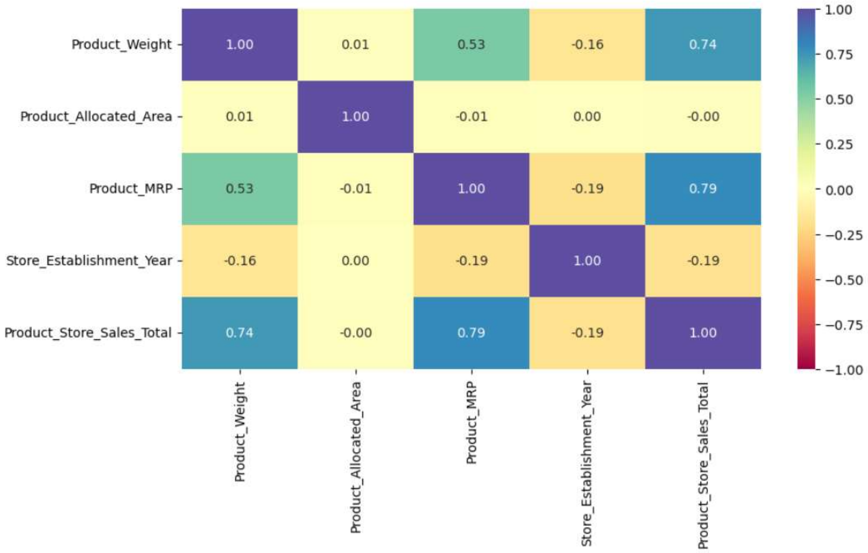
- Fruits and vegetables (14%) and Snack Foods (13%) have been bought the highest number of times from all the stores combined.
- Seafood (1%) has been bought the lowest number of times.
- The highest product type which is Fruits and Vegetables is 14 times of the lowest product type which is sea food.

Because our dataset has 4 different types of store outlets it's important to observe the sale from each outlet as shown below:



- Around 53% of the products are being sold from outlet OUT004. Almost equal number of products have been sold from the other three stores each.
- When compared to the four Store_Ids the products which are being sold from outlet OUT002 is low.

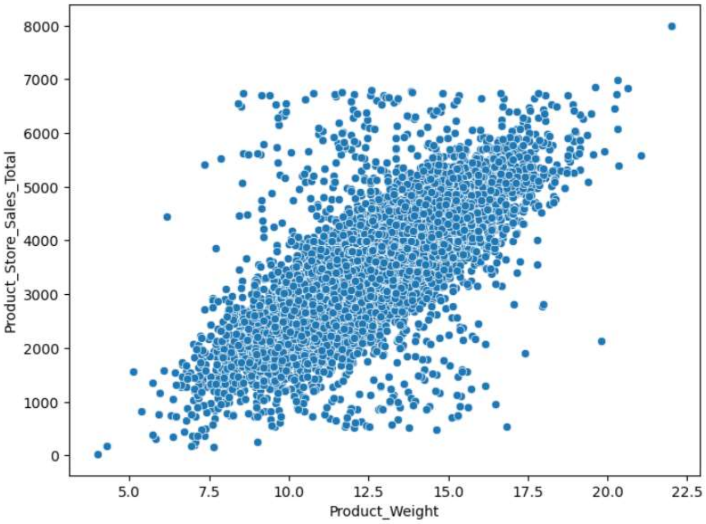
Now let's dive into the bivariate analysis to see the relationship among different variables. The heat map is shown below:



- Product weight and product MRP are highly correlated with our target variable i.e Product_Store_Sales_Total.
- Product weight and product MRP are moderately correlated with each other.
- There is not much correlation among the rest of the variables.

