ISSN No:-2456-2165

Automation of Candidate Hiring System Using Machine Learning

Joel Silas[1]
Pranali Udhan[2]
Prajakta Dahiphale[3]
Vaibhav Parkale[4]
Dr. Poonam Lambhate[5]
[1,2,3,4]Student, Department of Computer Engineering, JSPM's Jayawantrao Sawant College of Engineering, Pune
[5]Guide, Department of Computer Engineering, JSPM's Jayawantrao Sawant College of Engineering, Pune

Abstract:- The placement season is a very overwhelming duration for every job aspirant. This phase also involves anxiety due to the transition from academic to professional life. Not only the candidates but also the companies consider this period to have immense significance. They desire to hire the best candidates with the required skillset, mental and physical ability and commendable personality. Deserving and skilled candidates who will definitely be an asset to the organisation must be hired. In today's rat race not only qualifications but also the personality and soft skills of the candidate are considered to decide whether the candidate is eligible or not. The candidate's social media profile will help analyse his/her personality, thoughts and ideologies. Hence, a final eligibility score for each candidate will help the company decide that whether the candidate must be hired or not.

***Keywords*:-** *Ideal Candidate, Hiring, Personality Prediction, Emotions, Ideologies, Eligibility, Employment, Interview.*

I. INTRODUCTION

The academic and educational background will aid in determining a candidate's skill set and help shortlist him or her as a good fit for the position. The quality of the candidate's contribution to the organization will, however, depend on his behavior, attitude, mindset, ethics, and morale.

Cognitive-based Sentiment Analysis has received a lot of interest recently, with a focus on the automatic detection of user behavior, such as personality traits, based on online social media text. It is possible to establish the efficacy of the suggested model for eight significant personality qualities including introversion-extroversion, intuition-sensing, thinking-feeling, and judging-perceiving.

The datasets' evaluations show that the suggested model can successfully categorize the user's personality attributes. Organizations can decide on the hiring of personnel in an effective manner using the information gleaned from this research. Additionally, this research can contribute to the definition of procedures for the choice, management, and improvement of human resources, numerous policies, and service goods in a variety of industries, such as the military, government, or IT occupations.

➢ *Objectives*

- Candidate Screening
- Hiring deserving candidates
- Providing employment to deserving candidates
- Personality Analysis
- Automation of administration
- Decision making and feedback mechanism

Achievement of the above objectives indirectly helps attain the Sustainable Development Goal (SDG) 1 – 'End poverty in all its forms', by providing employability to deserving candidates

II. LITERATURE REVIEW

Algorithms for extracting embedded on textualized words from textual data for personality prediction systems have been researched[1,2]. It makes sense to conduct study on human behavior now that there is access to high-dimensional data on human emotions. Psychologists have been able to study human psychology and gain valuable insights from personality tests thanks to data gathered from our daily actions. With the aid of row and feature sampling approaches, machine learning models are a blessing for researchers since they can be used to learn extremely complex relationships and target their generalizability and robustness. Research and analysis can be transformed into personality psychology through it. Algorithms can operate on large datasets with several attributes without encountering collinearity problems by performing regularization through Ridge and Lasso Regression. In datasets, machine learning (ML) algorithms are extremely powerful and efficient at identifying patterns that people are unable to. The use of these ML models can result in automated personality assessments. On social media, people discuss and express their thoughts, feelings, likes, and opinions, revealing personality traits in the process. Big Five (OCEAN) personality traits[4,5] are determined using machine learning models using a variety of data elements. Industries frequently employ supervised machine learning techniques like Support Vector Machine and Naive Bayes to predict personality traits. Researchers have recently begun to develop unsupervised

learning techniques to find additional psychological structures in digital data.

III. PERSONALITY PREDICTION USING USER BEHAVIOR ON FACEBOOK

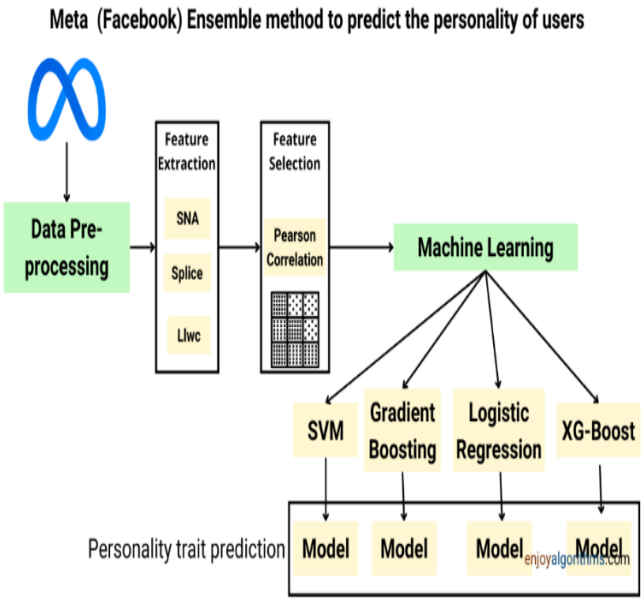

Fig 1 Personality Prediction Over Facebook Data

Social networking sites like Facebook[3], Twitter, Instagram, and Linkedin have risen to the top of the list of popular websites for internet users in recent years. Researchers may investigate and comprehend someone's online behaviors, preferences, and personality very well via social network activities. Different personalities are linked to the development of various social connections and interaction styles based on social position. The emergence of social networks has made it simple to infer user personalities from their social interactions. The Big Five model's many attributes and measurements are used to forecast Facebook users' personality traits in the manner shown in the image above.

➢ *Steps of Implementation*

- Data set is visualized
- Data set is preprocessed by handling null values and performing scaling
- Machine Learning models like SVM, gradient boosting, logistic regression can be trained on labelled data
- The performance of the models can be evaluated and the required hyper parameter tuning can be performed
- Now for any new candidate the personality prediction can be done.

IV. PROPOSED IMPLEMENTATION

System implemented is based on B/S architecture. It has two services Web Application as a user interface and Machine Learning for the predictive business logic. To enhance the user experience this can be developed as a single page application following the MVC Architecture.

Initially, the candidate is expected to create an account on this platform. Interviewers will be added by the administrator. The candidate must perform submission of social media profile link. The social media link provided by the candidate will be accessed. Web scrapping techniques will be applied to define a dataset comprising of the posts of the candidate. This will be passed through a series of natural language processing techniques in order to exercise text analytics. TF-IDF Vectorization will be applied to the processed posts to obtain their vectorised numerical equivalents[6,7] upon which classification algorithms like CatBoost, XgBoost, Support Vector Machine, Decision Tree Classifier can be applied to determine the class of the personality of the candidate. The possible classes of the personality can be a quartet of combinations of 8 important personality traits.

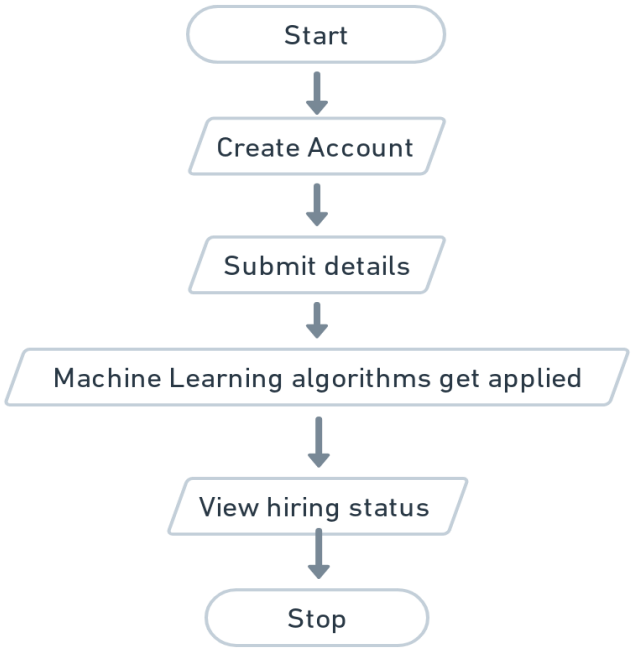

Fig 2 Execution Flowchart

V. CATBOOST ALGORITHM

➢ *Decision Tree:*

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It is a hierarchical structure such that at each node splitting is done based on a decision where the sequence and depth of decisions is based on concepts like impurity and entropy to obtain an optimal height tree ie. a tree incorporating minimum splits.

➢ *Boosting Algorithms:*

Boosting algorithms seek to improve the prediction power by training a sequence of weak models, each compensating the weaknesses of its predecessors. In each iteration, miss-classified data points are identified, their weights are increased and thus, weights of correct points are decreased in a sense so that the next classifier will pay extra attention to get them right.

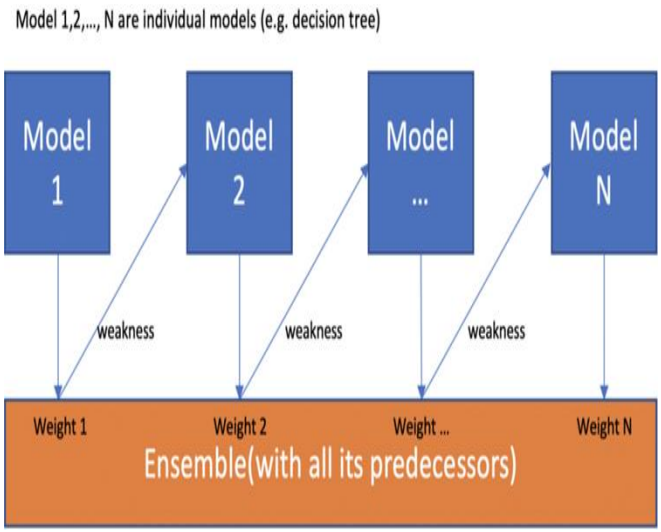

Fig 3 Boosting Ensemble Technique

➢ *Gradient Boosting Algorithms:*

Instead of adjusting weights of data points, Gradient boosting focuses on the difference between the prediction and the ground truth ie. it requires a loss function (expression of this difference called error).

$fit\ estimator\ F^1$
$for\ i\ in\ [1, M]$//M weak estimators
$Loss^i = \sum_{j=1}^{n} \left(Y_j - F^i(X_j)\right)^2$ // loss in i^{th} iteration
$calculate\ neg\ gradient: -\frac{\partial L^i}{\partial X_j} = -\frac{2}{n} * \left(Y_j - F^i(X_j)\right) \forall i$
$Fit\ a\ weak\ estimator\ H^i\ on\ \left(X, \frac{\partial L}{\partial X}\right)$
$//\rho\ changes\ the\ step\ size$
$Prediction: F^m(X) = F^i(X) + \rho * H^i(X) = F^1 + \rho * \sum_{i=1}^{m} H^i(X)$

➢ *CatBoost Algorithm (based on gradient boosted decision trees):*

- *Catboost package can be installed by executing pip install catboost*
- *Additional installations can include ipywidgets for data visualization, widgets extension and so on.*
- *Can be imported into the program as Catboost, CatboostRegressor, CatBoostClassifier and the method staged_predict can be called over the dataset to calculate the results taking into consideration only the trees in the range [0; i)*
- *The goal of training is to select the* model *y, depending on a set of* features *xi, that best solves the given problem (regression, classification, or multiclassification) for any input* object.

- *Preliminary Calculation of Splits:*

Before learning, the possible values of objects are divided into disjoint ranges (*buckets*) delimited by the threshold values (*splits*). The size of the quantization (the number of splits) is determined by the starting parameters (separately for numerical features and numbers obtained as a result of converting categorical features into numerical features). which is done by methods like

✓ MaxLogSum = max $\{\sum_{i=1}^{n} log(weight)\}$

✓ MinEntropy = min $\{\sum_{i=1}^{n} weight \cdot log(weight)\}$

✓ GreedyLogSum = max $\{\sum_{i=1}^{n} log(weight)\}$.. and so on

- *Transforming Categorical Features to Numerical* Features:

Before each split is selected, categorical features are transformed to numerical. This is done using various statistics on combinations of categorical features and numerical features which includes the following stages:

✓ Permutating the set of input objects in a random order.
✓ Converting the label value from a floating point to an integer.

Formula:

Calculating ctr for the i-th bucket ($i \in [0; k]$, creates $k + 1$ buckets):

$ctr_i = \frac{countInClass+prior}{totalCount+1}, where$

✓ *Countinclass is how many times the label value exceeded i for objects with the current categorical feature value. It only counts objects that already have this value calculated (calculations are made in the order of the objects after shuffling).*
✓ TotalCount is the total number of objects (up to the current one) that have a feature value matching the current one.
✓ Prior is a number (constant) defined by the starting parameters.

- *Transforming Text Features to Numerical Features:*

✓ *Loading and Storing Text Features*

Specify the Text column type in the column descriptions file if the dataset is loaded from a file. Use the text_features parameter in the Python package.

✓ *Tokenization*

Each value of a text feature is converted to a string sequence of words by splitting the original string by space.

✓ *Dictionary Creation*

A dictionary is a data structure that collects all values of a text feature, defines the minimum unit of text sequence representation called *token*, and assigns a number to each token.

- Examples: Consider the text: abra cadabra
- Letter — A symbol from the string. Dictionary = {a, b, c, d, r}.

- Word — An element from the sequence of strings. Dictionary = {'abra', 'cadabra'}.

- *Estimating Numerical Features:*

Numerical features are calculated based on the source tokenized which can be calculated by either of the following: Bag of words, top_tokens_count, Naïve Bayes, BM25

- *Training:*

Computed numerical features are passed to the regular CatBoost training algorithm.

- *Choosing the Tree Structure ie. Feature-Split Pairs for a Leaf:*

✓ A list is formed of possible candidates (feature-split pairs) to be assigned to a leaf as the split.
✓ A number of penalty functions are calculated for each object The split with the smallest penalty is selected. The resulting value is assigned to the leaf.
✓ This procedure is repeated for all the leaves

- *Once the training is done, testing / validation dataset can be used to predict the results*

- *Overfitting detector:*

If overfitting occurs, CatBoost can stop the training earlier than the training parameters dictate. For example, it can be stopped before the specified number of trees are built.

Before building each new tree, CatBoost checks the resulting loss change on the validation dataset. The overfit detector is triggered if the $Threshold$ value set in the starting parameters is greater than $CurrentP\,Value$:

$$CurrentP\,Value < Threshold$$

How $CurrentP\,Value$ is calculated from a set of values for the maximizing metric $score[i]$:

1. $ExpectedInc$ is calculated:

$$ExpectedInc = max_{i_1 \le i_2 \le i} 0.99^{i-i_1} \cdot (score[i_2] - score[i_1])$$

2. x is calculated:

$$x = \frac{ExpectedInc[i]}{max_{j \le i} score[j] - score[i]}$$

3. $CurrentP\,Value$ is calculated:

$$CurrentP\,Value = exp\left(-\frac{0.5}{x}\right)$$

- *Inc-To-Dec is an overfitting detection method described as follows:*

- *Score Functions:*

The common approach to solve supervised learning tasks is to minimize the loss function L:

$$L(f(x), y) = \sum_i w_i \cdot l(f(x_i), y_i) + J(f), where$$

- $l(f(x), y)$ is the value of the loss function at the point (x, y)
- w_i is the weight of the i-th object
- $J(f)$ is the regularization term.

For example, these formulas take the following form for linear regression:

- $l(f(x), y) = w_i((\theta, x) - y)^2$ (mean squared error)
- $J(f) = \lambda ||\theta||_{l2}$ (L2 regularization)

Types of score functions

Let's suppose that it is required to add a new tree to the ensemble. A score function is required in order to choose between candidate trees. Given a candidate tree f let a_i denote $f(x_i)$, w_i — the weight of i-th object, and g_i – the corresponding gradient of l. Let's consider the following score functions:

- $L2 = -\sum_i w_i \cdot (a_i - g_i)^2$
- $Cosine = \frac{\sum w_i \cdot a_i \cdot g_i}{\sqrt{\sum w_i a_i^2} \cdot \sqrt{\sum w_i g_i^2}}$

- *Cat Boost Algorithm Benefits:*

✓ By using oblivious decision trees (binary trees in which the same features are utilized to produce left and right splits for each level of the tree), CatBoost limits the number of features split per level to one, which helps to shorten prediction time.
✓ By using ordered target statistics, it successfully handles categorical features.
✓ It is simple to utilize with R and Python libraries.
✓ It may be used effectively with the default parameters, which cuts down on the time required for parameter tuning.

VI. RESULT ANALYSIS

Text analytics will be practiced using a variety of natural language processing algorithms on the textual information of the candidate posts. TF-IDF To establish the class of the candidate's personality, classification algorithms like CatBoost, XgBoost, Support Vector Machine, Logistic Regression, and Decision Tree Classifier can be used on the vectorized numerical equivalents of the processed postings. It has been found that the Decision Tree Classifier provides the least accurate results (50.26%), while the Catboost algorithm provides the best accuracy (67.20%).

➢ *Evaluation Parameters*

The results can be evaluated by the system using precision and recall methods.

- *Precision*

Precision is defined as the ratio of correctly classified positive samples (True Positive) to a total number of classified positive samples (either correctly or incorrectly).

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad \text{..... (1)}$$

- *Recall*

Recall is defined as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad \text{..... (2)}$$

➢ *Where,*

- True Positive = The label is correctly predicted to be positive
- True Negative = The label is correctly predicted to be negative
- False Positive = The label is incorrectly predicted to be positive
- False Negative = The label is incorrectly predicted to be negative

Table 1 Comparative Analysis of the Applied Classification Algorithms

Sr. No.	Model	Test Accuracy
1	CatBoost Classifier	0.672046
2	XGBoost Classifier	0.663977
3	Linear Support Vector classifier	0.662824
4	Support Vector classifier	0.647839
5	Logistic regression	0.628242
6	Decision Tree classifier	0.502594

- *Time and Space Complexity*

✓ Time complexity is defined as the total time taken by the algorithm to get executed and provide the output.

✓ Space complexity is defined as the space occupied by the algorithm in the memory (of the machine) for it to get executed

✓ It is a measure that can be used to compare the algorithms in order to find an optimal algorithm.

✓ It is preferred that an algorithm has lesser space and time complexity. Generally, major focus is towards reducing the time complexity.

✓ These can be represented by Big-Oh (O), Omega (Ω) or Theta (Θ) notation

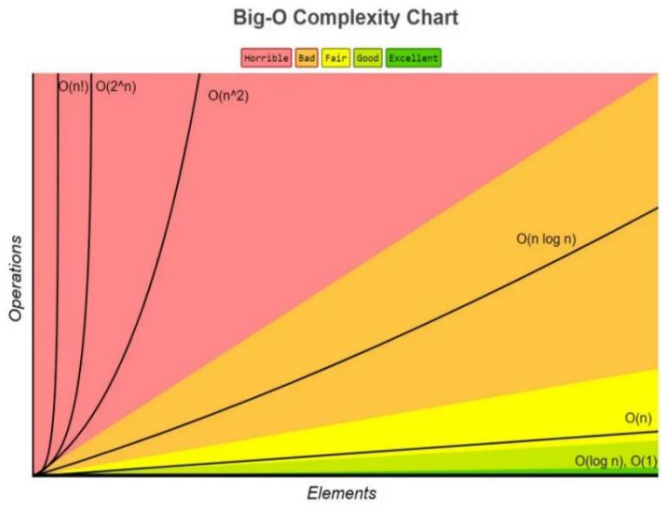

Fig 4 Complexity Comparison

Table 2 Time and Space Complexity

Sr. No.	Algorithm	Time Complexity	Space Complexity
1	Decision Tree Classifier	Training phase	
		O(n*m*log(n))	O(n*m)
		Testing phase	
		O(log(n))	O(n)
2	XgBoost	O(n*m*log(m))	O(n*m)
3	CatBoost	O(n*log(n))	O(n)
4	Logistic Regression	Training phase	
		$O(n*m^2)$	O(m)
		Testing phase	
		O(m)	O(1)
5	Support Vector Machine	O(n*log(n))	O(n)

➢ *Where*

- n = number of data points
- m = number of trees, for boosting algorithms features
- m = number of features, for other algorithms

Based on the comparative study of time and space complexity of the above mentioned machine learning algorithms it is observed that CatBoost Algorithm not only predicts at higher accuracy but also occupies lesser space and executes faster ie. lesser space and time complexity.

VII. CONCLUSION

- We can avoid lengthy psychological tests, by using the social profiles of the candidates to determine their personalities.
- With the help of the system developed the companies will be able to hire candidates not only based on their technical skillset but also by analysing and predicting their health condition and by identifying their personality traits.
- This can not only help the IT industries but also the government and military sector.

REFERENCES

[1]. Text based personality prediction from multiple social media by Christian et al. J
[2]. Research on recognizing required items based on opencv and machine learning by Qingyun Ma1 and Xubin Huang2
[3]. Personality Predictions Based on User Behavior on the Facebook Social Media Platform MICHAEL M. TADESSE , HONGFEI LIN, BO XU AND LIANG YANG
[4]. L. Goldberg, "The Structure of Phenotypic Personality Traits," Am. Psychologist, vol. 48, no. 1, 1993, pp. 26-34. 4. J.W. Pennebaker and L.A.King, "Linguistic Styles: Language Use as an Individual Difference," J. Personality and Social Psychology, vol. 77, no. 6, 1999, pp. 1296-1312
[5]. J. Digman, "Personality Structure: Emergence of the FiveFactor Model," Ann. Rev. Psychology, vol. 41, no. 1, 1990, pp. 417-440.
[6]. L. Goldberg. "The Structure of Phenotypic Personality Traits," Am. Psychologist, vol. 48, no. 1, 1993, pp. 26-34. 4. J.W. Pennebaker and L.A. King, "Linguistic Styles: Language Use as an Individual Difference," J. Personality and Social Psychology, vol. 77, no. 6, 1999, pp. 1296-1312
[7]. J.W. Pennebaker and L.A. King, "Linguistic Styles: Language Use as an Individual Difference," J. Personality and Social Psychology, vol. 77, no. 6, 1999, pp. 1296-1312.
[8]. https://raven.cam.ac.uk/auth/authenticate.html?ver=3&url=https://shib.raven.cam.ac.uk/dp/Authn/RemoteUser&date=20180506T172763ZXdesc-the Raven/Shibboleth