[5]:
```python
# Full Champion Evaluation Across All Market Shock Datasets

import os
import glob
import joblib
import pandas as pd
import numpy as np
from sklearn.metrics import average_precision_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# === Load Champion Models ===
model_paths = sorted(glob.glob("champion_packages/*.pkl"))
champion_models = [joblib.load(p) for p in model_paths if "_meta" not in p]
champion_names = [os.path.basename(p).replace(".pkl", "") for p in model_paths if "_meta" not in p]

# === Load All Market Shock Datasets ===
data_paths = sorted(glob.glob("market_shock_synthetic_datasets/*.csv"))

# === Result Collector ===
all_results = []

# Create output folder for champion stacks
os.makedirs("champion_stacks", exist_ok=True)

# === Evaluation Loop ===
for test_data_path in data_paths:
    dataset_name = os.path.basename(test_data_path).replace(".csv", "")
    df = pd.read_csv(test_data_path)
    X = df.drop("rare_event", axis=1)
    y = df["rare_event"]
    baseline = y.mean()

    # --- Individual Models ---
    individual_scores = []
    for model in champion_models:
        try:
            y_score = model.predict_proba(X)[:, 1]
        except AttributeError:
            y_score = model.decision_function(X)
        individual_scores.append(average_precision_score(y, y_score))

    best_individual_pr_auc = max(individual_scores)

    # --- Simple Ensemble ---
    def ensemble_predict_proba(models, X):
        preds = []
        for model in models:
            try:
                preds.append(model.predict_proba(X)[:, 1])
            except AttributeError:
                preds.append(model.decision_function(X))
        return np.mean(preds, axis=0)

    y_ensemble = ensemble_predict_proba(champion_models, X)
    ensemble_pr_auc = average_precision_score(y, y_ensemble)

    # --- Stacking ---
    meta_features = []
    for model in champion_models:
        try:
            meta_features.append(model.predict_proba(X)[:, 1])
        except AttributeError:
            meta_features.append(model.decision_function(X))
    meta_X = np.vstack(meta_features).T

    X_meta_train, X_meta_test, y_meta_train, y_meta_test = train_test_split(
        meta_X, y, test_size=0.3, stratify=y, random_state=42
    )
    meta_model = LogisticRegression(max_iter=1000).fit(X_meta_train, y_meta_train)
    meta_preds = meta_model.predict_proba(X_meta_test)[:, 1]
    meta_pr_auc = average_precision_score(y_meta_test, meta_preds)

    # Save meta-model
    joblib.dump(meta_model, f"champion_stacks/{dataset_name}_stacked.pkl")

    # --- Record All Results ---
    all_results.extend([
        {
            "Dataset": dataset_name,
            "Approach": "Baseline Only",
            "PR AUC": round(baseline, 3),
            "Lift Over Baseline": 0.0,
            "Notes": "No model, just prevalence"
        },
        {
            "Dataset": dataset_name,
            "Approach": "Best Individual Champion",
            "PR AUC": round(best_individual_pr_auc, 3),
            "Lift Over Baseline": round(best_individual_pr_auc - baseline, 3),
            "Notes": "Best PR AUC among individual champions"
        },
        {
            "Dataset": dataset_name,
            "Approach": "Simple Average Ensemble",
            "PR AUC": round(ensemble_pr_auc, 3),
            "Lift Over Baseline": round(ensemble_pr_auc - baseline, 3),
            "Notes": "Average probs across all models"
        },
        {
            "Dataset": dataset_name,
            "Approach": "Stacked Logistic Regression",
            "PR AUC": round(meta_pr_auc, 3),
            "Lift Over Baseline": round(meta_pr_auc - baseline, 3),
            "Notes": "Meta-model trained on model outputs"
        },
    ])

# === Final Leaderboard ===
leaderboard_df = pd.DataFrame(all_results)
leaderboard_df.to_csv("ensemble_lift_leaderboard.csv", index=False)
print(leaderboard_df)
```

```
                Dataset                    Approach  PR AUC  \
0        market_shock_sim               Baseline Only   0.165
1        market_shock_sim     Best Individual Champion   0.178
2        market_shock_sim      Simple Average Ensemble   0.177
3        market_shock_sim  Stacked Logistic Regression   0.172
4     marketshock_easy_s1               Baseline Only   0.141
5     marketshock_easy_s1     Best Individual Champion   0.306
6     marketshock_easy_s1      Simple Average Ensemble   0.207
7     marketshock_easy_s1  Stacked Logistic Regression   0.425
8     marketshock_easy_s2               Baseline Only   0.136
9     marketshock_easy_s2     Best Individual Champion   0.191
10    marketshock_easy_s2      Simple Average Ensemble   0.107
11    marketshock_easy_s2  Stacked Logistic Regression   0.455
12    marketshock_easy_s3               Baseline Only   0.133
13    marketshock_easy_s3     Best Individual Champion   0.149
14    marketshock_easy_s3      Simple Average Ensemble   0.098
15    marketshock_easy_s3  Stacked Logistic Regression   0.638
16  marketshock_extreme_s1               Baseline Only   0.261
```

[7]:
```python
leaderboard_df
```

[7]:

	Dataset	Approach	PR AUC	Lift Over Baseline	Notes
0	market_shock_sim	Baseline Only	0.165	0.000	No model, just prevalence
1	market_shock_sim	Best Individual Champion	0.178	0.013	Best PR AUC among individual champions
2	market_shock_sim	Simple Average Ensemble	0.177	0.012	Average probs across all models
3	market_shock_sim	Stacked Logistic Regression	0.172	0.007	Meta-model trained on model outputs

4	marketshock_easy_s1	Baseline Only	0.141	0.000	No model, just prevalence
5	marketshock_easy_s1	Best Individual Champion	0.306	0.164	Best PR AUC among individual champions
6	marketshock_easy_s1	Simple Average Ensemble	0.207	0.066	Average probs across all models
7	marketshock_easy_s1	Stacked Logistic Regression	0.425	0.283	Meta-model trained on model outputs
8	marketshock_easy_s2	Baseline Only	0.136	0.000	No model, just prevalence
9	marketshock_easy_s2	Best Individual Champion	0.191	0.055	Best PR AUC among individual champions
10	marketshock_easy_s2	Simple Average Ensemble	0.107	-0.029	Average probs across all models
11	marketshock_easy_s2	Stacked Logistic Regression	0.455	0.319	Meta-model trained on model outputs
12	marketshock_easy_s3	Baseline Only	0.133	0.000	No model, just prevalence
13	marketshock_easy_s3	Best Individual Champion	0.149	0.016	Best PR AUC among individual champions
14	marketshock_easy_s3	Simple Average Ensemble	0.098	-0.035	Average probs across all models
15	marketshock_easy_s3	Stacked Logistic Regression	0.638	0.505	Meta-model trained on model outputs
16	marketshock_extreme_s1	Baseline Only	0.261	0.000	No model, just prevalence
17	marketshock_extreme_s1	Best Individual Champion	0.261	0.000	Best PR AUC among individual champions
18	marketshock_extreme_s1	Simple Average Ensemble	0.258	-0.003	Average probs across all models
19	marketshock_extreme_s1	Stacked Logistic Regression	0.257	-0.004	Meta-model trained on model outputs
20	marketshock_extreme_s2	Baseline Only	0.255	0.000	No model, just prevalence
21	marketshock_extreme_s2	Best Individual Champion	0.257	0.002	Best PR AUC among individual champions
22	marketshock_extreme_s2	Simple Average Ensemble	0.247	-0.008	Average probs across all models
23	marketshock_extreme_s2	Stacked Logistic Regression	0.274	0.019	Meta-model trained on model outputs
24	marketshock_extreme_s3	Baseline Only	0.264	0.000	No model, just prevalence
25	marketshock_extreme_s3	Best Individual Champion	0.270	0.006	Best PR AUC among individual champions
26	marketshock_extreme_s3	Simple Average Ensemble	0.271	0.007	Average probs across all models
27	marketshock_extreme_s3	Stacked Logistic Regression	0.260	-0.004	Meta-model trained on model outputs
28	marketshock_hard_s1	Baseline Only	0.177	0.000	No model, just prevalence
29	marketshock_hard_s1	Best Individual Champion	0.191	0.014	Best PR AUC among individual champions
30	marketshock_hard_s1	Simple Average Ensemble	0.185	0.008	Average probs across all models
31	marketshock_hard_s1	Stacked Logistic Regression	0.191	0.014	Meta-model trained on model outputs
32	marketshock_hard_s2	Baseline Only	0.164	0.000	No model, just prevalence
33	marketshock_hard_s2	Best Individual Champion	0.167	0.003	Best PR AUC among individual champions
34	marketshock_hard_s2	Simple Average Ensemble	0.162	-0.001	Average probs across all models
35	marketshock_hard_s2	Stacked Logistic Regression	0.168	0.004	Meta-model trained on model outputs
36	marketshock_hard_s3	Baseline Only	0.155	0.000	No model, just prevalence
37	marketshock_hard_s3	Best Individual Champion	0.164	0.010	Best PR AUC among individual champions
38	marketshock_hard_s3	Simple Average Ensemble	0.157	0.003	Average probs across all models
39	marketshock_hard_s3	Stacked Logistic Regression	0.149	-0.006	Meta-model trained on model outputs
40	marketshock_medium_s1	Baseline Only	0.141	0.000	No model, just prevalence
41	marketshock_medium_s1	Best Individual Champion	0.146	0.005	Best PR AUC among individual champions
42	marketshock_medium_s1	Simple Average Ensemble	0.128	-0.012	Average probs across all models
43	marketshock_medium_s1	Stacked Logistic Regression	0.271	0.130	Meta-model trained on model outputs
44	marketshock_medium_s2	Baseline Only	0.144	0.000	No model, just prevalence
45	marketshock_medium_s2	Best Individual Champion	0.219	0.075	Best PR AUC among individual champions
46	marketshock_medium_s2	Simple Average Ensemble	0.151	0.008	Average probs across all models
47	marketshock_medium_s2	Stacked Logistic Regression	0.251	0.107	Meta-model trained on model outputs
48	marketshock_medium_s3	Baseline Only	0.140	0.000	No model, just prevalence
49	marketshock_medium_s3	Best Individual Champion	0.237	0.097	Best PR AUC among individual champions
50	marketshock_medium_s3	Simple Average Ensemble	0.195	0.055	Average probs across all models
51	marketshock_medium_s3	Stacked Logistic Regression	0.255	0.114	Meta-model trained on model outputs
52	stock_prediction_clean	Baseline Only	0.502	0.000	No model, just prevalence
53	stock_prediction_clean	Best Individual Champion	0.381	-0.122	Best PR AUC among individual champions
54	stock_prediction_clean	Simple Average Ensemble	0.327	-0.176	Average probs across all models
55	stock_prediction_clean	Stacked Logistic Regression	0.919	0.416	Meta-model trained on model outputs

[ ]: