iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** X **Month of publication:** October 2022

DOI: https://doi.org/10.22214/ijraset.2022.46990

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

Building a Merging and Acquisition Simulator by Knapsack and Dynamic Algorithm

Tamilselvan Arjunan
Assistant Manager, Data Science and Analytics

Abstract: ***This research paper's goal is to build a merger and acquisition simulator. Essentially, the simulation produces choices that can be compared, evaluated, and used to make decisions. The computer model was created so that users may alter the crucial decision-making factors that are involved in the purchase process.***
The simulator is a transaction analytics tool that proposes stock-level asset or product category swaps between a set of participating companies within an industry that optimize their positions in the market and are also cash efficient in execution. The user can define a set of rules, which set the boundaries of the proposed transactions that are evaluated in the simulation.
The simulation tool is a transaction analytics tool that proposes portfolio-level asset or product category swaps between a set of participating companies within an industry that optimize their positions in the market and are also cash efficient in execution.
Overview: The simulator offers a practical tool for developing and assessing various growth and diversification plans. If this procedure is oversimplified, the actual effects of the merger on the parent firm will become less clear, making both long-term and short-term financial planning very challenging. A model has been devised that makes use of a computer's capacity to quickly handle large amounts of data in order to get around several of these drawbacks. The long-term merger and acquisition computer model that has been created will optimize the impact that an acquisition will have on the parent company's earnings per share while taking into account the dividends received and the market value per share of the acquired firm.
Keywords: M&A simulator, Asset swap simulator, Optimizer

I. INTRODUCTION

The first and most important step is to set the boundaries of the market data on which the simulation algorithm evaluates transactions for the optimization of company portfolios.
The 'user' can define a set of rules, which set the boundaries of the proposed transactions that are evaluated in the simulation it also offers flexibility to override asset swaps proposed from the automated simulation algorithm by allowing the 'user' to define a set of pre-decided set of transactions known as first mover swaps

The major steps in creating a merger model are as follows:

1) *Input Data (Excel)*
- Geography-specific company level sales data by segment and sub-segment(s)
- Other supplementary company-level data such as Market Cap, EBIT, Debt, Cash position

2) *Simulator (Python Based Integrated with Excel)*
- Excel Identifies potential targets of interest for asset swap by taking into account multiple factors such as market concentration (HHI), relative market sha, EBITDA, Debt, Cash

3) *Output (Excel)*
- Final market state data showcasing an optimized portfolio of players
- Improved sales, relative market share, and profitability of participants

There 2 types of algorithms incorporated into the simulator.
- Greedy algorithm(Knapsack problem statement)
- Full optimization (Dynamic Optimization)

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

II. KNAPSACK PROBLEM

The knapsack problem is a combinatorial optimization problem that asks, given a set of objects, each with a weight and a value, how many of each item to include in a collection so that the total weight is as low as feasible or equal to a specified limit and the total value is as high as possible.

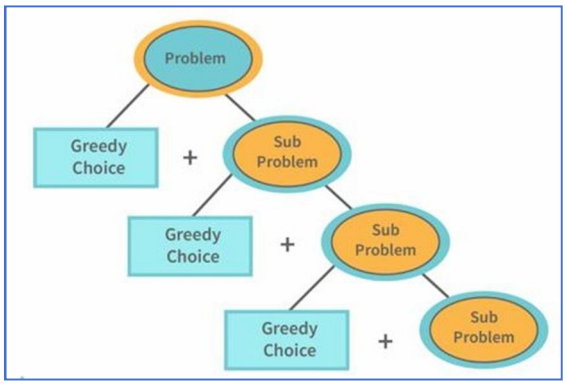


Figure 1. Knapsack problem statement

A Knapsack algorithm chooses the best solution at the moment, in order to ensure a globally optimal solution.
In order to guarantee an overall optimal solution, a greedy algorithm selects the best option at the time. The Knapsack strategy does not ensure that an ideal solution will be found.
A Knapsack approach moves more quickly than a dynamic one. Fast results.

III. DYNAMIC PROGRAMMING

Dynamic programming allows us to reduce time complexity for any recursive form that utilizes the same inputs again. The outcomes of subproblems can be preserved by being recorded. Time complexity is reduced to a polynomial ratio by this minimal optimization. For instance, temporal complexity would be linear if we recorded all the solutions to the Fibonacci summing. Time complexity may be made linear if we documented subproblems. Programmers can solve problems using the influence of variables and other elements by using a technique called dynamic programming, which allows them to do so. A problem should ideally be divided into smaller, more manageable portions before being solved independently. This strategy enables
Dynamic programming is a powerful optimization algorithm that can be used to solve a wide variety of problems. The objective of dynamic programming is to reduce the worst-case time to solve a problem. Keeping track of the data that needs to be processed and just processing it as it becomes available is the simplest approach to accomplish this.

However, this method can be costly and ineffective, therefore it's frequently preferable to employ memorization.

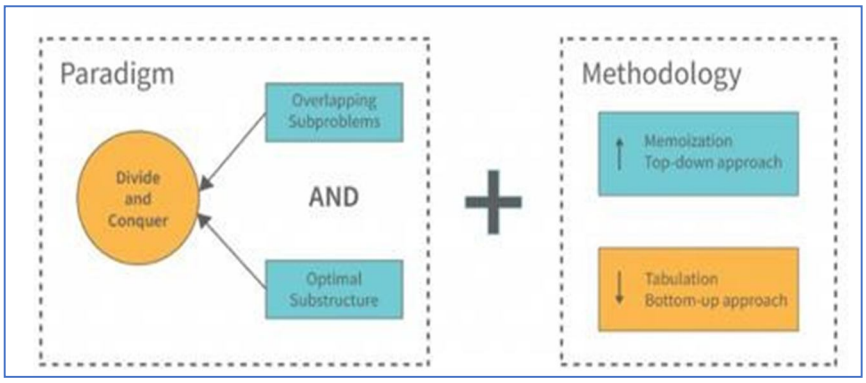
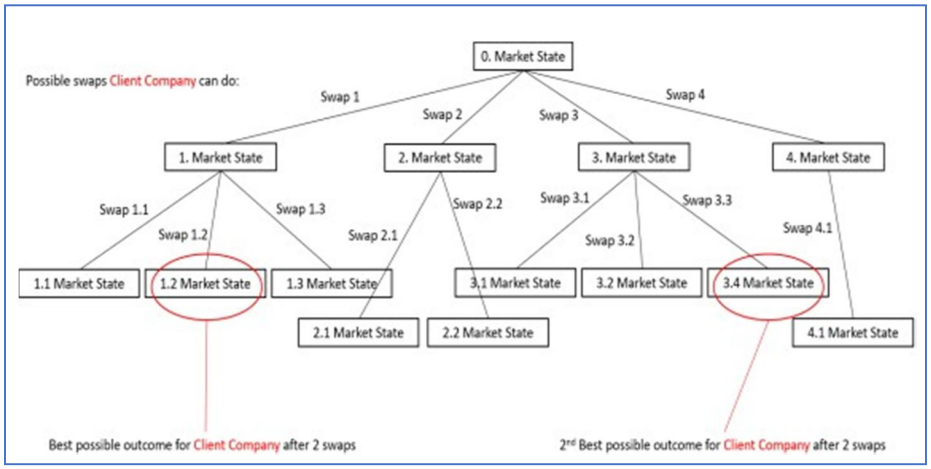


Figure 2. Full optimization

When employing memorization, the algorithm is initially tasked with locating the ideal solution to a particular issue.

IV. VISUALIZATION OF POSSIBILITIES (SWAP)



V. OVERVIEW

A. *Greedy Algorithm*

1) Bid: a request for the asset(s) by one company on their turn to another company
2) Ask: a response request for the asset(s) from the original "bidder" (
3) Tool supports negotiation, settling on swaps within fairness threshold
4) Accepts/rejects proposed swaps based on user-defined threshold parameters
5) "Bidder" exhausts the prioritized list of bids, either finding a successful swap and completing turn or moving onto next company's turn as "bidder"
6) Pros: examining dynamic market where competitors participate, seeing potential threats
7) Cons: not exploring other outcomes, final market state dependent on everyone, not fully optimized for client company (see below)

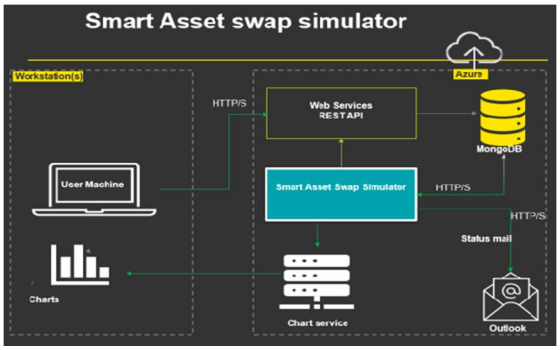
B. *Full Optimization*

1) Client Company: optimize their portfolio
2) Optimization goal set at the beginning
3) Ability to set the maximum number of swaps allowed
4) Adheres to user-defined thresholds for acceptable swaps (i.e. fair values, regulatory rules, etc.)

