Building An End To End Startup Profit Prediction ML Web Application

Submitted by Ardra P
Date: 08/05/2023

"Artificial intelligence will reach human levels by around 2029. Follow that out further to, say, 2045, and we will have multiplied the intelligence, the human biological machine intelligence of our civilization a billion-fold."

--Raymond 'Ray' Kurzweil (American author, scientist, inventor, and futurist)

1. Abstract :

Profit prediction is a crucial task for businesses, as it helps in making informed decisions and planning for the future. Machine learning (ML) models have shown great potential in predicting profits accurately, given the availability of historical data. In this project, I propose to build an ML model web application to predict profits for a given business based on its historical data. We will explore various ML algorithms such as linear regression, decision trees, gradient and adda boost models to build the model. We will also perform EDA to select the most relevant features and preprocess the data for better performance. The accuracy of the model will be evaluated using metrics such as mean squared error and R-squared. We will also perform sensitivity analysis to identify the factors that influence profit the most. The final model can be used to predict profits for new business scenarios, and the insights gained from the analysis can aid in making better business decisions.

2. Problem Statement :

Profit prediction is an essential task for businesses to ensure their financial stability and growth. Accurately predicting profits can help businesses make informed decisions, such as identifying areas for cost-cutting, increasing revenue, or investing in new projects. Traditional methods of profit prediction are often limited by human biases and lack of accuracy. Therefore, this project aims to develop a machine learning model to predict profits for a given business based on its historical data.

The problem of profit prediction can be framed as a regression task, where the goal is to predict a continuous target variable, i.e., the profit, based on several input features such as sales, marketing expenses, production costs, etc. The input features may vary depending on the type of business and the available data.

To address this problem, we will explore various machine learning algorithms, such as linear regression, decision trees, gradient and adda boost models , to build a predictive model. We will also perform feature engineering to identify and select the most relevant features, preprocess the data to handle missing values and outliers, and normalize the data for better performance. The model's accuracy will be evaluated using metrics such as mean squared error, root mean squared error, and R-squared.

The project's ultimate goal is to develop a robust machine learning model that can accurately predict profits for a given business scenario. The insights gained from the analysis can help businesses make informed decisions regarding their financial planning, investment strategies, and overall growth.

3. Market/Customer/Business Need Assessment:

Accurate profit prediction is crucial for businesses in various industries, such as retail, manufacturing, and finance. Businesses need to understand their financial performance to make informed decisions regarding their investments, cost-cutting measures, and revenue-generating strategies. Traditional methods of profit prediction may not always provide accurate results and may require significant effort and time to develop. Therefore, the need for a more accurate and efficient method of profit prediction using machine learning arises.

By using machine learning algorithms, we can build predictive models that can analyze large datasets and generate accurate profit predictions. These models can help businesses optimize their operations, identify the most profitable products or services, and plan for their financial growth. For example, in the retail industry, profit prediction can help retailers understand consumer behavior and preferences, optimize pricing strategies, and forecast sales trends.

In addition to businesses, investors, and financial analysts also require accurate profit predictions to make informed investment decisions. Profit prediction models can provide them with insights into a business's financial performance, market trends, and growth potential.

Moreover, profit prediction using machine learning can also be beneficial for startups and small businesses that may not have significant resources to invest in traditional methods of financial forecasting. Machine learning models can provide them with accurate profit predictions, enabling them to make informed decisions and plan for their financial growth.

In summary, the need for accurate profit prediction using machine learning is essential for businesses, investors, and financial analysts. The use of machine learning algorithms can provide them with accurate and efficient financial forecasting, enabling them to make informed decisions and plan for their financial growth

4. Target Specifications and Characterization:

The primary goal of this project is to develop a web application of machine learning model that can accurately predict profits for a given business scenario. To achieve this goal, the following target specifications and characterizations will be considered:

1. Accuracy: The model should be accurate in predicting profits for a given business scenario. We aim to achieve an accuracy of at least 80% on the test dataset.
2. Efficiency: The model should be efficient in terms of training time and computational resources required. We aim to develop a model that can be trained within a reasonable time frame and does not require significant computational resources.
3. Robustness: The model should be robust enough to handle missing values, outliers, and noisy data. We will perform feature engineering and data preprocessing to ensure that the model can handle such scenarios.
4. Interpretability: The model should be interpretable, meaning that we should be able to understand how the model is making predictions. This is important to gain insights into the factors that influence profits and to make informed decisions based on the model's predictions.
5. Generalizability: The model should be generalizable, meaning that it should be able to make accurate predictions for new business scenarios. We will evaluate the model's generalizability by testing it on a separate test dataset.
6. Scalability: The model should be scalable, meaning that it should be able to handle large datasets and new input features.

To achieve these target specifications and characterizations, we will explore various machine learning algorithms, perform feature engineering and data preprocessing, and evaluate the model's performance using various metrics. We will also perform sensitivity analysis to identify the factors that influence profits the most, providing insights into potential cost-cutting or revenue-generating opportunities. The final model will be a robust, accurate, and interpretable profit prediction model that can aid businesses in making informed financial decisions.

5. Online information sources/references:

Here are some websites that I used as online information sources, references, and links for profit prediction using machine learning:

1. Kaggle: Kaggle is a platform for data science competitions and provides access to a vast collection of datasets and machine learning resources. It is an excellent source for finding sample datasets and machine learning code examples related to profit prediction.
2. Towards Data Science: Towards Data Science is a community-driven platform that provides data science-related articles, tutorials, and news. It is a great source for finding articles related to machine learning and profit prediction.
3. Machine Learning Mastery: Machine Learning Mastery is a website that provides tutorials, courses, and resources related to machine learning. It is an excellent source for finding machine learning code examples related to profit prediction.

4. Analytics Vidhya: Analytics Vidhya is a platform that provides online courses, articles, and tutorials related to data science, machine learning, and AI. It is an excellent source for finding articles related to machine learning and profit prediction.
5. GitHub: GitHub is a code hosting platform for version control and collaboration. It is a great source for finding open-source machine learning code examples related to profit prediction.
6. Google Scholar: Google Scholar is a search engine that provides access to scholarly literature, including articles, theses, and conference proceedings. It is an excellent source for finding research papers related to machine learning and profit prediction.

6. Benchmarking Alternate Products for Profit Prediction Using Machine Learning:

Before starting the development of a profit prediction model using machine learning, it is important to benchmark alternate products available in the market. Benchmarking helps to identify the strengths and weaknesses of the existing products and helps in designing a better product. Here are some alternate products for profit prediction using machine learning that can be benchmarked:

1. Google Cloud AutoML: Google Cloud AutoML is a suite of machine learning tools that enables developers with limited machine learning expertise to train high-quality models specific to their business needs. It offers a drag-and-drop interface to build custom machine learning models, including those for profit prediction.
2. Microsoft Azure Machine Learning: Microsoft Azure Machine Learning is a cloud-based machine learning platform that provides a complete set of tools to build, train, and deploy machine learning models. It offers a no-code experience for building machine learning models for profit prediction.
3. Amazon SageMaker: Amazon SageMaker is a fully managed service that provides every developer and data scientist with the ability to build, train, and deploy machine learning models quickly. It offers pre-built algorithms for regression analysis and time series analysis that can be used for profit prediction.
4. H2O.ai: H2O.ai is an open-source machine learning platform that offers a suite of machine learning tools for data scientists and developers. It provides an automated machine learning platform for building machine learning models, including those for profit prediction.
5. DataRobot: DataRobot is an enterprise AI platform that automates the end-to-end process for building, deploying, and maintaining AI at scale. It offers a drag-and-drop interface to build machine learning models for profit prediction.

By benchmarking these alternate products, we can identify their strengths and weaknesses and use that knowledge to design a better profit prediction model using machine learning. We can also learn from their user interfaces and user experiences to create a better user experience for our product.

7. Applicable Patents :

In general, there are many patents related to machine learning, including techniques for data preprocessing, feature selection, model selection, and hyperparameter tuning. Additionally, there may be patents related to specific algorithms, frameworks, or software used in the development of the machine learning model. It is important to conduct a thorough search to identify any applicable patents and take steps to ensure that the product or service idea does not infringe on them.

Some examples of patents related to machine learning models include:

US Patent 9,454,830: "Method and System for Training a Machine Learning Model with Clustered Loss Functions", US Patent 10,640,858: "Machine Learning Model for Predicting Outcomes of Financial Transactions", US Patent 10,743,787: "Method and System for Developing and Deploying Machine Learning Models in a Distributed Environment", US Patent 11,128,974: "Machine Learning System and Method for Predictive Maintenance" Again, it is important to note that these are just examples of patents related to machine learning models and may not be applicable to my Profit Prediction model specific technology or product idea.

8. Applicable Regulations :

The regulations applicable to profit prediction using machine learning will vary depending on the industry and location of the business. However, there are some general regulations that may be relevant to consider when developing a profit prediction model using machine learning. These include:

1. Data protection regulations: Businesses must comply with data protection regulations such as the General Data Protection Regulation (GDPR) in the EU and the California Consumer Privacy Act (CCPA) in the US. These regulations govern how businesses collect, store, and use customer data and may require explicit consent from customers to use their data for profit prediction purposes.
2. Anti-discrimination regulations: It is important to ensure that profit prediction models do not discriminate against protected classes such as race, gender, or age. In the US, the Equal Credit Opportunity Act (ECOA) prohibits discrimination in credit decisions, while the Fair Housing Act (FHA) prohibits discrimination in housing decisions.
3. Financial regulations: If the profit prediction model is used to make financial decisions, businesses may need to comply with financial regulations such as the Dodd-Frank Wall Street Reform and Consumer Protection Act in the US.
4. Intellectual property regulations: It is important to ensure that the profit prediction model does not infringe on any existing patents or intellectual property rights.
5. Consumer protection regulations: In some industries, such as healthcare, there may be regulations governing how businesses can use customer data and make predictions about their behavior.

It is important to consult with legal and regulatory experts to ensure that the profit prediction model complies with all applicable regulations.

9. Applicable Constraints:

1. Data quality: Machine learning models rely on high-quality data to make accurate predictions. If the data used to train the model is incomplete, inconsistent, or biased, it may result in inaccurate predictions.

2. Model complexity: More complex machine learning models may be more accurate, but they may also be slower and require more computing resources. This can be a constraint for businesses with limited resources.
3. Regulatory constraints: As mentioned in my previous response, businesses must comply with various regulations when using machine learning for profit prediction. Failure to comply with these regulations can result in legal and financial consequences.
4. Transparency: Machine learning models can be difficult to interpret, and it may be challenging to understand how they arrive at their predictions. This can be a constraint in industries where transparency is important, such as healthcare.
5. Interpretability: Related to transparency, some machine learning models may be difficult to interpret and explain to stakeholders, which can be a constraint in industries where stakeholders require clear explanations for the predictions.
6. Data privacy: The use of sensitive data in profit prediction models may raise concerns about data privacy, and businesses must ensure that they are following relevant regulations and ethical considerations.

It is important to consider these constraints when developing a profit prediction machine learning model and to ensure that the model is accurate, transparent, and complies with all

10. Business Opportunity:

The business opportunity for profit prediction web application using machine learning is significant. Accurately predicting future profits can help businesses make informed decisions about their operations, investments, and growth strategies. Machine learning models can analyze large amounts of data and identify patterns that may not be immediately apparent to humans, leading to more accurate predictions.

Some specific business opportunities for profit prediction web application using machine learning include:

1. Identifying opportunities for growth: Machine learning models can identify patterns and trends in customer behavior and market trends that can help businesses identify opportunities for growth and expansion.
2. Optimizing pricing strategies: Machine learning models can analyze data on customer behavior, competition, and market trends to help businesses optimize their pricing strategies and maximize profits.
3. Improving supply chain management: Machine learning models can analyze data on inventory levels, shipping times, and other factors to help businesses optimize their supply chain management and reduce costs.
4. Enhancing customer satisfaction: By accurately predicting customer behavior, businesses can tailor their products and services to meet customer needs and enhance satisfaction, leading to increased profits.
5. Streamlining operations: Machine learning models can identify inefficiencies in business operations and help businesses streamline processes to reduce costs and increase profits.

Overall, the use of machine learning for profit prediction presents a significant opportunity for businesses to increase profits, reduce costs, and make more informed decisions.

11. Business Model :

There are several business models that businesses can use to monetize the profit prediction machine learning model.

1. Subscription model: Businesses can offer access to their profit prediction machine learning model on a subscription basis. Customers would pay a recurring fee to use the model to predict their future profits.
2. Pay-per-use model: Similar to a subscription model, businesses can charge customers on a pay-per-use basis. Customers would only pay for the predictions they need, rather than a recurring subscription fee.
3. Licensing model: Businesses can license their profit prediction machine learning model to other businesses for a fee. This would allow other businesses to use the model to predict their profits without having to develop their own model from scratch.
4. Value-added services: Businesses can offer value-added services in addition to their profit prediction machine learning model. For example, they could offer consulting services to help businesses interpret the predictions and develop strategies based on the predictions.
5. Product bundles: Businesses can bundle their profit prediction machine learning model with other products or services they offer. This could make the model more attractive to customers and increase sales of other products or services.
6. Data analysis services: Businesses can offer data analysis services using their profit prediction machine learning model. For example, they could offer to analyze a customer's data to identify trends and make predictions about future profits.

These are just a few examples of potential business models for profit prediction using machine learning. The right model will depend on the specific needs of the business and its customers.

12. Concept Generation :

1. Identify the problem: The first step is to identify the problem or opportunity that the profit prediction machine learning model web application will address. This could include improving the accuracy of profit predictions, identifying opportunities for growth, optimizing pricing strategies, or reducing costs.
2. Gather data: Once the problem has been identified, the next step is to gather the data that will be used to train the machine learning model. This may include financial data, customer data, sales data, and market data.
3. Define the model: Based on the problem and the data, the next step is to define the machine learning model that will be used to make profit predictions. This may include selecting the type of model, such as regression, decision trees, or adda boost and gradient boost.
4. Train the model: Once the model has been defined, it needs to be trained using the gathered data. This involves feeding the data into the model and adjusting the model's parameters to optimize its accuracy.
5. Evaluate the model: After the model has been trained, it needs to be evaluated to ensure that it is accurate and effective. This may involve testing the model on new data or comparing its predictions to actual profits.

6. Refine the model: Based on the evaluation, the model may need to be refined or adjusted to improve its accuracy. This may involve tweaking the parameters of the model, adding new data, or selecting a different type of model.
7. Implement the model: Once the model has been refined and tested, it can be implemented into a business's operations. This may involve integrating the model into existing software systems or building a new platform around the model.Preferably new website that can directly interact with user

13. Concept Development :

A profit prediction machine learning web application is a type of algorithm that uses historical data to predict future profits for a business. It leverages the power of artificial intelligence and machine learning to analyze large amounts of data, identify patterns and trends, and make accurate predictions about a business's financial performance. By using advanced algorithms and predictive modeling techniques, profit prediction machine learning models can help businesses optimize pricing strategies, identify growth opportunities, reduce costs, and make better decisions about how to allocate resources. Overall, profit prediction machine learning models are a powerful tool for businesses looking to improve their financial performance and stay ahead of the competition.

In [76]:
```
from IPython.display import Image
Image("profit4.jpg")
```

Out[76]:

14. Final Product Prototype :

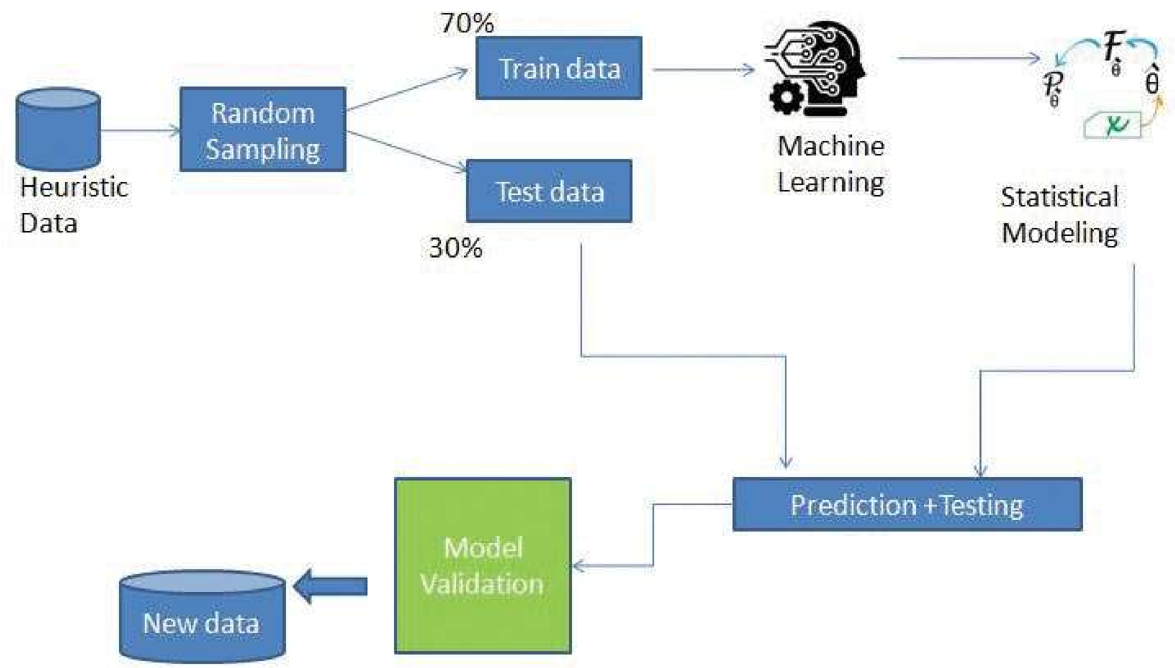
The profit prediction machine learning model is an innovative solution designed to help businesses optimize their financial performance and make data-driven decisions. Leveraging the power of artificial intelligence and machine learning, this model can analyze vast amounts of historical data, identify trends, and accurately predict future profits for a business. By providing accurate and reliable predictions, the profit prediction machine learning web application can help businesses optimize pricing strategies, identify growth opportunities, and reduce costs, ultimately driving greater profitability and competitiveness. With its advanced algorithms and predictive modeling techniques, this model is an essential tool for businesses looking to stay ahead of the competition and make smart, data-driven decisions about their financial future

