```
# Lab: Random Forest Algorithm
# 0. Installation and Imports

# Uncomment and run if not installed:
!pip install ucimlrepo

from ucimlrepo import fetch_ucirepo
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusi
import matplotlib.pyplot as plt
```

```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.7-py3-none-any.whl.metadata (5.5 kB)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.12/d
Requirement already satisfied: certifi>=2020.12.5 in /usr/local/lib/python3
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/d
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pyt
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/di
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-p
Downloading ucimlrepo-0.0.7-py3-none-any.whl (8.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.7
```

```
# Lab: Random Forest Algorithm
# 1. Load Dataset and Inspect

dermatology = fetch_ucirepo(id=33)

X = dermatology.data.features
y = dermatology.data.targets

print("Metadata:\n", dermatology.metadata)
print("\nVariables:\n", dermatology.variables)
print("Sample features:\n", X.head())
print("Sample targets:\n", y.head())
```

```
   disappearance of the granular layer  \
0                                    0
1                                    0
2                                    0
3                                    3
4                                    2

   vacuolisation and damage of the basal layer  spongiosis  \
0                                            0           3
1                                            0           0
2                                            2           3
3                                            0           0
4                                            3           2

   saw-tooth appearance of retes  follicular horn plug  \
0                              0                     0
1                              0                     0
2                              2                     0
3                              0                     0
4                              3                     0

   perifollicular parakeratosis  inflammatory monoluclear infiltrate  \
0                             0                                     1
1                             0                                     1
2                             0                                     2
3                             0                                     3
4                             0                                     2

   band-like infiltrate   age
0                     0  55.0
1                     0   8.0
2                     3  26.0
3                     0  40.0
4                     3  45.0

[5 rows x 34 columns]
Sample targets:
    class
0      2
1      1
2      3
3      1
4      3
```

```
# Lab: Random Forest Algorithm
# 2. Train-Test Split

X_train, X_test, y_train, y_test = train_test_split(
    X, y.values.ravel(), test_size=0.2, random_state=42
)

print("Training set size:", X_train.shape)
print("Testing set size:", X_test.shape)
```

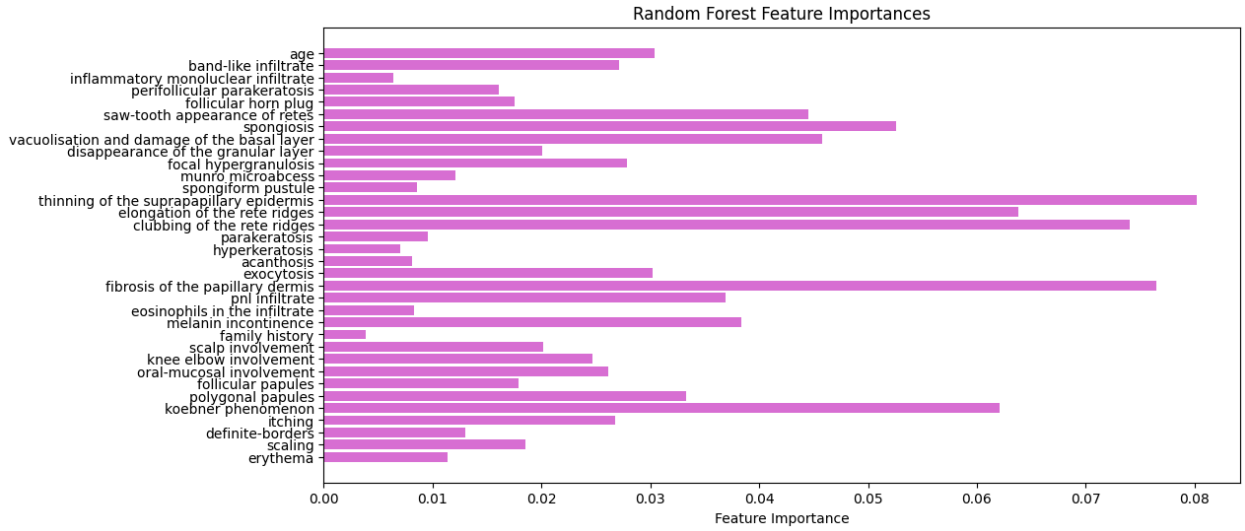
```
Training set size: (292, 34)
Testing set size: (74, 34)
```

```
# Lab: Random Forest Algorithm
# 3. Random Forest Training

rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
```

```
# Lab: Random Forest Algorithm
# 4. Evaluation and Output

print("Accuracy Score:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
Accuracy Score: 0.9864864864864865
Classification Report:
               precision    recall  f1-score   support

           1       1.00      1.00      1.00        31
           2       0.90      1.00      0.95         9
           3       1.00      1.00      1.00        13
           4       1.00      0.88      0.93         8
           5       1.00      1.00      1.00        10
           6       1.00      1.00      1.00         3

    accuracy                           0.99        74
   macro avg       0.98      0.98      0.98        74
weighted avg       0.99      0.99      0.99        74

Confusion Matrix:
 [[31  0  0  0  0  0]
 [ 0  9  0  0  0  0]
 [ 0  0 13  0  0  0]
 [ 0  1  0  7  0  0]
 [ 0  0  0  0 10  0]
 [ 0  0  0  0  0  3]]
```

```
# Lab: Random Forest Algorithm
# 5. Feature Importance Visualization

importances = rf.feature_importances_
features = X.columns

plt.figure(figsize=(12, 6))
plt.barh(features, importances, color='orchid')
plt.xlabel('Feature Importance')
plt.title('Random Forest Feature Importances')
plt.show()
```



Start coding or generate with AI.