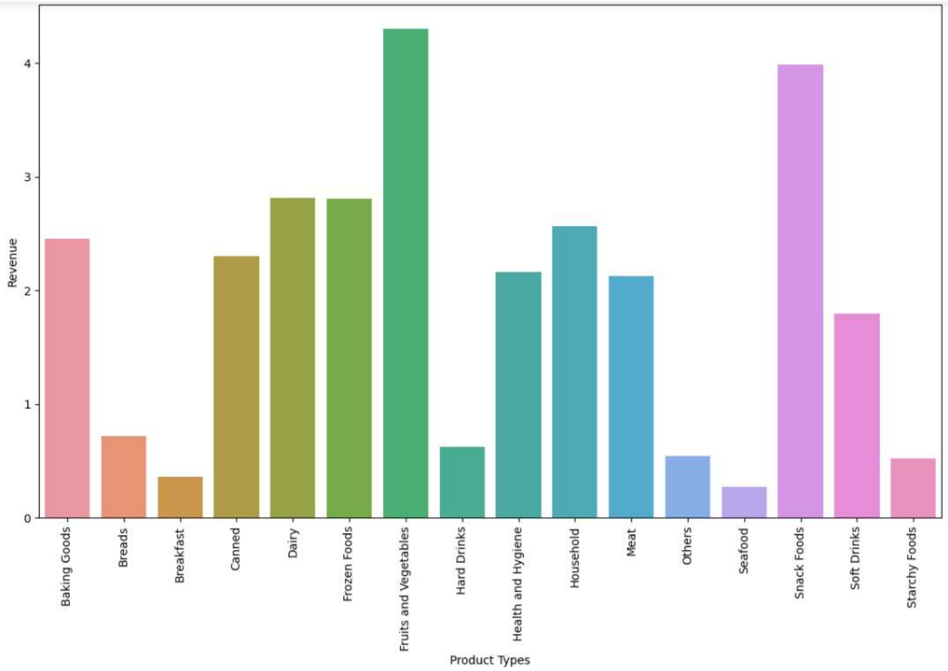
- Store_Establishment_Year is highly negatively correlated with our target variable i.e Product_Store_Sales_Total.

Let's check the distribution of our target variable i.e Product_Store_Sales_Total with the numeric columns.



- Product_Weight and Product_Store_Sales_Total are almost linearly correlated with each other.

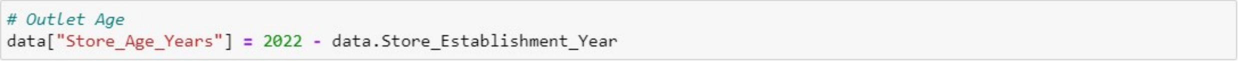
Let us see from which product type the company is generating most of the revenue.



- Fruits and vegetables and snack foods are the biggest contributors to the revenue of the company.
- Seafood's are the lowest contributors to the revenue of the company.
- Dairy and Frozen foods are contributing almost same to the revenue of the company.

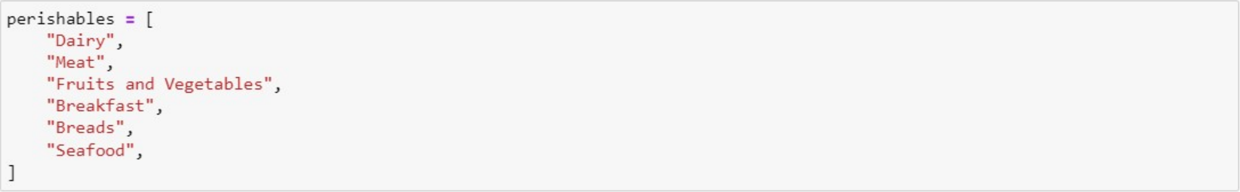
4.2 Data Preprocessing:

A store which has been in the business for a long duration is more trustworthy than the newly established ones. On the other hand, older stores may sometimes lack infrastructure if proper attention is not given. So let us calculate the current age of the store and incorporate that in our model.

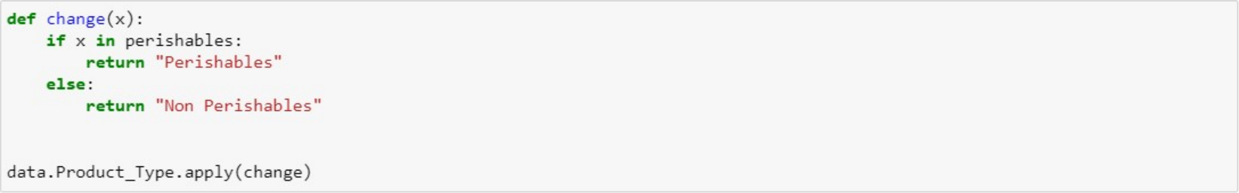


```
# Outlet Age
data["Store_Age_Years"] = 2022 - data.Store_Establishment_Year
```

We have 16 different product types in our dataset. So let us make two broad categories, perishables and non-perishables, in order to reduce the number of product types.



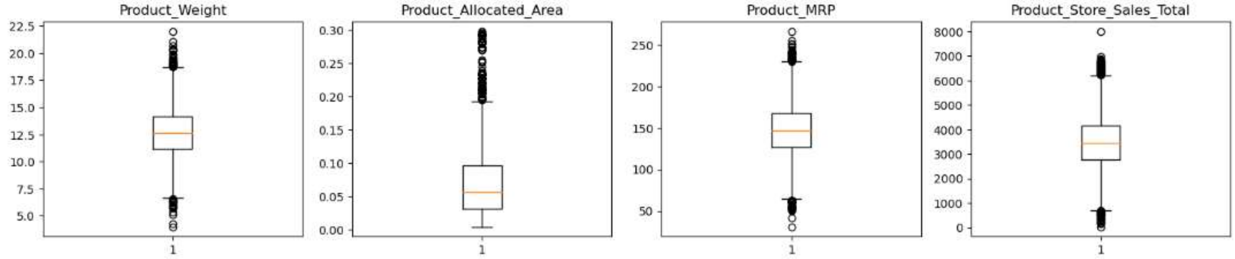
```
perishables = [
    "Dairy",
    "Meat",
    "Fruits and Vegetables",
    "Breakfast",
    "Breads",
    "Seafood",
]
```



```
def change(x):
    if x in perishables:
        return "Perishables"
    else:
        return "Non Perishables"


data.Product_Type.apply(change)
```

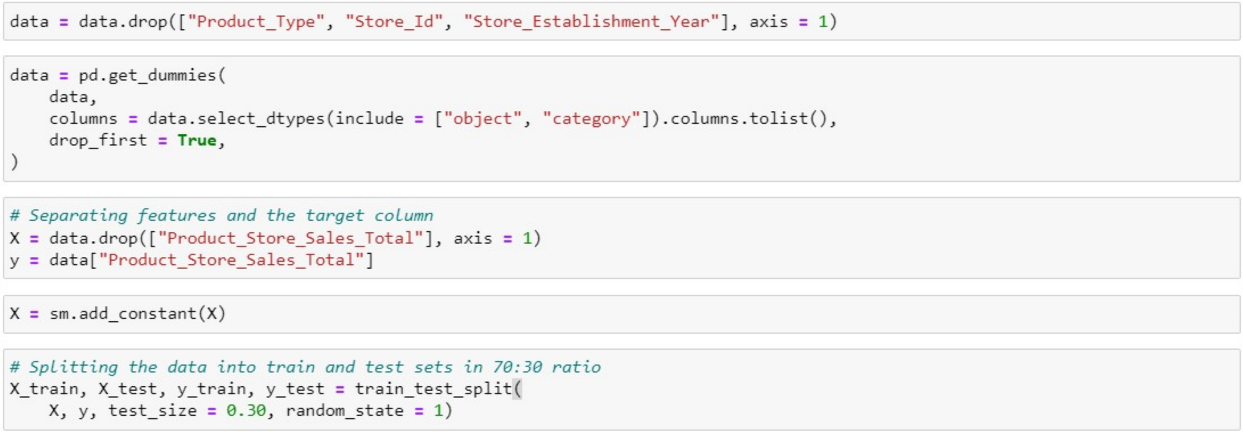
Let's check for outliers in the data.



- There are quite a few outliers in the data.
- However, we will not treat them as they are proper values.

So far we have done lots of exploratory data analysis let’s prepare the data for machine learning models.

- We want to forecast the Product_Store_Sales_Total.
- Let’s encode categorical features and drop the unnecessary columns
- Let’s split the data into train and test data.



```
data = data.drop(["Product_Type", "Store_Id", "Store_Establishment_Year"], axis = 1)
```

```
data = pd.get_dummies(
    data,
    columns = data.select_dtypes(include = ["object", "category"]).columns.tolist(),
    drop_first = True,
)
```

```
# Separating features and the target column
X = data.drop(["Product_Store_Sales_Total"], axis = 1)
y = data["Product_Store_Sales_Total"]
```

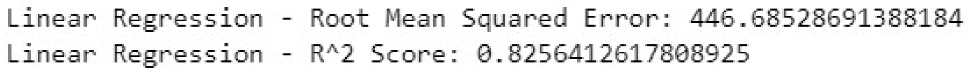
```
X = sm.add_constant(X)
```

```
# Splitting the data into train and test sets in 70:30 ratio
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size = 0.30, random_state = 1)
```

4.3 Modeling

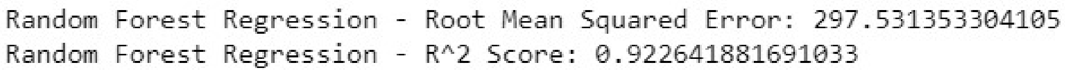
Let's build Linear Regression and Random Forest Regressor.

Linear Regression: After fitting the training data into the linear regression model the model has been tested on the test dataset, which was splitted above in the data preprocessing section. After testing the model the results are quite promising as shown below.



```
Linear Regression - Root Mean Squared Error: 446.68528691388184
Linear Regression - R^2 Score: 0.8256412617808925
```

Random Forest Regressor: After fitting the training data into the linear regression model the model has been tested on the test dataset, which was splitted above in the data preprocessing section. After testing the model the results are quite promising as shown below.



```
Random Forest Regression - Root Mean Squared Error: 297.531353304105
Random Forest Regression - R^2 Score: 0.922641881691033
```

- Lower RMSE: The Random Forest model's lower RMSE compared to the Linear Regression model indicates that it is making more accurate predictions. Since RMSE penalizes larger errors more severely, this also suggests that the Random Forest model has fewer large errors in its predictions than the Linear Regression model.
- Higher R^2 Score: An R^2 score of over 90% is typically considered very high and indicates that the model's predictions closely match the observed data.
- Model Comparison: When comparing the two models, the Random Forest Regression not only has a higher R^2 score but also a lower RMSE, which generally suggests that it is the better model for this problem.

5. Assumptions

The analysis operates under the assumption that past sales data is a reliable predictor of future trends, assuming there are no major shifts in customer behavior that our data doesn't capture. We also assume that the performance of the stores will stay stable over time, except for any unaccounted changes due to macroeconomic factors or local developments. Additionally, when categorizing products into perishable and non-perishable groups, we recognize that this broad classification may not fully capture the intricate variations in sales patterns that more detailed categories might reveal.

6. Limitations

The models we're using are largely dependent on past data, and there's a significant risk that they might not accurately predict upcoming market trends or changes in customer preferences. Both Linear Regression and Random Forest Regressor, the methods we're employing, come with their own set of limitations. For instance, Linear Regression works on the assumption that there's a straight-line relationship between variables, which isn't always the case in real-world scenarios. Moreover, any errors or gaps in the data can result in incorrect predictions.

7. Challenges

Understanding and modeling the complex, non-linear relationships between various factors (like product type, store location, and sales) was challenging. Categorizing products into perishable and non-perishable groups required careful consideration to avoid oversimplification. Managing large datasets and running computationally intensive models like Random Forest Regressor posed practical challenges, especially in optimizing for performance and speed.

8. Future Uses/Additional Applications:

Extend the model to predict inventory requirements, reducing overstocking and stock outs. Combine sales data with customer demographic and purchase behavior data to tailor marketing strategies. Use the model to assess potential sales and viability for new store locations or for expanding into new markets.

10. Recommendations:

- Prioritize inventory and marketing efforts on high-revenue products like fruits, vegetables, and snack foods.
- Develop targeted strategies for each store based on their unique sales patterns and customer demographics.

- Implement dynamic pricing strategies for products with high elasticity, identified through the analysis.

11. Implementation Plan

Train SuperKart staff on the insights and tools developed. Begin implementation of targeted marketing and inventory strategies based on the analysis. Monitor and adjust strategies based on real-time sales data. Start integrating customer behavior analysis for a more personalized shopping experience. Evaluate the impact of implemented strategies on sales and inventory efficiency. Explore potential for model application in new store location assessments and wider market expansion strategies.

12. Ethical Assessment:

- Ensure that customer and sales data are handled in compliance with data privacy laws and ethical standards.
- Regularly assess the models for potential biases, especially in recommendations that might favor certain stores or regions over others.
- Maintain transparency with stakeholders about how data is used in decision-making processes, especially in strategies that might affect employees or customers.

References:

1. Bohanec, Marko, Mirjana Kljajić Borštnar, and Marko Robnik-Šikonja. "Explaining machine learning models in sales predictions." *Expert Systems with Applications* 71 (2017): 416-428.
2. Bajaj, Purvika, et al. "Sales prediction using machine learning algorithms." *International Research Journal of Engineering and Technology (IRJET)* 7.6 (2020): 3619-3625.
3. Odegua, Rising. "Applied Machine Learning for Supermarket Sales Prediction." *Project: Predictive Machine Learning in Industry* (2020).
4. Tsoumakas, Grigorios. "A survey of machine learning techniques for food sales prediction." Artificial Intelligence Review 52.1 (2019): 441-447.