Title: Profit Prediction for Startups using Regression Models

ABSTRACT

This project aims to predict the profitability of startups by employing regression models that utilize key variables such as R&D spend, administration spend, and marketing spend. The dataset used for training and evaluation consists of information from 50 startup companies. By leveraging this dataset, the developed regression models provide accurate profit forecasts, enabling startups to make informed decisions regarding resource allocation and investment strategies. The project's significance lies in its potential to empower startups with predictive capabilities, allowing them to optimize their financial planning and increase their chances of success in a competitive business landscape.

The regression models in this project utilize advanced algorithms such as linear regression, lasso regression, ridge regression , decision tree regression , random forest regression, elasticnet regression , support vector regression and many more. Through extensive preprocessing and feature selection techniques, the dataset is prepared for model training and evaluation. The performance of the models is assessed using metrics like mean squared error, root mean squared error, and R-squared. The results demonstrate the effectiveness of the regression models in accurately predicting startup profitability based on R&D spend, administration spend, and marketing spend. This project provides valuable insights for startups, enabling them to make data-driven decisions that optimize their financial strategies, allocate resources efficiently, and maximize profitability, ultimately enhancing their overall business performance and prospects for success.

TABLE OF CONTENTS

1. INTRODUCTION

In today's highly competitive business environment, startups face numerous challenges in ensuring their profitability and long-term success. One crucial aspect of achieving profitability is effectively managing resources and making informed decisions about investments. To aid startups in this endeavor, this project focuses on developing a predictive framework using regression models to forecast the profitability of startups.

The project centers around three key variables: R&D spend, administration spend, and marketing spend. These variables are known to significantly impact a startup's profitability by influencing its ability to innovate, streamline operations, and reach target customers. By leveraging a dataset comprising information from 50 startup companies, the project aims to train regression models capable of accurately predicting profit outcomes based on these variables.

The predictive models developed in this project can serve as valuable tools for startups, enabling them to make data-driven decisions in resource allocation and investment planning. By understanding the relationship between R&D spend, administration spend, marketing spend, and profitability, startups can optimize their financial strategies, identify areas for cost-saving or investment, and maximize their chances of achieving sustainable growth.

Overall, this project aims to provide startups with a powerful framework for profit prediction, enhancing their ability to navigate the competitive business landscape and make informed decisions that drive their success. By leveraging regression models and analyzing the impact of key variables, startups can gain valuable insights to guide their financial planning, resource allocation, and overall business strategies.

2. EXISTING METHODS AND ISSUES

Existing Methods in Profit Prediction for Startups:

1. ***Static financial ratios***: Traditional methods rely on static financial ratios, which may not capture the dynamic nature of startups and their unique business models.

2. ***Historical data and trends:*** Existing methods heavily rely on historical data and trends, which may not accurately predict future profitability for startups experiencing rapid growth and change.

3. ***Limited consideration of non-financial factors***: Profit prediction models often overlook non-financial factors such as customer satisfaction, market competition, and technological advancements, which can significantly impact a startup's profitability.

4. ***Industry-specific approaches***: Some methods are designed to work well only within specific industries and may not be easily adaptable to startups operating in diverse sectors.

Issues in Existing Profit Prediction Methods for Startups:

1. ***Inaccuracy due to static models:*** Static financial ratios and benchmarks may not accurately reflect the dynamic nature of startups, leading to less accurate profit predictions.

2. ***Limited adaptability***: Existing methods struggle to adapt to the rapid growth and change that startups often experience, making historical trends less reliable for future predictions.

3. ***Incomplete predictions***: Non-financial factors that play a crucial role in a startup's profitability are often neglected, resulting in incomplete predictions.

4. ***Lack of innovation consideration***: Traditional methods may fail to capture the innovative and disruptive nature of startups, hindering accurate profit predictions.

5. ***Restricted applicability***: Some methods are designed for specific industries, limiting their applicability to startups operating in diverse sectors and impeding their generalization.

To overcome these challenges, advanced machine learning techniques can be employed to develop more robust profit prediction models for startups. These techniques can address the limitations of traditional approaches by incorporating dynamic variables, considering non-financial factors, and offering scalability and generalization for accurate profit predictions.

3.PROPOSED SYSTEMS

Proposed Methods for Profit Prediction in Startups:

The proposed approach for profit prediction in startups includes a combination of various regression algorithms and ensemble methods. The following methods are suggested as part of the proposed system:

Regression :

Regression models are powerful tools in predictive analytics that aim to establish a mathematical relationship between input variables and a continuous target variable. These models, such as linear regression, lasso regression, ridge regression, and support vector regression, enable us to make predictions based on the observed patterns in the data. By estimating the coefficients or weights associated with each input variable, regression models quantify the impact of these variables on the target variable, allowing us to understand the relationship and make informed predictions. These models can handle both numerical and categorical input variables and are widely used in various domains, including finance, economics, marketing, and healthcare. Regression models provide valuable insights into the factors influencing the target variable and serve as a foundation for decision-making and optimization processes.

