iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** IV **Month of publication:** April 2020

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

Career Prediction using Machine Learning Algorithms

Anmol Baroliy[1], Neha Shakya[2], Avdhesh Dwivedi[3], Shikha Sehrawat[4]
[1, 2, 3]*Student, Computer Science, Raj Kumar Goel Institute of Technology, Uttar Pradesh, India*

Abstract: ***Selecting a career profession is one of the important yet confusing decision for the students and their parents. With the plenty of career options and its sub- divisions available in the professional line, there is a high possibility of choosing and implementing a wrong choice according to the educational and personal traits of the student, which can lead to unhappy and stressful life. So, choosing a right career path for the student at the right age is very important for their lives. This paper mainly concentrates on the career area prediction of computer science domain candidates. This system is a part of a web application called EduSys. It is in learning module system's quiz section for Computer Science Engineers. These kinds of career recommender systems help students in picking the job role based on his/her performance and academic records. The quiz is designed based on the academic as well as personal traits having 36 parameters in total, which helps in suggesting the future career paths to students through machine learning analysis using different algorithms.***

Keywords: Career Prediction, Machine learning, Decision Tree, Naïve Bayes Classification, Stochastic Gradient Descent Classifier, kernel Support Vector Classification, Random Forest Classification.

I. INTRODUCTION

As students are going through their academics and pursuing their interested courses, it is very important for them to assess their capabilities and identify their interests so that they will get to know in which career area their interests and capabilities are going to put them in [2]. To reach the aim, students need to be planned and organized from initial stages of their education. Hence, it is very important to evaluate their performance, identify their personal traits and how close they are to their goal and assess whether they are in the right path that directs towards their goal. This quiz will assess the students technically and suggest the students and companies job roles suited on their performance. But here various factors including abilities of students. The total number of parameters that were taken into consideration as inputs are 36 and 12 job positions for computer science engineer. The job positions are Cloud Operations Engineer, System Analyst, Data Scientist, Network Architect, Database Administration, Mobile App Developer, Information Security Analyst, UX Designer, Technical Writer, Computer Hardware Engineer, Multimedia Programmer and Web Developer. Also, recruiters while recruiting the candidates after assessing them in all different aspects, these kinds of career recommender systems help them in deciding in which job role the candidate should be kept in based on his/her performance and other evaluations. For the best possible output classification and prediction advanced machine learning algorithms like Random Forest, Decision tree, Naive Bayes Classification, Stochastic Gradient Descent Classifier, kernel Support Vector Classification are used.

II. MACHINE LEARNING

Machine Learning is a technique where the machines are trained in such a way that it gains the ability to respond to a particular input or scenario based on the previous inputs it has learnt. Simply giving computers the ability to learn by using statistical techniques. Helps in solving very com-plex tasks and problems very easily and without involving much human labour. Various applications of machine learning include NLP, classification, prediction, image recognition, medical diagnosis, algorithm building, and much more. In this paper classification and prediction are being done. Majority of problems in machine learning can be solved using supervised and unsupervised learning. The supervised machine learning algorithms are those algorithms which needs external assistance. The input dataset is divided into train and test dataset. The train dataset has output variable which needs to be predicted or classified [4] If the final class labels are previously known and all the other data items are to be assigned with one of the available class labels, then it is call supervised. And if the final output classes and sets are not known and it is done by identifying the similarity between data point and their characteristics and finally, they are made into groups based on these characteristics then it is called un-supervised. Classification falls under supervised. Input parameters are given and based on their properties a predefined class label is assigned. There are other alternatives like clustering and regression. Based on the type of problem the apt model is chosen.

III.ALGORITHMS

Machine Learning algorithm is an evolution of the regular algorithm. It makes your programs "smarter", by allowing them to automatically learn from the data provided. The algorithm is mainly divided into training Phase and testing phase. If maximum number of predictions are right then model will have a good accuracy percentage and is reliable to continue with otherwise better to change the model. Algorithm used for career prediction are:

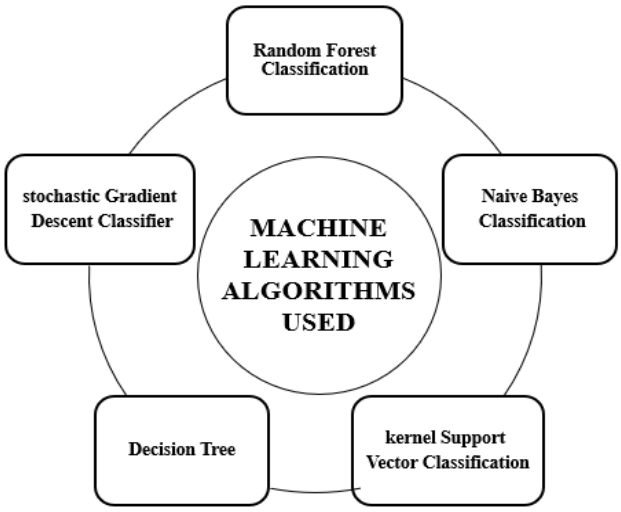
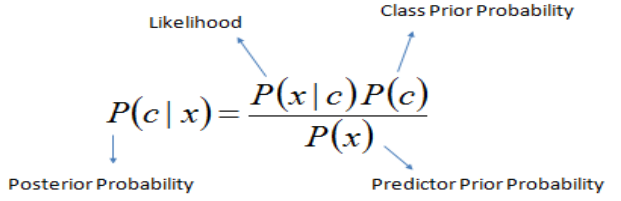


Fig.1 Different Types of algorithms used in the project

A. Naïve Bayes Classification

The Naive Bayes Classifier is very effective on many real data applications. The performance of Naïve Bayes usually benefits from a precise estimation of univariate conditional probabilities and from variable selection [1]. This is a classification technique based on *Bayes' theorem* with an assumption of independence between predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Naive Bayesian model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. Naïve Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticate classification methods. Bayes theorem [7] provides a way of calculating the posterior probability P(C/X) of class from P(C) is the prior probability of class, P(X) is the prior probability of predictor and P(X/C) is the likelihood which is the probability of predictor given class. Naïve Bayes classifier assumes that the effect of the value of a predictor (X) on a given class (C) is independent of the values of other predictors called conditional independence. Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c) shown in fig.2. Where P(c|x) is the posterior probability of class (target) given predictor (attribute), P(c) is the prior probability of class, P(x|c) is the likelihood which is the probability of predictor given class and P(x) is the prior probability of predictor.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Fig.2 Equation for Naïve Bayes

B. Random Forest Classification

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. Random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and then selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result and giving best solution possible.

C. *Decision Tree Classifier*

Decision trees are those type of trees which groups attributes by sorting them based on their values Decision Trees are extremely popular and one of the simple and easy to implement machine learning classification problems [4]. Decision trees laid basic foundation for many advanced algorithms like bagging, gradient boosting and random forest. The XG Boost algorithm discussed above is the advanced version of this general decision tree. A node denotes input variable (X) and a split on that variable, assuming the variable is numerical. The leaf which are also called the terminal nodes of the tree possess an output variable (y) which is vital for prediction [2]. Calculate information gain or entropy for each of the nodes before the split. Select the node that has more information gain or less entropy. Further split the node and reiterate the process. The process is iterated until there is no possibility to split, equation is given in fig.3.

$$H(S) = \sum_{x \in X} -p(x) \log_2 p(x) \qquad IG(A, S) = H(S) - \sum_{t \in T} p(t) H(t)$$

Fig.3 Information gain is the metric that measures how much entropy is reduced before to after split.

D. *Stochastic Gradient Descent Classifier*

SGD is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. Even though SGD has been around in the machine learning community for a long time, it has received a considerable amount of attention just recently in the context of large-scale learning. SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing problems. Extreme Gradient Boosting is a very efficient and scalable technique for the implementation of gradient boosting trees. It supports ranking, classification and regression. It is normally 10 times faster than gbm but its customization and result tendency is very good [9].

E. *Kernel Support Vector Classification*

Recent progresses have enabled additive kernel version of SVM efficiently solves such large-scale problems nearly as fast as a linear classifier [5]. In case of non-linearly separable data, the simple SVM algorithm cannot be used. Rather, a modified version of SVM, called Kernel SVM, is used. Basically, the kernel SVM projects the non-linearly separable data lower dimensions to linearly separable data in higher dimensions in such a way that data points belonging to different classes are allocated to different dimensions.

IV. IMPLEMENTATION

The implementation phase has various steps and the flow of implementation is shown in fig.4:

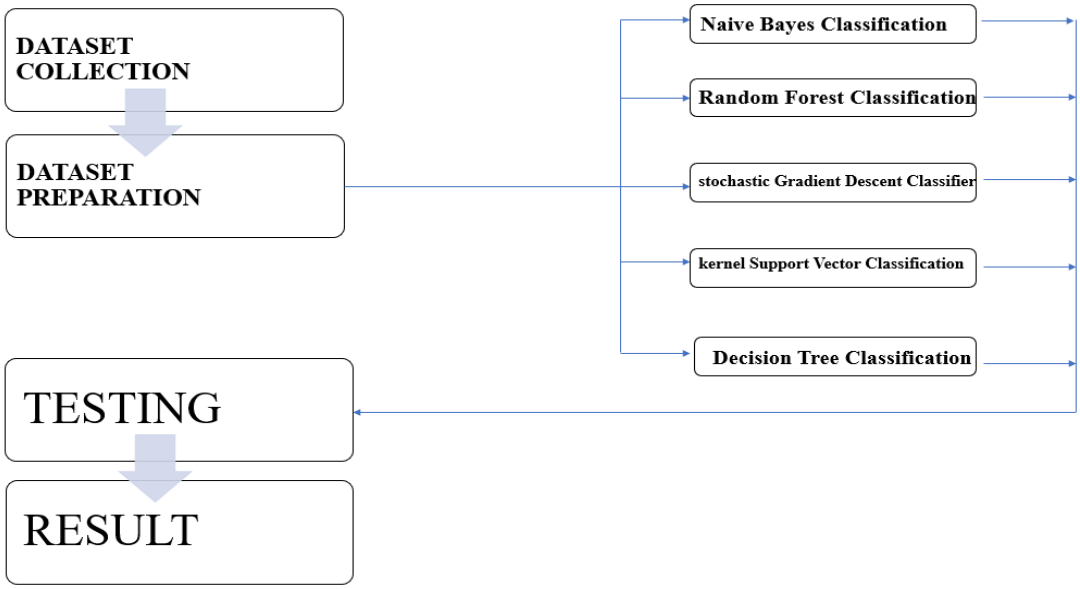


Fig. 4 Flow of implementation of the system

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

A. Dataset Collection

Collection of data is one of the major and most important tasks as the input we feed to the algorithms is a dataset. The algorithms efficiency and accuracy depend upon the correctness and quality of dataset [2]. The factors play vital role in deciding student's progress towards a career area, all these are taken in-to consideration. Data is collected in many ways like some amount of data is randomly generated using different databases available various dataset platforms and other from college alumni database.

B. Dataset preparation

Making the data useful is another vital task. Data collected from various means will be in an ambiguous format and there may be lot of in-valid data values and unwanted data. Cleaning all these data and replacing them with appropriate or approximate data and removing the missing data and replacing them with some fixed alternate values are the basic steps in pre-processing of data [2]. Even data collected may contain completely garbage values. It may not be in exact format or way that is meant to be. All such cases must be verified and replaced with alternate values to make data meaning meaningful and useful for further processing (Fig.5). Data must be kept in an organized format for use.

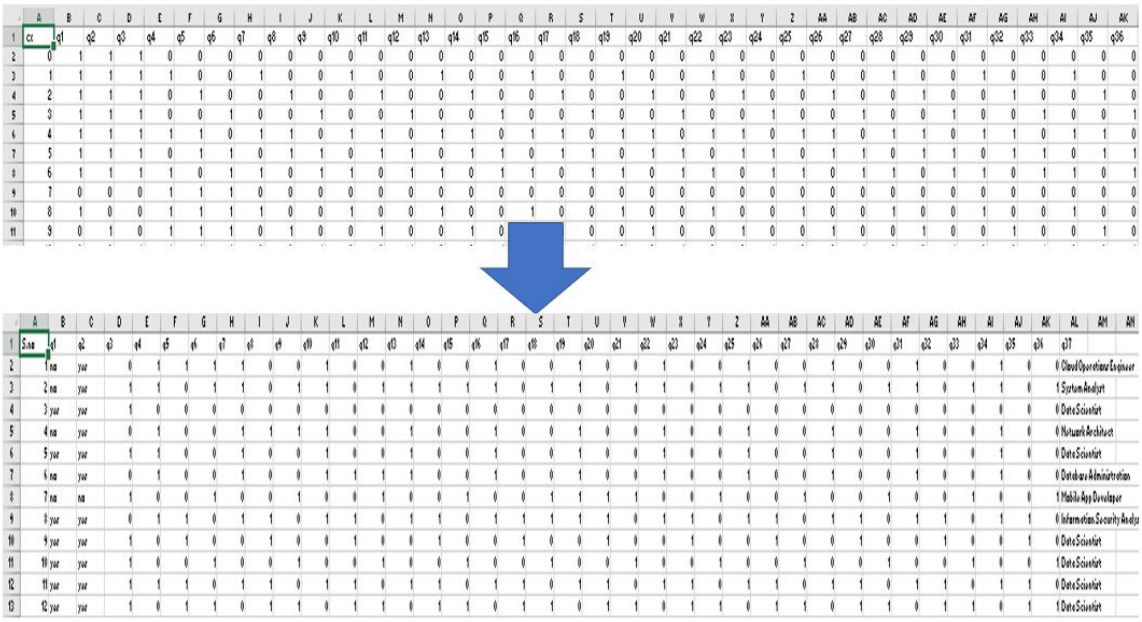


Fig.5 Data preparation (Deleting unwanted values and cleaning the dataset)

C. Algorithm Implementation

The dataset is sent through different algorithms and then the testing is to be done. Each algorithm has its own set of implementation criteria for the prediction and accuracy check. Example shown in fig.6.

```
[ ]  pred_naive=classifier_naive.predict(x)
     data["Predicted values By Naive"]=pred_naive
     data[['q37','Predicted values By Naive']]
```

	q37	Predicted values By Naive
0	Cloud Operations Engineer	Cloud Operations Engineer
1	System Analyst	System Analyst
2	Data Scientist	Data Scientist
3	Network Architect	Network Architect
4	Data Scientist	Data Scientist
...	...	...
79	Web Developer	Web Developer
80	Technical Writer	Technical Writer
81	Computer Hardware Engineer	Computer Hardware Engineer
82	Information Security Analyst	Information Security Analyst
83	Web Developer	Web Developer

Fig. 6 Implementing Naïve Bayes algorithm

D. Testing the Implementation

After processing of data and choosing the algorithms, next is testing. The performance of the algorithm, quality of data, and required output appears here. Training as discussed before is the process of making the machine to learn and giving it the capability to make further predictions. Whereas testing means already having a predefined data set with output and checking either it is giving the right prediction or not. Data correlation is the way in which one set of data may correspond or relate to another set. Statisticians and data analysts measure correlation of two numerical variables to find an insight about their relationships. The set of correlation values between pairs of its attributes form a matrix which is called a correlation matrix shown in fig.7.

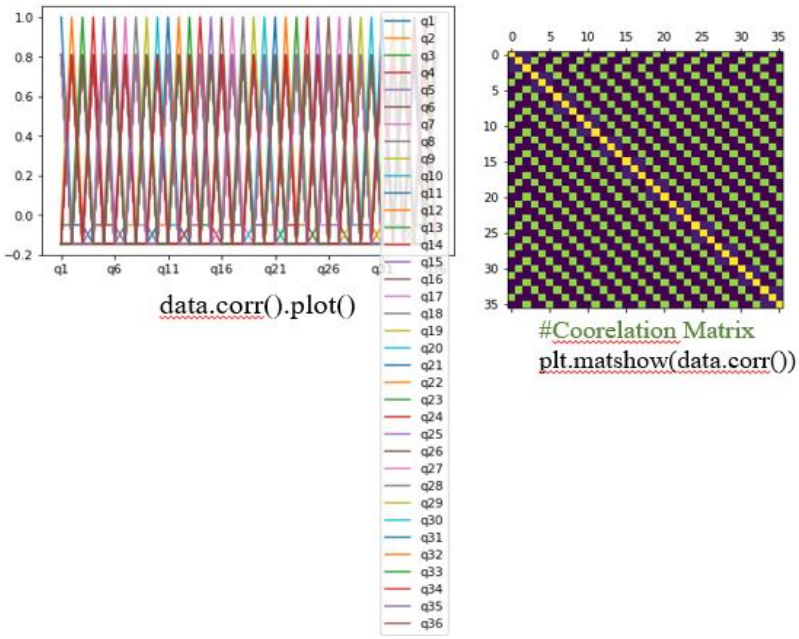


Fig.7 Data Correlation

A confusion matrix is a matrix (table) that can be used to measure the performance of a machine learning algorithm, usually a supervised learning one. Each row of the confusion matrix represents the instances of an actual class and each column represents the instances of a predicted class. In our dataset we are getting a multiclass confusion matrix. We can generalize this to the multi-class case. To do this we summarize over the rows and columns of the confusion matrix [6].

V. RESULTS AND CONCLUSION

A. Naïve Bayes Classification

Naïve Bayes gave more accuracy with 100 percent(fig.8).

```
[ ] #Naive Bayes Classification

    from sklearn.naive_bayes import GaussianNB
    classifier_naive = GaussianNB()
    classifier_naive.fit(x_train, y_train)
    y_naive_pred = classifier_naive.predict(x_test)

    #Accuracy

    classifier_naive.score(x_test,y_test)*100

    100.0
```

```
NAIVE BAYES RESULTS
                              precision    recall  f1-score   support

                Data Scientist       1.00      1.00      1.00         1
        Database Administration      1.00      1.00      1.00         2
Information Security Analyst         1.00      1.00      1.00         1
          Mobile App Developer       1.00      1.00      1.00         1
             Network Architect       1.00      1.00      1.00         1
             Technical Writer        1.00      1.00      1.00         1
                  UX Designer        1.00      1.00      1.00         2

                     accuracy                           1.00         9
                    macro avg       1.00      1.00      1.00         9
                 weighted avg       1.00      1.00      1.00         9

[[1 0 0 0 0 0 0]
 [0 2 0 0 0 0 0]
 [0 0 1 0 0 0 0]
 [0 0 0 1 0 0 0]
 [0 0 0 0 1 0 0]
 [0 0 0 0 0 1 0]
 [0 0 0 0 0 0 2]]
```

Fig. 8 Naïve Bayes results

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

B. Random Forest Classification

Random Forest gave accuracy with 55.55555555555556 percent(fig.9).

```
#Random Forest Classifier

from sklearn.ensemble import RandomForestClassifier

classifier_rfc = RandomForestClassifier(n_estimators=20, random_state=0)
classifier_rfc.fit(x_train, y_train)
y_rfc_pred = classifier_rfc.predict(x_test)

#accuracy
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_rfc_pred)*100
```

```
55.55555555555556
```

```
RANDOM FOREST CLASSIFIER RESULTS
                              precision    recall  f1-score   support

                Data Scientist       1.00      1.00      1.00         1
     Database Administration       0.00      0.00      0.00         2
Information Security Analyst       0.00      0.00      0.00         1
       Mobile App Developer        1.00      1.00      1.00         1
     Multimedia Programmer         0.00      0.00      0.00         0
         Network Architect         1.00      1.00      1.00         1
          Technical Writer         1.00      1.00      1.00         1
               UX Designer         1.00      0.50      0.67         2
             Web Developer         0.00      0.00      0.00         0

                  accuracy                           0.56         9
                 macro avg       0.56      0.50      0.52         9
              weighted avg       0.67      0.56      0.59         9

[[1 0 0 0 0 0 0 0 0]
 [0 0 0 0 1 0 0 0 1]
 [0 1 0 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 1 0 0]
 [0 0 0 0 1 0 0 1 0]
 [0 0 0 0 0 0 0 0 0]]
```

Fig. 9 Random Forest Classification results

C. Decision Tree Classifier

Decision tree gave accuracy with 66.666666666666666 percent(fig.10).

```
#Decision Tree Classifier

from sklearn.tree import DecisionTreeClassifier
classifier_dtc = DecisionTreeClassifier()
classifier_dtc.fit(x_train,y_train)
y_dtc_pred = classifier_dtc.predict(x_test)

#accuracy
accuracy_score(y_test,y_dtc_pred)*100
```

```
66.66666666666666
```

```
DECISION TREE RESULTS
                              precision    recall  f1-score   support

                Data Scientist       1.00      1.00      1.00         1
     Database Administration       1.00      0.50      0.67         2
Information Security Analyst       1.00      1.00      1.00         1
       Mobile App Developer        0.50      1.00      0.67         1
         Network Architect         0.00      0.00      0.00         1
          Technical Writer         0.33      1.00      0.50         1
               UX Designer         1.00      0.50      0.67         2

                  accuracy                           0.67         9
                 macro avg       0.69      0.71      0.64         9
              weighted avg       0.76      0.67      0.65         9

[[1 0 0 0 0 0 0]
 [0 1 0 0 0 1 0]
 [0 0 1 0 0 0 0]
 [0 0 0 1 0 0 0]
 [0 0 0 1 0 0 0]
 [0 0 0 0 0 1 0]
 [0 0 0 0 0 1 1]]
```

Fig.10 Decision tree results

D. Stochastic Gradient Descent Classifier

Stochastic gradient decent gave accuracy with 88.88888888888888 percent(fig.11).

```
#Stochastic Gradient Descent Classifier

from sklearn.linear_model import SGDClassifier

classifier_sgd= SGDClassifier(loss='modified_huber',shuffle=True,random_state=0)
classifier_sgd.fit(x_train,y_train)
y_sgd_pred = classifier_sgd.predict(x_test)

#accuracy
accuracy_score(y_test,y_sgd_pred)*100
```

```
88.88888888888889
```

```
STOCHASTIC GRADIENT DESCENT RESULTS
                              precision    recall  f1-score   support

  Cloud Operations Engineer       0.00      0.00      0.00         0
                Data Scientist       1.00      1.00      1.00         1
     Database Administration       1.00      1.00      1.00         2
Information Security Analyst       1.00      1.00      1.00         1
       Mobile App Developer        1.00      1.00      1.00         1
         Network Architect         0.00      0.00      0.00         1
          Technical Writer         1.00      1.00      1.00         1
               UX Designer         1.00      1.00      1.00         2

                  accuracy                           0.89         9
                 macro avg       0.75      0.75      0.75         9
              weighted avg       0.89      0.89      0.89         9

[[0 0 0 0 0 0 0 0]
 [0 1 0 0 0 0 0 0]
 [0 0 2 0 0 0 0 0]
 [0 0 0 1 0 0 0 0]
 [0 0 0 0 1 0 0 0]
 [1 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 2]]
```

```
[ ]  # Plot a graph btwn Actual vs Predcited values By SGD
     plt.plot(data["q1"], data["q37"], color="green" , linewidth='5')
     plt.plot(data["q1"], data["Predicted values By SGD"] , color="blue", linewidth='2')
     plt.title("Actual Vs Predicted by q1")
     plt.xlabel("q1")
     plt.ylabel("Actual Vs Predicted By SGD")
```

```
Text(0, 0.5, 'Actual Vs Predicted By SGD')
```

