Adidas
Sales & Price analysis

During the Pandemic

Bin, Rahull, Hayoung, Mayank, Meenakshi

Dataset Overview

US Adidas Monthly Sales Dataset For 2020-2021

9600 Observations

(From 2020 to 2021)

6 Products

52 Cities

50 States

3 Sales Channels

- Retailer Name/ID
- Price Per Unit
- Units Sold
- Sales Revenue
- Operating Profits and Margin

Problem Statement

Project Objective

Investigate the impact of Covid-19-induced shifts in consumer behavior and retail dynamics on the price sensitivity of Adidas product consumers

Methodology

Analyze price elasticities across different sales channels, regions, and product categories during the years 2020-2021

Suggestion

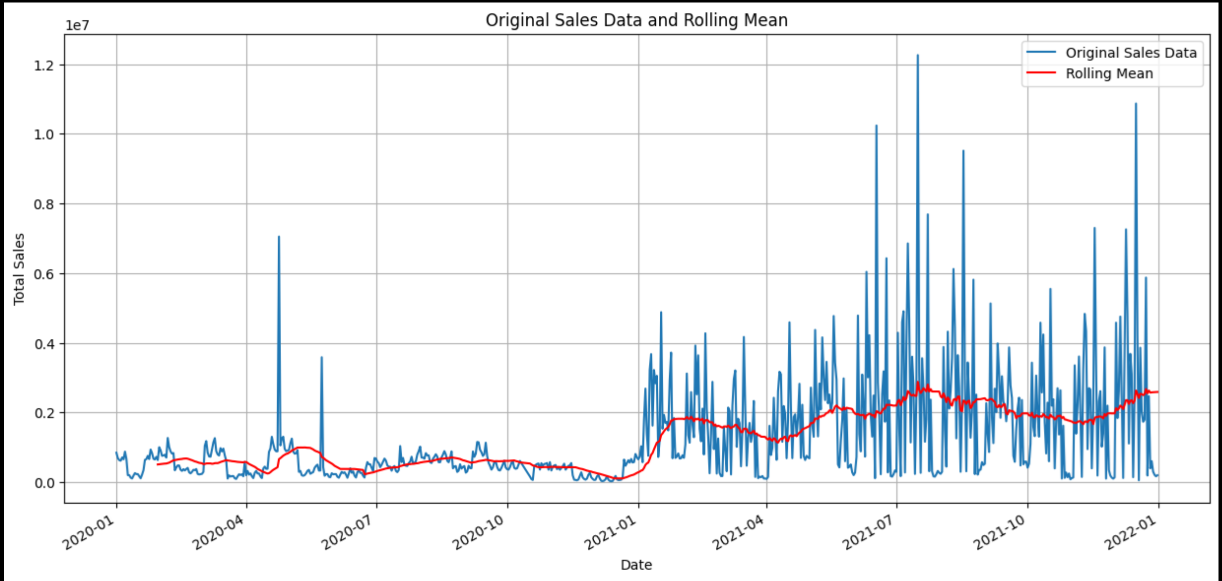
Offer Adidas the overall consumer behavior insights during pandemic across different geographies and products

TABLE OF CONTENTS

01

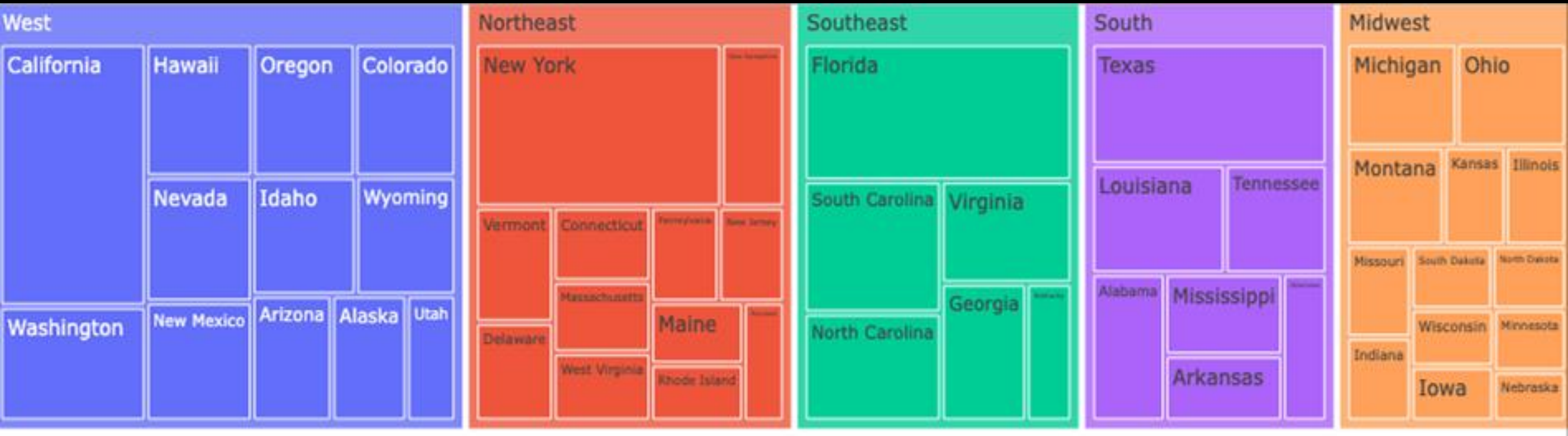
Exploratory Data Analysis

Sales Revenue in 2020 & 2021



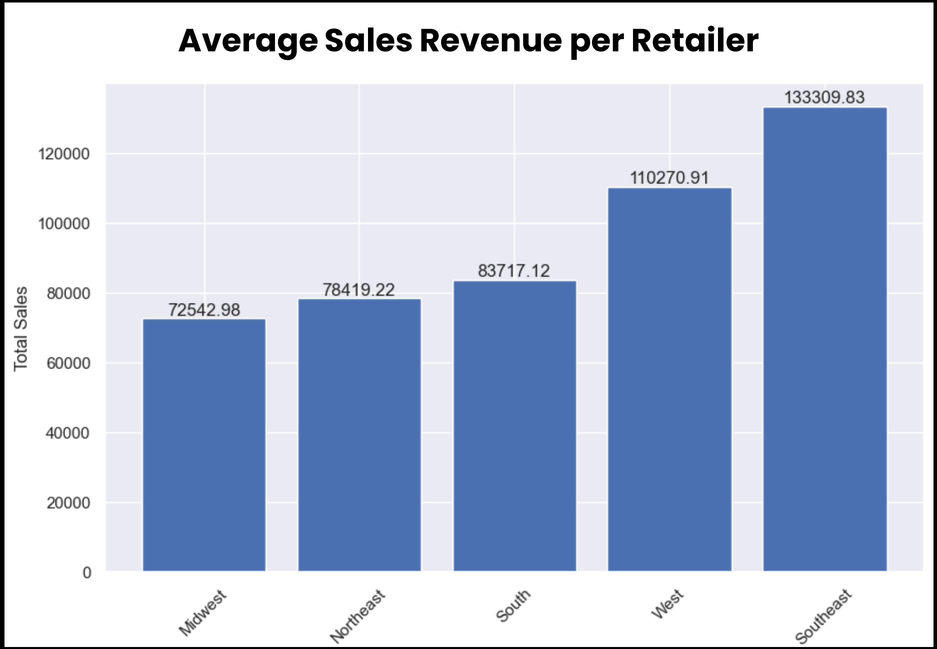
- Clear dip observed around the early to mid-2020, likely due to the impact of the COVID-19 pandemic
- Sales seem to recover somewhat after the dip and maintain a relatively consistent level towards the end of the year
- Rolling Mean: Window=30

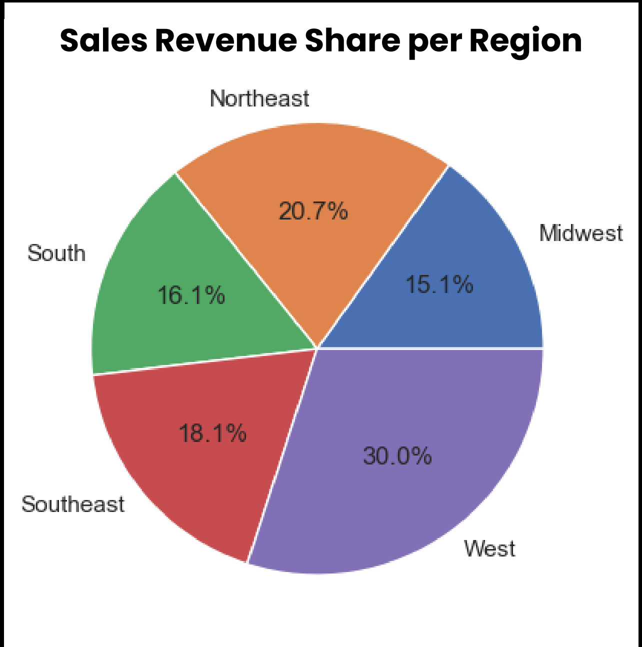
Sales Revenue By Region



California, New York, Florida, Texas, and Michigan recorded the highest sales in 2020-2021

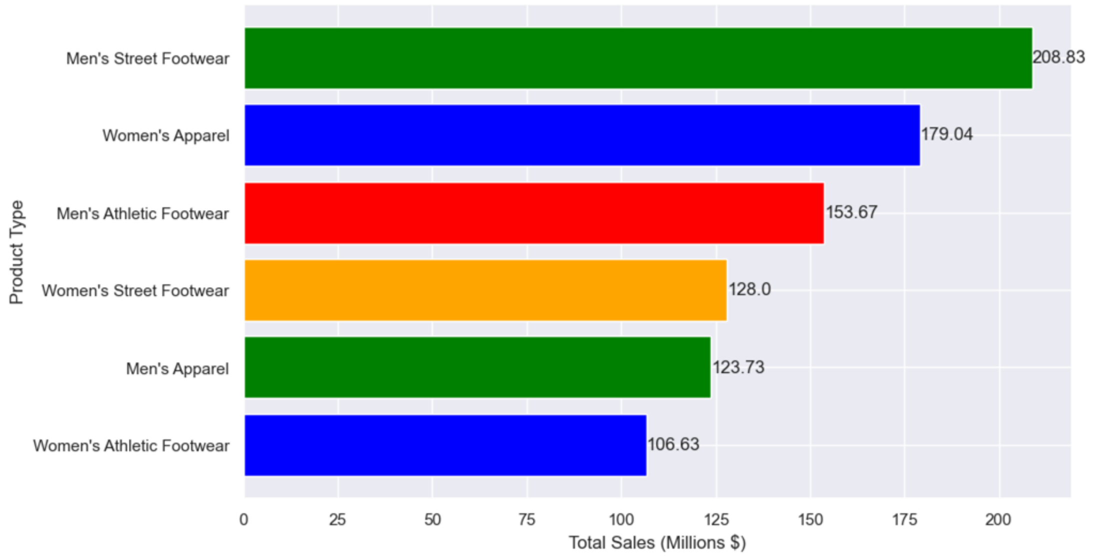
Sales Revenue By Region



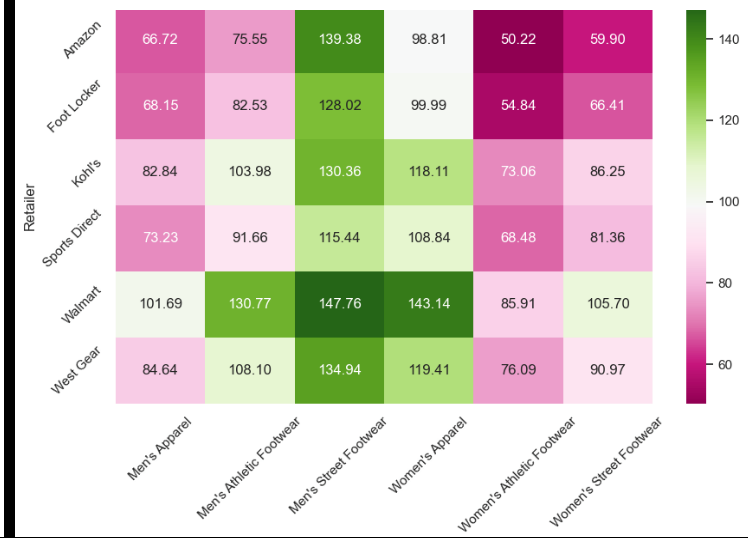


- West leads in total sales; Southeast in retailer average.
- Northeast's second in total sales contrasts with its near-bottom retailer average, indicating relatively high sales variance among retailers.

Sales Revenue by **Product**



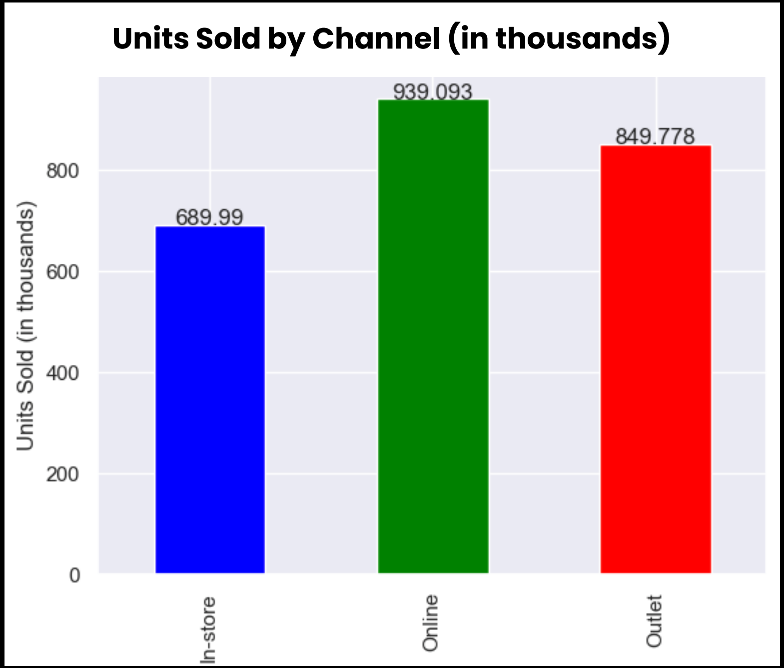
Sum Total Revenues per Product (Million$)

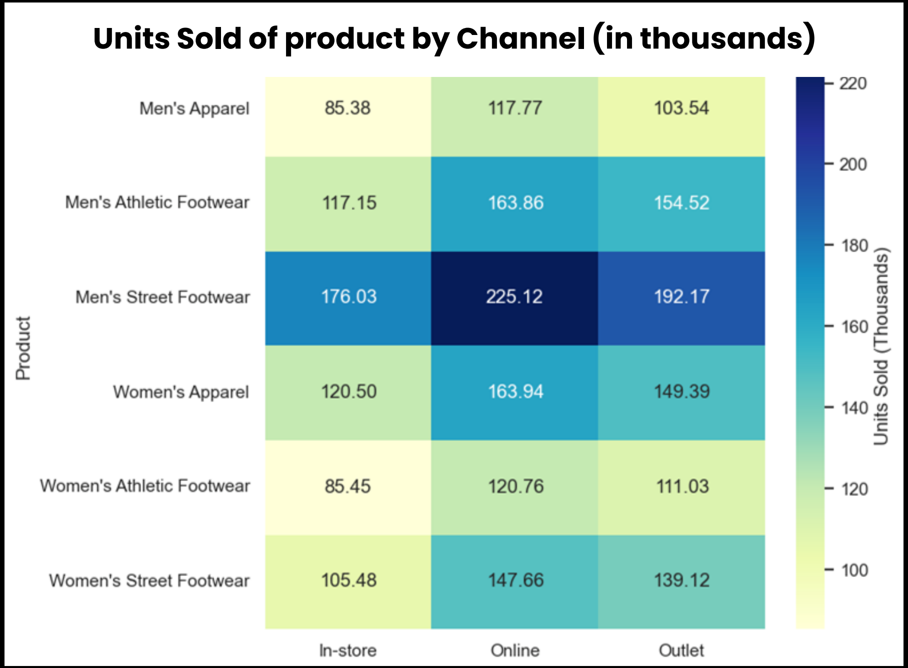


Average Sales Revenue for different Products across Retailers ($k)

- Total Sales are most notable for Men's Street Footwear and Women's Apparel
- Walmart holds the sales lead in all top three categories.

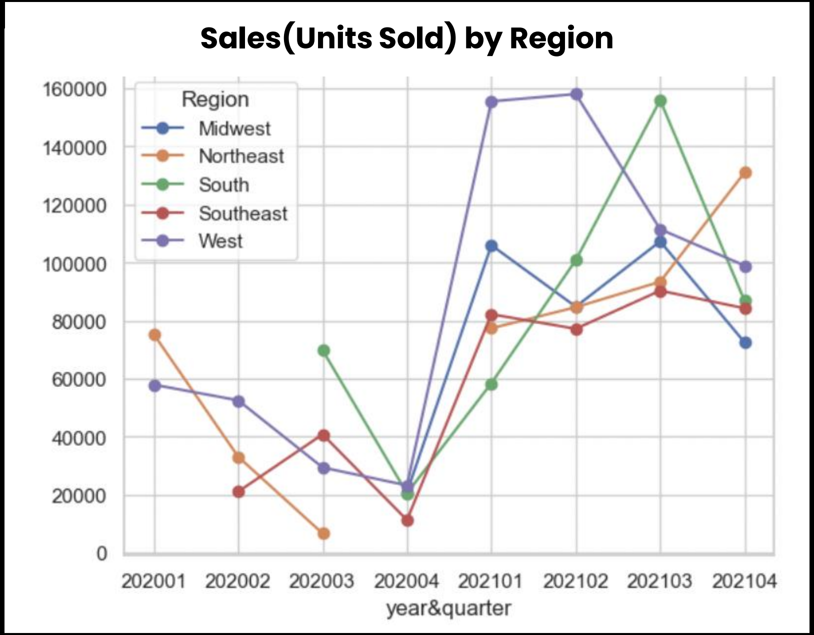
Units Sold (Sales) by Channel



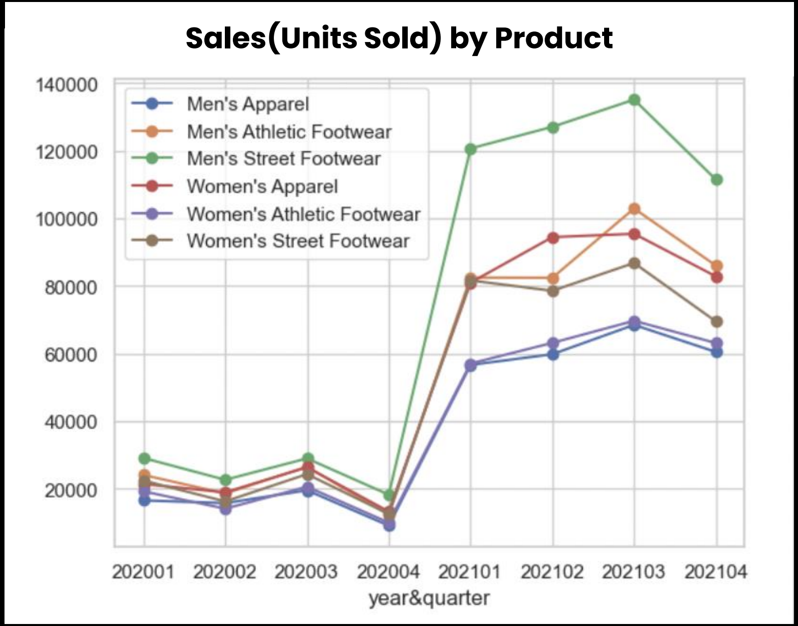


- The highest number of products were sold online
- Men's Street Footwear is the best-selling product across all channels, with online sales recording the highest units sold for all products

Sales Trends by Region and Product

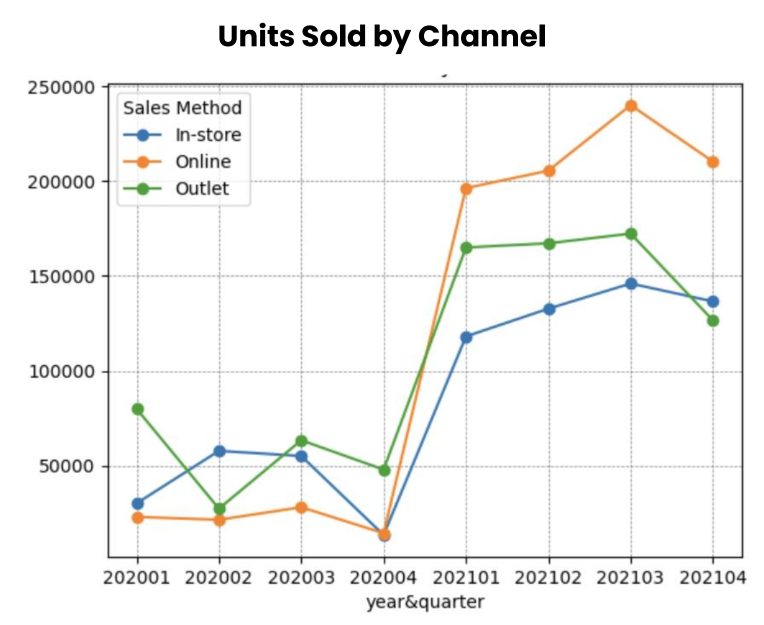


Adidas sales over the region seems to recover from the pandemic in 2021 Q1 but yet not so stable

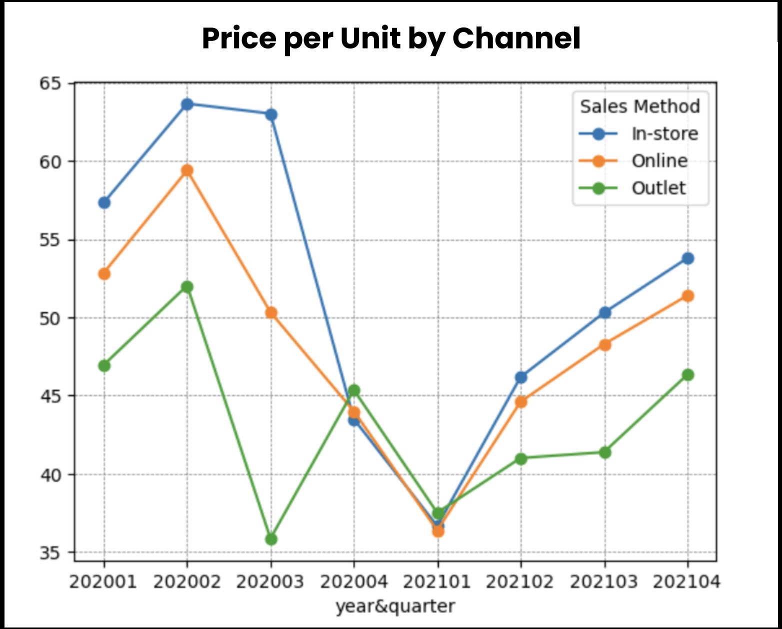


There has been an increasing disparity in sales across product categories

Sales Trends by Channel



During the pandemic, Online became the sought-after channel for adidas



Overall cut-down in price per unit explains the sharp increase in units sold in 2021

02

General Models

Multiple Linear Regression & Random Forest

Data Preprocessing

To predict 'Units Sold (sale)'

Dropping the derived variables columns

- Operating margin
- Operating profit
- Total Sales
- Retailer info

Creating month and year columns

- Year (2020, 2021)
- Month (Jan-Dec)

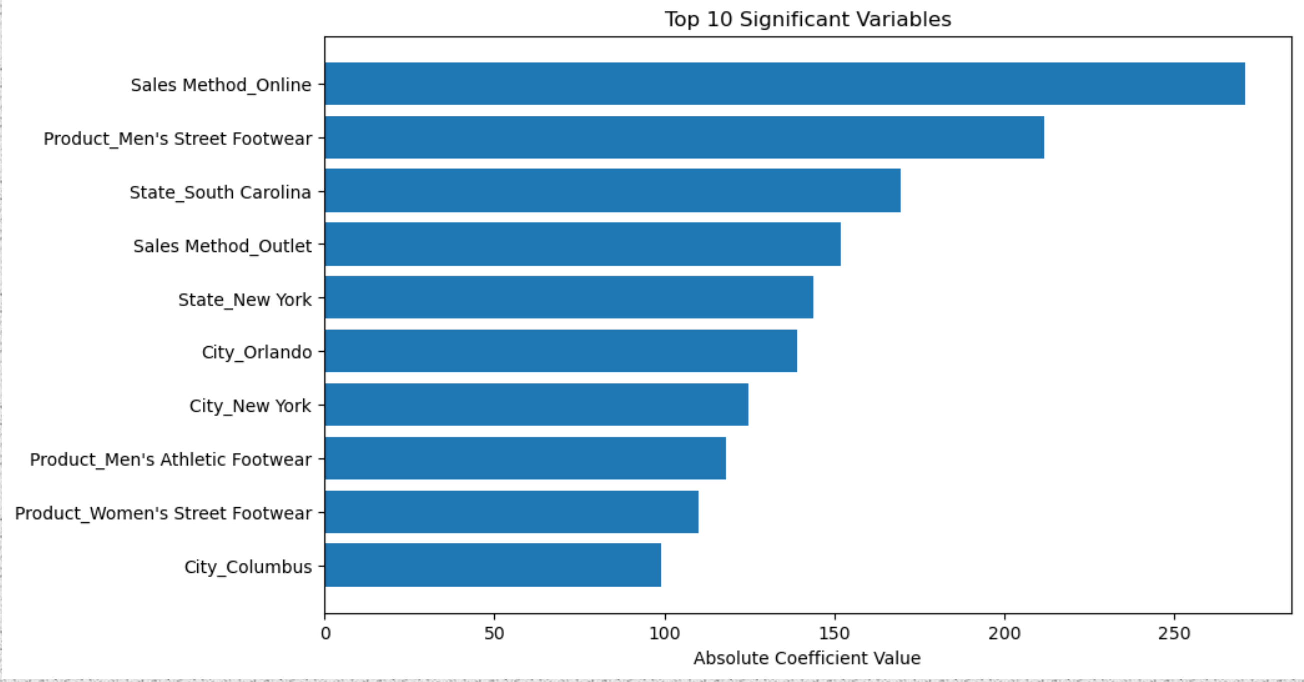
Dummy Encoding Categorical Variables

- Region, City, State
- Product Category
- Sales Channel
- Year, Month

Region, State, City, Product, Price per Unit, Units Sold, Sales Channel, Year, Month

Multiple Linear Regression

Feature Importance

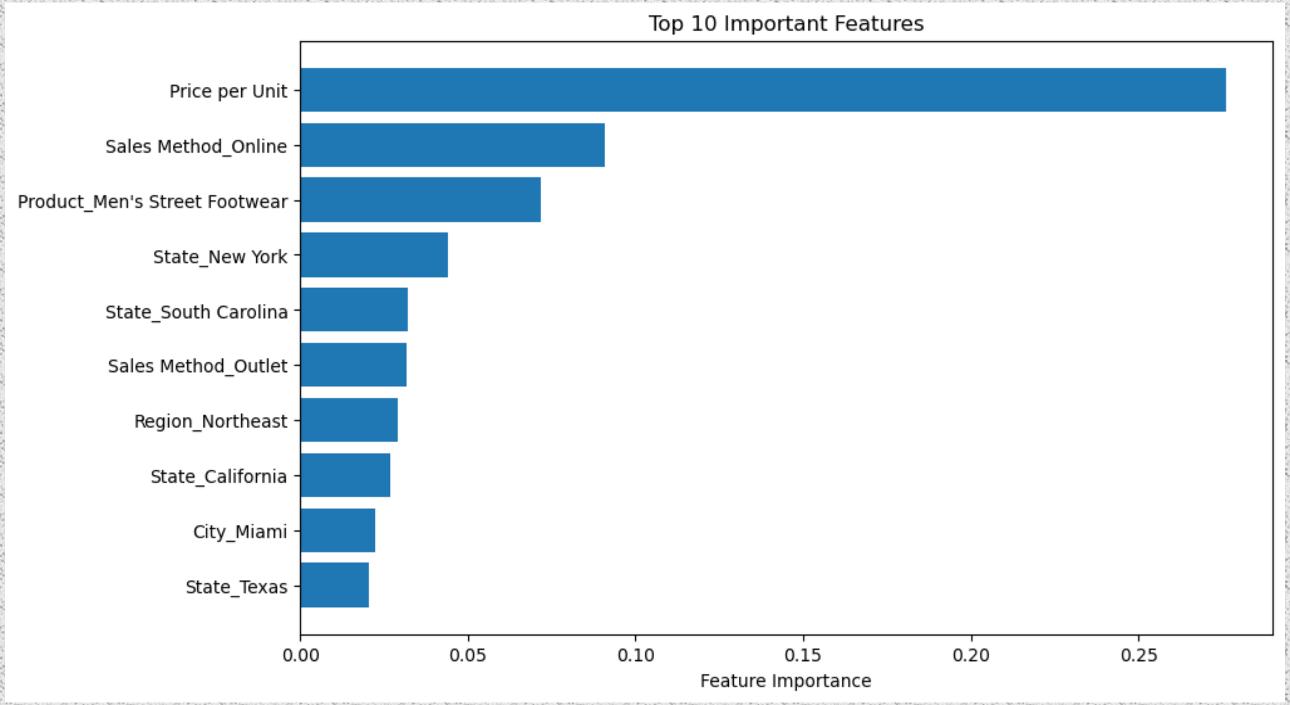


Pointers

- Train-test split : 80% - 20%
- F- Statistic : 124.6
- Significant variables cutoff at P-value < 0.05

Random Forest

Feature Importance



Pointers

- Train-test split :
  80% - 20%
- 5 fold cross validation based on MSE score
- Optimal number of trees: 1000

Model Comparison

Performance

Parameter	Random Forest	MLR
Test MSE	11,680	21,293.19
Test R-Squared	0.75	0.55
Best n_estimators (through Cross Validation)	1000	-

Comments

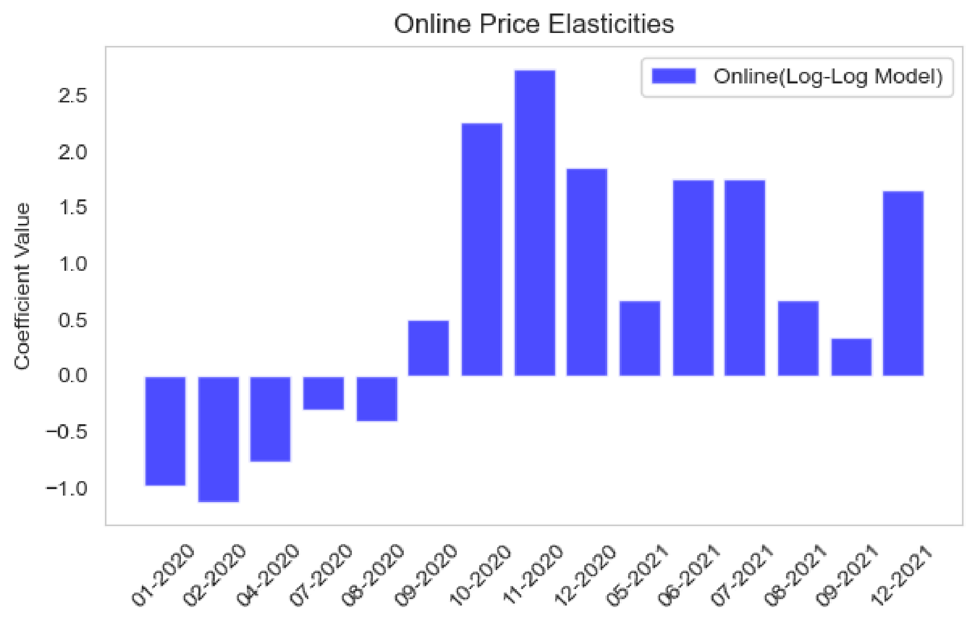
- There are non-linear relationships and interactions between variables that Random Forest is capturing but Linear Regression is not.
- Hence we can consider variable feature importance from Random Forest

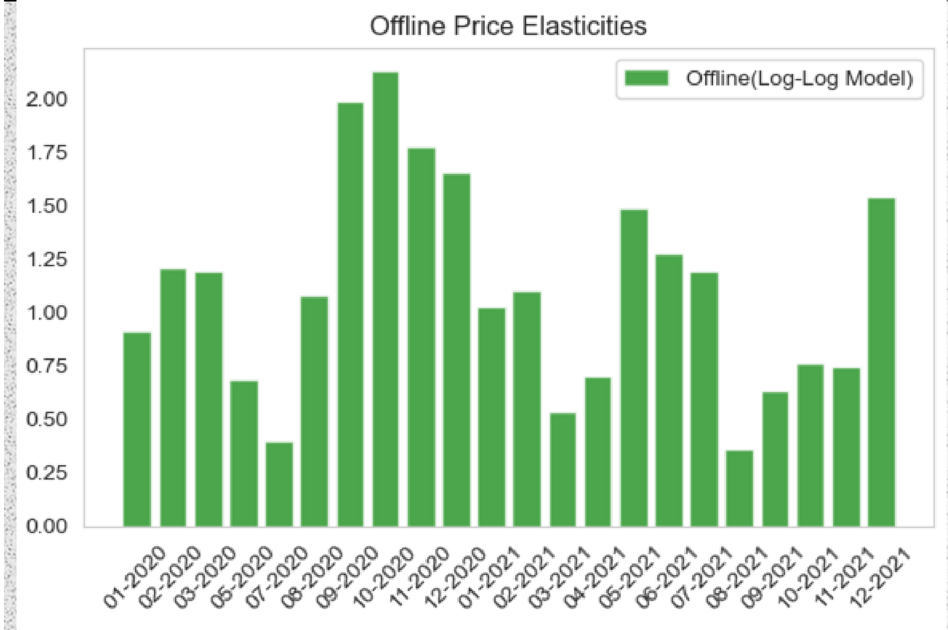
Hence ***Price Per Unit, Sales methods*** and a ***couple of demographic variables*** for states and regions can be noted as most important

03

Price Elasticities

Price Elasticities by Sales Channel

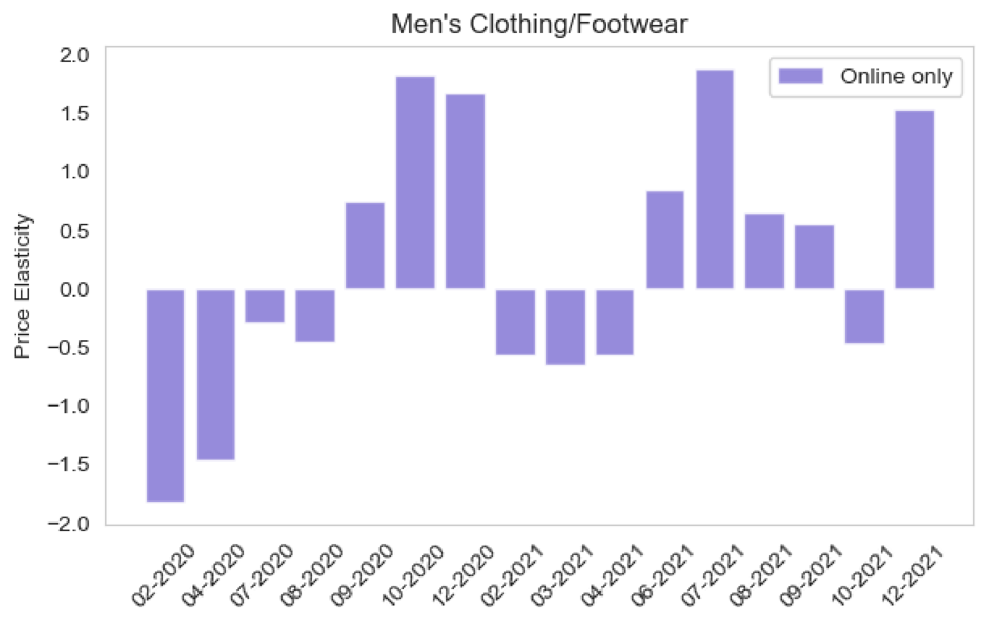


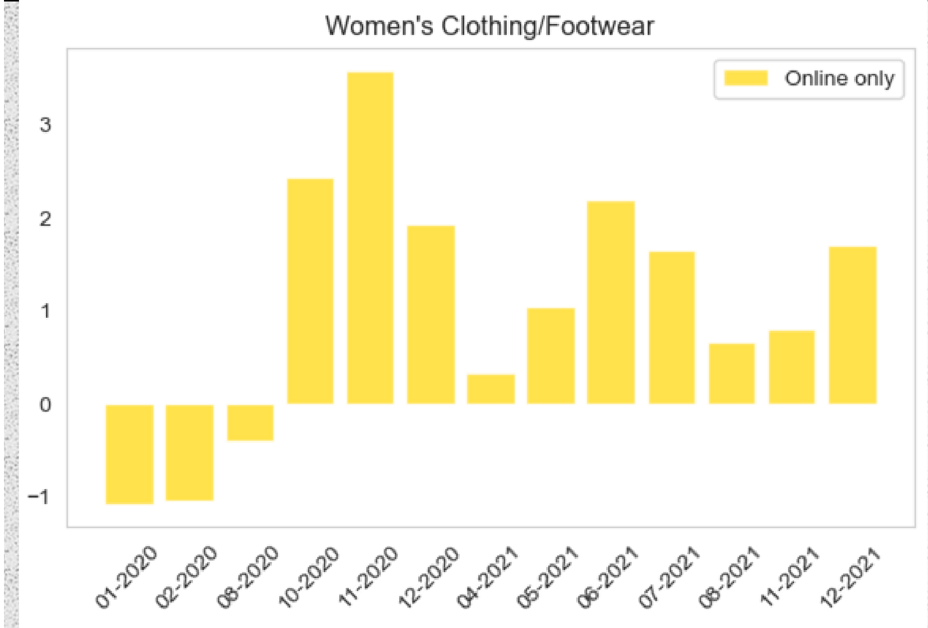


- Customers shopping online tended to compare prices more actively at the onset of Covid-19
- Shift in 2021 : Likely attributed to regional or product type level changes in customer behavior and skews

- Positive price elasticities during Covid-19
- In-store visits might have led to purchases regardless of prices due to product try-ons

Price Elasticities by Product Type

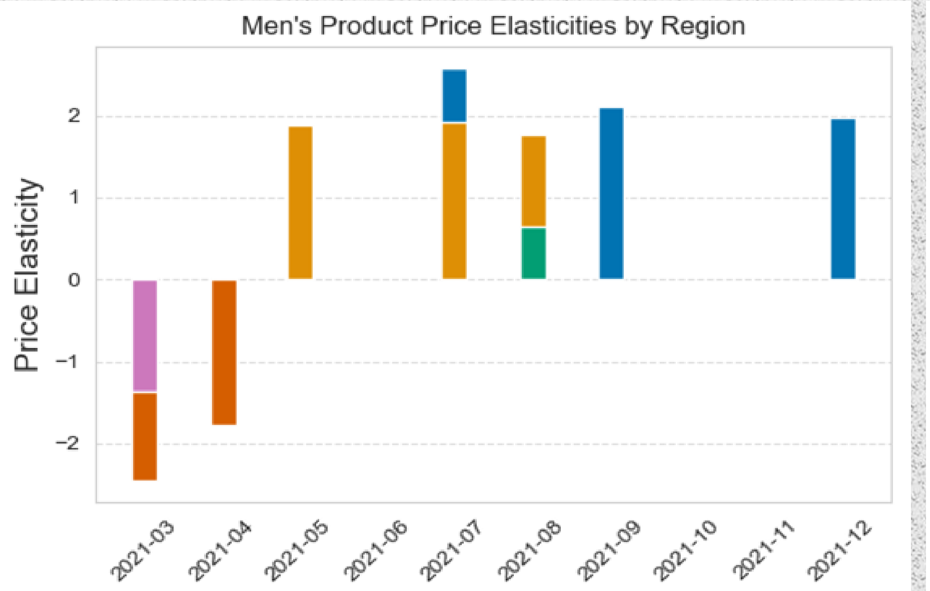


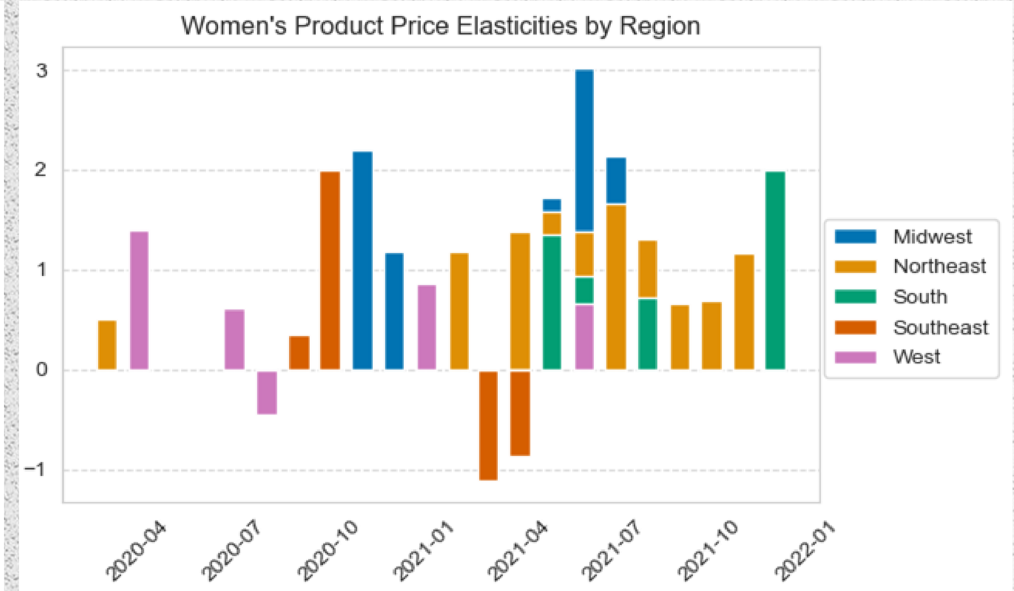


- Men's products showed a decrease in sales when prices went up in 2020 and 2021

- In contrast, women's products continued to sell well even when prices increased.
- Cognitive Bias : Higher prices → better quality.
- Limited traditional shopping options during covid → Increased online shopping

Product Price Elasticities by Region





- Limited data points for online men's category sales in 2020 resulted in insignificant coefficients
- North east, Midwest and South : Higher disposable incomes and population density combined with covid anxiety might explain the less price sensitivity
- Southeast and West : Negative price elasticities indicate higher price sensitivities and possible brand loyalty issues

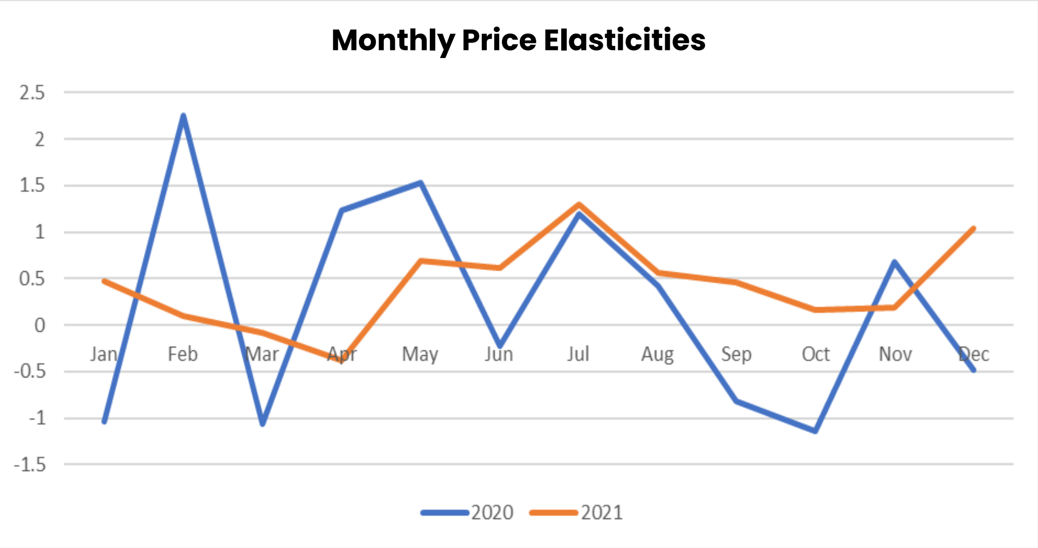
Monthly Price Elasticities (using Hierarchical Bayes)

2020

Month Year	Price Elasticities
Jan-20	-1.03
Feb-20	2.25
Mar-20	-1.06
Apr-20	1.23
May-20	1.53
Jun-20	-0.23
Jul-20	1.19
Aug-20	0.42
Sep-20	-0.82
Oct-20	-1.15
Nov-20	0.68
Dec-20	-0.49

2021

Month Year	Price Elasticities
Jan-21	0.48
Feb-21	0.10
Mar-21	-0.08
Apr-21	-0.38
May-21	0.69
Jun-21	0.62
Jul-21	1.30
Aug-21	0.56
Sep-21	0.47
Oct-21	0.16
Nov-21	0.19
Dec-21	1.04



- Hierarchical Bayes model suggests that for online sales channel the demand was mostly inelastic for 2021.
- In 2020, price elasticity fluctuated from mostly positive in the first half to negative in the second half.

04

Conclusion & Suggestion

Summary of the findings

/ Sales Strategy

/ Channel	Ensure competitive pricing in the online channel due to the distinctive price sensitive behavior observed in this domain
/ Category	Price increases in the men's product categories negatively impact the sales volume - targeted promotional offers to offset sales decrease
/ Region	Prioritize high margin sales in regions with low price sensitivity – Northeast, South and Southeast

THANKS!

DO YOU HAVE ANY QUESTIONS?

Appendix

```python
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm

df = pd.read_csv("Adidas US Sales Datasets.csv")
df['Invoice Date']=pd.to_datetime(df["Invoice Date"])
df["Month"] = df["Invoice Date"].dt.month_name()
df['Year']=df["Invoice Date"].dt.year
df=df[['Retailer','State','Region','City','Product', 'Sales Method','Month','Year','Units Sold','Price per Unit']]

# Perform one-hot encoding for categorical variables
df_encoded = pd.get_dummies(df, columns=['Retailer', 'Product','City', 'Region','Sales Method','State','Month','Year'],drop_first=True)


# Create a Linear Regression model
mlr_model = LinearRegression()

# Define the features (X) and the target variable (y)
X = df_encoded.drop(columns=['Units Sold'])
y = df_encoded['Units Sold']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

X_train.replace({False: 0, True: 1}, inplace=True)
X_test.replace({False: 0, True: 1}, inplace=True)

# Fit the model to your training data
mlr_model.fit(X_train, y_train)

# Assess feature significance using p-values
X_train_with_const = sm.add_constant(X_train)  # Add a constant term (intercept) to the features
mlr_model_sm = sm.OLS(y_train, X_train_with_const).fit()
print(mlr_model_sm.summary())
```

Multiple Linear Regression

```
Test Set Mean Squared Error (MSE): 21293.26
Test Set R-squared (R2) Score: 0.55
```

Appendix

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV, train_test_split
from sklearn.metrics import mean_squared_error,r2_score
from sklearn.model_selection import cross_val_score

X = df_encoded.drop(columns=['Units Sold'])
y = df_encoded['Units Sold']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

n_estimators_range = [10, 50, 100, 300,500,1000]

mse_scores = []

for n_estimators in n_estimators_range:

    rf_model = RandomForestRegressor(n_estimators=n_estimators, random_state=42)
    scores = cross_val_score(rf_model, X_train, y_train, cv=5, scoring='neg_mean_squared_error')
    mse_scores.append(-scores.mean())

best_estimators = n_estimators_range[mse_scores.index(min(mse_scores))]
best_model = RandomForestRegressor(n_estimators=best_estimators, random_state=42)
best_model.fit(X_train, y_train)
y_pred_test = best_model.predict(X_test)
mse_test = mean_squared_error(y_test, y_pred_test)
r2_val = r2_score(y_test, y_pred_test)

print(f"Best n_estimators: {best_estimators}")
print(f"Test Set Mean Squared Error (MSE): {mse_test:.2f}")
print(f"Test Set R-squared (R2) Score: {r2_val:.2f}")
```

Random Forest Regressor

```
Best n_estimators: 1000
Test Set Mean Squared Error (MSE): 11680.00
Test Set R-squared (R2) Score: 0.75
```

Appendix

```
sales_df=sales_df.sort_values(by=["year","month"],ascending=True)
dummy_df = pd.get_dummies(sales_df, columns=['Region', 'State','Retailer','City','Product'])
dummy_df

import numpy as np
import statsmodels.api as sm

dummy_df = pd.get_dummies(sales_df, columns=['Region', 'State', 'Retailer', 'City'])

short_dates = dummy_df["short_date"].unique()

online_df = pd.DataFrame(columns=['Short Date', 'Coefficient'])

for short_date in short_dates:
    month, year = map(int, short_date.split('/'))

    test11 = dummy_df[(dummy_df["year"] == year) & (dummy_df["month"] == month) &
                      (dummy_df["Sales Method"] == "Online")].copy(deep=True)

    X = np.log(test11[["Price per Unit"]])
    y = np.log(test11["Units Sold"])

    X = sm.add_constant(X)
    linMod = sm.OLS(y,X).fit()
    if linMod.pvalues[1]<0.05:
        online_df = pd.concat([online_df, pd.DataFrame({'Short Date': [short_date],
                                                        'Coefficient': [linMod.params[1]],
                                                        'P-value': [round(linMod.pvalues[1],3)]})],
                                                        ignore_index=True)
```

Price Elasticity of Units Sold - Online Channel Only

Appendix

```
#Month-level elasticity modelling for Men's product categories
sales_df=sales_df.sort_values(by=["year","month"],ascending=True)

import numpy as np
import statsmodels.api as sm

dummy_df = pd.get_dummies(sales_df, columns=['State', 'Retailer', 'City'])

short_dates = dummy_df["short_date"].unique()

men_df = pd.DataFrame(columns=['Short Date', 'Coefficient'])

for short_date in short_dates:
    month, year = map(int, short_date.split('/'))

    test11 = dummy_df[(dummy_df["year"] == year) & (dummy_df["month"] == month) &
                      (dummy_df["Sales Method"] == "Online")& dummy_df["Product"].isin(["Men's Apparel",
                                                                                        "Men's Athletic Footwear",
                                                                                        "Men's Street Footwear"])].copy(deep=True)

    X = np.log(test11[["Price per Unit"]])
    y = np.log(test11["Units Sold"])

    X = sm.add_constant(X)
    linMod = sm.OLS(y,X).fit()
    if linMod.pvalues[1]<0.05:
        men_df = pd.concat([men_df, pd.DataFrame({'Short Date': [short_date],
                                                  'Coefficient': [linMod.params[1]],
                                                  'P-value': [round(linMod.pvalues[1],3)]})],
                                                  ignore_index=True)
```

Price Elasticity – Online + Men's Product Categories

Appendix

```
#regional, month-level price elasticities for men's products
men_region_df = pd.DataFrame(columns=['Region', 'Short Date', 'Coefficient', 'P-value'])

for short_date in short_dates:
    month, year = map(int, short_date.split('/'))

    for region in dummy_df["Region"].unique():
        test11 = dummy_df[(dummy_df["year"] == year) & (dummy_df["month"] == month) &
                          (dummy_df["Sales Method"] == "Online") &
                          (dummy_df["Product"].isin(["Men's Apparel", "Men's Athletic Footwear", "Men's Street Footwear"])) &
                          (dummy_df["Region"] == region)].copy(deep=True)

        if test11.shape[0] > 0:
            X = np.log(test11[["Price per Unit"]])
            y = np.log(test11["Units Sold"])

            X = sm.add_constant(X)
            linMod = sm.OLS(y,X).fit()

            if linMod.pvalues[1] < 0.05:
                men_region_df = pd.concat([men_region_df,
                                          pd.DataFrame({'Region': [region],
                                                        'Short Date': [short_date],
                                                        'Coefficient': [linMod.params[1]],
                                                        'P-value': [round(linMod.pvalues[1], 3)]})], ignore_index=True)
men_region_df['Short Date'] = pd.to_datetime(men_region_df['Short Date'], format='%m/%Y')
```

Regional Monthly Price Elasticities Online + Men’s Product Categories

Appendix

```r
library(tidyverse)
library(readxl)
file_path <- "C:\\Users\\Mayank\\Marketing analytics\\Adidas US Sales Datasets.xlsx"
df <- read_xlsx(file_path)
library(rstan)

df <- df[df$`Units Sold` != 0,]

library(dplyr)

# Convert 'Invoice Date' to a datetime object
df$`Invoice Date` <- as.POSIXct(df$`Invoice Date`, format = "%Y-%m-%d %H:%M:%S")

# Convert selected columns to numeric
numeric_columns <- c('Price per Unit', 'Units Sold', 'Total Sales', 'Operating Profit', 'Operating Margin')
df[numeric_columns] <- lapply(df[numeric_columns], as.numeric)

# Extract year, month, and day
df$Year <- lubridate::year(df$`Invoice Date`)
df$Month <- lubridate::month(df$`Invoice Date`)
df$Day <- lubridate::day(df$`Invoice Date`)

# Concatenate "Month" and "Year" columns
df$MonthYear <- paste(df$Month, df$Year, sep = "")

# If you want the result as a character, you can convert it
df$MonthYear <- as.character(df$MonthYear)

# Assuming your data frame is named 'data' and columns are named 'Price per Unit' and 'Units Sold'
df$Price_log <- log(df$`Price per Unit`)
df$Units_log <- log(df$`Units Sold`)

# Select the desired columns
selected_columns <- c("Price_log", "Units_log", "MonthYear", "Sales Method", "Retailer", "Region")
df <- df %>% select(all_of(selected_columns))

# Filter records where Sales Method is "Online Store"
df <- df %>% filter(`Sales Method` == "Online")

# Calculate num_months
num_months <- length(unique(df$MonthYear))
num_regions <- length(unique(df$Region))
num_retailers <- length(unique(df$Retailer))
```

Hierarchical Bayes Price Elasticity Modelling (1/3) - Monthly Elasticities + Online Channel + Both Product Types

Appendix

```
stan_code <- "
data {
  int<lower=1> N;  // Number of data points
  vector[N] Units_log;  // Log-transformed Units Sold
  vector[N] Price_log;  // Log-transformed Price per Unit
  int<lower=1> num_months;  // Number of unique MonthYear values
  int<lower=1, upper=num_months> MonthYear[N];  // MonthYear IDs
  int<lower=1> num_regions;  // Number of unique Region values
  int<lower=1, upper=num_regions> Region[N];  // Region IDs
  int<lower=1> num_retailers;  // Number of unique Retailer values
  int<lower=1, upper=num_retailers> Retailer[N];  // Retailer IDs
}

parameters {
  real alpha_mu;  // Hyperparameter for alpha
  real alpha_sd;  // Hyperparameter for alpha
  real beta_mu;   // Hyperparameter for beta
  real beta_sd;   // Hyperparameter for beta
  real delta_mu;  // Hyperparameter for delta (Region effect)
  real delta_sd;  // Hyperparameter for delta
  real eta_mu;    // Hyperparameter for eta (Retailer effect)
  real eta_sd;    // Hyperparameter for eta
  real alpha[num_months];  // Intercept for each MonthYear
  real beta[num_months];  // Price elasticity for each MonthYear
  real delta[num_regions];  // Region effect for each Region
  real eta[num_retailers];  // Retailer effect for each Retailer
}

model {
  alpha ~ normal(alpha_mu, alpha_sd);
  beta ~ normal(beta_mu, beta_sd);
  delta ~ normal(delta_mu, delta_sd);
  eta ~ normal(eta_mu, eta_sd);

  for (n in 1:N) {
    Units_log[n] ~ normal(alpha[MonthYear[n]] + beta[MonthYear[n]] * Price_log[n] + delta[Region[n]] + eta[Retailer[n]], 1);
  }
}

generated quantities {
  real price_elasticity[num_months];
  for (m in 1:num_months) {
    price_elasticity[m] = beta[m];  // Price elasticity for each MonthYear
  }
}
"
```

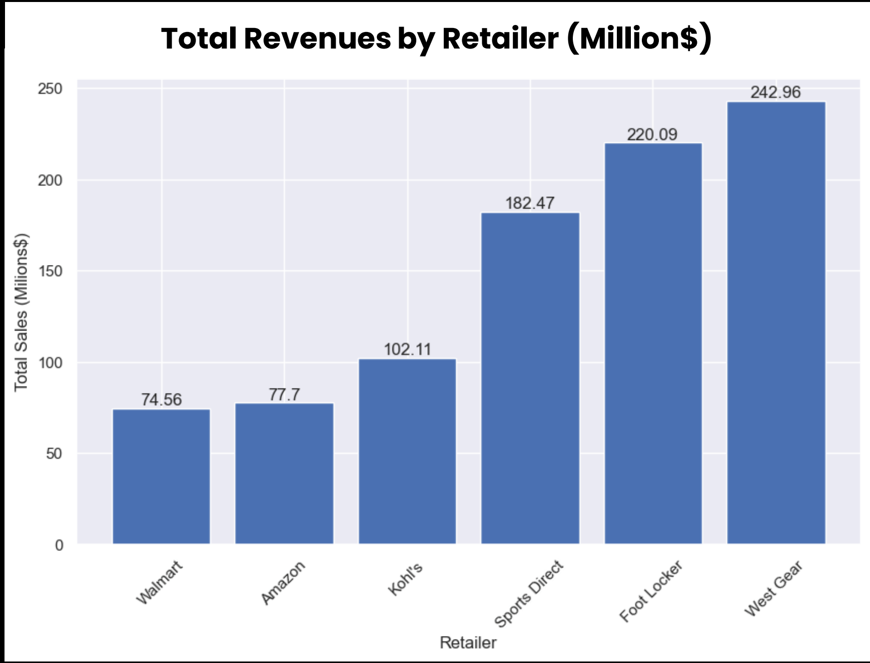
Hierarchical Bayes Model - Continued(2/3)

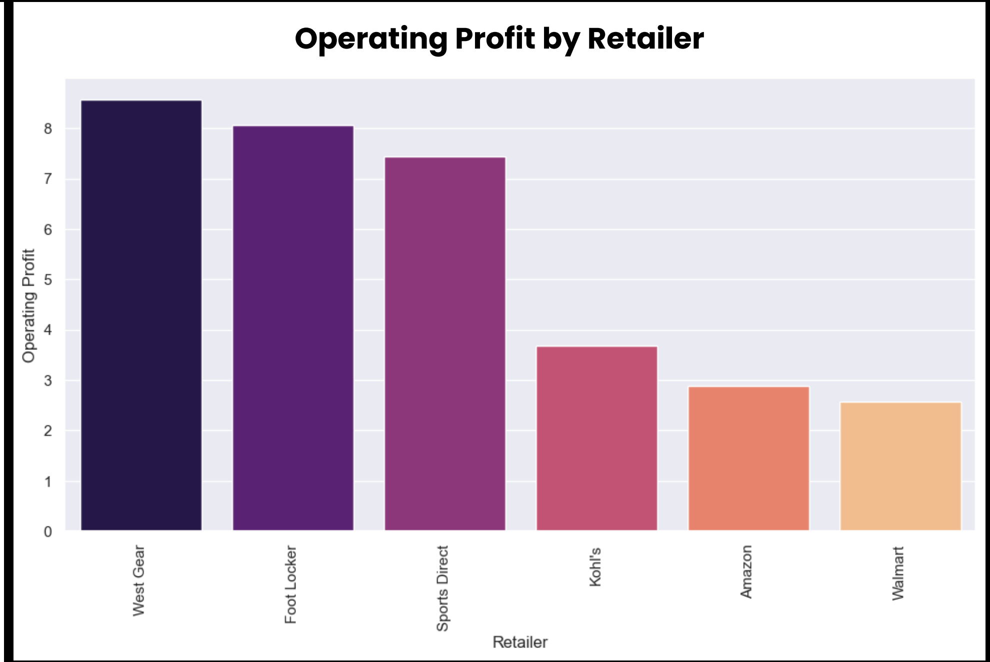
Appendix

```r
# Compile the Stan model
stan_model <- stan_model(model_code = stan_code)

# Get unique MonthYear values from your original dataframe 'df'
unique_months <- unique(df$MonthYear)

# Initialize an empty data frame to store the results
results_df <- data.frame(MonthYear = numeric(0), price_elasticity = numeric(0))

counter = 0
# Loop through each unique MonthYear value
for (month_year in unique_months) {
  counter = counter+1
  # Filter data for the current MonthYear
  data_subset <- df[df$MonthYear == month_year, ]

  # Fit the Stan model to the subset of data
  model_fit <- sampling(stan_model,
                        data = list(N = nrow(data_subset),
                                    Units_log = data_subset$Units_log,
                                    Price_log = data_subset$Price_log,
                                    num_months = length(unique_months),
                                    MonthYear = as.integer(factor(data_subset$MonthYear)),
                                    num_regions = nrow(data_subset),
                                    Region = as.integer(factor(data_subset$Region)),
                                    num_retailers = nrow(data_subset),
                                    Retailer = as.integer(factor(data_subset$Retailer))),
                        iter = 2000, chains = 4)
```

Hierarchical Bayes Model - Continued(3/3)

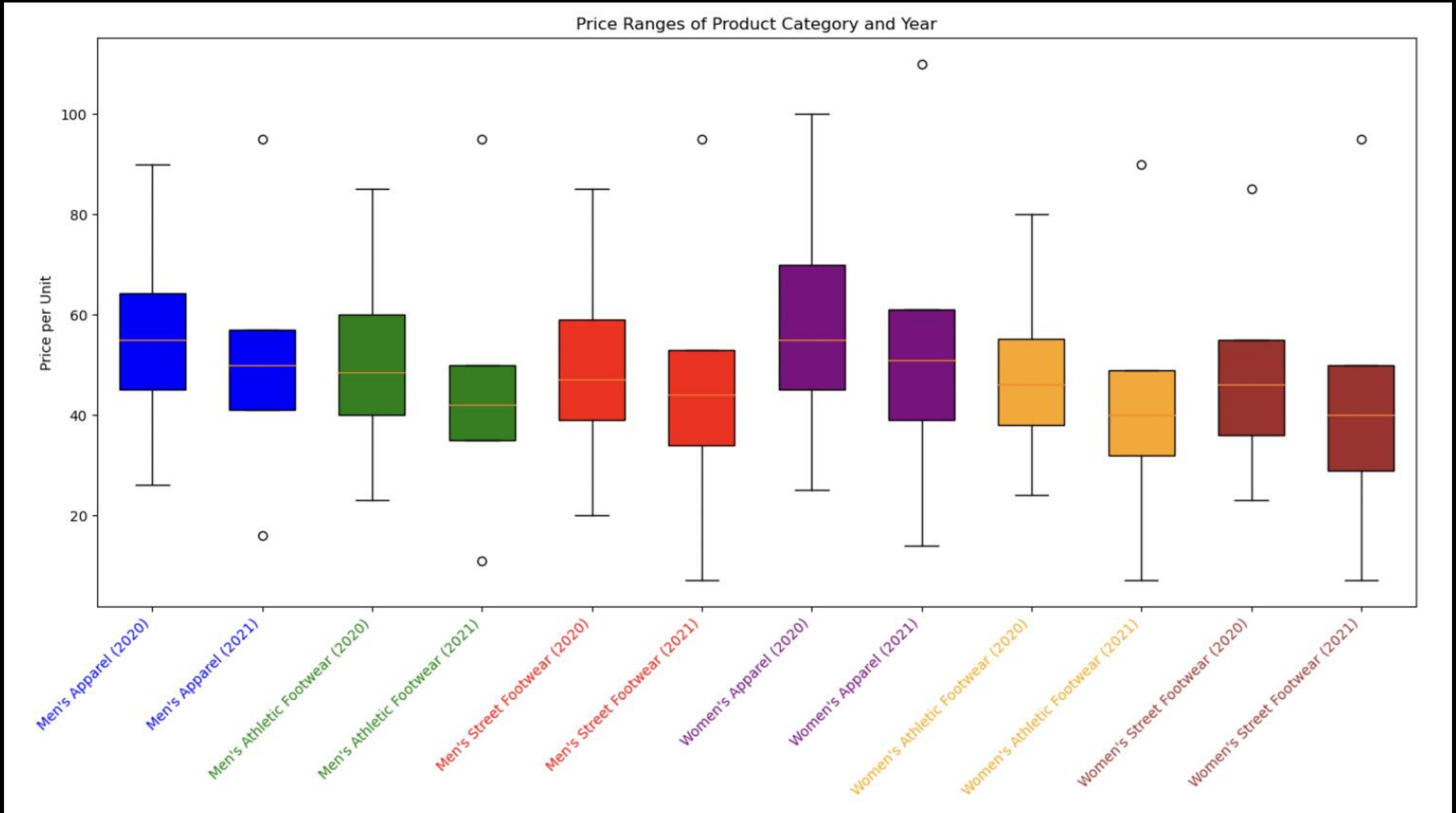
Revenues and Profit by Retailer





- West Gear, Foot Locker, Sports Direct are the top 3 Retailers for Adidas Sales
- Retailer's Total Revenue and Operating Profit generally show a proportional relationship

Price Ranges by Product



Price Elasticities for Different Regions