*This study investigates the relationship between Big Data management tools such as Relational Database Management System (RDBMS), Hadoop, and APIs, and how they can be separately interlinked with advanced data analytics, specifically neural networks. The purpose of the study is to compare a RDBMS with Hadoop when processing a 1.31 GB dataset, and then apply a Neural Network. To expand the scope, this study will also include the usage of APIs (Keras library) for implementing Neural Networks. This study was conducted using my personal laptop to load a 1.31 GB dataset into a RDBMS and Spark. I utilized Jupyter Notebooks to interact with these two technologies, exploring computing times, roadblocks faced, and other insights. Following this, we applied the same Neural Network to predict if certain jobs are more popular based on gender. Another aspect of the study involves utilizing an API; for this, we are employing Keras and a Convolutional Neural Network (CNN). Our aim is to evaluate the performance of the CNN model in classifying movie reviews as positive or negative based on their sentiment. The research findings indicate that using RDBMS or Hadoop for data processing is not as quick and straightforward as using an API like Keras, where you simply import the data without the need to worry about how to push it into databases, this becomes clear when modelling data using NN via Jupyter Notebooks.*

Relational Database Management Systems have been well-established since the late 1970s; at that time, the concept of Big Data was not the same as it is today. As technology rapidly advanced, the industry needed to process large amounts of data. To address this need, an open-source framework for writing and running distributed applications, called Hadoop, entered the scene. These two technologies, RDMS and Hadoop, are great; however, the implementation of both requires a high level of technical software skill. This is where APIs offer a solution to this problem, which the industry refers to as Machine Learning as a Service (MLaaS), e.g., Azure ML or AWS ML, just to mention a few

The intention of this paper is to explore all three technologies—RDBMS, Hadoop, and APIs—to determine which one is the best fit for data extraction and processing in the context of Neural Networks implementation. This consideration is crucial, given that many individuals interested in Machine Learning are not software developers, and the need for a 'plug-in' to deploy their ML models is evident.

The chosen topic is Big Data and Neural Networks, with NN being considered a type of Machine Learning (ML) process known as Deep Learning . The field of Big Data is constantly growing and encompasses a need for efficient data management and processing tools. Two well-known tools for handling and analyzing large datasets are Relational Database Management Systems (RDBMS) and Hadoop. However, the rampant advancement of Machine Learning and Neural Networks, the integration of these data management tools with advanced analytics technologies is the focus of this paper.

How do Relational Database Management Systems (RDBMS) and Hadoop compare in terms of efficiency and effectiveness in processing large datasets for the application of neural networks, and how can APIs, particularly the Keras library, streamline the implementation of neural network models in data analytics.

• Examine the current state of RDBMS, Hadoop, and APIs when used in modeling NN.

• Store a 1.31 GB dataset in both an RDBMS (SQL) and Hadoop, and then retrieve the data into a Jupyter Notebook to model a neural network.

• Utilize an API (Keras) to model a neural network and compare its performance in conjunction with RDBMS and Hadoop.

• Discuss the rationale behind the selection of the NN model for both scenarios

The current state of RDBMS has evolved substantially with enhancements in storage, speed, and scalability by using cloud-based solutions. The future holds a shift for RDBMS transitioning to a NoSQL database. To understand why NoSQL is taking over RDBMS, it is crucial to talk about: Schemas, where NoSQL uses dynamic instead of static schemas; the type of data to be stored, with NoSQL databases offering advantages for hierarchical data storage due to their flexible data models and scalability, while RDBMS are not that flexible; scalability, with NoSQL depending on horizontal scalability and RDBMS on vertical scalability; and other points where NoSQL surpasses RDBMS, including data warehouse, complexity, cloud, and big data handling, and output performance.