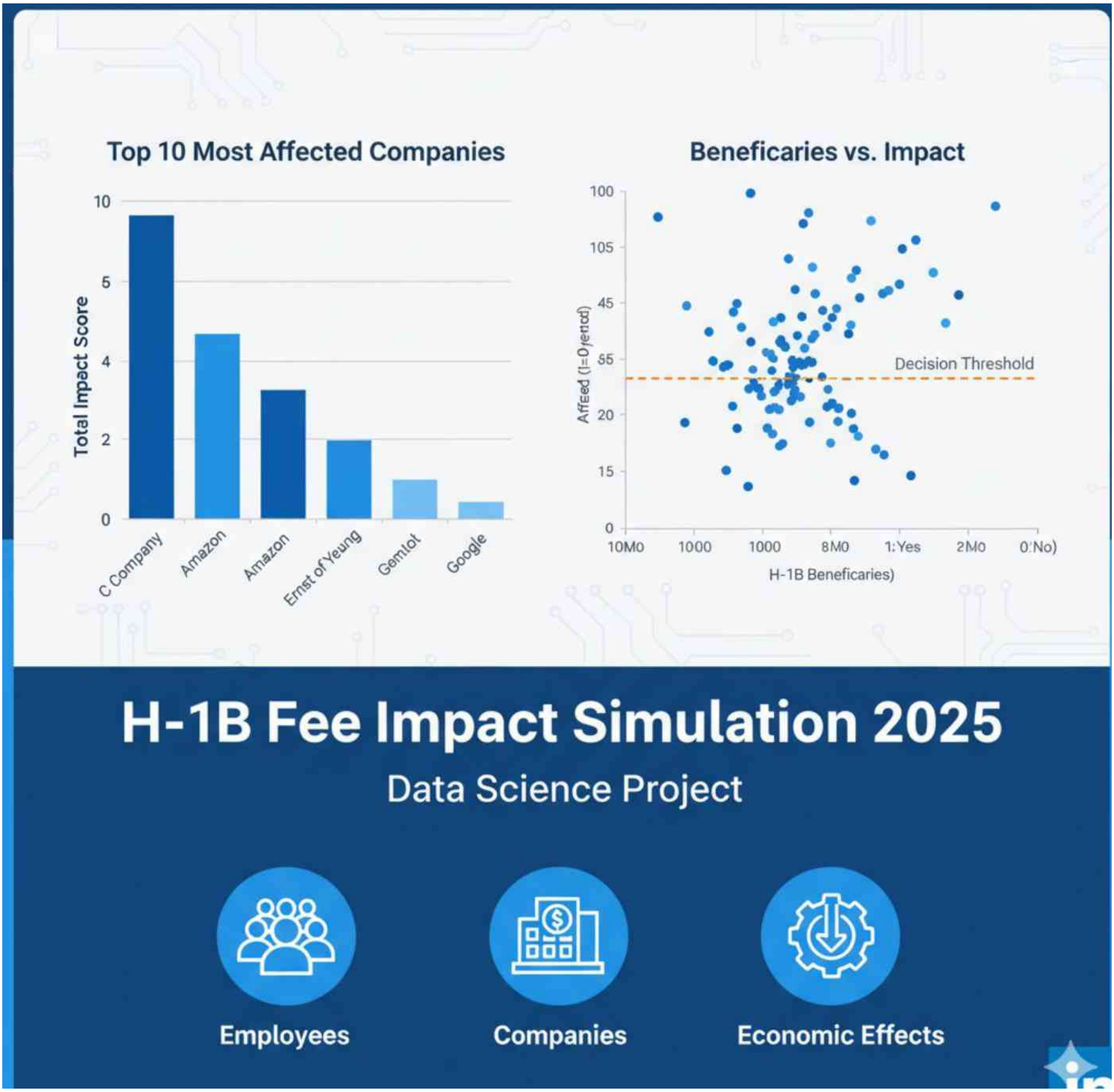

Top 10 Most Affected Companies
Total Impact Score
10
5
4
2
0
C Company
Amazon
Amazon
Ernst of Yeung
Gemlot
Google
Beneficaries vs. Impact
100
105
45
55
20
15
0
Decision Threshold
10M0
1000
1000
8M0
1:Yes
2M0
0:No)
H-1B Beneficaries)
H-1B Fee Impact Simulation 2025
Data Science Project
Employees
Companies
Economic Effects

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

# ---------------------------
# Step 1: Dataset with 50 companies
# ---------------------------
companies_50 = [
    "Amazon","TCS","Ernst & Young","Google","Microsoft",
    "Infosys","Meta","Intel","HCL","Accenture",
    "Wipro","Deloitte","Capgemini","IBM","JPMorgan",
    "Qualcomm","Cisco","Oracle","Salesforce","Uber",
    "Tesla","Snap","Adobe","VMware","ServiceNow",
    "Pinterest","Slack","Dropbox","Zoom","RedHat",
    "Palantir","Square","Stripe","Airbnb","LinkedIn",
    "Spotify","Atlassian","SAP America","SAP Labs 1","SAP Labs 2",
    "SAP Labs 3","SAP Labs 4","SAP Labs 5","SAP Labs 6","SAP Labs 7",
    "SAP Labs 8","SAP Labs 9","SAP Labs 10","SAP Labs 11","SAP Labs
12"
]

beneficiaries_50 = [
    10044,5505,8723,7649,5189,
    4926,5123,3242,3059,2800,
    2500,2300,2100,1900,2440,
    1700,1570,1500,1400,1300,
    1200,1100,1000,950,900,
    850,800,750,700,650,
    600,550,500,450,400,
    350,300,250,200,150,
    100,50,25,10,5,
    3,2,1,0,0
]

# ---------------------------
# Step 1a: Additional 4 companies
# ---------------------------
companies_4 = ['A Company', 'B Company', 'C Company', 'D Company']
beneficiaries_4 = [5000, 1700, 25000, 1100]
employees_total_4 = [15000, 2400, 27000, 1000]

# Convert to DataFrame
df_50 = pd.DataFrame({"Company": companies_50[:50], "Beneficiaries":
beneficiaries_50})
df_4 = pd.DataFrame({"Company": companies_4, "Beneficiaries":
```

```
beneficiaries_4, "Total_Employees": employees_total_4})

# Merge datasets (fill missing Total_Employees for 50-company data
with NaN)
df_50["Total_Employees"] = pd.NA
df = pd.concat([df_50, df_4], ignore_index=True)

# ----------------------------
# Step 2: Create target variable
# >2000 beneficiaries => highly affected (1), else 0
# ----------------------------
df["Affected"] = df["Beneficiaries"].apply(lambda x: 1 if x > 2000
else 0)

# ----------------------------
# Step 3: Train/Test Split
# ----------------------------
X = df[["Beneficiaries"]]
y = df["Affected"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# ----------------------------
# Step 4: Train Random Forest Classifier
# ----------------------------
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# ----------------------------
# Step 5: Evaluate Model
# ----------------------------
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)

print(f"Accuracy: {accuracy*100:.2f}%")
print("Confusion Matrix:")
print(cm)

# ----------------------------
# Step 6: Visualize Before Fee
# ----------------------------
plt.figure(figsize=(12,6))
sns.scatterplot(x="Beneficiaries", y="Affected", data=df, s=100)
plt.axhline(0.5, color='red', linestyle='--', label="Decision
Threshold")
```

```
plt.title("H-1B Beneficiaries vs Highly Affected Companies")
plt.xlabel("Number of H-1B Beneficiaries")
plt.ylabel("Highly Affected (1=Yes, 0=No)")
plt.legend()
plt.show()

# ---------------------------
# Step 7: Simulate Trump $100k Fee (20% reduction)
# ---------------------------
df["Beneficiaries_After_Fee"] = df["Beneficiaries"] * 0.8
df["Affected_After_Fee"] = df["Beneficiaries_After_Fee"].apply(lambda
x: 1 if x > 2000 else 0)

# Predict using trained model
X_new =
df[["Beneficiaries_After_Fee"]].rename(columns={"Beneficiaries_After_F
ee": "Beneficiaries"})
df["Predicted_Affected"] = model.predict(X_new)

# ---------------------------
# Step 8: Encode Real-world Impacts
# ---------------------------
df["Cost_Increase"] = df["Affected_After_Fee"] * 1
df["Layoffs_Risk"] = df["Affected_After_Fee"] * 1
df["Inflation_Contribution"] = df["Affected_After_Fee"] * 0.05
df["Economic_Instability"] = df["Affected_After_Fee"] * 1
df["Labor_Cost_Saving"] = df["Beneficiaries_After_Fee"] * 0.36 *
100000

# Total Impact Score
df["Total_Impact_Score"] = (
    df["Cost_Increase"]*0.3 +
    df["Layoffs_Risk"]*0.3 +
    df["Inflation_Contribution"]*0.2 +
    (df["Labor_Cost_Saving"]/1e6)*0.2
)

# ---------------------------
# Step 9: Show full results
# ---------------------------
print("\nSample of companies after Trump fee simulation:")
print(df[["Company", "Beneficiaries", "Beneficiaries_After_Fee",
"Affected_After_Fee",
          "Predicted_Affected", "Cost_Increase", "Layoffs_Risk",
          "Inflation_Contribution", "Labor_Cost_Saving",
"Total_Impact_Score"]].head(15))

# ---------------------------
```

```
# Step 10: Top 10 Companies Most Affected After Fee
# ---------------------------
top_affected = df.sort_values("Total_Impact_Score",
ascending=False).head(10)
print("\nTop 10 Companies Most Affected After $100k Fee:")
print(top_affected[["Company", "Beneficiaries",
"Beneficiaries_After_Fee", "Affected_After_Fee",
"Total_Impact_Score"]])

# ---------------------------
# Step 11: Visualization After Fee
# ---------------------------
plt.figure(figsize=(12,6))
sns.barplot(x="Company", y="Total_Impact_Score", data=top_affected)
plt.xticks(rotation=90)
plt.title("Top 10 Companies by Total Impact Score Due to $100k H-1B
Fee")
plt.ylabel("Total Impact Score (scaled)")
plt.show()
```

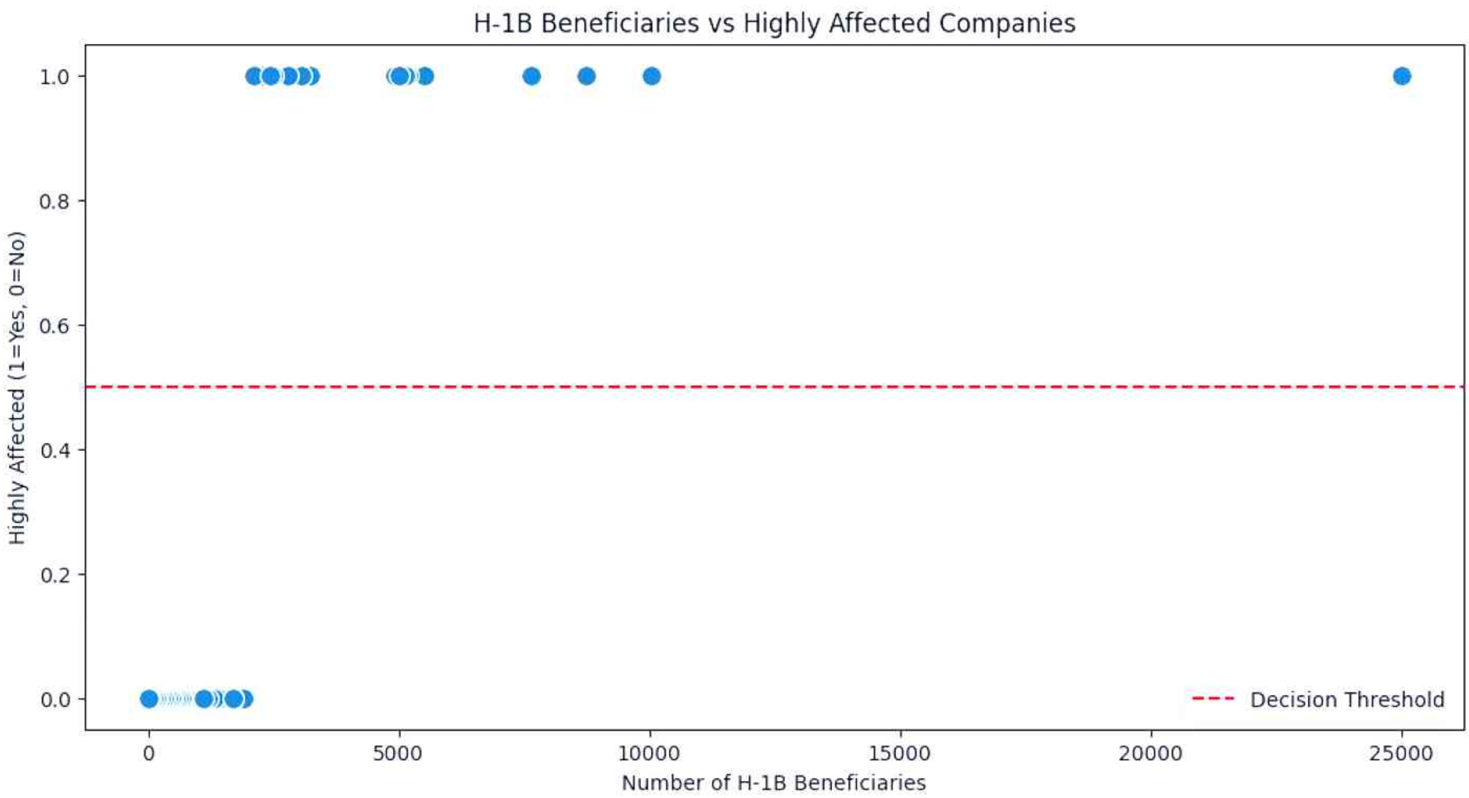
Accuracy: 100.00% Confusion Matrix: [[8 0] [0 3]]
Sample of companies after Trump fee simulation:

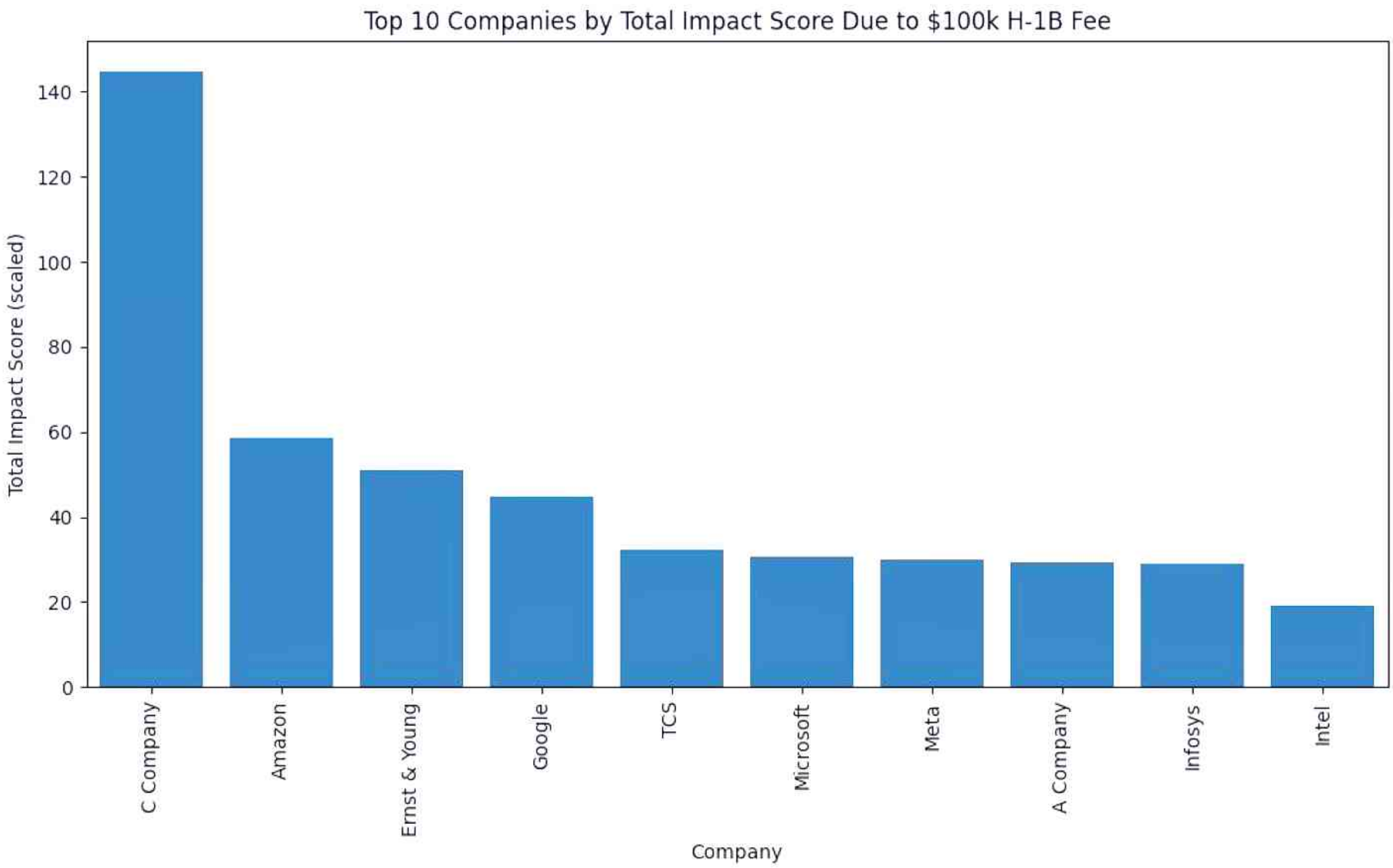
Company	Beneficiaries	Beneficiaries_After_Fee	Affected_After_Fee	Predicted_Affected	Cost_Increase	Layoffs_Risk	Inflation_Contribution	Labor_Cost_Saving	Total_Impact_Score
Amazon	10044	8035.2	1	1	1	1	0.05	289267200.0	58.46344
TCS	5505	4404.0	1	1	1	1	0.05	158544000.0	32.31880
Ernst & Young	8723	6978.4	1	1	1	1	0.05	251222400.0	50.85448
Google	7649	6119.2	1	1	1	1	0.05	220291200.0	44.66824
Microsoft	5189	4151.2	1	1	1	1	0.05	149443200.0	30.49864
Infosys	4926	3940.8	1	1	1	1	0.05	141868800.0	28.98376
Meta	5123	4098.4	1	1	1	1	0.05	147542400.0	30.11848
Intel	3242	2593.6	1	1	1	1	0.05	93369600.0	19.28392
HCL	3059	2447.2	1	1	1	1	0.05	88099200.0	18.22984
Accenture	2800	2240.0	1	1	1	1	0.05	80640000.0	16.73800
Wipro	2500	2000.0	0	0	0	0	0.00	72000000.0	14.40000
Deloitte	2300	1840.0	0	0	0	0	0.00	66240000.0	13.24800
Capgemini	2100	1680.0	0	0	0	0	0.00	60480000.0	12.09600
IBM	1900	1520.0	0	0	0	0	0.00	54720000.0	10.94400
JPMorgan	2440	1952.0	0	0	0	0	0.00	70272000.0	14.05440

Top 10 Companies Most Affected After $100k Fee:

Company	Beneficiaries	Beneficiaries_After_Fee	Affected_After_Fee	Total_Impact_Score
C Company	25000	20000.0	1	144.61000
Amazon	10044	8035.2	1	58.46344
Ernst & Young	8723	6978.4	1	50.85448
Google	7649	6119.2	1	44.66824
TCS	5505	4404.0	1	32.31880
Microsoft	5189	4151.2	1	30.49864
Meta	5123	4098.4	1	30.11848
A Company	5000	4000.0	1	29.41000

Company	Beneficiaries	Beneficiaries_After_Fee	Affected_After_Fee	Total_Impact_Score
Infosys	4926	3940.8	1	28.98376
Intel	3242	2593.6	1	19.28392


H-1B Beneficiaries vs Highly Affected Companies
1.0
0.8
0.6
0.4
0.2
0.0
Highly Affected (1=Yes, 0=No)
Decision Threshold
0
5000
10000
15000
20000
25000
Number of H-1B Beneficiaries


Top 10 Companies by Total Impact Score Due to $100k H-1B Fee
Total Impact Score (scaled)
0
20
40
60
80
100
120
140
C Company
Amazon
Ernst & Young
Google
TCS
Microsoft
Meta
A Company
Infosys
Intel
Company