International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 11 | November - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

USING MACHINE LEARNING FOR CAMPUS PLACEMENT PREDICTION

Yuvjeet
Department of Computer Science and Enginnering
Chandigrah University
Mohali, Punjab
artiarorasingh@gmail.com

Rahul Dey
Department of Computer Science and Enginnering
Chandigrah University
Mohali, Punjab
2003rahul.16@gmail.com

Abhishek Chamoli
Department of Computer Science and Enginnering
Chandigrah University
Mohali, Punjab
chamoli.work@gmail.com

Nandini Bhardwaj
Department of Computer Science and Enginnering
Chandigrah University
Mohali, Punjab
nandinibhardwaj2003@gmail.com

Neelamani samal
Department of Computer Science and Enginnering
Chandigrah University
Mohali, Punjab
neelamani.samal@gmail.com

***Abstract*— In the modern world, campus placements play a crucial role in shaping the career trajectories of students and determining the reputation of educational institutions. This study addresses the growing need for accurate and predictive tools in campus placement through the application of machine learning (ML) models. We focus on widely-used ML algorithms that have emerged as powerful tools for predicting fitting technical fields. The research utilizes a comprehensive dataset that incorporates diverse features, including academic performance for technical proficiency. By leveraging these features, this project explores the application of ML algorithms to predict suitable job profiles. By analyzing the academic and technical performance of student's data and identifying relevant features, ML models can learn patterns and relationships that contribute to predict the placement potential of current students, guiding them towards career paths and skill development that align with their strengths and interests.**

Keywords— Machine Learning, Campus Placement, Random Forest Classification, Logistic Regression, Decision Tree, K-Nearest Neighbour Gradient Boosting Classifier, Predictive Analytics.

I. INTRODUCTION

In today's fast-paced technological landscape, the process of campus placements plays a pivotal role in shaping the future of aspiring individuals. The challenge lies not only in securing a placement opportunity but also in aligning one's skills with the specific demands of diverse industries. As the nexus between education and industry grows stronger, the need for accurate, efficient, and fair placement prediction systems becomes increasingly apparent. This research paper delves into the intricate world of campus placement prediction, aiming to provide a comprehensive solution for both students and recruiters. Every year the TPO of GNDEC faces the challenging task of placing final year students in various companies of their respective field while simultaneously maintaining the quality of companies recruiting students. [1]

Traditionally, campus placements have been influenced by academic scores, but in the contemporary era, the scope has broadened significantly. This paper focuses on an innovative approach, where students input their academic performance in key subjects like Operating Systems, Software Engineering, Computer Networks, and more. The inclusion of a wide array of subjects recognizes the multidisciplinary nature of modern technology fields, acknowledging that expertise in various domains contributes to holistic professional development.

Moreover, the paper incorporates the vital aspect of certifications, acknowledging that theoretical knowledge must be complemented by practical skills. Students can choose from a plethora of certifications, ranging from app development to network security, allowing them to showcase their expertise in specialized areas. This not only benefits students but also enables recruiters to identify candidates with specific skill sets tailored to their organizational needs.

Understanding the diverse sectors within the tech industry, this research paper further allows students to express their preference for different fields such as development, security, finance, or marketing. Recognizing that each sector demands a unique skill set and mindset, this customization ensures a more accurate prediction model. Additionally, the choice between technical and managerial roles within these sectors adds a layer of complexity, reflecting the multifaceted career paths available to today's graduates.

The predictive models employed in this research—Decision Tree, Naive Bayes, Random Forest, SVM, KNN, and

Gradient Boost—represent the cutting edge of machine learning and data analytics. By harnessing the power of these algorithms, the paper aims to create a robust prediction system that not only matches students with their ideal career paths but also assists recruiters in identifying the most suitable candidates for their organizations.

III. DATASET

In order to enhance the accuracy and relevance of our Campus Placement Prediction system, a customized database was meticulously curated, tailored to the specific needs of our users. The database comprises essential subjects vital for technical proficiency, including Data Structures and Algorithms (DSA), Database Management Systems (DBMS), Computer Networks (CN), Operating Systems (OS), Mathematics, and Aptitude.

