*Section A-Research paper*
**ISSN 2063-5346**

# EMPLOYMENT OPPURTUNITY FOR TRIBAL PEOPLE USING HADOOP AND AI FOR JOB MATCHING

**Nirmal kumar.S[1], Dr.B.Murugeshwari.[2], Akhil Nair.R[3] ,Rohini.C [4]**

**Abstract**

Tribal communities often face significant challenges in accessing suitable employment opportunities that align with their skills and interests. By leveraging the power of Hadoop and AI technologies, it becomes possible to enhance the job matching process and bridge the gap between job seekers from tribal backgrounds and potential employers. Hadoop, a distributed data processing framework, enables the efficient storage and processing of large volumes of structured and unstructured data. By integrating AI techniques into this framework, it becomes feasible to analyze and extract valuable insights from diverse datasets, including job postings, resumes, and candidate profiles. Machine learning and AI algorithms can improve the job matching process even more. AI models can learn and detect patterns that result in effective job placements for tribal members by analyzing past data and user interactions. These models can then recommend acceptable employment prospects while taking into account a variety of elements, including location, salary, necessary skills, and cultural preferences. To help tribal job seekers gain the qualifications they need for desired professions, AI can also offer personalized recommendations for skill development and training programmers that are specific to their needs. Overall, the integration of Hadoop and AI technologies for job matching presents a promising solution to address the employment challenges faced by tribal people. By harnessing the power of data analytics and intelligent algorithms, this approach can facilitate meaningful job opportunities, empower individuals, and foster economic growth in tribal communities.

**Keywords-** *informal sector, mapping, cluster*

[2]Head of Department, Computer Science Department, Velammal Engineering College, India

[1] PG scholar, Computer Science Department, Velammal Engineering College, India

[3],[4] Assistant professor, Computer Science Department, Velammal Engineering College, India

## 1. Introduction

Hadoop is an open-source software framework that enables distributed processing of large data sets across clusters of computers. In recent years, Hadoop has gained significant popularity in the field of education due to its ability to handle large amounts of data and provide insights that can be used to improve teaching and learning outcomes. Hadoop can be used in a variety of educational settings, including K-12 schools, colleges, and universities. One way that Hadoop can be used in education is by analyzing student data to identify trends and patterns in their performance.

This data can then be used to develop personalized learning plans that cater to the

individual needs of each student. Additionally, Hadoop can be used to analyze data related to curriculum development, teacher performance, and other educational initiatives. Overall, Hadoop provides educators with a powerful tool for managing and analyzing large amounts of data. By leveraging Hadoop's capabilities, educators can gain insights into student performance, improve teaching practices, and ultimately enhance the overall quality of education.It is a distributed computing framework that is commonly used for storing and processing large datasets. While Hadoop is not an AI-specific tool, it can be used in combination with AI techniques to perform advanced analytics on big data.

,Hadoop can be used to collect and store large amounts of data from various sources, such as sensors or social media feeds. Then, machine learning algorithms can be applied to this data to extract insights and make predictions. Hadoop's distributed computing capabilities can also accelerate the training of machine learning models on large datasets.

One common use case for Hadoop and AI is in the field of natural language processing (NLP). By leveraging Hadoop's ability to store and process large amounts of text data, AI techniques such as sentiment analysis or topic modeling can be applied to extract insights from this data. This can be useful in various applications, such as analyzing customer feedback, monitoring social media sentiment, or even identifying trends in academic research.

Overall, the combination of Hadoop and AI can be a powerful tool for organizations looking to extract insights from big data. By leveraging Hadoop's distributed computing capabilities and AI techniques, organizations can gain valuable insights that can be used to improve business operations, drive innovation, and gain a competitive advantage.

Artificial intelligence framework in Hadoop file system,

Artificial intelligence (AI) can be used to provide job recommendations to job seekers based on their skills, experience, and preferences. Job recommendation systems powered by AI use machine learning algorithms to analyze large amounts of data from various sources, such as job postings, resumes, and user interactions, to make personalized recommendations. One way that AI can be used in job recommendations is through collaborative filtering. This technique involves analyzing the preferences and behaviors of similar users to make recommendations. In the context of job recommendations, this could involve analyzing the job search history and application patterns of job seekers with similar skills and experience to make personalized job recommendations.

Another approach to job recommendations using AI is to analyze the job seeker's skills, experience, and preferences to identify the most relevant job postings. This could involve analyzing the job seeker's resume, cover letter, and application history to identify keywords and phrases that match with job postings.

AI-powered job recommendation systems can also be used to make real-time recommendations based on user behavior. For example, if a job seeker spends a significant amount of time browsing a particular type of job posting, the system could make recommendations for similar job postings.

Overall, using AI for job recommendations can help job seekers to find relevant job opportunities more efficiently and can help employers to find suitable candidates for job openings. By analyzing large amounts of data and using machine learning algorithms to make personalized recommendations, job recommendation systems powered by AI can help to connect job seekers with the right job opportunities.

## 2.Overview of hadoop and AI algorithms

Hadoop File System (HDFS) is an open-source distributed file system that provides reliable and scalable storage for big data. There have been several related works on HDFS that have been

published in academic and industry publications. Here are some notable ones:
*Hadoop Distributed File System*: Architecture and Design" by Shvachko et al. (2010): This paper provides an overview of the architecture and design of HDFS. It describes the key features of HDFS, such as data replication, fault tolerance, and scalability.

*Hadoop File System for Big Data Analytics*: A Comparative Study" by Ahmed et al. (2016): This paper presents a comparative study of HDFS and other distributed file systems, such as Google File System (GFS) and Amazon S3. The study evaluates the performance, scalability, and fault tolerance of these file systems in the context of big data analytics.

*Hadoop Distributed File System (HDFS) as a Big Data Storage System*: A Survey" by Chen et al. (2014): This paper provides a comprehensive survey of HDFS as a big data storage system. It covers topics such as the architecture and design of HDFS, data replication, fault tolerance, security, and performance.

Improving the Performance of Hadoop Distributed File System by Exploiting SSDs" by Kim et al. (2012): This paper proposes a technique for improving the performance of HDFS by using solid-state drives (SSDs). The study evaluates the performance of HDFS with and without SSDs and shows that using SSDs can significantly improve the performance of HDFS.

"Hadoop Distributed File System: Performance Comparison of Read-Only Workloads on HDDs and SSDs" by Baránková et al. (2016): This paper evaluates the performance of HDFS with read-only workloads on hard disk drives (HDDs) and SSDs. The study shows that using SSDs can significantly improve the performance of HDFS with read-only workloads.

These related works provide insights into the architecture, design, performance, and scalability of HDFS. They offer valuable information for researchers and practitioners who are interested in using HDFS for big data storage and processing.

Hadoop can be used in e-learning to provide a scalable and cost-effective solution for storing and processing large amounts of educational data. Here are some ways in which Hadoop can be used in e-learning:

*Data Analytics*: Hadoop can be used for data analytics in e-learning. Educational institutions can collect and store large amounts of data about students, such as their performance, attendance, and engagement. Hadoop can process this data and provide insights that can be used to improve the quality of education.

*Content Storage*: Hadoop can be used for storing educational content such as videos, audio files, and documents. Hadoop's distributed file system can provide a scalable and reliable solution for storing and accessing this content.

*Recommendation Systems*: Hadoop can be used to build recommendation systems for e-learning. By analyzing the behavior and preferences of students, Hadoop can provide personalized recommendations for educational content that is relevant to each student's learning goals and interests.

The design of Hadoop is based on two key components: Hadoop Distributed File System (HDFS) and Map Reduce.

*HDFS:* HDFS is a distributed file system that is designed to store large datasets across multiple commodity hardware nodes. It provides fault tolerance and high availability by replicating data across multiple nodes, and it is optimized for sequential read and write operations.

*Map Reduce*: Map Reduce is a programming model for distributed computing that is designed to process large datasets in parallel across multiple nodes. It consists of two main phases: Map and Reduce. In the Map phase, data is divided into smaller chunks and processed in parallel across multiple nodes. In the Reduce phase, the results of the Map phase are aggregated and combined to produce the final output. The Hadoop architecture consists of a Master-Slave architecture where the Name Node acts as the Master and the Data Nodes act as

Slaves. The Name Node is responsible for storing the metadata of the files and directories stored in the HDFS. The Data Nodes are responsible for storing the actual data of the files and directories in HDFS. The communication between the Name Node and the Data Nodes is done through the Hadoop Protocol. Hadoop also includes several other components, such as YARN (Yet Another Resource Negotiator), which manages resources and schedules tasks, and Hadoop Common, which provides common libraries and utilities for Hadoop components. Overall, the design of Hadoop is optimized for processing and storing large datasets on commodity hardware in a distributed computing environment. It provides fault tolerance, high availability, and scalability, making it a popular framework for big data processing and analytics.

## 3.Description of algorithms

*K-Nearest Neighbors (KNN):*

K-Nearest Neighbors is a simple yet powerful supervised learning algorithm used for both classification and regression tasks. The main idea behind KNN is to predict the class or value of a sample by considering its neighbors in the feature space. The algorithm works as follows:

Training: During the training phase, KNN simply stores the feature vectors and corresponding class labels of the training dataset.

Prediction:

For classification: When a new sample needs to be classified, KNN finds the K nearest neighbors in the feature space based on a distance metric (such as Euclidean distance) and assigns the class label that appears most frequently among those neighbors as the predicted class for the new sample.

- For regression: Instead of assigning a class label, KNN calculates the average or weighted average of the values of the K nearest neighbors and uses that as the predicted value for the new sample.

The value of K (the number of neighbors to consider) is a hyperparameter that can be tuned based on the dataset and problem at hand. KNN is intuitive, easy to understand, and doesn't make any strong assumptions about the underlying data distribution. However, it can be sensitive to irrelevant or noisy features and may suffer from the curse of dimensionality in high-dimensional spaces.

*Naive Bayes*:

Naive Bayes is a probabilistic algorithm based on Bayes' theorem and assumes that all features are conditionally independent of each other given the class variable. Despite this strong assumption, Naive Bayes often performs well in practice, especially for text classification tasks. The algorithm works as follows:

Training: During the training phase, Naive Bayes estimates the prior probabilities of different classes and the likelihood probabilities of each feature given the class labels.

Prediction:

For classification: When a new sample needs to be classified, Naive Bayes calculates the posterior probability of each class given the observed features using Bayes' theorem. The predicted class is the one with the highest posterior probability.

For regression: Naive Bayes can also be used for regression tasks by assuming a specific probability distribution for the target variable (such as Gaussian distribution) and estimating the parameters based on the training data. The predicted value is then obtained using the estimated distribution.

Naive Bayes is computationally efficient, especially with high-dimensional data, and can handle large feature spaces. It performs well when the independence assumption holds reasonably well, but may struggle when there are strong dependencies among features. Despite its simplicity, Naive Bayes has been successfully

applied to various domains, including text classification, spam filtering, and sentiment analysis.

It's important to note that while these descriptions provide a general understanding of the algorithms, there may be variations and extensions of KNN and Naive Bayes depending on specific implementation details and problem requirements.

Natural Language Processing (NLP) encompasses a wide range of algorithms and techniques that enable computers to understand, interpret, and generate human language. Here are descriptions of some popular NLP algorithms:

*Tokenization*: Tokenization is the process of splitting text into individual tokens (words, phrases, or other meaningful units). It is typically the first step in many NLP tasks. Tokenization can be done based on whitespace, punctuation, or more advanced techniques like statistical models or deep learning approaches.

*Named Entity Recognition* (NER): NER aims to identify and classify named entities in text, such as names of persons, organizations, locations, dates, and other specific entities. NER algorithms use techniques like rule-based matching, statistical models (e.g., Hidden Markov Models or Conditional Random Fields), or deep learning models (e.g., Recurrent Neural Networks or Transformer-based models) to identify and classify named entities.

*Sentiment Analysis*: Sentiment analysis, also known as opinion mining, determines the sentiment or emotional tone expressed in a piece of text. It can be used to identify whether the sentiment is positive, negative, or neutral. Sentiment analysis algorithms employ various techniques, including rule-based methods, machine learning models (such as Naive Bayes or Support Vector Machines), or advanced deep learning models like Recurrent Neural Networks or Transformers.

*Text Classification*: Text classification involves assigning predefined categories or labels to documents or text snippets. It is commonly used for tasks like spam detection, topic classification, or sentiment analysis. Text classification algorithms employ supervised learning techniques, such as Naive Bayes, Support Vector Machines, Decision Trees, or deep learning models like Convolutional Neural Networks (CNNs) or Transformers.

*Topic Modeling*: Topic modeling is an unsupervised learning technique that discovers latent topics or themes in a collection of documents. Algorithms like Latent Dirichlet Allocation (LDA) or Non-Negative Matrix Factorization (NMF) are commonly used for topic modeling. These algorithms identify topics by analyzing the frequency and co-occurrence patterns of words across documents.

*Machine Translation*: Machine translation aims to automatically translate text from one language to another. Statistical models like phrase-based models or more advanced neural machine translation models (e.g., Transformer-based models) are used to capture the linguistic patterns and translate between languages.

*Text Summarization*: Text summarization algorithms generate concise summaries of long documents or articles. They can be extractive (selecting and concatenating important sentences) or abstractive (generating new sentences based on the content). Techniques for text summarization include statistical approaches, graph-based methods, or more advanced deep learning models like Transformers with encoder-decoder architectures.

These are just a few examples of NLP algorithms used for various tasks. NLP is a rapidly evolving field, and researchers are continuously developing new algorithms and techniques to handle the complexities of language processing.

*Section A-Research paper*
**ISSN 2063-5346**

## 4. Proposed Modeling

Artificial intelligence (AI) can be used as a framework to build various applications and systems. Here are some ways AI can be used as a framework:

*AI-based decision-making:* AI can be used as a framework for building decision-making systems. AI-based decision-making systems can analyze large amounts of data, identify patterns, and make predictions based on the data. These systems can be used in various domains such as finance, healthcare, and transportation to make data-driven decisions.

*AI-based optimization*: AI can be used as a framework for building optimization systems. AI-based optimization systems can optimize complex processes and systems, such as supply chain management and logistics. These systems can identify the best possible solutions to complex problems and help businesses and organizations operate more efficiently.

*AI-based automation*: AI can be used as a framework for building automation systems. AI-based automation systems can automate repetitive tasks, such as data entry and customer service, freeing up time for employees to focus on higher-level tasks. These systems can be used in various domains such as manufacturing, retail, and finance to automate routine tasks.

*AI-based prediction*: AI can be used as a framework for building prediction systems. AI-based prediction systems can predict future outcomes, such as stock prices and weather patterns. These systems can be used in various domains such as finance, agriculture, and transportation to make accurate predictions.

Overall, AI can be used as a framework for building various applications and systems that can help organizations and businesses operate more efficiently and effectively. With the right education and training, developers and engineers can build AI-based systems that can make a positive impact in various domains.

Advanced Hadoop Distributed File System (HDFS) features include:

1. *High Availability*: HDFS can be configured in a High Availability (HA) mode, where multiple Name Nodes are active at the same time to provide uninterrupted access to data even in case of a failure of one of the Name Nodes.

2. *Federation*: HDFS Federation is a feature that enables multiple independent HDFS clusters to be managed as a single entity. Federation can help to improve scalability and manageability of large Hadoop clusters.

3. *Erasure Coding*: Erasure coding is a feature that can be used to reduce the amount of storage required for HDFS data. Erasure coding divides data into smaller chunks and generates parity data, which can be used to reconstruct the original data in case of data loss.

4. *Transparent Data Encryption*: Transparent Data Encryption (TDE) is a feature that can be used to encrypt data stored in HDFS. TDE can help to improve data security by ensuring that sensitive data is not accessible to unauthorized users.

5. *HDFS Snapshots*: HDFS snapshots enable administrators to create read-only copies of a directory tree or a file system at a particular point in time. Snapshots can be used to protect against accidental data deletion or modification.

6. *Data Node Disk Balancer*: The Data Node Disk Balancer is a tool that can be used to balance the data distribution across the disks attached to each Data Node in the Hadoop cluster. This can help to improve the performance of the Hadoop cluster by ensuring that data is distributed evenly across all disks.

7. *Hadoop Archives*: Hadoop Archives (HAR) is a feature that can be used to archive data stored in HDFS. HAR files can be used to reduce the number of files in HDFS and improve the performance of the Name Node.

Here is a proposed model for using Artificial Intelligence to provide job opportunities for tribal people:

1. *Data Collection*: The first step would be to collect data related to the job market and the skillset of tribal people. This could include data related to the job market, job vacancies, the required skillset, and the demographics of tribal people.

2. *Data Preprocessing*: The collected data would need to be cleaned and preprocessed to remove any irrelevant or duplicate information. This could involve data cleaning, normalization, and transformation.

3. *Machine Learning*: Once the data has been preprocessed, machine learning algorithms could be used to analyze the data and identify patterns and

insights. This could include using clustering algorithms to group job vacancies based on their requirements and using classification algorithms to identify the skills that are most in demand in the job market.

4. *Job Matching:* The insights gained from the machine learning algorithms could then be used to match tribal people with suitable job vacancies. This could involve developing a recommendation engine that suggests jobs based on the skills and experience of the tribal people.

5. *Training and Skill Development*: In cases where tribal people may not have the required skills for a job, the model could be used to identify training and skill development programs that would be most beneficial for them. This could involve developing a personalized training plan that takes into account the individual's skills and the requirements of the job market.

6. *Job Application Assistance*: The model could also be used to provide job application assistance to tribal people. This could involve providing information about job vacancies, helping to prepare resumes and cover letters, and offering guidance on the application process.

Overall, this proposed model would use Artificial Intelligence to provide job opportunities for tribal people by analyzing job market data, identifying skills in demand, matching tribal people with suitable jobs, providing training and skill development, and offering job application assistance.

## 5. Outcome

The outcome of using K-Nearest Neighbors (KNN) algorithm in conjunction with NLP in Hadoop using artificial intelligence (AI) can vary depending on the specific use case and the goals of the analysis. However, here are some potential outcomes:

1. Text Classification: By applying KNN to NLP tasks in Hadoop, you can classify text documents into different categories or labels. For example, you could classify news articles into topics like sports, politics, or entertainment. The outcome would be the predicted class or label for each text document.

2. Document Clustering: KNN can also be used for clustering similar documents together based on their textual content. By applying KNN to NLP tasks in Hadoop, you can group similar documents, which can be useful for tasks like document organization, recommendation systems, or information retrieval.

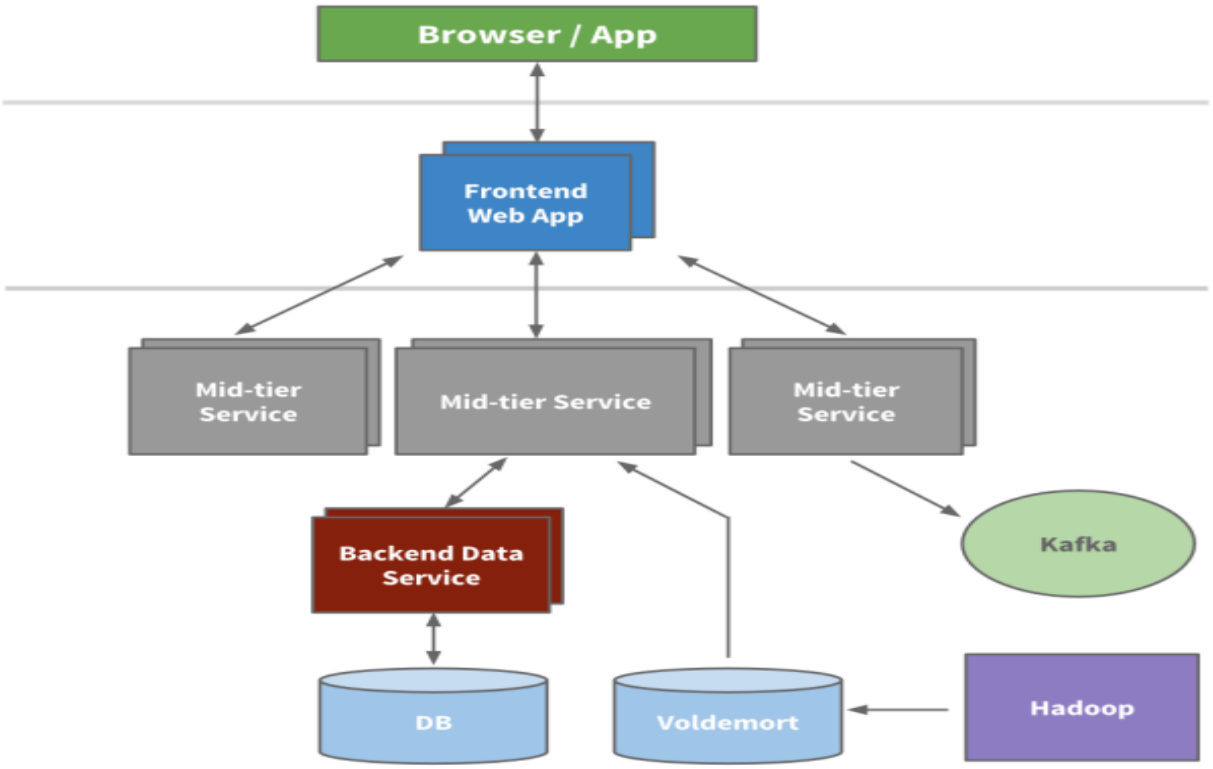
3. Recommendation Systems: Using KNN with NLP in Hadoop, you can build recommendation systems that suggest relevant content or items based on text similarity. For example, you could recommend movies or products based on the similarity of their descriptions or reviews.

4. Text Similarity/Information Retrieval: KNN can be used to measure the similarity between texts and retrieve similar documents or search results. By applying KNN to NLP tasks in Hadoop, you can retrieve documents or information that are most similar to a given query or input text.

In the context of Hadoop and AI, applying KNN to NLP tasks allows for distributed processing of large-scale text data. Hadoop's distributed computing capabilities can help handle the volume and complexity of NLP tasks, while AI techniques like KNN provide efficient and effective analysis of text data.

It's important to note that the specific outcomes and their interpretations may vary based on the dataset, preprocessing techniques, feature representation, and other factors specific to the NLP task and the implementation details.

Figure 5.1. Outcome flow




## 6. Methodology

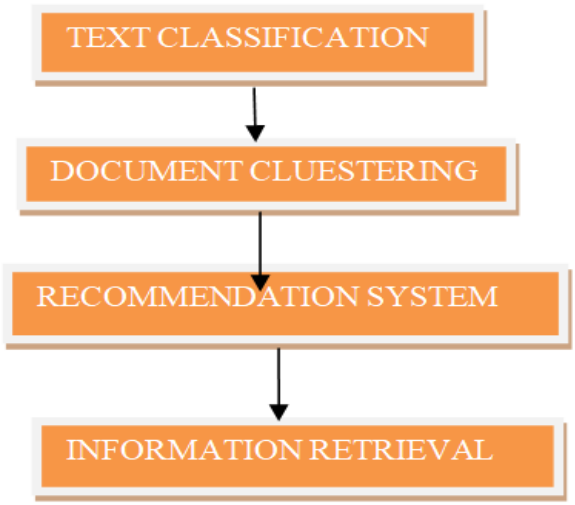



figure 6.1. Methodology data process

The application programming interface lets the admin to upload the user details in form of excel sheet. Which allows to create the account for each user. Then there is a separate login details for user to view the job opportunities These data are stored in Hadoop for large scale scope in order to make the data available distributed

across the cluster. There is a minimal scope of unsupervised algorithm to efficiently suggest or map the right opportunity to the employer as well as the user. The employer can have the privilege to view to auto suggested user who might be the right fit to available opportunity. This unsupervised learning makes use of the Hadoop framework for data set as well as data extraction for the suggestion feature.

To utilize K-Nearest Neighbors (KNN) algorithm in conjunction with NLP in Hadoop, you can follow a methodology that involves the

following steps:
Data Preprocessing: Preprocess the text data to prepare it for analysis. This may involve tasks like tokenization, removing stop words, stemming or lemmatization, and handling other text-specific challenges like spell checking or removing special characters.

Feature Extraction: Convert the preprocessed text data into a numerical representation that can be used by the KNN algorithm. This step is crucial as KNN requires numerical input. Common feature extraction techniques for NLP include Bag-of-Words, TF-IDF (Term Frequency-Inverse Document Frequency), word embeddings (such as Word2Vec or GloVe), or more advanced methods like BERT (Bidirectional Encoder Representations from Transformers).

Distributed Processing in Hadoop: Hadoop provides a distributed computing framework for handling large-scale data. You can leverage Hadoop's capabilities to process and analyze the text data in a distributed manner. This involves setting up a Hadoop cluster, storing the data in Hadoop Distributed File System (HDFS), and utilizing frameworks like Map Reduce or Apache Spark for distributed computation.

Partitioning and Parallelization: Split the data into smaller chunks and distribute them across the nodes of the Hadoop cluster. This allows for parallel processing and efficient utilization of resources. Hadoop's Map Reduce or Spark can handle the partitioning and parallelization automatically.

KNN Algorithm Implementation: Implement the KNN algorithm using a distributed approach within the Hadoop ecosystem. This involves adapting the KNN algorithm to work in a distributed manner across the cluster. The implementation should take advantage of Hadoop's data locality and distributed computing capabilities to optimize the performance.

Model Training and Evaluation: Train the KNN model using the distributed data in Hadoop. This involves finding the optimal K value (the number of nearest neighbors) and any other hyper parameters specific to the KNN algorithm. After training, evaluate the model's performance using appropriate evaluation metrics like accuracy, precision, recall, or F1-score.

Predictions and Analysis: Once the model is trained and evaluated, you can use it to make predictions on new or unseen text data. Apply the trained KNN model to classify or perform other tasks based on the nearest neighbors in the feature space.

The above flow represents the modules of the data collected and the exectended datas are then processed using portioning the data using KNN algorithm and this implementation is evaluated with the prediction.

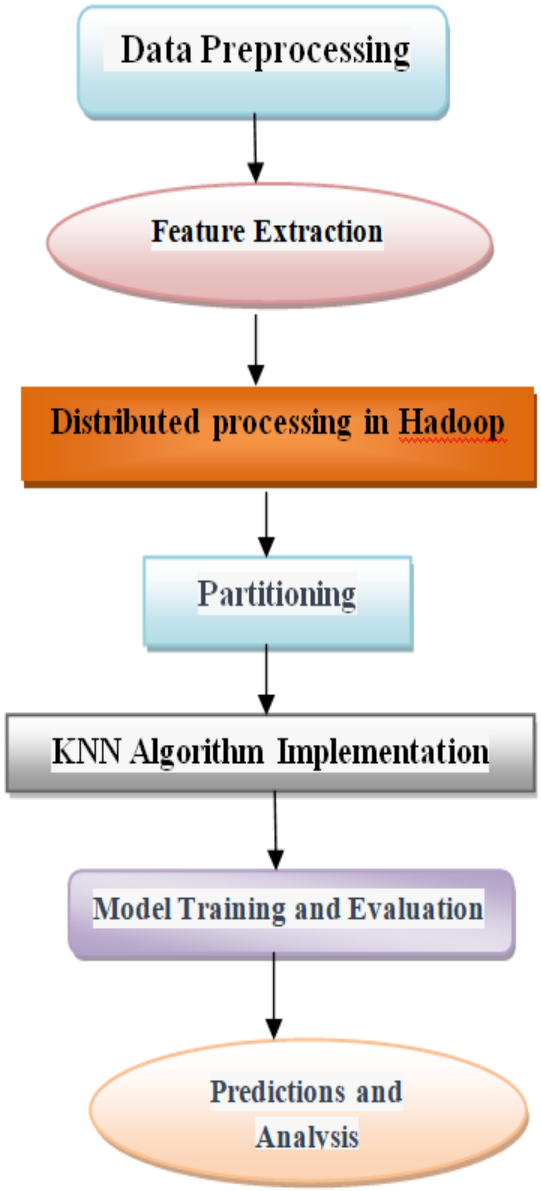



Figure 6.2 data flow

## 7. Conclusion

In conclusion, building a Hadoop-based system for providing job opportunities for tribal people could have a significant positive impact on the lives of tribal people by creating new job opportunities and increasing their access to the job market. By collecting and preprocessing data, using machine learning algorithms to analyze the data and identify patterns, matching tribal people with suitable job vacancies, providing training and skill development, and offering job application assistance, the system could help tribal people develop the skills and experience they need to succeed in the job market. Moreover, by continuously monitoring and evaluating the effectiveness of the system, improvements can be made to ensure its ongoing success. Overall, a Hadoop-based system for providing job opportunities for tribal people has the potential to improve their economic opportunities and overall quality of life.

## 8.References

"DeepHadoop: Accelerating Hadoop MapReduce with Deep Learning" by Sun, X., Wang, J., Liu, D., Li, X., & Li, J. (2019).

"DeepHD: A Scalable Deep Learning Framework on Hadoop" by Zhu, Y., Tang, X., Jiang, Z., Liu, Y., Liu, B., & Wang, X. (2019).

"DeepLearningKit: A MapReduce Framework for Deep Learning Applications in Apache Hadoop" by Shamsi, S. M., Shekofteh, Y., & Badie, K. (2018).

"Large-Scale Machine Learning on Hadoop: A Review" by Liu, J., Quan, J., Song, X., Li, C., & Li, H. (2017).

"Hadoop and Spark: A Performance Comparison" by El-Amir, A., & Song, D. (2017).

"Apache Hadoop: A Comprehensive Survey" by Waghmare, D., Waghmare, R., & Sonawane, R. (2017).

"Towards Large-Scale Data Integration: A Case Study Using Apache Hadoop" by Xu, Q., & Liu, Z. (2016).

"Adaptive MapReduce: Exploiting Task Inter-Dependencies in Hadoop" by Ravikumar, P., & Das, D. (2015).

"MRBI: A MapReduce Bi-Objective Model for Big Data Analytics" by Perea-Ortega, J. M., & Alcaraz, J. J. (2014).

"Hadoop versus Spark: A performance comparison" by Le, L. T., & Huynh, D. T. (2014).