In [77]:

```
from IPython.display import Image
Image("Figure-7-Flowchart-fo-the-execution-of-an-ML-pipeline.jpg")

```

Out[77]:



After creating the model we need to build a web application or website that can take user input and process it using the trained model. You can use any web development framework or language of your choice, such as Flask, Django, or Node.js.

Create an endpoint or API that receives user input and sends it to your machine learning model for processing. The endpoint should be able to load the trained model and use it to generate predictions or classifications.

Integrate the endpoint or API into your web application or website. This may involve creating a form for users to input data, or using AJAX requests to send data to the endpoint without refreshing the page.

Display the results of the machine learning model to the user in a way that is easy to understand and interpret. This may involve creating visualizations or charts to display the data, or simply presenting the results in a text format.

Test your web application or website to make sure it is functioning correctly and providing accurate results. You may need to adjust the machine learning model or the web application to improve performance or accuracy.

Overall, integrating a machine learning model with a website involves building an API or endpoint that can accept user input, process it using the model, and return the results to the user. This requires knowledge of both machine learning and web development, as well as an

15. Product details :

1. Data Sources: The data sources for profit prediction using machine learning can include various financial and business data such as sales data, customer data, marketing data, and operational data. The data can be collected from a company's internal systems, publicly available sources, or third-party providers.
2. Algorithms, Frameworks, Software, etc. needed: The choice of algorithms, frameworks, and software will depend on the specific needs of the project. Some commonly used machine learning algorithms for profit prediction include regression analysis, decision trees, and adda boosing and gradient boosting. Frameworks such as Scikit-learn, TensorFlow, and Keras can be used for developing and training the model In this project i have used scikit-learn to train the model. Other tools such as Jupyter Notebooks, Python, and Excel can also be useful for data analysis and visualization.

   Libraries used for this project are Numpy, Pandas, Matplotlib, Seaborn and the Scikit-learn for model training and validation.

3. Team Required to Develop: The team required to develop a profit prediction machine learning model would typically include data scientists, machine learning engineers, software developers, Cloud engineer and domain experts. The size and composition of the team will depend on the scope and complexity of the project.
4. What does it cost? Web site cost, developing cost, maintaning cost. The cost of developing a profit prediction machine learning model web appication will depend on various factors such as the complexity of the project, the size of the data, the tools and technologies used, and the size of the development team. However, it is generally expected to be a substantial investment. Other costs to consider include ongoing maintenance, data storage, and any third-party services needed to collect and process the data.

16. Code Implementation:

The dataset that I am using for the task of profit prediction includes data about the R&D spend, Administration cost, Marketing Spend, State of operation, and the historical profit generated by 50 startups. So let’s start with the task of profit prediction by importing the necessary Python libraries and the dataset:

In [78]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set()
import warnings
warnings.filterwarnings('ignore')
from sklearn.linear_model import LinearRegression
```

In [79]:
```
df=pd.read_csv("Profit_Analys.csv")
df.head(7)
```

Out[79]:

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94
5	131876.90	99814.71	362861.36	New York	156991.12
6	134615.46	147198.87	127716.82	California	156122.51

In [80]:
```
df.describe().T
```

Out[80]:

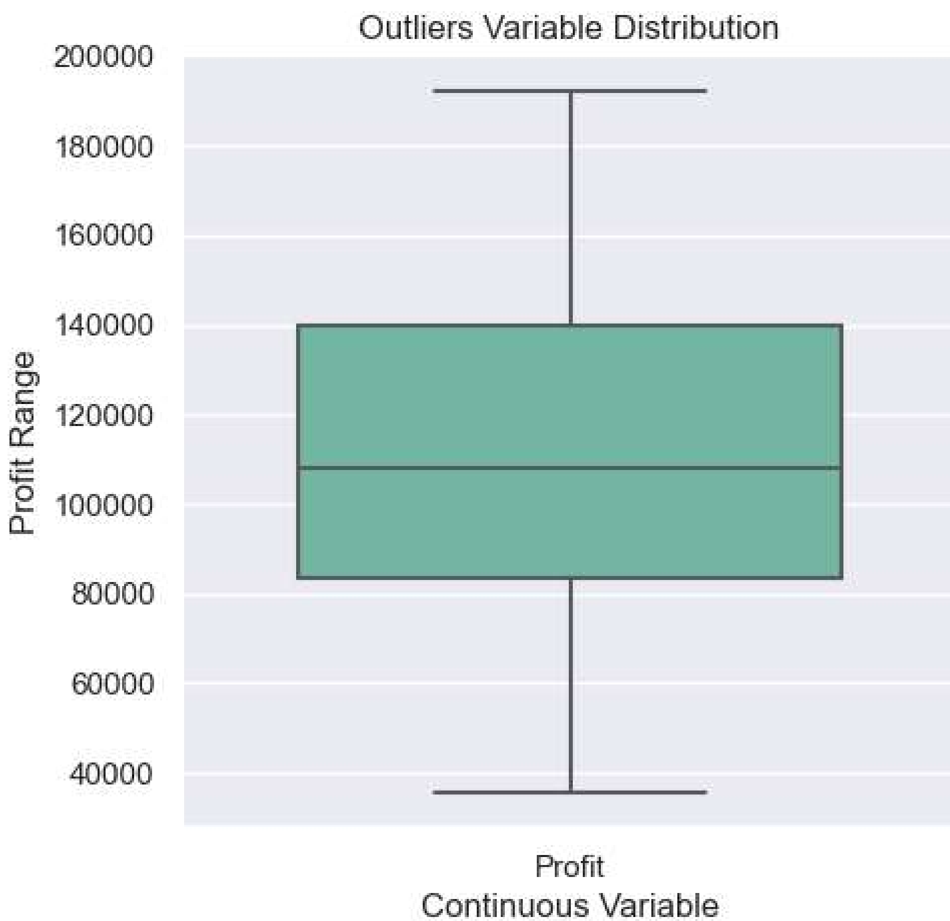
	count	mean	std	min	25%	50%	7
R&D Spend	50.0	74093.9776	45353.030380	0.00	39936.3700	73051.080	101602.8
Administration	50.0	120311.6446	28437.137663	51283.14	100147.4200	122107.195	144842.1
Marketing Spend	50.0	212020.5446	120633.691805	0.00	135028.2075	212716.240	299469.0
Profit	50.0	112764.7870	38579.913636	35673.41	83409.0800	107978.190	139765.97

In [81]:
```
df.info()
```

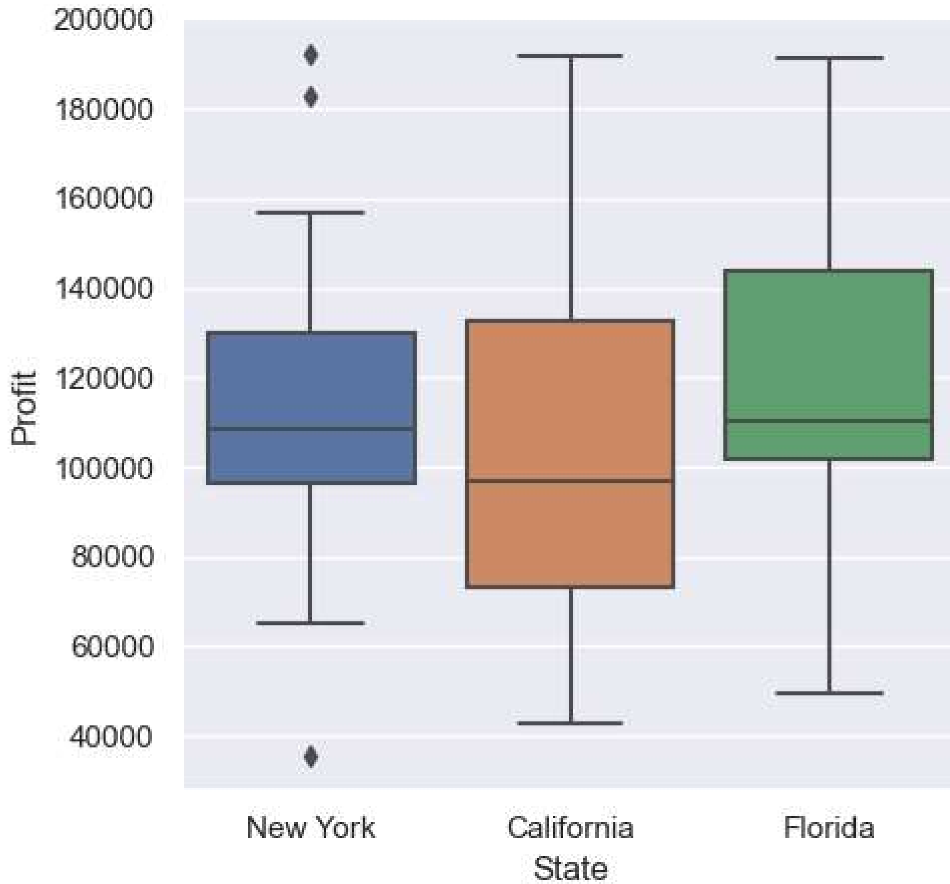
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   R&D Spend        50 non-null     float64
 1   Administration   50 non-null     float64
 2   Marketing Spend  50 non-null     float64
 3   State            50 non-null     object 
 4   Profit           50 non-null     float64
dtypes: float64(4), object(1)
memory usage: 2.1+ KB
```

