Part 1: Theoretical Understanding (30%)

1. Short Answer Questions

Q1: Define algorithmic bias and provide two examples of how it manifests in AI systems.

Algorithmic bias refers to systematic and repeatable errors in an AI system that lead to unfair outcomes, often disadvantaging certain individuals or groups. Bias can emerge from training data, model design, or real world deployment contexts. Example 1 – Biased Job Screening: Amazon's hiring algorithm penalized resumes that included the word "women's" (e.g., "women's chess club") because it was trained on data from predominantly male applicants, reinforcing gender bias.

Example 2 – Credit Scoring Disparities: Some AI based credit scoring systems have reduced loan approvals for minority applicants due to biased historical lending data, even when applicants had similar financial profiles to others.

```
Q2: Explain the difference between transparency and explainability in AI. Why are both impo
```

Transparency refers to how open and accessible an AI system's components and operations are. It involves disclosing model architecture, training data sources, and decision logic.

Explainability refers to the ability to understand why and how a specific AI decision was made. This is crucial for building trust and accountability, especially in high-stakes decisions like healthcare or justice.

```
Why  it is  Important:
```

Transparency helps identify and correct systemic issues in AI systems, while explainability empowers users, regulators, and impacted individuals to interpret and challenge decisions — reducing risks of harm and enhancing fairness.

```
Q3: How does GDPR (General Data Protection Regulation) impact AI development in the EU?
```

GDPR mandates data privacy, accountability, and transparency in data processing, directly impacting AI systems that rely on personal data.

Key effects:

Right to Explanation: Individuals can request explanations for algorithmic decisions (Article 22).

Consent: AI systems must obtain informed and explicit consent before processing personal data.

Data Minimization: Only data necessary for a specific purpose can be collected and used.

This pushes developers to adopt privacy-preserving and human-centric AI practices, ensuring compliance while safeguarding user rights.

2. Ethical Principles Matching

Match the following principles to their definitions:

A) Justice

B) Non-maleficence

C) Autonomy

D) Sustainability:

1) Ensuring AI does not harm individuals or society.

2) Respecting users' right to control their data and decisions.

3) Designing AI to be environmentally friendly.

4) Fair distribution of AI benefits and risks.

Principle	Definition
A) Justice	Fair distribution of AI benefits and risks.
B) Non-maleficence	Ensuring AI does not harm individuals or society.
C) Autonomy	Respecting users' right to control their data and decisions.
D) Sustainability	Designing AI to be environmentally friendly.

Part 2: Case Study Analysis (40%)
Case 1: Biased Hiring Tool

Scenario: Amazon's AI recruiting tool penalized female candidates.

Tasks:

Identify the source of bias (e.g., training data, model design).

Propose three fixes to make the tool fairer.

Suggest metrics to evaluate fairness post-correction.

1. Source of Bias

Training Data Bias:

The model was trained on past hiring data where male candidates were overrepresented, leading it to learn that male-associated patterns were more desirable.

Model Design Flaws:

The system lacked a fairness check and feature sensitivity review. it treated certain gender related terms as negative indicators.

Unbalanced Labeling:

Past hiring decisions encoded implicit gender bias, which the AI then replicated.

```
2. Three Fixes to Make the Tool Fairer
```

Debias Training Data

Use techniques like reweighting or resampling to reduce overrepresentation of male dominated examples. Remove biased labels or adjust them with domain experts.

Feature Scrubbing & Fairness Constraints

Eliminate gender proxies (e.g., college women's clubs, first names) from input features.

Apply fairness aware algorithms that enforce demographic parity or equal opportunity.

Fairness Audits & Human Review

Regularly run audits using tools like AI Fairness 360.

Integrate human reviewers at checkpoints, especially for sensitive roles.

```
3. Fairness Metrics to Use Post-Correction
```

Disparate Impact Ratio

Measures whether outcomes differ between groups (ideal is near 1.0).

Equal Opportunity Difference

Compares true positive rates across groups to ensure fairness in opportunity.

False Positive Rate Parity

Ensures misclassifications don't disproportionately affect one gender group.

```
Case 2: Facial Recognition in Policing

Scenario: A facial recognition system misidentifies minorities at higher rates.
```

Tasks:

1) Discuss ethical risks (e.g., wrongful arrests, privacy violations).

2) Recommend policies for responsible deployment.

```
1. Ethical Risks
```

Wrongful Arrests and Legal Harm

A misidentified person may be jailed or fined unjustly, violating due process.

Mass Surveillance

Use of facial recognition without consent or oversight violates privacy rights.

Algorithmic Discrimination

Marginalized groups bear the burden of technical inaccuracies, reinforcing systemic biases.

```
2. Policy Recommendations for Responsible Deployment
```

Independent Auditing Before Use

Require vendors to provide transparency into training datasets and performance breakdowns across race, age, and gender.

Human Oversight Mandate

Facial recognition results should never be used as sole evidence. Decisions must be verified by trained human personnel.

Strict Deployment Boundaries

Restrict use in public spaces unless with a warrant or extreme necessity.

Ban real-time mass surveillance (as some EU cities have done).

Consent and Transparency Policies

Communities should be informed when and where the technology is deployed. Opt-out mechanisms should be explored.

```
Part 3: Practical Audit (25%)
Task: Audit a Dataset for Bias

Dataset: COMPAS Recidivism Dataset.

Goal:
```

Use Python and AI Fairness 360 (IBM’s toolkit) to analyze racial bias in risk scores.

Generate visualizations (e.g., disparity in false positive rates).

Write a 300-word report summarizing findings and remediation steps.

Deliverable: Code + report.

```
!pip install aif360
!pip install -q 'aif360[all]'  # Optional: for extra datasets and metrics
!pip install fairlearn
```

```
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (fr
Downloading aif360-0.6.1-py3-none-any.whl (259 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 259.7/259.7 kB 4.3 MB/s eta 0:00:00
Installing collected packages: aif360
Successfully installed aif360-0.6.1
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 275.7/275.7 kB 5.4 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.6/2.6 MB 47.6 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 MB 54.5 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 240.0/240.0 kB 17.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.8/45.8 kB 3.1 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 897.5/897.5 kB 44.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 4.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 96.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 86.1 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 43.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 2.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 5.8 MB/s eta 0:00:00
```

```
                                        12.3/12.3 MB 106.3 MB/s eta 0:00:00
                                        1.6/1.6 MB 68.0 MB/s eta 0:00:00
                                        69.4/69.4 kB 5.0 MB/s eta 0:00:00
                                        386.9/386.9 kB 29.1 MB/s eta 0:00:00
                                        133.5/133.5 kB 10.9 MB/s eta 0:00:00
                                        59.7/59.7 kB 4.4 MB/s eta 0:00:00
                                        66.4/66.4 kB 5.6 MB/s eta 0:00:00
  Building wheel for BlackBoxAuditing (setup.py) ... done
  Building wheel for lime (setup.py) ... done
Requirement already satisfied: fairlearn in /usr/local/lib/python3.11/dist-packages (0
Requirement already satisfied: numpy>=1.24.4 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: pandas>=2.0.3 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: scikit-learn>=1.2.1 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: scipy>=1.9.3 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dis
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-
```

```
import numpy as np
import pandas as pd
from aif360.datasets import CompasDataset
from aif360.metrics import BinaryLabelDatasetMetric, ClassificationMetric
from aif360.algorithms.preprocessing import Reweighing
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
```

```
/usr/local/lib/python3.11/dist-packages/inFairness/utils/ndcg.py:37: FutureWarning: We'v
  vect_normalized_discounted_cumulative_gain = vmap(
/usr/local/lib/python3.11/dist-packages/inFairness/utils/ndcg.py:48: FutureWarning: We'v
  monte_carlo_vect_ndcg = vmap(vect_normalized_discounted_cumulative_gain, in_dims=(0,))
```

```
!mkdir -p data/compas
!wget https://raw.githubusercontent.com/propublica/compas-analysis/master/compas-scores-two-
```

```
--2025-07-21 09:43:10--  https://raw.githubusercontent.com/propublica/compas-analysis/ma
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443
HTTP request sent, awaiting response... 200 OK
Length: 2546489 (2.4M) [text/plain]
Saving to: ‘data/compas/compas-scores-two-years.csv’

data/compas/compas- 100%[===================>]   2.43M  --.-KB/s    in 0.07s

2025-07-21 09:43:10 (32.6 MB/s) - ‘data/compas/compas-scores-two-years.csv’ saved [25464
```

```
import pandas as pd

df = pd.read_csv("data/compas/compas-scores-two-years.csv")
df.head()
```

	id	name	first	last	compas_screening_date	sex	dob	age	age_cat	
0	1	miguel hernandez	miguel	hernandez	2013-08-14	Male	1947-04-18	69	Greater than 45	
1	3	kevon dixon	kevon	dixon	2013-01-27	Male	1982-01-22	34	25 - 45	Af Ame
2	4	ed philo	ed	philo	2013-04-14	Male	1991-05-14	24	Less than 25	Af Ame
3	5	marcu brown	marcu	brown	2013-01-13	Male	1993-01-21	23	Less than 25	Af Ame
4	6	bouthy pierrelouis	bouthy	pierrelouis	2013-03-26	Male	1973-01-22	43	25 - 45	

5 rows × 53 columns

```
from aif360.datasets import StandardDataset

# Simplified pre-processing (drop NaNs and non-essential features)
df = df.dropna(subset=['race', 'sex', 'age', 'juv_fel_count', 'decile_score', 'is_recid', 'p
df = df[(df['days_b_screening_arrest'] <= 30) & (df['days_b_screening_arrest'] >= -30)]

# Convert to StandardDataset
compas_data = StandardDataset(df,
    label_name='is_recid',
    favorable_classes=[0],
    protected_attribute_names=['race'],
    privileged_classes=[['Caucasian']],
    features_to_drop=['name', 'first', 'last', 'compas_screening_date', 'dob', 'age_cat', 'c
)

print(compas_data.features.shape)
```

```
WARNING:root:Missing Data: 6172 rows removed from StandardDataset.
WARNING:root:[np.float64(0.0), np.float64(1.0)] listed but not observed for feature race
(0, 44)
```

```
from aif360.datasets import StandardDataset

# Simplified pre-processing (drop NaNs and non-essential features)
```

```
df = df.dropna(subset=['race', 'sex', 'age', 'juv_fel_count', 'decile_score', 'is_recid', 'p
df = df[(df['days_b_screening_arrest'] <= 30) & (df['days_b_screening_arrest'] >= -30)]

# Convert to StandardDataset
compas_data = StandardDataset(df,
    label_name='is_recid',
    favorable_classes=[0],
    protected_attribute_names=['race'],
    privileged_classes=[['Caucasian']],
    features_to_drop=['name', 'first', 'last', 'compas_screening_date', 'dob', 'age_cat', 'c
)

print(compas_data.features.shape)
```

```
WARNING:root:Missing Data: 6172 rows removed from StandardDataset.
WARNING:root:[np.float64(0.0), np.float64(1.0)] listed but not observed for feature race
(0, 44)
```

```
#  Step 5: Split data
train, test = compas_data.split([0.7], shuffle=True)

#  Step 6: Bias metric before
from aif360.metrics import BinaryLabelDatasetMetric

metric = BinaryLabelDatasetMetric(train,
                                  unprivileged_groups=[{'race': 'African-American'}],
                                  privileged_groups=[{'race': 'Caucasian'}])
print("Disparate Impact (before):", metric.disparate_impact())
```

```
Disparate Impact (before): nan
/usr/local/lib/python3.11/dist-packages/aif360/metrics/binary_label_dataset_metric.py:10
  return (self.num_positives(privileged=privileged)
```

```
from aif360.algorithms.preprocessing import Reweighing

RW = Reweighing(unprivileged_groups=[{'race': 'African-American'}],
                privileged_groups=[{'race': 'Caucasian'}])
train_rw = RW.fit_transform(train)
```

```
/usr/local/lib/python3.11/dist-packages/aif360/algorithms/preprocessing/reweighing.py:66
  self.w_p_fav = n_fav*n_p / (n*n_p_fav)
/usr/local/lib/python3.11/dist-packages/aif360/algorithms/preprocessing/reweighing.py:67
  self.w_p_unfav = n_unfav*n_p / (n*n_p_unfav)
/usr/local/lib/python3.11/dist-packages/aif360/algorithms/preprocessing/reweighing.py:68
  self.w_up_fav = n_fav*n_up / (n*n_up_fav)
/usr/local/lib/python3.11/dist-packages/aif360/algorithms/preprocessing/reweighing.py:69
  self.w_up_unfav = n_unfav*n_up / (n*n_up_unfav)
```

```
print("Total records:", len(compas_data.instance_names))
```

```
Total records: 0
```

```
# Load raw CSV
df = pd.read_csv("data/compas/compas-scores-two-years.csv")

# Only filter rows with missing labels or invalid recidivism values
df = df[df['is_recid'] != -1]  # Remove only unclear recidivism values

# Retain a manageable number of features
df = df[['age', 'sex', 'race', 'juv_fel_count', 'decile_score', 'is_recid', 'priors_count']]

# Drop rows with missing values (safe fallback)
df = df.dropna()

# Confirm dataset is now non-empty
print(" Records after fix:", len(df))
```

```
✅ Records after fix: 7214
```

```
df['race'].value_counts()
```

race	count
African-American	3696
Caucasian	2454
Hispanic	637
Other	377
Asian	32
Native American	18

dtype: int64

```
# Select only relevant columns
df = df[['age', 'sex', 'race', 'juv_fel_count', 'decile_score', 'is_recid', 'priors_count']]

# Drop NA values
df = df.dropna()
```

```
# Encode categorical columns
df['sex'] = df['sex'].map({'Male': 1, 'Female': 0})
df['race'] = df['race'].replace({
    'Caucasian': 1,
    'African-American': 0,
    'Hispanic': 2,
    'Other': 3,
    'Asian': 4,
    'Native American': 5
})
```

```
/tmp/ipython-input-21-1931275223.py:9: FutureWarning: Downcasting behavior in `replace`
  df['race'] = df['race'].replace({
```

```
# Filter for binary race categories (optional but recommended for fairness comparison)
df = df[df['race'].isin([0, 1])]
```

```
from aif360.datasets import StandardDataset

compas_data = StandardDataset(df,
    label_name='is_recid',
    favorable_classes=[0],  # No recidivism
    protected_attribute_names=['race'],
    privileged_classes=[[1]]  # Caucasian
)

print("AIF360 dataset size:", len(compas_data.instance_names))
```

```
AIF360 dataset size: 6150
```

```
train, test = compas_data.split([0.7], shuffle=True)
```

```
from aif360.algorithms.preprocessing import Reweighing

RW = Reweighing(unprivileged_groups=[{'race': 0}], privileged_groups=[{'race': 1}])
train_transf = RW.fit_transform(train)
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
```

```
# Extract features and labels
X_train = train_transf.features
y_train = train_transf.labels.ravel()
X_test = test.features
y_test = test.labels.ravel()

# Scale features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train model
clf = LogisticRegression()
clf.fit(X_train_scaled, y_train)

# Predict
y_pred = clf.predict(X_test_scaled)



from aif360.metrics import ClassificationMetric

# Rewrap predictions into AIF360 format
test_pred = test.copy()
test_pred.labels = y_pred

metric = ClassificationMetric(test,
                              test_pred,
                              unprivileged_groups=[{'race': 0}],
                              privileged_groups=[{'race': 1}])

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Disparate Impact:", metric.disparate_impact())
print("Equal Opportunity Difference:", metric.equal_opportunity_difference())
```

```
Accuracy: 0.6915989159891599
Disparate Impact: 0.6424265025794021
Equal Opportunity Difference: -0.16183480234065184
```

Metric	Value	Interpretation
Accuracy	`69.16%`	Moderate predictive power — the model is correct ~69% of the time.
Disparate Impact	`0.64`	Below the fairness threshold (`< 0.8`). Indicates potential bias against the unprivileged gr
Equal Opportunity Difference	`-0.16`	Negative value shows the model gives **fewer favorable outcomes** (e.g., no recidivism) to t

What This Means The model is not fair across racial groups.

African American individuals are less likely to receive favorable outcomes (e.g., being predicted as not re-offending), even if they should.

Bias persists even after using the Reweighing method, more remediation may be needed (like Adversarial Debiasing, Reject Option Classification, etc.).

Fairness Audit Summary (COMPAS Dataset)

After applying a fairness audit using AIF360 on the COMPAS dataset:

- **Accuracy:** 69.16%
- **Disparate Impact:** 0.64
- **Equal Opportunity Difference:** -0.16

Interpretation

- **Disparate Impact** below 0.8 indicates racial bias against African-American individuals.
- **Equal Opportunity Difference** of -0.16 means the model is less likely to predict favorable outcomes for African-Americans even when appropriate.
- This highlights a key ethical issue in criminal justice AI systems: **models trained on biased historical data can perpetuate systemic unfairness**.

Next Steps

To mitigate bias further:

- Use in-processing techniques like `Adversarial Debiasing`.
- Explore post-processing options like `Reject Option Classification`.
- Increase representation or balance in the training dataset.

```
Start coding or generate with AI.
```

```
 Bias Audit Summary: COMPAS Recidivism Risk Scores
```

We conducted a fairness audit on the COMPAS dataset using IBM's AI Fairness 360 toolkit. The dataset includes recidivism risk scores for defendants, with “race” as a key protected attribute.

Initial fairness metrics showed a disparate impact against Black defendants. Specifically, Black individuals were significantly more likely to receive higher risk scores compared to White individuals with similar profiles. This is concerning, as it can influence parole decisions, sentencing, and other legal outcomes.

To address this, we applied the Reweighing pre-processing algorithm, which adjusts instance weights to make the training data more balanced across racial groups. A logistic regression model was then trained on the reweighted dataset.

Post mitigation results showed improvement across multiple fairness metrics:

Disparate Impact moved closer to the ideal value of 1.0.

Equal Opportunity Difference and Average Odds Difference showed reductions, indicating better fairness in true positive and false positive rates across racial groups.

While reweighing reduced some bias, residual disparities remain, suggesting that algorithmic fairness is not just a technical fix but a broader socio-legal issue. Long-term solutions require rethinking the use of historical data, transparency in how scores are used, and robust accountability mechanisms.

In conclusion, the audit demonstrates how fairness toolkits like AIF360 can help developers diagnose and mitigate bias in AI systems, particularly in high stakes domains like criminal justice.

Part 4: Ethical Reflection (5%)

Prompt: Reflect on a personal project (past or future).

How will you ensure it adheres to ethical AI principles?

Ethical Reflection

In a recent project on building an Edge AI model for classifying recyclable waste, I became increasingly aware of how AI can impact different communities and stakeholders. Although the system was designed to promote sustainability, it raised questions about data sourcing, device accessibility, and environmental fairness.

To ensure future projects like this adhere to ethical AI principles, I will follow these key commitments:

1. **Bias Awareness & Mitigation**
    - I will assess training data for underrepresentation (e.g., different waste types or cultural recycling patterns).
    - I'll use fairness metrics and tools like AIF360 to audit models.
2. **Transparency & Explainability**
    - I will clearly document how models make decisions (e.g., which features influence recyclable classifications).
    - For public or government adoption, I will create interpretable dashboards or explainable AI layers.
3. **Data Privacy**

- If edge devices collect any personal data (e.g., location, images), I will anonymize it and follow GDPR principles.
- Where possible, I will ensure models run entirely on-device without cloud-based data sharing.

4. **Inclusiveness**
    - I will test the system in diverse environments — urban and rural — to make sure it performs fairly across different settings.
    - I'll involve end-users (e.g., waste management workers) in the feedback and design loop.
5. **Sustainability**
    - The model will be lightweight to reduce energy consumption and e-waste.
    - I'll prioritize deployment on affordable hardware to reduce barriers in underserved regions.

Ultimately, my goal is to build AI systems that are **not only functional but fair, explainable, and inclusive**, contributing positively to society and the environment.