PROJECT REPORT

On

"PROFIT ANALYSIS OF STARTUPS USING MULTIPLE LINEAR REGRESSION"

Submitted in partial fulfillment of the requirements to

Malla Reddy University

For the award of the degree

BACHELOR** of **TECHNOLOGY

In

COMPUTER SCIENCE &ENGINEERING

Submitted by

2011CS020371	Thalagaturi Venkata Sai Meenan
2011CS020372	Yanamala Charan Tej Reddy
2011CS020373	Karumanchi Tejaswi
2011CS020374	Tunuri Sundeep
2011CS020375	Ummadi Reddy Vinithanjali

Under the guidance of

M.RAVINDER

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

ABSTRACT

Start-ups are becoming the motor that moves our economy. Google, Apple, and more recently start-ups Airbnb and Uber are companies with tremendous impact in worldwide economy, social interactions and government. Over the past decade, both in the US and Europe, there has been an exponential growth in start-up formation. Thus, it seems a relevant challenge understanding what makes this type of high-risk ventures successful and as such, attractive to investors and entrepreneurs. The ability to predict success is an invaluable competitive advantage for venture capitals on the hunt for investments since first-rate targets are those who have the potential for growing rapidly soon, which ultimately, allows investors to be one step ahead of competition.

We explored a 50 start-up companies database for this model – provided by the website Kaggle.com, with the objective of building a predictive model, through supervised learning, to accurately classify which start-ups are profitable and which aren't. As technologies of information evolve it became possible to achieve highly reliable results in data analysis by manipulating it with complex machine learning algorithms or data mining techniques to define features and characterize robust models.

- The goal is to predict the profit of start-ups on the bases of data provided which are on the bases of Research and Development Spend (R&D Spend), Administration Spend and Marketing Spend. We use multiple regression in this model because we have to predict profit (dependent variable) on bases of multiple field (independent variables) rather than one field just like we done in Simple Linear Regression. This model can help those people who want to invest in startup company by analyzing profit of the company. Using machine learning algorithm on this model we got an accuracy of 96%.

KEYWORDS

- Start-up
- Data analysis
- Research and Development Spend
- Administration Spend
- Marketing Spend
- Machine Learning
- Dependent variables and independent variables.

INDEX

1. INTRODUCTION

Start-ups are companies that make products that venture to an area or market in ways that haven't been done before. This makes start-ups risky and unpredictable as a new product or service may not work among its apparent users and may require constant adjustments before it gets market fit. Ultimately, a start-up is a high-risk company that is in the first stage of operations and commonly related to technology as a product or a service.

Start-ups are booming everywhere as more colleges, governments and private companies invest and stimulate people to pursue their ideas throughout these ventures. Companies are raising millions with ease and achieving billion-dollar valuation in a matter of years. Examples like Uber and Airbnb are changing societies in such impactful ways that regulation had to be created to keep pace with a new reality. Start-ups are having such impact that, ultimately it becomes every investor's ambition to be part of a large acquisition such as Facebook acquiring WhatsApp (a messaging app) for nineteen billion dollars which allowed venture capital fund to have a 50x return on investment. But there is a complication, start-ups are companies with an estimation of 90% probability failure, which means a lot of investments without proper returns.

For companies at every stage of development, accurately measuring profitability is crucial to the creation of effective business practices and financial management. When you know how to calculate profitability and evaluate the profit for startups, no matter the sector, you can become a successful investor. From startups to blue chips, every company needs to keep a close eye on the bottom line. Net income reflects the amount of revenue that remains as profit after accounting for all expenses, debt, income streams, and taxes.

Predicting the profit of a start-up is commonly defined as two-way strategy. With a focus on how a start-up or an investor could explore all this knowledge for a better decision making in investment strategy and monetary gain, this study intends, by applying data mining and machine learning techniques, to create a predictive model that has as the dependent variable a label to classify whether a start-up is profitable or not.

Improved areas of our society are already being improved by the application of machine learning. From healthcare, where by applying segmentation and predictive modelling it is possible to identify different types of treatment for a patient or to even diagnose the patient, to marketing personalization where companies benefit from knowing as much as possible from their clients to create customer-centric experiences all around. Fraud detection, financial services, insurance and even smart cars are all industries creating value, in a short to medium term, through the application of machine learning. It is possible to bring similar advantages to investors in start-ups, by giving the information about which start-ups are closer to a profitable event in their near future they can better choose where to put their chips and have higher returns on their investments.

This machine learning model will be quite helpful in such a situation where we need to find a profit based on how much we are spending in the market and for the market.

To generate the predictive model, we used a supervised machine learning algorithm which is Multiple Linear Regression. This algorithm fits the characteristics of the database. It provides a fast and simple technical implementation. Being able to accurately classify if a start-up had this event in its progress is not only incredibly valuable for all the investors in the start-up world but also, the application of different techniques and features to build models with higher predictive accuracy represents a step forward to not only the academic literature but also the industry.

The main objective of this project is to develop a predictive model to give the profit analysis of a start-up/company. Multiple linear regression takes the independent variable and calculates the dependent variable.

2. MULTIPLE LINEAR REGRESSION

Multiple linear regression (MLR), is a statistical technique that uses two or more explanatory variables to predict the outcome of a response variable. In other words, it can explain the relationship between multiple independent variables against one dependent variable. These independent variables serve as predictor variables, while the single dependent variable serves as the criterion variable. You can use this technique in a variety of contexts, studies and disciplines, including in econometrics and financial inference.

A multiple linear regression analysis is carried out to predict the values of a dependent variable "Y", and a given set of p independent variables (x_1, x_2, …., x_p). In multiple linear regression, the relationship between the dependent variable and the explanatory variables is represented by the following equation

$$y_\mathrm{i} = \mathrm{B}_0 + \mathrm{B}_1 x_1 + \mathrm{B}_2 x_2 + \ldots + \mathrm{B}_p x_p + \mathrm{E}$$

Here,

Y_i = dependent variable

$\mathrm{X_i}$ = explanatory variables

$\mathrm{B_0}$ = y-intercept (constant term)

$\mathrm{B_p}$ =slope coefficients for each explanatory variable

E = the model's error term (also known as the residuals)

Because of the multiple variables, which can be linear or nonlinear, this regression analysis model allows for more variance and precision when it comes to predicting outcomes, as well as understanding the impact of each explanatory variable on the model's total variance.

The major difference between linear regression and multiple linear regression is that in linear regression there is only one independent variable while when we check out, Multiple linear regression there is more than one independent variable.

Using this machine learning algorithm, we will find the profit of the startup and we will be taking profit as the dependent variable and Research and Development Spend (R&D Spend), Administration Spend, Marketing Spend and State as the independent variables. By cleaning the data, we will get more efficiency.

We will be splitting the data into train and test using the thumb rule (80% for training and 20% percent for the testing) to get accurate result.

When there are two or more variables, we will be using multiple linear regression to get more efficiency. And we will be using r^2 Score to statistically measure the representation of the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.

We will be using various graphical representations such as scatter plot, box plots, histogram, density plot, heat map to show the regression model.

3. DESIGN

This model aims to predict the profit of new start-ups based on a certain criterion using multiple linear regression. Prediction of the profits of the start-ups could benefit venture capital list in investing in a particular start-up or not.

Using this model data scientists working for investment group can derive insights from a data and therefore predict whether a particular start-up would be safe to invest or would drain the company's resources. Other useful insights from the data which was used in a model can procurement and can be used in categorizing and analyzing the efficiency of profits made by a start-up. This model also allows the user to check whether a start-up would be profitable in a particular market in which the start-up is aiming to launch. Therefore, this model not only loves investors to check whether a start-up is profitable to invest but also allows them to make proper decisions with start-up investments and its growth.

3.1 Libraries Used

NumPy stands for numerical python which was created in 2005 by Travis Oliphant. NumPy is an open-source project and one can use it freely. it is a python library used for working with arrays, linear algebra, Fourier transform and matrices.

Why Use NumPy?

- In Python we have lists that serve the purpose of arrays, but they are slow to process.
- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
- The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
- Arrays are very frequently used in data science, where speed and resources are very important.

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

- Fast and efficient for manipulating and analyzing data.
- Data from different file objects can be loaded.
- Easy handling of missing data (represented as NaN) in floating point as well as non-floating-point data
- Size mutability: columns can be inserted and deleted from Data Frame and higher dimensional objects
- Data set merging and joining.
- Flexible reshaping and pivoting of data sets
- Provides time-series functionality.
- Powerful group by functionality for performing split-apply-combine operations on data sets.

Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely -

histogram, bar charts, power spectra, error charts etc. It is used along with NumPy to provide an environment that is an effective open-source alternative for Matlab. It can also be used with graphics toolkits like PyQt and wxPython.

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.

Seaborn helps you explore and understand your data. Its plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

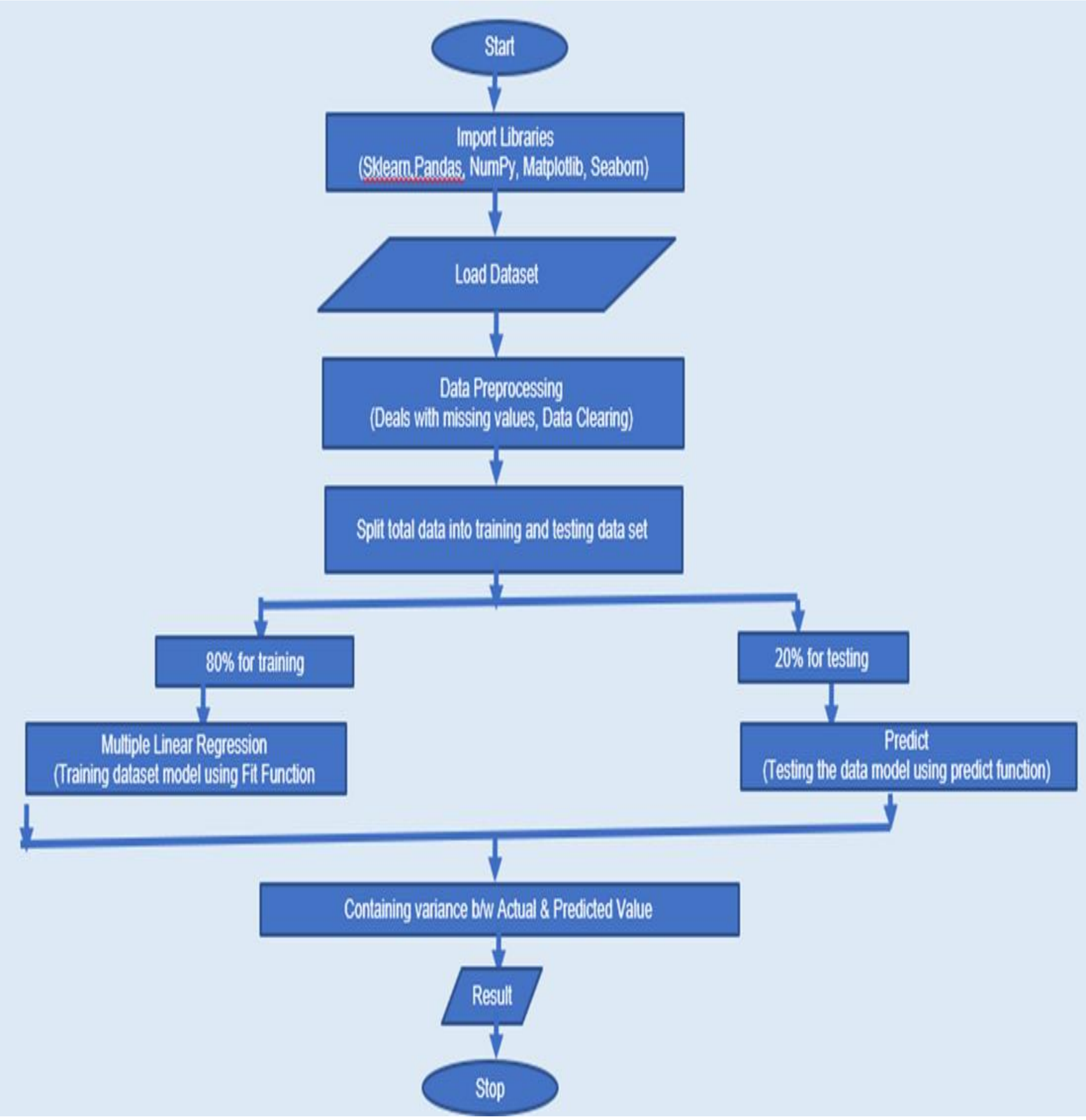
Tkinter is **the standard GUI library for Python**. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Creating a GUI application using Tkinter is an easy task.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the *Tkinter* module.
- Create the GUI application main window.

- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

3.2 FLOW CHART



4. IMPLEMENTATION

4.1 Part 1 — Data Preprocessing

Importing the libraries…

import numpy as np

i*mport* matplotlib.pyplot as plt

import pandas as pd

Importing the dataset

dataset = pd.read_csv('50_Startups.csv')

So, this is what the Dataset looks like…

S.No	R&D Spend	Administration	Marketing Spend	Profit
0	165349.2	136897.8	471784.1	192261.8
1	162597.7	151377.59	443898.53	191792.1
2	153441.51	101145.55	407934.54	191050.4
3	144372.41	118671.85	383199.62	182902
4	142107.34	91391.77	366168.42	166187.9
5	131876.9	99814.71	362861.36	156991.1
6	134615.46	147198.87	127716.82	156122.5
7	130298.13	145530.06	323876.68	155752.6
8	120542.52	148718.95	311613.29	152211.8
9	123334.88	108679.17	304981.62	149760
10	101913.08	110594.11	229160.95	146122
11	100671.96	91790.61	249744.55	144259.4
12	93863.75	127320.38	249839.44	141585.5
13	91992.39	135495.07	252664.93	134307.4
14	119943.24	156547.42	256512.92	132602.7
15	114523.61	122616.84	261776.23	129917
16	78013.11	121597.55	264346.06	126992.9
17	94657.16	145077.58	282574.31	125370.4
18	91749.16	114175.79	294919.57	124266.9
19	86419.7	153514.11	23443.91	122776.9
20	76253.86	113867.3	298664.47	118474

The above dataset contains the following features (independent variables)

1) R&D Spend-Total amount of money spent on Research and Development by the startup.

2) Administration-Total amount of money spent on Administration by the startup.

3) Marketing Spend-Total amount of money spent on Marketing by the startup.

4) State-The State or region in which the startup is launched or operates.

For storing the features in X, we use…

X = dataset.iloc[:, :-1]. values

The dependent variable Profit which tells you the Profit acquired by the Startup.

y = dataset.iloc[:, 4]. values

Machine learning models strictly work with numbers only but if we look at the data set, we can clearly deduce that the state is a string type variable and we cannot feed string type variables to the machine learning model. To overcome this challenge, we use the label in cooler object and create dummy variables using the onehotencoder object.

For example, let us consider we have only two states that is New York and California namely in a data set then one hot encoder will have only two columns.

Similarly for n different states the model would have n columns and each state would be represented by a series of 0s and 1s and all columns would be 0.

Encoding categorical data

Importing the Label Encoder Class along with OneHotEncoder

from sklearn.preprocessing *import* LabelEncoder, OneHotEncoder

Creating an Object of the Label Encoder class labelencoder = LabelEncoder()

As it is clear that the only categorical data is the name of the state which is stored at the 3rd Index in our Dataset so we encode that column!

X [:, 3] = labelencoder.fit_transform(X[:, 3])

onehotencoder = OneHotEncoder(categorical_features = [3])

X = onehotencoder.fit_transform(X).toarray()

Avoiding the Dummy Variable Trap

The Linear Regression equation would look like

Y = b (0) + b (1)x(1) + b (2)x(2) + b (3)x(3) + b (4)D(1) + b (5)D(2) + b (6)D(3) … b(n+3) D(m-1)

Here D (1) …D(m-1) are the m dummy variable's which we had defined earlier in LabelEncoder and OneHotEncoder

Well, if you are sharp enough you might have noticed that the even though there are m dummy variables, we have excluded the last dummy variable D(m)

The reason to that is a concept called **Dummy Variable Trap** in Machine Learning…and to avoid that we must always exclude the last Dummy Variable

If you are more interested then feel free to research a bit on Dummy Variable Trap!!

Dropping the 1st column out of the Dataset which contains one of the OneHotEncoded values…

X = X[:, 1:]

Splitting the dataset into the Training set and Test set

Importing the Libraries and Applying Cross Validation with 80% data as Training Data and 20% as Test Data.

from sklearn.model_selection *import* train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

4.2 Part 2— Fitting our Linear Regression Model

Fitting Multiple Linear Regression to the Training set

The Linear Regression equation would look like — — >

y = b (0) + b (1)x(1) + b (2)x(2) + b (3)x(3) + b (4)D(1) + b (5)D(2) + b (6)D(3) … b(n+3) D(m-1)

Importing the Linear Regression Class

from sklearn.linear_model *import* LinearRegression

Creating an object of the Linear Regression Class

regressor = LinearRegression()

Fit the created object to our training set

regressor.fit(X_train, y_train)

4.3 Part 3 — Predicting the Test set results

y_pred = regressor.predict(X_test)

Printing the predicted values

print(y_pred)

To see the difference b/w predicted and actual results we will also print the test values

print(y_test)

This model has an accuracy percentage of 96% from the data set used. Therefore, we conclude that multiple linear regression works best for the data we used compared to other data models. The efficiency of the model can be further tested with more data procurement. The machine learning model could be more efficient when dealing with statistically insignificant data during our predictions. From our preliminary analysis another method called backward elimination could be used to build an optical model with higher efficiency then 96% and we would like to test the theory in phase 2 of the project. Therefore, using of multiple linear regression was the most optimal choice for this project which yielded the percentage of 96% during our testing.

5. CODE

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

df=pd.read_csv('new_startup.csv')

df

df.head()

df.tail()

df.sample(10)

df.info ()

df=df.drop_duplicates()

df.shape

df.isnull(). sum ()

df

df.head()

target='Profit'
```

```
y=df[target]

x=df.drop(target,axis=1)

x.head()

df.shape

from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2, random_state=1)

x_train.shape,x_test.shape,y_test.shape

from sklearn.linear_model import LinearRegression

mlr=LinearRegression()

mlr.fit(x_train,y_train)

x_test.head()

x_test.shape

y_pred=mlr.predict(x_test)

print(y_pred)

mlr.predict([[0, 0, 0]])

df1=pd.DataFrame({'Actual': y_test,'Predicted': y_pred,'Variance':y_test-y_pred})

df1.head()

mlr.intercept_

print('intercet(b) is:',mlr.intercept_)

mlr.coef_
```

```
print ('coefficient (m) is:',mlr.coef_)

x.head()

from sklearn.metrics import r2_score

score=r2_score(y_test,y_pred)*100

print('score',score)

from sklearn import metrics

print ("Mean Absolute Error:",metrics.mean_absolute_error(y_test,y_pred))

print ("Mean Squared Error:",metrics.mean_squared_error(y_test,y_pred))

print ("Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test,y_pred)))

df.plot(kind='box',subplots=True,sharex=False,sharey=False)

plt.show()

df.hist(xrot=45, figsize=(10,10))

 df.plot(kind='density', subplots=True,sharex=True)

plt.show()

df.plot(kind='box',subplots=True,sharex=False,sharey=False)

plt.show()

import seaborn as sns

plt.figure(figsize=(8,5))

correlation=df.corr(). round (4)

sns.heatmap(data=correlation,annot=True)
```

```
plt.show()

from pandas.plotting import scatter_matrix

scatter_matrix(df)

plt.show()

df.to_csv('startup.csv')

import pandas as pd

from sklearn import linear_model

import tkinter as tk

import matplotlib.pyplot as plt

from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

print ('Intercept: \n', mlr.intercept_)

print ('Coefficients: \n', mlr.coef_)

root= tk.Tk()

canvas1 = tk.Canvas(root, width = 500, height = 300)

canvas1.pack()

Intercept_result = ('Intercept: ', mlr.intercept_)

label_Intercept = tk.Label(root, text=Intercept_result, justify = 'center')

canvas1.create window (260, 220, window=label_Intercept)

Coefficients_result  = ('Coefficients: ', mlr.coef_)

label_Coefficients = tk.Label(root, text=Coefficients_result, justify = 'center')
```

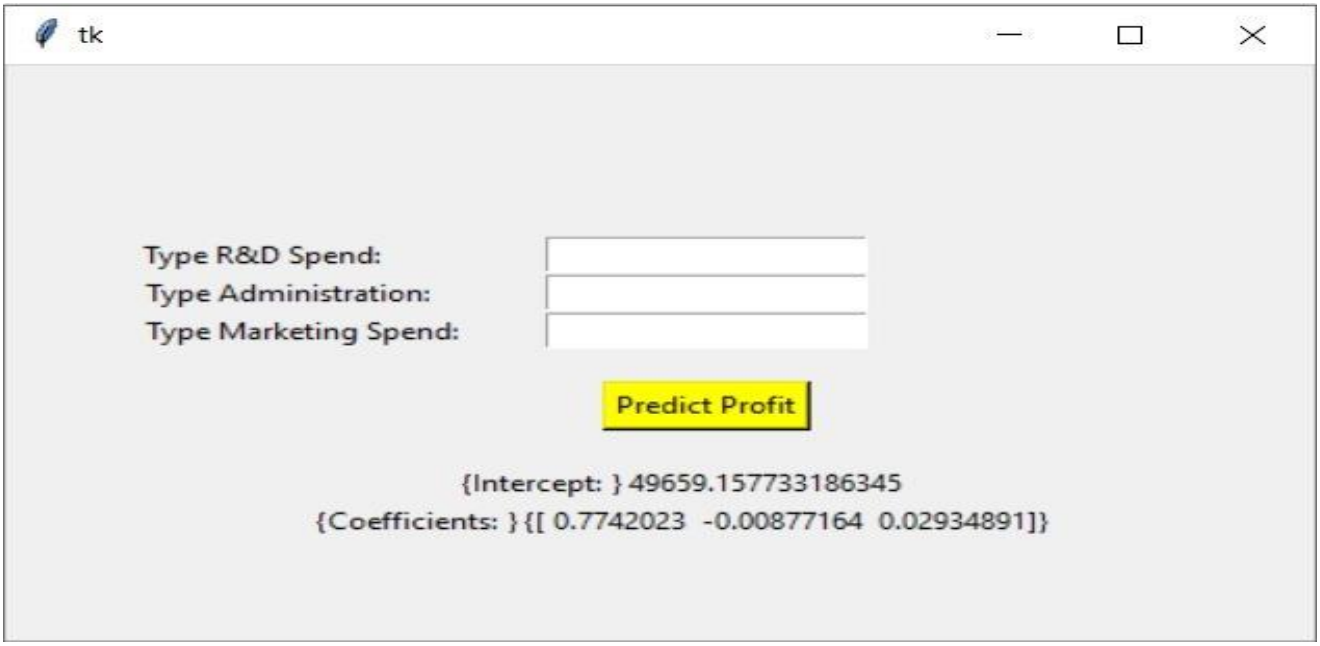
```
Canvas1.create_window (260, 240, window=label_Coefficients)

label1 = tk.Label(root, text='Type R&D Spend: ')

canvas1.create_window (100, 100, window=label1)

entry1 = tk.Entry (root)

canvas1.create_window (270, 100, window=entry1)

label2 = tk.Label(root, text='Type Administration: ')

canvas1.create_window (110, 120, window=label2)

entry2 = tk.Entry (root)

canvas1.create_window (270, 120, window=entry2)

label3 = tk.Label(root, text='Type Marketing Spend: ')

canvas1.create_window (116, 140, window=label3)

entry3 = tk.Entry (root)

canvas1.create_window (270, 140, window=entry3)

def values ():

    global New_RD_spend

    New_RD_spend = float (entry1.get ())

    global New_Administration

    New_Administration = float (entry2.get ())

    global New_Marketing_Spend

    New_Marketing_Spend = float (entry3.get ())
```

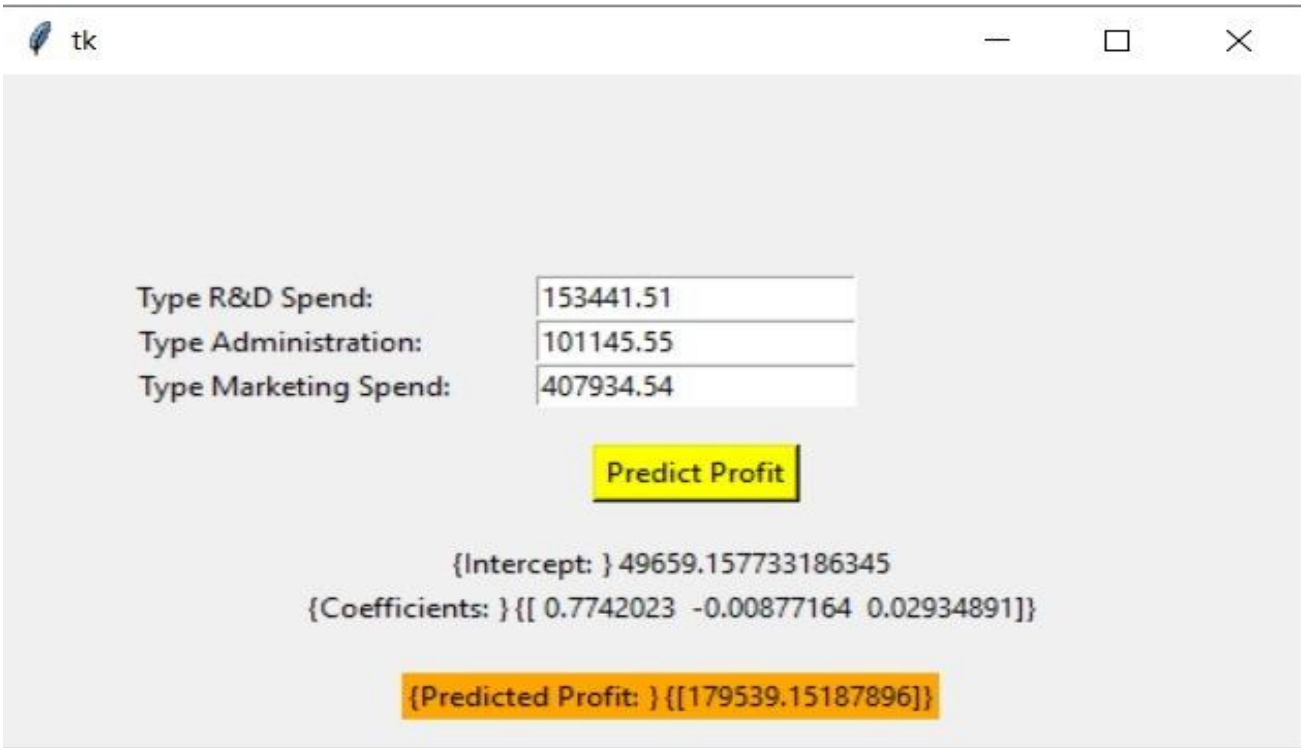
```
    Prediction_result = ('Predicted Profit: ', mlr.predict(([[New_RD_spend
,New_Administration,New_Marketing_Spend]])))

    label_Prediction = tk.Label(root, text= Prediction_result, bg='orange')

    canvas1.create_window (260, 280, window=label_Prediction)

button1 = tk.Button (root, text='Predict Profit', command=values, bg='yellow')

canvas1.create_window (270, 180, window=button1)

root.mainloop()

export=root.mainloop()
```

6. RESULTS

```
In [39]:  score=r2_score(y_test,y_pred)*100
          print('score',score)

          score 96.49618042059875
```

fig – r^2 score





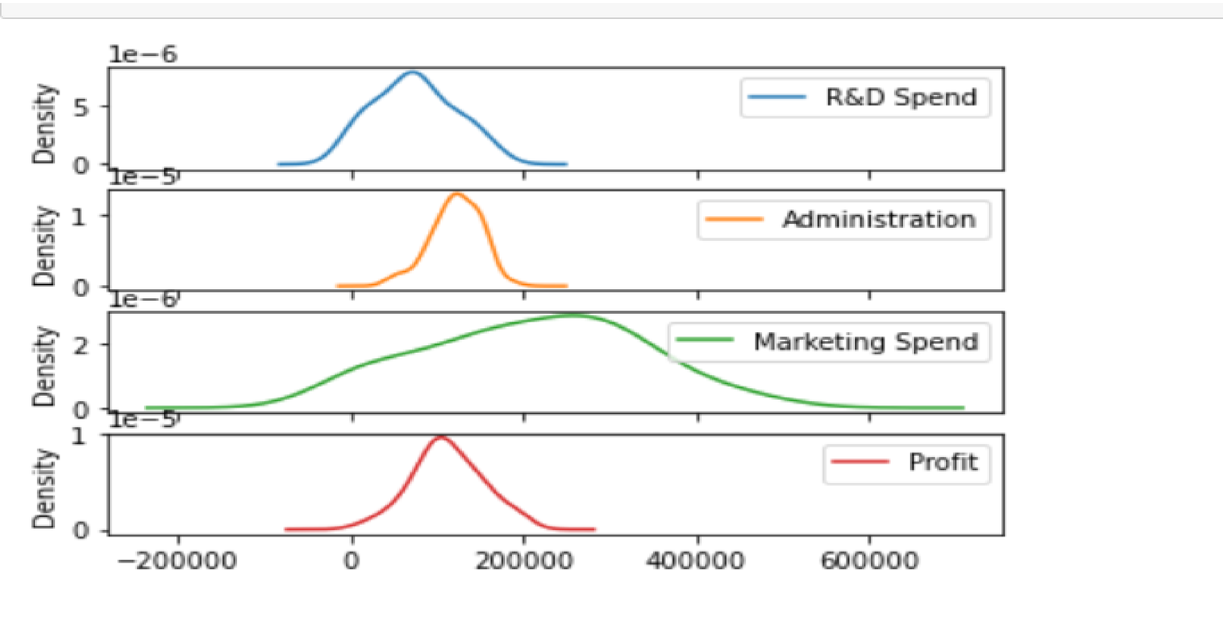


fig - Density plot

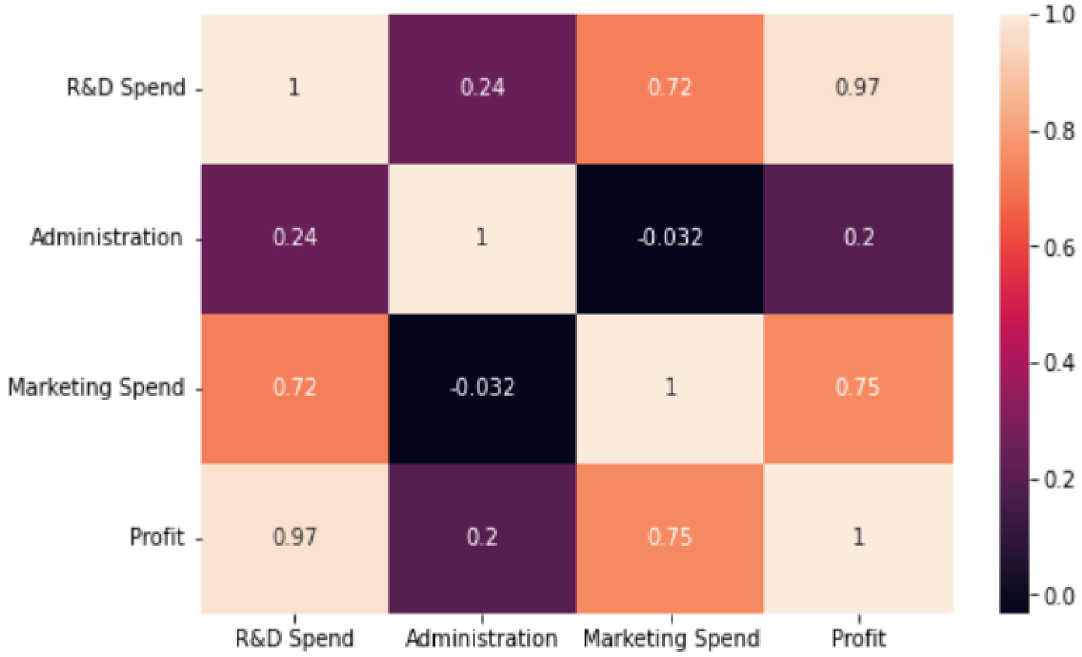


Fig – Correlation heat ma

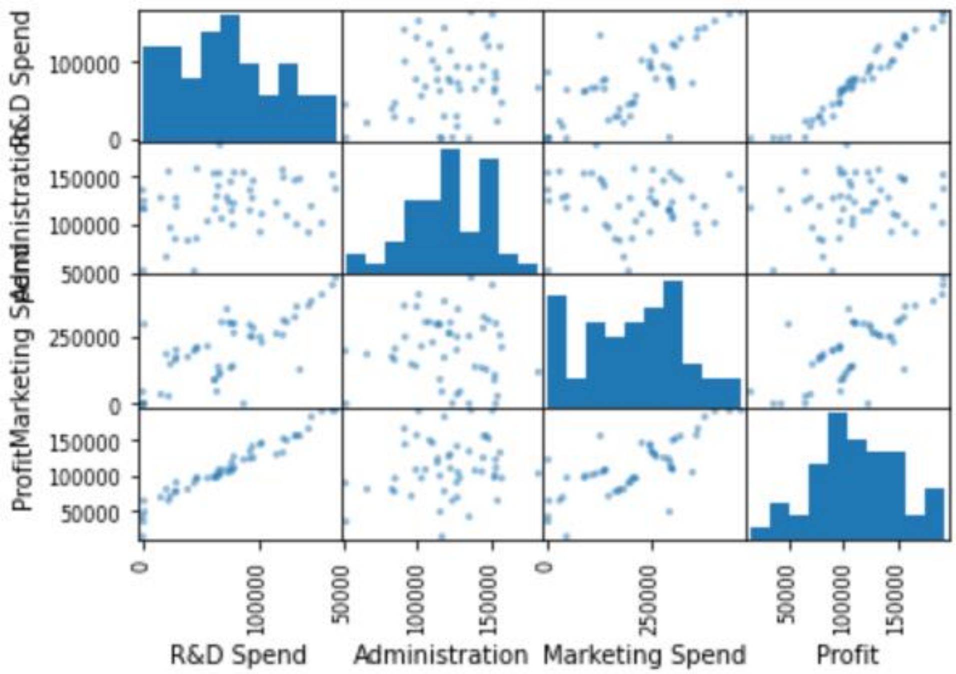


Fig – Scatter plot

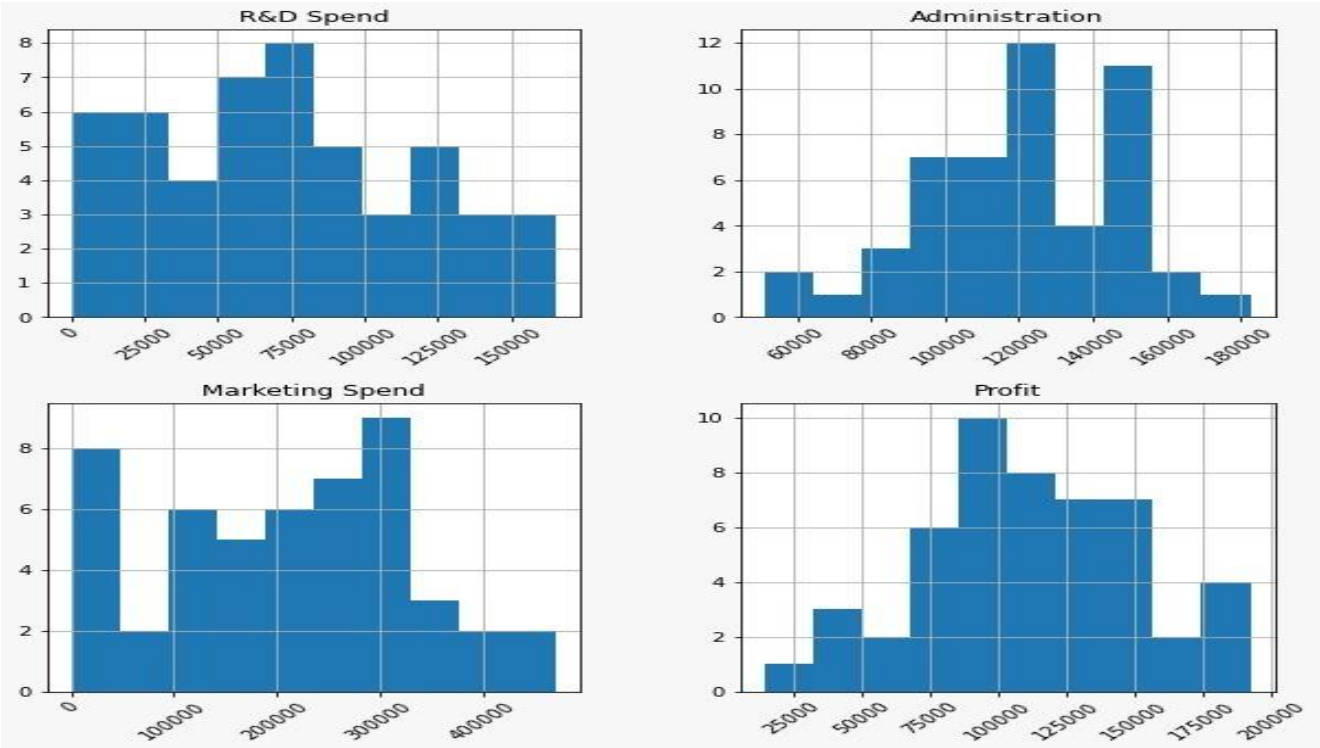


Fig - Histogram

7. CONCLUSION

In this project, we study the system for start-up predication of businesses to get the idea of profits so that any user who wants to know about business profits will use this application for their use. The users who have thought on investing and according to that they would invest money for the start-ups. Our goal is to assist start-ups who are looking for some investment and profit ideas so that they can make a suitable strategy for the future and know how much money their company will make. To forecast start-up profit, we use Multiple Linear Regression. After a period of qualitative study that includes getting to know the start-ups, start-up investors who typically do their own financial examination of possible target companies. Multiple Linear Regression can provide real-time exit predictions as well as a feature analysis that identifies features of a start-up that makes it a smart investment on the other hand, it could raise red flags.

From the dataset we got from Kaggle.com and using the machine learning algorithm (Multiple linear regression) we got an efficiency of 96% which proves that this model is a good model.

8. REFERENCES

Aggarwal, C. C. (2015). Data Mining: The Textbook. Springer International Publishing. http://doi.org/10.1007/978-3-319-14142-8

Artificial Intelligence and Machine Learning: Top 100 Influencers and Brands. (2016). Retrieved January 31, 2017, from http://www.onalytica.com/blog/posts/artificial-intelligence-machine-learning-top-100- influencers-and-brands/

Breiman, L., Friedman, J., Stone, C., & Olshen, R. (1984). Classification and regression trees.

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data Preprocessing for Supervised Leaning. INTERNATIONAL JOURNAL OF COMPUTER SCIENCE, I.