05_Feature_Engineering_Refinement

November 25, 2025

[20]:

```
import sys
import time
import joblib
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from pathlib import Path

# Optimization
import optuna

# Modeling
from sklearn.linear_model import LogisticRegression, SGDClassifier
from sklearn.svm import LinearSVC
from sklearn.naive_bayes import MultinomialNB
from lightgbm import LGBMClassifier

# Feature Selection & Ensembling
from sklearn.feature_selection import SelectKBest, chi2
from sklearn.calibration import CalibratedClassifierCV
from sklearn.ensemble import StackingClassifier, VotingClassifier
from sklearn.model_selection import StratifiedKFold, cross_val_predict,␣
 ↪cross_val_score
from sklearn.metrics import f1_score, classification_report, confusion_matrix

# Constants
RANDOM_STATE = 42
N_TRIALS = 30  # Number of Optuna trials per model
N_JOBS = -1    # Use all cores

# Paths
DATA_DIR = Path('../data')
FEATURES_DIR = Path('features')
MODELS_DIR = Path('models')
RESULTS_DIR = Path('results')
RESULTS_DIR.mkdir(exist_ok=True)
```

```
# Plotting style
plt.style.use('seaborn-v0_8-darkgrid')

print(f"Python Version: {sys.version}")
print(f"Optuna Version: {optuna.__version__}")
print("Setup Complete. Ready for Step 5.")
```

```
Python Version: 3.11.14 (main, Oct 31 2025, 23:04:14) [Clang 21.1.4 ]
Optuna Version: 4.6.0
Setup Complete. Ready for Step 5.
```

[21]:

```
# 5.1 Load Data and Features

# Load Dataframes
train_df = pd.read_csv(DATA_DIR / 'train.csv')
val_df = pd.read_csv(DATA_DIR / 'val.csv')
y_train = train_df['label'].values
y_val = val_df['label'].values

# Load the Best Feature Matrix from Step 3 (Hybrid: Word + Char)
# We use this for Linear Models (SGD, SVM, NB) which handle high-dim sparse␣
 ↪data well
import scipy.sparse as sp_sparse
X_train_hybrid = sp_sparse.load_npz(FEATURES_DIR / 'X_train_hybrid.npz')
X_val_hybrid = sp_sparse.load_npz(FEATURES_DIR / 'X_val_hybrid.npz')

print(f"Training Data Shape: {X_train_hybrid.shape}")
print(f"Validation Data Shape: {X_val_hybrid.shape}")
print(f"Labels Shape: {y_train.shape}")
```

```
Training Data Shape: (102000, 100000)
Validation Data Shape: (18000, 100000)
Labels Shape: (102000,)
```

[22]:

```
# 5.2 Feature Selection for Tree-Based Models

# We reduce dimensionality for LightGBM to improve training speed and␣
 ↪performance
# Linear models will keep using the full 100k set
TARGET_FEATURES = 5000

print(f"Selecting top {TARGET_FEATURES} features using Chi2 statistics...")
start_time = time.time()

selector = SelectKBest(chi2, k=TARGET_FEATURES)
X_train_selected = selector.fit_transform(X_train_hybrid, y_train)
X_val_selected = selector.transform(X_val_hybrid)
```

```
print(f"Selected Train Shape: {X_train_selected.shape}")
print(f"Selected Val Shape: {X_val_selected.shape}")
print(f"Selection Time: {time.time() - start_time:.2f}s")

# Save the selector for inference pipeline
joblib.dump(selector, MODELS_DIR / 'step5_chi2_selector.pkl')
print("Feature selector saved.")
```

```
Selecting top 5000 features using Chi2 statistics…
Selected Train Shape: (102000, 5000)
Selected Val Shape: (18000, 5000)
Selection Time: 0.90s
Feature selector saved.
```

[23]:

```
# 5.3 Define Bayesian Optimization Objectives

def objective_sgd(trial):
    """Optimize SGDClassifier (Logistic Regression / SVM approximation)"""
    params = {
        'alpha': trial.suggest_float('alpha', 1e-6, 1e-1, log=True),
        'penalty': trial.suggest_categorical('penalty', ['l2', 'l1',␣
 ↪'elasticnet']),
        'loss': trial.suggest_categorical('loss', ['modified_huber',␣
 ↪'log_loss']),
        # specific to elasticnet
        'l1_ratio': trial.suggest_float('l1_ratio', 0.05, 0.95) if trial.params.
 ↪get('penalty') == 'elasticnet' else 0.15
    }

    model = SGDClassifier(
        max_iter=1000,
        random_state=RANDOM_STATE,
        n_jobs=N_JOBS,
        early_stopping=True,
        **params
    )

    # 5-fold CV on full hybrid features
    cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=RANDOM_STATE)
    score = cross_val_score(model, X_train_hybrid, y_train, cv=cv,␣
 ↪scoring='f1_macro', n_jobs=N_JOBS)
    return score.mean()

def objective_lgbm(trial):
    """Optimize LightGBM on Reduced Feature Set"""
    params = {
        'n_estimators': trial.suggest_int('n_estimators', 100, 300),
        'learning_rate': trial.suggest_float('learning_rate', 0.05, 0.3),
        'num_leaves': trial.suggest_int('num_leaves', 20, 60),
        'max_depth': trial.suggest_int('max_depth', 3, 10),
        'subsample': trial.suggest_float('subsample', 0.6, 1.0),
        'colsample_bytree': trial.suggest_float('colsample_bytree', 0.6, 1.0),
        # 'device': 'gpu',
        # 'gpu_platform_id': 0,
        # 'gpu_device_id': 0
    }

    model = LGBMClassifier(
        random_state=RANDOM_STATE,
        n_jobs=-1, # Limit threads per model to avoid contention
        verbose=-1,
        **params
    )

    # Use SELECTED features (5K) for speed and stability
    cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=RANDOM_STATE)
    score = cross_val_score(model, X_train_selected, y_train, cv=cv,␣
 ↪scoring='f1_macro', n_jobs=1)
    return score.mean()

def objective_nb(trial):
    """Optimize Multinomial Naive Bayes"""
    alpha = trial.suggest_float('alpha', 1e-3, 10.0, log=True)
    model = MultinomialNB(alpha=alpha)

    cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=RANDOM_STATE)
    score = cross_val_score(model, X_train_hybrid, y_train, cv=cv,␣
 ↪scoring='f1_macro', n_jobs=N_JOBS)
    return score.mean()

def objective_svc(trial):
    """Optimize LinearSVC"""
    C = trial.suggest_float('C', 1e-2, 10.0, log=True)
    # LinearSVC doesn't support n_jobs, so we parallelize via cross_val_score
    model = LinearSVC(C=C, max_iter=2000, random_state=RANDOM_STATE,␣
 ↪dual='auto')

    cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=RANDOM_STATE)
    score = cross_val_score(model, X_train_hybrid, y_train, cv=cv,␣
 ↪scoring='f1_macro', n_jobs=N_JOBS)
    return score.mean()

print("Optimization objectives defined.")
```

```
Optimization objectives defined.
```

[24]:

```python
# 5.4 Execute Optimization Studies

optuna.logging.set_verbosity(optuna.logging.INFO)

studies = {}
best_params = {}

N_TRIALS_LGBM = 15

# List of models to tune
tasks = [
    ("SGD", objective_sgd),
    ("NaiveBayes", objective_nb),
    ("LinearSVC", objective_svc),
    ("LightGBM", objective_lgbm)
]

print("="*80)
print(f"STARTING HYPERPARAMETER TUNING ({N_TRIALS} trials per model)")
print("="*80)

for name, objective in tasks:
    trials = N_TRIALS_LGBM if name == "LightGBM" else N_TRIALS
    print(f"\nRunning optimization for {name}...")
    start_ts = time.time()

    study = optuna.create_study(direction='maximize')
    study.optimize(objective, n_trials=trials)

    studies[name] = study
    best_params[name] = study.best_params

    print(f"  Best F1: {study.best_value:.4f}")
    print(f"  Best Params: {study.best_params}")
    print(f"  Time: {time.time() - start_ts:.2f}s")

# Save best params
joblib.dump(best_params, RESULTS_DIR / 'step5_best_params.pkl')
print("\nTuning Complete. Parameters saved.")
```

```
[I 2025-11-25 11:09:02,126] A new study created in memory with name: no-name-
bc38e485-0518-4773-9825-e455c8eccd04

================================================================================
STARTING HYPERPARAMETER TUNING (30 trials per model)
================================================================================
Running optimization for SGD…
```

```
[I 2025-11-25 11:09:05,476] Trial 0 finished with value: 0.9159441294093913 and
parameters: {'alpha': 2.040934238948351e-05, 'penalty': 'l2', 'loss':
'log_loss'}. Best is trial 0 with value: 0.9159441294093913.
[I 2025-11-25 11:09:08,335] Trial 1 finished with value: 0.9068867092510778 and
parameters: {'alpha': 2.1926871089558056e-06, 'penalty': 'l2', 'loss':
'modified_huber'}. Best is trial 0 with value: 0.9159441294093913.
[I 2025-11-25 11:09:12,650] Trial 2 finished with value: 0.9019489904516307 and
parameters: {'alpha': 2.4836615947747726e-05, 'penalty': 'l1', 'loss':
'log_loss'}. Best is trial 0 with value: 0.9159441294093913.
[I 2025-11-25 11:09:17,369] Trial 3 finished with value: 0.8997832815652677 and
parameters: {'alpha': 0.0001127214829460273, 'penalty': 'l1', 'loss':
'modified_huber'}. Best is trial 0 with value: 0.9159441294093913.
[I 2025-11-25 11:09:21,611] Trial 4 finished with value: 0.8460481863022948 and
parameters: {'alpha': 0.0001674910901747046, 'penalty': 'l1', 'loss':
'log_loss'}. Best is trial 0 with value: 0.9159441294093913.
[I 2025-11-25 11:09:25,551] Trial 5 finished with value: 0.16328954357674233 and
parameters: {'alpha': 0.01599510270299278, 'penalty': 'elasticnet', 'loss':
'modified_huber', 'l1_ratio': 0.6405376661943031}. Best is trial 0 with value:
0.9159441294093913.
[I 2025-11-25 11:09:29,021] Trial 6 finished with value: 0.1 and parameters:
{'alpha': 0.015054497079254185, 'penalty': 'l1', 'loss': 'log_loss'}. Best is
trial 0 with value: 0.9159441294093913.
[I 2025-11-25 11:09:31,850] Trial 7 finished with value: 0.9177503796669544 and
parameters: {'alpha': 1.3999675521846313e-05, 'penalty': 'l2', 'loss':
'log_loss'}. Best is trial 7 with value: 0.9177503796669544.
[I 2025-11-25 11:09:34,624] Trial 8 finished with value: 0.9199874032219215 and
parameters: {'alpha': 7.4759096056390015e-06, 'penalty': 'l2', 'loss':
'log_loss'}. Best is trial 8 with value: 0.9199874032219215.
[I 2025-11-25 11:09:37,045] Trial 9 finished with value: 0.9008191241380485 and
parameters: {'alpha': 0.0017467776142629865, 'penalty': 'l2', 'loss':
'modified_huber'}. Best is trial 8 with value: 0.9199874032219215.
[I 2025-11-25 11:09:43,248] Trial 10 finished with value: 0.9196105103510762 and
parameters: {'alpha': 1.922821330931126e-06, 'penalty': 'elasticnet', 'loss':
'log_loss', 'l1_ratio': 0.07187386025753795}. Best is trial 8 with value:
0.9199874032219215.
[I 2025-11-25 11:09:49,540] Trial 11 finished with value: 0.9176343002173601 and
parameters: {'alpha': 1.0014704152897357e-06, 'penalty': 'elasticnet', 'loss':
'log_loss', 'l1_ratio': 0.05932265517320357}. Best is trial 8 with value:
0.9199874032219215.
[I 2025-11-25 11:09:54,140] Trial 12 finished with value: 0.9208126562033367 and
parameters: {'alpha': 4.947962119336253e-06, 'penalty': 'elasticnet', 'loss':
'log_loss', 'l1_ratio': 0.1013153010165367}. Best is trial 12 with value:
0.9208126562033367.
[I 2025-11-25 11:09:59,479] Trial 13 finished with value: 0.7694327454111802 and
parameters: {'alpha': 0.001168143473257208, 'penalty': 'elasticnet', 'loss':
'log_loss', 'l1_ratio': 0.4063384403311465}. Best is trial 12 with value:
```

```
0.9208126562033367.
[I 2025-11-25 11:10:04,359] Trial 14 finished with value: 0.9175806722643418 and
parameters: {'alpha': 7.737158118424287e-06, 'penalty': 'elasticnet', 'loss':
'log_loss', 'l1_ratio': 0.9293812787022451}. Best is trial 12 with value:
0.9208126562033367.
[I 2025-11-25 11:10:07,103] Trial 15 finished with value: 0.9083839991600854 and
parameters: {'alpha': 6.464958258331787e-05, 'penalty': 'l2', 'loss':
'log_loss'}. Best is trial 12 with value: 0.9208126562033367.
[I 2025-11-25 11:10:10,578] Trial 16 finished with value: 0.1 and parameters:
{'alpha': 0.09467887952172861, 'penalty': 'elasticnet', 'loss': 'log_loss',
'l1_ratio': 0.3857644279078905}. Best is trial 12 with value:
0.9208126562033367.
[I 2025-11-25 11:10:13,545] Trial 17 finished with value: 0.9204542841059199 and
parameters: {'alpha': 3.573711840609591e-06, 'penalty': 'l2', 'loss':
'log_loss'}. Best is trial 12 with value: 0.9208126562033367.
[I 2025-11-25 11:10:17,449] Trial 18 finished with value: 0.8872554982648282 and
parameters: {'alpha': 0.0006078869655377774, 'penalty': 'elasticnet', 'loss':
'modified_huber', 'l1_ratio': 0.23216722285143376}. Best is trial 12 with value:
0.9208126562033367.
[I 2025-11-25 11:10:20,443] Trial 19 finished with value: 0.9204802419518195 and
parameters: {'alpha': 4.0084759726438995e-06, 'penalty': 'l2', 'loss':
'log_loss'}. Best is trial 12 with value: 0.9208126562033367.
[I 2025-11-25 11:10:24,981] Trial 20 finished with value: 0.8944336878026838 and
parameters: {'alpha': 4.538448166337237e-05, 'penalty': 'elasticnet', 'loss':
'log_loss', 'l1_ratio': 0.6365647382680422}. Best is trial 12 with value:
0.9208126562033367.
[I 2025-11-25 11:10:27,824] Trial 21 finished with value: 0.9203741129771416 and
parameters: {'alpha': 5.624801585817183e-06, 'penalty': 'l2', 'loss':
'log_loss'}. Best is trial 12 with value: 0.9208126562033367.
[I 2025-11-25 11:10:30,786] Trial 22 finished with value: 0.920383773887219 and
parameters: {'alpha': 3.425577158481033e-06, 'penalty': 'l2', 'loss':
'log_loss'}. Best is trial 12 with value: 0.9208126562033367.
[I 2025-11-25 11:10:34,199] Trial 23 finished with value: 0.9181539350665787 and
parameters: {'alpha': 1.2522246415266808e-06, 'penalty': 'l2', 'loss':
'log_loss'}. Best is trial 12 with value: 0.9208126562033367.
[I 2025-11-25 11:10:37,198] Trial 24 finished with value: 0.9203795124578369 and
parameters: {'alpha': 4.26544862913276e-06, 'penalty': 'l2', 'loss':
'log_loss'}. Best is trial 12 with value: 0.9208126562033367.
[I 2025-11-25 11:10:39,843] Trial 25 finished with value: 0.9172860123732411 and
parameters: {'alpha': 1.5522657525301603e-05, 'penalty': 'l2', 'loss':
'log_loss'}. Best is trial 12 with value: 0.9208126562033367.
[I 2025-11-25 11:10:42,056] Trial 26 finished with value: 0.9077910308738615 and
parameters: {'alpha': 3.3124267864033565e-06, 'penalty': 'l2', 'loss':
'modified_huber'}. Best is trial 12 with value: 0.9208126562033367.
[I 2025-11-25 11:10:44,897] Trial 27 finished with value: 0.8943525697998049 and
parameters: {'alpha': 0.00025602625613428177, 'penalty': 'l2', 'loss':
'log_loss'}. Best is trial 12 with value: 0.9208126562033367.
[I 2025-11-25 11:10:48,945] Trial 28 finished with value: 0.8932998865970514 and
```

```
parameters: {'alpha': 3.8778158725957745e-05, 'penalty': 'l1', 'loss':
'log_loss'}. Best is trial 12 with value: 0.9208126562033367.
[I 2025-11-25 11:10:53,452] Trial 29 finished with value: 0.9157470335917477 and
parameters: {'alpha': 1.2891674658291655e-05, 'penalty': 'elasticnet', 'loss':
'log_loss', 'l1_ratio': 0.29645605345916437}. Best is trial 12 with value:
0.9208126562033367.
[I 2025-11-25 11:10:53,455] A new study created in memory with name: no-
name-9538cac0-28cf-493b-9a5e-b510c23bdd00

  Best F1: 0.9208
  Best Params: {'alpha': 4.947962119336253e-06, 'penalty': 'elasticnet', 'loss':
'log_loss', 'l1_ratio': 0.1013153010165367}
  Time: 111.33s

Running optimization for NaiveBayes…

[I 2025-11-25 11:10:54,166] Trial 0 finished with value: 0.904118779660932 and
parameters: {'alpha': 0.001064171238556927}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:10:54,851] Trial 1 finished with value: 0.9014087303829139 and
parameters: {'alpha': 0.12838171546040153}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:10:55,558] Trial 2 finished with value: 0.9023502546732717 and
parameters: {'alpha': 0.053005616864101004}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:10:56,255] Trial 3 finished with value: 0.8894444241761665 and
parameters: {'alpha': 7.576647112165318}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:10:56,936] Trial 4 finished with value: 0.9020481430290171 and
parameters: {'alpha': 0.0654744070301976}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:10:57,622] Trial 5 finished with value: 0.9000216200237541 and
parameters: {'alpha': 0.31274321385444226}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:10:58,446] Trial 6 finished with value: 0.888451733365395 and
parameters: {'alpha': 8.848080571609703}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:10:59,137] Trial 7 finished with value: 0.8983014761508616 and
parameters: {'alpha': 0.6743602050654508}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:10:59,824] Trial 8 finished with value: 0.8982122194077643 and
parameters: {'alpha': 0.7200109802742883}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:00,509] Trial 9 finished with value: 0.8953131169688145 and
parameters: {'alpha': 1.895451755288514}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:01,201] Trial 10 finished with value: 0.9040986448112702 and
parameters: {'alpha': 0.001046445403608703}. Best is trial 0 with value:
0.904118779660932.
```

```
[I 2025-11-25 11:11:01,887] Trial 11 finished with value: 0.9041080066205434 and
parameters: {'alpha': 0.0010422942025098253}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:02,586] Trial 12 finished with value: 0.904118682203659 and
parameters: {'alpha': 0.0010242269530889314}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:03,270] Trial 13 finished with value: 0.9038310801103921 and
parameters: {'alpha': 0.006426471290038563}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:03,969] Trial 14 finished with value: 0.9037986764240694 and
parameters: {'alpha': 0.007740699037515634}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:04,647] Trial 15 finished with value: 0.9037884140920103 and
parameters: {'alpha': 0.00812912290186258}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:05,338] Trial 16 finished with value: 0.9039048040795624 and
parameters: {'alpha': 0.004778313193621653}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:06,039] Trial 17 finished with value: 0.9039416393256211 and
parameters: {'alpha': 0.0024475193057783735}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:06,713] Trial 18 finished with value: 0.9031842728444891 and
parameters: {'alpha': 0.022181602545229356}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:07,409] Trial 19 finished with value: 0.9032626141088788 and
parameters: {'alpha': 0.021113211883201748}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:08,084] Trial 20 finished with value: 0.903940963026994 and
parameters: {'alpha': 0.0023108829733196074}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:08,788] Trial 21 finished with value: 0.9041181140854846 and
parameters: {'alpha': 0.0010286208722841454}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:09,484] Trial 22 finished with value: 0.9039316933340811 and
parameters: {'alpha': 0.002533668285571713}. Best is trial 0 with value:
0.904118779660932.
[I 2025-11-25 11:11:10,195] Trial 23 finished with value: 0.9041655461546293 and
parameters: {'alpha': 0.0011931231670689673}. Best is trial 23 with value:
0.9041655461546293.
[I 2025-11-25 11:11:10,888] Trial 24 finished with value: 0.9034174756126809 and
parameters: {'alpha': 0.015460837063868204}. Best is trial 23 with value:
0.9041655461546293.
[I 2025-11-25 11:11:11,583] Trial 25 finished with value: 0.9039011399642239 and
parameters: {'alpha': 0.002884995735416621}. Best is trial 23 with value:
0.9041655461546293.
[I 2025-11-25 11:11:12,270] Trial 26 finished with value: 0.9040421723181347 and
parameters: {'alpha': 0.0017635099556476213}. Best is trial 23 with value:
0.9041655461546293.
```

```
[I 2025-11-25 11:11:12,945] Trial 27 finished with value: 0.9038669752726914 and
parameters: {'alpha': 0.004054578366699656}. Best is trial 23 with value:
0.9041655461546293.
[I 2025-11-25 11:11:13,634] Trial 28 finished with value: 0.9034065713430536 and
parameters: {'alpha': 0.015336732192488047}. Best is trial 23 with value:
0.9041655461546293.
[I 2025-11-25 11:11:14,331] Trial 29 finished with value: 0.9014080590943288 and
parameters: {'alpha': 0.12970256412291298}. Best is trial 23 with value:
0.9041655461546293.
[I 2025-11-25 11:11:14,333] A new study created in memory with name: no-
name-27483c2f-f2ee-4f0a-97de-1de5c50cabe6
```

```
  Best F1: 0.9042
  Best Params: {'alpha': 0.0011931231670689673}
  Time: 20.88s

Running optimization for LinearSVC…
```

```
[I 2025-11-25 11:11:19,636] Trial 0 finished with value: 0.9128624733951665 and
parameters: {'C': 0.024980012938019085}. Best is trial 0 with value:
0.9128624733951665.
[I 2025-11-25 11:11:31,710] Trial 1 finished with value: 0.9183424890370817 and
parameters: {'C': 1.3932770666136087}. Best is trial 1 with value:
0.9183424890370817.
[I 2025-11-25 11:11:38,404] Trial 2 finished with value: 0.9233064027849676 and
parameters: {'C': 0.3652045317276682}. Best is trial 2 with value:
0.9233064027849676.
[I 2025-11-25 11:11:43,812] Trial 3 finished with value: 0.9224094835623493 and
parameters: {'C': 0.12142731666644255}. Best is trial 2 with value:
0.9233064027849676.
[I 2025-11-25 11:11:49,059] Trial 4 finished with value: 0.9124371021610399 and
parameters: {'C': 0.023168784892273822}. Best is trial 2 with value:
0.9233064027849676.
[I 2025-11-25 11:12:34,836] Trial 5 finished with value: 0.9100797855234987 and
parameters: {'C': 8.36103612367723}. Best is trial 2 with value:
0.9233064027849676.
[I 2025-11-25 11:12:40,296] Trial 6 finished with value: 0.9143579795853558 and
parameters: {'C': 0.030279947279803418}. Best is trial 2 with value:
0.9233064027849676.
[I 2025-11-25 11:13:02,418] Trial 7 finished with value: 0.9135532498230979 and
parameters: {'C': 3.680442871398865}. Best is trial 2 with value:
0.9233064027849676.
[I 2025-11-25 11:13:07,460] Trial 8 finished with value: 0.9185217006242399 and
parameters: {'C': 0.05568606185067378}. Best is trial 2 with value:
0.9233064027849676.
[I 2025-11-25 11:13:13,611] Trial 9 finished with value: 0.9235768467063007 and
parameters: {'C': 0.25386857090596615}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:13:20,753] Trial 10 finished with value: 0.9228663318484568 and
```

```
parameters: {'C': 0.4525187254533168}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:13:27,661] Trial 11 finished with value: 0.9232313247700802 and
parameters: {'C': 0.3886582652523945}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:13:33,073] Trial 12 finished with value: 0.9227778516003509 and
parameters: {'C': 0.14699601292647516}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:13:42,331] Trial 13 finished with value: 0.9210744086265408 and
parameters: {'C': 0.8476308479745394}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:13:47,662] Trial 14 finished with value: 0.9224299939309913 and
parameters: {'C': 0.12504159078600655}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:13:59,390] Trial 15 finished with value: 0.9184602996367388 and
parameters: {'C': 1.3611581366976022}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:14:05,521] Trial 16 finished with value: 0.923514409210059 and
parameters: {'C': 0.23395065821365574}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:14:11,285] Trial 17 finished with value: 0.9053069583718223 and
parameters: {'C': 0.010366525630080241}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:14:16,862] Trial 18 finished with value: 0.9233900952251913 and
parameters: {'C': 0.17923532027375877}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:14:21,914] Trial 19 finished with value: 0.9204988574932566 and
parameters: {'C': 0.07594668594593801}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:14:30,153] Trial 20 finished with value: 0.922271049698814 and
parameters: {'C': 0.6579248285008482}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:14:35,630] Trial 21 finished with value: 0.923253347035646 and
parameters: {'C': 0.17065948549006688}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:14:41,459] Trial 22 finished with value: 0.923418365368949 and
parameters: {'C': 0.19459832471696561}. Best is trial 9 with value:
0.9235768467063007.
[I 2025-11-25 11:14:47,462] Trial 23 finished with value: 0.9235844496938317 and
parameters: {'C': 0.24507915419346896}. Best is trial 23 with value:
0.9235844496938317.
[I 2025-11-25 11:14:53,594] Trial 24 finished with value: 0.9234305842113402 and
parameters: {'C': 0.27907374850431044}. Best is trial 23 with value:
0.9235844496938317.
[I 2025-11-25 11:14:58,634] Trial 25 finished with value: 0.9198850301626489 and
parameters: {'C': 0.06844099889095155}. Best is trial 23 with value:
0.9235844496938317.
[I 2025-11-25 11:15:08,229] Trial 26 finished with value: 0.921006006560553 and
```

```
parameters: {'C': 0.8612970336529235}. Best is trial 23 with value:
0.9235844496938317.
[I 2025-11-25 11:15:25,473] Trial 27 finished with value: 0.9155791901484218 and
parameters: {'C': 2.5647390961780387}. Best is trial 23 with value:
0.9235844496938317.
[I 2025-11-25 11:15:31,456] Trial 28 finished with value: 0.923576432608775 and
parameters: {'C': 0.2541542959364157}. Best is trial 23 with value:
0.9235844496938317.
[I 2025-11-25 11:15:39,438] Trial 29 finished with value: 0.9223953198405669 and
parameters: {'C': 0.6148501661255293}. Best is trial 23 with value:
0.9235844496938317.
[I 2025-11-25 11:15:39,441] A new study created in memory with name: no-
name-4bf282ac-e941-4a6a-90a1-4c33bbcb51cb
```

```
  Best F1: 0.9236
  Best Params: {'C': 0.24507915419346896}
  Time: 265.11s

Running optimization for LightGBM…
```

```
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:17:37,550] Trial 0 finished with value: 0.8998970240360317 and
parameters: {'n_estimators': 213, 'learning_rate': 0.18008626206710693,
'num_leaves': 35, 'max_depth': 5, 'subsample': 0.6945237016120651,
'colsample_bytree': 0.8999872747088535}. Best is trial 0 with value:
0.8998970240360317.
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:19:32,671] Trial 1 finished with value: 0.9028854830543628 and
parameters: {'n_estimators': 217, 'learning_rate': 0.23517357668890232,
'num_leaves': 20, 'max_depth': 5, 'subsample': 0.7824539409729636,
'colsample_bytree': 0.7959584930363208}. Best is trial 1 with value:
0.9028854830543628.
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:21:31,227] Trial 2 finished with value: 0.8969751595684663 and
parameters: {'n_estimators': 252, 'learning_rate': 0.1560541817198277,
'num_leaves': 26, 'max_depth': 4, 'subsample': 0.903186014977833,
'colsample_bytree': 0.8530746599972552}. Best is trial 1 with value:
0.9028854830543628.
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:23:23,073] Trial 3 finished with value: 0.8997087405126697 and
parameters: {'n_estimators': 289, 'learning_rate': 0.2940382016875403,
'num_leaves': 25, 'max_depth': 3, 'subsample': 0.9573174327034417,
'colsample_bytree': 0.741223688947292}. Best is trial 1 with value:
0.9028854830543628.
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
```

```
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:24:36,888] Trial 4 finished with value: 0.8763361661538861 and
parameters: {'n_estimators': 144, 'learning_rate': 0.09883656222759547,
'num_leaves': 60, 'max_depth': 4, 'subsample': 0.6558868219516923,
'colsample_bytree': 0.6662844680396283}. Best is trial 1 with value:
0.9028854830543628.
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:26:49,227] Trial 5 finished with value: 0.8759268107986319 and
parameters: {'n_estimators': 135, 'learning_rate': 0.051844162591289225,
'num_leaves': 41, 'max_depth': 7, 'subsample': 0.7925079467409215,
'colsample_bytree': 0.8741758421408636}. Best is trial 1 with value:
0.9028854830543628.
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:27:54,062] Trial 6 finished with value: 0.8863861685676248 and
parameters: {'n_estimators': 154, 'learning_rate': 0.2003867557063947,
'num_leaves': 35, 'max_depth': 3, 'subsample': 0.7690893677495266,
'colsample_bytree': 0.6899694543420741}. Best is trial 1 with value:
0.9028854830543628.
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
```

```
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:31:16,470] Trial 7 finished with value: 0.9035186126147979 and
parameters: {'n_estimators': 261, 'learning_rate': 0.13089015938510923,
'num_leaves': 56, 'max_depth': 7, 'subsample': 0.7185111205180461,
'colsample_bytree': 0.8122591803060994}. Best is trial 7 with value:
0.9035186126147979.
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:34:20,955] Trial 8 finished with value: 0.8903760471319878 and
parameters: {'n_estimators': 171, 'learning_rate': 0.056874887107906544,
'num_leaves': 49, 'max_depth': 9, 'subsample': 0.8043626766619841,
'colsample_bytree': 0.6131603607274235}. Best is trial 7 with value:
0.9035186126147979.
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:36:18,065] Trial 9 finished with value: 0.8853652886925354 and
parameters: {'n_estimators': 155, 'learning_rate': 0.07801919623950056,
'num_leaves': 59, 'max_depth': 6, 'subsample': 0.7994728520270431,
'colsample_bytree': 0.709437301977671}. Best is trial 7 with value:
0.9035186126147979.
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
```

```
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:41:26,688] Trial 10 finished with value: 0.9072704751582572 and
parameters: {'n_estimators': 300, 'learning_rate': 0.1283308355405813,
'num_leaves': 50, 'max_depth': 10, 'subsample': 0.6037700855650294,
'colsample_bytree': 0.987999511349397}. Best is trial 10 with value:
0.9072704751582572.
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:46:38,788] Trial 11 finished with value: 0.9073948516128842 and
parameters: {'n_estimators': 300, 'learning_rate': 0.1282327391772812,
'num_leaves': 51, 'max_depth': 10, 'subsample': 0.6030984016366676,
'colsample_bytree': 0.9641719040207022}. Best is trial 11 with value:
0.9073948516128842.
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:51:49,497] Trial 12 finished with value: 0.9065000959887072 and
parameters: {'n_estimators': 300, 'learning_rate': 0.11267386057164297,
'num_leaves': 49, 'max_depth': 10, 'subsample': 0.6069746526727431,
'colsample_bytree': 0.9999847922969234}. Best is trial 11 with value:
0.9073948516128842.
```

```
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:53:43,274] Trial 13 finished with value: 0.8959978764699779 and
parameters: {'n_estimators': 105, 'learning_rate': 0.1409152282474426,
'num_leaves': 48, 'max_depth': 9, 'subsample': 0.6074518259424065,
'colsample_bytree': 0.9900137613890698}. Best is trial 11 with value:
0.9073948516128842.
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
[I 2025-11-25 11:57:42,957] Trial 14 finished with value: 0.9082975168606583 and
parameters: {'n_estimators': 260, 'learning_rate': 0.20613981753369892,
'num_leaves': 44, 'max_depth': 10, 'subsample': 0.667326308413759,
'colsample_bytree': 0.9302491873440469}. Best is trial 14 with value:
0.9082975168606583.
```

```
  Best F1: 0.9083
  Best Params: {'n_estimators': 260, 'learning_rate': 0.20613981753369892,
'num_leaves': 44, 'max_depth': 10, 'subsample': 0.667326308413759,
'colsample_bytree': 0.9302491873440469}
  Time: 2523.52s

Tuning Complete. Parameters saved.
```

[25]:

```
# 5.5 Retrain Best Models & Generate OOF Predictions (Stacking Prep)

# Instantiate models with best params
models = {
    'SGD': SGDClassifier(
        max_iter=1000, random_state=RANDOM_STATE, n_jobs=N_JOBS,
        early_stopping=True, **best_params['SGD']
    ),
    'NB': MultinomialNB(**best_params['NaiveBayes']),
    # Wrap SVC in CalibratedClassifierCV to get probabilities for Stacking
    'SVC': CalibratedClassifierCV(
        LinearSVC(max_iter=2000, random_state=RANDOM_STATE, dual='auto',␣
 ↪**best_params['LinearSVC']),
        method='sigmoid', cv=3
    ),
    'LGBM': LGBMClassifier(
        random_state=RANDOM_STATE, n_jobs=4, verbose=-1,␣
 ↪**best_params['LightGBM']
    )
}

oof_preds = {}
test_preds = {} # For final validation

cv_strategy = StratifiedKFold(n_splits=5, shuffle=True,␣
 ↪random_state=RANDOM_STATE)

for name, model in models.items():
    print(f"Processing {name}...")

    # Select appropriate feature set
    X_train_use = X_train_selected if name == 'LGBM' else X_train_hybrid
    X_val_use = X_val_selected if name == 'LGBM' else X_val_hybrid

    # 1. Generate OOF Probabilities (for training meta-learner)
    # method='predict_proba' ensures we get probability columns
    oof_probs = cross_val_predict(
        model, X_train_use, y_train, cv=cv_strategy,
        method='predict_proba', n_jobs=N_JOBS
    )

    # 2. Train on full data and predict on Validation (for testing meta-learner)
    model.fit(X_train_use, y_train)
    val_probs = model.predict_proba(X_val_use)

    # Store
    oof_preds[name] = oof_probs
    test_preds[name] = val_probs

    # Save individual tuned model
    joblib.dump(model, MODELS_DIR / f'step5_tuned_{name}.pkl')
```

```python
print("\nOOF Generation Complete.")
```

```
Processing SGD…
Processing NB…
Processing SVC…
Processing LGBM…
```

```
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
```

```

OOF Generation Complete.
```

```
/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
```

[26]:

```python
# 5.6 Build and Evaluate Ensembles



# Prepare Meta-Features
# We concatenate the probability outputs of all 4 models
# Shape: (N_samples, N_models * N_classes)
X_meta_train = np.hstack([oof_preds[name] for name in models.keys()])
X_meta_val = np.hstack([test_preds[name] for name in models.keys()])

print(f"Meta-Feature Shape: {X_meta_train.shape}")

# --- A. Stacking Ensemble (Meta-Learner) ---
# We use Logistic Regression to learn how to weigh the base models
```

```python
meta_learner = LogisticRegression(random_state=RANDOM_STATE)
meta_learner.fit(X_meta_train, y_train)

stacking_pred = meta_learner.predict(X_meta_val)
stacking_f1 = f1_score(y_val, stacking_pred, average='macro')

# --- B. Soft Voting Ensemble ---
# Simply average the probabilities
avg_probs = np.mean([test_preds[name] for name in models.keys()], axis=0)
voting_pred = np.argmax(avg_probs, axis=1)
voting_f1 = f1_score(y_val, voting_pred, average='macro')

# --- C. Comparison with Single Best Model ---
# Find best individual model on validation set
single_scores = {}
for name, model in models.items():
    X_val_use = X_val_selected if name == 'LGBM' else X_val_hybrid
    pred = model.predict(X_val_use)
    score = f1_score(y_val, pred, average='macro')
    single_scores[name] = score

print("\nRESULTS TABLE:")
print("-" * 40)
print(f"{'Model':<20} | {'Macro-F1':<10}")
print("-" * 40)
for name, score in single_scores.items():
    print(f"{name + ' (Tuned)':<20} | {score:.4f}")
print("-" * 40)
print(f"{'Soft Voting':<20} | {voting_f1:.4f}")
print(f"{'Stacking (LogReg)':<20} | {stacking_f1:.4f}")
print("-" * 40)

# Save Meta Learner
joblib.dump(meta_learner, MODELS_DIR / 'step5_meta_learner.pkl')
```

```
Meta-Feature Shape: (102000, 16)

RESULTS TABLE:
----------------------------------------
Model                | Macro-F1
----------------------------------------
SGD (Tuned)          | 0.9268
NB (Tuned)           | 0.9065
SVC (Tuned)          | 0.9274
LGBM (Tuned)         | 0.9120
----------------------------------------
Soft Voting          | 0.9248
Stacking (LogReg)    | 0.9295
```

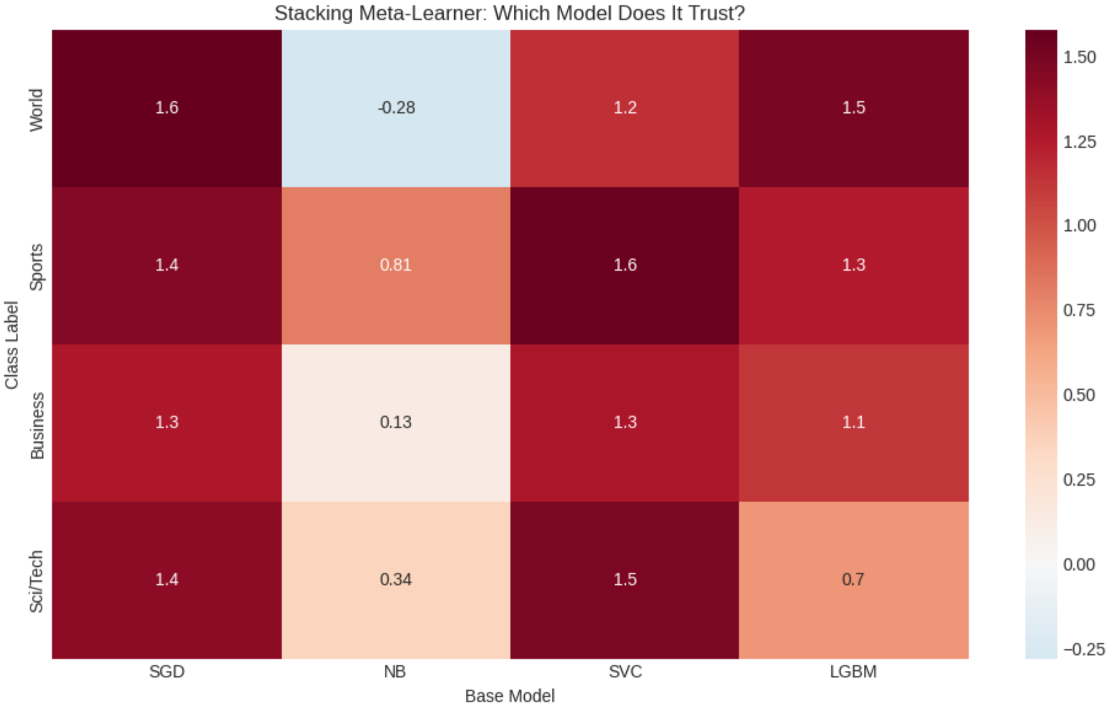
```
----------------------------------------

/home/dante/Desktop/prj/news/.venv/lib/python3.11/site-
packages/sklearn/utils/validation.py:2749: UserWarning: X does not have valid
feature names, but LGBMClassifier was fitted with feature names
  warnings.warn(
```

[26]:
```
['models/step5_meta_learner.pkl']
```

[27]:
```python
# 5.7 Meta-Learner Analysis



# The meta-learner has shape (n_classes, n_features)
# n_features = 4 models * 4 classes = 16 features
classes = ['World', 'Sports', 'Business', 'Sci/Tech']
model_names = list(models.keys())

# Create a DataFrame to visualize which model helps which class
# We sum the coefficients for the corresponding probability columns
coeffs = meta_learner.coef_ # Shape (4, 16)

importance_df = pd.DataFrame(index=classes, columns=model_names)

for i, cls in enumerate(classes):
    # The features are ordered: SGD_c0, SGD_c1... NB_c0, NB_c1...
    # We want to see how much the meta-learner weights the 'correct' class␣
 ↪probability from each model
    for j, model in enumerate(model_names):
        # The index of the probability column for class 'i' from model 'j'
        col_idx = j * 4 + i
        importance_df.loc[cls, model] = coeffs[i, col_idx]

print("Meta-Learner Coefficients (Diagonal):")
print("Values > 0 mean the Stacker trusts this model for this class.")
print(importance_df)

# Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(importance_df.astype(float), annot=True, cmap='RdBu_r', center=0)
plt.title("Stacking Meta-Learner: Which Model Does It Trust?")
plt.ylabel("Class Label")
plt.xlabel("Base Model")
plt.tight_layout()
plt.savefig(RESULTS_DIR / 'step5_stacking_weights.png')
plt.show()

print("\nStep 5 Complete.")
```

```
Meta-Learner Coefficients (Diagonal):
```

```
Values > 0 mean the Stacker trusts this model for this class.
               SGD        NB       SVC      LGBM
World     1.575814   -0.2811  1.15136  1.493773
Sports    1.439715  0.810591  1.556519  1.254916
Business  1.267826  0.129304  1.284509    1.1267
Sci/Tech  1.410645  0.344776  1.483867  0.696963
```



```
Step 5 Complete.
```