

# **Final Project Report**

## **Census Income Prediction**

### **Group 28**

Mihir Naimish Dalal  
Himanshu Randad

[dalal.mi@northeastern.edu](mailto:dalal.mi@northeastern.edu)  
[randad.hi@northeastern.edu](mailto:randad.hi@northeastern.edu)

u

**Percentage of Effort Contributed by Student 1: 50%**

**Percentage of Effort Contributed by Student 2: 50%**

**Signature of Student 1: Mihir Naimish Dalal**

**Signature of Student 2: Himanshu Randad**

**Submission Date: April 12, 2024**

## **Problem Setting**

This study goes beyond salary prediction; through the analysis of variables including age, education, occupation, and marital status, it seeks to identify and rectify unfair financial patterns. Using carefully collected data, it aims to be a catalyst against economic inequality by developing a tool that not only predicts salaries but also promotes a more equitable distribution of financial achievement. It tells the story of people's characteristics and income levels to create a future where everyone has access to economic opportunities.

## **Problem definition**

The Adult Income Prediction project is like a crystal ball for understanding if someone earns more than \$50,000 a year based on census details. Using smart computer techniques, we dig into factors like age, education, job, and marital status to unveil the secrets behind income levels. It's not just about predicting paychecks –it's about tackling unfair money patterns. By carefully collecting and analyzing data, we aim to build a tool that not only forecasts incomes but also helps fight financial inequalities, making sure everyone gets a fair shot at economic success.

## **Data sources**

Introduction: The Adult Census Income dataset, compiled by Ronny Kohavi and Barry Becker, provides valuable insights into the socio-economic landscape of individuals based on data extracted from the 1994 Census Bureau database. With 15 columns and 32,562 data values, this dataset offers a detailed perspective on various demographic and employment-related factors influencing income levels.

Link for the dataset - [Adult Census Income \(kaggle.com\)](https://www.kaggle.com/datasets/arslanarshad/adult-census-income)

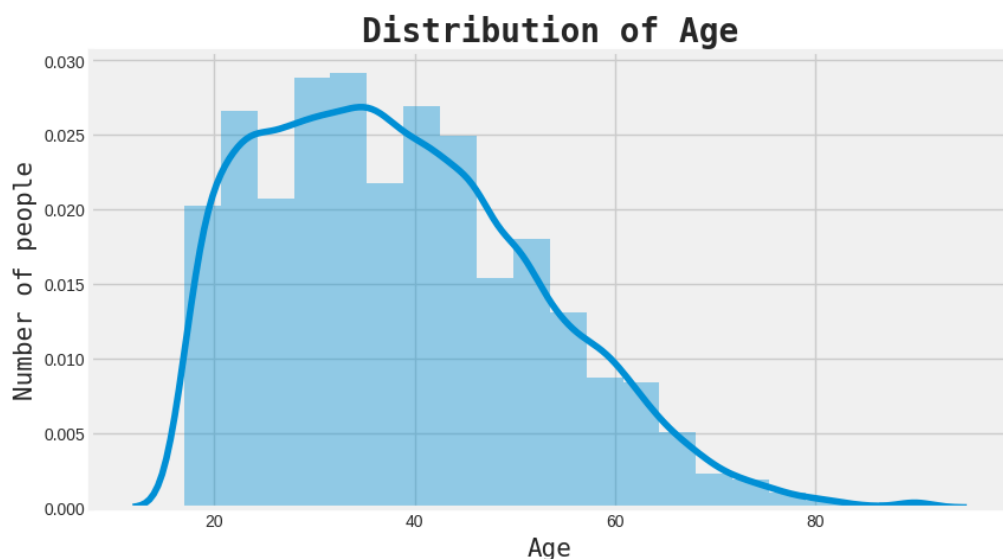
## **Data Description:**

1. Age: Represents the age of an individual, an integer greater than 0.
2. Work Class: Indicates the employment status of an individual, including options such as Private, Self-employed, Federal government, etc.
3. Final Weight (fnlwgt): Represents the estimated number of people the census entry represents, an integer greater than 0.
4. Education: Indicates the highest level of education achieved, with options such as Bachelors, High School graduate, Masters, etc.
5. Education Number: Numerical representation of the highest level of education achieved, an integer greater than 0.
6. Marital Status: Represents the marital status of an individual, including categories like Married, Divorced, never married, etc.

7. Occupation: Indicates the general type of occupation of an individual, including categories like Tech support, Sales, Farming, etc.
8. Relationship: Represents the individual's relative status to others, such as Wife, Husband, Unmarried, etc.
9. Race: Describes the individual's race, including options like White, Black, Asian Pacific-Islander, etc.
10. Sex: Represents the biological sex of the individual, either Male or Female.
11. Capital Gain: Indicates capital gains for an individual, an integer greater than or equal to 0.
12. Capital Loss: Indicates capital loss for an individual, an integer greater than or equal to 0.
13. Hours Per Week: Represents the number of hours an individual has reported working per week, a continuous variable.
14. Native Country: Indicates the country of origin for an individual, with options like United States, India, China, etc.
15. Label: Indicates whether an individual makes more than \$50,000 annually, represented as  $\leq 50k$  or  $> 50k$ .

### Data Visualization

**Distribution of Age:** A histogram was generated to understand the distribution of Age. The following results were obtained: Most of the people in the dataset were between the ages of 20 and 40.



**Distribution of Education and Income:** A bar plot was used to get insights into the education level of the population and what income level does each education level belong to. It was observed that most of the population had education level between High School to a Master's degree. The income was 0 for people in Preschool as it should be, and it gradually increased as the education level increased.

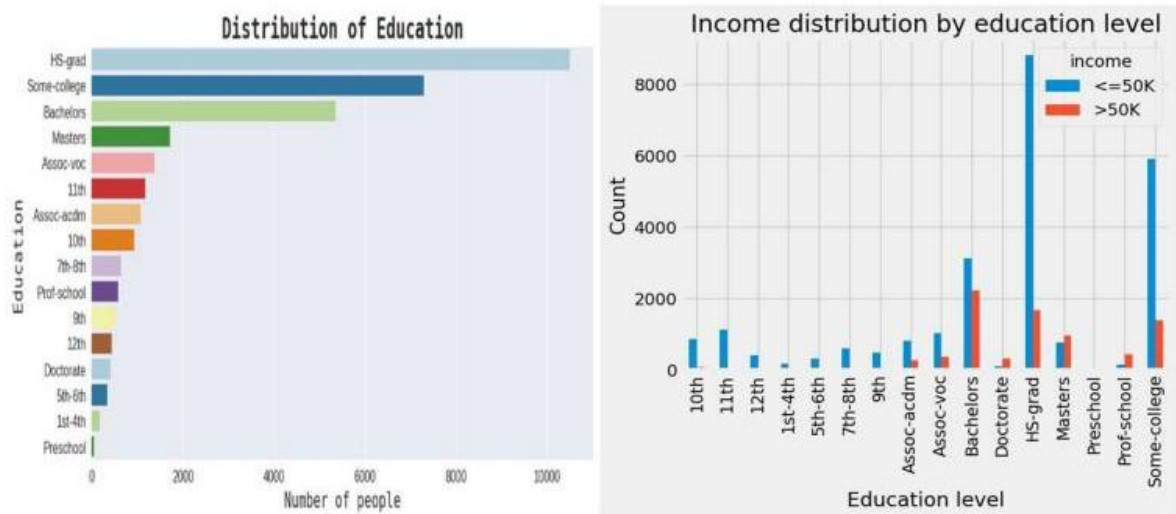
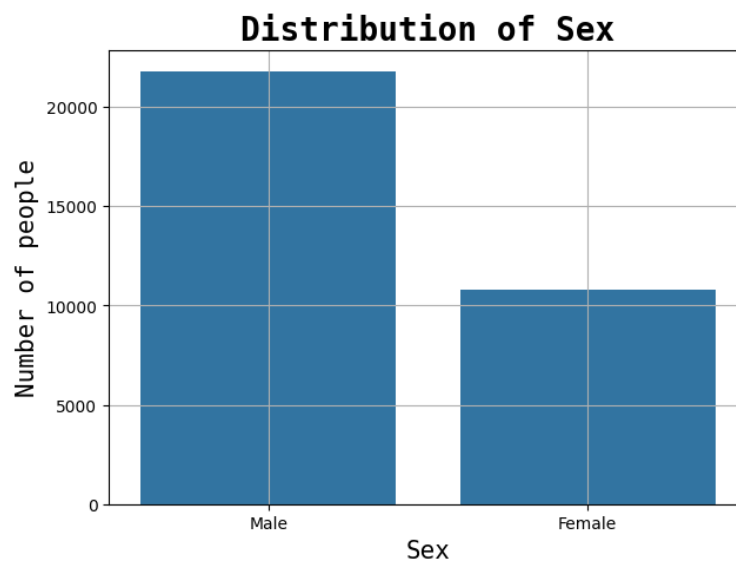
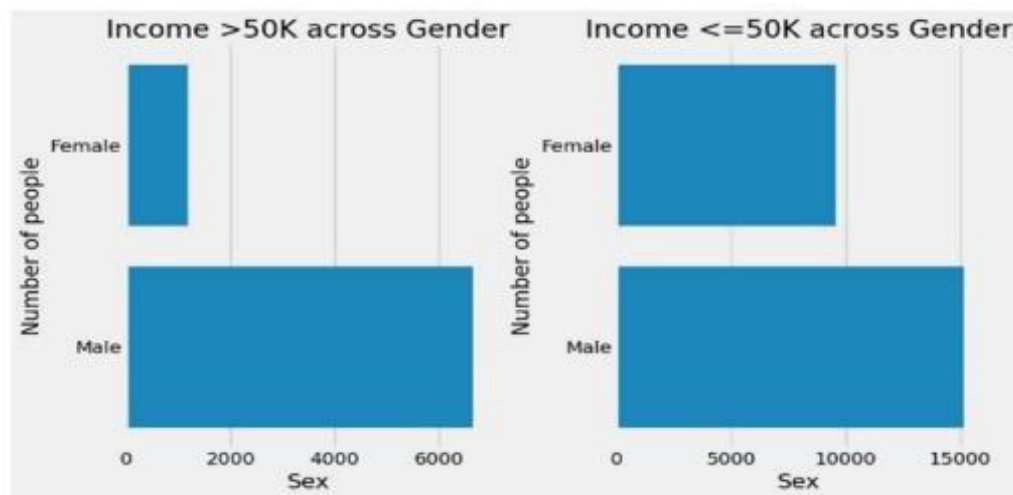


Figure 2 - Distribution of Education Level

**Distribution of Gender:** The genders were separated by a bar plot. The dataset is a male dominated dataset. More than half of the population is Male.

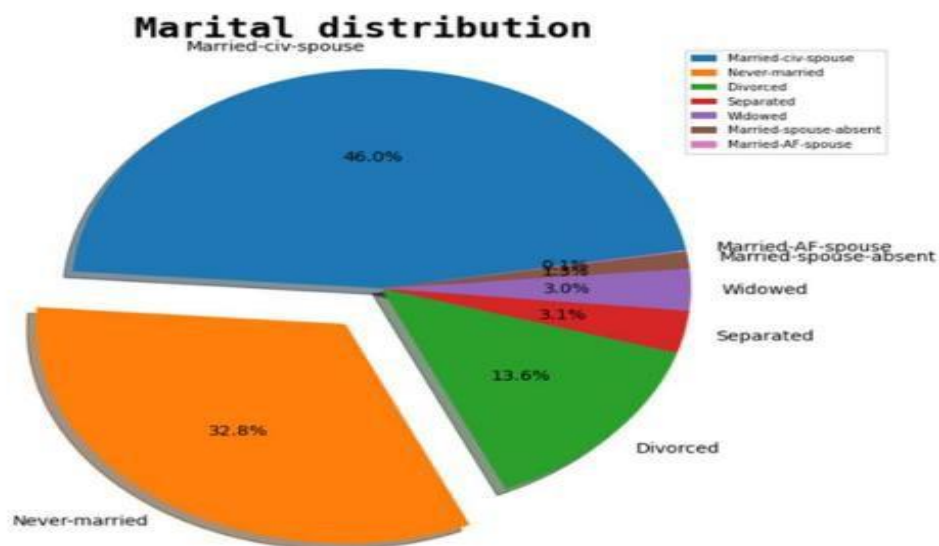


**Distribution of Income across gender:**



*Figure 4 - Distribution of Income across Genders*

**Distribution of Marital Status:** We can understand the distribution of Marital status of the dataset with the help of a pie chart.

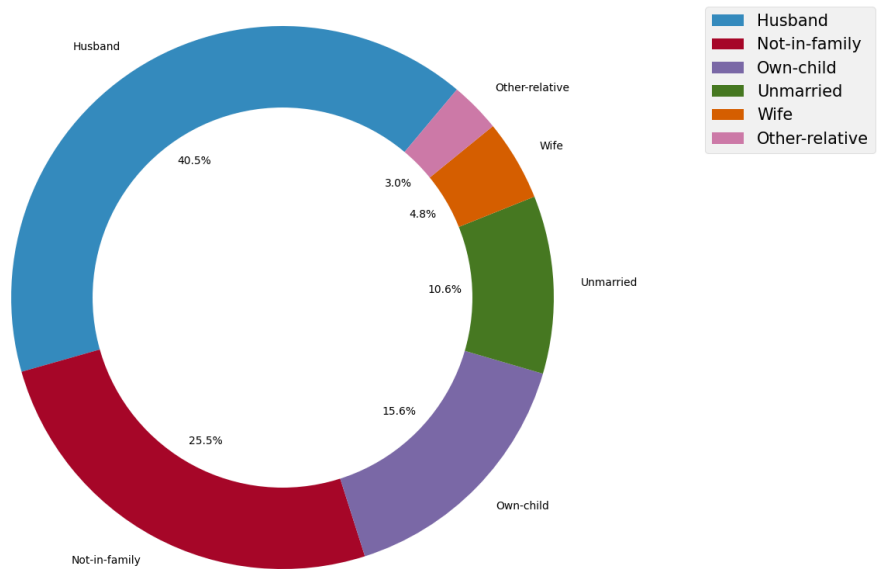


*Figure 5 - Distribution of Marital Status*

The top runners for this dataset are married to a civilian and then people who are never

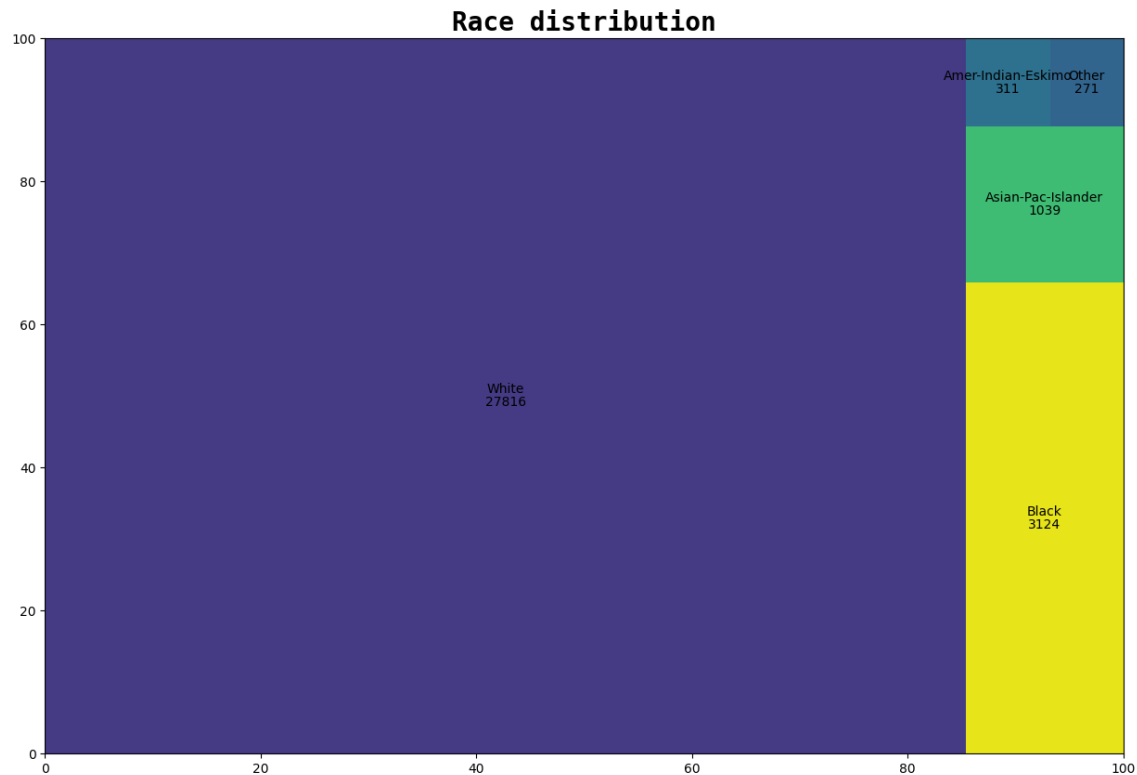
married and then people who are divorced. To further understand this aspect of the dataset we analyse the relationship of the dataset.

### Relationship distribution



Most of the people in the dataset are husbands which goes along with the previous bar chart of married to a civilian and then comes people who are not in a family which goes along with the previous pie chart of never married and divorced.

**Understanding distribution of race:** A tree map was plotted to understand the distribution of race.



The dataset population is dominated by White people. Other races which are present are listed in decreasing order as Black, Asian Pacific Islander, American Indians and Other.

**Understanding Correlation using a Heatmap:** A heatmap is used to understand correlation between different columns in a dataset. The following heatmap shows the correlation between different columns in our dataset. As the color becomes lighter the correlation between 2 variables increases.



Figure 8 - Heatmap of the dataset.

## Missing Values

### Data cleaning using the Interquartile Range (IQR) method –

Data cleaning using the Interquartile Range (IQR) method involves calculating the range between the first and third quartiles of numerical variables to identify outliers, typically defined as data points falling below  $Q1 - 1.5 * IQR$  or above  $Q3 + 1.5 * IQR$ . Once outliers are identified, they can be handled by removal, replacement with more appropriate values like the median or mean, or transformation techniques. Iterative steps may be necessary to ensure thorough cleaning, followed by validation to assess the impact of cleaning on the dataset's integrity. This method offers a systematic approach to improving data quality and reliability for subsequent analysis and modeling tasks. dataset had many missing values. All the missing values were in the form of “?”. The dataset had 4262 missing values distributed across 2400 rows. This accounted for 7.38% of the total dataset. Since the number wasn't so big it made sense to delete the rows having missing values without having a significant impact on the overall data integrity of the dataset. So, all the rows with missing values were eliminated from the dataset.



## Model Exploration:

- **Logistic Regression:** Think of logistic regression as a basic tool for categorizing things into two groups. It's akin to drawing a straight line on a graph to separate items into two distinct categories. However, when dealing with more complex situations beyond binary classification, logistic regression might not yield optimal results due to its simplicity.
- **Decision Trees:** Imagine making decisions by following a tree-like structure where each choice leads to more choices. Decision trees operate similarly. They excel at handling various types of information and uncovering connections between them. Nonetheless, they can become overly intricate and prone to errors in certain cases.
- **Random Forest:** Random forest operates like consulting multiple trees for advice and then aggregating the most popular responses. This approach is beneficial as it integrates diverse perspectives, often resulting in higher accuracy compared to individual trees. Additionally, random forest is less susceptible to errors because it considers the overall picture.
- **Support Vector Machines (SVM):** SVM involves determining the optimal way to draw a line between different groups of items on a graph. It excels at segregating mixed-up elements, particularly in scenarios involving multiple dimensions. However, SVM may require considerable time to identify the best line, especially with extensive datasets.
- **Gradient Boosting Machines (GBM):** GBM can be likened to assembling a team of experts, each tasked with rectifying the mistakes of their predecessors. They collaborate to arrive at the most informed decision. GBM is effective at achieving accuracy, even in complex scenarios.
- **Neural Networks:** Neural networks function as highly intelligent systems that learn from examples. They excel at identifying patterns across various datasets, such as images or text. Tailored neural network architectures cater to specific data types, rendering them potent tools for analysis.

### **Model Selection:**

Upon observing a distribution where approximately twenty-four percent of entries were labeled as >50k and seventy-six percent as ≤50k, we established a baseline by predicting the majority label ≤50k for each item. Given the dataset's size of 43,000 observations, potential viable models to explore include Gaussian Naive Bayes, Logistic Regression, Random Forest and Decision Trees. However, determining the best-performing model necessitates experimentation and comprehensive evaluation using a validation set.

### **Model Implementation and Results**

To analyze the patterns in this dataset 6 algorithms have been used on the training dataset and then tested on the test dataset. The models were –

#### **1.Logistic Regression:**

Logistic Regression works by estimating the probability that a given input belongs to a class using the logistic function, also known as the sigmoid function. The logistic function maps any input to a value between 0 and 1. If the probability is above a certain threshold (typically 0.5), the model predicts the default class; otherwise, it predicts the other class.

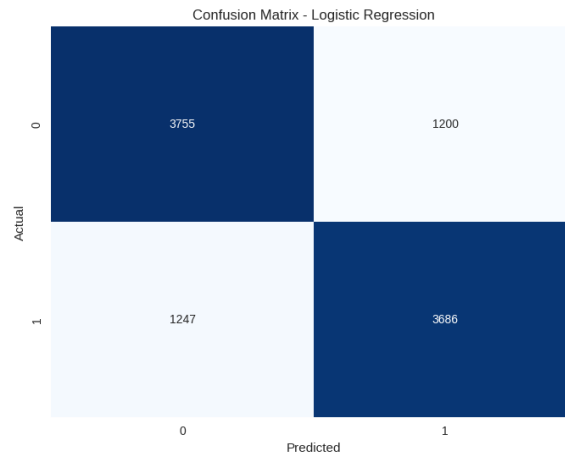
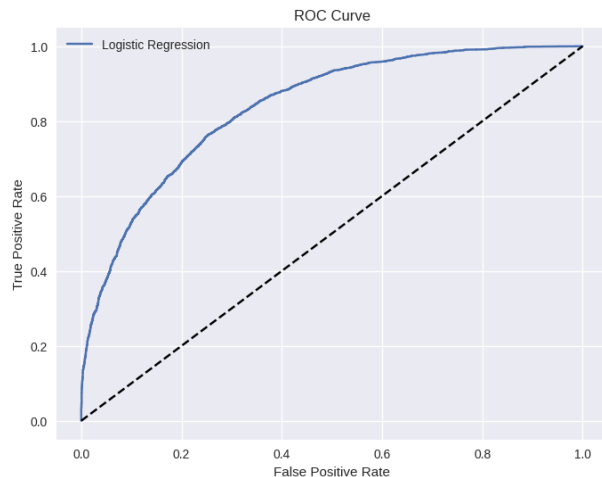
Advantages:

1. Simple and Interpretable
2. Efficient for Small Datasets
3. Low risk of Overfitting

Disadvantages:

- 1.Limited to Linear Decision Boundaries
- 2.Sensitivity to Outliers
- 3.Assumption of Independence of Observations

Implementation and Results – The following ROC curve and Confusion matrix were obtained after running the abovementioned model.



The Logistic Regression model exhibits balanced performance with an accuracy of 75.25%. It achieves a precision of 75.44%, indicating its ability to correctly classify positive instances. The recall score of 74.72% suggests that it effectively identifies most of the actual positive instances. The F1-score, a harmonic mean of precision and recall, is 75.08%, demonstrating a good balance between precision and recall. The ROC AUC score of 75.25% indicates the model's capability to discriminate between positive and negative classes.

## 2.Support Vector Classifier

SVC works by finding the hyperplane that maximizes the margin between the classes. It selects the support vectors, which are the data points closest to the hyperplane. SVC then makes predictions by classifying new data points based on which side of the hyperplane they fall on.

### Advantages

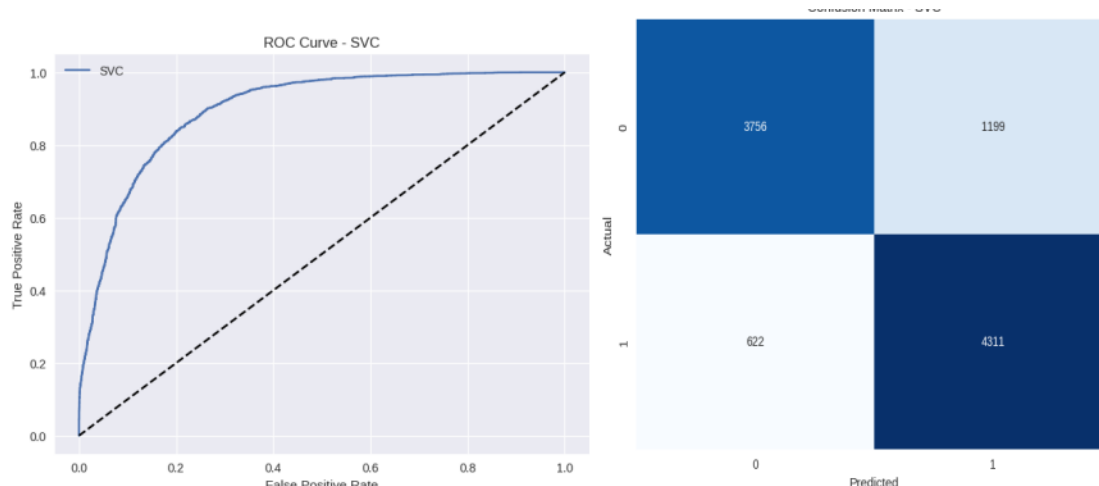
1. Effective in High-Dimensional Spaces
2. Robust to Overfitting
3. Versatile Kernels

### Disadvantages:

1. Computationally Intensive
2. Sensitivity to Noise
3. Difficulty in Interpreting Results

### Implementation and Results:

After running the abovementioned model, the following were obtained:



The Support Vector Classifier demonstrates robust performance with an accuracy of 81.58%. It achieves a precision of 78.24%, indicating a high proportion of correctly classified positive instances. With a recall of 87.39%, the model effectively captures most actual positive instances. The F1-score of 82.56% suggests a balanced trade-off between precision and recall. The ROC AUC score of 81.60% indicates good discrimination ability between positive and negative classes.

### 3. Naïve Bayes Classifier

Naive Bayes Classifier calculates the probability of a given sample belonging to each class based on the feature values. It assumes that the presence (or absence) of a particular feature is independent of the presence (or absence) of any other feature. Despite this naive assumption, Naive Bayes often performs well in practice, especially in text classification and spam filtering tasks.

Advantages:

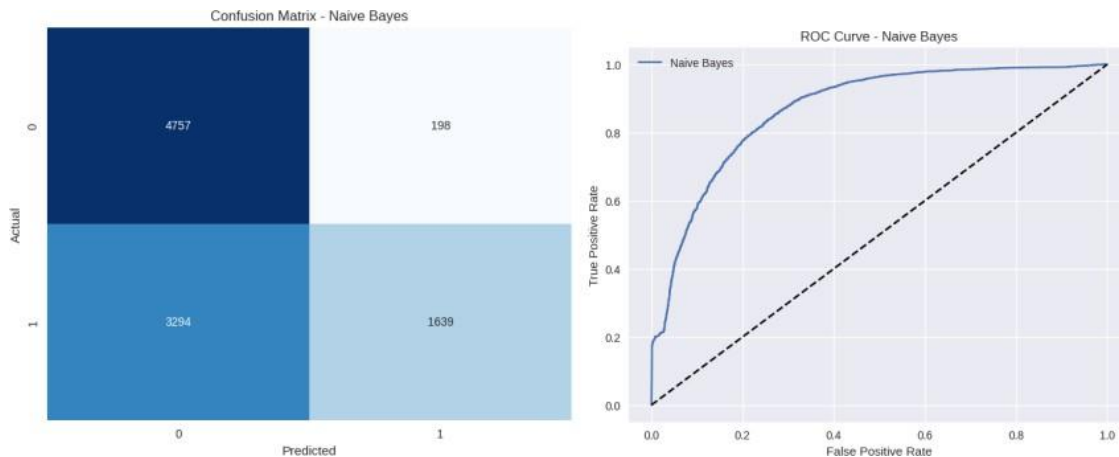
1. Simple and Fast
2. Works Well with High-Dimensional Data
3. Handles Missing Data

Disadvantages:

1. Assumption of Feature Independence
2. Sensitivity to Input Data Quality
3. Limited Expressiveness

Implementation and Results –

The following ROC curve and Confusion matrix were obtained after running the above-mentioned model.



The Naive Bayes Classifier achieves an accuracy of 64.68%, indicating moderate performance. Despite a high precision of 89.22%, suggesting a high proportion of true positive predictions, its recall score of 33.23% indicates the model misses a significant number of actual positive instances. Consequently, the F1-score of 48.42% reflects the imbalance between precision and recall. The ROC AUC score of 64.61% suggests fair discrimination ability between positive and negative classes.

#### 4. Decision Tree Classifier

Decision Tree Classifier builds a tree-like structure where each internal node represents a feature, each branch represents a decision based on that feature, and each leaf node represents a class label. The tree is constructed recursively by splitting the dataset into subsets based on the feature that best separates the classes at each node, typically using metrics like Gini impurity or information gain.

Advantages:

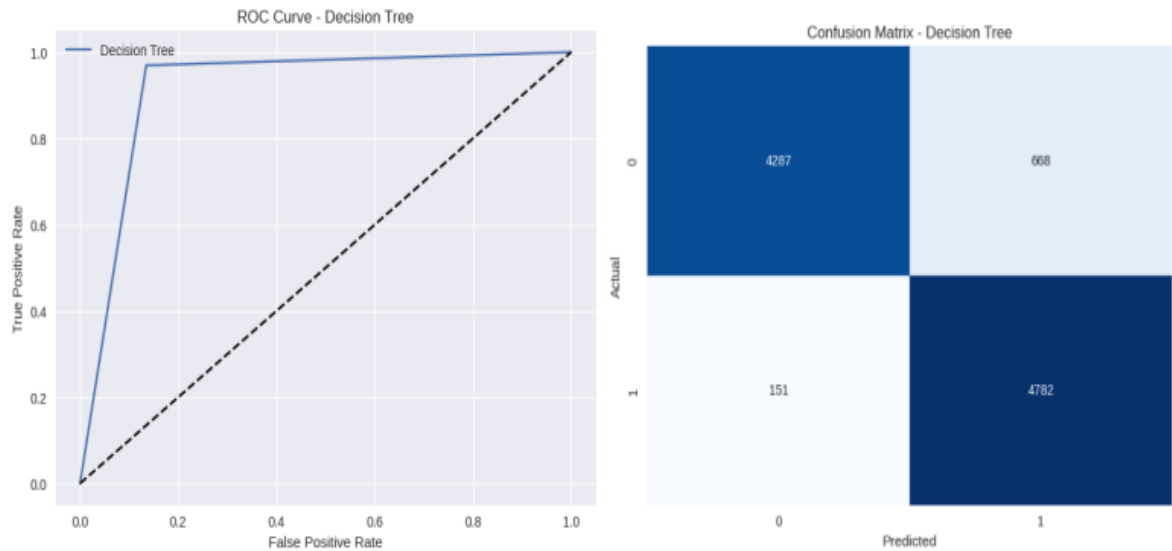
1. Interpretability
2. Handles Non-linear Relationships
3. Robust to feature

Disadvantages:

1. Overfitting
2. Instability
3. Bias Towards Dominant Classes

Implementation and Results –

The following ROC curve and Confusion matrix were obtained after running the abovementioned model.



The Decision Tree Classifier exhibits excellent performance with an accuracy of 91.42%. It achieves a precision of 87.28%, indicating a high proportion of correctly classified positive instances. With a recall of 96.94%, the model effectively captures almost all actual positive instances. The high F1-score of 91.86% indicates a strong balance between precision and recall. The ROC AUC score of 91.44% signifies outstanding discrimination ability between positive and negative classes.

## 5. Random Forest Classifier

Description: Random Forest Classifier builds a collection of decision trees by randomly selecting subsets of the training data and features for each tree. During prediction, each tree "votes" for the most popular class, and the class with the most votes is chosen as the final prediction.

Advantages:

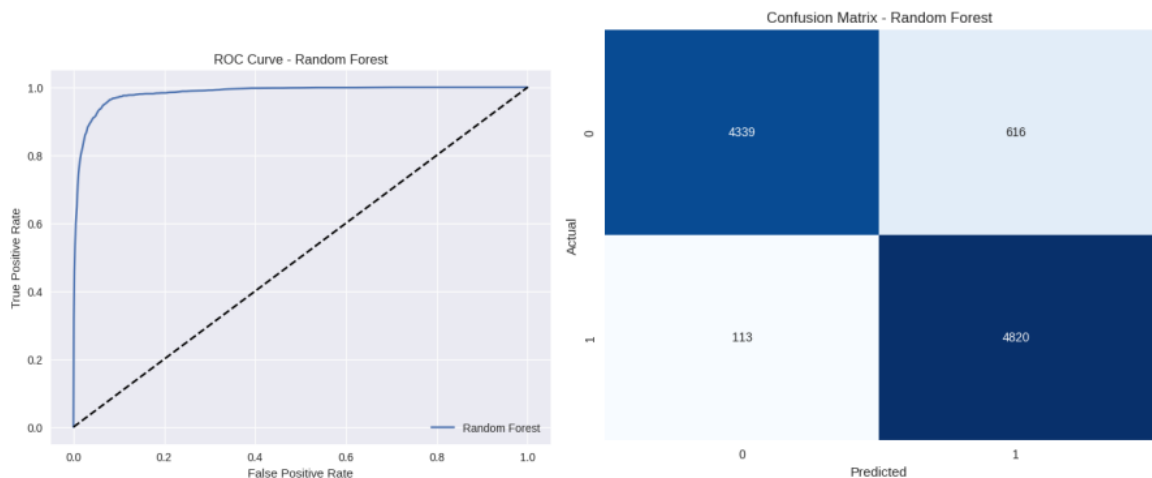
1. High Accuracy
2. Robust to Overfitting

Disadvantages:

1. Computationally Expensive
2. Less Interpretability
3. Biased Towards Dominant Classes

Implementation and Results:

The following ROC curve and Confusion matrix were obtained after running the above-mentioned model.



The Random Forest Classifier showcases outstanding performance with an accuracy of 92.58%. It achieves a precision of 88.63%, indicating a high proportion of true positive predictions among all positive predictions. With a recall of 97.65%, the model effectively captures almost all actual positive instances. The high F1-score of 92.92% indicates a strong balance between precision and recall. The ROC AUC score of 92.59% signifies exceptional discrimination ability between positive and negative classes.

Performance Evaluation Summary:

The following table summarizes the performance metrics across different models that were run. In summary, while all models demonstrate varying levels of performance, the Decision Tree and Random Forest Classifiers outperform others, showing robustness in accurately classifying positive instances while effectively capturing actual positive instances. These models would be preferable for tasks where precision and recall are both crucial, such as medical diagnosis or fraud detection.

	Model	Accuracy	Precision	Recall	F1-score	ROC AUC
0	Logistic Regression	75.252832	75.440033	74.721265	75.078929	75.251652
1	K-Nearest Neighbors	83.424353	79.431737	90.107440	84.433469	83.439189
2	Support Vector Classifier	81.583738	78.239564	87.391040	82.562482	81.596630
3	Naive Bayes Classifier	64.684466	89.221557	33.225218	48.419498	64.614627
4	Decision Tree Classifier	91.717233	87.743119	96.938982	92.112106	91.728825
5	Random Forest Classifier	92.627427	88.668138	97.709305	92.969428	92.638709


### Hyper Parameter Tuning:

Hyperparameter tuning is the process of selecting the optimal set of hyperparameters for a machine learning algorithm. Hyperparameters are parameters that are set prior to the training process and cannot be directly learned from the data. They govern the behavior of the learning algorithm and significantly affect its performance.

This code is performing hyperparameter tuning for a RandomForestClassifier using RandomizedSearchCV in scikit-learn. It searches for the optimal combination of hyperparameters such as the number of trees (`n_estimators`) and the maximum depth of the trees (`max_depth`) by sampling from predefined ranges. The aim is to find the best set of hyperparameters that maximize the model's performance, as evaluated by cross-validation. Finally, it evaluates the performance of the best model on a test set by computing accuracy and F1 score.

Output:

```

 Best Score: 0.913152372610526
Best Parameters: {'n_estimators': 50, 'max_depth': 80}
Accuracy: 0.926779935275081
F1 Score: 0.9300753332045587

```

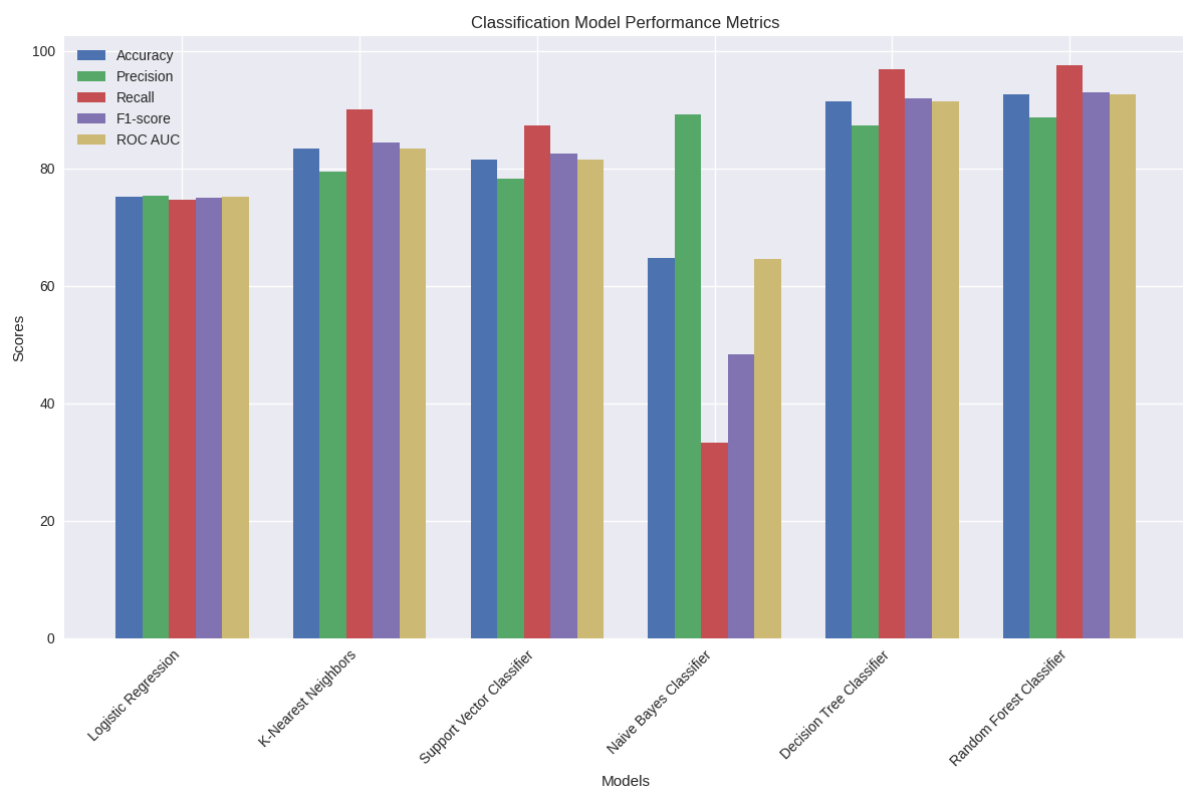
The RandomizedSearchCV process identified the best combination of hyperparameters for the Random Forest Classifier model.



- The best combination of hyperparameters found was {'n\_estimators': 50, 'max\_depth': 80}.
- After training the model with these optimal hyperparameters, it achieved an accuracy of approximately 92.68% on the test set.
- Additionally, the F1 score, which considers both precision and recall, was approximately 93.01% on the test set.

Overall, the Random Forest Classifier model, after hyperparameter tuning, demonstrates strong performance in terms of both accuracy and F1 score, indicating its effectiveness in making predictions on the given dataset.

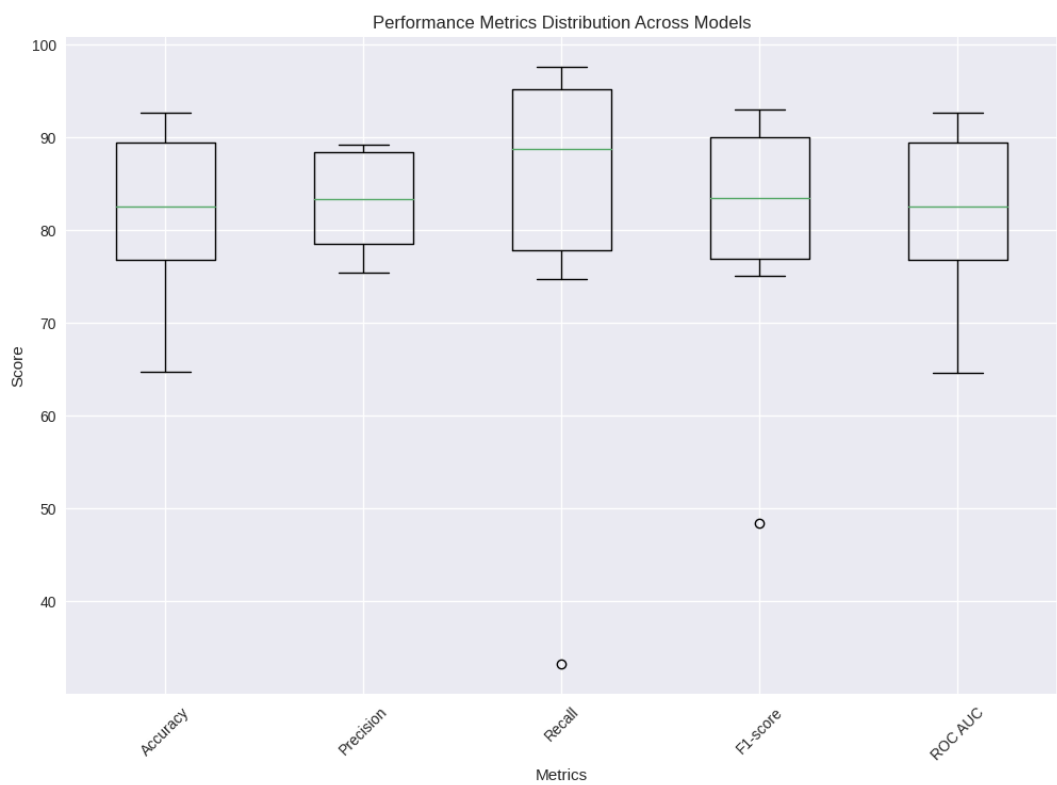
### Classification Model Performance:



The grouped bar chart illustrates the performance metrics (accuracy, precision, recall, F1-score, and ROC AUC) across six different classification models: Logistic Regression, K-Nearest Neighbors, Support Vector Classifier, Naive Bayes Classifier, Decision Tree Classifier, and Random Forest Classifier. Among the models, Decision Tree and Random Forest classifiers stand out with notably high scores across all metrics, indicating robust performance.

However, the Naive Bayes Classifier falls short in recall and F1-score compared to other models. This visualization offers a comparative analysis, enabling stakeholders to discern the strengths and weaknesses of each model across multiple evaluation criteria in a concise and accessible manner.

**Performance Metrics Distribution Across Models:**



The box plots illustrate the distribution of performance metrics—accuracy, precision, recall, F1-score, and ROC AUC—across six classification models: Logistic Regression, K-Nearest Neighbors, Support Vector Classifier, Naive Bayes Classifier, Decision Tree Classifier, and Random Forest Classifier. Each box plot shows the median (line within the box), interquartile range (box), and overall spread of scores (whiskers) for a particular metric. From the visualizations, we can observe the range and variability of performance among the models. For instance, while the Decision Tree and Random Forest classifiers generally exhibit high median scores across all metrics, the Naive Bayes Classifier shows a wider spread of scores, particularly in recall and F1-score, indicating greater variability in its performance.

## **Conclusion**

The Adult Income Prediction project aims to tackle unfair financial patterns by analyzing demographic variables like age, education, occupation, and marital status. Leveraging the Adult Census Income dataset, we've gained valuable insights into socio-economic factors influencing income levels. Through data visualization and analysis, we've identified significant trends and correlations within the dataset. Cleaning techniques like the Interquartile Range method ensured data integrity.

Our exploration of six classification algorithms revealed that Decision Tree and Random Forest classifiers perform best, accurately predicting income levels. Hyperparameter tuning further improved the Random Forest Classifier's performance.

Future work could include exploring feature engineering, ensemble methods, deep learning, fairness analysis, temporal analysis, and integrating external data for richer insights.

In summary, while our analysis provides valuable insights into income prediction and combating economic inequalities, ongoing research and refinement are needed to achieve our goal of promoting fairer financial opportunities for all.

## **Future Work**

**Feature Engineering:** Exploring feature interactions and creating new features from existing ones could enhance predictive power.

**Ensemble Methods:** Experimenting with stacking or boosting techniques could further improve model performance by combining predictions from multiple models.

**Deep Learning:** Investigating deep learning models, such as neural networks, may capture complex relationships in the data and improve predictive accuracy.

**Fairness and Bias Analysis:** Conducting thorough fairness and bias analysis on models is crucial to ensure equitable outcomes across demographic groups.

**Temporal Analysis:** Examining how income patterns have evolved over time could provide insights into societal changes and economic trends.

**External Data Integration:** Integrating external datasets, such as economic indicators, could enrich analysis and enhance predictive capabilities.

In summary, continuing research and refinement of methodologies are essential to achieve the project's goal of promoting a more equitable distribution of financial achievement.