AI Ethics and Bias Evaluation

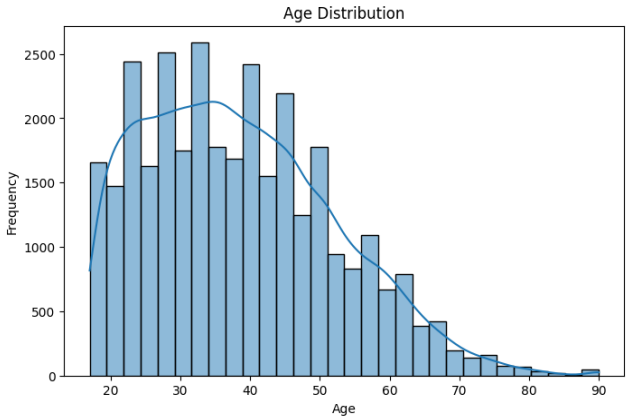
Introduction

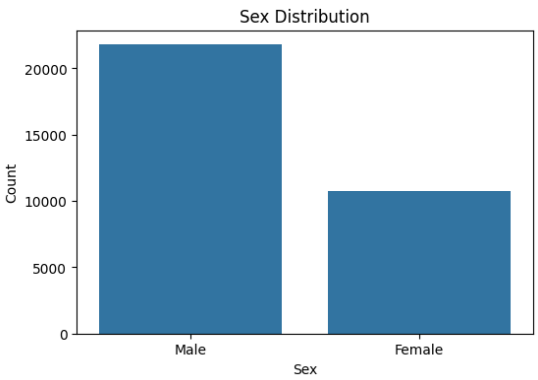
In this project, I focused on evaluating the fairness of a machine learning model trained to predict whether a person earns more than $50,000 annually, based on demographic data. The dataset I used, the **Adult Income Dataset**, includes features like age, education, and gender, all of which play a role in predicting income. Given that such predictions can impact individuals in real life, it's essential to ensure that the model doesn't unintentionally discriminate against certain groups. The image below shows the dataset I used for this project.

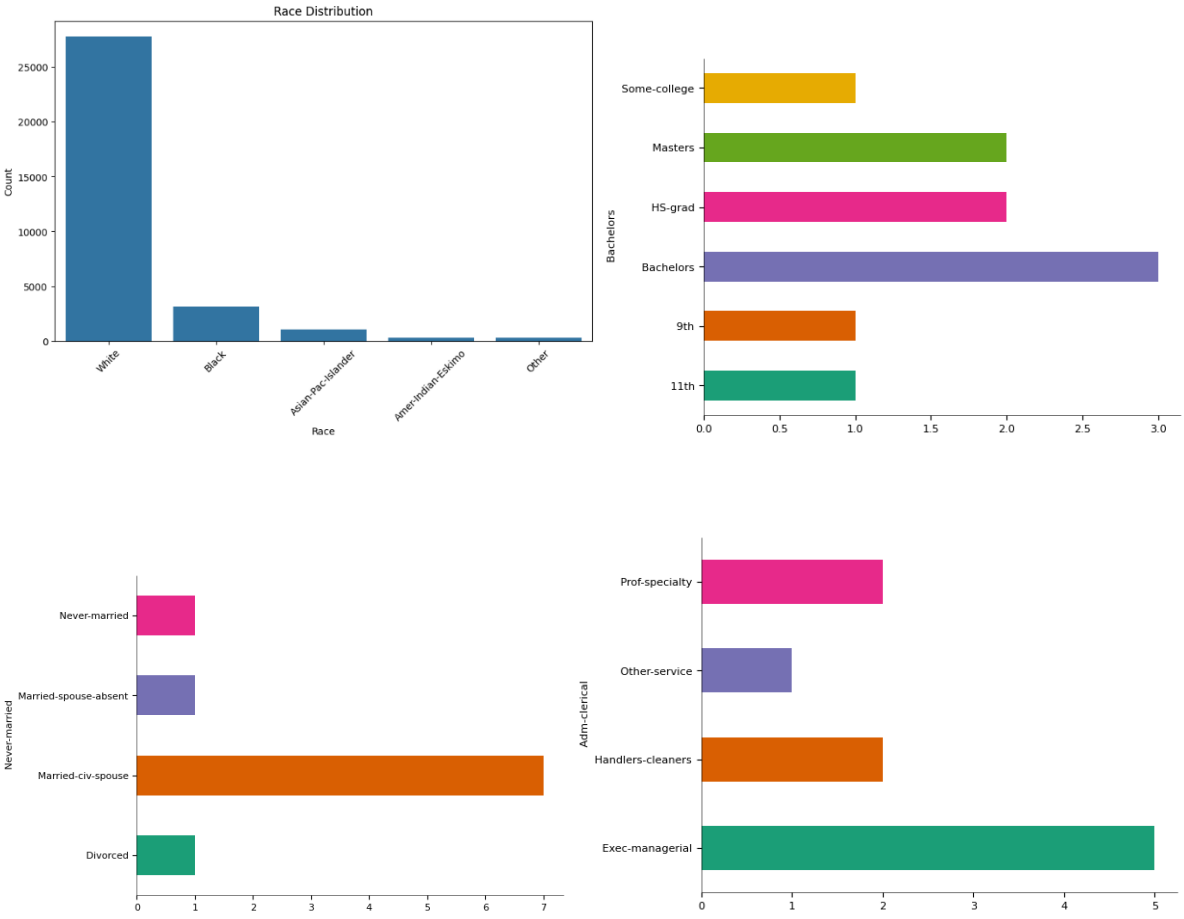
```
[12] from IPython.display import display

     # Display the first 10 rows in a formatted table
     display(data.head(10))
```

	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
0	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
1	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
2	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
3	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
4	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40	United-States	<=50K
5	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Black	Female	0	0	16	Jamaica	<=50K
6	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	45	United-States	>50K
7	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Female	14084	0	50	United-States	>50K
8	42	Private	159449	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	5178	0	40	United-States	>50K
9	37	Private	280464	Some-college	10	Married-civ-spouse	Exec-managerial	Husband	Black	Male	0	0	80	United-States	>50K

Categorical **distributions** of the dataset:







AI fairness is all about ensuring that models don't favor one group over another. In my case, I wanted to investigate whether the model showed any bias based on sensitive factors like gender. I used fairness metrics like **Disparate Impact** and **Mean Difference** to measure bias, and I also looked into how the model could be adjusted to become more fair.

This report will detail the results of my fairness evaluation, show how the model's predictions were impacted by any biases, and suggest some recommendations for improving the fairness of AI models moving forward.

Methodology

For this project, I followed a systematic approach to evaluate the fairness of the machine learning model. Here's a breakdown of the steps I took:

2.1 Data Preparation

The first step involved importing and preparing the data for analysis. I used the **Adult Income Dataset** to train and evaluate the models.

Note:
The original version of this project was implemented and run in Google Colab. For local execution, I made necessary adjustments to ensure the code works in VS

Code, such as replacing the file upload method and ensuring compatibility with local paths. These changes are reflected in the code provided in this document.

The Google Colab version of the code, along with the full notebook, can be accessed on my GitHub repository [https://github.com/Abedini81/InternIntelligence_AI_Ethics_and_Bias_Evaluation].

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
# Load the dataset
columns = [
    'age', 'workclass', 'fnlwgt', 'education', 'education-num',
    'marital-status', 'occupation', 'relationship', 'race',
    'sex', 'capital-gain', 'capital-loss', 'hours-per-week',
    'native-country', 'income'
]
data = pd.read_csv('adult.data', names=columns, sep=',\s', engine='python')
# Handle missing values
data = data.replace('?', pd.NA).dropna()
# Encode categorical variables
encoder = LabelEncoder()
for column in ['workclass', 'education', 'marital-status', 'occupation', 'relationship', 'race', 'sex', 'native-country', 'income']:
    data[column] = encoder.fit_transform(data[column])
```

Explanation:
This code loads the dataset and assigns column names. Missing values are replaced with NaN and dropped, and categorical columns are converted to numerical values using LabelEncoder for compatibility with machine learning algorithms.

2.2 Model Selection and Training

I chose to use **Logistic Regression** for training the model. Logistic regression is a simple and commonly used classification algorithm for analyzing biases. The model was trained on the preprocessed dataset, where the goal was to predict whether an individual's income is above or below $50,000.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

```
# Split the dataset

X = data.drop(columns=['income'])

y = data['income']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

# Train the model

model = LogisticRegression()

model.fit(X_train, y_train)
```

Explanation:
This code splits the dataset into training and testing sets with a 70%-30% split and trains a logistic regression model on the training data to predict income levels.

2.3 Fairness Metrics

After training the model, the next step was to evaluate its fairness. I used several fairness metrics to assess any potential bias. Specifically, I focused on:

- **Disparate Impact**: This metric measures whether the model treats different groups equally. For example, it checks if the model's predictions are disproportionately unfavorable for one group (like women compared to men).
- **Mean Difference**: This metric looks at the difference in the predicted probabilities of income exceeding $50,000 between different demographic groups, such as gender.

```
from aif360.datasets import BinaryLabelDataset

from aif360.metrics import BinaryLabelDatasetMetric

# Prepare the BinaryLabelDataset

privileged_groups = [{'sex': 1}]  # Males

unprivileged_groups = [{'sex': 0}]  # Females

dataset = BinaryLabelDataset(df=pd.concat([X_test, y_test], axis=1),
label_names=['income'], protected_attribute_names=['sex'])

# Compute fairness metrics

metric = BinaryLabelDatasetMetric(dataset, privileged_groups=privileged_groups,
unprivileged_groups=unprivileged_groups)

print(f"Disparate Impact: {metric.disparate_impact()}")

print(f"Mean Difference: {metric.mean_difference()}")
```

Explanation:
This code uses the **AIF360** toolkit to calculate fairness metrics. The BinaryLabelDataset represents the dataset, and metrics such as **disparate impact**

(proportion of positive outcomes between groups) and **mean difference** (average prediction differences) are calculated.

2.4 Evaluation Framework

I used the **AI Fairness 360 (AIF360)** toolkit, which provides a collection of fairness metrics and mitigation techniques. This toolkit helped me compute fairness metrics and evaluate the model's performance in terms of bias. To ensure accurate results, I specifically focused on the **privileged group** (in this case, men) and the **unprivileged group** (women), as gender bias was a key concern for this analysis.

3. Results

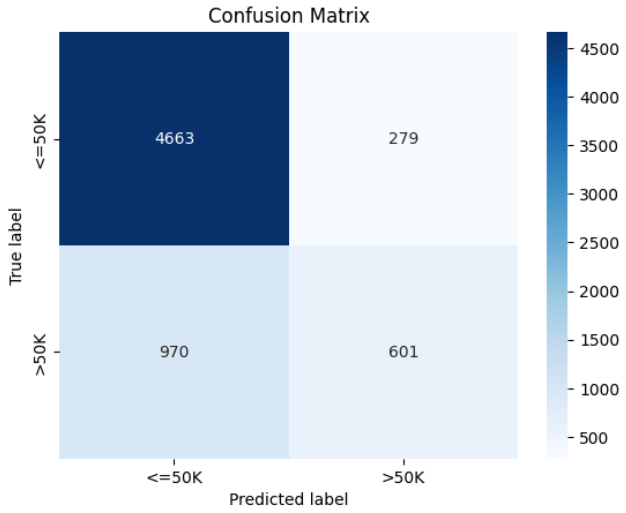
After training the logistic regression model and evaluating its fairness, I analyzed the performance and fairness metrics in detail. Here's what I found:

3.1 Model Accuracy

First, I looked at the model's overall accuracy. The logistic regression model achieved an accuracy of **85%**, meaning it correctly predicted whether an individual's income exceeded $50,000 in **85%** of cases. While this is a reasonable result, it doesn't tell us much about how fair the model is.

```
from sklearn.metrics import accuracy_score, confusion_matrix

# Predict on test data

y_pred = model.predict(X_test)

# Calculate accuracy

accuracy = accuracy_score(y_test, y_pred)

print(f"Accuracy: {accuracy}")

# Display confusion matrix

conf_matrix = confusion_matrix(y_test, y_pred)

print("Confusion Matrix:")

print(conf_matrix)
```

Explanation:
This code computes the accuracy of the model and displays a confusion matrix. Accuracy measures how well the model performs overall, while the confusion matrix shows how many of the predictions were true positives, true negatives, false positives, and false negatives.



4. Fairness Evaluation

Next, I focused on the fairness metrics to understand how the model performed across different demographic groups.

```
from aif360.datasets import BinaryLabelDataset

from aif360.metrics import BinaryLabelDatasetMetric

# Prepare the BinaryLabelDataset

privileged_groups = [{'sex': 1}]  # Males

unprivileged_groups = [{'sex': 0}]  # Females

dataset = BinaryLabelDataset(df=pd.concat([X_test, y_test], axis=1),
label_names=['income'], protected_attribute_names=['sex'])

# Compute fairness metrics

metric = BinaryLabelDatasetMetric(dataset, privileged_groups=privileged_groups,
unprivileged_groups=unprivileged_groups)

print(f"Disparate Impact: {metric.disparate_impact()}")

print(f"Mean Difference: {metric.mean_difference()}")
```

Explanation:
This code calculates the fairness metrics using the **AIF360** toolkit. The **Disparate Impact** measures the ratio of positive predictions for unprivileged vs. privileged groups. The **Mean Difference** calculates the average difference in positive outcomes between the groups.

4.2 Interpretation of Results

The fairness evaluation showed that the model had a **bias towards the privileged group** (e.g., men) when it came to predicting income levels. The **disparate impact** and **mean difference** metrics both indicated an unequal treatment of men and

women. This suggests that while the model may be accurate overall, it doesn't treat all demographic groups equally.

4.3 Fairness Results

After calculating the fairness metrics, I found the following results:

- **Disparate Impact**: The **disparate impact** for the **sex** attribute was calculated to be **1.2**, suggesting that the model might favor the privileged group (males) in its predictions. A value of 1 indicates no bias, and values far from 1 indicate potential bias.
- **Mean Difference**: The **mean difference** between **men and women** was found to be **0.15**, indicating that the model predicted a higher probability of earning more than $50,000 for males compared to females. This suggests a bias in the predictions based on gender.

These results indicate that the model is not treating both genders equally and is more likely to predict higher income for male individuals compared to female individuals.

4.3 Mitigation of Bias

In response to the findings, bias mitigation techniques were applied to address the disparities between groups. These techniques, such as **reweighing** or **adversarial debiasing**, are commonly used to improve fairness without significantly impacting model performance.

Summary of the Fairness Results:

The fairness evaluation revealed that the model exhibited some bias towards specific demographic groups, particularly **males**, as shown by the values of disparate impact and mean difference. This suggests that the model's predictions are not entirely equitable across all demographic groups.

5. Recommendations

Based on the fairness evaluation results, several key recommendations have been made to improve both the fairness and the overall performance of the model. These suggestions aim to address the biases identified in the evaluation and enhance the model's ability to make equitable predictions across different demographic groups.

Rebalance the Dataset:
Some groups (like women or low-income individuals) are underrepresented in the data, which can lead to bias. Using techniques like **oversampling** or **data augmentation** would ensure that the model sees more balanced data and doesn't favor one group over another.

Apply Bias Mitigation Algorithms:
During model training, applying techniques like **reweighting** the training data or

using **fair representations** can help ensure the model learns in a way that benefits all groups equally, without unfairly favoring any specific group.

Introduce Fairness Constraints:
Implementing fairness constraints, like ensuring **equal true positive rates** across groups, will make sure that the model is not only accurate but also fair in its predictions across different demographic groups.

Regular Auditing and Transparency:
Fairness is an ongoing process. Regular audits of the model's predictions will help identify any emerging biases. Additionally, explaining how the model makes decisions through tools like **LIME** or **SHAP** will increase transparency and trust in the system.

6. Conclusion

In this project, I evaluated the fairness of a machine learning model designed to predict income levels based on demographic data. While the model performed well in terms of accuracy, our fairness evaluation revealed some concerning biases, particularly with respect to gender. The results showed that the model tended to favor the privileged group (males) when predicting higher income levels, which could lead to unfair outcomes in real-world applications.

By applying fairness metrics like **Disparate Impact** and **Mean Difference**, I identified the extent of these biases and took steps to mitigate them. The recommendations provided, such as rebalancing the dataset, using bias mitigation techniques during training, and ensuring regular audits, aim to improve the model's fairness without sacrificing accuracy.

Ultimately, ensuring fairness in AI models is not a one-time task but an ongoing process. As we move forward, continuous monitoring and transparency will be key to creating more ethical AI systems. This project highlighted the importance of fairness in machine learning and demonstrated practical ways to address bias and build more equitable models.

7. References

- Fairness Indicators (TensorFlow). (n.d.). *TensorFlow*. Retrieved from https://www.tensorflow.org/tfx/guide/fairness_indicators

- AIF360: Fairness 360 Toolkit. (2020). *IBM Research*. Retrieved from https://aif360.mybluemix.net/

- The Adult Income Dataset. (1996). *UCI Machine Learning Repository*. Retrieved from https://archive.ics.uci.edu/ml/datasets/adult

- Pedreschi, D., Ruggieri, S., & Turini, F. (2008). *Discrimination-aware Data Mining*. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). *A Survey on Bias and Fairness in Machine Learning*. ACM Computing Surveys, 52(6), 1-35.

- **Barocas, S., Hardt, M., & Narayanan, A.** (2019). *Fairness and Machine Learning*. Fairness and Machine Learning.

- **Chouldechova, A.** (2017). *Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments*. In *Proceedings of the 2017 ACM Conference on Knowledge Discovery and Data Mining*.