Name: Micah Wagner, Dataset: https://www.kaggle.com/datasets/pulkit21aug/pyramid-scheme-profit-or-loss

In [5]:

```
import numpy as np
import pandas as pd
from sklearn.pipeline import Pipeline
from sklearn.base import TransformerMixin, BaseEstimator
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.compose import TransformedTargetRegressor
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import r2_score
```

In [6]:

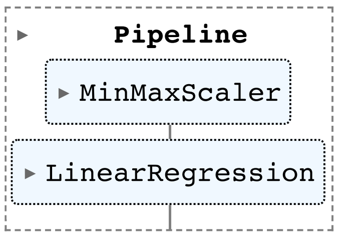
```
data = pd.read_csv("pyramid_scheme.csv")
my_data = data.drop(columns = ['cost_price', 'sales_commission'])
my_data.fillna(value=0, inplace = True)

xs = my_data.drop(columns = ["profit"])
ys = my_data["profit"]

train_x, test_x, train_y, test_y = train_test_split( xs, ys, train_size = 0.7)
print(train_x, train_y, test_x, test_y)
```

```
     Unnamed: 0  profit_markup  depth_of_tree
143         144              3             21
173         174              2             16
241         242              3              3
15           16              5             16
130         131              4             26
..          ...            ...            ...
328         329              5             10
286         287              5             27
358         359              4             26
435         436              3             13
430         431              3             16

[350 rows x 3 columns] 143   -13000
173   -11500
241     5000
15     -1000
130   -14500
       ...
328     5000
286   -12000
358   -14500
435    -5000
430    -8000
Name: profit, Length: 350, dtype: int64      Unnamed: 0  profit_markup  depth_of_tree
338         339              5             25
140         141              4             14
224         225              5              9
162         163              4             21
370         371              5             15
..          ...            ...            ...
113         114              5             11
306         307              3             14
30           31              4             30
121         122              3             15
466         467              5             23

[150 rows x 3 columns] 338   -10000
140    -2500
224     6000
162    -9500
370        0
       ...
113     4000
306    -6000
30    -18500
121    -7000
466    -8000
Name: profit, Length: 150, dtype: int64
```

In [7]:

```
steps = [
    ('scale', MinMaxScaler()),
    ('predict', LinearRegression(n_jobs = -1)),
]

pipe = Pipeline(steps)
pipe.fit(train_x, train_y)
```

Out[7]:



▸ Pipeline

▸ MinMaxScaler

▸ LinearRegression

In [8]:

```
predicted_ys = pipe.predict(test_x)
r2_score(test_y, predicted_ys)
```

Out[8]:

```
1.0
```

1. The reason I dropped the columns cost_price and sales_commission was because those two columns were all the same value, so there was no reason to train with that data. The reason I chose the columns profit_markup and depth_of_tree was because those were the only columns of useful information to predict the profit of the pyramid scheme. For instance, depth tree refers to how many levels of recruitment there are in the pyramid scheme, and this is vital to knowing the profit of the scheme since the returns are pormised to the investors from the captial of new investors. Additionally, profit_markup refers to the total profit from selling an item, including its cost of manufacturing. I thought this information was important to predict the profitability of a pyramid scheme since they all relate to the amount of capital in the scheme.
2. My model performed extremely well according to my metric. In fact, it preformed perfectly, which makes me think that my data is not a good dataset. Upon examining the dataset, by graphing the profit_markup on the x-axis, and the depth_of_tree on the y-axis, and the profit on the z-axis in excel, this resulted in what looked like a perfect plane. So what my model figured out was the plane equation to describe the data perfectly. I then took three points and calculated the plane equation using various linear algebra concepts, and arived at the following equation. profit = 3500*profit_markup - 1000*depth_of_tree - 2500. Plugging in values from the dataset demonstrates the validity of the equation.
3. Since my pipeline is perfoming a regression task, It would make sense to use R^2 because this metric is used to evaluate the performance of regression models by measuring how well they explain the variation in the target variable (1 meaning that the model perfectly explains the variation, and 0 meaning the model doesn't explain the variation at all).