PROFIT PREDICTION FOR 50 COMPANIES

Organization: Exposys data labs

KANISHKAA R

3rd year student of BTech Artificial Intelligence and Data Science

Easwari engineering college , Chennai.

aikanishkaa01@gmail.com

Keywords: Linear Regression, Lasso Regression , Elastic Net Regression, Ridge Regression

1.Abstract

The ability to predict profit is impossible without a computerized system as many factors must be taken into consideration. In this program, machine learning algorithms are used to predict profit from R&D costs, administration costs, and marketing expenses. Four machine learning algorithms are used in this analysis, such as linear regression, ridge regression, lasso regression, and elastic net regression, to derive a new prediction that is more reliable than a single algorithm.

2.Introduction

Data is produced everywhere in today's world, like when traveling to different locations (GPS data), browsing the internet (internet history), storing pictures, etc. In order to provide a personalized environment to the user, these information's are being used. The challenge is that these data are quite large, and they cannot be processed by a single person or even a team because their sources of production (if a mobile device is turned on then data is generated from that device) make them quite challenging to process. In order to provide users with what they want, Machine Learning makes use of all these data. A core concept of Machine Learning is used to predict the profit of a company, since determining or predicting the profit of any company has become quite challenging in recent years. As many factors affect a company's profit, including R&D costs, administration, marketing, and company standards, the profit of a company is affected by a number Increasing factors affect a company's profit, making things unpredictable for an average person. individual. By analyzing the history of the companies, such as their previous profit record, administration costs, a model is developed that recognizes patterns based on the factors that affect profit, so that profit can be predicted more accurately.

3.Methodology

3.1 Machine Learning

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

3.2 Linear Regression

Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.

3.3 Ridge Regression

Ridge regression is a method of estimating the coefficients of multiple-regression models in scenarios where the independent variables are highly correlated. It has been used in many fields including econometrics, chemistry, and engineering.

3.4 Elastic Net Regression

Elastic net is a penalized linear regression model that includes both the L1 and L2 penalties during training. Using the terminology from "The Elements of Statistical Learning," a hyperparameter "alpha" is provided to assign how much weight is given to each of the L1 and L2 penalties.

3.5 Lasso Regression

In statistics and machine learning, lasso (least absolute shrinkage and selection operator; also Lasso or LASSO) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model.

4. Proposed System and Implementation

The main intention is to predict the value of the dependent variable i.e., the value of the profit of the company based on the data of the company over the previous years. So, from all the techniques used before for the prediction of profit an average from all those predicted values of the dependent variable is computed and made as the predicted dependent variable.

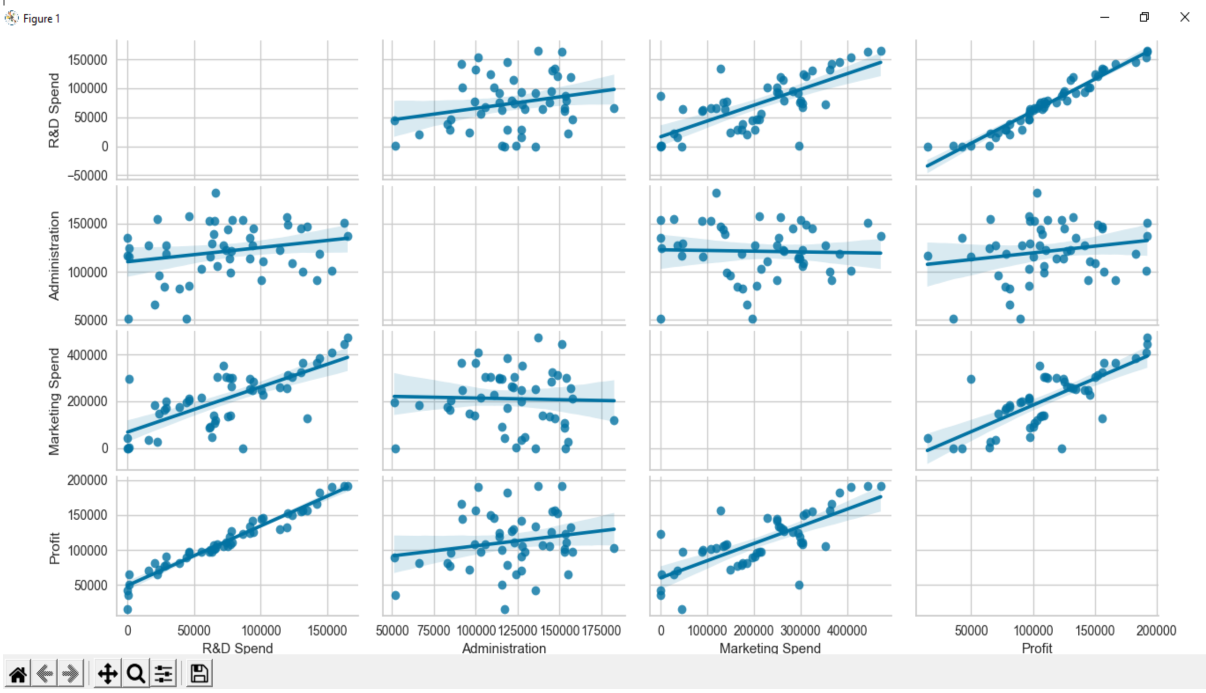
4.1 Dataset implementation

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from yellowbrick.regressor import PredictionError,ResidualsPlot
warnings.filterwarnings("ignore")
df=pd.read_csv('F:\\Exposys Data Science Interns\\50_Startups.csv')
print("dataset loaded..")
df.head()
df.columns
df.dtypes
df.describe()
df.corr()
```

```
*IDLE Shell 3.9.13*
File Edit Shell Debug Options Window Help
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: F:\Exposys Data Science Interns\Prediction_for_50_startups.py ====
dataset loaded..
```

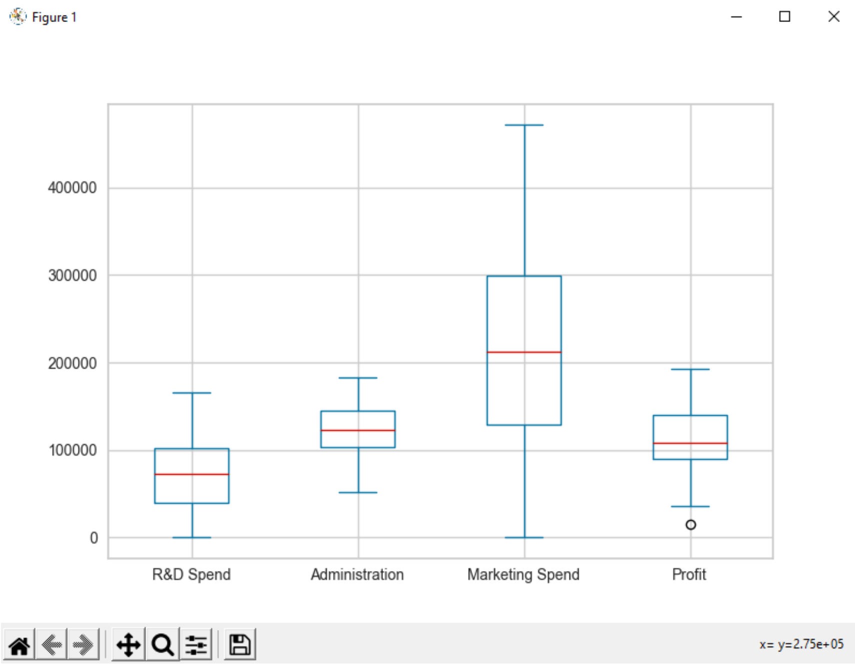
4.2 Pair plot visualization

```
#pairplot
sns.pairplot(df,kind="reg", diag_kind="")
plt.show()
```



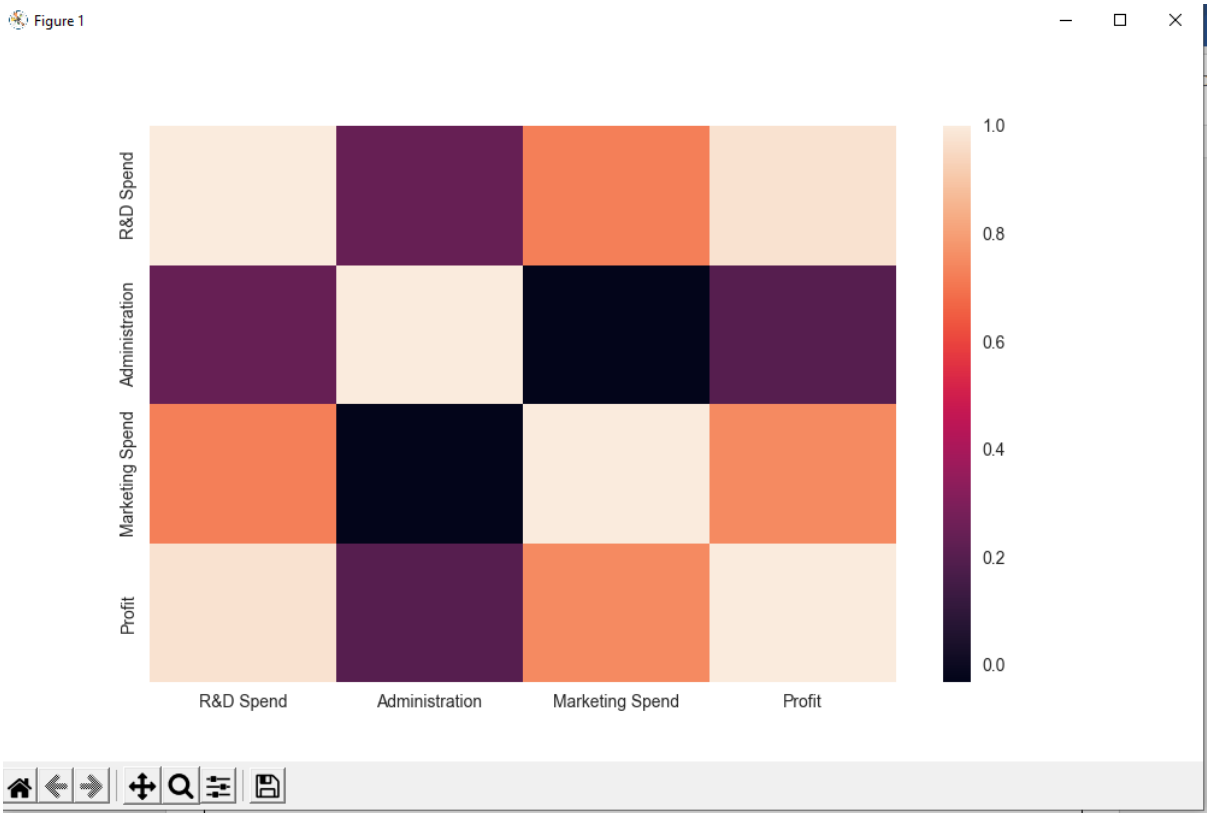
4.3 Box Plot Visualization

```
#BoxPlot
df.plot(kind ='box')
plt.show()
```



4.4 Correlation Heatmap

```
#heatmap
plt.figure(figsize=(10,6))
tc = df.corr()
sns.heatmap(tc)
plt.show()
```



4.5 Train and test data

```
x=df[['R&D Spend','Administration', 'Marketing Spend']]
y=df['Profit']
df_copy = df.copy()
print("copy of dataset is created..")
df_copy.head()
#train_and_test_the_data
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
print("X_train:",x_train.shape)
print("X_test:",x_test.shape)
print("Y_train:",y_train.shape)
print("Y_test:",y_test.shape)
```

IDLE Shell 3.9.13

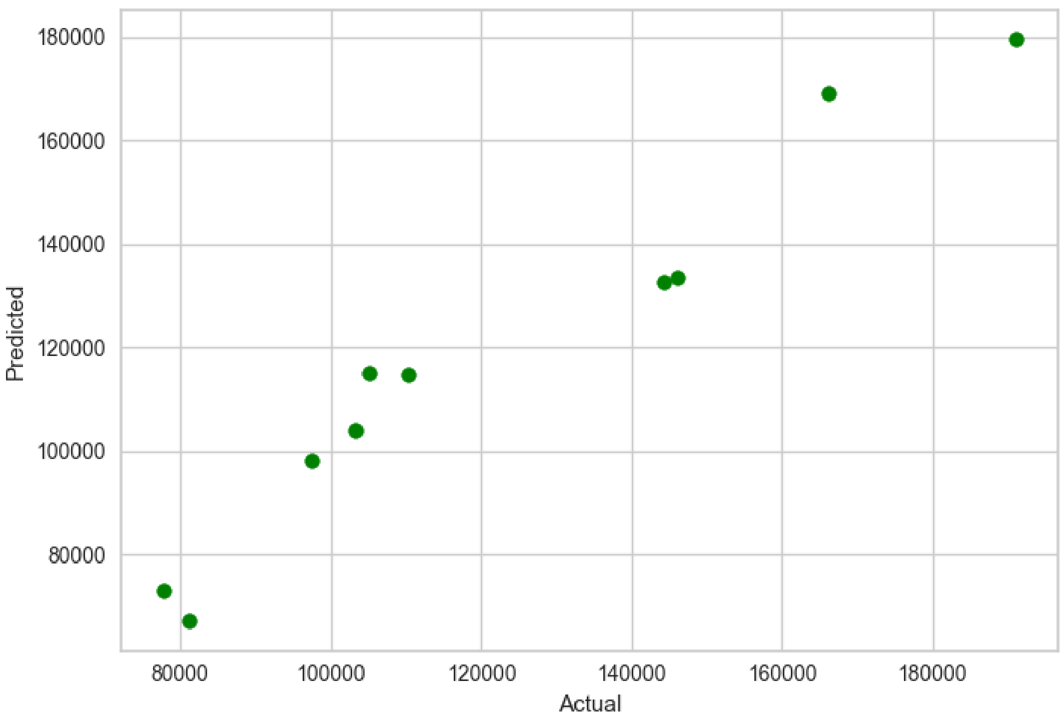
File Edit Shell Debug Options Window Help

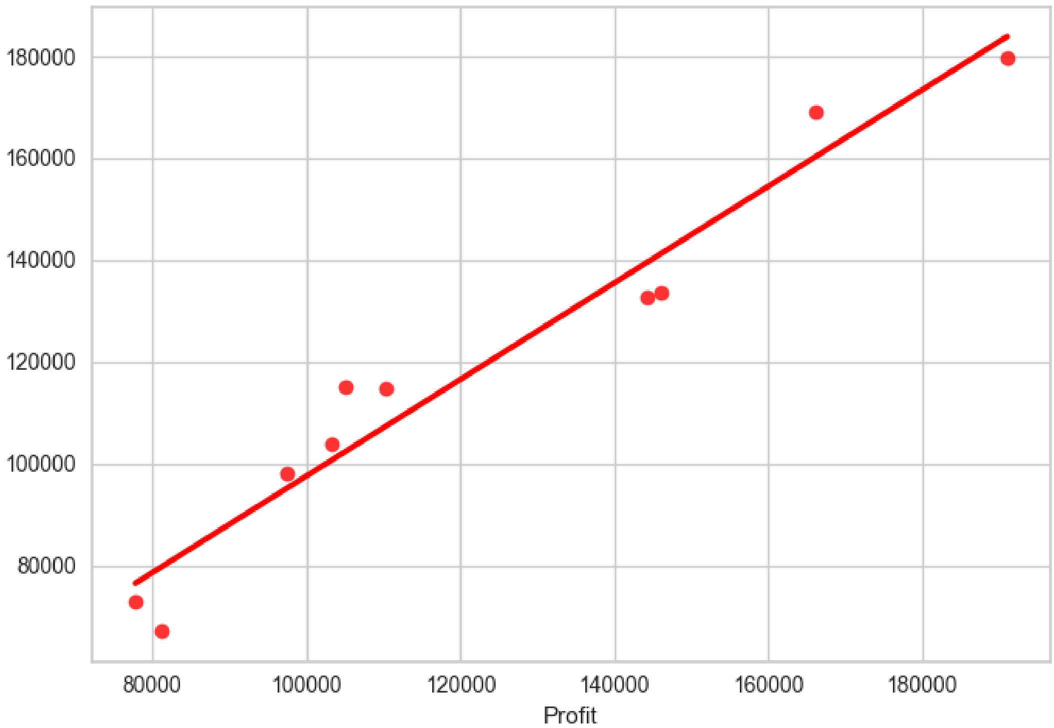
```
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: F:\Exposys Data Science Interns\Prediction_for_50_startups.py ====
dataset loaded..
copy of dataset is created..
X_train: (40, 3)
X_test: (10, 3)
Y_train: (40,)
Y_test: (10,)
```

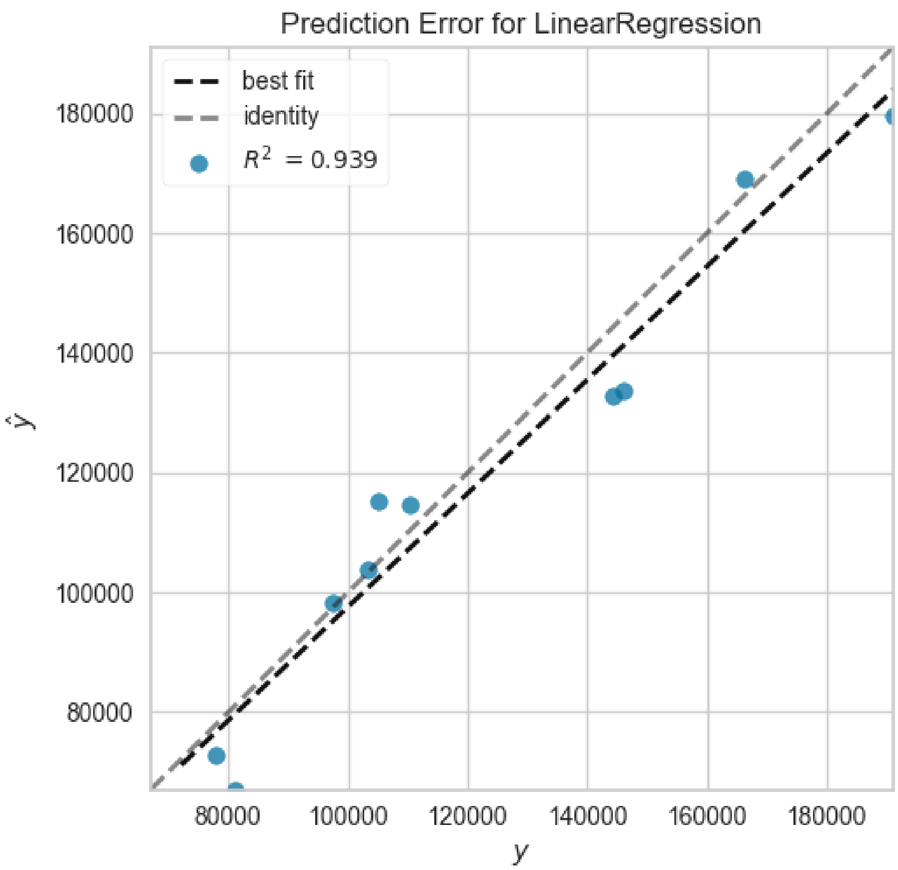
4.6 Building Model

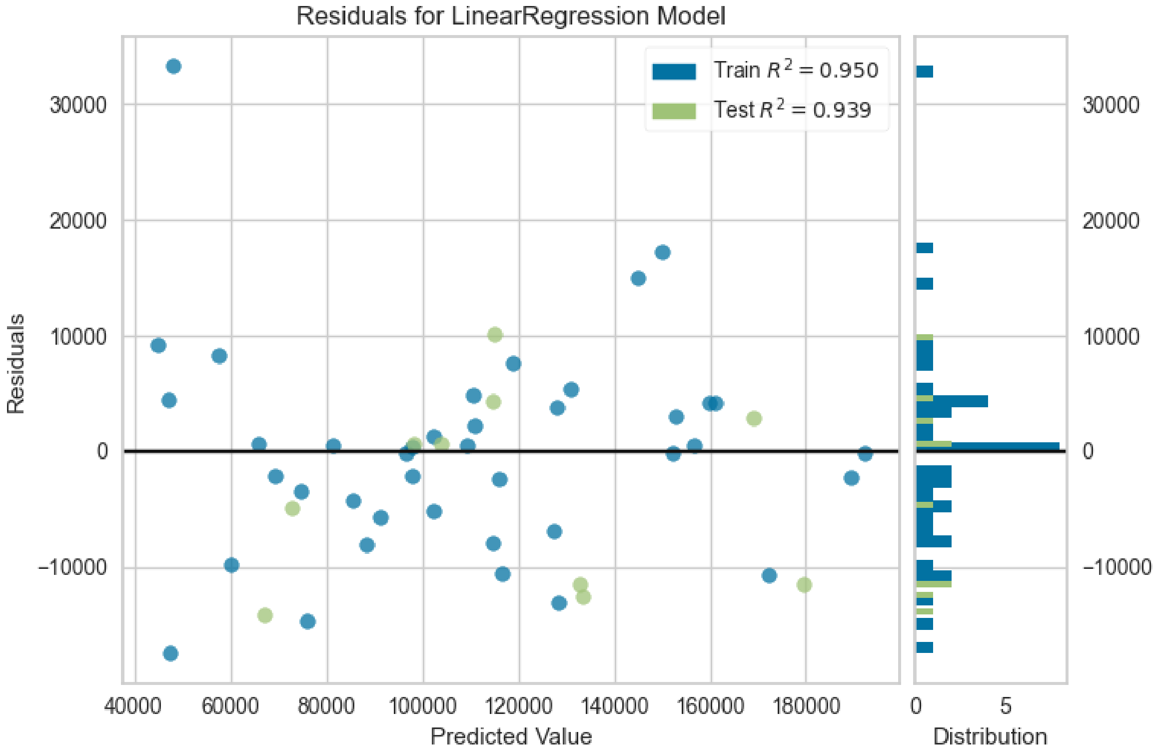
4.6.1 Linear Regression

```
#Build_model
print("********LINEAR_REGRESSION********")
lr=LinearRegression()
lr.fit(x_train,y_train)
y_pred=lr.predict(x_test)
print(y_pred)
#Accuracy_of_the_model
lr.score(x_test,y_test)
plt.scatter(y_test,y_pred,color='green')
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()
sns.regplot(x=y_test,y=y_pred,ci=None,color ='red')
plt.show()
pred_df=pd.DataFrame({'Actual Value':y_test,'Predicted Value':y_pred,'Difference':y_test-y_pred})
print(pred_df)
visualizer=PredictionError(lr)
visualizer.fit(x_train,y_train)
visualizer.score(x_test,y_test)
visualizer.poof()
visualizer=ResidualsPlot(lr)
visualizer.fit(x_train,y_train)
visualizer.score(x_test,y_test)
visualizer.poof()
Accuracy=r2_score(y_test,y_pred)*100
print(" Accuracy of the model is %.2f" %Accuracy)
```




180000
160000
140000
120000
100000
80000
80000
100000
120000
140000
160000
180000
Profit

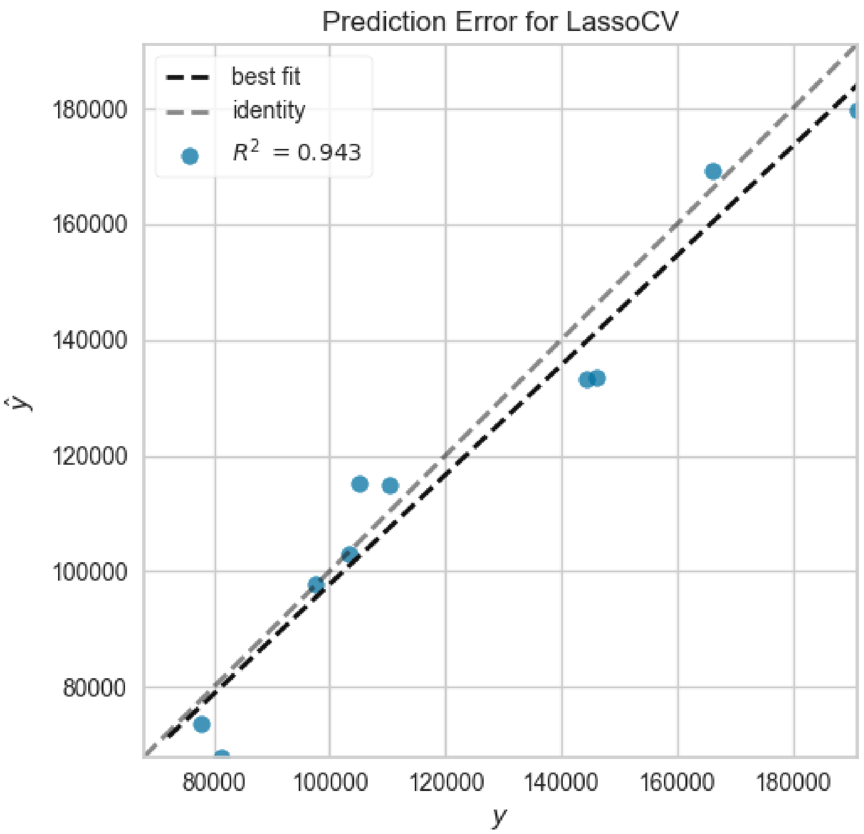

Prediction Error for LinearRegression
best fit
identity
$R^2 = 0.939$
180000
160000
140000
120000
100000
80000
$\hat{y}$
80000
100000
120000
140000
160000
180000
y

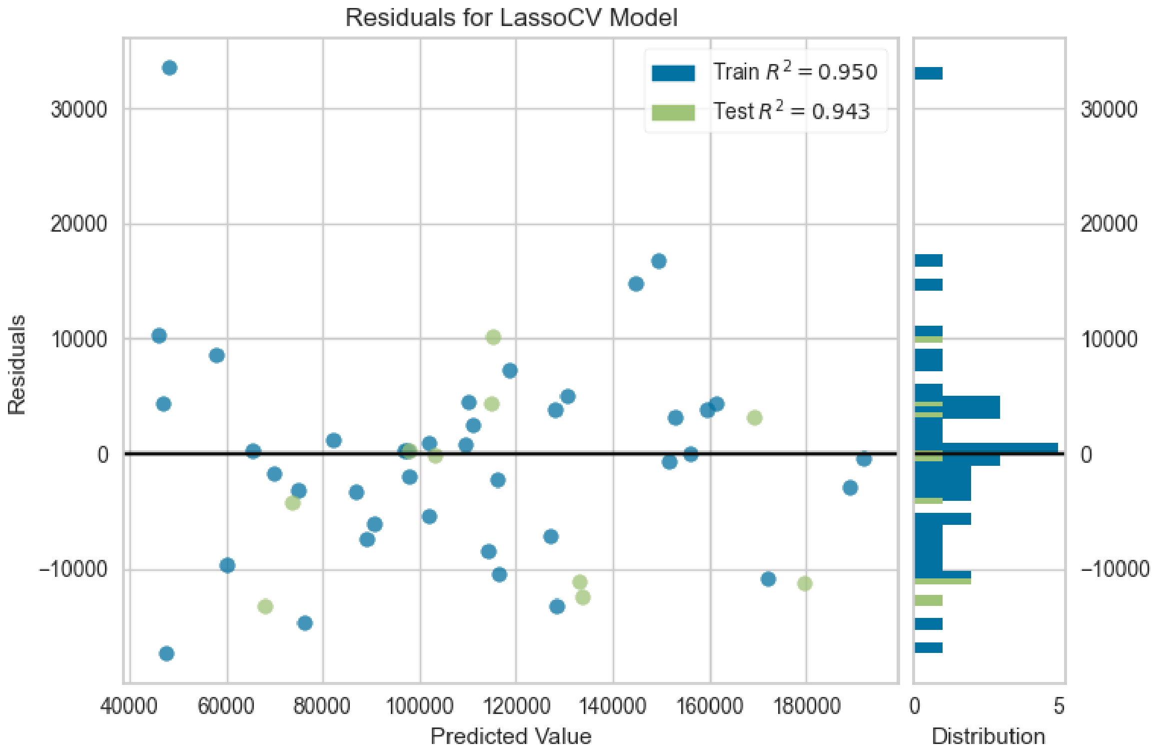



```
*******LINEAR_REGRESSION********
[103901.8969696  132763.05993126 133567.90370044  72911.78976736
 179627.92567224 115166.64864795  67113.5769057   98154.80686776
 114756.11555221 169064.01408795]
    Actual Value  Predicted Value    Difference
28     103282.38     103901.896970   -619.516970
11     144259.40     132763.059931  11496.340069
10     146121.95     133567.903700  12554.046300
41      77798.83      72911.789767   4887.040233
2      191050.39     179627.925672  11422.464328
27     105008.31     115166.648648 -10158.338648
38      81229.06      67113.576906  14115.483094
31      97483.56      98154.806868   -671.246868
22     110352.25     114756.115552  -4403.865552
4      166187.94     169064.014088  -2876.074088
 Accuracy of the model is 93.94 %
```

4.6.2 Lasso Regression

```
print("**********LASSO_REGRESSION******")
from sklearn.linear_model import LassoCV
lc=LassoCV()
lc.fit(x_train,y_train)
y_pred=lc.predict(x_test)
print(y_pred)
lc.score(x_test,y_test)
pred_df=pd.DataFrame({'Actual Value':y_test,'Predicted Value':y_pred,'Difference':y_test-y_pred})
print(pred_df)
Accuracy=r2_score(y_test,y_pred)*100
print(" Accuracy of the model is %.2f" %Accuracy,"%")
visualizer=PredictionError(lc)
visualizer.fit(x_train,y_train)
visualizer.score(x_test,y_test)
visualizer.poof()
visualizer=ResidualsPlot(lc)
visualizer.fit(x_train,y_train)
visualizer.score(x_test,y_test)
visualizer.poof()
```

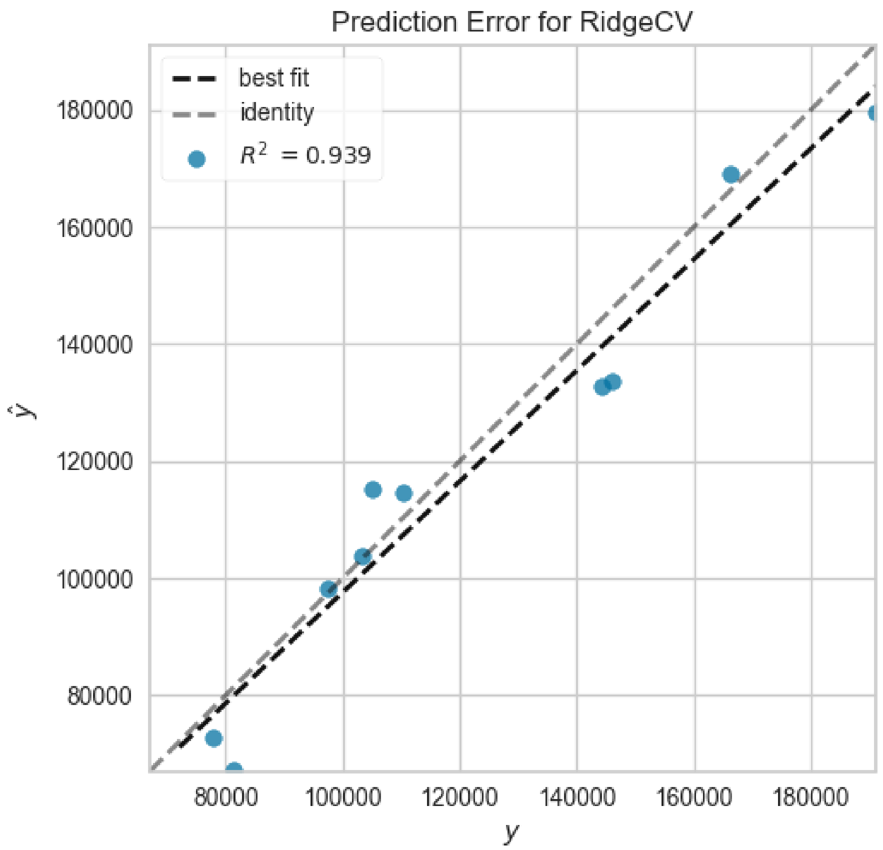


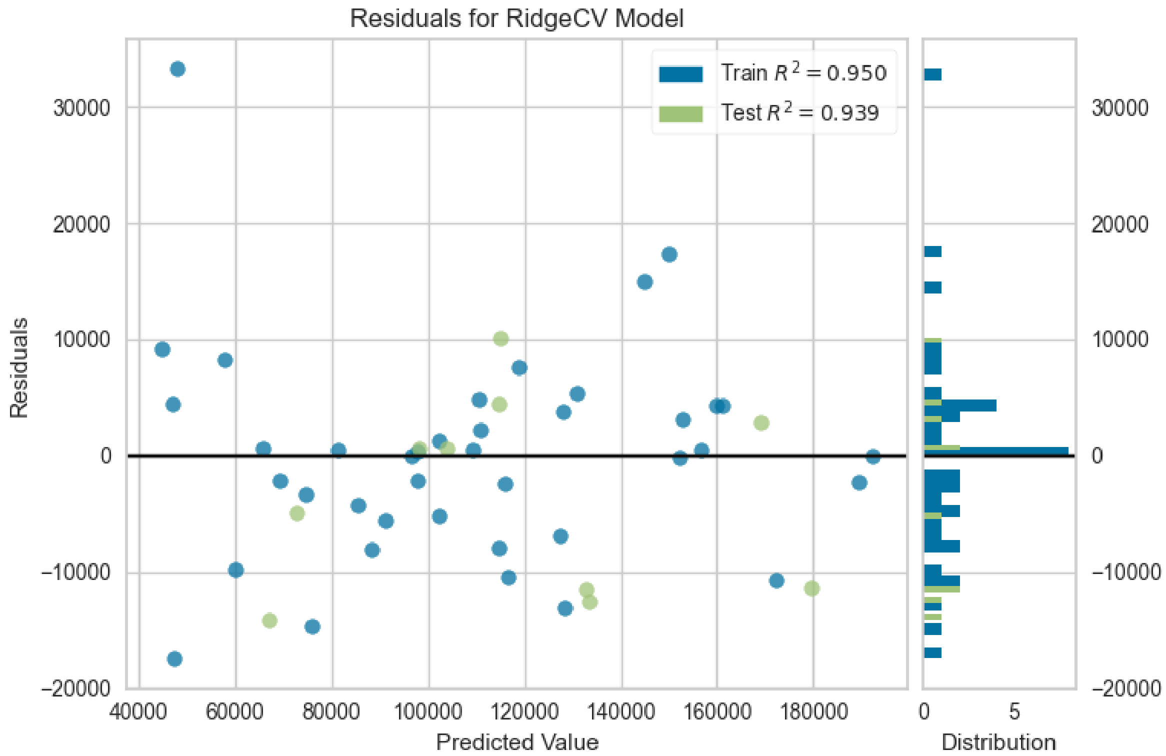



```
**********LASSO_REGRESSION******
[103100.31043118 133122.70252032 133666.98288214  73533.05124801
 179769.83698924 115150.16975146  68011.87435083  97757.45474323
 114787.87464772 169353.87012982]
    Actual Value  Predicted Value    Difference
28     103282.38    103100.310431    182.069569
11     144259.40    133122.702520  11136.697480
10     146121.95    133666.982882  12454.967118
41      77798.83     73533.051248   4265.778752
2      191050.39    179769.836989  11280.553011
27     105008.31    115150.169751 -10141.859751
38      81229.06     68011.874351  13217.185649
31      97483.56     97757.454743   -273.894743
22     110352.25    114787.874648  -4435.624648
4      166187.94    169353.870130  -3165.930130
 Accuracy of the model is 94.28 %
```

4.6.3 Ridge Regression

```
print("*********************RIDGE_REGRESSION*************************")
from sklearn.linear_model import RidgeCV
Rc=RidgeCV()
Rc.fit(x_train,y_train)
y_pred=Rc.predict(x_test)
print(y_pred)
Rc.score(x_test,y_test)
pred_df=pd.DataFrame({'Actual Value':y_test,'Predicted Value':y_pred,'Difference':y_test-y_pred})
print(pred_df)
Accuracy=r2_score(y_test,y_pred)*100
print(" Accuracy of the model is %.2f" %Accuracy,"%")
visualizer=PredictionError(Rc)
visualizer.fit(x_train,y_train)
visualizer.score(x_test,y_test)
visualizer.poof()
visualizer=ResidualsPlot(Rc)
visualizer.fit(x_train,y_train)
visualizer.score(x_test,y_test)
visualizer.poof()
```

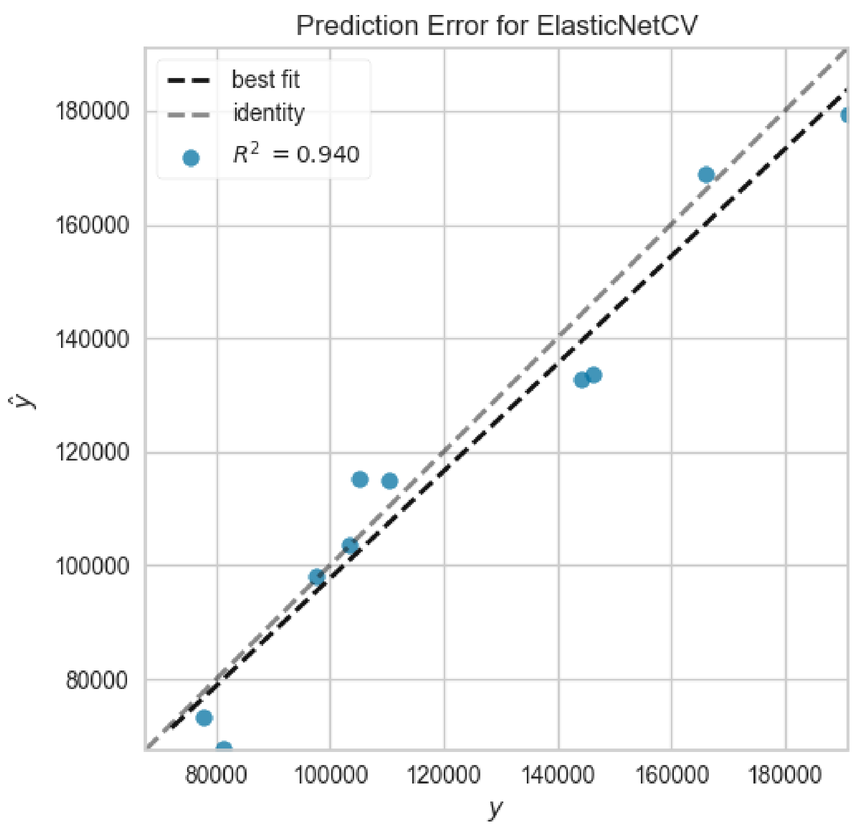


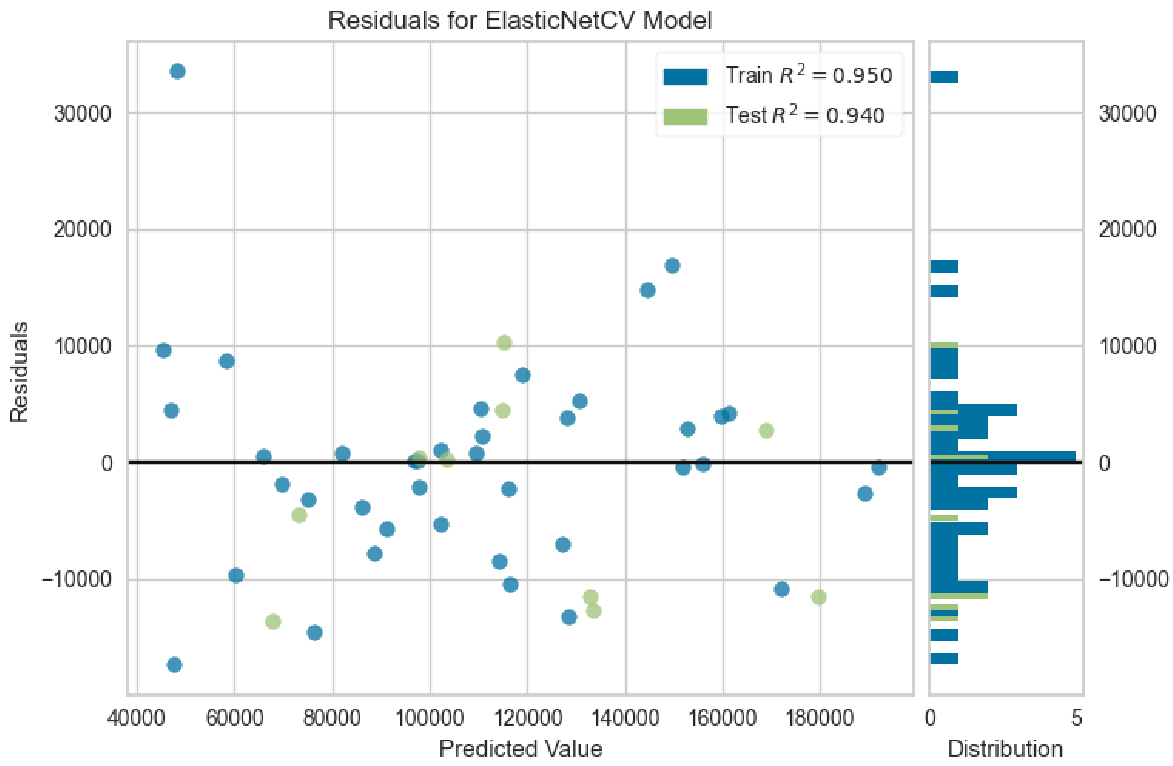



```
*********************RIDGE_REGRESSION*************************
[103901.29706004 132767.62003632 133573.11742993  72899.10396084
 179651.27220923 115176.368985    67099.24980247  98148.86957655
 114762.61530316 169082.14097    ]
    Actual Value  Predicted Value    Difference
28     103282.38    103901.297060   -618.917060
11     144259.40    132767.620036  11491.779964
10     146121.95    133573.117430  12548.832570
41      77798.83     72899.103961   4899.726039
2      191050.39    179651.272209  11399.117791
27     105008.31    115176.368985 -10168.058985
38      81229.06     67099.249802  14129.810198
31      97483.56     98148.869577   -665.309577
22     110352.25    114762.615303  -4410.365303
4      166187.94    169082.140970  -2894.200970
 Accuracy of the model is 93.94 %
```

4.6.4 Elastic Net Regression

```
print("************************ELASTICNET_REGRESSION*****************")
from sklearn.linear_model import ElasticNetCV
ENc=ElasticNetCV()
ENc.fit(x_train,y_train)
y_pred=ENc.predict(x_test)
print(y_pred)
ENc.score(x_test,y_test)
pred_df=pd.DataFrame({'Actual Value':y_test,'Predicted Value':y_pred,'Difference':y_test-y_pred})
print(pred_df)
Accuracy=r2_score(y_test,y_pred)*100
print(" Accuracy of the model is %.2f" %Accuracy,"%")
visualizer=PredictionError(ENc)
visualizer.fit(x_train,y_train)
visualizer.score(x_test,y_test)
visualizer.poof()
visualizer=ResidualsPlot(ENc)
visualizer.fit(x_train,y_train)
visualizer.score(x_test,y_test)
visualizer.poof()
```






```
************************ELASTICNET_REGRESSION*****************
[103576.47841299 132802.19954317 133495.02376736  73264.92468115
 179536.33264852 115325.4584633   67612.81661783  97948.43963948
 114867.23106708 169024.5022347 ]
    Actual Value  Predicted Value    Difference
28     103282.38    103576.478413   -294.098413
11     144259.40    132802.199543  11457.200457
10     146121.95    133495.023767  12626.926233
41      77798.83     73264.924681   4533.905319
2      191050.39    179536.332649  11514.057351
27     105008.31    115325.458463 -10317.148463
38      81229.06     67612.816618  13616.243382
31      97483.56     97948.439639   -464.879639
22     110352.25    114867.231067  -4514.981067
4      166187.94    169024.502235  -2836.562235
 Accuracy of the model is 94.02 %
```

5.Conclusion

In the above model trained dataset, machine learning algorithms such as linear regression, ridge regression, lasso regression, and elastic net regression were applied. For this ML model that can predict a company's profit value based on its R&D Spend, Administration Cost, and Marketing Spend, Lasso regression gave the highest accuracy among these four algorithms.

6.References

[1] Understanding Multiple Regression (The fundamental basis)

[2] https://towardsdatascience.com/understanding-multiple-regression-249b16bde83e.

[3] Support Vector Regression Tutorial for Machine Learning,

[4] https://www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machinelearning/ .

[5] An introduction to support vector regression, https://towardsdatascience.com/an-introduction-tosupport-vector-regression-svr-a3ebc1672c2

[6] How Random Forest works, https://builtin.com/data-science/random-forest-algorithm .

[7] Understanding Random Forest (How the Algorithm Works and Why it Is So Effective) and Decision

Tree, https://towardsdatascience.com/understanding-random-forest-58381e0602d2 .

[8] “The Politics”, by Aristotle in 1877.

[9] Decision Making, the right way to use the wisdom of crowds, https://hbr.org/2018/12/the-right-wayto-use-the-wisdom-of-crowds .

[10] What makes Python the best programming language for machine learning and the best programming language for AI? https://steelkiwi.com/blog/python-for-ai-and-machine-learning/ .

[11] Fabian Pedregosa; Gaël Varoquaux; Alexandre Gramfort; Vincent Michel; Bertrand Thirion; Olivier Grisel; Mathieu Blondel; Peter Prettenhofer; Ron Weiss; Vincent Dubourg; Jake Vanderplas; Alexandre

Passos; David Cournapeau; Matthieu Perrot; Édouard Duchesnay (2011).

[12] "Scikit-learn: Machine Learning in Python" http://jmlr.org/papers/v12/pedregosa11a.html . Journal of

Machine Learning Research. 12: 2825–2830.