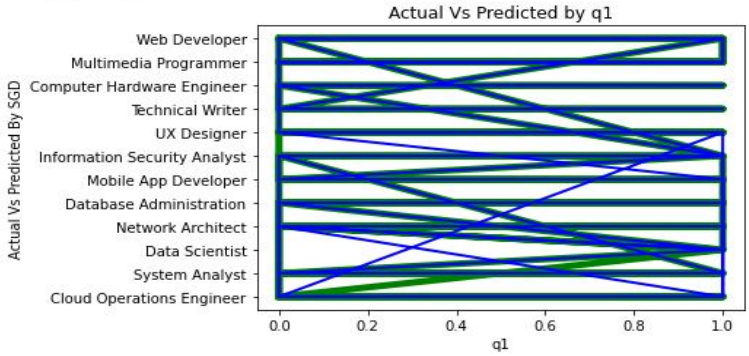


Fig. 11 Stochastic Gradient Descent Classifier result

E. Kernel Support Vector Classification

Kernel SVC gave accuracy with 66.666666666666666 percent(fig.12).

```
#Kernel Support Vector Classification

from sklearn.svm import SVC

classifier_svc = SVC(kernel='linear',random_state=100)
classifier_svc.fit(x_train,y_train)
y_svc_pred = classifier_svc.predict(x_test)

#accuracy
classifier_svc.score(x_test,y_test)*100
```

```
66.66666666666666
```

```
SVC RESULTS
                              precision    recall  f1-score   support

              Data Scientist       1.00      1.00      1.00         1
      Database Administration      1.00      0.50      0.67         2
Information Security Analyst       1.00      1.00      1.00         1
        Mobile App Developer       1.00      1.00      1.00         1
           Network Architect       0.00      0.00      0.00         1
           Technical Writer        1.00      1.00      1.00         1
                 UX Designer       1.00      0.50      0.67         2
               Web Developer       0.00      0.00      0.00         0

                    accuracy                           0.67         9
                   macro avg       0.75      0.62      0.67         9
                weighted avg       0.89      0.67      0.74         9

[[1 0 0 0 0 0 0 0]
 [0 1 0 0 0 0 0 1]
 [0 0 1 0 0 0 0 0]
 [0 0 0 1 0 0 0 0]
 [0 0 0 0 0 0 0 1]
 [0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 1 1]
 [0 0 0 0 0 0 0 0]]
```

Fig. 12 Kernel SVC results

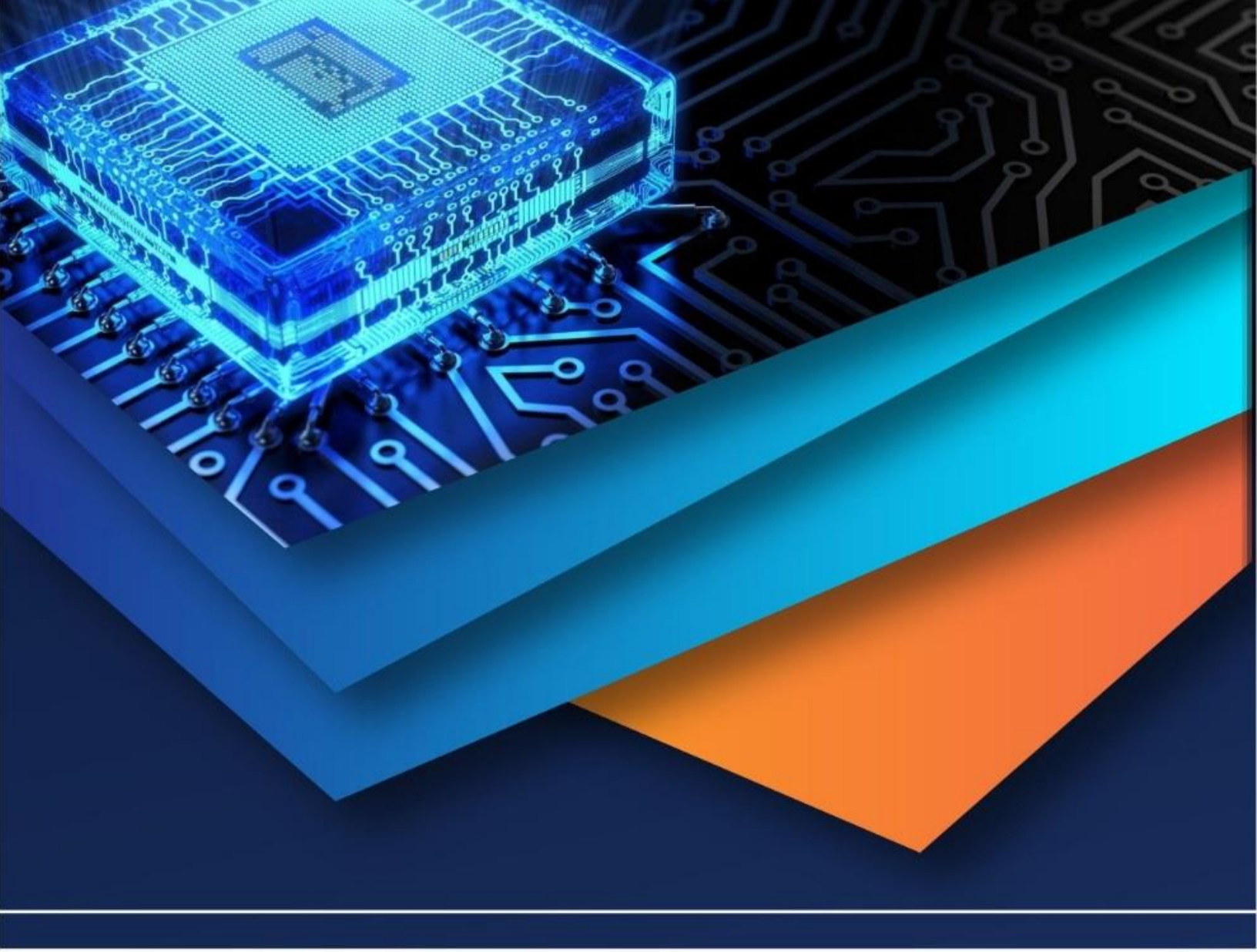
This paper concludes that the results of the analysis of all the algorithms is found to be useable for the career prediction of a computer Science engineering student. These results can be used for future projects related to career predictions.

VI. FUTURE SCOPE

Machine learning has lots of algorithms to analyse and implement to gain the benefits of them in various fields of human endeavours. All such algorithms work directly or indirectly making the use of idea presented by the fundamental learning techniques of supervised learning, unsupervised learning and reinforcement learning [8]. Finding and improving the model takes research, experimentation, perseverance and basic studies. The goal of algorithm is to predict a particular job role. The algorithm uses a many models with inputs related to both the student and the job. To obtain increasingly valuable inputs, we experimented with hundreds of features, including interaction terms that incorporate attributes of both the computer science student and the job. Predicting outcomes with career centre data is still in its infancy. The research is complex due to the vast amount of data requiring analysis and the newness of applying higher education data to the models. Nevertheless, there are vast opportunities to predict outcomes in employment and beyond, including student satisfaction, time to graduation, retention, and more. Data Science predict and compute the outcome which would have taken more time for the humans to proceed [10]. The possibilities are endless, and leveraging machine learning within university career canters is a natural starting point. Increase of more attributes can give the backbone to the system and correctness [3]. In future one can try to develop a system including recruiter's interest and can collect the data, making the interview for the suitable job easier and more effective for the student for that particular job.

REFERENCES

[1] Shreyas Harinath, Aksha Prasad, Suma H S, Suraksha A, Tojo Mathew, "Student placement prediction using machine learning," in *IRJET*, vol.6, April 2019, p.4577.
[2] K. Sripath Roy, K.Roopkanth , V.Uday Teja , V.Bhavana, J.Priyanka , " Student career prediction using advance machine learning techniques," in *IJET*, 2018, p.27
[3] Linsey S. Hugo "Predicting employment through machine learning" [Online]. Available:https://www.naceweb.org/career-development/trends-and-predictions/predicting-employment-through-machine-learning/., "Future scope", May 2019.
[4] Ayon Dey, "Machine Learning Algorithms: A Review," in *IJCSIT*, vol. 7, 2016, paper 11.3.4, p. 109.
[5] Xufeng Wang, "Accelerated stochastic gradient method for support vector machines classification with additive kernel," in *IEEE*, 2017.
[6] (2010) Bernd Klein, website. [Online]. Available: https://www.python-course.eu/confusion_matrix.php.
[7] S. Kanchana," Statistical Analysis Using Machine Learning Approach for Multiple Imputation of Missing Data", in *IJRASET*, vol.6, February 2018, p.2091.
[8] Manish Kumar Singh, Prof. G S Baluja, Dr. Dinesh Prasad Sahu, "Analyzing Machine Learning Algorithms & their Areas of Application", in *IJRASET*, vol.5, June 2017, p.656.
[9] Sandeep Kumar, "Performance Comparison of Machine Learning", in IJRASET, in vol. 5, December 2017, p.1040.
[10] P. Harika Reddy, K Rushikesh Surapaneni Gopi Siva Sai Teja, "Introduction to Data Science and Machine Learning", in IJRASET, in vol. 6, June 2018, p.1324.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY