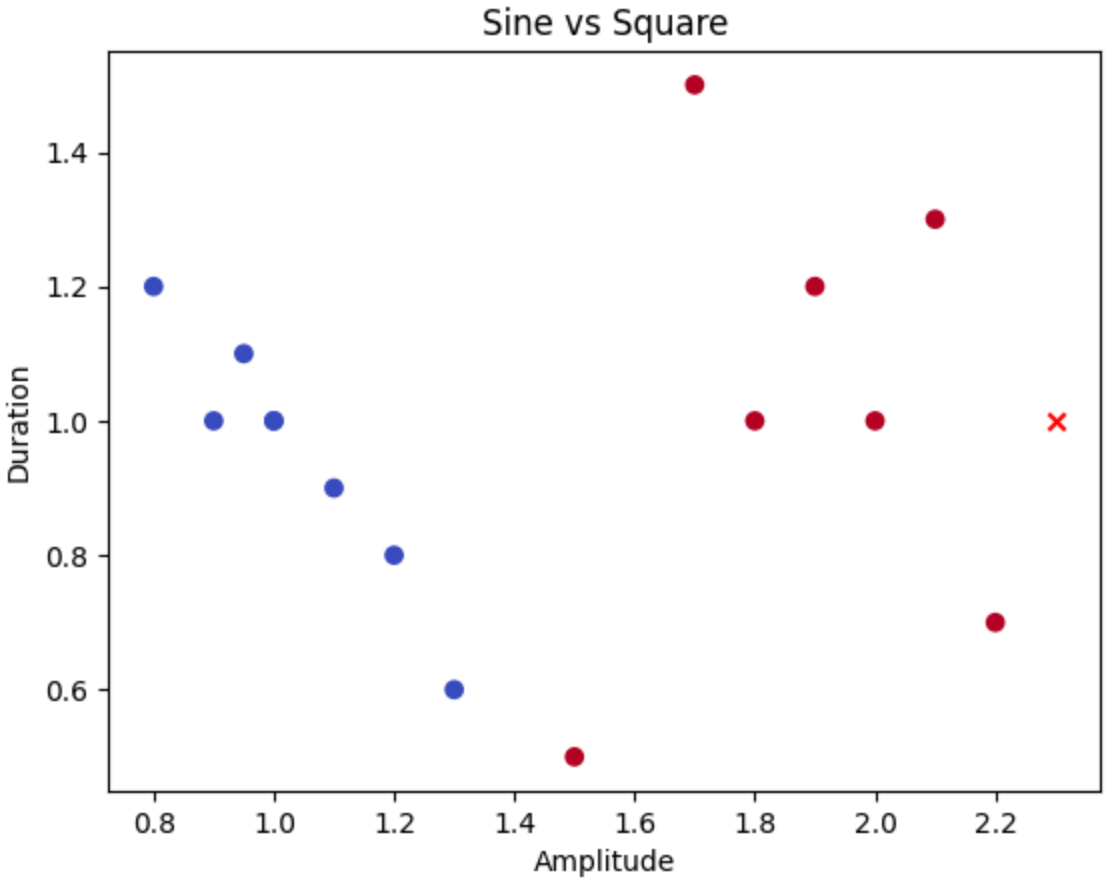
In [ ]:

