In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv("Iris (1).csv")
```

In [3]:
```
df
```

Out[3]:

	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
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [4]:
```
## DATA PREPROCESSING ##
```

In [5]:
```
df.head()
```

Out[5]:

	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
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [6]:
```
df.tail()
```

Out[6]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

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

Out[8]: (150, 6)

In [9]:
```
df.columns
```

Out[9]:
```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWid
thCm',
       'Species'],
      dtype='object')
```

In [10]:
```
df.duplicated().sum()
```

Out[10]: 0

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [12]:
```
df.isnull()
```

Out[12]:

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

150 rows × 6 columns

In [13]:
```
df['Species']
```

Out[13]:
```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
             ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object
```

In [14]:
```
count=df['Species'].value_counts()
print(count)
```

```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

In [15]:
```
df['SepalLengthCm']
```

Out[15]:
```
0      5.1
1      4.9
2      4.7
3      4.6
4      5.0
      ...
145    6.7
146    6.3
147    6.5
148    6.2
149    5.9
Name: SepalLengthCm, Length: 150, dtype: float64
```

In [16]:
```
df['SepalWidthCm']
```

Out[16]:
```
0      3.5
1      3.0
2      3.2
3      3.1
4      3.6
      ...
145    3.0
146    2.5
147    3.0
148    3.4
149    3.0
Name: SepalWidthCm, Length: 150, dtype: float64
```

In [17]:
```
df['PetalLengthCm']
```

Out[17]:
```
0      1.4
1      1.4
2      1.3
3      1.5
4      1.4
      ...
145    5.2
146    5.0
147    5.2
148    5.4
149    5.1
Name: PetalLengthCm, Length: 150, dtype: float64
```

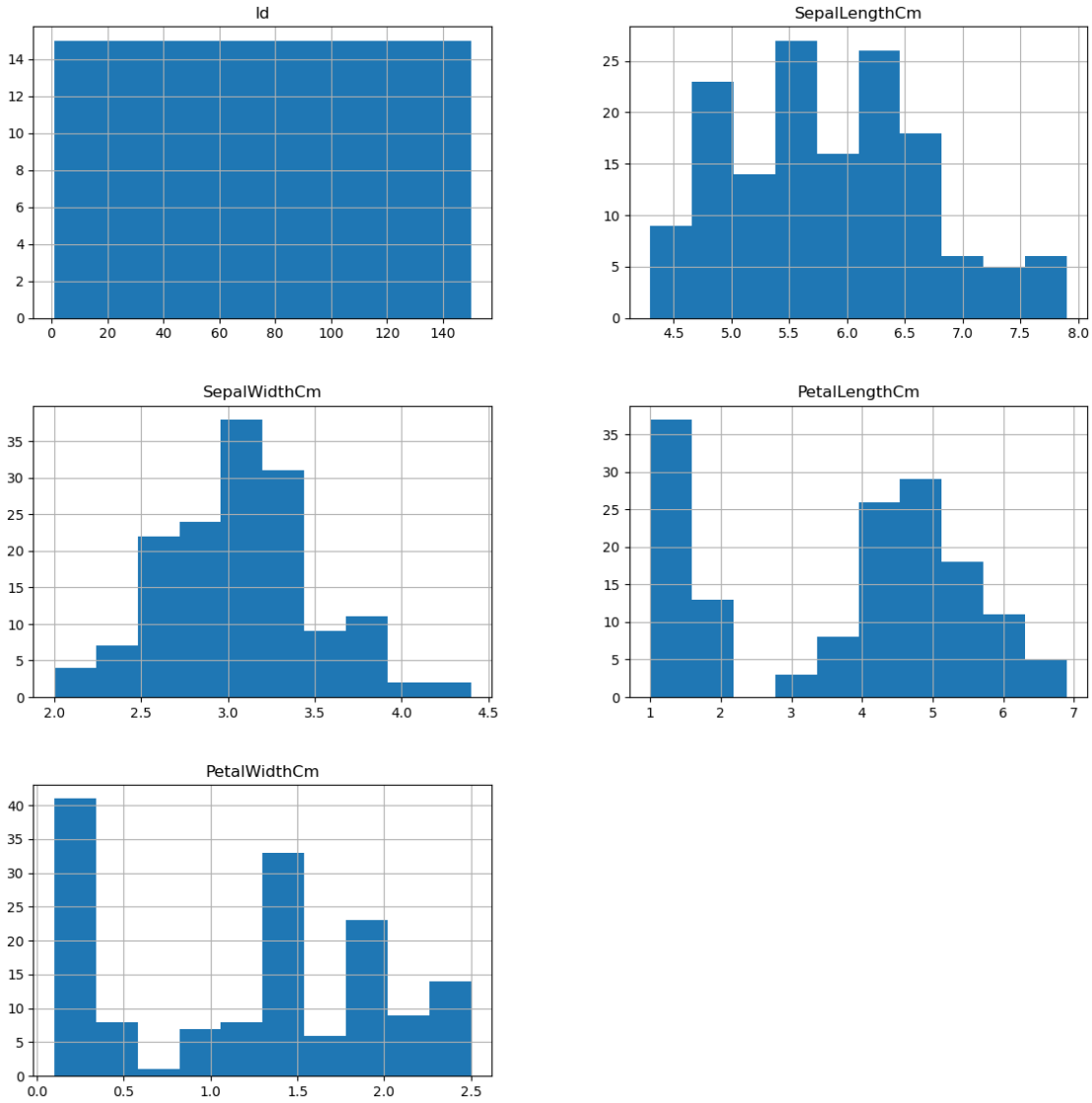
In [18]:
```
df['PetalWidthCm']
```

Out[18]:
```
0      0.2
1      0.2
2      0.2
3      0.2
4      0.2
      ...
145    2.3
146    1.9
147    2.0
148    2.3
149    1.8
Name: PetalWidthCm, Length: 150, dtype: float64
```

In [19]:
```
## VISUALIZATION ##
```

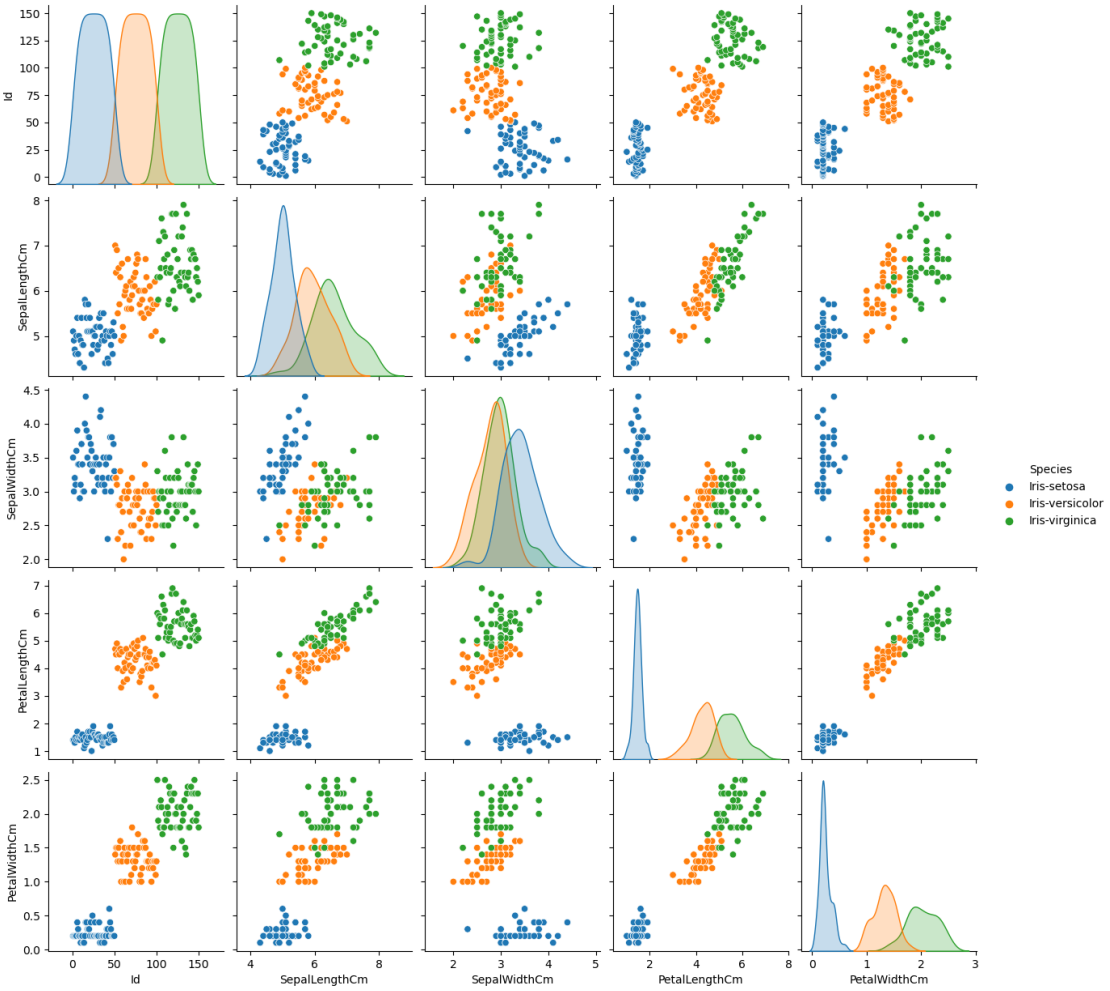
In [20]:
```
df.hist(figsize=(14,14))
plt.show
```

Out[20]: <function matplotlib.pyplot.show(close=None, block=None)>

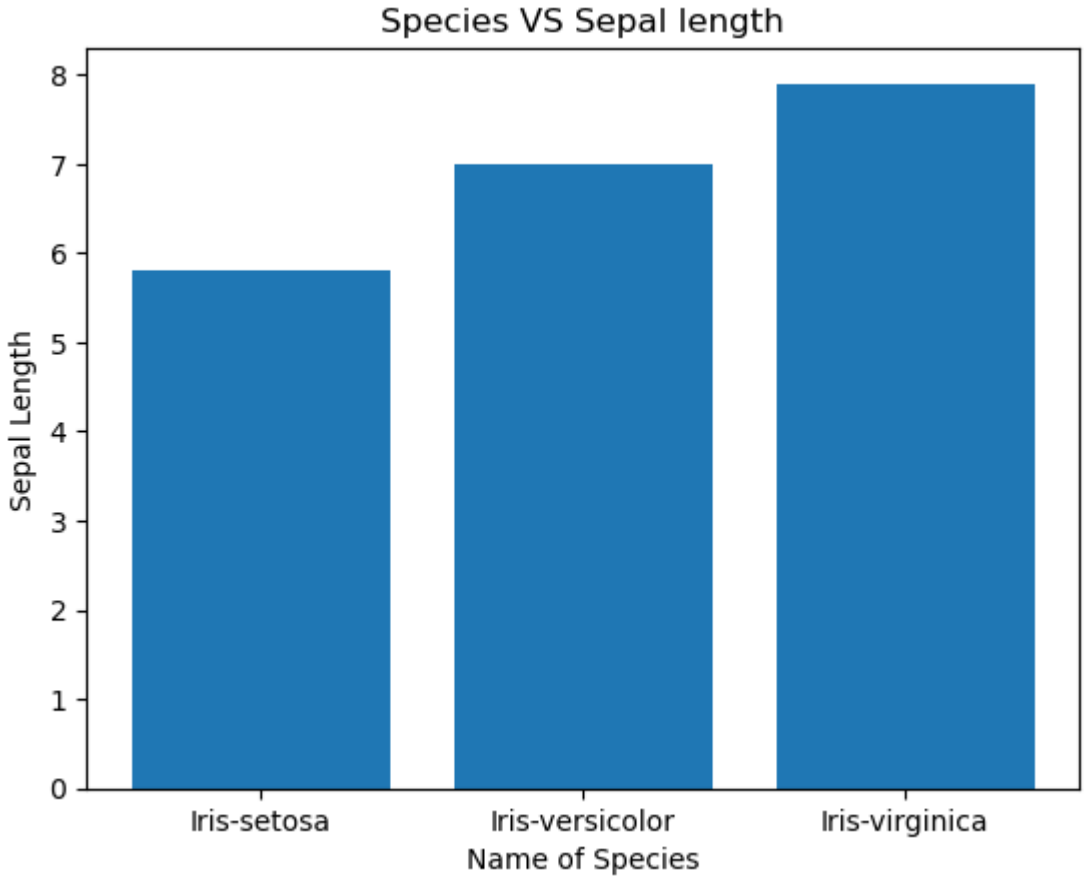


In [21]:
```
plt.figure(figsize=(7,7))
sns.pairplot(df,hue='Species')
plt.show()
```

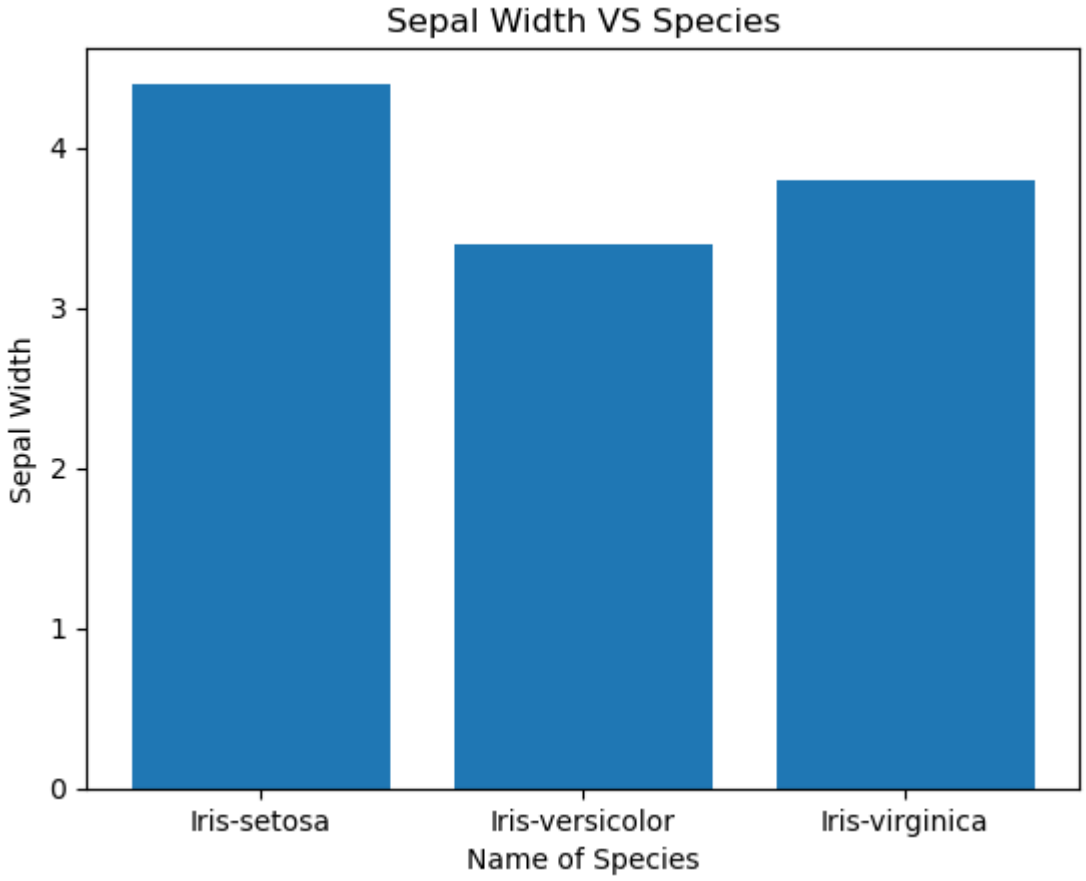
```
<Figure size 700x700 with 0 Axes>
```



In [22]:
```
plt.bar(df['Species'],df['SepalLengthCm'])
plt.title("Species VS Sepal length")
plt.xlabel("Name of Species")
plt.ylabel("Sepal Length")
plt.show()
```



In [23]:
```
plt.bar(df['Species'],df['SepalWidthCm'])
plt.title("Sepal Width VS Species")
plt.xlabel("Name of Species")
plt.ylabel("Sepal Width")
plt.show()
```

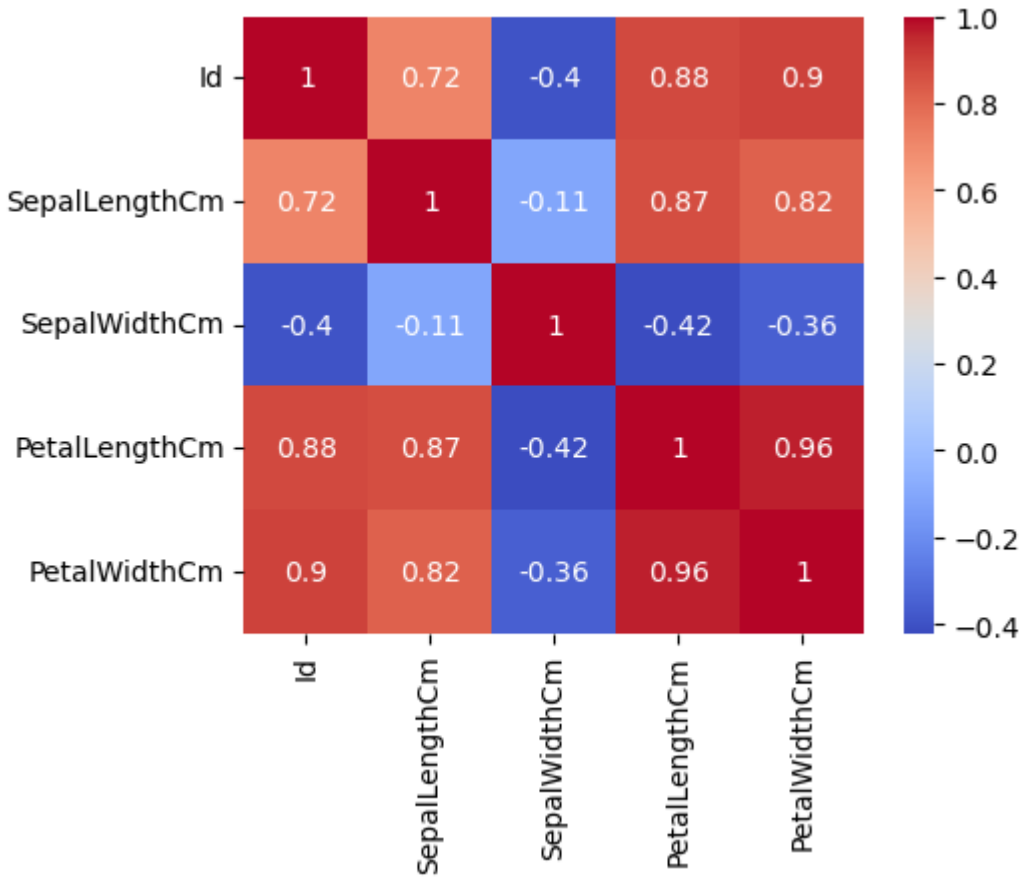


In [24]:
```
plt.bar(df['Species'],df['PetalLengthCm'])
plt.title("Petal Length VS Species")
plt.xlabel("Name of the Species")
plt.ylabel("Petal Length")
plt.show()
```

In [25]:
```
plt.bar(df['Species'],df['PetalWidthCm'])
plt.title("Petal Width VS Spcies")
plt.xlabel("Name of Species")
plt.ylabel("Petal Width")
plt.show()
```

In [26]:
```
corr = df.corr()
fig, ax = plt.subplots(figsize=(5,4))
sns.heatmap(corr, annot=True, ax=ax, cmap = 'coolwarm')
```

Out[26]: <AxesSubplot:>



In [44]:
```
## CHANGING THE CATEGORICAL ATTRIBUTES INTO NUMERIC DATA FOR BETTER ANALYS

df.replace({'Species':{'Iris-setosa':0,'Iris-virginica':1,"Iris-versicolor
```

In [45]:
```
## Splitting the data ##

x = df.drop(['Species'], axis=1)
y = df['Species']
print(len(x), len(y))
print(x)
print(y)
print(x.shape)
print(y.shape)
```

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

[150 rows x 5 columns]
0      0
1      0
2      0
3      0
4      0
      ..
145    1
146    1
147    1
148    1
149    1
Name: Species, Length: 150, dtype: int64
(150, 5)
(150,)
```

In [46]:
```
## Training and Test Data ##
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test=train_test_split(x,y,test_size=1/10,rando
```

In [47]:
```
## Linear Regression ##

from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x_train,y_train)
model.score(x_test,y_test)
```

Out[47]: 0.4858271252759866

In [48]:
```
# print metric to get performance
print("Accuracy: ",model.score(x_test, y_test) * 100)
```

```
Accuracy:  48.58271252759866
```

In [49]:
```
# decision tree
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(x_train, y_train)
```

Out[49]: DecisionTreeClassifier()

In [50]:
```
# print metric to get performance
print("Accuracy: ",model.score(x_test, y_test)*100 )
```

```
Accuracy:  93.33333333333333
```