

```
[1]: import os
import glob
import joblib
import pandas as pd
from sklearn.metrics import average_precision_score

# Paths
model_folder = "champion_packages"
data_folder = "market_shock_synthetic_datasets"

# Load all market shock synthetic datasets with baseline
def load_all_datasets_with_baseline(folder):
    files = sorted(glob.glob(os.path.join(folder, "*.csv")))
    datasets = []
    for f in files:
        df = pd.read_csv(f)
        X = df.drop("rare_event", axis=1)
        y = df["rare_event"]
        baseline_pr_auc = y.mean() # Random guess baseline = positive class rate
        datasets.append((os.path.basename(f), X, y, baseline_pr_auc))
    return datasets

datasets = load_all_datasets_with_baseline(data_folder)

# Load all trained champion models
model_files = sorted(glob.glob(os.path.join(model_folder, "*.pkl")))
models = []
for f in model_files:
    if "_meta" not in f:
        (os.path.basename(f).replace(".pkl", ""), joblib.load(f))

# Evaluate all models on all datasets
results = []

for model_name, model in models:
    for dataset_name, X, y, baseline in datasets:
        try:
            y_proba = model.predict_proba(X)[:, 1]
        except AttributeError:
            y_proba = model.decision_function(X)

        pr_auc = average_precision_score(y, y_proba)
        lift = pr_auc - baseline

        results.append((
            "Model": model_name,
            "Test Dataset": dataset_name.replace(".csv", ""),
            "PR AUC": round(pr_auc, 3),
            "Baseline PR AUC": round(baseline, 3),
            "Lift Over Baseline": round(lift, 3)
        ))

# Create summary DataFrame
results_df = pd.DataFrame(results)
results_df = results_df.sort_values(by=["Test Dataset", "PR AUC"], ascending=[True, False])

# Optional save
results_df.to_csv("champion_cross_eval_results.csv", index=False)

# View
print(results_df)
```

```
56 winner_noisy_overlap_w2_n20_d0.csv market_shock_sim 0.178 \
0 winner_baseline_easy_w5_n5_d0.csv market_shock_sim 0.175
14 winner_high_drift_w3_n10_d60.csv market_shock_sim 0.173
42 winner_mixed_realistic_w4_n15_d30.csv market_shock_sim 0.172
28 winner_imbalanced_sparse_w0_n10_d10.csv market_shock_sim 0.170
...
55 winner_mixed_realistic_w4_n15_d30.csv stock_prediction_clean 0.381
13 winner_baseline_easy_w5_n5_d0.csv stock_prediction_clean 0.348
27 winner_high_drift_w3_n10_d60.csv stock_prediction_clean 0.348
69 winner_noisy_overlap_w2_n20_d0.csv stock_prediction_clean 0.318
41 winner_imbalanced_sparse_w0_n10_d10.csv stock_prediction_clean 0.317

Baseline PR AUC Lift Over Baseline
56 0.165 0.013
0 0.165 0.010
14 0.165 0.008
42 0.165 0.007
28 0.165 0.005
...
55 0.502 -0.122
13 0.502 -0.155
27 0.502 -0.155
69 0.502 -0.184
41 0.502 -0.185

[70 rows x 5 columns]
```

[3]: results_df

	Model	Test Dataset	PR AUC	Baseline PR AUC	Lift Over Baseline
56	winner_noisy_overlap_w2_n20_d0.csv	market_shock_sim	0.178	0.165	0.013
0	winner_baseline_easy_w5_n5_d0.csv	market_shock_sim	0.175	0.165	0.010
14	winner_high_drift_w3_n10_d60.csv	market_shock_sim	0.173	0.165	0.008
42	winner_mixed_realistic_w4_n15_d30.csv	market_shock_sim	0.172	0.165	0.007
28	winner_imbalanced_sparse_w0_n10_d10.csv	market_shock_sim	0.170	0.165	0.005
...
55	winner_mixed_realistic_w4_n15_d30.csv	stock_prediction_clean	0.381	0.502	-0.122
13	winner_baseline_easy_w5_n5_d0.csv	stock_prediction_clean	0.348	0.502	-0.155
27	winner_high_drift_w3_n10_d60.csv	stock_prediction_clean	0.348	0.502	-0.155
69	winner_noisy_overlap_w2_n20_d0.csv	stock_prediction_clean	0.318	0.502	-0.184
41	winner_imbalanced_sparse_w0_n10_d10.csv	stock_prediction_clean	0.317	0.502	-0.185

70 rows x 5 columns

[5]: ranked_df = results_df.sort_values(by=["Test Dataset", "PR AUC"], ascending=[True, False])

[7]: ranked_df["Best Performer"] = ranked_df.groupby("Test Dataset")["PR AUC"].transform("max") == ranked_df["PR AUC"]
ranked_df["Significant Lift"] = ranked_df["Lift Over Baseline"].map({True: "🏆", False: "❌"})

[9]: ranked_df["Significant Lift"] = ranked_df["Lift Over Baseline"] > 0.05).map({True: "🏆", False: "❌"})

[11]: ranked_df

	Model	Test Dataset	PR AUC	Baseline PR AUC	Lift Over Baseline	Best Performer	Significant Lift
56	winner_noisy_overlap_w2_n20_d0.csv	market_shock_sim	0.178	0.165	0.013	🏆	❌
0	winner_baseline_easy_w5_n5_d0.csv	market_shock_sim	0.175	0.165	0.010		❌
14	winner_high_drift_w3_n10_d60.csv	market_shock_sim	0.173	0.165	0.008		❌
42	winner_mixed_realistic_w4_n15_d30.csv	market_shock_sim	0.172	0.165	0.007		❌
28	winner_imbalanced_sparse_w0_n10_d10.csv	market_shock_sim	0.170	0.165	0.005		❌
...
55	winner_mixed_realistic_w4_n15_d30.csv	stock_prediction_clean	0.381	0.502	-0.122	🏆	❌
13	winner_baseline_easy_w5_n5_d0.csv	stock_prediction_clean	0.348	0.502	-0.155		❌
27	winner_high_drift_w3_n10_d60.csv	stock_prediction_clean	0.348	0.502	-0.155		❌
69	winner_noisy_overlap_w2_n20_d0.csv	stock_prediction_clean	0.318	0.502	-0.184		❌

70 rows x 7 columns

[]:	
[]:	