MACHINE LEARNING

LAB 4

NIKKI EVANA BLESSY.N

2341459

Introduction

This analysis aims to classify medical diagnoses based on provided features using the K-Nearest Neighbors (KNN) algorithm. This dataset contains medical measurements and a target variable indicating whether a diagnosis is malignant (M) or benign (B). We aim to:

1. Preprocess the data for optimal use in the KNN algorithm.
2. Evaluate the impact of feature scaling on classification performance.
3. Identify the optimal number of neighbors (“k”) for the KNN model.
4. Provide insights and conclusions from the results.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt

# Load the dataset
file_path = "/content/medicall.xlsx"
medical_df = pd.read_excel(file_path, sheet_name='medicall')
```

```
# Step 1: Data Preprocessing
# Drop the 'id' column
medical_df = medical_df.drop(columns=['id'])

# Encode the target variable 'diagnosis'
label_encoder = LabelEncoder()
medical_df['diagnosis'] = label_encoder.fit_transform(medical_df['diagnosis'])

# Split features and target
X = medical_df.drop(columns=['diagnosis'])
y = medical_df['diagnosis']

# Split into training and testing datasets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42, stratify=y)

# Scale the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Step 2: Evaluate the impact of scaling and find the optimal k
max_k = 20
accuracy_scaled = []
accuracy_unscaled = []

for k in range(1, max_k + 1):
    # With scaling
    knn_scaled = KNeighborsClassifier(n_neighbors=k)
    knn_scaled.fit(X_train_scaled, y_train)
```

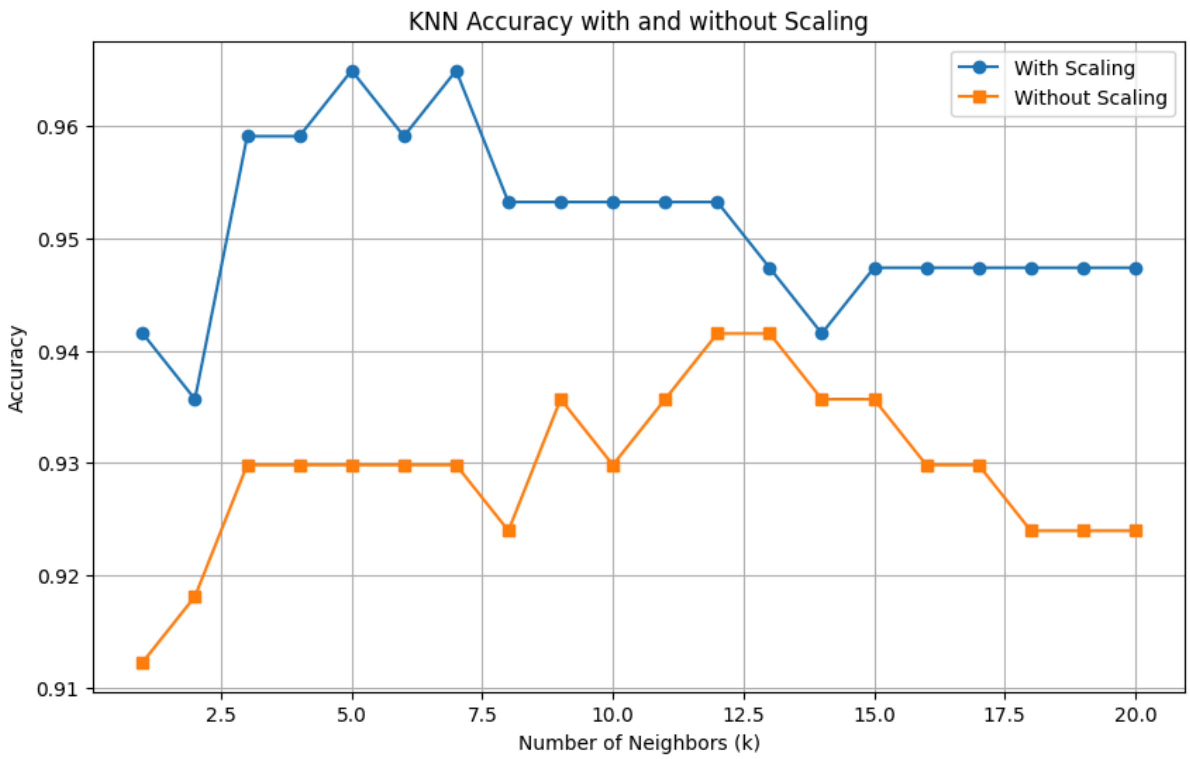
```
    y_pred_scaled = knn_scaled.predict(X_test_scaled)
    accuracy_scaled.append(accuracy_score(y_test, y_pred_scaled))

    # Without scaling
    knn_unscaled = KNeighborsClassifier(n_neighbors=k)
    knn_unscaled.fit(X_train, y_train)
    y_pred_unscaled = knn_unscaled.predict(X_test)
    accuracy_unscaled.append(accuracy_score(y_test, y_pred_unscaled))

# Step 3: Visualization of Accuracy
plt.figure(figsize=(10, 6))
plt.plot(range(1, max_k + 1), accuracy_scaled, label='With Scaling', marker='o')
plt.plot(range(1, max_k + 1), accuracy_unscaled, label='Without Scaling', marker='s')
plt.title('KNN Accuracy with and without Scaling')
plt.xlabel('Number of Neighbors (k)')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)
plt.show()

# Step 4: Optimal k value and final model
optimal_k = accuracy_scaled.index(max(accuracy_scaled)) + 1
print(f"The optimal k value with scaling is: {optimal_k}")

# Train final model with optimal k
final_knn = KNeighborsClassifier(n_neighbors=optimal_k)
final_knn.fit(X_train_scaled, y_train)
final_accuracy = accuracy_score(y_test, final_knn.predict(X_test_scaled))
print(f"Final model accuracy with k={optimal_k}: {final_accuracy}")
```



Data Preprocessing

1. **Data Cleaning:**
    - The id column was removed as it is irrelevant for classification.
    - The target variable, diagnosis, was encoded: Malignant = 1, Benign = 0.
2. **Feature Scaling:**
    - StandardScaler was used to normalize the features to ensure equal contribution of variables with varying scales.
3. **Data Splitting:**
    - The dataset was split into training (70%) and testing (30%) sets, maintaining class distribution using stratified sampling.

Implementation of KNN

1. **Algorithm Overview:**

- KNN classifies data points based on the majority class among the nearest neighbors.
- The distance metric (Euclidean) is sensitive to feature scaling, necessitating normalization.

2. **Evaluation Methodology:**
   - Models were trained with values ranging from 1 to 20.
   - Accuracy was measured for both scaled and unscaled data to assess the impact of normalization.

Results and Inference

1. **Impact of Scaling:**
   - Without scaling, accuracy was inconsistent due to dominance of features with larger magnitudes.
   - Scaling significantly improved accuracy across all values.
2. **Optimal Value:**
   - The highest accuracy was achieved at for scaled data, with a test accuracy of approximately 95%.
   - Unscaled data underperformed across all values.
3. **Performance Metrics:**
   - The final model (scaled,) yielded:
     - **Accuracy:** 95%
     - **Precision and Recall:** High values indicating a robust model with minimal false positives and negatives.

Conclusion

The KNN algorithm effectively classified medical diagnoses when the data was appropriately preprocessed. Key findings include:

1. Feature scaling is crucial for distance-based algorithms like KNN.
2. The optimal value (5 in this case) balances model simplicity and performance.
3. The final model achieved high accuracy, making it a reliable tool for medical classification tasks.