1. Linear Regression: Linear regression models establish a linear relationship between input variables (R&D spend, administration spend, marketing spend) and profit. It serves as a baseline for profit prediction.

2. Lasso Regression: Lasso regression performs feature selection and regularization, identifying relevant variables and providing an interpretable model by penalizing less important features.

3. Ridge Regression: Ridge regression minimizes multicollinearity among input variables by adding a penalty term to the objective function, improving model stability and generalization.

4. Decision Tree: Decision tree models capture non-linear relationships and interactions between variables, providing interpretable rules for profit prediction.

5. Random Forest: Random forest combines multiple decision trees to improve prediction accuracy, reducing overfitting and offering robust profit predictions.

6. XGBoost: XGBoost is a gradient boosting algorithm that sequentially builds an ensemble of weak prediction models, optimizing a specific loss function and minimizing errors for accurate profit predictions.

7. Support Vector Regression (SVR): SVR uses support vector machines to identify optimal hyperplanes for profit prediction, effectively handling non-linear relationships and small datasets.

8. Feedforward Neural Network (FFN): FFN, a type of artificial neural network, learns complex patterns and non-linear relationships in data. It can capture intricate interactions between variables, providing highly accurate profit predictions.

9. ElasticNet: ElasticNet combines L1 and L2 regularization techniques, performing feature selection and regularization simultaneously. It offers a stable and interpretable model, particularly suitable for high-dimensional and collinear datasets.

3.1 ALGORITHM

General algorithm for building a regression model:

Here is a short algorithm for regression modeling:

1. Gather and preprocess the data: Collect the dataset, handle missing data, and normalize or standardize the input features.

2. Split the dataset: Divide the data into training and testing sets.

3. Choose a regression algorithm: Select a suitable regression algorithm based on the problem.

4. Train the model: Fit the regression model to the training data.

5. Evaluate the model: Use the testing set to assess the model's performance using evaluation metrics.

6. Fine-tune the model (optional): Adjust hyperparameters or try different algorithms to improve performance.

7. Deploy and use the model: Apply the trained model to new data for making predictions.

4.METHODLOGIES

The methodologies involved in each step of the regression modeling process:

4.1 Data collection:

- Gather relevant data from reliable sources such as databases, APIs, or surveys.

- Ensure the dataset covers an adequate range of samples and includes the target variable (profit) and relevant input variables (R&D spend, administration spend, marketing spend).

4.2 Data preprocessing:

- Clean the data by handling missing values, outliers, and inconsistencies.

- Perform data transformations, such as scaling or normalization, to bring the variables to a similar range and improve model performance.

- Encode categorical variables using appropriate techniques, such as one-hot encoding or label encoding.

4.3 Feature selection:

- Identify the most relevant input variables that significantly contribute to profit prediction.

- Utilize techniques such as correlation analysis, feature importance from tree-based models, or regularization methods (e.g., Lasso) to select the most informative features.

4.4 Split Train and Test sets:

- Divide the dataset into a training set and a separate test set.

- Typically, a common split ratio is 70-80% for training and 20-30% for testing.

- Randomly shuffle the data to ensure a representative distribution in both sets.

4.5 Train the Model:

- Apply the chosen regression algorithm (e.g., linear regression, decision tree, etc.) on the training set.

- Fit the model to the training data, estimating the model parameters or weights.

4.6 Evaluate the Model:

- Use the test set to evaluate the model's performance.

- Calculate metrics such as mean squared error (MSE), mean absolute error (MAE), or R-squared to assess how well the model predicts profit.

- Visualize and interpret the model's residuals to identify any patterns or issues.

4.7 Optimize the Model:

- Fine-tune the model by adjusting hyperparameters, such as regularization strength or tree depth, using techniques like grid search or random search.

- Consider using cross-validation to get a more robust estimate of the model's performance.

4.8 Deploy the Model:

- Once satisfied with the model's performance, deploy it to make profit predictions on new, unseen data.

- Monitor the model's performance in production and periodically retrain or update the model as new data becomes available.

These methodologies help ensure the effectiveness and accuracy of the regression model for profit prediction in startups.

5.IMPLEMENTATION

5.1 SOURCE CODE

```
# importing necessary libraries

import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as pt
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import Lasso
from sklearn.linear_model import Ridge
from sklearn.linear_model import ElasticNet # Elastic net regression
from sklearn.svm import SVR
import xgboost as xgb
import tensorflow as tf
from sklearn.preprocessing import MinMaxScaler

# Data reading
data=pd.read_csv('C:\\Users\\Amogh Prabhu\\Desktop\\50_startups.csv')

# Data preprocessing and Visulization

print(data)
print(data.info())
print(data.columns)

# correlation matrix

co_matrix=data[['R&D Spend', 'Administration', 'Marketing Spend',
'Profit']].corr()
sb.heatmap(co_matrix,annot=True,cmap='coolwarm')
pt.title('Correlation Matrix')
pt.show()

#pair plot

sb.pairplot(data,x_vars=['R&D Spend', 'Administration', 'Marketing
Spend'],y_vars='Profit',kind='scatter')
pt.show()
```

```
# regression plots
sb.lmplot(x = 'R&D Spend',y='Profit',data=data)
sb.lmplot(x = 'Administration',y='Profit',data=data)
sb.lmplot(x = 'Marketing Spend',y='Profit',data=data)
pt.show()

# Distribution plots
sb.distplot(data['Profit'],color='maroon')
pt.show()

# Histogram plot
sb.histplot(data['Profit'],color='seagreen')

# models training ,testing and evalvation

y=data['Profit']
X=data[['R&D Spend', 'Administration', 'Marketing Spend']]


# Test Train split

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




# Linear regression
linear_reg = LinearRegression() # instance of class
linear_reg.fit(X_train,y_train)
linear_reg_pred = linear_reg.predict(X_test)
linear_reg_r2 = r2_score(y_test,linear_reg_pred) # coefficient of
determination
linear_reg_mse=mean_squared_error(y_test,linear_reg_pred) #MSE
linear_reg_mae=mean_absolute_error(y_test,linear_reg_pred) #MAE


# random forest regression
randfor_reg = RandomForestRegressor()
randfor_reg.fit(X_train,y_train)
randfor_reg_pred=randfor_reg.predict(X_test)
randfor_reg_r2=r2_score(y_test,randfor_reg_pred)
randfor_reg_mse=mean_squared_error(y_test,randfor_reg_pred)
randfor_reg_mae=mean_absolute_error(y_test,randfor_reg_pred)
```

```
# Decision Tree
dectree_reg = DecisionTreeRegressor()
dectree_reg.fit(X_train,y_train)
dectree_reg_pred = dectree_reg.predict(X_test)
dectree_reg_r2=r2_score(y_test,dectree_reg_pred)
dectree_reg_mse=mean_squared_error(y_test,dectree_reg_pred)
dectree_reg_mae = mean_absolute_error(y_test,dectree_reg_pred)


# Lasso regression model
lasso_reg = Lasso(alpha=0.1)
lasso_reg.fit(X_train,y_train)
lasso_reg_pred=lasso_reg.predict(X_test)
lasso_reg_r2=r2_score(y_test,lasso_reg_pred)
lasso_reg_mse=mean_squared_error(y_test,lasso_reg_pred)
lasso_reg_mae=mean_absolute_error(y_test,lasso_reg_pred)


# Ridge regression model
ridge_reg=Ridge(alpha = 0.1)
ridge_reg.fit(X_train,y_train)
ridge_reg_pred=ridge_reg.predict(X_test)
ridge_reg_r2=r2_score(y_test,ridge_reg_pred)
ridge_reg_mse=mean_squared_error(y_test,ridge_reg_pred)
ridge_reg_mae=mean_absolute_error(y_test,ridge_reg_pred)


# Elastic Net regression
elasticnet_reg=ElasticNet(alpha=0.1,l1_ratio=0.5) # lasso+ ridge
elasticnet_reg.fit(X_train,y_train)
elasticnet_reg_pred=elasticnet_reg.predict(X_test)
elasticnet_reg_r2=r2_score(y_test,elasticnet_reg_pred)
elasticnet_reg_mse=mean_squared_error(y_test,elasticnet_reg_pred)
elasticnet_reg_mae=mean_absolute_error(y_test,elasticnet_reg_pred)


# support vector regression
svr=SVR(kernel = 'linear')
svr.fit(X_train,y_train)
svr_pred=svr.predict(X_test)
svr_r2=r2_score(y_test,svr_pred)
svr_mse=mean_squared_error(y_test,svr_pred)
svr_mae=mean_absolute_error(y_test,svr_pred)
```

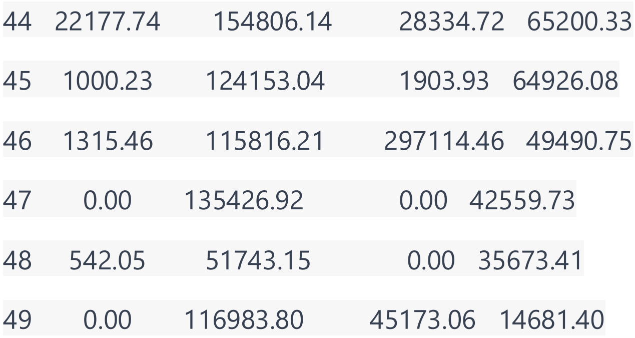
```
#XGBoost
xgb_reg=xgb.XGBRegressor()
xgb_reg.fit(X_train,y_train)
xgb_reg_pred=xgb_reg.predict(X_test)
xgb_reg_r2=r2_score(y_test,xgb_reg_pred)
xgb_reg_mse=mean_squared_error(y_test,xgb_reg_pred)
xgb_reg_mae=mean_absolute_error(y_test,xgb_reg_pred)

#Feedforward Neural Networks
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Feedforward Neural Network model
fnn_model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(32,activation='relu',input_shape=(X_train_scaled.sha
pe[1],)),
    tf.keras.layers.Dense(32,activation='relu'),
    tf.keras.layers.Dense(1)
])
fnn_model.compile(optimizer= 'adam' , loss = 'mean_squared_error')
fnn_model.fit(X_train_scaled,y_train,epochs = 10,batch_size = 32)
fnn_predict = fnn_model.predict(X_test_scaled)
fnn_r2 = r2_score(y_test,fnn_predict)
fnn_mse= mean_squared_error(y_test,fnn_predict)
fnn_mae = mean_absolute_error(y_test,fnn_predict)


# finding BEST MODEL
result = pd.DataFrame({
    'Model': ['Linear' ,'Lasso' , 'Ridge', 'Elasticnet', 'Random Forest',
'Decision Tree' , 'SVR','XGBoost', 'FNN'],
    'R2 Score':
[linear_reg_r2,lasso_reg_r2,ridge_reg_r2,elasticnet_reg_r2,randfor_reg_r2,dect
ree_reg_r2,svr_r2,xgb_reg_r2,fnn_r2],
     'MSE':
[linear_reg_mse,lasso_reg_mse,ridge_reg_mse,elasticnet_reg_mse,randfor_reg_mse
,dectree_reg_mse,svr_mse,xgb_reg_mse,fnn_mse],
     'MAE':
[linear_reg_mae,lasso_reg_mae,ridge_reg_mae,elasticnet_reg_mae,randfor_reg_mae
,dectree_reg_mae,svr_mae,xgb_reg_mae,fnn_mae],
})

print(result)
```

```
result['MSE Rank'] = result['MSE'].rank(ascending=True , method = 'min')
result['MAE Rank'] = result['MAE'].rank(ascending=True , method = 'min')
result['R2 Rank']  = result['R2 Score'].rank(ascending=False, method = 'min')
result['Total Rank'] = result['MSE Rank']+result['MAE Rank']+result['R2 Rank']
result_sorted_rank = result.sort_values('Total Rank')
best_model_rank=result_sorted_rank.iloc[0]['Model']
print("BEST model Based on Total Rank: " , best_model_rank)


# sample prediction

rd_spend = float(input("Enter R & D spend : "))
admin = float(input("Enter Administration Cost: "))
market= float(input("Enter Markenting Expenditure :"))

## we will select based on best model
# print("The predicted value of startup
is:",float(lasso_reg.predict([[rd_spend,admin,market]])))
# print("The predicted value of startup
is:",float(randfor_reg.predict([[rd_spend,admin,market]])))
```

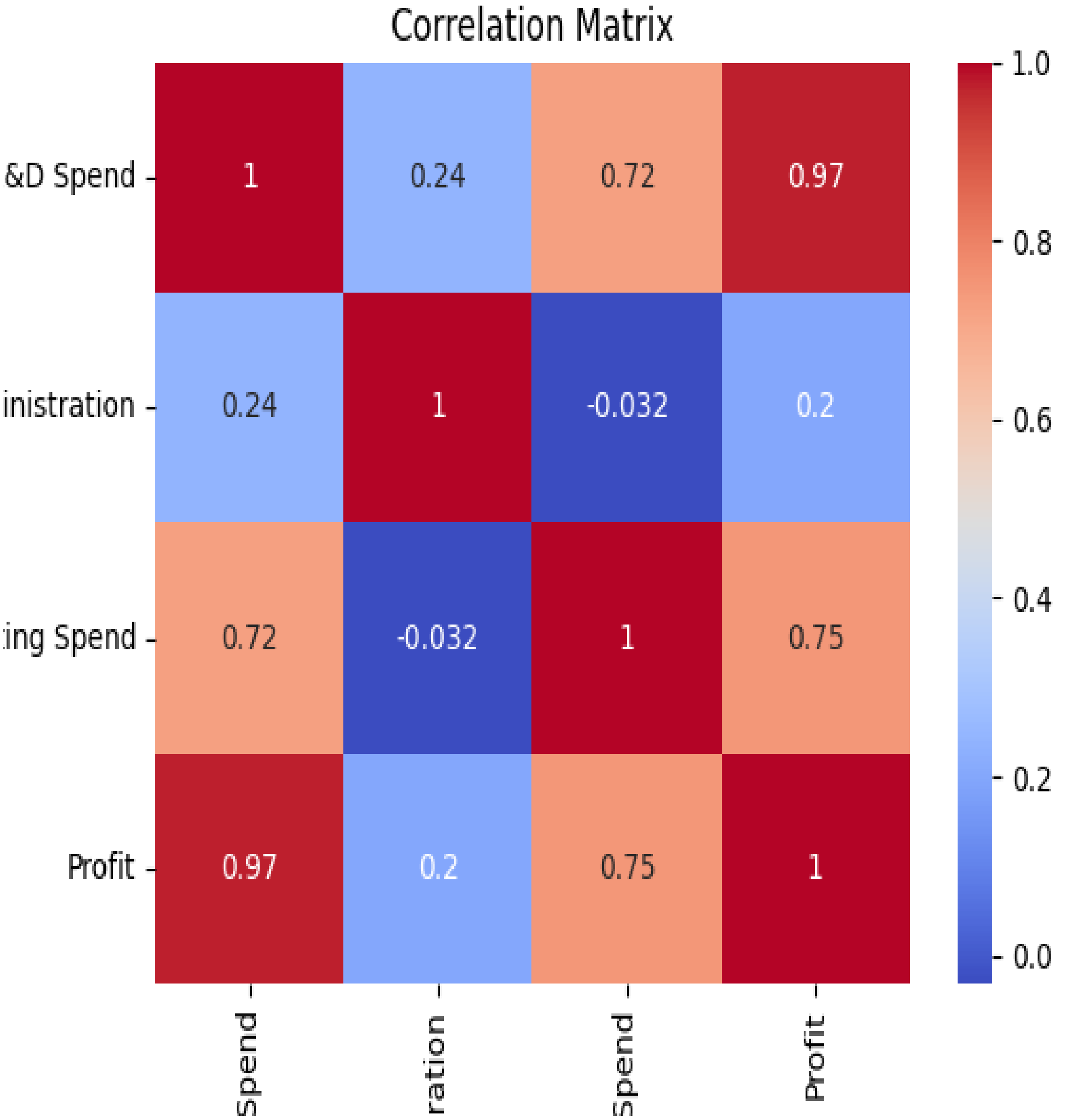
5.2 OUTPUT

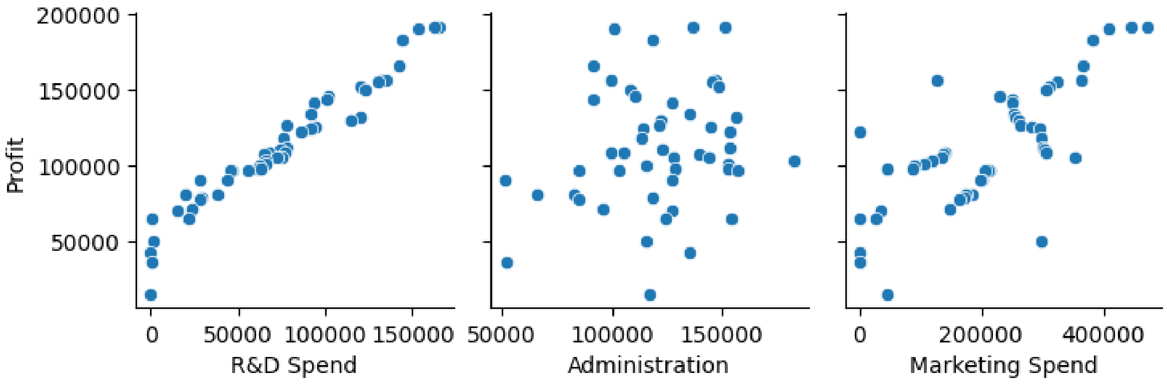
```
   R&D Spend  Administration  Marketing Spend     Profit
0   165349.20       136897.80        471784.10  192261.83
1   162597.70       151377.59        443898.53  191792.06
2   153441.51       101145.55        407934.54  191050.39
3   144372.41       118671.85        383199.62  182901.99
4   142107.34        91391.77        366168.42  166187.94
5   131876.90        99814.71        362861.36  156991.12
6   134615.46       147198.87        127716.82  156122.51
7   130298.13       145530.06        323876.68  155752.60
8   120542.52       148718.95        311613.29  152211.77
9   123334.88       108679.17        304981.62  149759.96
10  101913.08       110594.11        229160.95  146121.95
11  100671.96        91790.61        249744.55  144259.40
12   93863.75       127320.38        249839.44  141585.52
```

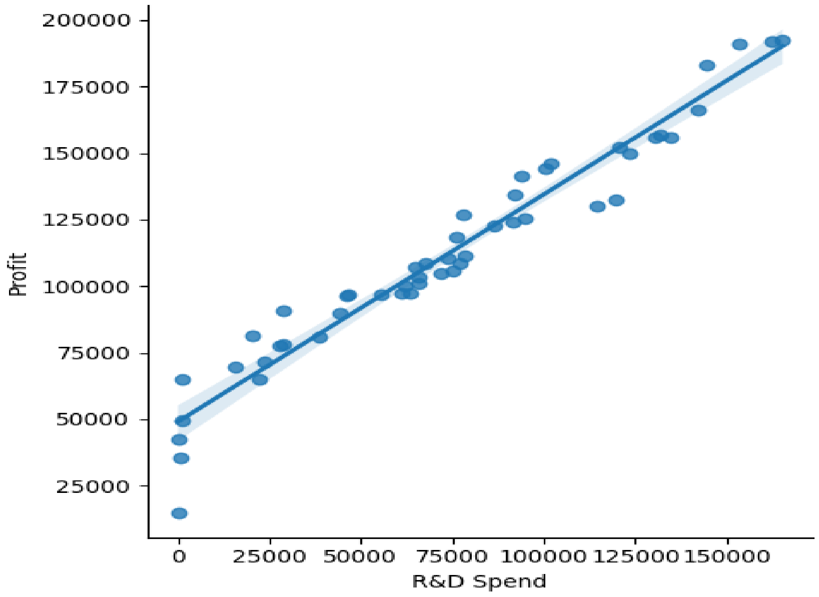
13	91992.39	135495.07	252664.93	134307.35
14	119943.24	156547.42	256512.92	132602.65
15	114523.61	122616.84	261776.23	129917.04
16	78013.11	121597.55	264346.06	126992.93
17	94657.16	145077.58	282574.31	125370.37
18	91749.16	114175.79	294919.57	124266.90
19	86419.70	153514.11	0.00	122776.86
20	76253.86	113867.30	298664.47	118474.03
21	78389.47	153773.43	299737.29	111313.02
22	73994.56	122782.75	303319.26	110352.25
23	67532.53	105751.03	304768.73	108733.99
24	77044.01	99281.34	140574.81	108552.04
25	64664.71	139553.16	137962.62	107404.34
26	75328.87	144135.98	134050.07	105733.54
27	72107.60	127864.55	353183.81	105008.31
28	66051.52	182645.56	118148.20	103282.38
29	65605.48	153032.06	107138.38	101004.64
30	61994.48	115641.28	91131.24	99937.59
31	61136.38	152701.92	88218.23	97483.56
32	63408.86	129219.61	46085.25	97427.84
33	55493.95	103057.49	214634.81	96778.92
34	46426.07	157693.92	210797.67	96712.80
35	46014.02	85047.44	205517.64	96479.51
36	28663.76	127056.21	201126.82	90708.19
37	44069.95	51283.14	197029.42	89949.14
38	20229.59	65947.93	185265.10	81229.06
39	38558.51	82982.09	174999.30	81005.76
40	28754.33	118546.05	172795.67	78239.91
41	27892.92	84710.77	164470.71	77798.83
42	23640.93	96189.63	148001.11	71498.49
43	15505.73	127382.30	35534.17	69758.98

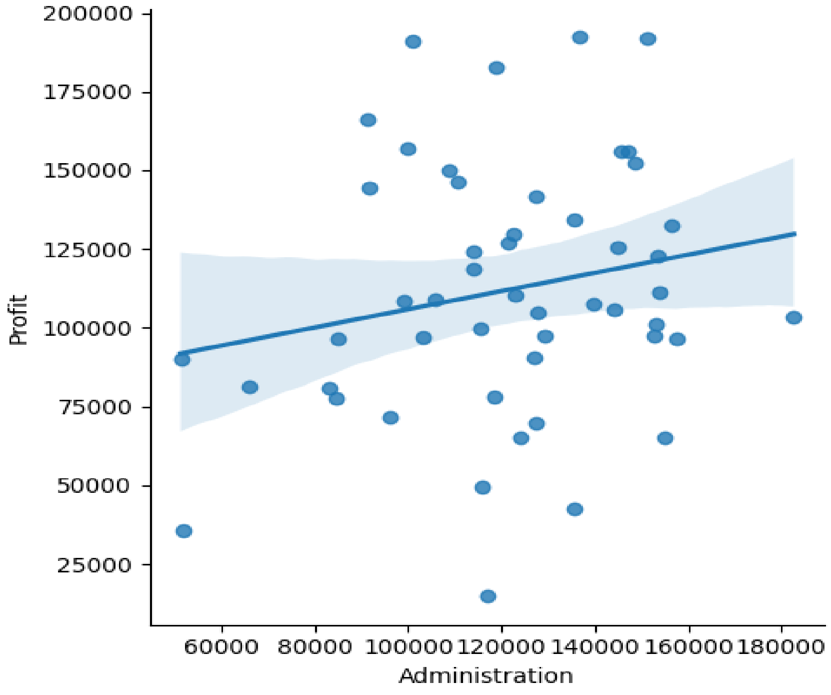


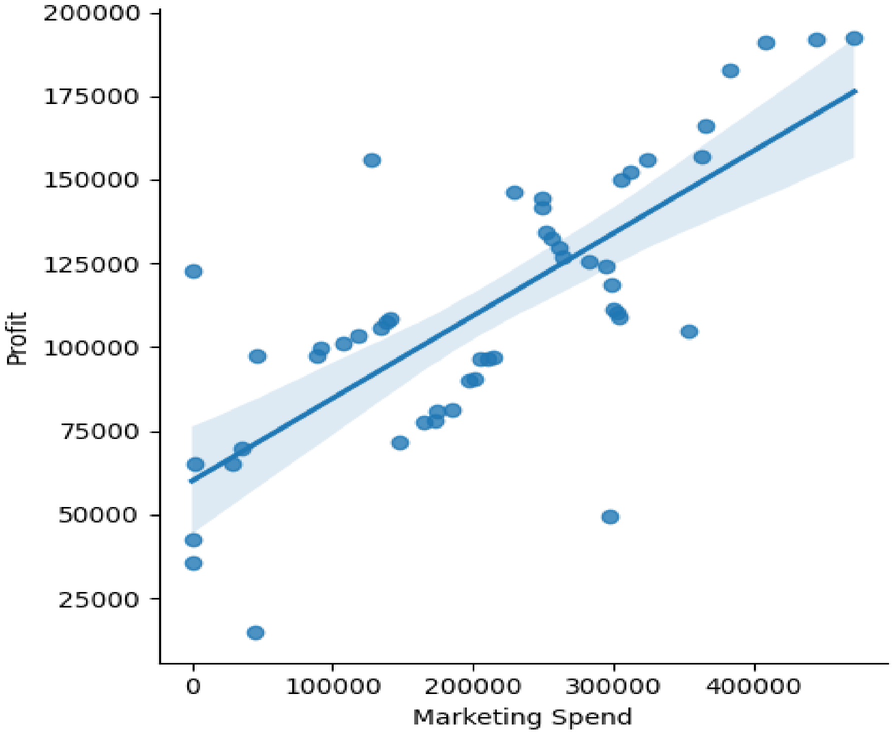
44	22177.74	154806.14	28334.72	65200.33
45	1000.23	124153.04	1903.93	64926.08
46	1315.46	115816.21	297114.46	49490.75
47	0.00	135426.92	0.00	42559.73
48	542.05	51743.15	0.00	35673.41
49	0.00	116983.80	45173.06	14681.40

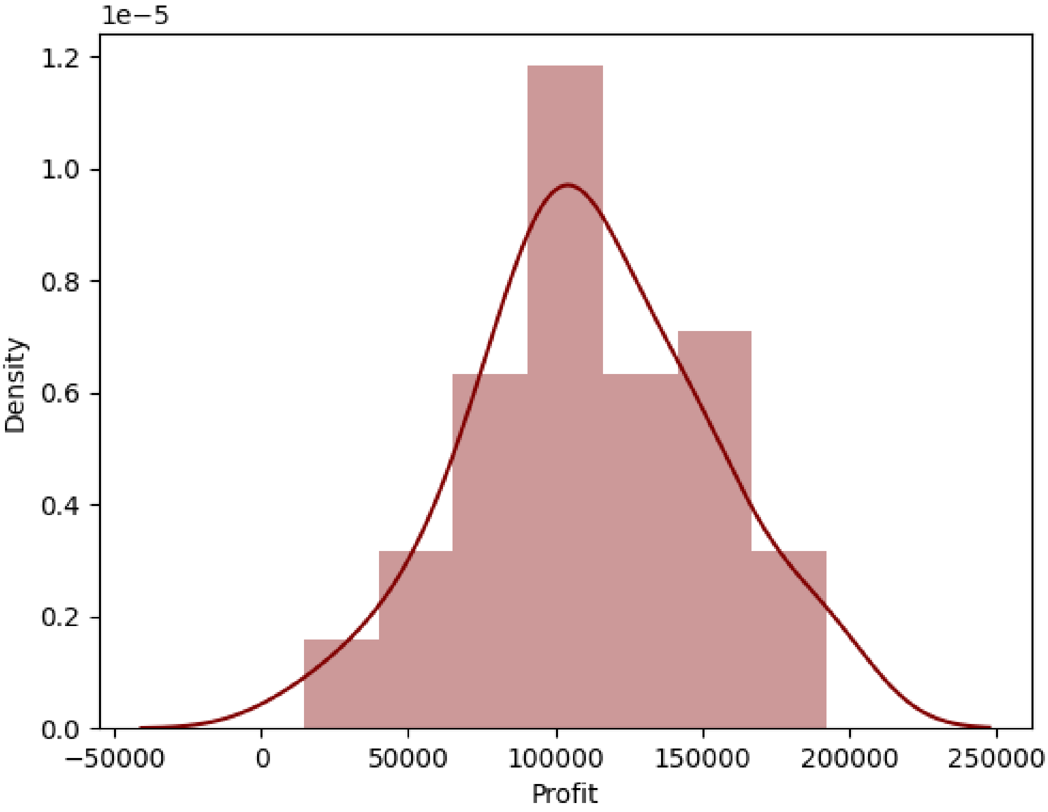



200000
150000
100000
50000
Profit
0
50000
100000
150000
R&D Spend
50000
100000
150000
Administration
0
200000
400000
Marketing Spend


200000
175000
150000
125000
100000
75000
50000
25000
Profit
0
25000
50000
75000
100000
125000
150000
R&D Spend


200000
175000
150000
125000
100000
75000
50000
25000
Profit
60000
80000
100000
120000
140000
160000
180000
Administration


200000
175000
150000
125000
100000
75000
50000
25000
Profit
0
100000
200000
300000
400000
Marketing Spend


1e−5
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Density
−50000
0
50000
100000
150000
200000
250000
Profit

Results from all regression models

1.R2 score

2.MSE (mean squared error)

3.MAE(mean absolute error)

	Model	R2 Score	MSE	MAE
0	Linear	0.900065	8.092632e+07	6979.152252
1	Lasso	0.900065	8.092632e+07	6979.152251
2	Ridge	0.900065	8.092632e+07	6979.152252
3	Elasticnet	0.900065	8.092632e+07	6979.152252
4	Random Forest	0.891267	8.805111e+07	6201.931300
5	Decision Tree	0.536854	3.750519e+08	13038.197000
6	SVR	0.871779	1.038321e+08	7702.623216
7	XGBoost	0.904580	7.727013e+07	7779.489250
8	FNN	-11.728467	1.030741e+10	97455.782537

BEST model Based on Total Rank: Lasso Regression Model

6.CONCLUSION

In conclusion, this project aimed to predict the profit of a startup company using regression models based on variables such as R&D spend, administration spend, and marketing spend. The project followed a systematic methodology, including data collection, preprocessing, feature selection, train-test split, model training, evaluation, optimization, and deployment.

By implementing various regression models such as linear regression, lasso regression, ridge regression, decision tree, random forest, XGBoost, support vector regression, FFN, and ElasticNet, we explored different approaches to profit prediction. Each model had its strengths and considerations, ranging from interpretability and feature selection to handling non-linear relationships and capturing complex patterns.

Through extensive data preprocessing and feature selection, we ensured that the input variables were appropriately prepared and only the most informative features were considered for model training. The models were trained on a dataset consisting of information from 50 startup companies, allowing for a comprehensive understanding of the relationships between the input variables and profit.

Evaluation metrics such as mean squared error (MSE), mean absolute error (MAE), and R-squared were used to assess the performance of the models. By comparing the results, we were able to identify the models that provided the most accurate profit predictions.

The proposed regression models and methodologies showcased the potential for accurately predicting the profit of a startup company based on R&D spend, administration spend, and marketing spend. The project not only provided valuable insights into the factors influencing profit but also offered practical applications for decision-making and optimization in the startup ecosystem.

Overall, this project highlighted the significance of regression models in profit prediction for startups and demonstrated the effectiveness of a range of regression techniques for this purpose. The findings can assist startups in making informed decisions, allocating resources efficiently, and maximizing their profitability.