Faculty of Engineering & Technology

Electrical & Computer Engineering Department

ENCS5341 Machine Learning and Data Science

Assignment3

Prepared by:

Jana Herzallah	1201139	section (1)
Lana Badwan	1200071	section (2)

Instructor: Dr.Yazan Abufarha

Date: 21/1/2024

Table of contents:

Table of figures

1. Introduction:

In this report, we take on a journey to explore and evaluate various machine learning models applied to a dataset aimed at predicting income levels. The dataset encompasses a multitude of features, ranging from demographic attributes such as age, workclass, and education to socio-economic indicators like marital status and occupation. Our primary objective is to discern whether an individual's income surpasses the $50,000 threshold or not. The models under scrutiny include fundamental algorithms tailored to the intricacies of the dataset:

1. Nearest Neighbor Baseline (k=1 and k=3):

- Leveraging the K-Nearest Neighbors (KNN) classifier as a baseline, we experiment with different values of k, denoting the number of neighbors considered.

2. Logistic Regression with SGD and Validation:

- Fine-tuning logistic regression with SGD through a validation set and exploration of different regularization parameters (alpha).

3. Support Vector Machine (SVM) with Hinge Loss:

- Evaluating SVM with hinge loss, incorporating a validation set and varying regularization parameters (C).

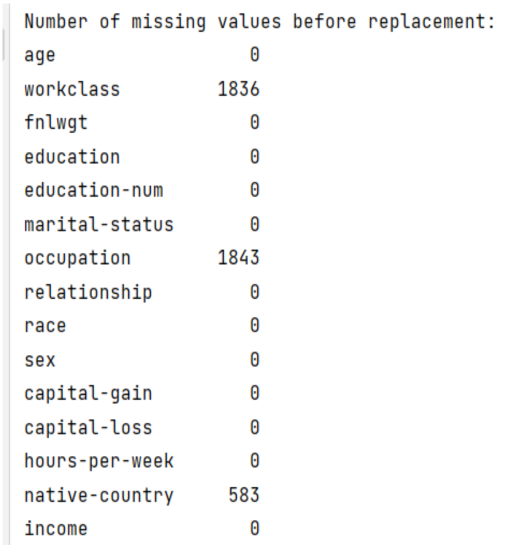
Our journey unfolds with an in-depth Exploratory Data Analysis (EDA), shedding light on the dataset's characteristics. Descriptive statistics and visualizations provide insights into the distribution of variables and relationships within the data.

For the evaluation of these models, key metrics such as accuracy, precision, recall, and F1-score take center stage. The report also delves into the application of grid search to optimize hyperparameters for select models.

Navigating through the project menu, users can seamlessly explore the dataset, benchmark the nearest neighbor baseline, delve into logistic regression with SGD and validation, assess SVM with hinge loss and validation, and gracefully exit the program. This comprehensive and systematic approach aims to discern the most effective model for predicting income levels on the given dataset, offering valuable insights into the interplay between diverse machine learning techniques and real-world socio-economic data.

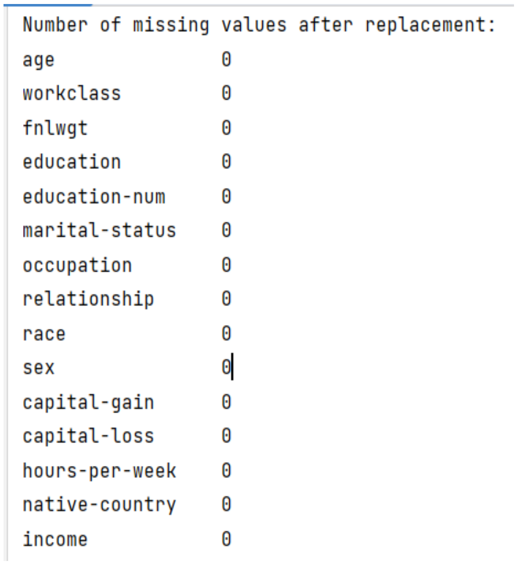
2. Dataset Description:

The dataset under consideration is a comprehensive and widely cited example frequently encountered in machine learning courses, particularly for data pre-processing and introductory practices in the field. Comprising 15 columns and 32561 examples as training data and 16281 testing data, the focal point is the 'Income' attribute, divided into two classes: <=50K and >50K. The dataset incorporates 14 attributes and a label, ranging from continuous variables such as age, fnlwgt, education-num, capital-gain, capital-loss, and hours-per-week, to categorical features including workclass, education, marital-status, occupation, relationship, race, sex, and native-country. This rich array of attributes offers a diverse perspective on the factors influencing an individual's annual income. The categorical variables span a spectrum of personal and socio-economic aspects, including work environment, educational background, marital status, occupation type, and demographic details like race and gender. There were some missing data that we filled by the mean if it’s a numerical field and by the mode if it’s a categorical data.[1]



```
Number of missing values before replacement:
age                  0
workclass         1836
fnlwgt               0
education            0
education-num        0
marital-status       0
occupation        1843
relationship         0
race                 0
sex                  0
capital-gain         0
capital-loss         0
hours-per-week       0
native-country     583
income               0
```

Figure 1:missing data before replacement

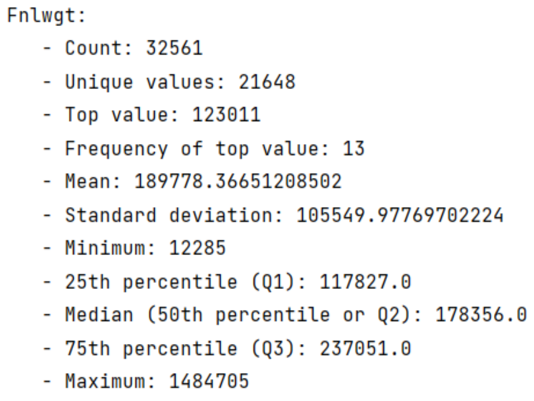


```
Number of missing values after replacement:
age               0
workclass         0
fnlwgt            0
education         0
education-num     0
marital-status    0
occupation        0
relationship      0
race              0
sex               0
capital-gain      0
capital-loss      0
hours-per-week    0
native-country    0
income            0
```

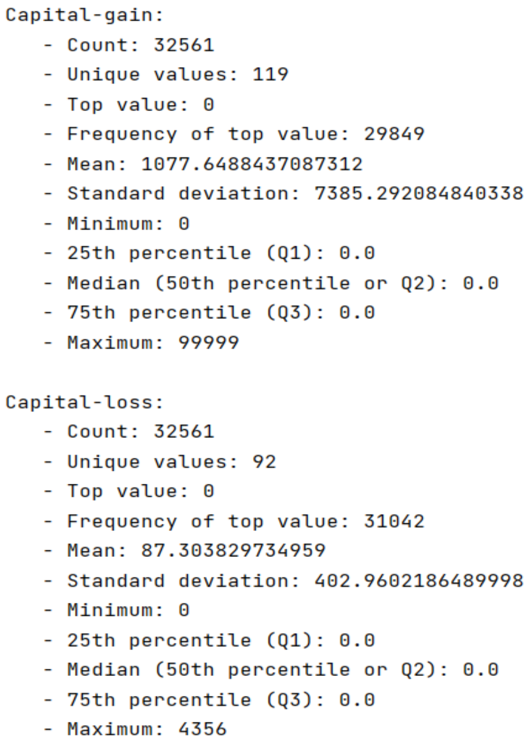
Figure 2 :missing data after replacement

2.1 Dataset Overview and Descriptive Statistics: Unveiling the Characteristics of the Adult Income Dataset:

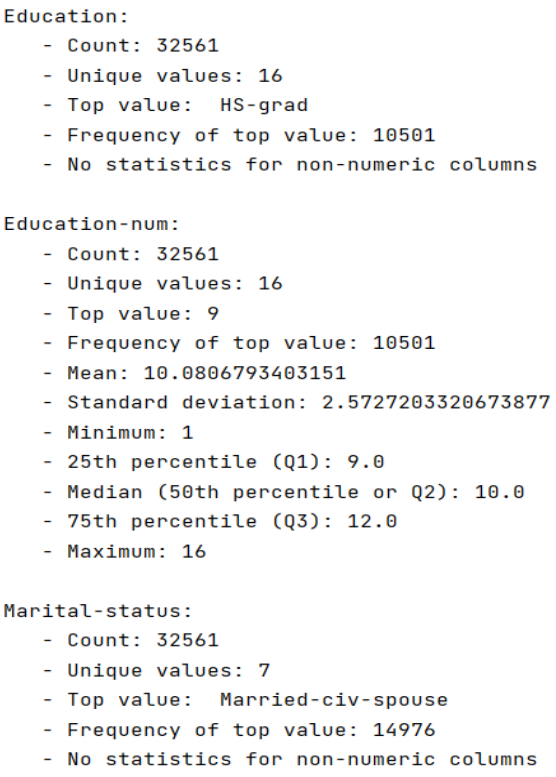
This section provides an in-depth exploration of the dataset used in the machine learning project, offering insights into its structure, key features, and summary statistics. It aims to give readers a comprehensive understanding of the data's composition and initial properties.



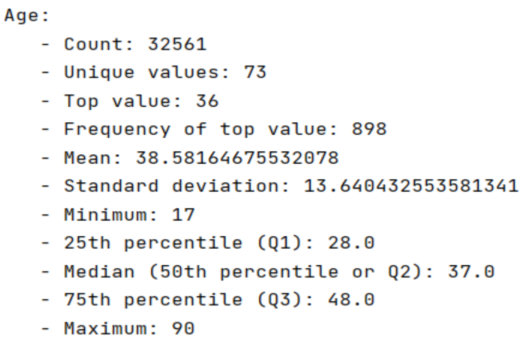
```
Fnlwgt:
   - Count: 32561
   - Unique values: 21648
   - Top value: 123011
   - Frequency of top value: 13
   - Mean: 189778.36651208502
   - Standard deviation: 105549.97769702224
   - Minimum: 12285
   - 25th percentile (Q1): 117827.0
   - Median (50th percentile or Q2): 178356.0
   - 75th percentile (Q3): 237051.0
   - Maximum: 1484705
```



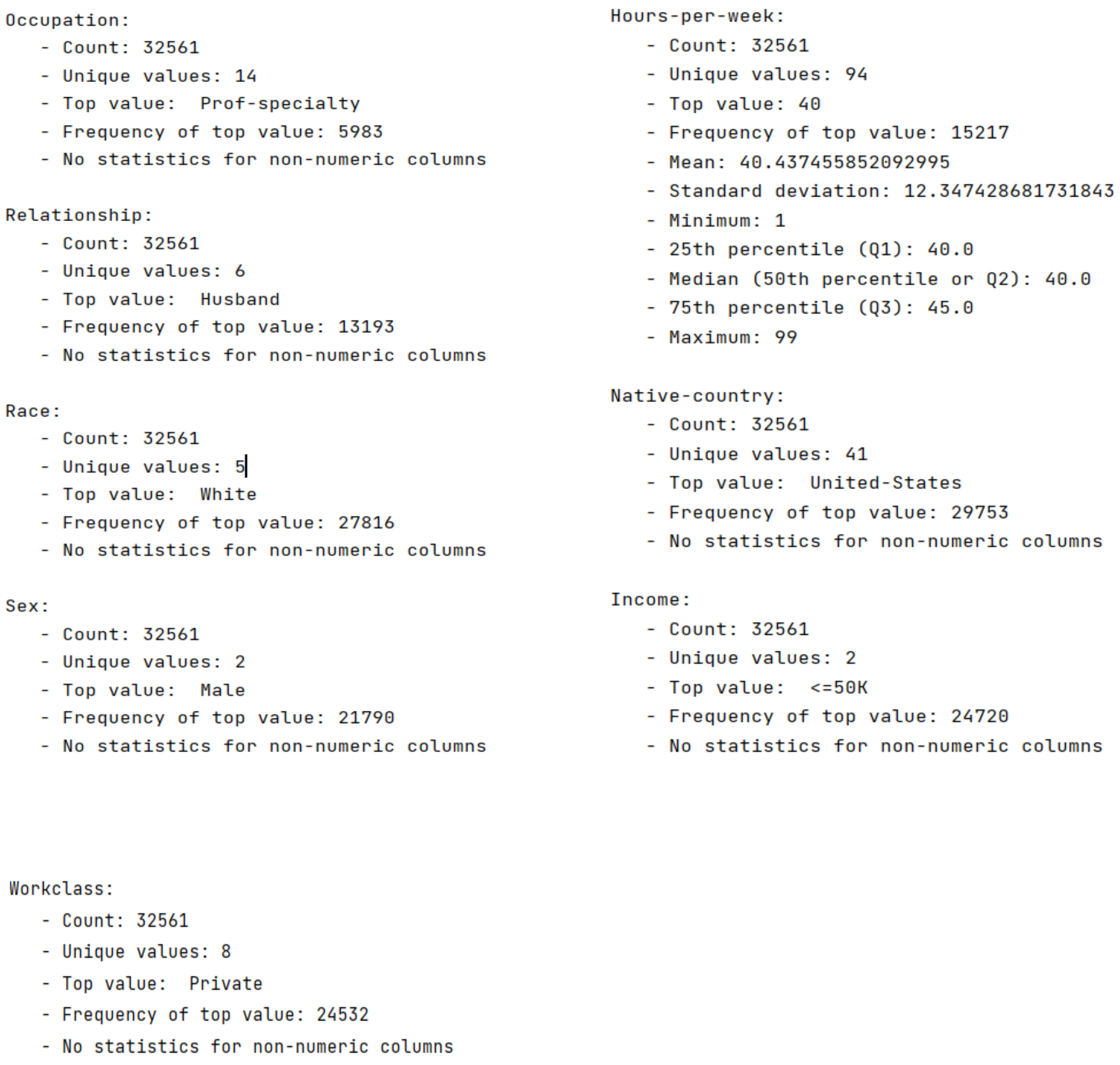
```
Capital-gain:
   - Count: 32561
   - Unique values: 119
   - Top value: 0
   - Frequency of top value: 29849
   - Mean: 1077.6488437087312
   - Standard deviation: 7385.292084840338
   - Minimum: 0
   - 25th percentile (Q1): 0.0
   - Median (50th percentile or Q2): 0.0
   - 75th percentile (Q3): 0.0
   - Maximum: 99999

Capital-loss:
   - Count: 32561
   - Unique values: 92
   - Top value: 0
   - Frequency of top value: 31042
   - Mean: 87.303829734959
   - Standard deviation: 402.9602186489998
   - Minimum: 0
   - 25th percentile (Q1): 0.0
   - Median (50th percentile or Q2): 0.0
   - 75th percentile (Q3): 0.0
   - Maximum: 4356
```



```
Education:
   - Count: 32561
   - Unique values: 16
   - Top value:  HS-grad
   - Frequency of top value: 10501
   - No statistics for non-numeric columns

Education-num:
   - Count: 32561
   - Unique values: 16
   - Top value: 9
   - Frequency of top value: 10501
   - Mean: 10.0806793403151
   - Standard deviation: 2.5727203320673877
   - Minimum: 1
   - 25th percentile (Q1): 9.0
   - Median (50th percentile or Q2): 10.0
   - 75th percentile (Q3): 12.0
   - Maximum: 16

Marital-status:
   - Count: 32561
   - Unique values: 7
   - Top value:  Married-civ-spouse
   - Frequency of top value: 14976
   - No statistics for non-numeric columns
```



```
Age:
   - Count: 32561
   - Unique values: 73
   - Top value: 36
   - Frequency of top value: 898
   - Mean: 38.58164675532078
   - Standard deviation: 13.640432553581341
   - Minimum: 17
   - 25th percentile (Q1): 28.0
   - Median (50th percentile or Q2): 37.0
   - 75th percentile (Q3): 48.0
   - Maximum: 90
```



```
Occupation:
    - Count: 32561
    - Unique values: 14
    - Top value:  Prof-specialty
    - Frequency of top value: 5983
    - No statistics for non-numeric columns

Relationship:
    - Count: 32561
    - Unique values: 6
    - Top value:  Husband
    - Frequency of top value: 13193
    - No statistics for non-numeric columns

Race:
    - Count: 32561
    - Unique values: 5
    - Top value:  White
    - Frequency of top value: 27816
    - No statistics for non-numeric columns

Sex:
    - Count: 32561
    - Unique values: 2
    - Top value:  Male
    - Frequency of top value: 21790
    - No statistics for non-numeric columns
```

```
Hours-per-week:
    - Count: 32561
    - Unique values: 94
    - Top value: 40
    - Frequency of top value: 15217
    - Mean: 40.437455852092995
    - Standard deviation: 12.347428681731843
    - Minimum: 1
    - 25th percentile (Q1): 40.0
    - Median (50th percentile or Q2): 40.0
    - 75th percentile (Q3): 45.0
    - Maximum: 99

Native-country:
    - Count: 32561
    - Unique values: 41
    - Top value:  United-States
    - Frequency of top value: 29753
    - No statistics for non-numeric columns

Income:
    - Count: 32561
    - Unique values: 2
    - Top value:  <=50K
    - Frequency of top value: 24720
    - No statistics for non-numeric columns
```

```
Workclass:
    - Count: 32561
    - Unique values: 8
    - Top value:  Private
    - Frequency of top value: 24532
    - No statistics for non-numeric columns
```

Figure 3:Quantitative measures for features

2.2 Dataset Visualization:

This section conducts Exploratory Data Analysis (EDA) to uncover underlying patterns, trends, and relationships. Through visualizations and statistical summaries, it aims to provide a nuanced perspective on the dataset, facilitating a deeper appreciation of its intricacies and potential challenges for machine learning modeling.

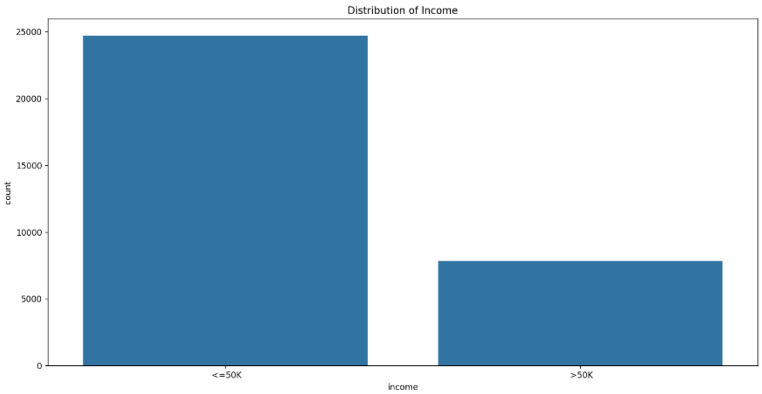


Figure 4: Distribution of income among training dataset

We can notice that most of the dataset has the label of <=50k which tells that the data is more concentrated in this class more than the other class and that will affect the training models.

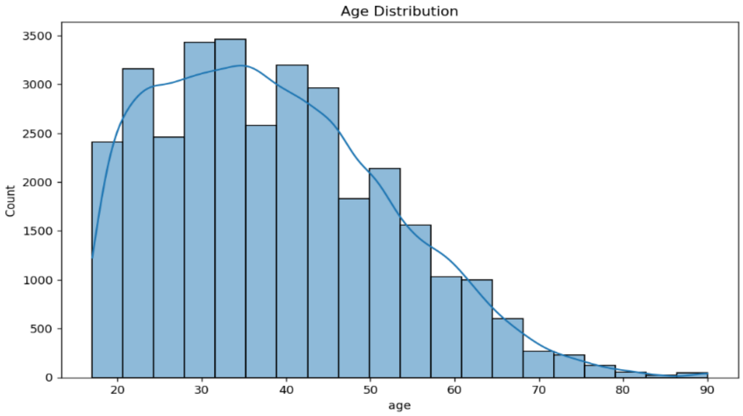


Figure 5 : Distribution of age among training dataset

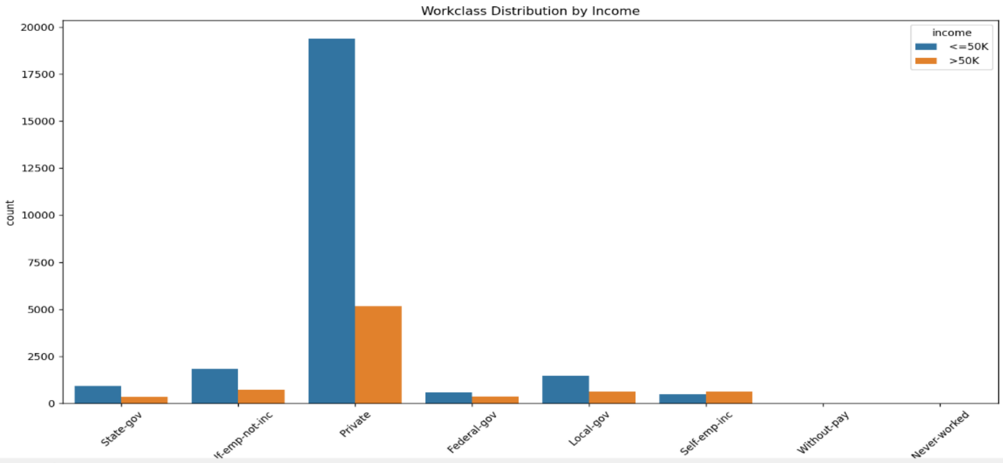


Figure 6:Distribution of workclass against income

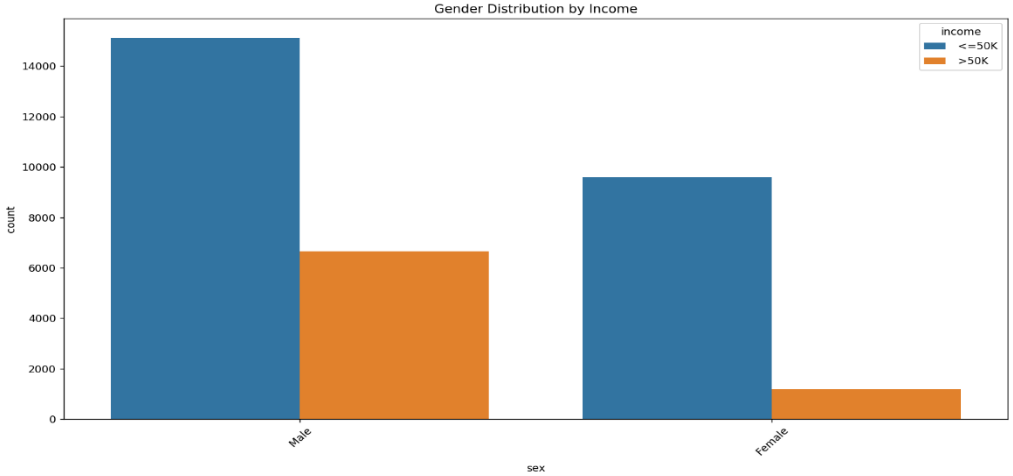


Figure 7:Distribution of Gender against income

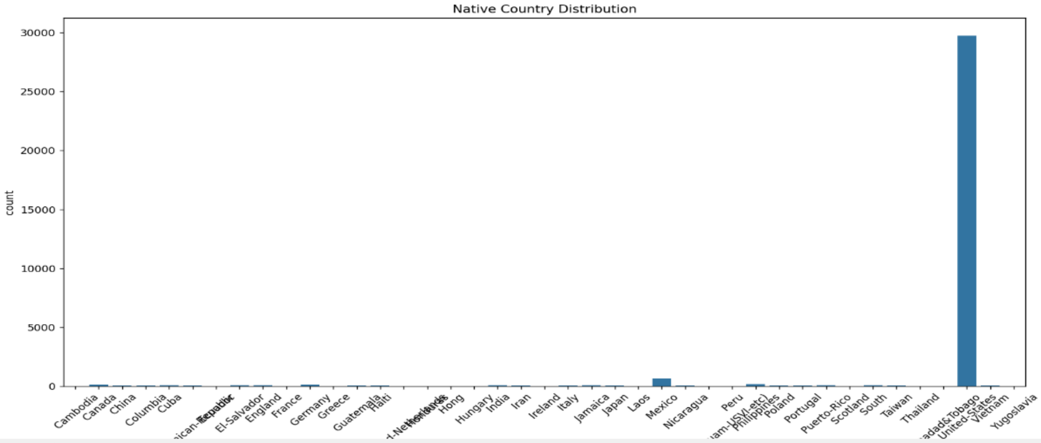


Figure 8:Distribution of native country against income

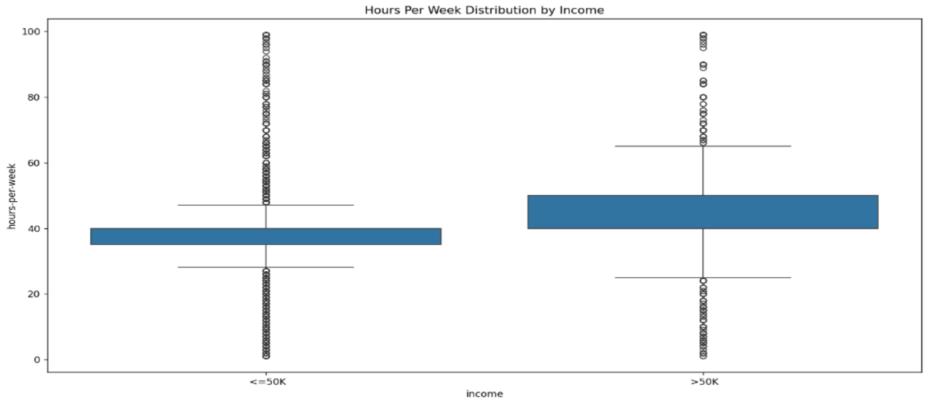


Figure 9:Distribution of Hours per week against income

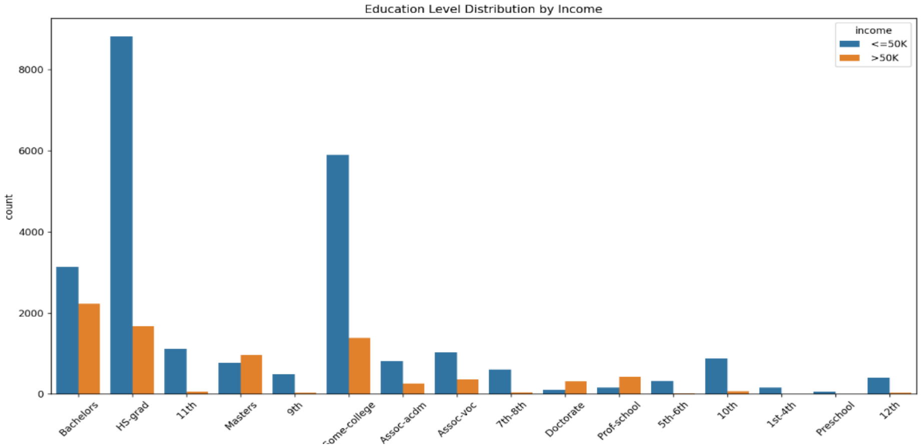


Figure 10:Distribution of Education level against income

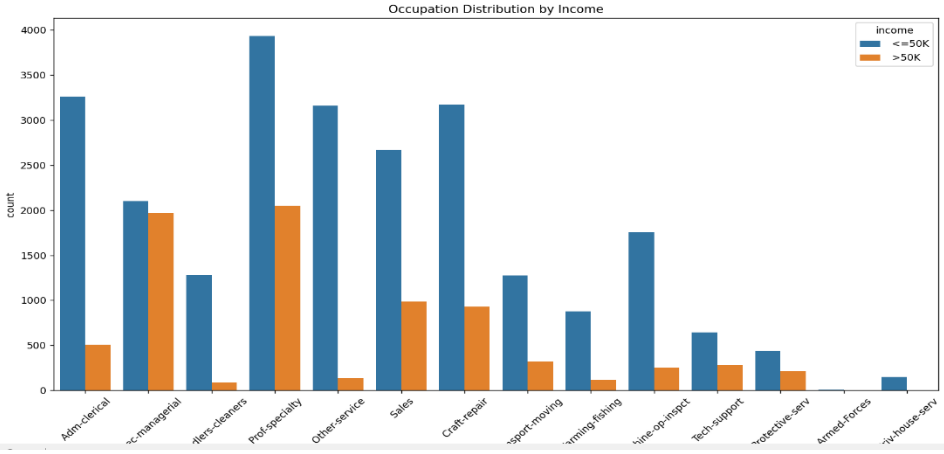


Figure 11:Distribution of Occupation against income

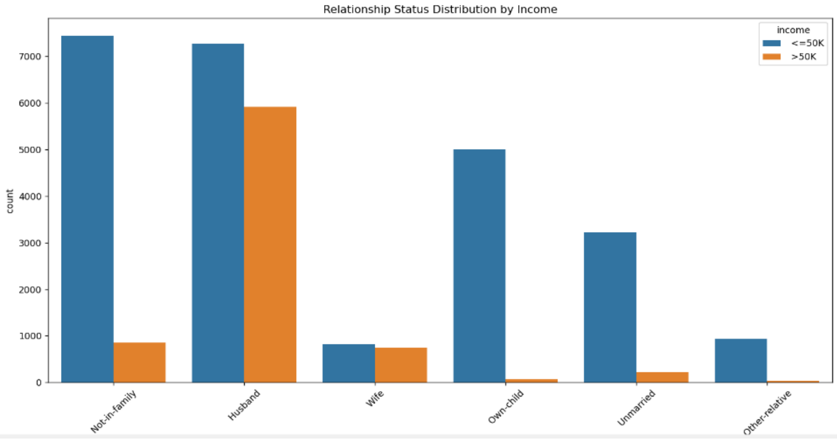


Figure 12:Distribution of Relationship against income

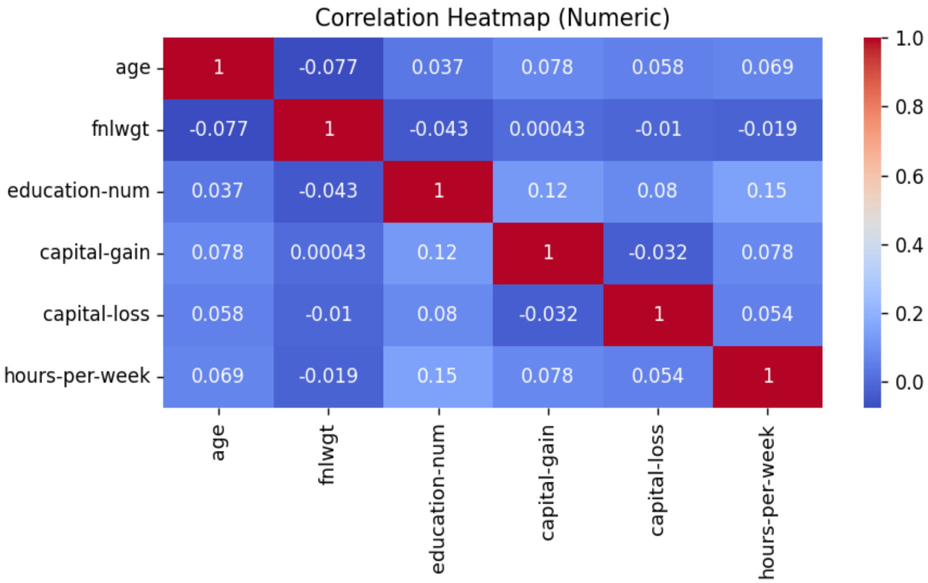


Figure 13:correlation heatmap for numeric data

These visualizations and statistics collectively offer a comprehensive understanding of the dataset, helping in feature selection, identifying potential patterns, and informing the subsequent steps in the machine learning pipeline.

3. Experiments and Results:

3.1 Dataset Preprocessing:

Before applying KNN model and the other two models, the dataset went through some steps of preprocessing summarized by:

1.Data Loading and Initial Exploration:

- ➢ pd.read_csv (Data Loading)

we loaded the dataset into a Pandas DataFrame, replacing "?" with NaN during the loading process.

2.Missing Value Replacement:

- ➢ replace_missing_values (Missing Value Imputation)

we used this replacing function to replace missing (NaN) values in the DataFrame by having numeric columns are imputed with the mean, while categorical columns are imputed with the mode.

3.One-Hot Encoding for Categorical Columns:

- ➢ convert_categorical_to_numeric (One-Hot Encoding)

we applied one-hot encoding to convert categorical columns into a numerical format. Then we created dummy variables for each category, dropping the first to avoid multicollinearity.

4.Splitting Data into Training and Testing Sets:

- ➢ split_data (Data Splitting)

we used this function to load the testing data and then we got it encoded. And by the end of this function all data (training and testing) are ready to be used in the models.

3.2 Training Models:

3.2.1 Base line model:

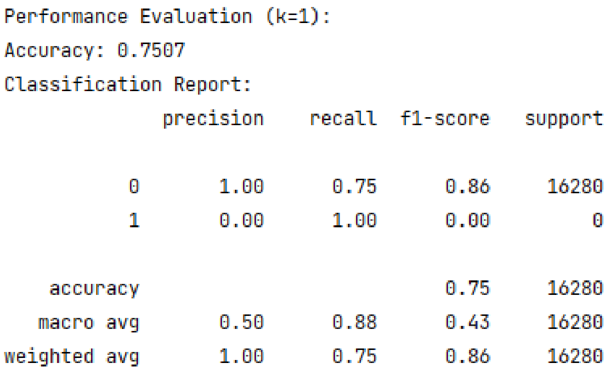
In this analysis, we employed the K-Nearest Neighbors (KNN) algorithm to create baseline models for binary classification, focusing on predicting the income class (>50K which is considered class 1 or <=50K which is class 0) based on a set of features. The evaluation

was conducted for two different values of k: k=1 and k=3.

Interpretation:

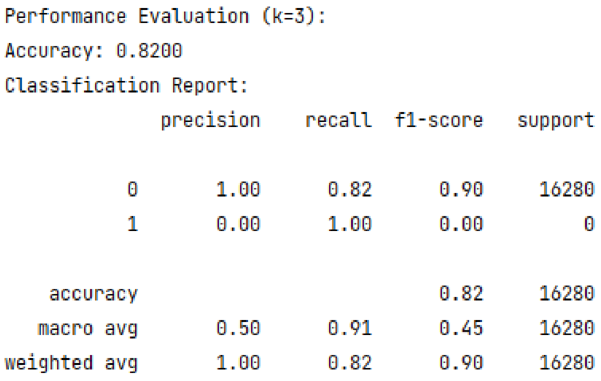
- The k=3 model exhibits superior performance compared to k=1, reflected in higher accuracy and improved precision, recall, and F1-score for class 0 which is <=50k that we explained above how its more frequent in the dataset.

- The dataset demonstrates class imbalance, particularly for class 1, resulting in metrics being unavailable for this class in both cases.



```
Performance Evaluation (k=1):
Accuracy: 0.7507
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.75      0.86     16280
           1       0.00      1.00      0.00         0

    accuracy                           0.75     16280
   macro avg       0.50      0.88      0.43     16280
weighted avg       1.00      0.75      0.86     16280
```

Figure 14:classification report for KNN, k=1



```
Performance Evaluation (k=3):
Accuracy: 0.8200
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.82      0.90     16280
           1       0.00      1.00      0.00         0

    accuracy                           0.82     16280
   macro avg       0.50      0.91      0.45     16280
weighted avg       1.00      0.82      0.90     16280
```

Figure 15 : classification report for KNN, k=3

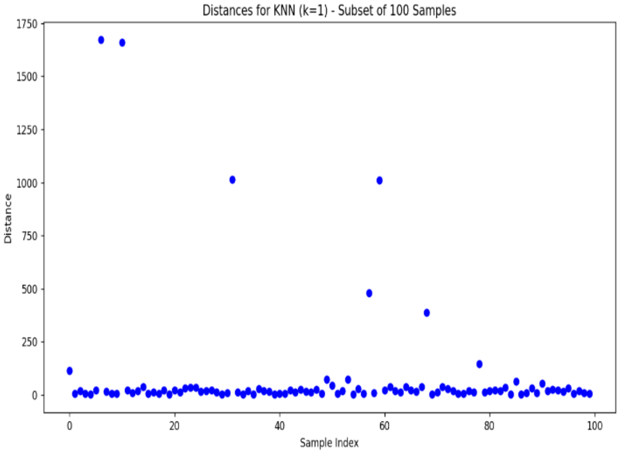


Figure 17: Distances of KNN between testing data and its k nearest neighbor, k=1

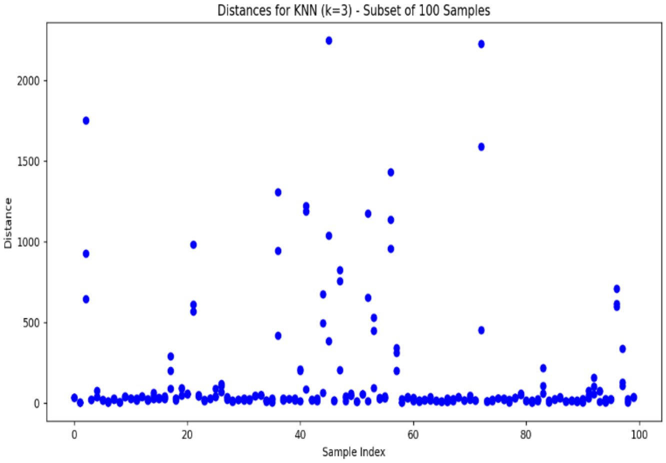


Figure 16: Distances of KNN between testing data and its k nearest neighbor, k=3

The scatter plot visualization shows the Euclidean distances between each data point in a subset of the test set and its k-nearest neighbors in a k-nearest neighbors (KNN) classification model. Each point on the plot represents a different sample, with the x-axis indicating the sample index and the y-axis representing the distances to its k-nearest neighbors. Blue dots on the plot represent the distances for each of the k-nearest neighbors. The visualization provides insight into the distribution of distances, helping to understand how well the KNN algorithm performs on the selected subset of the test set. Points close together indicate small distances, while more spread-out points may suggest larger distances to nearest neighbors.

3.2.2 Logistic Regression with Stochastic Gradient Decent:

The process involved checking how well a logistic regression model performs when we tweak a parameter called "alpha." We tried four different values for alpha: 0.0001, 0.001, 0.01, and 0.1.

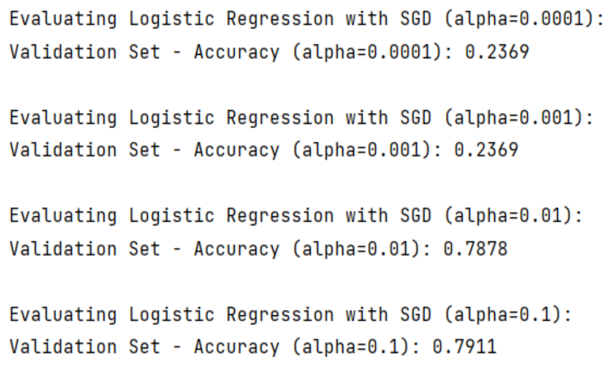
First, we trained the model with each alpha value using one part of the data. Then, we tested it on another part to see how accurate it was. For the lower alpha values (0.0001 and 0.001), the model didn't do well, with an accuracy of only around 0.24. But when we increased alpha to 0.01 and 0.1, the accuracy improved a lot, reaching about 0.79.

We decided that an alpha value of 0.1 worked the best because it gave the highest accuracy during this testing phase.

After picking the best alpha (0.1), we trained the model on the entire training dataset and then tested it on a separate test dataset. The final accuracy on the test set was 0.886, which is pretty good.

However, when looking more closely at the details (classification report), we noticed a challenge in predicting one of the class1 (>50k). The precision, recall, and F1-score for this class were all zero, and the support was reported as zero too. This raised some questions about the imbalance in the test data or potential issues in how we evaluated the model.

In conclusion, the process highlighted the importance of choosing the right alpha value to get the best results from the logistic regression model. It also pointed out potential areas for improvement, such as addressing class imbalances or tweaking other aspects of the model.



```
Evaluating Logistic Regression with SGD (alpha=0.0001):
Validation Set - Accuracy (alpha=0.0001): 0.2369

Evaluating Logistic Regression with SGD (alpha=0.001):
Validation Set - Accuracy (alpha=0.001): 0.2369

Evaluating Logistic Regression with SGD (alpha=0.01):
Validation Set - Accuracy (alpha=0.01): 0.7878

Evaluating Logistic Regression with SGD (alpha=0.1):
Validation Set - Accuracy (alpha=0.1): 0.7911
```

Figure 18: hyper parameters in logistic regression

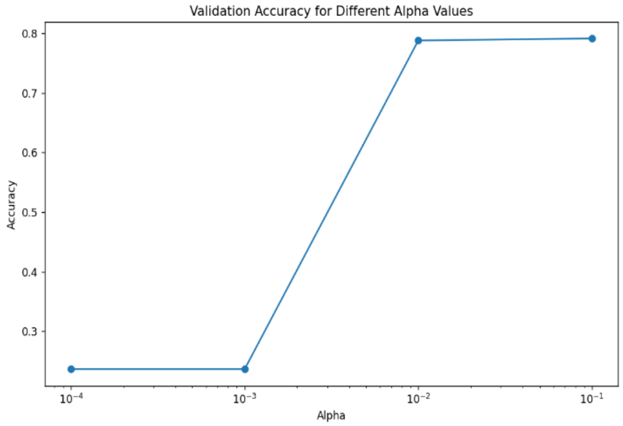
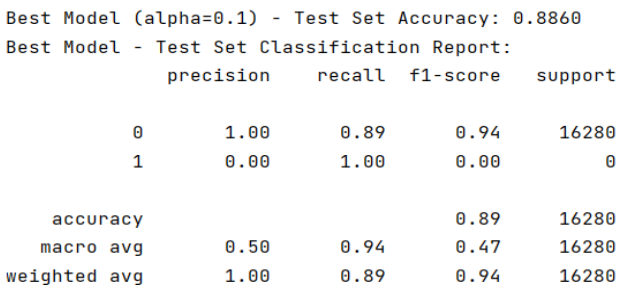


Figure 19:accuracy in terms of alpha

After finding the best alpha on validation set, we trained the best model with the highest value of alpha. And these are the classification report that were obtained.

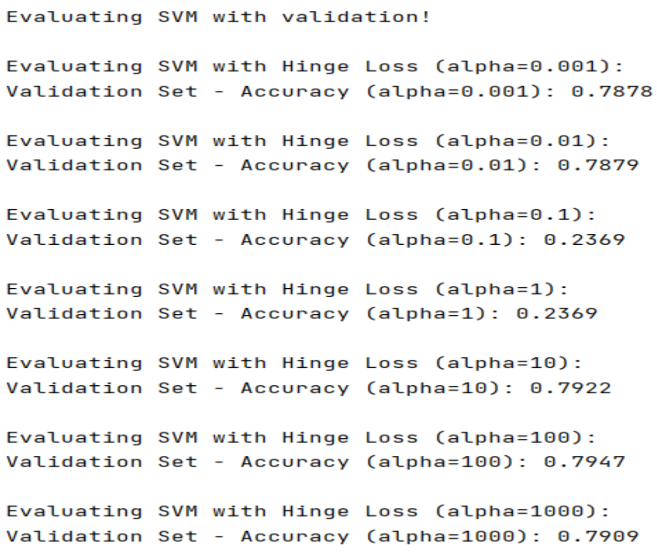


```
Best Model (alpha=0.1) - Test Set Accuracy: 0.8860
Best Model - Test Set Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.89      0.94     16280
           1       0.00      1.00      0.00         0

    accuracy                           0.89     16280
   macro avg       0.50      0.94      0.47     16280
weighted avg       1.00      0.89      0.94     16280
```

Figure 20:best logistic regression model

3.2.2 SVM with validation:

The procedure for evaluating the Support Vector Machine (SVM) with Hinge Loss involves splitting the data into training and validation sets for hyperparameter tuning. A loop iterates over different alpha values, initializing and training linear SVM models using the SGD Classifier with Hinge Loss. The models are evaluated on the validation set, and accuracies are recorded. A plot is created to visualize the relationship between validation accuracy and alpha values. The alpha associated with the highest validation accuracy is selected as optimal. A new SVM model is trained on the entire training set using the best alpha, and its performance is evaluated on the test set. Results, including validation accuracies and test set metrics, are displayed throughout the process for analysis. This systematic approach aims to identify the regularization parameter that optimizes the SVM model's generalization to unseen data.



```
Evaluating SVM with validation!

Evaluating SVM with Hinge Loss (alpha=0.001):
Validation Set - Accuracy (alpha=0.001): 0.7878

Evaluating SVM with Hinge Loss (alpha=0.01):
Validation Set - Accuracy (alpha=0.01): 0.7879

Evaluating SVM with Hinge Loss (alpha=0.1):
Validation Set - Accuracy (alpha=0.1): 0.2369

Evaluating SVM with Hinge Loss (alpha=1):
Validation Set - Accuracy (alpha=1): 0.2369

Evaluating SVM with Hinge Loss (alpha=10):
Validation Set - Accuracy (alpha=10): 0.7922

Evaluating SVM with Hinge Loss (alpha=100):
Validation Set - Accuracy (alpha=100): 0.7947

Evaluating SVM with Hinge Loss (alpha=1000):
Validation Set - Accuracy (alpha=1000): 0.7909
```

Figure 21: hyper parameters in SVM

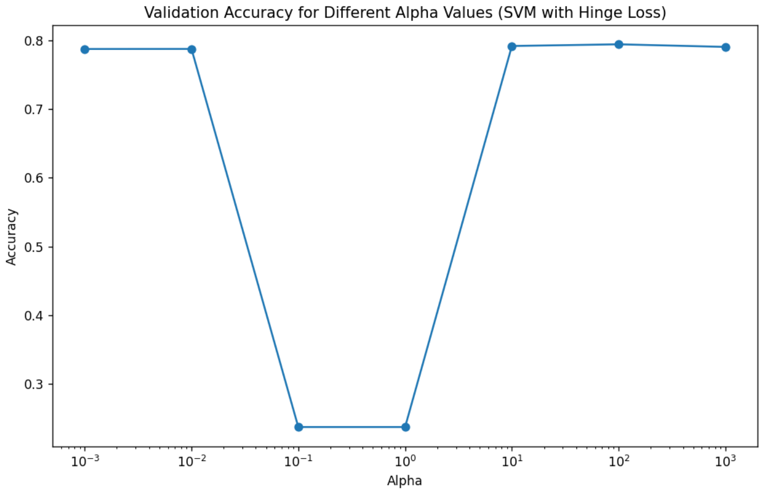
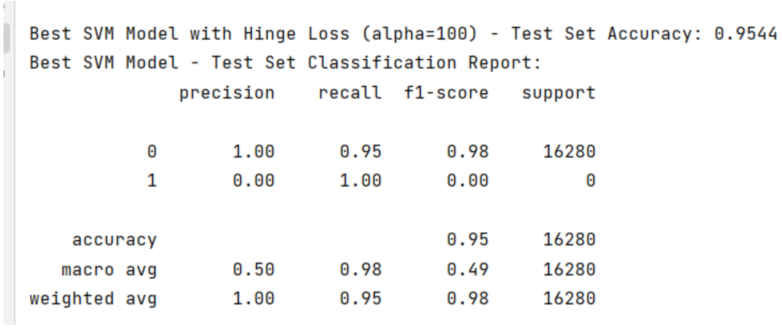


Figure 22: accuracy in terms of C



```
Best SVM Model with Hinge Loss (alpha=100) - Test Set Accuracy: 0.9544
Best SVM Model - Test Set Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.95      0.98     16280
           1       0.00      1.00      0.00         0

    accuracy                           0.95     16280
   macro avg       0.50      0.98      0.49     16280
weighted avg       1.00      0.95      0.98     16280
```

Figure 23:best SVM model

4. Analysis:

In the analysis of the K-Nearest Neighbors (KNN) classification model, it became evident that the model's predictive performance was significantly influenced by the class distribution within the dataset. Notably, the dataset exhibited a considerable imbalance between the two income classes, with a higher frequency of instances falling into the category of income less than or equal to $50,000 (<=50k) compared to the higher income bracket (>50k). This class imbalance posed a challenge for the KNN algorithm, as it tended to favor predicting the majority class due to its reliance on the majority class within the k-nearest neighbors.

To address this challenge, logistic regression was introduced as an alternative classification model. Logistic regression is well-suited for scenarios with imbalanced class distributions and provides a probabilistic interpretation of the predicted outcomes. By adopting logistic regression, the model could better handle the imbalanced dataset and mitigate the bias towards the majority class observed in the KNN model.

An interesting finding during the analysis was the impact of class frequency differences on the accuracy of the classification models. The class imbalance contributed to variations in the accuracy metrics, with the models demonstrating higher accuracy in predicting the majority class but struggling to perform well on the minority class. This observation emphasized the importance of considering class distribution and employing strategies to address imbalances for robust model evaluation.

Additionally, the computational complexity associated with Support Vector Machine (SVM) models, especially when using kernels, led to the exploration of alternative models. Stochastic Gradient Descent (SGD) emerged as a favorable choice due to its efficiency in handling large datasets and providing good performance in terms of accuracy. The adoption of SGD as a classification model proved to be a pragmatic solution, offering a balance between computational efficiency and predictive accuracy, especially in scenarios involving substantial data volumes.

5. Conclusions:

In our exploration of machine learning models for income prediction, we discovered key insights into model performance. The influence of class imbalances significantly impacted accuracy, notably in KNN. Logistic Regression emerged as a robust choice, mitigating bias toward the majority class.

Hyperparameter exploration emphasized the critical role of parameters like alpha in Logistic Regression. SVM, while powerful, posed computational challenges, leading to the adoption of Stochastic Gradient Descent for efficiency.

Despite successes, our models have limitations, particularly sensitivity to imbalances. Addressing these challenges requires advanced techniques and careful consideration of model complexity and efficiency.

As we progress from KNN to Logistic Regression with SGD and Validation and finally to SVM with Hinge Loss, we anticipate an ascending trend in accuracy, precision, recall, and F1-score. The models become progressively adept at capturing the underlying patterns and nuances within the dataset. It's crucial to note that the choice of algorithm and hyperparameter tuning significantly impacts the overall predictive performance, and this progression underscores the iterative refinement of models for enhanced accuracy and reliability in predicting income levels.

6. References:

https://www.kaggle.com/datasets/wenruliu/adult-income-dataset