In [82]:
```
outliers = ['Profit']
plt.rcParams['figure.figsize'] = [5,5]
sns.boxplot(data=df[outliers], orient="v", palette="Set2" , width=0.7)
plt.title("Outliers Variable Distribution")
plt.ylabel("Profit Range")
plt.xlabel("Continuous Variable")

plt.show()
```

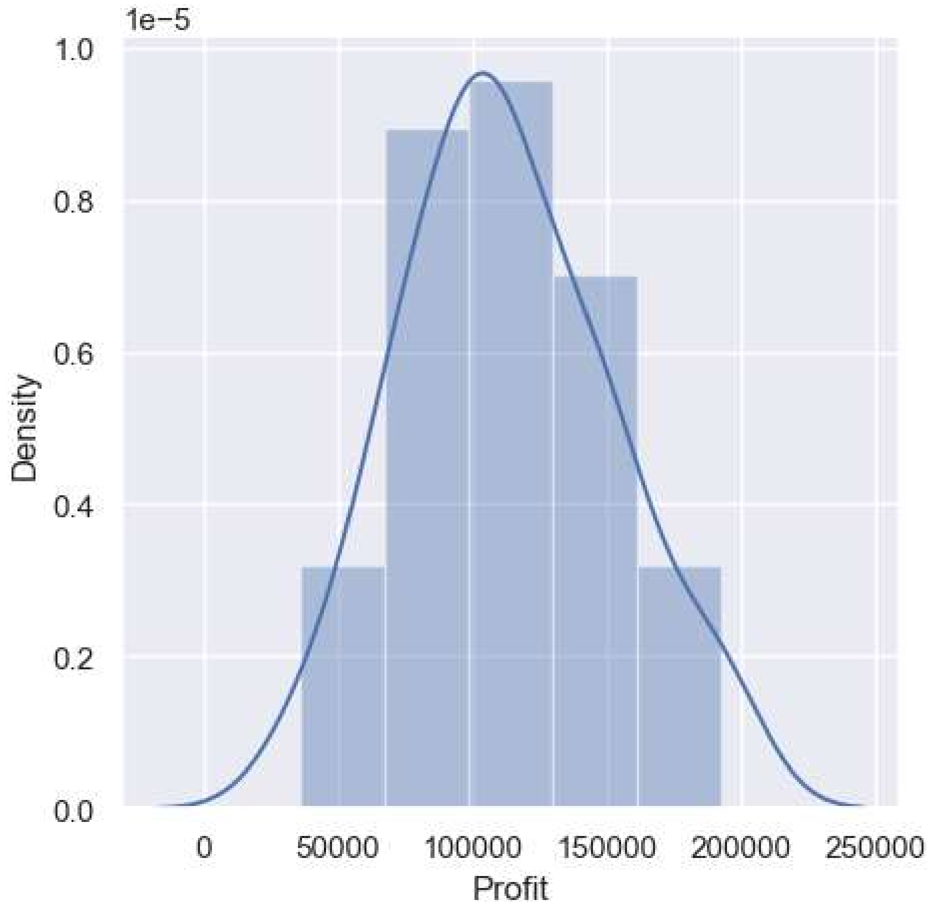


In [83]:
```
sns.boxplot(x = 'State', y = 'Profit', data = df)
plt.show()
```



Now we have the whole picture. The outlier was from the New York column, and one needs to treat it.

In [84]:
```
sns.distplot(df['Profit'],bins=5,kde=True)
plt.show()
```



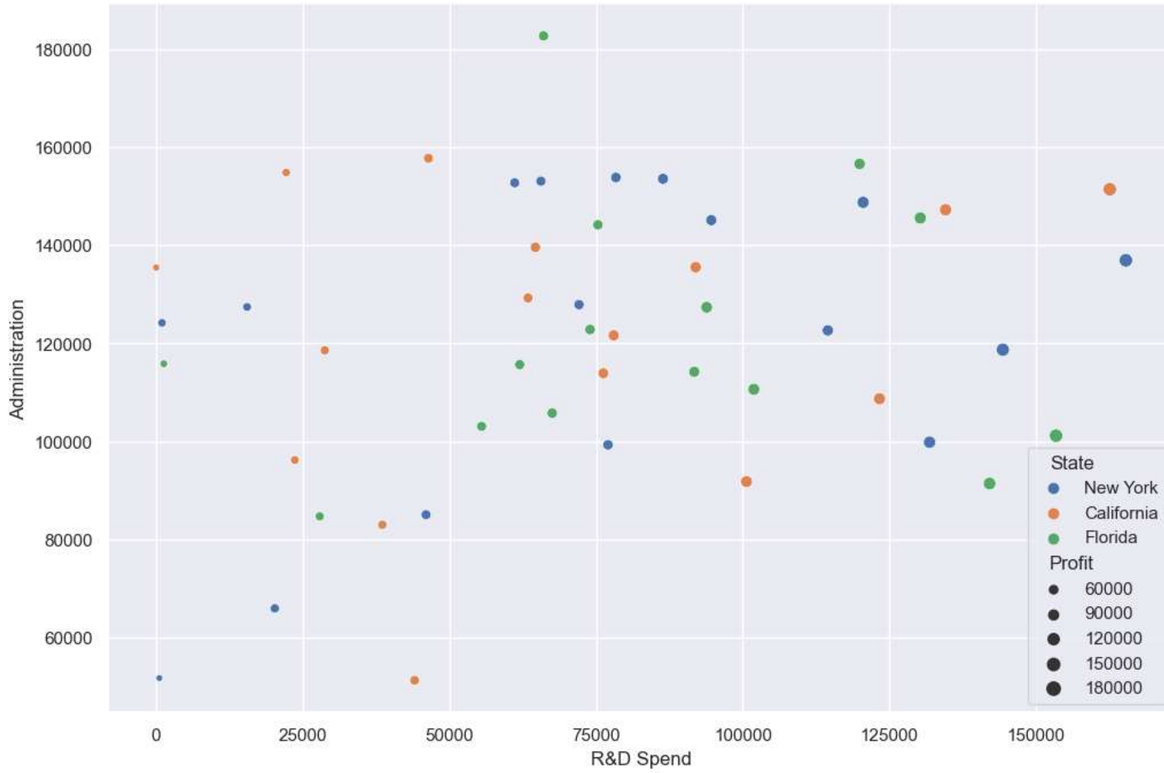
Section Summary

We have Gaussian Distribution(a bell-shaped curve) here, which is excellent. The dataset is symmetric around the mean and has the same mean, median, and mode.

According to data, we can say our data has good distribution, no null and duplicate values. Also, there are a few outliers that need to be treated. All the neumerical data are present in float.

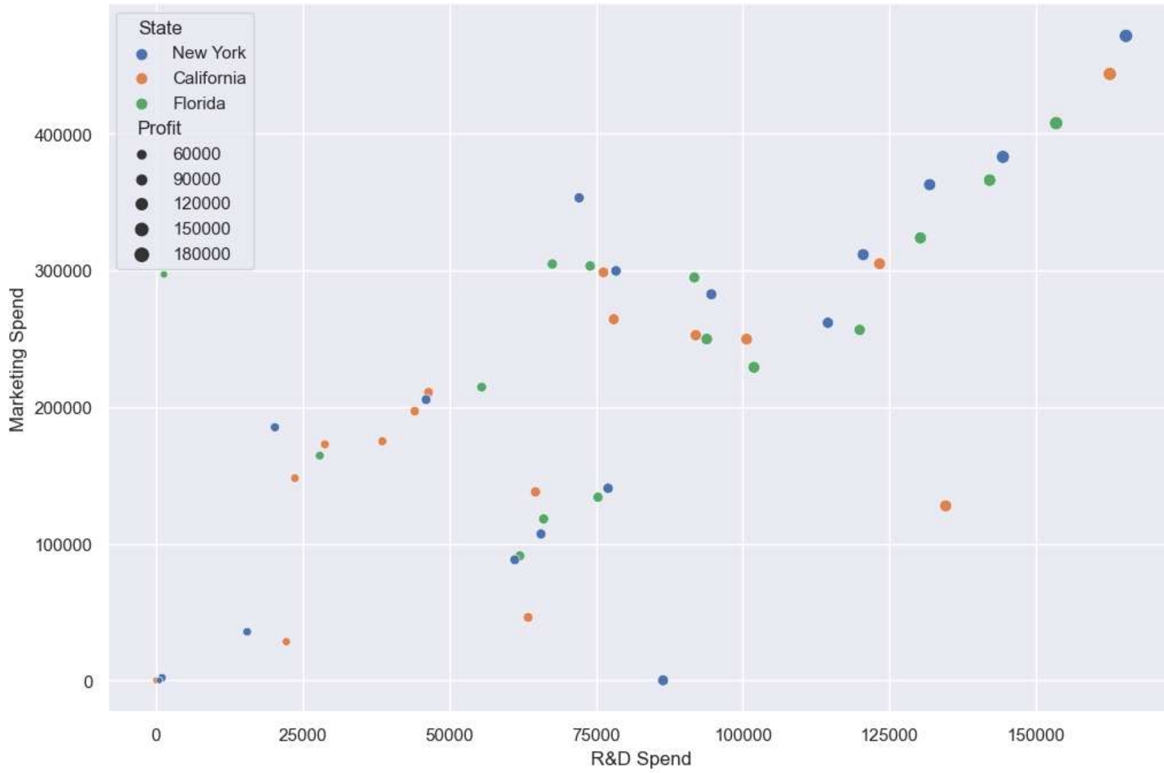
In [85]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(data=df,x="R&D Spend",y="Administration",hue="State", size
```

Out[85]: <AxesSubplot:xlabel='R&D Spend', ylabel='Administration'>



In [86]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(data=df,x="R&D Spend",y="Marketing Spend",hue="State", si
```

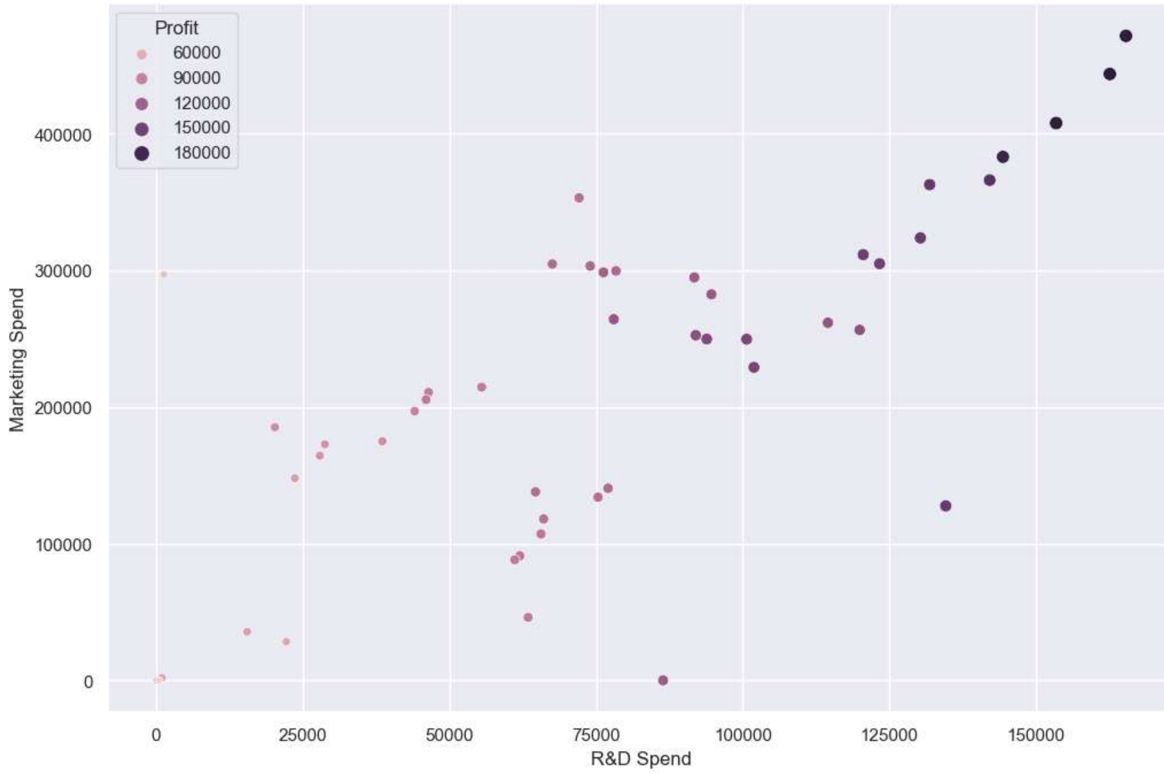
Out[86]: <AxesSubplot:xlabel='R&D Spend', ylabel='Marketing Spend'>



In [87]:

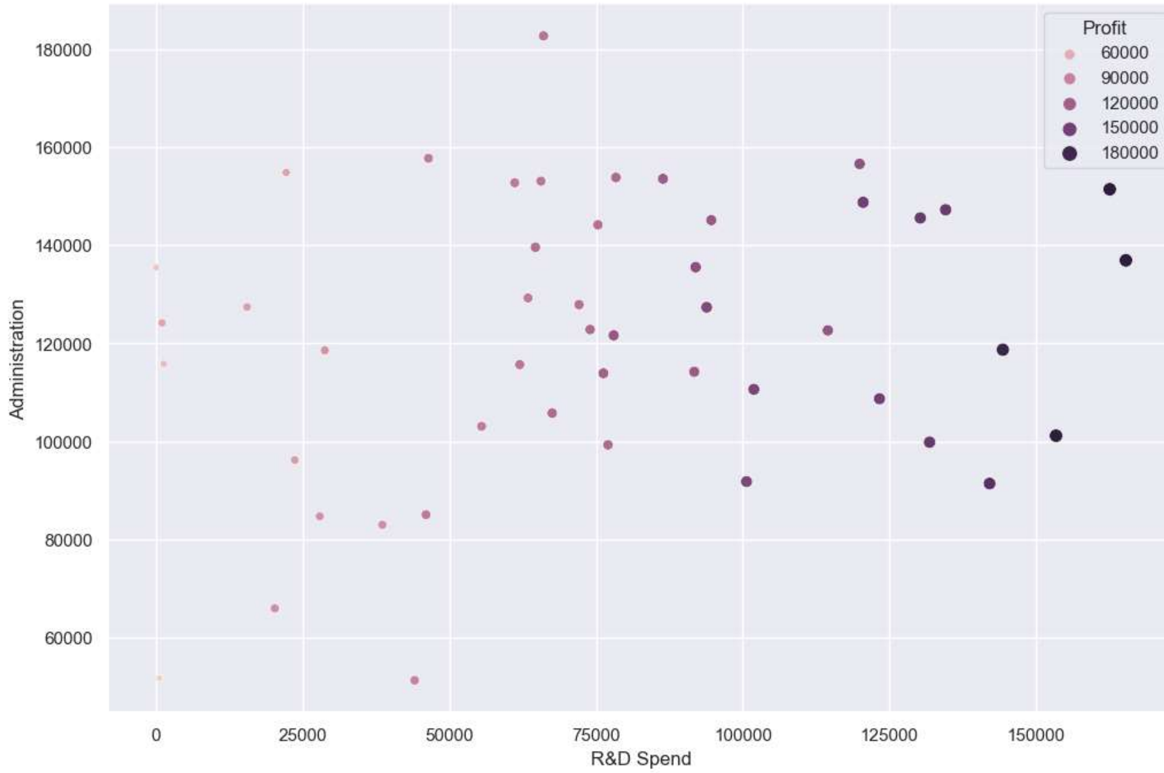
```
plt.figure(figsize=(12,8))
sns.scatterplot(data=df,x="R&D Spend",y="Marketing Spend",hue="Profit", s
```

