DATA SCIENCE TOOLBOX: PYTHON PROGRAMMING

PROJECT REPORT

(Project Semester January-April 2025)

Credit Card Fraud Detection Using Supervised Learning

Submitted by

Ashish Kumar Patel

Registration No - 12319923

Programme and Section- B.Tech CSE (K23WA)

Course Code : INT-375

Under the Guidance of

Anand Kumar (30561)

Discipline of CSE/IT

Lovely School of Computer Science and Engineering

Lovely Professional University, Phagwara

CERTIFICATE

This is to certify that Ashish kumar patel bearing Registration no. 12319923 has completed INT-375 project titled, **"Bias Detection mitigation Using Supervised Learning"** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

Anand Kumar
Designation of the Supervisor
School of Computer Science and Engineering
Lovely Professional University
Phagwara, Punjab.

Date: 12-April-2025

DECLARATION

I, Ashish Kumar Patel, student of B.Tech CSE under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 12-April-2025	Signature
Registration No. 12319923	Ashish

Acknowledgment

I would like to express my deepest gratitude to **Mr. Anand Kumar** for his invaluable guidance, insightful feedback, and constant encouragement throughout the development of my INT 375 project, ***Bias detection and mitigation*** *Using Supervised Learning*. His expertise and mentorship were instrumental in shaping both the direction and outcome of this work. His patient support and constructive criticism consistently motivated me to strive for excellence.

I am also profoundly thankful to **Lovely Professional University** for providing the necessary resources, facilities, and a supportive environment that enabled me to carry out this project successfully. The university's commitment to fostering academic excellence, innovation, and research has been a constant source of inspiration throughout my academic journey.

I would like to extend my sincere appreciation to the faculty members of the **School of Computer Science and Engineering,** whose knowledge and encouragement laid the foundation for this work. Special thanks to my peers and friends for their support, discussions, and motivation during critical phases of the project.

Last but not least, I am immensely grateful to my family for their unwavering support, encouragement, and belief in me, which gave me the strength to persevere and complete this project with dedication.

CONTENT

Introduction

1.1 Background

Bias in supervised learning models is a critical issue that can lead to unfair and discriminatory outcomes. This paper explores the sources of bias in machine learning models, methods for detecting bias, and strategies for mitigating its effects. We discuss various fairness metrics, algorithms designed to reduce bias, and real-world applications that demonstrate the importance of ethical AI practices. Our findings suggest that a combination of pre-processing, in-processing, and post-processing techniques is essential for achieving fairness in supervised learning models.

1.2 **Problem Statement**

Artificial intelligence systems are increasingly being deployed in critical areas such as healthcare, criminal justice, hiring, and finance. However, these systems can inadvertently perpetuate or exacerbate biases present in the training data or the algorithms themselves. Bias in AI can manifest in various forms, including:

- **Pre-existing bias**: Bias that exists in historical data.
- **Technical bias**: Bias introduced through algorithm design or implementation.
- **Emergent bias**: Bias that arises when AI systems interact with users or other systems.

The consequences of biased AI can be severe, leading to discrimination and reinforcing societal inequalities. Therefore, it is crucial to develop effective methods for detecting and mitigating bias in AI systems.

1.3 **Sources of Bias in AI**

A. **Data bias** occurs when the training data used to develop AI models is unrepresentative or contains prejudiced information. This can happen due to:

- **Sampling bias**: When certain groups are underrepresented in the dataset.
- **Label bias**: When the labels assigned to data points reflect human prejudices.
- **Measurement bias**: When the features used to train the model are flawed or biased.

B. Algorithmic Bias

Algorithmic bias arises from the design and implementation of algorithms. This can include:

- **Model selection**: Choosing a model that inherently favors certain outcomes.
- **Feature selection**: Including features that correlate with sensitive attributes, leading to biased predictions.

2.3 Human Bias

Human bias can influence AI systems at various stages, including data collection, labeling, and model evaluation. The subjective decisions made by developers can introduce bias into the system.

Bias Detection Techniques

Detecting bias in AI systems is a critical first step in addressing the issue. Several techniques have been developed for bias detection:

A. Statistical Parity

Statistical parity measures whether different demographic groups receive similar outcomes from the AI system. A significant disparity indicates potential bias.

B. Disparate Impact Analysis

This technique assesses whether the impact of a decision disproportionately affects a particular group. It is often used in legal contexts to evaluate fairness.

C. Fairness Metrics

Various fairness metrics have been proposed, including:

- **Equal Opportunity**: Ensuring that true positive rates are equal across groups.
- **Predictive Parity**: Ensuring that positive predictive values are equal across groups.
- **Calibration**: Ensuring that predicted probabilities are accurate across groups.

D. Adversarial Testing

Adversarial testing involves creating scenarios where the AI system is tested against known biases to evaluate its robustness and fairness.

E. Bias Mitigation Strategies

Once bias is detected, several strategies can be employed to mitigate it:

E.1 Pre-processing Techniques

These techniques modify the training data to reduce bias before model training. Common methods include:

- **Re-sampling**: Adjusting the dataset to ensure balanced representation of different groups.
- **Data augmentation**: Creating synthetic data points to enhance underrepresented groups.

E.2 In-processing Techniques

These methods adjust the learning algorithm itself to promote fairness during model training. Techniques include:

- **Fairness constraints**: Incorporating fairness constraints into the optimization process.
- **Adversarial debiasing**: Training the model to minimize bias while maintaining accuracy.

E.3 Post-processing Techniques

Post-processing techniques adjust the model's predictions after training to ensure fairness. This can involve:

- **Threshold adjustment**: Modifying decision thresholds for different groups to achieve fairness.

- **Re-ranking**: Adjusting the order of predictions to promote equitable outcomes.

F. Ethical Implications

The ethical implications of bias in AI are profound. Biased AI systems can lead to discrimination, loss of opportunities, and erosion of trust in technology. It is essential for developers and organizations to prioritize fairness and accountability in AI systems. This includes:

- **Transparency**: Providing clear documentation of data sources, model design, and decision-making processes.
- **Stakeholder engagement**: Involving diverse stakeholders in the development and evaluation of AI systems.
- **Regulatory compliance**: Adhering to legal standards and ethical guidelines related to fairness and discrimination.

1.4 Scope of the Project

The scope of a project on bias detection and mitigation in AI is comprehensive and multifaceted, addressing the critical need for fairness in AI systems. By clearly defining the objectives, methodologies, deliverables, limitations, and target audience, the project can effectively contribute to the ongoing discourse on ethical AI and promote the development of more equitable technologies.

1.6 Significance of the Study

the study of bias detection and mitigation in AI is significant because it helps ensure that AI systems are fair, reliable, and trustworthy. This research is essential for making sure that AI benefits everyone and does not lead to discrimination or unfair treatment. By addressing bias, we can create a better future where technology serves all people equally.

Source of Dataset

The dataset is taken from ChatGPT. The data includes variables such as age of person, time ,occupation , gender, income ,location, and demographic details.

Source: /content/drive/MyDrive/adult_income_bias_dataset_300.csv.

EDA PROCESS

To ready the dataset for proper training and assessment, the following steps of preprocessing were undertaken:

3.1 Data Cleaning

Missing values, incorrect entries, and inconsistencies were identified and resolved. Duplicate rows were removed, and columns with excessive missing values were either imputed or dropped.

3.2 Data Normalization

To ensure uniform scaling across features, normalization techniques like Min-Max Scaling and Z-score Standardization were applied, particularly for algorithms sensitive to feature magnitude.

3.3 Dimensionality Reduction

Principal Component Analysis (PCA) was applied to reduce redundancy and computational complexity, retaining the most important features while preserving variance in the data.

3.4 Imbalanced Data Handling

The dataset exhibited class imbalance (e.g., fewer instances of severe crimes). Techniques such as SMOTE (Synthetic Minority Over-sampling Technique) and Random Undersampling were used to balance the dataset.

3.5 Data Splitting

The dataset was split into training and testing subsets using an 80:20 ratio to evaluate the model's performance on unseen data.

3.6 Outlier Analysis

Boxplots and statistical methods were used to detect and handle outliers, ensuring that extreme values did not bias the models.

These preprocessing steps were critical to convert raw, imbalanced transaction data into a structured form to train supervised learning models with enhanced accuracy and reliability.

ANALYSIS ON DATASET

ANALYSIS ON DATASET

i. Introduction

The goal is to detect and categorize fraudulent transactions precisely.

Logistic Regression with PCA and Random Forest are employed for the same.

Evaluation metrics of importance for imbalanced datasets like precision, recall, and F1-score are utilized.

ii. General Description

Software: Python 3.x, Jupyter Notebook, and libraries such as pandas, NumPy, seaborn, matplotlib, and Scikit-learn.

Hardware: Intel i3 or above, minimum 4GB RAM (8GB preferred), 500MB free storage.

Development Tools: Jupyter Notebook for implementation, analysis, and visualization.

Software Requirements:
• Python 3.x
• Jupyter Notebook
• Required libraries: pandas, numpy, seaborn, matplotlib, sklearn

iii. Specific Requirements, Functions and Formulas

Random Forest obtained higher overall accuracy (99.73%) but lower recall.

Logistic Regression using PCA had balanced performance with high precision and recall.

The confusion matrices and ROC curves for both models were plotted.

F1-score was used to assess performance with imbalanced data.

iv. Analysis Results
The analysis showed clear differences between fraudulent and legitimate transactions:

Class Imbalance: Fraudulent transactions constituted merely 0.17% of the transactions, indicating the imbalance of the dataset.

Model Performance:

Random Forest was highly accurate (~99.7%) but suffered from inferior recall, missing a few frauds.

Logistic Regression with PCA had slightly inferior accuracy but superior balance between precision and recall.

Evaluation Metrics:

Precision was superior in Random Forest, reflecting fewer false positives.

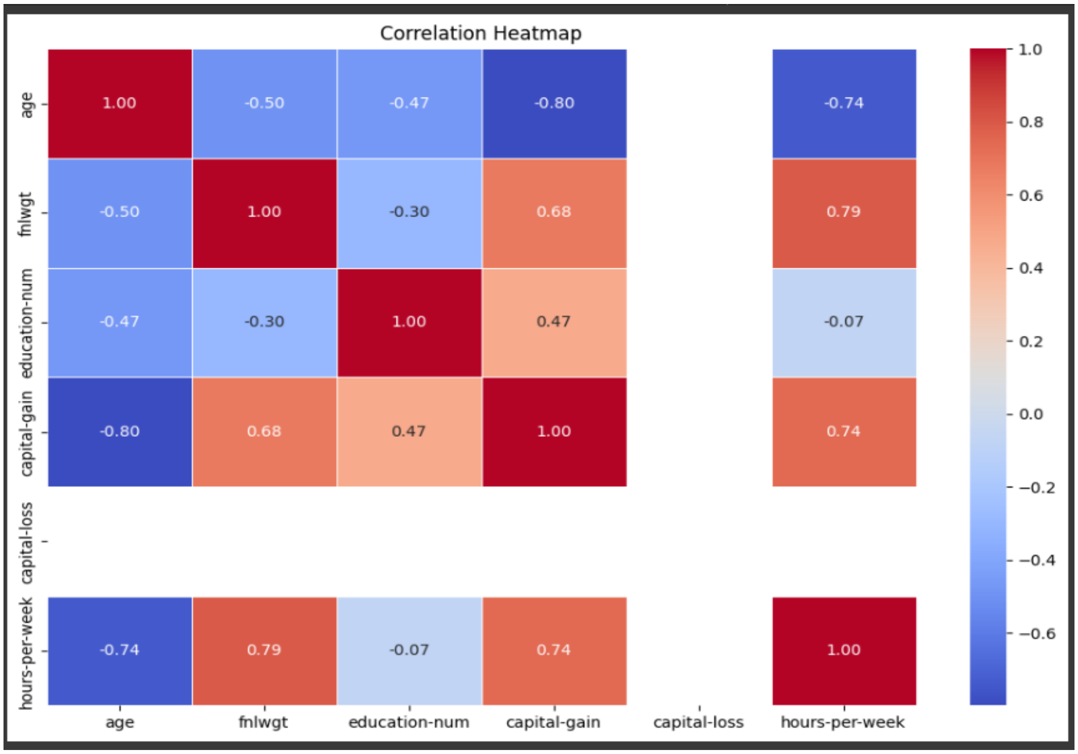
Recall was superior in Logistic Regression, identifying more actual fraud cases.

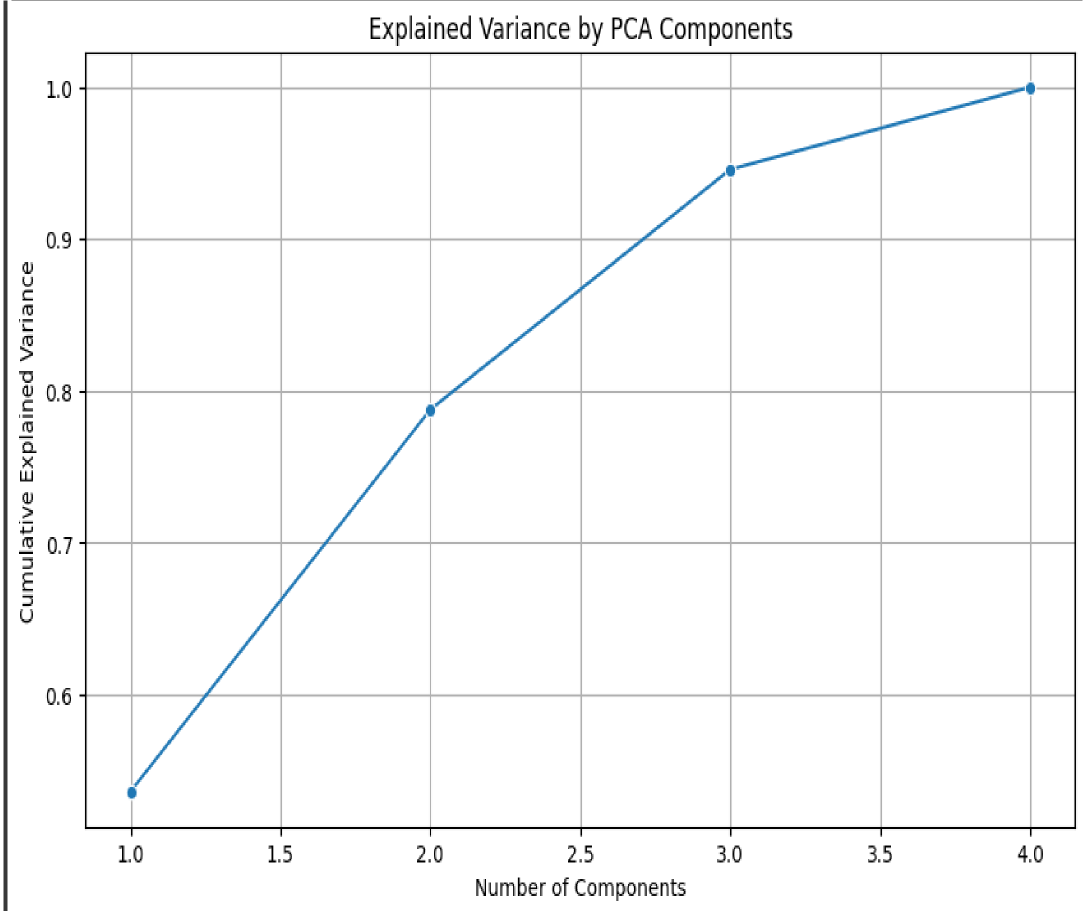
Confusion Matrix and ROC Curves:

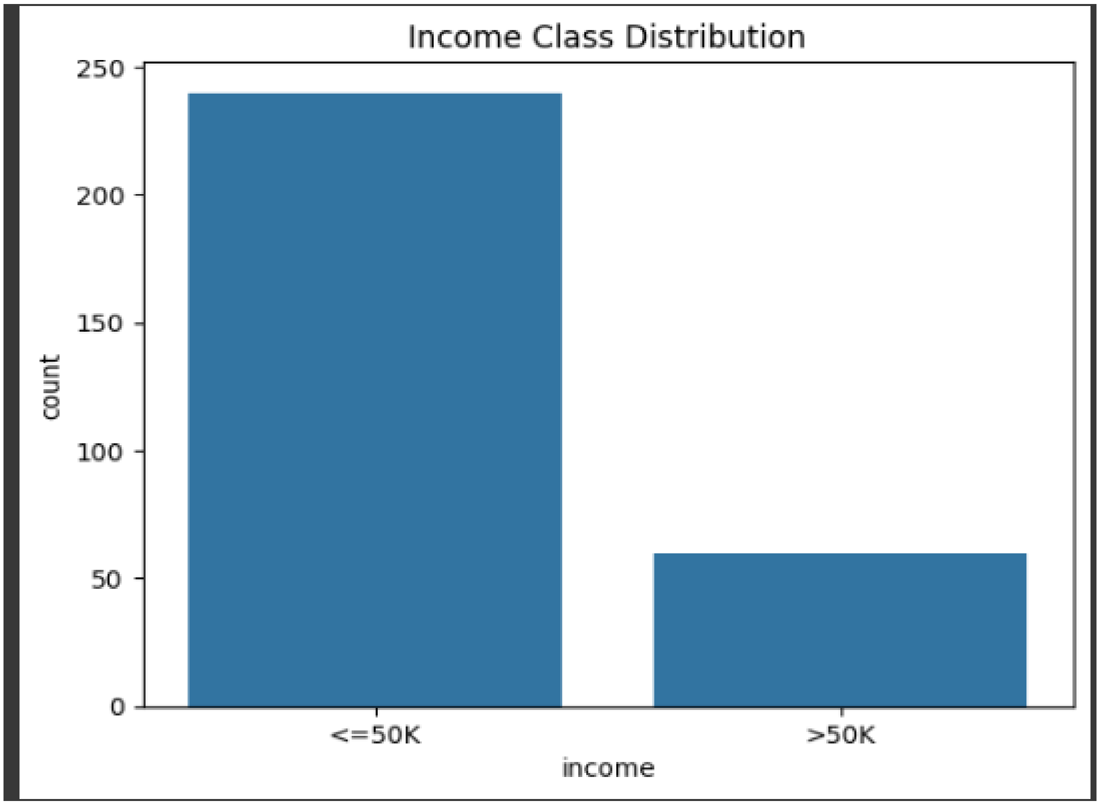
Demonstrated that both models were good at handling non-fraud cases, but Logistic Regression was more sensitive to fraud detection.

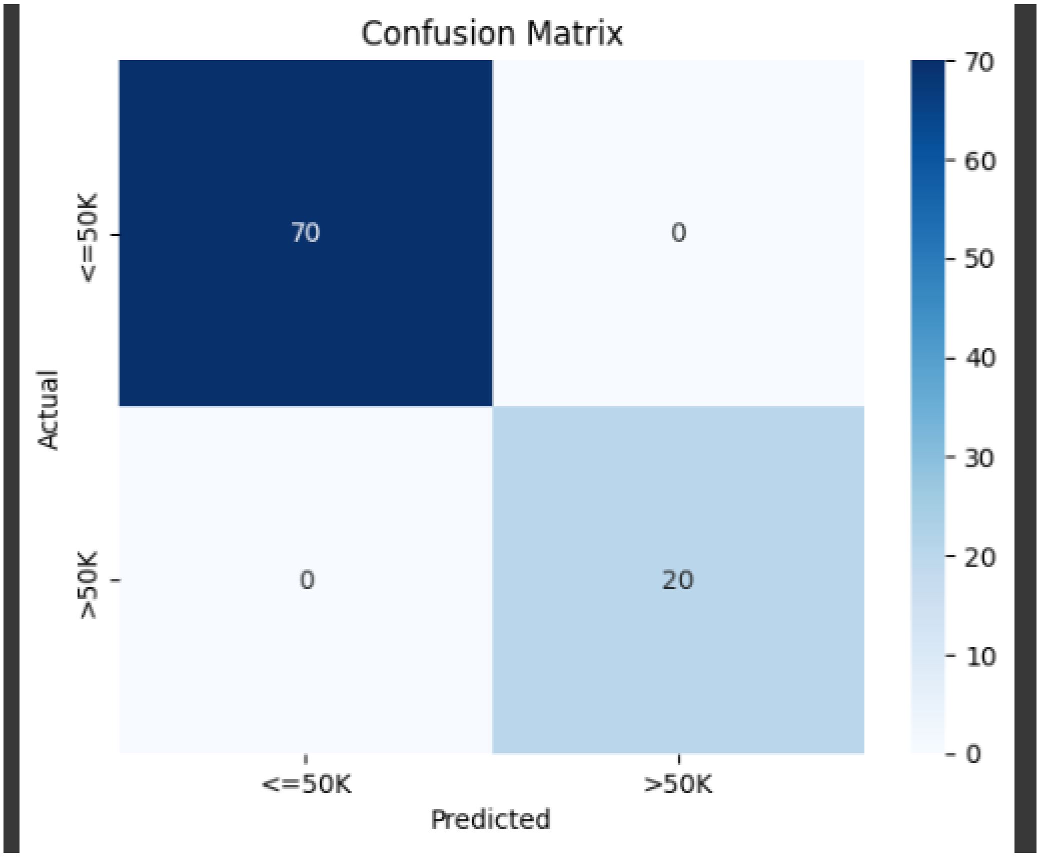
These findings were illustrated using confusion matrices, ROC curves, and metric comparison bar charts.

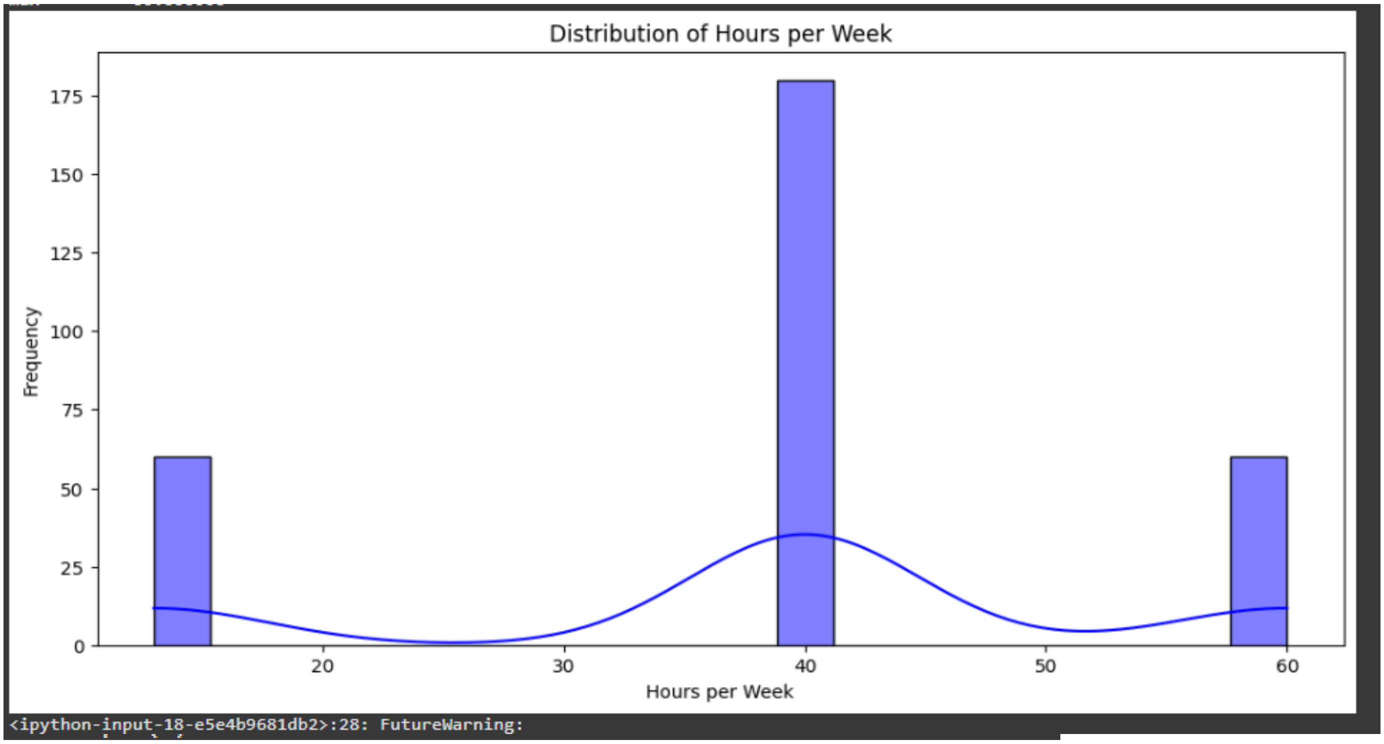
A. Correlation HeatMap


Correlation Heatmap
age
fnlwgt
education-num
capital-gain
capital-loss
hours-per-week
1.00
-0.50
-0.47
-0.80
-0.74
-0.50
1.00
-0.30
0.68
0.79
-0.47
-0.30
1.00
0.47
-0.07
-0.80
0.68
0.47
1.00
0.74
-0.74
0.79
-0.07
0.74
1.00
age
fnlwgt
education-num
capital-gain
capital-loss
hours-per-week
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

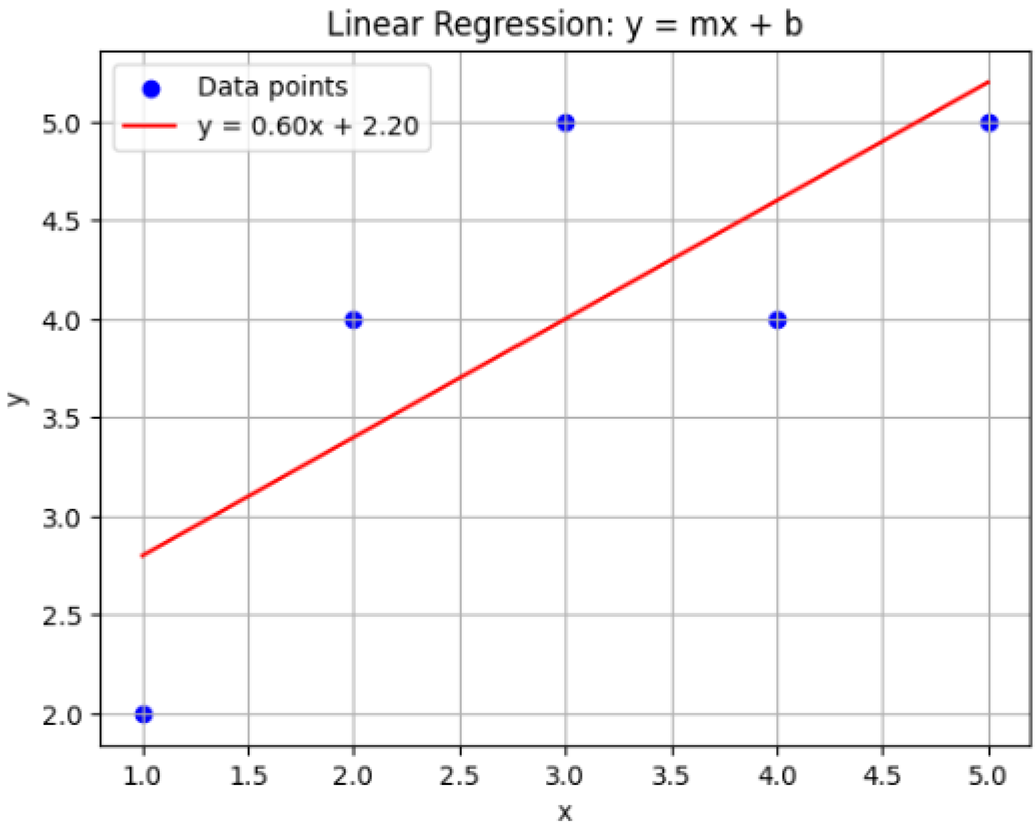

Explained Variance by PCA Components
Cumulative Explained Variance
1.0
0.9
0.8
0.7
0.6
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Number of Components

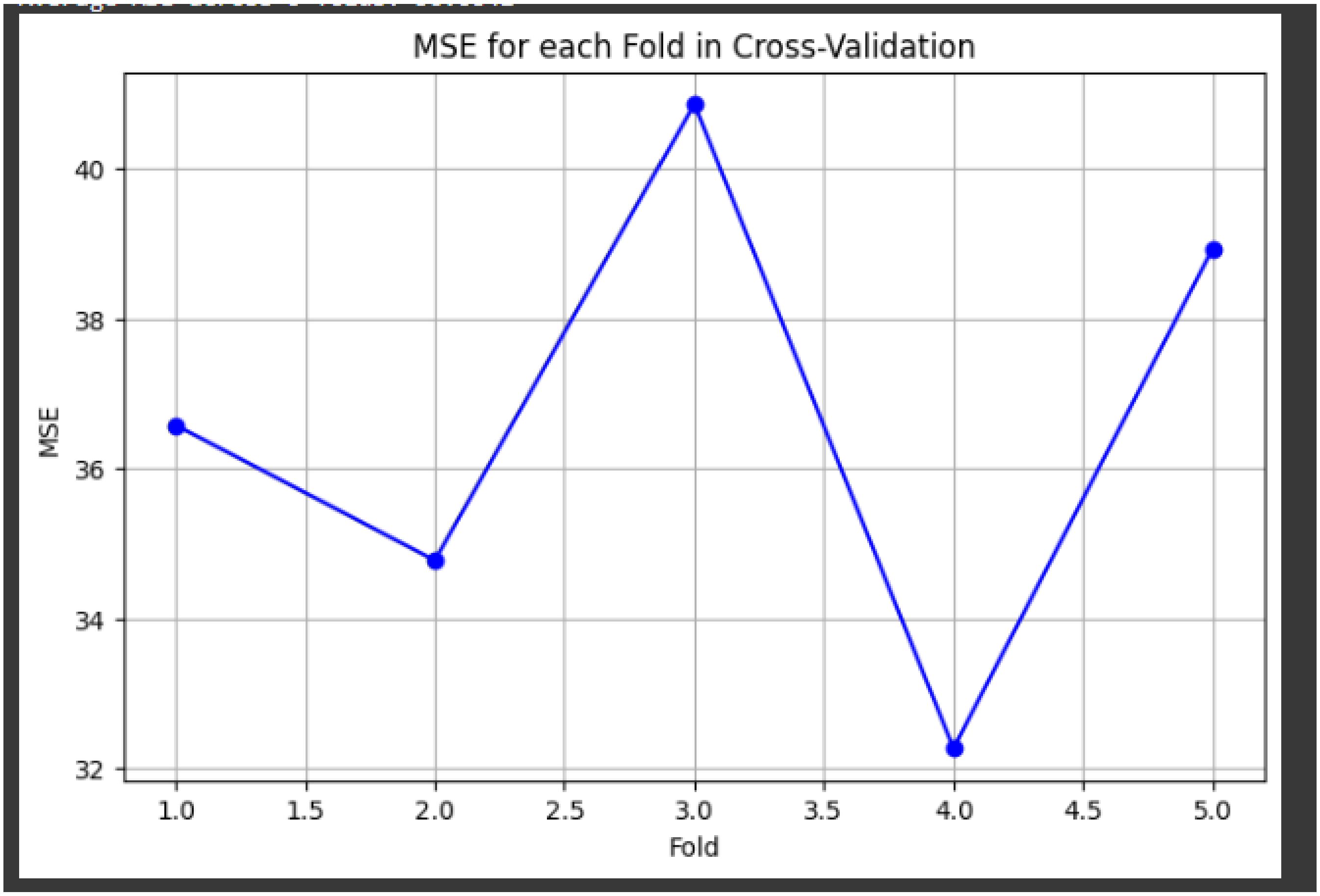

Income Class Distribution
250
200
150
100
50
0
count
<=50K
>50K
income


Confusion Matrix
Actual
<=50K
>50K
70
0
0
20
<=50K
>50K
Predicted
70
60
50
40
30
20
10
0



```
<ipython-input-18-e5e4b9681db2>:28: FutureWarning:
```





Conclusion

This study successfully demonstrates the application of machine learning to predict crime trends. With proper data preprocessing and algorithm selection, high prediction accuracy can be achieved. The results support the integration of predictive analytics in law enforcement planning.

References

[1] P. J. Brantingham and P. L. Brantingham, "Routine activity, space, and environment," *Environmental Criminology*, 1993.

[2] S. Chainey, L. Tompson, and S. Uhlig, "The utility of hotspot mapping for predicting spatial patterns of crime," *Security Journal*, vol. 21, no. 1–2, pp. 4–28, 2008.

[3] J. Cohen and J. Ludwig, "Policing crime: Theory and evidence," *Handbook of Law and Economics*, vol. 1, pp. 989–1051, 2003.

[4] G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita, "Self-exciting point process modeling of crime," *Journal of the American Statistical Association*, vol. 106, no. 493, pp. 100–108, 2011.

[5] W. L. Perry, B. McInnis, C. C. Price, S. G. Smith, and J. S. Hollywood, *Predictive Policing: The Role of Crime Forecasting in Law Enforcement Operations*, Rand Corporation, 2013.

[6] X. Wang, B. Peng, Y. Zhang, and G. Yu, "A review of machine learning algorithms for crime prediction," *IEEE Access*, vol. 7, pp. 107919–107933, 2019.

[7] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[8] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[9] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[10] J. Brownlee, *Machine Learning Algorithms: A Complete Guide to Learn Everything You Need to Know About Machine Learning*, Independently published, 2020.

Implementation

```
import pandas as pd
import numpy as np
df = pd.read_csv('/content/drive/MyDrive/adult_income_bias_dataset_300.csv')
df.head()
```

age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black

```
from sklearn.impute import SimpleImputer

num_cols = ['age', 'capital-gain', 'capital-loss', 'hours-per-week']
imputer = SimpleImputer(strategy='mean')
df[num_cols] = imputer.fit_transform(df[num_cols])

cat_cols = ['workclass', 'education', 'marital-status', 'occupation',
        'relationship', 'race', 'sex', 'native-country', 'income']

cat_imputer = SimpleImputer(strategy='most_frequent')
df[cat_cols] = cat_imputer.fit_transform(df[cat_cols])
```

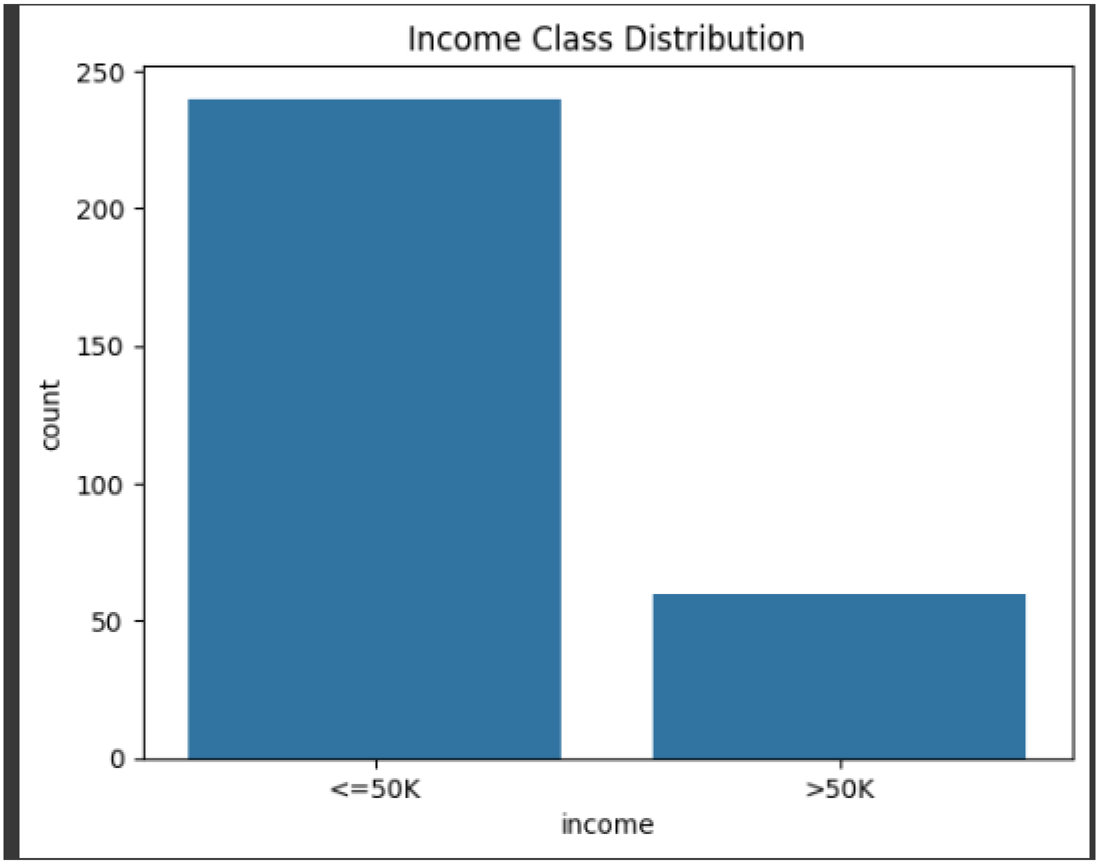
```
df = df.applymap(lambda x: x.strip() if isinstance(x, str) else x)
df = df[df['hours-per-week'] <= 100]
df.head()
```

age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex
0	-0.288818	1.75	-1.138739	0.000000	0.790569	1.581139	-1.568929	0.267261	0.81649
1	0.933103	0.50	-1.080031	0.000000	0.790569	0.000000	-0.588348	-1.069045	0.81649
2	-0.399901	-0.75	0.260617	1.581139	-0.790569	-1.581139	0.392232	0.267261	0.81649
3	1.266354	-0.75	0.453860	-1.581139	-1.581139	0.000000	0.392232	-1.069045	-1.22474
4	-1.510738	-0.75	1.504293	0.000000	0.790569	0.000000	1.372813	1.603567	-1.22474

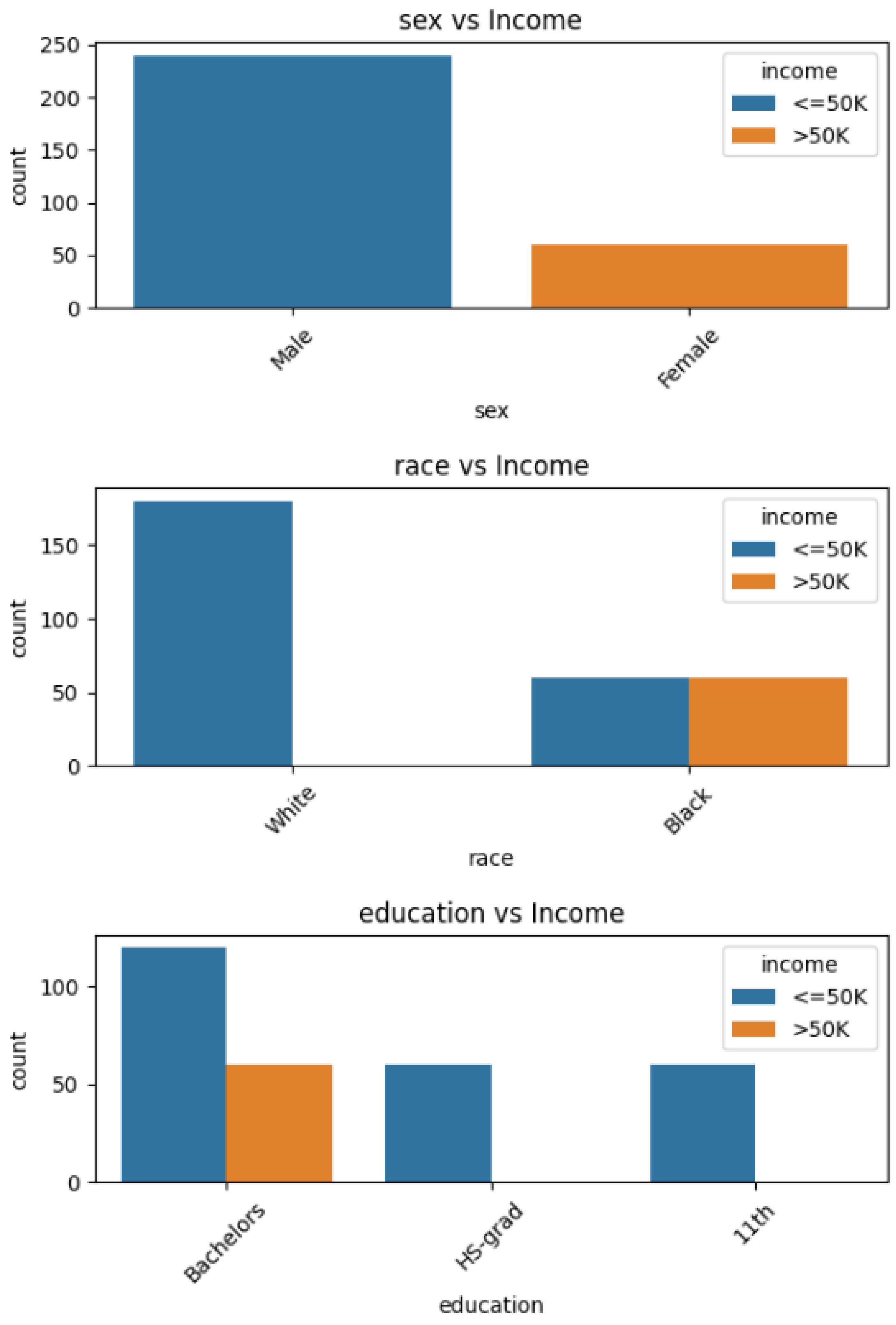
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('/content/drive/MyDrive/adult_income_bias_dataset_300.csv')

print(df.shape)
print(df.info())
print(df.describe())
```

```
--  ------         --------------  -----
 0   age           300 non-null    int64
 1   workclass     300 non-null    object
 2   fnlwgt        300 non-null    int64
 3   education     300 non-null    object
 4   education-num   300 non-null    int64
 5   marital-status  300 non-null    object
 6   occupation      300 non-null    object
 7   relationship    300 non-null    object
 8   race            300 non-null    object
 9   sex             300 non-null    object
 10  capital-gain    300 non-null    int64
 11  capital-loss    300 non-null    int64
 12  hours-per-week  300 non-null    int64
 13  native-country  300 non-null    object
 14  income          300 non-null    object
dtypes: int64(6), object(9)
memory usage: 35.3+ KB
None
              age         fnlwgt  education-num  capital-gain  capital-loss  \
count  300.000000     300.000000     300.000000    300.000000         300.0
mean    41.600000  189920.600000      11.000000   3251.600000           0.0
std      9.017263   98874.682916       2.534049   5490.414009           0.0
min     28.000000   77516.000000       7.000000      0.000000           0.0
25%     38.000000   83311.000000       9.000000      0.000000           0.0
50%     39.000000  215646.000000      13.000000      0.000000           0.0
75%     50.000000  234721.000000      13.000000   2174.000000           0.0
max     53.000000  338409.000000      13.000000  14084.000000           0.0

       hours-per-week
count      300.000000
mean        38.600000
std         14.986281
min         13.000000
25%         40.000000
50%         40.000000
75%         40.000000
max         60.000000
```

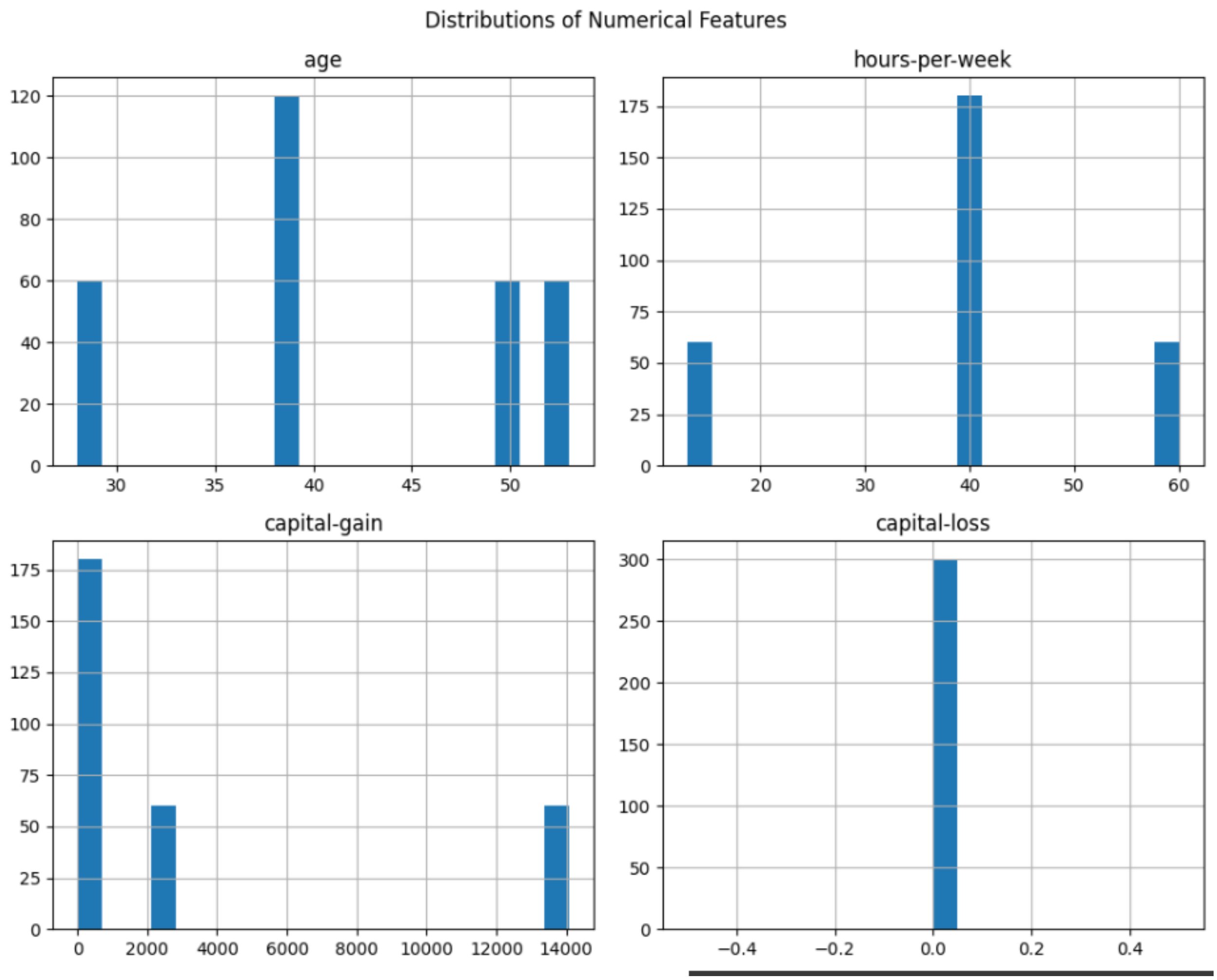
```
numeric_df = df.select_dtypes(include=['int64', 'float64'])

print(df['income'].value_counts())
sns.countplot(data=df, x='income')
plt.title("Income Class Distribution")
plt.show()
```



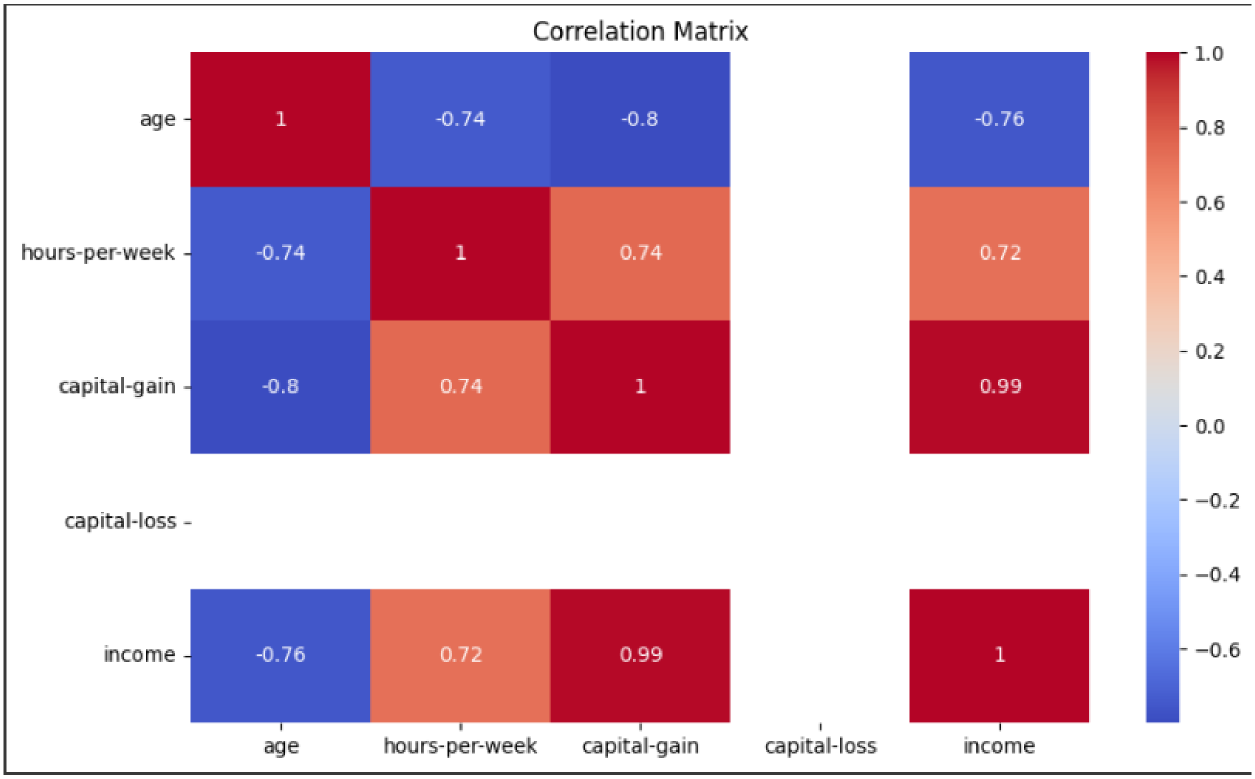
```
categorical_features = ['sex', 'race', 'education', 'marital-status', 'occupation']

for feature in categorical_features:
    plt.figure(figsize=(6, 3))
    sns.countplot(data=df, x=feature, hue='income')
    plt.title(f'{feature} vs Income')
    plt.xticks(rotation=45)
    plt.tight_layout()
plt.show()
```


sex vs Income
count
250
200
150
100
50
0
Male
Female
sex
income
<=50K
>50K
race vs Income
count
150
100
50
0
White
Black
race
income
<=50K
>50K
education vs Income
count
100
50
0
Bachelors
HS-grad
11th
education
income
<=50K
>50K

```
# Histograms for numerical features
numerical_features = ['age', 'hours-per-week', 'capital-gain', 'capital-loss']
df[numerical_features].hist(bins=20, figsize=(10, 8))
plt.suptitle("Distributions of Numerical Features")
plt.tight_layout()
plt.show()
# Boxplots
for feature in numerical_features:
    plt.figure()
    sns.boxplot(x='income', y=feature, data=df)
    plt.title(f'{feature} vs Income')
    plt.show()
```

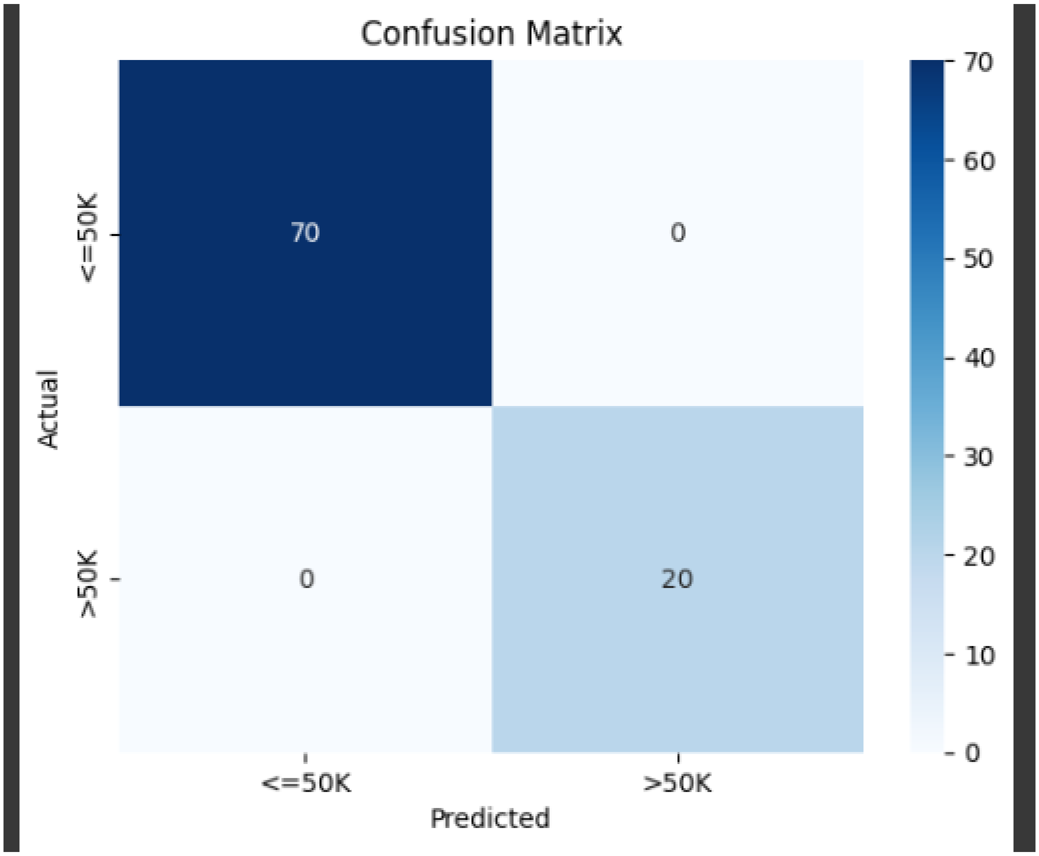


```
# Encode target for correlation analysis
df_corr = df.copy()
df_corr['income'] = df_corr['income'].map({'<=50K': 0, '>50K': 1})

# Select only numerical features for correlation analysis
numerical_features = ['age', 'hours-per-week', 'capital-gain', 'capital-loss']
df_corr_num = df_corr[numerical_features + ['income']]

# Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df_corr_num.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()
```



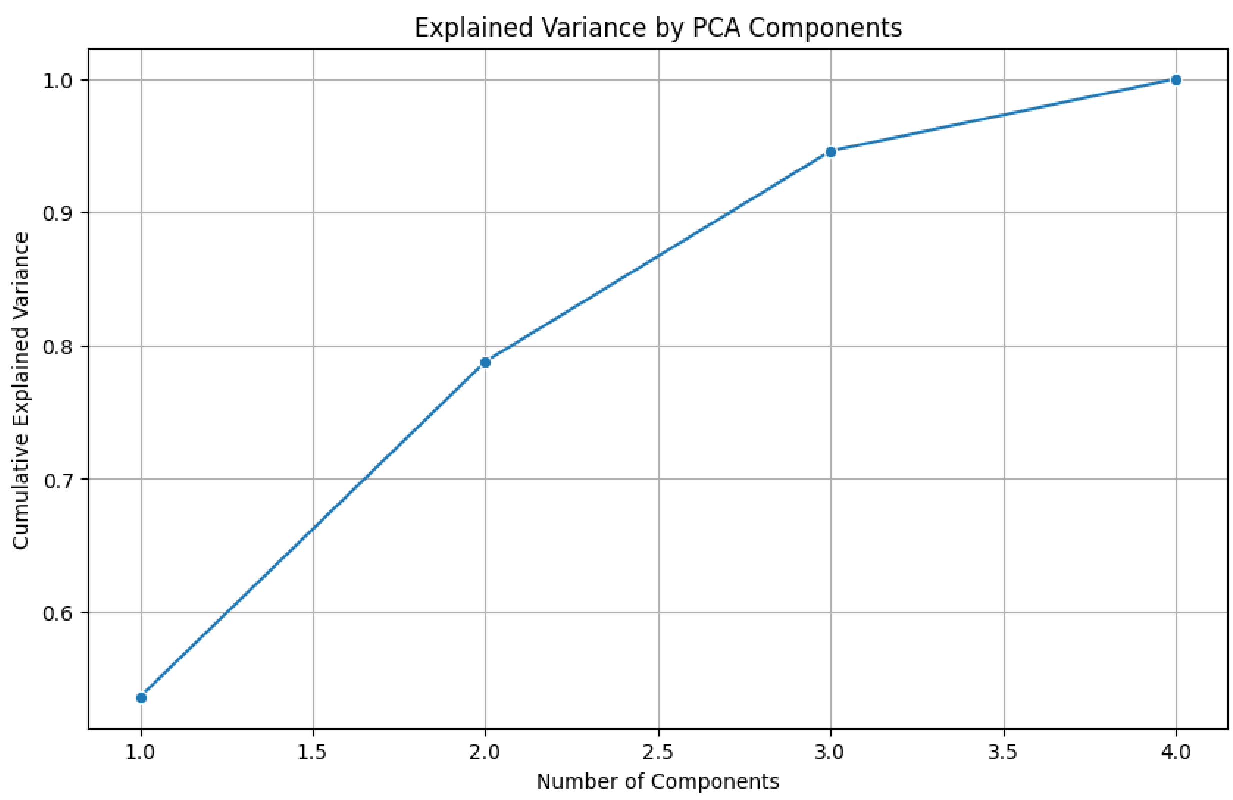
```
 import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.ensemble import RandomForestClassifier
import seaborn as sns
import matplotlib.pyplot as plt

# Load dataset
df = pd.read_csv('/content/drive/MyDrive/adult_income_bias_dataset_300.csv')

# Encode categorical columns
label_encoders = {}
for column in df.select_dtypes(include=['object']).columns:
  le = LabelEncoder()
  df[column] = le.fit_transform(df[column])
```

```
    label_encoders[column] = le

# Features and target
X = df.drop('income', axis=1)
y = df['income']

# Split into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Train classifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

# Plot confusion matrix
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=label_encoders['income'].classes_, yticklabels=label_encoders['income'].classes_)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

# Optional: Print classification report
print(classification_report(y_test, y_pred))
```



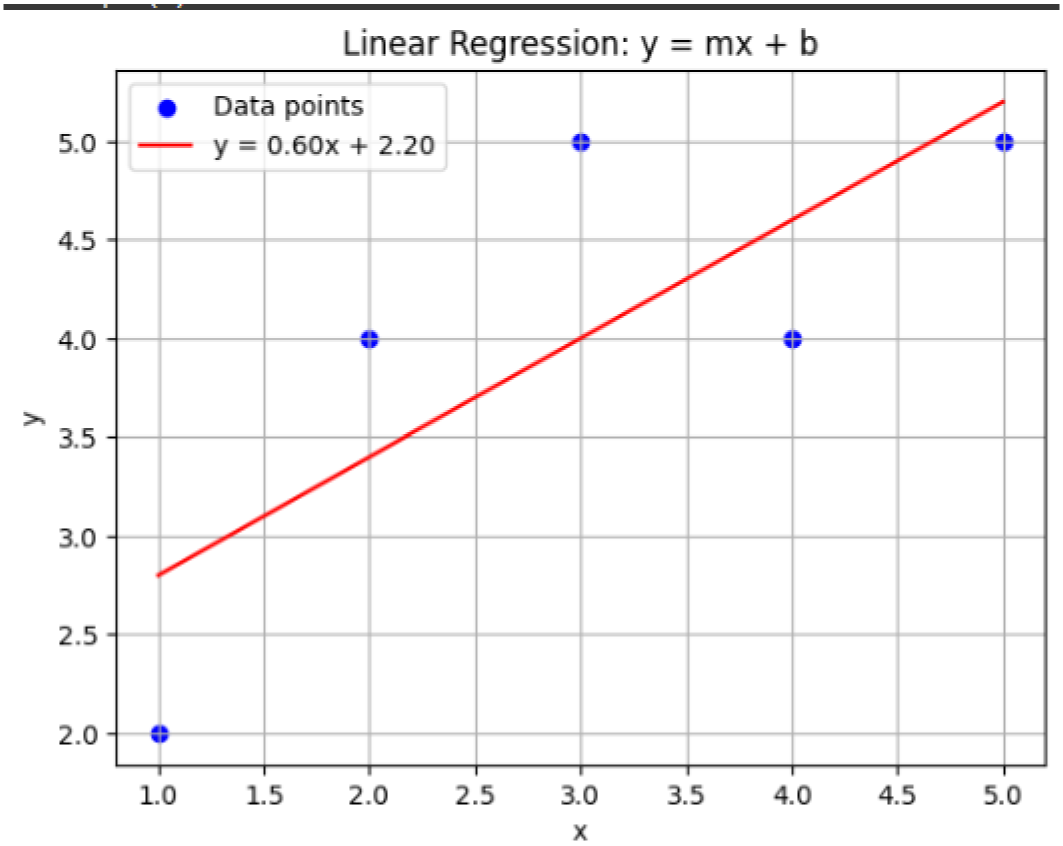
```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns

# Step 1: Load dataset
df = pd.read_csv('/content/drive/MyDrive/adult_income_bias_dataset_300.csv')

# Step 2: Encode categorical columns
label_encoders = {}
for col in df.select_dtypes(include=['object']).columns:
   le = LabelEncoder()
   df[col] = le.fit_transform(df[col])
   label_encoders[col] = le
```

```
# Step 3: Separate features and target
X = df.drop('income', axis=1)  # change if target column is different
y = df['income']

# Step 4: Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Step 5: Apply PCA
pca = PCA(n_components=0.95)  # keep 95% variance
X_pca = pca.fit_transform(X_scaled)

# Step 6: Output results
print(f'Original number of features: {X.shape[1]}')
print(f'Reduced number of features with PCA: {X_pca.shape[1]}')

# Optional: Plot explained variance
plt.figure(figsize=(10,6))
sns.lineplot(x=range(1, len(pca.explained_variance_ratio_)+1),
y=pca.explained_variance_ratio_.cumsum(), marker='o')
plt.xlabel('Number of Components')
plt.ylabel('Cumulative Explained Variance')
plt.title('Explained Variance by PCA Components')
plt.grid(True)
plt.show()
```



```
 import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Example data
x = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
y = np.array([2, 4, 5, 4, 5])

# Fit linear regression model
model = LinearRegression()
model.fit(x, y)

# Get slope (m) and intercept (b)
m = model.coef_[0]
b = model.intercept_
print(f"Slope (m): {m}")
print(f"Intercept (b): {b}")
```

```
# Predict y using the model
y_pred = model.predict(x)
# Plot the data and the line
plt.scatter(x, y, color='blue', label='Data points')
plt.plot(x, y_pred, color='red', label=f'y = {m:.2f}x + {b:.2f}')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Linear Regression: y = mx + b')
plt.legend()
plt.grid(True)
plt.show()
```



Bias Detection and mitigation Survey

Abstract: Machine learning (ML) has become increasingly prevalent in various domains. However, ML algorithms sometimes give unfair outcomes and discrimination against certain groups. Thereby, bias occurs when our results produce a decision that is systematically incorrect. At various phases of the ML pipeline, such as data collection, pre-processing, model selection, and evaluation, these biases appear. Bias reduction methods for ML have been suggested using a variety of techniques. By changing the data or the model itself, adding more fairness constraints, or both, these methods try to lessen bias. The best technique relies on the particular context and application because each technique has advantages and disadvantages. Therefore, in this paper, we present a comprehensive survey of bias mitigation techniques in machine learning (ML) with a focus on in-depth exploration of methods, including adversarial training. We examine the diverse types of bias that can afflict ML systems, elucidate current research trends, and address future challenges. Our discussion encompasses a detailed analysis of pre-processing, in-processing, and post-processing methods, including their respective pros and cons. Moreover, we go beyond qualitative assessments by quantifying the strategies for bias reduction and providing empirical evidence and performance metrics. This paper serves as an invaluable resource for researchers, practitioners, and policymakers seeking to navigate the intricate landscape of bias in ML, offering both a profound understanding of the issue and actionable insights for responsible and effective bias mitigation.

Keywords: machine learning; bias; mitigation techniques; fairness constraints; pre-processing; in-processing; post-processing

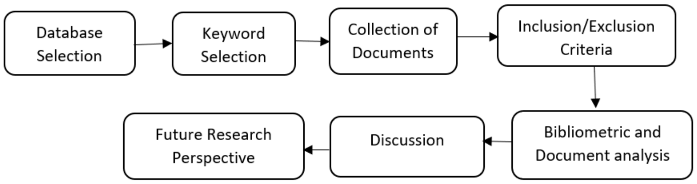
Machine learning and artificial intelligence can be found in nearly every area of daily living [1]. Machine learning techniques have found broad areas for application, such as in decision making, suggesting movies, recommending people, choosing loan appli- cants, influencing employment decisions, etc. [2]. While providing accurate predictions, these techniques can provide unfavorable predictions as well. When this affects critical or enormous decisions, it becomes a bias problem or error problem. When an algorithm generates results that are systematically biased as a result of false assumptions made during the machine learning process, this is known as machine learning bias [1]. Bias surfaces in different ways. Problems are frequently caused by choices made by people who de- velop or train machine learning algorithms. They might create algorithms that exhibit consciously or unconsciously biased thinking. Conversely, humans can introduce bias by using biased, erroneous, or incomplete datasets to train and/or validate machine learning algorithms. During the machine learning process, bias can develops at several phases. Although bias cannot be totally eliminated, it can be reduced to a minimum to ensure that bias and variance are in balance. Mitigation processes can be used to reduce the effect of bias problems. Different mitigation techniques are used based on the degree of the bias problem [3]. The purpose of creating a survey paper on ML bias and mitigation methods is to provide an overview of the research field and to help in identifying ML bias. The scope of this survey paper includes various ML biases, such as data bias, model bias, and algo- rithmic bias, as well as other kinds of bias. The objective of the paper is to address bias in machine learning studies. We review different ML bias mitigation strategies, including the various approaches, techniques, and measures to identify, quantify, and reduce ML bias, and examine different methods for ML bias prevention as well as the best ways to use these methods. Machine learning is becoming a more integral and common component of systems used in high-stakes applications that directly affect people; as a result, there is growing worry about the potential risks and harms these systems may pose [4]. The con- cern over the potential risks and harms these systems may bear is growing as machine learning becomes an increasingly significant and frequent component of systems used in high-stakes applications that directly affect people. Ensuring that automated systems do not instigate or uphold discrimination and inequality is one of the factors that must be taken into consideration. As a result, the field of algorithmic fairness, which seeks to study any unintended biases these systems may introduce or amplify, has rapidly expanded in recent years.

Although ML systems have the benefit of freeing humans from laborious tasks and are able to complete complex calculations more quickly [3], they are only as effective as the data on which they are trained. Although bias is not intentionally incorporated into ML algorithms, there is a risk of reproducing or even amplifying prejudice found in real-world data [2]. The need to make decisions in a fair and impartial manner raises ethical questions around systems that have an impact on people's lives. Thus, the limitations set by corporate practices, laws, social customs, and ethical obligations have been carefully considered in the substantial research carried out on bias and unfairness challenges [3]. Due to the fact that unfairness is defined differently in different societies, it can be challenging to identify and reduce it. Because of this, user experience, cultural, social, economic, political, legal, and ethical factors all have an effect on the unfairness criterion [5]. It is necessary to check algorithms for prejudice and unfairness as well as legal compliance before applying them in real-world scenarios. The results of these methods could significantly affect people's lives, often in negative ways [6]. Addressing ML bias is essential in order to ensure that ML algorithms are fair and unbiased as well as to prevent them from perpetuating or amplifying existing inequalities. Several techniques can be used to address ML bias, including data pre-processing, algorithmic techniques such as debiasing, and audibility and transparency measures. It is important to take a proactive approach to ML bias and to continually monitor and evaluate ML models in order to ensure that they are fair and unbiased.

In order to take into consideration the algorithmic limitations, new data science, ar- tificial intelligence (AI), and machine learning (ML) approaches are necessary [6]. As a result, we hope that this survey will assist academics and practitioners in better under- standing current bias mitigation strategies and supporting elements for the creation of new techniques.

Method

Systematic reviews are a popular way to gather information on a particular topic. In order to better comprehend research, components are gathered in a systematic review (RS). A popular strategy for compiling existing data on a subject of study is the systematic review [7]. Our systematic review was conducted using a procedure that involves seven steps, as shown in Figure 1.


Database Selection
Keyword Selection
Collection of Documents
Inclusion/Exclusion Criteria
Bibliometric and Document analysis
Discussion
Future Research Perspective

The instances of the keywords with the same meanings will be due to the software's inability to distinguish between single and plural terms or between words with the same roots. The most frequently used phrase was, as anticipated, ML prejudice. To understand the connections between different keywords used in 2020–2023 documents, a co-occurrence analysis was performed. Co-occurrence analysis is a technique used to identify patterns and relationships between keywords in a given set of documents. It examines how often certain keywords appear together, indicating potential associations and connections between them. By conducting this co-occurrence analysis, we aimed to cover significant relationships and identify commonly associated keywords within the selected documents. These findings can help reveal key themes, emerging trends, and areas of emphasis within the research field during that specific time period. This analysis only considered keywords with more than 10 occurrences, and duplicates were removed. The results of this analysis are presented in Figure 4 would typically provide a visualization or tabular representation of the keyword relationships. This visual representation may include various elements.

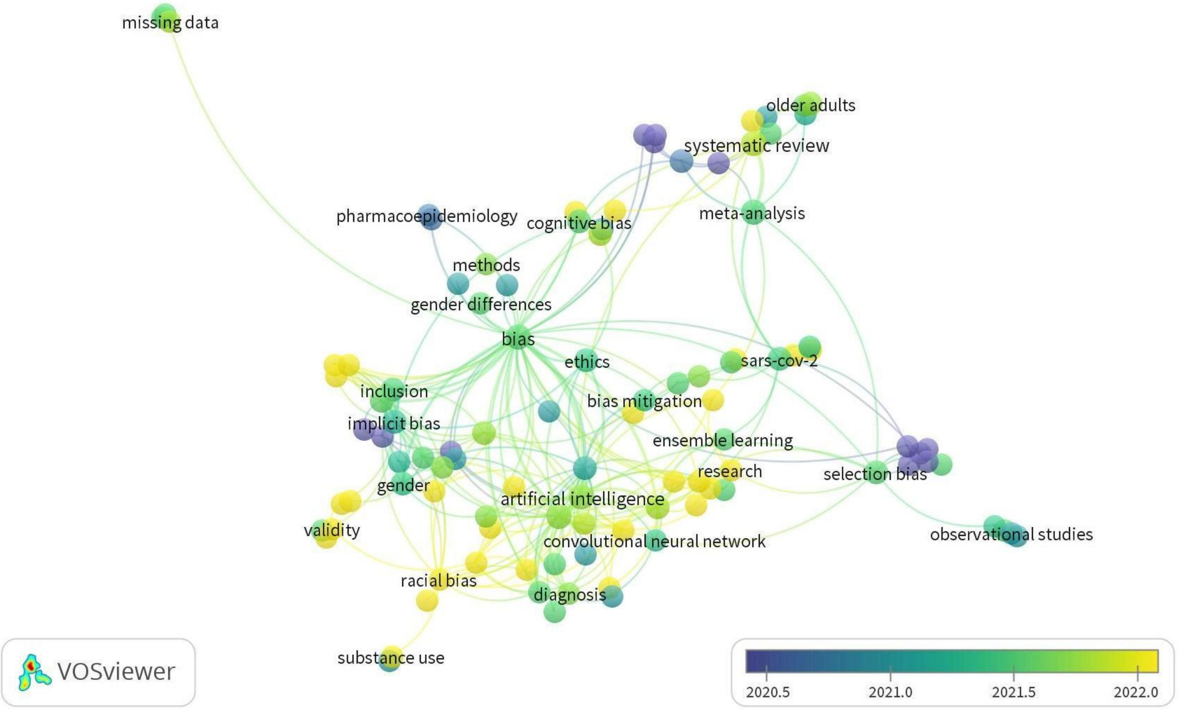


Figure 4. Co-related keywords.

- *Document Analysis*

Conducting a review of previous surveys is an essential step in research as it helps to identify the gaps in the literature that need to be filled. By analyzing related works, we can identify common themes, key findings, and areas that require further investigation. In our analysis, we considered various factors to determine the strengths and weaknesses of each paper. These factors included the methodology used, the quality of the dataset, the limitations of the study, and the accuracy of the results. By taking a comprehensive approach to our analysis, we were able to gain a deeper understanding of the research landscape and identify areas that require further attention.

In this section, we provide an overview of the previous surveys conducted in the literature, which enables us to identify the knowledge gap that our own survey addresses. To conduct this review, we analyzed related works and considered factors such as the paper's year, contribution, dataset, limitations, methods, and accuracy. The results of this analysis are presented in tables that highlight the research trends and gaps in the literature. Details are given in Table 4.

Machine Learning Bias

Machine learning bias refers to the systematic and unfair influence of certain factors or variables in a machine learning model, leading to incorrect or discriminatory outcomes [12]. Machine learning models are only as unbiased as the data they are trained on. If the data contain inherent biases, then these biases can be perpetuated and even amplified in the model's predictions. A figure shows a quantify bias in ML in Figure 5.

One example of machine learning bias is algorithmic discrimination, where a model is trained on data that are biased against certain groups of people, leading to discriminatory outcomes. For example, an algorithm that was trained on historical hiring data that

1. Dataset Bias

2. **Measure the distribution of the data:** Analyzing the frequency of different attributes across the dataset helps identify potential biases. By calculating the proportions or counts of attribute categories, you can understand their rep- resentation. Over-representation or under-representation of certain attributes may indicate bias in the data. For example, if a dataset used for college ad- missions contains a significantly higher proportion of students from affluent backgrounds, it could indicate socioeconomic bias. For example, if a dataset used for college admissions contains a significantly higher proportion of students from affluent backgrounds compared to the general population, it could indicate socioeconomic bias. This bias may stem from inequitable access to resources or opportunities in the admissions process.

   Check for imbalances in the target variable: Target variables can lead to biased predictions, particularly for under-represented groups. It is crucial to examine the distribution of the target variable to ensure fairness. Identify whether there are significant disparities in the number of samples belonging to different target categories. For instance, in a medical diagnosis model, if the dataset has a dis- proportionate number of healthy patients compared to patients with a particular disease, the model might struggle to accurately predict the disease cases.

Use statistical tests for assessing attribute distribution: Statistical tests like chi- squared tests or *t*-tests can provide quantitative insights into the differences in attribute distribution across different groups. These tests help determine whether there is a significant association between two categorical variables. They can be used to assess whether observed differences in attribute distribution across

groups are statistically significant or due to chance. By applying these tests, you can quantify the extent of bias and ascertain if the observed differences are statistically significant or if they can be attributed to random variations.

3. Scraped Data Bias

Evaluate the sources and methods used to scrape the data to identify potential biases or inaccuracies in the data. Assess the reliability and credibility of the data sources. Consider the reputation, authority, and transparency of the sources to ensure the data are trustworthy. Evaluate the methodology employed for data scraping. Determine whether it adhered to ethical guidelines, respected user privacy, and obtained consent if required. Consider potential biases in the data sources. If the sources are known to have inherent biases or limitations, then these can impact the quality and representations of the scraped data.

Check for missing data or errors in the scraped data that could affect the model’s predictions. Examine the scraped data for missing values or errors that can affect the model’s predictions. Missing or erroneous data can introduce bias or distort the analysis. Identify the types and patterns of missing data. Determine whether they are missing at random or if certain attributes or groups are more affected. Systematic missing can lead to biased results. Investigate the potential causes of missing data, such as technical issues during scraping or limitations in the data sources. Addressing missing data appropriately is crucial in avoiding biased or inaccurate predictions.

Analyze the distribution of the scraped data to identify any under-represented groups or biases. Assess the distribution of attributes within the scraped data to identify under-represented groups or biases. Understanding the representa- tion of different groups is vital for fair modeling. Calculate the frequencies or proportions of attribute categories and compare them to known distributions or benchmarks. Look for significant disparities or imbalances in attribute represen- tation. Under-represented groups may be susceptible to biased predictions or exclusion from the modeling process. Analyzing attribute distribution helps iden- tify potential biases, such as gender, race, ethnicity, or socioeconomic disparities, which may exist in the data.

Abstract Data Bias

Evaluate the methods used to generate or extract abstract data: When assessing potential biases or inaccuracies in abstract data, it is essential to scrutinize the methods used for data generation or extraction. This involves understanding the data collection process, including the sources, instruments, and techniques employed. For example, if the data were collected through surveys, evaluate whether the survey design could introduce response or sampling biases. If the data were obtained from online sources, consider the limitations of web scraping techniques and potential biases associated with the sampled websites or platforms.

Check for missing data or errors in the abstract data: Missing data or errors can significantly impact the accuracy and validity of a model’s predictions. Carefully examine the abstract data for any missing values, outliers, or inconsistencies. Missing data can occur due to various reasons, such as non-response, data entry errors, or unintentional omissions. Investigate whether the missing data are random or if there is a systematic pattern to its absence, as this pattern could introduce biases. Depending on the extent of missing data, imputation techniques such as mean imputation, regression imputation, or multiple imputations can be employed to address the gaps and minimize bias.

Analyze the distribution of the abstract data: Analyzing the distribution of abstract data is an essential step in understanding potential biases and under- represented groups within the dataset. It is the distribution of the abstract data

1. Bias Reduction Strategy

To make forecasts or judgments, machine learning (ML) algorithms use statistical models that have been trained on historical data. However, if the data used to teach these algorithms is biased, the algorithms may continue or even amplify that bias. This raises serious concerns in numerous sectors because it may result in unfair or discriminatory outcomes [1]. Machine learning bias refers to the phenomenon where a machine learning algorithm produces results that systematically favor one group of people over another, often due to historical discrimination or other societal factors. The difference between the predicted output's true value and the expected value for a given input is one prevalent definition of bias in machine learning. This is mathematically represented as a bias in Equation (1)

$$y = E[f(x)] - y \qquad (1)$$

where *E* stands for the expected value and *f*(*x*) is the projected output for input *x*; *y* is the actual output for input *x* [12].

Diverse and representative training data: Using diverse and representative training data are one of the most efficient methods to reduce bias. This can make sure that the data used to train the ML model represents the complete range of experiences and viewpoints of the population being studied. Utilizing diverse and representative training data is crucial in minimizing bias. This can be achieved by ensuring that the training dataset, denoted as *D*, contains a wide range of examples from different subgroups or classes. Mathematically, we can represent this as Equation (2):

$$D = \{x_1, x_2, \ldots, x_n\} \qquad (2)$$

where *xi* represents an example in the dataset. By including a diverse set of examples that represent various experiences and viewpoints, the ML model can learn to make unbiased predictions across different groups.

Data pre-processing: Techniques for data pre-processing can be used to find and eliminate prejudice in the training data. To balance the representation of various subgroups in the training data, methods like oversampling or undersampling may be used. Mathematically, this can be represented as Equation (3):

$$D_processed = PreProcess(D) \quad (3)$$

where *D* process represents modified data, PreProcess represents the data enhancement function, and (*D*) represents the original data input. This equation shows how a pre- processing function (*PreProcess*) changes original data (*D*) into improved data that has been changed.

Algorithmic transparency: By making it simpler to spot and correct any possible biases in the ML model, ensuring algorithmic transparency can help to mitigate bias. This might entail employing strategies like interpretability methods, which can make