STAT 463 FINAL PROJECT

Group 5

Chau Phan

Dylan Nguyen

TABLE OF CONTENTS

BIG MART SALES DATASET

The Big Mart Sales dataset contains **8,523 rows** of sales records across various products and outlets. It includes features like item type, price, visibility, and store characteristics.

The goal is to predict product sales (**OutletSales, in thousands of dollars per Product)** using regression models.

This dataset is ideal for practicing data cleaning, feature engineering, and sales forecasting in a retail context.

1 DATA EXPLORATION

2 DATA PREPARATION

3 MODEL BUILDING

4 MODEL EVALUATION

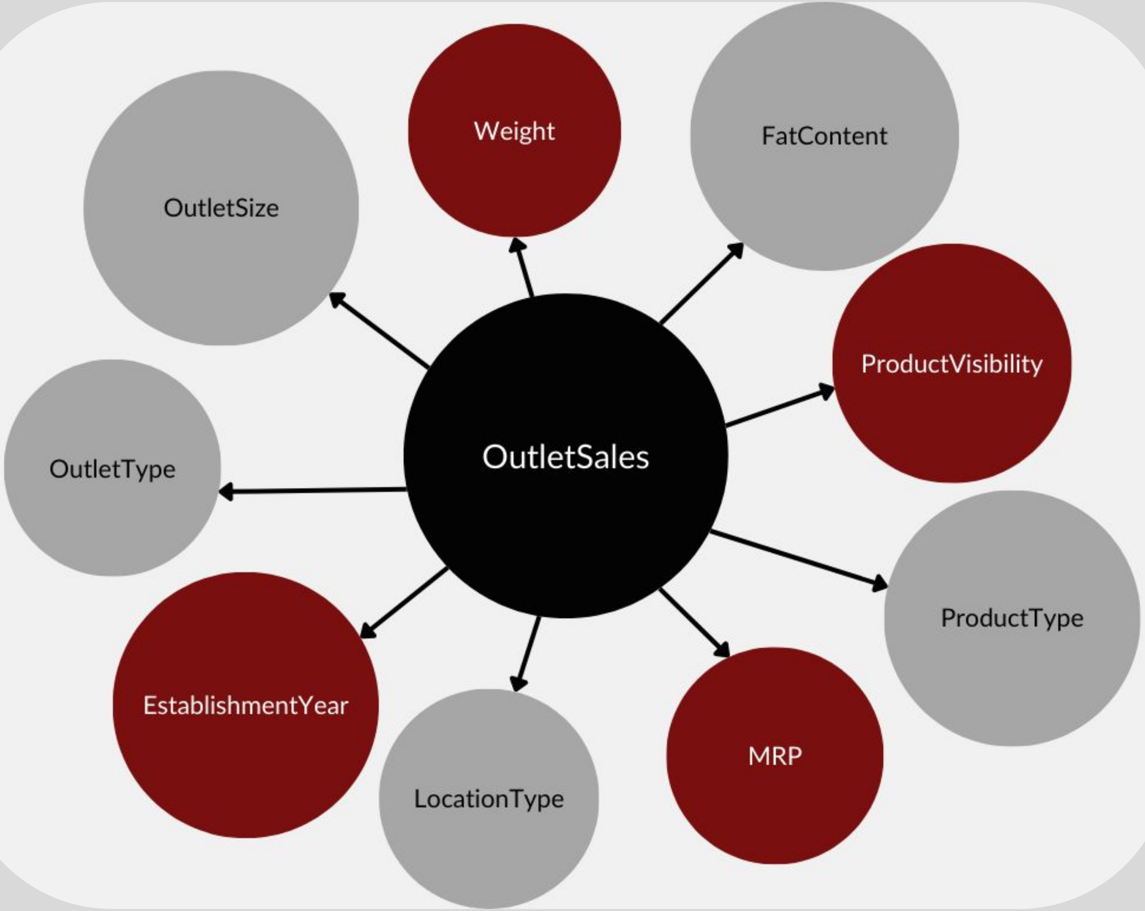
5 CONCLUSION/ RECOMMENDATIONS

VARIABLES

quantitative variables

categorical variables

response



DATA PREPARATION

Cleaning: Removed NAs, encoded categoricals, numeric formatting on both train_set and test_set.

```r
# remove NA values - imputation
train_set <- train_set %>%
  filter(!is.na(Weight))
test_set <- test_set %>%
  filter(!is.na(Weight))
```

```r
train_set$Weight <- as.numeric(train_set$Weight)
train_set$ProductVisibility <- as.numeric(train_set$ProductVisibility)
train_set$MRP <- as.numeric(train_set$MRP)
train_set$EstablishmentYear <- as.numeric(train_set$EstablishmentYear)
train_set$OutletSales <- as.numeric(train_set$OutletSales)

test_set$Weight <- as.numeric(test_set$Weight)
test_set$ProductVisibility <- as.numeric(test_set$ProductVisibility)
test_set$MRP <- as.numeric(test_set$MRP)
test_set$EstablishmentYear <- as.numeric(test_set$EstablishmentYear)
test_set$OutletSales <- as.numeric(test_set$OutletSales)
```

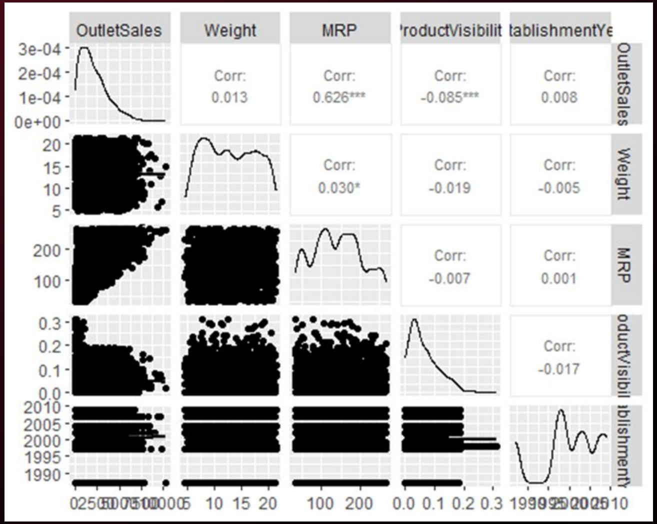
DATA PREPARATION

```
# For test_set
test_set <- test_set %>%
  mutate(FatContent = recode_factor(FatContent, 'LF' = 'Low Fat', 'low fat' =
'Low Fat', 'reg' = 'Regular')) %>%
  mutate(ProductType = recode_factor(ProductType,
                                     'Baking Goods' = 'Baking Goods',
                                     'Fruits and Vegetables' = 'Fruits and
Vegetables',
                                     'Household' = 'Household',
                                     'Dairy' = 'Dairy',
                                     'Hard Drinks' = 'Hard Drinks',
                                     'Frozen Foods' = 'Frozen Foods',
                                     'Snack Foods' = 'Snack Foods',
                                     'Canned' = 'Canned',
                                     'Meat' = 'Meat',
                                     'Health and Hygiene' = 'Health and
Hygiene',
                                     'Soft Drinks' = 'Soft Drinks',
                                     'Starchy Foods' = 'Starchy Foods',
                                     'Breakfast' = 'Breakfast',
                                     'Others' = 'Others',
                                     'Breads' = 'Breads',
                                     'Seafood' = 'Seafood')) %>%
  mutate(OutletSize = recode_factor(OutletSize, 'Small' = 'Small', 'Medium' =
'Medium', 'High' = 'High')) %>%
  mutate(LocationType = recode_factor(LocationType, 'Tier 1' = 'Tier 1',
'Tier 2' = 'Tier 2', 'Tier 3' = 'Tier 3')) %>%
  mutate(OutletType = recode_factor(OutletType,
                                    'Supermarket Type1' = 'Supermarket
Type1',
                                    'Grocery Store' = 'Grocery Store',
                                    'Supermarket Type2' = 'Supermarket
Type2',
                                    'Supermarket Type3' = 'Supermarket
Type3'))
```

```
# For train_set
train_set <- train_set %>%
  mutate(FatContent = recode_factor(FatContent, 'LF' = 'Low Fat', 'low fat' =
'Low Fat', 'reg' = 'Regular')) %>%
  mutate(ProductType = recode_factor(ProductType,
                                     'Baking Goods' = 'Baking Goods',
                                     'Fruits and Vegetables' = 'Fruits and
Vegetables',
                                     'Household' = 'Household',
                                     'Dairy' = 'Dairy',
                                     'Hard Drinks' = 'Hard Drinks',
                                     'Frozen Foods' = 'Frozen Foods',
                                     'Snack Foods' = 'Snack Foods',
                                     'Canned' = 'Canned',
                                     'Meat' = 'Meat',
                                     'Health and Hygiene' = 'Health and
Hygiene',
                                     'Soft Drinks' = 'Soft Drinks',
                                     'Starchy Foods' = 'Starchy Foods',
                                     'Breakfast' = 'Breakfast',
                                     'Others' = 'Others',
                                     'Breads' = 'Breads',
                                     'Seafood' = 'Seafood')) %>%
```

Cleaning: Removed NAs, encoded categoricals, numeric formatting on both train_set and test_set.

DATA EXPLORATION

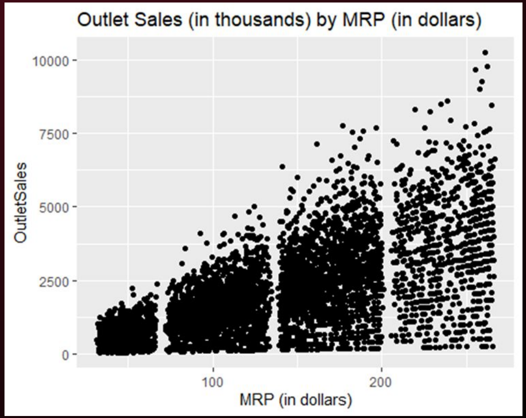


We created a scatter plot matrix and other explorations between predictors and response in order to see the general relationships that we may want to explore later with regression analysis.

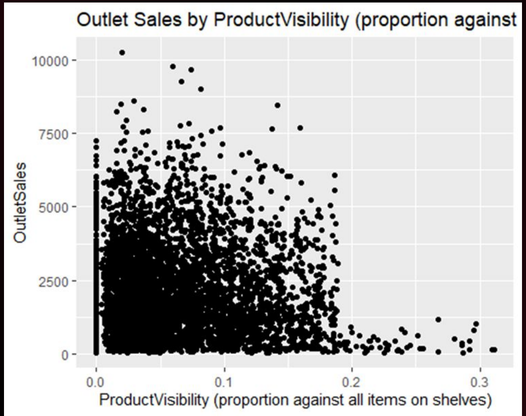
DATA EXPLORATION

Outlet Sales Response vs Individual Predictors:

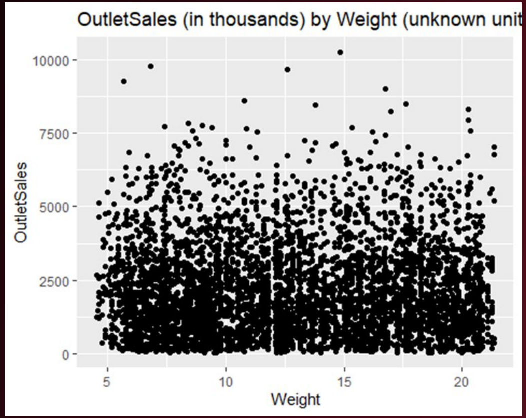
- MRP (Maximum Retail Price in $)
- Product Visibility
- Weight Predictors



Maximum Retail Price
(MRP in $)



ProductVisibility
(by proportion)

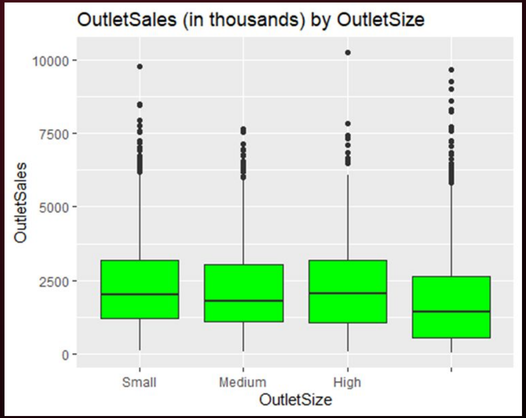


Weight
(Unknown Units)

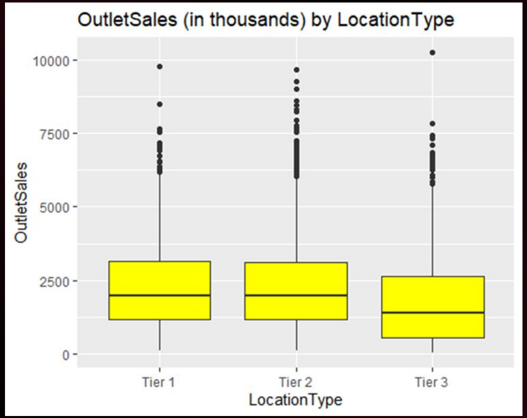
DATA EXPLORATION

Outlet Sales Response vs Individual Predictors:

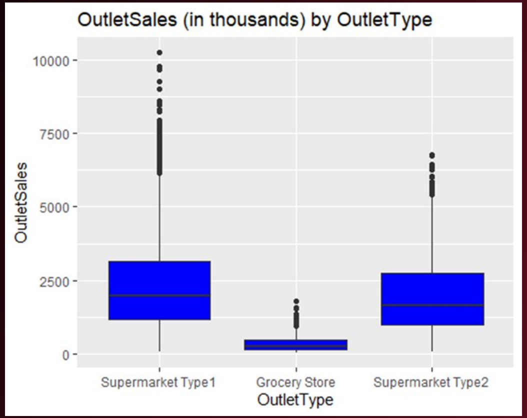
- Outlet Size (arbitrary)
- Location Type (arbitrary tiers)
- Weight Predictors



OutletSize (arbitrary)



Location Type (arbitrary tiers)

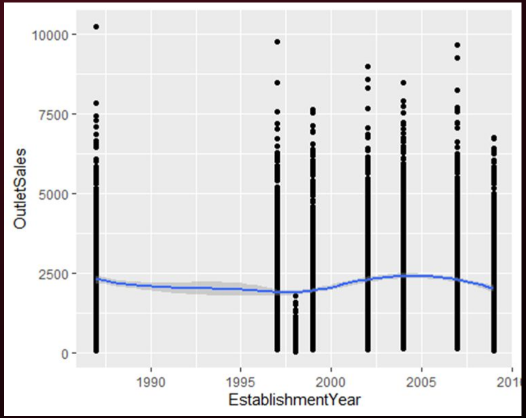


Outlet Type (supermarket type)

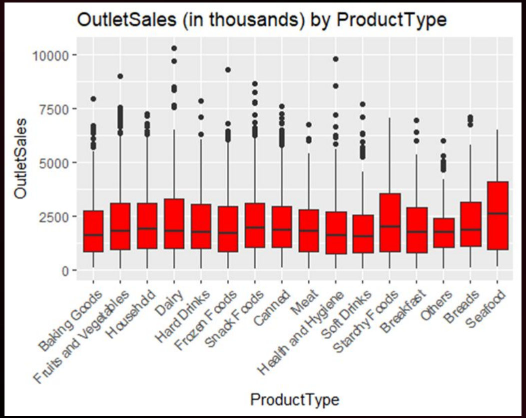
DATA EXPLORATION

Outlet Sales Response vs Individual Predictors:

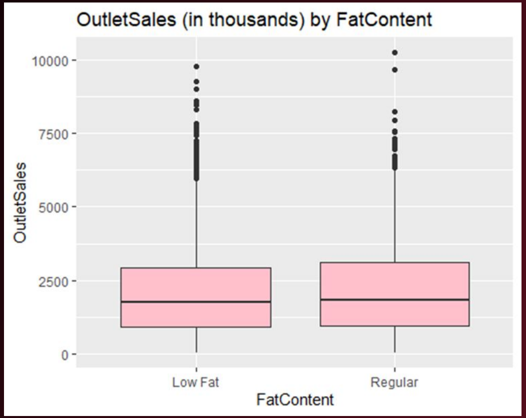
- Establishment Year
- Product Type (seafood, drinks, produce, etc.)
- Fat Content



Establishment Year

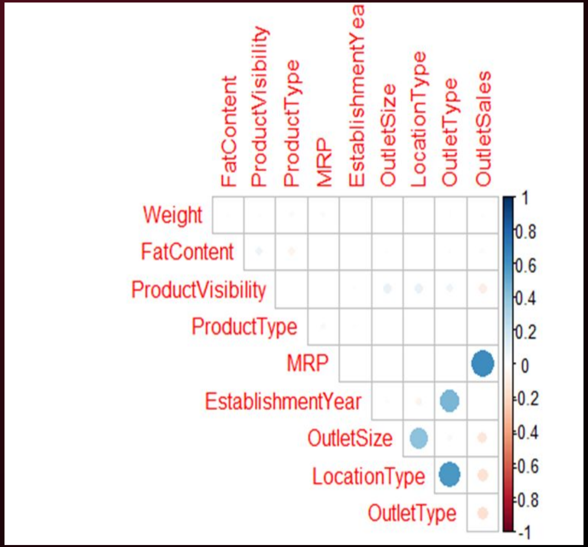


Product Type (seafood, drinks, produce, etc.)



Fat Content

DATA EXPLORATION



- Check multicollinearity for all transformations and features
- Ensure that predictors are effectively "doing their own job" or contributing to the model in their own way.

VIF values < 10 = Good!

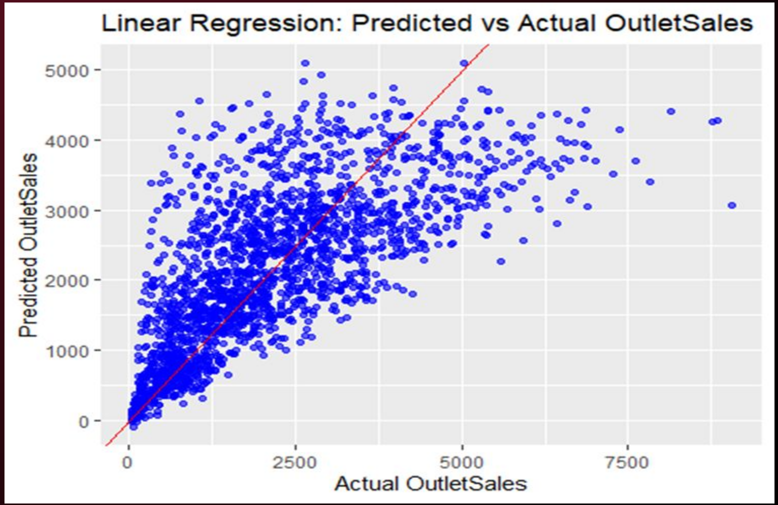
	VIF <dbl>
Weight	1.020760
FatContentRegular	1.223233
ProductVisibility	1.056084
ProductTypeFruits and Vegetables	2.565595
ProductTypeHousehold	2.360316
ProductTypeDairy	1.928924
ProductTypeHard Drinks	1.364494
ProductTypeFrozen Foods	2.110778
ProductTypeSnack Foods	2.500517
ProductTypeCanned	1.881521
ProductTypeMeat	1.541225
ProductTypeHealth and Hygiene	1.850724
ProductTypeSoft Drinks	1.656593
ProductTypeStarchy Foods	1.234980
ProductTypeBreakfast	1.166535
ProductTypeOthers	1.254662
ProductTypeBreads	1.358170
ProductTypeSeafood	1.090114
LocationTypeTier 2	2.113173
LocationTypeTier 3	3.380596
poly(MRP, 6)1	1.015241
poly(MRP, 6)2	1.014478
poly(MRP, 6)3	1.019447
poly(MRP, 6)4	1.018052
poly(MRP, 6)5	1.009908
poly(MRP, 6)6	1.010726
poly(EstablishmentYear, 2)1	1.298671
poly(EstablishmentYear, 2)2	2.271473

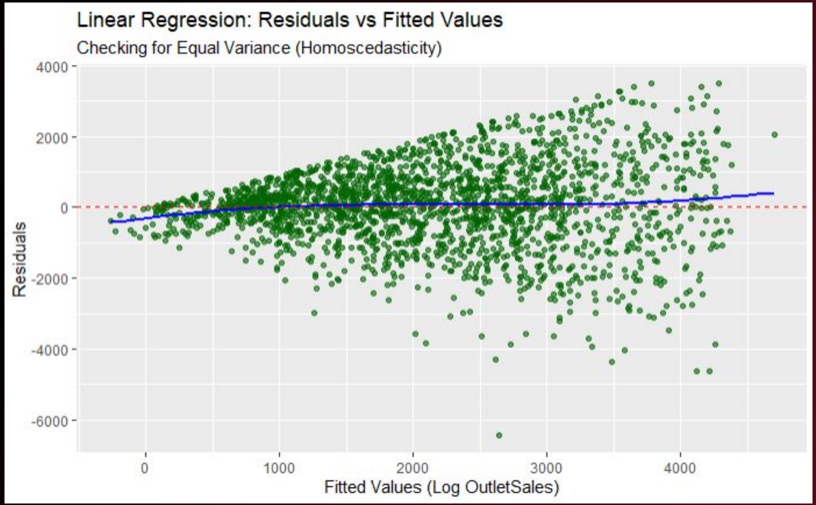
Model with polynomial predictors

	VIF <dbl>
Weight	1.018883
FatContentRegular	1.219977
ProductVisibility	1.019271
ProductTypeFruits and Vegetables	2.557385
ProductTypeHousehold	2.342842
ProductTypeDairy	1.918911
ProductTypeHard Drinks	1.351354
ProductTypeFrozen Foods	2.099249
ProductTypeSnack Foods	2.487738
ProductTypeCanned	1.878941
ProductTypeMeat	1.539136
ProductTypeHealth and Hygiene	1.837589
ProductTypeSoft Drinks	1.638864
ProductTypeStarchy Foods	1.233139
ProductTypeBreakfast	1.163154
ProductTypeOthers	1.252755
ProductTypeBreads	1.357668
ProductTypeSeafood	1.086329
MRP	1.014958
EstablishmentYear	1.285868
LocationTypeTier 2	1.778327
LocationTypeTier 3	1.517195

Model w/ no Transformations

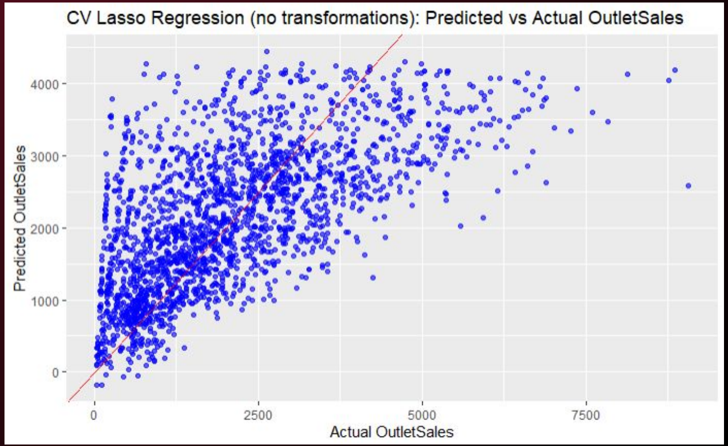
MODEL BUILDING – LINEAR REGRESSION

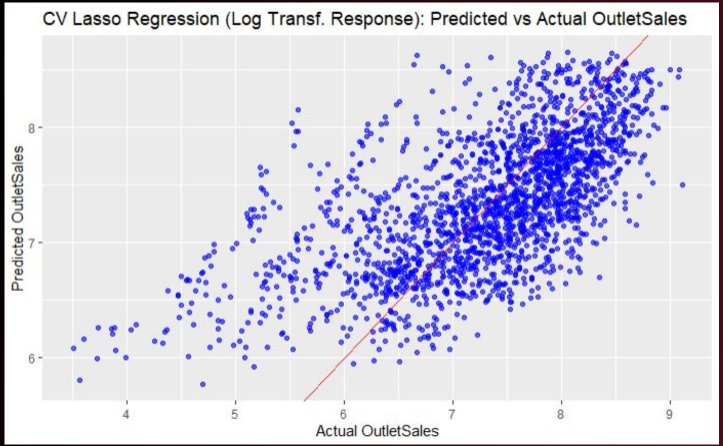




- RMSE 1153, Test R^2 0.41
- Easy to understand, gives a clear view of how each feature affects sales.

MODEL BUILDING

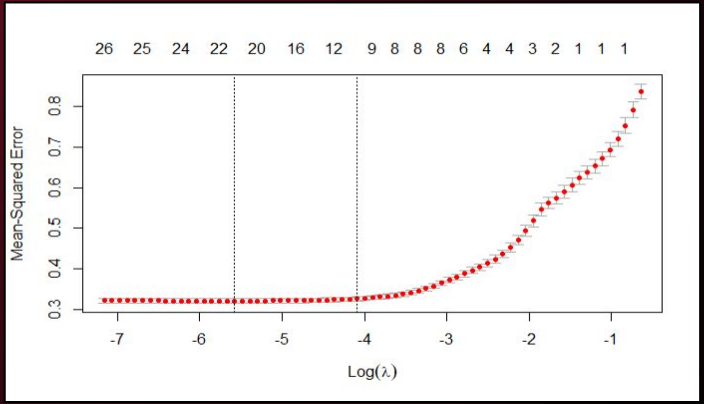


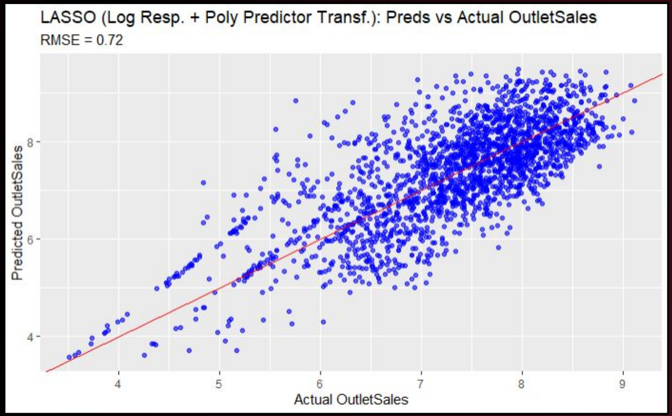


- Lasso v1: RMSE 1149, Test R^2 0.41
- Slightly improved the model, no significant change.

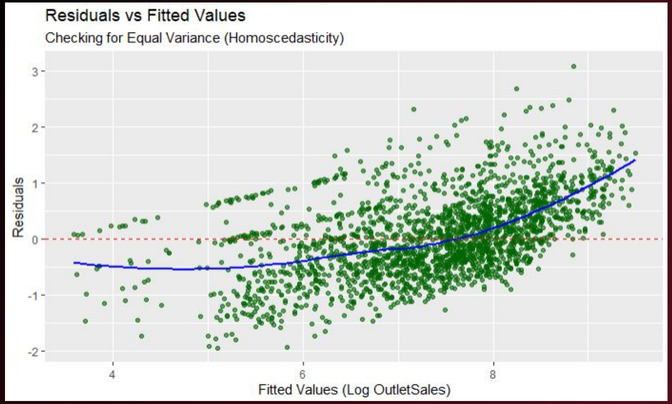
- Lasso v2 (Log Response): RMSE 0.72, Test R^2 0.405
- Slightly improved the model, no significant change.

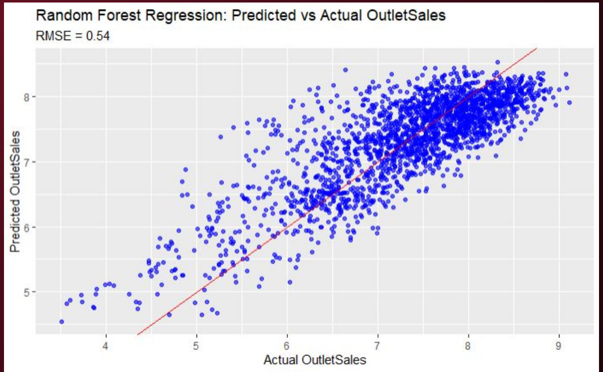
MODEL BUILDING



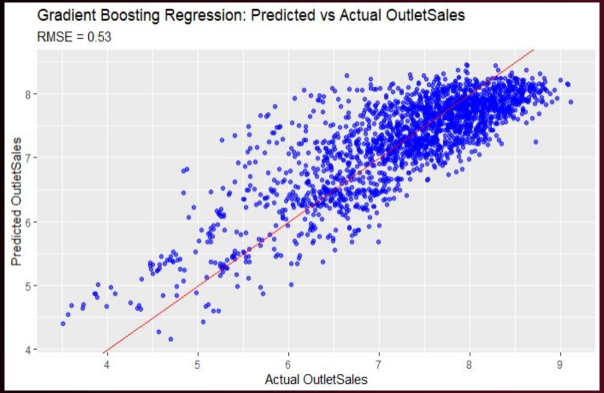


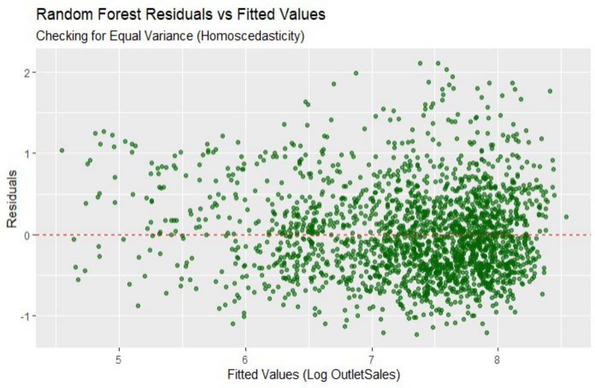
- 3rd Lasso Model (Log + Poly): RMSE 0.721, Test R^2 0.405
- Similar to linear but also removes less important features automatically.



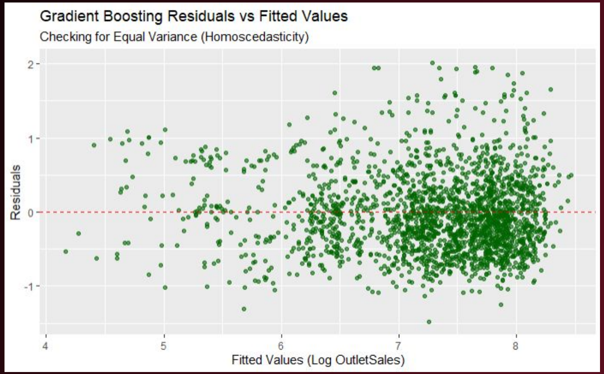


Normal QQ Plot Residuals





Equal Variance
QQ Plot Residuals



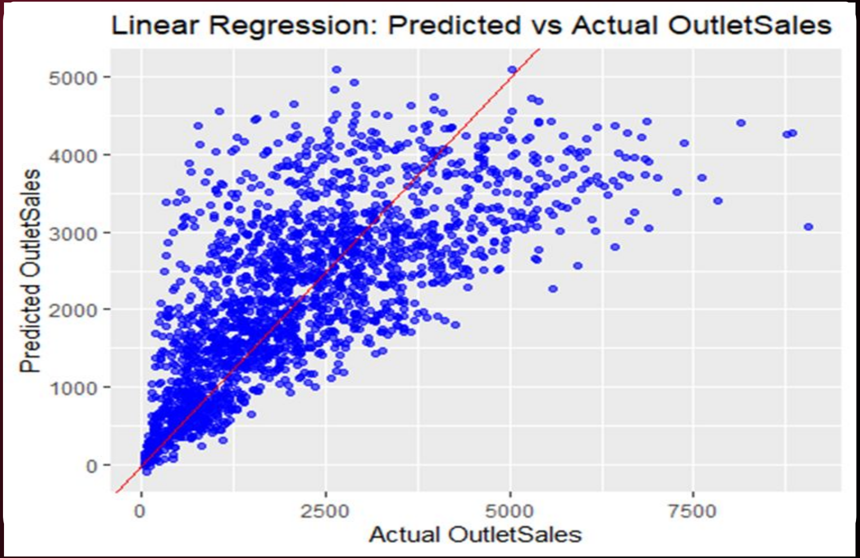
- Random Forest: RMSE 0.54, Test R^2 0.66
- More accurate, but very complex — hard to explain why predictions are made.

- Boosting Regression: lowest RMSE 0.53, Test R^2 0.68
- Best predictive accuracy, low interpretability.

Model Selection & Evaluations

- **Lasso Regression** was meant to reduce unnecessary features.
  ➢ But it didn't improve performance much.
  ➢ It didn't remove many predictors either — not much gained.
- **Gradient Boosting** had the best accuracy (lowest RMSE, highest R^2)
  ➢ But it's a black box — hard to explain how it makes decisions.
  ➢ Not ideal for business users who want to understand what drives sales
  ➢ Also better suited for time-based data, which we don't have.

- **Random Forest** handled non-linear patterns better
  ➢ But like boosting, it lacks transparency for stakeholders.



Model Selection & Evaluations

Why We Chose Linear Regression?

✅ Performs almost as well as Lasso
✅ Easy to interpret — shows exactly how each factor (like MRP) affects sales
✅ Clear predictor insights help decision-makers take action
✅ No multicollinearity issues (VIF < 3 for all predictors)
❌ Non-linearity issues exist, but interpretability is more important for this case

Model Selection & Evaluations

Most Significant Predictors:	Other Significant Predictors:	Insignificant Variables (or Highly Collinear):
• **Maximum Retail Price** (MRP) • **Product Visibility** (by Proportion of all Products)	• ProductType • Establishment Year • Location Type	• Outlet Size • Outlet Type • Weight • Fat Content (Regular or Low Fat)

Recommendations

01 Focus on High-MRP Products

➢ Products with higher Maximum Retail Price (MRP) are strongly linked to higher sales.

➢ Consider promoting premium items or adjusting pricing strategies for key products.

02 Boost Visibility of Low-Performing Products

➢ Product Visibility has a positive impact — items that are more visible tend to sell more.

➢ Improve shelf placement, in-store signage, and digital visibility for underperforming items.

03 Leverage Top-Performing Product Types

➢ Some categories, like **seafood**, significantly boost sales.

➢ Expand inventory or promotions around these high-performing product types.

Thank you