

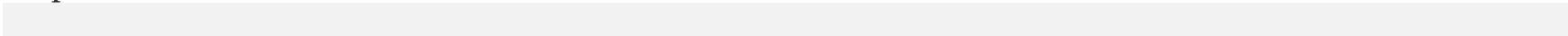
Traditional computer programming has been around for more than a century, with the first known computer program dating back to the mid 1800s. **Traditional Programming** refers to any manually created program that uses input data and runs on a computer to produce the output.

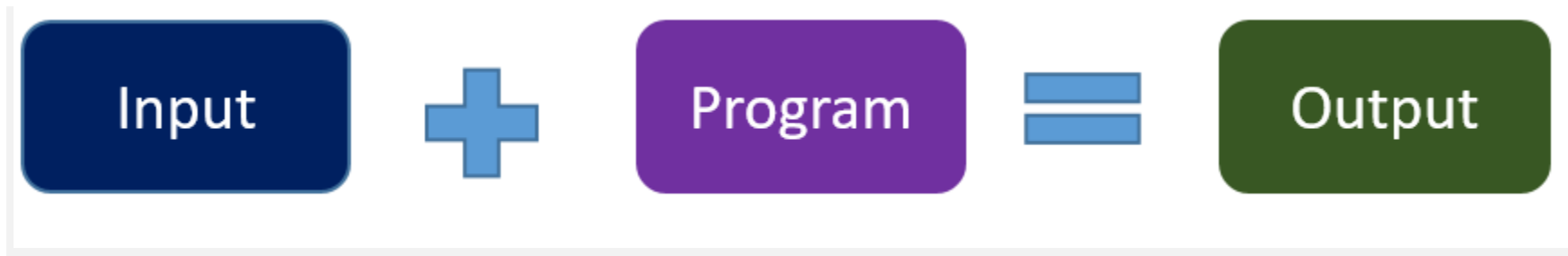
But for decades now, an advanced type of programming has revolutionized business, particularly in the areas of intelligence and embedded analytics. In **Machine Learning**, also known as augmented analytics, the input data and output are fed to an algorithm to create a program. This yields powerful insights that can be used to predict future outcomes.

**Here's a closer comparison of traditional programming versus machine learning that would be useful for a product manager:**

### **Traditional Programming**

Traditional programming is a manual process — meaning a person (programmer) creates the program. But without anyone programming the logic, one has to manually formulate or code rules. We have the input data, and someone (programmer) coded a program that uses that data and runs on a computer to produce the desired output.





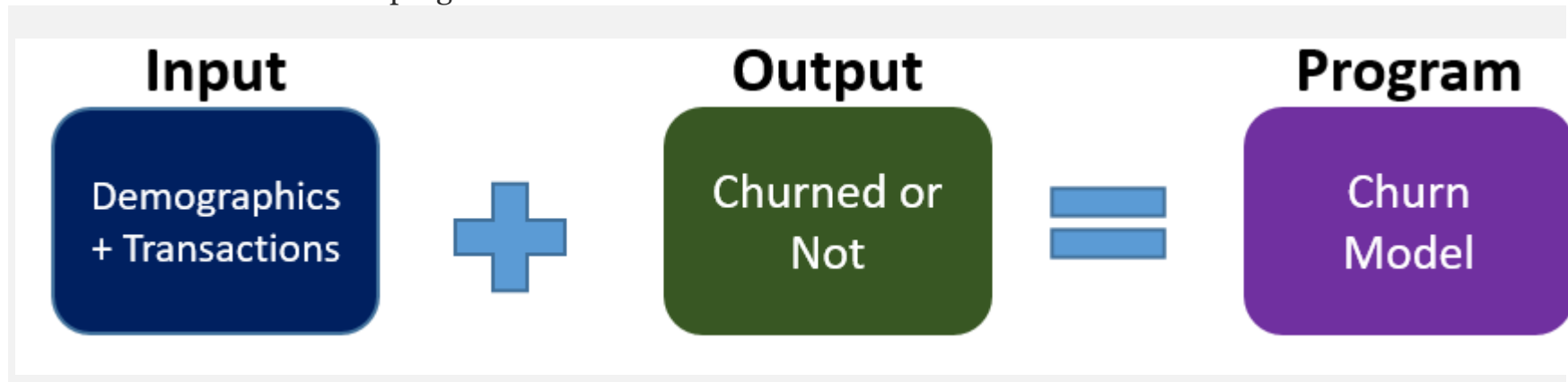
## Machine Learning

Machine Learning, on the other hand, the input data and output are fed to an algorithm to create a program.

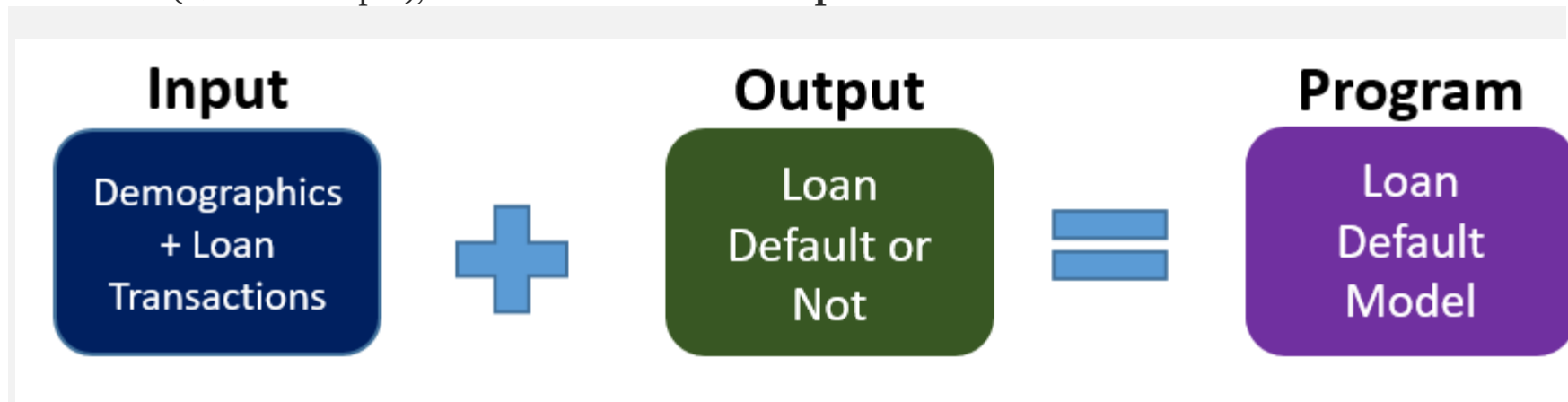


This is the basic difference between traditional programming and machine learning. Without anyone programming the logic, In Traditional programming one has to manually formulate/code rules while in Machine Learning the algorithms automatically formulate the rules from the data, which is very powerful. .

For example, if you feed in customer demographics, transactions as input and the observed output if they churned or not in the past, the algorithm will formulate the program which would know how to predict if someone would **churn or not**. That program is called a model



Customers, in general, have a lot of such input and output data and they can feed that to the algorithms to create predictive models. For example, feed-in customer information/loan transactions (input) and how many defaulted on the loan (observed output), and it will create a model to **predict who will default on the loan**.

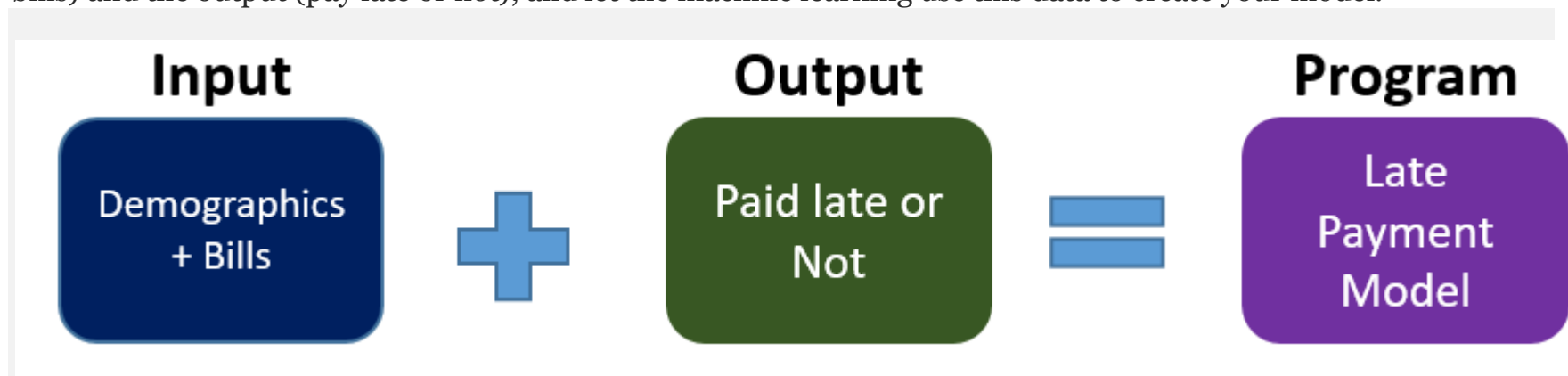


## Simple Framework

A product manager can use this framework to predict business outcomes in any situation where you have input and historical output data:

1. Identify the business question you would like to ask.
2. Identify the historical input.
3. Identify the historically observed output (i.e., data samples for when the condition is true and for when it's false).

For instance, if you want to **predict who will pay the bills late**, identify the input (customer demographics, bills) and the output (pay late or not), and let the machine learning use this data to create your model.



## **Summary**

So the problem a company needs to solve to create a predictive model is to identify data samples for when the condition (churn) is true and when the condition (churn) is false and pass this data to a predictive algorithm to create the model. That is a simplistic way to look at predicting outcomes.

This is what you as a product manager can figure out ways to treat your business data as a financial asset. Formulate the question that has good business value, assemble the data (historical) that has samples of both positive and negative past outcomes/conditions and point the algorithm to your data so it can learn powerful rules that can be used to predict future business outcomes.