Final Project

This is my final project. I had hoped to do a custom clustering to find types of less successful developers on top of directly answering my questions, but didn't get to it in time.

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
# https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html
# Various parts of the pandas api reference have been used, will not be citing every
raw_data = pd.read_csv('survey_results_public.csv', index_col="ResponseId")
raw_data.shape
```

Out[2]: (65437, 113)

Analysis of developers who didn't learn to code in college

In [3]:
```
# Only using columns that I find relevant for the question I want to ask
data = raw_data.filter(items=[
    "MainBranch",
    "Employment",
    "EdLevel",
    "LearnCode",
    "Country",
    "ConvertedCompYearly",
    "JobSat"
])

# Filtering results for full time, employed developers in the US
data["Country"] = data["Country"] == "United States of America"
data["MainBranch"] = data["MainBranch"] == "I am a developer by profession"
data["Employment"] = data["Employment"] == "Employed, full-time"
data["LearnCode"] = data["LearnCode"].str.contains("School")

# Removing NA values
data.dropna(inplace=True, subset=[
    "MainBranch",
    "Employment",
    "EdLevel",
    "LearnCode",
    "Country",
    "ConvertedCompYearly",
    "JobSat"
])

# Dollar amount is poverty line for a 4 person family
data["Successful"] = (data["JobSat"] > 5) & (data["ConvertedCompYearly"] > 32150)

# https://stackoverflow.com/questions/17071871/how-do-i-select-rows-from-a-dataframe-
data = data.loc[data["Country"] & data["MainBranch"] & data["Employment"]]
data = data.filter(items=[
#     "EdLevel",
    "LearnCode",
#     "JobSat"
    "Successful"
])
```

```
data.dropna(inplace=True)

data.describe()
```

Out[3]:

	LearnCode	Successful
count	2631	2631
unique	2	2
top	True	True
freq	1425	2118

In [4]:
```
data.value_counts()
```

Out[4]:
```
LearnCode  Successful
True       True          1165
False      True           953
True       False          260
False      False          253
Name: count, dtype: int64
```

This data shows that roughly 80% of software developers are considered "successful" by my very limited classification. More importantly, however, this shows that the proportion is roughly the same, regardless of whether you learned to code in college.

This shows that college is not the only path to success. If it doesn't work for you, then you don't have to go to college to be a successful developer.

Analysis of Developers in different industries

In [5]:
```
# Only using columns that I find relevant for the question I want to ask
data = raw_data.filter(items=[
    "MainBranch",
    "Employment",
    "EdLevel",
    "Country",
    "Industry",
    "ConvertedCompYearly",
    "JobSat"
])

# Filtering results for full time, employed developers in the US
data["Country"] = data["Country"] == "United States of America"
data["MainBranch"] = data["MainBranch"] == "I am a developer by profession"
data["Employment"] = data["Employment"] == "Employed, full-time"

# Removing NA values
data.dropna(inplace=True, subset=[
    "MainBranch",
    "Employment",
    "EdLevel",
    "Country",
    "Industry",
    "ConvertedCompYearly",
    "JobSat"
])

# Dollar amount is poverty line for a 4 person family
data["Successful"] = (data["JobSat"] > 5) & (data["ConvertedCompYearly"] > 32150)
```

```
# https://stackoverflow.com/questions/17071871/how-do-i-select-rows-from-a-dataframe-
data = data.loc[data["Country"] & data["MainBranch"] & data["Employment"]]

data = data.filter(items=[
    "Industry",
    "Successful"
])

# https://stackoverflow.com/questions/32387266/converting-categorical-values-to-binar
data = pd.get_dummies(data)

data.dropna(inplace=True)

data.describe()
```

Out[5]:

	Successful	Industry_Banking/Financial Services	Industry_Computer Systems Design and Services	Industry_Energy	Industry_Fir
count	2618	2618	2618	2618	
unique	2	2	2	2	
top	True	False	False	False	
freq	2107	2498	2539	2570	

In [6]:
```
# https://stackoverflow.com/questions/29763620/how-to-select-all-columns-except-one-i
x = data.loc[:, data.columns != 'Successful'].to_numpy()
y = data["Successful"].to_numpy()

sample_count, feature_count = x.shape

weights = np.zeros(feature_count)
bias = 1
lr = 0.001
epochs = 1000

for _ in range(epochs):
    y_pred = np.dot(x, weights) + bias

    dw = (1 / sample_count) * np.dot(x.T, (y_pred - y))
    db = (1 / sample_count) * np.sum(y_pred - y)

    weights -= lr * dw
    bias -= lr * db
```

In [7]:
```
def print_linreg_equation(intercept: float, weights: list[float]):
    print(f"Y = {intercept:.2f}", end='')
    for i, w in enumerate(weights):
        print(f" {'+' if w > 0 else '-'} {w if w > 0 else -w:.2f}*X{i + 1}", end='')
    print()

print_linreg_equation(bias, weights)
```

```
Y = 0.88 - 0.01*X1 - 0.00*X2 - 0.00*X3 - 0.01*X4 - 0.01*X5 - 0.01*X6 - 0.00*X7 - 0.00*
X8 - 0.01*X9 - 0.01*X10 - 0.00*X11 - 0.02*X12 - 0.01*X13 - 0.03*X14 - 0.01*X15
```

In [8]:
```
def model_eval(pred, y):
    rss = np.sum((pred - y)**2)
    rse = (rss/(len(y) - 2))**(1/2)
    tss = np.sum((y - np.mean(y))**2)
    r2 = 1 - rss/tss
    return rss, rse, tss, r2
```

```
_, rse, _, r2 = model_eval(y_pred, y)
print("RSE:", rse)
print("R^2:", r2)
```

```
RSE: 0.4019525560430364
R^2: -0.02771225554692136
```

Looking at the R-squared value here, we can obviously see that the industry you choose to work in has effectively no correlation with your level of success in life. So restricting yourself to specific industries because they are more prestigious or because you think you will be more successful is, in reality, only restricting your possible job opportunities.

(I know that a linear model is not ideal for this kind of data, but I just wanted to establish a lack of relation between the inputs and outputs. I don't need to make any predictions or do anything complicated with this model.)

Analysis of developers in management

In [9]:

```
# Only using columns that I find relevant for the question I want to ask
data = raw_data.filter(items=[
    "MainBranch",
    "Employment",
    "EdLevel",
    "ICorPM",
    "Country",
    "ConvertedCompYearly",
    "JobSat"
])

# Filtering results for full time, employed developers in the US
data["Country"] = data["Country"] == "United States of America"
data["MainBranch"] = data["MainBranch"] == "I am a developer by profession"
data["Employment"] = data["Employment"] == "Employed, full-time"
data["ICorPM"] = data["ICorPM"].str.contains("manager")

# Removing NA values
data.dropna(inplace=True, subset=[
    "MainBranch",
    "Employment",
    "EdLevel",
    "ICorPM",
    "Country",
    "ConvertedCompYearly",
    "JobSat"
])

# Dollar amount is poverty line for a 4 person family
data["Successful"] = (data["JobSat"] > 5) & (data["ConvertedCompYearly"] > 32150)

# https://stackoverflow.com/questions/17071871/how-do-i-select-rows-from-a-dataframe-
data = data.loc[data["Country"] & data["MainBranch"] & data["Employment"]]
data = data.filter(items=[
#    "EdLevel",
    "ICorPM",
#    "JobSat"
    "Successful"
])

data.dropna(inplace=True)

data.describe()
```

Out[9]:

	ICorPM	Successful
count	2631	2631
unique	2	2
top	False	True
freq	2374	2117

In [10]:
```
data.value_counts()
```

Out[10]:
```
ICorPM  Successful
False   True          1902
        False          472
True    True           215
        False           42
Name: count, dtype: int64
```

This once again shows that regardless of whether you are an individual contributor or a people manager, there is still roughly a 80% success ratio for software developers. The ratio for people managers is admittedly slightly higher, at 83.6%, but I'd say that this is within the margin for error with this relatively small number of people managers in this dataset. It could also mean that there is a slight positive association between being a manager and being successful, which would also make sense.