Code Written By: Ankit Kotnala

Importing Essential Libraries & Loading Dataset

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
df = pd.read_csv("Iris.csv")
df
```

Out[2]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

Exploring the dataset

In [3]:

```
df.head()
```

Out[3]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [4]:

```
df.tail()
```

Out[4]:

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica

146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

```
In [5]: df.shape
```

```
Out[5]: (150, 5)
```

```
In [6]: df.size
```

```
Out[6]: 750
```

```
In [7]: df.columns
```

```
Out[7]: Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
               'species'],
              dtype='object')
```

```
In [8]: df.values
```

```
Out[8]: array([[5.1, 3.5, 1.4, 0.2, 'setosa'],
               [4.9, 3.0, 1.4, 0.2, 'setosa'],
               [4.7, 3.2, 1.3, 0.2, 'setosa'],
               [4.6, 3.1, 1.5, 0.2, 'setosa'],
               [5.0, 3.6, 1.4, 0.2, 'setosa'],
               [5.4, 3.9, 1.7, 0.4, 'setosa'],
               [4.6, 3.4, 1.4, 0.3, 'setosa'],
               [5.0, 3.4, 1.5, 0.2, 'setosa'],
               [4.4, 2.9, 1.4, 0.2, 'setosa'],
               [4.9, 3.1, 1.5, 0.1, 'setosa'],
               [5.4, 3.7, 1.5, 0.2, 'setosa'],
               [4.8, 3.4, 1.6, 0.2, 'setosa'],
               [4.8, 3.0, 1.4, 0.1, 'setosa'],
               [4.3, 3.0, 1.1, 0.1, 'setosa'],
               [5.8, 4.0, 1.2, 0.2, 'setosa'],
               [5.7, 4.4, 1.5, 0.4, 'setosa'],
               [5.4, 3.9, 1.3, 0.4, 'setosa'],
               [5.1, 3.5, 1.4, 0.3, 'setosa'],
               [5.7, 3.8, 1.7, 0.3, 'setosa'],
               [5.1, 3.8, 1.5, 0.3, 'setosa'],
               [5.4, 3.4, 1.7, 0.2, 'setosa'],
               [5.1, 3.7, 1.5, 0.4, 'setosa'],
               [4.6, 3.6, 1.0, 0.2, 'setosa'],
               [5.1, 3.3, 1.7, 0.5, 'setosa'],
               [4.8, 3.4, 1.9, 0.2, 'setosa'],
               [5.0, 3.0, 1.6, 0.2, 'setosa'],
               [5.0, 3.4, 1.6, 0.4, 'setosa'],
               [5.2, 3.5, 1.5, 0.2, 'setosa'],
               [5.2, 3.4, 1.4, 0.2, 'setosa'],
               [4.7, 3.2, 1.6, 0.2, 'setosa'],
               [4.8, 3.1, 1.6, 0.2, 'setosa'],
               [5.4, 3.4, 1.5, 0.4, 'setosa'],
               [5.2, 4.1, 1.5, 0.1, 'setosa'],
               [5.5, 4.2, 1.4, 0.2, 'setosa'],
               [4.9, 3.1, 1.5, 0.2, 'setosa'],
               [5.0, 3.2, 1.2, 0.2, 'setosa'],
               [5.5, 3.5, 1.3, 0.2, 'setosa'],
               [4.9, 3.6, 1.4, 0.1, 'setosa'],
               [4.4, 3.0, 1.3, 0.2, 'setosa'],
               [5.1, 3.4, 1.5, 0.2, 'setosa'],
               [5.0, 3.5, 1.3, 0.3, 'setosa'],
[4.5, 2.3, 1.3, 0.3, 'setosa'],
[4.4, 3.2, 1.3, 0.2, 'setosa'],
[5.0, 3.5, 1.6, 0.6, 'setosa'],
[5.1, 3.8, 1.9, 0.4, 'setosa'],
[4.8, 3.0, 1.4, 0.3, 'setosa'],
[5.1, 3.8, 1.6, 0.2, 'setosa'],
[4.6, 3.2, 1.4, 0.2, 'setosa'],
[5.3, 3.7, 1.5, 0.2, 'setosa'],
[5.0, 3.3, 1.4, 0.2, 'setosa'],
[7.0, 3.2, 4.7, 1.4, 'versicolor'],
[6.4, 3.2, 4.5, 1.5, 'versicolor'],
[6.9, 3.1, 4.9, 1.5, 'versicolor'],
[5.5, 2.3, 4.0, 1.3, 'versicolor'],
[6.5, 2.8, 4.6, 1.5, 'versicolor'],
[5.7, 2.8, 4.5, 1.3, 'versicolor'],
[6.3, 3.3, 4.7, 1.6, 'versicolor'],
[4.9, 2.4, 3.3, 1.0, 'versicolor'],
[6.6, 2.9, 4.6, 1.3, 'versicolor'],
[5.2, 2.7, 3.9, 1.4, 'versicolor'],
[5.0, 2.0, 3.5, 1.0, 'versicolor'],
[5.9, 3.0, 4.2, 1.5, 'versicolor'],
[6.0, 2.2, 4.0, 1.0, 'versicolor'],
[6.1, 2.9, 4.7, 1.4, 'versicolor'],
[5.6, 2.9, 3.6, 1.3, 'versicolor'],
[6.7, 3.1, 4.4, 1.4, 'versicolor'],
[5.6, 3.0, 4.5, 1.5, 'versicolor'],
[5.8, 2.7, 4.1, 1.0, 'versicolor'],
[6.2, 2.2, 4.5, 1.5, 'versicolor'],
[5.6, 2.5, 3.9, 1.1, 'versicolor'],
[5.9, 3.2, 4.8, 1.8, 'versicolor'],
[6.1, 2.8, 4.0, 1.3, 'versicolor'],
[6.3, 2.5, 4.9, 1.5, 'versicolor'],
[6.1, 2.8, 4.7, 1.2, 'versicolor'],
[6.4, 2.9, 4.3, 1.3, 'versicolor'],
[6.6, 3.0, 4.4, 1.4, 'versicolor'],
[6.8, 2.8, 4.8, 1.4, 'versicolor'],
[6.7, 3.0, 5.0, 1.7, 'versicolor'],
[6.0, 2.9, 4.5, 1.5, 'versicolor'],
[5.7, 2.6, 3.5, 1.0, 'versicolor'],
[5.5, 2.4, 3.8, 1.1, 'versicolor'],
[5.5, 2.4, 3.7, 1.0, 'versicolor'],
[5.8, 2.7, 3.9, 1.2, 'versicolor'],
[6.0, 2.7, 5.1, 1.6, 'versicolor'],
[5.4, 3.0, 4.5, 1.5, 'versicolor'],
[6.0, 3.4, 4.5, 1.6, 'versicolor'],
[6.7, 3.1, 4.7, 1.5, 'versicolor'],
[6.3, 2.3, 4.4, 1.3, 'versicolor'],
[5.6, 3.0, 4.1, 1.3, 'versicolor'],
[5.5, 2.5, 4.0, 1.3, 'versicolor'],
[5.5, 2.6, 4.4, 1.2, 'versicolor'],
[6.1, 3.0, 4.6, 1.4, 'versicolor'],
[5.8, 2.6, 4.0, 1.2, 'versicolor'],
[5.0, 2.3, 3.3, 1.0, 'versicolor'],
[5.6, 2.7, 4.2, 1.3, 'versicolor'],
[5.7, 3.0, 4.2, 1.2, 'versicolor'],
[5.7, 2.9, 4.2, 1.3, 'versicolor'],
[6.2, 2.9, 4.3, 1.3, 'versicolor'],
[5.1, 2.5, 3.0, 1.1, 'versicolor'],
[5.7, 2.8, 4.1, 1.3, 'versicolor'],
[6.3, 3.3, 6.0, 2.5, 'virginica'],
[5.8, 2.7, 5.1, 1.9, 'virginica'],
[7.1, 3.0, 5.9, 2.1, 'virginica'],
[6.3, 2.9, 5.6, 1.8, 'virginica'],
[6.5, 3.0, 5.8, 2.2, 'virginica'],
[7.6, 3.0, 6.6, 2.1, 'virginica'],
[4.9, 2.5, 4.5, 1.7, 'virginica'],
[7.3, 2.9, 6.3, 1.8, 'virginica'],
[6.7, 2.5, 5.8, 1.8, 'virginica'],
[7.2, 3.6, 6.1, 2.5, 'virginica'],
[6.5, 3.2, 5.1, 2.0, 'virginica'],
[6.4, 2.7, 5.3, 1.9, 'virginica'],
[6.8, 3.0, 5.5, 2.1, 'virginica'],
[5.7, 2.5, 5.0, 2.0, 'virginica'],
[5.8, 2.8, 5.1, 2.4, 'virginica'],
[6.4, 3.2, 5.3, 2.3, 'virginica'],
[6.5, 3.0, 5.5, 1.8, 'virginica'],
[7.7, 3.8, 6.7, 2.2, 'virginica'],
[7.7, 2.6, 6.9, 2.3, 'virginica'],
[6.0, 2.2, 5.0, 1.5, 'virginica'],
[6.9, 3.2, 5.7, 2.3, 'virginica'],
[5.6, 2.8, 4.9, 2.0, 'virginica'],
[7.7, 2.8, 6.7, 2.0, 'virginica'],
[6.3, 2.7, 4.9, 1.8, 'virginica'],
[6.7, 3.3, 5.7, 2.1, 'virginica'],
[7.2, 3.2, 6.0, 1.8, 'virginica'],
[6.2, 2.8, 4.8, 1.8, 'virginica'],
[6.1, 3.0, 4.9, 1.8, 'virginica'],
[6.4, 2.8, 5.6, 2.1, 'virginica'],
[7.2, 3.0, 5.8, 1.6, 'virginica'],
[7.4, 2.8, 6.1, 1.9, 'virginica'],
[7.9, 3.8, 6.4, 2.0, 'virginica'],
[6.4, 2.8, 5.6, 2.2, 'virginica'],
[6.3, 2.8, 5.1, 1.5, 'virginica'],
[6.1, 2.6, 5.6, 1.4, 'virginica'],
[7.7, 3.0, 6.1, 2.3, 'virginica'],
[6.3, 3.4, 5.6, 2.4, 'virginica'],
[6.4, 3.1, 5.5, 1.8, 'virginica'],
[6.0, 3.0, 4.8, 1.8, 'virginica'],
[6.9, 3.1, 5.4, 2.1, 'virginica'],
[6.7, 3.1, 5.6, 2.4, 'virginica'],
[6.9, 3.1, 5.1, 2.3, 'virginica'],
[5.8, 2.7, 5.1, 1.9, 'virginica'],
[6.8, 3.2, 5.9, 2.3, 'virginica'],
[6.7, 3.3, 5.7, 2.5, 'virginica'],
[6.7, 3.0, 5.2, 2.3, 'virginica'],
[6.3, 2.5, 5.0, 1.9, 'virginica'],
[6.5, 3.0, 5.2, 2.0, 'virginica'],
[6.2, 3.4, 5.4, 2.3, 'virginica'],
[5.9, 3.0, 5.1, 1.8, 'virginica']], dtype=object)
```

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Describing the dataset

In [10]:

```
df.describe()
```

Out[10]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Exploratory Data Analysis

Checking for null values

In [11]:

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

Out[11]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

Checking for duplicate values

In [12]:

```
df.duplicated().sum()
```

Out[12]:

```
1
```

Duplicated Value

In [13]:

```
df[df.duplicated()]
```

Out[13]:

	sepal_length	sepal_width	petal_length	petal_width	species
142	5.8	2.7	5.1	1.9	virginica

Dropping duplicated value

In [14]:

```
df.drop_duplicates(inplace = True)
```

In [15]:

```
df.duplicated().sum()
```

Out[15]:

```
0
```

Checking values under the species column

In [16]:

```
df['species'].unique()
```

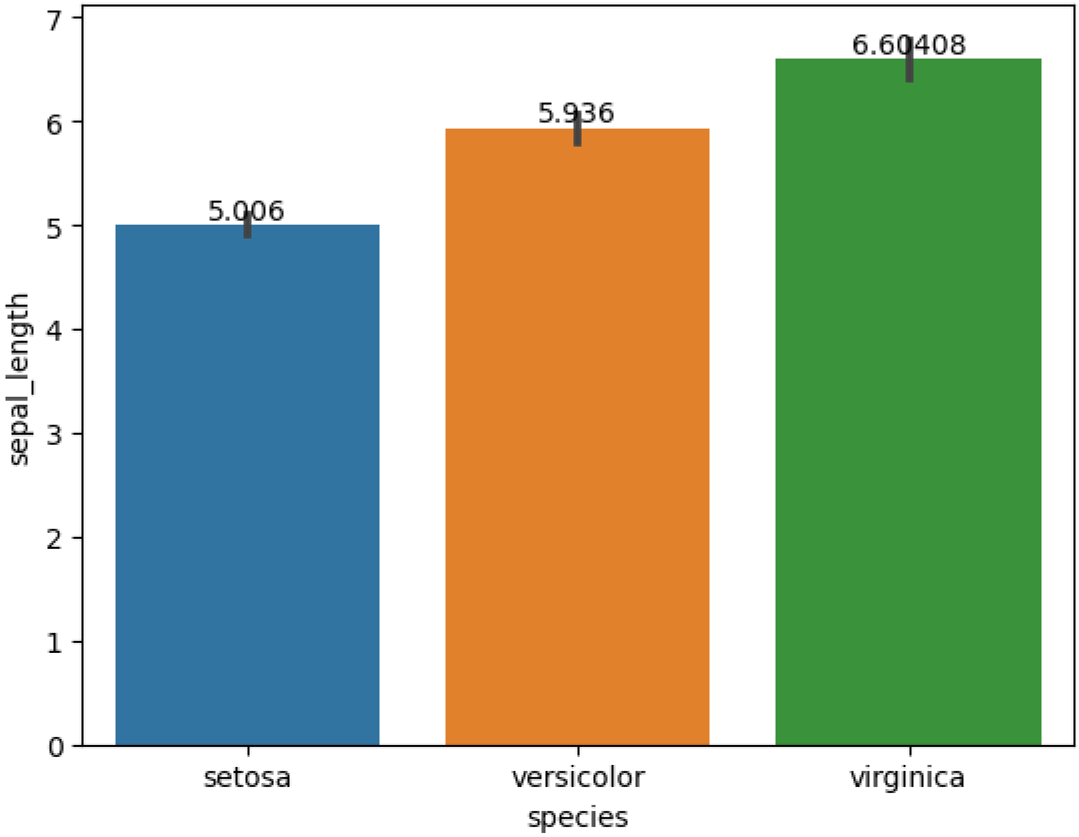
Out[16]:

```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

Visualization using Seaborn

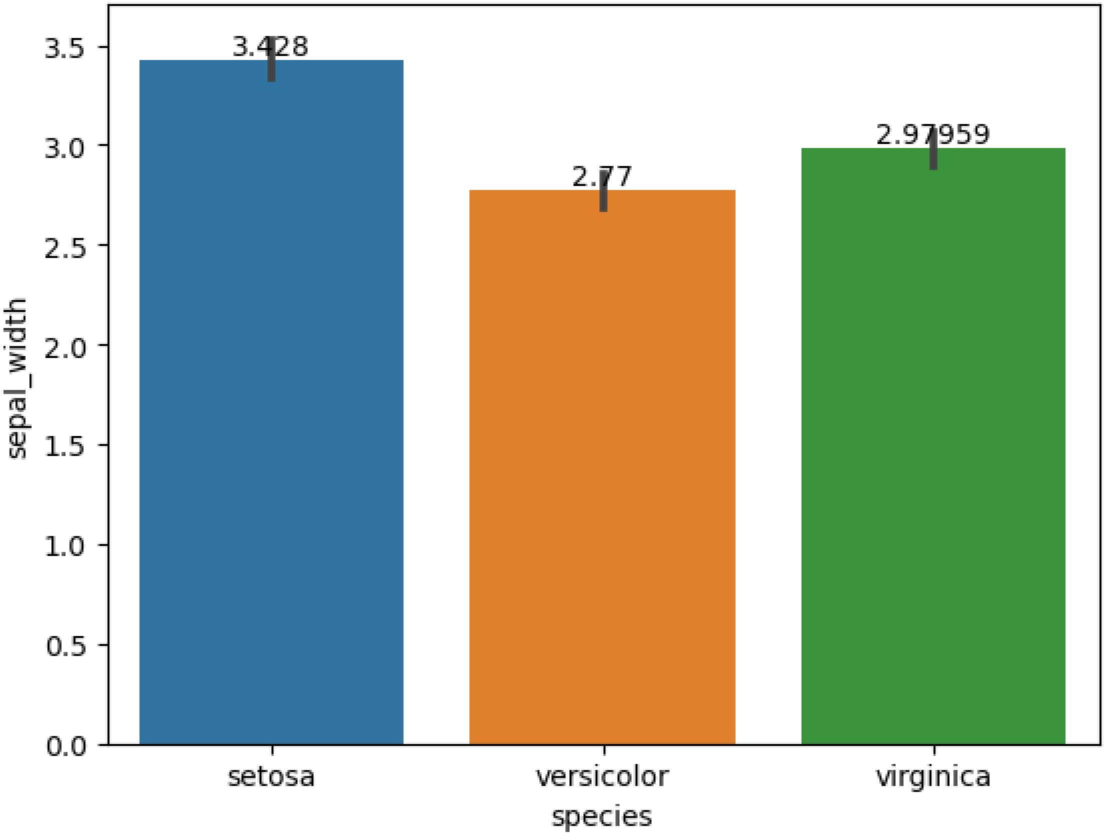
In [17]:

```
ax = sns.barplot(x = "species", y = "sepal_length", data = df)
for bars in ax.containers:
    ax.bar_label(bars)
```



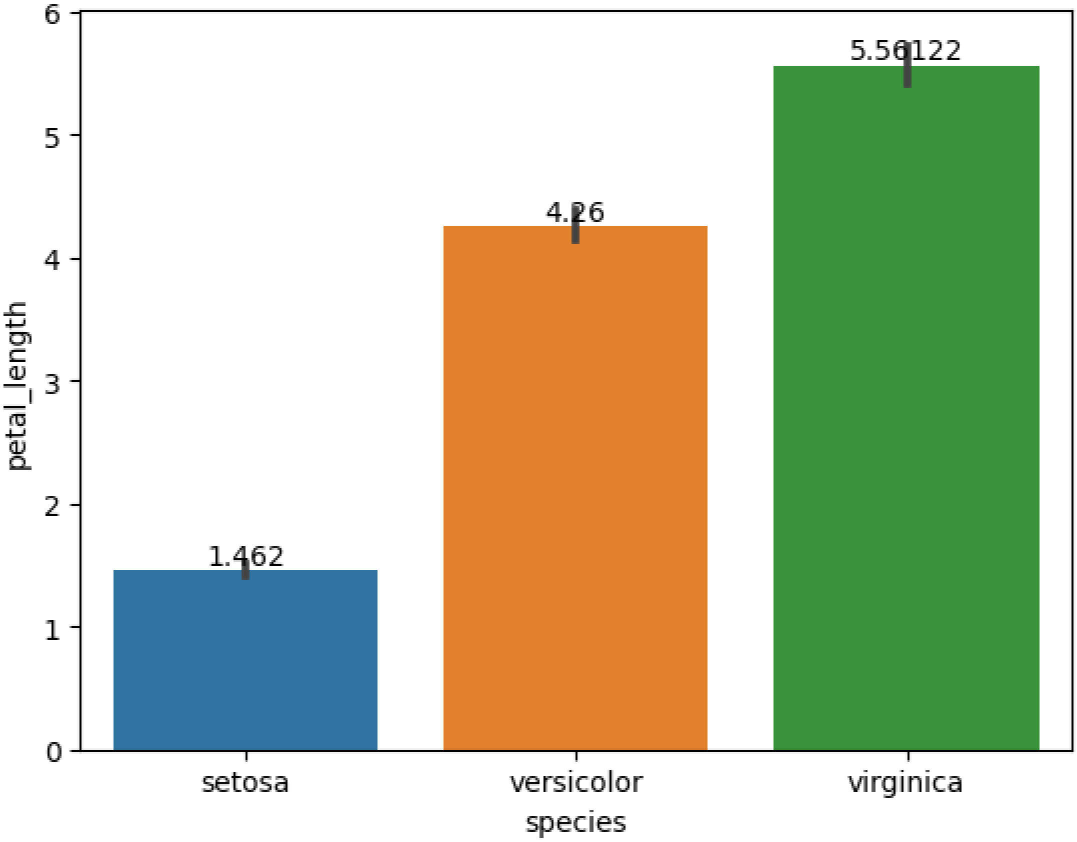
In [18]:

```
ax = sns.barplot(x = "species", y = "sepal_width", data = df)
for bars in ax.containers:
    ax.bar_label(bars)
```



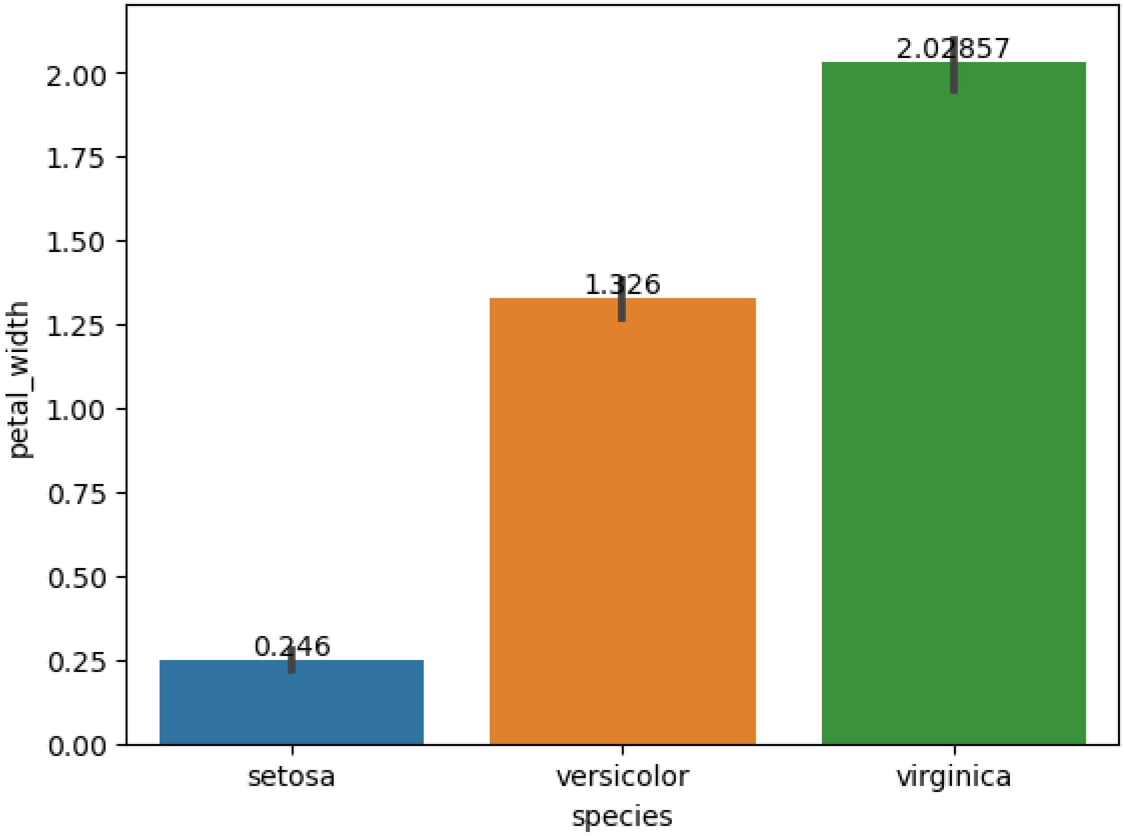
In [19]:

```
ax = sns.barplot(x = "species", y = "petal_length", data = df)
for bars in ax.containers:
    ax.bar_label(bars)
```



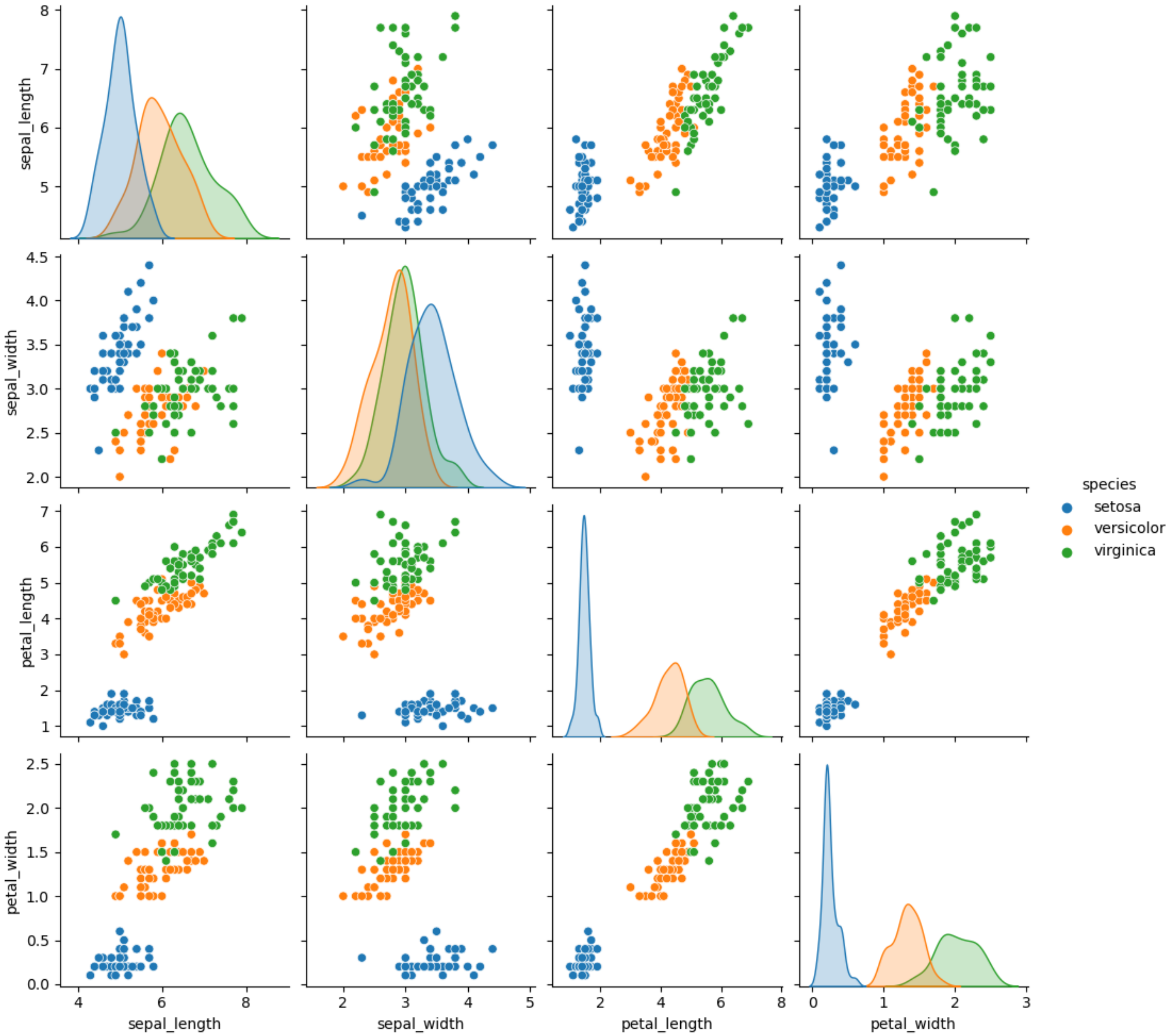
In [20]:

```
ax = sns.barplot(x = "species", y = "petal_width", data = df)
for bars in ax.containers:
    ax.bar_label(bars)
```



In [21]:
```
sns.pairplot(df, hue = 'species')
```

Out[21]:
```
<seaborn.axisgrid.PairGrid at 0x1e054747e10>
```



Mapping the values under the species column

In [22]:

```
species_mapping = {label: idx for idx, label in enumerate(np.unique(df['species']))}
species_mapping
```

Out[22]:

```
{'setosa': 0, 'versicolor': 1, 'virginica': 2}
```

In [23]:

```
df['species'] = df['species'].map(species_mapping)
df['species']
```

Out[23]:

```
0      0
1      0
2      0
3      0
4      0
      ..
145    2
146    2
147    2
148    2
149    2
Name: species, Length: 149, dtype: int64
```

In [24]:

```
df.corr()
```

Out[24]:

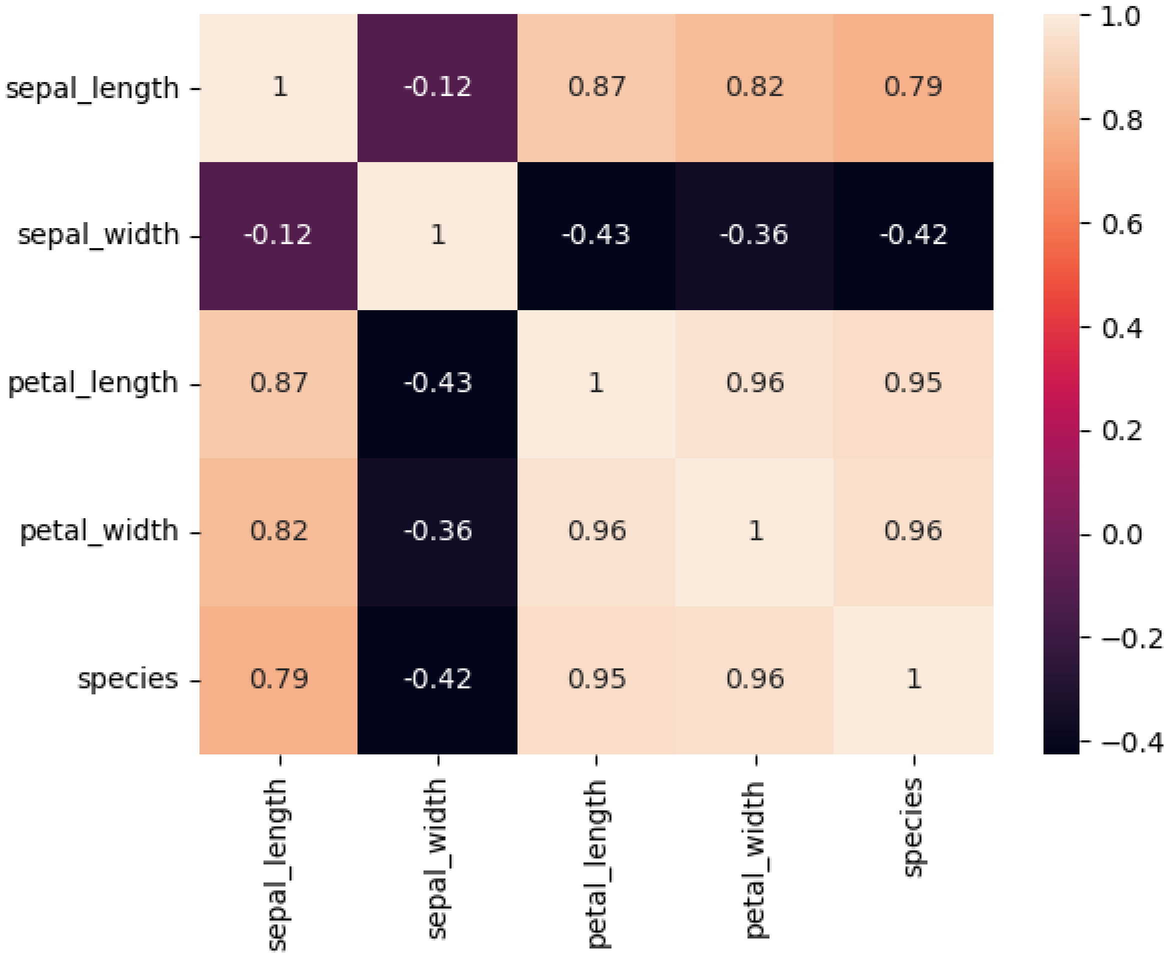
	sepal_length	sepal_width	petal_length	petal_width	species
sepal_length	1.000000	-0.118129	0.873738	0.820620	0.786971
sepal_width	-0.118129	1.000000	-0.426028	-0.362894	-0.422987
petal_length	0.873738	-0.426028	1.000000	0.962772	0.949402
petal_width	0.820620	-0.362894	0.962772	1.000000	0.956514
species	0.786971	-0.422987	0.949402	0.956514	1.000000

In [25]:

```
sns.heatmap(df.corr(), annot = True)
```

Out[25]:

```
<Axes: >
```



Machine Learning Begins !!

In [26]:

```
x = df.iloc[:, df.columns != 'species']
y = df.iloc[:, df.columns == 'species']
```

In [27]:

```
x
```

Out[27]:

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2

3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

149 rows × 4 columns

In [28]:

```
y
```

Out[28]:

	species
0	0
1	0
2	0
3	0
4	0
...	...
145	2
146	2
147	2
148	2
149	2

149 rows × 1 columns

In [29]:

```
#train test split
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25)
```

In [30]:

```
x_train
```

Out[30]:

	sepal_length	sepal_width	petal_length	petal_width
16	5.4	3.9	1.3	0.4
81	5.5	2.4	3.7	1.0
44	5.1	3.8	1.9	0.4
127	6.1	3.0	4.9	1.8
122	7.7	2.8	6.7	2.0
...	...	...	...	...
137	6.4	3.1	5.5	1.8

148	6.2	3.4	5.4	2.3
93	5.0	2.3	3.3	1.0
40	5.0	3.5	1.3	0.3
145	6.7	3.0	5.2	2.3

111 rows × 4 columns

In [31]:

```
x_test
```

Out[31]:

	sepal_length	sepal_width	petal_length	petal_width
5	5.4	3.9	1.7	0.4
25	5.0	3.0	1.6	0.2
144	6.7	3.3	5.7	2.5
90	5.5	2.6	4.4	1.2
133	6.3	2.8	5.1	1.5
77	6.7	3.0	5.0	1.7
116	6.5	3.0	5.5	1.8
65	6.7	3.1	4.4	1.4
30	4.8	3.1	1.6	0.2
119	6.0	2.2	5.0	1.5
111	6.4	2.7	5.3	1.9
64	5.6	2.9	3.6	1.3
37	4.9	3.6	1.4	0.1
128	6.4	2.8	5.6	2.1
98	5.1	2.5	3.0	1.1
114	5.8	2.8	5.1	2.4
126	6.2	2.8	4.8	1.8
94	5.6	2.7	4.2	1.3
54	6.5	2.8	4.6	1.5
140	6.7	3.1	5.6	2.4
32	5.2	4.1	1.5	0.1
7	5.0	3.4	1.5	0.2
135	7.7	3.0	6.1	2.3
10	5.4	3.7	1.5	0.2
22	4.6	3.6	1.0	0.2
84	5.4	3.0	4.5	1.5
85	6.0	3.4	4.5	1.6
9	4.9	3.1	1.5	0.1
6	4.6	3.4	1.4	0.3

91	6.1	3.0	4.6	1.4
99	5.7	2.8	4.1	1.3
108	6.7	2.5	5.8	1.8
97	6.2	2.9	4.3	1.3
59	5.2	2.7	3.9	1.4
106	4.9	2.5	4.5	1.7
79	5.7	2.6	3.5	1.0
1	4.9	3.0	1.4	0.2
36	5.5	3.5	1.3	0.2

In [32]:

```
y_train
```

Out[32]:

	species
16	0
81	1
44	0
127	2
122	2
...	...
137	2
148	2
93	1
40	0
145	2

111 rows × 1 columns

In [33]:

```
y_test
```

Out[33]:

	species
5	0
25	0
144	2
90	1
133	2
77	1
116	2
65	1
30	0
119	2

111	2
64	1
37	0
128	2
98	1
114	2
126	2
94	1
54	1
140	2
32	0
7	0
135	2
10	0
22	0
84	1
85	1
9	0
6	0
91	1
99	1
108	2
97	1
59	1
106	2
79	1
1	0
36	0

In [34]:

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(random_state = 42)
rf
```

Out[34]:

```
▾        RandomForestClassifier
RandomForestClassifier(random_state=42)
```

In [35]:

```
import time
from sklearn.model_selection import GridSearchCV
np.random.seed(42)
start = time.time()

param_dict = {'max_depth': [2,3,4,5],
              'bootstrap': [True, False],
              'max_features': ['auto', 'sqrt', 'log2', None],
              'criterion': ['gini', 'entropy']}

cv_rf = GridSearchCV(rf, cv = 10, param_grid = param_dict, n_jobs = 3)

cv_rf.fit(x_train, y_train)
print("Best Parameters: ", cv_rf.best_params_)
end = time.time()
print('Time taken: %0.2f'%(end-start))
```

```
Best Parameters:  {'bootstrap': True, 'criterion': 'gini', 'max_depth': 3, 'max_feature
s': 'auto'}
Time taken: 47.09
```

```
C:\Users\Administrator\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:90
9: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Ple
ase change the shape of y to (n_samples,), for example using ravel().
  self.best_estimator_.fit(X, y, **fit_params)
C:\Users\Administrator\anaconda3\Lib\site-packages\sklearn\ensemble\_forest.py:424: Futu
reWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3.
To keep the past behaviour, explicitly set `max_features='sqrt'` or remove this paramete
r as it is also the default value for RandomForestClassifiers and ExtraTreesClassifiers.
  warn(
```

In [36]:

```
rf.set_params(criterion = 'gini', max_features = 'log2', max_depth = 2)
```

Out[36]:

```
▾                 RandomForestClassifier
RandomForestClassifier(max_depth=2, max_features='log2', random_state=42)
```

In [37]:

```
rf.fit(x_train, y_train)
```

```
C:\Users\Administrator\AppData\Local\Temp\ipykernel_1088\3512478466.py:1: DataConversion
Warning: A column-vector y was passed when a 1d array was expected. Please change the sh
ape of y to (n_samples,), for example using ravel().
  rf.fit(x_train, y_train)
```

Out[37]:

```
▾                 RandomForestClassifier
RandomForestClassifier(max_depth=2, max_features='log2', random_state=42)
```

In [38]:

```
y_pred = rf.predict(x_test)
y_pred
```

Out[38]:

```
array([0, 0, 2, 1, 2, 2, 2, 1, 0, 1, 2, 1, 0, 2, 1, 2, 2, 1, 1, 2, 0, 0,
       2, 0, 0, 1, 1, 0, 0, 1, 1, 2, 1, 1, 1, 1, 0, 0], dtype=int64)
```

In [39]:

```
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
accuracy_rf = accuracy_score(y_pred, y_test)
accuracy_rf
```

Out[39]:

```
0.9210526315789473
```

In [40]:

```
print(classification_report(y_pred, y_test))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        12
           1       0.93      0.87      0.90        15
           2       0.83      0.91      0.87        11

    accuracy                           0.92        38
   macro avg       0.92      0.93      0.92        38
```

```
weighted avg       0.92      0.92      0.92        38
```

In [41]:

```
cmat = confusion_matrix(y_test, y_pred)
cmat
```

Out[41]:

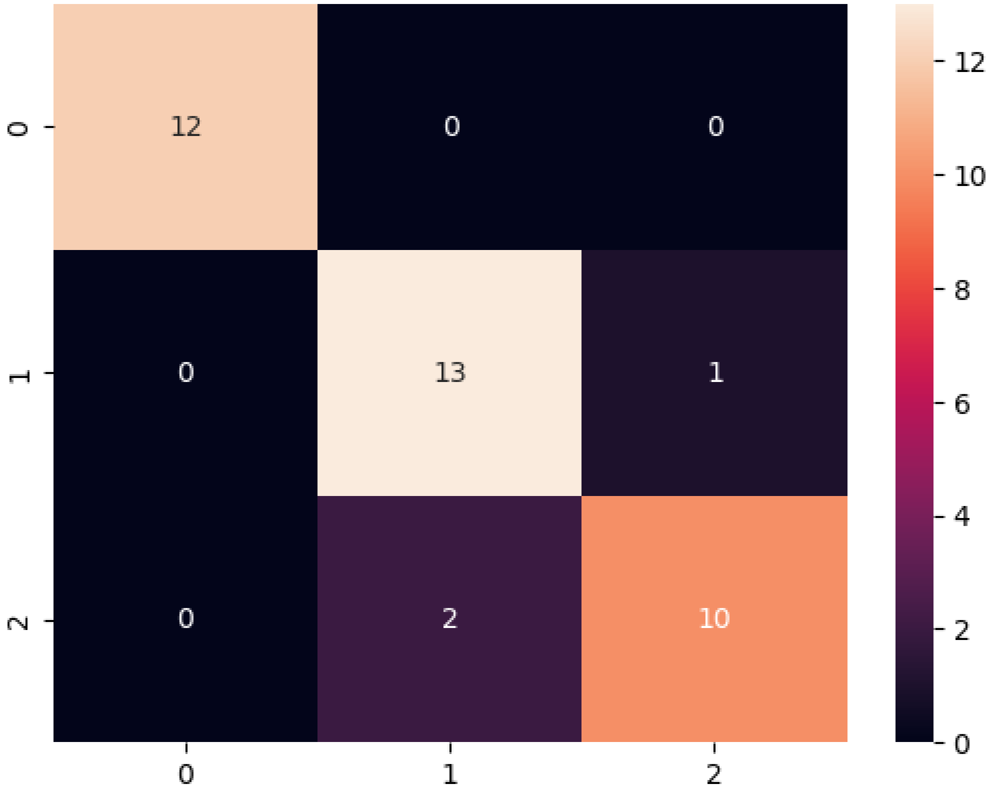
```
array([[12,  0,  0],
       [ 0, 13,  1],
       [ 0,  2, 10]], dtype=int64)
```

In [42]:

```
#in heatmap
sns.heatmap(cmat, annot = True)
```

Out[42]:

```
<Axes: >
```



In [ ]:

In [ ]:

In [ ]:

In [ ]: