

```
In [26]: import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score
from sklearn.feature_selection import SelectKBest, mutual_info_classif, SelectFromModel
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [27]: df = pd.read_csv('CleanLoanData.csv')
```

```
In [28]: df.head()
```

```
Out[28]:
```

	Unnamed: 0	Loan_ID	Gender	Married	Dependents	Education	Self_Employed
0	0	0	1	0	0	0	0
1	1	1	1	1	1	0	0
2	2	2	1	1	0	0	1
3	3	3	1	1	0	1	0
4	4	4	1	0	0	0	0

```
In [29]: df.columns
```

```
Out[29]: Index(['Unnamed: 0', 'Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
               'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
               'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
              dtype='object')
```

```
In [30]: df.isnull().sum()
```

```
Out[30]: Unnamed: 0      0
         Loan_ID      0
         Gender      0
         Married      0
         Dependents   0
         Education    0
         Self_Employed 0
         ApplicantIncome 0
         CoapplicantIncome 0
         LoanAmount    0
         Loan_Amount_Term 0
         Credit_History 0
         Property_Area 0
         Loan_Status    0
         dtype: int64
```

```
In [31]: df = df.drop(columns=['Unnamed: 0', 'Loan_ID'])
```

```
In [32]: df.columns
```

```
Out[32]: Index(['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed',
               'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
               'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
              dtype='object')
```

```
In [33]: df['Loan_Status'].value_counts()
```

```
Out[33]: Loan_Status
1      422
0      192
Name: count, dtype: int64
```

```
In [34]: X = df.drop('Loan_Status', axis=1).values
         y = df['Loan_Status'].values
```

```
In [35]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30,
```

```
In [36]: def evaluate_model(model_name, y_true, y_pred):

    # Calculate metrics
    accuracy = accuracy_score(y_true, y_pred)
    precision = precision_score(y_true, y_pred, average='weighted')
    recall = recall_score(y_true, y_pred, average='weighted')
    f1 = f1_score(y_true, y_pred, average='weighted')
    cm = confusion_matrix(y_true, y_pred)

    # Create a report
    report = classification_report(y_true, y_pred)

    # Output results
    metrics = {
        'Model Name': model_name,
        'Accuracy': accuracy,
        'Precision': precision,
        'Recall': recall,
        'F1 Score': f1,
```

```

'Classification Report': report
}

# Plot Confusion Matrix
plt.figure(figsize=(4, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False,
            xticklabels=np.unique(y_true), yticklabels=np.unique(y_true))
plt.title(f'Confusion Matrix for {model_name}')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()

return metrics

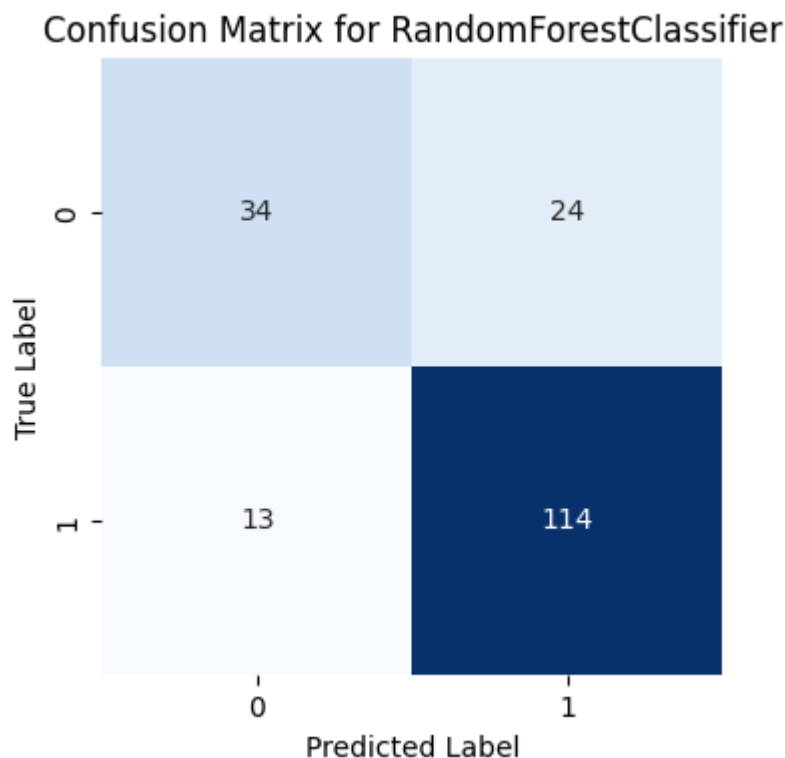
```

```

In [37]: rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
# Train the model
rf_classifier.fit(X_train, y_train)
# Make predictions
y_pred = rf_classifier.predict(X_test)
# Evaluate the model
evaluation_results = evaluate_model('RandomForestClassifier', y_test, y_pr

# Print the evaluation results
for key, value in evaluation_results.items():
    if key == 'Classification Report':
        print(value) # Print report separately for better readability
    else:
        print(f"{key}: {value:.4f}" if isinstance(value, float) else f"{key}

```



Model Name:
 RandomForestClassifier
 Accuracy: 0.8000
 Precision: 0.7939
 Recall: 0.8000
 F1 Score: 0.7937

	precision	recall	f1-score	support
0	0.72	0.59	0.65	58
1	0.83	0.90	0.86	127
accuracy			0.80	185
macro avg	0.77	0.74	0.75	185
weighted avg	0.79	0.80	0.79	185

In [38]: *#LogisticRegression*

```
LR = LogisticRegression()
# Train the model
LR.fit(X_train, y_train)
# Make predictions
y_pred = LR.predict(X_test)
# Evaluate the model
evaluation_results = evaluate_model('LogisticRegression', y_test, y_pred)
# Print the evaluation results
for key, value in evaluation_results.items():
    if key == 'Classification Report':
        print(value) # Print report separately for better readability
    else:
        print(f"{key}: {value:.4f}" if isinstance(value, float) else f"{key}
```

/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):

STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=1000).

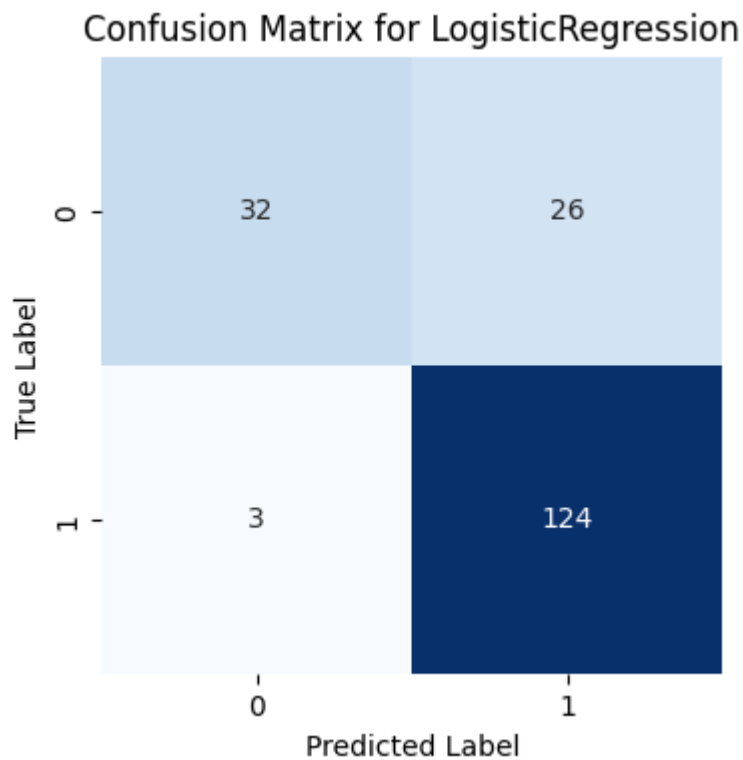
You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
```



Model Name:

LogisticRegression

Accuracy: 0.8432

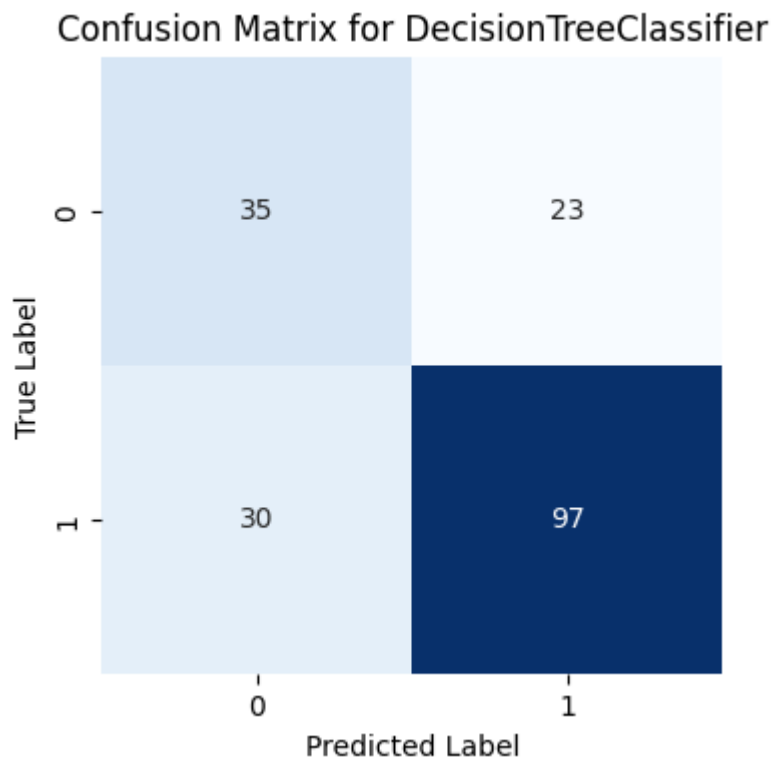
Precision: 0.8541

Recall: 0.8432

F1 Score: 0.8304

	precision	recall	f1-score	support
0	0.91	0.55	0.69	58
1	0.83	0.98	0.90	127
accuracy			0.84	185
macro avg	0.87	0.76	0.79	185
weighted avg	0.85	0.84	0.83	185

```
In [39]: DT = DecisionTreeClassifier()
# Train the model
DT.fit(X_train, y_train)
# Make predictions
y_pred = DT.predict(X_test)
# Evaluate the model
evaluation_results = evaluate_model('DecisionTreeClassifier', y_test, y_pr
# Print the evaluation results
for key, value in evaluation_results.items():
    if key == 'Classification Report':
        print(value) # Print report separately for better readability
    else:
        print(f"{key}: {value:.4f}" if isinstance(value, float) else f"{key}:
```



Model Name:

DecisionTreeClassifier

Accuracy: 0.7135

Precision: 0.7237

Recall: 0.7135

F1 Score: 0.7176

	precision	recall	f1-score	support
0	0.54	0.60	0.57	58
1	0.81	0.76	0.79	127
accuracy			0.71	185
macro avg	0.67	0.68	0.68	185
weighted avg	0.72	0.71	0.72	185

```
In [40]: ##### Random Forest Classifier with grid search

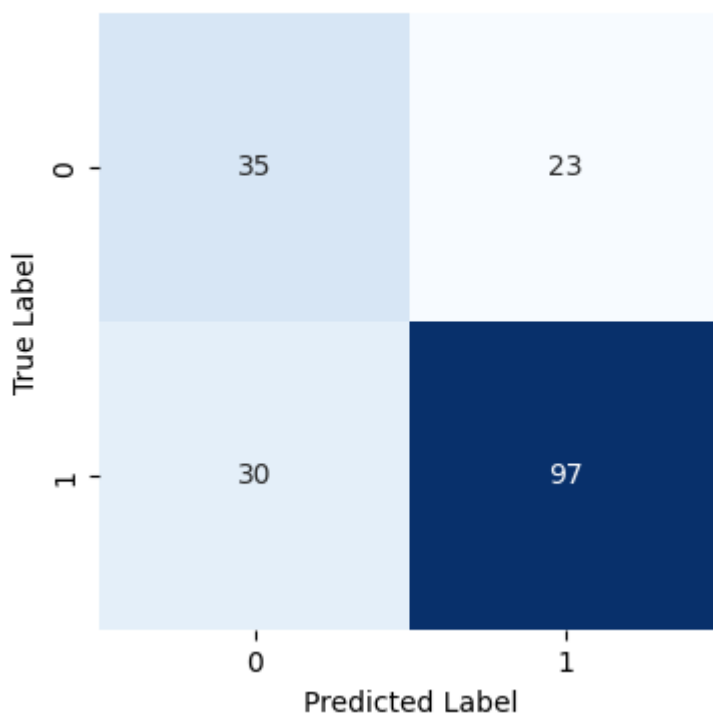
from sklearn.model_selection import GridSearchCV
# Define the Random Forest Classifier
rf_classifier = RandomForestClassifier(random_state=42)
# Define the hyperparameters for grid search
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
# Initialize GridSearchCV
grid_search = GridSearchCV(estimator=rf_classifier, param_grid=param_grid,
# Fit the grid search to the data
grid_search.fit(X_train, y_train)
# Get the best estimator from grid search
best_rf_classifier = grid_search.best_estimator_
# Make predictions using the best model y_pred = best_rf_classifier.predict(X_test)
# Evaluate the model
```

```

evaluation_results = evaluate_model('RandomForestClassifier', y_test, y_pr
# Print the evaluation results
for key, value in evaluation_results.items():
    if key == 'Classification Report':
        print(value) # Print report separately for better readability
    else:
        print(f"{key}: {value:.4f}" if isinstance(value, float) else f"{
# Print the best parameters found by grid search
print("\nBest hyperparameters found by GridSearchCV:")
print(grid_search.best_params_)

```

Confusion Matrix for RandomForestClassifier



Model Name:

RandomForestClassifier

Accuracy: 0.7135

Precision: 0.7237

Recall: 0.7135

F1 Score: 0.7176

	precision	recall	f1-score	support
0	0.54	0.60	0.57	58
1	0.81	0.76	0.79	127
accuracy			0.71	185
macro avg	0.67	0.68	0.68	185
weighted avg	0.72	0.71	0.72	185

Best hyperparameters found by GridSearchCV:

```
{'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}
```

In [74]: `### LogisticRegression with grid search`

```

# Define the Logistic Regression model
LR = LogisticRegression(random_state=42)
# Define the hyperparameters for grid search

```

```
param_grid = {'C': [0.01, 0.1, 1, 10], # Regularization strength
              'solver': ['liblinear', 'lbfgs'], # Solver type
              'max_iter': [100, 200, 300] # Maximum iterations for conver
}
# Initialize GridSearchCV
grid_search = GridSearchCV(estimator=LR, param_grid=param_grid, cv=5, scor
# Fit the grid search to the data
grid_search.fit(X_train, y_train)
# Get the best estimator from grid search
best_LR = grid_search.best_estimator_
# Make predictions using the best model
y_pred = best_LR.predict(X_test)
# Evaluate the model
evaluation_results = evaluate_model('LogisticRegression', y_test, y_pred)
# Print the evaluation results
for key, value in evaluation_results.items():
    if key == 'Classification Report':
        print(value) # Print report separately for better readability
    else:
        print(f"{key}: {value:.4f}" if isinstance(value, float) else f"{
# Print the best hyperparameters found by grid search
print("\nBest hyperparameters found by GridSearchCV:")
print(grid_search.best_params_)
```

```

/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=1000).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=2000).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=2000).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=2000).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=2000).
You might also want to scale the data as shown in:

```

```

https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=200).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=100).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=100).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=100).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):

```

STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=300).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
```

STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=300).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
```

STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=300).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
```

STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=300).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
```

STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=300).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```

ression
    n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=1000).
You might also want to scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
ression
    n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=1000).
You might also want to scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
ression
    n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=1000).
You might also want to scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
ression
    n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=1000).
You might also want to scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
ression
    n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=1000).
You might also want to scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
ression
    n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=1000).

```

```
0).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=200).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=200).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=200).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=200).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear mo
```

```

del/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 2
00 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=20
0).
You might also want to scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-reg
ression
    n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_mo
del/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 1
00 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=10
0).
You might also want to scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-reg
ression
    n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_mo
del/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 3
00 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=30
0).
You might also want to scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-reg
ression
    n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_mo
del/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 3
00 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=30
0).
You might also want to scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-reg
ression
    n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_mo
del/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 3
00 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=30
0).
You might also want to scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html

```

Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=100).

You might also want to scale the data as shown in:
<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=300).

You might also want to scale the data as shown in:
<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=100).

You might also want to scale the data as shown in:
<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=100).

You might also want to scale the data as shown in:
<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

```

Increase the number of iterations to improve the convergence (max_iter=100).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=100).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=200).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=200).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=200).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression

```

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=100).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=100).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=200).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=300).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=100).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=300).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=200).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=300).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=300).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 1
```

```
00 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=100).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=200).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=200).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=200).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=300).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
```

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=100).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=300).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=200).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 200 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence (max_iter=200).

You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

```
Increase the number of iterations to improve the convergence (max_iter=300).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=300).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=300).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=300).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

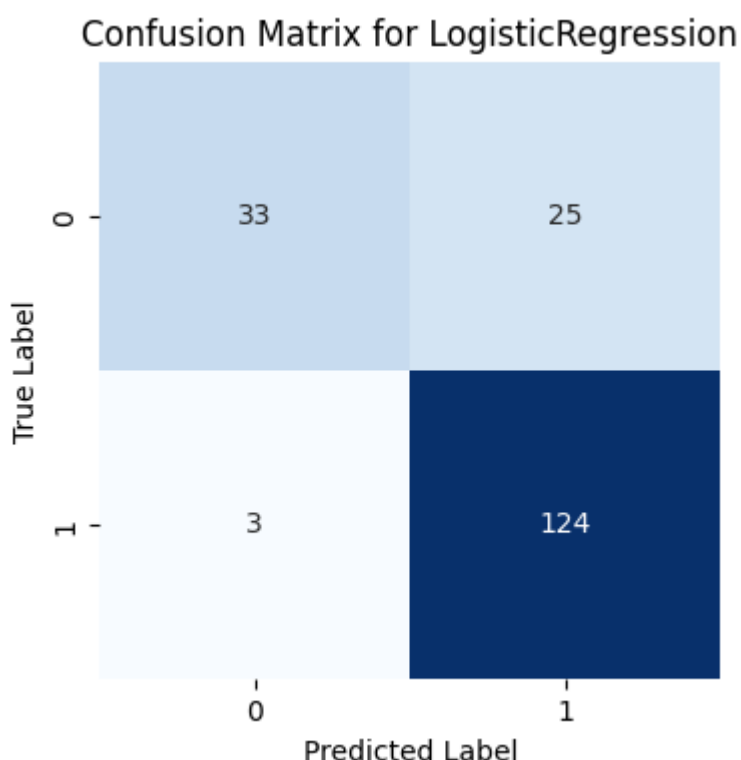
Increase the number of iterations to improve the convergence (max_iter=300).
You might also want to scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
```

```

/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 300 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=3000).
You might also want to scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
n_iter_i = _check_optimize_result(

```



Model Name:
 LogisticRegression
 Accuracy: 0.8486
 Precision: 0.8587
 Recall: 0.8486
 F1 Score: 0.8370

	precision	recall	f1-score	support
0	0.92	0.57	0.70	58
1	0.83	0.98	0.90	127
accuracy			0.85	185
macro avg	0.87	0.77	0.80	185
weighted avg	0.86	0.85	0.84	185

Best hyperparameters found by GridSearchCV:
 {'C': 1, 'max_iter': 300, 'solver': 'lbfgs'}

```

In [75]: import pandas as pd
import numpy as np

df = pd.read_csv("CleanLoanData.csv")

```

```
TARGET = "Loan_Status"

# Drop ID / index-like columns (common in cleaned CSVs)
drop_cols = [c for c in ["Unnamed: 0", "Loan_ID"] if c in df.columns]

X = df.drop(columns=drop_cols + [TARGET])
y = df[TARGET]

print("X shape:", X.shape)
print("Target distribution:\n", y.value_counts())
```

```
X shape: (614, 11)
Target distribution:
Loan_Status
1      422
0      192
Name: count, dtype: int64
```

```
In [76]: # Check missing values
print("Missing values per column:\n", X.isnull().sum().sort_values(ascending=True))

# If you want to fill (safe defaults)
# Numeric -> median
X = X.fillna(X.median(numeric_only=True))

# If you had categorical columns, you'd fill them by mode; your file is a
```

```
Missing values per column:
Gender      0
Married     0
Dependents  0
Education   0
Self_Employed  0
ApplicantIncome  0
CoapplicantIncome  0
LoanAmount  0
Loan_Amount_Term  0
Credit_History  0
dtype: int64
```

```
In [77]: from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

```
In [78]: from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, precision_score, recall_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC

models = {
    "Random Forest": (RandomForestClassifier(random_state=42), {
        "n_estimators": [200, 500],
        "max_depth": [None, 5, 10]
    }),
    "Decision Tree": (DecisionTreeClassifier(random_state=42), {
        "max_depth": [None, 5, 10, 20],
        "min_samples_split": [2, 5, 10]
    })
}
```

```

    }),
    "SVM": (SVC(), {
        "C": [0.1, 1, 10],
        "kernel": ["rbf", "linear"]
    })
}

def evaluate_model(name, best_estimator, X_te, y_te):
    y_pred = best_estimator.predict(X_te)
    return {
        "Model": name,
        "Accuracy": accuracy_score(y_te, y_pred),
        "Precision": precision_score(y_te, y_pred, average="weighted", zero_div
        "Recall": recall_score(y_te, y_pred, average="weighted", zero_div
        "F1": f1_score(y_te, y_pred, average="weighted", zero_division=0)
    }

baseline_results = []
best_estimators_baseline = {}

for name, (model, grid) in models.items():
    gs = GridSearchCV(model, grid, cv=5, scoring="f1_weighted", n_jobs=-1)
    gs.fit(X_train, y_train)
    best_estimators_baseline[name] = gs.best_estimator_
    baseline_results.append(evaluate_model(name, gs.best_estimator_, X_te
    print(name, "best params:", gs.best_params_)

baseline_df = pd.DataFrame(baseline_results).sort_values("F1", ascending=
baseline_df

```

Random Forest best params: {'max_depth': 5, 'n_estimators': 200}

Decision Tree best params: {'max_depth': 5, 'min_samples_split': 10}

SVM best params: {'C': 1, 'kernel': 'linear'}

Out [78]:

	Model	Accuracy	Precision	Recall	F1
0	Random Forest	0.853659	0.869641	0.853659	0.840441
2	SVM	0.845528	0.863337	0.845528	0.830296
1	Decision Tree	0.829268	0.827612	0.829268	0.820010

In [80]: `from sklearn.feature_selection import SelectKBest, chi2`
`import matplotlib.pyplot as plt`

```

k = 5
selector_chi2 = SelectKBest(score_func=chi2, k=k)
selector_chi2.fit(X_train, y_train)

chi2_scores = pd.DataFrame({
    "Feature": X_train.columns,
    "Score": selector_chi2.scores_
}).sort_values("Score", ascending=False)

display(chi2_scores.head(15))

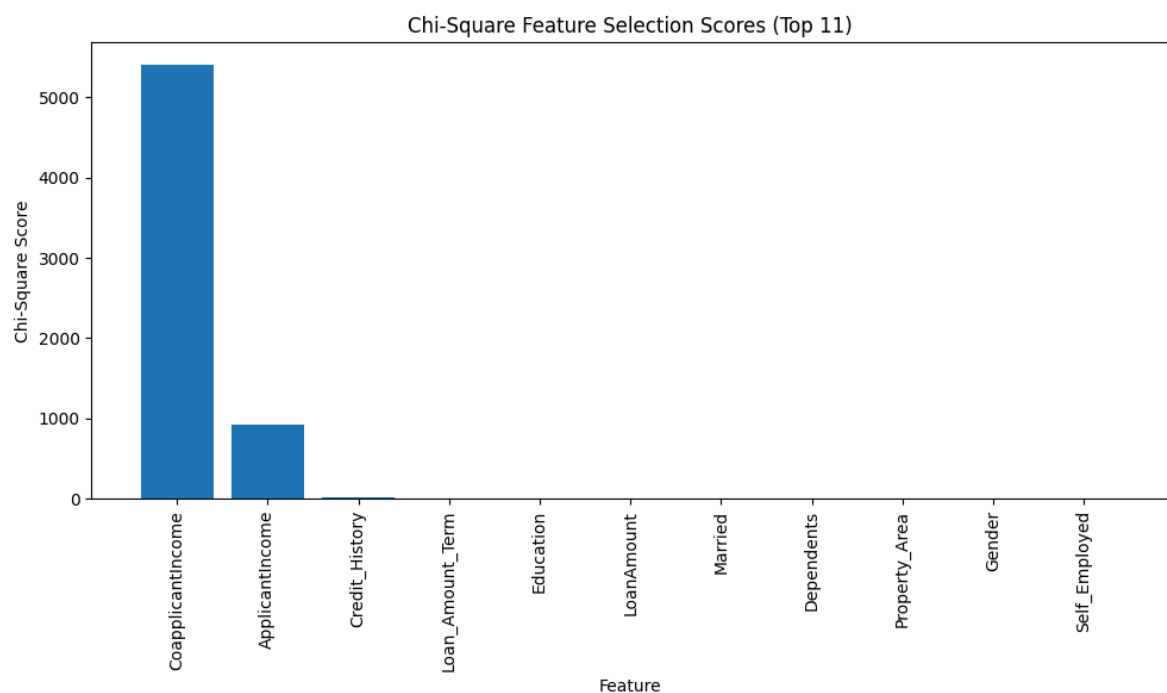
# Plot (top 20 for readability)
top_n = min(20, len(chi2_scores))
plt.figure(figsize=(10, 6))
plt.bar(chi2_scores["Feature"].head(top_n), chi2_scores["Score"].head(top
plt.xlabel("Feature")

```

```
plt.ylabel("Chi-Square Score")
plt.title(f"Chi-Square Feature Selection Scores (Top {top_n})")
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()

selected_features_chi2 = X_train.columns[selector_chi2.get_support()]
selected_features_chi2
```

	Feature	Score
6	CoapplicantIncome	5411.745582
5	ApplicantIncome	919.590309
9	Credit_History	17.733446
8	Loan_Amount_Term	3.851114
3	Education	2.803899
7	LoanAmount	0.949964
1	Married	0.899920
2	Dependents	0.645990
10	Property_Area	0.366629
0	Gender	0.037776
4	Self_Employed	0.005088



```
Out[80]: Index(['Education', 'ApplicantIncome', 'CoapplicantIncome', 'Loan_Amount_Term',
               'Credit_History'],
              dtype='object')
```

```
In [82]: from sklearn.feature_selection import RFE
         from sklearn.linear_model import LogisticRegression

         k = 5
```

```

lr = LogisticRegression(max_iter=2000)
rfe_lr = RFE(estimator=lr, n_features_to_select=k)
rfe_lr.fit(X_train, y_train)

rfe_lr_df = pd.DataFrame({
    "Feature": X_train.columns,
    "Ranking": rfe_lr.ranking_
}).sort_values("Ranking", ascending=True)

display(rfe_lr_df.head(15))

plt.figure(figsize=(10, 6))
plt.bar(rfe_lr_df["Feature"].head(20), rfe_lr_df["Ranking"].head(20))
plt.xlabel("Feature")
plt.ylabel("RFE Ranking (1 = Selected)")
plt.title("RFE Feature Ranking (Logistic Regression)")
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()

selected_features_rfe_lr = X_train.columns[rfe_lr.support_]
selected_features_rfe_lr

```

/opt/anaconda3/envs/Meshary/lib/python3.13/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 2000 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max_iter=2000).

You might also want to scale the data as shown in:

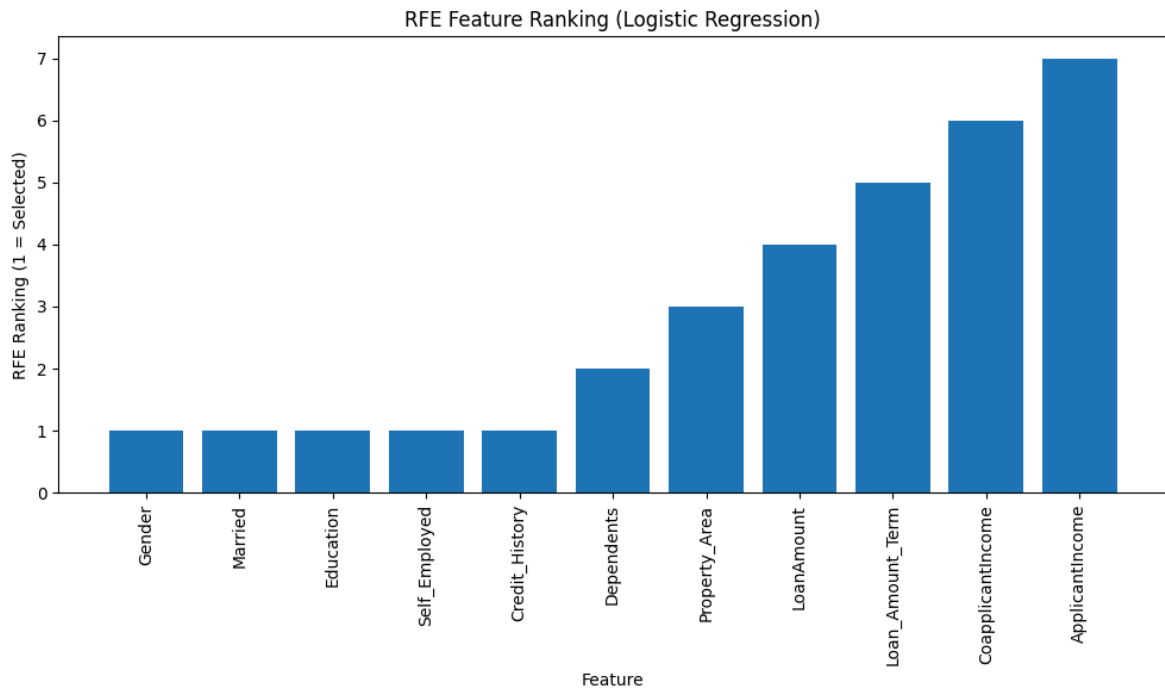
<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

n_iter_i = _check_optimize_result(

	Feature	Ranking
0	Gender	1
1	Married	1
3	Education	1
4	Self_Employed	1
9	Credit_History	1
2	Dependents	2
10	Property_Area	3
7	LoanAmount	4
8	Loan_Amount_Term	5
6	CoapplicantIncome	6
5	ApplicantIncome	7



```
In [83]: rf_for_rfe = RandomForestClassifier(n_estimators=200, random_state=42)
rfe_rf = RFE(estimator=rf_for_rfe, n_features_to_select=k)
rfe_rf.fit(X_train, y_train)

selected_features_rfe_rf = X_train.columns[rfe_rf.support_]
selected_features_rfe_rf
```

```
Out[83]: Index(['Dependents', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
               'Credit_History'],
              dtype='object')
```

```
In [84]: rf = RandomForestClassifier(random_state=42)
rf.fit(X_train, y_train)

importances = rf.feature_importances_

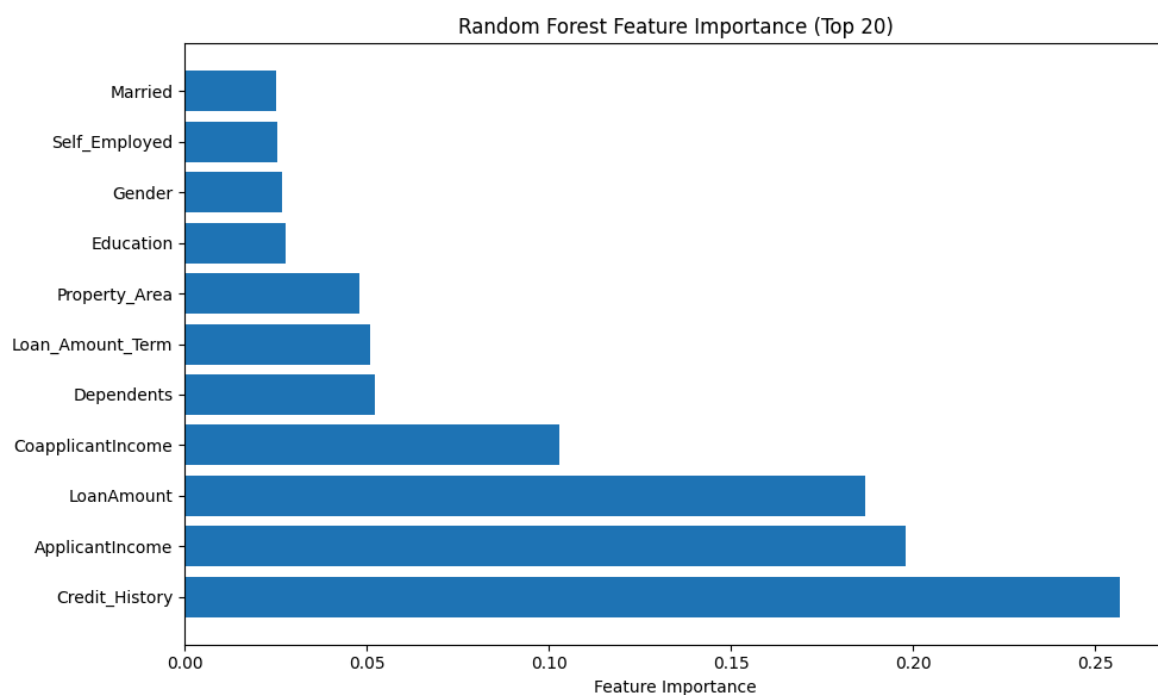
rf_imp = pd.DataFrame({
    "Feature": X_train.columns,
    "Importance": importances
}).sort_values("Importance", ascending=False)

display(rf_imp.head(15))

plt.figure(figsize=(10, 6))
plt.barh(rf_imp["Feature"][::-1].head(20)[::-1], rf_imp["Importance"][::-1].head(20)[::-1])
plt.xlabel("Feature Importance")
plt.title("Random Forest Feature Importance (Top 20)")
plt.tight_layout()
plt.show()

k = 10
selected_features_rf_imp = rf_imp["Feature"].head(k).values
selected_features_rf_imp
```

	Feature	Importance
9	Credit_History	0.256638
5	ApplicantIncome	0.197874
7	LoanAmount	0.186801
6	CoapplicantIncome	0.102891
2	Dependents	0.052350
8	Loan_Amount_Term	0.050767
10	Property_Area	0.047926
3	Education	0.027605
0	Gender	0.026765
4	Self_Employed	0.025417
1	Married	0.024968



```
Out[84]: array(['Credit_History', 'ApplicantIncome', 'LoanAmount',
                'CoapplicantIncome', 'Dependents', 'Loan_Amount_Term',
                'Property_Area', 'Education', 'Gender', 'Self_Employed'],
              dtype=object)
```

```
In [85]: def gridsearch_on_selected_features(selected_features, label):
        Xtr = X_train[selected_features]
        Xte = X_test[selected_features]

        results = []
        for name, (model, grid) in models.items():
            gs = GridSearchCV(model, grid, cv=5, scoring="f1_weighted", n_job
            gs.fit(Xtr, y_train)
            results.append(evaluate_model(f"{name} ({label})", gs.best_estima
        return pd.DataFrame(results).sort_values("F1", ascending=False)

        results_chi2 = gridsearch_on_selected_features(selected_features_chi2, "C
        results_rfe_lr = gridsearch_on_selected_features(selected_features_rfe_lr
```

```

results_rfe_rf = gridsearch_on_selected_features(selected_features_rfe_rf)
results_rfimp = gridsearch_on_selected_features(selected_features_rf_imp,

final_compare = pd.concat([baseline_df, results_chi2, results_rfe_lr, res
final_compare.sort_values("F1", ascending=False)

```

Out [85]:

	Model	Accuracy	Precision	Recall	F1
3	Random Forest (Chi2)	0.861789	0.875999	0.861789	0.850399
4	Decision Tree (Chi2)	0.853659	0.857488	0.853659	0.844763
12	SVM (RF-Imp)	0.853659	0.862569	0.853659	0.842703
0	Random Forest	0.853659	0.869641	0.853659	0.840441
6	Random Forest (RFE-LR)	0.853659	0.869641	0.853659	0.840441
7	SVM (RFE-LR)	0.853659	0.869641	0.853659	0.840441
13	Random Forest (RF-Imp)	0.853659	0.869641	0.853659	0.840441
8	Decision Tree (RFE-LR)	0.845528	0.855656	0.845528	0.832798
1	SVM	0.845528	0.863337	0.845528	0.830296
9	Random Forest (RFE-RF)	0.845528	0.863337	0.845528	0.830296
2	Decision Tree	0.829268	0.827612	0.829268	0.820010
14	Decision Tree (RF-Imp)	0.829268	0.827612	0.829268	0.820010
10	SVM (RFE-RF)	0.837398	0.857076	0.837398	0.819951
11	Decision Tree (RFE-RF)	0.813008	0.807801	0.813008	0.805150
5	SVM (Chi2)	0.813008	0.852830	0.813008	0.783576

In []: