iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35746

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 9 Issue VI Jun 2021- Available at www.ijraset.com

Prophecy on Programming Language using Machine Learning Algorithms

Komal Bhaskar Thube[1], Suhasini Vijaykumar[2]
[1, 2]*Bharati Vidyapeeth Institute of Management & Information Technology*

Abstract: ***A programming language is a computer language developers use to develop software programs, scripts, or other sets of instruction for computers to execute. It is difficult to determine which programming language is widely used. In our work, I have analyzed and compared the classification results of various machine learning models and find out which programming language is widely used by developers. I have used Support Vector Machine (SVM), K neighbor classifier (KNN),Decision Tree Classifier(CART) for our comparative study. My task is to analyze different data and to classify them for the efficiency of each algorithm in terms of accuracy, precision, recall, and F1 Score. My best accuracy was 94.29% percent which was found using SVM. These techniques are coded in python and executed in Jupyter NoteBook, the Scientific Python Development Environment. Our experiments have shown that SVM is the best for predictive analysis and from our study that SVM is the well-suited algorithm for the prediction of the most widely used programming language.***
Keywords: Support Vector Machine (SVM), K Nearest Neighbor (KNN), and Decision Tree Classifier (CART).

I. INTRODUCTION

A programming language is a formal language comprising a set of instructions that turn out numerous sorts of output. It's a notational system for describing computation throughout a machine-readable and human-readable form. It is a tool for developing executable models for a class of problem domains. It is also used in computer programming to implement algorithms. The amount of programming languages in use has increased. It is difficult to determine programming languages are most widely used. One language might occupy the bigger range of programmer hours, a different one have additional lines of code, a third might utilize the most CPU time, and so on. Some languages are extraordinarily standard for specific forms of applications. As an example, Kotlin is used for any kind of development, be it server-side, client-side web, and Android; and other languages are regularly used to write many various kinds of applications. The main goal of this paper is to analyze different algorithms and to predict the most widely used programming language by developers. Algorithms are tested with regard to the accuracy, precision.

II. DATASET

The dataset had picked from kaggle.com. It includes different programming languages as per the 2020 Stack Overflow developer survey that was selected for analysis. It is openly accessible. In this dataset, there are different languages where developers have done extensive development work over the past year. There are no missing values in the dataset and there are numerical features in the dataset.

III. DATA VISUALIZATION

The mission of this paper is to analyze different algorithms and predicting the most widely used programming language. Algorithms are tested concerning accuracy, precision. I eagerly want to study it more for a better and reliable result.

IV. MODEL SELECTION

Model selection is a process of choosing different types of machine learning approaches -e.g. SVM, KNN, etc. or choosing different hyperparameters or sets of features for the same machine learning approach - e.g. deciding between the polynomial degrees/complexities for linear regression In my dataset, I have the outcome variable or Dependent variable. A classification algorithm of supervised learning is applied to it. I have chosen three different types of classification algorithms in Machine Learning.

1) Support Vector Machines(SVM)
2) K- Nearest Neighbor (KNN)
3) Decision Tree Classifier(CART)

The architecture works along with the binary classification of tagged info. There are total three algorithms implemented to find out the best one. The algorithms that had selected were based on their ability to predict. The machine learning algorithms used for this particular problem are. Support VectorMachine(SVM), K-NearestNeighbor(KNN), Decision Tree Classifier(CART). The output values of the algorithms are properly equated to decide the most accurate prediction of the programming language.

A. Support Vector Machine

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification issues. After giving an SVM model a set of labeled training data for each category, they are able to categorize new text. The objective of the support vector machine algorithm is to search a hyperplane in N-dimensional space (N — the number of features) that clearly classifies the data points. To separate the two classes of data points, several possible hyperplanes might be chosen. My objective is to search out a plane that has the utmost margin, i.e. The utmost distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points may be classified with more confidence. Hyperplanes are decision boundaries that facilitate classifying the data points. Data points falling on either aspect of the hyperplanes are often attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is simply a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine once when the number of features exceeds 3.

B. K-Nearest Neighbors

The k-nearest neighbors (KNN) is a very easy, efficient-to-implement supervised algorithm that can utilize as a way to solve both classification and regression problems. An algorithm called supervised when a machine learning algorithm depends on a tagged input value to educate itself a function that makes an appropriate result when inserting a new untagged value. 1. The KNN Algorithm Loads the input value. 2. Then start by putting K to a selected amount of neighbors. 3. Equate the space among the query and current example in the data 3.1. 3.2 plus space in between and the index of the particular value to a sequential collection. 4. Sort the sequential set of spaces and indices from smallest to largest by the spaces. 5. Choose the 1st K entries from the sorted set. 6. Take in the tags of the chosen K entries.7. If regression, send the mean of the K tags.8.If classification, send the mode of the K tags.

$$\sqrt{(x0-x1)^2+(y0-y1)^2} \quad (1)$$

C. Decision Tree

A decision tree is like a flow of a couple of chart structures in which an inner node presents variables. Here branch specifies a decision rule, and every one of the leaf nodes specifies the result. The upper node inside the decision tree is called the root node. It gains knowledge of the division of the attribute value. It makes division of tree in the reverse way named recursive partitioning. This chart of flow structure helps us in judgement taking. Its view shape like a flowchart diagram that comfortably copies the human-like dreaming. So for this reason decision trees are not so complex to comprehend and believe. Decision Tree inspection is a normal, guess-worthy structure tool that has software spread in various areas. Mostly, decision trees are created through a mathematical approach that defines the path to divide a data set with respect to various rules. It is a broadly used and very realistic method for supervised learning. A Decision Tree is a non-parametric supervised learning process that utilizes for both classification and regression sectors. The approach is to make a structure that defines the result of a target variable through processing simple decision regulation permitted from the data values. The decision regulations are normally in the shape of if or else statements. The deeper the tree, the more complicated the regulations and fitting the structure. In general, ideas work after every decision tree algorithm is given aside: (a) Select the best variable using variable choosing techniques to split the records. (b) Make that variable a decision node and split the dataset into little subsets. (c) Continue tree making by reprise this process recursively for every child as long as one of the terms will match i. All the tuples from to the exact variable value. ii. No, be left attributes. iii. No left-out instances.

V. DISCUSSION

After the successful implementation of the machine learning model, it is very important to measure that how effective the model is and how the model performs on the dataset that I have chosen for our research. In our, various execution parameters to figure out which machine learning algorithm will be the best option for the predicting most widely used programming language. From the dataset, I have taken 70% of the data for training purposes, and the rest 30% was used for testing. Both random state and 10 old cross-validations are deployed to find out the best possible result and the more satisfying result.

A. Performance Metrics

This paper mainly focuses on the comparison of different classification problems and from that classified performance matrix focus on classification. To predict a widely-used programming language, the tagged variable 1 means it is a positive instance and that refers to developers who have done extensive development work. On the contrary, 0 means it is a negative instance and that indicates that developers have not done extensive development work.

B. Confusion MAtrix

A confusion matrix is always recognized as an easily understandable matrix while it is arguably the most common matrix to determine the precision and rightness of a prototype. A confusion matrix is also a summary of prediction results on a classification problem. The confusion matrix structure accommodates the users to conceptualize the effects of the confusion matrix. Here instances are actual classes that are represented by each row of the matrix in. In contrast, a single column adheres to the exemplification in a pre-defined class or in the opposite way.

Table 1: Confusion Matrix

	Predictive Negative	Predictive Positive
Actual Negative	True Negative (TN)	False Positive (FP)
Actual Positive	False Negative (FN)	True Positive (TP)

1) *True Positives (TP):* In this situation both the predicted value and the actual value is correct (true) (1), i.e., while a classifier predicts that the kotlin programming language is used widely and the kotlin is that is used widely. So, the classifier is predicting the correct decision in this situation.
2) *True Negatives (TN):* True Negatives state that the predicted value and the actual value are false (0), i.e., while a classifier predicts that a C++ programming language is not widely used and (actual value) is a C++ programming language that is not widely used. So, the classifier is predicting the correct decision.
3) *False Positives (FP):* This situation refers to the predicted value is correct(true) (1) but the actual value is wrong(false) (0), i.e., while a classifier predicts that C++ programming language is widely used but the kotlin programming language is widely used. So, the classifier is unable to predict the correct decision for that case.
4) *False Negatives (FN):* False Negative states that the predicted value is false (0) but the actual value is correct(true) (1), i.e., while the classifier predicts that the kotlin programming language is not widely used but the kotlin programming language is widely used. So, the classifier is unable to predict correct decisions.

Thus, the classifier's accuracy is higher when more TP and TN are found inside the confusion matrix. Similarly, when the amount of FP and the FN increases in a Confusion Matrix. So, the best situation would be when none of the FP and FN would be founded inside the model. If it happens, the model can give us 100% accuracy.

C. Accuracy

Accuracy mean is the proportion of correct anticipation assembled by the classification data model over the complete number of anticipation that the classifier assembled. If the target variable classes in a dataset are nearly balanced, I can expect a good accuracy.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

D. Precision

Precision is called that which generates the ratio of True Positives to the summation of True Positives and False Positives. Simply high precision means that an algorithm-generated mostly appropriate results than inappropriate.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

E. Recall

The recall is to measure our model correctly identifying True Positives. A high recall shows that an algorithm-generated maximum appropriate results. The formula of recall is given below.

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

F. F1 Score

The F1 Score is that the Harmonic Mean between precision and recall. Additionally, the weighted average of precision and recall is known as the F1 score. The span of the F1 score is from 0 to 1. F1 score represents how accurate the classifier is and also shows how durable that is at the same time.

$$F1\,Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (5)$$

VI. RESULTS

A. Kotlin Programming Language

The Kotlin is a cross-platform, general-purpose, free, open-source, statically typed “pragmatic” programming language and initially designed for the JVM (Java Virtual Machine) and Android that combines object-oriented and functional programming features. The kotlin are often used for any kind of development, be it server-side, client-side web, and Android. The kotlin supports other platforms such as embedded. The Kotlin is an expressive and concise programming language that reduces common errors and it has code safety. Easily integrates into existing apps. Works on multiplatform and it is easy to learn. Every developer desires to write the least possible code and still accomplish the objective. The Kotlin allows writing the least code and so it improves app performance.

1) Support Vector Machine (SVM)

The Accuracy of SVM mode is 0.94.29

```
accuracy score: 0.9429

Classification Report:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97       325
           1       1.00      0.20      0.33        25

    accuracy                           0.94       350
   macro avg       0.97      0.60      0.65       350
weighted avg       0.95      0.94      0.92       350


Confusion Matrix:
 [[325   0]
 [ 20   5]]
Support Vector Machine Average Accuracy:        0.9286
Support Vector Machine Decision Tree Accuracy SD:               0.0143
```

2) K Neighbor Classifier (KNN)

The Accuracy of K Nearest Neighbours mode is 0.94

```
accuracy score: 0.9400

Classification Report:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97       325
           1       0.83      0.20      0.32        25

    accuracy                           0.94       350
   macro avg       0.89      0.60      0.65       350
weighted avg       0.93      0.94      0.92       350


Confusion Matrix:
 [[324   1]
 [ 20   5]]
K-Nearest Neighbour Average Accuracy:    0.9314
K-Nearest Neighbour Accuracy SD:                0.0190
```

3) Decision Tree Classifier (CART)

The Accuracy of Decision mode is 0.88

```
accuracy score: 1.0000

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       325
           1       1.00      1.00      1.00        25

    accuracy                           1.00       350
   macro avg       1.00      1.00      1.00       350
weighted avg       1.00      1.00      1.00       350


Confusion Matrix:
 [[325   0]
 [  0  25]]

Decision Tree Average Accuracy:          0.8829
Decision Tree Accuracy SD:               0.0349
```

B. Python Programming Language

Python is an interpreted, object-oriented, and high-level programming language with dynamic semantics. Python is a high-level built-in data structure, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, furthermore as to be used as a scripting language to connect existing components together. It can be used for server-side web development, software development, and system scripting handling big data, and performing complex mathematics. It also supports modules and packages and encourages code reusability.

1) Support Vector Machine (SVM)

The Accuracy of SVM mode is 0.91

```
accuracy score: 0.9171

Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.95      0.92       183
           1       0.94      0.88      0.91       167

    accuracy                           0.92       350
   macro avg       0.92      0.92      0.92       350
weighted avg       0.92      0.92      0.92       350


Confusion Matrix:
 [[174   9]
 [ 20 147]]

Support Vector Machine Average Accuracy:        0.7371
Support Vector Machine Decision Tree Accuracy SD:               0.0686
```

2) K Neighbor Classifier (KNN)

The Accuracy of K Nearest Neighbours mode is 0.77

```
accuracy score: 0.7743

Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.92      0.81       183
           1       0.88      0.61      0.72       167

    accuracy                           0.77       350
   macro avg       0.80      0.77      0.77       350
weighted avg       0.80      0.77      0.77       350


Confusion Matrix:
 [[169  14]
 [ 65 102]]

K-Nearest Neighbour Average Accuracy:    0.6286
K-Nearest Neighbour Accuracy SD:                0.0639
```

3) Decision Tree Classifier(CART)

The Accuracy of Decision mode is 0.66

```
accuracy score: 1.0000

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       183
           1       1.00      1.00      1.00       167

    accuracy                           1.00       350
   macro avg       1.00      1.00      1.00       350
weighted avg       1.00      1.00      1.00       350


Confusion Matrix:
 [[183   0]
 [  0 167]]

Decision Tree Average Accuracy:          0.6686
Decision Tree Accuracy SD:               0.0615
```

C. C# Programming Language:

C# is a strongly typed object-oriented programming language. C# is simple, modern, flexible, open-source, and versatile. Its vast popularity is attributed to Reusable components for faster software development. Data types inside C# are more versatile and error-free. C# is used to build web apps and dynamic websites using the .NET platform or other open-source platforms. Software system development needs easy-to-maintain and scalable programming languages. C# is a programming language that has all these attributes. This permits developers to form a simple adjustment and sleek maintenance.

1) Support Vector Machine (SVM)

The Accuracy of SVM mode is 0.87

```
accuracy score: 0.8743

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.96      0.91       221
           1       0.91      0.73      0.81       129

    accuracy                           0.87       350
   macro avg       0.89      0.84      0.86       350
weighted avg       0.88      0.87      0.87       350


Confusion Matrix:
 [[212   9]
 [ 35  94]]

Support Vector Machine Average Accuracy:        0.8029
Support Vector Machine Decision Tree Accuracy SD:              0.0534
```

2) K Neighbor Classifier (KNN)

The Accuracy of K Nearest Neighbours mode is 0.78

```
accuracy score: 0.7829

Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.93      0.84       221
           1       0.81      0.53      0.64       129

    accuracy                           0.78       350
   macro avg       0.79      0.73      0.74       350
weighted avg       0.79      0.78      0.77       350


Confusion Matrix:
 [[205  16]
 [ 60  69]]

K-Nearest Neighbour Average Accuracy:    0.7029
K-Nearest Neighbour Accuracy SD:                0.0514
```

3) Decision Tree Classifier (CART)

The Accuracy of Decision mode is 0.74

```
accuracy score: 1.0000

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       221
           1       1.00      1.00      1.00       129

    accuracy                           1.00       350
   macro avg       1.00      1.00      1.00       350
weighted avg       1.00      1.00      1.00       350


Confusion Matrix:
 [[221   0]
 [  0 129]]

Decision Tree Average Accuracy:          0.7429
Decision Tree Accuracy SD:               0.0599
```

D. C++ Programming Language

C++ may be a powerful general procedural, imperative computer programming language as an associate extension of the C programming language. C++ will be accustomed develop operational systems, browsers, and so on. C++ supports alternative ways of programming like procedural, object-oriented, and practical. This makes C++ powerful further as well as versatile.

1) Support Vector Machine (SVM)

The Accuracy of SVM mode is 0.79

```
Classification Report:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97       275
           1       0.98      0.77      0.87        75

    accuracy                           0.95       350
   macro avg       0.96      0.88      0.92       350
weighted avg       0.95      0.95      0.95       350


Confusion Matrix:
 [[274   1]
 [ 17  58]]

Support Vector Machine Average Accuracy:        0.7943
Support Vector Machine Decision Tree Accuracy SD:               0.0333
```

2) K Neighbor Classifier (KNN)

The Accuracy of K Nearest Neighbors mode is 0.84

```
accuracy score: 0.8400

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.99      0.91       275
           1       0.85      0.31      0.45        75

    accuracy                           0.84       350
   macro avg       0.85      0.65      0.68       350
weighted avg       0.84      0.84      0.81       350


Confusion Matrix:
 [[271   4]
 [ 52  23]]

  K-Nearest Neighbour Average Accuracy:    0.8029
  K-Nearest Neighbour Accuracy SD:                0.0518
```

3) Decision Tree Classifier (CART)

The Accuracy of Decision mode is 0.80

```
accuracy score: 1.0000

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       275
           1       1.00      1.00      1.00        75

    accuracy                           1.00       350
   macro avg       1.00      1.00      1.00       350
weighted avg       1.00      1.00      1.00       350


Confusion Matrix:
 [[275   0]
 [  0  75]]

Decision Tree Average Accuracy:          0.8000
Decision Tree Accuracy SD:               0.0527
```

Kotlin is a language that is designed to be used on an industrial scale, targeting any type of application development, be it backend, desktop, frontend, on the web as well as mobile. Kotlin is more loved by developers and is enriched with all the bases missed by Python. Kotlin as an additional language will help us target more solutions and use cases where Python is not best at. Kotlin has features that C# doesn't have. Kotlin has its capabilities to build awesome applications with less code. After applying the machine learning algorithms in three i.e Support Vector Machine, k Nearest Neighbor, and DecisionTree. SVM gives the highest accuracy 0f 94.29% for Kotlin Programming Language. So, I propose that SVM is the best-suited algorithm for the prediction of widely-used programming languages.

VII. CONCLUSION

In this paper, I have provided explanations of different Machine Learning approaches and their applications in the programming language used to analyze the data and have given very well prediction accuracy.

REFERENCES

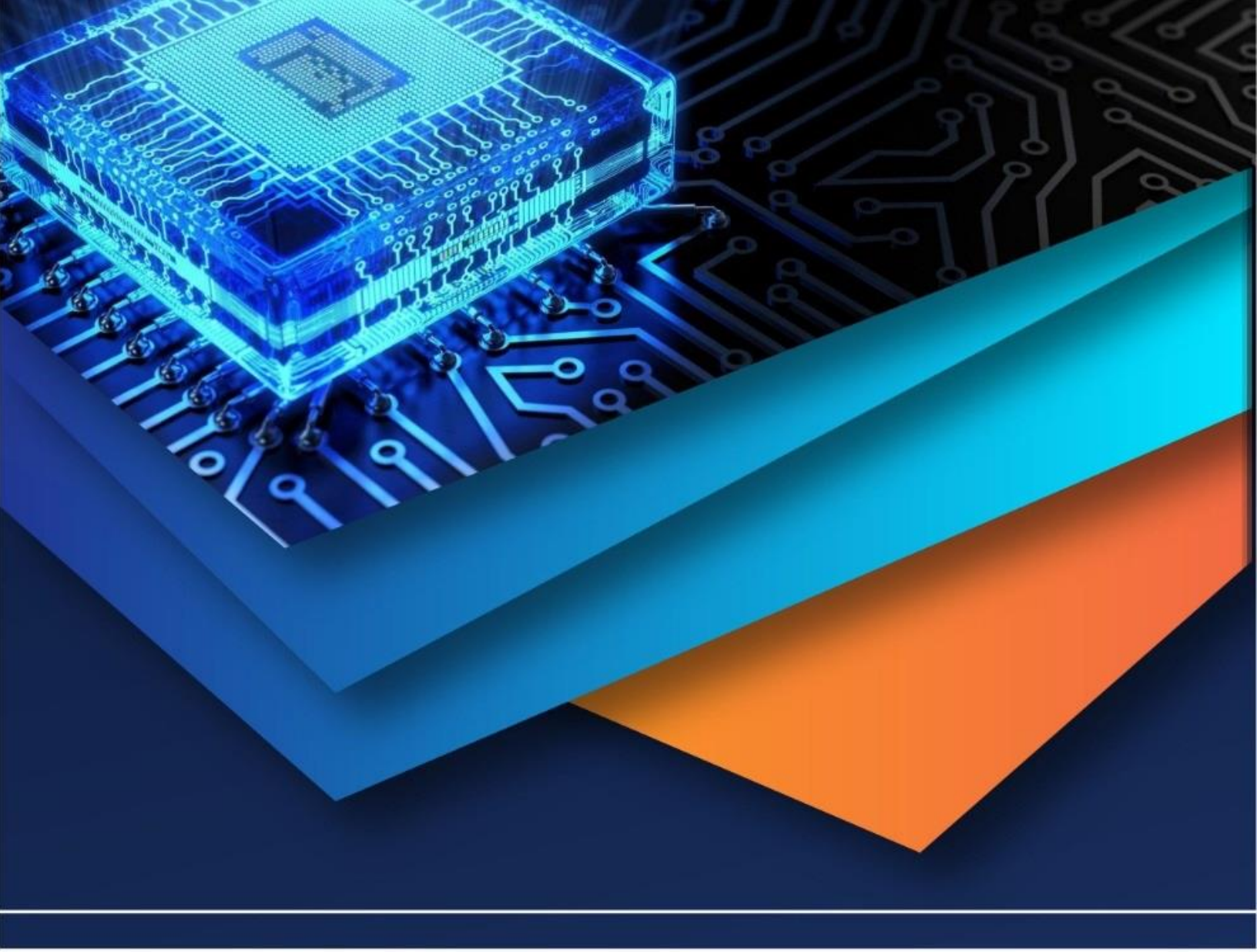
[1] Baquero, Juan F., et al. "Predicting the programming language: Extracting knowledge from stack overflow posts." Colombian Conference on Computing. Springer, Cham, 2017.

[2] Innes, Mike, et al. "On machine learning and programming languages." Association for Computing Machinery (ACM), 2018.

[3] Alreshedy, Kamel, et al. "Predicting the Programming Language of Questions and Snippets of StackOverflow Using Natural Language Processing." arXiv preprint arXiv:1809.07954 (2018).

[4] Baquero, Juan F., et al. "Predicting the programming language: Extracting knowledge from stack overflow posts." Colombian Conference on Computing. Springer, Cham, 2017.

[5] Saha, Avigit K., Ripon K. Saha, and Kevin A. Schneider. "A discriminative model approach for suggesting tags automatically for stack overflow questions." 2013 10th Working Conference on Mining Software Repositories (MSR). IEEE, 2013.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY