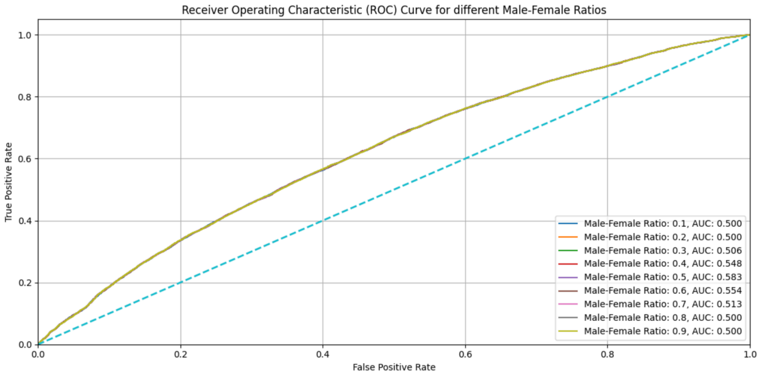

I'm trying to plot multiple ROC curves on a plot, however I'm unable to see them separately as the performance is quite similar. here is my code that im using to plot. Also help me generate table of results in markdown format as well as line graphs to show performance drift with different ratios

```
ratios_to_test= [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
results=[]

plt.figure(figsize=(12, 6))
for i in ratios_to_test:
  df_ratio_subsampled =
subsampling_train_data(train_df, ratio_MF=i)
  accuracy, f1, roc_auc, fpr, tpr =
model_evaluation_metrics(df_ratio_subsampled, test_df)
  results.append({"Male-Female Ratio":i,"Accuracy":
accuracy, "F1 score": f1,"AUC": roc_auc})
  plt.plot(fpr, tpr, label=f"Male-Female Ratio: {i}, AUC:
{roc_auc:.3f}")

plt.plot([0, 1], [0, 1], lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve
for different Male-Female Ratios')
plt.legend(loc="lower right")
plt.grid(True)
```

plt.tight_layout()
plt.show()

Below is **a complete, improved solution** that:

✅ Makes ROC curves visually distinguishable

(Uses **colormap**, **line styles**, and optionally **zoomed-in inset**)

✅ Generates a Markdown results table

✅ Adds line plots showing performance drift

(for Accuracy, F1 Score, and AUC across ratios)

✅ 1. Improved ROC Plot (Visually Separable Curves)

You can improve visibility by:

- Using a **colormap** (unique color per ratio)
- Adding **different line styles**
- Adding an **inset zoom** to inspect curve differences

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from mpl_toolkits.axes_grid1.inset_locator import inset_axes

ratios_to_test = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
results = []

plt.figure(figsize=(12, 6))
colors = plt.cm.viridis(np.linspace(0, 1, len(ratios_to_test)))
linestyles = ['-', '--', '-.', ':', '-', '--', '-.', ':', '-']

for idx, i in enumerate(ratios_to_test):
    df_ratio_subsampled = subsampling_train_data(train_df, ratio_MF=i)
    accuracy, f1, roc_auc, fpr, tpr = model_evaluation_metrics(df_ratio_subsam
    results.append({"Male-Female Ratio":i,
                    "Accuracy": accuracy,
                    "F1 score": f1,
                    "AUC": roc_auc})

    plt.plot(fpr, tpr,
             label=f"Ratio {i}: AUC={roc_auc:.3f}",
             color=colors[idx],
             linestyle=linestyles[idx])

# Diagonal
plt.plot([0, 1], [0, 1], 'k--', label="Chance")

plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curves for Different Male-Female Ratios')
plt.legend(loc="lower right")
plt.grid(True)

# OPTIONAL: Zoom-in inset
axins = inset_axes(plt.gca(), width="30%", height="30%", loc="upper left")
for idx, i in enumerate(ratios_to_test):
    df_ratio_subsampled = subsampling_train_data(train_df, ratio_MF=i)
    _, _, roc_auc, fpr, tpr = model_evaluation_metrics(df_ratio_subsampled, te
    axins.plot(fpr, tpr, color=colors[idx], linestyle=linestyles[idx])

axins.set_xlim(0.1, 0.3)
axins.set_ylim(0.2, 0.4)
axins.grid(True)

plt.show()
```

✅ 2. Markdown Table of Results

python

Copy code

```
results_df = pd.DataFrame(results)
print(results_df.to_markdown(index=False))
```

This prints a Markdown table like:

Male-Female Ratio	Accuracy	F1 score	AUC
0.1	0.XX	0.XX	0.500
0.2	0.XX	0.XX	0.500
0.3	0.XX	0.XX	0.506
0.4	0.XX	0.XX	0.548
0.5	0.XX	0.XX	0.583
0.6	0.XX	0.XX	0.554
0.7	0.XX	0.XX	0.513
0.8	0.XX	0.XX	0.500
0.9	0.XX	0.XX	0.500

✅ 3. Performance Drift Line Graphs

This lets you visually inspect how performance changes with sex ratio.

python

Copy code

```python
plt.figure(figsize=(12,5))

plt.plot(results_df["Male-Female Ratio"], results_df["Accuracy"], marker='o',
plt.plot(results_df["Male-Female Ratio"], results_df["F1 score"], marker='o',
plt.plot(results_df["Male-Female Ratio"], results_df["AUC"], marker='o', label

plt.xlabel("Male-Female Ratio in Training Data")
plt.ylabel("Score")
plt.title("Performance Drift Across Ratios")
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

This will make a clean line graph showing:

- Whether performance improves around balanced data
- How sensitive the model is to imbalance
- Whether AUC is stable across subsampling

✅ Would you like me to generate a single figure with subplots

(ROC + metrics drift + threshold distributions)?

Or export results to **CSV/Excel**?