In [1]:
```
import pandas as pd
```

In [25]:
```
df=pd.read_csv("D:/AIML/Iris.csv")
```

In [26]:
```
df.head(2)
```

Out[26]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa

In [27]:
```
df.isna().sum()
```

Out[27]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [28]:

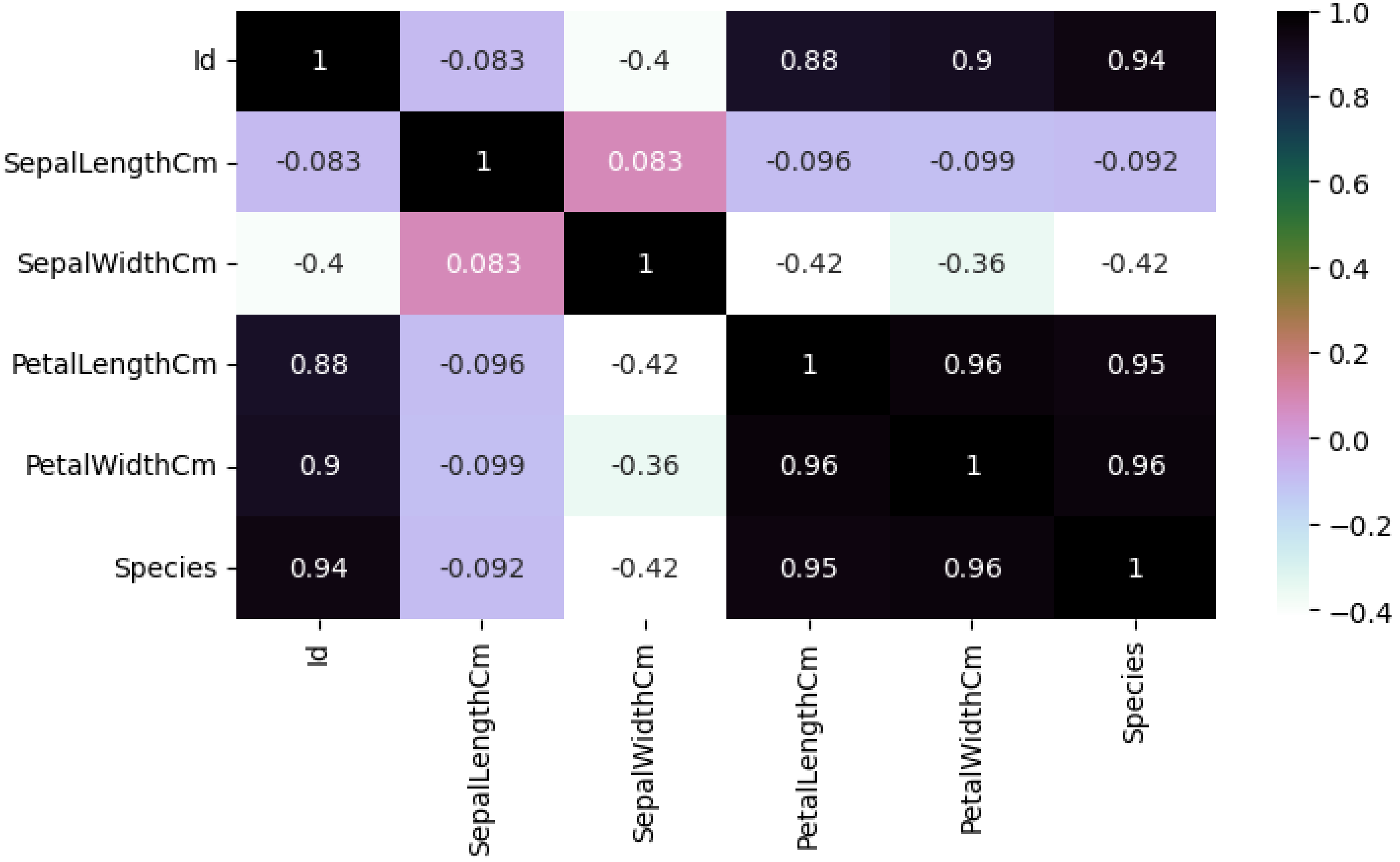
```
df["Species"]=df["Species"].astype("category")
df["Species"]=df["Species"].cat.codes
df
```

Out[28]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	0
1	2	4.9	3.0	1.4	0.2	0
2	3	4.7	3.2	1.3	0.2	0
3	4	4.6	3.1	1.5	0.2	0
4	5	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	2
146	147	6.3	2.5	5.0	1.9	2
147	148	6.5	3.0	5.2	2.0	2
148	149	6.2	3.4	5.4	2.3	2
149	150	5.9	3.0	5.1	1.8	2

150 rows × 6 columns

In [29]:
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(8,4))
sns.heatmap(df.corr(), annot=True, cmap='cubehelix_r')
plt.show()
```



In [30]:
```
df.drop("Id", axis=1, inplace = True)
```

In [36]:

```
from sklearn.model_selection import train_test_split
train,test=train_test_split(df,test_size=0.2)
print(train.shape)
print(test.shape)
```

```
(120, 5)
(30, 5)
```

In [39]:

```
train_x=train[["SepalLengthCm","SepalWidthCm","PetalLengthCm","PetalWidthCm"]]
train_y=train.Species
test_x=train[["SepalLengthCm","SepalWidthCm","PetalLengthCm","PetalWidthCm"]]
test_y=train.Species
```

In [47]:

```
train_x.head(1)
```

Out[47]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
71	6.1	2.8	4.0	1.3

In [46]:

```
test_x.head(1)
```

Out[46]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
71	6.1	2.8	4.0	1.3

In [45]:

```
train_y.head(1)
```

Out[45]:

```
71    1
Name: Species, dtype: int8
```

In [91]:

```
from sklearn import svm
model=svm.SVC()

model.fit(train_x,train_y)

prediction1 = model.predict(train_x)

from sklearn.metrics import accuracy_score
accuracy = accuracy_score(test_y,prediction1)
print("Accuracy: {:.2f}%".format(accuracy * 100))
```

```
Accuracy: 96.67%
```

In [80]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import svm
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier
```

In [98]:

```
model=LogisticRegression()
model.fit(train_x,train_y)

prediction2=model.predict(train_x)

accuracy = accuracy_score(test_y,prediction2)
print("Accuracy: {:.2f}%".format(accuracy*100))
```

```
Accuracy: 96.67%
```

In [95]:

```
model = DecisionTreeClassifier()
model.fit(train_x, train_y)
prediction3 = model.predict(train_x)
accuracy = accuracy_score(test_y,prediction3)
print("Accuracy: {:.2f}%".format(accuracy*100))
```

```
Accuracy: 100.00%
```

In [96]:

```
model = KNeighborsClassifier(n_neighbors=3)
model.fit(train_x, train_y)
prediction4 = model.predict(test_x)
accuracy = accuracy_score(test_y,prediction4)
print("Accuracy: {:.2f}%".format(accuracy*100))
```

```
Accuracy: 95.83%
```

In [99]:

```
df.isnull()
```

Out[99]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False

150 rows × 5 columns

In [101]:

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

Out[101]:

```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [ ]:

```

```