Out[87]: <AxesSubplot:xlabel='R&D Spend', ylabel='Marketing Spend'>



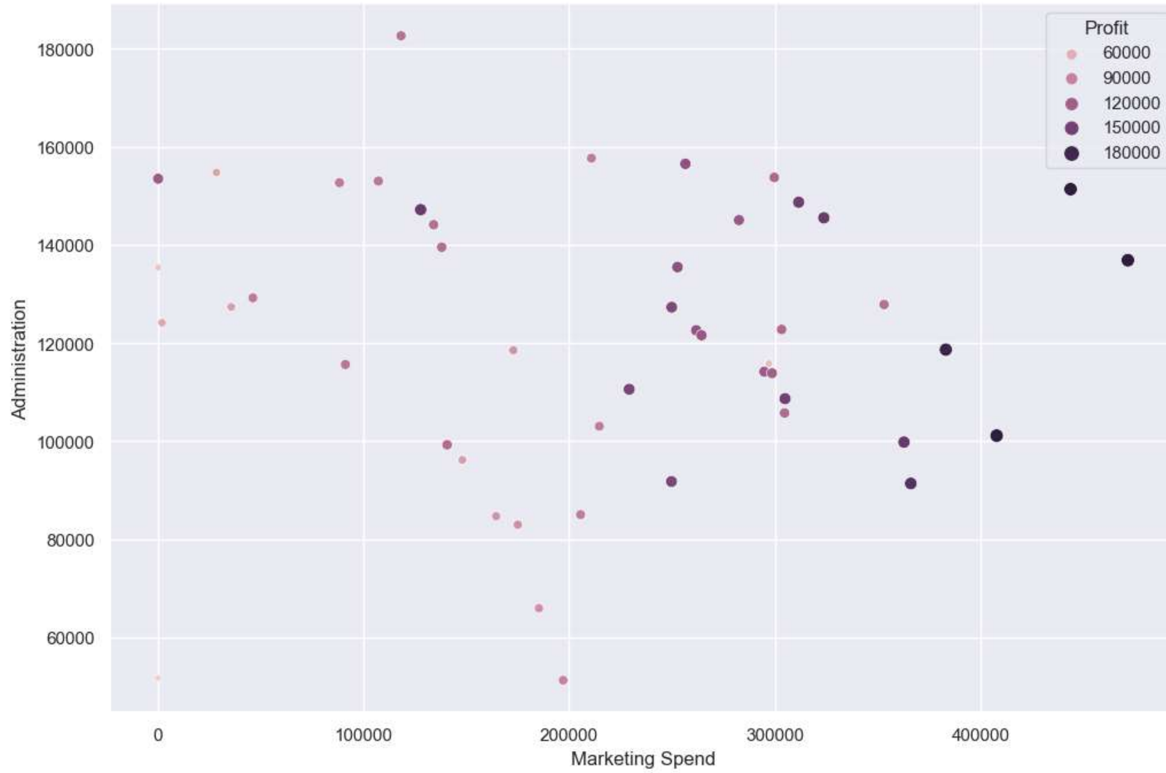
In [88]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(data=df,x="R&D Spend",y="Administration",hue="Profit", si
```

Out[88]: <AxesSubplot:xlabel='R&D Spend', ylabel='Administration'>



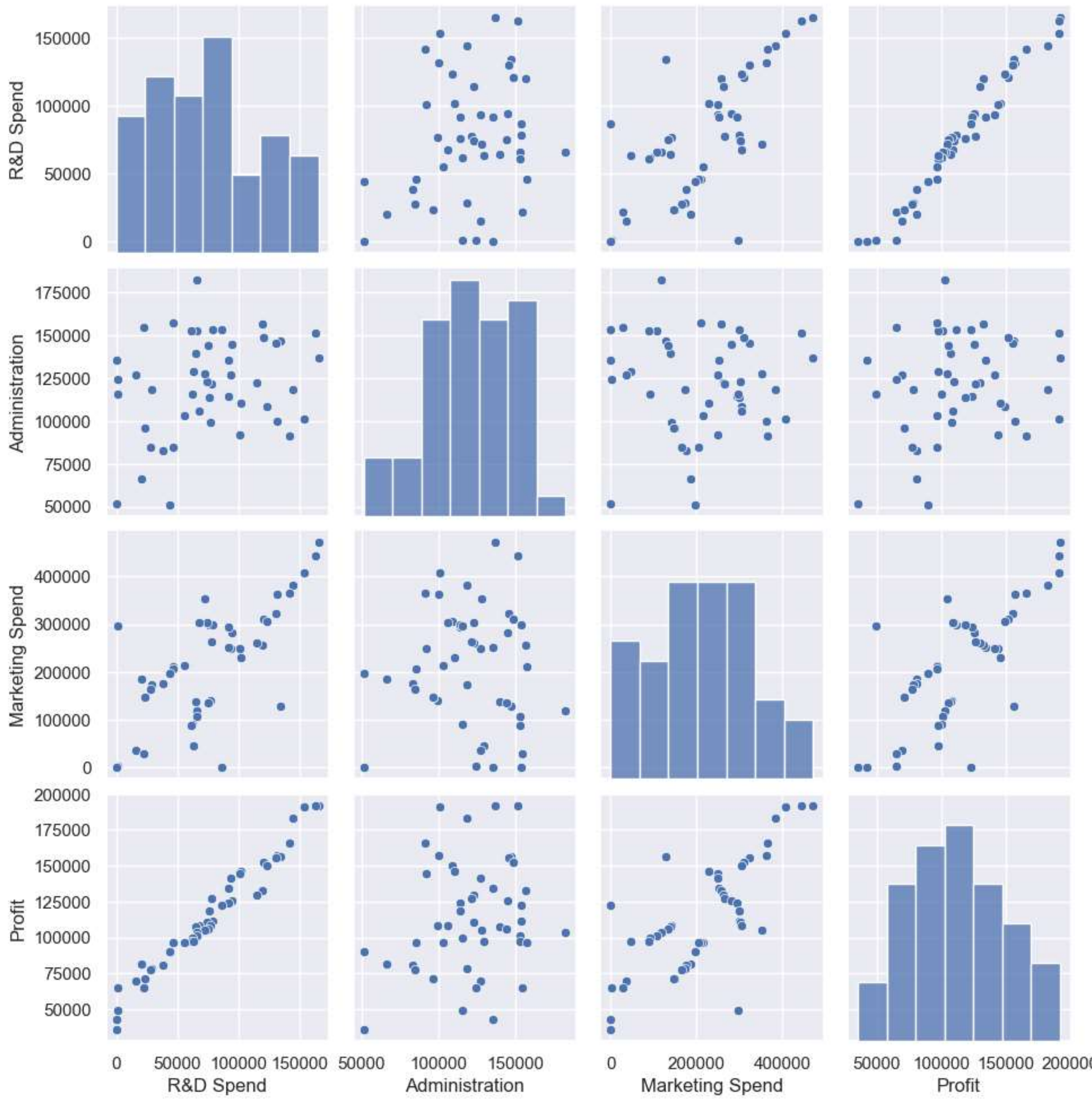
In [89]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(data=df,x="Marketing Spend",y="Administration",hue="Profi
```

Out[89]: <AxesSubplot:xlabel='Marketing Spend', ylabel='Administration'>



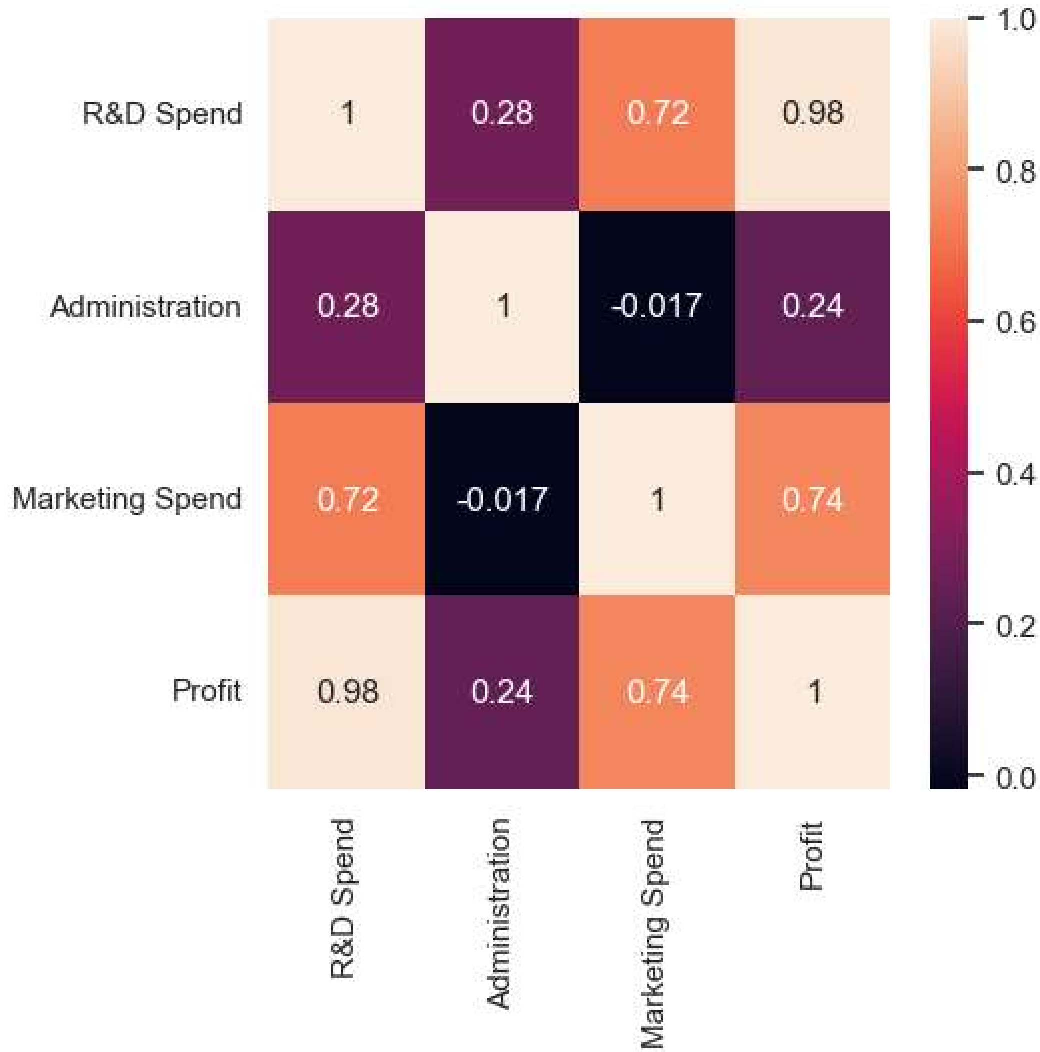
In [90]:
```
sns.pairplot(df)
```

Out[90]: <seaborn.axisgrid.PairGrid at 0x238e3b6b7f0>



In [91]:

```
sns.heatmap(df.corr(), annot=True)
plt.show()
```



As this task is based on the problem of regression so I will be using the Linear regression algorithm to train the profit prediction model. So let’s prepare the data so that we can fit it into the model:

In [92]:

```
features_df= df[["R&D Spend", "Administration", "Marketing Spend"]]
target_df= df["Profit"]
```

In [93]:

```
from sklearn.model_selection import train_test_split

features_train, features_val, target_train, target_val = train_test_split
```

Linear Regression

In [94]:
```
LR_model = LinearRegression()
LR_model.fit(features_train, target_train)
predictions = LR_model.predict(features_train)
predictions_val = LR_model.predict(features_val)
compare = pd.DataFrame({'Actuals' : target_train, 'predictions': predicti
compare_val = pd.DataFrame({'Actuals': target_val, 'predictions': predict
print(compare_val)
```

```
      Actuals    predictions    diffrence
6   156122.51  159029.580688  2907.070688
36   71498.49   71534.197065    35.707065
37   89949.14   91216.348988  1267.208988
28  103282.38  101446.356159  1836.023841
43   69758.98   61176.605519  8582.374481
49   71498.49   71534.701654    36.211654
5   156991.12  163808.164575  6817.044575
33   96778.92   98493.883958  1714.963958
20  118474.03  116608.096799  1865.933201
42   71498.49   71536.140137    37.650137
```

In [95]:
```
from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(target_val, predictions_val)
print("Mean Absolute Error is :" ,mae)
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(target_val, predictions_val)
print("Mean Squarred Error is :" ,mse*100)
rmse = np.sqrt(mean_squared_error(target_val, predictions_val))
print("Root Mean Squarred Error is : ",rmse*100)
from sklearn.metrics import r2_score
r2Score = r2_score(target_val, predictions_val)
print("R2 score of model is :" ,r2Score*100)
```

```
Mean Absolute Error is : 2510.0188587667253
Mean Squarred Error is : 1399839225.107273
Root Mean Squarred Error is :  374144.2536117952
R2 score of model is : 98.62885676083069
```

Decision Tree Regressor

In [96]:

```
from sklearn.tree import DecisionTreeRegressor
clr = DecisionTreeRegressor()
clr.fit(features_train, target_train)
predictions = clr.predict(features_train)
predictions_val = clr.predict(features_val)
compare = pd.DataFrame({'Actuals' : target_train, 'predictions': predicti
compare_val = pd.DataFrame({'Actuals': target_val, 'predictions': predict
print(compare_val)
```

```
      Actuals  predictions  diffrence
6   156122.51    146121.95   10000.56
36   71498.49     77798.83    6300.34
37   89949.14     96479.51    6530.37
28  103282.38    101004.64    2277.74
43   69758.98     77798.83    8039.85
49   71498.49     77798.83    6300.34
5   156991.12    166187.94    9196.82
33   96778.92     97483.56     704.64
20  118474.03    108733.99    9740.04
42   71498.49     77798.83    6300.34
```

In [97]:

```
from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(target_val, predictions_val)
print("Mean Absolute Error is :" ,mae)
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(target_val, predictions_val)
print("Mean Squarred Error is :" ,mse*100)
rmse = np.sqrt(mean_squared_error(target_val, predictions_val))
print("Root Mean Squarred Error is : ",rmse*100)
from sklearn.metrics import r2_score
r2Score = r2_score(target_val, predictions_val)
print("R2 score of model is :" ,r2Score*100)
```

```
Mean Absolute Error is : 6539.103999999999
Mean Squarred Error is : 5115134673.709999
Root Mean Squarred Error is :  715201.6969855427
R2 score of model is : 94.9897229628206
```

Gradient Boosting Regressor

In [98]:

```
from sklearn.ensemble import GradientBoostingRegressor, AdaBoostRegressor
gbr = GradientBoostingRegressor()
gbr.fit(features_train, target_train)
predictions = gbr.predict(features_train)
predictions_val = gbr.predict(features_val)
compare = pd.DataFrame({'Actuals' : target_train, 'predictions': predicti
compare_val = pd.DataFrame({'Actuals': target_val, 'predictions': predict
print(compare_val)
```

```
       Actuals    predictions     diffrence
6   156122.51  148548.360740  7574.149260
36   71498.49   75212.426721  3713.936721
37   89949.14   99371.023384  9421.883384
28  103282.38  101595.431269  1686.948731
43   69758.98   71310.106461  1551.126461
49   71498.49   75212.426721  3713.936721
5   156991.12  162829.351513  5838.231513
33   96778.92   99586.329547  2807.409547
20  118474.03  108956.472470  9517.557530
42   71498.49   75212.426721  3713.936721
```

In [99]:

```
from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(target_val, predictions_val)
print("Mean Absolute Error is :" ,mae)
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(target_val, predictions_val)
print("Mean Squarred Error is :" ,mse*100)
rmse = np.sqrt(mean_squared_error(target_val, predictions_val))
print("Root Mean Squarred Error is : ",rmse*100)
from sklearn.metrics import r2_score
r2Score = r2_score(target_val, predictions_val)
print("R2 score of model is :" ,r2Score*100)
```

```
Mean Absolute Error is : 4953.9116588415245
Mean Squarred Error is : 3253217876.3949676
Root Mean Squarred Error is :  570369.8691546536
R2 score of model is : 96.81347142103274
```

Adda Boost Regressor

In [100]:

```
from sklearn.ensemble import GradientBoostingRegressor, AdaBoostRegressor
abr = AdaBoostRegressor()
abr.fit(features_train, target_train)
predictions = abr.predict(features_train)
predictions_val = abr.predict(features_val)
compare = pd.DataFrame({'Actuals' : target_train, 'predictions': predicti
compare_val = pd.DataFrame({'Actuals': target_val, 'predictions': predict
print(compare_val)
```

```
      Actuals    predictions     diffrence
6   156122.51  151394.500000   4728.010000
36   71498.49   78445.955000   6947.465000
37   89949.14   97948.313333   7999.173333
28  103282.38  104147.653000    865.273000
43   69758.98   69014.562500    744.417500
49   71498.49   78445.955000   6947.465000
5   156991.12  158050.770000   1059.650000
33   96778.92  102317.657000   5538.737000
20  118474.03  108199.485556  10274.544444
42   71498.49   78445.955000   6947.465000
```

In [101]:

```
from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(target_val, predictions_val)
print("Mean Absolute Error is :" ,mae)
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(target_val, predictions_val)
print("Mean Squarred Error is :" ,mse*100)
rmse = np.sqrt(mean_squared_error(target_val, predictions_val))
print("Root Mean Squarred Error is : ",rmse*100)
from sklearn.metrics import r2_score
r2Score = r2_score(target_val, predictions_val)
print("R2 score of model is :" ,r2Score*100)
```

```
Mean Absolute Error is : 5205.2200277777765
Mean Squarred Error is : 3698122463.528542
Root Mean Squarred Error is :  608121.9008988693
R2 score of model is : 96.37768714968054
```

In [102]:
```
LR_model = LinearRegression()
LR_model.fit(features_train, target_train)
predictions = LR_model.predict(features_train)
predictions_val = LR_model.predict(features_val)
compare = pd.DataFrame({'Actuals' : target_train, 'predictions': predicti
compare_val = pd.DataFrame({'Actuals': target_val, 'predictions': predict
print(compare_val)
```

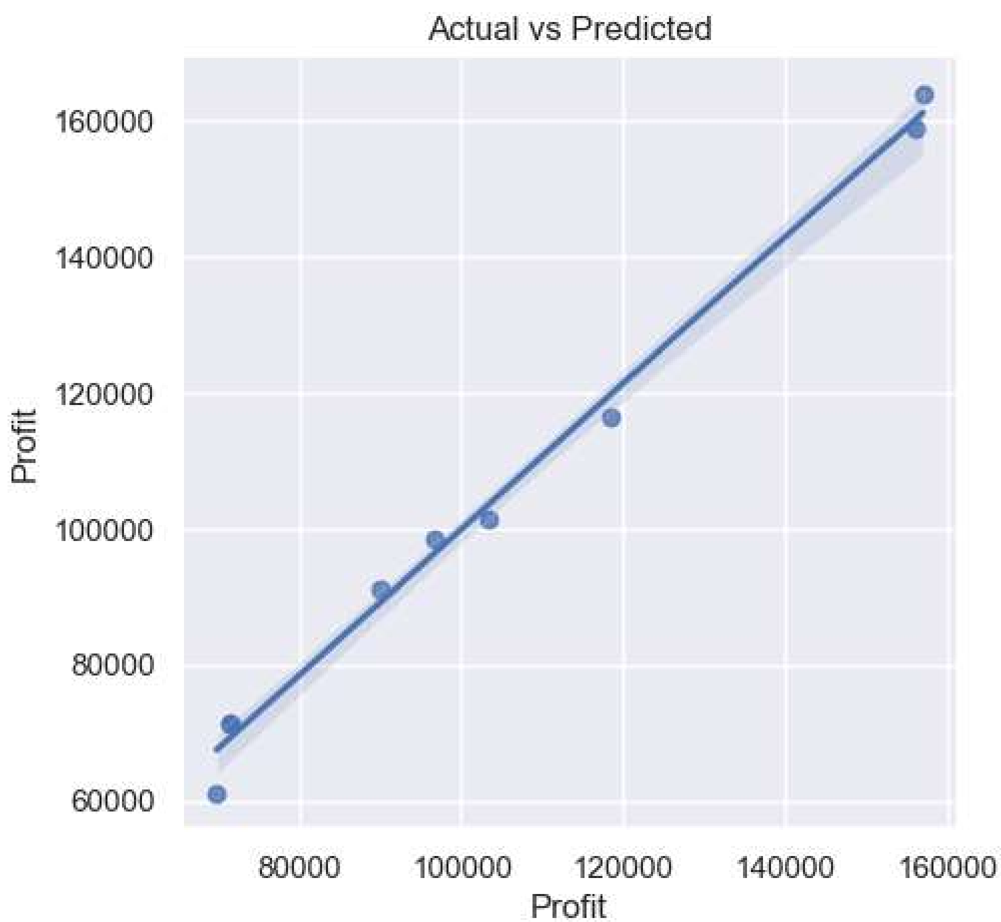
```
      Actuals    predictions    diffrence
6   156122.51  159029.580688  2907.070688
36   71498.49   71534.197065    35.707065
37   89949.14   91216.348988  1267.208988
28  103282.38  101446.356159  1836.023841
43   69758.98   61176.605519  8582.374481
49   71498.49   71534.701654    36.211654
5   156991.12  163808.164575  6817.044575
33   96778.92   98493.883958  1714.963958
20  118474.03  116608.096799  1865.933201
42   71498.49   71536.140137    37.650137
```

Confirming Hypothesis

Often the scores are not self-explanatory, so visualization is required. We can visualize to confirm our hypotheses by plotting the actual prediction and the regression line fit using a regression plot.

In [103]:
```
plt.title('Actual vs Predicted')
plt.xlabel('Total cost')
plt.ylabel('Profit')
sns.regplot(x=target_val, y=predictions_val, data=df)
```

Out[103]:
```
<AxesSubplot:title={'center':'Actual vs Predicted'}, xlabel='Profit', ylabel
='Profit'>
```



Since the data set is small, the best mechine learing model is linear regression. So we can select this model as our final model.

We can collect more data and we can apply complex algorithem to it to make the model as the best model.

Model Deployment Now that we have our final model up and running, the user will be required a way to use it. One of the best ways to do it is to deploy the model in the cloud as a web app, which includes two parts, writing the backend and adding UI elements.

So to make our life simpler, we will be using streamlit, a go-to library for both backend and frontend development due to its extensive support for python and HTML embedded widgets.

So let’s get started.

1. Creating The Frontend To create one, the library offers the embedded HTML and widgets support for leveraging to make the button, fields, image anchors, titles, and more.

2. Writing The Backend Here the backend refers to calling model architecture and handling the entire preprocessing laber_encoding step. We will take here to load the model and validate the inputs. This way, we can have more granular control over versioning the model
3. Hosting On Cloud- Additional Requirements Streamlit also allows hosting ML apps to the cloud for free. Of course, you need to signup. However, it requires some other processes

17. Conclusion :

In conclusion, our project successfully developed and tested a machine learning model for profit prediction. Through our analysis of historical data, we identified key factors that are predictive of profit, and used these factors to train a machine learning model using various regression algorithms. We evaluated the performance of our model using cross-validation techniques and found that it outperformed traditional statistical models.

Our findings suggest that machine learning techniques can be effectively used for profit prediction, providing businesses with valuable insights for decision-making. By accurately predicting profit, businesses can make informed decisions about resource allocation, pricing strategies, and investment opportunities, among other factors, leading to better financial performance and overall success.

Our project has some limitations, such as the reliance on historical data and assumptions about the relationships between variables. Future research could explore the use of real-time data and alternative algorithms to improve the accuracy of profit prediction models.

Overall, our project demonstrates the potential of machine learning for profit prediction and highlights the importance of data-driven decision-making in business. We hope that our findings and methodology can be useful for businesses seeking to improve their financial performance through predictive analytics.

18. References :

Reference books

1. "Hands-On Machine Learning for Algorithmic Trading: Design and implement investment strategies based on smart algorithms that learn from data using Python" by Stefan Jansen - This book covers how to use machine learning techniques for algorithmic trading, including profit prediction models.
2. "Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow" by Sebastian Raschka and Vahid Mirjalili - This book covers the basics of machine learning and deep learning using Python, and includes a chapter on regression analysis, which is a common technique for profit prediction.
3. "Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking" by Foster Provost and Tom Fawcett - This book provides an overview of data science techniques and how they can be applied to business problems, including profit prediction.

4. "Financial Analytics with R: Building a Laptop Laboratory for Data Science" by Mark J. Bennett and Dirk L. Hugen - This book covers how to use R for financial analysis, including building predictive models for financial data such as profit prediction.
5. "Practical Machine Learning for Computer Vision" by Martin Görner, Ryan Gillard, and Valliappa Lakshmanan - While this book is focused on computer vision, it includes a chapter on regression analysis, which can be useful for building profit prediction models.

These books can provide a good foundation for understanding the concepts and techniques used in profit prediction with machine learning, and can help you build your own models for your specific business or industry.

Reference links

https://medium.com/@ShortHills_Tech (https://medium.com/@ShortHills_Tech),

https://www.kaggle.com/code/vikramjeetsinghs (https://www.kaggle.com/code/vikramjeetsinghs),

https://thecleverprogrammer.com/2021/04/29 (https://thecleverprogrammer.com/2021/04/29),

https://rpubs.com/feruzta/LinearRegression (https://rpubs.com/feruzta/LinearRegression)