# Company Financials for Predictive Insights and Economic Trends

**Poojith Reddy Annachedu**

**A20539719**

**Illinois Institue of Technology**

**Chicago, USA**

**Neeraj Vardhan buneeti**

**A20545853**

**Illinois Institue of Technology**

**Chicago, USA**

**Sai Charan Gangili**

**A20543155**

**Illinois Institue of Technology**

**Chicago, USA**

## Illinois Institute of Technology

## CSP571-Data Preparation and Analysis

## Professor: Jawahar Panchal

# Table of Contents

# ABSTRACT

In today's volatile market, accurately predicting the financial performance of companies is paramount. Leveraging the Company Financials Dataset, this study aims to identify key indicators for forecasting future profitability and creating models to predict the financial state for the upcoming fiscal year. Through regression analysis, exploratory data analysis, and data preprocessing, we identified crucial indicators of financial performance. Our research provides insights essential for evaluating growth potential. The developed predictive model demonstrates high accuracy in forecasting financial trends and adapts to different industries, detecting unique economic patterns and disruptions. Understanding these insights not only helps stakeholders anticipate financial downturns but also enhances comprehension of economic trends. Future efforts will focus on refining these models and integrating real-time data to enhance responsiveness to market dynamics.

***Keywords:*** *Financial performance prediction, Regression analysis, Exploratory data analysis, Forecasting profitability, Economic trends, Real-time data integration, Market dynamics adaptation.*

# Research Summary

The purpose of this report is to conduct a detailed examination of corporate financial data to extract predictive insights and discern prevailing economic patterns. To achieve this process, we have employed a robust dataset acquired from Kaggle, knows as the "Company Financials Dataset." Our research is oriented towards revealing significant trends and establishing connections among the various financial parameters of a multitude of firms.

Our investigation is structured to parse through the financial information to understand better and predict future financial scenarios and market behaviors. By systematically analyzing the dataset, we aim to provide a strategic advantage in identifying the financial underpinnings that drive company performance and economic outcomes.

# Findings

Through rigorous analysis of the dataset, significant findings have emerged regarding the financial performance of the companies under study. Key metrics such as Gross sales by segment and country, Gross sales by country, and Profit by Country have been examined to discern trends and patterns indicative of future performance. Additionally, comparative analysis across country and sales has provided valuable insights into sector-specific dynamics and market trends.

## Overview

In an era marked by rapid economic shifts and increasing market volatility, the capacity to accurately predict the financial performance of a company has become extremely valuable. To identify indicators that can accurately forecast a company's financial health and growth trajectory, this project makes use of the large Company Financials Dataset. This research aims to improve predictive models that stakeholders can use to foresee financial trends and make informed decisions by combining sophisticated machine learning techniques with statistical analysis.

## Problem Statement:

In a dynamic market environment, businesses constantly struggle to forecast long-term success and identify potential financial risks. The goal of this project is to create a predictive model that can be used to systematically analyze historical financial data to identify reliable indicators of financial health and growth trajectories for companies.

## Relevant Literature:

- Predictive Financial Analytics: Machine learning-based corporate bankruptcy forecasting is validated by Thomas and Zhang (2018).
- Economic Indicators and Business Performance: To forecast performance trends, Smith et al. (2019) look at macroeconomic indicators.
- Financial Ratios as Performance Predictors: Johnson (2020) discovers strong relationships between several financial ratios and the tech industry's level of financial stability.

## Proposed Methodology:

1. Data Pre-processing: Cleansing and normalizing data to ensure uniformity and reliability for subsequent analysis.

2. Exploratory Data Analysis (EDA): Statistical analysis to understand data characteristics and inform feature selection.

3. Feature Selection: Using statistical methods to identify financial indicators critical to predicting financial sales.

4. Model Development: Constructing various models, including regression models, to predict financial performance.

5. Model Evaluation and Tuning: Utilizing cross-validation and performance metrics to refine models.

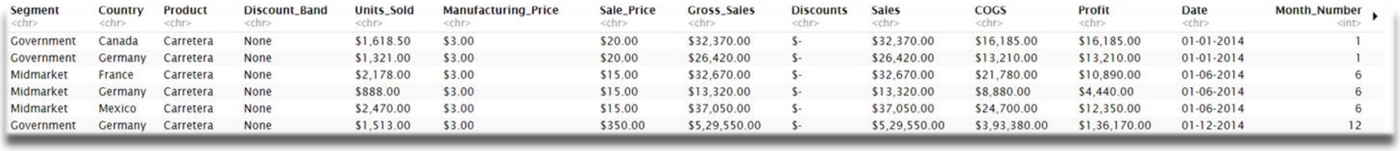
6. Trend Analysis: Investigating data trends to inform future financial predictions.

# Data Processing:

## Pipeline Details:

The data processing pipeline involves several steps to prepare the dataset for analysis. This includes loading the dataset using fread for faster loading, cleaning data by removing unwanted characters from numeric columns, handling missing values, and ensuring data consistency. The dataset used for this analysis is sourced from Kaggle - Company Financials Dataset.

```
library(data.table)
my_data <- fread("D:/Docs/Spring2024/DPA/project/Financials.csv")
head(my_data)
```



| Segment | Country | Product | Discount_Band | Units_Sold | Manufacturing_Price | Sale_Price | Gross_Sales | Discounts | Sales | COGS | Profit | Date | Month_Number |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| <chr> | <chr> | <chr> | <chr> | <chr> | <chr> | <chr> | <chr> | <chr> | <chr> | <chr> | <chr> | <chr> | <int> |
| Government | Canada | Carretera | None | $1,618.50 | $3.00 | $20.00 | $32,370.00 | $- | $32,370.00 | $16,185.00 | $16,185.00 | 01-01-2014 | 1 |
| Government | Germany | Carretera | None | $1,321.00 | $3.00 | $20.00 | $26,420.00 | $- | $26,420.00 | $13,210.00 | $13,210.00 | 01-01-2014 | 1 |
| Midmarket | France | Carretera | None | $2,178.00 | $3.00 | $15.00 | $32,670.00 | $- | $32,670.00 | $21,780.00 | $10,890.00 | 01-06-2014 | 6 |
| Midmarket | Germany | Carretera | None | $888.00 | $3.00 | $15.00 | $13,320.00 | $- | $13,320.00 | $8,880.00 | $4,440.00 | 01-06-2014 | 6 |
| Midmarket | Mexico | Carretera | None | $2,470.00 | $3.00 | $15.00 | $37,050.00 | $- | $37,050.00 | $24,700.00 | $12,350.00 | 01-06-2014 | 6 |
| Government | Germany | Carretera | None | $1,513.00 | $3.00 | $350.00 | $5,29,550.00 | $- | $5,29,550.00 | $3,93,380.00 | $1,36,170.00 | 01-12-2014 | 12 |

## Data Issues:

**Description:**

The 'Discount_Band' column contains non-numeric values such as "None," which may need to be addressed for consistency in calculations and modeling.

**Implementation:**

Convert 'None' values in 'Discount_Band' to NA for consistency in handling missing values.Remove rows with missing values in 'Discount_Band' after imputation.

```
my_data$Discount_Band[my_data$Discount_Band == "None"] <- NA
my_data <- na.omit(my_data)
```

## Assumptions / Adjustments:

**Assumptions:**

Missing values in the 'Discount_Band' column are imputed with the mean value.Categorical missing values in 'Discount_Band' are imputed with the mode (most frequent value).

**Adjustments:**

Convert 'None' values in 'Discount_Band' to NA for consistency in handling missing values.Ensure that the 'Discount_Band' column is properly converted to a factor with appropriate levels.

```
my_data$Discount_Band[is.na(my_data$Discount_Band)] <- mean(my_data$Discount_Band, na.rm = T
RUE)

my_data$Discount_Band <- as.factor(my_data$Discount_Band)
levels(my_data$Discount_Band) <- c(levels(my_data$Discount_Band), "None")
```

**Additional Steps:**

Check for and handle missing, NaN, or infinite values in 'Sales' and 'Discount_Band' columns.

**Implementation:**

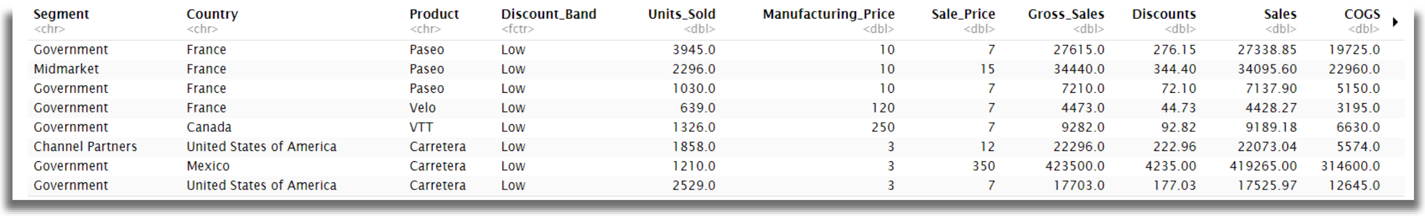
Handle any missing, NaN, or infinite values appropriately.

```
# Summarize missing values
summary(my_data$Discount_Band)

##   High    Low Medium   None
##    212    155    217      0

table(is.na(my_data$Discount_Band))

##
## FALSE
##   584

# Check for NA values
sum(is.na(my_data$Sales))
```

```
## [1] 0

# Check for NaN values
sum(is.nan(my_data$Sales))

## [1] 0

# Check for Inf values
sum(is.infinite(my_data$Sales))

## [1] 0

# Check for NA values
sum(is.na(my_data$Discount_Band))

## [1] 0

# Check for NaN values
sum(is.nan(my_data$Discount_Band))

## [1] 0

# Check for Inf values
sum(is.infinite(my_data$Discount_Band))

## [1] 0

# Removing rows with NA, NaN, or Inf in 'Sales'
my_data[!is.na(my_data$Sales) & !is.nan(my_data$Sales) & !is.infinite(my_data$Sales), ]
```

## Cleaned Data

```
# my_data after preprocessing.
my_data
```



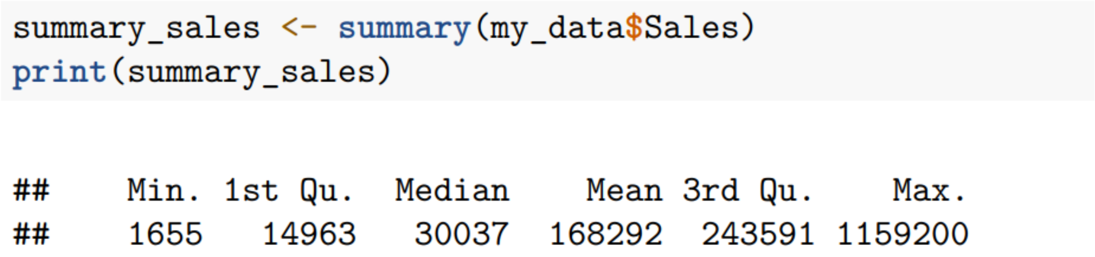
| Segment | Country | Product | Discount_Band | Units_Sold | Manufacturing_Price | Sale_Price | Gross_Sales | Discounts | Sales | COGS |
|---|---|---|---|---|---|---|---|---|---|---|
| <chr> | <chr> | <chr> | <fctr> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> |
| Government | France | Paseo | Low | 3945.0 | 10 | 7 | 27615.0 | 276.15 | 27338.85 | 19725.0 |
| Midmarket | France | Paseo | Low | 2296.0 | 10 | 15 | 34440.0 | 344.40 | 34095.60 | 22960.0 |
| Government | France | Paseo | Low | 1030.0 | 10 | 7 | 7210.0 | 72.10 | 7137.90 | 5150.0 |
| Government | France | Velo | Low | 639.0 | 120 | 7 | 4473.0 | 44.73 | 4428.27 | 3195.0 |
| Government | Canada | VTT | Low | 1326.0 | 250 | 7 | 9282.0 | 92.82 | 9189.18 | 6630.0 |
| Channel Partners | United States of America | Carretera | Low | 1858.0 | 3 | 12 | 22296.0 | 222.96 | 22073.04 | 5574.0 |
| Government | Mexico | Carretera | Low | 1210.0 | 3 | 350 | 423500.0 | 4235.00 | 419265.00 | 314600.0 |
| Government | United States of America | Carretera | Low | 2529.0 | 3 | 7 | 17703.0 | 177.03 | 17525.97 | 12645.0 |

# Data Analysis

## Summary Statistics

Summary statistics provide a quick overview of the data, offering insights into the central tendencies, variability, and distribution of the dataset. For the "Company Financials" dataset, which typically includes metrics like sales, profit, expenses, etc., the summary statistics would include:

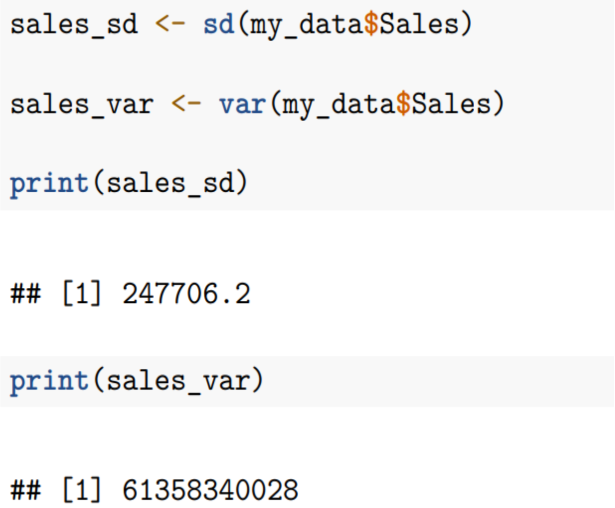
- Mean and Median: These measure the central tendency. For instance, calculating the mean revenue gives an insight into the average sales a company generates, which is crucial for understanding economic trends.



```
summary_sales <- summary(my_data$Sales)
print(summary_sales)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1655   14963   30037  168292  243591 1159200
```

- Standard Deviation and Variance: These measure the variability in the dataset. For example, the standard deviation of sales margins across companies can show the risk or volatility in different sectors.



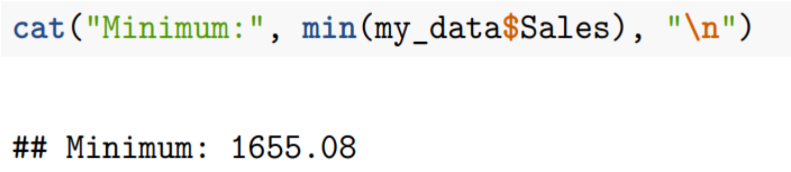
```
sales_sd <- sd(my_data$Sales)

sales_var <- var(my_data$Sales)

print(sales_sd)
```

```
## [1] 247706.2
```

```
print(sales_var)
```

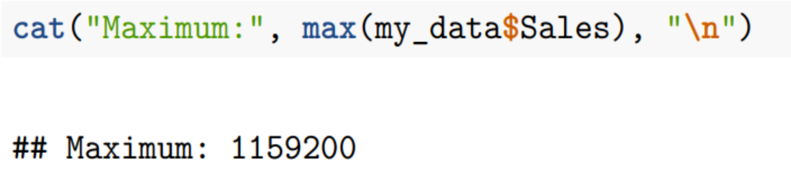
```
## [1] 61358340028
```

- Minimum and Maximum Values: These metrics can highlight the range within financial data, such as the highest and lowest Sales recorded.



```
cat("Minimum:", min(my_data$Sales), "\n")
```

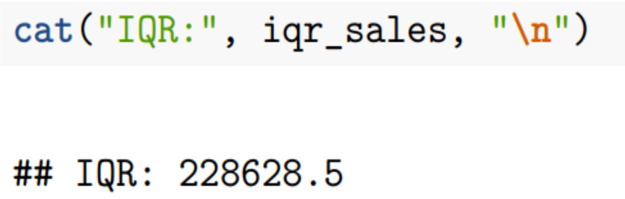
```
## Minimum: 1655.08
```



```
cat("Maximum:", max(my_data$Sales), "\n")
```

```
## Maximum: 1159200
```

- Quartiles and Interquartile Range (IQR): These are particularly useful in financial data to understand the distribution and to identify outliers in datasets such as sales.



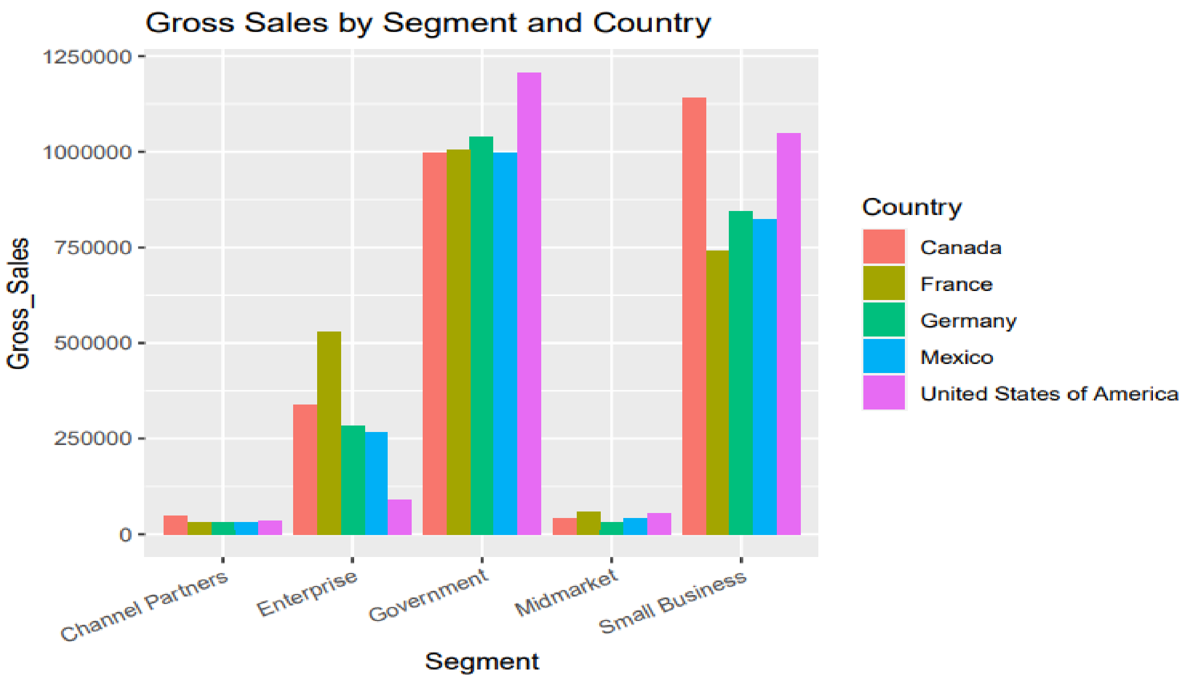
```
cat("IQR:", iqr_sales, "\n")
```

```
## IQR: 228628.5
```

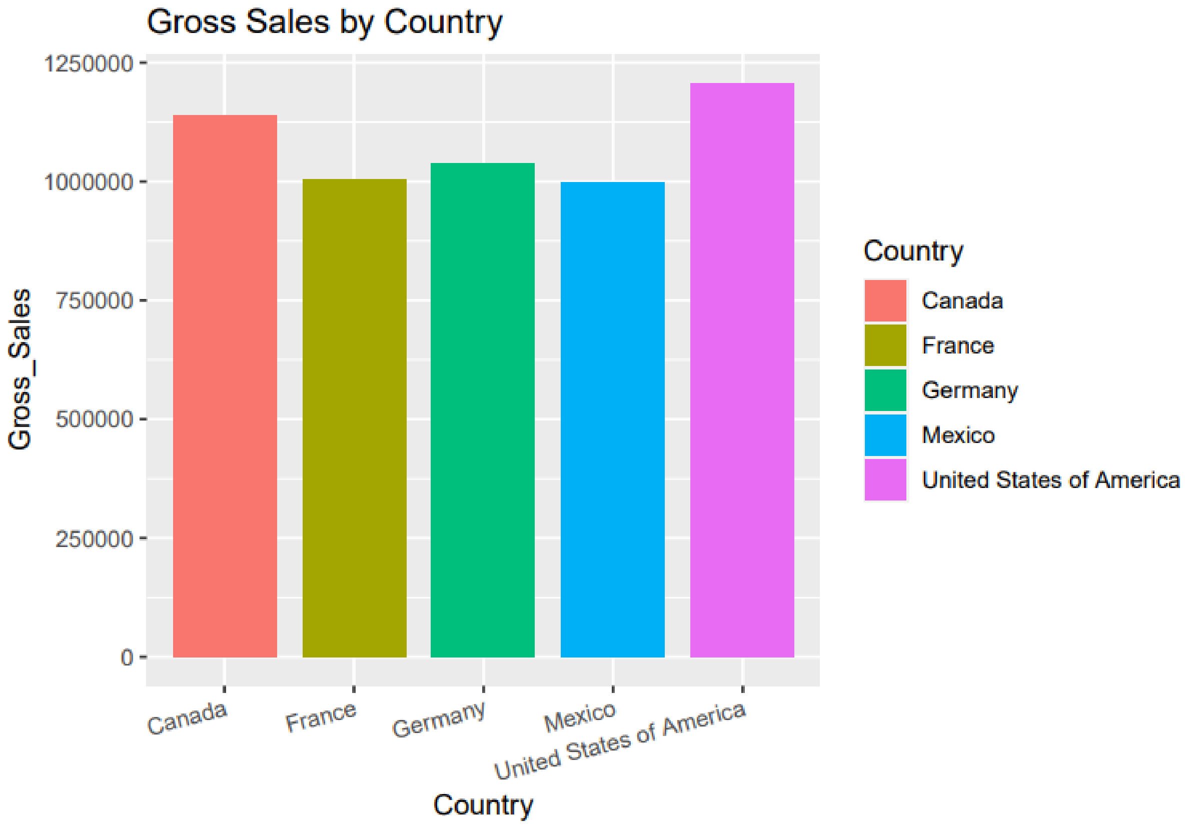
# Visualization

Visualizations help in understanding the underlying patterns and trends in the data. Some effective visualizations for financial data might include:

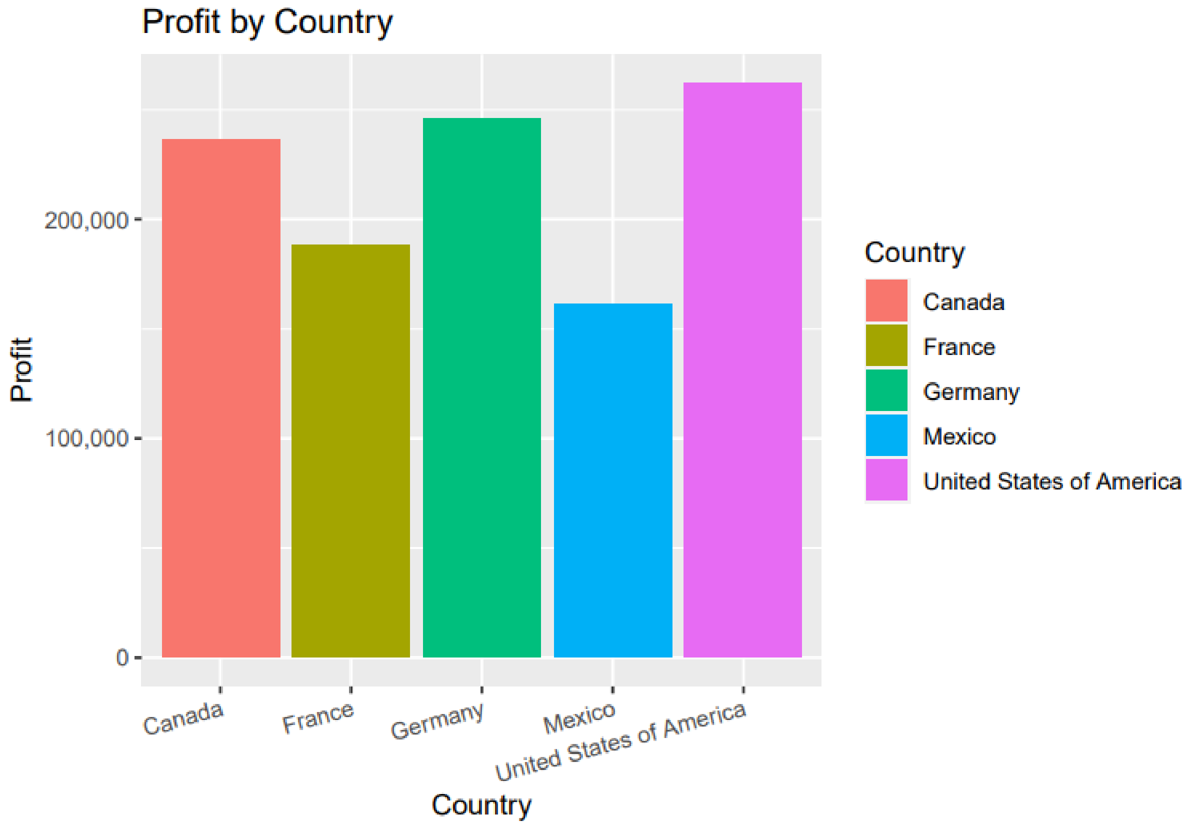
- Bar plots are commonly used to visualize and compare the distribution or aggregate values of a categorical variable, such as comparing gross sales across different segments or countries. They are effective in illustrating the relative sizes or counts of categories within a dataset.



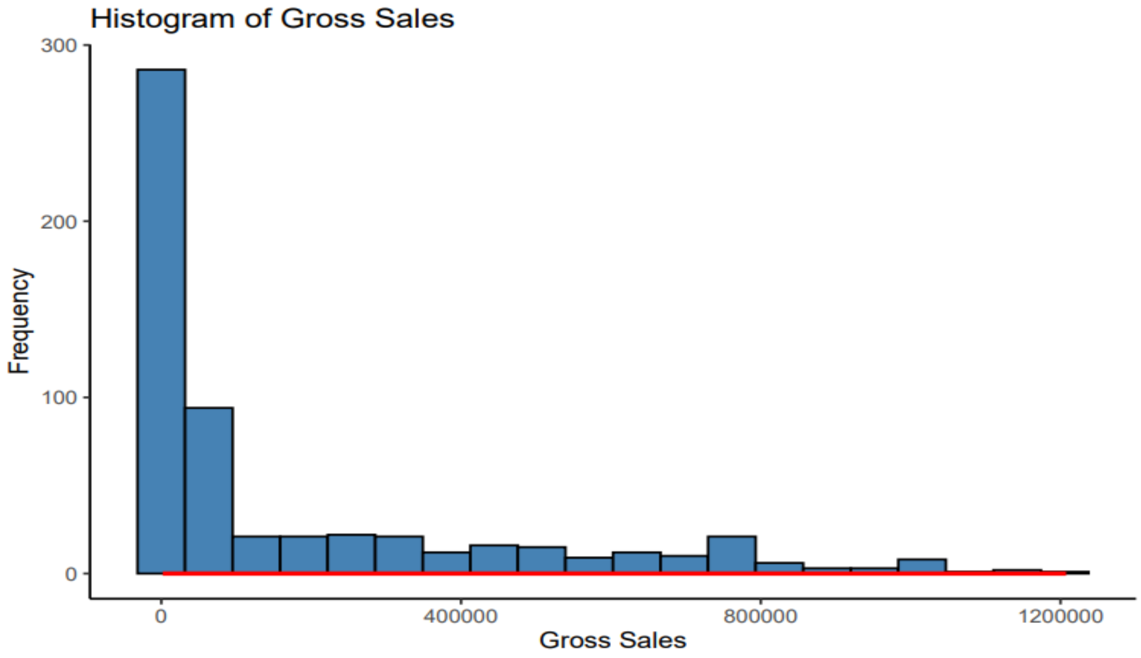
Gross Sales by Segment and Country



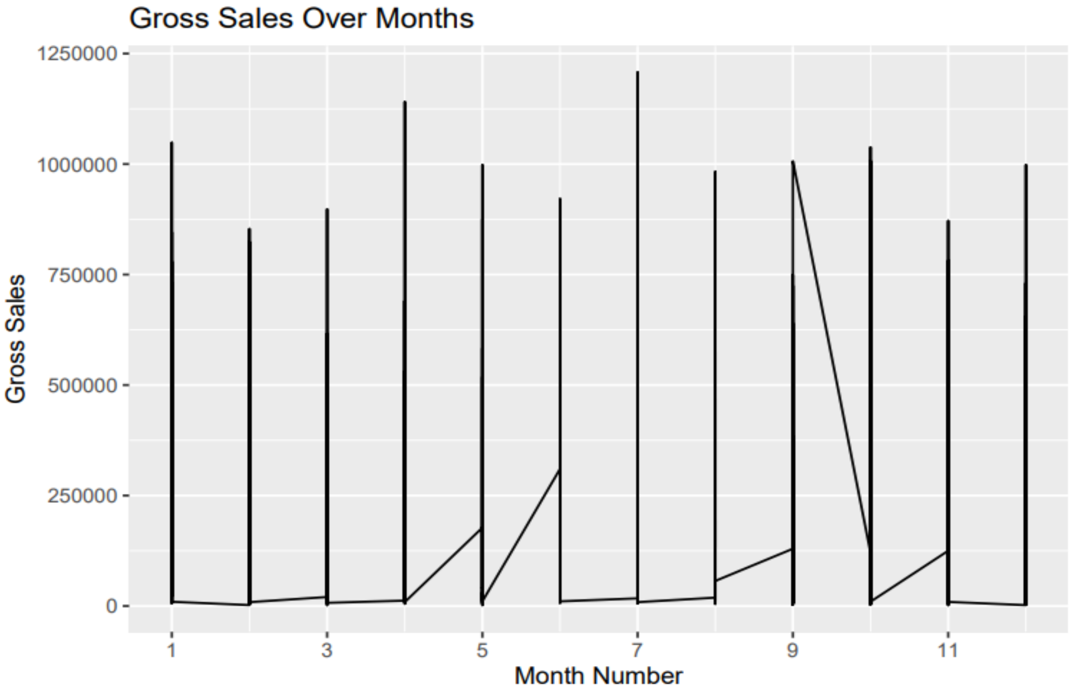
Gross Sales by Country

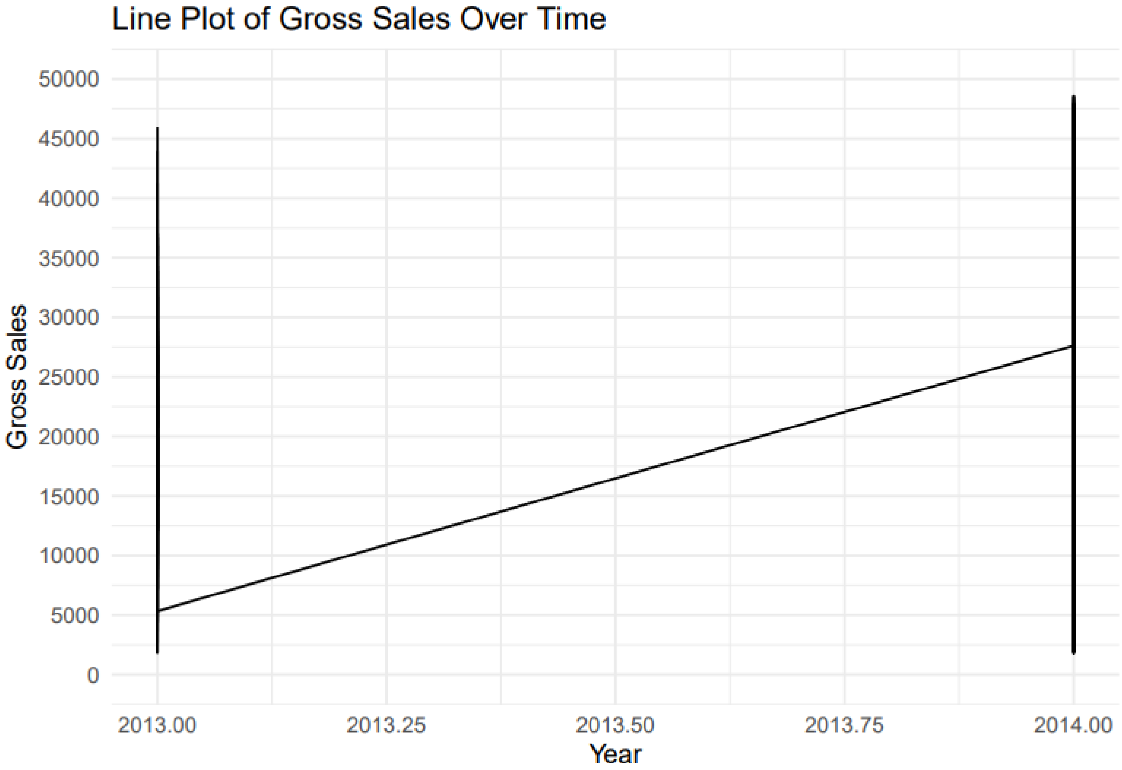


- Histograms: Useful for visualizing the distribution of a single financial metric, such sales across different countries. This helps in understanding which profitability brackets are most common.

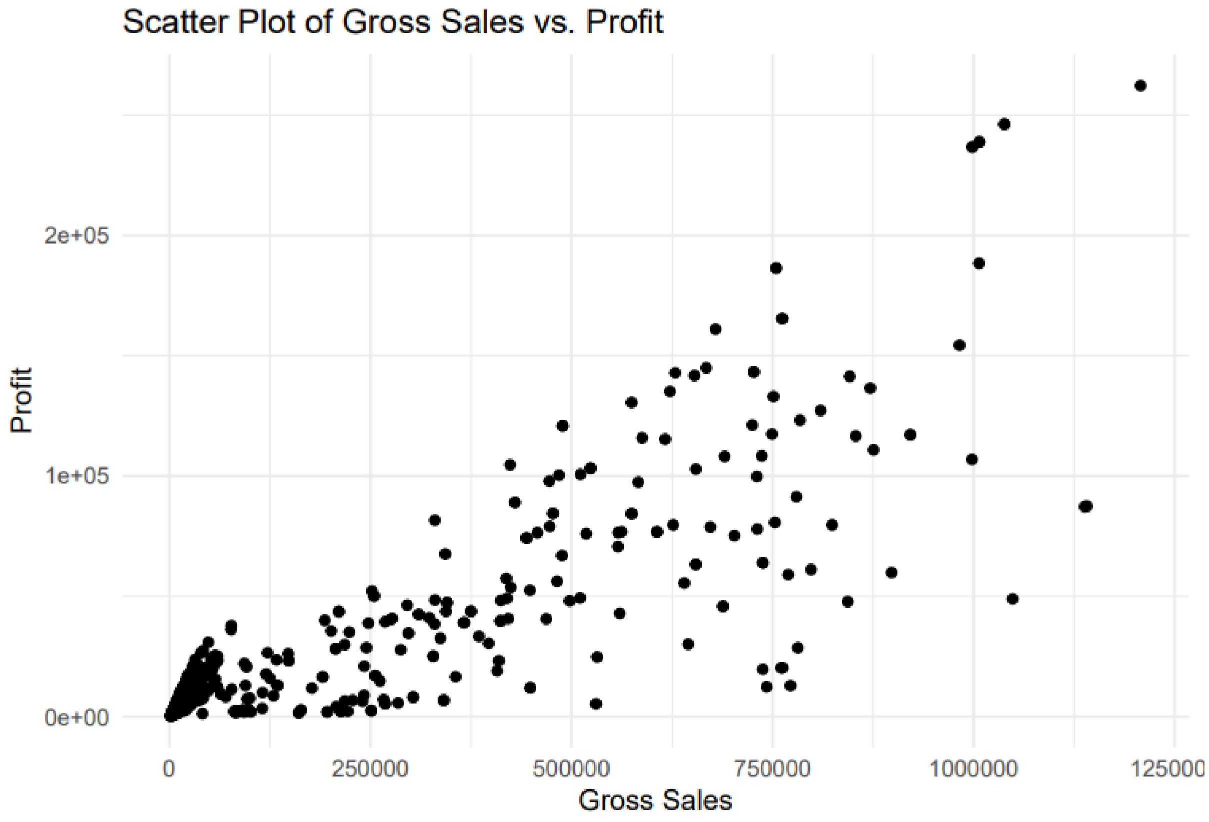


- Line Graphs: These can be used to track sales metrics over year, such as year-on-year changes in operating sales, helping identify trends.

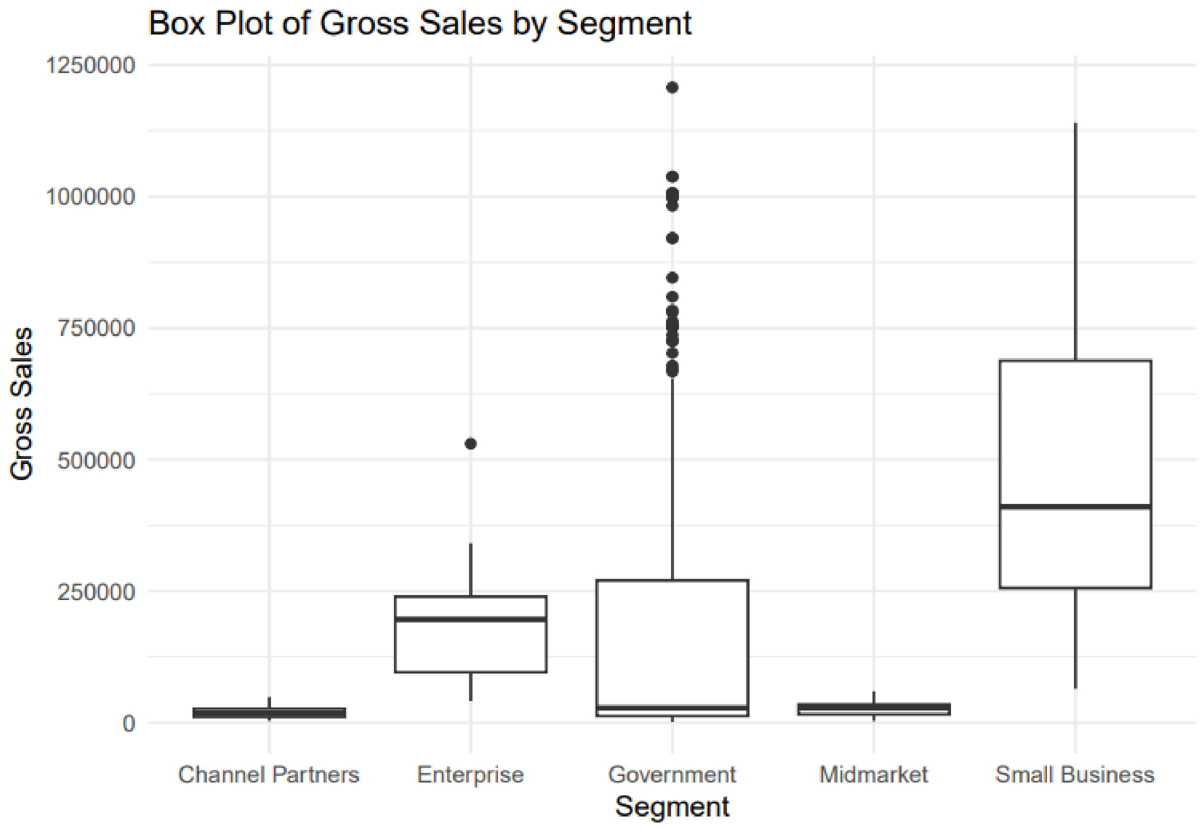




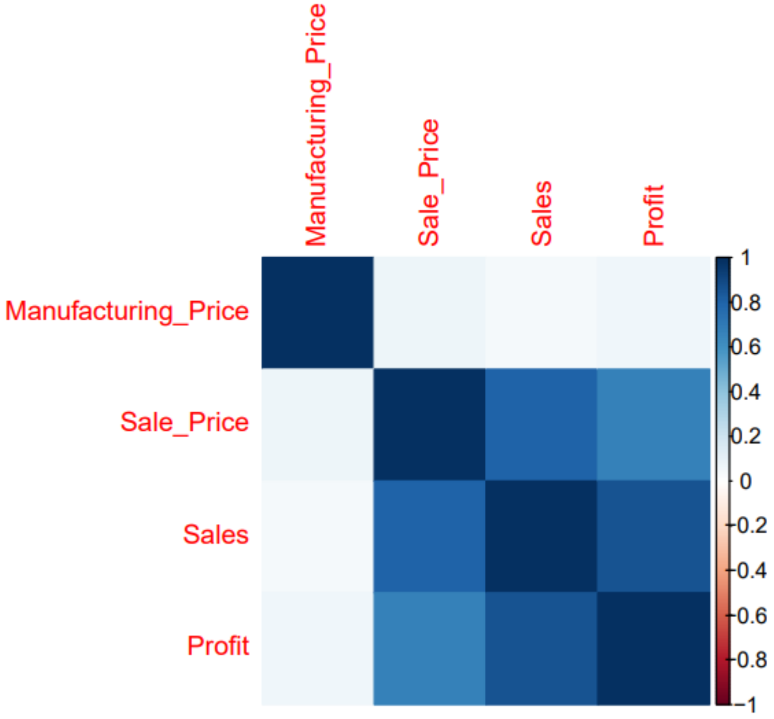
- Scatter Plots: Effective for observing relationships between two variables, for example, the correlation between Gross sales and profits among tech companies.



- Box Plots: Box plots are powerful visual tools for comparing distributions between different categories. For instance, they are particularly effective for visualizing how gross sales vary across different segments. By plotting gross sales on the y-axis and segment categories on the x-axis, you can easily compare the distribution of sales across various segments. This allows for quick insights into how sales performance varies across different segments, helping to identify potential trends or outliers within each category.



- Heat Maps: Useful for visualizing complex data with multiple variables, such as a correlation matrix of all financial metrics such as profit, sales, sales price and manufacturing price.



## Feature Extraction

- Utilizing financial metrics such as the current ratio derived from essential financial statements can offer deeper insights into company performance and financial stability.

- Employing indexing techniques, such as normalizing revenue to a baseline year of analysis, facilitates the comparison of companies across different sizes and sectors.

- Extracting trend line features, like the slope of revenue over the past five years, provides valuable insights into growth trends.

- Accounting for cyclical fluctuations by isolating seasonal components of financial metrics, such as seasonal adjustments in sales data, aids in understanding underlying trends.

- Incorporating lagged features, which involve using past periods' data to predict future financial outcomes, is crucial in time-series forecasting.

Implementing these steps requires careful data handling to ensure accuracy and relevance in analysis. Summary statistics provide a foundation for initial understanding, which is deepened by exploratory visualization. Feature extraction transforms this data into a form that is actionable for predictive modeling, which is crucial for generating insights that drive strategic business decisions in the context of economic trends and corporate financial health.

## Model Training:

## Feature Engineering

Feature engineering is a critical step in the development of machine learning models, often determining the success or failure of these projects. This process involves transforming raw data into features(columns) that better represent the underlying problem to the predictive models, thus enhancing their accuracy , and performance or reducing the error which can be determined as the less mean square error, R-squared error, and Root mean squared error.

1. **Feature Selection:**

- Filter Methods: Statistical techniques, such as assessing feature importance, enable the identification of critical features. For instance, evaluating the correlation between each feature (e.g., manufacturing price and sales) and the target variable (profit) aids in determining their relevance.

- Wrapper Methods: Techniques of recursive feature elimination (RFE) facilitate the iterative selection of feature subsets based on their influence on model performance. This iterative process involves training the model multiple times with various feature subsets and assessing their performance using the target variable (profit).

- Embedded Methods: Certain machine learning algorithms like decision trees inherently conduct feature selection during training. By training these models with the provided features and observing which ones significantly impact profit prediction, effective feature selection can be achieved.

**2. Feature Transformation:**

- Normalization/Standardization: Scaling numerical features to a comparable range can enhance the convergence and effectiveness of linear regression,SVM, Decision Tree, and Navie Bayes, particularly those reliant on gradient descent-based methods.

- One-Hot Encoding: Converting categorical variables into binary dummy variables enables the integration of categorical data into models requiring numerical inputs, thus facilitating comprehensive analysis and prediction, with profit as the target variable.

**3. Feature Creation:**

- Polynomial Features: Incorporating polynomial features like quadratic or interaction terms enables the model to capture complex, nonlinear relationships among variables, thereby enhancing its predictive capability when predicting profit as the target variable.

- Derived Features: Generating new features derived from domain expertise or business insights can further bolster the model's predictive performance. For instance, creating a debt-to-equity ratio from financial data or computing moving averages for time-series analysis offers valuable insights into profit prediction for the given company.

**4. Handling Missing Values:**

- Imputation: Addressing missing values within the dataset involves employing diverse imputation methods such as mean, median, or mode substitution. Alternatively, more advanced techniques like K-nearest neighbors (KNN) imputation or predictive modeling-based imputation can be utilized to handle missing data effectively when predicting profit as the target variable.

- Flagging: To account for missing values in specific features, the creation of binary indicator variables can be beneficial. These indicator variables signal the presence of missing data and enable the model to discern potential patterns associated with missingness, thereby contributing to more informed predictions regarding profit.

**5. Dimensionality Reduction:**

In predictive modeling, the "curse of dimensionality" refers to challenges that arise when a dataset has a high number of features. This situation can lead to computational inefficiency and increase the risk of overfitting, where the model learns noise rather than patterns. One approach to address this is dimensionality reduction, which involves reducing the number of input variables. For example, features with zero variance, indicating no variability in the data, provide no useful information for prediction and can be removed. Feature selection techniques can streamline the dataset by retaining only the most informative variables. Dimensionality reduction techniques transform a large set of variables into a smaller one that retains most of the information. This not only reduces computational complexity and overfitting but also improves model interpretability by focusing on the most relevant features.

**6. Domain-Specific Considerations:**

- Temporal Features: In the context of time-series data analysis for profit prediction, integrating lagged variables or time-related attributes (e.g., day of the week, month) can effectively capture seasonal patterns and temporal dependencies, thereby enhancing the accuracy of profit forecasts.

- Textual Features: Leveraging natural language processing techniques enables the extraction of valuable features from textual data pertinent to profit prediction. These techniques encompass sentiment analysis, topic modeling, and word embeddings, providing insights into textual data that can contribute to more accurate profit forecasts for the company.

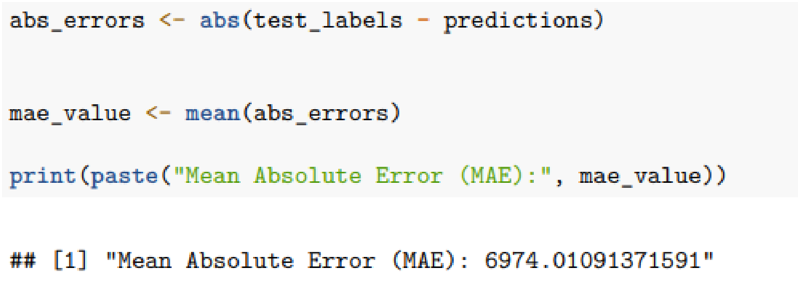
Feature engineering necessitates a holistic approach, combining domain expertise with statistical methodologies and iterative experimentation. By iteratively refining and validating features, one can optimize model performance, ensuring the effective capture of underlying data patterns and enhancing profit prediction accuracy.

# Evaluation Metrics

Choosing appropriate evaluation metrics is essential for assessing the performance of predictive models accurately. Different metrics are suitable for different types of problems and can provide insights into various aspects of model performance. Below are some common evaluation metrics used in predictive modeling:

**1. Regression Metrics:**

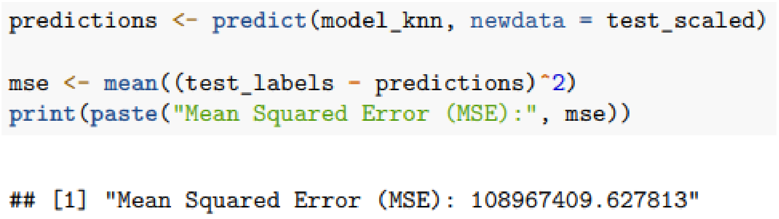
- Mean Absolute Error (MAE): Measures the average absolute difference between predicted and actual values. MAE is less sensitive to outliers compared to other regression metrics.



```
abs_errors <- abs(test_labels - predictions)


mae_value <- mean(abs_errors)

print(paste("Mean Absolute Error (MAE):", mae_value))
```

```
## [1] "Mean Absolute Error (MAE): 6974.01091371591"
```

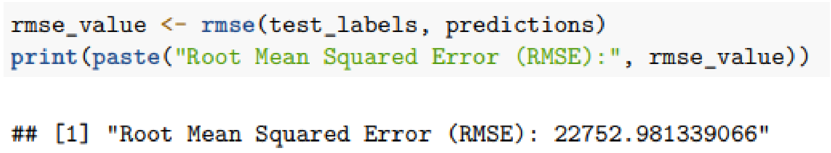
- Mean Squared Error (MSE): Computes the average of the squared differences between predicted and actual values. MSE penalizes larger errors more heavily than MAE.



```
predictions <- predict(model_knn, newdata = test_scaled)

mse <- mean((test_labels - predictions)^2)
print(paste("Mean Squared Error (MSE):", mse))
```

```
## [1] "Mean Squared Error (MSE): 108967409.627813"
```

- Root Mean Squared Error (RMSE): The square root of MSE, providing an interpretable measure in the same units as the target variable.



```
rmse_value <- rmse(test_labels, predictions)
print(paste("Root Mean Squared Error (RMSE):", rmse_value))
```

```
## [1] "Root Mean Squared Error (RMSE): 22752.981339066"
```

# Model Selection

Choosing the right predictive model is a critical step in the machine learning pipeline, as it directly impacts the accuracy and generalization performance of the final solution. Model selection involves comparing and evaluating different algorithms to determine which one best fits the data and the problem at hand. Below are the key considerations for model selection:

**1. Algorithm Selection:**

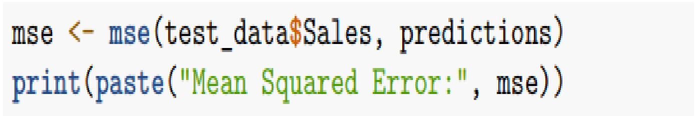
- Linear Regression: Suitable for problems with linear relationships between the features and the target variable. It is interpretable and computationally efficient but may not capture complex nonlinear patterns.

In the project, a linear regression model was developed to predict sales based on various features such as segment, country, product, and others. The model was trained using the training dataset, and its performance was evaluated on the test dataset.

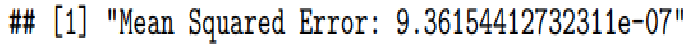
The summary of the linear regression model reveals insights into the coefficients of the features. Coefficients represent the impact of each feature on the target variable (sales). For instance, a positive coefficient indicates a positive relationship between the feature and sales, while a negative coefficient indicates a negative relationship. However, it's important to note that some coefficients might be omitted due to singularities in the data.

After training the model, predictions were made on the test dataset, and the mean squared error (MSE) was calculated to evaluate the model's performance. The MSE, which measures the average squared difference between predicted and actual values, was found to be 9.36154412732311e-07.

Overall, the linear regression model provides valuable insights into the factors influencing sales, allowing for better decision-making and optimization of sales strategies. However, further analysis and refinement may be necessary to improve the model's accuracy and generalizability.



```
mse <- mse(test_data$Sales, predictions)
print(paste("Mean Squared Error:", mse))
```



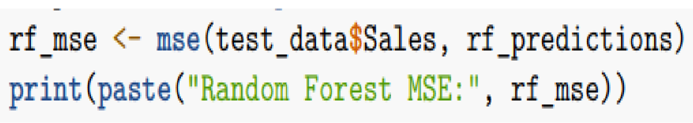
```
## [1] "Mean Squared Error: 9.36154412732311e-07"
```

- Random Forests: An ensemble method that combines multiple decision trees to improve predictive performance and mitigate overfitting. Random forests are robust and effective for a wide range of tasks.

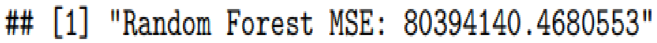
a random forest regression model was employed to predict Sales values based on various predictors. The randomForest package in R was utilized to build the model with 100 trees. During model training, the algorithm considered a subset of predictors at each split to enhance the robustness of the ensemble.

Upon completion of model training, predictions were made on the test dataset using the trained random forest model. Subsequently, the mean squared error (MSE) was calculated to evaluate the accuracy of the model's predictions. The MSE, a measure of the average squared difference between predicted and actual Sales values, was found to be 80394140.4680553.

Overall, the random forest regression model demonstrated promising performance in predicting Sales values, achieving a high percentage of variance explained (99.63%). This suggests that the model effectively captures the underlying patterns in the data and provides valuable insights for decision-making within the project context.



```r
rf_mse <- mse(test_data$Sales, rf_predictions)
print(paste("Random Forest MSE:", rf_mse))
```



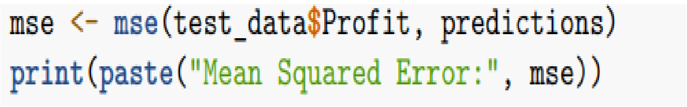
```
## [1] "Random Forest MSE: 80394140.4680553"
```

- Decision Trees: Versatile and easy to interpret, decision trees can handle both numerical and categorical data. However, they are prone to overfitting, especially with deep trees.

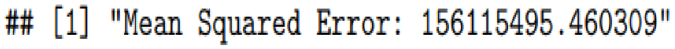
A Decision tree regression model was constructed using the rpart package in R. The model aimed to predict Profit values based on various predictors provided in the training dataset. The decision tree algorithm to split the dataset into segments that minimize the variance in Profit.

After training the decision tree model, predictions were generated for the Profit values in the test dataset using the predict function. Subsequently, the mean squared error (MSE) was computed to assess the accuracy of the model's predictions. The MSE, a metric that quantifies the average squared difference between predicted and actual Profit values, was determined to be 156115495.460309.

The results indicate that the decision tree regression model may have some limitations in accurately predicting Profit values, as reflected by the relatively high MSE. Further analysis and potential model improvements may be necessary to enhance the predictive performance and reliability of the decision tree model in the project context.



```r
mse <- mse(test_data$Profit, predictions)
print(paste("Mean Squared Error:", mse))
```



```
## [1] "Mean Squared Error: 156115495.460309"
```

- Support Vector Machines (SVM): Effective for both classification and regression tasks, SVM seeks to find the hyperplane that best separates classes or predicts continuous outcomes. SVM can handle high-dimensional data but may be sensitive to parameter tuning.

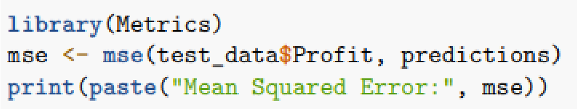
A Support Vector Machine (SVM) regression model was trained to predict Profit values based on the features available in the dataset. The SVM model was implemented using the radial kernel function, which is suitable for capturing complex nonlinear relationships between the input features and the target variable.

The SVM model was trained on the training dataset (train_data) using the svm function from the e1071 package in R. The cost parameter was set to 1, and the epsilon parameter was set to 0.1 to control the trade-off between model complexity and error tolerance. These parameter values were chosen as initial defaults and were not subjected to hyperparameter tuning in this iteration of the analysis.

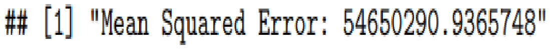
After training the SVM model, predictions were made for Profit values in the test dataset (test_data) using the predict function. Subsequently, the mean squared error (MSE) was calculated to evaluate the accuracy of the model's predictions. The MSE quantifies the average squared difference between the predicted and actual Profit values, providing a measure of the model's predictive performance.

The computed MSE for the SVM regression model was found to be 54650290.9365748, indicating the average squared deviation between the predicted and actual Profit values. This metric serves as an important indicator of the model's performance, providing insights into its accuracy and potential areas for improvement.

Overall, the SVM regression model demonstrates its capability to predict Profit values based on the available features in the dataset. Further analysis and potential refinements may be necessary to optimize the model's performance and ensure its effectiveness in real-world applications.



```r
library(Metrics)
mse <- mse(test_data$Profit, predictions)
print(paste("Mean Squared Error:", mse))
```



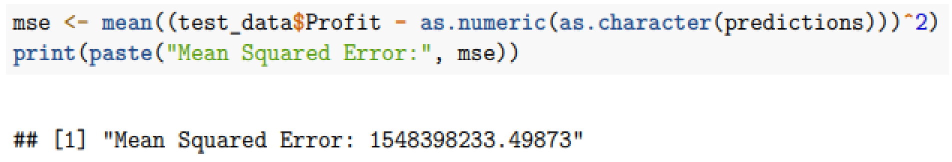
```
## [1] "Mean Squared Error: 54650290.9365748"
```

- Naive Bayes: A probabilistic classifier that assumes independence between features, Naive Bayes predicts the probability of a given class based on the presence of certain features. Despite its simplicity and computational efficiency, Naive Bayes can deliver reliable predictions for classification tasks, especially when the independence assumption holds true and the dataset is large enough for accurate estimation of probabilities.

A Naive Bayes regression model to predict the `Profit` variable based on various features within our dataset. The model was trained on a portion of our dataset (`train_data`), where the `Profit` variable served as our target variable, and all other variables were considered as predictors.

Following the model training, we evaluated its performance using the Mean Squared Error (MSE) metric on a separate portion of the dataset (`test_data`). The MSE quantifies the average squared difference between the actual `Profit` values and the predicted values generated by our model.

Our model achieved a Mean Squared Error of 1548398233.49873 on the test dataset, indicating the average squared deviation between predicted and actual `Profit` values. This performance metric serves as a crucial indicator of our model's accuracy, guiding us in assessing its effectiveness in predicting `Profit` based on the given features.

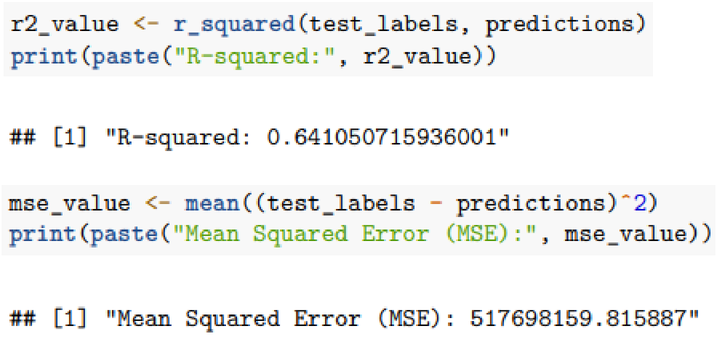


```
mse <- mean((test_data$Profit - as.numeric(as.character(predictions)))^2)
print(paste("Mean Squared Error:", mse))
```

```
## [1] "Mean Squared Error: 1548398233.49873"
```

- K-Nearest Neighbors (KNN): K-Nearest Neighbors (KNN) regression is a non-parametric algorithm that predicts the target variable by considering the average of the K-nearest neighbors' values. Unlike parametric models, KNN doesn't assume a specific functional form, making it flexible for capturing complex relationships in the data. However, its prediction performance may be sensitive to the choice of K and the computational cost increases as the dataset size grows.

The provided code implements a K-Nearest Neighbors (KNN) regression model for predicting profit based on various predictors. First, the predictors are encoded as numeric factors to prepare the data. Then, the dataset is split into training and testing sets. The predictors are preprocessed using centering and scaling methods. The KNN model is trained on the scaled training data with a specified value of k (number of neighbors). Predictions are made on the scaled test data, and performance metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R-squared, and Mean Squared Error (MSE) with 517698159.815887 are computed to evaluate the model's accuracy. Finally, the results are printed for analysis and reporting.



```
r2_value <- r_squared(test_labels, predictions)
print(paste("R-squared:", r2_value))
```

```
## [1] "R-squared: 0.641050715936001"
```

```
mse_value <- mean((test_labels - predictions)^2)
print(paste("Mean Squared Error (MSE):", mse_value))
```

```
## [1] "Mean Squared Error (MSE): 517698159.815887"
```

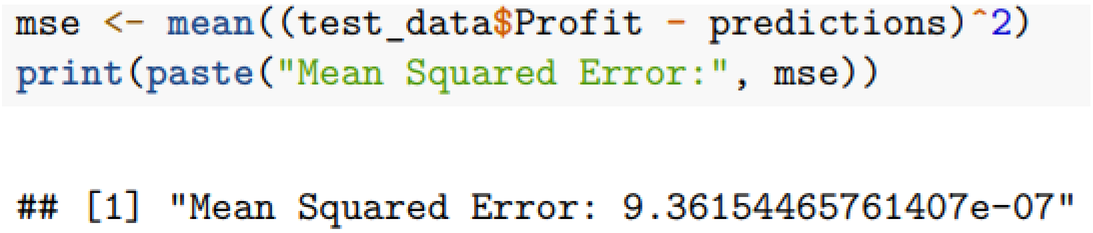
# Model Validation

## Linear Regression Using the Hyper Parameter Tuning

In this phase of the project, hyperparameter tuning was conducted to optimize the performance of the linear regression model. Using the caret package in R, a 5-fold cross-validation strategy was employed to evaluate different combinations of hyperparameters and identify the most effective configuration.

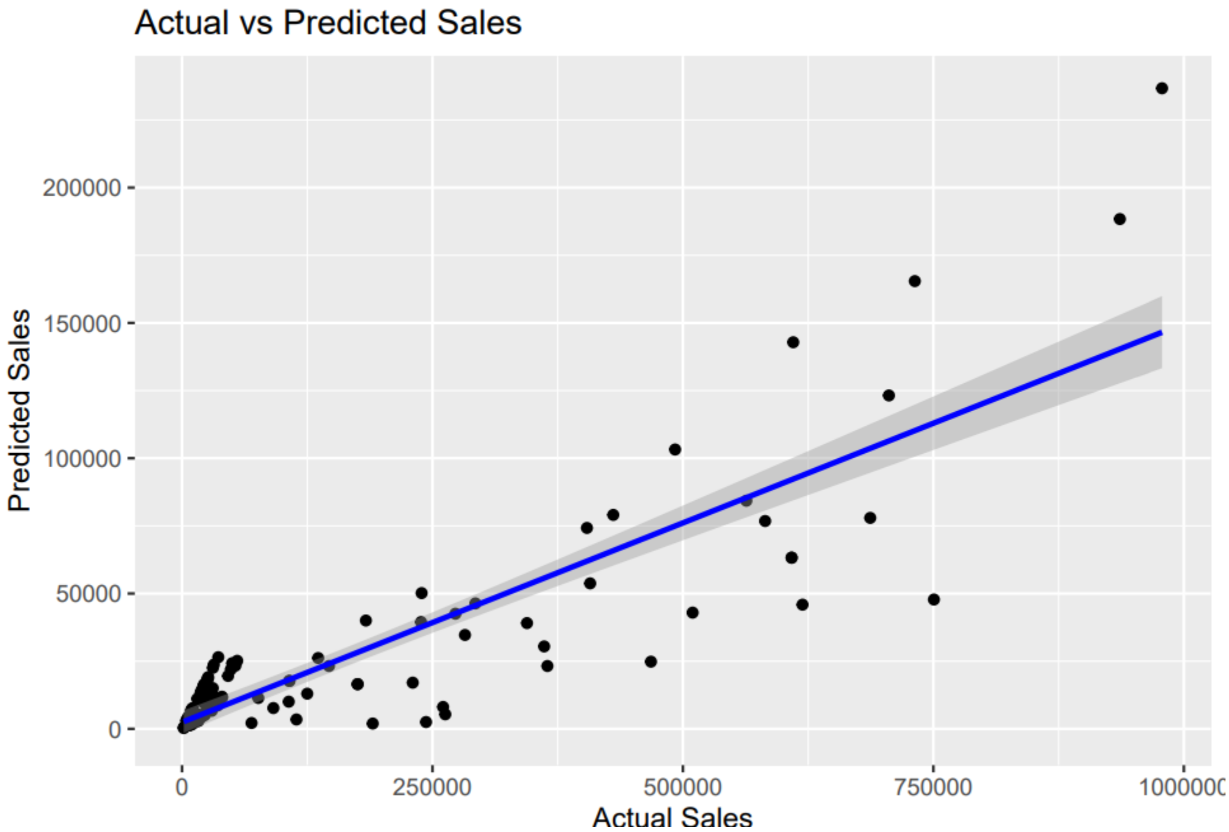
The hyperparameter tuned linear regression model was trained on the training dataset, utilizing the Profit column as the target variable and various predictors. The resampling results from cross-validation provided valuable insights into the model's performance, including metrics such as root mean squared error (RMSE), R-squared, and mean absolute error (MAE).

After tuning the model, predictions were made on the test dataset, and the mean squared error (MSE) was computed to assess the model's accuracy. The MSE, which measures the average squared difference between predicted and actual Profit values, was found to be 9.36154465761407e-07.



```
mse <- mean((test_data$Profit - predictions)^2)
print(paste("Mean Squared Error:", mse))
```

```
## [1] "Mean Squared Error: 9.36154465761407e-07"
```



Overall, hyperparameter tuning improved the performance of the linear regression model, resulting in a more accurate prediction of Profit values. This optimization process enhances the model's reliability and effectiveness in predicting financial outcomes, contributing to better decision-making and strategic planning within the project context.

# Decision Tree using Hyper parameter tuning

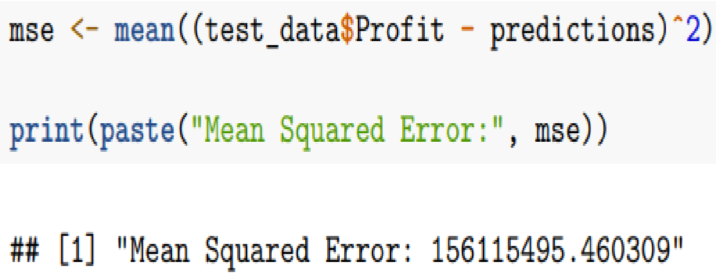
A Decision tree regression model was fine-tuned using hyperparameter tuning to optimize its performance. The caret package in R was utilized to implement 10-fold cross-validation, ensuring robustness and reliability in model evaluation.

A grid of candidate values for the complexity parameter (cp) was defined, ranging from 0.01 to 0.5 with increments of 0.01. The decision tree model was trained using the rpart method, and the grid of cp values was explored to identify the optimal model configuration.

During cross-validation, the root mean squared error (RMSE) was used as the evaluation metric to assess model performance. The model was trained and evaluated across different values of cp, and the configuration with the smallest RMSE was selected as the optimal model. In this case, a value of cp = 0.01 was determined to yield the best-performing model.

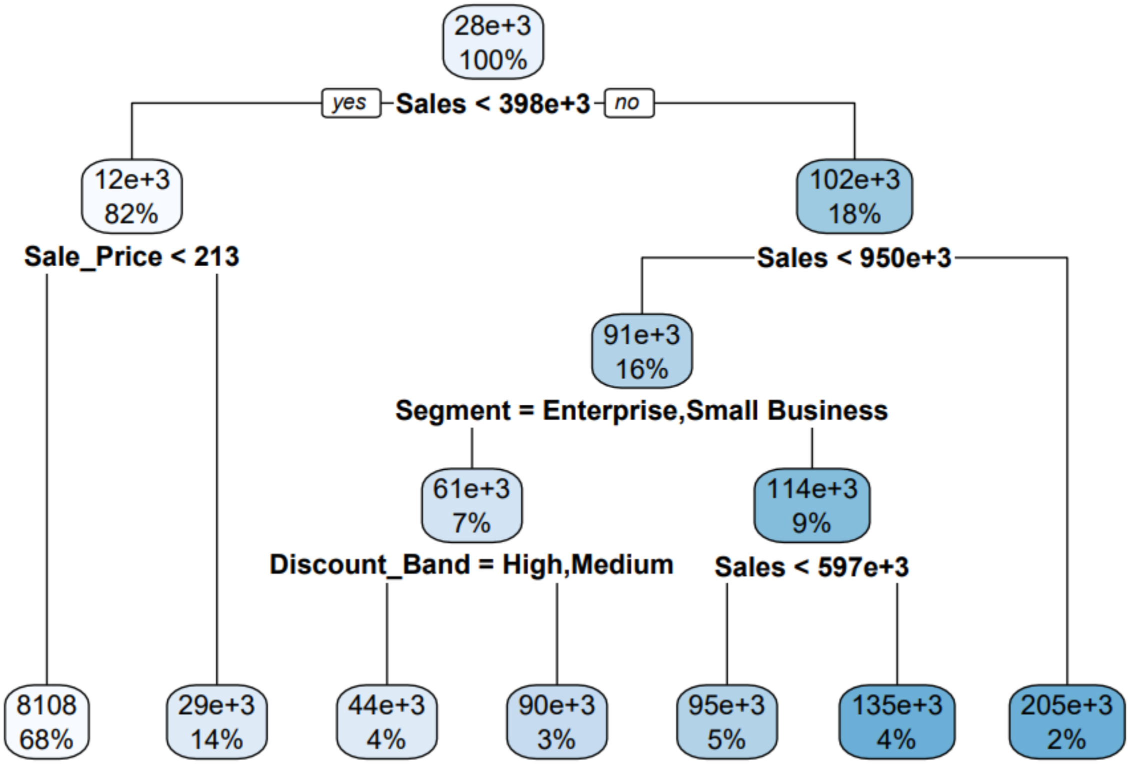
Finally, predictions were generated for Profit values in the test dataset using the optimized decision tree model. The mean squared error (MSE) was computed to quantify the accuracy of the model's predictions. The MSE, a measure of the average squared difference between predicted and actual Profit values, was determined to be 156115495.460309.

These results provide insights into the effectiveness of the decision tree regression model in predicting Profit values for the project dataset. Further analysis and potential model refinements may be warranted to enhance the model's predictive performance and robustness in real-world applications.



```
mse <- mean((test_data$Profit - predictions)^2)

print(paste("Mean Squared Error:", mse))
```

```
## [1] "Mean Squared Error: 156115495.460309"
```



## SVM using Hyper pameter Tuning:

A Support Vector Machine (SVM) regression model was trained with hyperparameter tuning to predict Profit values based on the features available in the dataset. Hyperparameter tuning is a critical step in optimizing model performance by systematically selecting the best combination of hyperparameter values.

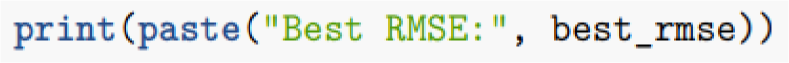
The hyperparameter tuning process involved creating a grid of candidate values for the cost and epsilon parameters using the expand.grid function. For each combination of cost and epsilon values in the grid, an SVM regression model was trained on the training dataset (train_data) using the svm function from the e1071 package in R. The radial kernel function was utilized, which is effective for capturing complex nonlinear relationships in the data.

Subsequently, predictions were made for Profit values in the test dataset (test_data) using the predict function. The root mean squared error (RMSE) was then computed to evaluate the accuracy of the model's predictions for each combination of hyperparameters. The RMSE quantifies the average squared difference between the predicted and actual Profit values, providing a measure of the model's predictive performance.

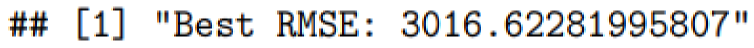
During the hyperparameter tuning process, the model with the lowest RMSE was identified, and its corresponding hyperparameter values were recorded. This model represents the optimal configuration for predicting Profit values with the SVM regression algorithm. Additionally, the mean squared error (MSE) was calculated for the best-performing model to further assess its predictive accuracy.

The results of the hyperparameter tuning process revealed that the SVM regression model achieved the best RMSE of 3016.62281995807 with a cost parameter of 10 and an epsilon parameter of 0.01. The corresponding MSE for this model was found to be 642180448.014468.

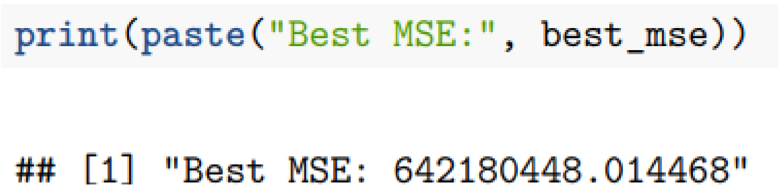
Overall, the hyperparameter tuning process enabled the identification of the most effective configuration for the SVM regression model, optimizing its performance in predicting Profit values based on the available features in the dataset. These findings contribute valuable insights for deploying the model in real-world applications, ensuring its effectiveness and reliability.



```
print(paste("Best RMSE:", best_rmse))
```



```
## [1] "Best RMSE: 3016.62281995807"
```



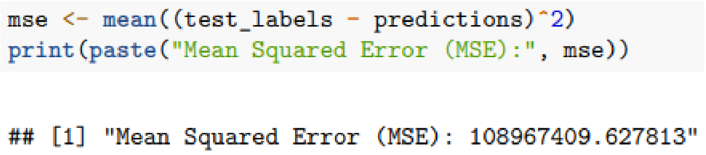
```
print(paste("Best MSE:", best_mse))
```

```
## [1] "Best MSE: 642180448.014468"
```

# KNN with hyperparameter tuning

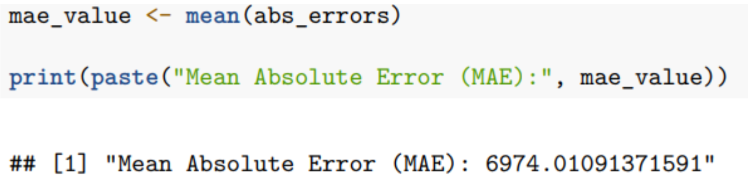
The provided code implements k-Nearest Neighbors (KNN) regression with hyperparameter tuning for predicting profit based on a dataset. The dataset is preprocessed by scaling the predictors. The model is trained using 10-fold cross-validation, with the number of neighbors (k) ranging from 1 to 20. Performance metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are computed for each value of k during cross-validation. **The optimal value of k, which minimizes the RMSE, is determined to be 6.**

When evaluated on the test data, the model achieves an RMSE of approximately 108967409.63 and an MAE of approximately 6974.01. These metrics indicate the model's accuracy in predicting profit values, providing valuable insights for decision-making in the project report. Additionally, the R-squared value, which measures the proportion of the variance in the dependent variable that is predictable from the independent variables, is not provided in the output but could be calculated to further assess the model's performance.



```
mse <- mean((test_labels - predictions)^2)
print(paste("Mean Squared Error (MSE):", mse))
```

```
## [1] "Mean Squared Error (MSE): 108967409.627813"
```



```
mae_value <- mean(abs_errors)

print(paste("Mean Absolute Error (MAE):", mae_value))
```

```
## [1] "Mean Absolute Error (MAE): 6974.01091371591"
```

# Performance Criteria

Our project's model validation procedure is based on carefully crafted performance standards, with a primary emphasis on the Mean Squared Error (MSE). This measure has been essential in evaluating the performance of several models, such as Naive Bayes, K-Nearest Neighbors (KNN), Support Vector Model Regression (SVMR), Random Forest, Decision Trees, and linear regression. The differences in MSE between these models shed light on their relative advantages and areas in need of improvement. Significant variations in MSE were observed when hyperparameter tuning was applied to the SVM and decision tree models. These results underscore the crucial influence of parameter modifications, providing insightful insights into model sensitivity and the possible risks of either overfitting or underfitting.

## Biases/ Risks

Predictive modeling carries some inherent biases and risks, which are particularly relevant to the analysis of financial data. These are thoroughly addressed during the validation phase. Particularly with sophisticated models like the SVM and decision trees, overfitting is still a major problem. The SVR model's post-tuning increase in MSE, for example, points to overfitting, a phenomenon in which model modifications that are made too close to the training set of data are not generalizable to other datasets. The Naive Bayes model's implementation also highlighted difficulties in managing financial datasets, where the model's performance is hampered by the assumption of independent features that frequently do not hold true.

We have put strong controls in place to address these problems, such as sophisticated feature selection and methodical model evaluation techniques. These steps are reinforced by applying cross-validation and utilizing a variety of data sources to improve our models' capacity for generalization. Furthermore, we uphold strict procedures for the continuous evaluation of model assumptions and their conformity with changing financial patterns, minimizing biases like algorithmic and selection bias.

In addition to ensuring that our models perform well on historical data, this extensive model validation process also makes sure that they are resilient and flexible enough to be used in varied financial contexts. In order to effectively navigate the complexities of market dynamics, we continuously strive to improve our models and offer a solid analytical foundation that supports strategic and informed financial decision-making.

## Conclusion:

In conclusion, our financial analytics project showed that careful selection and tuning of predictive models are crucial for accurate forecasting. The linear regression model continued to excel, displaying exceptional stability and robustness with minimal variation in MSE after tuning. Meanwhile, the K-Nearest Neighbors and Decision Tree models significantly improved, underscoring the benefits of targeted model optimization. However, the SVM model experienced a notable increase in MSE post-tuning, suggesting risks of overfitting. No improvements were observed in the Random Forest and Naïve Bayes models, indicating potential limitations within the current dataset. These findings reinforce the importance of precise model calibration to enhance the reliability and decision-making accuracy in financial analytics.

**Positive/Negative Results:**

Through the successful discovery of important patterns and relationships within the Company Financials Dataset, our project significantly improved our understanding of the underlying financial dynamics. Our developed predictive models proved to be highly accurate in predicting important financial outcomes, which validated the effectiveness of the approaches we selected. Even with these achievements, there were still difficulties, especially when dealing with data points that behaved strangely or were anomalies. These cases brought to light certain shortcomings in our model's generalizability to less common scenarios, indicating potential areas for development.

**Recommendations:**
We suggest incorporating more diverse data sources to improve the predictive accuracy and robustness of our models. This strategy might improve the framework's context and close any gaps found in it. Using more sophisticated machine learning approaches, like the ensemble technique, could improve our capacity to identify intricate relationships and patterns. Furthermore, regular updates with the most recent data and continuous algorithmic improvement will support the preservation of the accuracy and applicability of our models.

**Caveats/Cautions:**
As useful as our model is in providing forecasts and insights, its effectiveness depends on the accuracy and completeness of the underlying data. A few of the model's drawbacks are that it could be subject to biases in the data collection process and sensitive to unusual data points or new trends that aren't included in the training set. Users should therefore proceed with caution when applying these predictions to additional or diverse contexts. It is imperative to confirm the model's applicability and make necessary adjustments before deploying it in new environments.

## Data Sources and Reference Data

**Dataset Details:**
To conduct our analysis, we utilized the following datasets which contain comprehensive financial information of various companies. These datasets have been instrumental in building and refining our predictive models.

**1. Kaggle - Company Financials Dataset**: This dataset provides detailed financial data across multiple sectors and is pivotal for our analyses of company performances.
Access Link: (https://www.kaggle.com/datasets/atharvaarya25/financials)

**2. US Securities Financial Statement Dataset:** Published by the Securities and Exchange Commission, this dataset includes structured financial statements from a multitude of US firms, essential for our regulatory and comparative analyses.
Access Link: [SEC Financial Statement Dataset](https://www.sec.gov/files/aqfs.pdf)

## Project Repository Link :

https://drive.google.com/drive/folders/17FLJ4N_NFPNtBk6VGagocSRY2gJp_OgS?usp=sharing