Implementation of Naive Bayes (June 2024)

Kristina Ghimire (THA077BCT023)[1], Punam Shrestha (THA077BCT038)[1]

[1]Department of Electronics and Computer Engineering, IOE, Thapathali Campus, Kathmandu 44600, Nepal
Corresponding author: Kristina Ghimire(ghimirekristina10@gmail.com)

ABSTRACT This study explores the application of Naive Bayes classifiers on the Titanic and Income datasets, focusing on predictive modeling of survival rates and income levels, respectively. Through data preprocessing, feature engineering, and model selection, Gaussian Naive Bayes and Categorical Naive Bayes were employed to effectively handle continuous and categorical data types. Results demonstrate the classifiers' robust performance in accurately predicting survival outcomes based on demographic and travel-related factors in the Titanic dataset, and income levels influenced by demographic and work-related attributes in the Income dataset. These findings underscore Naive Bayes' versatility and effectiveness in extracting meaningful insights from diverse datasets for predictive analytics.

INDEX TERMS Bayes Theorem, Categorical Naive Bayes, Gaussian Naive Bayes, Likelihood.

I INTRODUCTION

The Naive Bayes Classifier is a powerful and efficient machine learning model rooted in Bayes' Theorem. It is especially effective for classification tasks that involve high-dimensional datasets, where the number of features is large. One of its key advantages is the assumption of independence among features, which simplifies computations and makes the model scalable. This assumption significantly reduces computational complexity, allowing the Naive Bayes Classifier to process large datasets quickly and efficiently.

Despite its simplicity, the Naive Bayes Classifier often delivers high accuracy, particularly when the independence assumption approximately holds. This classifier is straightforward to implement and interpret, making it a popular choice among practitioners. It also has the advantage of handling irrelevant features well; because of the independence assumption, the effects of such features tend to cancel out.

The Naive Bayes Classifier excels at handling high-dimensional data. Considering each feature's contribution independently, simplifies the computational process, even when the dataset has many features. Another strength is its ability to handle missing data by ignoring the missing values during probability calculations, thus avoiding the need for complex imputation methods.

One of the unique aspects of the Naive Bayes Classifier is its probabilistic output. It provides not just a classification but also the probabilities of each class. This probabilistic interpretation is valuable in many applications, especially in decision-making processes where confidence in predictions is essential.

However, the assumption that all features are independent given the class label is rarely true in real-world scenarios, which can sometimes affect the classifier's performance. Additionally, if a particular feature-class combination is not observed in the training data, it can lead to zero probability. This issue can be mitigated using techniques like Laplace smoothing.

Overall, the Naive Bayes Classifier is a valuable tool in the machine learning toolbox due to its simplicity, efficiency, and effectiveness in various applications. Its ability to handle high-dimensional data, provide probabilistic interpretations, and maintain performance even with the independence assumption makes it a go-to choice for many classification tasks.

II RELATED WORK

Naive Bayes is widely studied and used in classification and regression problems.

Navoneel Chakrabarty et al. [1] employ machine learning to tackle economic inequality by predicting income levels using the UCI Adult Dataset. They preprocess data, including handling missing values and encoding categorical features, and split it into training and testing sets. Utilizing the Extra Trees Classifier for feature selection and Gradient Boosting Classifier (GBC) for modeling, they achieve high training (88.73%) and validation (88.16%) accuracies. Naive Bayes wasn't the primary model used; instead, the GBC outperformed due to extensive hyper-parameter tuning. Their approach advances predictive accuracy and suggests future exploration of hybrid or advanced preprocessing methods.

Peling et al. [2] explore the application of the Naive Bayes algorithm in predicting students' study periods. The study utilizes data mining techniques to analyze factors influencing study duration, aiming to enhance educational planning and student support systems. By leveraging Naive Bayes, the authors demonstrate a methodical approach to understanding and predicting student behavior patterns. This research contributes valuable insights into educational data analytics, offering potential improvements in student success strategies and resource allocation within academic institutions.

Harahap et al. [3] explores an effective application of machine learning in consumer behavior analysis. The study demonstrates the method's capability to predict purchase decisions based on historical data, leveraging the simplicity and efficiency of Naïve Bayes. By focusing on practical implementation and results, the authors provide insights into how this technique can enhance marketing strategies and customer engagement strategies. Overall, the paper contributes valuable findings for both academia and industry seeking to utilize machine learning for predictive analytics in consumer-oriented domains.

Hemanth et al. [4] compare the performance of Decision Tree and Naive Bayes algorithms in predicting income classes. While Decision Trees show superior accuracy according to their findings, Naive Bayes remains a pivotal benchmark in machine learning, known for its simplicity and efficiency in handling large datasets with categorical features. The study underscores Decision Tree's strengths in complex classification tasks, yet acknowledges Naive Bayes' enduring utility in scenarios demanding rapid model training and interpretation.

These works contribute to the understanding and use of Naive Bayes to make more accurate predictions based on given features.

III METHODOLOGY

A DATASET DESCRIPTION

The performance evaluation of target classification using the Naive Bayes involved conducting experiments on the two datasets: Titanic and Income.

The Titanic dataset is widely used in machine learning and analysis. This dataset contains 11 features like age, sex, ticket class, fare, cabin, name, passenger ID, Pclass, Sibling or spouse, parent or children, ticket, fare, and cabin. The dataset aims to predict the survival rate based on passenger features. Data preprocessing techniques like imputing missing values and feature engineering are commonly applied. This dataset is used for implementing the model that can predict categorical targets like the Naive Bayes Classifier.

The Census Income dataset is commonly used for predictive tasks and classification to predict whether an individual earns more than $50K or not based on the individual's features. The dataset consists of 48842 instances with 14 attributes that impact the prediction process. The dataset includes features such as age, education, occupation, marital status, relationship, race, sex, capital gain, capital loss, hours worked per week, and native country. Data preprocessing techniques like imputing missing values, encoding categorical variables, and feature engineering are commonly applied.

The use of the Naive Bayes on these datasets was done to evaluate the prediction of the Naive Bayes and different ways to improve the accuracy.

B THEORETICAL FORMULATION

Naive Bayes is an easy-to-implement and understand algorithm based on Bayes' theorem. It requires little training data and is efficient in both the training and prediction phases. It handles large datasets well, scaling linearly with the number of features and data points, making it suitable for real-time applications.

Despite its simplicity, Naive Bayes works well with high-dimensional data, especially in text classification where features (words) are numerous. It effectively manages irrelevant features due to the assumption of conditional independence, which reduces the impact of non-informative features.

Compared to complex models like SVMs, neural networks, or ensemble methods, Naive Bayes is less prone to overfitting and performs well even with limited training data. Its computational efficiency is a significant advantage for real-time applications. Moreover, it provides interpretable probabilities for class membership, unlike more complex models.

Naive Bayes is a probabilistic classifier based on Bayes' theorem, with a naive assumption of feature independence. This allows it to update probability estimates as new information becomes available, making it effective for large and high-dimensional datasets. The formula used is shown in figure 1.

The term ”Naive Bayes” combines two key ideas. First, it's based on Bayes' Theorem, which calculates the probability of a hypothesis given prior knowledge, named after Thomas Bayes. Second, it makes a ”naive” or simple assumption for calculation that features are conditionally independent given the class label. This means each feature's presence is assumed to be unrelated to any other feature, given the class. While this assumption isn't usually true in real-world scenarios, where features often interact, Naive Bayes often performs well, especially with high-dimensional data and small datasets. Its simplicity and computational efficiency come from this naive assumption, making it easy to implement and scalable for large datasets.

1 Working of Naive Bayes Classifiers

The Naive Bayes classifier works through two main phases: training and testing.

Training Phase

During the training phase, the Naive Bayes classifier learns the statistical properties of the features in the training dataset relative to the class labels. It applies Bayes' theorem under the ”naive” assumption that features are conditionally independent given the class label. This phase involves estimating two main probabilities:

1. **Prior probabilities**
   These are the probabilities of each class occurring in the dataset.

2. **Conditional probabilities or Likelihood**
   These are the probabilities of each feature given in each class.

For example, in Gaussian Naive Bayes, these conditional probabilities are based on the mean and variance of each feature for each class. This training process allows the classifier to build a probabilistic model that captures the relationships between features and classes.

Testing Phase

Once trained, the Naive Bayes classifier is evaluated in the testing phase using unseen data. This phase is crucial as it assesses how well the model generalizes to new data. The classifier applies the learned probabilities to calculate the posterior probability of each class for new instances. It then predicts the class with the highest probability as the output. Key performance metrics such as accuracy, precision, recall, and F1 score are computed to measure how effectively the classifier predicts unseen data. The testing phase helps validate the model's assumptions and decisions made during training, ensuring it performs reliably in real-world applications.

These two phases together form the core operational cycle of the Naive Bayes classifier, enabling it to learn from data, make predictions, and assess its performance accurately.

2 Variants of Naive Bayes Classifiers

There are several variants of Naive Bayes classifiers suited to different types of data.

GaussianNB

Gaussian Naive Bayes (GaussianNB) is used for continuous features that follow a normal distribution. It computes probabilities using the Gaussian probability density function, making it effective for handling continuous data without discretization. GaussianNB is efficient and quick to train, requiring only the estimation of the mean and variance for each class.

CategoricalNB

Categorical Naive Bayes (CategoricalNB) is designed for categorical data, where features take on discrete values. This variant is ideal for domains like marketing and social sciences, where categorical data is common. CategoricalNB can directly handle categorical features, making it useful for tasks like document classification, customer segmentation, and sentiment analysis. Its simplicity, efficiency, and clear probabilistic interpretation make CategoricalNB a valuable tool in various applications.

C MATHEMATICAL FORMULAE

1 Bayes Theorem

Naive Bayes is based on Bayes Theorem which is a fundamental concept in probability theory and statistics. It helps us move from our initial belief (prior probability) to a more informed belief (posterior probability) by considering new evidence and how likely that evidence is under different scenarios. The formula of Bayes Theorem is shown in equation 1.

2 Prior Probability

Prior probability is the initial probability of a hypothesis before any new evidence is considered. It reflects our knowledge or belief about the hypothesis based on previous information or assumptions. In the context of Bayes' Theorem, the prior probability is updated with new evidence to calculate the posterior probability, providing a more informed estimate of the hypothesis's likelihood. The general formula of prior probability is shown in equation 2.

3 Likelihood

Likelihood measures the probability of observing evidence given that a particular hypothesis is true. In Bayes' Theorem, the likelihood is used to update prior probability (or initial belief), helping to determine the posterior probability (or new belief) of the hypothesis. The general formula of likelihood for n features is shown in equation 4.

1. **Gaussian Naive Bayes**
   In Gaussian Naive Bayes, the likelihood is used to handle continuous data assuming the continuous data or features follow a Gaussian (normal) distribution. It allows the model to handle continuous data without the need for discretization into categorical data. Here, all the categorical data should be converted to numerical data. The formula is shown in equation 5.

2. **Categorical Naive Bayes**
   In Categorical Naive Bayes, the likelihood is used to handle categorical data. It is useful for datasets with categorical features as there is no need to convert them into numerical or other forms. The formula is shown in equation 6.

4 Precision

Precision measures the accuracy of positive predictions, reflecting the ratio of true positives to total predicted positives. High precision indicates reliable positive predictions with few false positives, essential for applications sensitive to incorrect classifications. The formula of precision is shown in equation 7.

5 Recall

Recall (sensitivity) measures the proportion of true positive predictions relative to all actual positives. Balancing recall with precision is crucial; high recall ensures identifying most positives but may increase false positives, impacting overall model performance. The formula of recall is shown in equation 8.

6 F1 Score

The F1 score combines precision and recall into a single metric using their harmonic mean. A higher F1 score indicates a balanced performance between precision and recall, making the model suitable for applications requiring both accurate identification of positives and low false positives. The formula of the f1 score is shown in equation 9.

7 Macro Average

Macro average calculates the mean performance metric (e.g., precision, recall, F1 score) across all classes in multi-class classification, treating each class equally. A higher macro average signifies consistent performance across diverse categories, indicating strong generalization capabilities. The formula of the macro average is shown in equation 10.

8 Weighted Average

Weighted average considers class distribution by calculating a mean weighted by instances in each class. Higher weighted averages indicate robust model performance across all classes, demonstrating the ability to handle imbalanced data and make accurate predictions. The formula of the weighted average is shown in equation 11.

9 Accuracy

Accuracy measures correctly predicted instances relative to the total number of instances in a dataset. High accuracy indicates effective model performance suitable for real-world applications, accurately predicting both positive and negative instances. The formula of accuracy is shown in equation 12.

D SYSTEM BLOCK DIAGRAM

The system block diagram is shown in figure 1. The Naive Bayes classifier operates through several key phases. Initially, the dataset undergoes preprocessing to handle missing values, outliers, and irrelevant features. Categorical data is then encoded into numerical form for compatibility with the algorithm. During training, the classifier learns the statistical relationships between features and class labels, assuming conditional independence of features given the class. The testing phase evaluates the model's performance on unseen data, ensuring it generalizes well. Predictions are made using learned probabilities, guiding practical decisions in various applications. Optimization steps such as hyperparameter tuning and feature selection refine the model's effectiveness and generalization capabilities, ensuring it performs robustly across different datasets and scenarios.

E DATA PREPROCESSING PIPELINES

1 Imputing Missing Values

A dataset often contains missing values, which can hinder the analysis as they do not contribute any meaningful information. To address this issue, missing values should be imputed using various methods such as filling them with the mean, median, or mode (highest frequency) of the available data. Imputing missing values ensures that the dataset remains complete and allows for more accurate and reliable statistical analyses, enhancing the overall quality and utility of the data.

2 Treating Outliers

Outliers are values that fall far outside the typical range of data points in a dataset. They can skew statistical analyses and lead to misleading results. To identify and treat outliers, several methods can be employed. One common approach involves using z-scores, where we calculate how many standard deviations a data point is away from the mean. Points beyond a certain threshold, often set at around ±3 standard deviations, are considered outliers and can be removed or adjusted. Another method involves using boxplots, which visually display the spread of data and help identify values that lie significantly beyond the whiskers, determined by the InterQuartile Range (IQR). Data points outside the calculated bounds of the IQR are typically treated as outliers and either corrected or excluded from further analysis to ensure the dataset remains reliable and representative.

3 Categorization of Numerical Features

Categorizing a numerical target variable involves converting its continuous values into discrete categories or classes, which simplifies predictive tasks by reducing complexity. This transformation is particularly useful in methods such as clustering or decision trees, where working with discrete categories can streamline processing and enhance prediction accuracy. One common technique for encoding nominal categorical values is label encoding. In label encoding, each unique category is assigned a different integer. This allows algorithms to interpret categorical data as numerical data, facilitating their use in various machine learning algorithms. However, it's essential to note that label encoding implies an ordinal relationship between categories, which may not always be appropriate for nominal variables without inherent order.

4 Rectifying class imbalance

In datasets, classes aren't always balanced. Some classes might have many more examples than others, causing models to favor those classes during predictions due to their sheer numbers. To fix this, we can balance the classes by resampling the data. Upsampling involves increasing the number of instances in the minority class, while downsampling decreases instances in the majority class. This ensures that each class contributes equally to the model training, improving its ability to predict all classes accurately, not just the majority.

5 Feature Selection Using Elastic Net

Feature selection and dimensionality reduction are crucial for simplifying models and reducing computational complexity and cost. Various methods exist for feature selection, such as PCA analysis, Lasso regularization, Elastic Net regularization, and more. PCA analysis summarizes multiple features into a smaller set of principal components, which may not directly preserve the original feature values but capture their variance effectively. Regularization techniques like Lasso and Elastic Net help in selecting the most relevant features by penalizing less important ones based on their coefficients. However, it's essential to consider that sometimes all features may be important, necessitating correlation analysis to understand relationships between features and ensure comprehensive model coverage. This approach ensures that models are efficient, interpretable, and perform optimally without unnecessary computational burden.

6 Normalization of data

Normalization involves scaling numerical data into a standardized range. There are several normalization methods. Z-score normalization adjusts data to have a mean of 0 and a standard deviation of 1. On the other hand, min-max scaling transforms data to fit within a fixed range, typically between 0 and 1. These techniques ensure that data from different scales can be compared directly and are essential for many machine learning algorithms to perform effectively and efficiently.

7 Discretization of Continuous data

Using continuous data directly in classification tasks, such as with Naive Bayes, isn't always ideal. Continuous data can vary across a wide range of values, making predictions uncertain and potentially less accurate. To address this, we can discretize continuous data by dividing it into groups or bins. This process, known as discretization, allows us to categorize continuous variables into distinct groups based on specified criteria, such as equal width or equal frequency bins. By doing so, we simplify the data representation and make it more suitable for classification tasks, improving the interpretability and effectiveness of models like decision trees.

IV INSTRUMENTATION DETAILS

Python stands out as a versatile and straightforward programming language renowned for its simplicity and readability. Unlike lower-level languages, Python is interpreted, meaning code is executed line by line, which facilitates quick testing and debugging. In Python, variables are dynamically typed, meaning they don't require explicit declaration for memory allocation; instead, memory is allocated automatically when a value is assigned to a variable. Jupyter Notebook is an interactive computing environment that enables users to create and share documents containing live code, equations, visualizations, and explanatory text. It supports various programming languages, including Python, and is extensively utilized in data science for its flexibility and user-friendly interface. Scikit-learn (or sklearn) is a prominent Python library that provides a range of tools for machine learning and statistical modeling. When using Naive Bayes Classifier from scikit-learn, you can easily implement the classifier with a few lines of code, leveraging its simplicity and effectiveness. Scikit-learn also integrates well with other libraries mentioned, such as Matplotlib for data visualization and Seaborn for statistical graphics, allowing for comprehensive data analysis and visualization workflows in Python.

V EXPERIMENTAL RESULTS

A PROBLEM 1: Titanic Dataset

The titanic dataset is to predict whether the instance survived or not based on given features.

1 Study of Dataset

Evaluating the total dataset, 549 instances did not survive and 342 survived, as visualized in Figure 2. The survival rate for males was less than for females. Based on Pclass features, males from Pclass 3 survived the least. The plot of survival rate based on gender and Pclass is shown in Figure 3. Additionally, the plot of survival rate based on Pclass and Fare in Figure 4 shows the majority of the rate despite fare range. The correlation between the features was visualized using a heatmap as shown in Figure 5. The features Age and Fare have higher range values, so their impact on the target is more significant. The histogram plot for the Fare feature shows a right-skewed distribution in Figure 7. Similarly, the histogram plot for the Age feature shows an almost normal distribution in Figure 6.

2 Removal of Unwanted Features

Features like PassengerId, Name, Ticket, and Cabin have no impact on survival rate analysis and can be discarded.

3 Duplicate Values Removal

Evaluating the dataset, 111 data entries are duplicated. Therefore, we remove the duplicated data from the dataset, reducing it from 891 to 780.

4 Imputing Missing Values

The features Age and Embarked contain some missing values. The Age feature is numerical data, so the missing values are imputed using the mean. The Embarked feature is categorical data, so the missing values are imputed using the mode.

5 Outliers Removal

There are some outliers in the dataset. We visualized a boxplot to check for outliers and remove them. After evaluating the boxplot of each feature, we calculated the InterQuartile Range (IQR) of each feature to determine the lower bound and upper bound of the data. The data less than the lower bound and greater than the upper bound are assumed to be outliers and removed from the original dataset. The plot before removing outliers and after removing outliers is shown in Figures 8 and 9 respectively. After removing outliers, the dataset was reduced from 780 to 647.

6 Resampling

After the removal of outliers, the dataset contains 410 not survived data and 237 survived data. So, we need to resample the data. The plot before resampling classes with the majority class not surviving and after resampling is shown in Figures 10 and 11.

7 Label Encoding for Categorical Data

For Gaussian Naive Bayes, all the features should be numerical data. So, the features Gender and Embarked are label encoded.

8 Combination of Features to Form New Features

The features SibSp and Parch contain siblings or children and parent numbers. So a new feature Family is created by adding both SibSp and Parch and the person.

9 Feature Selection Using Elastic Net

We use the Elastic Net Regularization method to select important features from the dataset. We fit the features and target to get the important features. All the features were selected.

10 Normalization of Dataset

The dataset contains different range values. The Age and Fare contain higher range values which have more impact on the prediction. To standardize all the features to a similar scale, first, we split the dataset into 80% training data and 20% testing data. Then, the training features are fitted into the StandardScaler model and transformed, and the testing features are transformed based on the training features.

11 Training and Testing

Using GaussianNB, we fit the training features and target. On prediction, the model achieved 76% accuracy, as shown in Table 1. The confusion matrix based on the prediction is shown in Figure 12. Around 66 are true positives and 57 are true negatives. The precision for those who did not survive is 0.78 for 82 instances and for those who survived is 0.73 for 82 instances. The recall for those who did not survive is 0.70 for 82 instances and for those who survived is 0.80 for 82 instances. The F1 score for those who did not survive is 0.74 for 82 instances and for those who survived is 0.76 for 82 instances. The total number of instances for testing is 164, which gives 0.75 as the macro and weighted average of precision. Similarly, the macro and weighted average recall is 0.77. The macro and weighted F1 score is 0.73.

We also used CategoricalNB for training features and targets. Before training, all the numerical but continuous data is discretized. On prediction, the model obtained 71% accuracy, as shown in Table 2. The confusion matrix based on the prediction is shown in Figure 13. Around 63 are true positives and 57 are true negatives. The precision for those who did not survive is 0.75 for 82 instances and for those who survived is 0.72 for 82 instances. The recall for those who did not survive is 0.70 for 82 instances and for those who survived is 0.77 for 82 instances. The F1 score for those who did not survive is 0.72 for 82 instances and for those who survived is 0.74 for 82 instances. The total number of instances for testing is 164, which gives 0.73 as the macro and weighted average of precision. Similarly, the macro and weighted average recall is 0.77.

B PROBLEM 2: Census Income Dataset

This study aims to predict whether the person's income exceeds $50k/year or not using the census income dataset.

1 Study of Dataset

Evaluating the total dataset, 7,841 instances have an income of more than 50K, and 24,750 have an income of less than or equal to 50K, as visualized in Figure 14. Visualizing the plot for income more than 50K based on Gender and Marital Status in Figure 17 shows that males have a higher income than females despite their marital status, except for married-civ-spouse where females are more. Among all males, married-AF-spouse has the highest number.

2 Duplicate Values Removal

Evaluating the dataset, 11 instances are duplicated among 32,561 total instances. The removal of these duplicate values is unnecessary.

3 Imputing Missing Values

The missing value in this dataset is in the form of ”?” instead of ”NaN” and is present in the features workclass, occupation, and native.country. All these features are categorical data. So, we impute the missing values using the highest frequency value or mode.

4 Outliers Removal

There are some outliers in the dataset. We visualized a boxplot to check for outliers and remove them. After evaluating the boxplot of each feature, we calculated the InterQuartile Range (IQR) of each feature to determine the data's lower and upper bounds. The data less than the lower bound and greater than the upper bound are assumed to be outliers and are removed from the original dataset. The plot before removing outliers and after removing outliers is shown in Figures 15 and 16 respectively. After removing outliers, the dataset was reduced from 32,561 to 19,004. This outlier range is huge, so we avoided the outlier removal.

5 Label Encoding for Categorical Data

For Gaussian Naive Bayes, all the features should be numerical data. So, the target income is renamed as income_¿50K, and all its values are labeled as 0 for income less than or equal to 50K and 1 for income more than 50K. Other features like workclass, education, marital.status, occupation, relationship, race, sex, and native.country are categorical data. Therefore, all these features are labeled as numerical data.

6 Resampling

The dataset contains 7,841 instances with an income of more than 50K and 24,720 instances with an income less than or equal to 50K. Therefore, we need to resample the data. The plot before resampling classes with the majority class and after resampling is shown in Figures 18 and 19.

7 Feature Selection Using Elastic Net

We use the Elastic Net Regularization method to select important features from the dataset. We convert the target classes to numbers using Label Encoding. Then, we fit the features and target to get the important features. The selected features are 'age', 'fnlwgt', and 'hours.per.week'. The confusion matrix based on feature selection is displayed in Figure 20. We also performed a decision tree without removing any features, which shows results in Figure 21.

8 Training and Testing

Using GaussianNB, we fit the training features and target by removing outliers. On prediction, the model achieved 80% accuracy, as shown in Table 3. The confusion matrix based on the prediction is shown in Figure **??**. Around 526 are true positives and 2,511 are true negatives. The precision for income less than or equal to 50K is 0.91 for 3,019 instances, and for income more than 50K, it is 0.51 for 782 instances. The recall for income less than or equal to 50K is 0.83 for 3,019 instances, and for income more than 50K, it is 0.67 for 782 instances. The F1 score for income less than or equal to 50K is 0.87 for 3,019 instances, and for income more than 50K, it is 0.58 for 782 instances. The total number of instances for testing is 3,801, which gives 0.71 as the macro and 0.83 as the weighted average of precision. Similarly, the macro average recall is 0.75 and the weighted average recall is 0.80. The macro average F1 score is 0.72 and the weighted F1 score is 0.81.

Using GaussianNB, we fit the training features and target. On prediction, the model achieved 100% accuracy, as shown in Table 5. The confusion matrix based on the prediction is shown in Figure 23. Around 4,981 are true positives and 4,907 are true negatives. The precision, recall, and F1 score for income less than or equal to 50K is 1.0 for 4,981 instances, and for income more than 50K, it is 1.0 for 4,907 instances. The total number of instances for testing is 9,888.

We also used CategoricalNB for training features and targets by removing outliers. Before training, all the numerical but continuous data is discretized. On prediction, the model obtained 82% accuracy, as shown in Table **??**. The confusion matrix based on the prediction is shown in Figure 22. Around 550 are true positives and 2,573 are true negatives. The precision for income less than or equal to 50K is 0.92 for 3,019 instances, and for income more than 50K, it is 0.55 for 782 instances. The recall for income less than or equal to 50K is 0.85 for 3,019 instances, and for income more than 50K, it is 0.70 for 782 instances. The F1 score for income less than or equal to 50K is 0.88 for 3,019 instances, and for income more than 50K, it is 0.62 for 782 instances. The total number of instances for testing is 3,801, which gives 0.73 as the macro and 0.84 as the weighted average of precision. Similarly, the macro average recall is 0.78 and the weighted average recall is 0.82. The macro average F1 score is 0.75 and the weighted F1 score is 0.83.

VI DISCUSSION AND ANALYSIS

The Titanic dataset analysis provides valuable insights into the factors influencing survival rates among passengers. Initially, the dataset underwent several preprocessing steps to ensure data integrity and readiness for modeling. Duplicate entries were identified and removed, reducing the dataset to a more manageable size. Missing values in the 'Age' and 'Embarked' features were imputed using statistical measures like the mean and mode, respectively. This ensured that all data points were accounted for without compromising the dataset's overall quality.

Exploratory data analysis revealed significant trends in survival rates based on gender and Pclass. Females generally had a higher survival rate compared to males. The influence of passenger class (Pclass) on survival rates was also pronounced, with passengers in the lower classes (specifically Pclass 3) experiencing lower survival rates. This was evident from visualizations such as bar plots and correlation heatmaps, which highlighted these correlations.

Feature engineering played a crucial role in enhancing the dataset's predictive power. The creation of the 'Family' feature, derived from combining 'SibSp' (siblings/spouses) and 'Parch' (parents/children) and the passenger itself, aimed to capture the familial context of passengers aboard. This new feature provided additional insights into survival outcomes beyond individual characteristics. Using Elastic Net regularization, the dataset underwent feature selection to identify the most influential predictors of survival. This method balances the strengths of both Ridge and Lasso regressions, ensuring a robust selection of features while handling multicollinearity effectively. In this case, all features were retained after Elastic Net regularization, suggesting each feature contributed uniquely to predicting survival outcomes.

Normalization of the dataset was crucial to standardize the range of values across different features. This step involved scaling numerical features like 'Age' and 'Fare' to a comparable range, ensuring that no single feature disproportionately influenced the model's predictions due to its scale.

During model training and evaluation, Gaussian Naive Bayes (NB) and Categorical Naive Bayes were employed. These algorithms, suited for their respective types of data (continuous and discrete), provided insights into the dataset's predictability. Gaussian NB achieved a commendable 76% accuracy, indicating its ability to classify survival outcomes based on continuous features effectively. Similarly, Categorical NB, after discretizing continuous data, achieved 71% accuracy, highlighting its capability to handle categorical features and provide reliable predictions. The confusion matrices further illustrated the models' performance metrics, including precision, recall, and F1 scores. These metrics are crucial in assessing the models' ability to correctly classify survivors and non-survivors, providing a balanced view of their predictive capabilities. For instance, both models showed reasonable precision and recall values across survival and non-survival categories, suggesting they were effective in making informed predictions based on the Titanic dataset's features.

Overall, the Titanic dataset analysis underscores the importance of data preprocessing, feature engineering, and model selection in achieving accurate predictions. By leveraging these insights, data scientists can better understand historical events like the Titanic disaster and extract meaningful patterns from complex datasets.

In the Income dataset analysis, we aimed to predict income levels using various factors like Age and Fare details. To prepare the data for modeling, we first removed duplicate entries to keep the dataset clean and avoid repetition. For missing values marked with ”?”, we filled them using the mode for categorical features. The categorical features in the dataset are ’workclass’, ’occupation’, and ’native.country’. This approach helped ensure that all data points were accounted for and reduced any biases that could affect our analysis later on.

Exploratory data analysis highlighted a significant impact on income levels based on gender, marital status, and educational status. Visualizations illustrated these trends clearly, showcasing how factors such as gender and marital status influence individuals' earning potential. For instance, males were observed to have higher income levels compared to females across various marital statuses. Feature selection using Elastic Net regularization identified ’age’, ’fnlwgt’ (final weight), and ’hours per week’ as key predictors of income. This step provided valuable insights into the demographic and work-related factors that contribute most significantly to higher earnings. By focusing on these predictors, the model could effectively capture the variability in income levels among individuals in the dataset.

Normalization of the dataset standardized the range of values across different features, ensuring that no single feature disproportionately influenced the model's predictions due to its scale. This preprocessing step is crucial in maintaining the integrity of predictive models, allowing for fair comparisons and accurate predictions across diverse datasets.

During model training and evaluation, Gaussian Naive Bayes (NB) and Categorical Naive Bayes were employed, each tailored to handle different types of data. Gaussian NB, suitable for continuous features like ’age’ and ’hours per week’, demonstrated strong predictive performance with a remarkable 100% accuracy. This high accuracy suggests that the model effectively differentiated between income levels based on continuous variables, providing robust predictions. Conversely, Categorical NB, after discretizing continuous data into bins, achieved 82% accuracy. This approach leveraged the discretized nature of categorical features to make informed predictions about income levels, demonstrating its efficacy in handling discrete data types effectively. The confusion matrices provided further insights into the models' performance metrics, including precision, recall, and F1 scores. These metrics offered a comprehensive assessment of the models' ability to classify individuals based on their income levels, highlighting their strengths and areas for improvement in predictive accuracy. In conclusion, the Income dataset analysis emphasizes the significance of data preprocessing, feature selection, and model evaluation in predicting income levels based on different factors.

VII CONCLUSION

In conclusion, our analysis of the Titanic and Income datasets highlights the pivotal role of meticulous data preprocessing, thoughtful feature engineering, and judicious model selection in achieving precise predictions. For the Titanic dataset, preprocessing involved removing duplicates and imputing missing values, ensuring data integrity. Feature engineering, such as creating the ’Family’ feature, enriched our understanding of survival factors beyond individual demographics. Gaussian Naive Bayes effectively leveraged the Gaussian distribution assumption to predict survival based on continuous features like age and fare. In contrast, Categorical Naive Bayes handled discrete data like gender and passenger class adeptly, revealing insights into survival dynamics across different groups.

Similarly, in the Income dataset analysis, preprocessing steps such as handling missing values and normalizing numerical features prepared the data for modeling. Feature engineering efforts, including creating composite features and selecting informative predictors through Elastic Net regularization, enhanced the model's predictive power. Gaussian Naive Bayes excelled in predicting income levels based on continuous variables like age and hours worked per week, while Categorical Naive Bayes effectively categorized individuals into income groups based on categorical attributes such as occupation and education level.

Overall, these analyses underscore the versatility of Naive Bayes classifiers in handling diverse data types and extracting meaningful patterns for predictive tasks. By demonstrating strong performance across both datasets, Naive Bayes lays a solid foundation for exploring more advanced modeling techniques in future research and applications, ensuring robust insights and informed decision-making in complex domains.

VIII APPENDICES

A Equations

Bayes Theorem

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \tag{1}$$

where,

$P(C|X)$ is the posterior probability of class C given features X.

$P(X|C)$ is the likelihood of X given C.

$P(C)$ is the prior probability of C.

$P(X)$ is the probability of X.

Prior Probability

The prior probability for a class C_k is given as:

$$P(C_k) = \frac{N_k}{N} \tag{2}$$

where,
N_k is the number of instances belonging to class C_k.
N is the total number of instances in the training dataset.

Likelihood
For a feature vector $X = X_1, X_2, \cdot, X_n$ assuming conditional independence between the features, the Likelihood $P(X|C)$ is given as:

$$P(X|C) = P(X_1, X_2, \ldots, X_n|C) \tag{3}$$

or,

$$P(X|C) \approx P(X_1|C) \cdot P(X_2|C) \cdot \ldots \cdot P(X_n|C) \tag{4}$$

Gaussian Naive Bayes
The likelihood for Gaussian Naive Bayes is:

$$P(X_i|C) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left(-\frac{(X_i - \mu)^2}{2\sigma^2}\right) \tag{5}$$

Categorical Naive Bayes
The likelihood for Categorical Naive Bayes is:

$$P(X_i = j|C) = \theta_{i,j} \tag{6}$$

where,
X_i is the observed value of the categorical feature.
j is the specific category X_i belongs to.
$\theta_{i,j}$ is the probability of X_i belonging to category j given class C.

Precision

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{7}$$

Recall

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{8}$$

F1 Score

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

Macro Average

$$\text{Macro Average} = \frac{1}{J}\sum_{j=1}^{J} \text{Metric}_j \tag{10}$$

Weighted Average

$$\text{Weighted Average} = \sum_{j=1}^{J} \frac{N_j}{N} \cdot \text{Metric}_j \tag{11}$$

Accuracy

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}} \tag{12}$$

where,

True Positive = Model predicted correctly for true data
False Positive = Model predicted incorrectly for true data
True Negative = Model predicted correctly for false data
False Negative = Model predicted incorrectly for false data
Metric can be either precision, recall or F1 score.

B Figures

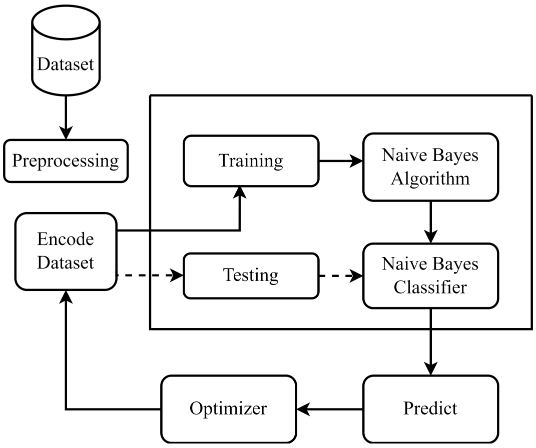

Figure 1: System Block Diagram

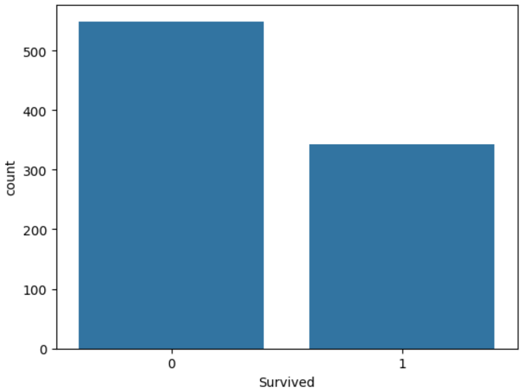

Figure 2: Titanic Plot of Survival Count

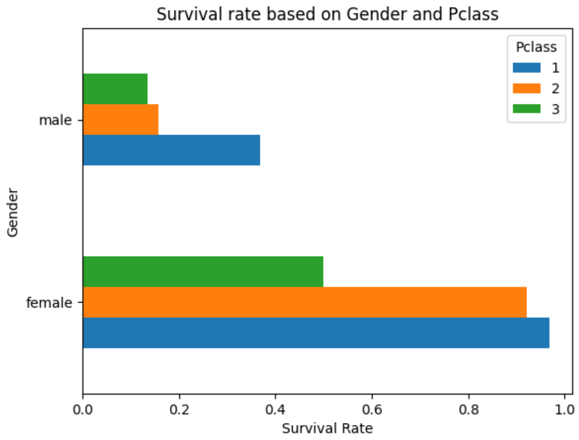

Figure 3: Titanic Barplot of Survival based on Gender and Pclass

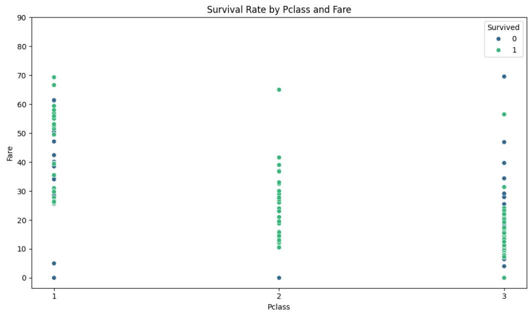

Figure 4: Titanic Plot of Survival based on Fare and Pclass

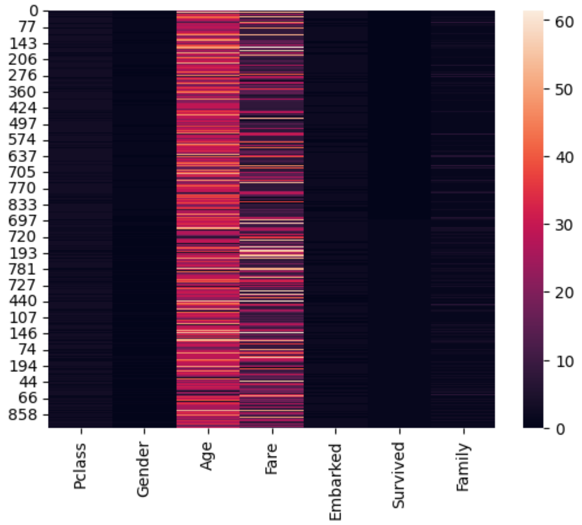

Figure 5: Titanic Heatmap showing Correlation

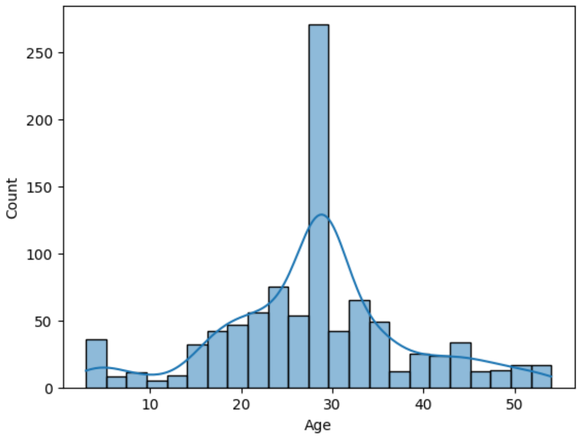

Figure 6: Titanic Histogram Plot of Age

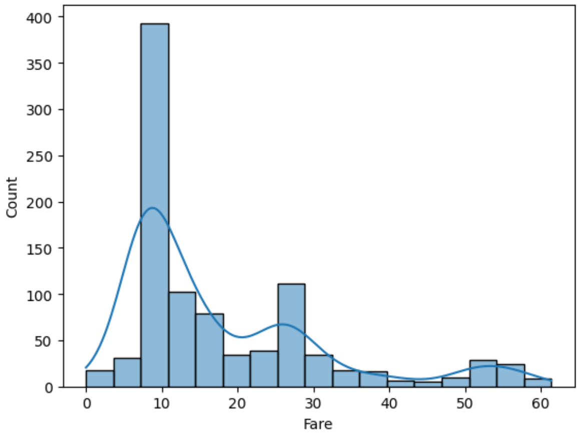

Figure 7: Titanic Histogram Plot of Fare

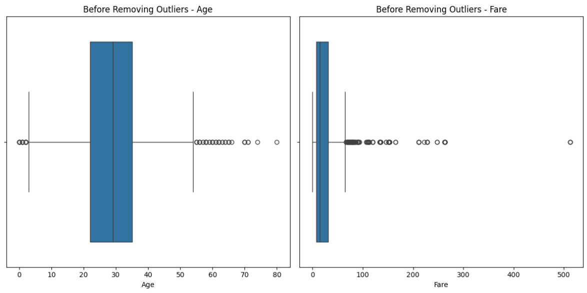

Figure 8: Titanic Plot Before Removing Outliers

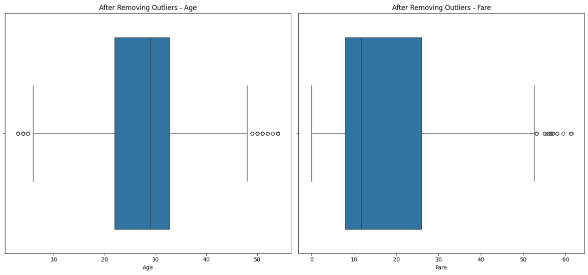

Figure 9: Titanic Plot After Removing Outliers

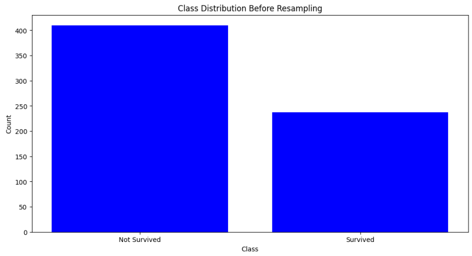

Figure 10: Titanic Plot Before Resampling

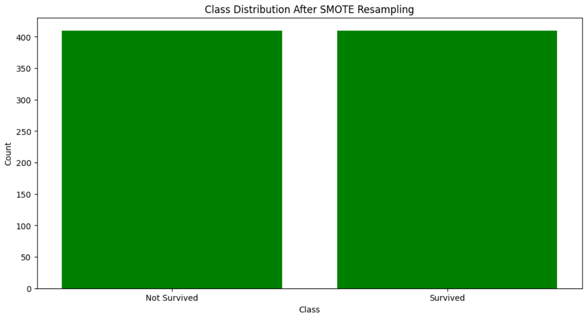

Figure 11: Titanic Plot After Resampling

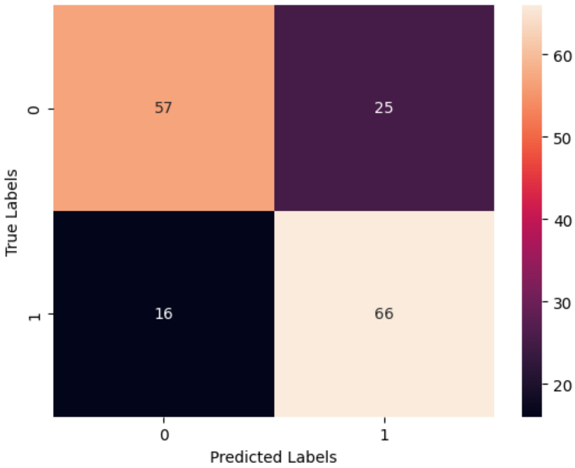

Figure 12: Titanic Confusion Matrix for Gaussian

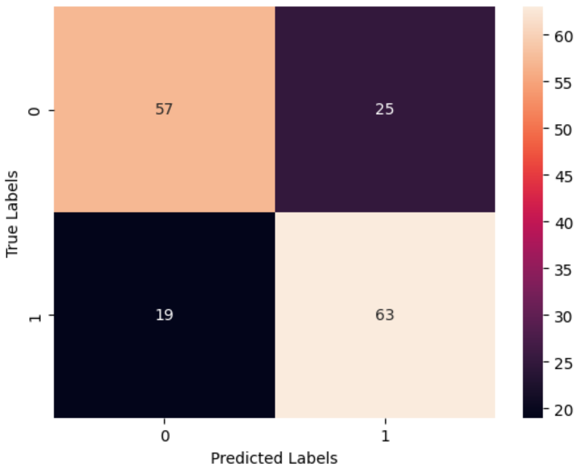

Figure 13: Titanic Confusion Matrix for Categorical

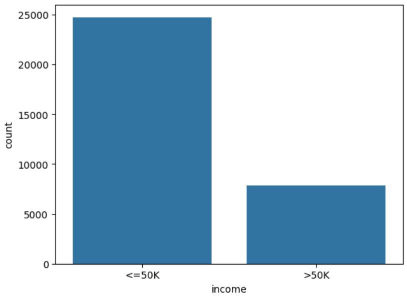

Figure 14: Income count Plot

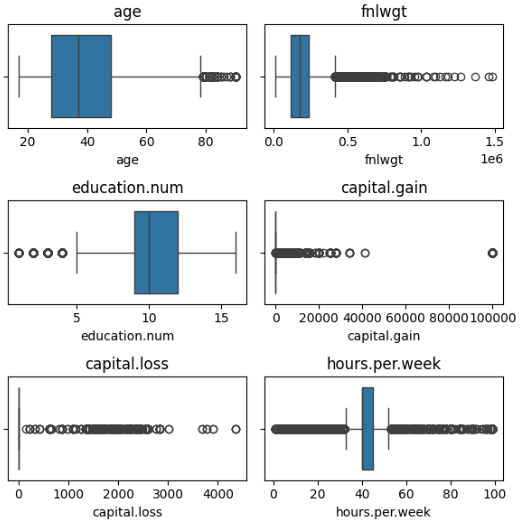

Figure 15: Income Plot Before Removing Outliers

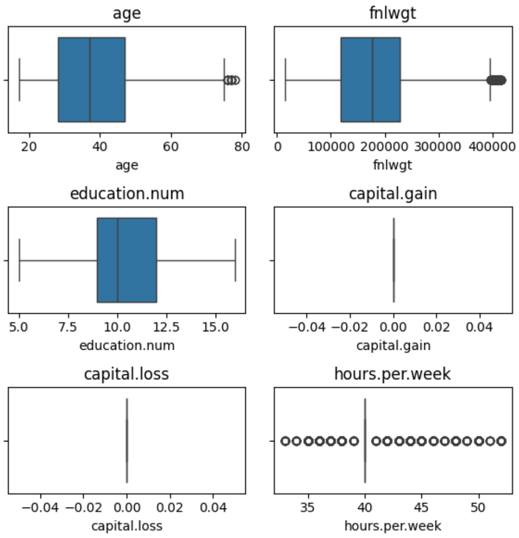

Figure 16: Income Plot After Removing Outliers

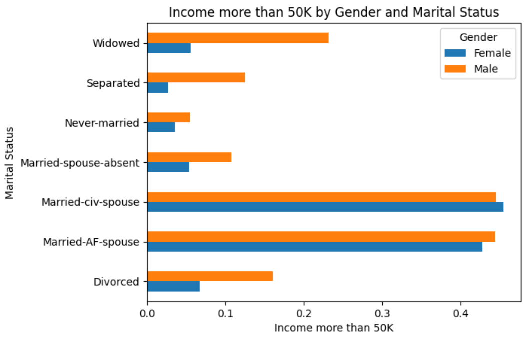

Figure 17: Income Plot Based on Gender and Marital Status

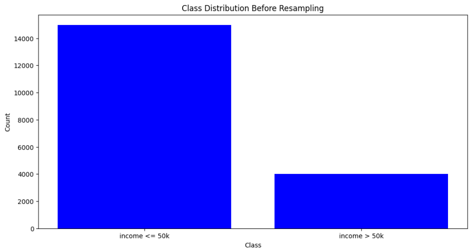

Figure 18: Income Plot Before Resampling

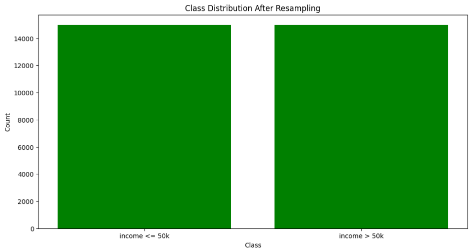

Figure 19: Income Plot After Resampling

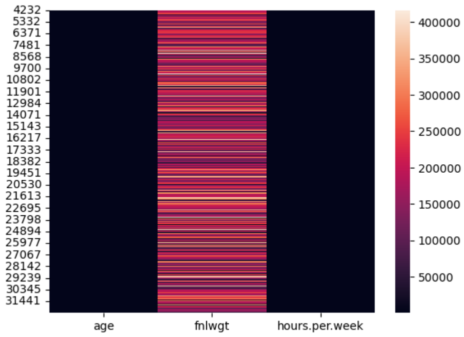

Figure 20: Income Correlation Plot having selected features

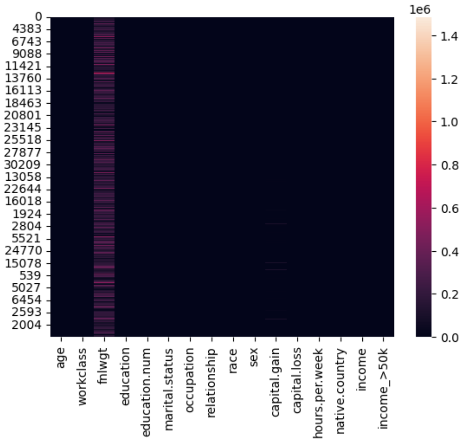

Figure 21: Income Correlation Plot having all features

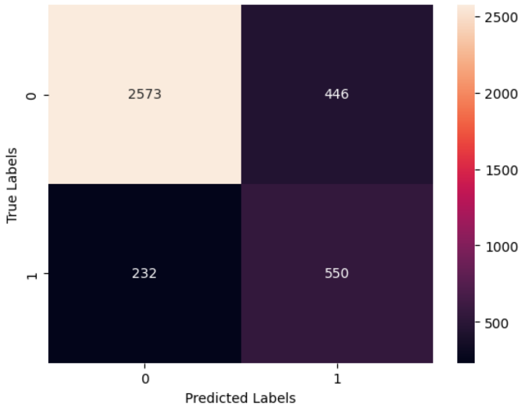

Figure 22: Income Confusion Matrix for Categorical without outliers

Table 1: Titanic Classification report using GaussianNB

	Precision	Recall	F1 score
Not Survived	0.78	0.70	0.74
Survived	0.73	0.80	0.76
accuracy			0.75
macro avg	0.75	0.75	0.75
weighted avg	0.75	0.75	0.75

Table 2: Titanic Classification report using CategoricalNB

	Precision	Recall	F1 score
Not Survived	0.75	0.70	0.72
Survived	0.72	0.77	0.74
accuracy			0.73
macro avg	0.73	0.73	0.73
weighted avg	0.73	0.73	0.73

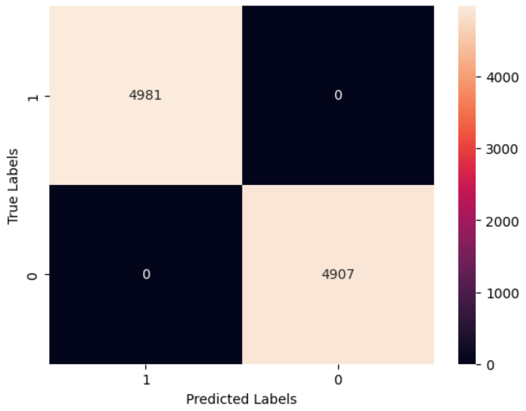

Figure 23: Income Confusion Matrix for Gaussian

C Tables

Table 3: Income Classification report using GaussianNB Removing Outliers

	Precision	Recall	F1 score
Income <= 50K	0.91	0.83	0.87
Income > 50K	0.51	0.67	0.58
accuracy			0.80
macro avg	0.71	0.75	0.72
weighted avg	0.83	0.80	0.81

Table 4: Income Classification report using CategoricalNB Removing Outliers

	Precision	Recall	F1 score
Income <= 50K	0.92	0.85	0.88
Income > 50K	0.55	0.70	0.62
accuracy			0.82
macro avg	0.73	0.78	0.75
weighted avg	0.84	0.82	0.83

Table 5: Income Classification report using GaussianNB

	Precision	Recall	F1 score
Income <= 50K	1.0	1.0	1.0
Income > 50K	1.0	1.0	1.0
accuracy			1.0
macro avg	1.0	1.0	1.0
weighted avg	1.0	1.0	1.0

REFERENCES

[1] N. Chakrabarty and S. Biswas, “A statistical approach to adult census income level prediction,” in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. IEEE, 2018, pp. 207–212.

[2] I. B. A. Peling, I. N. Arnawan, I. P. A. Arthawan, and I. G. N. Janardana, “Implementation of data mining to predict period of students study using naive bayes algorithm,” *Int. J. Eng. Emerg. Technol*, vol. 2, no. 1, p. 53, 2017.

[3] F. Harahap, A. Y. N. Harahap, E. Ekadiansyah, R. N. Sari, R. Adawiyah, and C. B. Harahap, “Implementation of naïve bayes classification method for predicting purchase,” in *2018 6th International Conference on Cyber and IT Service Management (CITSM)*. IEEE, 2018, pp. 1–5.

[4] D. Hemanth *et al.*, “Higher classification accuracy of income class using decision tree algorithm over naive bayes algorithm,” 2022.

Kristina Ghimire is currently pursuing her undergraduate degree in Computer Engineering at IOE, Thapathali Campus. Her research interests encompass various areas, including data mining, machine learning, and deep learning.(THA077BCT023)

Punam Shrestha is currently pursuing her undergraduate degree in Computer Engineering at IOE, Thapathali Campus. Her research interests encompass various areas, including data mining, and web development.(THA077BCT038)