21162101012_CBA_Yash_Lakhtariya

DMW Practical 8

The Iris flower data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems. It is sometimes called Anderson's Iris data set because Edgar Anderson collected the data to quantify the morphologic variation of Iris flowers of three related species. The data set consists of 50 samples from each of three species of Iris (Iris Setosa, Iris virginica, and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

This dataset became a typical test case for many statistical classification techniques. Kindly implement the Naive Bayes Classification for famous Iris Flower. Dataset that consists of 3 classes of flowers. In this, there are 4 independent variables namely the, sepal_length, sepal_width, petal_length and petal_width. The dependent variable is the species which we will predict using the four independent features of the flowers.

```
[ ] from google.colab import files
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    import io
```

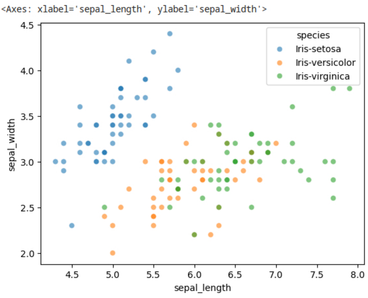
```
[ ] uploaded = files.upload()
```

```
Choose Files  No file chosen    Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving iris.csv to iris.csv
```

```
[ ] df = pd.read_csv(io.BytesIO(uploaded['iris.csv']))
```

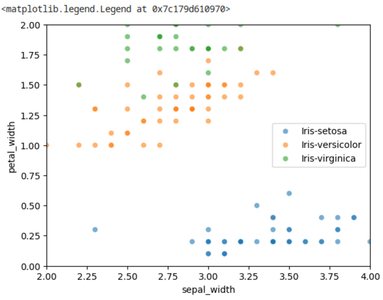
```
[ ] sns.scatterplot(data=df, x='sepal_length', y='sepal_width', hue='species', alpha=0.6)
```

```
<Axes: xlabel='sepal_length', ylabel='sepal_width'>
```



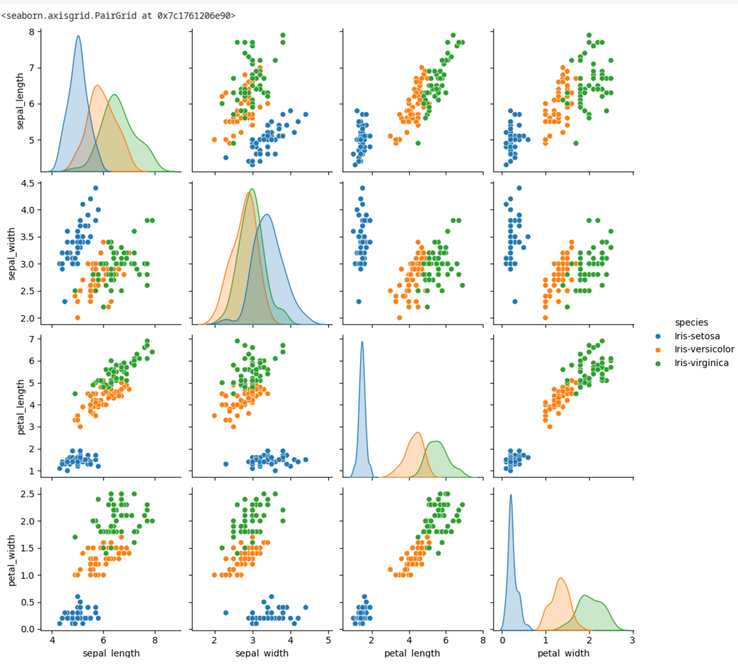

```
[ ] sns.scatterplot(data=df, x='sepal_width', y='petal_width', hue='species', alpha=0.6)
    plt.xlim(2, 4)
    plt.ylim(0, 2)
    plt.legend()
```

```
<matplotlib.legend.Legend at 0x7c179d610970>
```




```
[ ] sns.pairplot(data=df, hue='species')
```

```
<seaborn.axisgrid.PairGrid at 0x7c1761206e90>
```




```
[ ] from sklearn.model_selection import train_test_split
    from sklearn.naive_bayes import GaussianNB
    from sklearn.metrics import accuracy_score
```

```
[ ] x = df[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']]
    y = df['species']
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.4, random_state=42)
```

```
[ ] clf = GaussianNB()
    clf.fit(x_train, y_train)
```

```
▾ GaussianNB
GaussianNB()
```

```
[ ] y_pred = clf.predict(x_test)
    accuracy = accuracy_score(y_test, y_pred)
    print('Accuracy of this classifier:', accuracy)
```

```
Accuracy of this classifier: 0.9666666666666667
```