EXPOSYS DATA LABS

P.M R. Residency
Ground Floor, No-5/3 Sy. No.10/6-1
Doddaballapur Main Road
Yelahanka Bengaluru, Karnataka 560064

Project report on

“STARTUPS PROFIT PREDICTION USING DATA SCIENCE”

A project dissertation submitted in partial fulfilment of the requirement for the award of

INTERNSHIP

By

Sathwik P S

Under the Guidance of

EXPOSYS DATA LABS

Duration of Internship:
1 Month

Start Date:
20-10-2023

End Date:
21-11-2023

ABSTRACT

In the given dataset, R&D Spend, Administration Cost and Marketing Spend of 50 Companies are given along with the profit earned. The target is to prepare an ML model which can predict the profit value of a company if the value of its R&D Spend, Administration Cost and Marketing
Spend are given.

i) Construct Different Regression algorithms

ii) Divide the data into train set and test set

iii) Calculate different regression metrics

iv) Choose the best model

Language: Python

TABLE OF CONTENTS

1. INTRODUCTION

Predicting a startup's likelihood of success and profitability is still a vital endeavour for stakeholders, investors, and entrepreneurs in the ever-changing world of entrepreneurship. A useful tool for this project is the 50-startups dataset, which provides a plethora of data on a range of characteristics that may affect a startup's profitability.

This study intends to build a predictive model for the profit margins of these companies by utilizing the concepts of Multiple Linear Regression to utilise the dataset's broad range of parameters, including R&D investment, marketing expenditures, administration costs, and state-wise location. Through the application of a statistical methodology that investigates the correlations between several independent variables and the dependent variable of profit, this analysis aims to provide detailed understanding of the factors that influence startup success.

The goal is to create a strong model that can estimate and forecast startup earnings based on operational characteristics using this predictive framework. Through an examination of the interactions between these various elements, we hope to provide insightful advice to business strategists, investors, and entrepreneurs who are trying to make judgments in the dynamic startup ecosystem.

The results of this Multiple Linear Regression analysis are intended to clarify the importance of many aspects impacting a startup's financial performance in addition to offering precise profit projections. These kinds of insights might enable stakeholders to better allocate resources, hone business plans, and raise the possibility of long-term financial success in the cutthroat world of startups.

2. EXISTING METHOD

By examining a variety of factors that could have an impact on a startup's success, multiple linear regression predicts the profitability of new ventures. As promising as this technique is, there are several established methodologies in this field that must be acknowledged.

1. **Single Linear Regression:** This method involves predicting a dependent variable (profit in this case) based on one independent variable. While it simplifies the analysis, it may not capture the complexity of multiple factors influencing startup profitability.

2. **Feature Engineering:** Prior to regression analysis, feature engineering involves selecting, transforming, or creating new features from the available dataset. Techniques like normalization, scaling, or polynomial transformations can enhance the predictive power of the model.

3. **Regularization Techniques:** Approaches like Ridge Regression or Lasso Regression help prevent over fitting in the Multiple Linear Regression model. They introduce penalty terms to the regression equation, reducing the impact of less relevant variables and improving model generalization.

4. **Cross-Validation:** This technique involves splitting the dataset into training and validation sets, allowing for better model evaluation. Techniques like k-fold cross-validation help ensure the model's predictive accuracy and generalizability.

5. **Model Evaluation Metrics:** R-squared, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Adjusted R-squared are common metrics used to evaluate the goodness-of-fit and predictive accuracy of the Multiple Linear Regression model.

6. **Variable Selection Methods:** Techniques like stepwise regression or forward/backward selection help in selecting the most relevant variables for the model, improving its efficiency and interpretability.

These existing methodologies form a foundation for predicting startup profitability using Multiple Linear Regression. Incorporating these techniques and leveraging the 50-startups dataset can lead to a more robust, accurate, and insightful predictive model for estimating the profit margins of startups.

3. PROPOSED METHOD WITH ARCHITECTURE

Designing an architecture for predicting startup profitability using Multiple Linear Regression involves several steps, from data preprocessing to model evaluation. Here's a proposed method and architecture for this task:

Proposed Method:

1. **Data Collection and Preprocessing:**
   - Obtain the 50-startups dataset containing features such as R&D spending, marketing expenditures, administration costs, and state-wise location.
   - Check for missing values, perform data cleaning, and handle categorical variables (if any) through encoding techniques like one-hot encoding.
   - Split the dataset into training and testing sets.
2. **Feature Engineering:**
   - Normalize or scale numerical features to ensure they're on similar scales, preventing dominance by larger magnitude features.
   - Consider feature transformation techniques like polynomial features or interaction terms to capture non-linear relationships between predictors and the target variable.
3. **Model Development:**
   - Implement Multiple Linear Regression using libraries like scikit-learn or stats models in Python or an equivalent in other programming languages.
   - Train the model using the training dataset, fitting the regression equation to predict the profit based on the selected features.
4. **Model Evaluation:**
   - Validate the model using the testing dataset to assess its predictive performance.

- Utilize evaluation metrics such as R-squared, Mean Squared Error

Architecture:

1. **Data Collection Module:**
   - Interface to collect and import the 50-startups dataset or any relevant startup data for analysis.
2. **Preprocessing Module:**

- Handles data cleaning, missing value imputation, encoding categorical variables, and splitting the dataset into training and testing subsets.

3. **Feature Engineering Module:**
   - Performs feature scaling, transformation, and selection to prepare the dataset for model training.
4. **Multiple Linear Regression Model Module:**
   - Implements the Multiple Linear Regression algorithm using appropriate libraries and packages.
5. **Model Evaluation Module:**
   - Evaluates the trained model using testing data and various evaluation metrics to assess its accuracy and performance.
6. **Visualization and Reporting Module:**
   - Visualizes insights, such as feature importance, model performance metrics, and residual plots.
   - Generates reports summarizing the model's predictive capabilities and insights gleaned from the analysis.
7. **Deployment/Integration Module:**
   - Provides the option to deploy the trained model for real-time predictions or integrate it into existing systems for profitability forecasting.

4. SYSTEM REQUIREMENTS

Hardware Requirements:

- Computer/Server: You'll need a computer or server to run the machine learning models and host the system.
- Processor (CPU): Core i3/i5/i7
- Memory (RAM): minimum 8GB of RAM
- Storage:. Solid State Drives (SSDs) are preferable

Software Requirements:

- Operating System: Windows, macOS, or Linux.
- Python: Python 3.x, with Python 3.7 or higher recommended.
- Python Libraries:
    1. NumPy: For numerical operations
    2. pandas: For data manipulation
    3. scikit-learn: For machine learning algorithms
    4. streamlit: Streamlit is an open-source Python library for building web applications.
- Integrated Development Environment (IDE): PyCharm, Visual Studio Code, Jupyter Notebook, or Spyder.

5. METHODOLOGY

1. **Data Collection and Understanding:**

    - Acquire the 50-startups dataset containing various features like R&D spending, marketing expenditures, administration costs, and state-wise location.
    - Understand the nature of the data, features, and the target variable (profit) by conducting exploratory data analysis (EDA).

2. **Data Preprocessing:**

    - Handle missing values, outliers, and perform necessary data cleaning procedures.
    - Encode categorical variables using techniques like one-hot encoding for state-wise locations.
    - Split the dataset into training and testing subsets (e.g., 80-20 split or using cross-validation).

3. **Feature Engineering:**

    - Normalize or scale numerical features to ensure they're on the same scale, preventing biases in model training due to feature magnitudes.
    - Explore feature transformation techniques like polynomial features or interaction terms to capture non-linear relationships between predictors and profit.

4. **Model Development:**

    - Utilize libraries like scikit-learn, statsmodels, or similar tools in Python to implement Multiple Linear Regression.
    - Fit the model on the training dataset using the selected features to predict the profit for startups.

5. **Model Evaluation:**

    - Evaluate the trained model using the testing dataset to assess its predictive performance.
    - Calculate evaluation metrics such as R-squared, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Adjusted R- squared to gauge the model's accuracy and goodness-of-fit.
    - Conduct residual analysis to validate assumptions and ensure model adequacy

6. Fine-tuning and Validation:

- Fine-tune the model by considering regularization techniques like Ridge Regression or Lasso Regression to prevent overfitting.
- Perform cross-validation to validate the model's stability and generalization on different subsets of the data.

7. Interpretation and Insights:

- Interpret the coefficients of the features in the Multiple Linear Regression equation to understand their impact on startup profitability.
- Derive actionable insights by analysing the importance of different factors influencing profit.

8. Model Deployment or Utilization:

- Optionally,
- deploy the trained model for real-time profit predictions for new startup data or integrate it into existing systems for forecasting purposes.

9. Documentation and Reporting:

- Document the entire methodology, including preprocessing steps, model development, and evaluation procedures.
- Prepare a comprehensive report summarizing the findings, model performance, insights gained, and recommendations based on the analysis.

6. IMPLEMENTATION

Snippet code:

```
import pandas as pd
import numpy as np
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
data=pd.read_csv("/sathwik/mini_projects/2.CompanyProfitPrediction/dataset/50_Startups.csv")
print(data.head())
data.isnull().sum()
X=data.iloc[:,:-1]
Y=data.Profit
X.head()
Y.head()

# Find relation between profit through histogram
sns.histplot(data.Profit)
sns.pairplot(data)

## Preparing test and train data
from sklearn.model_selection import train_test_split
x1,x2,y1,y2=train_test_split(X,Y,test_size=0.2,random_state=42)
from sklearn.linear_model import LinearRegression
model_linear=LinearRegression()
model_linear.fit(x1,y1)

# Accuracy of Linear Regression Model
accu=model_linear.score(x2,y2)*100
print("{:.2f}%".format(accu))
new_arr=[170450,140900,850000]
new_arr=np.array(new_arr).reshape(1,-1)
profit=model_linear.predict(new_arr)[0]
print("Profit would be:{:.2f}/-".format(profit))

# Using Random Forest for same prediction
data=pd.read_csv('datasets/50_Startups.csv')
print(data.head())

## Define function to reshape inputs to pass to prediction model
def reshape(arr):
    arr=np.array(arr).reshape(1,-1)
    return arr
data.head()

## Split columns which are input & output
from sklearn.model_selection import train_test_split
X=data.drop(['Profit'],axis=1)
y=data['Profit']
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)

## Join the data to perform any manipulations further
This acts as duplicate of 'data' DataFrame
```

```
train_data = X_train.join(y_train)
train_data.head()

# Draw Histogram to figure out the data
train_data.hist(figsize=(15,8))

# Heat map to find correlation
import matplotlib.pyplot as plt
plt.figure(figsize=(15,8))
sns.heatmap(train_data.corr(),annot=True,cmap="YlGnBu")
train_data.head()

## Transform the train_data to have logarithmic bell curve
train_data['R&D Spend']=np.log(train_data['R&D Spend']+1)
train_data['Administration']=np.log(train_data['Administration']+1)
train_data['Marketing Spend']=np.log(train_data['Marketing Spend']+1)
train_data['Profit']=np.log(train_data['Profit']+1)
train_data.hist(figsize=(15,8))

## Train the Random Forest Regerssor
from sklearn.ensemble import RandomForestRegressor
forest=RandomForestRegressor()
forest.fit(X_train,y_train)

# Find Accuracy of the model
forest.score(X_test,y_test)
vals=[250000,350000,150000]
vals=reshape(vals)
forest.predict(vals)[0]

## Further more imporve the Random Forest by tweaking properties and
training again
from sklearn.model_selection import GridSearchCV
param_grid={
    'n_estimators':[30,50,100],
    'max_features':[8,12,20],
    'min_samples_split':[2,4,6,8]
}

grid_search=GridSearchCV(forest,param_grid,cv=5,
                        scoring='neg_mean_squared_error',
                        return_train_score=True)
grid_search.fit(X_train,y_train)
best_forest=grid_search.best_estimator_

## Calculate Accuracy of imporved random forest
best_forest.score(X_test,y_test)
linear_accuracy=model_linear.score(X_test,y_test)*100
forest_accuracy=forest.score(X_test,y_test)*100
best_forest_accuracy=best_forest.score(X_test,y_test)*100

print("Linear Regression Accuracy is:{:.2f}%".format(linear_accuracy))
print("Simple random forest Accuracy is:{:.2f}%".format(forest_accuracy))
print("Optimized random forest Accuracy
is:{:.2f}%".format(best_forest_accuracy))
```

```
# Linear Regression is most accurate, therefore will use that
1. Save all models into disk
import pickle as pkl

linear_name="linear_model.sav"
forest_name='forest_model.sav'
best_forest_name='best_forest_model.sav'

pkl.dump(model_linear,open(linear_name,'bw'))
pkl.dump(forest,open(forest_name,'bw'))
pkl.dump(best_forest,open(best_forest_name,'bw'))

print("Linear model saved as:",linear_name)
print("Basic Forest model saved as:",forest_name)
print("Best Forest model saved as:",best_forest_name)
```

7. CONCLUSION

The machine learning project aims to predict startup profitability using Multiple Linear Regression. By leveraging the 50-startups dataset, the project involves data collection, pre-processing, feature engineering, model development, evaluation, and deployment. The proposed architecture encompasses modules for data collection, pre-processing, feature engineering, model development, model evaluation, visualization and reporting, and deployment/integration. The system requirements include hardware specifications and software dependencies such as Python, NumPy, pandas, scikit-learn, and streamlit. The methodology involves data collection and understanding, data preprocessing, feature engineering, model development, model evaluation, fine-tuning and validation, interpretation and insights, model deployment or utilization, and documentation and reporting. The project's objective is to offer valuable guidance to entrepreneurs, investors, and business strategists by providing accurate profit predictions and shedding light on the factors influencing startup financial performance

The project's proposed method and architecture, along with the systematic methodology, provide a comprehensive framework for predicting startup profitability using Multiple Linear Regression. By following these steps, stakeholders can gain valuable insights and make informed decisions in the dynamic landscape of entrepreneurship.

- **Accuracy Scores:**
  - Simple random forest Accuracy is:88.25%
  - Optimized random forest Accuracy is:88.32%
  - LinearRegression: 90.41%

  Linear Regression is most accurate. Hence this model can be considered for the prediction.