Subjects in the Custom Database:

1. Data Structures and Algorithms
2. Database Management Systems
3. Computer Networks
4. Operating Systems
5. Mathematics
6. Aptitude

User Evaluation Criteria:

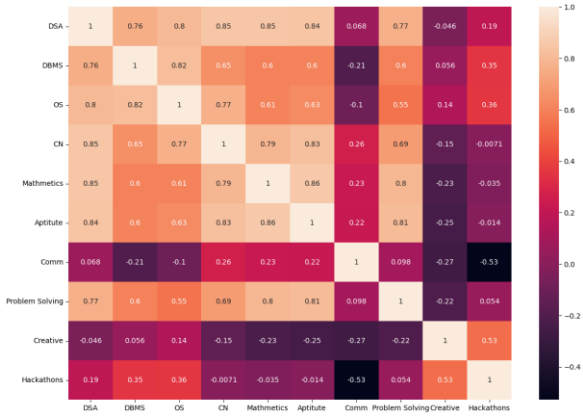
To further refine the prediction process, users are encouraged to self-assess their problem-solving abilities and creativity on a scale of 1 to 10. These criteria play a significant role in determining a candidate's suitability for different job roles. The user's self-assessment in Problem Solving reflects their ability to analyse and solve complex problems, while Creativity gauges their innovative thinking and adaptability.

Skills Assessment:

In addition to subject scores, users are prompted to highlight two specific skills they possess. These skills serve as valuable indicators of the user's expertise and are essential for shaping their career trajectory. By allowing users to input these skills, our system accommodates a wide array of talents and specializations, ensuring a holistic and personalized prediction process.

Heatmap:

A heatmap is a graphical representation of data where individual values in a matrix are represented as colours. Heatmaps are commonly used to visualize correlations in data sets, especially large datasets, and to understand complex patterns within the data. Each cell in the heatmap represents the relationship between two variables by its colour. The below heatmap represents the relation between different different variables in our dataset.



IV. MACHINE LEARNING MODELS

1) Decision Tree:

Decision Tree is a versatile, interpretable, and widely used machine learning algorithm for both classification and regression tasks. It works by recursively splitting the dataset into subsets based on the most significant attribute at each node, eventually creating a tree-like structure of decisions. A tree is either a leaf node labelled with a class or a structure consisting of a test node linked to two or more subtrees.[2]

The decision tree algorithm calculates information gain or Gini impurity to decide the best attribute for splitting the data at each node. The formulas for these metrics are as follows:

Information Gain (for Classification):

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Gini Impurity (for Classification):

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Decision Trees can be graphically represented as a tree structure where nodes represent decisions based on attribute values and edges represent the outcomes of those decisions.

2) Naive Bayes:

Naive Bayes is a probabilistic classifier based on Bayes' theorem. It assumes that features are conditionally independent given the class label, which simplifies the calculation of probabilities.

A Naive Bayesian classifier is a simple probabilistic classifier based on applying Bayesian theorem with strong (naive) independence assumptions.[3]

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Naive Bayes makes the naive assumption that features are conditionally independent, simplifying *P*(Features | Class) to the product of individual feature probabilities.

Since Naive Bayes is a probabilistic model, it doesn't have a graphical representation in the same way decision trees do. Instead, it relies on probability calculations based on the input features and class probabilities.

3) Random Forest:

Random Forest is an ensemble learning method that combines multiple decision trees to create a more accurate and robust model. It introduces randomness by training each tree on a random subset of the data and using random feature subsets.

A forecast is generated using Random Forest, which may estimate the possibility of putting an understudy in an organization.[4] For classification, it typically takes a majority vote, and for regression, it takes the average of predictions made by individual trees. A Random Forest is essentially a collection of decision trees. Visualizing a Random Forest involves showing multiple decision trees together.

Random forest classification is based on the following formula:

Prediction = mode(predictions of individual trees)

Where:

1. prediction is the predicted class of the new data point
2. mode() is the function that returns the most frequent element in a list
3. predictions of individual trees are a list of predictions from the individual trees in the random forest

4) Support Vector Machine (SVM):

SVM is a powerful supervised learning algorithm used for both classification and regression tasks. The Support Vector Machine (SVM) helps in identifying the hyperplane for classifying the data samples.[5]

For a linear SVM, the equation of the hyperplane is w·x+b=0, where w is the weight vector, x is the input feature vector, and b is the bias. The goal is to find w and b that maximize the margin between the classes while minimizing classification errors.

In a 2D feature space, the SVM hyperplane is a straight line that best separates the classes. In a 3D feature space, it's a plane, and in higher dimensions, it's a hyperplane.

5) K-Nearest Neighbours (KNN):

K-nearest neighbors is an algorithm that falls under the category of supervised learning algorithms. The classification as per this algorithm is done based on the distances between the training data and the testing data.[6] For classification, KNN calculates the majority class among its *k* nearest neighbours. For regression, it takes the average of the *k* nearest neighbours' target values. In a 2D feature space, KNN involves identifying the *k* nearest data points to a test point. The class of the test point is determined by the majority class among its *k* nearest neighbours.

6) Gradient Boosting:

Gradient boosting classifier was first introduced in 1999 by Jerome H. Friedman.[9] Each tree corrects the errors of its predecessor, leading to a strong predictive model. Gradient Boosting minimizes the loss function by adding weak learners (usually decision trees) iteratively. It combines the predictions of individual trees, giving more weight to the ones that reduce the loss more effectively.

Gradient Boosting combines decision trees sequentially, with each tree focusing on the mistakes made by the previous ones. It continually refines its predictions, resulting in a strong ensemble model.

Gradient boosting classifier is based on the following formula:

$$f(x) = f_0(x) + \sum_{i=1}^{m} a_i h\ (x; \theta_i)$$

where:

1. f(x) is the predicted class of the new data point.
2. f0(x) is a simple initial model, such as a decision tree.
3. αi is the learning rate for the ith model.
4. h(x;θi) is the ith model.
5. M is the number of models in the ensemble.

V. METHODOLOGY

1) Exploratory Data Analysis (EDA)

EDA is the initial step of deciphering data by first showing the visual representation using different tools available in a data processing tool.[7] During this phase, we meticulously examined the dataset to gain insights into the distribution, correlation, and outliers within the academic scores, certification data, and career preferences. Visualization techniques such as histograms, scatter plots, and heatmaps were employed to unearth hidden patterns and anomalies. By understanding the dataset's nuances, we could make informed decisions regarding feature selection and preprocessing techniques, laying a solid foundation for subsequent analyses.

2) Feature Engineering:

Feature engineering is a central task in data preparation for machine learning.[8] Leveraging the insights from EDA, we engineered new features that encapsulate the essence of a candidate's academic proficiency, certification expertise, and career preferences. Techniques such as one-hot encoding, feature scaling, and dimensionality reduction were employed to prepare the data for model training. Through meticulous feature selection, we identified the most influential attributes that significantly contribute to placement predictions, ensuring the accuracy and efficiency of our models.

3) Building Models on the Data:

The heart of our research lies in the construction of predictive models that accurately forecast campus placements based on the provided inputs. Utilizing a diverse set of algorithms including Decision Tree, Naive Bayes, Random Forest, SVM, KNN, and Gradient Boost we embarked on an extensive model-building journey. Each algorithm was fine-tuned and optimized to achieve the highest predictive accuracy. Cross-validation techniques were employed to ensure the models' robustness and reliability. Model evaluation metrics such as accuracy, precision, recall, and F1-score were utilized to gauge their performance. Comparative analyses were conducted to identify the most suitable algorithm for our prediction system, ensuring its effectiveness in real-world scenarios.

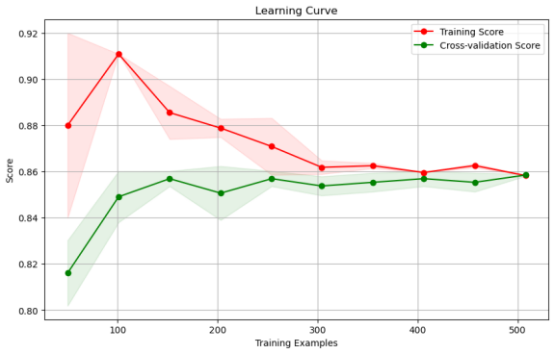
4) Building a website:

To democratize access to our placement prediction system, we developed an intuitive and user-friendly website. Leveraging modern web technologies such as HTML, CSS, and JavaScript, coupled with backend frameworks like Django or Flask, we created an interactive platform where users can input their academic scores, certifications, career preferences, and role choices. The website seamlessly integrates with the trained machine learning models, providing instant placement predictions in a visually appealing format. User experience and interface design were paramount, ensuring accessibility for a wide range of users, from students seeking placements to recruiters scouting for talent. Regular updates and enhancements were made to the website, reflecting the continuous evolution of our prediction system based on the latest data and technological advancements.

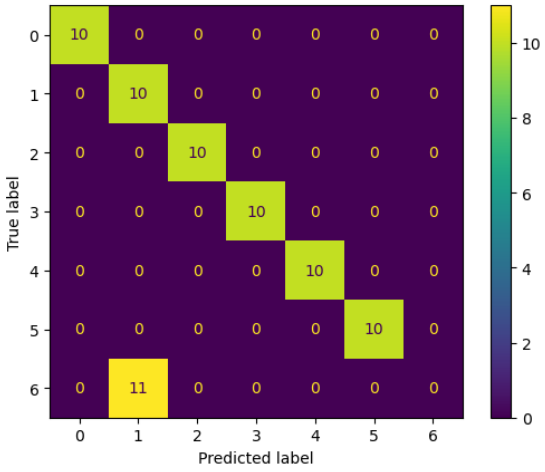
VI. RESULTS

The unambiguousness of the various models used will be depicted in this section with their accuracy percentage, confusion matrix and learning curve as testimony.

Decision tree



The presented learning curve corresponds to a Decision Tree model and reveals insights into its training progression. Initially, both training and cross-validation accuracies show an upward trajectory, peaking around 100 training examples. Subsequently, the training accuracy stabilizes, while the cross-validation accuracy plateaus or slightly diminishes. This pattern suggests that the model effectively learns from the initial dataset but encounters diminishing returns beyond a certain point. Notably, the small gap between training and cross-validation accuracies implies a lack of overfitting, indicating a balanced model. Further improvements might stem from hyperparameter tuning or exploring alternative algorithms. In concise terms, the Decision Tree model exhibits effective learning and generalization initially but reaches a point of diminishing returns, requiring thoughtful optimization for enhanced performance.



Classification Report:

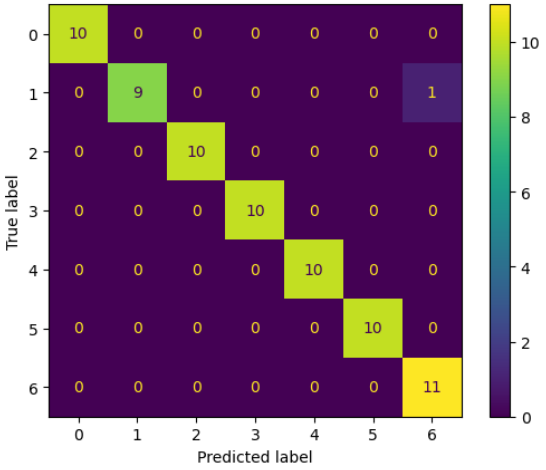
	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	10
1	0.48	1.00	0.65	10
Accuracy	--	--	0.85	71
Macro Avg	0.78	0.86	0.81	71
wt. Avg	0.77	0.85	0.80	71

Random Forest

The learning curve for the Random Forest model depicts a compelling pattern. Initially, the training accuracy surges to a remarkably high level (e.g., 96%) with a small number of training examples, indicative of the model's ability to capture intricate patterns in the data. However, as the number of training examples increases, the training accuracy undergoes fluctuations, suggesting potential sensitivity to the specific data subsets used.

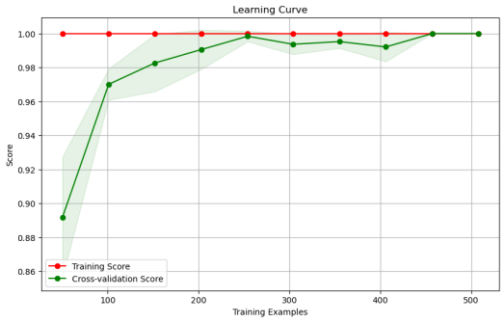
The cross-validation accuracy follows a distinct trend, exhibiting a more gradual increase. Around 250 training examples, a notable improvement occurs, highlighting the ensemble strength of the Random Forest. The model demonstrates solid generalization, with a relatively small gap between training and cross-validation accuracies, indicating effective mitigation of overfitting.

The oscillations in the training accuracy suggest that the model may benefit from additional regularization or fine-tuning of hyperparameters to enhance stability. Overall, the Random Forest model exhibits powerful learning capabilities and generalization, with potential for further refinement through parameter optimization.



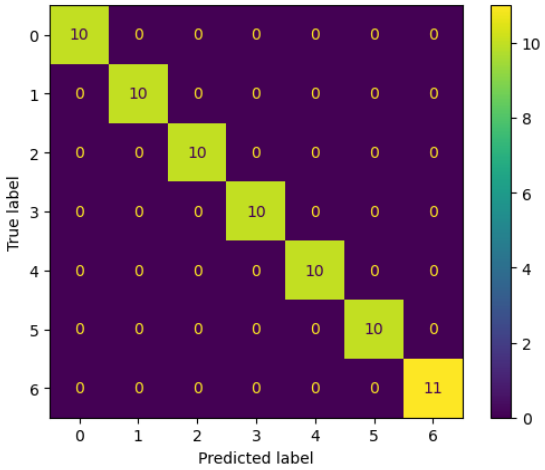
	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	10
1	1.00	0.90	0.95	10
Accuracy	--	--	0.99	71
Macro Avg	0.99	0.99	0.99	71
wt. Avg	0.99	0.99	0.99	71

Gradiant Boost



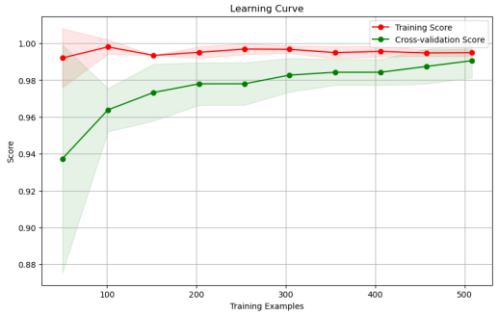
The learning curve for the Gradient Boost model reveals a remarkable trend. In the training set, the model achieves perfect accuracy (1.00) with a small number of examples, emphasizing its ability to fit the training data precisely. The model maintains this flawless performance as the training set size increases, indicating robust learning and adaptability.

The cross-validation accuracy, while not perfect, shows a clear upward trajectory, reaching a high level (e.g., 1.00) with larger training sets. This signifies the model's capacity to generalize well to unseen data. The initial improvement in cross-validation accuracy indicates effective learning, with subsequent iterations maintaining high performance.



	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	10
Accuracy	--	--	1.00	71
Macro Avg	1.00	1.00	1.00	71
wt. Avg	1.00	1.00	1.00	71

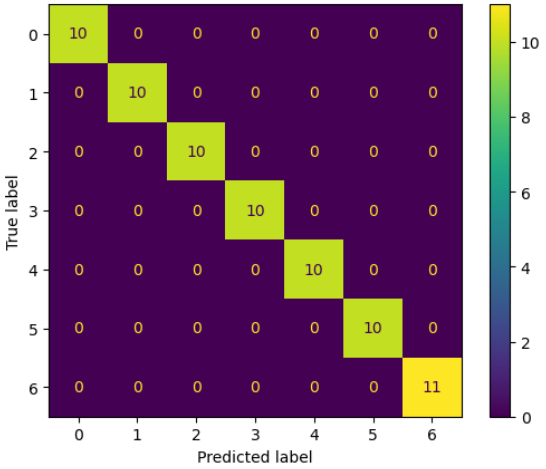
Logistic Regression



The learning curve for the Logistic Regression model illustrates a compelling pattern. In the training set, the model achieves consistently high accuracy, indicating its proficiency in capturing patterns within the data. The accuracy remains consistently close to 1.00, emphasizing the model's strong fit to the training examples.

In the cross-validation set, the accuracy starts at a respectable level and gradually improves with additional training examples. The final accuracy of 0.998 on the cross-validation set suggests the model's ability to generalize well.

The small gap between training and cross-validation accuracies indicates minimal overfitting, portraying a well-balanced model.



	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	10
Accuracy	--	--	1.00	71
Macro Avg	1.00	1.00	1.00	71
wt. Avg	1.00	1.00	1.00	71

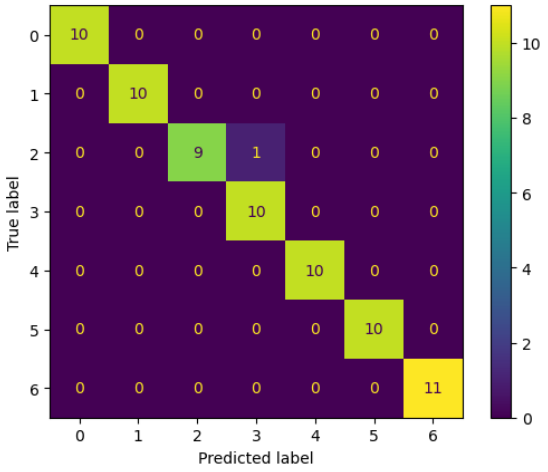
K – Nearest Neighbours

sThe learning curve for the K-Nearest Neighbors (KNN) model showcases an interesting pattern. In the training set, the model consistently achieves high accuracy, reaching near-perfect scores (e.g., 0.999) with a larger number of training examples. This suggests that the model effectively memorizes the training data and adapts well to the patterns within it.

In the cross-validation set, the accuracy starts at a relatively high level and maintains a stable performance as the number of training examples increases. The accuracy on the cross-validation set indicates the model's ability to generalize to

new data, although it does not reach the perfect scores observed in the training set.

The small gap between training and cross-validation accuracies suggests minimal overfitting, indicating a robust KNN model. KNN is known for its simplicity and effectiveness, particularly in capturing complex relationships in data. Further optimization might involve tuning the hyperparameters, such as the number of neighbors, to potentially enhance the model's performance. Overall, the KNN model demonstrates strong learning and generalization capabilities.



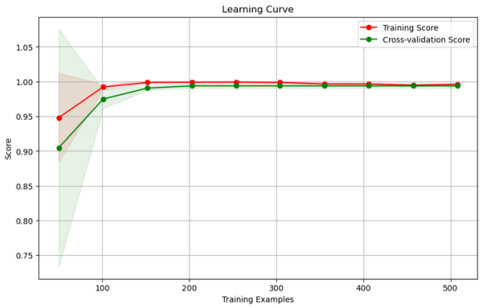
	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	10
Accuracy	--	--	0.99	71
Macro Avg	0.99	0.99	0.99	71
wt. Avg	0.99	0.99	0.99	71

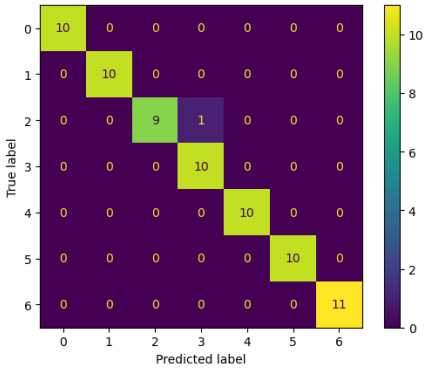
Support Vector Machine

The learning curve for the Support Vector Machine (SVM) model depicts a clear trend. In the training set, the model achieves consistently high accuracy, reaching near-perfect scores (e.g., 1.00) with a larger number of training examples. This signifies the SVM's effectiveness in capturing complex relationships within the training data.

In the cross-validation set, the accuracy starts at a respectable level and experiences a gradual improvement with additional training examples. The upward trend in cross-validation accuracy indicates the model's ability to generalize well to unseen data, reflecting the robustness of SVM in handling various patterns.

The small gap between training and cross-validation accuracies suggests minimal overfitting, portraying a well-generalizing model. Further optimization could involve tuning kernel parameters or exploring different kernel functions to potentially enhance the model's performance. Overall, the SVM model demonstrates strong learning and generalization capabilities.





	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	10
Accuracy	--	--	0.99	71
Macro Avg	0.99	0.99	0.99	71
wt. Avg	0.99	0.99	0.99	71

VII. CONCLUSION

In conclusion, By harnessing the power of ML, this project aims to provide a valuable tool provide's a valuable guidance for educational institutions and students to make informed career decisions. The project's relevance lies in its potential to enhance carrer decesion, making them more data-driven and objective.

VIII. REFERENCE

[1] A. Giri, M. V. V. Bhagavath, B. Pruthvi and N. Dubey, "A Placement Prediction System using k-nearest neighbors classifier," 2016 Second International Conference on Cognitive Computing and Information Processing (CCIP), Mysuru, India, 2016, pp. 1-4, doi: 10.1109/CCIP.2016.7802883.

[2] N. Kumar, A. S. Singh, T. K and E. Rajesh, "Campus Placement Predictive Analysis using Machine Learning," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2020, pp. 214-216, doi: 10.1109/ICACCCN51052.2020.9362836.

[3] A. S. Sharma, S. Prince, S. Kapoor and K. Kumar, "PPS — Placement prediction system using logistic regression," 2014 IEEE International Conference on MOOC, Innovation and Technology in Education (MITE), Patiala, India, 2014, pp. 337-341, doi: 10.1109/MITE.2014.7020299.sss

[4] J. Nagaria and S. V. S, "Utilizing Exploratory Data Analysis for the Prediction of Campus Placement for Educational Institutions," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-7, doi: 10.1109/ICCCNT49239.2020.9225441.

[5] Rao, Abhishek S., S. V. Aruna Kumar, Pranav Jogi, K. Chinthan Bhat, B. Kuladeep Kumar, and Prashanth Gouda. "Student placement prediction model: a data mining perspective for outcome-based education system." International Journal of Recent Technology and Engineering (IJRTE) 8 (2019): 2497-2507.

[6] Pal, Ajay Kumar, and Saurabh Pal. "Classification model of prediction for placement of students." International Journal of Modern Education and Computer Science 5, no. 11 (2013): 49.

[7] Quinlan, J. Ross. "Learning decision tree classifiers." ACM Computing Surveys (CSUR) 28.1 (1996): 71-72.

[8] Nargesian, Fatemeh, et al. "Learning Feature Engineering for Classification." Ijcai. Vol. 17. 2017.

[9] Chen, Tianqi, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, and Tianyi Zhou. "Xgboost: extreme gradient boosting." R package version 0.4-2 1, no. 4 (2015): 1-4.