Startup Profit Prediction using Linear Regression

Introduction

Startups invest in various domains like R&D, marketing, and administration to generate profits. However, predicting the profitability of a startup based on these investments can be complex. Machine learning models, particularly regression models, can help in forecasting startup profits based on historical data and investment patterns.

Problem Statement

Given a dataset of 50 startups with various investment amounts and their profits, predict the future profit of a startup based on its expenditures across departments (R&D Spend, Administration, Marketing Spend) and its operational state.

Objectives

- To preprocess the dataset efficiently for building a predictive model.
- To apply Linear Regression for predicting profits.
- To evaluate the model's performance using appropriate metrics.
- To visualize and interpret the relationship between actual and predicted profits.

Implementation

Model Used:

- Linear Regression

Preprocessing:

- OneHotEncoding for categorical feature ('State')

Pipeline:

- Built using Scikit-learn's Pipeline and ColumnTransformer

Train-Test Split:

- 80% training, 20% testing (random_state = 42)

Key Libraries:

- pandas

- numpy
- matplotlib
- seaborn
- scikit-learn

Code Snippets:

python

```
# Load the dataset
import pandas as pd
df = pd.read_csv("50_Startups.csv")

# Feature and target selection
X = df.drop("Profit", axis=1)
y = df["Profit"]

# Preprocessing
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
preprocessor = ColumnTransformer(transformers=[('cat', OneHotEncoder(drop='first'), ['State'])
],
   remainder='passthrough'
)

# Model pipeline
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LinearRegression
model = Pipeline(steps=[('preprocessor', preprocessor),('regressor', LinearRegression())])

# Train-test split
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=42)

# Model training
model.fit(X_train, y_train)

# Prediction
y_pred = model.predict(X_test)

# Evaluation
from sklearn.metrics import mean_squared_error, r2_score
print("MSE:", mean_squared_error(y_test, y_pred))
print("R2 Score:", r2_score(y_test, y_pred))
```

Results

- **Mean Squared Error (MSE):** 9,133,600.5
- **R^2 Score:** 0.91

The model explains **91%** of the variance in the dataset, showing strong predictive capability.

Challenges

- Handling categorical variables without introducing multicollinearity.
- Limited dataset size (only 50 records).
- Integrating preprocessing and modeling steps into a single clean pipeline.

Future Scope

- Expand dataset size for better generalization.
- Add additional features like number of employees, year of establishment, etc.
- Experiment with more advanced models such as Ridge Regression, Lasso Regression, and Decision Trees.
- Deploy the trained model through a web application for real-time predictions.

Conclusion

The Linear Regression model successfully predicted startup profits based on investments in different departments. While the model demonstrated strong performance on the available dataset, expanding the dataset and exploring advanced techniques can further enhance predictive power and real-world applicability.

References

- Scikit-learn Documentation: https://scikit-learn.org/stable/
- Pandas Documentation: https://pandas.pydata.org/
- Dataset Source: Kaggle - 50_Startups Dataset