Project Risk Management With Logistic Regression and Gaussian Naïve Bayes

Minh Thanh, Nguyen
MSIS3033.N22.CTTT
University of Information Technology
20521920@gm.uit.edu.vn

Van Tan, Nguyen
MSIS3033.N22. CTTT
University of Information Technology
20521880@gm.uit.edu.vn

Truong Thin, Tong
MSIS3033.N22. CTTT
University of Information Technology
20521958@gm.uit.edu.vn

Vinh Quang, Quach
MSIS3033.N22. CTTT
University of Information Technology
20521811@gm.uit.edu.vn

Van Sy, Le
MSIS3033.N22. CTTT
University of Information Technology
20521854@gm.uit.edu.vn

***Abstract*—Project risk management is a critical process for organizations to ensure the successful delivery of projects in today's competitive business environment. This research paper investigates the application of machine learning algorithms, specifically logistic regression and Naive Bayes, for predicting project risk. By leveraging historical project data and a set of selected features, we aim to develop accurate and automated models that can identify and quantify project risks.**

Keywords—Project risk, Logistic Regression, Gaussian Naïve Bayes

I. INTRODUCTON

This research aims to explore the application of two popular machine learning algorithms, logistic regression and Naive Bayes, in predicting project risk. Logistic regression is a widely used classification algorithm that estimates the probability of a binary outcome based on a set of predictor variables. On the other hand, Naive Bayes is a probabilistic algorithm that calculates the likelihood of an event occurring given the presence of certain independent features. By employing these algorithms, we aim to develop predictive models that can identify and quantify the risk associated with a project based on its characteristics and historical data.

The primary objective of this study is to assess the effectiveness of these machine learning techniques in predicting project risk compared to traditional manual methods. By analyzing historical project data, we aim to evaluate the developed models' accuracy, precision, and F1 score. Additionally, we will investigate the impact of different feature selection and preprocessing techniques on the predictive performance of the models.

II. RELATED WORKS.

1. *"Prediction of risk factors of software development project by using multiple logistic regression*"; Thitima Christiansen, Pongpisit Wuttidittachotti, Somchai PrakanCharoen and Sakda Ari-Ong Vallipakorn. In this paper, they analyzed multiple logistic regressions from 70 related projects and registered the project's risk profile. After using the model to predict, they obtained 34 projects equivalent to 91.9% risk and 29 projects equivalent to 87.9% risk-free projects. Then, the overall prediction accuracy was 90%. [7]
2. "*Incremental Estimation of Project Failure Risk with Naive Bayes Classifier"* by Toshiki Mori; Shurei Tamura; Shingo Kakui . In this paper , they conducted experiments with data of 104 projects from an organization in which the prediction results obtained using Naïve Bayes classifier were compared with those obtained using the Poisson regression model in each development phase: low-level design (LD), coding (CD), and unit testing (UT) .Those obtained using Naïve Bayes classifier were 0.702, 0.748, and 0.764, respectively, in LD, CD, and UT.[8]
3. "*Applied artificial intelligence for predicting construction projects delay "*; Christian Nnaemeka Egwim, Hafiz Alaka, Luqman Olalekan Toriola-Coker, Habeed Balogun, Funlade Sunmola. In this paper they use many different models including

Gaussian Naive Bayes. After using many models to evaluate and compare different models, they obtained relatively high accuracy scores. in which the Gaussian Naive Bayes model has the highest accuracy score of 75%. [9]

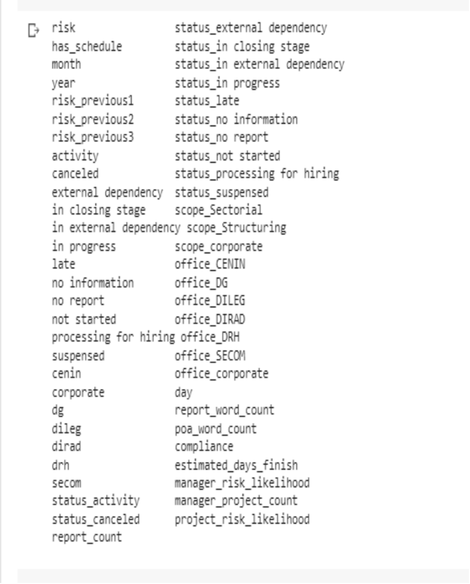
4. *"CREDIT RISK ESTIMATION WITH MACHINE LEARNING AND ARTIFICIAL NEURAL NETWORKS ALGORITHMS"* , : İlker Yıldız. In this paper , he/she use "German Credit" data on the Kaggle platform , which conclude customers information and credit status are found as "good" and "bad". By using these data, it is aimed to evaluate new credit application requests. The data set used was passed through various pre-data processing steps and models such as Logistic Regression, Artificial Neural Networks, K-NN, Support Vector, Gaussian Naïve Bayes, ..etc were trained . In this one , the accuracy achieved using the Gaussian Naïve Bayes model is 64,4% [10]

III. Data Collection

The dataset we have chosen for our risk management analysis is obtained from a reputable source on GitHub, which is one of the large communities and platforms for datasets, specifically from the following repository "Predicting Project Risk" provided by Marcio Fonseca. Our initial raw data consisted of 19 columns of data where,

- project: project identifier
- risk: label: 1 - high risk, 0 - low risk
- status: project status: {'tramitando para contratação', 'em andamento', 'não iniciado', 'sem relatório', 'atrasado', 'dependência externa', 'suspenso', 'em dependência externa', 'cancelado', 'em fase de encerramento', 'atividade', 'sem informação'}
- compliance: index for compliance with project management process
- report_count: number of reports available for the project
- has_schedule: 1 - project has schedule, 0 - otherwise
- scope: project scope: {"Corporativo", "Setorial", "Estruturante"}
- office: project sponsor office: {'Corporativo', 'CENIN', 'SECOM', 'DILEG', 'DG', 'DIRAD', 'DRH'}
- month: month of project report publication
- year: year of project report publication
- day: day of project report publication
- risk_previous1: project appeared in the risk report in the last month
- risk_previous2: project appeared in the risk report in the last two months
- risk_previous3: project appeared in the risk report in the last three months
- project_risk_likelihood: maximum likelihood risk probability estimation (with Laplace smoothing)
- report_word_count: number of words in report
- poa_word_count: number of words in "points of attention" section
- estimated_days_finish: estimated days to finish project
- manager_risk_likelihood: maximum likelihood risk probability estimation for managers (with Laplace smoothing)
- manager_project_count: number of projects the manager is responsible for in a given month

Then we will use Python to dataset before processing. After that, we preprocessed the data first to convert everything to English and encode those columns with many values into new columns. Our dataset then has 59 columns and 962 rows.



```
risk                    status_external dependency
has_schedule            status_in closing stage
month                   status_in external dependency
year                    status_in progress
risk_previous1          status_late
risk_previous2          status_no information
risk_previous3          status_no report
activity                status_not started
canceled                status_processing for hiring
external dependency     status_suspensed
in closing stage        scope_Sectorial
in external dependency scope_Structuring
in progress             scope_corporate
late                    office_CENIN
no information          office_DG
no report               office_DILEG
not started             office_DIRAD
processing for hiring office_DRH
suspensed               office_SECOM
cenin                   office_corporate
corporate               day
dg                      report_word_count
dileg                   poa_word_count
dirad                   compliance
drh                     estimated_days_finish
secom                   manager_risk_likelihood
status_activity         manager_project_count
status_canceled         project_risk_likelihood
report_count
```

Then we will use those values to predict project risk, that is shown through the 2 models below: Logistic Regression and Gaussian Naive Bayes.

IV. METHODOLOGIES

Logistic Regression is one of the most important analytic tools in the social and natural sciences. Furthermore, logistic regression is a predictive analysis method to predict a binary outcome. The model delivers a binary or dichotomous outcome limited to two possible outcomes: yes/no, 0/1, or true/false based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.

How Does the **Logistic Regression** *Algorithm Work?*

Assume that the probability of event A will happen is $\hat{p}$ and $1 - \hat{p}$ is alternative probability of A. In the scenario where we have a large amount of data, how can we know if the next event is A or not?

The odds are simply the ratio of the proportions for the two possible outcomes.

$$ODDS = \frac{\hat{p}}{1 - \hat{p}}$$

In that:

+ Odds: ratio between the two components occurs.

+ $\hat{p}$: is the probability that an event will occur

+ $1 - \hat{p}$: is the probability that the event will not happen

We all know the equation of the best fit line in linear regression is:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

In that:

+ y is dependent variable

+ $x_1, .. x_n$ independent variables

+ β_0: coefficient of freedom

+ $\beta_1, ... \beta_n$: partial slopes.

And instead of y we are taking the probability $\hat{p}$. But there is an issue here, the value of $\hat{p}$ will exceed 1 or go below 0 and we know that range of $\hat{p}$ is (0 – 1). To overcome this issue, we take ODDS of $\hat{p}$:

$$y = \beta_0 + \beta_1 x => \frac{\hat{p}}{1-\hat{p}} = \beta_0 + \beta_1 x$$

We know that odds can always be positive which means the range will always be. (0, +∞). The logistic regression solution to this difficulty is to transform the odds $\frac{\hat{p}}{1-\hat{p}}$ (using the natural logarithm). We use the term ***log odds*** or ***logit*** for this transformation.

$$\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = \beta_0 + \beta_1 x => \hat{p} = \frac{1}{1+e^{-(\beta_0 + \beta_1 x)}}$$

We now have our logistic function, also called a sigmoid function.

Sigmoid function is used to map predictions to probabilities. The sigmoid function has an S shaped curve. It is also called sigmoid curve.

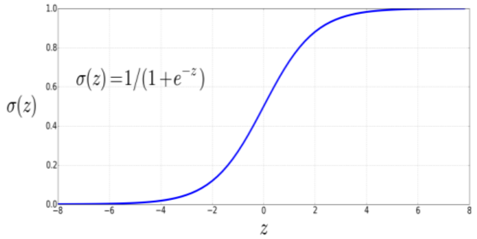


Figure 3 Sigmoid graph has a S-shaped curve.

The sigmoid function from the prior section thus gives us a way to take an instance x and compute the probability P (y = 1|x). How do we make a decision about which class to apply to a test instance x? For a given x, we say yes if the probability P (y = 1|x) is more than .5, and no otherwise. We call .5 the decision boundary:

$$decision(x) = \begin{cases} 1 \; if \; P(y = 1|x) > 0.5 \\ 0 \, , otherwise \end{cases}$$

Naive Bayes classifier - Naive Bayes classifier is a simple and easy classifier. In Naive Bayes

Classifier, the model applies the Bayes Theorem with the probabilistic approach to provide the outcome of the prediction [1] [2] [3] [4] [5]. Naive Bayes takes each dataset parameter as an independent variable [1]. The project risk is calculated through the Bayes principle, where the risk parameter is the target feature y, and the rest of the parameter is the feature X. Naive Bayes also assigns a posterior class probability to an instance [5]. Additionally, in the equation, the target feature is probability Y and the feature is probability X. Therefore, we obtain an equation:

$$P(y_j \mid \mathbf{x}_i) = \frac{P(\mathbf{x}_i \mid y_j) P(y_j)}{P(\mathbf{x}_i)}$$

Figure 4 Naïve Bayes classifier equation

Gaussian Naive Bayes - Gaussian Naive Bayes is one of the variants of the Naive Bayes classifier. The model inherits from the former properties, with an additional combination of Gaussian distribution to deal with continuous data in the dataset. GNB computes conditional class probabilities of a feature X and then predicts the most probable class of a vector of training data, where the training set is separated by the mean and standard deviation. [1] [6] The equation of GNB is the same as the probability density as it applies the Gaussian distribution:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Figure 5 Guassian Naïve Bayes classifier equation

V. Results

After conducting numerous experiments and testing various metrics, we decided to choose accuracy and F1 score for our model evaluation. Accuracy and F1 score are popular metrics to evaluate a model. While accuracy prioritizes the model's number of correct predictions made, F1 score takes both precision and recall scores into one single metric. Additionally, to have a clearer picture of the classification problem, we graphed a confusion matrix to address four values: True Positive, True Negative, False Positive and False Negative.

This is the evaluation table we got after predicting risk from 9 features mentioned in data collection:

Model	Train – Test	F1-Score		Accuracy
		0	1	
Logistic Regression	7 - 3	0.54	0.66	0.609
	8 - 2	0.55	0.67	0.618
	9 - 1	0.52	0.63	0.581
Guassian Naïve Bayes	7 - 3	0.81	0.39	0.714
	8 - 2	0.81	0.38	0.714
	9 - 1	0.84	0.39	0.743

Table 1 Evaluation of 2 models

The prediction results show that the accuracy of the two algorithms have a high accuracy point around 60 and 70 percent. With train-test ratios of 7-3, 8-2 and 9-1, respectively, Logistic Regression gives an exact ratio of 0.609, 0.618 and 0.581, respectively. In addition, Gaussian Naïve Bayes gives a better accuracy rate, where score is nearly the same, with the 9-1 ratio is a little bit higher than 7-3 and 8-2 ratio.

Analyzing more deeply into F1-Score, we can see that the F1-Score of the two algorithms is at a rather large difference, the difference is Logistic Regression only has F1 score at class 0 smaller than class 1, indicating that the model has a little trouble in predicting class 0. On the other hands, there is a huge difference between two classes at Gaussian Naïve Bayes, where class 0 is very high, and class 1 is very low. This is indicating that Gaussian Naïve Bayes had more difficulty in predicting class 1 than class 0.

For Feature importances, since Gaussian Naïve Bayes is unable to use features of importance, we only got the Logistic Regression:

First, the Logistic Regression (figure 6, figure 9, and figure 12) represents the important features of the three train sets – giving almost the same result.

When observing three figures, it is intuitively clear that year is the most important factor. For the least important factor, there is a distinct difference between the three ratios. In 7-3 (figure 6), the least important feature is has_schedule, while both month and report_word_count are the least important in 8-2 (figure 9) and lastly, both status_late and month have the lowest cofficient in 9-1 (figure 12).

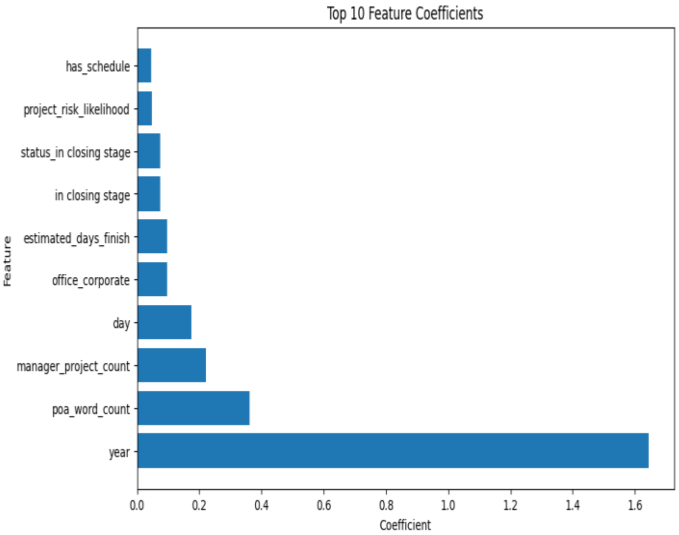


Figure 6 Features importance of Logistic Regression when spliting train- test into 7 – 3

For Confusion Matrix, we can come into conclusion that there is also a significant difference between in both between the model train-test ratio and the model. Here is the Logistic Regression confusion matrices:

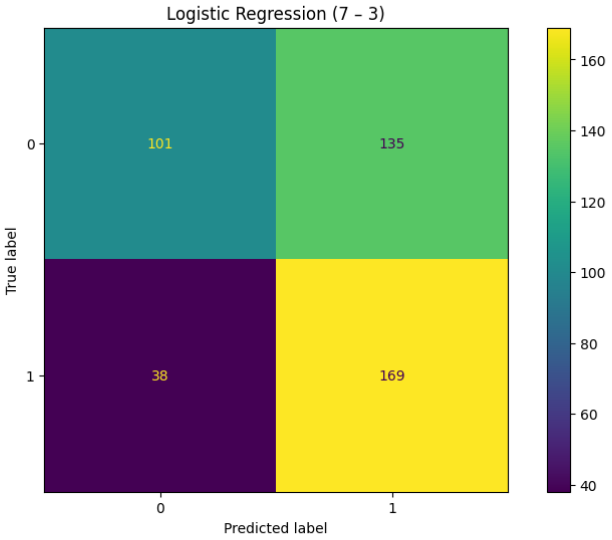
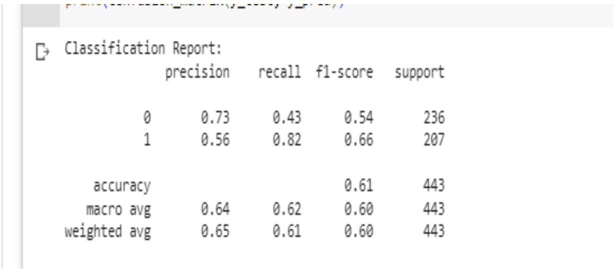


Figure 7 Confusion matrix display of Logistic Regression when splitting train-test into 7-3



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

           0       0.73      0.43      0.54       236
           1       0.56      0.82      0.66       207

    accuracy                           0.61       443
   macro avg       0.64      0.62      0.60       443
weighted avg       0.65      0.61      0.60       443
```

Figure 8 Classification report with train – test 7-3 respectively of Logistic Regression algorithm

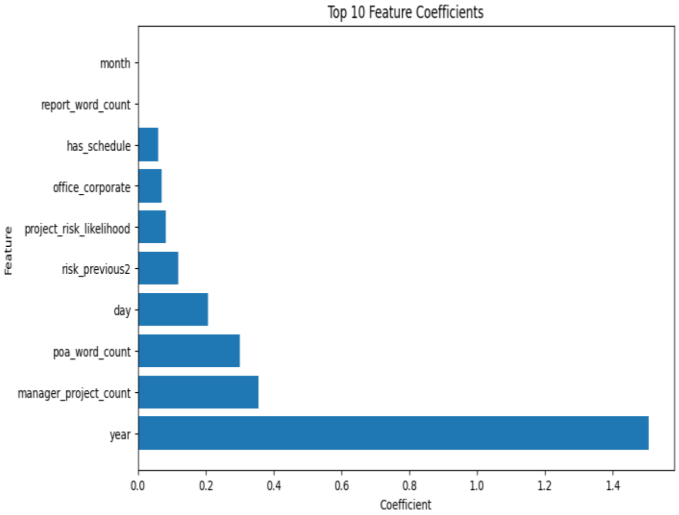


Figure 9 Features importance of Logistic Regression when splitting train-test into 8 – 2.

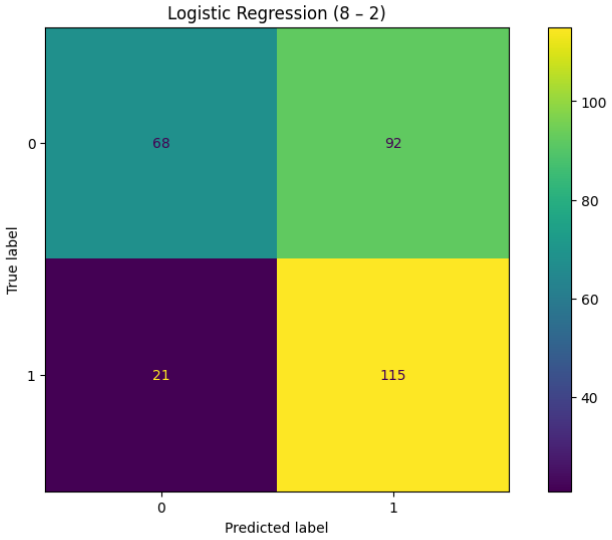
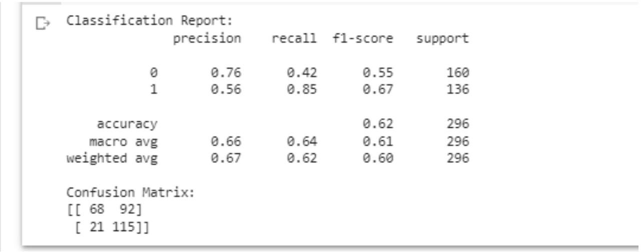


Figure 10 Confusion matrix display of Logistic Regression when splitting train-test into 8-2



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

           0       0.76      0.42      0.55       160
           1       0.56      0.85      0.67       136

    accuracy                           0.62       296
   macro avg       0.66      0.64      0.61       296
weighted avg       0.67      0.62      0.60       296

Confusion Matrix:
[[ 68  92]
 [ 21 115]]
```

Figure 11 Result with train – test 8-2 respectively of Logistic Regression

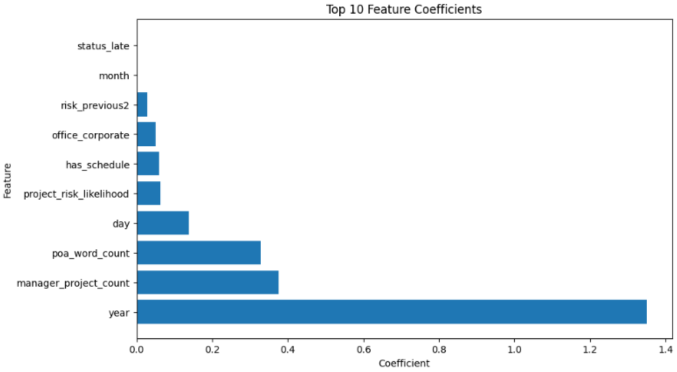


Figure 12 Features importance of Logistic Regression when splitting train- test into 9-1.

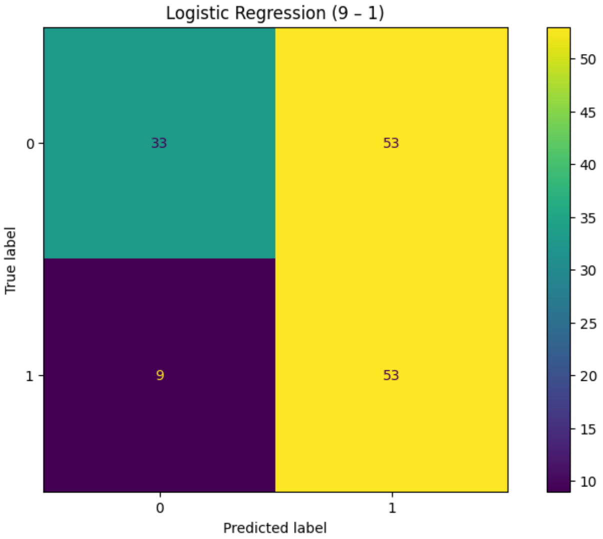
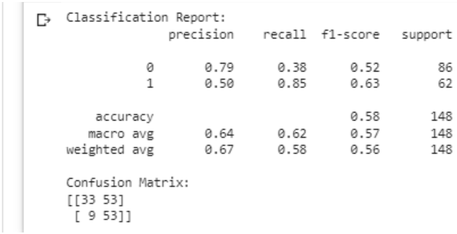


Figure 13 Confusion matrix display of Logistic Regression when splitting train-test into 9-1



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

           0       0.79      0.38      0.52        86
           1       0.50      0.85      0.63        62

    accuracy                           0.58       148
   macro avg       0.64      0.62      0.57       148
weighted avg       0.67      0.58      0.56       148

Confusion Matrix:
[[33 53]
 [ 9 53]]
```

Figure 14 Result with train – test 9-1 respectively of Logistic Regression

In ratio 7-3, there is a significant amount of correct prediction, with low False Positive and only the value of False Negative is noticeable since the number is very high. The rest of the ratio has the same result type as the ratio 7-3. This is indicating that the model had some difficulties on trying to get a correct prediction on the "false" side, meaning that there is a lot of projects supposed to be high risk, but the model predicted them to be low risk.

Based on the feature important results from the Logistic Regression model when dividing the ship test into 7 - 3, 8 - 2, and 9 - 1. Factor 'Year' has the highest importance with a value of 1.65 with a ratio of 7-3. Besides, two factors 'poa_word_count' and 'manager_project_count' have a big influence following 'Year' which the factor 'poa_word_count' has the best coefficient at a train-test ratio of 7-3 is 0.375 and 'manager_project_count' has the best coefficient at a train-test ratio of 9-1 is 0.385. In general, the train-test ratio 9-1 has the best results.

As for Gaussian Naive Bayes, we have a better result:

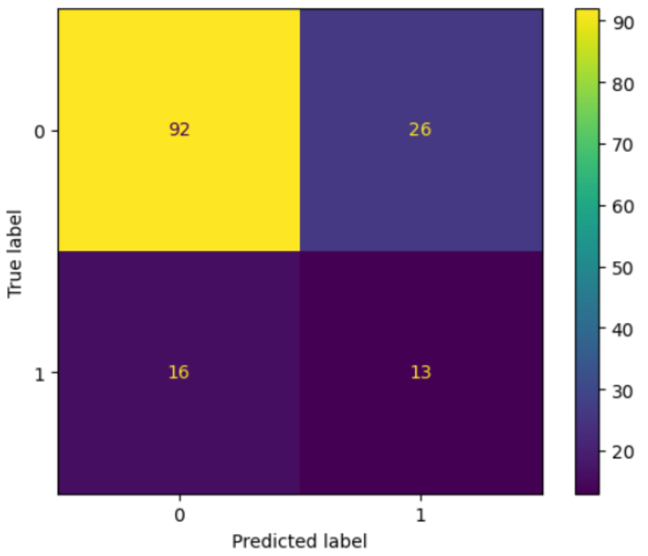
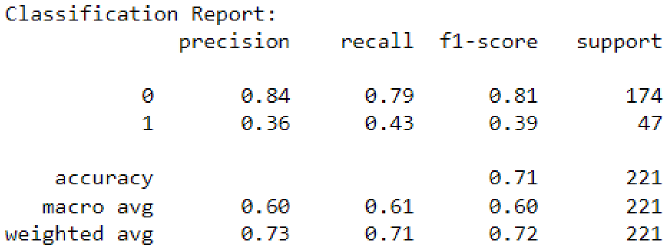


Figure 15 Confusion matrix display of Guassian Naïve Bayes when splitting train-test into 7-3



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

           0       0.84      0.79      0.81       174
           1       0.36      0.43      0.39        47

    accuracy                           0.71       221
   macro avg       0.60      0.61      0.60       221
weighted avg       0.73      0.71      0.72       221
```

Figure 16 Classification report with train – test 7-3 respectively of Guassian Naïve Bayes

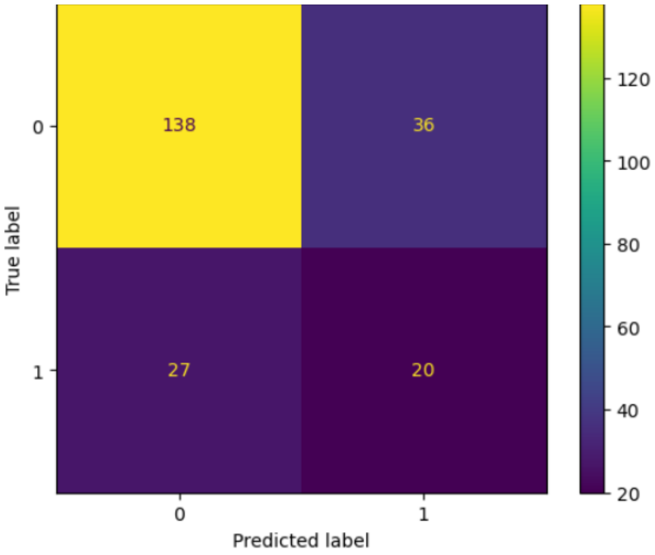
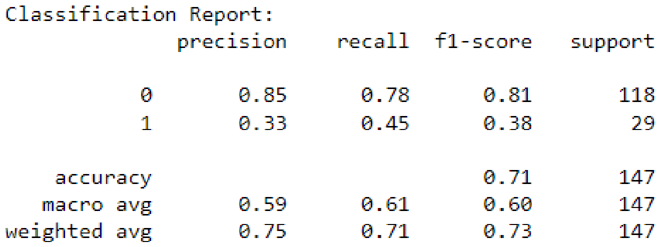


Figure 17 Confusion matrix display of Gassian Naïve Bayes when splitting train-test into 8-2



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

           0       0.85      0.78      0.81       118
           1       0.33      0.45      0.38        29

    accuracy                           0.71       147
   macro avg       0.59      0.61      0.60       147
weighted avg       0.75      0.71      0.73       147
```

Figure 18 Classification report with train – test 8-2 respectively of Guassian Naïve Bayes

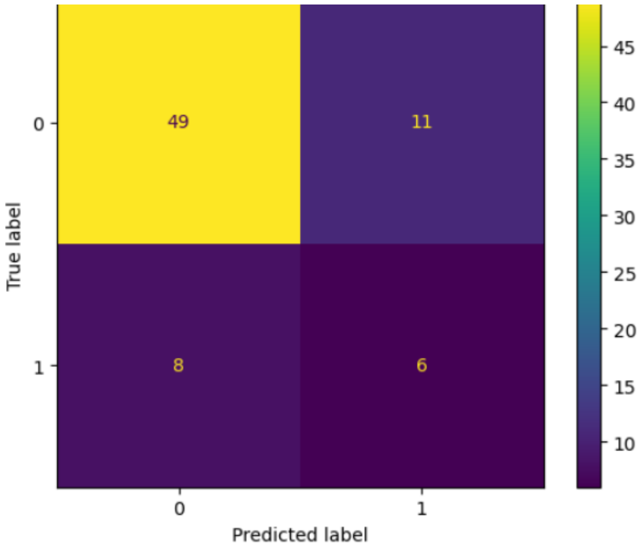
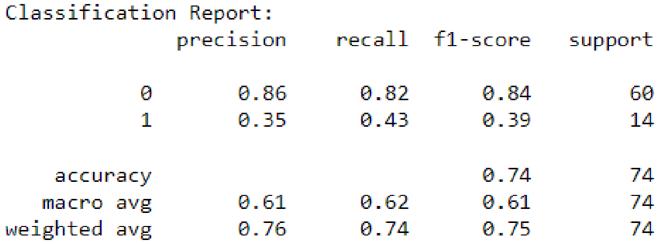


Figure 19 Confusion matrix display of Guassian Naïve Bayes when splitting train-test into 9-1



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

           0       0.86      0.82      0.84        60
           1       0.35      0.43      0.39        14

    accuracy                           0.74        74
   macro avg       0.61      0.62      0.61        74
weighted avg       0.76      0.74      0.75        74
```

Figure 20 Classification report with train – test 9-1 respectively of Guassian Naïve Bayes

All the ratio got the most True Positive value, with a very small value at True Negative, False Positive and False Negative. This indicates that the model performed well, especially when it had a lot of correct predictions on projects that had low risk and low.

We do not apply the Important Feature to the Gaussian Naive Bayes because it does not directly calculate the importance of each feature for the classification decision.

VI. Conclusion

In this research, we explored the application of two popular machine learning algorithms, Logistic Regression and Naive Bayes in predicting project risk. Our study indicates that the Logistic Regression algorithm exhibited comparatively lower accuracy, whereas the Naive Bayes algorithm consistently demonstrated higher accuracy. This can be suggested that Naïve Bayes algorithm holds significant potential in providing higher accuracy. By integrating machine learning algorithm like Navie Bayes, we can effectively predict and mitigate risks in project management processes. Through these endeavors, we could aim to enhance risk management practices and optimize project outcomes.

VII. References

[1] Hajer Kamel, Dhahir Abdulah and Jamal M.Al-Tuwaijari, "Cancer Classification Using Gaussian Naive Bayes Algorithm", IEC2019, 2019 Rish, "An empirical study of the naive Bayes classifier", IJCAI, 2001

[2] Marlis Ontivero-Ortega, Agustin Lage-Castellanos, Giancarlo Valente, Rainer Goebel and Mitchell Valdes-Sosa, "Fast Gaussian Naïve Bayes for searchlight classification analysis", NeuroImage 2017

[3] Adnan Ahmed Rafique, Ahmad Jalal and Abrar Ahmed, "Scene Understanding and Recognition: Statistical Segmented Model using Geometrical Features and Gaussian Naïve Bayes", ICAEM, 2019

[4] Daniel Berrar, Bayes' Theorem and Naive Bayes Classifier, Reference Module in Life Sciences, 2018

[5] Ronei Marcos de Moraes and Liliane , Santos Machado, "Gaussian Naive Bayes for Online Training Assessment in Virtual Reality-Based Simulators", Mathware & Soft Computing 16, 2009

[6] Prediction of risk factors of a software development project by using multiple logistic regression; Thitima Christiansen, Pongpisit, Somchai PrakanCharoen, and Sakda Ari-Ong Vallipakorn

[7] Toshiki Mori, Shurei Tamura and Shingo Kakui, "Incremental Estimation of Project Failure Risk with Naive Bayes Classifier", ACM / IEEE, 2013

[8] Christian Nnaemeka Egwim, Hafiz Alaka, Luqman Olalekan Toriola-Coker, Habeed Balogun and Funlade Sunmola, "Applied artificial intelligence for predicting construction projects delay", Vol 6, December 2021

[9] İlker Yıldız, "Customer Credit Risk Assessment using Artificial Neural Networks", MECS, 2016