BSc(Hons) Data Science & Business Analytics

Introduction to Machine Learning (CS3273)

Lecturer: Ms. WMKS Ilmini

Semester 05

Assignment 01

GKP Sewmini

D/DBA/22/0025

Introduction

The Wheat Seeds Dataset contains measurements of seven different features of wheat kernels belonging to three different varieties. The dataset consists of 210 observations, with 70 observations for each variety. The goal of this analysis is to explore the dataset and develop an artificial neural network (ANN) model to predict the variety of a wheat seed based on its features.

Seven geometric parameters of wheat kernels :

1. area A,
2. perimeter P,
3. compactness C = 4*pi*A/P^2,
4. length of kernel,
5. width of kernel,
6. asymmetry coefficient
7. length of kernel groove.

All of these parameters were real-valued continuous.

Exploratory Data Analysis (EDA)

Loading the dataset and performing some basic exploratory data analysis. The dataset contains eight columns: 'Area', 'Perimeter', 'Compactness', 'length_k', 'width_k', 'as_coe', 'len_k_grov', and 'variety'. The 'variety' column is the target variable, which will use to train our ANN model.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings as wr
wr.filterwarnings('ignore')
import os
```

```
df = pd.read_csv("seeds_dataset.csv")
print(df.head())
```

```
    Area  Perimeter  Compactness  length_k  width_k  as_coe  len_k_grov  \
0  15.26      14.84       0.8710     5.763    3.312   2.221       5.220
1  14.88      14.57       0.8811     5.554    3.333   1.018       4.956
2  14.29      14.09       0.9050     5.291    3.337   2.699       4.825
3  13.84      13.94       0.8955     5.324    3.379   2.259       4.805
4  16.14      14.99       0.9034     5.658    3.562   1.355       5.175

   variety
0        1
```

```
1         1
2         1
3         1
4         1
```

The dataset contains no missing values and all features have numerical data types. Then calculated the number of unique values for each feature and found that 'variety' has three unique values, corresponding to the three different wheat varieties.

```
df.shape
```

```
(210, 8)
```

```
df.columns.tolist()
```

```
['Area',
 'Perimeter',
 'Compactness',
 'length_k',
 'width_k',
 'as_coe',
 'len_k_grov',
 'variety']
```

```
df.isnull().sum()
```

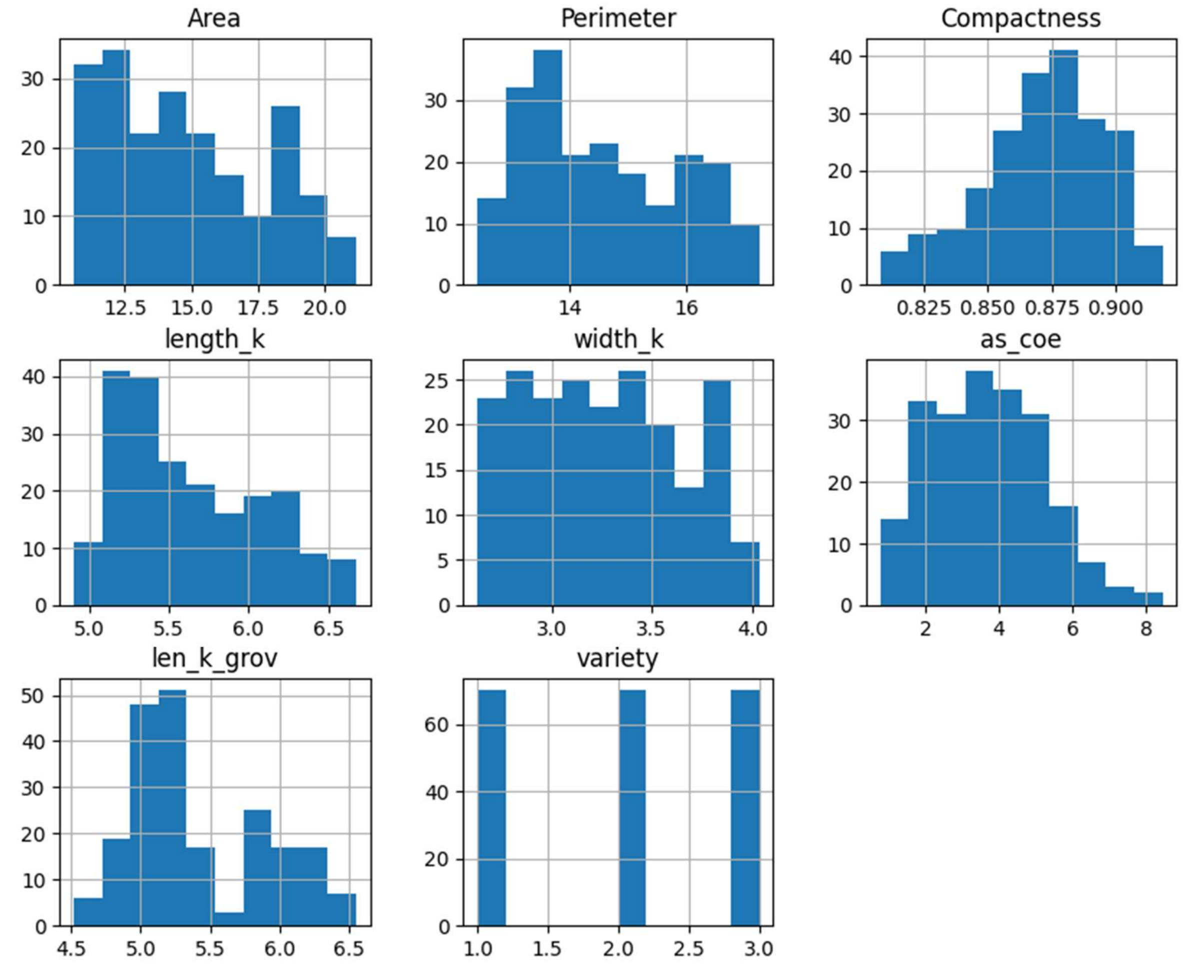
```
Area         0
Perimeter     0
Compactness    0
length_k      0
width_k       0
as_coe        0
len_k_grov    0
variety       0
dtype: int64
```

```
df.nunique()
```

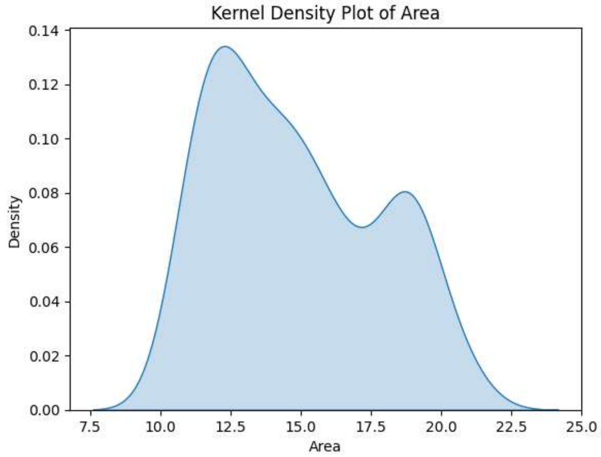
```
Area         193
Perimeter     170
Compactness    186
length_k      188
width_k       184
as_coe        207
len_k_grov    148
variety        3
dtype: int64
```

Distributions of each feature using histograms. Most features have a roughly normal distribution, with some features having a slight positive skew. Then a kernel density plot for the 'Area' feature, which showed a similar distribution.

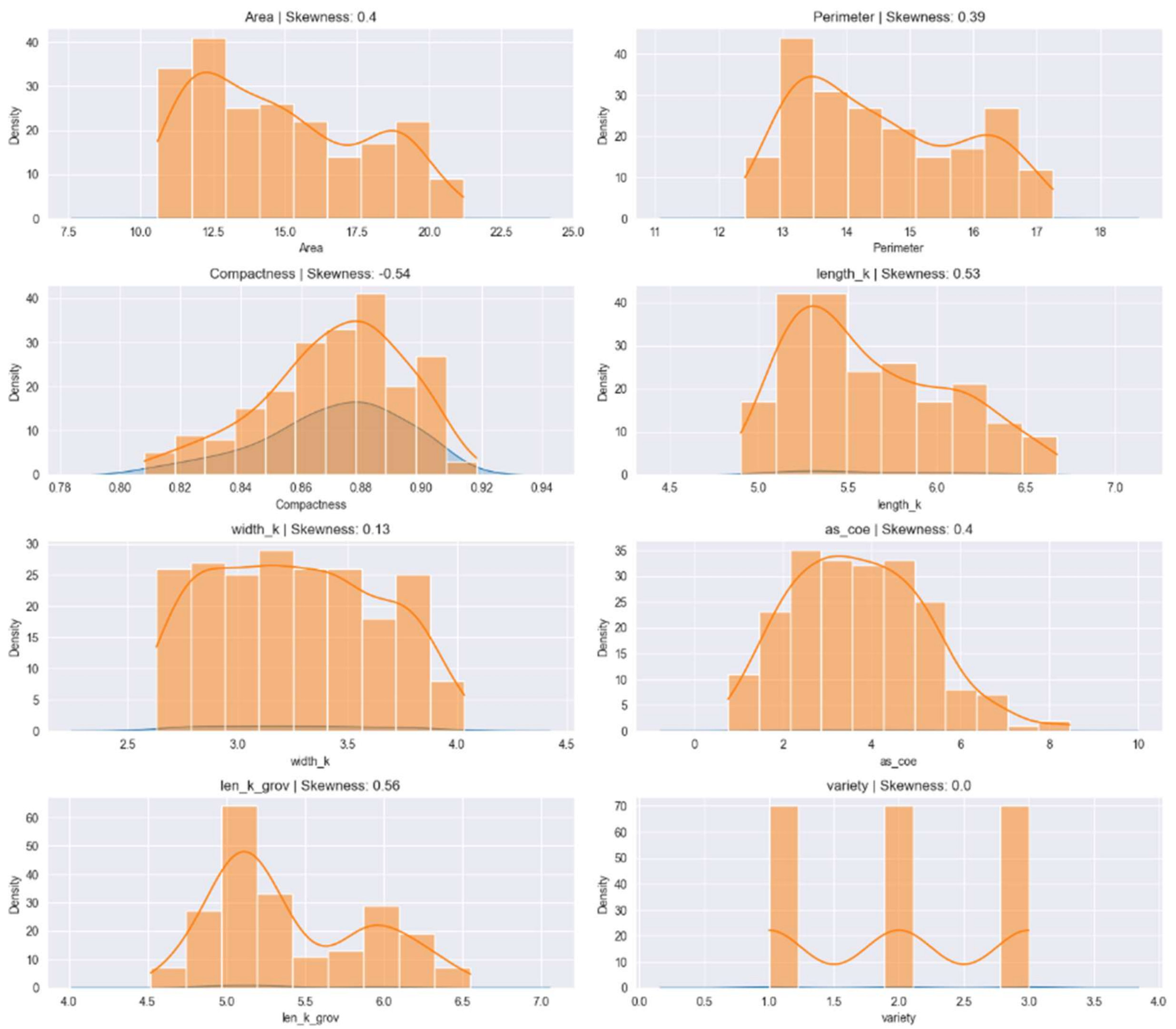
```
df.hist(bins=10, figsize=(10, 8))
plt.show()
```



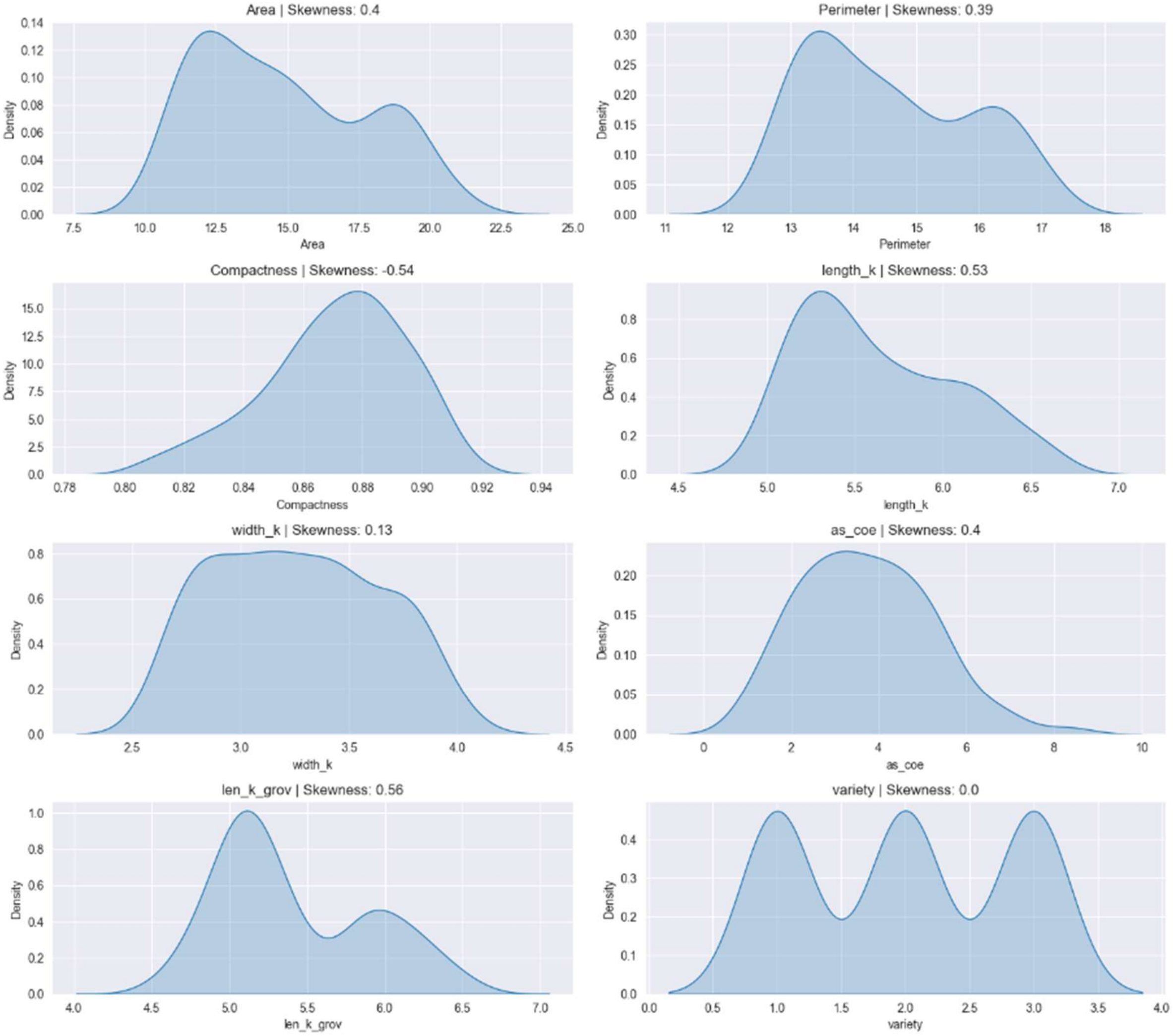
```
sns.kdeplot(data=df, x='Area', fill=True)

plt.xlabel('Area')
plt.ylabel('Density')
plt.title('Kernel Density Plot of Area')

plt.show()
```



```
sns.set_style("darkgrid")

numerical_columns = df.select_dtypes(include=["int64", "float64"]).columns

plt.figure(figsize=(14, len(numerical_columns) * 3))
for idx, feature in enumerate(numerical_columns, 1):
    plt.subplot(len(numerical_columns), 2, idx)
    sns.kdeplot(df[feature], shade=True)
    sns.histplot(df[feature], kde=True)
    plt.title(f"{feature} | Skewness: {round(df[feature].skew(), 2)}")

plt.tight_layout()
plt.show()
```

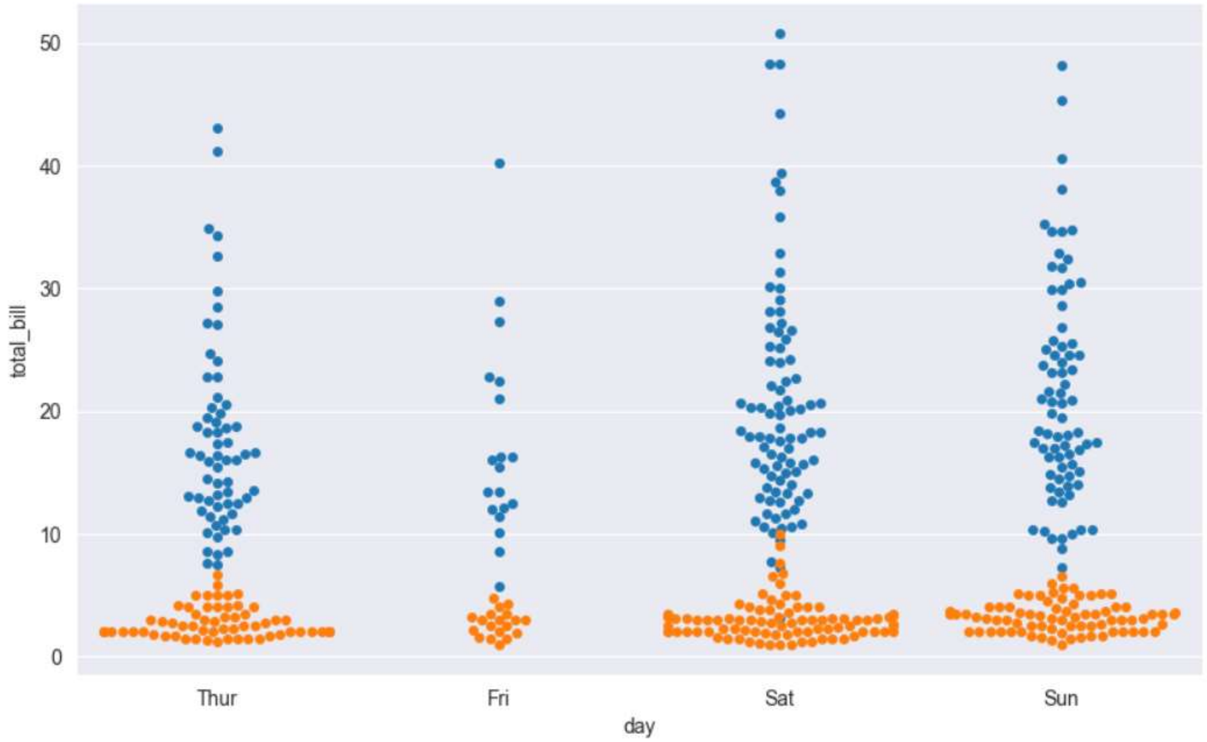


```
sns.set_style("darkgrid")

numerical_columns = df.select_dtypes(include=["int64", "float64"]).columns

plt.figure(figsize=(14, len(numerical_columns) * 3))
for idx, feature in enumerate(numerical_columns, 1):
    plt.subplot(len(numerical_columns), 2, idx)
    sns.kdeplot(df[feature], shade=True)
    plt.title(f"{feature} | Skewness: {round(df[feature].skew(), 2)}")

plt.tight_layout()
plt.show()
plt.show()
```

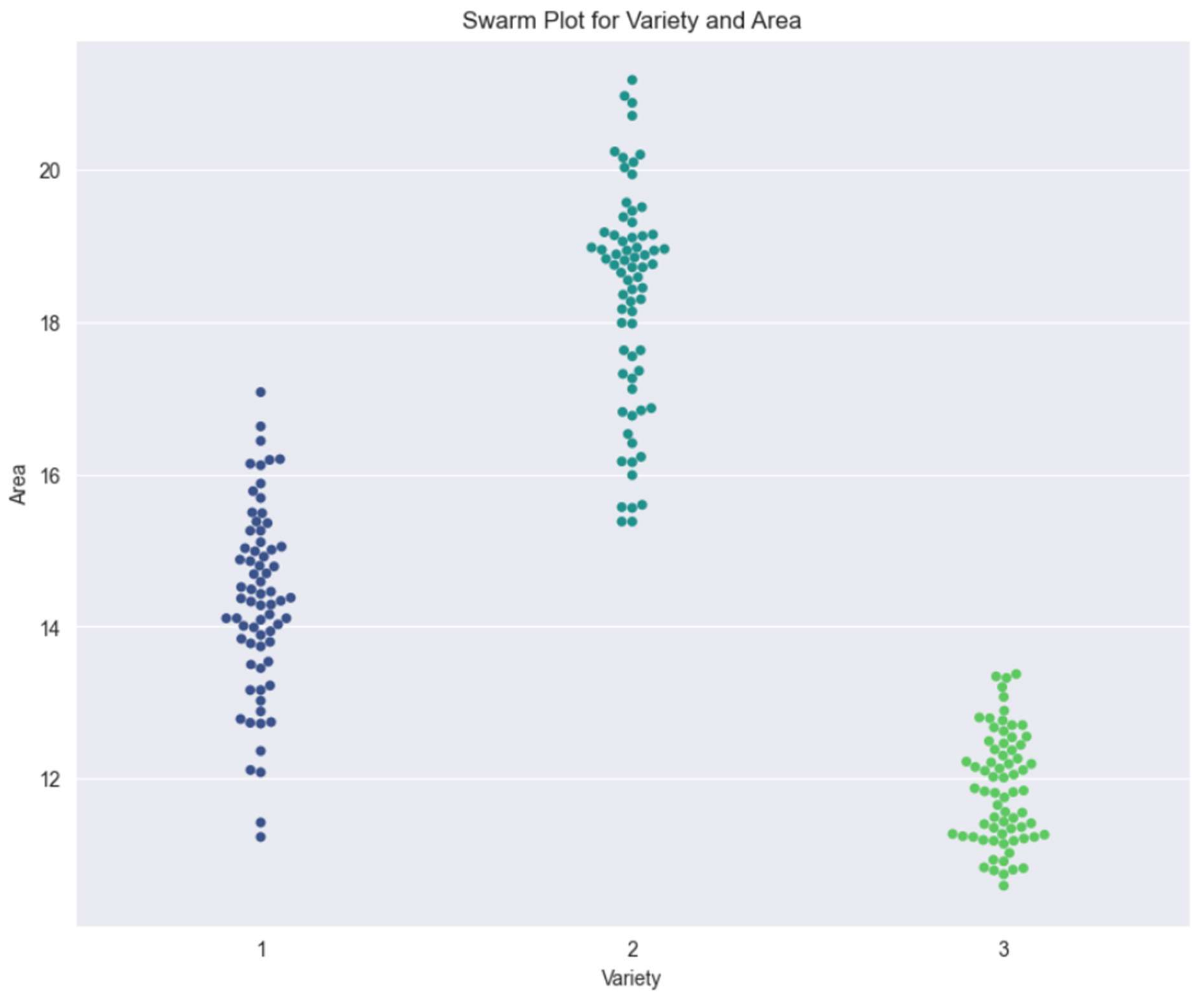


Swarm plot to visualize the relationship between 'variety' and 'Area'. There is some overlap in the 'Area' values for the different varieties, but variety 3 tends to have smaller 'Area' values than varieties 1 and 2.

```
#swarm plot
tips = sns.load_dataset("tips")

plt.figure(figsize=(10, 6))
sns.swarmplot(x="day", y="total_bill", data=tips)
sns.swarmplot(x="day", y="tip", data=tips)

plt.show()
```

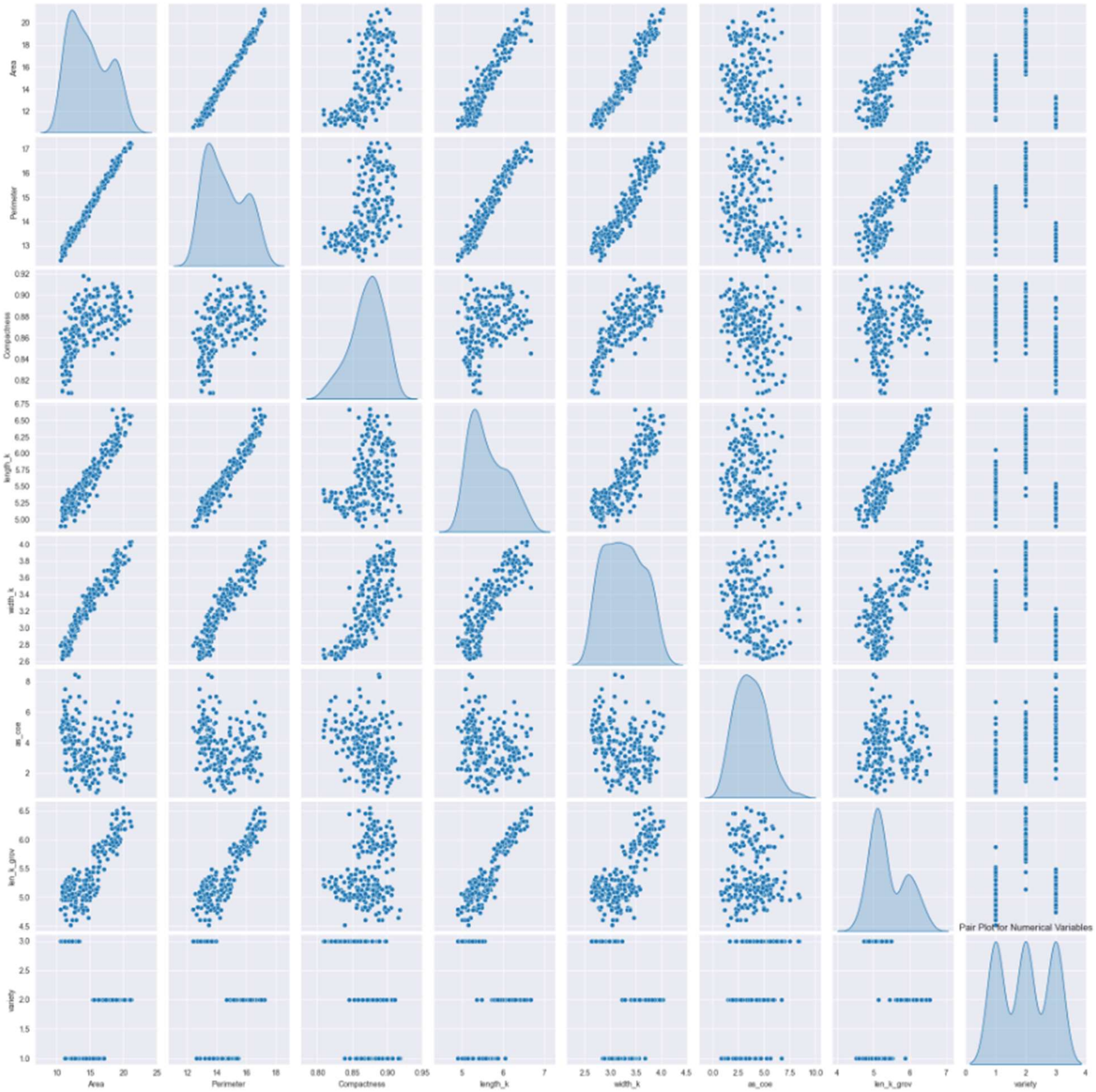


```
#swarm plot for variety and area
plt.figure(figsize=(10, 8))

sns.swarmplot(x="variety", y="Area", data=df, palette='viridis')

plt.title('Swarm Plot for Variety and Area')
plt.xlabel('Variety')
plt.ylabel('Area')
plt.show()
```



To further explore the relationships between features, there's pair plot. The pair plot showed that there is a positive correlation between 'Area' and 'Perimeter', as well as between 'length_k' and 'width_k'. There is also a negative correlation between 'Compactness' and 'length_k' and 'width_k'.

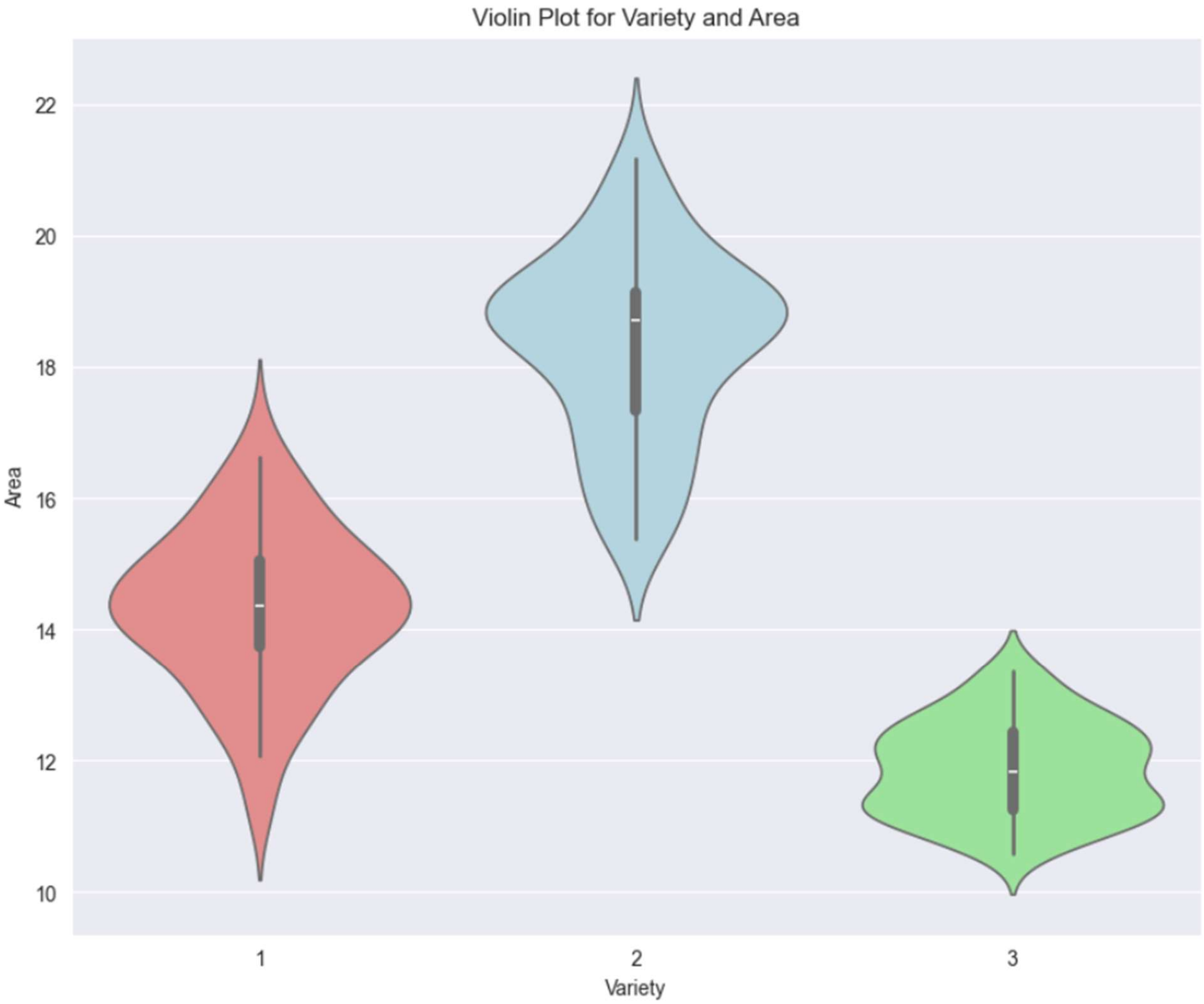
```
#pair plot
plt.figure(figsize=(10, 8))
sns.pairplot(df, vars=numerical_columns, kind='scatter', diag_kind='kde',
palette='viridis')
plt.title('Pair Plot for Numerical Variables')
plt.show()
```



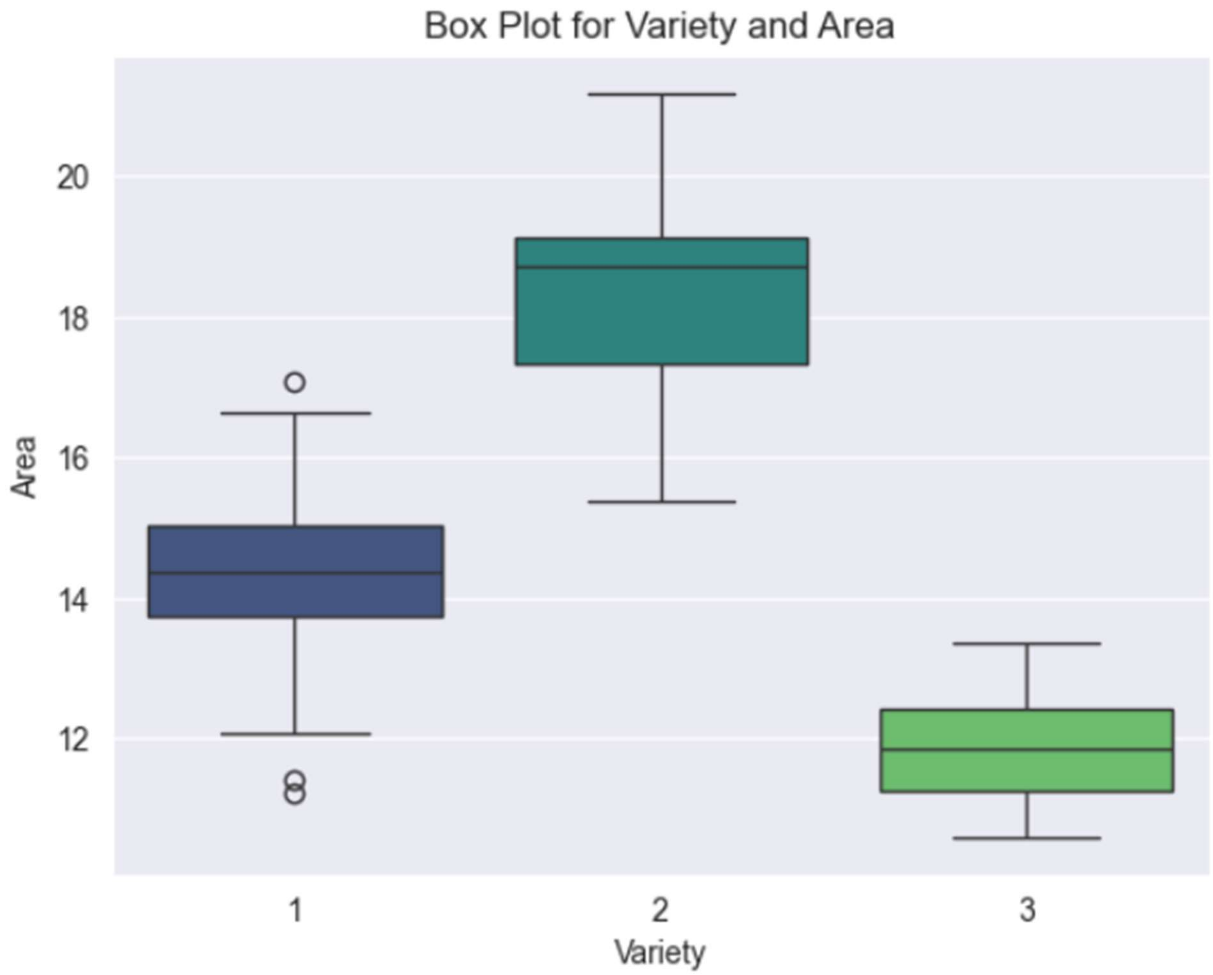
Finally, the violin plot and box plot to visualize the distribution of 'Area' values for each variety. The violin plot showed that the distribution of 'Area' values varies by variety, with variety 3 having a smaller range of values than varieties 1 and 2. The box plot showed that there are some outliers in the 'Area' values for each variety.

```
#violin plot
df['variety'] = df['variety'].astype(str)  # Convert 'variety' to 
categorical
plt.figure(figsize=(10, 8))
```

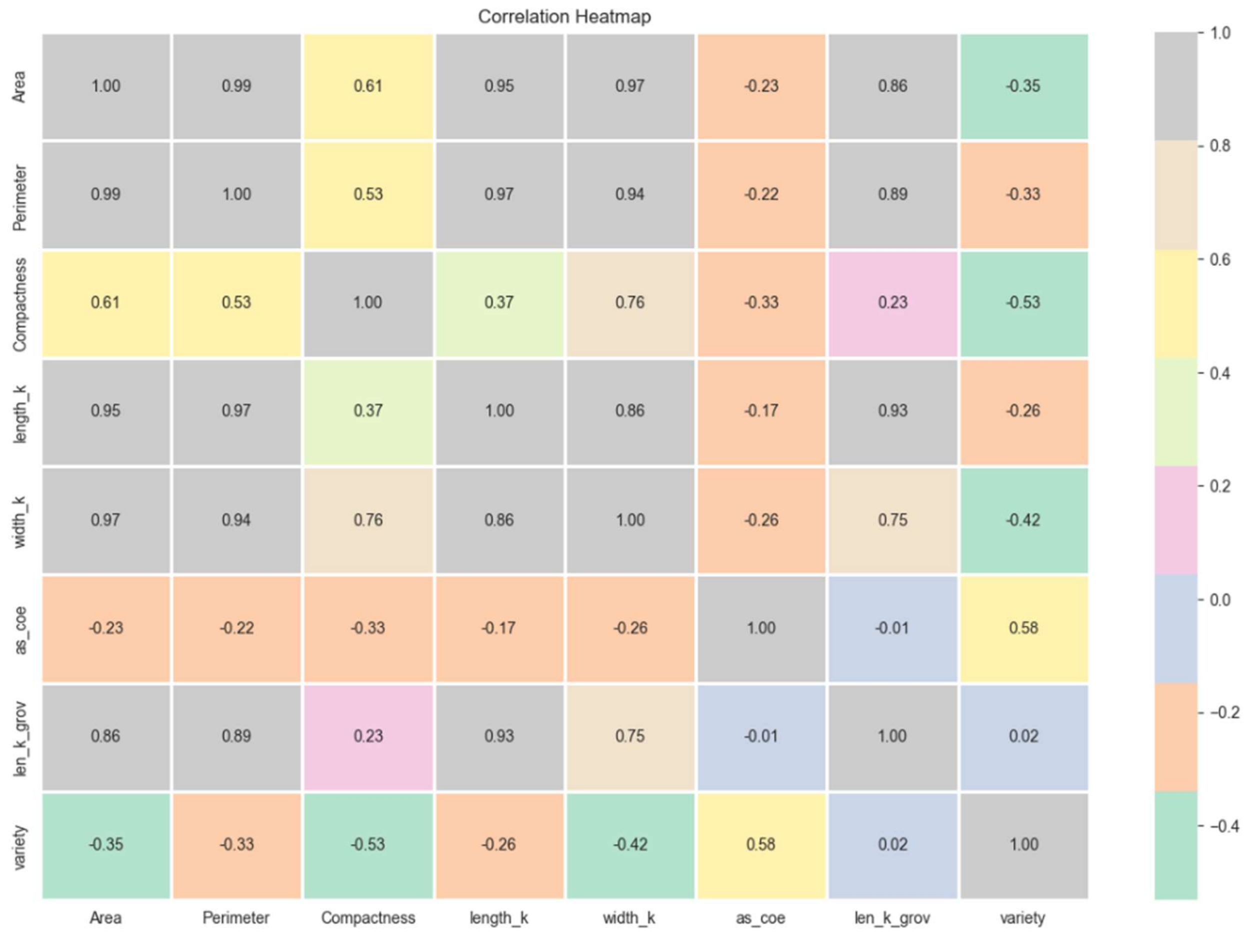
```
sns.violinplot(x="variety", y="Area", data=df, palette={
                '1': 'lightcoral', '2': 'lightblue', '3': 'lightgreen'})
plt.title('Violin Plot for Variety and Area')
plt.xlabel('Variety')
plt.ylabel('Area')
plt.show()
```



```
#boxplot
sns.boxplot(x="variety", y="Area", data=df, palette='viridis')
plt.title('Box Plot for Variety and Area')
plt.xlabel('Variety')
plt.ylabel('Area')
plt.show()
```



```
#correlation matrix
plt.figure(figsize=(15, 10))
sns.heatmap(df.corr(), annot=True, fmt='.2f', cmap='Pastel2',
linewidths=2)
plt.title('Correlation Heatmap')
plt.show()
```



Artificial Neural Network (ANN) Model

The dataset was loaded into a pandas DataFrame and split into features (X) and labels (y). The 'variety' column was dropped from the X DataFrame and converted to one-hot encoding using the to_categorical function from Keras. The X DataFrame was then scaled using the MinMaxScaler function from Scikit-Learn. Finally, the dataset was split into training and testing sets using the train_test_split function from Scikit-Learn.

```python
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.utils import to_categorical
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
```

```
X = df.drop('variety', axis=1)
y = to_categorical(df['variety'])
scaler = MinMaxScaler()
X = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

The ANN model was built using the Sequential model from Keras. The model consisted of three layers:

- Input layer: A dense layer with 32 neurons and ReLU activation function.
- Hidden layer: A dense layer with 16 neurons and ReLU activation function.
- Output layer: A dense layer with 3 neurons and softmax activation function.

```
model = Sequential()
model.add(Dense(32, activation='relu', input_shape=(7,)))
model.add(Dense(16, activation='relu'))
model.add(Dense(3, activation='softmax'))
```

The model was compiled using the Adam optimizer and categorical cross-entropy loss function. Accuracy was used as the evaluation metric.

```
model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
model = Sequential()
model.add(Dense(16, activation='relu', input_shape=(7,)))
model.add(Dense(8, activation='relu'))
model.add(Dense(4, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
model.fit(X_train, y_train, epochs=50, batch_size=32,
validation_split=0.2)
loss, accuracy = model.evaluate(X_test, y_test)
print('Test accuracy:', accuracy)
Test accuracy: 0.6904761791229248
```

The model was trained using the fit function from Keras. The training set was used to train the model for 50 epochs with a batch size of 32. A validation split of 20% was used to evaluate the model's performance during training.

The model's performance was evaluated using the evaluate function from Keras on the testing set. The model achieved a test accuracy of 69.05%.

Conclusion

The wheat seeds dataset was analyzed and an Artificial Neural Network (ANN) model was developed to classify the seeds into three different varieties based on seven features. The dataset consisted of 210 samples, with each sample having seven features and a corresponding label indicating the variety of wheat seed.

The data was first preprocessed by splitting it into training and testing sets, and then scaling the features using MinMaxScaler. The target variable was converted to one-hot encoding using to_categorical function. An ANN model was then developed using the Sequential model from Keras. The model consisted of three layers, with the input layer having seven neurons, the hidden layer having 16 neurons, and the output layer having three neurons corresponding to the three varieties of wheat seeds. The ReLU activation function was used for the input and hidden layers, while the softmax activation function was used for the output layer.

The model was compiled using the categorical cross-entropy loss function and the Adam optimizer. The model was then trained using the fit function, with a batch size of 32 and a total of 50 epochs. The validation split was set to 0.2 to evaluate the model's performance on unseen data.

The trained model achieved a test accuracy of 69.05%, which indicates that the model can accurately classify wheat seeds into their respective varieties with a reasonable degree of accuracy. However, there is still room for improvement, and future work could focus on optimizing the model's hyperparameters and architecture.

In conclusion, this project demonstrated the application of ANN models in classifying wheat seeds based on their features. The developed model can potentially be used in agriculture to assist in seed classification and improve crop yield. However, further research is required to improve the model's accuracy and generalization performance.