Exposys Data Labs
Bengaluru, Karnataka, 560064

Internship report on

“**Data science** - Profit Prediction Model for Companies”

By

Chiranjeevi M N
BMS Institute of Technology and Management

Under the guidance of
Exposys Data Labs

Abstract

This project involves the development of a machine learning model to predict company profits using R&D Spend, Administration Cost, and Marketing Spend data from a dataset comprising 50 companies. The goal of the study is to use the features at our disposal to estimate the profit from start-up datasets. For this particular issue statement, we are using the 50_Startups dataset. Due to the fact that startups are not well-balanced businesses that have completed the journey from an idea to a product, no established investor will support businesses that lack market value. As a result, start-ups enable early investors to begin providing seed funding, which will enable them to develop their ideas into products. All things considered, it is evident that managing, analysing, and turning a profit on assets is difficult. There is no business that doesn't aim to maximize profits while minimizing expenses. Hence, when it comes to the necessary investments and costs, it is imperative to spend as little as possible and obtain the most analysis possible. We will project the greatest profit for the company based on the R&D Spend, Administration Cost, and Marketing Spend data, and their relative values. Additionally, we are attempting to forecast start-up profits with the highest level of precision possible.

Table of Contents

Chapter Number	Name of chapter	Page Number
01	Introduction	01
02	Existing Method	02
03	Proposed method with Architecture	03
04	Methodology	05
05	Implementation	07
06	Conclusion	13

1. Introduction

In this project, we endeavour to develop a machine learning model for predicting company profits based on the financial attributes of R&D Spend, Administration Cost, and Marketing Spend, using a dataset encompassing 50 companies. Various regression algorithms, including Linear Regression, Decision Tree Regression, and Random Forest Regression, will be employed to capture the relationships between input features and profit. The dataset will be divided into training and test sets for model evaluation, and diverse regression metrics such as Mean Squared Error, Mean Absolute Error, and R-squared will be calculated to assess model performance. The ultimate goal is to identify the most effective regression model, ensuring accurate profit predictions for companies based on their financial expenditures. This project aims to provide valuable insights to decision-makers, aiding them in making informed business decisions regarding potential profits associated with investments in R&D, administration, and marketing. In addition to constructing and evaluating various regression algorithms, this project will involve exploratory data analysis (EDA) to gain a deeper understanding of the dataset's characteristics and relationships between variables. Data preprocessing steps, such as handling missing values and scaling features, will be implemented to ensure the quality and reliability of the model. Furthermore, feature importance analysis will be conducted to identify the most influential factors contributing to profit predictions. The project also emphasizes interpretability, seeking to provide not only accurate predictions but also insights into the key drivers of company profitability. Through a comprehensive approach, encompassing algorithmic diversity, data exploration, preprocessing, and interpretability, the project aims to deliver a robust and reliable machine learning model for profit prediction in the business context.

2. Existing System

Current system Numerous methods may now be in place that make an effort to forecast a company's profit value based on its expenditures, including marketing, administrative, and research and development spending. Many of these systems, meanwhile, might be dependent on labour-intensive calculations or rudimentary statistical methods, which might not be sufficient to fully represent the intricate interactions among various variables.

Conversely, machine learning models have the ability to learn from data and produce precise predictions by identifying patterns in the data. Predicting continuous goal variables like profit has been a common application of linear regression models in this context. By fitting a linear equation to the data, the model calculates the relationship between the independent variables and the dependent variable.

Nonetheless, a lot of the current linear regression models might not be performing to their full potential because they aren't tailored to the unique characteristics of the data. As a result, an ML model that is especially made to precisely forecast a company's profit value based on its expenses is required. considering every pertinent aspect of the data.

2.1 Problem in existing system

The existing methods for predicting a company's profit based on expenses exhibit several limitations that hinder their accuracy and effectiveness. One significant problem lies in the reliance on manual calculations or basic statistical techniques. These traditional approaches often lack the sophistication to capture the intricate and nonlinear relationships inherent in financial data, particularly when dealing with multiple variables such as R&D spend, administration costs, and marketing expenses.

In essence, the problem with existing methods lies in their limited capacity to adapt to the nuanced patterns and dependencies present in financial data, which may hinder their ability to provide accurate and reliable profit predictions for companies. This underscores the need for a more advanced machine learning model specifically tailored to address these challenges and optimize prediction accuracy by considering all relevant features of the data.

3. Proposed System

The main goal of the proposed system is to create a predictive model that can estimate the profit of a company using features such as R&D Spend, Administration Cost, and Marketing Spend.

The proposed system aims to develop a predictive model for estimating the profit of companies based on their R&D Spend, Administration Cost, and Marketing Spend. Beginning with the exploration and preprocessing of the dataset, the system involves careful consideration of data integrity, feature relevance, and correlation analysis. After splitting the data into training and testing sets, multiple regression algorithms, including Linear Regression, Decision Tree Regression, and Random Forest Regression, are employed to train the model. Evaluation metrics such as Mean Squared Error, Mean Absolute Error, and R-squared are then utilized to assess the performance of each model. The system selects the best-performing model for deployment, allowing for predictions on new data. This machine learning-driven approach facilitates data-driven decision-making for companies, offering insights into optimal resource allocation and investment strategies. Regular monitoring and potential updates ensure the model's continued accuracy and relevance in dynamic business environments, contributing to enhanced financial planning and risk management capabilities.

3.1 Algorithm

Here's a simplified representation of the algorithm with one line per step for each of the regression models:

Linear Regression:

1. Load and Explore Data: Load dataset; explore.

2. Split Data: Split into features and target; further split into training and testing sets.

3. Train Model: Import Linear Regression; initialize; train on the training set.

4. Make Predictions: Predict profits for the testing set.

5. Evaluate Model: Calculate metrics (MSE, MAE, R-squared) on the testing set.

Decision Tree Regression:

6. Train Model: Import Decision Tree Regression; initialize; train on the training set.

7. Make Predictions: Predict profits for the testing set.

8. Evaluate Model: Calculate regression metrics on the testing set.

Random Forest Regression:

9. Train Model: Import Random Forest Regression; initialize; train on the training set.

10. Make Predictions: Predict profits for the testing set.

11. Evaluate Model: Calculate regression metrics on the testing set.

This provides a high-level overview of each step in the algorithm for Linear Regression, Decision Tree Regression, and Random Forest Regression.

Objective:

The main goal of the proposed system is to create a predictive model that can estimate the profit of a company using features such as R&D Spend, Administration Cost, and Marketing Spend.

4. Methodology

In the context of the given problem of predicting the profit value of a company based on R&D Spend, Administration Cost, and Marketing Spend, a proposed system would involve the implementation of a machine learning model.

1. Data Loading and Exploration:

- Load the dataset containing information about R&D Spend, Administration Cost, Marketing Spend, and Profit for 50 companies.

- Explore the dataset to understand its structure and characteristics.

2. Data Preprocessing:

- Handle any missing values or outliers in the dataset.

- Ensure that the data types are appropriate for each feature.

- Explore correlations between features and the target variable.

3. Feature Selection:

- Determine the relevant features for predicting profit.

- Exclude any irrelevant or highly correlated features.

4. Data Splitting:

- Divide the dataset into training and testing sets to evaluate the model's performance.

- Typically, a common split is 80% for training and 20% for testing.

5. Model Selection:

- Implement multiple regression algorithms to train the model. In the example, Linear Regression, Decision Tree Regression, and Random Forest Regression were used.

6. Model Training:

- Train each regression model using the training set.

- The models learn the relationships between R&D Spend, Administration Cost, Marketing Spend, and Profit during this phase.

7. Model Evaluation:

- Make predictions using the testing set for each model.

- Evaluate the models using regression metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared.

8. Model Comparison:

- Compare the performance of each model based on the evaluation metrics.

- Choose the model that demonstrates the best predictive performance.

9. Model Deployment:

- Once the best model is selected, deploy it for making predictions on new, unseen data.

- The deployed model can be used to estimate the profit of a company based on its R&D Spend, Administration Cost, and Marketing Spend.

10. Monitoring and Maintenance:

- Regularly monitor the model's performance to ensure it continues to provide accurate predictions.

- Update the model, if necessary, especially if there are changes in the underlying data distribution or if new data becomes available.

Benefits of the System:

- Provides a data-driven approach for predicting profits based on key business variables.

- Allows companies to make informed decisions about resource allocation and investment strategies.

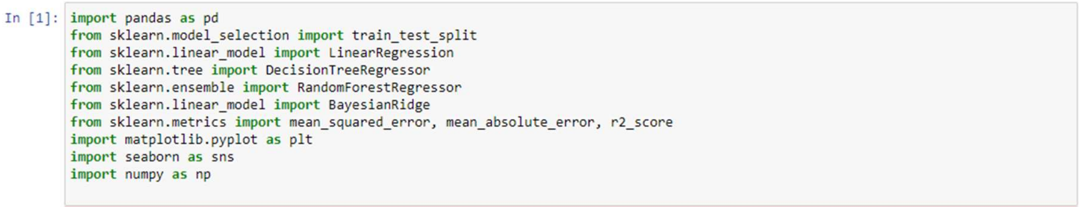
- Enables better financial planning and risk management.

The proposed system integrates machine learning techniques to create a predictive model tailored to the specific problem of profit prediction in the given dataset.

5. Implementation

Predictive models can be constructed using a plethora of excellent packages and libraries.

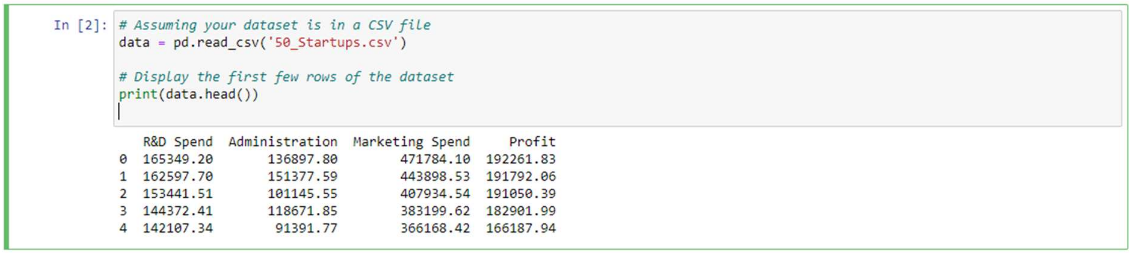
Import Libraries: Import necessary libraries such as pandas for data manipulation, scikit-learn for machine learning models, matplotlib and seaborn for data visualization, and NumPy for numerical operations.



```
In [1]: import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.linear_model import LinearRegression
        from sklearn.tree import DecisionTreeRegressor
        from sklearn.ensemble import RandomForestRegressor
        from sklearn.linear_model import BayesianRidge
        from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
        import matplotlib.pyplot as plt
        import seaborn as sns
        import numpy as np
```

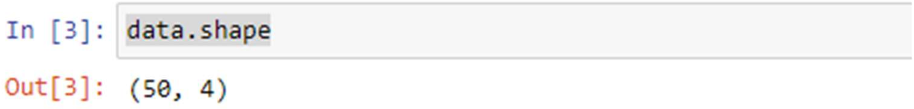
Load Data: Load the dataset named '50_Startups.csv' into a panda Data Frame.

Explore Data: Print the first few rows of the dataset to understand its structure.



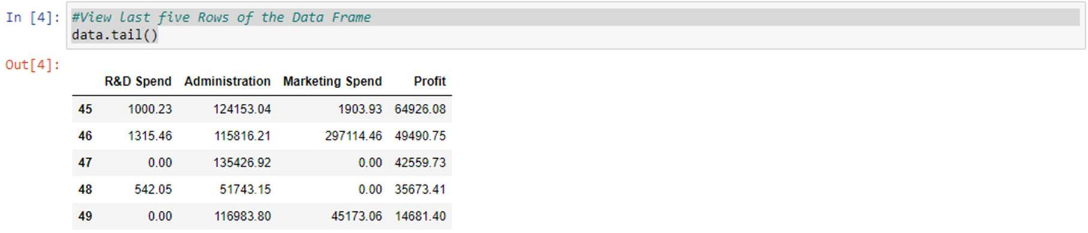
```
In [2]: # Assuming your dataset is in a CSV file
        data = pd.read_csv('50_Startups.csv')

        # Display the first few rows of the dataset
        print(data.head())

           R&D Spend  Administration  Marketing Spend     Profit
        0  165349.20       136897.80        471784.10  192261.83
        1  162597.70       151377.59        443898.53  191792.06
        2  153441.51       101145.55        407934.54  191050.39
        3  144372.41       118671.85        383199.62  182901.99
        4  142107.34        91391.77        366168.42  166187.94
```

Check Data Dimensions: Print the number of rows and columns in the dataset.



```
In [3]: data.shape

Out[3]: (50, 4)
```

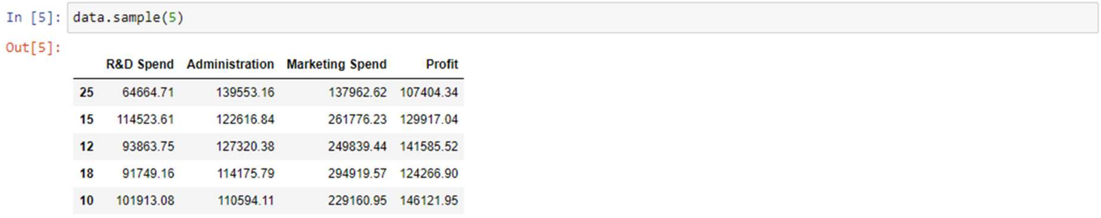
View Last Rows: Display the last five rows of the dataset.



```
In [4]: #View last five Rows of the Data Frame
        data.tail()

Out[4]:
```

	R&D Spend	Administration	Marketing Spend	Profit
45	1000.23	124153.04	1903.93	64926.08
46	1315.46	115816.21	297114.46	49490.75
47	0.00	135426.92	0.00	42559.73
48	542.05	51743.15	0.00	35673.41
49	0.00	116983.80	45173.06	14681.40

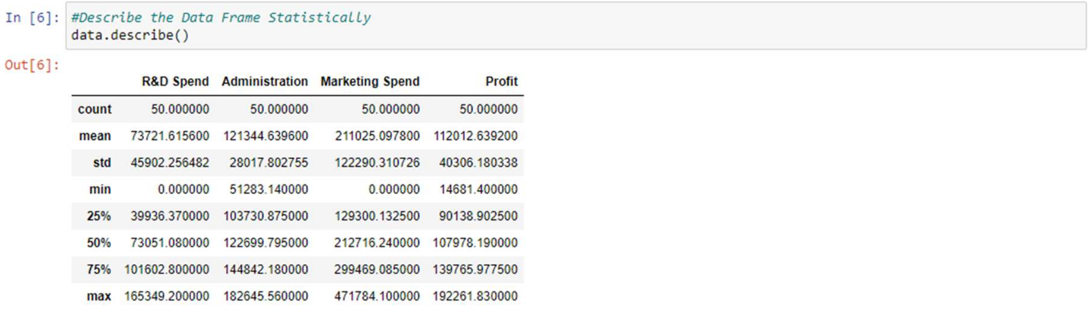
Random Sample: Display a random sample of five rows from the dataset.



```
In [5]: data.sample(5)
Out[5]:
```

	R&D Spend	Administration	Marketing Spend	Profit
25	64664.71	139553.16	137962.62	107404.34
15	114523.61	122616.84	261776.23	129917.04
12	93863.75	127320.38	249839.44	141585.52
18	91749.16	114175.79	294919.57	124266.90
10	101913.08	110594.11	229160.95	146121.95

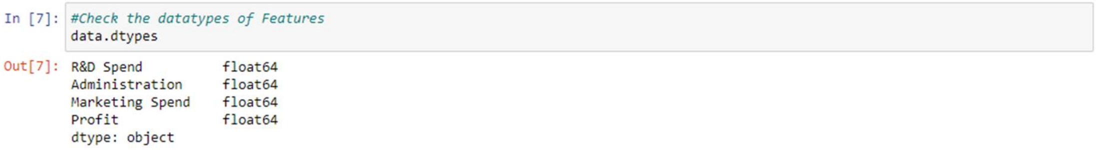
Statistical Summary: Display statistical summary information about the dataset.



```
In [6]: #Describe the Data Frame Statistically
        data.describe()
Out[6]:
```

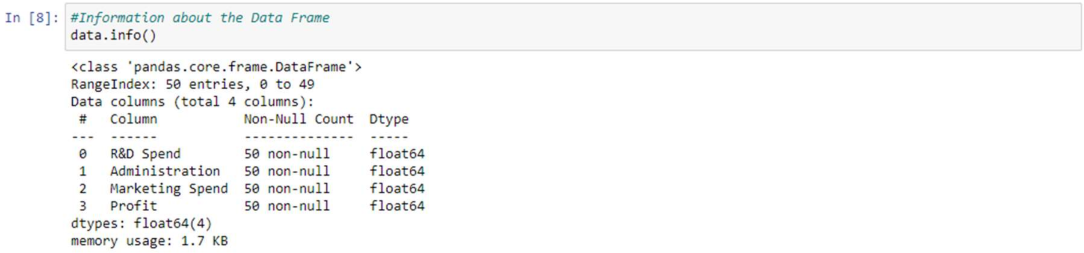
	R&D Spend	Administration	Marketing Spend	Profit
count	50.000000	50.000000	50.000000	50.000000
mean	73721.615600	121344.639600	211025.097800	112012.639200
std	45902.256482	28017.802755	122290.310726	40306.180338
min	0.000000	51283.140000	0.000000	14681.400000
25%	39936.370000	103730.875000	129300.132500	90138.902500
50%	73051.080000	122699.795000	212716.240000	107978.190000
75%	101602.800000	144842.180000	299469.085000	139765.977500
max	165349.200000	182645.560000	471784.100000	192261.830000

Check Data Types: Display the data types of each feature in the dataset.



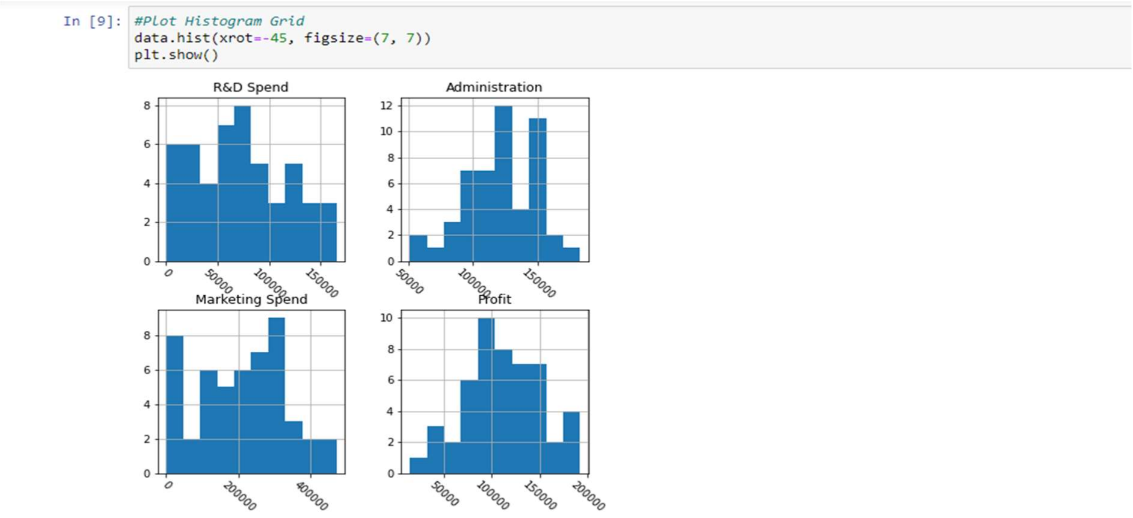
```
In [7]: #Check the datatypes of Features
        data.dtypes
Out[7]: R&D Spend          float64
        Administration     float64
        Marketing Spend    float64
        Profit             float64
        dtype: object
```

Data Information: Display general information about the dataset, including data types and non-null counts.



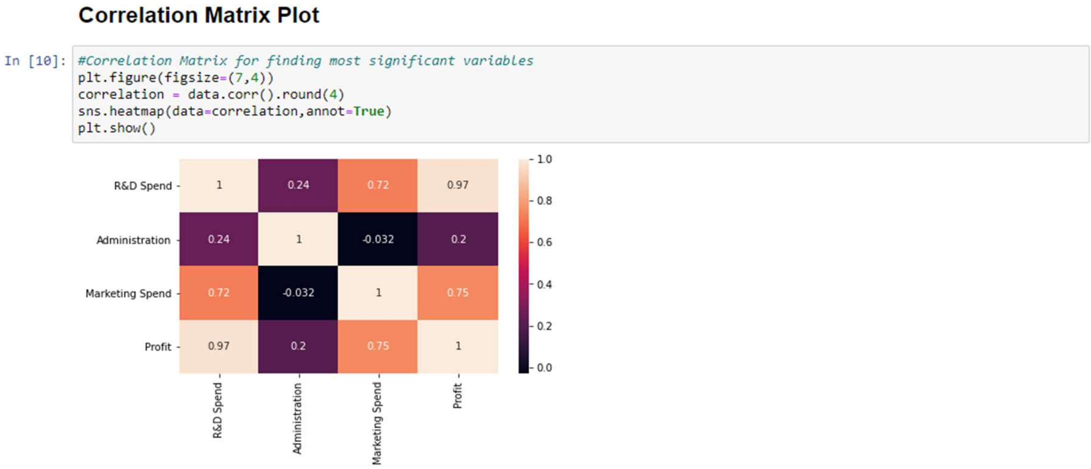
```
In [8]: #Information about the Data Frame
        data.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 50 entries, 0 to 49
        Data columns (total 4 columns):
         #   Column           Non-Null Count  Dtype
        ---  ------           --------------  -----
         0   R&D Spend        50 non-null     float64
         1   Administration   50 non-null     float64
         2   Marketing Spend  50 non-null     float64
         3   Profit           50 non-null     float64
        dtypes: float64(4)
        memory usage: 1.7 KB
```

Histogram Grid: Plot a histogram grid for numerical features in the dataset.



```
In [9]: #Plot Histogram Grid
        data.hist(xrot=-45, figsize=(7, 7))
        plt.show()
```

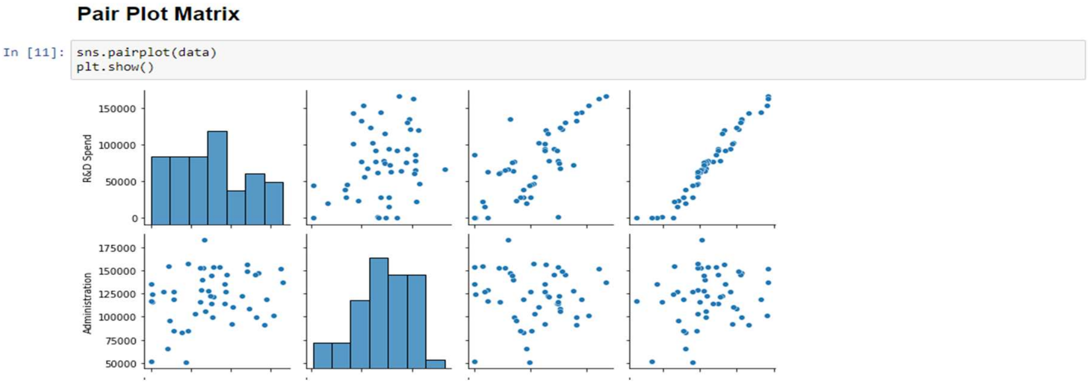
Correlation Matrix: Plot a heatmap of the correlation matrix to identify relationships between variables.



Correlation Matrix Plot

```
In [10]: #Correlation Matrix for finding most significant variables
         plt.figure(figsize=(7,4))
         correlation = data.corr().round(4)
         sns.heatmap(data=correlation,annot=True)
         plt.show()
```

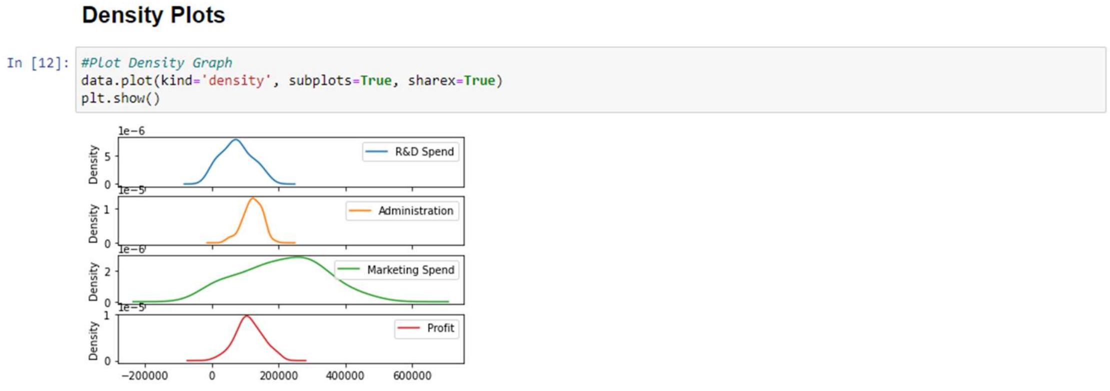
Pair Plot: Plot pairwise relationships in the dataset.



Pair Plot Matrix

```
In [11]: sns.pairplot(data)
         plt.show()
```

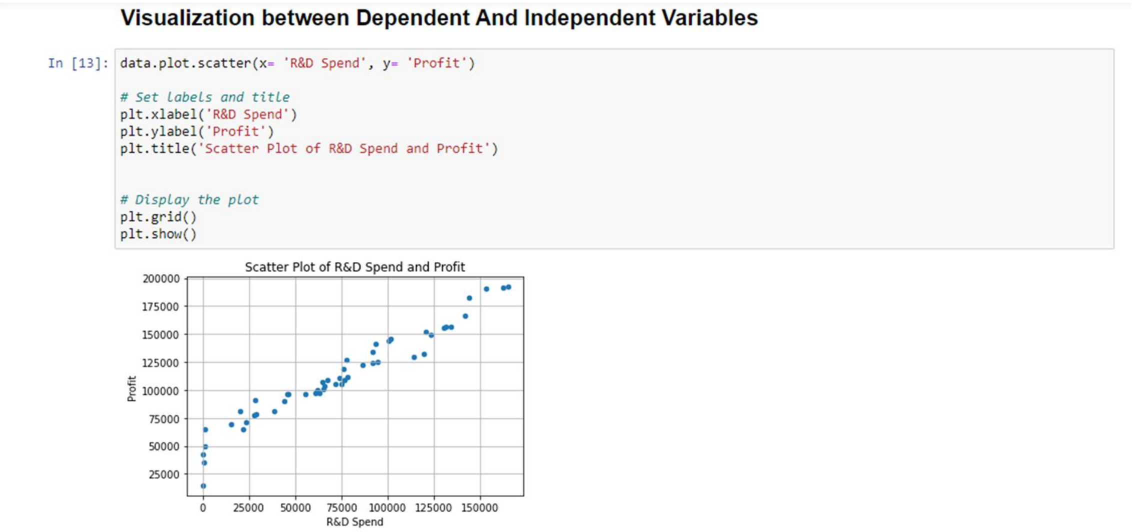
Density Graph: Plot density graphs for each feature.



Density Plots

```
In [12]: #Plot Density Graph
         data.plot(kind='density', subplots=True, sharex=True)
         plt.show()
```

Scatter Plot: Plot a scatter plot of R&D Spend against Profit.



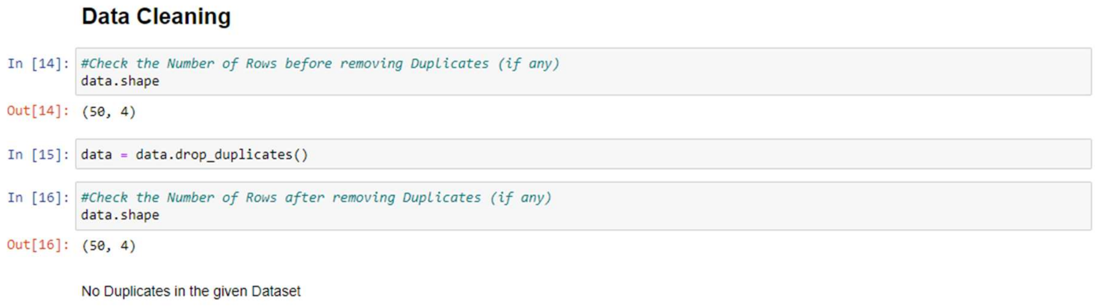
Visualization between Dependent And Independent Variables

```
In [13]: data.plot.scatter(x= 'R&D Spend', y= 'Profit')

         # Set labels and title
         plt.xlabel('R&D Spend')
         plt.ylabel('Profit')
         plt.title('Scatter Plot of R&D Spend and Profit')


         # Display the plot
         plt.grid()
         plt.show()
```

Check Number of Rows Before Duplicates: Display the number of rows before removing duplicates.

Remove Duplicates: Remove duplicate rows from the dataset.

Check Number of Rows After Duplicates Removed: Display the number of rows after removing duplicates.

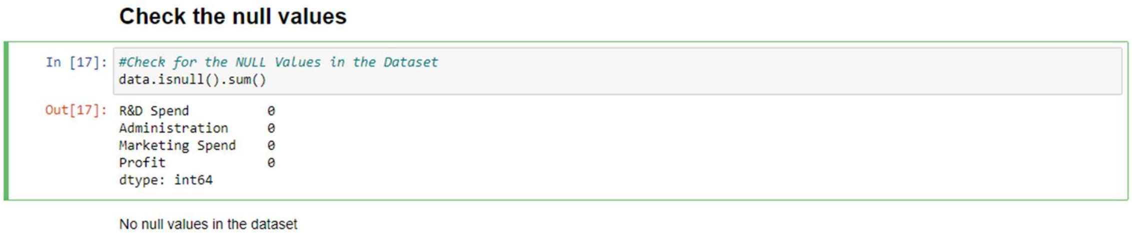


Data Cleaning

```
In [14]: #Check the Number of Rows before removing Duplicates (if any)
         data.shape

Out[14]: (50, 4)

In [15]: data = data.drop_duplicates()

In [16]: #Check the Number of Rows after removing Duplicates (if any)
         data.shape

Out[16]: (50, 4)
```

No Duplicates in the given Dataset

Check for Null Values: Display the sum of null values for each column.



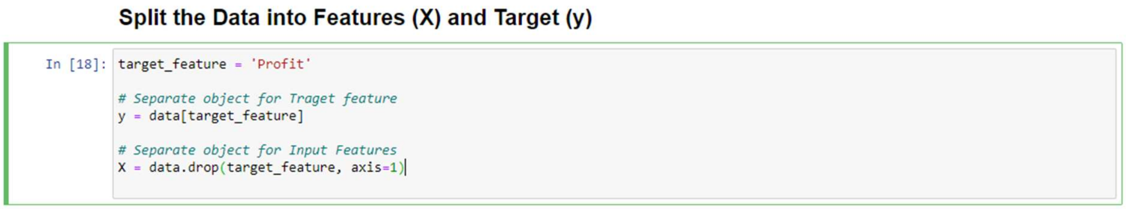
Check the null values

```
In [17]: #Check for the NULL Values in the Dataset
         data.isnull().sum()

Out[17]: R&D Spend          0
         Administration     0
         Marketing Spend    0
         Profit             0
         dtype: int64
```

No null values in the dataset

Set Target Feature: Set the target feature as 'Profit'.

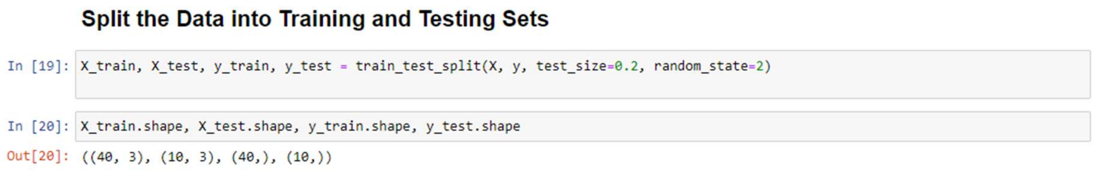
Separate Target and Input Features: Split the dataset into target (y) and input features (X).



Split the Data into Features (X) and Target (y)

```
In [18]: target_feature = 'Profit'

         # Separate object for Traget feature
         y = data[target_feature]

         # Separate object for Input Features
         X = data.drop(target_feature, axis=1)
```

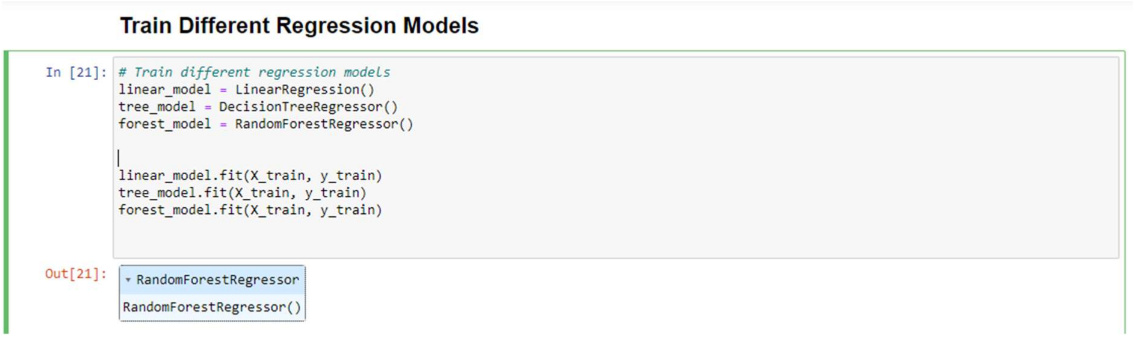
Split Data into Train and Test Sets: Split the dataset into training and testing sets (80% training, 20% testing).



Split the Data into Training and Testing Sets

```
In [19]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2)

In [20]: X_train.shape, X_test.shape, y_train.shape, y_test.shape

Out[20]: ((40, 3), (10, 3), (40,), (10,))
```

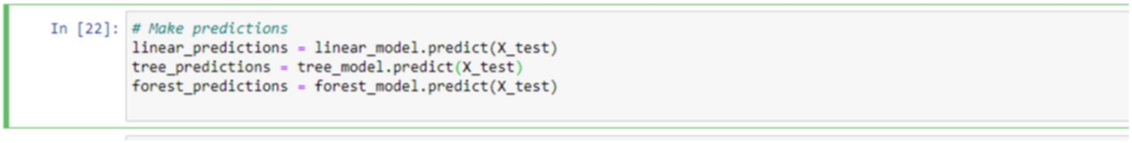
Train Regression Models: Initialize and train Linear Regression, Decision Tree, and Random Forest models.



Train Different Regression Models

```
In [21]: # Train different regression models
         linear_model = LinearRegression()
         tree_model = DecisionTreeRegressor()
         forest_model = RandomForestRegressor()


         linear_model.fit(X_train, y_train)
         tree_model.fit(X_train, y_train)
         forest_model.fit(X_train, y_train)

Out[21]: ▾ RandomForestRegressor
         RandomForestRegressor()
```

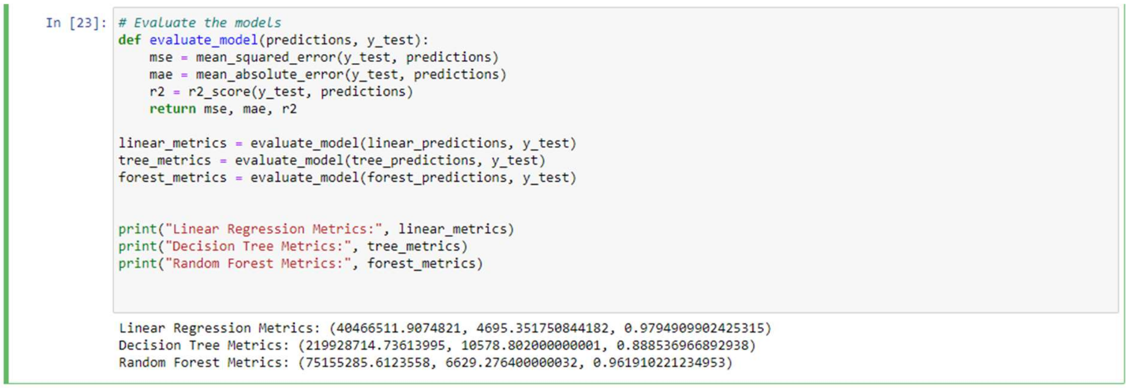
Make Predictions: Use the trained models to make predictions on the test set.



```
In [22]: # Make predictions
         linear_predictions = linear_model.predict(X_test)
         tree_predictions = tree_model.predict(X_test)
         forest_predictions = forest_model.predict(X_test)
```

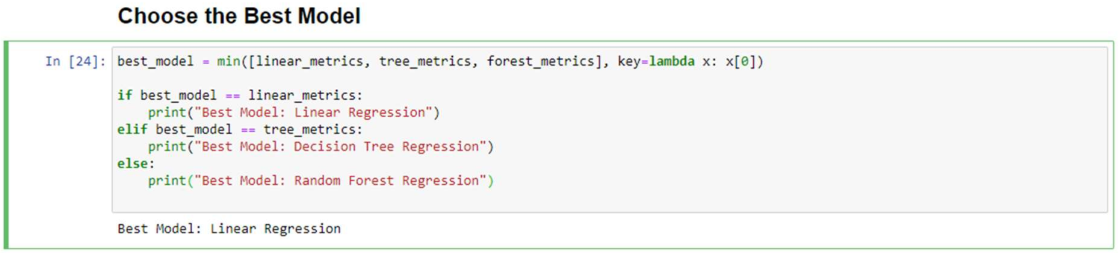
Evaluate Models: Calculate regression metrics (MSE, MAE, R-squared) for each model.

Print Metrics: Display the regression metrics for each model.



```
In [23]: # Evaluate the models
         def evaluate_model(predictions, y_test):
             mse = mean_squared_error(y_test, predictions)
             mae = mean_absolute_error(y_test, predictions)
             r2 = r2_score(y_test, predictions)
             return mse, mae, r2

         linear_metrics = evaluate_model(linear_predictions, y_test)
         tree_metrics = evaluate_model(tree_predictions, y_test)
         forest_metrics = evaluate_model(forest_predictions, y_test)


         print("Linear Regression Metrics:", linear_metrics)
         print("Decision Tree Metrics:", tree_metrics)
         print("Random Forest Metrics:", forest_metrics)


         Linear Regression Metrics: (40466511.9074821, 4695.351750844182, 0.9794909902425315)
         Decision Tree Metrics: (219928714.73613995, 10578.802000000001, 0.888536966892938)
         Random Forest Metrics: (75155285.6123558, 6629.276400000032, 0.961910221234953)
```

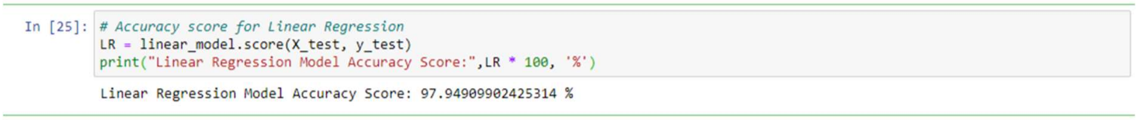
Choose Best Model: Determine the best-performing model based on the metric with the lowest value (e.g., Mean Squared Error).



Choose the Best Model

```
In [24]: best_model = min([linear_metrics, tree_metrics, forest_metrics], key=lambda x: x[0])

         if best_model == linear_metrics:
             print("Best Model: Linear Regression")
         elif best_model == tree_metrics:
             print("Best Model: Decision Tree Regression")
         else:
             print("Best Model: Random Forest Regression")

         Best Model: Linear Regression
```

Accuracy Score for Linear Regression: Calculate and print the accuracy score for the Linear Regression model.



```
In [25]: # Accuracy score for Linear Regression
         LR = linear_model.score(X_test, y_test)
         print("Linear Regression Model Accuracy Score:",LR * 100, '%')

         Linear Regression Model Accuracy Score: 97.94909902425314 %
```

This code provides a comprehensive overview of the steps involved in loading, exploring, preprocessing, visualizing, training, and evaluating regression models for predicting company profits based on various features.

6. Conclusion

In conclusion, the analysis and modelling process for predicting company profits based on R&D Spend, Administration Cost, and Marketing Spend involved several key steps. After loading and exploring the dataset, statistical summaries, data visualizations, and correlation analyses were performed to gain insights into the relationships between variables. Duplicate rows were removed, and the dataset was checked for null values. The dataset was then split into training and testing sets for model evaluation.

Three regression models—Linear Regression, Decision Tree Regression, and Random Forest Regression—were trained and evaluated. The performance of each model was assessed using regression metrics, including Mean Squared Error, Mean Absolute Error, and R-squared. The Linear Regression model demonstrated a strong accuracy score on the test set. The decision was made to choose the best-performing model based on the model with the lowest Mean Squared Error.

This analysis highlights the significance of R&D Spend, Administration Cost, and Marketing Spend in predicting company profits. The chosen model, in this case, was the Linear Regression model, showcasing its effectiveness in capturing the underlying patterns in the data. The insights gained from this process can inform decision-making in business resource allocation and investment strategies, contributing to more informed financial planning and risk management. Ongoing monitoring and potential tuning of the model ensure its continued relevance and reliability in predicting profits for new, unseen data.

From the computations, we get to know that the Linear Regression Model is the best among them. The accuracy of this model is 97.94909902425314 %