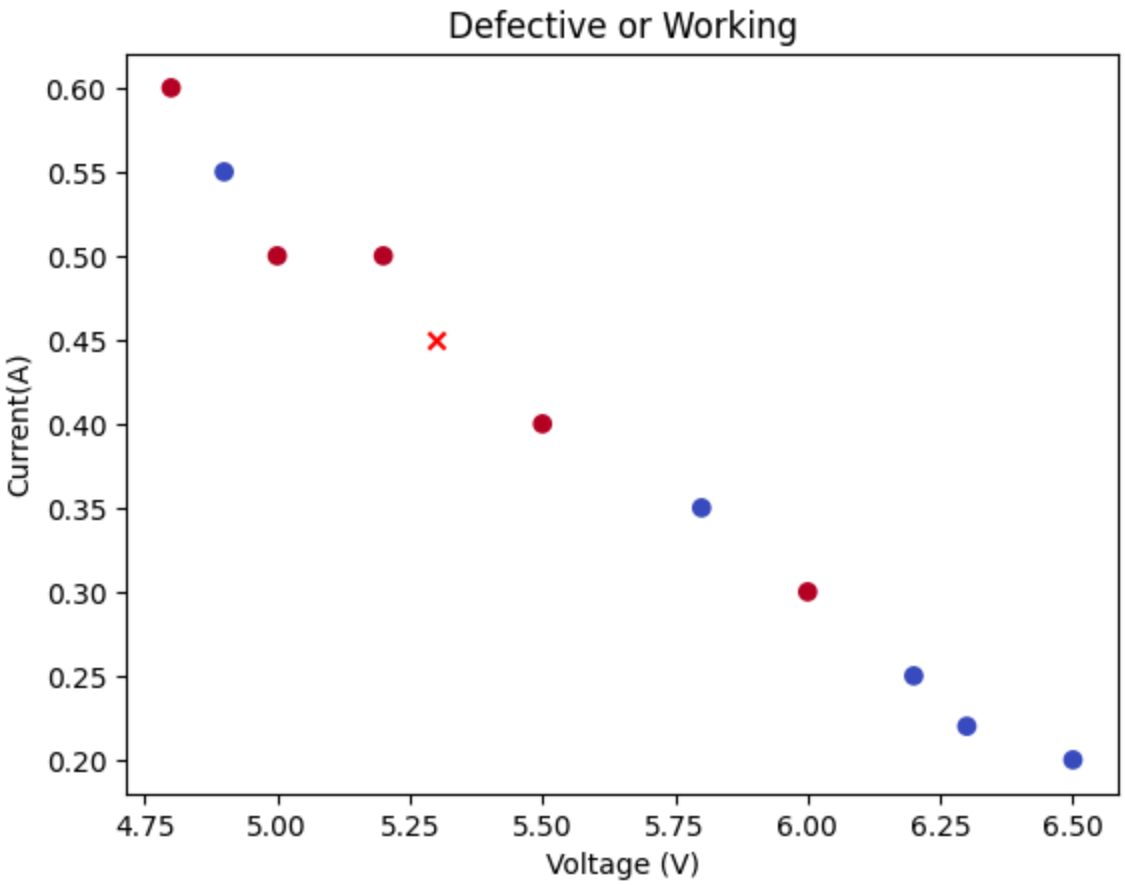
```
from sklearn.neighbors import KNeighborsClassifier
import matplotlib.pyplot as plt
import pandas as pd
url="https://docs.google.com/spreadsheets/d/1ZlVczEcOSM3FKTsFpWgiIUvBd1-6VJQo
df=pd.read_csv(url)
x=df['Amplitude']
y=df['Duration']
classes=df['Label'].map({'Sine':0,'Square':1})
data=list(zip(x,y))
knn=KNeighborsClassifier(n_neighbors=1)
knn.fit(data,classes)
newx=2.3
newy=1
newp=[(newx,newy)]
prediction=knn.predict(newp)
plt.scatter(x,y,c=classes,cmap='coolwarm')
pc='red' if prediction[0]==1 else 'blue'
plt.scatter(newx,newy,c=pc,marker='x')
plt.xlabel('Amplitude')
plt.ylabel('Duration')
plt.title('Sine vs Square')
plt.show()
```



In [ ]:

```
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
```

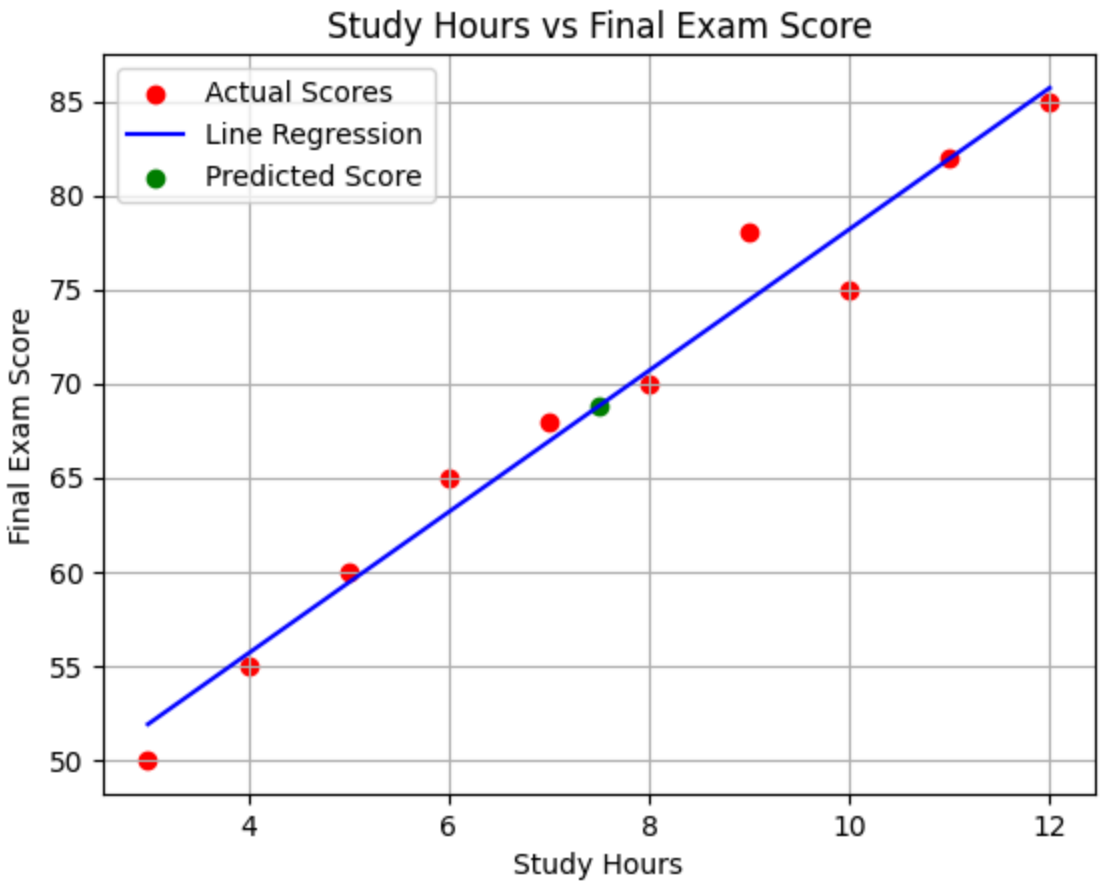
```
url="https://docs.google.com/spreadsheets/d/1RotohRlpi22vGQlt-IMk9fgl1jXzt6rh
df=pd.read_csv(url)
x=df['Voltage (V)']
y=df['Current(A)']
classes=df['Status'].map({'Defective':0,'Working':1})
d=list(zip(x,y))
knn=KNeighborsClassifier(n_neighbors=1)
knn.fit(d,classes)
new_x=5.30
new_y=0.45
new_p=[(new_x,new_y)]
prediction=knn.predict(new_p)
plt.scatter(x,y,c=classes,cmap='coolwarm')
pc='red' if prediction[0]==1 else 'blue'
plt.scatter(new_x,new_y,c=pc,marker='x')
plt.xlabel('Voltage (V)')
plt.ylabel('Current(A)')
plt.title('Defective or Working')
plt.show()
```



In [ ]:

```
import matplotlib.pyplot as plt
from scipy import stats
import pandas as pd
df=pd.read_csv("https://docs.google.com/spreadsheets/d/109Tl6jF7jBZsTKuRYGVT0
x=df['Study Hours']
y=df['Final Exam Score']
slope,intercept,r,p,std_err=stats.linregress(x,y)
def myfun(x):
  return slope*x+intercept
```

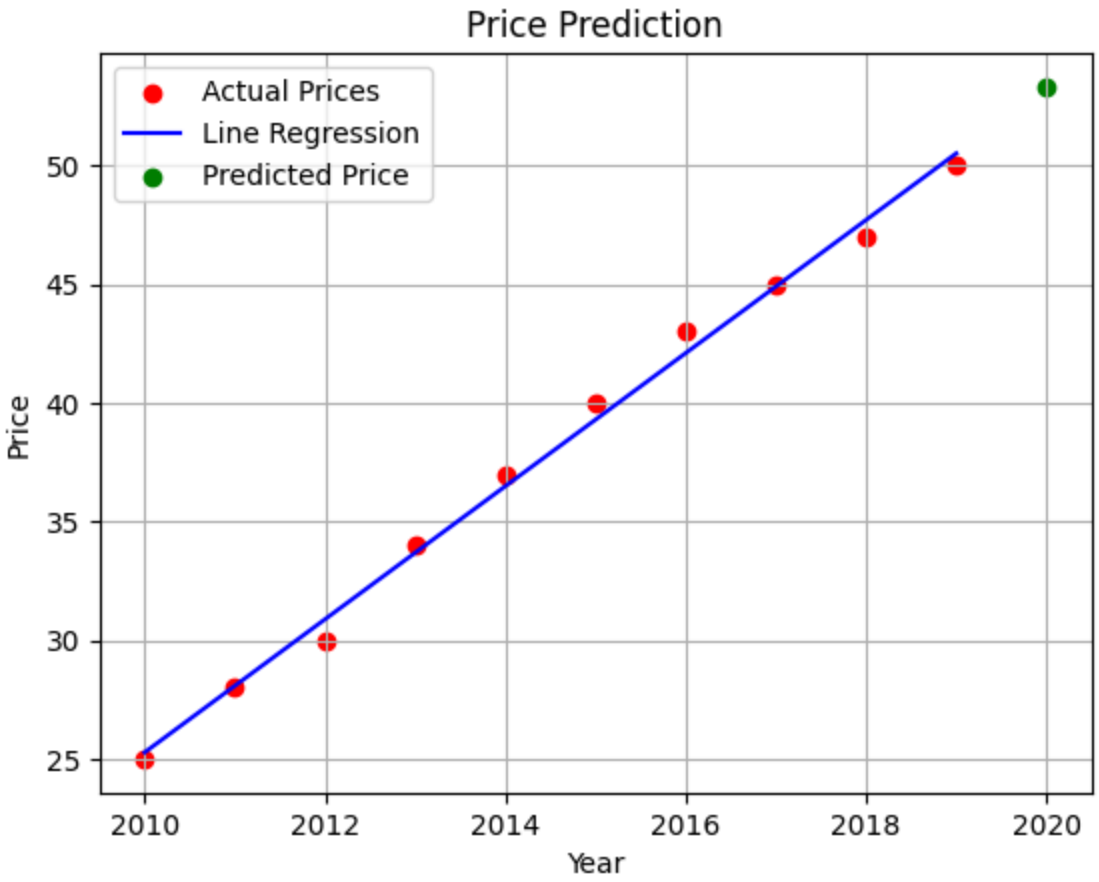
```
mymodel=list(map(myfun,x))
plt.scatter(x,y,color='red',label='Actual Scores')
plt.plot(x,mymodel,color='blue',label='Line Regression')
plt.scatter(7.5,myfun(7.5),color='green',label='Predicted Score')
plt.legend()
plt.title('Study Hours vs Final Exam Score')
plt.xlabel('Study Hours')
plt.ylabel('Final Exam Score')
plt.grid(True)
plt.show()
```



In [48]:

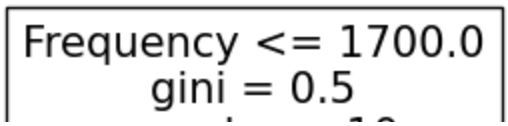
```
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
df=pd.read_csv("https://docs.google.com/spreadsheets/d/10domE5N12-sk1Z8lhIka5
x=df['Year']
y=df['Price']
slope,intercept,r,p,std_err=stats.linregress(x,y)
def myfun(x):
  return slope*x+intercept
mymodel=list(map(myfun,x))
plt.scatter(x,y,color='red',label='Actual Prices')
plt.plot(x,mymodel,color='blue',label='Line Regression')
plt.scatter(2020,myfun(2020),color='green',label='Predicted Price')
plt.legend()
plt.title('Price Prediction')
plt.xlabel('Year')
plt.ylabel('Price')
```

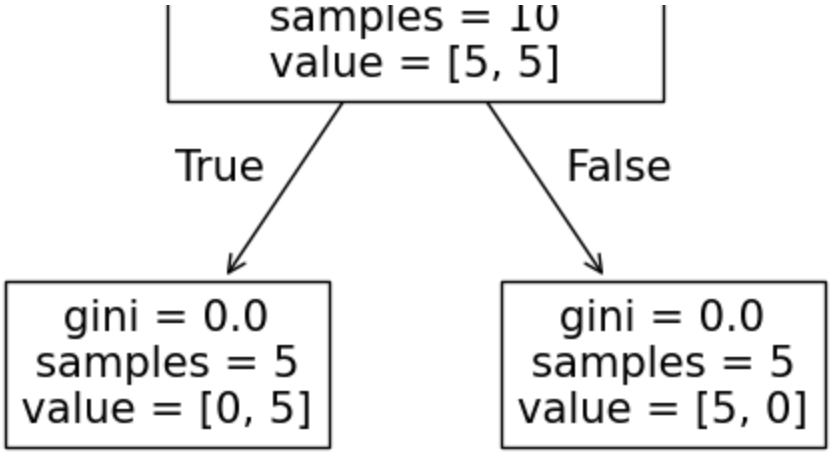
```
plt.grid(True)
plt.show()
```



In [47]:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
url="https://docs.google.com/spreadsheets/d/1lagVzfL2OMyGtdIVLMW5vFmdnrw4LwxX
df=pd.read_csv(url)
d={'T1':0,'T2':1,'T3':2}
df['Transmitter']=df['Transmitter'].map(d)
d={'Good':0,'Bad':1}
df['Signal Quality']=df['Signal Quality'].map(d)
features=['Transmitter','Signal Strength','Frequency']
x=df[features]
y=df['Signal Quality']
dt=DecisionTreeClassifier()
dt.fit(x,y)
tree.plot_tree(dt,feature_names=features)
plt.show()
```





In [49]:

```
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.cluster import KMeans
df=pd.read_csv("https://docs.google.com/spreadsheets/d/1844wXQMP4j3w_I0bBlSE6
x=df['Voltage']
y=df['Current']
d=list(zip(x,y))
kmeans=KMeans(n_clusters=2)
kmeans.fit(d)
plt.scatter(x,y,c=kmeans.labels_,cmap='coolwarm')
plt.xlabel('Voltage')
plt.ylabel('Current')
plt.title('Voltage vs Current')
plt.show()
```