Name of the skill:
Algorithm Design and Complexity Analysis.

Introduction: The term "Analysis of algorithms" was coined by Donald Knuth. In computer science, the analysis of algorithms is the process of finding the computational complexity of algorithms – the amount of time, storage, or other resources needed to execute them. Usually, this involves determining a function that relates the length of an algorithm's input to the number of steps it takes (its time complexity) or the number of storage locations it uses (its space complexity).

Classification of Skill
Hard and Technical Skill. This skill requires technical knowledge and a strong mathematical background.

Prerequisites:

Proof techniques: Extremely important topic of mathematics to learn algorithms. Mostly, two proof techniques are used. Induction and contradiction. Proof is not a thing which can be just learnt, it’s a skill that has to be mastered.

Probability: Basic probability is used mostly in the analysis of some algorithms. One most popular example is hashing. Probability is used more extensively in the analysis of some randomized data structures and algorithms like skip lists and randomized quicksort or randomized median finding.

Recurrence relations: Also a useful tool in analyzing the complexity of recursive algorithms like merge sort. It’s good to know how to solve recurrence relations.

Number theory: Widely used in cryptography. There are few must know topics in number theory which are important to learn to solve problems using data structures and algorithms. Some of them are
A. Prime numbers and finding prime numbers using Sieve's algorithm
B. Greatest common divisor and Euclid's algorithm to find GCD.
C. Little modular arithmetic might also help.

Related Software Engineering Areas:

Used in Software Construction.
Specifically, the skill has a major utilization in the “Construction Design” as well as “Coding” areas of Software Engineering.
Also in the area of “Computing Foundations”, it is widely used in Problem Solving Techniques where appropriate design of algorithm and analysis of complexity is a critical feature.

Rationale for Skill:

Algorithm Analysis is the most important step for software development. It concerns "understanding" why and what the system does . No one can set an algorithm without profound understanding of the problem and its solution.

Algorithm Analysis is done before coding. It is important to be able to measure, or at least make educated statements about, the space and time complexity of an algorithm. This will allow us to compare the merits of two alternative approaches to a problem we need to solve, and also to determine whether a proposed solution will meet required resource constraints before we invest money and time coding.

It is important to study algorithms and complexity for the following reasons:

Useful in developing analytical skills
To be able to write robust programs, whose behaviour we can reason about
To learn (well-known) strategies for solving computational problems
To learn about the inherent difficulty of computational problems

Roles for the Skill:

Machine Learning Researcher
Working on innovative Machine Learning Models, developing and researching algorithms and analyzing complexities.
Machine Learning Developer
Analyst Programmer, Operation Research

Calibrating the algorithms to meet the client's requirements.
Reproduce and analyze the client's situation and propose new efficient and effective solutions.
Computer Vision Developer

Collaborate with application and integration developers to solve complex technological problems and develop outstanding immersive applications.
Analyze and improve algorithms to process images.
Data Scientist

Work Related to Skill (Activities and Artifacts):

There are many ways to write an algorithm. Some are very informal, some are quite formal and mathematical in nature, and some are quite graphical. The instructions for connecting a DVD player to a television are an algorithm. The development of an algorithm (a plan) is a key step in solving a problem. Once we have an algorithm, we can translate it into a computer program in some programming language. Algorithm development process consists of five major steps.

Step 1: Getting the description of the problem.
Step 2: Analyzing the problem.
Step 3: Developing a high-level algorithm.
Step 4: Refining the algorithm by adding more detail.
Step 5: Reviewing the algorithm.

Also, after developing the algorithm, we must check for the complexity of the algorithm in order to make the software more efficient and optimized.

Real-World Example/Scenario of Skill:

Page Rank Algorithm used by Google

Page rank algorithm is designed by two PHd students i.e Larry page and Sergey Brin. This had given birth to Google.
Search engine is an algorithm to find things efficiently in the world wide web, the algorithm developed by google is called Page rank algorithm.
This website shows a nice visualization of working of the Page Rank Algorithm as well as shows the simulation process. http://bl.ocks.org/emeeks/f448eef177b5fe94b1c0

Some other real world applications of the skill are:

Face Detection Algorithm
Dijkstra's algorithm
Huffman Coding
Heaps
Dynamic Programming

Role of Academia or Industry in Cultivating the Skill:

Algorithm Design and Complexity Analysis courses at universities help students in developing techniques needed to design algorithms and analyze complexities. The courses prepares students on:

Various algorithm design techniques
String Pattern Matching
Modulo Arithmetic Algorithms
Geometrical and Network Flow Algorithms

Many of these programs offer co-op work and internships or placements, where students can apply their knowledge and skills from what they have learnt. As the tech industry is growing rapidly, companies in all areas such as business, robotics, mobile application development and many more employ Algorithm Designers or Analyst Programmers.

Tools supporting the skill:

Microsoft BI: The Microsoft BI suite comprises a set of products that offer services ranging from data visualization to advanced algorithm and data analysis.

Weka: Also known as Waikato Environment is a machine learning software developed at the University of Waikato in New Zealand. It is best suited for data analysis and predictive modeling. It contains algorithms and visualization tools that support machine learning.

Knime: KNIME is the best integration platform for data analytics and reporting developed by KNIME.com AG. It operates on the concept of the modular data pipeline. KNIME consists of various machine learning and data mining components embedded together. KNIME has been used widely for pharmaceutical research.

Apache Mahout: Apache Mahout is a project developed by Apache Foundation that serves the primary purpose of creating machine learning algorithms. It focuses mainly on data clustering, classification, and collaborative filtering.

Skill Self-Assessment:
Score: 7 (Out of 10)
Reason: Not much experience working in this area of software engineering, but I worked as a junior Machine Learning Developer intern for 5 months and learned some basic algorithm design techniques as well as tried to optimize the application by doing complexity analysis. So I think 7 is an appropriate score for the skill.

References:

[1] SWEBOK3

[2] https://www.quora.com/How-is-algorithm-useful-in-the-context-of-software-development

[3] https://en.wikipedia.org/wiki/Analysis_of_algorithms

[4] https://stackoverflow.com/questions/8843632/prerequisites-for-understanding-algorithms

[5]https://www.quora.com/Which-mathematics-topics-need-to-be-strong-for-understanding-algorithms-and-data-structure

[6] Wegner, P., 1997, Why Interaction is More Powerful than Algorithms, Comm. ACM

[7] Gödel, K., 1931, Metamathematics, van Norstrand (New York).

[8] Jon Bentley, 1984, Programming pearls: algorithm design techniques

[9] Steven Skiena, 2008, Review of the algorithm design manual