Engineering a Novel Recruitment System Using Gradient Boosted Decision Tree Algorithm and Business Intelligence Principles

Dr. Shaik Javed Parvez[1], Abarna Arulprakasam[2], I. Thowbik Ahamed[3], J. Jane Carolyn[4]

[1]Department of Information Technology, Hindustan Institute of Technology and Science, Chennai, India. E-mail: shaikjp@hindustanuniv.ac.in

[2]Department of Information Technology, Hindustan Institute of Technology and Science, Chennai, India. E-mail: abarna5820@gmail.com

[3]Department of Information Technology, Hindustan Institute of Technology and Science, Chennai, India. E-mail: thowbikahamed04@gmail.com

[4]Department of Information Technology, Hindustan Institute of Technology and Science, Chennai, India. E-mail: carolynjane1819@gmail.com

Abstract

Using Business Intelligence to strategize the use of technology and reduce the rate of an Indian economic issue of unemployment by developing a Start-up web application using By coordinating with the Indian Sports and Cultural Institutions, and establishing agreements with NGOs/Corporates and other sponsorship providers to raise stipends/employment for underprivileged Indian youth using a novel algorithm with the GBDT algorithm for a suitable and secure recruiter recommendation system using firebase encryption. Since the Indian youth require a vast stage of opportunities and growth in variety of employment, the creatives and the sports industry are primarily focused to display limelight on those nurtured with inborn talents using our recruiter recommendation system using the 3 BI principles namely relevance, query intelligence and personalization. This app will seek to eradicate insufficient assessments, location barriers, inaccessibility of academies due to lack of funds, inadequate cultural categories, loopholes in the database due to weak security, and manual screening and verification of UUP details, this paper aims to give an overview and a studied literature review including proposed concepts and describes the need, significance and goal of developing an algorithm using business intelligence and machine learning to solve a socio-economic crisis.

Keywords: Business Intelligence, Indian, Youth, Cultural, Unemployment, GDBT, Recruiter Recommendation System.

DOI: 10.47750/pnr.2022.13.S03.074

INTRODUCTION

"The final solution for unemployment is *Work*", the late 30th president of the United States, Calvin Coolidge declared. Millions of opportunities and jobs are being offered at different rates of speed, platforms and yet the great unemployment move is still at large. Observations in [1] state arts and cultural industries suffer a lack of talent and proper guidance to the same but its original value is more than $166 billion per year, as the ICT industry is booming, India especially suffers a lack of cultural participation and growth, globally. In [1], the creatives, fine arts, sports, and games industry directly gives economic support to many workforce/laborers over a wide area of occupations such as chartered accountants, finance specialists, backstage workers, advertising specialists, IT personnel, trainers, and more. According to [1], the National Endowment for Arts stated that unemployment in arts was higher than the normal workforce before the recessions and technology boosts. India alone has reached 10.3% more unemployment in the overall economy. Professionals, artists are directly affected by the massive unemployment shift, and by 64%, a total of 129,000 workers are displaced aimlessly.

Fig 1.1. CMIE Series, Unemployment rate of India as of May 2021

1.1 Problem Statement

After establishing the primary issue of cultural unemployment, the issue of having the right platform to provide a vast spectrum of categories for choice and inclusiveness of all communities and languages still remains

unchanged. Hence the creatives and sports industry lack an exclusive platform to connect job providers and the pool of talent with the support of funds for onboarding.

1.2 Objective of the Study

To analyze the current economic state of unemployment and assess the categories for employment to rise and pave way for Big Data to promote better and more efficient platforms, with machine learning algorithms to develop the ideal RRS-recruiter recommendation system for cultural employment using GBDT for complex data and rank other algorithms to seek out the boon and bane of having a functional RRS and an existing system or online job site. This paper will introduce a broader view to the existing methodology and prioritize Gradient Boosted Decision Tree Algorithm to filter the recommendation system and provide users more chances of being placed using the following Business Intelligence principles and metrics:

1. **Relevance Strategy**- The target position will consist of running and returning the "relevant" candidates that match the criteria provided.
2. **Query Intelligence**-Not only the same but similar candidates with the same interests and skills are taken into account.
3. **Personalization**- For the recruiter to customize their search option since many times they might find suitable candidates outside the recommended list, this enables them to filter out manually for the highly competitive roles.

1.3 Conceptual Framework - Novelty

The conceptual framework of the proposed methodology consisted of a system workflow to feed the collected structured and unstructured data into Spark Analytics tool to interpret attributes that will align with the offers and criteria and get them scheduled with gradient values and route preliminary tests via notification senders and a third-party conference tool hosted by the academy.

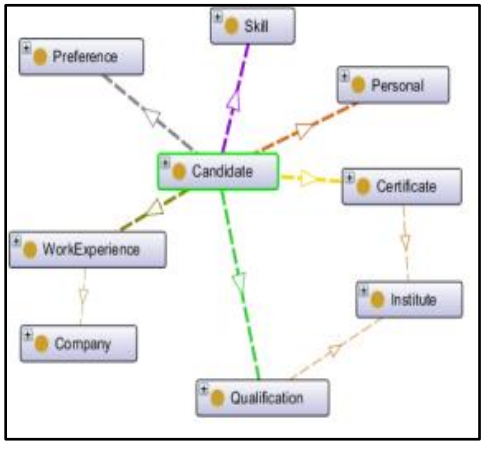


Fig 1.3.2: Candidate's gradient attributes

Related Works

Global recessions and pandemics like the Spanish flu and COVID-19 have increased the socio-economic crisis of unemployment on a global scale, and has shown a deteriorating economic slowdown, in the graduate employee's section; where one being knowledgeable but unable to get a job in the current market due to various reasons, has only increased over the years, not due to unavailability of jobs but poor categories, industries, and selection of candidature. [2].

The common reasons stated by [2] about the gradual decrease in the rate of graduate employment include huge supply of graduates versus low demand of jobs in particular sectors, a huge pool of talent in multiple areas of recruitment versus poor platforms, and less to no sustainability of those cultural jobs `due to heavy payment of royalty fees to the cultural academies, outdated curriculum and infrastructure is outdated and not being attentive to global trends and national involvement in the economic standards, corruption, politics, and occurrences of pandemics. It is observed and mystified that enterprises, startups, and cloud-intelligent platforms bring connectivity within people, places, jobs, and positions in this fast-changing environment. According to [3] business intelligence is used to combine models and algorithms to establish the process of job seeking, efficient and flexible. For instance [3] stated, the Bayesian decision tree will give numerical values based on the profile management system to the job providers and the k-means clustering algorithm will help in studying the current market demand. The QoS or Quality-of-Service is maintained and facilitated by these intelligent algorithms and principles such as the Keep-It-Simple-and-Smart principle that will solve other control problems, and deep reinforcement learning is used to schedule the jobs for continuous job requests.

Most online recruitment systems do not analyze the value calculated taking into account the graduates and the job providers have an inversely related value and the characteristics of the graduates deeply to categorize the right job and profile, according to [4], business intelligence is used to quicken the process of choosing according to relevance but also considering certain attributes that are necessary more deeply in Linkedin-which is known to be a market leader [4,5] to all industries except cultural employment but recruiters find it moderate to hard to find the right candidates on the most popular job site due to inadequate criteria and mismanaged profiles, Monster.com, Naukri, etc.

Not only does BI & A filter out the contents but can connect to ML algorithms and perform the suitable processes to get the desired outcome.

Using the Apriori algorithm, students' resume pool will be filtered from vast data by analyzing various information concerning the job description and skills needed to the skills required. Usually, this machine learning algorithm attains the best out of every candidate hence the rate of choosing is very low but very accurate, although it improves employee

satisfaction and saves time for a tedious recruitment process. The system uses features including recommendation algorithms, functions for specific user requirements, and employment market strategizing tools in Big Data to make further reasonable recommendations for graduates and college students. [6], [7].

According to [8], most academies, institutes, and private sectors use online services to recruit and accept applicants for their industry, hence the distribution of "online" resumes has been overwhelming for recruiters today. [10] has initiated a web portal, convenient and user friendly called "Smart Applicant Ranker" that uses a ranking algorithm to substitute into RS(c, j) = ((ri × wi) + (rj × wj)) to save the data securely. SAR uses the advantage of developing individual ranks according to the resumes or details shared by the user to narrow the search to match their requirements by developing smaller sets of data, the ranks may vary from user to domain and criteria.

According to [9], a particular sector i.e the music industry was taken into account for the cultural employment analysis; music producers and artists officially upload their tracks or audios into the platforms that always require to make a royalty payment, that is to pay a couple of dollars to publish the song to a global audience. This arose an issue for unsigned artists as they do not belong to record labels that can fund them to publish a track without self-investment. Besides finding a free platform available to musicians to earn without having to pay, the total revenue the musicians receive is also not for the artist but went to the owners of the site as well [9].

Fig 2.1 Existing System Flow of Music Employees

According to [10] vast amounts of data are taken to analyze the deserving candidates for candidature. Hence the similarity and reliability of data are two modules taken into consideration to focus on, as most resumes will be similar in their respective categories and departments. But [10] created datasets to help assess the real-time similarity in resumes and how recruiter recommendation systems respond using CNN. They have extracted huge amounts of data by scraping the web from popular job sites and creating a CSV file to display the features to be considered.

The following algorithms are used to form a grouped result and then compared to each other algorithm to find the effective method using CNN, support vector machine (SVM), naive Bayes, term frequency-inverse document frequency, in addition over 1000 applicants' datasets is taken and modeled to classify unlabelled and labeled information in [10],[11].

It was observed that accurate job and resume classification are undoubtedly what the job seeker and the recruiter consider to be vital and hard to ignore. An automatic text classification system was built by [10] who state to have utilized frequency-inverse document along with the frequency with Machine Learning and CNN in order to train the model with clustering each into labeled clusters and establishing comparative results through pooling and more convolutions, strict criteria values will be developed as Neural Networks is used [11],[12], [13].

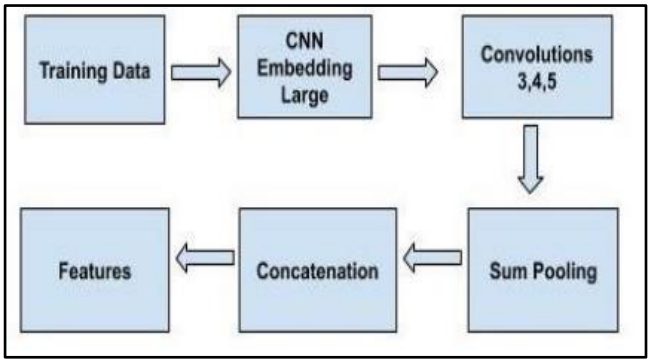


Fig 2.2: RRS using CNN

PROPOSED METHODOLOGY

3.1 App Development

Hence through developing a software aggregate application such as Swiggy/OYO/Uber, this platform can connect an Indian youth to train her/himself culturally (as India's cultural participation rate in national competitions is low and will also save lakhs of people from taking the wrong career path and ending up unemployed/poor). If the youth of India got to do what they are naturally talented or passionate about, the nation's economic/social crisis would not have existed.

This app will be able to provide:

1. Categories for different sports, music, other cultural opportunities for students to post their videos of their talents.
2. An organized unified user profile system to maintain their personal details.
3. Connection of students under 30 with fewer resources to Indian Cultural Institutions to arrange direct interaction where the fee structure and duration will be decided according to the occupational details provided by the user.

Novelty in the Recruiter Recommendation System:

Each candidate's response will be taken by the application and processed through the recruiter recommendation system and as per the scholarship criteria that is fixed, the scores and rank will be evaluated, this novel solution is created to eradicate the hindrances of finances and the obstacle of fee payment to the academies or Creatives and Sports communities to take the candidate in, this will be embedded in the application's dashboard as a feature.

Following are the competitive benefits:

1. Naukri.com, LinkedIn, and many other employment platforms, this app will focus only on the cultural aspect for talented youth to become employed as Leaders, Sportsmen, Musicians in various Government ministries [14].
2. NTC, Edmundo, Coach Up are some of the 4+ stars rated Sports training apps that only provide recorded training videos only if paid for premium membership, the apps don't provide Government approved institutions as well, some are not even allowed access for Indians [15].
3. Yousician, VoCo Vocal Coach, Vox Tools are a few Vocals and Musical Instrument training applications that let the programmed software check the user's vocal pitch and melody without any live feedback from an experienced professional which also requires the user to have a premium membership to access lessons [16].

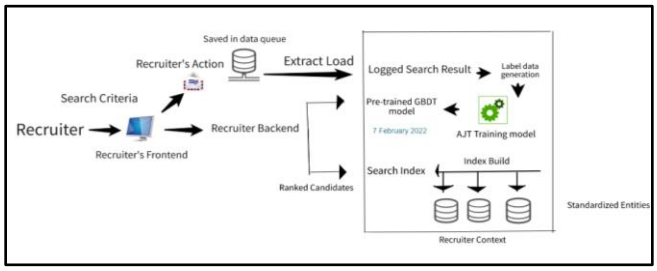


Fig 3.1.1 Architecture- Recruiter Overview

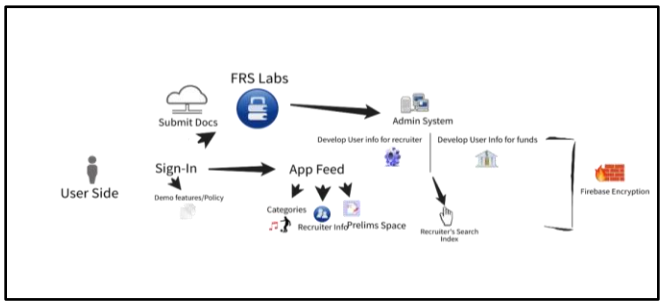


Fig 3.1.2 Architecture- User Overview

3.2 Why Gradient Boosted Decision Tree?

Since linear regression algorithms, K-means algorithm, and apriori are comparatively seamless to interpret and decode they cannot find non-linear correlations in large datasets like LinkedIn. Hence Gradient Boosted Decision Tree algorithm is used to concatenate various models so as to align the eligibility criteria given by the job providers and develop tree structures. Besides having a larger hypothesis space where GBDT is primarily used it also includes a few more perks, such as better processing with feature collinearity, determining vast features from others that come with different ranges, missing feature values, etc [18],[19].

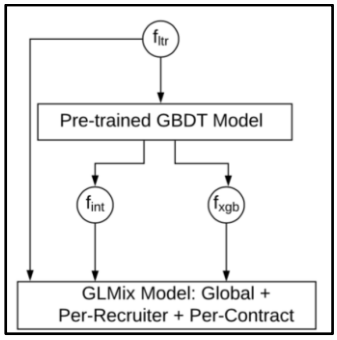


Fig 3.2.1 GDBT for RRS

GBDT alone gave some significant benefits over linear regression, but it also failed to address certain critical search-related issues. Searches for dentists, for example, returned people with software engineering titles since the search models prioritized job-seekers. Pairwise optimizing is a technique that creates features. Essentially, this method adds a pairwise ranking objective to GBDT, allowing candidates to be compared inside the same framework. and evaluate which candidate better fits the current search context despite it being heavily criteria-based [20].

3.2.1 Proposed Algorithm

Given a candidate recommendation to a recruiter by the platform's admin system that is specially selected using IDs that is called using (r-request, c-context, re-recruiuter, ca-candidated, co-contract), an AJT mix training procedure with the scores developed in the GBDT tree features using response values, and the following formula:

д(A(response values and variables) | {z} Value and Variable Reader) = βдlobal · mixed response values| {z } level variable + βr e · f ca mem | {z }

Where A(response values and variables) is the sum formula of the mixed and non-fixed values for attributes to field the categories and other significant factors from the dataset that would require to be processed in the match-score formula, these are global variables for the engine to recognize and adapt, when needed from the admin side.

д(A(response values)) = lд A(response values and variables) 1−A(r,c,r e,ca,co), the ideal function for logit, the terms will form the variables needed for the scoring in GBDT and LTR to make the candidates list stand out using A(r,c,re,ca,co) and hence a fixed effect model will be generated for the rest of the query processing to follow, business intelligence principles : query intelligence, personalization and relevance strategy will be implemented and trained as values and features in the AJT Mix Model, that will capture the re recruiter values and generate more details with the contract co priorities as well. Hence the set containing the variables the learning to rank feature tree will generate specific vectors for each candidate.

Table 1: Sample Clusters -Interpretation

ID	Skills	Interpretation
1	Realistic, Constructive drawing	Contemporary Painter
2	Thrust, Foot Hold	Kabadi Player
3	Orchestral Experience	Music Instructor

Table 2: Sample Clusters - Personalization

ID	Location	Documents
1	Trichy, India	Parents Qualification, Identity Verification
2	Theni, India	Talent Certificates

In the below equation, a "match score" will be generated where cm is the candidate's match aligning with the recruiter's details ™ and wtn,wcm are binary vectors and intent clusters to specify properties that we can give as exceptions.

$$matchScore(c_m|t_n) = \vec{w}_{t_n} \bullet \vec{w}_{c_m} = \sum_i w_{t_n,i} \cdot w_{c_m,i} ,$$

The AJT mix model + recruiter response + contract operation with features for interaction values will be processed through the following formula and then the above matchScore algorithm will be substituted as AJT will train the score of the GBDT score using the parameters called.

д(A(response values and variables) | {z} Value and Variable Reader) = βдlobal · mixed response values| {z } level variable + βr e · f ca mem | {z }

where,

•д being a quantile function in the logit link.

•Variables r for request, re for recruiter, c for context, ca for candidates and co for contract.

•β being a global vector pairing variable.

•βr being a model vector for the variable re, recruiter.

•βco being the model vector for the variable co for contracts.

•fall = fl t r Đ fxдb Đ font representing the rate of learnings done with a fixed value of 0.7 as scores processed by the gradient boosted decision tree may vary from 0.7 onwards, a fixed value is given to call the threshold of a beginning score as other factors may affect the "fall" variable and set non-linear models difficult to define, hence this font sets a interaction score that we are able to modify.

Hence this score will be a modifiable LTR valid value formed using Gradient Boosted Decision Tree formed by different values (such as from 0.7) and consequently linking an intertwined role encrypted into an individual set.

This technique of using the Learning Rate algorithm is found best working with Boosting algorithm, gradient boosted decision tree allows clustering and tree formation of non-linear data and increased efficiency of being trained and tested with multiple datasets and mix models and hence declaring the gradient features and attributes is represented mathematically as follows:

$$p(D|\alpha, \beta) = \prod_{d=1}^{M} \int_{\theta_d} p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn}, \beta) \right) \mathrm{d}\theta_d .$$

Where,

- α- corpus parameter
- β - corpus parameter
- θd being the parameter to open and read documents offline and online.
- zdn being a word level parameter along with wdn.

3.3 AJT Mix Model

This proposed mix model architecture seeks to combine the learning-to-rank algorithm that is to be embedded with the interaction features and clustered to form GBDT model scores. Once the model scores are generated, they are passed on to to merge and be trained with the AJT mix model that is stored in the model store. By using an indiscriminate and unpredictable alteration using tree features and interactions and Gradient Boosted Decision Tree model scores, combining with our AJT mix-model, it can prioritize and call specific functions to model the gradient scores again, train and include the **candidate's test submitted**, **location** and **societal qualifications** and then stored in the admin's database to verify.

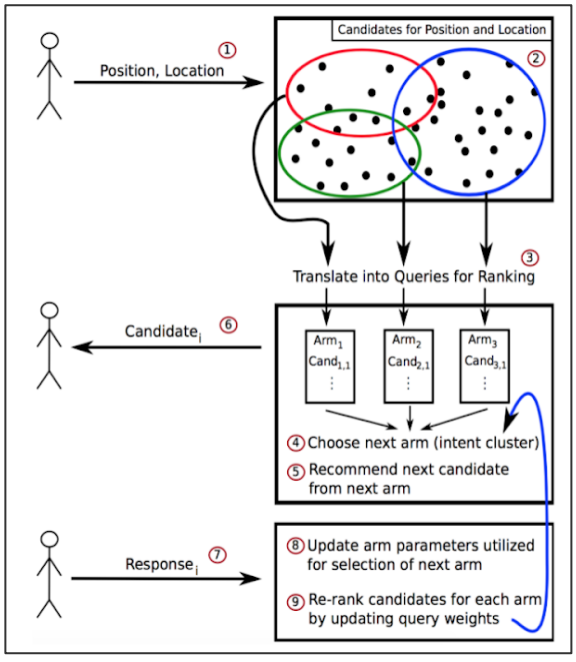


Fig 3.2.1 GDBT for RRS

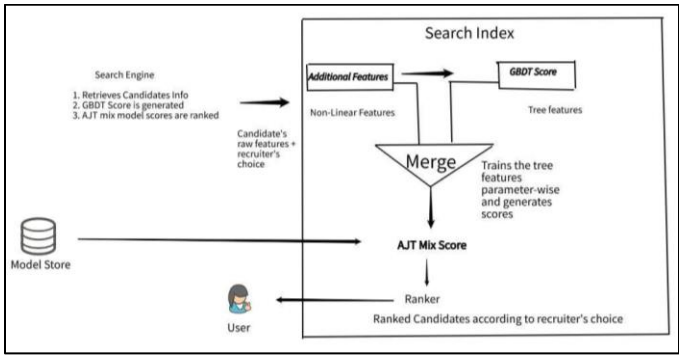


Fig 3.2.2 Functional Architecture

3.4 Algorithm for GDBT and AJT Mix Model

STEP 1: Unclassified data undergo query processing and labeling.

STEP 2: Classifies the entities of the datasets provided.

STEP 3: Gradient values are given as exceptions among other attributes.

STEP 4: Entities are being differentiated into trees and represented as clusters for training the data.

STEP 5: Sending search query to generate first level Gradient Boosted Decision Tree score and returning top candidates with their scores.

STEP 6: AJT performs the second level scoring for the returned results using specified raw entities.

STEP 7: Re-ranking the candidates based on AJT scoring

STEP 8: Returning the list of ranked candidates.

3.5 Pseudocode for Gradient Boosted Decision Tree with AJT Mix Model

```
IF objective EQUALS "cross_entropy":
        SET self. objective TO CrossEntropy()
    ELSEIF isinstance(objective, Objective):
     SET self. objective TO objective
    ELSE:
        logger. warning(type(objective))
        raise ValueError("Only support
`cross_entropy`. Actually: {}". format(objective))
    IF loss EQUALS "logistic":
        SET self. loss TO logistic_loss
    ELSEIF hasattr(loss, "_call_"):
        SET self. loss TO loss
  ELSE:
        raise ValueError("arg `loss` is only
supported `logistic`. ")
    SET self. activate TO self. objective. activate
    SET self. trees TO [] # type: List[Node]
    SET self. max_leaves TO max_leaves
    SET self. max_depth TO max_depth
    SET self. gamma TO gamma
    SET self. num_iter TO num_iter
    SET self. eta TO eta
    SET self. reg_lambda TO reg_lambda
    SET self. subsample TO subsample
    SET self. subsample_freq TO subsample_freq
    SET self. colsample_bytree TO colsample_bytree
    SET self. training_loss TO None
    SET self. validation_loss TO None
    # f to save predictions
    SET self. f TO None
  DEFINE FUNCTION fit(self,
        x: np. ndarray,
        t: np. ndarray,
        SET validation_data: Tuple[np. ndarray, np.
 ndarray] TO None, verbose=1,
 random_seed=1):#F        """
 SET :p aram np.ndarray x: feature numpy array.
shape TO (n_samples, n_features)
 SET :p aram np.ndarray t: numpy array of the
target variable. shape TO (n_samples, )
    :param [np.ndarray, np.ndarray] | None validation_data:
 validation data consisting of (x, y)
 When anything other than None is given, validation
loss is calculated using this data FOR each
iteration.
    :param int verbose:
    :RETURN:
    """
    IF verbose >= 2:
      logger. setLevel("DEBUG")
    ELSEIF verbose >= 1:
      logger. setLevel("INFO")
    ELSE:
      logger. setLevel('WARNING')
    IF len(x. shape) EQUALS 1:
      SET x TO x. reshape(-1, 1)
    assert len(x) EQUALS len(t)
    SET state TO np. random.
RandomState(seed=random_seed)
    SET n_features TO x. shape[1]
    SET n_data TO x. shape[0]
    SET n_select_columns TO int(n_features * self.
colsample_bytree)
    SET n_select_data TO int(n_data * self. subsample)
    # 'f' initialized with zero vector
    SET self. f TO np. zeros_like(t, dtype=np. float)
   SET self. training_loss TO []
    IF validation_data is not None:
      SET self. validation_loss TO []
    SET use_idx TO range(0, n_data)
    FOR i IN range(self. num_iter):
      logger. info('start build new Tree')
      # 1. choice data FOR each 'subsample_freq'
      IF i % self. subsample_freq EQUALS 0:
        logger. debug('update use data')
        SET use_idx TO state. choice(range(0,
 n_data), size=n_select_data, replace=False )
      # 2. choice the data you want to use
      SET use_columns TO state. choice(range(0,
n_features), size=n_select_columns, replace=
False)
      logger. debug('use column index: {}'.
```

```
 format(','. join([str(i) FOR i IN use_columns])))
        # calculate slope and hessian with the
 previous predicted value and the desired value
        SET X_train TO x[use_idx]
        SET t_train TO t[use_idx]
        SET grad, hess TO self. objective(self.
f[use_idx], t_train)
        SET root_node TO Node(x=X_train,
t=t_train, grad= grad, hess=hess, lam=self.
reg_lambda, use_columns=use_columns)
        # repeat UNTIL max_leaves is reached
        FOR leave IN range(self. max_leaves):
           SET best_gain, best_node TO root_node.
 calculate_best_split(self. max_depth)
           # when there is no best_node, it is time
when there is no way to divide, so it ends
           IF best_node is None:
              logger. info('best node is None. Stop build node.')
              break
           # end when gain does not reach the set value
'gamma'
           IF best_gain < self. gamma:
              logger. info(f'best gain {best_gain:.3e}
 below gamma {self. gamma:.3e}. stop build node.')
              break
           ELSE:
              logger. info(f'build new node
 {best_node} gain={best_gain:.4f}')
              best_node. build()
        ELSE:
           logger. info(f'reach to {self. max_leaves}
 nodes. stop build node.')
        self. trees. append(root_node)
        # make this prediction across the data
        SET f_i TO root_node. predict(x)
        self. f += self. eta * f_i
        SET train_loss TO self. _current_train_loss(t)
        logger. info('=' * 30)
        logger. info('end tree iteration')
        logger. info('iterate:{0}\tloss:{1:.2e}'.
format(i, train_loss))
        IF len(self. training_loss) > 0:
           SET diff TO self. training_loss[-1] - train_loss
           logger. info(f'(improve: {diff:.3e})')
        self. training_loss. append(train_loss)
        IF validation_data is not None:
           SET valid_x, valid_t TO validation_data
           SET pred TO self. predict(valid_x)
           SET pred_loss TO self. loss(pred, valid_t). mean()
           self. validation_loss. append(pred_loss)
           logger.          info('valid          loss:\t{0:.3e}'.
format(pred_loss))
     RETURN self
  DEFINE FUNCTION _current_train_loss(self, t)
-> float:#F
     """
 calculating the value of a loss function during
 learning
     :param np.ndarray t: target values
     :RETURN: loss values
     :rtype: float
     """
     SET a TO self. activate(self. f)
     SET loss TO self. loss(a, t)
     RETURN loss. mean()
  DEFINE FUNCTION predict(self, x,
use_trees=None) -> np. ndarray:#F      """
     :param np.ndarray x:
     :param None | int use_trees:
 the number of trees used FOR prediction. when it is
larger than the number of trees that exist,
 predict using all trees when negative values are set
     :RETURN:
     """
 IF use_trees and 1 <= use_trees <= len(self. trees):
        SET nodes TO self. trees[:use_trees]
     ELSE:
        SET nodes TO self. trees
     SET a TO np. zeros_like(x[:, 0])
     FOR i, tree IN enumerate(nodes):
       a += self. eta * tree. predict(x)
     SET pred TO self. activate(a)
     RETURN pred
  DEFINE FUNCTION show_network(self) ->
dict:#F
     """
 dict outputs a learned tree with a nice feeling.
     Returns:
     """
     SET data TO {}
     SET tree_data TO []
     FOR i, tree IN enumerate(self. trees):
        SET d_i TO tree. show_network()
        tree_data. append({
           'index': i,
           'data': d_i
        })
     SET data['trees'] TO tree_data
     RETURN data
  DEFINE FUNCTION feature_importance(self,
 type='gain') -> dict:#F
     """
 calculating feature importance
     Args:
        type: importance type. `'"gain"'` or `'"split"'`.
     Returns:
 importance dict.key is index of feature.
     """
     SET data TO defaultdict(float)
     FOR t IN self. trees:
        SET d_i TO t. feature_importance(type)
        IF d_i is None:
           continue
        FOR k, v IN d_i. items():
           data[k] += v
   RETURN data
```

3.6 Screening and Verification

To keep the screening and verification simple, live and secure, eliminate risks, and remain compliant, FRS labs has featured a plugin for an assisted Video KYC including identity checks and fraud detection [21].

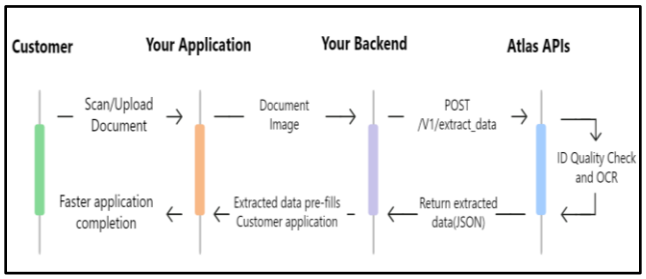


Fig 3.3.1 Assisted KYC Flow, FRS Labs

With a flexible deployment and minimal integration, the application can accommodate user documents on APK, iOS, and android, as their Aldus dashboard provides capturing face, ID, and geolocation of the customer. And as part of verification, the ID is verified against issuing authority, the face is matched against Aadhaar and PAN, geolocation is verified to be within India, and liveness checks ensure that the video call is live authentic [22].

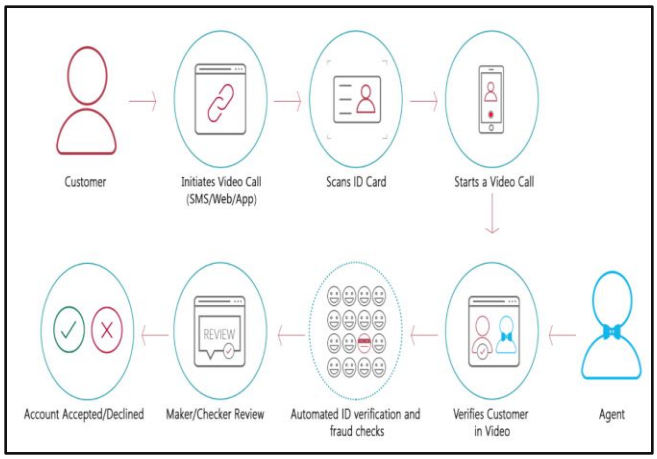


Fig 3.2.1 Video KYC Flow, FRS Labs

3.7 Data Privacy and Security

We use Firebase encryption to protect the app data with the help of Firebase, before any data is written to disk, Firestore encrypts it automatically. No separate processes required to specially use the services provided in the respective environments or engage in a set-up to newly configure but when legitimate personnel read the information, it is instantaneously open to be read confidentially. Since this type of encryption is on the side of the server, Google can automate instead of the user manually performing it, including tight key access controls and auditing, by utilizing a similar key encryption to manage and store, encrypt and decrypt as well. Every encryption key uses individual master keys that will follow a revolving role each time the user logs on and off that makes use of a 256-bit AES- Advanced Encryption Standard for each firestore key [23].

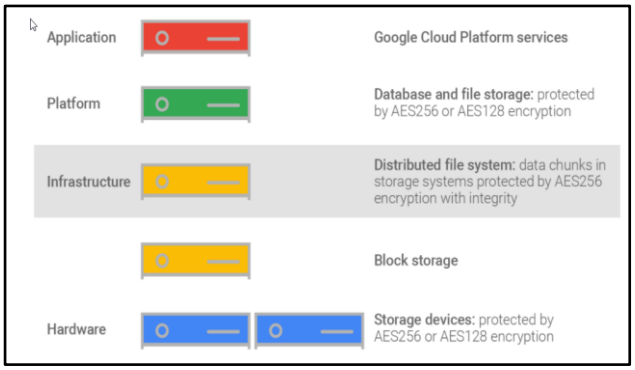


Fig 3.3.1 Google key O-Auth

This type of google authentication is “O-Auth”, completely open in both the client and in the side if the server as well to manage and encrypt on both the sides and then send it to firestore to be stored hence encryption here occurs twice with our personal firestore keys and then with the keys Google provides [24].

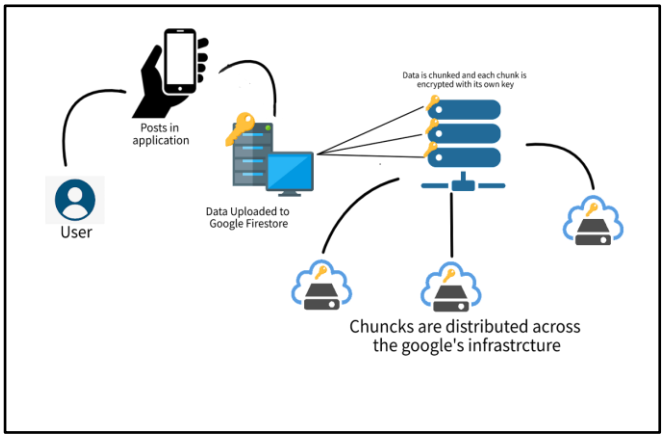


Fig 3.7.1: Architecture - Authentication

3.8 Algorithm for Authentication

STEP 1: Checking for saved passwords in the device or connecting to other apps via OAuth login method

STEP 2: Logging in via email and password method

STEP 3: Creating a new account if the account is not available in the database

STEP 4: Create account using Oauth method by linking with other apps or create using email and password

STEP 5: After completion storing password and logging in

3.9 Pseudo Code for Authentication

```
Adding dependencies{
        Import BoM(Bill of materials) fro firebase platform
        Declaring dependencies for firebase auth library
}
```

```
creating auth function{
        Initializing firebase Auth
}
creating check function{
        checking whether A user is already logged in
        return starting page if loggedin
}
function AuthComplete{
        if
                login successful
                return current user
                show the starting page
        else
                Display message "Authentication Failed"
}
```

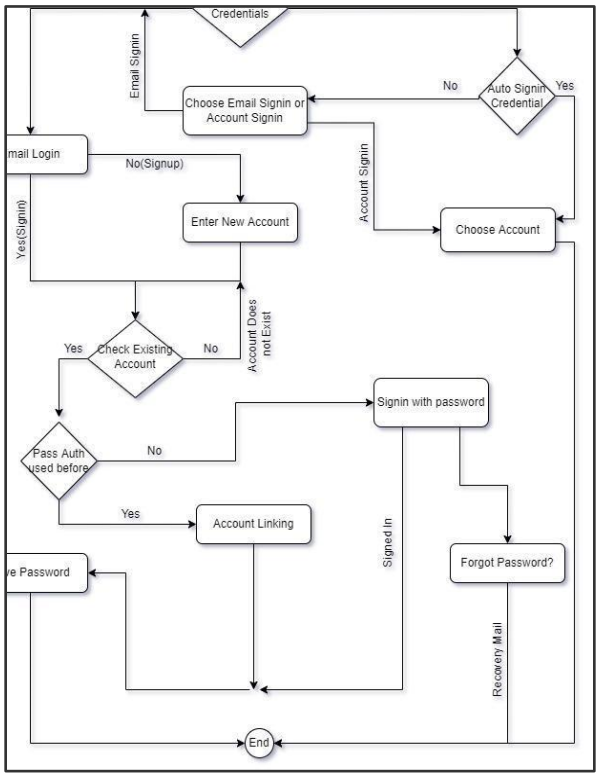
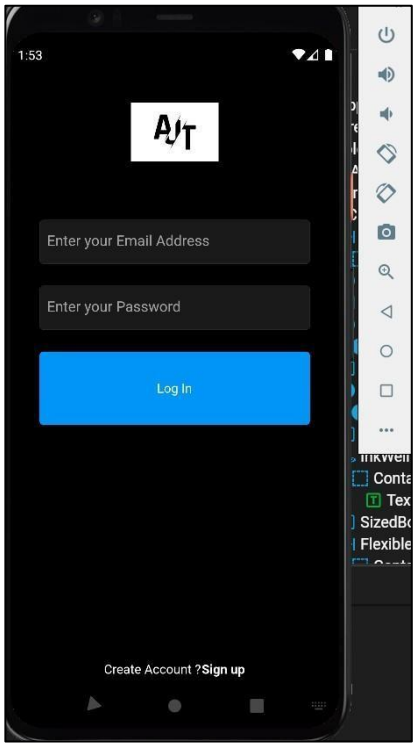


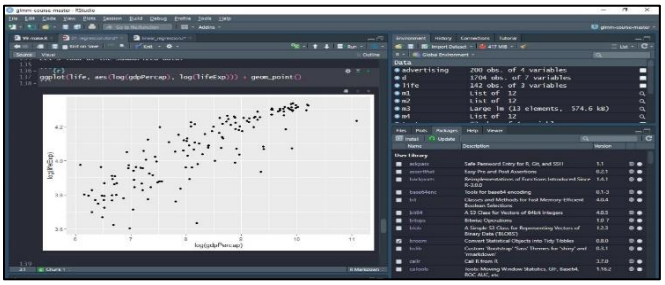
Fig 3.7.1 Authentication Flowchart

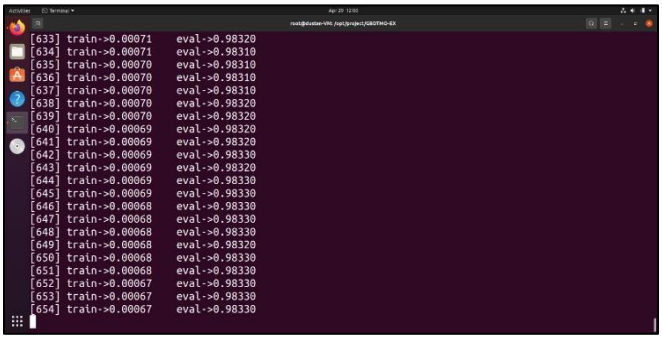
SIMULATION AND ANALYSIS

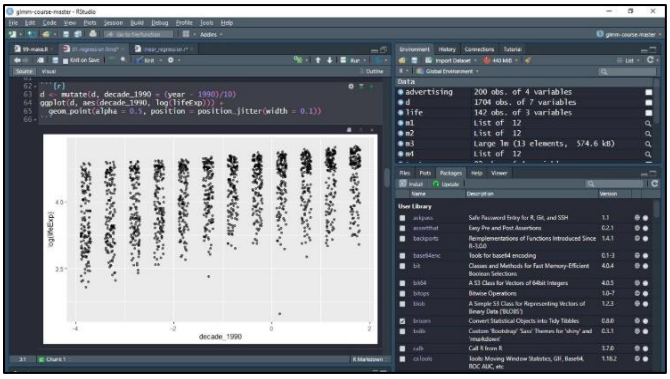
The following are the results from the simulator run on Android Studio using python, the function current_train_loss, reads the documents and segrates them into clusters, the function show_network labels the unstructured clusters into forming trees according to the categories or attributes. A sign-up page including a dashboard to navigate to each candidate's unified user profiles and find criteria for applying for scholarships, documents uploading sections, community of other users, search engine for various categories of Creatives and Sports jobs, policy and privacy maintenance and a chatbot.



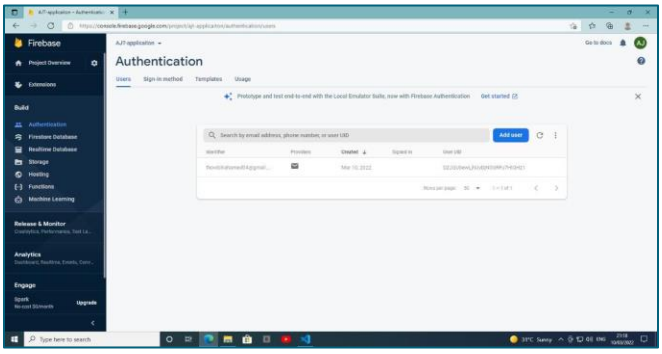
The AJT Mix Model is simulated in matlab and mathworks and the algorithm is integrated into the Gradient Boosted Decision Tree code and gives Gradient Features namely for the Mix Model to train the algorithm namely, Talent, Location and Societal Qualifications. Using the original GBDT model scores in the engine and through a query transformation and hence combining with our AJT mix-model using matlab and scripring from the original GLM-Mix Model, it can deliver recruiter-level and contract-level personalization to prioritize the **candidate's test submitted**, **location** and **societal qualifications** and then stored in the admin's database to verify.







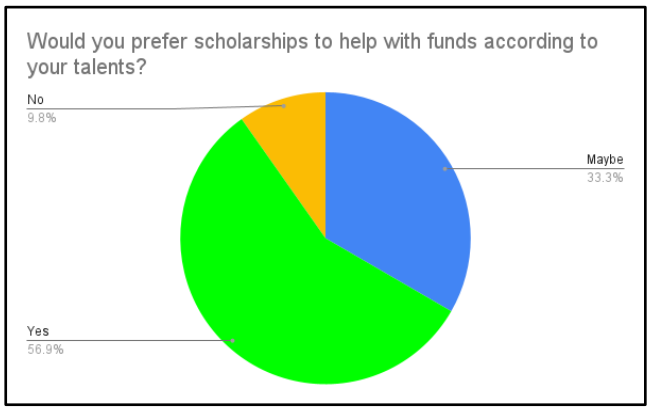
To store and encrypt data, we used firestore's Authcomplete and Authfunction that encrypts the user data with AES-256 encryption. We connected it to Google apps for integrating Open authentication and Single Sign On.



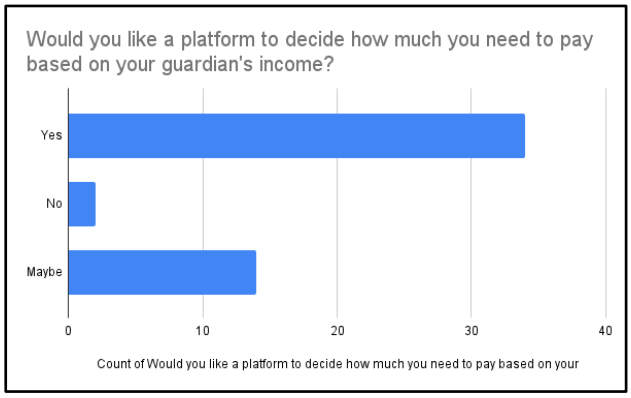
RESULTS - GRAPHS

A student survey was taken as a dataset containing responses from ages 18-26 answering the following questions to analyze the interests of future pursuers of the application in real time - 2022 and the responses were overwhelmingly positive and the scholarship initiation and the Creatives & Sports employment is observed as the need of the hour from the below graphs taken from Google Forms and Survey.

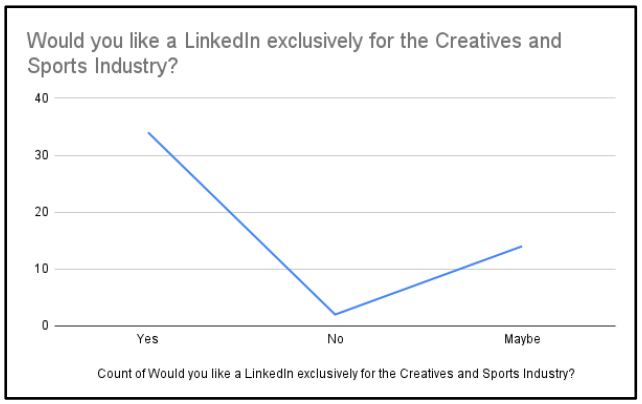
1. Would you prefer scholarships to help with funds according to your talents?



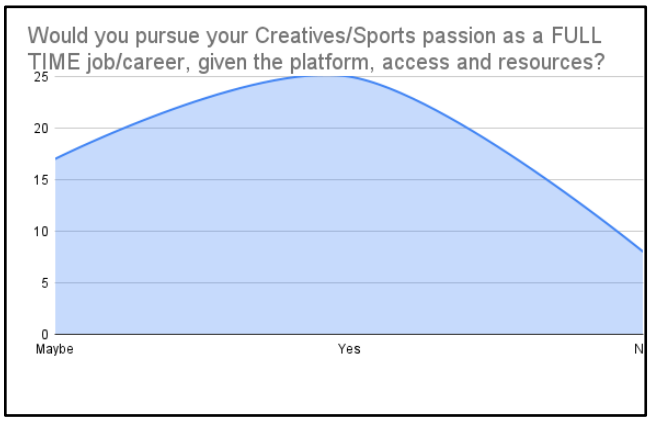
2. Would you like a platform to decide how much you need to pay based on your guardian's income?



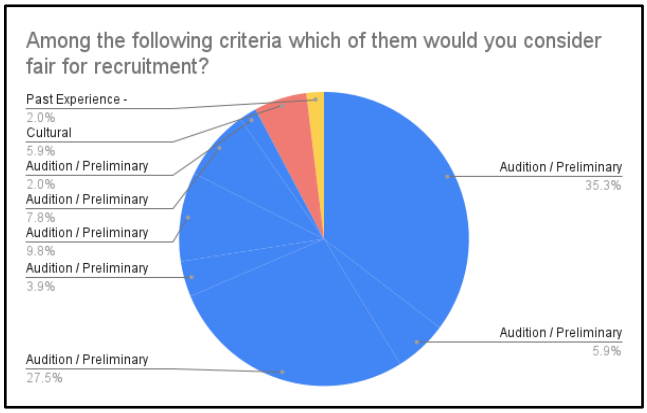
3. Would you like a job portal exclusively for the Creatives and Sports Industry?



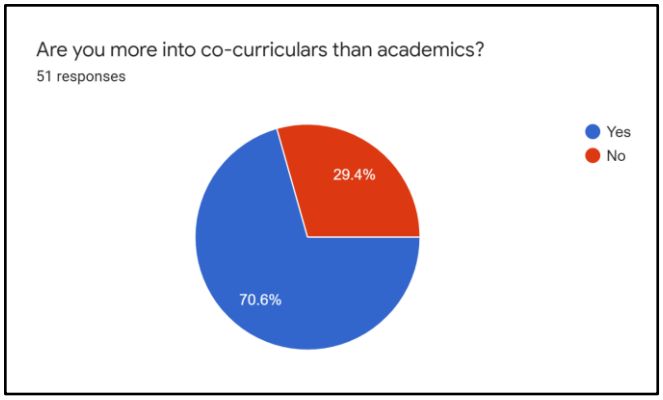
4. Would you pursue your Creatives/Sports passion as a FULL TIME job/career, given the platform, access and resources?



5. Which criteria would you consider fair for recruitment?



6. Are you more interested in co-curriculars than academics?



Conclusion & Future Works

The application along with the respective stakeholders requires internal management and survey of rural datasets to avoid technological hassles according to various locations, channelizing and outsourcing the application for the academies and NGOs to fund the users is certainly a prime factor of the wings of the project. Considering its economic growth and rising literacy levels, India's labor market strategy still faces a big issue in the form of youth unemployment. In India, young job seekers account for 49% of the net unemployed. The lack of any type of social security system, as well as suitable and adequate outlets for vocational training, is India's biggest concern when it comes to youth unemployment.

In a world where there are so many chances for economic progress, India needs increased national and international cooperation among labor market actors. Hence not only is the impact of cultural employment a boon to India’s graduate employment but also can leverage the global economic ratio at a broad spectrum.

References

Shahriar, Mohammad Shibli, Islam, K.M. Anwarul, Zayed, Nurul Mohammad, Hasan, K.B.M. Rajibul, Raisa, Tahsin Sharmila, The Impact of COVID-19 on Bangladesh's Economy: A Focus on Graduate Employability, The Journal of Asian Finance, Economics and Business, 2020. https://doi.org/10.13106/jafeb.2021.vol8.no3.1395

Zarrin Tasnim, F.M. Javed Mehedi Shamrat, Shaikh Muhammad Allyar, Implementation of an Intelligent Online Job Portal Using Machine Learning Algorithms, 2nd International Conference On Emerging Technologies In Data Mining And Information Security (Iemis 2020). http://dx.doi.org/10.1007/978-981-15-9774-9_55

Guilherme Pinho, João Arantes, Tiago Marques Frederico Branco, Manuel Au-Yong-Oliveira, The Use of LinkedIn for ICT Recruitment, 2021. https://link.springer.com/conference/worldcist

Yanhui Ding, Yongxin Zhang Lin Li, Weizhi Xu, Hu Wang, 48th International Conference on Information Technology in Medicine and Education, A Reciprocal Recommender System for Graduates’ Recruitment, 2016. 10.1109/IPEC49694.2020.9115188

Hao Wu, Qian Liu, Zhifang Zhang, 2020 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Analysis of University Students Employment Recommendation System Based on Apriori Algorithm, 2020. 10.1109/IPEC49694.2020.9115188

Okta Purnama Rahadian, Mira Hidayati, Martin Sujono, Abba Suganda Girsang, Sani Muhammad Isa, IEEE, Business Intelligence For a Digital Music Content Provider, 2018. https://doi.org/10.1109/INAPR.2018.8627051

Ashif Mohamed, Wickram Bagawathinathan, Usama Iqbal, Anuradha Jayakody, Shahik Shamrath,Smart Talents Recruiter – Resume Ranking and Recommendation System, IEEE, 2018. https://doi.org/10.1109/ICIAFS.2018.8913392

Randy Joy M. Ventayen, Amy P. Balcita, Josephine S. Dela Cruz, UNSIGNED Music Lounge: Promoting Unsigned Artist Thru Online Music Distribution Portal, IEEE, 2019. https://doi.org/10.1109/HNICEM.2018.8666323

Mr. Ramraj S, Dr.V. Sivakumar, Kaushik Ramnath G, IEEE International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE-2020), July 29-31, 2020, Odisha, India, Real-Time Resume Classification System Using LinkedIn Profile Descriptions, 2020. https://doi.org/10.1109/CISPSSE49931.2020.9212209

Shayma Boukari, Sondes Fayech, Rim Faiz, Authorized licensed use limited to: University of Saskatchewan. on July 05, at 22:21: 37 UTC from IEEE Xplore Huntalent: A candidates recommendation system for automatic recruitment via LinkedIn 2021. http://odell.radonc.med.ufl.edu/papers/Yan_IEEErobotics_04.pdf

Laurence Moroney, Firebase Cloud Messaging, Springer, 2017. https://doi.org/10.1007/978-1-4842-2943-9_9

Zhigao Chen, Wenqian Liang, Xiong Gao, Zhijian Zhou,Meng Wu,

Research on the Accurate Recommendation Management System for Employment of College Graduates on Hadoop,5th International Conference on Big Data and Information Analytics 2019. https://doi.org/10.1109/BigDIA.2019.8802855

F. M. Javed Mehedi Shamrat, Zarrin Tasnim, Imran Mahmud, Ms. Nusrat Jahan, Naimul Islam Nobel, Application Of K-Means Clustering Algorithm To Determine The Density Of Demand Of Different Kinds Of Jobs, International Journal Of Scientific & Technology Research Volume 9, Issue 02, February 2020. http://www.ijstr.org/paper-references.php?ref=IJSTR-0120-29776

Guangshun Zhang, Shiyuan Zhou, Xiaoyun Xia, Serhat Yüksel, Halim Baş, And Hasan Dincer, Strategic Mapping of Youth Unemployment With Interval-Valued Intuitionistic Hesitant Fuzzy DEMATEL Based on 2-Tuple Linguistic Values, Digital Object Identifier 10.1109/ACCESS.2020. https://www.researchgate.net/publication/339022641_Strategic_Mapping_of_Youth_Unemployment_With_Interval-Valued_Intuitionistic_Hesitant_Fuzzy_DEMATEL_Based_on_2-Tuple_Linguistic_Values

Casimiro A. Curbelo Montañez, And William Hurst, A Machine Learning Approach for Detecting Unemployment Using the Smart Metering Infrastructure, Digital Object Identifier 10.1109/ACCESS.2020. https://doi.org/10.1109/ACCESS.2020.2969468

Laurence Moroney, Using Authentication in Firebase, 11 November 2017. https://link.springer.com/chapter/10.1007%2F978-1-4842-2943-9_2#citeas

Yi Wei, Li Pan, Shijun Liu, Lei Wu, And Xiangxu Meng, Drl-Scheduling: An Intelligent QoS-Aware Job Scheduling Framework for Applications in Clouds, Digital Object Identifier 10.1109/ACCESS.2018. https://doi.org/10.1109/ACCESS.2018.2872674

Aymen Hamrouni; Hakim Ghazzai; Turki Alelyani; Yehia Massoud, Low-Complexity Recruitment for Collaborative Mobile Crowdsourcing Using Graph Neural Networks, IEEE Internet of Things Journal (Volume: 9, Issue: 1, Jan.1, 1 2022). https://doi.org/10.1109/JIOT.2021.3086410

Pooja Singh; Lalit Kumar Singh, Reliability and Safety Engineering for Safety-Critical Systems in Computer Science: A Study into the Mismatch Between Higher Education and Employment in Brazil and India, IEEE Transactions on Education (Volume: 64, Issue: 4, Nov. 2021). https://doi.org/10.1109/TE.2021.3057611

Jorge Hochstetter; Jaime Díaz; Mauricio Diéguez; Roberto Espinosa; Jeferson Arango-López, Assessing Transparency in eGovernment Electronic Processes, IEEE Access (Volumne:10, 2021). https://doi.org/10.1109/ACCESS.2021.3137799

Two Derivative Algorithms of Gradient Boosting Decision Tree for Silicon Content in Blast Furnace System Prediction, IEEE Access: Volume: 8, 2020. https://doi.org/10.1109/ACCESS.2020.3034566

Zhendong Zhang; Cheolkon Jung GBDT-MO: Gradient-Boosted Decision Trees for Multiple Outputs, IEEE Transactions on Neural Networks and Learning Systems (Volume: 32, Issue: 7, July 2021). https://doi.org/10.1109/TNNLS.2020.3009776

Yabin Shao; Chuanlong Wang, HIBoosting: A Recommender System Based on a Gradient Boosting Machine, IEEE Access: Volume: 7, 2019. https://doi.org/10.1109/ACCESS.2019.2956342

The Economic And Employment Impact Of The Arts And Music Industry, Hearing Held In Washington, Dc, Serial No. 111–12, 2009. https://www.gpo.gov/fdsys/pkg/CHRG-111hhrg48055/pdf/CHRG-111hhrg48055.pdf

Ibrahim, S. (2022). Commutativity of high-order linear time-varying systems. Advances in Differential Equations and Control Processes, 27, 73-83.