VI. ARCHITECTURE

A. *Data Feed Source*

Market Data: Market Data Can Be Sourced From Various Sector Or Industry Focused Databases. Financial Data: Financial Data Is Sourced From public Listed companies And The data Parameters Include Revenue, EBIT%, Cash, Debt, Market Capitalization, EV/Revenue, EV/EBITDA Among Others.



ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

VII. ALGORITHMS

We will go through the process of how each algorithm we can apply in asset swapping: First, the greedy algorithm: The companies follow a sequence order to act as bidders and ask for assets from other participating companies to consolidate their core positions. Companies take a turn in the sequence order to act as bidders. This process runs for one or more rounds and attempts to improve the competitive position of each transacting party either in the product category.

	Cereal Bars	Chilled Lunch Kits	Chilled Pizza	Cheese
Item Value(Vi)	5	6	3	7
Item Weight(Wi)	11	15	16	18

Simulations of genuine negotiations involving all market participants.

To serve as a bidder and request assets from other participating corporations in order to consolidate their key positions, the companies move in a predetermined order. Companies act as bidders sequentially one at a time. This bid-ask procedure lasts for one or more rounds and aims to strengthen each transacting party's competitive position, either at the product category market state level.

The purpose of the knapsack problem is to select which items to fit into the bag without exceeding the weight limit of what can be carried.

As shown below inside knapsack algorithm we will keep weights(sales) of bidding of company. For values we can keep responding company value which is what we are trying to optimize . Capacity is sales value. Knapsack algorithm will give asset which can be optimize and increase the benefits both bidding and responding company.

We solve the problem with an integer programming solver by setting up each item as a binary variable (0 or 1). A zero (0) is a decision to not place the item in the knapsack while a one (1) is a decision to include it. More generally, suppose there are nn items with values v1,...,vnv1,...,vn and weights w1,...,wnw1,...,wn with a total weight limit of the sack W=14W=14. The decision variables are x1,...,xnx1,...,xn that can be 0 or 1. The following optimization formulation represents this problem as an integer program:

$$\max n\sum i=1vixi \text{ subject to } n\sum i=1wixi\leq W$$
$$xi\in\{0,1\}$$

An alternative to using optimization is a greedy algorithm where the items are successively selected based on a metric such as the highest value to weight ratio. This is done until the weight limit is exceeded. While this approach is computationally fast and intuitive, it may give suboptimal results because a constraint limit (weight) is not reached. The greedy algorithm is an example of a heuristic (rule-based) approach that is often specific to the application. Heuristics may be valuable to initialize the optimization solution or identify at least one feasible solution that can be improved with optimization.

A. Secondly, Full Optimization Process

There are different types of probability based on below logistic regression equation and then we will find the probability of success or failure.

For example calculate value of cash probability, leverage probability, debt probability, regulatory probability.

If events A and B are independent events, then P(A and B)=P(A)·P(B) .

So swap possibility = P(cash probability) x P(leverage probability) x P(Debt probability) x P(regulatory probability)

Using brute force method examine absolutely everything, definitely find the best result

Time-consuming, but feasible depending on market size can save 2nd, 3rd, 4th, etc. best results as alternatives for clients to consider

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

B. Time-saving Strategies

Exclude unnecessary product categories from consideration limit Full Optimization simulation to include only the largest companies, then after the computer simulation finds best swap results for the client, plug them into the full market simulation as Manual swaps and effect on full market. Apply tighter constraints on swap acceptance if reasonable.

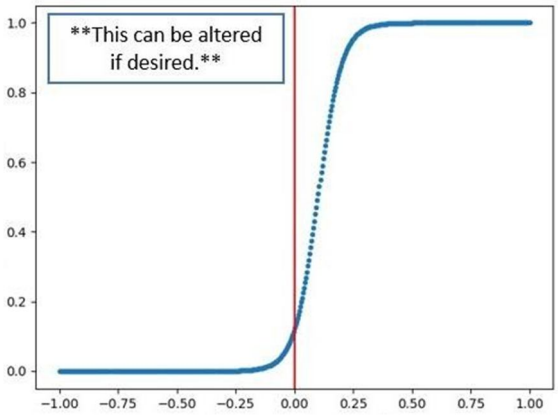
C. Stochastic Approach

Apply probability of swap acceptance based on various factors- i.e. the probability of accepting a swap between Comp 1 and Comp 2 is a combination of the following probabilities:

$$P_{swap} = P_{\Delta comp1} * P_{\Delta comp2} * P_{FavorComp1} * P_{FavorComp2}$$

D. Logistic Regularization

Applying hyperbolic tangent function with range spanning 0 to 1 and gentle slope (could adopt another functional form to follow a specific qualitative business motivation)



We can also apply transactions cost post swapping assets.

There are two methods of defining transaction costs:

Percent of the purchase price (meaning the value of the asset) with both minimum and maximum values A sliding scale where the percent of the purchase price diminishes as the value of the asset increase, with both minimum and maximum values Percentage of purchase price is user defined.

VIII. TYPES OF SWAPS TWO PARTY MULTI SWAPS

These are the exchange of an arbitrary number of assets between two parties.

A. Multi-party Swaps

These are the exchange of an arbitrary number of assets between an arbitrary number of parties.

While the most flexible, this scenario is technically equivalent to a fill industry optimization and is the hardest to explain.

While the algorithm is for n parties, the user defines the max number of parties that can be involved in a single swap. For example, the user can set a maximum of 5 where no more than 5 companies can participate in the swap.

IX. BENEFITS OF MERGING AND ACQUISITION SIMULATOR

The likelihood of merger and acquisition success, or selecting acquisition possibilities that provide the maximum return on investment, is the most crucial factor (ROI).

This simulator is a game-changer for conventional opportunity sourcing when identifying acquisition prospects with possibly value-added synergies based on their overall company objectives.

Simulator gives stakeholders the opportunity to investigate prospective acquisition targets and the effects of deals on strategy and financial performance through machine learning.

To find patterns that are invisible to humans, it is possible to aggregate and analyse data sets from many sources, both correlated and uncorrelated. The target screening process is scalable because by examining these patterns, market insights and trends can be discovered and forecasts can be made much faster than by a human.

Decision-making can be considerably aided by these algorithm-based analyses, and dynamic real- time visuals help decision-makers better understand complex relationships.

Virtual data rooms powered by the cloud have nearly completely supplanted physical ones in recent years, revolutionizing M&A due diligence procedures. This advancement has increased the effectiveness of evaluating and examining crucial presumptions relating to the anticipated expansion of the intended transaction. The data gathered and pooled in VDRs can be mined for even more value using automation, simulator- based analytical tools, and dynamic visualization approaches. This enables the buy- and sell-side to make decisions more quickly and effectively with better advice and insights. At least one of the following three approaches—the cost approach, the market approach, and the income method—is typically employed when valuing a corporation. Alternatively, to increase computation quality, valuation adjustment formulas can be constructed that are specific to the target and the desired criteria. Income-based valuation strategies, on the other hand, are often intrinsic valuation techniques. As an illustration, using the discounted cash flow approach. The predicted future earnings of the company are used to determine the enterprise value, which are then reduced to the present value using a suitable discount rate. As a result, this tool can assist in gathering and extracting discount variables and predictions of cash flows derived from market- based benchmarks that take the company's risk into account. On-demand scenario analysis and multi-variable sensitivity are made possible by its robust data processing capabilities. Analytics can provide better insights, improving the customer experience through quicker responsiveness, click-through capabilities, among others. All of this contributes to the valuation process's clarification and offers information for making wise strategic business decisions.

X. CONCLUSION

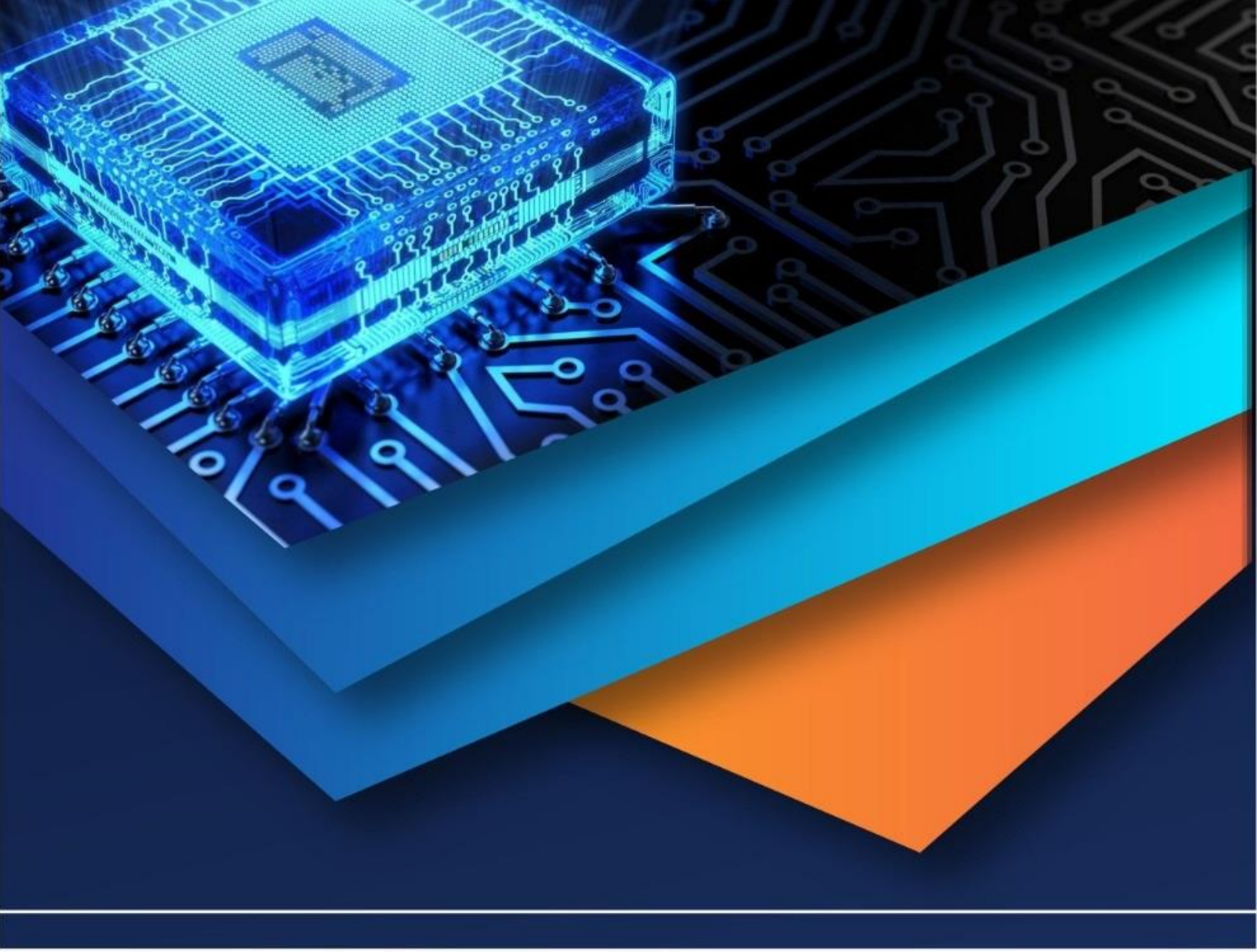
The use of data analytics and AI-based activities in the M&A process is projected to grow in the future since they improve the value generation, effectiveness, insights, and decision-making of M&A transactions across all industries. The objective of the dynamic programming approach is to reduce the number of steps necessary to accomplish a specific goal. Finding a technique to divide an issue into smaller components and then solving each component separately is the basic principle. The issue is then resolved as effectively as possible. The greedy strategy, on the other hand, focuses on increasing the total amount of labour that can be accomplished. When there is not enough knowledge to make a choice, this strategy is frequently used. The objective is to reduce the number of steps needed. Dynamic programming is a potent tool that can be applied in a variety of contexts. Problems that are challenging to solve can be handled using it.

REFERENCES

[1] Martello, S., and Toth, P. [1990]: Knapsack Problems; Algorithms and Computer Implementations. Wiley, Chichester 1990.
[2] Papadimitriou, C.H., and Steiglitz, K. [1982]: Combinatorial Optimization; Algorithms and Complexity. Prentice-Hall, Englewood Cliffs 1982, Sections 16.2, 17.3, and 17.4
[3] Pisinger, D. [1999]: Linear time algorithms for knapsack problems with bounded weights. Journal of Algorithms 33 (1999), 1–14
[4] (2011). Logistic Regression. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_493.
[5] Brealey, Richard A., Stewart C. Myers, and Franklin Allan. 2020. Principles of corporate finance. 13th ed, 832. McGraw Hill Education..
[6] Eiteman, David K., Arthur I. Stonehill, and Michael H. Moffett. 2016. Chapter 18 Multinational capital budgeting and cross-border acquisitions. In Multinational business finance, 14th ed. Pearson..

Biography of Author:

Tamilselvan Arjunan is working as an Assistant manager. He has a total of 7 years of hands-on experience in Machine learning, Data Science and Python. He has built many AI-based products for clients. He is certified in Data Science and Python. He completed a bachelor’s degree in mechanical engineering from Anna University.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY