



INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv



Prediction of Cross-Platform and Native Apps Technology Opportunities for Beginner Developers Using C 4.5 and Naïve Bayes Algorithms

Wawan Gunawan ^{a,*}, Raychal Ababil Wiradiputra ^a, Anggi Puspita Sari ^b, Deddy Prayama ^c, Esron Rikardo Nainggolan ^d

^a Department of Informatics Technology, Universitas Mercu Buana, Jl. Meruya Selatan No 1, Jakarta, 11650, Indonesia

^b Department of Information Systems, Universitas Bina Sarana Informatika, Jl. Meruya Selatan No 1, Jakarta, 11650, Indonesia

^c Department of Information Technology, Politeknik Negeri Padang, Limau Manis, Padang, 25176, Indonesia

^d Department of Information Systems, Universitas Nusa Mandiri, Jalan Jatiwaringin No. 2, Jakarta Timur, 17411, Indonesia

Corresponding author: *wawan.gunawan@mercubuana.ac.id

Abstract—The competition between native and cross-platform app development makes application development simpler, safer, and more scalable. However, developers must have sufficient fundamentals, and the industry must conduct good research to shorten development time and minimize expenses. In order to solve these problems, this study made a prediction that discusses the technology that has a chance to survive in the industry so as not to be left behind in technology. Using Naïve Bayes and C 4.5 algorithms into a dataset with nine programming languages related to mobile app development. Results obtained in This research show Dart as a programming language that supports cross-platform frameworks and Kotlin as a programming language that supports native app frameworks is a technology that would have the opportunity in the future with an accuracy level above 90% with Naïve Bayes and C 4.5 algorithms. These results are obtained by testing an algorithm model using MAPE, consistent dataset sharing, and careful data processing. This research Can help entry-level developers learn and deepen the fundamentals of technology and can add knowledge to the industry in choosing a technology.

Keywords— Accuracy; naïve bayes; c 4,5; framework; programming language.

Manuscript received 28 Dec. 2022; revised 24 Feb. 2023; accepted 26 Apr. 2023. Date of publication 31 Dec. 2023. International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



I. INTRODUCTION

Many entry-level developers do not seem to understand learning technologies in mobile app development. They learn from current global trends and do not consider the future opportunities of technology based on an industry perspective. The preference for technology is always correlated with the time and cost spent on development. Typically, mobile apps are developed with native programming languages specific to platforms such as Java/Kotlin in the case of Android, Swift/Objective-C in the case of iOS, and C# in the case of Windows Phone. Maintaining multiple codebases for a particular platform can be time-consuming and costly. Programming languages play an important role in the development of an application, both in terms of processing time, the performance of an application, the complexity of the

application to be developed, and the cost of developing the application.

Furthermore, the JavaScript programming language is predicted to last for the next five years compared to a cross-platform framework supporting the programming language. Some studies revealed the excellence and popularity of JavaScript programming language compared to cross-platform frameworks, making development easy in terms of time and cost; it is impossible to abandon native Apps completely [1]–[3]. Related to research on students' experience in using Native Apps and non-Native Apps, where they prefer Native Apps over non-Native Apps in terms of reliability and convenience, cross-platform frameworks consume more resources, one of which is the battery by taking 6% to 8% more, despite the underlying device, compared to native apps t regardless of the device used, compared to the Native Apps themselves [4]–[6] Some things that should be considered to ensure the selection of technology in

developing an application, including the expertise of the team formed, reliability and support from vendors, UI customization, framework experience and capabilities, consistency between various platforms, security, and simplicity get the information needed in application development on the framework used [7] and a cross-platform development approach where there is no concrete software implementation in mind, but rather a common way to solve the challenge of developing a single application but running the application on multiple platforms[8].

Previous research compares the Naïve Bayes and C 4.5 algorithms with four cases and various attributes. Compare accuracy, precision, and recall values individually, then calculate the algorithm's advantages applied in each test. The results showed that, of the 4 cases, 1 case showed the same results with an accuracy value above 90%, 2 cases showed results where both algorithms were superior in each case where in the first case, the C 4.5 algorithm was superior with an average accuracy value of 95%. In the second case, the Naive Bayes algorithm was superior, with an average accuracy value of 66%. The third case of Naïve Bayes algorithm excels with an average accuracy score of 81%. The last case of the Naïve Bayes algorithm is only superior to precision with an average value of 62.72%, followed by accuracy and recall on the C 4.5 algorithm, which produces positive results with an average value of 70.84% for accuracy and 70.57% for recall. The method used to generate the data shown above is to compare accuracy, recall, and precision to evaluate or test the two algorithms.

Furthermore, for dataset division, the dataset starts from 50 data to the actual amount of data. For training and testing, data starts from the actual amount of data to 50 or close to 50. After that, a summary is made to evaluate which algorithm is superior to the two algorithms tested in the study by comparing one by one in each test directly the results without calculating the average value. Furthermore, research on the same topic and with the same number of attributes showed results that did not differ significantly between the two algorithms. In another study on the same topic with a difference of 2 classes, consisting of 2 attributes and three attributes. It turned out that the Naïve Bayes algorithm used another method with 12 attributes in 60% of training data and 40% of testing data. The results showed that the C 4.5 algorithm was better than Naïve Bayes in terms of accuracy, precision, recall, log loss, and specificity [9]–[12]. In another study, the Java cross-platform graphical interface framework presented in this paper provides a new, scalable, and flexible solution that targets different Java development application scenarios. It can be applied to develop cross-platform graphical applications for Java Standard and Micro Editions where the same code can also be executed and integrated into Android mobile applications without significant changes or readaptations of the codebase [13].

In previous studies, using Optical Character Recognition (OCR) and the Naïve Bayes algorithm, a system was created that automatically classifies whether an image contains promotional offers or not without human intervention with an accuracy of 94.31% and an average precision of 94.33% compared to using the Random Forest and KNN algorithms [14]. Previous research examined IoT's application by

predicting COVID-19 risk levels and obtained accuracy results of 99% and 3% better than Random Forest.

Furthermore, Clustering is done using the K-means algorithm as SAFE and INSECURE [15]. In a study that discusses the classification of the Naïve Bayes algorithm in predicting the smooth level of MSME core rental payments with 13 attributes (which have been classified into six attributes), the results are 81.81% accuracy, 66.66% precision, and 100% recall with an AUC value of 0.800 and processed through the Rapid Miner application [16]. In previous studies examining effectiveness between Java and Kotlin, the results found that Kotlin was superior in terms of effectiveness and consistency [17]. Another study discussing the application of QR-Code in the E-Market standalone application made based on Android stated that the application was able to facilitate the process of buying and selling transactions by 90% and sales product promotion by 90% thus increasing people's purchasing power and proving that in the future there would be many applications that are able to facilitate people in their daily lives [18]. In another study discussing Mobile Applications for Monitoring Drug Addition to Infusion Fluids, it is known that the application is designed very simply with the Java programming language that supports native app frameworks [19]. Therefore, it helps support the assumption that native app frameworks have a higher chance than cross-platform frameworks, so Kotlin would survive in the industry when it comes to Android app development.

This study implemented the Gaussian method using the Naive Bayes algorithm and C 4.5 for decision trees on programming language data to provide entry-level developers with opportunities to develop mobile-based applications. The C 4.5 algorithm is the algorithm that forms the decision tree because it is a well-known method of classification and prediction, and the Naïve Bayes algorithm is famous for predicting the likelihood that it would occur in the future even with minimal data in the past. In previous research conducted by Kurniawan [9], the method used in this study involves a method of comparing each test value, and the result obtained is how many algorithms are superior after being compared, then dividing the dataset using integers and consistent, but for testing, it is inconsistent because the data in each case study is different.

In this study, the method applied is a dataset division method that varies from 90% training data and 10% testing data to 90% training data and 10% testing data with 9 attributes in terms of accuracy, recall, precision, average calculation and Mean Absolute Percentage Error (MAPE) applied to evaluate the feasibility level of the algorithm, after that the results are compared to predict opportunities in terms of programming languages that support Cross-Platform frameworks and Native Apps, such as Dart, JavaScript, Kotlin, C, C++, C#, Java, Swift, Objective-C, and others. Next step, this study also aims to determine the opportunities of several programming languages that have the potential to be most widely adopted in the mobile application industry so that entry-level developers can learn and adjust and have the skills to be able to survive and develop in the industrial world. While in the realm of industry, can be a reference in determining what technology would be used in developing

mobile-based applications. For a more detailed comparison between the previous study and mine can be seen in Table 1.

TABLE I
COMPARISON METHOD

No.	Method	Research	
		Previous	Now
1.	Dataset division	Divided by an integer	Divided by percentage
2.	Model validation	None	MAPE (statistical)
3.	Comparison	One by one	One by one
4.	Case study implementation	Inconsistent	Consistent
5.	Attributes	Random (based on case studies)	Consistent

II. MATERIAL AND METHOD

A. Data Mining

Data mining, officially known as knowledge discovery in Databases (KDD), describes a relatively new technique of using iterative and interactive processes to gain new knowledge/patterns/models using large amounts of existing data. KDD involves investigating models in big data sets using techniques at the intersection of machine learning, statistics, and database systems. It facilitates pattern checking, such as data categorization through cluster studies, abnormal record recognition, also known as anomaly detection, and related rules or dependencies [20], [21]. In a previous data mining study, this research yielded new insights into the graduation data of students at Garuda High School for further processing [22].

B. Classification

Classification refers to finding models to visualize and differentiate data classes to predict the class of an object from unknown class labels. The method consists of two processes, including the training stage, which analyzes the training data and then generates classification rules. The second step is classification, where test data is used to test the accuracy of classification rules. Previous studies used classifications that provide classes, which would later be used to predict student enrollment activities [23].

C. Prediction

Prediction is the first stage of the decision-making process. It is important to know the actual problems in decision-making to predict. Prediction means the idea of quantity. For example, market demand for one or more products in the future. This is an important part of medical decision-making in medical research involving prediction. Choosing the preferred treatment plan may be the same as choosing the treatment plan with the best predictions [24].

Past data is systematically integrated through specific methods and processed to produce future conditions. The purpose of prediction is to provide insight to decision-makers and policymakers regarding potential uncertainties and risks that may arise and can be considered in planning. By making such predictions, planners and decision-makers can consider alternatives. The predicted results may not be precise due to the uncertainty of future circumstances or events. However, assuming all factors are determined correctly, the predicted

result would be close to the expected result. In previous studies using predictions, the results obtained can be considered in the future [22].

D. C 4.5 Algorithms

The C 4.5 algorithm is arguably one of the most efficient decision tree algorithms for classification. This algorithm is used to generate a decision tree. The decision tree describes the prediction of the target variable with the basic idea of relying on selection attributes with the highest priority or having the highest gain value based on entropy. Related research using the C 4.5 algorithm method discussed developing a predictive model to prevent diabetes as early as possible with 49 data records and five attributes, including age, blood pressure, pulse, weight, and blood sugar levels. Then, the results obtained are 90% accuracy [25]. The stages of this algorithm are shown below [23].

- 1) Determine the Entropy value:

$$Entropy(S) = \sum_{i=1}^n - P_i \times \log_2 P_i \quad (1)$$

Notes:

S : The set of cases.

n : The number of partitions S.

pi : The proportion of Si towards S.

- 2) Determine the Information Gain value of each attribute:

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} \times Entropy(S_i) \quad (2)$$

Notes:

S : The set of cases.

A : Attribute.

n : The number of attribute partitions A.

|Si| : Number of cases in partition i.

|S| : Number of cases within S.

- 3) Estimate the Split Info value of each attribute:

$$Split\ Info(S,A) = - \sum_{i=1}^n \frac{S_i}{S} \log_2 \frac{S_i}{S} \quad (3)$$

Notes:

S : The set of cases.

A : Attribute.

Si : Number of samples for attribute i.

- 4) Determine the Gain Ratio value of each attribute:

$$Gain\ Ratio = \frac{Gain(S,A)}{Split\ Info(S,A)} \quad (4)$$

Notes:

S : The set of cases.

A : Attribute.

Gain(S, A) : Information of gain on attribute A.

Split Info(S, A) : Information of gain on attribute A.

The attribute with the highest Profit Ratio is considered root. After that, it is necessary to calculate the Profit Ratio value of each attribute that does not support the root. Otherwise, the attribute is considered a branch when the Profit Ratio is lower than the root Gain Ratio.

E. Naïve Bayes Algorithms

The Naïve Bayes algorithm refers to the Bayes theorem or a simplification technique based on the Bayesian algorithm originally proposed by Thomas Bayes, one of the British

scientists who intended to build a classifier based on a conditional probability model. The Naïve Bayes Classifier model is a classification method that indicates the model has a decent accuracy, precision, recall, and f1 value. The major feature of this method is the independence assumption of each situation and considers selecting the best class marker according to this probability and the loss of valuation error [26]–[29]. A related study involving the Naive Bayes algorithm method discusses whether a person is eligible to get a money loan from credit or not given with 10 data records and nine attributes, including name, gender, age, type of occupation, loan amount, repayment period, guarantee, income, and category. Then, the results showed that this system scored a precision value of 82%, an accuracy value of 80%, and a recall value of 94%, which implicitly said that this system could be considered successful in predicting a person's eligibility to get a loan from credit [30].

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \quad (5)$$

Notes:

- X : Data with an unknown class
- H : Data hypothesis X is a specific class
- P(H|X) : Probability of hypothesis H based on condition x
- P(H) : Probability of hypothesis H (before evidence is observed)
- P(X|H) : Probability of X based on the condition.
- P(X) : The probability of X

F. Programming Languages

A programming language refers to standard instructions on instructing a computer program that serves a specific function. It is defined as a set of syntax and semantic rules used to define computer programs. This language allows a

programmer to specify what data would be processed by the computer, the way this data would be stored/forwarded, and the actions to be initiated in various situations [31].

G. Mean

Arithmetic means a value that is commonly referred to as the average value. This value is gained by summing up the whole value and dividing it by the number of values. The formula [32], [33] estimates the mean value.

$$\bar{x} = \frac{x_1+x_2+\dots+x_n}{n} \quad (6)$$

Notes:

- \bar{x} = average
- x = the data to be found, average
- n = amount of data

H. Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) is estimated by dividing the absolute error over each period by the actual observed value within the period. Therefore, this approach comes in handy as the size or magnitude of the prediction variable is critical in evaluating the accuracy of the prediction. MAPE represents the amount of error in the prediction compared to the actual value of the series. MAPE also compares the accuracy of the same or opposite methods on two different series and measures the accuracy of the model's estimated value denoted by MAPE [34].

$$MAPE = \frac{\sum_{t=1}^n \left| \frac{a-b}{a} \right|}{n} \times 100\% \quad (7)$$

Notes:

- n = total data
- a = actual value
- b = estimation value

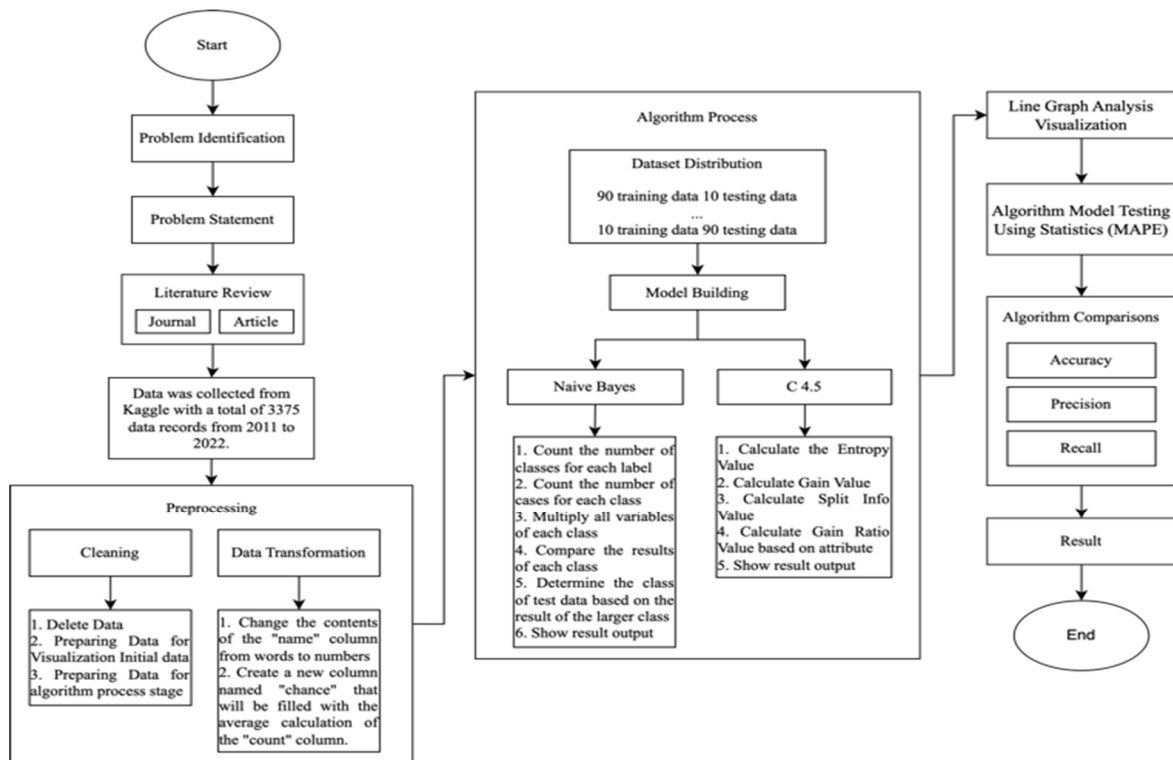


Fig. 1 Stages of the research

In Figure 1, the calculation process of the C 4.5 algorithm uses formulas number 1 until number 4. Later, the Naïve Bayes algorithm calculation process uses formula number 5, while the calculation of Mean Absolute Percentage Error (MAPE) uses formula number 7.

I. Dataset

Referring to Figure 1, the collected data consists of programming language data, which is 3375 data from 2011 to 2021 extracted from the website <https://www.kaggle.com>. The data represents secondary data as it was processed by an open data provider (Kaggle) before publication. Table II shows the initial dataset. This dataset is utilized to analyze the possibilities in the future related to the changes in programming languages for developing mobile-based applications.

TABLE II
DATASET

	Name	Year	Quarter	Count
0	Ruby	2011	3	965
1	JavaScript	2011	3	694
2	Python	2011	3	619
3	PHP	2011	3	560
4	Java	2011	3	420
...
3370	Go	2022	1	204
3371	PHP	2022	1	174
3372	TypeScript	2022	1	169
3373	C	2022	1	151
3374	C #	2022	1	148

J. Pre-Processing dan Labeling

Based on Figure 1, the pre-processing used is data cleaning and data transformation to remove irrelevant columns/data and convert the data from character to numeric to facilitate the computational process. Table III displays the pre-processing results.

TABLE III
PRE-PROCESSING PROCESS

	Name	Year	Count
207	1	2013	124
249	1	2013	272
304	1	2013	241
364	1	2013	624
412	1	2014	627

Based on Figure 1, the labeling process is applied to determine whether the programming language is possible by calculating the average in each programming language's "count" column. Then, it is converted from characters to numbers to facilitate the computation process. Table IV displays the labeling results.

TABLE IV
LABELING PROCESS

	Name	Year	Count	Chance
1	1	2013	124	0
2	1	2013	272	0
3	1	2013	241	0
4	1	2013	264	0
5	1	2014	672	0
...
32	1	2020	13641	1
33	1	2021	3973	1
34	1	2021	13126	1
35	1	2021	1041	1
36	1	2021	7782	1

III. RESULT AND DISCUSSION

Precision, recall, and accuracy value comparison for Naïve Bayes and C 4.5 models can be seen in Table V.

TABEL V
ALGORITHM PROCESS RESULTS WITH 90% TRAINING 10% TESTING

Programming language	Naïve Bayes Algorithm			C 4.5 Algorithm		
	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)
Dart	100	100	100	100	100	100
JavaScript	100	67	75	100	100	100
Kotlin	100	100	100	100	100	100
C	50	100	50	100	100	100
C++	100	100	100	100	100	100
C #	100	100	100	100	100	100
Java	100	100	100	100	100	100
Swift	100	100	100	100	100	100
Objective-c	100	100	100	100	100	100
Average	94.44	96.33	91.67	100	100	100

TABEL VI
ALGORITHM PROCESS RESULTS WITH 10% TRAINING 90% TESTING

Programming language	Algorithm Naïve Bayes			C 4.5 Algorithm		
	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)
Dart	94	100	100	100	100	100
JavaScript	0	0	75	100	100	100
Kotlin	0	0	100	100	100	100
C	55	100	50	100	100	100
C++	61	100	100	100	100	100
C #	55	100	100	100	100	100
Java	33	100	100	100	100	100
Swift	0	0	100	100	100	100
Objective-c	0	0	100	100	100	100
Average	33.11	55.56	56.39	63.11	100	100

The results of 10% testing data are based on table V. The C 4.5 algorithm excels with an average accuracy rate of 100% for nine programming languages and for the Naïve Bayes algorithm, gaining 75% accuracy for the JavaScript programming language and 50% for the C programming language, as for the 90% testing data based on Table VI, the C 4.5 algorithm excels with an average accuracy of 71%, with the Kotlin, Java, and Swift programming languages being the top 3 with over 80% accuracy and for the Naïve Bayes algorithm getting the average accuracy rate of 56% with the dart programming language as being the highest accuracy rate.

TABEL VII
ALGORITHM PROCESS RESULTS WITH 80% TRAINING 20% TESTING

Programming language	Naïve Bayes Algorithm			C 4.5 Algorithm		
	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)
Dart	100	100	100	100	100	100
JavaScript	100	100	75	100	100	100
Kotlin	100	100	100	100	100	100
C	83	100	50	100	100	100
C++	100	100	100	100	100	100
C #	100	100	100	100	100	100
Java	100	100	100	100	100	100
Swift	100	100	100	100	100	100
Objective-c	100	100	100	100	100	100
Average	98.11	100	98,61	100	100	100

TABLE VIII
ALGORITHM PROCESS RESULTS WITH 20% TRAINING 80% TESTING

Programming language	Naïve Bayes Algorithm			C 4.5 Algorithm		
	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)
Dart	92	100	100	100	100	100
JavaScript	79	100	75	100	100	100
Kotlin	100	81	100	100	100	100
C	100	39	50	100	100	100
C++	100	79	100	100	100	100
C #	80	100	100	100	100	100
Java	100	73	100	100	100	100
Swift	100	46	100	100	100	100
Objective-c	100	92	100	100	100	100
Average	94.56	78.89	85.70	90.67	100	100

The 20% testing data results are based on table VII. C 4.5 algorithm and Naïve Bayes algorithm are balanced with 98% average accuracy, with C and C# programming languages ranked in the bottom 2 with 87% accuracy. Furthermore, with 80% testing data based on Table VIII, the C 4.5 algorithm excels with a 92% average accuracy rate, with the Kotlin, C, C++, C#, Java, and Dart programming languages being the top 5 with over 95% accuracy and for the Naïve Bayes algorithm getting a level of accuracy with an average of 85% with the Dart and Objective-C programming languages being the top 2 with 96% accuracy.

TABLE IX
ALGORITHM PROCESS RESULTS WITH 70% TRAINING 30% TESTING

Programming language	Naïve Bayes Algorithm			C 4.5 Algorithm		
	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)
Dart	86	100	100	100	100	100
JavaScript	100	100	75	100	100	100
Kotlin	100	100	100	100	100	100
C	100	88	50	100	100	100
C++	100	100	100	100	100	100
C #	60	100	100	100	100	100
Java	100	100	100	100	100	100
Swift	100	100	100	100	100	100
Objective-c	100	67	100	100	100	100
Average	94	95	93.94	95.89	100	100

TABLE X
ALGORITHM PROCESS RESULTS WITH 30% TRAINING 70% TESTING

Programming language	Naïve Bayes Algorithm			C 4.5 Algorithm		
	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)
Dart	100	100	100	100	100	100
JavaScript	100	100	75	100	100	100
Kotlin	100	100	100	100	100	100
C	81	100	50	100	100	100
C++	64	100	100	100	100	100
C #	82	100	100	100	100	100
Java	41	100	100	100	100	100
Swift	100	64	100	100	100	100
Objective-c	76	100	100	100	100	100
Average	82.67	96	85.39	93.78	100	100

For the 30% testing data results based on table IX. The C 4.5 algorithm is 3% superior to the Naïve Bayes algorithm with an average accuracy rate of 96% and 93% with the Dart, C, C#, Java, Swift, Objective-C programming languages and 100% accuracy rate for the C 4.5 algorithm and the JavaScript, Kotlin, C++, Java, Swift programming languages. While with 70% of testing data based on table X, the C 4.5 algorithm outperformed 9% with an average accuracy rate of 94% and 85% with JavaScript, C ++, C#, Java, Swift programming language with a 100% accuracy rate for the C

4.5 algorithm and Dart, JavaScript, and Kotlin programming languages with a 100% accuracy rate for the Naïve Bayes algorithm.

TABLE XI
ALGORITHM PROCESS RESULTS WITH 60% TRAINING 40% TESTING

Programming language	Naïve Bayes Algorithm			C 4.5 Algorithm		
	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)
Dart	86	100	100	100	100	100
JavaScript	100	83	75	100	100	100
Kotlin	100	100	100	100	100	100
C	82	100	50	100	100	100
C++	89	100	100	100	100	100
C #	100	100	100	100	100	100
Java	100	100	100	100	100	100
Swift	100	100	100	100	100	100
Objective-c	100	100	100	100	100	100
Average	95.22	98.11	96.22	95.56	100	100

TABLE XII
ALGORITHM PROCESS RESULTS WITH 40% TRAINING 60% TESTING

Programming language	Naïve Bayes Algorithm			C 4.5 Algorithm		
	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)
Dart	92	100	100	100	100	100
JavaScript	53	100	75	100	100	100
Kotlin	73	100	100	100	100	100
C	82	100	50	100	100	100
C++	65	100	100	100	100	100
C #	94	100	100	100	100	100
Java	100	80	100	100	100	100
Swift	100	90	100	100	100	100
Objective-c	69	100	100	100	100	100
Average	80.89	96.67	84.80	95.11	100	100

In the results of 40% test data based on table XI. The C 4.5 algorithm performed 1% ahead of the Naïve Bayes algorithm with average accuracy of 97% and 96%, indicating both algorithms almost balanced with Kotlin, C, C++, Java, Swift, and Objective-C programming languages being the top 5 with 100% accuracy for the C 4.5 algorithm and Kotlin, C#, Java, Swift, and Objective-c programming languages with 100% accuracy for the Naïve Bayes algorithm. Furthermore, with 60% testing data based on Table XII, the C 4.5 algorithm outperforms the Naïve Bayes algorithm by 7% with an average accuracy of 91% and 84% with the Dart, Kotlin, Java, and Swift programming languages with 100% accuracy for the C 4.5 algorithm and the Dart, C#, and Swift programming languages being the top 3 with an accuracy rate above 90% for the Naïve Bayes algorithm.

TABLE XIII
ALGORITHM PROCESS RESULTS WITH 50% TRAINING 50% TESTING

Programming language	Naïve Bayes Algorithm			C 4.5 Algorithm		
	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)
Dart	88	100	100	100	100	100
JavaScript	100	100	75	100	100	100
Kotlin	100	100	100	100	100	100
C	90	100	50	100	100	100
C++	100	92	100	100	100	100
C #	90	100	100	100	100	100
Java	80	100	100	100	100	100
Swift	100	100	100	100	100	100
Objective-c	90	100	100	100	100	100
Average	93.11	99.11	96.24	96.11	100	100

In the 50% testing data results based on table XIII. The C 4.5 algorithm outperforms the Naïve Bayes algorithm by 1% with an average accuracy rate of 97% and 96%, which means

both algorithms are almost balanced, with the Dart and C programming languages being the bottom 2 with an accuracy rate of 94% and 83% for the C 4.5 algorithm and the JavaScript, Kotlin, and Swift programming languages being the top 3 with an accuracy rate of 100% for the Naïve Bayes algorithm.

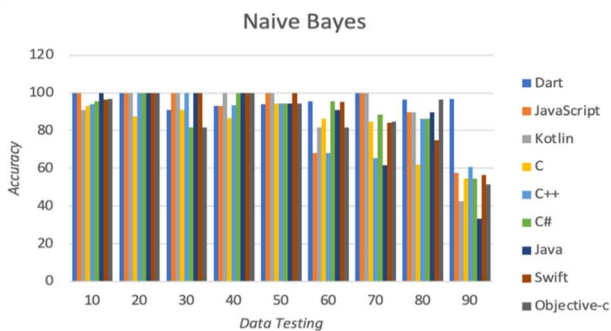


Fig. 2 Data visualization of Naïve Bayes results

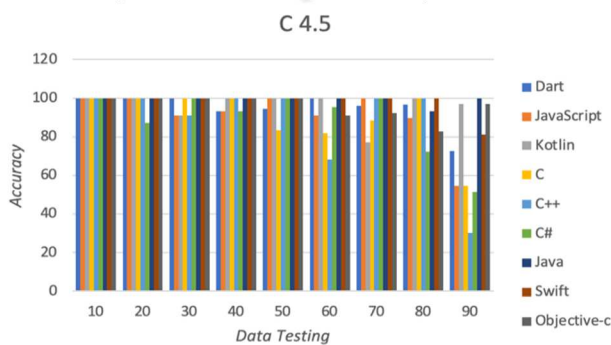


Fig. 3 Data visualization of C 4.5 results

Figure 2 states that the results of the Naïve Bayes algorithm, if the testing data is more than 90%, 6 out of 9 programming languages show an accuracy value below 80%, even up to 30%. Meanwhile, in Picture 3, it is revealed that the results of the C 4.5 algorithm, if the testing data is more than 70%, 8 out of 9 programming languages show an accuracy value below 80%, even up to 30%. The comparison of precision, recall, and accuracy values for Naïve Bayes and C 4.5 models can be seen in Table XIII, referring to Table V to Table XIII.

TABLE XIV
DATA RECAPITULATION OF PRECISION, RECALL, AND ACCURACY BASED ON AVERAGE

Data Testing	Precision	Recall	Accuracy
10%	C 4.5	C 4.5	C 4.5
20%	C 4.5	Naive Bayes	Naive Bayes
30%	C 4.5	C 4.5	C 4.5
40%	C 4.5	C 4.5	C 4.5
50%	C 4.5	C 4.5	C 4.5
60%	C 4.5	Naive Bayes	C 4.5
70%	C 4.5	C 4.5	C 4.5
80%	Naive Bayes	C 4.5	C 4.5
90%	C 4.5	C 4.5	C 4.5

Table XIV shows that in 20% of the testing data, the Naïve Bayes model excels in recall and accuracy values. In 60% of data testing, the Naïve Bayes model only excels in the recall value. In 80% of testing data, the Naïve Bayes model only excels in precision value. The C 4.5 model excels in the precision, recall, and overall accuracy values on the other testing data. Whereas previous research used 12 attributes

with 40% testing data distribution, the C 4.5 algorithm excels in precision, recall, and accuracy. The results of this study, with nine attributes and 40% testing data distribution, show that the C 4.5 algorithm is still ahead regardless of the difference in attributes.

Then, according to Figure 2 and Figure 3, the calculation of the average for each programming language and test data is carried out. Next, the Mean Absolute Percentage Error is calculated to see how well the prediction on each model is used. Then, sort by highest to lowest accuracy for each programming language followed by the Mean Absolute Percentage Error in each programming language. Table XV presents a reference to see the feasibility of the algorithm model [35], followed by Table XVI presents a reference to see the feasibility of the algorithm accuracy [36], and Table XVII presents the results of the above calculations.

TABLE XV
MAPE SCOPE GUIDANCE

Range MAPE	Description
< 10%	Forecasting model capability is very good
10 – 20%	Forecasting model capability is good
20 – 50%	Forecasting model capability is worthy
> 50%	Forecasting model capability is poor

TABLE XVI
ACCURACY SCORE SCOPE GUIDANCE

Accuracy Score	Group
0,90 – 1,0	Excellent
0,80 – 0,90	Good
0,80 – 0,70	Worthy
0,70 – 0,60	Less
< 0,60	Poor

TABLE XVII
ACCURACY RECAPITULATION DATA BASED ON AVERAGE VALUE WITH MAPE

Programming language	Accuracy (%)		MAPE (%)	
	Naive Bayes	C 4.5	Naive Bayes	C 4.5
Dart	96.30	94.80	37	37
Kotlin	90.42	96.09	41	41
Swift	90.05	97.92	53	53
C #	88.99	88.86	31	31
Objective-c	87.86	95.88	36	36
JavaScript	87.02	91.04	37	37
Java	85.54	99.23	32	32
C++	85.35	87.71	23	23
C	77.45	89.79	22	22

After that, evidence was collected by comparing the actual data in 2015 with the data predicted to increase in 2022. Table XVIII displays a comparison of the data for the two years.

TABLE XVIII
COMPARISON OF ACTUAL DATA WITH FORECASTING DATA

Name	Year		Percentage	Condition
	2015	2022		
Dart	16.060,00	45.922,00	65%	↑
JavaScript	1.036.613,00	275.437,00	-73%	↓
Kotlin	18.110,00	19.200,00	6%	↑
C	114.195,00	90.268,00	-21%	↓
C++	206.343,00	168.453,00	-18%	↓
C #	161.275,00	102.178,00	-37%	↓
Java	441.422,00	228.703,00	-48%	↓
Swift	36.920,00	15.053,00	-59%	↓
Objective-c	71.513,00	12.232,00	-83%	↓

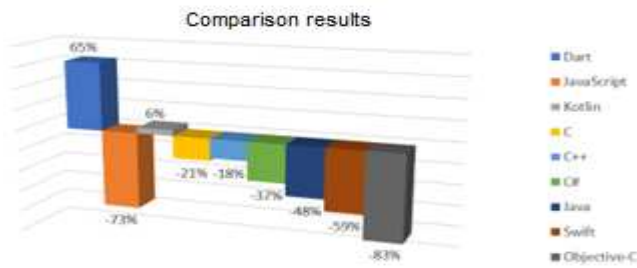


Fig. 4 Comparison results data

A previous study conducted in 2018 predicted that the JavaScript programming language would endure for the next five years, proven by the number of frameworks that support the programming language. Nevertheless, the results of this study, by comparing data in 2015 as actual data and 2022 as predicted data in Table XVIII, indicate the Dart programming language at 65% and Kotlin 6% are proven to improve in 2022 and possess future opportunities for mobile application development and data visualization as shown in Figure 4.

IV. CONCLUSION

Many entry-level developers who are unsure where to learn the technology as the basis for developing mobile applications are concerned that focusing on one technology may not necessarily be the technology that would survive for the next 5 years. To deal with this, an application model is developed to predict what programming languages will likely endure. This model was developed using 2 different algorithms, namely the Naïve Bayes and C 4.5 algorithms, with various testing data ranging from 10% to 90%. Afterward, the results of the process were carried out, and the performance of the C 4.5 algorithm exceeded the overall average in precision, recall, and accuracy values. Referring to the data in Table VII, the Dart and Kotlin programming languages are technologies that would have opportunities in the future with an accuracy rate above 90% with both the Naïve Bayes and C 4.5 algorithms and the qualified technology used by these programming languages is the native apps framework for Kotlin which is expected to increase by 6% and the cross-platform framework for Dart which is expected to increase by 65%. Therefore, it is expected that entry-level developers may begin to learn the basics to fundamentals in depth to develop a mobile application using cross-platform technology.

REFERENCES

- [1] M. Ilhami, "Tren dan Peluang Cross-Platform Mobile App untuk Developer Pemula," *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, vol. 1, no. 2, pp. 402-411, Des. 2021.
- [2] K. Gunnarsson and O. Herber, "The Most Popular Programming Languages of GitHub's Trending Repositories," DiVA, to be published. DiVA, id: diva2:1463849.
- [3] "Stack Overflow Developer Survey 2022." <https://survey.stackoverflow.co/2022/#technology-top-paying-technologies> (accessed Oct. 28, 2022).
- [4] T. Dorfer, L. Demetz, and S. Huber, "Impact of mobile cross-platform development on CPU, memory and battery of mobile devices when using common mobile app features," *Procedia Comput Sci.*, vol. 175, pp. 189-196, Aug. 2020.
- [5] P. M. Léger, T. An Nguyen, P. Charland, S. Sénécal, H. G. Lapierre, and M. Fredette, "How Learner Experience and Types of Mobile Applications Influence Performance: The Case of Digital Annotation," *Computers in the Schools*, vol. 36, no. 2, pp. 83-104, Apr. 2019.
- [6] J. Z. Blanco and D. Lucrédio, "A holistic approach for cross-platform software development," *Journal of Systems and Software*, vol. 179, p. 110985, Sep. 2021, doi: 10.1016/j.jss.2021.110985.
- [7] "The Six Best Cross-Platform App Development Frameworks | Kotlin." <https://kotlinlang.org/docs/cross-platform-frameworks.html#8-educational-materials> (accessed Dec. 16, 2022).
- [8] C. Rieger and T. A. Majchrzak, "Towards the definitive evaluation framework for cross-platform app development approaches," *Journal of Systems and Software*, vol. 153, pp. 175-199, Jul. 2019.
- [9] Y. I. Kurniawan, "Perbandingan Algoritma Naive Bayes dan C.45 dalam Klasifikasi Data Mining," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 4, pp. 455-464, Oct. 2018.
- [10] N. Nanni and A. Sudransyah, "Perbandingan Kinerja Algoritma Naive Bayes dan C4.5 Untuk Klasifikasi Harga Pangan," *PROtek : Jurnal Ilmiah Teknik Elektro*, vol. 7, no. 1, pp. 20-24, May. 2020.
- [11] H. Sulaiman, "Perbandingan Algoritma Decision Tree C4.5 Dan Naive Bayes pada Analisa Tekstur Gray Level Co-Occurrence Matrix Menggunakan Citra Wajah," *SISTEMASI*, vol. 10, no. 2, pp. 470-479, May. 2021.
- [12] K. L. Kohsasih and Z. Situmorang, "Analisis Perbandingan Algoritma C4.5 dan Naive Bayes Dalam Memprediksi Penyakit Cerebrovascular," *Jurnal Informatika*, vol. 9, no. 1, pp. 13-17, Apr. 2022.
- [13] S. Cirani, M. Picone, L. Veltri, L. Zaccomer, and F. Zanichelli, "ZWT: A new cross-platform graphical interface framework for Java applications," *SoftwareX*, vol. 12, p. 100599, Jul. 2020, doi: 10.1016/j.softx.2020.100599.
- [14] Hubert, P. Phoenix, R. Sudaryono, and D. Suhartono, "Classifying Promotion Images Using Optical Character Recognition and Naive Bayes Classifier," *Procedia Comput Sci.*, vol. 179, pp. 498-506, Jan. 2021.
- [15] N. Deepa, J. Sathya Priya, and T. Devi, "Towards applying internet of things and machine learning for the risk prediction of COVID-19 in pandemic situation using Naive Bayes classifier for improving accuracy," *Mater Today Proc.*, vol. 62, pp. 4795-4799, Jan. 2022.
- [16] R. Rachman and R. N. Handayani, "Analisis Perbandingan Algoritma C4.5 dan Naive Bayes Dalam Memprediksi Penyakit Cerebrovascular," *Jurnal Informatika*, vol. 8, no. 2, pp. 111-122, Aug. 2021.
- [17] L. Ardito, R. Coppola, G. Malnati, and M. Torchiano, "Effectiveness of Kotlin vs. Java in android app development tasks," *Inf Softw Technol.*, vol. 127, p. 106374, Nov. 2020.
- [18] D. Ramayanti, W. Gunawan, and I. I. Faishal, "Implementasi QR-Code pada Aplikasi E-Market Mandiri untuk Pemberdayaan Ekonomi Kreatif Berbasis Android," *Jurnal Informatika*, vol. 8, no. 1, pp. 34-40, Feb. 2021.
- [19] V. Ayumi, "Mobile Application for Monitoring of Addition of Drugs to Infusion Fluids," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 48-56, Nov. 2019, doi: 10.32628/cseit195616.
- [20] R. Rastogi and M. Bansal, "Diabetes prediction model using data mining techniques," *Measurement: Sensors*, vol. 25, p. 100605, Feb. 2023, doi: 10.1016/j.measen.2022.100605.
- [21] D. L. Aditya and D. Fitriah, "Comparative study of fuzzy c-means and k-means algorithm for grouping customer potential in brand limback," *Jurnal Riset Informatika*, vol. 3, no. 4, pp. 327-334, Sep. 2021.
- [22] E. Yaodah Kodratillah, Daririn and C. Naya, "Penerapan data mining untuk prediksi kelulusan siswa menggunakan algoritma naive bayes pada SMK Garuda," *Jurnal Teknologi Pelita Bangsa*, vol. 12, no. 4 pp. 201-206, Des. 2021.
- [23] N. Yahya and A. Jananto, "Komparasi Kinerja Algoritma C. 45 Dan Naive Bayes Untuk Prediksi Kegiatan Penerimaanmahasiswa Baru (Studi Kasus: Universitas Stikubank Semarang)," *Proceeding SENDI U.*, pp. 221-228, Jul. 2019.
- [24] M. W. Kattan and T. A. Gerds, "A Framework for the Evaluation of Statistical Prediction Models," *Chest*, vol. 158, no. 1, pp. S29-S38, Jul. 2020.
- [25] S. Ucha Putri, E. Irawan, F. Rizky, "Implementasi Data Mining Untuk Prediksi Penyakit Diabetes Dengan Algoritma C4.5," *Kesatria : Jurnal Penerapan Sistem Informasi (Komputer dan Manajemen)*, vol. 2, no. 1, pp. 39-46, Jan. 2021.
- [26] T. A.M and A. Yaqin, "Perbandingan Algoritma Naive Bayes, K-Nearest Neighbors dan Random Forest untuk Klasifikasi Sentimen Terhadap BPJS Kesehatan pada Media Twitter," *InComTech : Jurnal Telekomunikasi dan Komputer*, vol. 12, no. 1, pp. 01-10, Apr. 2022.

- [27] Z. Ye, P. Song, D. Zheng, X. Zhang, and J. Wu, "A Naive Bayes model on lung adenocarcinoma projection based on tumor microenvironment and weighted gene co-expression network analysis," *Infect Dis Model.*, vol. 7, no. 3, pp. 498–509, Sep. 2022.
- [28] B. Ruyu, H. Mo, and L. Haifeng, "A Comparison of Credit Rating Classification Models Based on Spark- Evidence from Lending-club," *Procedia Comput Sci.*, vol. 162, pp. 811–818, Jan. 2019.
- [29] D. Fitriyah, D. Fitriyah, W. Gunawan, and A. P. Sari, "Studi Komparasi Algoritma Klasifikasi C5.0, SVM dan Naive Bayes dengan Studi Kasus Prediksi Banjir," *Techno.Com.*, vol. 21, no. 1, pp. 1–11, Feb. 2022.
- [30] D. A. Kurniawan and Y. I. Kurniawan, "Aplikasi Prediksi Kelayakan Calon Anggota Kredit Menggunakan Algoritma Naïve Bayes," *Jurnal Teknologi dan Manajemen Informatika*, vol. 4, no. 1, Jan. 2018, doi: 10.26905/jtmi.v4i1.1831.
- [31] "Bahasa Pemrograman - Pengertian, Jenis, Macam, Tingkatan & Contoh." <https://www.dosenpendidikan.co.id/bahasa-pemrograman/> (accessed Aug. 17, 2022).
- [32] Morissan, "Karakteristik Data," in *Statistik Sosial*, 1st ed. Jakarta, Indonesia: KENCANA, 2016, Accessed on: Des. 11, 2022.
- [33] H. H. Elmousalami and A. E. Hassanien, "Day Level Forecasting for Coronavirus Disease (COVID-19) Spread: Analysis, Modeling and Recommendations," doi:10.48550/arxiv.2003.07778.
- [34] S. Prayudani, A. Hizriadi, Y. Y. Lase, Y. Fatmi, and Al-Khowarizmi, "Analysis Accuracy Of Forecasting Measurement Technique On Random K-Nearest Neighbor (RKNN) Using MAPE And MSE," *Journal of Physics: Conference Series*, vol. 1361, no. 1, p. 012089, Nov. 2019, doi: 10.1088/1742-6596/1361/1/012089.
- [35] "Cara Menghitung Mean Absolute Percentage Error (MAPE) - khoiri.com." <https://www.khoiri.com/2020/12/pengertian-dan-cara-menghitung-mean-absolute-percentage-error-mape.html> (accessed Dec. 10, 2022).
- [36] A. T. Nugraha, "Aplikasi pemesanan travel menggunakan Chatbot dengan Machine Learning," 2020, Accessed: Jan. 17, 2023. [Online]. Available: <http://etheses.uin-malang.ac.id/20994/>.