Annexe B: Optimisation d'hyper-paramètres de LightGBM

1 Milestone 2: build the best possible classifier

1.1 By Emiliano Aviles and Cassandre Hamel

[1]:
```
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.decomposition import TruncatedSVD, PCA
from sklearn.feature_selection import SelectKBest, chi2, mutual_info_classif
from sklearn.cluster import KMeans
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import cross_val_score
from sklearn.metrics import make_scorer, f1_score
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
from lightgbm import LGBMClassifier
from tqdm import tqdm
from sklearn.model_selection import GridSearchCV
import optuna
```

[2]:
```
# Load data
data_train = np.load('data_train.npy', allow_pickle = True)
data_test = np.load('data_test.npy', allow_pickle = True)
vocab_map = np.load('vocab_map.npy', allow_pickle=True)

# Load labels_train from CSV and extract the 'label' column
labels_train_df = pd.read_csv('label_train.csv')  # Assuming this is your labels␣
↪file
labels_train = labels_train_df['label'].values  # Extract the labels as a NumPy␣
↪array

# Convert training data to a DataFrame for visualization
df_train = pd.DataFrame(data_train)

# Add column names using vocab_map
df_train.columns = vocab_map

# Add the target labels to the DataFrame
df_train['TARGETT'] = labels_train

# Separate features and target variable
X = df_train.drop(columns=['TARGETT'])
```

```
y = df_train['TARGETT']

# Step 1: Apply TF-IDF Transformation to the Term Count Matrix
print("Applying TF-IDF Transformation...")

tfidf_transformer = TfidfTransformer()
X_tfidf = tfidf_transformer.fit_transform(X)

print("Transformation applied!")
```

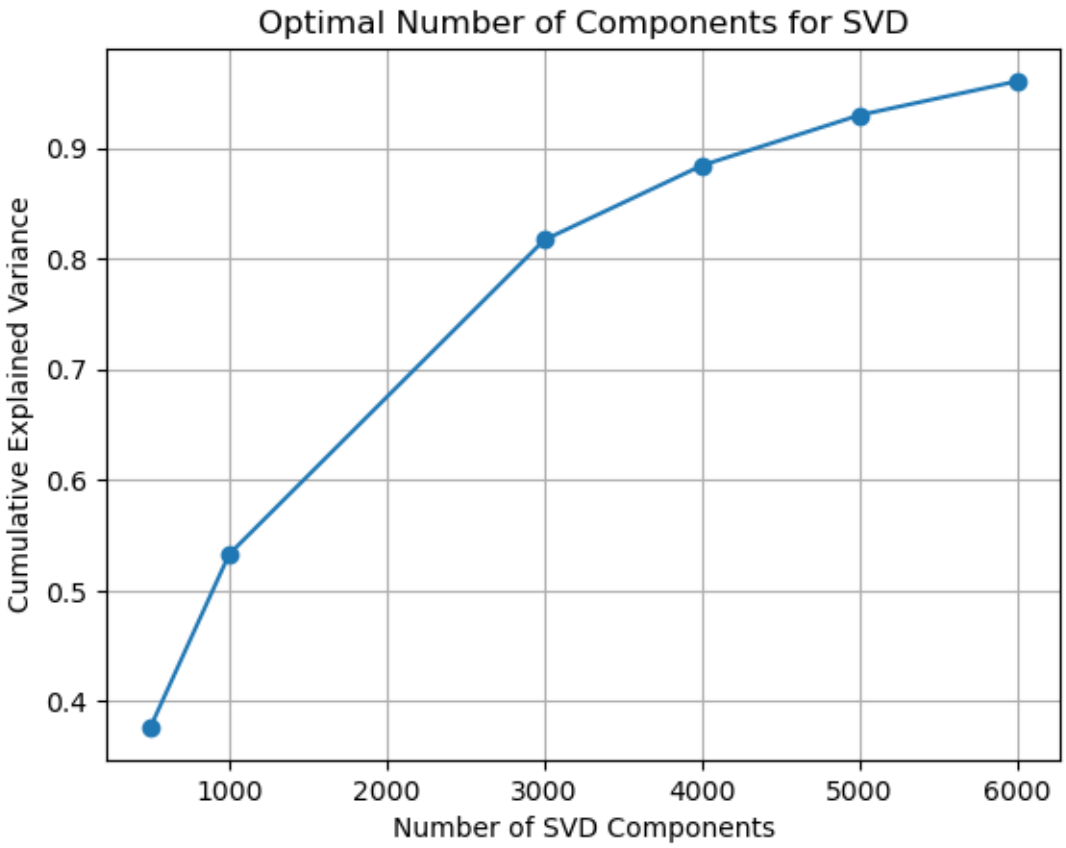
```
Applying TF-IDF Transformation...
Transformation applied!
```

[4]:
```
# Step 2: Determine Optimal SVD Components Based on Explained Variance
explained_variance_ratio = []
n_components_range = [500, 1000, 3000, 4000, 5000, 6000]  # Test up to 1000␣
 ↪components in steps of 20 for efficiency

for n in n_components_range:
    svd = TruncatedSVD(n_components=n, random_state=42)
    svd.fit(X_tfidf)
    cumulative_variance = svd.explained_variance_ratio_.sum()
    explained_variance_ratio.append(cumulative_variance)
    print(f"{n} components explain {cumulative_variance:.2%} of the variance")

# Plot Explained Variance to Find Elbow Point
plt.figure()
plt.plot(n_components_range, explained_variance_ratio, marker='o')
plt.xlabel("Number of SVD Components")
plt.ylabel("Cumulative Explained Variance")
plt.title("Optimal Number of Components for SVD")
plt.grid()
plt.show()
```

```
500 components explain 37.66% of the variance
1000 components explain 53.29% of the variance
3000 components explain 81.69% of the variance
4000 components explain 88.41% of the variance
5000 components explain 92.96% of the variance
6000 components explain 96.02% of the variance
```



[5]:
```
# Step 3: Transform Data Using Optimal SVD Components
svd_optimal = TruncatedSVD(n_components=5000, random_state=42)
X_svd_optimal = svd_optimal.fit_transform(X_tfidf)

print("X_svd_otimal found!")
```

```
X_svd_otimal found!
```

[4]:
```
# Step 4: Implement different combinations of hyper-parameters for finding a␣
↪good LightGBM model

# Define a parameter grid for LightGBM
param_grid = {
    'n_estimators': [100, 200, 300],              # Number of boosting rounds
    'learning_rate': [0.01, 0.05, 0.1],           # Step size shrinkage
    'num_leaves': [31, 50, 100],                  # Maximum number of leaves in␣
↪a tree
    'max_depth': [-1, 10, 20],                    # Maximum depth of the tree␣
↪(-1 means no limit)
```

```
    'min_child_samples': [10, 20, 50],              # Minimum number of samples␣
 ↪per leaf
    'min_child_weight': [1e-3, 1e-2, 1e-1],         # Minimum sum of instance␣
 ↪weight (hessian) needed in a child
    'subsample': [0.8, 1.0],                        # Fraction of samples used␣
 ↪for each tree (boosting round)
    'colsample_bytree': [0.8, 1.0],                 # Fraction of features used␣
 ↪for each tree
    'lambda_l1': [0, 0.1, 1],                       # L1 regularization
    'lambda_l2': [0, 0.1, 1],                       # L2 regularization
}

# Define the evaluation metric
f1_scorer = make_scorer(f1_score, average='macro')

# Function to train and evaluate a LightGBM model with GPU support
def evaluate_lgbm_model(params, X, y):
    model = LGBMClassifier(**params, n_jobs=-1)  # Use GPU acceleration
    scores = cross_val_score(model, X, y, cv=3, scoring=f1_scorer, n_jobs=-1)
    print(f"Model params: {params}")
    print(f"Mean F1 Score: {scores.mean():.4f}\n")
    return scores.mean()

# Model 1: Default settings with a small number of trees
params_1 = {
    'n_estimators': 100,
    'learning_rate': 0.1,
    'num_leaves': 31,
    'max_depth': -1,
    'random_state': 42,
    'n_jobs': -1
}

# Model 2: Deeper trees with a lower learning rate
params_2 = {
    'n_estimators': 200,
    'learning_rate': 0.05,
    'num_leaves': 50,
    'max_depth': 15,
    'subsample': 0.8,
    'colsample_bytree': 0.8,
    'random_state': 42,
    'n_jobs': -1
}

# Model 3: Fewer leaves but higher regularization
params_3 = {
```

```
    'n_estimators': 150,
    'learning_rate': 0.05,
    'num_leaves': 20,
    'max_depth': 10,
    'lambda_l1': 0.1,
    'lambda_l2': 0.1,
    'min_child_samples': 20,
    'subsample': 0.9,
    'colsample_bytree': 0.9,
    'random_state': 42,
    'n_jobs': -1
}

# Evaluate each model
print("Evaluating LightGBM Models...\n")
f1_model_1 = evaluate_lgbm_model(params_1, X_svd_optimal, y)
f1_model_2 = evaluate_lgbm_model(params_2, X_svd_optimal, y)
f1_model_3 = evaluate_lgbm_model(params_3, X_svd_optimal, y)

# Summary of results
results = {
    "Model 1 (Default)": f1_model_1,
    "Model 2 (Deeper Trees)": f1_model_2,
    "Model 3 (Regularization)": f1_model_3
}

print("\nFinal Results Summary:")
for model, score in results.items():
    print(f"{model}: F1 Score = {score:.4f}")
```

```
Evaluating LightGBM Models...

Model params: {'n_estimators': 100, 'learning_rate': 0.1, 'num_leaves': 31,
'max_depth': -1, 'random_state': 42, 'n_jobs': -1}
Mean F1 Score: 0.6166

Model params: {'n_estimators': 200, 'learning_rate': 0.05, 'num_leaves': 50,
'max_depth': 15, 'subsample': 0.8, 'colsample_bytree': 0.8, 'random_state': 42,
'n_jobs': -1}
Mean F1 Score: 0.6045

Model params: {'n_estimators': 150, 'learning_rate': 0.05, 'num_leaves': 20,
'max_depth': 10, 'lambda_l1': 0.1, 'lambda_l2': 0.1, 'min_child_samples': 20,
'subsample': 0.9, 'colsample_bytree': 0.9, 'random_state': 42, 'n_jobs': -1}
Mean F1 Score: 0.6144


Final Results Summary:
```

```
Model 1 (Default): F1 Score = 0.6166
Model 2 (Deeper Trees): F1 Score = 0.6045
Model 3 (Regularization): F1 Score = 0.6144
```

[5]:
```
# Further dimensionality reduction with GPU acceleration and F1_scores for
# different randomized hyper-parameter combinations

# Define the evaluation metric
f1_scorer = make_scorer(f1_score, average='macro')

# Function to optimize with Optuna
def objective(trial, X, y):
    # Step 1: Optimize `k` for feature selection
    k = trial.suggest_int('k', 500, 5000, step=500)

    # Choose the feature selection method
    feature_selection_method = trial.suggest_categorical('method', ['chi2',␣
↪'mutual_info'])

    # Automatically switch to mutual_info_classif if the data contains negative␣
↪values
    if feature_selection_method == 'chi2' and (X < 0).any():
        print("Data contains negative values, switching to mutual_info_classif")
        selector = SelectKBest(mutual_info_classif, k=k)

    elif feature_selection_method == 'chi2':
        selector = SelectKBest(chi2, k=k)
    else:
        selector = SelectKBest(mutual_info_classif, k=k)

    X_reduced = selector.fit_transform(X, y)

    # Step 2: Optimize LightGBM hyperparameters
    params = {
        'objective': 'binary',
        'n_estimators': trial.suggest_int('n_estimators', 100, 1000),
        'learning_rate': trial.suggest_float('learning_rate', 0.01, 0.1, log =␣
↪True),
        'num_leaves': trial.suggest_int('num_leaves', 20, 100),
        'max_depth': trial.suggest_int('max_depth', 5, 20),
        'min_child_samples': trial.suggest_int('min_child_samples', 10, 50),
        'min_child_weight': trial.suggest_float('min_child_weight', 1e-3, 1e-1,␣
↪log = True),
        'subsample': trial.suggest_float('subsample', 0.5, 1.0),
        'colsample_bytree': trial.suggest_float('colsample_bytree', 0.5, 1.0),
        'lambda_l1': trial.suggest_float('lambda_l1', 1e-3, 1.0, log = True),
        'lambda_l2': trial.suggest_float('lambda_l2', 1e-3, 1.0, log = True),
```

```
        'scale_pos_weight': len(y[y == 0]) / len(y[y == 1]),
        'device': 'gpu',  # Use GPU acceleration
        'random_state': 42,
        'n_jobs': -1
    }

    model = LGBMClassifier(**params)

    # Evaluate using cross-validation
    scores = cross_val_score(model, X_reduced, y, cv=3, scoring=f1_scorer,␣
↪n_jobs=-1)
    return scores.mean()

# Optimize hyperparameters using Optuna with a progress bar
def optimize_with_optuna(X, y, n_trials=50):
    study = optuna.create_study(direction='maximize')

    with tqdm(total=n_trials, desc="Optuna Trials Progress") as pbar:
        def callback(study, trial):
            # Update progress bar on each completed trial
            pbar.update(1)
            # Print the best trial so far
            print(f"Trial {trial.number} completed - F1 Score: {trial.value:.
↪4f}, Best F1 Score: {study.best_value:.4f}")

        study.optimize(lambda trial: objective(trial, X, y), n_trials=n_trials,␣
↪callbacks=[callback])

    print(f"\nBest F1 Score: {study.best_value:.4f}")
    print("Best Parameters:", study.best_params)

    # Visualize optimization results
    plot_optimization_history(study).show()
    plot_param_importances(study).show()
    plot_parallel_coordinate(study).show()
    plot_slice(study).show()
    plot_contour(study).show()

    return study.best_params

# Load your preprocessed data after applying SVD
print("\nApplying Optuna Optimization on SVD Data...")
best_params = optimize_with_optuna(X_svd_optimal, y, n_trials=50)

# Train final model using the best parameters found
def train_final_model(X, y, params):
    k = params.pop('k')
```

```
    method = params.pop('method')

    if method == 'chi2':
        selector = SelectKBest(chi2, k=k)
    else:
        selector = SelectKBest(mutual_info_classif, k=k)

    X_reduced = selector.fit_transform(X, y)
    model = LGBMClassifier(**params)
    scores = cross_val_score(model, X_reduced, y, cv=3, scoring=f1_scorer,␣
↪n_jobs=-1)
    print(f"\nFinal Model F1 Score: {scores.mean():.4f}")

print("\nTraining Final Model with Optimized Parameters...")
train_final_model(X_svd_optimal, y, best_params)
```

```
[I 2024-11-09 00:00:20,680] A new study created in memory with name: no-
name-7db9edad-f876-4f96-afac-57df4be238a7


Applying Optuna Optimization on SVD Data...

Optuna Trials Progress:   0%|
| 0/50 [00:00<?, ?it/s][I 2024-11-09 00:08:59,949] Trial 0 finished with value:
0.6235023474823428 and parameters: {'k': 3500, 'method': 'mutual_info',
'n_estimators': 630, 'learning_rate': 0.06887655111248715, 'num_leaves': 95,
'max_depth': 10, 'min_child_samples': 42, 'min_child_weight':
0.08082561158362306, 'subsample': 0.8270125925783023, 'colsample_bytree':
0.6693087804091462, 'lambda_l1': 0.01750099457880224, 'lambda_l2':
0.007862327009207089}. Best is trial 0 with value: 0.6235023474823428.
Optuna Trials Progress:   2%|
| 1/50 [08:39<7:04:04, 519.27s/it]

Trial 0 completed - F1 Score: 0.6235, Best F1 Score: 0.6235
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 00:16:14,275] Trial 1 finished with value: 0.6388731121585408 and
parameters: {'k': 2500, 'method': 'chi2', 'n_estimators': 778, 'learning_rate':
0.09849944377720078, 'num_leaves': 81, 'max_depth': 13, 'min_child_samples': 41,
'min_child_weight': 0.012313138723784483, 'subsample': 0.971393047389842,
'colsample_bytree': 0.7559740050109527, 'lambda_l1': 0.04921963431788272,
'lambda_l2': 0.1454697035361137}. Best is trial 1 with value:
0.6388731121585408.
Optuna Trials Progress:   4%|
| 2/50 [15:53<6:15:26, 469.30s/it]

Trial 1 completed - F1 Score: 0.6389, Best F1 Score: 0.6389

[I 2024-11-09 00:21:47,344] Trial 2 finished with value: 0.6349504748115119 and
parameters: {'k': 500, 'method': 'mutual_info', 'n_estimators': 747,
'learning_rate': 0.034158171020010245, 'num_leaves': 65, 'max_depth': 20,
```

```
'min_child_samples': 41, 'min_child_weight': 0.0013289687413224754, 'subsample':
0.9841016783562776, 'colsample_bytree': 0.5011307455863958, 'lambda_l1':
0.17345465288074843, 'lambda_l2': 0.7731588572643261}. Best is trial 1 with
value: 0.6388731121585408.
Optuna Trials Progress:   6%|
| 3/50 [21:26<5:18:53, 407.09s/it]

Trial 2 completed - F1 Score: 0.6350, Best F1 Score: 0.6389
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 00:27:23,404] Trial 3 finished with value: 0.6645184794395632 and
parameters: {'k': 2000, 'method': 'chi2', 'n_estimators': 431, 'learning_rate':
0.0536215084884131, 'num_leaves': 28, 'max_depth': 9, 'min_child_samples': 23,
'min_child_weight': 0.0015402458314191078, 'subsample': 0.8763184288853227,
'colsample_bytree': 0.716580823198006, 'lambda_l1': 0.49965496855785124,
'lambda_l2': 0.012150113550378932}. Best is trial 3 with value:
0.6645184794395632.
Optuna Trials Progress:   8%|
| 4/50 [27:02<4:50:36, 379.05s/it]

Trial 3 completed - F1 Score: 0.6645, Best F1 Score: 0.6645

[I 2024-11-09 00:31:36,016] Trial 4 finished with value: 0.6715103657668321 and
parameters: {'k': 500, 'method': 'mutual_info', 'n_estimators': 156,
'learning_rate': 0.03986684817335969, 'num_leaves': 66, 'max_depth': 8,
'min_child_samples': 32, 'min_child_weight': 0.01593864316845239, 'subsample':
0.7822470285684844, 'colsample_bytree': 0.7018125791698697, 'lambda_l1':
0.05888211668867544, 'lambda_l2': 0.0033992662402508077}. Best is trial 4 with
value: 0.6715103657668321.
Optuna Trials Progress:  10%|
| 5/50 [31:15<4:10:05, 333.46s/it]

Trial 4 completed - F1 Score: 0.6715, Best F1 Score: 0.6715
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 00:38:37,307] Trial 5 finished with value: 0.6356735409325686 and
parameters: {'k': 3000, 'method': 'chi2', 'n_estimators': 610, 'learning_rate':
0.0542364587691214, 'num_leaves': 100, 'max_depth': 14, 'min_child_samples': 46,
'min_child_weight': 0.04402146646529704, 'subsample': 0.6087363455809343,
'colsample_bytree': 0.5594503717423809, 'lambda_l1': 0.7009412543903374,
'lambda_l2': 0.5956970412873357}. Best is trial 4 with value:
0.6715103657668321.
Optuna Trials Progress:  12%|
| 6/50 [38:16<4:26:26, 363.32s/it]

Trial 5 completed - F1 Score: 0.6357, Best F1 Score: 0.6715

[I 2024-11-09 00:49:28,688] Trial 6 finished with value: 0.6219549811829174 and
parameters: {'k': 4000, 'method': 'mutual_info', 'n_estimators': 530,
'learning_rate': 0.07331089059280722, 'num_leaves': 91, 'max_depth': 11,
'min_child_samples': 29, 'min_child_weight': 0.002501671791224599, 'subsample':
0.8669337888075992, 'colsample_bytree': 0.5164032770334324, 'lambda_l1':
```

```
0.005248738388270759, 'lambda_l2': 0.6184266612182531}. Best is trial 4 with
value: 0.6715103657668321.
Optuna Trials Progress:  14%|
| 7/50 [49:08<5:27:52, 457.49s/it]

Trial 6 completed - F1 Score: 0.6220, Best F1 Score: 0.6715
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 00:57:53,688] Trial 7 finished with value: 0.6707802203405543 and
parameters: {'k': 3000, 'method': 'chi2', 'n_estimators': 529, 'learning_rate':
0.037863905165411174, 'num_leaves': 30, 'max_depth': 10, 'min_child_samples':
50, 'min_child_weight': 0.0032469564611588243, 'subsample': 0.708325421182449,
'colsample_bytree': 0.9880528967862803, 'lambda_l1': 0.005467755211785072,
'lambda_l2': 0.8500971036391992}. Best is trial 4 with value:
0.6715103657668321.
Optuna Trials Progress:  16%|
| 8/50 [57:33<5:30:49, 472.62s/it]

Trial 7 completed - F1 Score: 0.6708, Best F1 Score: 0.6715

[I 2024-11-09 01:03:05,656] Trial 8 finished with value: 0.6542518893311363 and
parameters: {'k': 1000, 'method': 'mutual_info', 'n_estimators': 265,
'learning_rate': 0.06475293212586594, 'num_leaves': 97, 'max_depth': 8,
'min_child_samples': 30, 'min_child_weight': 0.0027426172075112544, 'subsample':
0.5450404367844313, 'colsample_bytree': 0.8476411231660823, 'lambda_l1':
0.5454463319154875, 'lambda_l2': 0.010112889095416739}. Best is trial 4 with
value: 0.6715103657668321.
Optuna Trials Progress:  18%|
| 9/50 [1:02:44<4:48:38, 422.40s/it]

Trial 8 completed - F1 Score: 0.6543, Best F1 Score: 0.6715
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 01:08:34,610] Trial 9 finished with value: 0.6785981923832853 and
parameters: {'k': 2000, 'method': 'chi2', 'n_estimators': 577, 'learning_rate':
0.030911229866324802, 'num_leaves': 27, 'max_depth': 6, 'min_child_samples': 37,
'min_child_weight': 0.001226971697247327, 'subsample': 0.9769599746886701,
'colsample_bytree': 0.68470393266994, 'lambda_l1': 0.7981968843340821,
'lambda_l2': 0.00694286584658936}. Best is trial 9 with value:
0.6785981923832853.
Optuna Trials Progress:  20%|
| 10/50 [1:08:13<4:22:21, 393.55s/it]

Trial 9 completed - F1 Score: 0.6786, Best F1 Score: 0.6786
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 01:14:23,408] Trial 10 finished with value: 0.679943999640126 and
parameters: {'k': 1500, 'method': 'chi2', 'n_estimators': 947, 'learning_rate':
0.01577286783889854, 'num_leaves': 44, 'max_depth': 5, 'min_child_samples': 10,
'min_child_weight': 0.007277059364642142, 'subsample': 0.683996363122032,
'colsample_bytree': 0.8271557658639406, 'lambda_l1': 0.001026049745987376,
'lambda_l2': 0.0011779925731107844}. Best is trial 10 with value:
```

```
0.679943999640126.
Optuna Trials Progress:  22%|
| 11/50 [1:14:02<4:06:54, 379.85s/it]

Trial 10 completed - F1 Score: 0.6799, Best F1 Score: 0.6799
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 01:20:16,502] Trial 11 finished with value: 0.6756428554818606 and
parameters: {'k': 1500, 'method': 'chi2', 'n_estimators': 988, 'learning_rate':
0.016254640442826644, 'num_leaves': 43, 'max_depth': 5, 'min_child_samples': 11,
'min_child_weight': 0.0056174847535963, 'subsample': 0.6846787330410691,
'colsample_bytree': 0.8566408254074589, 'lambda_l1': 0.0011938167946807512,
'lambda_l2': 0.0011638015617297854}. Best is trial 10 with value:
0.679943999640126.
Optuna Trials Progress:  24%|
| 12/50 [1:19:55<3:55:25, 371.71s/it]

Trial 11 completed - F1 Score: 0.6756, Best F1 Score: 0.6799
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 01:26:52,441] Trial 12 finished with value: 0.678301021476866 and
parameters: {'k': 2000, 'method': 'chi2', 'n_estimators': 968, 'learning_rate':
0.017132813124089052, 'num_leaves': 48, 'max_depth': 5, 'min_child_samples': 10,
'min_child_weight': 0.006789526803632723, 'subsample': 0.5974447816351263,
'colsample_bytree': 0.8127225741119175, 'lambda_l1': 0.0010947690420178506,
'lambda_l2': 0.0010541851175190113}. Best is trial 10 with value:
0.679943999640126.
Optuna Trials Progress:  26%|
| 13/50 [1:26:31<3:53:44, 379.05s/it]

Trial 12 completed - F1 Score: 0.6783, Best F1 Score: 0.6799
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 01:34:41,481] Trial 13 finished with value: 0.6983529782815862 and
parameters: {'k': 5000, 'method': 'chi2', 'n_estimators': 820, 'learning_rate':
0.010183235579285525, 'num_leaves': 21, 'max_depth': 5, 'min_child_samples': 17,
'min_child_weight': 0.027099125170458416, 'subsample': 0.6856996203818133,
'colsample_bytree': 0.6163067050211289, 'lambda_l1': 0.1824725483279566,
'lambda_l2': 0.05119971825214225}. Best is trial 13 with value:
0.6983529782815862.
Optuna Trials Progress:  28%|
| 14/50 [1:34:20<4:03:44, 406.23s/it]

Trial 13 completed - F1 Score: 0.6984, Best F1 Score: 0.6984
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 01:47:27,934] Trial 14 finished with value: 0.7001649224770166 and
parameters: {'k': 5000, 'method': 'chi2', 'n_estimators': 825, 'learning_rate':
0.010131646440713127, 'num_leaves': 20, 'max_depth': 16, 'min_child_samples':
18, 'min_child_weight': 0.022835633748607285, 'subsample': 0.6635012554748161,
'colsample_bytree': 0.9903848575850792, 'lambda_l1': 0.16132514277328733,
'lambda_l2': 0.05527462600442084}. Best is trial 14 with value:
```

```
0.7001649224770166.
Optuna Trials Progress:  30%|
| 15/50 [1:47:07<5:00:18, 514.81s/it]

Trial 14 completed - F1 Score: 0.7002, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 02:00:15,821] Trial 15 finished with value: 0.6978986794180684 and
parameters: {'k': 5000, 'method': 'chi2', 'n_estimators': 791, 'learning_rate':
0.010581345794844255, 'num_leaves': 21, 'max_depth': 16, 'min_child_samples':
19, 'min_child_weight': 0.025971086510508154, 'subsample': 0.625853287145187,
'colsample_bytree': 0.9980257160979499, 'lambda_l1': 0.17230751726742707,
'lambda_l2': 0.06940623033308055}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  32%|
| 16/50 [1:59:55<5:34:53, 590.99s/it]

Trial 15 completed - F1 Score: 0.6979, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 02:15:13,423] Trial 16 finished with value: 0.6725663973129259 and
parameters: {'k': 5000, 'method': 'chi2', 'n_estimators': 842, 'learning_rate':
0.01313060795042486, 'num_leaves': 37, 'max_depth': 18, 'min_child_samples': 18,
'min_child_weight': 0.023253724962008063, 'subsample': 0.7402655435340252,
'colsample_bytree': 0.6096060253869919, 'lambda_l1': 0.16110086500412385,
'lambda_l2': 0.03766099134484302}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  34%|
| 17/50 [2:14:52<6:15:45, 683.19s/it]

Trial 16 completed - F1 Score: 0.6726, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 02:35:00,917] Trial 17 finished with value: 0.6493134697032221 and
parameters: {'k': 4500, 'method': 'chi2', 'n_estimators': 699, 'learning_rate':
0.022991893148314794, 'num_leaves': 54, 'max_depth': 15, 'min_child_samples':
17, 'min_child_weight': 0.05330634482282394, 'subsample': 0.5126954145859455,
'colsample_bytree': 0.9212697510290253, 'lambda_l1': 0.10526454041913386,
'lambda_l2': 0.17190572065080484}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  36%|
| 18/50 [2:34:40<7:25:11, 834.73s/it]

Trial 17 completed - F1 Score: 0.6493, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 02:44:23,859] Trial 18 finished with value: 0.6982218055746815 and
parameters: {'k': 4000, 'method': 'chi2', 'n_estimators': 873, 'learning_rate':
0.010999622677976526, 'num_leaves': 22, 'max_depth': 17, 'min_child_samples':
25, 'min_child_weight': 0.030624798118370898, 'subsample': 0.6427337819761698,
'colsample_bytree': 0.6204060404185878, 'lambda_l1': 0.01771693230685754,
'lambda_l2': 0.02916870823089359}. Best is trial 14 with value:
```

```
0.7001649224770166.
Optuna Trials Progress:  38%|
| 19/50 [2:44:03<6:29:06, 753.10s/it]

Trial 18 completed - F1 Score: 0.6982, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 02:54:40,655] Trial 19 finished with value: 0.675259248699437 and
parameters: {'k': 4500, 'method': 'chi2', 'n_estimators': 457, 'learning_rate':
0.023125081736810937, 'num_leaves': 37, 'max_depth': 12, 'min_child_samples':
15, 'min_child_weight': 0.0164755319161435, 'subsample': 0.7792707514330273,
'colsample_bytree': 0.7689094939099113, 'lambda_l1': 0.3549890899925461,
'lambda_l2': 0.18111350266370133}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  40%|
| 20/50 [2:54:19<5:56:05, 712.17s/it]

Trial 19 completed - F1 Score: 0.6753, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 03:16:42,173] Trial 20 finished with value: 0.6400477672993913 and
parameters: {'k': 4500, 'method': 'chi2', 'n_estimators': 883, 'learning_rate':
0.02357323353619814, 'num_leaves': 79, 'max_depth': 19, 'min_child_samples': 23,
'min_child_weight': 0.08124899778715115, 'subsample': 0.7347873452247844,
'colsample_bytree': 0.9251632143946606, 'lambda_l1': 0.2917063526089418,
'lambda_l2': 0.02600706963620178}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  42%|
| 21/50 [3:16:21<7:12:37, 895.08s/it]

Trial 20 completed - F1 Score: 0.6400, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 03:26:14,114] Trial 21 finished with value: 0.6970400646605142 and
parameters: {'k': 4000, 'method': 'chi2', 'n_estimators': 867, 'learning_rate':
0.010714825622239823, 'num_leaves': 21, 'max_depth': 17, 'min_child_samples':
22, 'min_child_weight': 0.03285441277563135, 'subsample': 0.6448513033806181,
'colsample_bytree': 0.6271077047340818, 'lambda_l1': 0.02253117453027686,
'lambda_l2': 0.06370029484356125}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  44%|
| 22/50 [3:25:53<6:12:26, 798.10s/it]

Trial 21 completed - F1 Score: 0.6970, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 03:38:13,909] Trial 22 finished with value: 0.6861677912804366 and
parameters: {'k': 5000, 'method': 'chi2', 'n_estimators': 709, 'learning_rate':
0.01240834157541727, 'num_leaves': 34, 'max_depth': 16, 'min_child_samples': 26,
'min_child_weight': 0.04381053115834528, 'subsample': 0.5628524512927825,
'colsample_bytree': 0.6043065288676716, 'lambda_l1': 0.007508486601910011,
'lambda_l2': 0.022155182771081028}. Best is trial 14 with value:
```

```
0.7001649224770166.
Optuna Trials Progress:  46%|
| 23/50 [3:37:53<5:48:34, 774.60s/it]

Trial 22 completed - F1 Score: 0.6862, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 03:47:43,036] Trial 23 finished with value: 0.6933621495888218 and
parameters: {'k': 4000, 'method': 'chi2', 'n_estimators': 885, 'learning_rate':
0.01313052129076421, 'num_leaves': 21, 'max_depth': 18, 'min_child_samples': 14,
'min_child_weight': 0.021978731791736733, 'subsample': 0.6621244769490866,
'colsample_bytree': 0.6318554442191525, 'lambda_l1': 0.01160840385713188,
'lambda_l2': 0.0931379463492358}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  48%|
| 24/50 [3:47:22<5:08:56, 712.95s/it]

Trial 23 completed - F1 Score: 0.6934, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 03:55:55,134] Trial 24 finished with value: 0.6972775200863893 and
parameters: {'k': 3500, 'method': 'chi2', 'n_estimators': 664, 'learning_rate':
0.01025046635915053, 'num_leaves': 29, 'max_depth': 14, 'min_child_samples': 21,
'min_child_weight': 0.01053141695905916, 'subsample': 0.5778056135509324,
'colsample_bytree': 0.5518968809538352, 'lambda_l1': 0.05657174084808934,
'lambda_l2': 0.040194332750961785}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  50%|
| 25/50 [3:55:34<4:29:27, 646.68s/it]

Trial 24 completed - F1 Score: 0.6973, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 04:05:01,409] Trial 25 finished with value: 0.6817443789127808 and
parameters: {'k': 4500, 'method': 'chi2', 'n_estimators': 811, 'learning_rate':
0.01915236423110352, 'num_leaves': 21, 'max_depth': 20, 'min_child_samples': 16,
'min_child_weight': 0.03319995673680757, 'subsample': 0.6489569761162599,
'colsample_bytree': 0.5773453907936382, 'lambda_l1': 0.09597146616746188,
'lambda_l2': 0.017132864135275504}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  52%|
| 26/50 [4:04:40<4:06:37, 616.56s/it]

Trial 25 completed - F1 Score: 0.6817, Best F1 Score: 0.7002

[I 2024-11-09 04:25:41,399] Trial 26 finished with value: 0.6514386466622276 and
parameters: {'k': 5000, 'method': 'mutual_info', 'n_estimators': 915,
'learning_rate': 0.013359518006329856, 'num_leaves': 53, 'max_depth': 16,
'min_child_samples': 26, 'min_child_weight': 0.015875021726395534, 'subsample':
0.7088622496054039, 'colsample_bytree': 0.6690811427345249, 'lambda_l1':
0.036407725822951785, 'lambda_l2': 0.24088000055989}. Best is trial 14 with
value: 0.7001649224770166.
```

```
Optuna Trials Progress:  54%|
| 27/50 [4:25:20<5:08:02, 803.60s/it]

Trial 26 completed - F1 Score: 0.6514, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 04:38:44,318] Trial 27 finished with value: 0.6679004432672967 and
parameters: {'k': 4000, 'method': 'chi2', 'n_estimators': 757, 'learning_rate':
0.01940545991273601, 'num_leaves': 33, 'max_depth': 18, 'min_child_samples': 35,
'min_child_weight': 0.0655563887565127, 'subsample': 0.5215847746051075,
'colsample_bytree': 0.9401402116089065, 'lambda_l1': 0.003124359574553715,
'lambda_l2': 0.04875664616286717}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  56%|
| 28/50 [4:38:23<4:52:22, 797.40s/it]

Trial 27 completed - F1 Score: 0.6679, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 04:56:42,528] Trial 28 finished with value: 0.6690052279032539 and
parameters: {'k': 4500, 'method': 'chi2', 'n_estimators': 990, 'learning_rate':
0.011912182339568849, 'num_leaves': 41, 'max_depth': 14, 'min_child_samples':
26, 'min_child_weight': 0.0348352880535964, 'subsample': 0.7778003235095833,
'colsample_bytree': 0.730752743277788, 'lambda_l1': 0.10218705324252138,
'lambda_l2': 0.09902381579224875}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  58%|
| 29/50 [4:56:21<5:08:34, 881.64s/it]

Trial 28 completed - F1 Score: 0.6690, Best F1 Score: 0.7002

[I 2024-11-09 05:05:22,394] Trial 29 finished with value: 0.6897992364115403 and
parameters: {'k': 3500, 'method': 'mutual_info', 'n_estimators': 680,
'learning_rate': 0.014654421390742349, 'num_leaves': 26, 'max_depth': 12,
'min_child_samples': 13, 'min_child_weight': 0.0888399091958248, 'subsample':
0.8036120159736874, 'colsample_bytree': 0.6636401491002956, 'lambda_l1':
0.015904383380494828, 'lambda_l2': 0.005209892874757266}. Best is trial 14 with
value: 0.7001649224770166.
Optuna Trials Progress:  60%|
| 30/50 [5:05:01<4:17:42, 773.11s/it]

Trial 29 completed - F1 Score: 0.6898, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 05:13:07,325] Trial 30 finished with value: 0.6945871799794533 and
parameters: {'k': 5000, 'method': 'chi2', 'n_estimators': 398, 'learning_rate':
0.010869383870873103, 'num_leaves': 26, 'max_depth': 7, 'min_child_samples': 19,
'min_child_weight': 0.0560525837753051, 'subsample': 0.8462113345121274,
'colsample_bytree': 0.7898953595865762, 'lambda_l1': 0.025023126670856093,
'lambda_l2': 0.017078653826955236}. Best is trial 14 with value:
0.7001649224770166.
```

```
Optuna Trials Progress:  62%|
| 31/50 [5:12:46<3:35:32, 680.65s/it]

Trial 30 completed - F1 Score: 0.6946, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 05:25:22,511] Trial 31 finished with value: 0.6946233856708292 and
parameters: {'k': 5000, 'method': 'chi2', 'n_estimators': 812, 'learning_rate':
0.010343724894521405, 'num_leaves': 20, 'max_depth': 16, 'min_child_samples':
20, 'min_child_weight': 0.01906166860484486, 'subsample': 0.6211051426041365,
'colsample_bytree': 0.9720906240667817, 'lambda_l1': 0.2823416099727132,
'lambda_l2': 0.07440224903814252}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  64%|
| 32/50 [5:25:01<3:29:06, 697.01s/it]

Trial 31 completed - F1 Score: 0.6946, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 05:39:21,252] Trial 32 finished with value: 0.6902992301066835 and
parameters: {'k': 5000, 'method': 'chi2', 'n_estimators': 777, 'learning_rate':
0.011879518971010415, 'num_leaves': 25, 'max_depth': 17, 'min_child_samples':
24, 'min_child_weight': 0.029027562315866794, 'subsample': 0.6272684645593155,
'colsample_bytree': 0.9951410942085699, 'lambda_l1': 0.18177502869272483,
'lambda_l2': 0.4002989996285582}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  66%|
| 33/50 [5:39:00<3:29:32, 739.53s/it]

Trial 32 completed - F1 Score: 0.6903, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 05:55:05,121] Trial 33 finished with value: 0.6869669395882917 and
parameters: {'k': 4500, 'method': 'chi2', 'n_estimators': 816, 'learning_rate':
0.010140570365677622, 'num_leaves': 33, 'max_depth': 17, 'min_child_samples':
17, 'min_child_weight': 0.02560437039059345, 'subsample': 0.6696467896061185,
'colsample_bytree': 0.9024725858922235, 'lambda_l1': 0.04177419222745204,
'lambda_l2': 0.05015976351596896}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  68%|
| 34/50 [5:54:44<3:33:33, 800.83s/it]

Trial 33 completed - F1 Score: 0.6870, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 06:06:42,015] Trial 34 finished with value: 0.6904467684406882 and
parameters: {'k': 3500, 'method': 'chi2', 'n_estimators': 912, 'learning_rate':
0.013860124207377087, 'num_leaves': 23, 'max_depth': 15, 'min_child_samples':
13, 'min_child_weight': 0.01143862364422078, 'subsample': 0.7112237601601473,
'colsample_bytree': 0.9548731307179379, 'lambda_l1': 0.07692400174729012,
'lambda_l2': 0.11917740446512033}. Best is trial 14 with value:
0.7001649224770166.
```

```
Optuna Trials Progress:  70%|
| 35/50 [6:06:21<3:12:24, 769.65s/it]

Trial 34 completed - F1 Score: 0.6904, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 06:24:38,301] Trial 35 finished with value: 0.6531631612401383 and
parameters: {'k': 4000, 'method': 'chi2', 'n_estimators': 736, 'learning_rate':
0.011656759451785329, 'num_leaves': 74, 'max_depth': 13, 'min_child_samples':
19, 'min_child_weight': 0.013774503444858404, 'subsample': 0.6359560994745858,
'colsample_bytree': 0.6498653499361856, 'lambda_l1': 0.16625471645408518,
'lambda_l2': 0.02828863583361207}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  72%|
| 36/50 [6:24:17<3:21:02, 861.64s/it]

Trial 35 completed - F1 Score: 0.6532, Best F1 Score: 0.7002

[I 2024-11-09 06:38:46,947] Trial 36 finished with value: 0.6704786397146423 and
parameters: {'k': 5000, 'method': 'mutual_info', 'n_estimators': 610,
'learning_rate': 0.018083825962850467, 'num_leaves': 37, 'max_depth': 19,
'min_child_samples': 25, 'min_child_weight': 0.04004397564726283, 'subsample':
0.5959706770720533, 'colsample_bytree': 0.8901575908785717, 'lambda_l1':
0.22644358347027155, 'lambda_l2': 0.2844179744154226}. Best is trial 14 with
value: 0.7001649224770166.
Optuna Trials Progress:  74%|
| 37/50 [6:38:26<3:05:50, 857.74s/it]

Trial 36 completed - F1 Score: 0.6705, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 06:48:54,605] Trial 37 finished with value: 0.6816486735361683 and
parameters: {'k': 4500, 'method': 'chi2', 'n_estimators': 779, 'learning_rate':
0.014401624266039924, 'num_leaves': 30, 'max_depth': 15, 'min_child_samples':
29, 'min_child_weight': 0.02024327221590036, 'subsample': 0.5674937289640749,
'colsample_bytree': 0.5254642104869957, 'lambda_l1': 0.46186907295047025,
'lambda_l2': 0.06664199212952247}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  76%|
| 38/50 [6:48:33<2:36:32, 782.72s/it]

Trial 37 completed - F1 Score: 0.6816, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 07:02:36,872] Trial 38 finished with value: 0.643648776802693 and
parameters: {'k': 3000, 'method': 'chi2', 'n_estimators': 928, 'learning_rate':
0.020795205359595656, 'num_leaves': 61, 'max_depth': 11, 'min_child_samples':
21, 'min_child_weight': 0.06238771418816673, 'subsample': 0.9059378496029925,
'colsample_bytree': 0.5837274757550602, 'lambda_l1': 0.12216897721640527,
'lambda_l2': 0.016774583977245107}. Best is trial 14 with value:
0.7001649224770166.
```

```
Optuna Trials Progress:  78%|
| 39/50 [7:02:16<2:25:40, 794.58s/it]

Trial 38 completed - F1 Score: 0.6436, Best F1 Score: 0.7002

[I 2024-11-09 07:11:36,472] Trial 39 finished with value: 0.6659850182329032 and
parameters: {'k': 3500, 'method': 'mutual_info', 'n_estimators': 656,
'learning_rate': 0.03080112510196184, 'num_leaves': 31, 'max_depth': 10,
'min_child_samples': 32, 'min_child_weight': 0.00813661150475623, 'subsample':
0.6926952216018916, 'colsample_bytree': 0.6979727207322979, 'lambda_l1':
0.07658417805622386, 'lambda_l2': 0.13466858285472927}. Best is trial 14 with
value: 0.7001649224770166.
Optuna Trials Progress:  80%|
| 40/50 [7:11:15<1:59:40, 718.09s/it]

Trial 39 completed - F1 Score: 0.6660, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 07:39:13,672] Trial 40 finished with value: 0.6322308508322149 and
parameters: {'k': 5000, 'method': 'chi2', 'n_estimators': 851, 'learning_rate':
0.015782598198865253, 'num_leaves': 87, 'max_depth': 19, 'min_child_samples':
28, 'min_child_weight': 0.0276599162723646, 'subsample': 0.7519489644972677,
'colsample_bytree': 0.7451350270842187, 'lambda_l1': 0.011621850033540706,
'lambda_l2': 0.0100141877285638}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  82%|
| 41/50 [7:38:52<2:29:58, 999.82s/it]

Trial 40 completed - F1 Score: 0.6322, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 07:46:40,175] Trial 41 finished with value: 0.6971941887952223 and
parameters: {'k': 3500, 'method': 'chi2', 'n_estimators': 642, 'learning_rate':
0.010412416672602403, 'num_leaves': 25, 'max_depth': 14, 'min_child_samples':
22, 'min_child_weight': 0.009912894684146986, 'subsample': 0.5741275903938494,
'colsample_bytree': 0.5488216507891006, 'lambda_l1': 0.05883633631116622,
'lambda_l2': 0.03720670283551231}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  84%|
| 42/50 [7:46:19<1:51:10, 833.83s/it]

Trial 41 completed - F1 Score: 0.6972, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 07:53:49,337] Trial 42 finished with value: 0.6950404325218367 and
parameters: {'k': 2500, 'method': 'chi2', 'n_estimators': 725, 'learning_rate':
0.011285075244233712, 'num_leaves': 28, 'max_depth': 13, 'min_child_samples':
20, 'min_child_weight': 0.00551685583848621, 'subsample': 0.6024880376553918,
'colsample_bytree': 0.5401778381509712, 'lambda_l1': 0.0493425563169085,
'lambda_l2': 0.04013215712061319}. Best is trial 14 with value:
0.7001649224770166.
```

```
Optuna Trials Progress:  86%|
| 43/50 [7:53:28<1:23:06, 712.43s/it]

Trial 42 completed - F1 Score: 0.6950, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[I 2024-11-09 08:02:56,735] Trial 43 finished with value: 0.6976204984887215 and
parameters: {'k': 4000, 'method': 'chi2', 'n_estimators': 784, 'learning_rate':
0.0100669603276488, 'num_leaves': 24, 'max_depth': 15, 'min_child_samples': 16,
'min_child_weight': 0.012461721675270307, 'subsample': 0.5358641577301217,
'colsample_bytree': 0.5048948280662204, 'lambda_l1': 0.9571120770254014,
'lambda_l2': 0.054457001201312215}. Best is trial 14 with value:
0.7001649224770166.
Optuna Trials Progress:  88%|
| 44/50 [8:02:36<1:06:17, 662.92s/it]

Trial 43 completed - F1 Score: 0.6976, Best F1 Score: 0.7002
Data contains negative values, switching to mutual_info_classif

[W 2024-11-09 08:06:26,352] Trial 44 failed with parameters: {'k': 4000,
'method': 'chi2'} because of the following error: KeyboardInterrupt().
Traceback (most recent call last):
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\optuna\study\_optimize.py", line 197, in _run_trial
    value_or_values = func(trial)
  File "C:\Users\emi_r\AppData\Local\Temp\ipykernel_4648\1199570481.py", line
63, in <lambda>
    study.optimize(lambda trial: objective(trial, X, y), n_trials=n_trials,
callbacks=[callback])
  File "C:\Users\emi_r\AppData\Local\Temp\ipykernel_4648\1199570481.py", line
25, in objective
    X_reduced = selector.fit_transform(X, y)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\_set_output.py", line 316, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\base.py", line 1101, in fit_transform
    return self.fit(X, y, **fit_params).transform(X)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\base.py", line 1473, in wrapper
    return fit_method(estimator, *args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_univariate_selection.py", line 567, in fit
    score_func_ret = self.score_func(X, y)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\_param_validation.py", line 186, in wrapper
    return func(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_mutual_info.py", line 571, in
mutual_info_classif
```

```
    return _estimate_mi(
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_mutual_info.py", line 317, in _estimate_mi
    mi = Parallel(n_jobs=n_jobs)(
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 74, in __call__
    return super().__call__(iterable_with_config)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1918, in __call__
    return output if self.return_generator else list(output)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1847, in _get_sequential_output
    res = func(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 136, in __call__
    return self.function(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_mutual_info.py", line 167, in _compute_mi
    return _compute_mi_cd(x, y, n_neighbors)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_mutual_info.py", line 129, in _compute_mi_cd
    r = nn.kneighbors()[0]
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\neighbors\_base.py", line 903, in kneighbors
    chunked_results = Parallel(n_jobs, prefer="threads")(
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 74, in __call__
    return super().__call__(iterable_with_config)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1918, in __call__
    return output if self.return_generator else list(output)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1847, in _get_sequential_output
    res = func(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 136, in __call__
    return self.function(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\neighbors\_base.py", line 704, in _tree_query_parallel_helper
    return tree.query(*args, **kwargs)
KeyboardInterrupt
[W 2024-11-09 08:06:26,463] Trial 44 failed with value None.
Optuna Trials Progress:  88%|
| 44/50 [8:06:05<1:06:17, 662.86s/it]
```

[11]:
```
# Further dimensionality reduction with GPU acceleration and F1_scores for
# different randomized hyper-parameter combinations
```

```
# Define the evaluation metric
f1_scorer = make_scorer(f1_score, average='macro')

# Function to optimize with Optuna
def objective(trial, X, y):
    # Step 1: Optimize `k` for feature selection
    k = 5000

    # Choose the feature selection method
    feature_selection_method = trial.suggest_categorical('method', ['chi2',␣
↪'mutual_info'])

    # Automatically switch to mutual_info_classif if the data contains negative␣
↪values
    if feature_selection_method == 'chi2' and (X < 0).any():
        feature_selection_method = "mutual_info"
        selector = SelectKBest(mutual_info_classif, k=k)

    elif feature_selection_method == 'chi2':
        feature_selection_method = "mutual_info"
        selector = SelectKBest(mutual_info_classif, k=k)
    else:
        selector = SelectKBest(mutual_info_classif, k=k)

    X_reduced = selector.fit_transform(X, y)

    # Step 2: Optimize LightGBM hyperparameters
    params = {
        'objective': 'binary',
        'n_estimators': trial.suggest_int('n_estimators', 500, 2000),
        'learning_rate': trial.suggest_float('learning_rate', 0.005, 0.011, log␣
↪= True),
        'num_leaves': trial.suggest_int('num_leaves', 20, 60),
        'max_depth': trial.suggest_int('max_depth', 5, 20),
        'min_child_samples': trial.suggest_int('min_child_samples', 10, 50),
        'min_child_weight': trial.suggest_float('min_child_weight', 1e-3, 1e-1,␣
↪log = True),
        'subsample': trial.suggest_float('subsample', 0.7, 1.0),
        'colsample_bytree': trial.suggest_float('colsample_bytree', 0.5, 0.9),
        'lambda_l1': trial.suggest_float('lambda_l1', 1e-3, 0.5, log = True),
        'lambda_l2': trial.suggest_float('lambda_l2', 1e-3, 0.5, log = True),
        'scale_pos_weight': len(y[y == 0]) / len(y[y == 1]),
        'device': 'gpu',  # Use GPU acceleration
        'random_state': 42,
        'n_jobs': -1
    }
```

```
    model = LGBMClassifier(**params)

    # Evaluate using cross-validation
    scores = cross_val_score(model, X_reduced, y, cv=3, scoring=f1_scorer,␣
 ↪n_jobs=-1)
    return scores.mean()

# Optimize hyperparameters using Optuna with a progress bar
def optimize_with_optuna(X, y, n_trials=50):
    study = optuna.create_study(direction='maximize')

    with tqdm(total=n_trials, desc="Optuna Trials Progress") as pbar:
        def callback(study, trial):
            # Update progress bar on each completed trial
            pbar.update(1)
            # Print the best trial so far
            print(f"Trial {trial.number} completed - F1 Score: {trial.value:.
 ↪4f}, Best F1 Score: {study.best_value:.4f}")

        study.optimize(lambda trial: objective(trial, X, y), n_trials=n_trials,␣
 ↪callbacks=[callback])

    print(f"\nBest F1 Score: {study.best_value:.4f}")
    print("Best Parameters:", study.best_params)

    # Visualize optimization results
    plot_optimization_history(study).show()
    plot_param_importances(study).show()
    plot_parallel_coordinate(study).show()
    plot_slice(study).show()
    plot_contour(study).show()

    return study.best_params

# Load your preprocessed data after applying SVD
print("\nApplying Optuna Optimization on SVD Data...")
best_params = optimize_with_optuna(X_svd_optimal, y, n_trials=50)

# Train final model using the best parameters found
def train_final_model(X, y, params):
    k = params.pop('k')
    method = params.pop('method')

    if method == 'chi2':
        selector = SelectKBest(chi2, k=k)
    else:
```

```
        selector = SelectKBest(mutual_info_classif, k=k)

    X_reduced = selector.fit_transform(X, y)
    model = LGBMClassifier(**params)
    scores = cross_val_score(model, X_reduced, y, cv=3, scoring=f1_scorer,␣
↪n_jobs=-1)
    print(f"\nFinal Model F1 Score: {scores.mean():.4f}")

print("\nTraining Final Model with Optimized Parameters...")
train_final_model(X_svd_optimal, y, best_params)
```

```
[I 2024-11-09 09:31:22,282] A new study created in memory with name: no-
name-80b57a1d-2871-4bc8-9fc0-4a1b2fc0ce35


Applying Optuna Optimization on SVD Data...

Optuna Trials Progress:   0%|
| 0/50 [00:00<?, ?it/s][I 2024-11-09 10:04:06,135] Trial 0 finished with value:
0.6614296543159686 and parameters: {'method': 'chi2', 'n_estimators': 1376,
'learning_rate': 0.00639155861370827, 'num_leaves': 53, 'max_depth': 17,
'min_child_samples': 12, 'min_child_weight': 0.055721936571308354, 'subsample':
0.7475829520661277, 'colsample_bytree': 0.6468418313358647, 'lambda_l1':
0.0042925695974801665, 'lambda_l2': 0.16077133773867605}. Best is trial 0 with
value: 0.6614296543159686.
Optuna Trials Progress:   2%|
| 1/50 [32:43<26:43:48, 1963.85s/it]

Trial 0 completed - F1 Score: 0.6614, Best F1 Score: 0.6614

[I 2024-11-09 10:26:58,761] Trial 1 finished with value: 0.6566770832231886 and
parameters: {'method': 'mutual_info', 'n_estimators': 1288, 'learning_rate':
0.00850974608630668, 'num_leaves': 57, 'max_depth': 7, 'min_child_samples': 38,
'min_child_weight': 0.013275190901935222, 'subsample': 0.9862182030268651,
'colsample_bytree': 0.6889038446119251, 'lambda_l1': 0.005111217334000554,
'lambda_l2': 0.1546021948099615}. Best is trial 0 with value:
0.6614296543159686.
Optuna Trials Progress:   4%|
| 2/50 [55:36<21:32:51, 1616.07s/it]

Trial 1 completed - F1 Score: 0.6567, Best F1 Score: 0.6614

[I 2024-11-09 10:49:35,342] Trial 2 finished with value: 0.6674406627580236 and
parameters: {'method': 'chi2', 'n_estimators': 1788, 'learning_rate':
0.007171319372972953, 'num_leaves': 33, 'max_depth': 6, 'min_child_samples': 17,
'min_child_weight': 0.002824283850844851, 'subsample': 0.7992721293121012,
'colsample_bytree': 0.7122201031875582, 'lambda_l1': 0.287318119600479,
'lambda_l2': 0.003326844005339016}. Best is trial 2 with value:
0.6674406627580236.
Optuna Trials Progress:   6%|
| 3/50 [1:18:13<19:33:06, 1497.58s/it]
```

```
Trial 2 completed - F1 Score: 0.6674, Best F1 Score: 0.6674

[I 2024-11-09 11:13:11,381] Trial 3 finished with value: 0.6676001615666287 and
parameters: {'method': 'mutual_info', 'n_estimators': 1271, 'learning_rate':
0.010495140184636141, 'num_leaves': 35, 'max_depth': 14, 'min_child_samples':
15, 'min_child_weight': 0.007918555723829216, 'subsample': 0.8617521742752678,
'colsample_bytree': 0.706139505917572, 'lambda_l1': 0.007303767159317069,
'lambda_l2': 0.001454836283222907}. Best is trial 3 with value:
0.6676001615666287.
Optuna Trials Progress:   8%|
| 4/50 [1:41:49<18:43:27, 1465.39s/it]

Trial 3 completed - F1 Score: 0.6676, Best F1 Score: 0.6676

[I 2024-11-09 11:29:29,899] Trial 4 finished with value: 0.6855099777876106 and
parameters: {'method': 'chi2', 'n_estimators': 975, 'learning_rate':
0.0078345744775295, 'num_leaves': 32, 'max_depth': 17, 'min_child_samples': 21,
'min_child_weight': 0.06030633325457074, 'subsample': 0.7882451318642654,
'colsample_bytree': 0.6661863083362384, 'lambda_l1': 0.26025734766254527,
'lambda_l2': 0.017932629723310587}. Best is trial 4 with value:
0.6855099777876106.
Optuna Trials Progress:  10%|
| 5/50 [1:58:07<16:07:21, 1289.82s/it]

Trial 4 completed - F1 Score: 0.6855, Best F1 Score: 0.6855

[W 2024-11-09 11:38:44,280] Trial 5 failed with parameters: {'method': 'chi2',
'n_estimators': 1877, 'learning_rate': 0.005095848390205523, 'num_leaves': 42,
'max_depth': 15, 'min_child_samples': 36, 'min_child_weight':
0.004489229396645799, 'subsample': 0.9271506928095047, 'colsample_bytree':
0.6733356253328151, 'lambda_l1': 0.006859318883013227, 'lambda_l2':
0.0037527562866533737} because of the following error: KeyboardInterrupt().
Traceback (most recent call last):
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1650, in _get_outputs
    yield from self._retrieve()
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1762, in _retrieve
    time.sleep(0.01)
KeyboardInterrupt

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\optuna\study\_optimize.py", line 197, in _run_trial
    value_or_values = func(trial)
  File "C:\Users\emi_r\AppData\Local\Temp\ipykernel_4648\1773060294.py", line
64, in <lambda>
    study.optimize(lambda trial: objective(trial, X, y), n_trials=n_trials,
```

```
callbacks=[callback])
  File "C:\Users\emi_r\AppData\Local\Temp\ipykernel_4648\1773060294.py", line
50, in objective
    scores = cross_val_score(model, X_reduced, y, cv=3, scoring=f1_scorer,
n_jobs=-1)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\_param_validation.py", line 213, in wrapper
    return func(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\model_selection\_validation.py", line 712, in cross_val_score
    cv_results = cross_validate(
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\_param_validation.py", line 213, in wrapper
    return func(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\model_selection\_validation.py", line 423, in cross_validate
    results = parallel(
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 74, in __call__
    return super().__call__(iterable_with_config)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 2007, in __call__
    return output if self.return_generator else list(output)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1703, in _get_outputs
    self._abort()
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1614, in _abort
    backend.abort_everything(ensure_ready=ensure_ready)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\_parallel_backends.py", line 620, in abort_everything
    self._workers.terminate(kill_workers=True)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\executor.py", line 75, in terminate
    self.shutdown(kill_workers=kill_workers)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\externals\loky\process_executor.py", line 1303, in shutdown
    executor_manager_thread.join()
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\threading.py", line 1060, in
join
    self._wait_for_tstate_lock()
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\threading.py", line 1080, in
_wait_for_tstate_lock
    if lock.acquire(block, timeout):
KeyboardInterrupt
[W 2024-11-09 11:38:46,035] Trial 5 failed with value None.
Optuna Trials Progress:  10%|
| 5/50 [2:07:23<19:06:33, 1528.75s/it]
```

[12]:
```
# Further dimensionality reduction with GPU acceleration and F1_scores for
# different randomized hyper-parameter combinations

# Define the evaluation metric
f1_scorer = make_scorer(f1_score, average='macro')

# Function to optimize with Optuna
def objective(trial, X, y):
    # Step 1: Optimize `k` for feature selection
    k = 5000

    # Choose the feature selection method
    feature_selection_method = trial.suggest_categorical('method', ['chi2',␣
↪'mutual_info'])

    # Automatically switch to mutual_info_classif if the data contains negative␣
↪values
    if feature_selection_method == 'chi2' and (X < 0).any():
        feature_selection_method = "mutual_info"
        selector = SelectKBest(mutual_info_classif, k=k)

    elif feature_selection_method == 'chi2':
        feature_selection_method = "mutual_info"
        selector = SelectKBest(mutual_info_classif, k=k)
    else:
        selector = SelectKBest(mutual_info_classif, k=k)

    X_reduced = selector.fit_transform(X, y)

    # Step 2: Optimize LightGBM hyperparameters
    params = {
        'objective': 'binary',
        'n_estimators': trial.suggest_int('n_estimators', 500, 900),
        'learning_rate': trial.suggest_float('learning_rate', 0.008, 0.011, log␣
↪= True),
        'num_leaves': trial.suggest_int('num_leaves', 10, 35),
        'max_depth': trial.suggest_int('max_depth', 8, 15),
        'min_child_samples': trial.suggest_int('min_child_samples', 10, 20),
        'min_child_weight': trial.suggest_float('min_child_weight', 1e-3, 1e-1,␣
↪log = True),
        'subsample': trial.suggest_float('subsample', 0.7, 1.0),
        'colsample_bytree': trial.suggest_float('colsample_bytree', 0.5, 0.9),
        'lambda_l1': trial.suggest_float('lambda_l1', 0.1, 0.5, log = True),
        'lambda_l2': trial.suggest_float('lambda_l2', 1e-3, 0.005, log = True),
        'scale_pos_weight': len(y[y == 0]) / len(y[y == 1]),
        'device': 'gpu',  # Use GPU acceleration
        'random_state': 42,
```

```
        'n_jobs': -1
    }

    model = LGBMClassifier(**params)

    # Evaluate using cross-validation
    scores = cross_val_score(model, X_reduced, y, cv=3, scoring=f1_scorer,␣
↪n_jobs=-1)
    return scores.mean()

# Optimize hyperparameters using Optuna with a progress bar
def optimize_with_optuna(X, y, n_trials=50):
    study = optuna.create_study(direction='maximize')

    with tqdm(total=n_trials, desc="Optuna Trials Progress") as pbar:
        def callback(study, trial):
            # Update progress bar on each completed trial
            pbar.update(1)
            # Print the best trial so far
            print(f"Trial {trial.number} completed - F1 Score: {trial.value:.
↪4f}, Best F1 Score: {study.best_value:.4f}")

        study.optimize(lambda trial: objective(trial, X, y), n_trials=n_trials,␣
↪callbacks=[callback])

    print(f"\nBest F1 Score: {study.best_value:.4f}")
    print("Best Parameters:", study.best_params)

    # Visualize optimization results
    plot_optimization_history(study).show()
    plot_param_importances(study).show()
    plot_parallel_coordinate(study).show()
    plot_slice(study).show()
    plot_contour(study).show()

    return study.best_params

# Load your preprocessed data after applying SVD
print("\nApplying Optuna Optimization on SVD Data...")
best_params = optimize_with_optuna(X_svd_optimal, y, n_trials=50)

# Train final model using the best parameters found
def train_final_model(X, y, params):
    k = params.pop('k')
    method = params.pop('method')

    if method == 'chi2':
```

```
        selector = SelectKBest(chi2, k=k)
    else:
        selector = SelectKBest(mutual_info_classif, k=k)

    X_reduced = selector.fit_transform(X, y)
    model = LGBMClassifier(**params)
    scores = cross_val_score(model, X_reduced, y, cv=3, scoring=f1_scorer,␣
↪n_jobs=-1)
    print(f"\nFinal Model F1 Score: {scores.mean():.4f}")

print("\nTraining Final Model with Optimized Parameters...")
train_final_model(X_svd_optimal, y, best_params)
```

```
[I 2024-11-09 11:41:56,842] A new study created in memory with name: no-
name-75e8398a-7e5a-44e0-b984-2161dc792c1a


Applying Optuna Optimization on SVD Data...

Optuna Trials Progress:   0%|
| 0/50 [00:00<?, ?it/s][I 2024-11-09 11:51:26,826] Trial 0 finished with value:
0.6984387270672284 and parameters: {'method': 'mutual_info', 'n_estimators':
670, 'learning_rate': 0.01187858610630135, 'num_leaves': 18, 'max_depth': 16,
'min_child_samples': 10, 'min_child_weight': 0.003980666732697284, 'subsample':
0.9800182771904905, 'colsample_bytree': 0.7087435550647764, 'lambda_l1':
0.16784378354599674, 'lambda_l2': 0.0033458383404920385}. Best is trial 0 with
value: 0.6984387270672284.
Optuna Trials Progress:   2%|
| 1/50 [09:29<7:45:29, 569.98s/it]

Trial 0 completed - F1 Score: 0.6984, Best F1 Score: 0.6984

[I 2024-11-09 11:58:42,255] Trial 1 finished with value: 0.698081505785651 and
parameters: {'method': 'chi2', 'n_estimators': 546, 'learning_rate':
0.011064065905722337, 'num_leaves': 18, 'max_depth': 17, 'min_child_samples':
16, 'min_child_weight': 0.060076259195530904, 'subsample': 0.7274340758065704,
'colsample_bytree': 0.5173935164192187, 'lambda_l1': 0.200065810704011,
'lambda_l2': 0.0013131883909397169}. Best is trial 0 with value:
0.6984387270672284.
Optuna Trials Progress:   4%|
| 2/50 [16:45<6:32:39, 490.83s/it]

Trial 1 completed - F1 Score: 0.6981, Best F1 Score: 0.6984

[I 2024-11-09 12:06:11,942] Trial 2 finished with value: 0.7002171421570988 and
parameters: {'method': 'chi2', 'n_estimators': 840, 'learning_rate':
0.010206429264446818, 'num_leaves': 12, 'max_depth': 15, 'min_child_samples':
18, 'min_child_weight': 0.001955097597595167, 'subsample': 0.9284268456128003,
'colsample_bytree': 0.5932170358734847, 'lambda_l1': 0.3224206427371498,
'lambda_l2': 0.0015206910065073405}. Best is trial 2 with value:
0.7002171421570988.
```

```
Optuna Trials Progress:   6%|
| 3/50 [24:15<6:09:46, 472.04s/it]

Trial 2 completed - F1 Score: 0.7002, Best F1 Score: 0.7002

[I 2024-11-09 12:17:08,731] Trial 3 finished with value: 0.6966606246259159 and
parameters: {'method': 'mutual_info', 'n_estimators': 808, 'learning_rate':
0.00819144037644812, 'num_leaves': 18, 'max_depth': 16, 'min_child_samples': 11,
'min_child_weight': 0.0633293949612287, 'subsample': 0.7073819947677166,
'colsample_bytree': 0.8306038682487653, 'lambda_l1': 0.12278294791356036,
'lambda_l2': 0.0019528120983153506}. Best is trial 2 with value:
0.7002171421570988.
Optuna Trials Progress:   8%|
| 4/50 [35:11<6:57:49, 544.98s/it]

Trial 3 completed - F1 Score: 0.6967, Best F1 Score: 0.7002

[I 2024-11-09 12:28:16,282] Trial 4 finished with value: 0.7048656272916073 and
parameters: {'method': 'chi2', 'n_estimators': 884, 'learning_rate':
0.008552871705252547, 'num_leaves': 18, 'max_depth': 15, 'min_child_samples':
12, 'min_child_weight': 0.02433182001711402, 'subsample': 0.9022878985957729,
'colsample_bytree': 0.688174345860965, 'lambda_l1': 0.2173588901875657,
'lambda_l2': 0.002597627293608841}. Best is trial 4 with value:
0.7048656272916073.
Optuna Trials Progress:  10%|
| 5/50 [46:19<7:21:53, 589.18s/it]

Trial 4 completed - F1 Score: 0.7049, Best F1 Score: 0.7049

[W 2024-11-09 12:31:41,964] Trial 5 failed with parameters: {'method': 'chi2'}
because of the following error: KeyboardInterrupt().
Traceback (most recent call last):
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\optuna\study\_optimize.py", line 197, in _run_trial
    value_or_values = func(trial)
  File "C:\Users\emi_r\AppData\Local\Temp\ipykernel_4648\2937233955.py", line
64, in <lambda>
    study.optimize(lambda trial: objective(trial, X, y), n_trials=n_trials,
callbacks=[callback])
  File "C:\Users\emi_r\AppData\Local\Temp\ipykernel_4648\2937233955.py", line
26, in objective
    X_reduced = selector.fit_transform(X, y)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\_set_output.py", line 316, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\base.py", line 1101, in fit_transform
    return self.fit(X, y, **fit_params).transform(X)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\base.py", line 1473, in wrapper
    return fit_method(estimator, *args, **kwargs)
```

```
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_univariate_selection.py", line 567, in fit
    score_func_ret = self.score_func(X, y)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\_param_validation.py", line 186, in wrapper
    return func(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_mutual_info.py", line 571, in
mutual_info_classif
    return _estimate_mi(
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_mutual_info.py", line 317, in _estimate_mi
    mi = Parallel(n_jobs=n_jobs)(
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 74, in __call__
    return super().__call__(iterable_with_config)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1918, in __call__
    return output if self.return_generator else list(output)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1847, in _get_sequential_output
    res = func(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 136, in __call__
    return self.function(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_mutual_info.py", line 167, in _compute_mi
    return _compute_mi_cd(x, y, n_neighbors)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_mutual_info.py", line 129, in _compute_mi_cd
    r = nn.kneighbors()[0]
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\neighbors\_base.py", line 903, in kneighbors
    chunked_results = Parallel(n_jobs, prefer="threads")(
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 74, in __call__
    return super().__call__(iterable_with_config)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1918, in __call__
    return output if self.return_generator else list(output)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1847, in _get_sequential_output
    res = func(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 136, in __call__
    return self.function(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\neighbors\_base.py", line 704, in _tree_query_parallel_helper
```

```
    return tree.query(*args, **kwargs)
KeyboardInterrupt
[W 2024-11-09 12:31:41,967] Trial 5 failed with value None.
Optuna Trials Progress:  10%|
| 5/50 [49:45<7:27:46, 597.02s/it]
```

[14]:
```python
# Further dimensionality reduction with GPU acceleration and F1_scores for
# different randomized hyper-parameter combinations

# Define the evaluation metric
f1_scorer = make_scorer(f1_score, average='macro')

# Function to optimize with Optuna
def objective(trial, X, y):
    # Step 1: Optimize `k` for feature selection
    k = 5000

    # Choose the feature selection method
    feature_selection_method = trial.suggest_categorical('method', ['chi2',␣
↪'mutual_info'])

    # Automatically switch to mutual_info_classif if the data contains negative␣
↪values
    if feature_selection_method == 'chi2' and (X < 0).any():
        feature_selection_method = "mutual_info"
        selector = SelectKBest(mutual_info_classif, k=k)

    elif feature_selection_method == 'chi2':
        feature_selection_method = "mutual_info"
        selector = SelectKBest(mutual_info_classif, k=k)
    else:
        selector = SelectKBest(mutual_info_classif, k=k)

    X_reduced = selector.fit_transform(X, y)

    # Step 2: Optimize LightGBM hyperparameters
    params = {
        'objective': 'binary',
        'n_estimators': trial.suggest_int('n_estimators', 500, 900),
        'learning_rate': trial.suggest_float('learning_rate', 0.008, 0.011, log␣
↪= True),
        'num_leaves': trial.suggest_int('num_leaves', 10, 35),
        'max_depth': trial.suggest_int('max_depth', 8, 16),
        'min_child_samples': trial.suggest_int('min_child_samples', 10, 20),
        'min_child_weight': trial.suggest_float('min_child_weight', 1e-3, 1e-1,␣
↪log = True),
        'subsample': trial.suggest_float('subsample', 0.7, 1.0),
```

```
        'colsample_bytree': trial.suggest_float('colsample_bytree', 0.5, 0.9),
        'lambda_l1': trial.suggest_float('lambda_l1', 0.1, 0.5, log = True),
        'lambda_l2': trial.suggest_float('lambda_l2', 1e-3, 0.005, log = True),
        'scale_pos_weight': len(y[y == 0]) / len(y[y == 1]),
        'device': 'gpu',  # Use GPU acceleration
        'random_state': 42,
        'n_jobs': -1
    }

    model = LGBMClassifier(**params)

    # Evaluate using cross-validation
    scores = cross_val_score(model, X_reduced, y, cv=3, scoring=f1_scorer,␣
↪n_jobs=-1)
    return scores.mean()

# Optimize hyperparameters using Optuna with a progress bar
def optimize_with_optuna(X, y, n_trials=50):
    study = optuna.create_study(direction='maximize')

    with tqdm(total=n_trials, desc="Optuna Trials Progress") as pbar:
        def callback(study, trial):
            # Update progress bar on each completed trial
            pbar.update(1)
            # Print the best trial so far
            print(f"Trial {trial.number} completed - F1 Score: {trial.value:.
↪4f}, Best F1 Score: {study.best_value:.4f}")

        study.optimize(lambda trial: objective(trial, X, y), n_trials=n_trials,␣
↪callbacks=[callback])

    print(f"\nBest F1 Score: {study.best_value:.4f}")
    print("Best Parameters:", study.best_params)

    # Visualize optimization results
    plot_optimization_history(study).show()
    plot_param_importances(study).show()
    plot_parallel_coordinate(study).show()
    plot_slice(study).show()
    plot_contour(study).show()

    return study.best_params

# Load your preprocessed data after applying SVD
print("\nApplying Optuna Optimization on SVD Data...")
best_params = optimize_with_optuna(X_svd_optimal, y, n_trials=50)
```

```python
# Train final model using the best parameters found
def train_final_model(X, y, params):
    k = params.pop('k')
    method = params.pop('method')

    if method == 'chi2':
        selector = SelectKBest(chi2, k=k)
    else:
        selector = SelectKBest(mutual_info_classif, k=k)

    X_reduced = selector.fit_transform(X, y)
    model = LGBMClassifier(**params)
    scores = cross_val_score(model, X_reduced, y, cv=3, scoring=f1_scorer,␣
 ↪n_jobs=-1)
    print(f"\nFinal Model F1 Score: {scores.mean():.4f}")

print("\nTraining Final Model with Optimized Parameters...")
train_final_model(X_svd_optimal, y, best_params)
```

```
[I 2024-11-09 12:33:17,095] A new study created in memory with name: no-
name-e1b9a9e4-94f4-4d9c-bf5f-f81eeab00bcc


Applying Optuna Optimization on SVD Data...

Optuna Trials Progress:   0%|
| 0/50 [00:00<?, ?it/s][I 2024-11-09 12:43:51,165] Trial 0 finished with value:
0.6930744083448476 and parameters: {'method': 'mutual_info', 'n_estimators':
881, 'learning_rate': 0.010688843953413173, 'num_leaves': 21, 'max_depth': 13,
'min_child_samples': 11, 'min_child_weight': 0.013606967452614206, 'subsample':
0.7487306715327516, 'colsample_bytree': 0.6046850236878447, 'lambda_l1':
0.41126809491501, 'lambda_l2': 0.0011420179224282073}. Best is trial 0 with
value: 0.6930744083448476.
Optuna Trials Progress:   2%|
| 1/50 [10:34<8:37:49, 634.07s/it]

Trial 0 completed - F1 Score: 0.6931, Best F1 Score: 0.6931

[I 2024-11-09 12:51:13,429] Trial 1 finished with value: 0.695925014248005 and
parameters: {'method': 'chi2', 'n_estimators': 726, 'learning_rate':
0.009726785827453076, 'num_leaves': 12, 'max_depth': 11, 'min_child_samples':
10, 'min_child_weight': 0.0014674428295185834, 'subsample': 0.8053354930997136,
'colsample_bytree': 0.6681923138417887, 'lambda_l1': 0.2315225082071698,
'lambda_l2': 0.010820363349000 95}. Best is trial 1 with value:
0.695925014248005.
Optuna Trials Progress:   4%|
| 2/50 [17:56<6:56:59, 521.24s/it]

Trial 1 completed - F1 Score: 0.6959, Best F1 Score: 0.6959

[I 2024-11-09 13:07:04,292] Trial 2 finished with value: 0.6866350371644613 and
```

```
parameters: {'method': 'mutual_info', 'n_estimators': 791, 'learning_rate':
0.009876098893112063, 'num_leaves': 34, 'max_depth': 13, 'min_child_samples':
10, 'min_child_weight': 0.0034072708634088843, 'subsample': 0.9611554300446059,
'colsample_bytree': 0.7400336195894457, 'lambda_l1': 0.1500031953978912,
'lambda_l2': 0.003026635299289609}. Best is trial 1 with value:
0.695925014248005.
Optuna Trials Progress:   6%|
| 3/50 [33:47<9:21:58, 717.42s/it]

Trial 2 completed - F1 Score: 0.6866, Best F1 Score: 0.6959

[I 2024-11-09 13:15:22,166] Trial 3 finished with value: 0.6930384556805059 and
parameters: {'method': 'mutual_info', 'n_estimators': 869, 'learning_rate':
0.009375324645110877, 'num_leaves': 11, 'max_depth': 16, 'min_child_samples':
12, 'min_child_weight': 0.0010220136378053053, 'subsample': 0.8853207346954495,
'colsample_bytree': 0.7277246002302993, 'lambda_l1': 0.11620006485345813,
'lambda_l2': 0.0015066119475974156}. Best is trial 1 with value:
0.695925014248005.
Optuna Trials Progress:   8%|
| 4/50 [42:05<8:03:34, 630.74s/it]

Trial 3 completed - F1 Score: 0.6930, Best F1 Score: 0.6959

[I 2024-11-09 13:25:28,030] Trial 4 finished with value: 0.6988410534595509 and
parameters: {'method': 'mutual_info', 'n_estimators': 893, 'learning_rate':
0.00958039052963381, 'num_leaves': 17, 'max_depth': 14, 'min_child_samples': 14,
'min_child_weight': 0.0013388304989459827, 'subsample': 0.903329508634547,
'colsample_bytree': 0.6523407701596748, 'lambda_l1': 0.23053533305662516,
'lambda_l2': 0.003580997320692697}. Best is trial 4 with value:
0.6988410534595509.
Optuna Trials Progress:  10%|
| 5/50 [52:10<7:46:19, 621.77s/it]

Trial 4 completed - F1 Score: 0.6988, Best F1 Score: 0.6988

[I 2024-11-09 13:34:37,973] Trial 5 finished with value: 0.6997424766024821 and
parameters: {'method': 'mutual_info', 'n_estimators': 829, 'learning_rate':
0.010155843523711306, 'num_leaves': 18, 'max_depth': 14, 'min_child_samples':
13, 'min_child_weight': 0.0028492748722602802, 'subsample': 0.8813998166056892,
'colsample_bytree': 0.5476442278874959, 'lambda_l1': 0.1325947833337639,
'lambda_l2': 0.0034446075571824006}. Best is trial 5 with value:
0.6997424766024821.
Optuna Trials Progress:  12%|
| 6/50 [1:01:20<7:18:03, 597.35s/it]

Trial 5 completed - F1 Score: 0.6997, Best F1 Score: 0.6997

[I 2024-11-09 13:43:15,265] Trial 6 finished with value: 0.6989487209395842 and
parameters: {'method': 'mutual_info', 'n_estimators': 578, 'learning_rate':
0.00974013345377984, 'num_leaves': 23, 'max_depth': 16, 'min_child_samples': 20,
'min_child_weight': 0.025812581204700542, 'subsample': 0.8324451764420886,
'colsample_bytree': 0.5691275779087289, 'lambda_l1': 0.35934258982242906,
```

'lambda_l2': 0.0021136548419622026}. Best is trial 5 with value:
0.6997424766024821.
Optuna Trials Progress: 14%|
| 7/50 [1:09:58<6:49:20, 571.18s/it]

Trial 6 completed - F1 Score: 0.6989, Best F1 Score: 0.6997

[I 2024-11-09 13:51:29,424] Trial 7 finished with value: 0.6994426792874752 and
parameters: {'method': 'mutual_info', 'n_estimators': 678, 'learning_rate':
0.008115404312120126, 'num_leaves': 18, 'max_depth': 8, 'min_child_samples': 10,
'min_child_weight': 0.0026039071057793847, 'subsample': 0.9164724350702601,
'colsample_bytree': 0.5761668491435968, 'lambda_l1': 0.4437882410445249,
'lambda_l2': 0.0012783481106497091}. Best is trial 5 with value:
0.6997424766024821.
Optuna Trials Progress: 16%|
| 8/50 [1:18:12<6:22:39, 546.66s/it]

Trial 7 completed - F1 Score: 0.6994, Best F1 Score: 0.6997

[I 2024-11-09 13:57:22,998] Trial 8 finished with value: 0.6949751299603436 and
parameters: {'method': 'chi2', 'n_estimators': 627, 'learning_rate':
0.01006760137421175, 'num_leaves': 10, 'max_depth': 12, 'min_child_samples': 12,
'min_child_weight': 0.08335444924498518, 'subsample': 0.8328485410175201,
'colsample_bytree': 0.5096848334696149, 'lambda_l1': 0.10293788217034536,
'lambda_l2': 0.0018647012107114512}. Best is trial 5 with value:
0.6997424766024821.
Optuna Trials Progress: 18%|
| 9/50 [1:24:05<5:32:18, 486.30s/it]

Trial 8 completed - F1 Score: 0.6950, Best F1 Score: 0.6997

[I 2024-11-09 14:06:19,372] Trial 9 finished with value: 0.6988029043973946 and
parameters: {'method': 'mutual_info', 'n_estimators': 619, 'learning_rate':
0.010903591792561129, 'num_leaves': 21, 'max_depth': 14, 'min_child_samples':
14, 'min_child_weight': 0.08492041802380491, 'subsample': 0.8981301951348848,
'colsample_bytree': 0.6749711245546355, 'lambda_l1': 0.10231506964408842,
'lambda_l2': 0.002110022376944986}. Best is trial 5 with value:
0.6997424766024821.
Optuna Trials Progress: 20%|
| 10/50 [1:33:02<5:34:30, 501.76s/it]

Trial 9 completed - F1 Score: 0.6988, Best F1 Score: 0.6997

[I 2024-11-09 14:20:15,484] Trial 10 finished with value: 0.6941653700297841 and
parameters: {'method': 'chi2', 'n_estimators': 774, 'learning_rate':
0.008602573817131266, 'num_leaves': 28, 'max_depth': 10, 'min_child_samples':
17, 'min_child_weight': 0.0050046332239653425, 'subsample': 0.9959895721805444,
'colsample_bytree': 0.8547824187232378, 'lambda_l1': 0.15883402311182432,
'lambda_l2': 0.004591326626568717}. Best is trial 5 with value:
0.6997424766024821.
Optuna Trials Progress: 22%|
| 11/50 [1:46:58<6:32:39, 604.09s/it]

```
Trial 10 completed - F1 Score: 0.6942, Best F1 Score: 0.6997

[I 2024-11-09 14:27:47,876] Trial 11 finished with value: 0.6986074682541398 and
parameters: {'method': 'mutual_info', 'n_estimators': 691, 'learning_rate':
0.008035364883963322, 'num_leaves': 16, 'max_depth': 8, 'min_child_samples': 15,
'min_child_weight': 0.004104649445960542, 'subsample': 0.9443553565737817,
'colsample_bytree': 0.5366318815513, 'lambda_l1': 0.336679415434775,
'lambda_l2': 0.0031917871971142986}. Best is trial 5 with value:
0.6997424766024821.
Optuna Trials Progress:  24%|
| 12/50 [1:54:30<5:53:21, 557.94s/it]

Trial 11 completed - F1 Score: 0.6986, Best F1 Score: 0.6997

[I 2024-11-09 14:35:00,233] Trial 12 finished with value: 0.6971417115762918 and
parameters: {'method': 'mutual_info', 'n_estimators': 522, 'learning_rate':
0.00888854307728752, 'num_leaves': 17, 'max_depth': 8, 'min_child_samples': 13,
'min_child_weight': 0.002339737722428696, 'subsample': 0.776662771930227,
'colsample_bytree': 0.6064402770591645, 'lambda_l1': 0.49903149908511624,
'lambda_l2': 0.0014363151512491016}. Best is trial 5 with value:
0.6997424766024821.
Optuna Trials Progress:  26%|
| 13/50 [2:01:43<5:20:36, 519.90s/it]

Trial 12 completed - F1 Score: 0.6971, Best F1 Score: 0.6997

[I 2024-11-09 14:45:15,124] Trial 13 finished with value: 0.695803766786279 and
parameters: {'method': 'mutual_info', 'n_estimators': 811, 'learning_rate':
0.00814660629415372, 'num_leaves': 26, 'max_depth': 10, 'min_child_samples': 16,
'min_child_weight': 0.008395922965909944, 'subsample': 0.7074565351160615,
'colsample_bytree': 0.5046068634083083, 'lambda_l1': 0.16869099215377786,
'lambda_l2': 0.002800153506201209}. Best is trial 5 with value:
0.6997424766024821.
Optuna Trials Progress:  28%|
| 14/50 [2:11:58<5:29:09, 548.59s/it]

Trial 13 completed - F1 Score: 0.6958, Best F1 Score: 0.6997

[W 2024-11-09 14:46:39,304] Trial 14 failed with parameters: {'method': 'chi2'}
because of the following error: KeyboardInterrupt().
Traceback (most recent call last):
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\optuna\study\_optimize.py", line 197, in _run_trial
    value_or_values = func(trial)
  File "C:\Users\emi_r\AppData\Local\Temp\ipykernel_4648\1408981353.py", line
64, in <lambda>
    study.optimize(lambda trial: objective(trial, X, y), n_trials=n_trials,
callbacks=[callback])
  File "C:\Users\emi_r\AppData\Local\Temp\ipykernel_4648\1408981353.py", line
26, in objective
    X_reduced = selector.fit_transform(X, y)
```

```
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\_set_output.py", line 316, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\base.py", line 1101, in fit_transform
    return self.fit(X, y, **fit_params).transform(X)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\base.py", line 1473, in wrapper
    return fit_method(estimator, *args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_univariate_selection.py", line 567, in fit
    score_func_ret = self.score_func(X, y)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\_param_validation.py", line 186, in wrapper
    return func(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_mutual_info.py", line 571, in
mutual_info_classif
    return _estimate_mi(
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_mutual_info.py", line 317, in _estimate_mi
    mi = Parallel(n_jobs=n_jobs)(
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 74, in __call__
    return super().__call__(iterable_with_config)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1918, in __call__
    return output if self.return_generator else list(output)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1847, in _get_sequential_output
    res = func(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 136, in __call__
    return self.function(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_mutual_info.py", line 167, in _compute_mi
    return _compute_mi_cd(x, y, n_neighbors)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\feature_selection\_mutual_info.py", line 129, in _compute_mi_cd
    r = nn.kneighbors()[0]
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\neighbors\_base.py", line 903, in kneighbors
    chunked_results = Parallel(n_jobs, prefer="threads")(
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 74, in __call__
    return super().__call__(iterable_with_config)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1918, in __call__
```

```
    return output if self.return_generator else list(output)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1847, in _get_sequential_output
    res = func(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 136, in __call__
    return self.function(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\neighbors\_base.py", line 704, in _tree_query_parallel_helper
    return tree.query(*args, **kwargs)
KeyboardInterrupt
[W 2024-11-09 14:46:39,306] Trial 14 failed with value None.
Optuna Trials Progress:  28%|
| 14/50 [2:13:22<5:42:57, 571.59s/it]
```

[16]:

```python
# Further dimensionality reduction with GPU acceleration and F1_scores for
# different randomized hyper-parameter combinations

# Define the evaluation metric
f1_scorer = make_scorer(f1_score, average='macro')

# Function to optimize with Optuna
def objective(trial, X, y):
    # Step 1: Optimize `k` for feature selection
    k = 5000

    # Choose the feature selection method
    feature_selection_method = trial.suggest_categorical('method', ['chi2',␣
 ↪'mutual_info'])

    # Automatically switch to mutual_info_classif if the data contains negative␣
 ↪values
    if feature_selection_method == 'chi2' and (X < 0).any():
        feature_selection_method = "mutual_info"
        selector = SelectKBest(mutual_info_classif, k=k)

    elif feature_selection_method == 'chi2':
        feature_selection_method = "mutual_info"
        selector = SelectKBest(mutual_info_classif, k=k)
    else:
        selector = SelectKBest(mutual_info_classif, k=k)

    X_reduced = selector.fit_transform(X, y)

    # Step 2: Optimize LightGBM hyperparameters
    params = {
        'objective': 'binary',
```

```
        'n_estimators': trial.suggest_int('n_estimators', 800, 1100),
        'learning_rate': trial.suggest_float('learning_rate', 0.008, 0.011, log␣
↪= True),
        'num_leaves': trial.suggest_int('num_leaves', 10, 35),
        'max_depth': trial.suggest_int('max_depth', 15, 18),
        'min_child_samples': trial.suggest_int('min_child_samples', 10, 20),
        'min_child_weight': trial.suggest_float('min_child_weight', 1e-3, 1e-1,␣
↪log = True),
        'subsample': trial.suggest_float('subsample', 0.7, 1.0),
        'colsample_bytree': trial.suggest_float('colsample_bytree', 0.5, 0.9),
        'lambda_l1': trial.suggest_float('lambda_l1', 0.1, 0.5, log = True),
        'lambda_l2': trial.suggest_float('lambda_l2', 1e-3, 0.005, log = True),
        'scale_pos_weight': len(y[y == 0]) / len(y[y == 1]),
        'device': 'gpu',  # Use GPU acceleration
        'random_state': 42,
        'n_jobs': -1
    }

    model = LGBMClassifier(**params)

    # Evaluate using cross-validation
    scores = cross_val_score(model, X_reduced, y, cv=3, scoring=f1_scorer,␣
↪n_jobs=-1)
    return scores.mean()

# Optimize hyperparameters using Optuna with a progress bar
def optimize_with_optuna(X, y, n_trials=50):
    study = optuna.create_study(direction='maximize')

    with tqdm(total=n_trials, desc="Optuna Trials Progress") as pbar:
        def callback(study, trial):
            # Update progress bar on each completed trial
            pbar.update(1)
            # Print the best trial so far
            print(f"Trial {trial.number} completed - F1 Score: {trial.value:.
↪4f}, Best F1 Score: {study.best_value:.4f}")

        study.optimize(lambda trial: objective(trial, X, y), n_trials=n_trials,␣
↪callbacks=[callback])

    print(f"\nBest F1 Score: {study.best_value:.4f}")
    print("Best Parameters:", study.best_params)

    # Visualize optimization results
    plot_optimization_history(study).show()
    plot_param_importances(study).show()
    plot_parallel_coordinate(study).show()
```

```
    plot_slice(study).show()
    plot_contour(study).show()

    return study.best_params

# Load your preprocessed data after applying SVD
print("\nApplying Optuna Optimization on SVD Data...")
best_params = optimize_with_optuna(X_svd_optimal, y, n_trials=50)

# Train final model using the best parameters found
def train_final_model(X, y, params):
    k = params.pop('k')
    method = params.pop('method')

    if method == 'chi2':
        selector = SelectKBest(chi2, k=k)
    else:
        selector = SelectKBest(mutual_info_classif, k=k)

    X_reduced = selector.fit_transform(X, y)
    model = LGBMClassifier(**params)
    scores = cross_val_score(model, X_reduced, y, cv=3, scoring=f1_scorer,␣
 ↪n_jobs=-1)
    print(f"\nFinal Model F1 Score: {scores.mean():.4f}")

print("\nTraining Final Model with Optimized Parameters...")
train_final_model(X_svd_optimal, y, best_params)
```

```
[I 2024-11-09 15:18:05,371] A new study created in memory with name: no-name-
bf07ec09-cb27-482f-bb14-eab3eb4e893a


Applying Optuna Optimization on SVD Data...

Optuna Trials Progress:   0%|
| 0/50 [00:00<?, ?it/s][I 2024-11-09 15:35:58,380] Trial 0 finished with value:
0.6798648219376991 and parameters: {'method': 'chi2', 'n_estimators': 1050,
'learning_rate': 0.00847845995291866, 'num_leaves': 33, 'max_depth': 18,
'min_child_samples': 15, 'min_child_weight': 0.04090500804279365, 'subsample':
0.752039872067227, 'colsample_bytree': 0.7300477177557985, 'lambda_l1':
0.4200000044292351, 'lambda_l2': 0.004549231685950845}. Best is trial 0 with
value: 0.6798648219376991.
Optuna Trials Progress:   2%|
| 1/50 [17:53<14:36:17, 1073.01s/it]

Trial 0 completed - F1 Score: 0.6799, Best F1 Score: 0.6799

[I 2024-11-09 15:43:50,243] Trial 1 finished with value: 0.6977905947557176 and
parameters: {'method': 'mutual_info', 'n_estimators': 1054, 'learning_rate':
0.010724349165623485, 'num_leaves': 10, 'max_depth': 15, 'min_child_samples':
```

```
18, 'min_child_weight': 0.0015205769370585478, 'subsample': 0.939117189600813,
'colsample_bytree': 0.6759421127051858, 'lambda_l1': 0.17993160574221476,
'lambda_l2': 0.004400850749727704}. Best is trial 1 with value:
0.6977905947557176.
Optuna Trials Progress:   4%|
| 2/50 [25:44<9:35:30, 719.39s/it]

Trial 1 completed - F1 Score: 0.6978, Best F1 Score: 0.6978

[I 2024-11-09 15:53:38,798] Trial 2 finished with value: 0.6930264553285609 and
parameters: {'method': 'chi2', 'n_estimators': 1039, 'learning_rate':
0.009430585249899075, 'num_leaves': 12, 'max_depth': 18, 'min_child_samples':
14, 'min_child_weight': 0.002181030285158791, 'subsample': 0.8636607791297642,
'colsample_bytree': 0.8702736210160187, 'lambda_l1': 0.17662285370088554,
'lambda_l2': 0.0011698144188551122}. Best is trial 1 with value:
0.6977905947557176.
Optuna Trials Progress:   6%|
| 3/50 [35:33<8:36:43, 659.65s/it]

Trial 2 completed - F1 Score: 0.6930, Best F1 Score: 0.6978

[I 2024-11-09 16:05:43,382] Trial 3 finished with value: 0.6983346185490896 and
parameters: {'method': 'mutual_info', 'n_estimators': 998, 'learning_rate':
0.008339529649628672, 'num_leaves': 21, 'max_depth': 18, 'min_child_samples':
11, 'min_child_weight': 0.002228926327306755, 'subsample': 0.9383876387300494,
'colsample_bytree': 0.6830674306169544, 'lambda_l1': 0.1546340503850816,
'lambda_l2': 0.004171344295072093}. Best is trial 3 with value:
0.6983346185490896.
Optuna Trials Progress:   8%|
| 4/50 [47:38<8:45:23, 685.28s/it]

Trial 3 completed - F1 Score: 0.6983, Best F1 Score: 0.6983

[I 2024-11-09 16:18:06,401] Trial 4 finished with value: 0.6974358160061835 and
parameters: {'method': 'mutual_info', 'n_estimators': 1042, 'learning_rate':
0.008273246933733323, 'num_leaves': 21, 'max_depth': 18, 'min_child_samples':
16, 'min_child_weight': 0.003037589038022129, 'subsample': 0.7894923508050208,
'colsample_bytree': 0.6716274952515652, 'lambda_l1': 0.10442748271896009,
'lambda_l2': 0.0016061070182448144}. Best is trial 3 with value:
0.6983346185490896.
Optuna Trials Progress:  10%|
| 5/50 [1:00:01<8:49:34, 706.10s/it]

Trial 4 completed - F1 Score: 0.6974, Best F1 Score: 0.6983

[I 2024-11-09 16:32:49,487] Trial 5 finished with value: 0.6796402783699462 and
parameters: {'method': 'mutual_info', 'n_estimators': 852, 'learning_rate':
0.010921850217731255, 'num_leaves': 32, 'max_depth': 18, 'min_child_samples':
18, 'min_child_weight': 0.028474879137336366, 'subsample': 0.9702598540850994,
'colsample_bytree': 0.7120581889213304, 'lambda_l1': 0.4562336354092253,
'lambda_l2': 0.0010794600504231149}. Best is trial 3 with value:
0.6983346185490896.
```

```
Optuna Trials Progress:  12%|
| 6/50 [1:14:44<9:21:56, 766.28s/it]

Trial 5 completed - F1 Score: 0.6796, Best F1 Score: 0.6983

[I 2024-11-09 16:52:06,391] Trial 6 finished with value: 0.68209416103972 and
parameters: {'method': 'chi2', 'n_estimators': 976, 'learning_rate':
0.008828359610512171, 'num_leaves': 32, 'max_depth': 16, 'min_child_samples':
17, 'min_child_weight': 0.005296567061000446, 'subsample': 0.9960855746201465,
'colsample_bytree': 0.8824295833961453, 'lambda_l1': 0.2201702505726328,
'lambda_l2': 0.0010089605473625827}. Best is trial 3 with value:
0.6983346185490896.
Optuna Trials Progress:  14%|
| 7/50 [1:34:01<10:40:41, 893.98s/it]

Trial 6 completed - F1 Score: 0.6821, Best F1 Score: 0.6983

[I 2024-11-09 17:01:58,147] Trial 7 finished with value: 0.7001367094865953 and
parameters: {'method': 'chi2', 'n_estimators': 1087, 'learning_rate':
0.009382545971421487, 'num_leaves': 13, 'max_depth': 17, 'min_child_samples':
14, 'min_child_weight': 0.00118617817537115, 'subsample': 0.891661431012198,
'colsample_bytree': 0.7441311682245657, 'lambda_l1': 0.277922414142598,
'lambda_l2': 0.0013715968271366924}. Best is trial 7 with value:
0.7001367094865953.
Optuna Trials Progress:  16%|
| 8/50 [1:43:52<9:18:26, 797.77s/it]

Trial 7 completed - F1 Score: 0.7001, Best F1 Score: 0.7001

[I 2024-11-09 17:13:32,323] Trial 8 finished with value: 0.6915112234626059 and
parameters: {'method': 'chi2', 'n_estimators': 891, 'learning_rate':
0.00968423543216792, 'num_leaves': 27, 'max_depth': 15, 'min_child_samples': 11,
'min_child_weight': 0.0043197747574356725, 'subsample': 0.8943782483877747,
'colsample_bytree': 0.519107309070411, 'lambda_l1': 0.1522340264735486,
'lambda_l2': 0.003289849118290137}. Best is trial 7 with value:
0.7001367094865953.
Optuna Trials Progress:  18%|
| 9/50 [1:55:26<8:43:00, 765.38s/it]

Trial 8 completed - F1 Score: 0.6915, Best F1 Score: 0.7001

[W 2024-11-09 17:19:44,646] Trial 9 failed with parameters: {'method': 'chi2',
'n_estimators': 1003, 'learning_rate': 0.008374075645216441, 'num_leaves': 16,
'max_depth': 15, 'min_child_samples': 12, 'min_child_weight':
0.057087991149234046, 'subsample': 0.7777826621377115, 'colsample_bytree':
0.7274754249936122, 'lambda_l1': 0.33646450846829307, 'lambda_l2':
0.0011630416638220137} because of the following error: KeyboardInterrupt().
Traceback (most recent call last):
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\optuna\study\_optimize.py", line 197, in _run_trial
    value_or_values = func(trial)
  File "C:\Users\emi_r\AppData\Local\Temp\ipykernel_4648\1832754857.py", line
```

```
64, in <lambda>
    study.optimize(lambda trial: objective(trial, X, y), n_trials=n_trials,
callbacks=[callback])
  File "C:\Users\emi_r\AppData\Local\Temp\ipykernel_4648\1832754857.py", line
50, in objective
    scores = cross_val_score(model, X_reduced, y, cv=3, scoring=f1_scorer,
n_jobs=-1)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\_param_validation.py", line 213, in wrapper
    return func(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\model_selection\_validation.py", line 712, in cross_val_score
    cv_results = cross_validate(
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\_param_validation.py", line 213, in wrapper
    return func(*args, **kwargs)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\model_selection\_validation.py", line 423, in cross_validate
    results = parallel(
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\sklearn\utils\parallel.py", line 74, in __call__
    return super().__call__(iterable_with_config)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 2007, in __call__
    return output if self.return_generator else list(output)
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1650, in _get_outputs
    yield from self._retrieve()
  File "D:\miniCondaa\envs\gpu_kaggle1IFT3395\lib\site-
packages\joblib\parallel.py", line 1762, in _retrieve
    time.sleep(0.01)
KeyboardInterrupt
[W 2024-11-09 17:19:44,654] Trial 9 failed with value None.
Optuna Trials Progress:  18%|
| 9/50 [2:01:39<9:14:12, 811.03s/it]
```

[21]:
```
df_test = pd.DataFrame(data_test)

data_test_tfidf = tfidf_transformer.fit_transform(df_test)

data_test = svd_optimal.transform(data_test_tfidf)

print(data_test)
```

```
[[ 3.08304737e-01  4.42997915e-02  3.40074637e-03 ...  6.22914683e-03
  -5.93609243e-04 -4.24540480e-03]
 [ 3.24578814e-01  4.05390551e-02  6.04733872e-02 ... -1.97552263e-03
  -9.57668820e-03  5.16290543e-03]
```

```
[ 3.43138687e-01  1.03279088e-01 -5.07147022e-02 ... -1.42421415e-04
 -4.32751164e-03 -1.13001943e-02]
...
[ 3.33930928e-01 -8.88954928e-02 -4.08795741e-02 ... -6.80090850e-04
 -5.55554963e-04 -4.25924417e-03]
[ 3.06479371e-01  3.96557887e-02 -3.50367504e-02 ...  2.59855955e-03
  5.87417768e-03 -8.88125952e-03]
[ 2.31641716e-01  1.40174678e-02  6.62313916e-02 ...  3.92276494e-03
  8.97419554e-03 -2.94823902e-03]]
```

[22]:
```python
# Define the evaluation metric
f1_scorer = make_scorer(f1_score, average='macro')

# Best hyperparameters from Optuna
best_params = {
    'method': 'mutual_info_classif',
    'n_estimators': 884,
    'learning_rate': 0.008552871705252547,
    'num_leaves': 18,
    'max_depth': 15,
    'min_child_samples': 12,
    'min_child_weight': 0.02433182001711402,
    'subsample': 0.9022878985957729,
    'colsample_bytree': 0.688174345860965,
    'lambda_l1': 0.2173588901875657,
    'lambda_l2': 0.002597627293608841,
    'objective': 'binary',
    'scale_pos_weight': len(y[y == 0]) / len(y[y == 1]),
    'device': 'gpu',  # Ensure GPU acceleration
    'random_state': 42,
    'n_jobs': -1
}

# Train final model using the best parameters found
def train_final_model(X, y, params, data_test):
    k = 5000  # Fixed value for k
    method = params.pop('method')

    # Step 1: Feature selection
    if method == 'chi2':
        selector = SelectKBest(mutual_info_classif, k=k)
    else:
        selector = SelectKBest(mutual_info_classif, k=k)

    # Apply feature selection on training data
    X_reduced = selector.fit_transform(X, y)
```

```
    # Step 2: Train the model using the best parameters
    model = LGBMClassifier(**params)
    model.fit(X_reduced, y)

    # Apply the same feature selection on test data
    X_test_reduced = selector.transform(data_test)

    # Make predictions on the test set
    y_test_pred = model.predict(X_test_reduced)

    # Reshape the output as requested
    IDs = np.array(range(len(y_test_pred)))
    output = np.hstack((IDs.reshape(len(IDs), 1), y_test_pred.
 ↪reshape(len(y_test_pred), 1)))

    # Save the predicted labels for the test set to a CSV file
    np.savetxt('test_predictions_lightGBM.csv', output, delimiter=',', fmt='%d',␣
 ↪header='ID,label', comments='')
    print("\nPredictions saved to 'test_predictions_lightGBM.csv'")

# Load your preprocessed data after applying SVD
print("\nTraining final model with the best parameters and generating␣
 ↪predictions...")
train_final_model(X_svd_optimal, y, best_params, data_test)
```

```
Training final model with the best parameters and generating predictions...
[LightGBM] [Warning] lambda_l1 is set=0.2173588901875657, reg_alpha=0.0 will be
ignored. Current value: lambda_l1=0.2173588901875657
[LightGBM] [Warning] lambda_l2 is set=0.002597627293608841, reg_lambda=0.0 will
be ignored. Current value: lambda_l2=0.002597627293608841
[LightGBM] [Warning] lambda_l1 is set=0.2173588901875657, reg_alpha=0.0 will be
ignored. Current value: lambda_l1=0.2173588901875657
[LightGBM] [Warning] lambda_l2 is set=0.002597627293608841, reg_lambda=0.0 will
be ignored. Current value: lambda_l2=0.002597627293608841
[LightGBM] [Info] Number of positive: 2298, number of negative: 7124
[LightGBM] [Info] This is the GPU trainer!!
[LightGBM] [Info] Total Bins 1275000
[LightGBM] [Info] Number of data points in the train set: 9422, number of used
features: 5000
[LightGBM] [Info] Using GPU Device: NVIDIA GeForce GTX 1050, Vendor: NVIDIA
Corporation
[LightGBM] [Info] Compiling OpenCL Kernel with 256 bins...
[LightGBM] [Info] GPU programs have been built
[LightGBM] [Info] Size of histogram bin entry: 8
[LightGBM] [Info] 5000 dense feature groups (44.93 MB) transferred to GPU in
0.077283 secs. 0 sparse feature groups
```

```
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.243897 -> initscore=-1.131430
[LightGBM] [Info] Start training from score -1.131430
[LightGBM] [Info] Increasing preallocd_max_num_wg_ to 1250 for launching more
workgroups
[LightGBM] [Warning] lambda_l1 is set=0.2173588901875657, reg_alpha=0.0 will be
ignored. Current value: lambda_l1=0.2173588901875657
[LightGBM] [Warning] lambda_l2 is set=0.002597627293608841, reg_lambda=0.0 will
be ignored. Current value: lambda_l2=0.002597627293608841

Predictions saved to 'test_predictions_lightGBM.csv'
```

[9]:
```python
import matplotlib.pyplot as plt
import numpy as np

# Data for optimal and non-optimal hyperparameters
optimal_f1_scores = [0.6984, 0.6981, 0.7002, 0.6967, 0.7049]
non_optimal_f1_scores = [
    0.6614, 0.6567, 0.6674, 0.6676, 0.6855, 0.6235, 0.6389, 0.6350, 0.6645, 0.
 ↪6715,
    0.6357, 0.6220, 0.6708, 0.6543, 0.6786, 0.6799, 0.6756, 0.6783, 0.6984, 0.
 ↪7002,
    0.6979, 0.6726, 0.6493, 0.6982, 0.6753, 0.6400, 0.6970, 0.6862, 0.6934, 0.
 ↪6973,
    0.6817, 0.6514, 0.6679, 0.6690, 0.6898, 0.6946, 0.6946, 0.6903, 0.6870, 0.
 ↪6904,
    0.6532, 0.6705, 0.6816, 0.6436, 0.6660, 0.6322, 0.6972, 0.6950, 0.6976
]

# Ajuster le graphique pour aligner les essais optimaux plus près de la moitié␣
 ↪des essais non optimaux
trials_non_optimal = np.arange(1, len(non_optimal_f1_scores) + 1)

# Ajuster les essais optimaux pour commencer plus près de la moitié des essais␣
 ↪non optimaux
mid_point = len(non_optimal_f1_scores) // 2
trials_optimal_adjusted_mid = np.arange(mid_point, mid_point +␣
 ↪len(optimal_f1_scores))

# Tracé des scores F1 pour les hyperparamètres optimaux et non optimaux
plt.figure(figsize=(12, 6))

# Tracé des hyperparamètres optimaux (décalé pour aligner vers la moitié des non␣
 ↪optimaux)
plt.plot(trials_optimal_adjusted_mid, optimal_f1_scores, label='Optimal␣
 ↪Hyperparameters', marker='o', linestyle='-', color='blue')

# Tracé des hyperparamètres non optimaux
```

```
plt.plot(trials_non_optimal, non_optimal_f1_scores, label='Non-Optimal␣
 ↪Hyperparameters', marker='o', linestyle='--', color='orange')

# Configuration du graphique
plt.title("Comparaison des F1 Scores pour les hyperparamètres optimaux et non␣
 ↪optimaux (Aligné à mi-chemin)")
plt.xlabel("Numéro d'essai")
plt.ylabel("F1 Score")
plt.xticks(range(1, len(non_optimal_f1_scores) + 1, 5))
plt.ylim(0.60, 0.72)
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```